

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи контролю дій користувачів у
Інтернет кафе”

КБГЗ - 2025

Виконав здобувач вищої освіти
IV курсу, групи КІ-21-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Лебедко О.В.
« ____ » _____ 2025 р.

Керівник проекту
доктор технічних наук, професор
_____ Коваленко О.В.
« ____ » _____ 2025 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 17 » січня 2025 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Лебедку Олександрю Вадимовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи контролю дій користувачів у Інтернет кафе*

2. Керівник роботи *Коваленко Олександр Володимирович, докт. техн. наук, професор*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 47-02 від 17.01.2025 року

3. Строк подання студентом роботи до захисту 23.05.2025 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи контролю дій користувачів у Інтернет кафе*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 17 » січня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2025 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2025 р.	
3.	Розробка моделі компонента	20.03.2025 р.	
4.	Розробка структур даних	25.03.2025 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2025 р.	
6.	Програмування алгоритмів	10.04.2025 р.	
7.	Оформлення ПЗ	17.04.2025 р.	
8.	Попередній захист роботи	23.05.2025 р.	

Дата видачі завдання
« 17 » січня 2025 р.

Підпис керівника

Коваленко О.В.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2025 р.

Підпис здобувача

Лебедко О.В.
(прізвище та ініціали)

АНОТАЦІЯ

Лебедко О.В. Програмне забезпечення системи контролю дій користувачів у Інтернет кафе. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи контролю дій користувачів у Інтернет кафе.

Метою розробки є програмне забезпечення системи контролю дій користувачів у Інтернет кафе.

Результат роботи – програмна реалізація системи контролю дій користувачів у Інтернет кафе.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, система контролю дій

ABSTRACT

Lebedko O.V. Software for the control system of user actions in an Internet cafe. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the control system of user actions in an Internet cafe.

The purpose of the development is the software for the control system of user actions in an Internet cafe.

The result of the work is the software implementation of the control system of user actions in an Internet cafe.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on a PC with OS Windows 10/11.

The program was developed in the Python environment.

Keywords: computer engineering, action control system

ВСТУП

Актуальність теми. Відеоспостереження являє собою систему, що складається з різних компонентів. Правильний підбір устаткування для системи відеоспостереження дозволить коректно побудувати охоронні системи відеоспостереження й забезпечити найбільш високий рівень безпеки офісу й житла.

Відеоспостереження в Кропивницькому й будь-якому іншому місті України просто життєво необхідно. Охоронне відеоспостереження дозволить установити особистість зловмисників, визначити їхнього наміру заздалегідь, до фактичного здійснення пограбування. Класичні способи охорони будинку – сталеві двері, ґрати й жалюзі-роллети на вікнах, надійні замки, сигналізація – всі вони здатні протистояти інтелектуальному злому лише часово. Лише охоронне відеоспостереження здатне запобігти проникненню – об'єкт із відеокамерами по периметрі свідомо небезпечний для злочинців.

Система відеоспостереження для будинку може містити в собі наступні види діяльності й компоненти:

- Проектування системи.
- Підбор відеокамер і їхній монтаж.
- Підбор записуючого пристрою.
- IP відеоспостереження; устаткування й програми.
- Підбор пристроїв передачі сигналу.
- Забезпечення постійного живлення.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи контролю дій користувачів у Інтернет кафе.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем контролю дій користувачів у Інтернет кафе.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

- Дослідження системи контролю дій користувачів у Інтернет кафе.
- Програмна реалізація системи контролю дій користувачів у Інтернет кафе.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі контролю дій користувачів у Інтернет кафе.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи контролю дій користувачів у Інтернет кафе, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ – 2025

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Якщо потрібна система відеоспостереження, купити необхідно й DVR відеореєстратор, що дозволяє переглядати відео в реальному часі на моніторі або при підключенні до мережі, а також дивитися архівні записи з реєстратора або віддаленого пристрою, записувати на віддалений сервер, управляти камерами Speed Dome і камерами, що мають трансфокатор.

Відеокамера спостереження:

- IP-відеокамера або аналогова.
- Кольорова або чорно-біла.
- З об'єктивом або без нього.
- Внутрішня або зовнішня.
- Купольна або проста.
- З інфрачервоним підсвічуванням або без її.

Пристрою відеозапису:

- DVR відеореєстратор.
- Плата відеозахвата.

А також додаткові пристрої:

- Монітор.
- Об'єktiv для відеокамер.
- Пристрою передачі відеосигналу.
- Гермокожухи й кронштейни.
- ІЧ-прожектора.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1.2 Область застосування

Централізоване керування відеокамерами може виконуватися за допомогою двох інструментів:

- фірмового ПЗ, що супроводжує камери відеоспостереження;
- ПЗ незалежних розроблювачів.

Перший варіант являє собою безкоштовний додаток до обладнання, що поставляється. Розробки виготовлювачів не дозволяють інтеграцію в мережу відеоприсрої сторонніх виробників. У другому випадку практично завжди варто враховувати додаткові витрати. Однак програмний продукт незалежних розроблювачів характеризується більшою функціональністю.

Тверда прив'язка програм до власних камер відеоспостереження викликана технологічними особливостями розвитку IPССТV: немає уніфікованих стандартів для IP-камер і їхнього функціонування. Тому виробники прагнуть забезпечити підтримку фірмових стандартів устаткування, уникаючи подорожчання свого кінцевого продукту.

Інакше кажучи, додаткові інвестиції в розробку програм для ір-відеоспостереження приводять до природного підвищення вартості встаткування. Це є вкрай небажаним рішенням, тому що ринок постійно поповнюється новими недорогими IP-камерами китайських виробників (Китай, Тайвань).

Широкі функціональні можливості ПЗ незалежних розроблювачів пояснюються тим, що всі їх інтелектуальні й фінансові ресурси направляються на створення максимально універсальних, функціональних і гнучких програмних продуктів.

Безкоштовні версії програмних продуктів або умовно безкоштовні застосовуються маркетологами ІТ-компаній як рекламні мети для залучення корпоративних клієнтів.

Ціль очевидна: протестувавши пробний зразок, його комфортність і функціональність, споживач купить повну версію за гарні гроші.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Безкоштовне ПЗ, по суті, є урізаною варіацією основної програми. Обмеження стосуються кількості інтегрованих камер, періоду використання продукту (до 30 днів), глибини архівування відеофайлів. Також тестові системи обмежують (у деяких випадках) подальшу модернізацію й нарощування.

Перед установкою програм і налаштування відеоспостереження необхідно купити камеру. Вибір камери відеоспостереження онлайн – важливий крок у налаштуванні відеоспостереження.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи контролю дій користувачів у Інтернет кафе, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ – 2025

					VKPB-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Програмні продукти централізованого відеоспостереження розробляються для різних платформ – MacOS, Linux, Windows.

Давайте ж докладніше розглянемо програми різних виробників і розберемося, де скачати платну й безкоштовну версію програми для відеоспостереження.

XProject Go

Один зі світових лідерів IT-сегмента компанія Milestone (Данія) також має у своєму розпорядженні безкоштовну версію свого провідного продукту XProject Go. До можливостей даної версії ставиться підтримка максимум восьми IP-камер і одного сервера, зберігання архіву протягом п'яти днів, avi-формат для відеозапису.

Платні версії і їхні можливості пропонуються по наростаючі: перший платний продукт цієї програми забезпечує підтримку 26 камер, необмежений період зберігання інформації, формати jpg, avi, віддалене керування через веб-клієнта, 5 користувачів.

Самий продуктивний варіант (XProject Corporate) працює з необмеженою кількістю камер і користувачів, забезпечує нелімітований період зберігання відеоархіву. Також ця версія відрізняється розвинутою аналітичною функціональністю, можливістю інтегрувати ПЗ і встаткування інших виробників. Детектор руху в цій системі функціонує незалежно від відеокамери.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

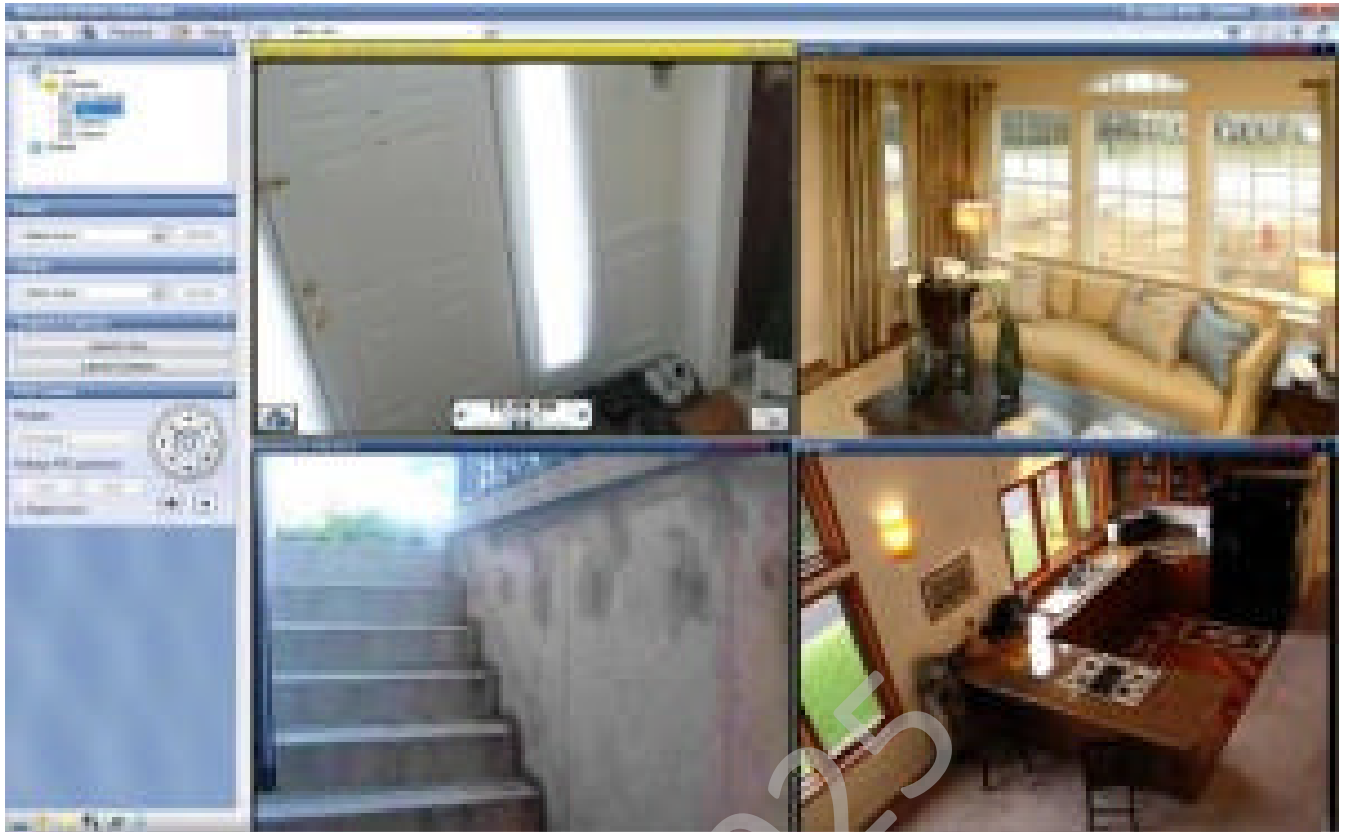


Рисунок 2.1 – Інтерфейс користувача XProject Go

Сервером може служити звичайний комп'ютер (ОЗУ – 1 або 2 ГБ, процесор 2,4 ГГц). Процедура реєстрації на сайті обов'язкова тільки при застосуванні ПЗ більше місяця. Продукт працює на платформі різних версій Windows.

Ben Software Security Spy

Для комп'ютерів Mac OS розроблений Ben Software Security Spy. Тестовий продукт доступний для використання протягом місяця.

Камери підтримуються аналогові, USB, IP. Система передбачає нелімітовану підтримку серверів і обсягів архіву (кодеки MPEG4, MJPEG).

Великий набір функцій включає:

- 10-секундний буфер запису. Приміром, після активації детектора руху зберігається запис самої події й того, що йому передувало);
- оповіщення по електронній пошті відповідальних осіб;

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

- убудований веб-сервер;
- автоматичне завантаження даних на заданий FTP-сервер.

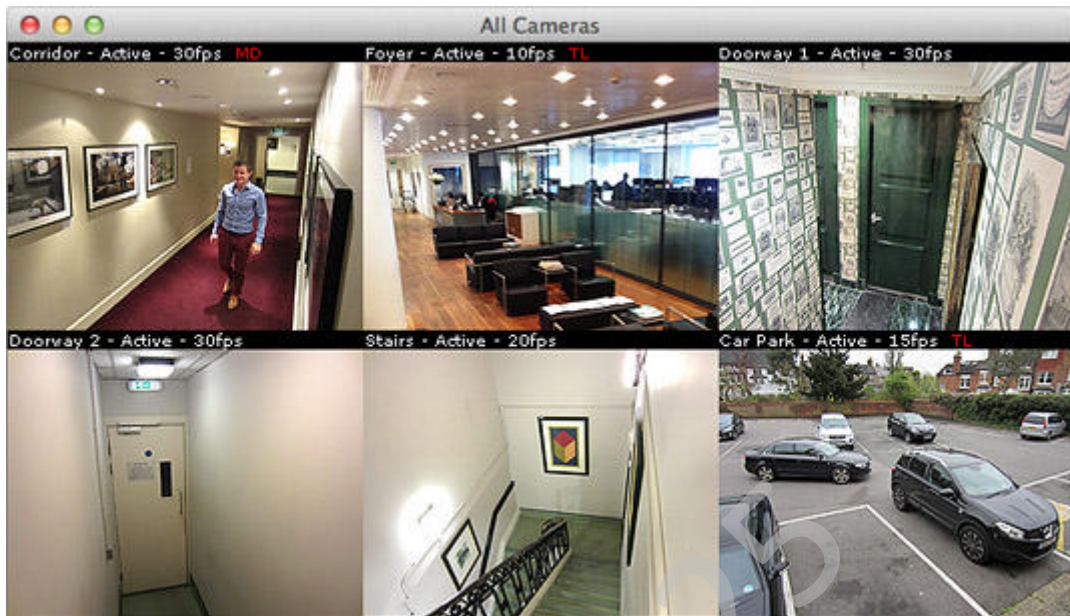


Рисунок 2.2 – Інтерфейс користувача Ben Software Security Spy

Zone Minder

Для операційної системи Linux розроблена безкоштовна програма Zone Minder. Проект існує завдяки добровільним пожертвуванням, тому ПЗ надається безкоштовно. Можливості програми обмежуються тільки потужністю сервера або ПК, що обробляє й зберігає відеопотоки.

