

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
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи для ідентифікації особи у
мережі в режимі реального часу на основі технології LFIS”

Виконав здобувач вищої освіти
IV курсу, групи КІ-23-МБ
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Мізякін Д.В.
«___» _____ 2026 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.М.
«___» _____ 2026 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
« 27 » лютого 2026 року

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

Мізякіну Данилі Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS*

2. Керівник роботи *Якименко Наталія Миколаївна, канд. фіз.-мат. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 126-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту 23.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS*

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 « 27 » лютого 2026 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	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Якименко Н.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Мізякін Д.В.
(прізвище та ініціали)

АНОТАЦІЯ

Мізякін Д.В. Програмне забезпечення системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

Метою розробки є програмне забезпечення системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

Результат роботи – програмна реалізація системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

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 HTML, ASP, Jscript, Visual C++.

Ключові слова: комп'ютерна інженерія, ідентифікації особи, мережа, LFIS

ABSTRACT

Miziakin D.V. Software for a real-time network personal identification system based on LFIS technology. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this issue of the qualification work for the first (bachelor's) level of higher education, software has been developed that is intended for a real-time network personal identification system based on LFIS technology.

The purpose of the development is software for a real-time network personal identification system based on LFIS technology.

The result of the work is a software implementation of a real-time network personal identification system based on LFIS technology.

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 PCs with Windows 10/11.

The program is developed in HTML, ASP, Jscript, Visual C++.

Keywords: computer engineering, personal identification, network, LFIS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	17
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	21
3.1 Опис функціонування системи	21
3.2 Розробка структурної схеми.....	26
3.3 Розробка функціональної схеми	28
3.4 Розробка діаграми процесів.....	31
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	34
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	34
4.2 Захист розробленого програмного забезпечення.....	58
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	60
6 ОСНОВНІ ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64

					ВКРБ-123.26.0041.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS	Літ.	Аркуш	Аркушів
Розроб.	Мізякін Д.В.					Б	1	71
Перев.	Якименко Н.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23-МБ		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

EOM	– електронно—обчислювальна машина
IT	– інформаційні технології
API	– прикладний програмний інтерфейс
AppWizard	– засоби автоматизованого створення додатків
CEBIT	– комп'ютерна виставка
FAR	– False Acceptance Rate
FRR	– False Rejection Rate
ISO/IEC	– стандарт
JTC1	– міжнародний комітет зі стандартизації в області IT
LFIS	– Live Face Identification System, це біометрична система розпізнавання осіб
Md5	– алгоритм шифрування
MFC	– Microsoft Foundation Class library
NIST	– національний інститут стандартизації
OLE	– технологія зв'язування й вбудовування об'єктів
PIN	– personal identification number
SC37	– спеціальний біопараметричний комітет
SSL	– Secure Sockets Layer
USB	– universal serial bus

ВСТУП

Актуальність теми. Розпізнавання осіб – це технологія використання на комп'ютері таких нововведень, як пошукові системи або програмне забезпечення, для ідентифікації людей або пошуку зі схожими особами. Сфери й області застосування технології розпізнавання досить об'ємні – це й відеоспостереження, маркетинг, рекламна область, здійснення платежів, медицина. Завдяки розвитку й ефективності вона використовується й застосовується для встановлення злочинців і допомагає у встановленні особистості людей. У сучасних системах відеоспостереження функція розпізнавання особи в режимі реального часу є однієї із ключових.

При спробі використовувати інтелектуальну відеоаналітику, засновану на русі, таку як: виявлення руху, відключення камери, вторгнення і т.д., для превентивної безпеки, користувачі часто дратуються через неправильні повідомлення. Ці спрацьовування виникають через те, що аналітика, заснована на русі, видає сигнал навіть тоді, коли в поле зору камери з'являються тварини або рухаються дерева через сильний вітер. В остаточному підсумку це приводить до того, що просто відключають цей механізм уже через кілька днів після установки. Навіть співробітники служби безпеки перестають звертати увагу на такі події.

Ідентифікація особи у відеоспостереженні гарантує, що повідомлення буде відправлятися тільки тоді, коли у відзначеній області присутня саме людина.

На сьогоднішній день, для втілення такого механізму, можуть застосовуватися два основні способи:

– Процес розпізнавання здійснюється на стороні камери, що знімає, і потім передаються мета-дані й відеопотік на сервер. Переваги методу – підключення великої кількості камер до медіа-сервера. Даний метод істотного навантаження на ресурси й продуктивність не виявить. Недоліки способу є, що буде потрібно встановлювати відеокамери одного виробника.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

– Інший метод. Спеціальне програмне забезпечення встановлюється на самому сервері. Тоді з камери спостереження приймається тільки відеопотік. Систему зручно експлуатувати у випадку, якщо відеоспостереження вже встановлене на об'єкті. Просто замінити застаріле серверне обладнання.

Заміна устаткування, покупка ліцензії на кожний канал – усе це коштує не «дуже» дешево. Але й самі камери із вбудованим детектором осіб мають чималу вартість. Також потрібно буде зробити заміну раніше встановлених відеокамер.

Для кожного конкретного випадку вибір індивідуальний.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

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

- Огляд існуючих систем для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.
- Дослідження системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.
- Програмна реалізація системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

По недавній оцінці компанії Mordor Intelligence, глобальний ринок систем і засобів розпізнавання осіб в 2025 р. вийшов на оцінку 4,4 млрд дол. США, і до 2028 р. аналітики очікували його росту із середньорічними темпами 17,6%. Експерти Marketsandmarkets трохи більш стримані, але в цілому підтверджують ці величини: 3.2 млрд дол. в 2025-м і ріст із CAGR = 16,6% до 2027 р.

Якість розпізнавання осіб останнім часом досяглося феноменальної результативності: приміром, серійні системи Thales у ході натурних випробувань, проведених Дирекцією науки й технологій американського Департаменту національної безпеки (US Homeland Security Science and Technology Directorate), точно ідентифікують 99,44% учасників експерименту, затрачаючи на кожний акт розпізнавання не більш 5 секунд. Іншими словами, усе було прекрасне, поки не гримнув коронавірус.

Тепер же Національний інститут стандартів і технологій США (National Institute of Standards and Technology, NIST) рапортує про вкрай невтішний для цього сегмента ІТ-ринку факт: носіння маски катастрофічно погіршує якість розпізнавання особи. Експерти NIST копітко досліджували майже 90 комерційно доступних алгоритмів розпізнавання, імplementованих у самих різних системах. Кожний із цих алгоритмів зазнавав величезних труднощів, зіставляючи зображення осіб з масками з них повністю відкритими оригіналами з вихідної бази. Навіть кращі серед алгоритмів розпізнавання демонстрували зниження результативності в межах від 5 до 50% – залежно від того, наскільки значну частину особи закривала маска.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1.2 Область застосування

По даним J'son & Partners Consulting, до кінця 2026 року частка технологій розпізнавання осіб у загальному обсязі українського біометричного ринку склала майже 50%, а протягом чотирьох років цей сегмент показував ріст на рівні 106,7% у рік. Зростаючий попит експерти зв'язують із технологічним проривом в області машинного навчання, а також з інтересом великих інвесторів до українських стартапам алгоритми, що розробляють, face recognition. Зараз технологія розпізнавання осіб одержує в Україні істотну підтримку з боку держави, у тому числі на рівні законодавства.

Краще всіх росте фінансовий сегмент, тому що існує законодавча база й безліч сценаріїв для підвищення заробітку за допомогою технології. Інший сегмент – це транспорт. Впровадження технології підвищує безпека і якість транспортних пасажиропотоків.

Конкурентну перевагу одержує той, хто має серйозну експертизу в розробці й впровадженні закінчених багатокомпонентних рішень. Необхідно адаптувати системи під індивідуальні вимоги проекту, а також інтегрувати рішення зі сторонніми продуктами й системами верхнього рівня підприємства. Також ідеться про можливість розроблювача забезпечити замовникові повноцінну технічну підтримку й подальший супровід проекту

Оскільки технологія розпізнавання осіб уже дійшла до такого рівня готовності, що її можна використовувати в комерційних проектах, багато компаній впроваджують подібні платформи – наприклад, усім відомий 3D-сканер особи Face ID від Apple уже використовується на сотнях мільйонів пристроїв. На сьогоднішній день аналогічні технології активно використовуються в системах комп'ютерного зору й відеоаналітиці

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Серед найпоширеніших сфер застосування можна виділити наступні:

Контроль доступу до об'єктів або систем

За даними агентства J'son & Partners, з 2014 року частка систем контролю доступу з розпізнаванням осіб виросла з 0,7% до 11%. Розпізнавання осіб використовується для розмежування доступу в технологічні зони, особливо у випадках, коли співробітники передають свої пропуски в тимчасове користування. Сучасні системи розпізнавання осіб оснащені технологією захисту від підміни осіб, яка автоматично відрізняє реальну особу від світлини. Це дає можливість відслідковувати випадки підміни власної особи чужою світлиною або проходу співробітника, що несе фото відсутнього колеги.

У ЖКГ-сфері технологія забезпечує доступ мешканців багатоквартирного будинку в під'їзд або приміщення. Для освітніх установ застосовується інтеграція з контрольно-пропускною системою, використання особи як додаткового ідентифікатора для контролю учнів, інтеграція з рейтинговою системою й кампусними картами студента й викладача. Рішення, засноване на ідентифікації по особі, у порівнянні з іншими біометричними модальностями, такими як райдужна оболонка, відбиток долоні або пальця, не вимагає прямого контакту користувача зі зчитувачем, при цьому процес максимально відкритий. Також обране рішення не вимагає заміни систем керування доступом, які побудовані на різних платформах, а виходить, і не припускає відповідних витрат. Застосування системи знизило ризики, пов'язані з людським фактором при наданні доступу в офіс, а також виключило несанкціонований прохід шляхом передачі пропуску. Також рішення підвищив зручність користувачів, як співробітників, так і гостей. Впровадження комплексу біометричної ідентифікації спростило подальшу автоматизацію процесу доступу відвідувачів в офіс – дозволило впровадити термінали самостійної реєстрації.

Виявлення порушників

МВС тестує камери-відеореєстратори з функцією розпізнавання осіб. Якщо випробування будуть успішними, функція розпізнавання з'явиться в

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

портативних відеореєстраторах, які вже застосовують поліцейські. Камера по розміру ледве менше рації й кріпиться на одяг людини. Вона аналізує відеопотік і виділяє з нього особи людей, дані про яких через інтернет (у пристрій можна вставити цим-карту) відправляє на сервер. У деяких країнах, наприклад, у США й у Китаї, у розпорядженні поліції є не тільки такі пристрої, але й розумні окуляри.

Для багатьох організацій, зокрема, для мережного ритейла, актуальне завдання – виявити людину з «чорного списку» (що раніше зробила крадіжку або підозрюваного в незаконних діях).

Визначення портрета покупця

Ціль системи в цьому сегменті – проаналізувати поведінка покупця (наприклад, на касі) і підвищити лояльність до мережі. Такі системи вже використовує 59% ритейлерів одягу у Великобританії. В 2017 році Walmart розробив технологію, яка допомагає визначити, наскільки покупець задоволений відвідуванням магазину. Якщо система виявить покупця з нещасною особою, вона подає сигнал про це співробітникам магазину. Провідні українські ритейлери також або тестують, або вже використовують розпізнавання осіб, часто купуючи «коробкові» рішення. Технологія розпізнавання осіб у десятки раз прискорила контроль планogramм, скоротила на 10% кількість людей, що йдуть із магазину без покупок, і на 20% – втрати магазинів. Керівництво вважає, що поліпшення позитивно вплинули на товарообіг. Система допомагала відслідковувати кількість людей у чергах, визначала самі відвідувані відділи в магазинах, а також розпізнавала особи, стать, вік і настрій покупців.

Ідентифікація в банківському секторі

З першого липня 2018 року українські банки почали збір біометричних даних клієнтів. Пройшовши ідентифікацію в системі, клієнт може за допомогою голосу й фотозображення віддалено відкрити рахунок або замовити випуск карти на порталі держпослуг.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Керування робочим часом

Інтерес у роботодавців до системи обліку робочого часу співробітників (СКРЧ) росте з кожним роком. Здійснення контролю відбувається за допомогою програми, вбудованої в планшет при вході в офіс або інше місце роботи.

Камера планшету фіксує час приходу й відходу співробітника, вносить інформацію в систему, яка при запиті формує звіт про присутність кожного зі співробітників на робочому місці. Такі системи особливо затребувані в тих роботодавців, які мають багато найманого персоналу, що працює позмінно й із гнучким графіком. Система дозволяє спростити всю паперову роботу й контроль, надаючи достовірну інформацію з відпрацьованого співробітниками часу.

Оплата послуг

У березні стартап ORBL представив прототип розробки з одним з європейських провайдерів платіжних систем на виставці eShow у Барселоні. Це кіоск для оплати товарів або послуг за допомогою особи. Для реєстрації необхідно скачати мобільний застосунок, за допомогою якого можна прив'язати банківську карту або електронний гаманець, а також сфотографуватися. При оплаті в кіоску кошти списуються з рахунку покупця. Технологія дозволяє зробити покупку, навіть якщо покупець за якимись причинами не може скористатися звичними методами оплати. А також знизити ризик користування картою третьою особою. Ще один успішний кейс – використання технології для придбання лотерейних квитків, продаж яких обмежена віковим цензом.

Прохід на стадіони

Керівництво звернулося до розроблювачів після масових безладь на стадіонах в 2013 році, коли вболівальники влаштовували бійки, вибігали на поле, кидали файери. Це загрожувало не тільки величезними штрафами й порожніми трибунами на матчах клубу, але й було небезпечно для інших глядачів. Після впровадження системи клуби змогли суттєво заощадити й більше заробляти. Хулігани стали розуміти, що покарання неминуче, система не забороняє, а запобігає.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Вокзали

Є технологічний рішення, з яким пасажиром взагалі не прийдеться діставати паспорт або посадковий талон в аеропорті або на вокзалі. Паперові документи замінює розпізнавання осіб, біометрія, ця практика вже поширена в Європі. Почати впровадження можна було б із внутрішніх рейсів, а далі масштабувати.

Розумне місто

Безпека – базовий, але не єдиний сценарій використання технологій розпізнавання осіб в інфраструктурі міст. Безпека виступає як базис і інфраструктура для інших сервісів – фінансових, транспортних і так далі. Дані в знеособленому форматі можна використовувати для оплати проїзду на транспорті, оптимізації пасажиропотоку, пошуку зниклих людей, безбар'єрному проході на заходи, в аеропорти, оптимізації руху, міських послугах, ЖКГ.

Доступ до іспитів

При використанні навчального порталу студенти інституту одержують доступ до матеріалів курсу й до здачі тестів і іспитів, тоді як раніше це було неможливо через використання системою стандартного механізму доступу по паролю. За даними розроблювача, близько 5% студентів намагалися використовувати треті особи для здачі іспитів, але система запобігла всім випадкам шахрайства. Прокторинг – процедура спостереження й контролю над дистанційним випробуванням, дуже розвинений у США, де прокторингові компанії працюють в основному з освітніми установами. В Україні університети тільки починають використовувати його при прийманні іспитів. Рішення верифікують особу за допомогою біометрії в режимі відеопотоку й аналізують поведінку людини перед монітором.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

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

Оскільки шахрайство стає дедалі складнішим, підприємствам потрібна біометрична автентифікація, яка не лише перевіряє особу, але й надійно підтверджує її в режимі реального часу, захищає компанію від різних атак та безперебійно функціонує для реальних користувачів.

Нижче наведено шість найнадійніших рішень для біометричної автентифікації у 2026 році. Кожне з них має різні сильні сторони; правильний варіант залежить від вашого профілю ризику, бази користувачів та потреб у дотриманні вимог.

iDenfy

iDenfy лідирує в рейтингу, поєднуючи біометричну автентифікацію (зіставлення обличчя, документа, живості) з перевіркою особи – все це в робочому процесі, який мінімізує тертя. Для компаній, яким потрібна як гарантія ідентифікації, так і фінансова легітимність, пропонує видатний пакет послуг.

Чому варто обрати iDenfy:

– Точне зіставлення обличчя та документа: Система з високою точністю порівнює селфі користувача з фотографією в його посвідченні особи (паспорті, водійських правах тощо), виявляючи невідповідності, такі як підроблені посвідчення особи або невідповідні фотографії, зменшуючи шахрайство на етапі входу.

– Виявлення живості: не просто вимагає селфі, а визначає, чи людина дійсно присутня (моргає, повертає голову тощо). Це блокує підробку фотографій чи відео.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

– Перевірка власності банківського рахунку: підтверджує, що користувач володіє банківським рахунком, який він має намір використовувати, або має законний зв'язок з ним. Це запобігає шахрайським транзакціям через рахунки, що належать іншим особам.