Тому немає обмежень по кількості камер, розмірів архіву або користувачів. Система підтримує аналогові камери, IP-, USB-камери самих різних виробників, є детектор руху з автоматичним запуском режиму запису й повідомлення по SMS або електронній пошті.

Zone Minder як вільна платформа дозволяє інтеграцію програмних продуктів інших виробників. Дана програма використовується не тільки приватними клієнтами, невеликими компаніями, але й великими організаціями.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

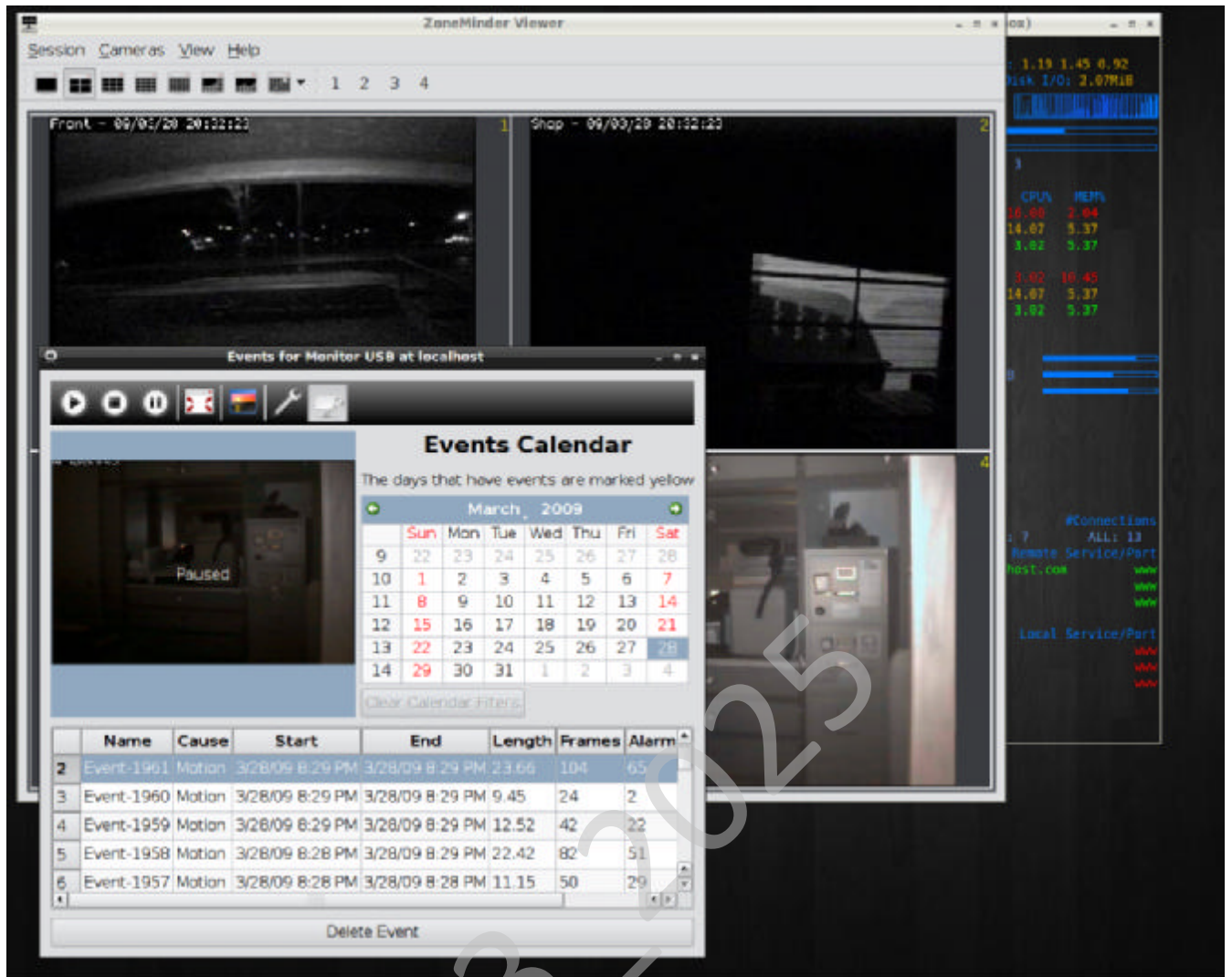


Рисунок 2.3 – Інтерфейс користувача Zone Minder

Smart IP

Варто відзначити умовно безкоштовну версію Smart IP, що в обмеженому варіанті називається Smart Start. Обмеження в урізаній версії поширюються на кількість камер (не більше 16), на обсяг відеоархіву (до 1 ТБ), кількість серверів (не більше 1). При цьому не обмежується число користувачів. У всім іншому продукт має всі функції основної програми. До цікавих функцій умовно безкоштовного варіанта ставиться можливість пошуку в архіві. Він здійснюється по швидкості й напрямку руху об'єкта, по розмірі, кольору. Крім того, пошук об'єкта, що рухається, можна виконувати в певній області. Шукати можна також зникнення або поява об'єктів, перетинання умовної лінії й т.д.



Рисунок 2.4 – Інтерфейс користувача Smart IP

Нарощування Smart Start виконується за допомогою установки відновлення набору драйверів – без переустановки всієї системи.

Безкоштовні програми для відеоспостереження

Для рішення простих завдань відеоспостереження (квартира, периметр перед будинком, під'їзд) існує безліч програм, доступних для завантаження в інтернеті.

Кодос-відеомережа – ПЗ для безкоштовного завантаження. Продукт передбачає підключення максимум чотирьох відеокамер різних виробників (аналогових за допомогою пристрою відеозахвата, IP, веб камери). Для IP-камер є спеціальна програма Globoss, що дозволяє інтегрувати 12 камер.

При цьому програма реалізовує стандартний набір функцій: детектор руху й звуку, налаштування часу включення запису з відеокамер, керування доступом

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

для перегляду відеофайлів. Також ПЗ характеризується досить комфортним користувальницьким інтерфейсом.

Abel Cam – програма для вільного поширення в мережі. Вона дозволяє здійснювати запис із веб-камери й редагування відеофайлу. Продукт взаємодіє з камерами будь-якого типу (веб-камери, бездротові) і одночасно з декількома джерелами. У програмі є функція веб-сервера з підтримкою SSI, PHP, CGI, проху, Ajax. Налаштування системи дозволяють включати режим запису з появою об'єкта, що рухається, позначати мертві зони.

AV100 – утиліта Arecont Vision призначається для систем, що володіють високим дозволом. Вона здійснює контроль над 16 відеокамерами, робить запис HDTV-Якості при швидкості 240 к/с. Утиліта надається безкоштовно.

GOTCHA Multicam – універсальна програма, проста й доступна, що дозволяє інтеграцію декількох відеокамер. Налаштування активації камер залежать від бажання користувача: по початку руху, по припиненню руху. Ця безкоштовна програма має всю функціональність професійного ПЗ.

Відеодані програма записує у форматі mv2. Їх можна відобразити за допомогою убудованого плеєра. Детектор руху налаштовується для гнучкої обробки тривожних подій (запис, взаємодія з іншим охоронним пристроєм).

Всі перераховані програми допускають як використання професійного встаткування, так і налаштування відеоспостереження через веб камеру. Так само, всі розглянуті програми можуть бути скачани безкоштовно.

Функціональність програм визначається їхніми версіями. Тому рекомендується самостійно протестувати найбільш підходящі для заданих умов варіанти (тип і кількість камер, вимоги до ПК або сервера, конкретні завдання для відеоспостереження). Всі програми й утиліти (якщо не зазначено інакше) призначені для роботи на платформі операційної системи Windows і її версій.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13



Рисунок 2.5 – Інтерфейс користувача GOTCHA Multicam

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це об'єктноорієнтована мова програмування високого рівня загального призначення з відкритим кодом. Це визначення може бути важким для новачків, тому розглянемо кожну характеристику окремо, щоб зрозуміти, що вона означає:

- Відкритий вихідний код: це безкоштовно та доступно для подальших покращень, таких як додавання корисних функцій або виправлення помилок.
- Об'єктноорієнтована: заснована не на функціях, але в об'єктах з певними атрибутами й методами.
- Високий рівень: зручний для людини, а не для комп'ютера.
- Загальне призначення: можна використовувати для створення будь-яких програм.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веброзробці, розробці програмного забезпечення та інших галузях.

Переваги та недоліки Python

Переваги:

– Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам. Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java.

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв'язанні проблеми. Їм не

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

потрібно витратити багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

– Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження.

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек.

– Портативність. У багатьох мовах, таких як C/C++, потрібно змінити свій код, щоб запустити програму на різних платформах. З Python все інакше. Ви тільки пишете один раз і запускаєте її будь-де.

Недоліки:

– Низька швидкість. Вище ми обговорювали, що це інтерпретована мова з динамічною типізацією. Порядкове виконання коду часто призводить до повільного виконання. Динамічна природа Python також є причиною її низької швидкості, оскільки їй доводиться виконувати додаткову роботу при виконанні коду. Тому вона не підходить для цілей, де швидкість важливий аспект проєкту.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

– Неєфективна для пам'яті. Ця мова програмування використовує великий обсяг пам'яті, це може бути недоліком при створенні програм, коли віддають перевагу оптимізації пам'яті.

– Слабка у мобільних обчисленнях. Python зазвичай використовується у серверному програмуванні. Ми не бачимо – її на стороні клієнта або в мобільних програмах з таких причин: вона не заощаджує пам'ять і має повільну обчислювальну потужність у порівнянні з іншими мовами.

– Доступ до бази даних. Програмувати на цій мові легко, але коли ми взаємодіємо з базою даних, її не вистачає. Рівень доступу до бази даних у Python примітивний та недостатньо розвинений у порівнянні з іншими популярними технологіями.

– Помилки виконання. Це мова з динамічною типізацією, тому тип даних змінної може змінюватись у будь-який час. Змінна, що містить ціле число, у майбутньому може містити рядок, що може призвести до помилок виконання.

Застосування Python:

– Для аналізу даних. Дані стали цінним активом у будь-якій сучасній галузі, і більшість компаній зацікавлені у збиранні, обробці та аналізі релевантних даних, щоб витягти з них цінну інформацію для бізнесу. І тут Python виходить за межі будь-якої конкуренції. Python особливо цінна тим, що крім великої стандартної бібліотеки надає величезний набір додаткових модулів, розроблених спеціально для аналітичних цілей. Найвідоміші бібліотеки Python для аналізу даних – це pandas і NumPy . Ці інструменти дозволяють робити з вашими даними майже все, наприклад, очищати і аналізувати їх, вивчати статистику або візуалізувати приховані тенденції у ваших даних.

– Для візуалізації даних. Візуалізація даних – це окрема частина аналізу даних, яка допомагає нам подавати інформацію, необроблену чи очищену, у більш змістовній формі. Тут Python знову входить у гру, пропонуючи широкий спектр інструментів візуалізації даних. Найпопулярніші з них – matplotlib і

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

перетворювати, розділяти, об'єднувати або видаляти файли, перевіряти їх наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

- Фахівець з даних.
- Аналітик даних.
- Інженер даних.
- Інженер з машинного навчання.
- Журналіст даних.
- Архітектор даних.
- Повний стек веброзробника.
- Backend-розробник.
- DevOps-інженер.
- Інженер-програміст

Можемо зробити висновок, що Python ще довго буде популярною мовою, хоч і має низку недоліків. Цю мову використовують для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу, аналізу даних, машинного навчання, інженерії даних та для багатьох інших областей. Це перспективна і затребувана навичка, яка необхідна у всіх галузях.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи контролю дій користувачів у Інтернет кафе.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

- а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;
- б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;
- в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;
- г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;
- д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;
- е) провести розрахунки по визначенню економічної ефективності розробленої системи;
- ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;
- з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

У роботі необхідно вести відеоспостереження у рамках розробляємої системи контролю дій користувачів у Інтернет кафе. Налаштувати систему можна всього за кілька простих кроків.

1. Завантажите зі сторінки завантаження.

2. Запустите на комп'ютері, мобільному пристрої або планшеті, що стане сервером – центром системи відеоспостереження. Тут будуть оброблятися камери й інші джерела, записуватися матеріали в архів і т.п. Якщо ще не активована, у тому числі й при першому запуску, вона запуситься в пробному режимі. Можливо перемикається між безкоштовним, пробним і комерційним режимами через Головне меню.

3. При першому запуску програма для відеоспостереження сканує мережа в пошуках доступних камер. Всі знайдені камери будуть додані на Головне вікно, за винятком камер, що вимагають авторизацію – такі камери будуть перераховані в списку в меню "+" у нижній панелі. Можливо додати їх, вибравши їх зі списку й увівши необхідні дані для доступу. Можливо продовжувати роботу під час пошуку. Однак, якщо пошук займає занадто багато часу або якщо всі потрібні камери вже були знайдені й додані, пошук можна зупинити, нажавши на шкалу прогресу пошуку.

4. Знайдені камери додаються зі схемою за замовчуванням: Універсальна камера – Розклад – Детектор руху – Архів і перегляд, з налаштуваннями за замовчуванням, що підходять для більшості випадків. Можливо додати, видалити або переналаштувати модулі так, як потрібно.

– Камера: можна вибрати дозвіл і частоту відновлення картинки (для USB камер) або потік із уже заданими параметрами (для мережних камер). Аналогові

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

камери визначаються або як USB, або як IP камери залежно від того, як вони підключені.

– Детектор руху: виділите зону або кілька зон, у яких буде відслідковуватися рух, задайте перед- і постзапис і чутливість датчика.

– Розклад: задайте час роботи наступних за Розкладом модулів. За замовчуванням настроєний працювати цілодобово.

– Перегляд і Архів (або просто Архів): Можливо вказати, куди зберігати відео, час зберігання (запису з більше старою датою будуть віддалені), максимальний обсяг, що архіву дозволено займати на диску. За замовчуванням час зберігання – 1 година.

5. Встановите, щоб програма запускаласть автоматично зі стартом операційної системи.

6. Щоб переглянути записи в архіві, натисніть на маленьку іконку "Play" у правому верхньому куті зображення з камери на головному вікні або на більшу іконку "Play" у нижній панелі. Відкриється убудований програвач архіву, де Ви зможете переглянути й експортувати запису.

Хочете робити знімки того, що відбувається на екрані комп'ютера? Скористайтесь модулем "Захват екрана".

Потрібно, щоб програма для відеоспостереження працювала непомітно для інших користувачів? Відзначте галочкою опцію "Схований режим" у меню Установки.

Потрібний віддалений доступ для перегляду камер і їхніх записів з будь-якої крапки миру? Це також можливо. Для цього використовуйте дані з Діалогу підключення.

Якщо віддалений доступ не потрібний, програма може працювати без виходу в Інтернет.

Можливо додавати облікові записи оператора й давати персоналу доступ до перегляду й керування камерами, або використовувати безліч інших корисних можливостей.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Головне вікно системи безпеки складається із шапки вікна (1), так званого прев'ю – осередків перегляду (2) (зображення в режимі реального часу з камер або інших джерел відео) і висувної нижньої панелі (3).