– Плавний інтерфейс користувача: зручність для мобільних пристроїв та веб-сайтів, чіткі інструкції для користувачів (як зробити фотографію документа, освітлення тощо), швидкі відповіді, що зменшують кількість відмов. Також підтримується регульований стандарт (KYC, AML тощо), що дозволяє компаніям отримувати відповідність вимогам та безпеку без шкоди для зручності використання.

Якщо ви розширюєтеся на міжнародний рівень або хочете спростити свою гру дотримання вимог, iDenfy – це золотий стандарт. Він безпечний, зручний у використанні та позбавляє вас перевантаження постачальниками.

Veriff

Veriff – це сильний вибір для компаній, яким потрібна швидкість і масштабованість. Він робить акцент на швидкому автоматизованому прийнятті рішень з резервною ручною підтримкою, завдяки чому більшість користувачів перевіряються за лічені секунди, але випадки, що потребують додаткової перевірки, обробляються належним чином.

Чому варто обрати Veriff:

– Високий рівень автоматизації: значна частина процесів верифікації вирішується автоматично – зіставлення обличчя + документа + активність – без втручання людини. Це пришвидшує адаптацію та зменшує витрати.

– Прийняття рішень у режимі реального часу: якщо всі перевірки пройдено успішно, користувача швидко схвалюють; якщо є якісь розбіжності, є чіткий шлях «потребує перевірки». Ця чіткість допомагає кінцевим користувачам та внутрішнім командам зрозуміти, чому щось затримується.

– Потужна мобільна оптимізація: добре працює на різних пристроях; хороший зворотний зв'язок щодо проблем із камерою, освітленням тощо, що

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

означає менше помилкових відхилень через помилку користувача.

– Масштабована інфраструктура: розроблена для високої пропускну здатності; багато клієнтів повідомляють про здатність обробляти піки потреби в перевірці (під час запусків, стрибків трафіку тощо) без шкоди для точності.

Entrust IDV

Пропозиція Entrust IDV популярна серед компаній, які бажають забезпечити надійну біометричну автентифікацію та перевірку особи з глобальним охопленням. Вона використовується в різних галузях – від фінтех до подорожей – де підтвердження особи є важливим для довіри, але не менш важливим є і користувацький досвід.

Чому варто обрати Entrust:

– Різноманітність документів та глобальне покриття: підтримує широкий спектр міжнародних посвідчень особи, що важливо для компаній, що працюють у багатьох країнах. Користувачі на ринках, що розвиваються, часто мають менш поширені типи документів; це допомагає їм враховувати їх потреби.

– Обличчя + активність + перевірка документа: поєднує всі три перевірки, щоб шахраї не могли прослизнути з підробленим документом або підробленим обличчям. Багатошаровість додає надійності.

– Налаштовувані профілі ризику: дозволяють компаніям налаштовувати суворість перевірки залежно від контексту, наприклад, користувачів з низьким або високим рівнем ризику, геолокації та суми транзакцій. Це допомагає збалансувати безпеку та труднощі.

– Надійні інструменти для забезпечення відповідності та аудиту: Надає детальні журнали, чіткі обґрунтування рішень та підтримку правових/нормативних стандартів (KYC, AML, GDPR тощо). Допмагає компаніям проходити аудити та проявляти належну перевірку.

Plaid

Plaid часто відомий завдяки можливості підключення до банків, але його біометричні компоненти (у поєднанні з процесами перевірки особи) роблять його

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

переконливим варіантом для компаній, яким потрібна швидка та безпечна адаптація або авторизація платежів.

Чому варто обрати Plaid:

– Швидке прив'язування облікових записів із перевіркою особи: Коли користувачі прив'язують свої банківські рахунки, Plaid дедалі більше гарантує перевірку особи власника облікового запису за допомогою перевірки документів або обличчя, що зменшує шахрайство під час налаштування платежів або АСН-переказів.

– Зручний користувацький досвід: мінімальна кількість кроків, що означає, що користувачі часто з'єднуються зі своїм банком та проходять верифікацію за один раз, що покращує коефіцієнти конверсії.

– Потужні інструменти для розробників: хороші API, зрозумілі SDK, корисні цикли зворотного зв'язку (наприклад, якщо документ відхилено, інструкції щодо виправлення). Це пришвидшує інтеграцію та мінімізує кількість запитів на підтримку.

– Гарна відповідність нормативним вимогам: Розроблено з урахуванням ризиків; підходить для галузей, де важлива ідентифікація та точність платежів. Добре працює з додатковими системами KYC/KYB для забезпечення відповідності вимогам.

Plaid ідеально підходить для стартапів або платіжних додатків, яким потрібні швидкі, масштабовані банківські з'єднання з акцентом на задоволенні клієнтів. Якщо ви цінуєте швидкість та зручність розробки, Plaid саме для вас.

LexisNexis

LexisNexis – це не просто «постачальник послуг розпізнавання обличчя/живості» – їхня біометрична автентифікація є частиною ширшого комплексу послуг: ідентифікація, платежі, санкції, ризики ланцюга поставок тощо. Якщо вам потрібна біометрія плюс усе інше, вони пропонують цілісну платформу.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Чому варто обрати LexisNexis:

– Інтегроване шарування ризиків: Перевірки обличчя/активності поєднуються з верифікацією особи, верифікацією облікового запису, перевіркою на наявність санкцій та перевіркою негативних ЗМІ. Це допомагає виявляти складне шахрайство, яке проявляється за кількома векторами.

– Сильні сторони у сфері регулювання та дотримання вимог: вагомий досвід роботи в регульованих галузях – банківській справі, страхуванні тощо. Повна документація, журнали аудиту, контроль політик, захист конфіденційності.

– Зв'язок одержувача платежу з банківським рахунком: допомагає забезпечити не лише ідентифікацію особи, але й правильне надсилання платежів законних користувачів (через перевірені реквізити рахунку) та їх відсутність у разі переадресації.

– Адаптивні варіанти впровадження: пропонує інтеграцію API, веб-портал та гнучке налаштування суворості перевірки на основі ризику або вартості транзакції. Це допомагає компаніям налаштувати, наскільки суворими або обмеженими мають бути їхні біометричні перевірки залежно від контексту.

LexisNexis особливо ефективний для підприємств та транснаціональних корпорацій, де вартість одного шахрайського платежу може бути руйнівною. Поєднуючи перевірку банківського рахунку з перевірками ідентифікації та відповідності, вони забезпечують впевненість у тому, що платежі є точними, безпечними та законними.

Jumio

Jumio залишається основним вибором для компаній з високими вимогами до дотримання нормативних вимог, великими базами користувачів та потребами в надійній ідентифікації та біометричній автентифікації. Він добре підходить для регульованих середовищ, де точність є дуже важливою.

Чому варто обрати Jumio:

– Ретельна перевірка документів, обличчя та біометричних даних: висока

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

точність, широка бібліотека документів та перевірки на актуальність, стійкі до підробки.

- Висока надійність транзакцій: обробляє велику кількість користувачів без зниження продуктивності; хороша продуктивність, що означає швидку перевірку, навіть коли в черзі багато запитів.

- Комплексні журнали аудиту та управління: детальна реєстрація рішень щодо перевірки, можливості ручного перегляду та потужні інструменти нагляду допомагають у регульованих секторах (фінанси, охорона здоров'я, уряд).

- Глобальні операції: Працює в багатьох країнах та підтримує різні мови та шрифти, що допомагає користувачам з різних юрисдикцій відчувати підтримку та зменшує труднощі, коли документи відрізняються за форматом.

Як вибрати правильного постачальника біометричної автентифікації

Навіть чудове програмне забезпечення допомагає лише тоді, коли воно відповідає потребам вашого бізнесу. Ось практичні фільтри, які можна використовувати:

- Ризик проти тертя: Вищий рівень безпеки (глибші біометричні перевірки) часто означає більше тертя з боку користувачів. Знайдіть постачальника, який дозволяє адаптивну верифікацію – легкий біометричний потік для низькоризикових транзакцій, потужніший потік для транзакцій з вищим ризиком.

- Пристрій користувача та географія: якщо ваші користувачі переважно мобільні або знаходяться в регіонах зі старими пристроями чи змінним інтернетом, оберіть варіанти, відомі мобільними SDK та хорошою толерантністю до різноманітної якості зображення.

- Конфіденційність, зберігання даних та відповідність вимогам: перевірте, як зберігаються біометричні дані, як захищені шаблони, чи відповідає система місцевим законам про конфіденційність (GDPR, CCPA тощо). Підприємства та регульовані сектори часто потребують вищих стандартів.

- Продуктивність та рівень помилкових відхилень: вам потрібен низький

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

рівень помилкових відхилень (блокування хороших користувачів) та низький рівень помилкових прийняття (прослизання шахраїв). Запитуйте статистику або пробні запуски. Краще використовувати постачальників, які публікують показники упередженості/успіху.

– Підтримка та масштабованість: Наскільки добре підтримується продукт (обслуговування клієнтів, документація)? Чи може він масштабуватися без перебоїв, коли обсяг верифікацій різко зростає?

Висновок

Вибір правильного сервісу верифікації у 2026 році залежить від потреб вашого бізнесу – чи то iDenfy, Plaid, чи інші постачальники послуг. Кожен із цих сервісів пропонує щось особливе, від технологій штучного інтелекту до глобальної відповідності або гнучкості, зручної для стартапів. Оберіть той, який відповідає вашому стилю, і ви захистите себе від шахрайства, одночасно полегшуючи життя своїм клієнтам.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для реалізації програми мною була використана мова програмування Visual C++. У зв'язку з тим, що сьогодні рівень складності програмного забезпечення дуже високий, розробка застосунків Windows з використанням тільки якої-небудь мови програмування значно утрудняється. Програміст повинен затратити масу часу на рішення стандартних завдань по створенню багатовіконного інтерфейсу. Реалізація технології зв'язування й вбудовування об'єктів – OLE – зажадає від програміста ще більш складної роботи. Щоб полегшити роботу програміста практично всі сучасні компілятори з мови C++ містять спеціальні бібліотеки класів. Такі бібліотеки містять у собі практично весь програмний інтерфейс Windows і дозволяють користуватися при програмуванні засобами більш високого рівня, чим звичайні виклики функцій. За

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

рахунок цього значно спрощується розробка застосунків, що мають складний інтерфейс користувача, полегшується підтримка технології OLE і взаємодія з базами даних. Сучасні інтегровані засоби розробки застосунків Windows дозволяють автоматизувати процес створення застосунка. Для цього використовуються генератори застосунків. Програміст відповідає на питання генератора застосунків і визначає властивості застосунка – чи підтримує воно багатовіконний режим, технологію OLE, тривимірні органи керування, довідкову систему. Генератор застосунків, створить застосунок, що відповідає вимогам, і надасть вихідні тексти. Користуючись їм як шаблоном, програміст зможе швидко розробляти свої застосунки. Подібні засоби автоматизованого створення застосунків включені в компілятор Microsoft Visual C++ і називаються MFC AppWizard. Заповнивши кілька діалогових панелей, можна вказати характеристики застосунка й одержати його тексти, постачені великими коментарями. MFC AppWizard дозволяє створювати одновіконні й багатовіконні застосунки, а також застосунки, що не мають головного вікна, – замість нього використовується діалогова панель. Можна також включити підтримку технології OLE, баз даних, довідкової системи. Звичайно, MFC AppWizard не всесильний. Прикладну частину застосунка програмістові прийдеться розробляти самостійно. Вихідний текст застосунка, створений MFC AppWizard, стане тільки основою, до якої потрібно підключити інше. Але працюючий шаблон застосунка – це вже половина всієї роботи. Вихідні тексти застосунків, автоматично отриманих від MFC AppWizard, можуть становити сотні рядків тексту. Набір його вручну був би дуже стомлюючий. Потрібно відзначити, що MFC AppWizard створює тексти застосунків тільки з використанням бібліотеки класів MFC (Microsoft Foundation Class library). Тому тільки вивчивши мову C++ і бібліотеку MFC, можна користуватися засобами автоматизованої розробки й створювати свої застосунки в найкоротший термін. Як уже згадувався, MFC – це базовий набір (бібліотека) класів, написаних мовою C++ і призначених для спрощення й прискорення процесу програмування для Windows. Бібліотека містить багаторівневу ієрархію

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

класів, що нараховує близько 200 членів. Вони дають можливість створювати Windows-застосунки на базі об'єктно-орієнтованого підходу. З погляду програміста, MFC являє собою каркас, на основі якого можна писати програми для Windows. Бібліотека MFC розроблялася для спрощення завдань, що стоять перед програмістом. Як відомо, традиційний метод програмування під Windows вимагає написання досить довгих і складних програм, що мають ряд специфічних особливостей. Зокрема, для створення тільки каркаса програми таким методом знадобиться близько 75 рядків коду. У міру ж збільшення складності програми її код може досягати воістину неймовірних розмірів. Однак та ж сама програма, написана з використанням MFC, буде приблизно в три рази менше, оскільки більшість приватних деталей приховано від програміста.

Одною з основних переваг роботи з MFC є можливість багаторазового використання того самого коду. В зв'язку з тим, що бібліотека містить багато елементів, загальних для всіх Windows-застосунків, немає необхідності щораз писати їх заново. Замість цього їх можна просто успадковувати (говорячи мовою об'єктно-орієнтованого програмування). Крім того, інтерфейс, забезпечуваний бібліотекою, практично незалежний від конкретних деталей, його що реалізують. Тому програми, написані на основі MFC, можуть бути легко адаптовані до нових версій Windows (на відміну від більшості програм, написаних звичайними методами). Ще однією істотною перевагою MFC є спрощення взаємодії із прикладним програмним інтерфейсом (API) Windows. Будь-який застосунок взаємодіє з Windows через API, що містить кілька сотень функцій. Значний розмір API утрудняє спроби зрозуміти й вивчити його цілком. Найчастіше навіть складно простежити, як окремі частини API зв'язані один з одним! Але оскільки бібліотека MFC поєднує (шляхом інкапсуляції) функції API у логічно організовану безліч класів, інтерфейсом стає значно легше управляти.

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 для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

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

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-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

2D ідентифікація особи

Більшість програм і технології ідентифікації опираються на 2D зображення. Але це зроблене не тому, що 2D-зображення особи дуже точне, а просто для зручності.

Переважає більшість камер одержують картинку без якої-небудь глибини. І загальнодоступні світлини, які можна використовувати для баз даних, усі 2D.