У верхній панелі вікна програма розташовані стандартні кнопки "Згорнути", "Розгорнути", "Закрити". Пересувати вікно й змінювати його розмір можна за допомогою миші.

У заголовку вікна (1) відображається інформація про IP-Адресу сервера (при віддаленому підключенні), версії (4) і режимі програми (5), а також про назву вікна, у якому Ви зараз перебуваєте (6). У правому верхньому куті кожного осередку перегляду (2) є маленькі іконки архіву (7) і налаштувань (8) – натисканням на них відкривається вікно перегляду архіву цієї камери або її детальних налаштувань. Якщо у Вас активована Pro-Ліцензія, починаючи з версії 15.10.21, архів відкриється прямо на головному вікні, паралельно Ви зможете переглядати інші камери в режимі реального часу.

Якщо камера підтримує PTZ-керування (функції " Поворот-Нахил-масштабування"), поруч буде також іконка PTZ для виклику PTZ-контроля. Якщо настроєно звук для камери, то буде показуватися також і маленька іконка "Динамік" (зелений колір іконки – звук включений, білий колір іконки – звук для Головного вікна виключений). На Головному вікні може бути включений звук максимум з 1 камери.

Під цими маленькими іконками Ви побачите час останнього виявленого руху (9) (якщо в ланцюжку цієї камери (відеоджерела) є й включений модуль "Детектор руху"). При натисканні на цей час буде відкритий архів камери на моменті початку цього руху. До того ж, при виявленні руху рамка осередку перегляду (10) стане червоною, поступово (протягом 3 хвилин) вертаючись до звичайного білого кольору (при наведенні курсору – блакитному). якщо виявлений рух усе ще триває, замість часу буде значок " Res".

У лівому верхньому куті системи безпеки прев'ю видно назву цієї камери (11) – воно задається в налаштуваннях модулів "Перегляд і Архів" або

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

"Перегляд". Кликніть на назву тут, і налаштування модуля будуть відразу відкриті.

Клацання кнопкою мишки в будь-якому місці усередині осередку перегляду (12) відкриє вікно Швидких налаштувань, де в лівій висувній панелі Можливо побачити список підключених модулів і швидко його редагувати.

Нижня панель, що висувається (3) містить наступні іконки: "Список", "Плюс", "Гайка", "Play", "Вікно" і "Інфо". Щоб сховати/показати панель, потягніть неї долілиць/нагору або просто кликніть по ній мишкою.

– "Список" – іконка головного меню: при натисканні відкриває **Головне меню**, у якому можна:

– налаштувати віддалений доступ (включаючи Ретранслятор і облікові записи користувачів/операторів);

– виключити/включити "Дерево камер";

– виключити/включити всі джерела;

– установити програму в автозавантаження або деінсталювати програму;

– знайти навчальне відео;

– перемкнутися між безкоштовним; пробним або комерційним режимами роботи програми;

– купити ліцензію й активувати комерційну версію;

– знайти інформацію про програму (у меню "Інформація");

– залишити відкликання;

– перейти на офіційний сайт розроблювача;

– перемкнути мова інтерфейсу.

– Іконка "Плюс" використовується для додавання нової камери. Якщо в мережі виявлені камери, додати їх можна прямо із цього меню. Якщо камер не виявлено, можна додати випадкову камеру миру, дублювати останню схему, повторити пошук або повторити пошук із заданими логіном, паролем, підмережею або портом, які може використовувати шукана камера.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Також у цьому меню Можливо дублювати останню переглянута камеру, що буде особливо зручно, коли потрібно скопіювати складну настроєну схему системи безпеки й застосувати її для наступних камер з майже ідентичними налаштуваннями.

– **"Гайка"** аналогічна іконці налаштувань у верхньому правому куті прев'ю – вона використовується для доступу до детальних налаштувань останньої переглянутої камери (джерела) і найбільш точного й наочного налаштування всіх можливостей.

– **"Play"** – або "Архів" – аналогічна іконці виклику архіву у верхньому правому куті осередку перегляду: діє так само, якщо активовані звичайні комерційні ліцензії; при наявності активованих Pro-Ліцензій діє трохи інакше: з версії 15.10.21 вона дозволяє вибрати, які архіви хочете відкрити в строєному переглядачі, де можна переглянути всі записи цієї камери й експортувати фрагменти в обраному форматі.

– **"Вікно"** – меню Перегляду – пропонує:

– вибрати варіанти перегляду прев'ю всіх камер, у тому числі режим збільшений перегляд однієї або декількох камер. Якщо був обраний варіант, при якому показується лише частина камер, ліворуч і/або праворуч з'явиться стрілка «Попередні камери» і «Наступні камери» для перемикання між вікнами з декількома джерелами. Також тут можна вибрати опцію "Сховати/показати панель сторінок", що дозволяє створити вгорі програми вкладки, яким можна дати назва по якій-небудь ознаці, і групувати камери, перетаскуючи їх у потрібну групу. Кількість камер у групах може бути нерівним. Якщо хочете закрити якусь конкретну вкладку, а не всі відразу, то пройдіть у Режим перегляду -> Керування переглядом -> Перегляд: що набудовується -> Видалити поточну вкладку;

– форсовано змінити швидкість відновлення картинки клієнтської частини (це значення ніяк не відіб'ється на серверній частині) і розмір шрифту для назв камер і системних повідомлень (меню "Налаштування FPS і шрифту");

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

- включити/виключити показ такої інформації про камери, як її швидкість відновлення картинки, формат потоку, строк зберігання архіву, на прев'ю працюючих камер, а також загальну інформацію про обліковий запис – навантаження програми на процесор і пам'ять;
- включити/виключити візуалізацію рухів;
- включити/виключити перекреслення непрацюючих камер (тобто поверх не камер, що обновляються, буде показуватися червоний хрест, що зручно при маленькому розмірі прев'ю-віконець на головному екрані);
- відрегулювати гучність;
- включити/виключити режим поверх всіх вікон і задати напівпрозорість;
- редагувати білий список для "Розпізнавача номерів" (тільки при наявності PRO ліцензії);
- відкрити, завантажити й використовувати eMap – інтерактивну карту об'єкта під спостереженням.
- Натискання на іконку **"Інфо"** відкриває екран з підказкою про призначення іконок панелі й пропозицією скористатися розділом "Довідка".

Інтерфейс вікна швидких налаштувань

Інтерфейс вікна швидких налаштувань містить панель, що висувається (**"Ліву панель"**) зі списком використовуваних для даної камери можливостей програми, віконця напроти яких дозволяють оперативно включити або виключити їх. Натисніть на назву модуля, щоб відкрити його налаштування. Для перемикання на наступну/попередню камеру (ланцюжок) використовуйте стрелочки.

Нижня панель швидких налаштувань крім іконок, описаних вище, містить також **"Стрілочку"**, **"Червоний хрест"**, **"Поворот"** і **"Гучність"**.

"Стрілочка" використовується для повернення в попереднє вікно, у цьому випадку – у режим перегляду (головне вікно).

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

(7) – Великий хрест – іконка скасування. Нажавши на неї, Ви вийдете з налаштувань модуля без збереження останніх змін.

(8) – Більша галочка – іконка "ОК". Натисніть на неї, щоб зберегти останні зміни налаштувань і закрити діалог.

(9) – **Групові налаштування** (де присутні) – опція дозволить застосувати якісь налаштування до усім або частини модулів такого ж типу (наприклад, до усім "Детекторам Руху"). Це зручно, коли потрібно швидко й однаково налаштувати безліч модулів одного типу.

Зміна порядку камер на головному вікні

Можливо змінити порядок розташування камер на головному вікні, просто перетягнувши зображення з камери відеоспостереження на потрібне місце (тобто обміняти осередку перегляду місцями).

Керування поворотними камерами й зуммування

У можна управляти PTZ-Функціями камери (поворот, нахил, масштабування) або, якщо камера не підтримує цю можливість, користуватися цифровим зуммуванням.

1. Керування PTZ. Якщо PTZ-камера була успішно визначена в , а також розпізнані команди для керування її PTZ-Функціями, у правому верхньому куті прев'ю камери в Головному вікні або вікні Швидких налаштувань з'явиться іконка PTZ контролю. При натисканні на іконку буде відкритий PTZ-контроль. Натисніть кнопки стрілок для повороту у відповідну сторону. Натисніть на кнопки + або – для наближення/віддалення відповідно.

Якщо камера підтримує PTZ і контроль керування цими функціями з'являється, але натискання на стрілки або +/- не приносить результату, відкрийте налаштування модуля "Універсальна камера" цієї камери й спуститесь до секції налаштування PTZ. Спробуйте інші опції з осередку "Доступні PTZ". Якщо використовується нестандартний порт для PTZ, укажіть його в наданій для цього осередку. Якщо це не допомагає, спробуйте використовувати інший URL для одержання зображення з камери.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

2. Цифровий зум. Цифровий зум – це метод електронного наближення об'єктів без використання лінзи. Якщо картинка з камери досить великого дозволу й потрібно наблизити віддалений об'єкт, включите зуммування в Головному меню або меню вікна Швидких налаштувань і можете наближати/віддаляти об'єкта на зображенні по кличі або колесом миші (прокручування нагору – наближення, долілиць – віддалення).

3. Керування PTZ-Функціями камери в браузері.

Підвищити ефективність оперативного відеоспостереження допоможуть уже убудовані в програму детектор руху й шуму. Як тільки спрацює один з детекторів, програма покаже відбувається на eMap, Ви побачите червоний значок датчика й з легкістю зможете визначити, у який з кімнат відбувся рух або шумовий поріг перевищив зазначений Вами ліміт.

3.2 Розробка структурної схеми

Можливості програми для відеоспостереження:

Робота з камерами/джерелами:

- Підтримка всіх видів веб-камер.
 - Підтримка сотень брендів і моделей IP-камер (включаючи бездротові WiFi, ONVIF, H.264, JPEG/MJPEG, MPEG-4, Fisheye і PTZ камери).
 - Необмежений дозвіл картинки (кількість мегапікселів).
 - Можливість переглядати світові Інтернет-камери.
 - Можливість робити знімки екрана (скріншоти) із всіх моніторів одночасно, за допомогою програми для відеоспостереження.
 - Вибір якості звуку й гучності.
- Професійне відеоспостереження для всіх:
- Швидкий перехід між камерами й керування групами за допомогою "Дерева камер".
 - Групові налаштування.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

- Хмарний сервіс Cloud.
 - Професійне розширення функцій за допомогою Pro.
 - Можливість налаштування камер, модулів і ланцюжків через файл конфігурації в Pro.
 - Об'єднання серверів у мережу.
 - Мульти-серверне й мульти-клієнтське підключення для відеоспостереження будь-якого масштабу.
 - Облікові записи (профілі) користувачів із гнучким налаштуванням прав доступу.
 - Захист налаштувань програми для відеоспостереження паролем.
 - Повноекранний перегляд.
 - Можливість використовувати кілька моніторів (стільки, скільки витримає сам ваш комп'ютер).
 - Ручне керування PTZ камерами.
 - Поворот зображення на 0-360°.
 - Детектор звуку.
 - Швидке збереження знімка картинки з камери одним натисканням.
- Інтелектуальні модулі й можливості:
- Корекція зображення камер Fisheye ("риб'яче око").
 - eMap – інтерактивна карта об'єкта з розставленими на ній камерами.
 - Синхронізація з SD картами пам'яті камер.
 - Контроль касових операцій.
 - Синхронізація з активними директоріями LDAP.
 - Проста інтеграція із системами автоматизації завдяки модулям "Відправлення HTTP запити" і "HTTP перемикач".
 - Можливість створити свій власний хмарний сервіс відеоспостереження.
 - Автоматичне маскування зон (размиття об'єктів приватної власності, які заборонені знімати).

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

- Автоматичне спостереження за об'єктами, що рухаються, із зуммуванням (PTZ спостереження) і їхня візуалізація.
- Детектор проблем для моніторингу "здоров'я" системи.
- Керування PTZ, у тому числі при перегляді через браузер.
- Швидке включення/вимикання ланцюжків сторонніми програмами ("HTTP перемикач") і через інтерфейс програми для відеоспостереження ("Швидке включення/вимикання").
- Пошук в архіві виявлених рухів по даті й в обраній зоні.
- RTSP і HTTP трансляція – емуляція IP камер.

Віддалений доступ:

- Віддалений перегляд за допомогою мобільних пристроїв (підтримка Android, iPhone, iPad).
- Online перегляд всіх камер зі звуком.
- Повний віддалений доступ до налаштувань, архіву й камерам, через інтерфейс програми для відеоспостереження.
- Веб сервер і трансляція в Інтернет.
- Вбудовування камер на Ваш сайт.
- Віддалений доступ навіть при динамічному IP адресі.
- Перегляд архівів камер через веб браузер.

Детектор руху:

- Можливість задати зони детектування будь-якої форми й розміру.
- Візуалізація виявленого руху із вказівкою часу останнього поміченого руху.
- Можливість указати максимальний розмір ігнорованих об'єктів, поріг чутливості датчика руху.
- Передзапис дозволяє зберегти в архів кілька секунд, що передують початку події.
- Постзапис дозволяє протягом деякого часу продовжувати запис після закінчення події.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

- Затримка спрацьовування.
- Поліпшений алгоритм проти помилкових спрацьовувань (реакції на тварин, зміни погодних умов).

Архів:

- Можливість указати максимальний розмір займаного простору.
- Циклічний запис для економії місця.
- Регульований рівень стиску.
- Можливість запису архіву на різні диски (у т.ч. RAID-масиви).
- Можливість запису архіву на мережні диски (NAS і т.д).
- Архівні записи зі звуком і відеорядом.
- Видалення непотрібних записів прямо із програвача.
- Захист від видалення записів операторами.
- Зручний убудований програвач із пошуком по русі або по даті/часу.
- Експорт будь-якого відрізка в обраному форматі прямо із програвача.
- Одночасний перегляд декількох архівів.

Повідомлення й реакції:

- Відправлення SMS повідомлень на зазначений номер.
- Відправлення повідомлення на E-Mail (із прикріпленням відео або зображення, у т.ч. на кілька адрес).
- Автоматичне розгортання вікна програми.
- Звуковий сигнал (сирена, умовний звук).
- Автоматичний запуск зазначеної програми або скрипта.
- Повідомлення на панелі стану для Android.
- Завантаження зображень і відео зі звуком на FTP-сервер.
- Збереження матеріалу у файл в обраному форматі (з можливістю автоматичного перезапису).

Топ-можливості:

- Схований режим роботи програми для відеоспостереження.
- Вибір формату зберігаемого відео (H.264, MPEG-4, MP4, MJPEG).