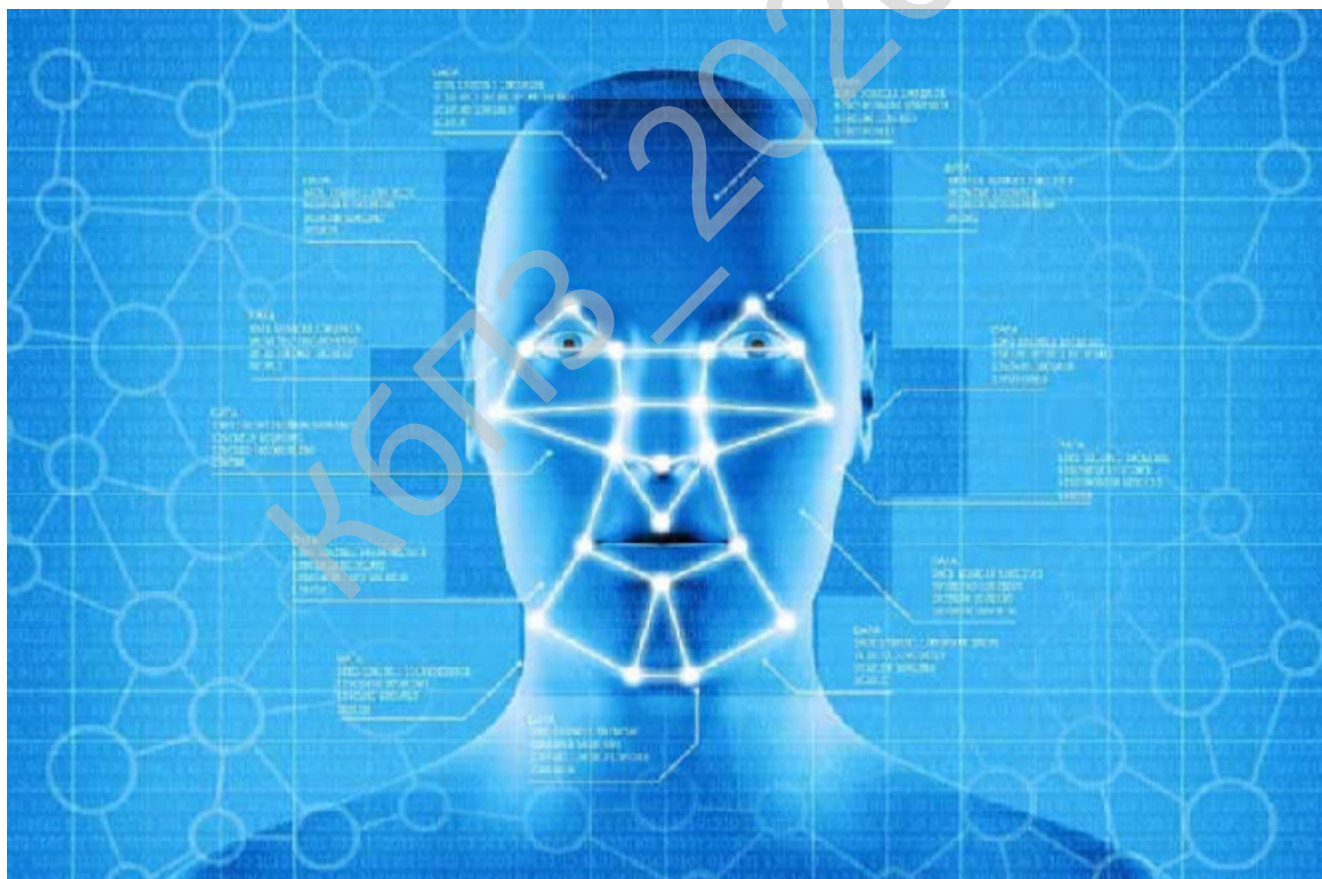


Рисунок 3.1 – Ідентифікація особи – 2D розпізнавання осіб у реальному часі

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Чому 2D-зображення особи не є точним?

Плоский знімок особи випробовує недолік в ідентифікуючих особливостях, таких як глибина. За допомогою плоского зображення комп'ютер може вимірювати, наприклад, відстань між зіницями й ширину рота. Але він не може визначити довжину або опуклість чола.

Крім того, отриманий знімок сильно залежить від спектра видимого світла. Це означає, що двомірна візуалізація особи не працює в темряві. Така технологія ідентифікації в реальному часі може бути ненадійною при яскравому або темному висвітленні.

Для обходу цих недоліків треба використовувати інші методи.

3D ідентифікація особи

Технологія, що з'явився не дуже давно, ідентифікації осіб опирається на 3D-моделювання. Така модель забезпечує більшу точність.

Для одержання х-мірного зображення поверхні особи людину в режимі реального часу використовуються відмітні риси. Де найбільш очевидні кості й цупкі тканини, наприклад вигини очниць, підборіддя або носа. Такі області особи не міняються згодом, вони – унікальні.

Використовуючи вісь виміру й глибину, на яку не впливає висвітлення, 3D-Ідентифікація особи в реальному часі може застосовуватися в темряві. Дозволяє визначати об'єкт під різними кутами огляду з можливістю розпізнавання до 90 градусів (особа в профіль).

Застосовуючи ПЗ для 3D, розпізнавання осіб зазнає серії проміжних кроків для перевірки особистості людину:

Розпізнавання осіб – виявлення

Одержання зображення може бути виконане шляхом оцифровки існуючої світлини (2D) або з використанням відео для одержання живого зображення об'єкта (3D).

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22



Рисунок 3.2 – Ідентифікація особи – 3D розпізнавання осіб у реальному часі

Центрування

Як тільки система виявляє особу, визначається положення голови, розмір і поза. Як вказувалося раніше, об'єкт може бути розпізнаний до 90 градусів. У той час як у режимі 2D голова повинна бути повернена як мінімум на 35 градусів у напрямку камери.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Вимір

Потім вимірюються криві грані в міліметровому (або мікрохвильовому) масштабі й створюється шаблон.

Представлення

Система переводить шаблон в унікальний код. Це кодування дає кожному шаблону набір чисел для вистави особливостей особи суб'єкта.

Узгодження

Якщо знімок є тривимірним, а база даних містить тривимірні зображення, то зіставлення буде виконано без яких-небудь змін.

Однак у цей час існує проблема, з якої зустрічаються інформаційні бази, що перебувають в 2D-зображеннях. 3D забезпечує живий змінний об'єкт, що рухається, у порівнянні із плоским стійким знімком.

Нові технології вирішують цю проблему. При зйомці тривимірного об'єкта вираховуються різні точки (звичайно три). Наприклад, зовнішня й внутрішня частина ока й кінчик носа будуть визначені й обмірювані. Як тільки ці виміри будуть виконані, до картинки буде застосований алгоритм (покрокова процедура), щоб перетворити його у двомірне. Після перетворення програмне забезпечення буде порівнювати отриманий результат з 2D-зображеннями в базі, щоб знайти потенційний збіг.

Перевірка й(або) ідентифікація особи

При перевірці, вихідна інформація зіставляється тільки одному результату в базі даних (1:1). Це робиться, щоб перевірити, чи є суб'єкт тем, ким він говорить. Якщо метою є ідентифікація, то кадр рівняється з усіма файлами бази, що дає оцінку для кожної потенційної відповідності (1: N).

Далі розглянемо, як біометрія шкіри може допомогти перевірити збіги.

Біометричні технології ідентифікації

Аналіз текстури шкіри виявився успішним інструментом для розпізнавання за рахунок значного підвищення дозволу зображення. Метод

використовує біометрію шкіри або унікальність текстури, для одержання ще більш точних результатів.

Процес працює майже так само, як розпізнавання осіб у реальному часі. Вибирається ділянка шкіри, називаний відбитком. Потім цей відбиток розбивається на більш дрібні блоки. Використовуючи алгоритми, щоб перетворити розглянута ділянка в математичне рівняння, система буде розрізняти будь-які лінії, пористість і фактичну текстуру. Згідно Identix, завдяки комбінації технології розпізнавання осіб з аналізом текстури шкіри, точність ідентифікації може збільшитися на 20-25 відсотків.

Розпізнавання осіб у реальному часі – тепловізійний аналіз

На сьогоднішній день дослідження з розпізнавання осіб в основному зосереджені на видимому спектрі. Вирішуючи такі проблеми, як зміни освітленості, поза й дозвіл зображення. Проте, для відеоспостереження в нічний час відсутність висвітлення не дозволяє ефективно використовувати камери, що працюють у спектрі видимого світла.

Тепловізійний аналіз вимірює випромінювання в інфрачервоному спектрі середньої довжини хвилі й довгохвильової частини спектра. Яке природно випромінюється живою тканиною, і тому є досить практичним способом візуалізації для роботи вночі. Однак, оскільки більшість баз даних і списків спостереження містять тільки результати видимого спектра, важко зіставити невідоме теплове зображення з набором відомих видимих знімків.

Технології розпізнавання й ідентифікації стають усе більш помітними в нашому світі. За останні кілька років ми стали свідками серйозних розробок у цій технології. Детектор осіб у відеоспостереженні – одна з новітніх розробок біометричних ідентифікаторів, яка не вимагає багато часу на обробку або втручання людини в перевірку.

Інші біометричні ідентифікатори, такі як сканери відбитків пальців і розпізнавання голосу, вимагають набагато більше різних пристроїв для функціонування.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