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

- Маркування (накладення дати й часу або довільного тексту на зображення).
- Робота з розкладу.
- Регульована гучність звуку з камер у режимі перегляду.
- Швидке налаштування: копіювання настроєних ланцюжків.
- Різні варіанти розкладки камер.
- Автоматичне листання зображень із камер.
- Відновлення в 1 клацання миші.
- Автоматичні повідомлення про нові версії програми для відеоспостереження.

Переваги програми:

- Крос-платформенність: працює на Windows, Linux, Mac OS і мобільних платформах..
- Безкоштовне завантаження.
- Повнофункціональна пробна версія, що може бути використана нескінченно.
- Не обмежена за часом використання безкоштовна версія.
- Швидкий старт: готовий працювати відразу після завантаження з оптимальними налаштуваннями за замовчуванням.
- Не потрібна установка.
- Не потрібні права адміністратора.
- Простота наступних налаштувань.
- Принципово новий інтуїтивно-зрозумілий графічний інтерфейс.
- Якісна техпідтримка.
- Динамічний постійний розвиток і додавання сучасних можливостей.

Сфери застосування:

- Спостереження за приміщеннями в неробочий час (Інтернет кафе, офіси, магазини, склади).

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

– Охоронне спостереження (контроль і охорона периметра, безпека автомобіля).

– Інтеграція із системами "Розумний будинок" і системами автоматизації.

– Спостереження за працівниками (на виробництві, в офісах, у лікарні, будинку).

– Спостереження за діяльністю дітей і службовців за комп'ютером, контроль відвідуваних сайтів і ресурсів.

– Спостереження за домашніми вихованцями.

– Спостереження за автоматикою на виробництві.

Мінімальні системні вимоги:

– Для Windows – Windows XP (SP2 тільки x64, SP3 – x32/x64), Windows Vista, Windows 7, Windows 8, 64- і 32-бітні версії; Windows Server 2000/2003/2008/10/11.

– Для Linux – 32- і 64-бітні версії із графічною оболонкою XWindows або консоллю/сервером. Підтримувані дистрибутиви: .

– Ubuntu останніх версій (10, 11, 12, 13, 14).

– Debian 6 Server, Debian 7.

– openSUSE 12.1.

– Mint 12.

– CentOS версії 6 і вище..

– Red Hat Enterprise Linux версії 6 і вище.

У консольній версії Linux для підтримки звуку потрібен пакет libasound2.

Розпаковувати архів _linux.tgz потрібно саме на тій машині, де встановлений Linux, або виконати команду `chmod a+x .app`, після чого запустити як звичайно: `./app`

Якщо встановлюєте 32-бітну на Linux 64-біт (хоча рекомендуємо скачати 64-біт), може знадобитися встановити `ia 32-libs`.

– Для Mac версії 10.7 і вище.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

– Для Android – операційна система Android 2.3.3 (API LEVEL 10) і вище (рекомендується Android 4.2), 20 Мб оперативної пам'яті, дозвіл екрана – від 450x450 пікс.

– Для iPhone/iPad – iOS версії 5 і вище.

Режими

Дане програмне забезпечення може використовуватися в одному із чотирьох режимів: безкоштовному, пробному й комерційному.

– Безкоштовний режим – базовий – дозволяє використовувати до 8 джерел відео (камер) і до 3 модулів у кожній схемі необмежений час і управляти програмою з того комп'ютера, де вона встановлена.

– Lite режим містить самі необхідному й прості в освоєнні модулі. Дозволяє використовувати до 4 джерел відео, але не більше шести модулів у кожній схемі. Відновлення програми в Lite не підтримуються: використовуйте версію, на якій вироблялася активація, протягом необмеженого часу. Ліцензії продовження на Lite не потрібні.

– Програмне забезпечення в пробному режимі можна використовувати необмежену кількість модулів у будь-яких комбінаціях, а також контролювати систему віддалено. Задані налаштування скидаються на дефолтні через 4 години з моменту останньої зміни. Також задані налаштування скидаються при закритті програми. Зверніть увагу, що час зберігання Архіву в пробній версії становить максимум 1 година, по витіканню часу Архів буде перезаписаний.

– Стандартний режим доступний після покупки ліцензії на будь-яку кількість камер. Цей режим дозволяє використовувати будь-які модулі в будь-яких сполученнях. Доступна функція віддаленого контролю програми. Загальна кількість використовуваних відеоджерел/камер задається активованою ліцензією або декількома ліцензіями.

– Перемикання між режимами виробляється через Головне меню -> Інформація або Головне меню -> Інформація -> Про програму.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Налаштування призначень

Звуковий сигнал

Цей модуль дозволяє програвати на сервері заданий звук при надходженні зображення. В Pro звуковий сигнал може програватися й на клієнтах. Якщо хочете, щоб сигнал спрацьовував при виявленні руху, підключите його після детектора руху. Можна вибрати звуковий файл у форматі wav із запропонованих або вказати шлях до файлу вручну, інтервал програвання файлу (звук буде відтворюватися з кожним приходячим кадром, але не частіше зазначеного інтервалу) і тривалість програвання. Натисніть «Протестувати», щоб перевірити правильність зазначених налаштувань.

Запуск додатка

Цей модуль дозволяє запускати обраний додаток. Якщо хочете, щоб сигнал спрацьовував при виявленні руху, підключите його після детектора руху. Укажіть шлях до програми, що запускається, з використанням макросів і інтервал запуску (програма буде запускатися з кожним приходячим кадром, але не частіше зазначеного інтервалу). Натисніть «Протестувати», щоб перевірити правильність зазначених налаштувань.

Спливаюче вікно (на клієнті)

Цей модуль дозволяє автоматично розвертати програму зі згорнутого стану при надходженні зображення.

Відправлення sms

Цей модуль дозволяє організувати автоматичне відправлення sms-повідомлень на телефон. Можна встановити інтервал відправлення повідомлень, а також задати макроси, які уточняють, яка камера й у скільки виявила рух.

Відправлення email

Цей модуль дозволяє організувати автоматичне відправлення листів на електронну пошту. Можна встановити інтервал відправлення email і прикріплювати вступником з камери відео й зображення (максимальна кількість зображень, що прикріплюються, інтервал між зображеннями, частоту кадрів і

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

тривалість прикріплюється відео й максимальний розмір повідомлень). Також використовуйте макроси для роботи з детекторами руху й проблем.

Максимальний розмір повідомлення дозволить одержувати повідомлення, не перевищуючі певний розмір (наприклад, 2 Mb). Максимальна кількість зображень, що прикріплюються, розраховується виходячи із заданого максимального розміру повідомлення й розміру кожного зображення з даної камери (залежно від дозволу й розміру картинки, які в більшості випадків задаються в налаштуваннях камери в її адмін-сторінці).

Виберіть Ваш поштовий сервер зі списку або вкажіть свій. Якщо Ваш поштовий сервер не зазначений у списку, Можливо вказати параметри вручну. Уведіть адресу SMTP-сервера й порт, а також укажіть спосіб шифрування передачі даних. Уведіть логін і пароль від електронної пошти. У рядку «Адреса одержувача» укажіть той email, на який буде вироблятися відправлення. Укажіть тему й текст листа (опціонально). Після налаштування модуля Можливо протестувати його роботу, нажавши «Протестувати».

Якщо Ви підключаєте модуль відправлення email після детектора руху, залежно від виставленого інтервалу й кількості картинок, що прикріплюються, програма буде перевіряти наявність руху через певні проміжки часу й зберігати зображення для наступного відправлення. Наприклад, при інтервалі в 1 хвилину й 2 зображеннях, що прикріплюються, кожних 30 секунд буде перевірятися, є чи в заданому секторі рух. Якщо рух є, що відповідає картинка буде збережена. Через наступних 30 секунд програма знову перевірить детектор на наявність руху й збереже зображення, якщо рух є, відразу після чого буде сформоване й відправлений лист. Якщо руху в момент перевірки ні, картинка збережена не буде. Якщо після закінчення часу інтервалу між повідомленнями картинок не буде, лист не буде відправлено.

Увага: якщо хочете переглядати відео, одержувані на зазначений email, на мобільному пристрої, рекомендуємо використовувати формат MP4, оскільки він підтримується більшістю ОС мобільних пристроїв.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Натисніть «Протестувати», щоб перевірити правильність зазначених налаштувань.

Відправлення НТТР запиту

Цей модуль дозволяє формувати й відсилати НТТР запити. Якщо хочете, щоб запит відсилався при виявленні руху, підключайте модуль після детектора руху. Установіть прапорець "Показати параметри" для завдання параметрів запиту. Використовуйте **макрос {ID}** у значеннях параметрів для підстановки ідентифікатора модуля. Можливо побачити результуючий запит у поле "Кінцевий URL".

НТТР Відправлення на іншу

Цей модуль дозволяє передавати відео й аудіо по мережі іншим копіям . Укажіть номер порту й шлях на віддаленому сервері , на якому буде здійснюватися передача відео й аудіо інформації.

Для обмеження доступу на віддаленому сервері, задайте логін і пароль для доступу до нього. У відповідному полі Ви побачите адресу, по якому буде здійснюватися доступ до відео або зображення.

RTSP Транслятор

Цей модуль дозволяє передавати дані по мережі, використовуючи rtsp-трансляцію. Для трансляції в libx264 потрібно використовувати додаткове ПЗ (ffmpeg). Можливо скачати його з нашого сайту або використовувати свій. Завантажуючи ffmpeg з нашого сайту, Ви приймаєте умови ліцензійної угоди GNU GPLv2+. Якщо ви транслюєте в mjpeg або mpeg4, де використовується убудований ffmpeg, то завантажувати додаткове ПЗ не потрібно. Ви також можете вибрати ступінь стиску відеопотоку.

Перегляд

Цей модуль дозволяє переглядати поточне зображення в режимі реального часу. У цьому модулі можна вказати ім'я, що буде відображатися на зображенні даного Перегляду. Модуль підсвічується кольором у колірному інтервалі від червоного до білого, якщо протягом останніх 5 хвилин був виявлений рух (чим

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

колір світліше, тим більше часу пройшло з моменту виявлення руху). При цьому на зображенні з'являється час останнього виявлення руху. При наведенні миші на таке зображення – рамка здобуває рожевий колір. Якщо рух не був зафіксований протягом останніх 5 хвилин – рамка біла, при наведенні – блакитна.

Детектор дня

Цей модуль автоматично відсіває кадри при недостатнім освітленні.

Лічильник відвідувачів

За допомогою цього модуля Можливо вести автоматичний підрахунок перетинань лінії, заданої в Детекторі руху, з виводом цієї інформації на екран. Лічильник відвідувачів працює тільки при підключенні після Детектора руху (опція "Безперервне спрацьовування для можливості..." повинна бути відключена). Односпрямований підрахунок підходить для входів і/або виходів, де відвідувачі рухаються в одному напрямку. Кожне перетинання лінії додасть 1 до лічильника відвідувачів. Двонаправлений підрахунок підходить, якщо рух може бути як усередину, так і назовні. 1 до лічильника відвідувачів додається при кожному другому перетинанні лінії. Інтервал скидання лічильника задає, як часто починати відлік заново (наприклад, кожний новий день або на початку зміни). Також доступна функція збереження звітів у форматі CSV.

Розклад

Цей модуль дозволяє встановити дні тижня й час, коли підключені до нього модулі будуть працювати.

Можливо вказати дні тижня й інтервал часу роботи, наприклад, з 17:00 до 23:59 (запис ведеться до кінця зазначеної хвилини, тобто до 23:59:59) по буднях. Також можна окремо вказати час для будніх днів і встановити цілодобовий запис у вихідні дні.

Детектор проблем

Цей фільтр відтворює повідомлення про позаштатні ситуації, що виникають у ході роботи (відсутність відновлення картинки, відсутність звуку, поворот або зрушення камери, затемнення або засвітло картинки, закінчення

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

місця на диску, вичерпання оперативної пам'яті, перезапуск сервера). Також відображає на шкалі інтервалів в архіві, коли запису не було через виниклі проблеми (камера відключилася, була засвічена й т.д.). Здійснює запис log файлу на диск у форматі CSV. Для включення цієї функції поставте галочку в розширених налаштуваннях модуля в пункті "запис журналу подій у файл".

Маскування зон

Дана функція дозволяє сховати частина кадру, що може містити конфіденційну інформацію, від сторонніх користувачів. Наша компанія піклується про захист недоторканності приватної власності, особливо в більших містах, де в об'єктив камера відеоспостереження може з легкістю потрапити вікно Ваших сусідів.

Об'єднувач

За допомогою модуля "Об'єднувач" Можливо поєднувати зображення із двох відео джерел в одне. Підключите до входу об'єднувача джерела із зображеннями, тоді на виході буде картинка із двох джерел відразу.

Можна задати взаємне розташування картинок (поєднувати горизонтально або вертикально), а також міняти порядок картинок.

Звертаємо увагу, що в результаті об'єднання одержите також тільки 1 звуковий потік.

Зменшення

За допомогою модуля "Зменшення" Можливо змінити розмір вступник картинки.

У налаштуваннях модуля виберіть розмір результуючої картинки у відсотках від розміру вихідної.

Поворот зображення

За допомогою модуля "Поворот зображення" Можливо повернути або перевернути зображення з камери.

За допомогою слайдера в налаштуваннях модуля можна вибрати кут повороту зображення. Якщо відзначено галочкою опцію "Прив'язати значення до

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

інтервалу в 90 градусів" виставлене значення буде округлено в меншу сторону до 0, 90 або 180 градусів. Якщо галочку зняти, можна задати довільний кут повороту.

Корекція панорамного зображення

Цей модуль дозволяє усунути перекручування панорамних камер (камер "риб'яче око / fisheye") і одержати картинку з камери у звичному форматі.

Реле-перемикач

Цей модуль – перемикач потоку. При надходженні сигналу від керуючого модуля (за замовчуванням – модуль, підключений до Реле першим) керований ланцюжок замикається й починає працювати.

Наприклад, підключите до Реле дві камери. Після першої камери, але перед Реле підключите Детектор руху. При детектуванні руху з першої камери в такому випадку буде включатися друга камера (залежно від обраного режиму роботи Реле можливі різні типи поведіння після закінчення руху з першої камери) – її відеопотік буде передаватися до наступному послі Реле модулям.

Можливо вибрати режим роботи Реле:

Синхронно із сигналом – спрацьовувати й працювати, тільки коли від керуючого модуля йдуть спрацювання, закінчувати при надходженні не-спрацювання від керуючого модуля.

1-й сигнал – включення, 2-й – вимикання – спрацьовувати при першому керуючому сигналі, відключатися при наступному керуючому сигналі.

Одноразовий, з вимиканням через: – спрацьовувати тільки в момент спрацювання, надходження сигналу від керуючого модуля.