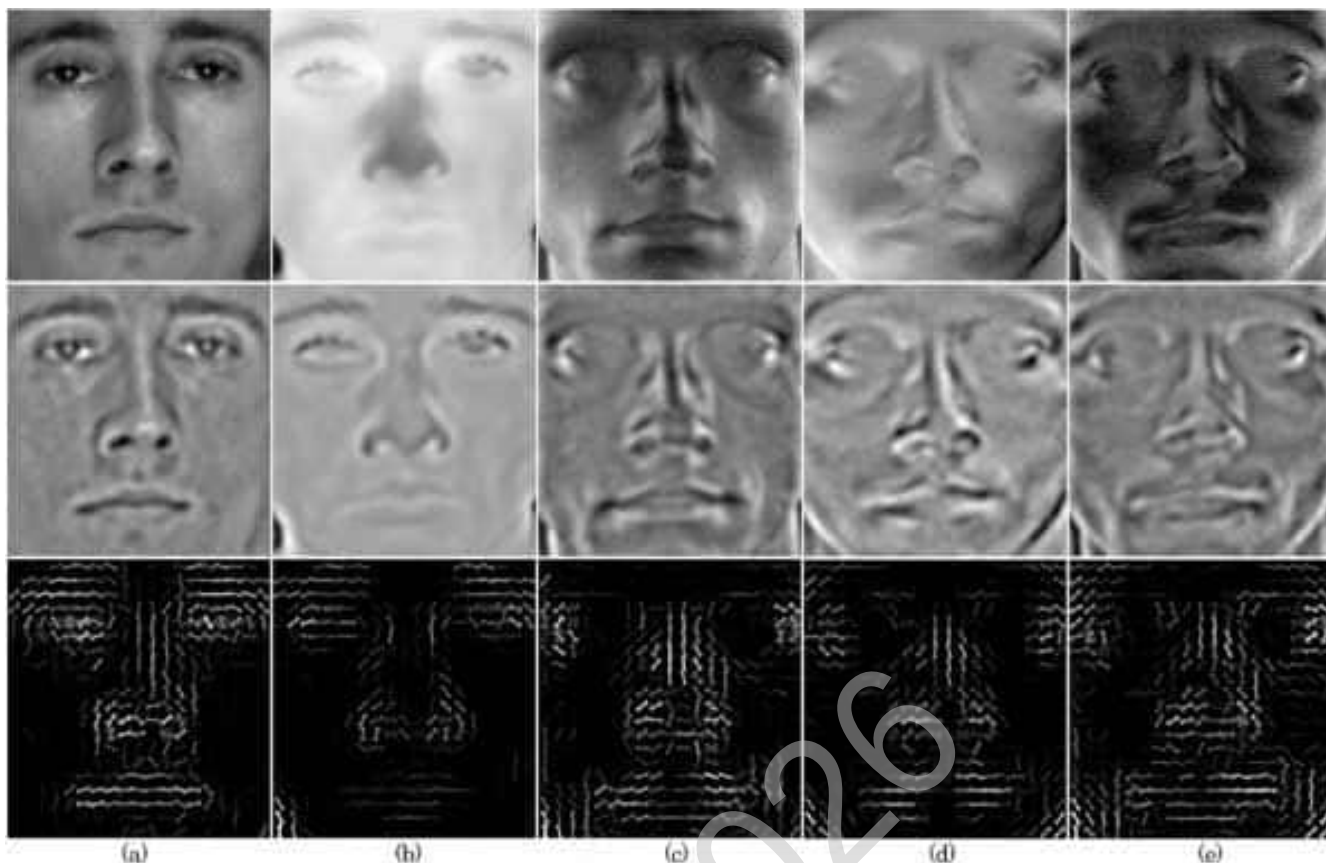


Рисунок 3.3 – Окремий об'єкт, видимий у видимому (а) і тепловому (b-e) спектрах

Розпізнавання осіб – високоефективна біометрична технологія ідентифікації, що володіє більшим потенціалом.

3.2 Розробка структурної схеми

LFIS – Біометричні системи: розпізнавання обличчя та відбитків пальців. Технології розпізнавання облич та біометричні технології мають безліч застосувань для ефективнішого вирішення проблем.

Цей тип технології використовує датчик або камеру, яка фіксує ідентифікаційні ознаки людини (морфологічні вимірювання, такі як відбиток пальця або зображення обличчя у 2D або 3D), які перетворюються на цифрові дані за допомогою алгоритму або спеціалізованого програмного забезпечення, а

потім порівнюються із запам'ятованими посиланнями для швидкої та ефективної ідентифікації людей.

Чи то спрощення процесу перевірки особи, спрощення, пришвидшення та підвищення безпеки автентифікації, чи створення персоналізованого досвіду для вашої діяльності, чи доповнення до вашого програмного рішення, ми можемо допомогти вам з вашим проектом розпізнавання облич.

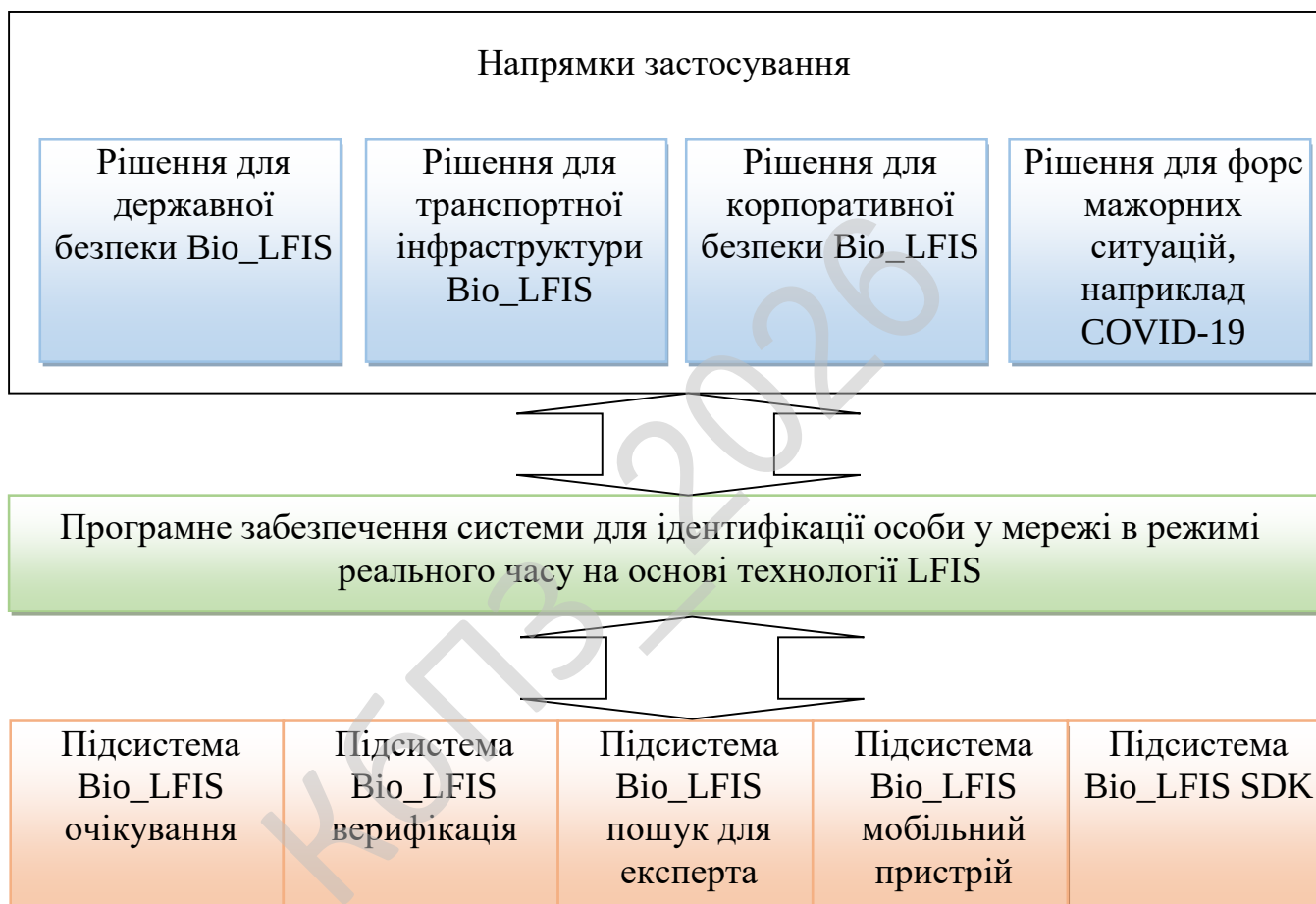


Рисунок 3.4 – Структурна схема системи

Програмне забезпечення для розпізнавання облич LFIS є передовим у сфері технологій та досягає чудових результатів, оскільки воно інтегрує найновіші розробки в галузі розпізнавання облич, такі як швидке відстеження облич, ідентифікація між наборами зображень (mp) та прискорення GPU, а також методи глибокого навчання, які покращують його з кожною взаємодією.

LFIS – це програмне забезпечення для біометричної розпізнавання обличчя, яке дозволяє ідентифікувати людей за допомогою камер і генерувати сповіщення в режимі реального часу, якщо обличчя відповідає особам, збереженим у списку спостереження.

Його можна використовувати з широким спектром камер з роздільною здатністю від 720p до 4K, він адаптується до кожного середовища та не потребує специфічного обладнання, яке збільшує кінцеву вартість. Він не потребує VMS, рекордерів чи додаткових баз даних.

Його можна інтегрувати за допомогою SDK та адаптувати до конкретної проблеми, щоб забезпечити біометричне рішення для обличчя: контроль присутності, контроль пропускної здатності тощо.

3.3 Розробка функціональної схеми

Сьогодні розпізнавання особи – це зручна й практична функція ідентифікації без пароля. Сама технологія відноситься до області застосування теорії розпізнавання образів, яка виникла значно раніше сучасних комп'ютерних систем. Розпізнавання образів – невід'ємна частина діяльності мозку. Тому в спектрі комп'ютерних дисциплін завдання розпізнавання ставляться до проблематики штучного інтелекту.