Швидке включення/вимикання

Цей модуль дозволяє швидко включити наступні модулі в ланцюжку через кнопку в нижній панелі або Головному Меню. Додайте його в потрібне місце в ланцюжку й, коли необхідно, наприклад, щоб почати швидкий запис, натисніть кнопку в нижній панелі або виберіть пункт у Головному Меню. Можна задати назву модуля (ID). Якщо назва однакове в різних ланцюжках, всі такі модулі будуть спрацьовувати одночасно.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Детектор об'єктів

Модуль автоматично виявляє предмети в зоні детектування. Можливо вказати розмір потрібного об'єкта, а програма з легкістю визначить його місце розташування на кадрі. Такий модуль допоможе контролювати завантаженість паркування.

Детектор осіб

Цей модуль автоматично виявляє особи всіх людей, що переміщуються в зоні детектування.

PTZ-спостереження

Цей модуль дозволяє переміщати PTZ-камеру у бік, що рухаються об'єктів. Для роботи даного модуля камера повинна підтримувати функції PTZ, сам модуль повинен бути підключений у ланцюжок після модуля "Детектор руху". Налаштуйте чутливість, спосіб детектування й інші параметри в налаштуваннях модуля "Детектор руху". У цьому випадку модуль " PTZ-спостереження" буде працювати відповідним чином.

Можна задати швидкість реакції на об'єкти, що рухаються (камера буде повертатися, якщо рух довший заданої тривалості); час повернення у вихідну позицію й очікування до початку наступного спостереження; максимально припустимий поворот камери; максимальне збільшення; а також розмір кроку повороту й збільшення (впливає на швидкість/кут повороту).

Маркування

За допомогою модуля "Маркування" Можливо додати дату й час, довільний текст і GPS координати прямо в одержуване з камери зображення. Можна задати місце розташування маркування, формат дати/часу й розмір шрифту.

HTTP маркування

Цей модуль – комбінація "HTTP перемикача" і "Маркування". Здатний приймати по мережі команду й включати/виключати (зі збереженням метаданих) наступні модулі (всі модулі далі в ланцюжку після HTTP маркірувальника), при цьому передаючи їм заданий ідентифікатор (ID). По

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

ідентифікаторі (ID) можна знайти цей запис в архіві в браузері й переглянути її (у такому випадку, щоб переглядати архів онлайн, у ланцюжку повинен бути включений модуль "Веб Сервер"). Як і HTTP перемикач, працює за принципом розмикача/замикача ланцюга. При одержанні команди на включення замикає ланцюг. При одержанні команди на вимикання розмикає ланцюг. При одержанні команди на продовження – продовжує тримати ланцюг замкнутої. Якщо протягом тривалого часу не прийшла команда на продовження/вимикання, тоді ланцюг розмикається автоматично.

Наприклад, підключите HTTP маркірувальник перед архівом і відправте запит на URL включення, указавши ідентифікатор (наприклад, "id=123"). Щоб продовжити запис в архів, відправте команду на продовження. Для зупинки запису в архів відправте команду вимикання.

Запит можна посилати через програми автоматизації або через адресний рядок веб браузера.

HTTP перемикач

Цей модуль дозволяє включати/виключати наступні модулі (всі модулі далі в ланцюжку після HTTP перемикача) при прийманні http-запиту. Працює за принципом розмикача/замикача ланцюга – при одержанні команди на вимикання розмикає ланцюг, при одержанні команди на включення замикає ланцюг.

URL-шлях для включення: при надходженні запиту по цьому URL модуль не буде блокувати відеопотік, і, якщо зазначено таймаут, по його витіканню модуль почне блокувати відеопотік.

URL-шлях для вимикання: при надходженні http-запиту, модуль заблокує потік на заданий інтервал часу (щоб примусово відключити блокування, необхідно надіслати запит на URL включення).

Наприклад, підключите HTTP перемикач перед архівом і відправте запит на URL вимикання, якщо треба зупинити запис в архів. Щоб відновити запис в архів, відправте запит на URL включення. Запит можна посилати через програми автоматизації або через адресний рядок веб браузера.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

записувати в архів події, що передують безпосередньому виявленню руху. Використовуйте пост-запис, щоб зберегти в архів кадри після закінчення руху.

Задавши **розмір об'єкта**, можна зменшити рівень помилкового спрацьовування на рух не об'єктів, що цікавлять: занадто великих або занадто дрібних об'єктів, а також на повну зміну картинки, наприклад, при переміщенні камери.

Налаштування в детекторі руху міняють колір залежно від того, чи блокують вони відео або пропускають далі по ланцюжку. Жовтий колір – відео заблоковано, зелений – відео передається наступним модулям.

Якщо протягом останніх 5 хвилин був виявлений рух у заданому секторі, що відповідає прев'ю в режимі перегляду підсвічується кольором у колірному інтервалі від червоного до білого (чим колір світліше, тим більше часу пройшло з моменту виявлення руху). При цьому на зображенні з'являється час останнього виявлення руху. При наведенні миші на таке зображення – рамка здобуває рожевий колір. Якщо рух не був зафіксований протягом останніх 5 хвилин – рамка біла, при наведенні – блакитна.

За допомогою опції "**Детектувати тільки відсутність руху протягом**" можна змусити детектор руху працювати навпаки – спрацьовувати, коли рух у поле зору камери припинилося (наприклад, при моніторингу лічильників або встаткування). Слайдером задайте час, протягом якого рух повинне відновити, інакше відбудеться спрацьовання наступних модулів.

Чекбокс "**Безперервне спрацьовування для можливості пошуку руху в архіві**": якщо поставити галочку, детектор руху буде фактично переведений у стан "Проникний" – запис в архів буде йти безупинно. Використовується, якщо детектор руху не потрібний (навпроти, потрібна постійний запис), але хочете використовувати опцію пошуку по русі в архіві. Детектор буде використовуватися тільки для збору даних про рух.

Перегляд архіву

Запису, зроблені модулем "Перегляд і Архів", будуть зберігатися в каталог користувача за замовчуванням або по заданому Вами шляхи. Архів постачений зручним убудованим програвачем, перейти в який Можливо за допомогою кнопки "Play" у вікні перегляду всіх камер або у вікні швидких налаштувань. У програвачі Можливо переглядати запису, у той час як запис триває. Часова шкала архіву регулюється від секунд до днів, таким чином, Можливо легко й швидко знайти необхідний часовий відрізок і почати відтворення саме із цього моменту. Масштаб часової шкали змінюється перетягання бігунка вправо або вліво або прокручуванням коліщати мишки. Завдяки перемотуванню знайти потрібний момент не тільки легко, але й швидко – натисніть стрілки, розташовані з боків часової шкали, для переміщення на хвилину вперед/назад при безперервному записі або на наступне/попередній рух, якщо запис ведеться по детекторі руху. Також Можливо набудовувати швидкість відтворення записів, щоб, приміром, пролистати у прискореному режимі не фрагменти, що цікавлять, й зосередитися на відрізках, що більше заслуговують пильної уваги. Також швидкий перехіді забезпечує календар, де можливо вибрати дату й час події й відкрити його.

У меню програвача архіву є опції "Відкривати архів на" ("Стартова позиція"), "Експорт", "Видалення" і "Пошук по русі". Перша дозволить вибрати, з якого моменту починати відтворення при наступних відкриттях архіву (за 5 хвилин до кінця, з початку або з того моменту, де Ви зупинилися востаннє – цей варіант використовується за замовчуванням). За допомогою **Експорту** Можливо **експортувати будь-який відрізок** у потрібному форматі (mjpg, wmv, MPEG-4 avi, H.264 avi), зберегти кадри в JPEG, наприклад, для передачі відео/фото доказів у слідчі органи. Виберіть цю опцію – над часовою шкалою з'являться маркери – білі трикутники. Перетаскуючи їх, виберіть потрібний відрізок і знову виберіть пункт "Експорт". З'явиться нове вікно, у якому Можливо дати ім'я фрагменту, вибрати директорію для збереження й указати формат, у який його необхідно конвертувати.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

За допомогою "**Видалення**" можна безповоротно видалити фрагмент із архіву. Працює він по схожому з "Експортом" принципу – за допомогою білих маркерів виділяється відрізок, потім у меню знову вибирається цей пункт.

Калькулятор мінімальних системних вимог допоможе оцінити необхідний дисковий простір для планованої системи відеоспостереження.

Професійні можливості в Архіві

Візуалізація виявленого руху

Функція візуалізації руху не просто виявить об'єкти, що рухаються, але й забезпечить їхній безперервний супровід (трасування).

Можливість створити в архіві відрізки, що не видаляються

Виділіть певний відрізок часу в Архіві й позначте його невидаляємим як (Архів-> Головне меню-> Видалення інтервалу-> Задати не видаляється інтервал, що). Ця функція дозволить не втратити важливі відрізки запису й зведе на мінімум можливість їхнього видалення операторами.

Пошук по русі

Пошук по русі робить навігацію в Архіві простіше й зручніше. Функція дозволить знайти події з рухом в інтересуючій області. Виберіть цей пункт і виділіть на зображенні з камери прямокутник, де буде виконаний пошук по русі. Знайдені події будуть виділені на часовій шкалі зеленим кольором. Одночасно може бути виділено кілька таких зон для пошуку. Щоб видалити яку-небудь зону, کلیکніть усередині її. Щоб видалити всі виділені області, виберіть відповідний пункт у меню.

Структурна схема системи зображена на рисунку 3.1.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53



Рисунок 3.1 – Структурна схема системи

Клієнт-серверна архітектура

ПЗ для відеоспостереження засновано на архітектурі "Клієнт-сервер".

Це означає, що програма складається з 2 частин – Сервер і Клієнт.

Сервер обробляє камери й інші джерела сигналу й у цілому виконує всю роботу, у той час як **Клієнт** використовується для перегляду й зв'язку із сервером.

Запускаючи програму для web камер з файлу, що виконується, Ви запускаєте одночасно й Сервер, і Клієнт – робота сервера не видна неозброєним оком, а Клієнт – це те, що видно як головне вікно.

Пізніше, використовуючи віддалений доступ, Ви будете використовувати переваги клієнт-серверної архітектури для підключення клієнтською частиною з одного пристрою до серверної частини на іншому. До одного сервера може бути підключена необмежена кількість клієнтських частин. Підключення відбувається за допомогою Діалогу підключення (Головне меню -> Віддалений доступ -> Підключення до віддаленого сервера), де потрібно буде замінити "локальний" IP адресу (localhost) комп'ютера, що підключається, на "віддалений" IP адреса або адреса Ретранслятора.

Оскільки сервер є важливою частиною програми, підключення до сервера (локальному або віддаленому) необхідно для виконання всіх основних операцій (наприклад, активація ліцензійної версії, робота з камерами й т.д.) Якщо з якоїсь причини підключення встановити не вдалося, робота буде неможлива. У такому випадку, будь ласка, перевірте правильність адреси для підключення.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. З лівого боку зображено блок розробленого ПЗ, яке інстальовано на сервері системи відеоспостереження.

Через інтерфейс ПЗ можна проводити перегляд дій які проходять у приміщеннях Інтернет кафе з аналізом зображення та сповіщенням при виявленні руху при заданих умовах. Крім цього розроблене ПЗ має розширену систему налаштувань під певні дії та архів роботи камер з гнучкою системою сортування. Архів зберігає зображення у форматі *.jpg та відео потік у *.avi, це забезпечує універсальність дій при перегляді.

Для фіксування дій у приміщеннях Інтернет кафе необхідно знати точні часові дані, для цього розроблено модуль синхронізації часу з камерами та Інтернет сервером точного відліку часу.

Відеозображення з камер отримується через захищені канали по локальній

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

мережі через протокол Microsoft Media Server. IP-камери підключено напряду до маршрутизаторів. Характеристики потоку IP-камери – 48 біт RGB кольору та 1280x1024 роздільна здатність. Цього достатньо для отримання чіткого зображення та розпізнавання облич осіб у зоні бачення IP-камери.

Для налаштування IP-камери можна використовувати WEB інтерфейс маршрутизатора.

Розроблена у бакалаврській дипломній роботі система контролю дій користувачів у Інтернет кафе забезпечує гарантовано стійкій відеосигнал з мінімальною кількістю шумів та з максимальною різкістю зображення у приміщеннях Інтернет кафе. Система ґрунтується на новітніх алгоритмах обробки відеозображень. Після проголошення незалежності України і переходу її економіки на ринкову, з'явилася гостра потреба в системах відеоспостереження з якісними алгоритмами обробки потоків відео інформації.

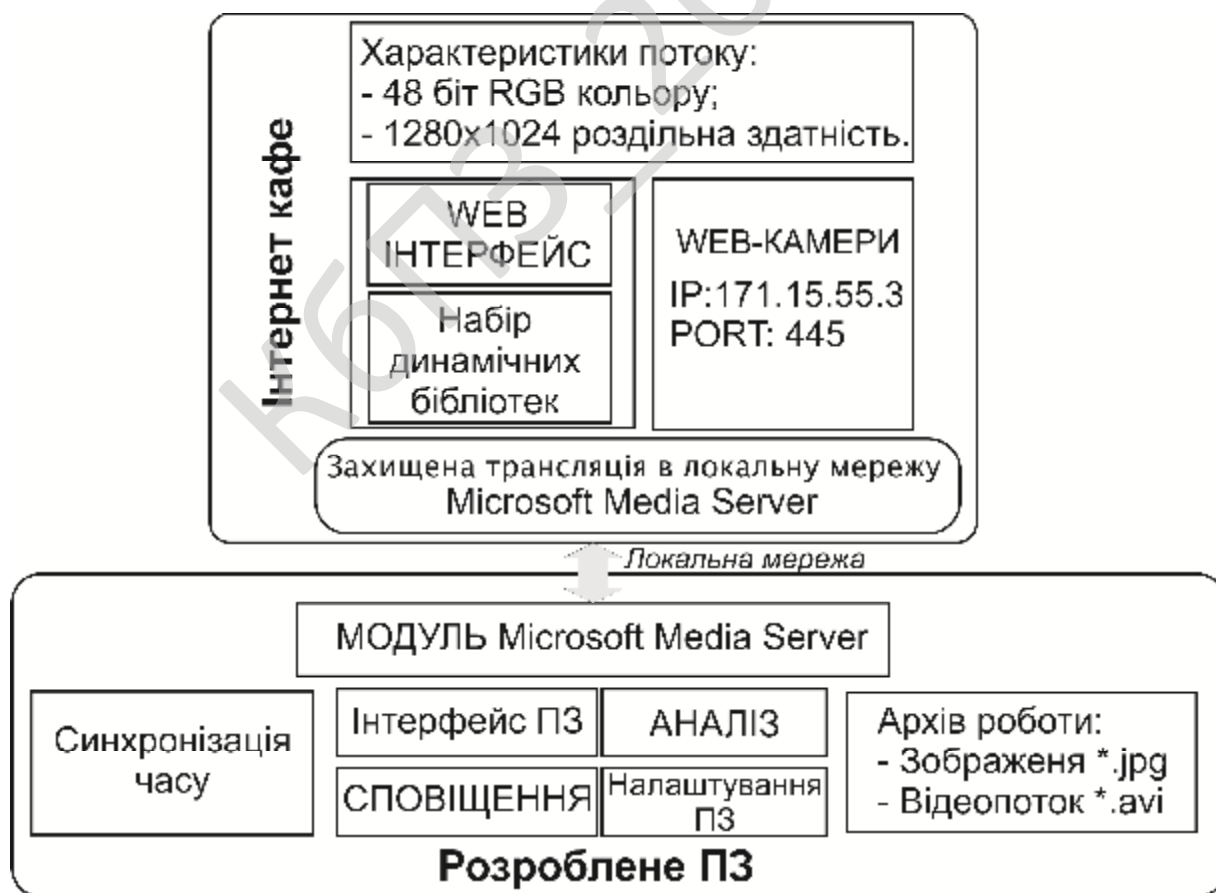


Рисунок 3.2 – Функціональна схема роботи системи

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

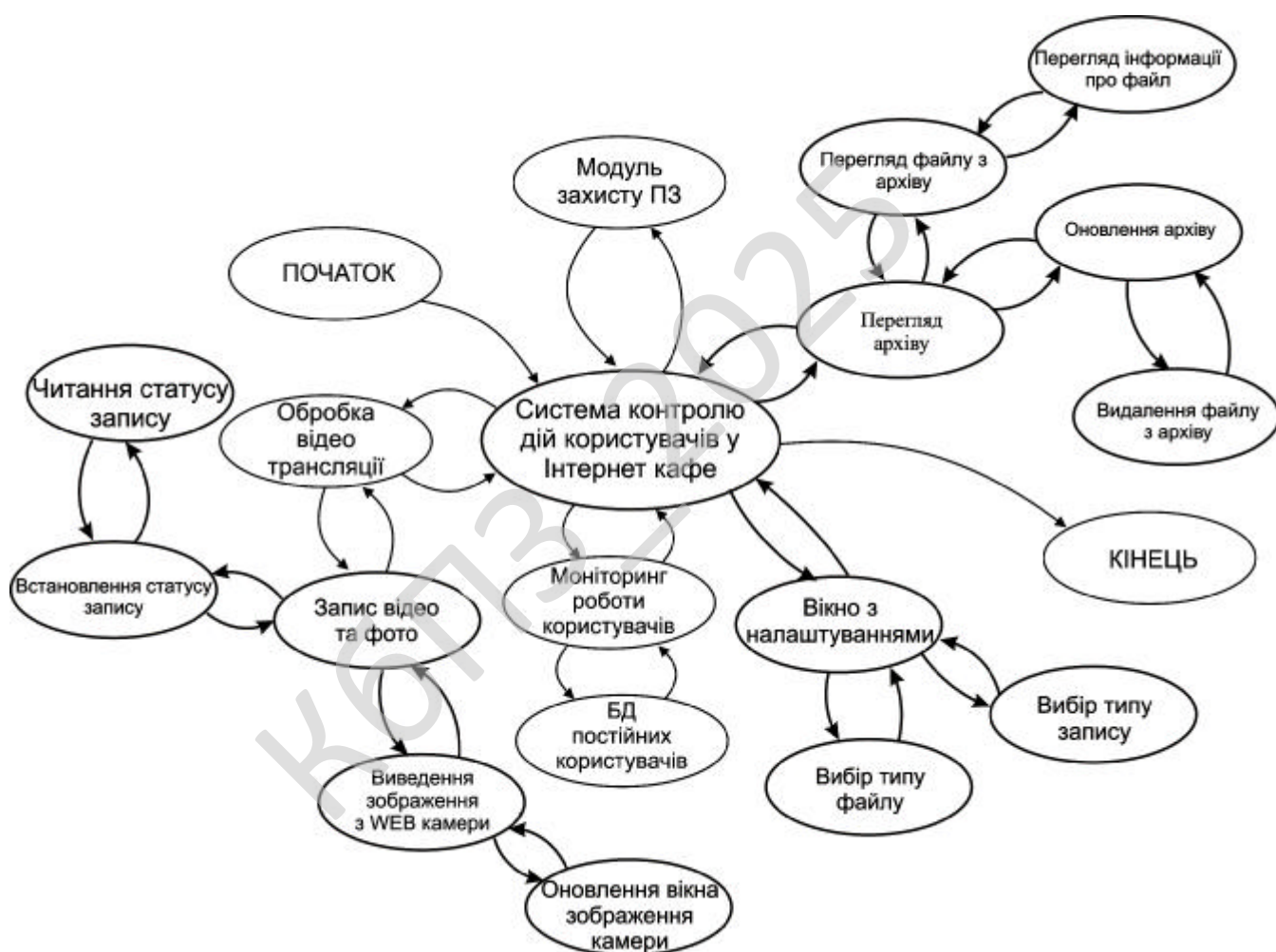


Рисунок 3.3 – Діаграма взаємодії процесів

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ - 2025

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми, на рисунку 4.2 зображено роботу підпрограм.

Під час роботи над бакалаврською дипломною роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

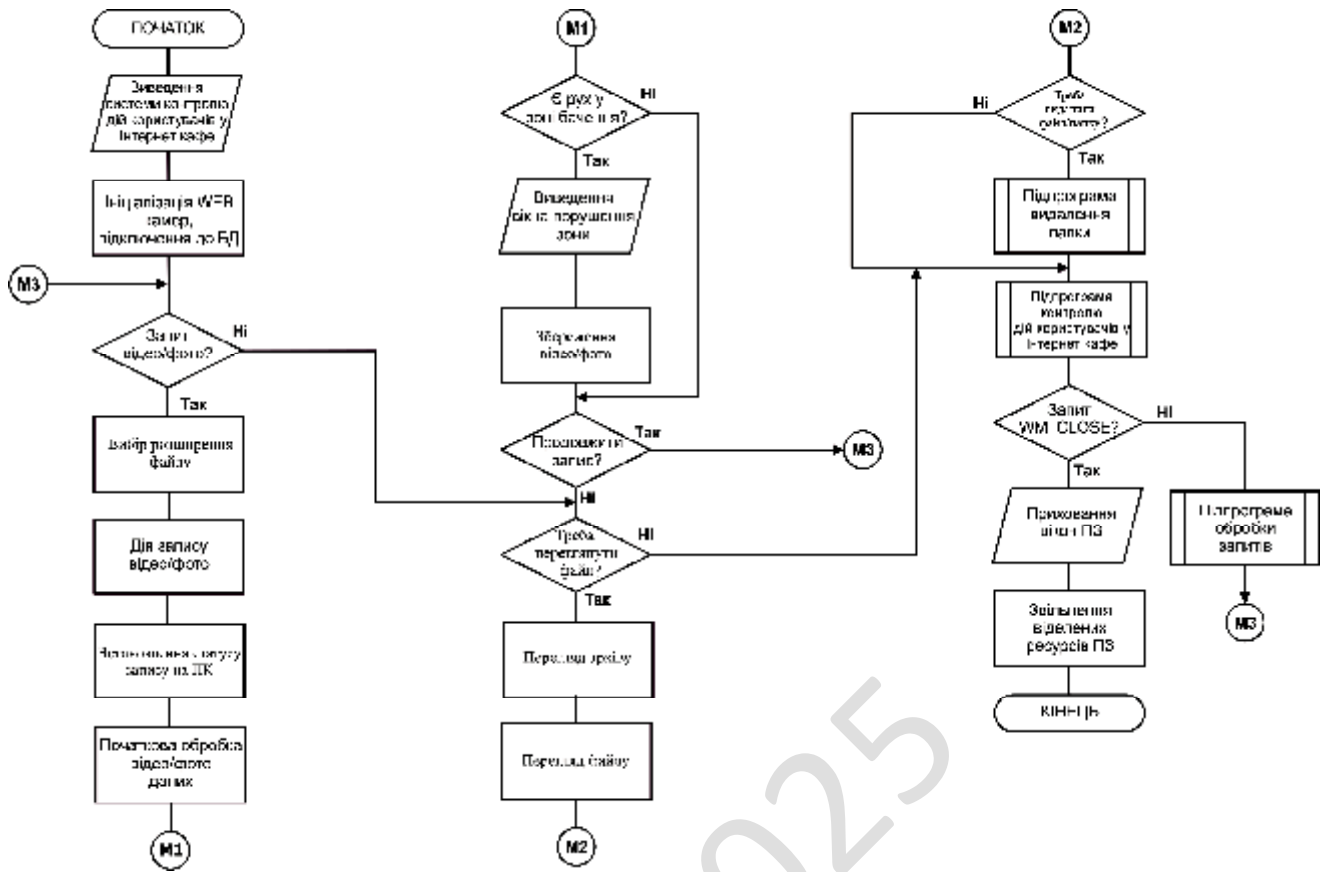


Рисунок 4.1 – Блок-схема основної програми

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Також при розробці бакалаврської дипломної роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів.

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (association relationship);
- включення (include relationship);
- розширення (extend relationship);
- узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізнити один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці. Кратність можна задати рівною одиниці (1), можна вказати діапазон: "нуль або одиниця" (0..1), "багато" (0 .. *), "одиниця або більше" (1 .. *). Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку можна задати і більш складні кратності, наприклад 0..1, 3..4, 6 .. *, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовани ромбиком.

Опис системи

Система контролю дій користувачів у Інтернет кафе призначена для моніторингу активності клієнтів, обмеження доступу до небажаних ресурсів та ведення журналу дій для забезпечення безпеки та ефективного управління роботою закладу.

Архітектура системи

Система складається з клієнтського додатка, встановленого на кожному комп'ютері Інтернет кафе, сервера для обробки даних та бази даних для зберігання інформації про користувачів, сесії та дії. Основні модулі системи включають:

1. Клієнтський додаток.
2. Сервер управління.
3. База даних.
4. Модуль обробки статистики та звітності.

Опис коду системи

```
import sqlite3
import datetime
import socket
import threading

# Клас для роботи з базою даних
class Database:
    def __init__(self, db_name):
        self.connection = sqlite3.connect(db_name)
```

					VKPB-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66


```

        action_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        self.cursor.execute("INSERT INTO actions (user_id, action_time,
action_description) VALUES (?, ?, ?)", (user_id, action_time, description))
        self.connection.commit()

# Клас для серверної частини системи
class ControlServer:
    def __init__(self, host, port):
        self.server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.server.bind((host, port))
        self.server.listen(5)
        self.database = Database("internet_cafe.db")

# Обробляє підключення клієнтів
def handle_client(self, client_socket, client_address):
    client_ip = client_address[0]
    print(f"З'єднання з {client_ip} встановлено")
    username = client_socket.recv(1024).decode("utf-8")
    user_id = self.database.add_user(username, client_ip)

    try:
        while True:
            action = client_socket.recv(1024).decode("utf-8")
            if action == "END":
                self.database.end_user_session(user_id)
                print(f"Сесія для {username} завершена")
                break
            else:
                self.database.add_action(user_id, action)
                print(f"Дія {action} зареєстрована для {username}")

    except:
        self.database.end_user_session(user_id)
        print(f"Підключення з {username} втрачено")

    client_socket.close()

# Запускає сервер
def start(self):
    print("Сервер контролю запущено")
    while True:
        client_socket, client_address = self.server.accept()
        client_thread = threading.Thread(target=self.handle_client,

```

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

```
args=(client_socket, client_address))
    client_thread.start()

# Запуск серверної частини
if __name__ == "__main__":
    control_server = ControlServer("0.0.0.0", 9999)
    control_server.start()
```

Модулі системи

1. Клієнтський додаток – встановлюється на кожному комп'ютері. Він передає дані про дії користувача на сервер.

2. Сервер управління – приймає дані від клієнтів, зберігає їх у базі даних і обробляє.

3. База даних – зберігає інформацію про користувачів, їх сесії та дії.

4. Модуль обробки статистики – генерує звіти про активність користувачів.

Розрахунки і підтвердження проектних рішень обраної структури бази даних

Припустимо, що Інтернет кафе має 20 комп'ютерів. У середньому кожен користувач виконує 100 дій за сесію. База даних здатна обробити 2000 записів за одну зміну. Зробимо розрахунок обсягу даних, що зберігається в системі:

1. Середній розмір запису користувача – 100 байтів.

2. Обсяг даних за одну зміну – 100 байтів * 2000 записів = 200 кілобайтів.

3. Обсяг даних за місяць (30 змін) – 200 кілобайтів * 30 змін = 6 мегабайтів.

Цей розрахунок підтверджує, що обрана структура бази даних є оптимальною і може ефективно зберігати дані за тривалий період.

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищаються за допомогою використання алгоритму Md5. Він отримує на вході повідомлення довільної довжини і створює на виході

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

дайджест повідомлення довжиною 128 біт. Алгоритм складається з наступних кроків:

1. Додавання недостаючих біт. Повідомлення доповнюється так, щоб його довжина стала рівна 448 по модулю 512 (довжина $448 \bmod 512$). Це означає, що довжина доданого повідомлення на 64 біта менше, ніж число, кратне 512. Додавання проводиться завжди, навіть якщо повідомлення має потрібну довжину. Наприклад, якщо довжина повідомлення 448 біт, воно доповнюється 512 бітами до 960 біт. Таким чином, число біт, що додаються, знаходиться в діапазоні від 1 до 512.

Додавання складається з одиниці, за якою слідує необхідна кількість нулів.

2. Додавання довжини. 64-бітове представлення довжини початкового (до додавання) повідомлення в бітах приєднується до результату першого кроку. Якщо первинна довжина більша, ніж 264, то використовуються тільки останні 64 біта. Таким чином, поле містить довжину початкового повідомлення по модулю 264.

В результаті перших двох кроків створюється повідомлення, довжина якого кратна 512 бітам. Це розширене повідомлення представляється як послідовність 512-бітових блоків Y_0, Y_1, \dots, Y_{l-1} , при цьому загальна довжина розширеного повідомлення рівна $L * 512$ бітам. Таким чином, довжина отриманого розширеного повідомлення кратна шістнадцяти 32-бітовим словам.

3. Ініціалізація MD-буфера. У алгоритмі Md5 використовується 128-бітовий буфер для зберігання проміжних і остаточних результатів хеш-функції. Буфер може бути представлений як чотири 32-бітові регістри (A, B, C, D). Ці регістри ініціалізувалися наступними шістнадцятковими числами:

$$A = 01234567$$

$$B = 89abcdef$$

$$C = Fedcba98$$

$$D = 76543210$$

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

4. Обробка послідовності 512-бітових (16-словних) блоків. Основою алгоритму Md5 є модуль, що складається з чотирьох циклічних обробок, позначений як Hmd5. Чотири цикли мають схожу структуру, але кожен цикл використовує свою елементарну логічну функцію, ff , що позначається, fg , fh і fi відповідно.