Алгоритм роботи технології розпізнавання осіб складається із двох етапів: ідентифікація (хто ця людина?) і верифікація (а чи той це людей, за яку він себе видає?). Послідовність дій звичайно така:

1. Виділення особи

Виділяється особа людини на зображенні.

2. Обчислення антропометричних крапок

Обчислюються антропометричні крапки. Система знаходить опорні крапки на особі, які визначають індивідуальні характеристики. Алгоритм обчислення характеристик різний для кожної із систем і є головним секретом

розроблювачів. Раніше основною опорною крапкою для алгоритмів були ока, але алгоритми еволюціонували й стали враховувати мінімум 68 крапок на особі (розташовані по контуру особи, визначають положення й форму підборіддя, око, носа й рота, відстань між ними).

3. Нормалізація особи

Проводяться додаткові перетворення зображення (усунення нахилу голови, корекція кольору особи і так далі) з метою одержання чіткого фронтального знімка.

4. Обчислення й аналіз дескрипторів

Обчислюється дескриптор – набір характеристик, що описують особу незалежно від сторонніх факторів (вік, зачіска, макіяж). Аналізуються спеціальні локальні ознаки, що характеризують, наприклад, текстуру певних областей на особі. Зіставлення різних дескриптерів дозволяє оцінити, ставляться чи два отримані зображення особи до тому самому людині.

5. Верифікація особи

Рівняється отриманий вектор особи (цифровий шаблон) з наявним у базі особами.

Методи розпізнавання осіб

Методи добування ознак умовно діляться на дві групи:, що використовують локальні й глобальні ознаки особи. При використанні локальних – алгоритм виділяє окремі частини (ока, ніс, рот і ін.) і вже по них розпізнає особа. При використанні глобальних – оперує з усім особою в цілому. Кількість існуючих методів виділення ознак і їх класифікації велике, але ті самі методи використовуються для виділення як локальних, так і глобальних ознак.

Один з найпоширеніших методів – нейромережевий. Використання згорточних нейронних мереж на відміну від «класичної» математики робить розпізнавання менш залежним від кооперативності поведінки людей, знижує вимоги до монтажу камер, поліпшує якість розпізнавання в широкому діапазоні зовнішніх умов, включаючи мінливу освітленість.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

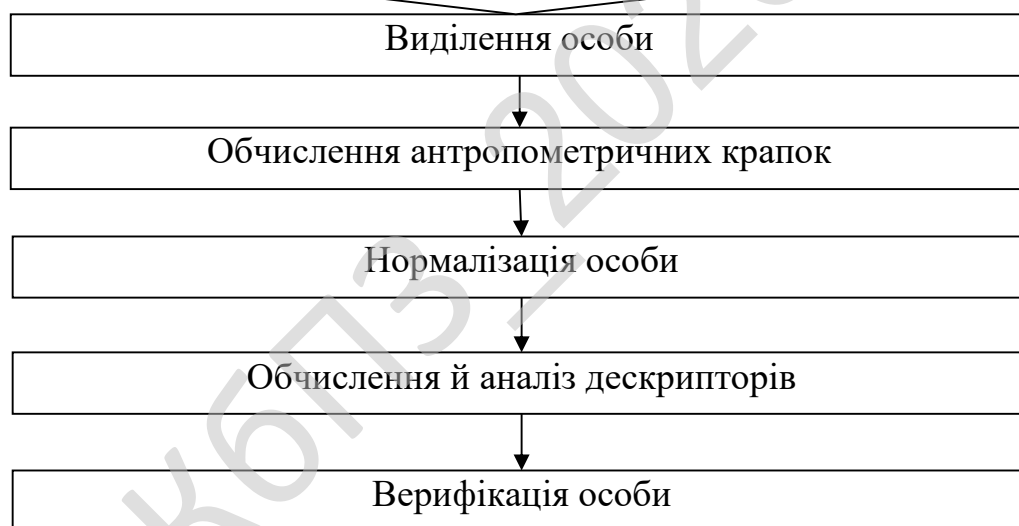
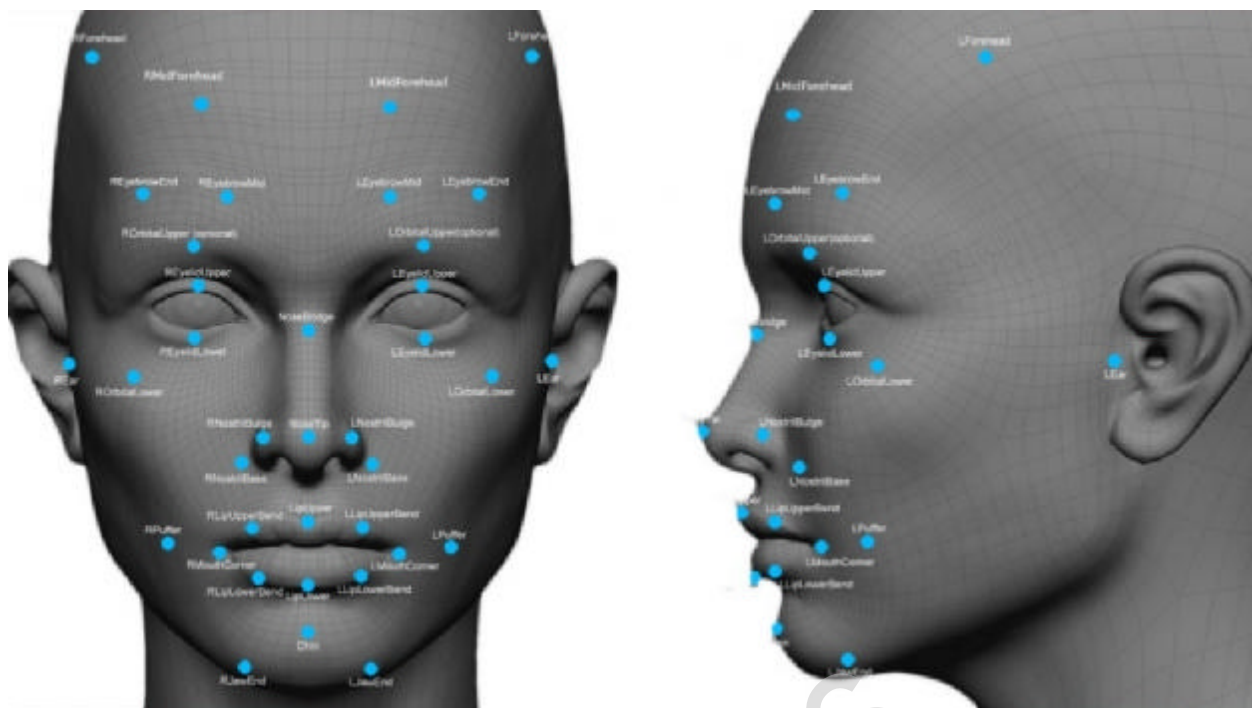


Рисунок 3.5 – Функціональна схема системи

Зображення особи з камери відправляється в систему, де обробляється біометричним алгоритмом. Спочатку картинка переводиться в масив крапок, потім він перетворюється в масив чисел, які обробляються математичними алгоритмами. На виході ми одержуємо біометричну модель – файл із певною послідовністю чисел.

Відновити вихідну світлинку людини з такого файлу не можна. Система просто щораз однаково перетворить картинку й порівнює отримані цифри. При цьому біометрична модель відрізняється від звичайного фотозображення: вона працює з відеопотоками (закодована послідовність кадрів) і знеособленою інформацією.

Технологія, заснована на надточних нейронних мережах, не вимагає дорогого устаткування й легко вбудовується в існуючі бізнеси-процеси. Розпізнавання відбувається шляхом обчислення відстані між дескрипторами осіб. Високоякісні дескриптори повинні показувати мала відстань між особами однієї людини й велика відстань між особами різних людей. Дескриптори можуть бути отримані за допомогою спеціальних алгоритмів, що перетворюють вхідні зображення у вектора значень.

3.4 Розробка діаграми процесів

Відповідно до методичних рекомендацій розроблення графічної частини кваліфікаційної бакалаврської роботи розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3.

Розроблена діаграма взаємодії процесів використовується для представлення та візуалізації процесів обробки даних тобто структурного проектування бакалаврської роботи.