Кожен цикл приймає на вхід поточний 512-бітовий блок Y_q , що обробляється в даний момент, і 128-бітове значення буфера ABCD, яке є проміжним значенням дайджесту, і змінює вміст цього буфера. Кожен цикл також використовує четверту частину 64-елементної таблиці $T[1..64]$, побудованої на основі функції \sin . i -ий елемент T , $T[i]$, що позначається, має значення, рівне цілій частині від $232 * \text{abs}(\sin(i))$, і задане в радіанах. Оскільки $\text{abs}(\sin(i))$ є числом між 0 і 1, кожен елемент T є цілим, яке може бути представлене 32 бітами. Таблиця забезпечує “випадковий” набір 32-бітових значень, які повинні ліквідувати будь-яку регулярність у вхідних даних.

Для отримання Md_{q+1} вихід чотирьох циклів складається по модулю 232 з Md_q . Складання виконується незалежно для кожного з чотирьох слів в буфері.

5) Вихід Md5. Після обробки всіх L 512-бітових блоків виходом L -ої стадії є 128-бітовий дайджест повідомлення.

Детальніше логіку кожного з чотирьох циклів виконання одного 512-бітового блоку розглянуто нижче. Кожен цикл складається з 16 кроків, що оперують з буфером ABCD.

$$A \leftarrow B + \text{CLSS}(A + f(B, C, D) + X[k] + T[i]),$$

A, B, C, D – чотири слова буфера; після виконання кожного окремого кроку відбувається циклічне зрушення вліво на одне слово.

f – одна з елементарних функцій ff, fg, fh, fi .

CLSs – циклічне зрушення вліво на s біт 32-бітового аргументу.

$X[k] = M[q * 16 + k]$ – кодує 32-бітове слово в q -ому 512 блоці повідомлення.

$T[i]$ – i -е 32-бітове слово в матриці T .

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання бакалаврської дипломної роботи.

Розроблене програмне забезпечення системи контролю дій користувачів у Інтернет кафе складається з наступних функціональних блоків:

- Навігаційне меню: Налаштування; Фільтри; Отриманий матеріал; Допомога.
- Вікна виведення зображення, розділу обрання вхідного сигналу.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

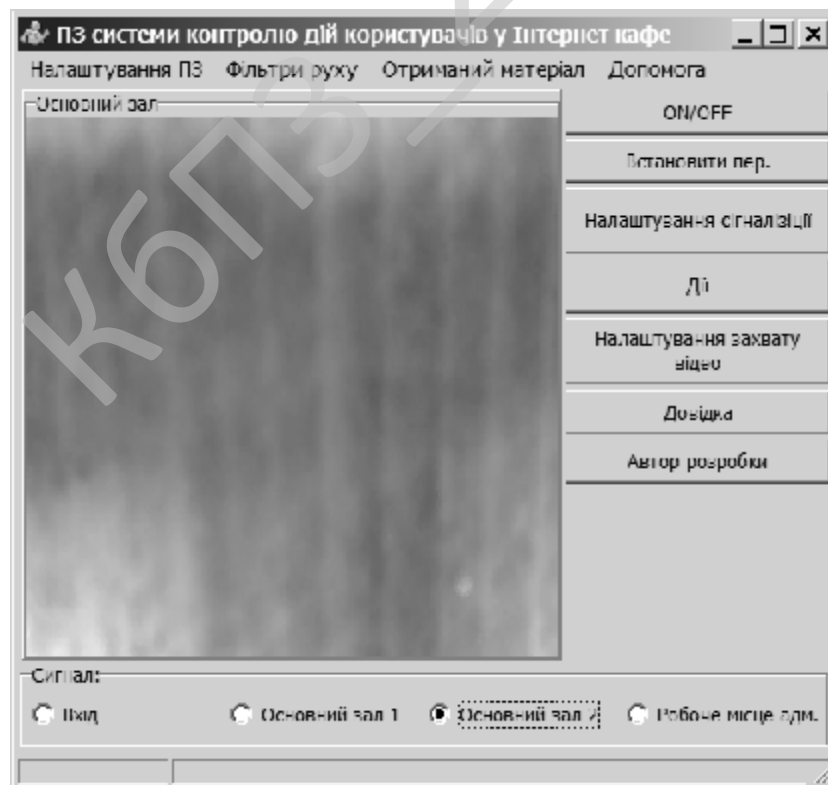


Рисунок 5.1 – Головне вікно розробленого ПЗ

Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з'явиться вікно показане на рисунку 5.2.

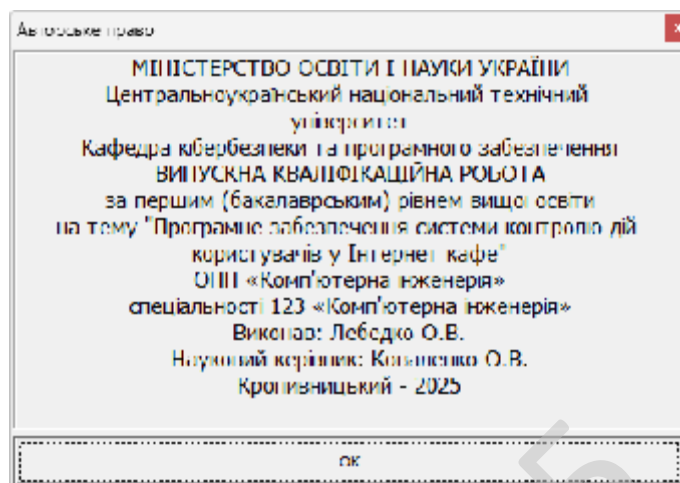


Рисунок 5.2 – Вікно розробника ПЗ

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

- При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

- Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи контролю дій користувачів у Інтернет кафе.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем контролю дій користувачів у Інтернет кафе.
- Досліджена система контролю дій користувачів у Інтернет кафе.
- На основі отриманих результатів досліджень створена програмна реалізація системи контролю дій користувачів у Інтернет кафе.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання контролю дій користувачів у Інтернет кафе.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи контролю дій користувачів у Інтернет кафе. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Md5.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

КБПЗ_2025

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
2. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
3. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.
4. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447
5. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
6. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchov, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
7. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
8. Smirnov, O., Neskrodieva, T., Fedorov, E., Rudakov, K., Neskrodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

9. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.

10. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>

11. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

12. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

13. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

14. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

15. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

16. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

17. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

18. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

19. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

20. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

21. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

22. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

23. Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

24. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

25. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

26. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

27. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

28. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

29. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

30. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

31. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

32. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

33. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

34. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

35. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

36. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

37. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

38. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

39. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнoукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

40. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» Комп'ютерні науки та кібербезпека. № 4. С. 30-37. 2019.

41. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

42. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

43. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

44. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для

модельовання трафіку у мережі. Центральнoукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

45. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

46. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 36-39.

47. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

48. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". - Випуск 2 (118). т.2. - Х.: ХУПС - 2014. - С. 64-67

49. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної науково-практичної конференції "Проблеми та перспективи розвитку ІТ-індустрії". м. Харків. 17-18 квітня 2014р. – Харків: ХНЄУ. - 2014. - С. 240.

50. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лешко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник. – Кіровоград: КНТУ 2013. – 257с.

51. Смірнов О.А., Дреєв О.М., Доренський О.П. «Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі. Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – Х.: ХУПС – 2013. – С. 234-239.

					ВКРБ-123.25.0034.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Перелік документів, що розробляються.....	5
8 Етапи розробки.....	6
9 Порядок контролю та приймання.....	6

					ВКРБ-123.25.0034.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Лебедко О.В.				<i>Програмне забезпечення системи контролю дій користувачів у Інтернет кафе</i>	Літ.	Аркуш	Аркушів
Перевірів	Коваленко О.В.					Б	1	6
Н. Контр.	Коваленко А.С.					ЦНТУ КІ-21-2		
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи контролю дій користувачів у Інтернет кафе.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 47-02 від 17.01.2025 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи контролю дій користувачів у Інтернет кафе.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					ВКРБ-123.25.0034.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи контролю дій користувачів у Інтернет кафе;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРБ-123.25.0034.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Python.

					ВКРБ-123.25.0034.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 84 аркуші.

8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

					ВКРБ-123.25.0034.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2025 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 3.06.2025 р.

					ВКРБ-123.25.0034.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
першим (бакалаврським) рівнем вищої освіти

_____ Коваленко О.В.

*Програмне забезпечення системи контролю дій користувачів у Інтернет
кафе*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 21

Літера: РП

Кропивницький – 2025 року

Основна програма

```
import socket
import threading
import datetime
import time
from authentication import Authentication
from roles import Roles
from session_limit import SessionLimit
from remote_lock import RemoteLock
from session_warning import SessionWarning
from site_filter import SiteFilter
from software_monitor import SoftwareMonitor
from traffic_analysis import TrafficAnalysis
from alerts import Alerts
from backup_manager import BackupManager
from mobile_monitor import MobileMonitor
from firewall_integration import FirewallIntegration
from multi_server_support import MultiServerSupport
from report_scheduler import ReportScheduler
from access_rules import AccessRules
from video_surveillance import VideoSurveillance
from hardware_monitor import HardwareMonitor
from email_reports import EmailReports
from usb_control import USBControl
from dlp_system import DLPSystem
from client_gui import ClientGUI

# Основний клас ControlServer для управління сервером
class ControlServer:

    def __init__(self, host="0.0.0.0", port=9999):
        self.host = host
        self.port = port
        self.server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.threads = []
        self.modules_initialized = False
        self._initialize_modules()
        print(f"Сервер створено за адресою {self.host}:{self.port}")

    # Ініціалізація всіх додаткових модулів
    def _initialize_modules(self):
        if not self.modules_initialized:
            self.auth = Authentication()
            self.roles = Roles()
            self.session_limit = SessionLimit()
            self.remote_lock = RemoteLock()
            self.session_warning = SessionWarning()
            self.site_filter = SiteFilter()
            self.software_monitor = SoftwareMonitor()
            self.traffic_analysis = TrafficAnalysis()
            self.alerts = Alerts(admin_email="admin@example.com")
            self.backup_manager = BackupManager()
            self.mobile_monitor = MobileMonitor()
            self.firewall = FirewallIntegration()
            self.multi_server_support = MultiServerSupport()
            self.scheduler = ReportScheduler()
```

```

self.access_rules = AccessRules()
self.video_surveillance = VideoSurveillance()
self.hardware_monitor = HardwareMonitor()
self.email_reports = EmailReports(
    smtp_server="smtp.example.com",
    smtp_port=587,
    sender_email="system@example.com",
    sender_password="password123"
)

self.usb_control = USBControl()
self.dlp_system = DLPSystem()
self.modules_initialized = True
print("Усі модулі успішно ініціалізовані.")

# Метод для запуску сервера
def start_server(self):
    try:
        self.server.bind((self.host, self.port))
        self.server.listen(5)
        print(f"Сервер запущено за адресою {self.host}:{self.port}.")
        while True:
            client_socket, client_address = self.server.accept()
            print(f"Нове підключення: {client_address}")
            client_thread = threading.Thread(target=self.handle_client,
args=(client_socket, client_address))
            client_thread.start()
            self.threads.append(client_thread)
        except Exception as e:
            print(f"Помилка при запуску сервера: {str(e)}")

# Метод для обробки підключення клієнта
def handle_client(self, client_socket, client_address):
    client_ip = client_address[0]
    print(f"Обробка підключення від {client_ip}.")
    username = client_socket.recv(1024).decode("utf-8")
    print(f"Користувач {username} підключився.")

    if not self.auth.authenticate(username, "password123"):
        print(f"Аутентифікація для {username} не пройдена.")
        client_socket.close()
        return

    self.session_limit.start_session(username)
    self.roles.assign_role(username, "Користувач")
    self.hardware_monitor.check_cpu_status()
    self.hardware_monitor.check_memory_status()

    try:
        while True:
            action = client_socket.recv(1024).decode("utf-8")
            if action == "END":
                print(f"Сесія для {username} завершена.")
                self.session_limit.is_session_active(username)
                break
            else:
                print(f"Отримано дію: {action}")
                self.traffic_analysis.monitor_traffic()

```

```
        self.site_filter.block_site("http://example.com")
except Exception as e:
    print(f"Помилка під час обробки дій клієнта: {str(e)}")
finally:
    client_socket.close()

# Метод для зупинки сервера
def stop_server(self):
    for thread in self.threads:
        thread.join()
    self.server.close()
    print("Сервер зупинено.")
```

КБПЗ_2025

Файл authentication.py

```
import hashlib

# Клас Authentication для управління аутентифікацією
class Authentication:

    def __init__(self):
        self.user_credentials = {}

    # Метод додавання нового користувача
    def add_user(self, username, password):
        if username in self.user_credentials:
            print("Помилка: Користувач з таким ім'ям вже існує.")
            return

        if not password or len(password) < 6:
            print("Помилка: Пароль має містити мінімум 6 символів.")
            return

        password_hash = self._hash_password(password)
        self.user_credentials[username] = password_hash
        print(f"Користувач {username} успішно доданий.")
        print(f"Збережений хеш пароля: {password_hash}")

    # Метод для перевірки аутентифікації користувача
    def authenticate(self, username, password):
        if username not in self.user_credentials:
            print(f"Помилка: Користувач {username} не знайдений.")
            return False

        password_hash = self._hash_password(password)
        if self.user_credentials[username] == password_hash:
            print(f"Користувач {username} успішно аутентифікований.")
            return True
        else:
            print("Помилка: Неправильний пароль.")
            return False

    # метод хешування пароля
    def _hash_password(self, password):
        try:
            password_hash = hashlib.sha256(password.encode()).hexdigest()
            if not password_hash:
                print("Помилка: Хеш пароля не створений.")
                return password_hash
        except Exception as e:
            print(f"Помилка при хешуванні пароля: {str(e)}")
            return ""
```

Файл roles.py

```
# Клас Roles для управління ролями користувачів
class Roles:

    def __init__(self):
        self.roles = {}

    # Метод для призначення ролі користувачеві
    def assign_role(self, username, role):
        if not username:
            print("Помилка: Ім'я користувача не може бути порожнім.")
            return
        if role not in ["Адміністратор", "Модератор", "Користувач"]:
            print("Помилка: Недопустима роль.")
            return

        self.roles[username] = role
        print(f"Користувачу {username} призначено роль {role}.")

    # Метод для отримання ролі користувача
    def get_role(self, username):
        if username not in self.roles:
            print(f"Попередження: Для користувача {username} роль не
встановлена.")
            return "Користувач"
        return self.roles[username]
```

Файл session_limit.py

```
import datetime

# Клас для управління часом сесії користувача
class SessionLimit:

    def __init__(self, session_duration_minutes=60):
        self.session_duration =
datetime.timedelta(minutes=session_duration_minutes)
        self.sessions = {}

    # Метод початку нової сесії
    def start_session(self, username):
        if username in self.sessions:
            print(f"Попередження: Сесія для {username} вже розпочата.")
            return

        self.sessions[username] = datetime.datetime.now()
        print(f"Сесія для {username} розпочата о {self.sessions[username]}")

    # Метод перевірки активності сесії
    def is_session_active(self, username):
        if username not in self.sessions:
            print(f"Попередження: Сесія для {username} не знайдена.")
            return False

        session_start = self.sessions[username]
        time_elapsed = datetime.datetime.now() - session_start
        if time_elapsed < self.session_duration:
            print(f"Сесія для {username} активна. Час, що минув:
{time_elapsed}.")
            return True
        else:
            print(f"Сесія для {username} завершена. Час, що минув:
{time_elapsed}.")
            return False
```

Файл site_filter.py

```
# Клас SiteFilter для управління списком заблокованих сайтів
class SiteFilter:

    def __init__(self):
        self.blocked_sites = set()

    # Метод для додавання сайту до списку блокування
    def block_site(self, site):
        if not site.startswith("http"):
            print(f"Помилка: Сайт {site} не є дійсною URL-адресою.")
            return

        if site in self.blocked_sites:
            print(f"Попередження: Сайт {site} вже заблокований.")
        else:
            self.blocked_sites.add(site)
            print(f"Сайт {site} додано до списку блокування.")

    # Метод перевірки, чи сайт заблокований
    def is_site_blocked(self, site):
        if site in self.blocked_sites:
            print(f"Сайт {site} заблоковано.")
            return True
        else:
            print(f"Сайт {site} дозволено.")
            return False
```

Файл traffic_analysis.py

```
import psutil
import time

# Клас TrafficAnalysis для моніторингу мережевого трафіку
class TrafficAnalysis:

    def __init__(self, monitor_duration=10):
        self.monitor_duration = monitor_duration
        self.traffic_data = []

    # Метод для моніторингу мережевого трафіку
    def monitor_traffic(self):
        print("Запуск моніторингу мережевого трафіку...")
        for _ in range(self.monitor_duration):
            net_io = psutil.net_io_counters()
            sent = net_io.bytes_sent
            received = net_io.bytes_recv
            self.traffic_data.append((sent, received))
            print(f"Дані трафіку – Відправлено: {sent} байт, Отримано:
{received} байт")
            time.sleep(1)
        print("Моніторинг завершено.")

    # Метод для аналізу трафіку
    def analyze_traffic(self):
        total_sent = sum(data[0] for data in self.traffic_data)
        total_received = sum(data[1] for data in self.traffic_data)
        print(f"Загальний обсяг відправлених даних: {total_sent} байт")
        print(f"Загальний обсяг отриманих даних: {total_received} байт")
```

Файл alerts.py

```
import smtplib
from email.mime.text import MIMEText

# Клас Alerts для надсилання повідомлень адміністраторам
class Alerts:

    def __init__(self, admin_email):
        self.admin_email = admin_email
        self.smtp_server = "smtp.example.com"
        self.smtp_port = 587
        self.sender_email = "system@example.com"
        self.sender_password = "password123"

    # Метод для відправки оповіщення
    def send_alert(self, subject, message):
        try:
            msg = MIMEText(message)
            msg['Subject'] = subject
            msg['From'] = self.sender_email
            msg['To'] = self.admin_email

            with smtplib.SMTP(self.smtp_server, self.smtp_port) as server:
                server.starttls()
                server.login(self.sender_email, self.sender_password)
                server.sendmail(self.sender_email, self.admin_email,
                                msg.as_string())
            print(f"Оповіщення надіслано адміністратору
{self.admin_email}.")
        except Exception as e:
            print(f"Помилка при надсиланні оповіщення: {str(e)}")
```

Файл backup_manager.py

```
import os
import datetime

# Клас BackupManager для керування резервними копіями та логами
class BackupManager:

    def __init__(self, backup_dir="backups", log_dir="logs"):
        self.backup_dir = backup_dir
        self.log_dir = log_dir
        self._create_directories()

    # Метод для створення необхідних директорій
    def _create_directories(self):
        if not os.path.exists(self.backup_dir):
            os.makedirs(self.backup_dir)
            print(f"Директорія {self.backup_dir} створена.")
        if not os.path.exists(self.log_dir):
            os.makedirs(self.log_dir)
            print(f"Директорія {self.log_dir} створена.")

    # Метод для автоматичного видалення старих файлів
    def delete_old_files(self, directory, days_old=30):
        now = datetime.datetime.now()
        for filename in os.listdir(directory):
            file_path = os.path.join(directory, filename)
            if os.path.isfile(file_path):
                file_time =
datetime.datetime.fromtimestamp(os.path.getmtime(file_path))
                if (now - file_time).days > days_old:
                    os.remove(file_path)
                    print(f"Файл {filename} видалений, оскільки йому більше
{days_old} днів.")
```

Файл `firewall_integration.py`

```
import subprocess

# Клас FirewallIntegration для роботи з брандмауером
class FirewallIntegration:

    def __init__(self):
        self.rules = []

    # Метод для додавання правила в брандмауер
    def add_firewall_rule(self, rule):
        try:
            subprocess.run(rule, shell=True, check=True)
            self.rules.append(rule)
            print(f"Правило брандмауера додано: {rule}")
        except subprocess.CalledProcessError as e:
            print(f"Помилка при додаванні правила: {str(e)}")
```

КБПЗ_2025

Файл report_scheduler.py

```
import schedule
import time

# Клас ReportScheduler для планування звітів
class ReportScheduler:

    def __init__(self):
        self.reports = []

    # Метод для додавання звіту
    def add_report(self, report_name):
        self.reports.append(report_name)
        print(f"Звіт {report_name} додано до розкладу.")

    # Метод для планування щотижневих звітів
    def schedule_weekly_reports(self):
        schedule.every().monday.at("09:00").do(self._generate_reports)
        print("Щотижневі звіти заплановано на кожен понеділок о 09:00.")

    # Метод для виконання генерації звітів
    def _generate_reports(self):
        for report in self.reports:
            print(f"Генерується звіт: {report}")

    # Метод для запуску розкладу
    def run_scheduler(self):
        print("Запуск планувальника звітів...")
        while True:
            schedule.run_pending()
            time.sleep(1)
```

Файл access_rules.py

```
# Клас AccessRules для управління правилами доступу
class AccessRules:

    def __init__(self):
        self.rules = {} # Зберігає правила у форматі: username -> list of rules

    # Метод додавання правила доступу
    def add_rule(self, username, rule):
        if not username:
            print("Помилка: Ім'я користувача не може бути порожнім.")
            return
        if username not in self.rules:
            self.rules[username] = []
        if rule in self.rules[username]:
            print(f"Попередження: Правило '{rule}' вже додано для {username}.")
        else:
            self.rules[username].append(rule)
            print(f"Правило '{rule}' додано для користувача {username}.")

    # Метод перевірки правила доступу
    def check_rule(self, username, rule):
        if username not in self.rules:
            print(f"Попередження: Для користувача {username} немає встановлених
правил.")
            return False
        if rule in self.rules[username]:
            print(f"Правило '{rule}' знайдено для користувача {username}.")
            return True
        else:
            print(f"Правило '{rule}' відсутнє для користувача {username}.")
            return False
```

Файл video_surveillance.py

```
# Клас VideoSurveillance для роботи з системами відеоспостереження
class VideoSurveillance:

    def __init__(self):
        self.cameras = [] # Список підключених камер

    # Метод для додавання камери
    def add_camera(self, camera_id):
        if camera_id in self.cameras:
            print(f"Попередження: Камера з ID {camera_id} вже підключена.")
        else:
            self.cameras.append(camera_id)
            print(f"Камера з ID {camera_id} додана.")

    # Метод для перегляду списку підключених камер
    def list_cameras(self):
        print("Список підключених камер:")
        for camera in self.cameras:
            print(f"Камера ID: {camera}")
```

КБПЗ_2025

Файл hardware_monitor.py

```
import psutil

# Клас HardwareMonitor для моніторингу апаратного стану
class HardwareMonitor:

    def check_cpu_status(self):
        cpu_percent = psutil.cpu_percent(interval=1)
        print(f"Завантаження CPU: {cpu_percent}%")
        if cpu_percent > 80:
            print("Попередження: Високе завантаження CPU.")

    def check_memory_status(self):
        memory = psutil.virtual_memory()
        print(f"Використана пам'ять: {memory.percent}%")
        if memory.percent > 85:
            print("Попередження: Низький рівень доступної пам'яті.")

    def check_disk_status(self):
        disk = psutil.disk_usage('/')
        print(f"Заповнено місце на диску: {disk.percent}%")
        if disk.percent > 90:
            print("Попередження: Недостатньо місця на диску.")
```

Файл email_reports.py

```
import smtplib
from email.mime.text import MIMEText

# Клас EmailReports для надсилання звітів електронною поштою
class EmailReports:

    def __init__(self, smtp_server, smtp_port, sender_email, sender_password):
        self.smtp_server = smtp_server
        self.smtp_port = smtp_port
        self.sender_email = sender_email
        self.sender_password = sender_password

    # Метод для надсилання звіту
    def send_report(self, recipient_email, subject, report_content):
        try:
            msg = MIMEText(report_content)
            msg['Subject'] = subject
            msg['From'] = self.sender_email
            msg['To'] = recipient_email

            with smtplib.SMTP(self.smtp_server, self.smtp_port) as server:
                server.starttls()
                server.login(self.sender_email, self.sender_password)
                server.sendmail(self.sender_email, recipient_email,
                                msg.as_string())
                print(f"Звіт успішно надісланий на {recipient_email}.")
        except Exception as e:
            print(f"Помилка при надсиланні звіту: {str(e)}")
```

Файл `usb_control.py`

```
import psutil

# Клас USBControl для моніторингу підключених USB-пристроїв
class USBControl:

    def __init__(self):
        self.connected_devices = []

    # Метод для перевірки підключених пристроїв
    def monitor_usb_devices(self):
        print("Перевірка підключених USB-пристроїв...")
        devices = psutil.disk_partitions()
        for device in devices:
            if 'removable' in device.opts:
                if device.device not in self.connected_devices:
                    self.connected_devices.append(device.device)
                    print(f"Новий USB-пристрій виявлено: {device.device}")
```

КБПЗ_2025

Файл client_gui.py

```
from tkinter import *

# Клас ClientGUI для створення графічного інтерфейсу клієнтської програми
class ClientGUI:

    def __init__(self, root):
        self.root = root
        self.root.title("Клієнтська програма")
        self.create_widgets()

    # Метод для створення елементів інтерфейсу
    def create_widgets(self):
        Label(self.root, text="Введіть ім'я користувача:").pack(pady=5)
        self.username_entry = Entry(self.root)
        self.username_entry.pack(pady=5)

        Label(self.root, text="Введіть пароль:").pack(pady=5)
        self.password_entry = Entry(self.root, show="*")
        self.password_entry.pack(pady=5)

        Button(self.root, text="Увійти", command=self.login).pack(pady=10)

    # Метод для обробки події входу
    def login(self):
        username = self.username_entry.get()
        password = self.password_entry.get()
        print(f"Спроба входу з ім'ям: {username} і паролем: {password}")
```

Файл Extended.py

```

import os
import pandas as pd
import re
import time
import schedule
import smtplib
from email.mime.text import MIMEText
from datetime import datetime, timedelta

class ExtendedSystem:

    def __init__(self):
        self.data_storage = []
        self.search_results = []
        self.anomaly_logs = []
        self.language_support = ["English", "Ukrainian"]
        self.document_control_logs = []
        self.plan_security_checks()

    def advanced_search(self, search_term):
        if not search_term:
            print("Попередження: Пошуковий запит не може бути порожнім.")
            return
        print(f"Початок пошуку для запиту: {search_term}")
        for record in self.data_storage:
            if re.search(search_term, record, re.IGNORECASE):
                self.search_results.append(record)
                print(f"Знайдено: {record}")
        if not self.search_results:
            print("Нічого не знайдено для запиту.")

    def add_record(self, record):
        if not record:
            print("Помилка: Порожній запис не може бути доданий.")
            return
        self.data_storage.append(record)
        print(f"Запис '{record}' успішно додано до сховища даних.")

    def filter_data(self, keyword):
        print(f"Фільтрація даних за ключовим словом: {keyword}")
        filtered_data = [record for record in self.data_storage if keyword in
record]
        for record in filtered_data:
            print(f"Знайдений запис: {record}")
        return filtered_data

    def generate_reports(self):
        if not self.data_storage:
            print("Попередження: Немає даних для створення звіту.")
            return
        df = pd.DataFrame(self.data_storage, columns=["Data"])
        filename = f"report_{datetime.now().strftime('%Y%m%d_%H%M%S')}.xlsx"
        df.to_excel(filename, index=False)
        print(f"Звіт створено у файлі {filename}.")

```

```

def send_sms_notification(self, message, phone_number):
    print(f"Відправка SMS на номер {phone_number} з повідомленням:
{message}")

def anomaly_detection(self, activity_log):
    print("Початок аналізу для виявлення аномальної активності.")
    for log in activity_log:
        if "error" in log.lower() or "unauthorized" in log.lower():
            self.anomaly_logs.append(log)
            print(f"Аномальна активність виявлена: {log}")

def set_language(self, language):
    if language not in self.language_support:
        print(f"Помилка: Мова {language} не підтримується.")
        return
    print(f"Мова інтерфейсу змінена на {language}.")

def monitor_document_control(self, document_name, user_action):
    log_entry = f"Документ: {document_name}, Дія: {user_action}, Час:
{datetime.now()}"
    self.document_control_logs.append(log_entry)
    print(f"Зареєстровано дію з документом: {log_entry}")

def plan_security_checks(self):
    print("Планування автоматичних перевірок безпеки.")
    schedule.every().monday.at("08:00").do(self.perform_security_check)
    schedule.every().friday.at("18:00").do(self.perform_security_check)

def perform_security_check(self):
    print("Виконується автоматична перевірка безпеки.")
    self.anomaly_detection(self.data_storage)
    print("Перевірка безпеки завершена.")

def run_scheduler(self):
    print("Запуск планувальника подій.")
    while True:
        schedule.run_pending()
        time.sleep(1)

def email_notification(self, subject, message, recipient_email):
    try:
        msg = MIMEText(message)
        msg['Subject'] = subject
        msg['From'] = "system@example.com"
        msg['To'] = recipient_email
        with smtplib.SMTP("smtp.example.com", 587) as server:
            server.starttls()
            server.login("system@example.com", "password123")
            server.sendmail("system@example.com", recipient_email,
msg.as_string())
        print(f"Повідомлення надіслано на {recipient_email}.")
    except Exception as e:
        print(f"Помилка при надсиланні повідомлення: {str(e)}")

```