Основні складові елементи діаграми взаємодії процесів це потоки даних:

- Репозиторії, потік сховища даних.
- Потоки зовнішні по відношенню до системи сутності.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Потоки даних гібридні між елементами трьох попередніх типів.

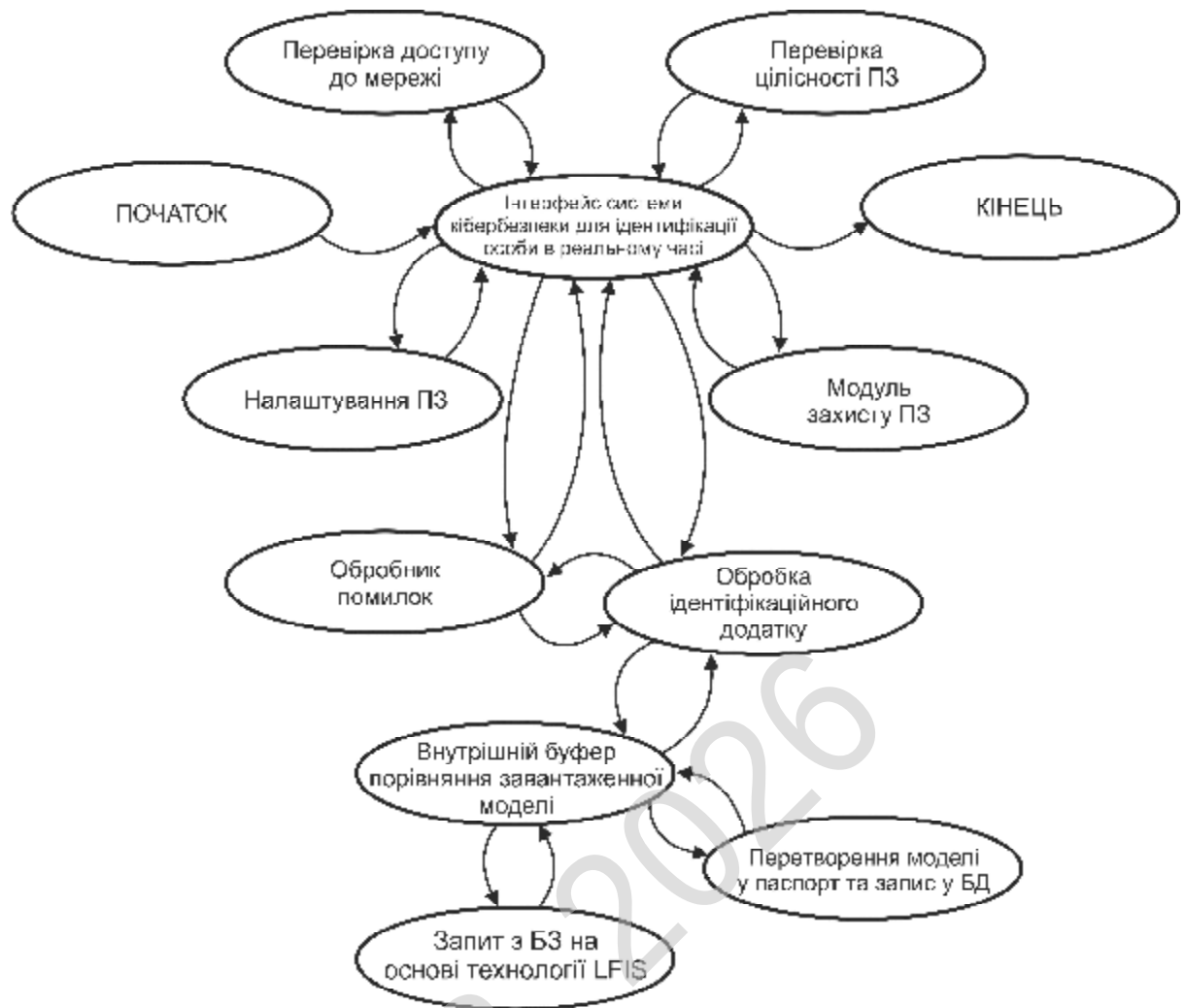


Рисунок 3.6 – Діаграма взаємодії процесів

Відповідно до документації основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Для схематичного представлення системи що розробляється необхідно спочатку представити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи в цілому у подальшому. Використовується

модель проектування, графічне представлення «потоків» даних в інформаційній системі. Розроблена діаграма взаємодії процесів системи в подальшому уточнюється шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Таким чином у результаті після розгляду, вищеописаної системи, схеми структурної, функціональної, діаграми взаємодії процесів перейдемо до опису та розгляду блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

К6ПЗ_2026

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

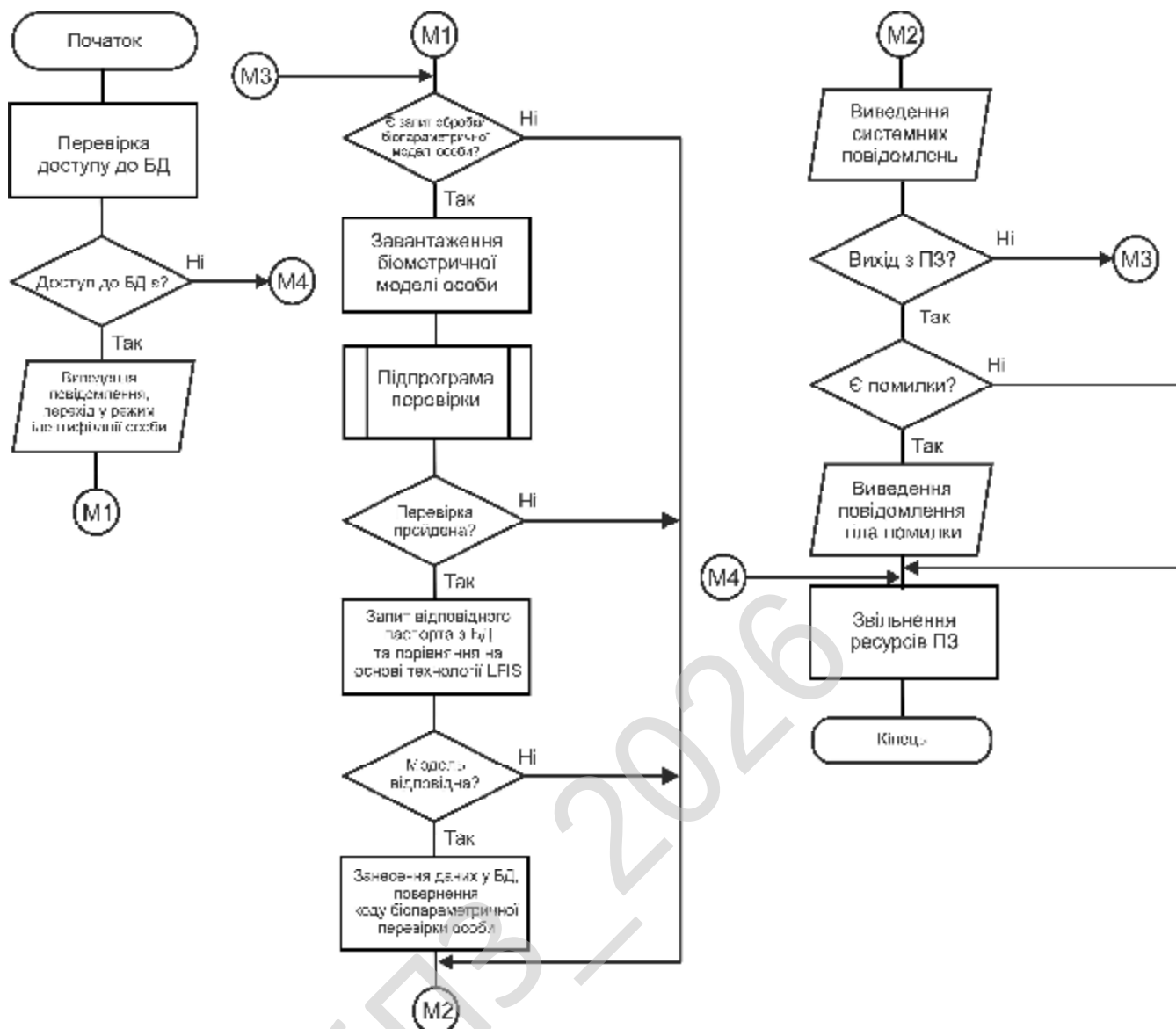


Рисунок 4.1 – Блок-схема основної програми

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як

інтерпретованого коду можлива кодогенерація. Було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів; Діаграма компонент.

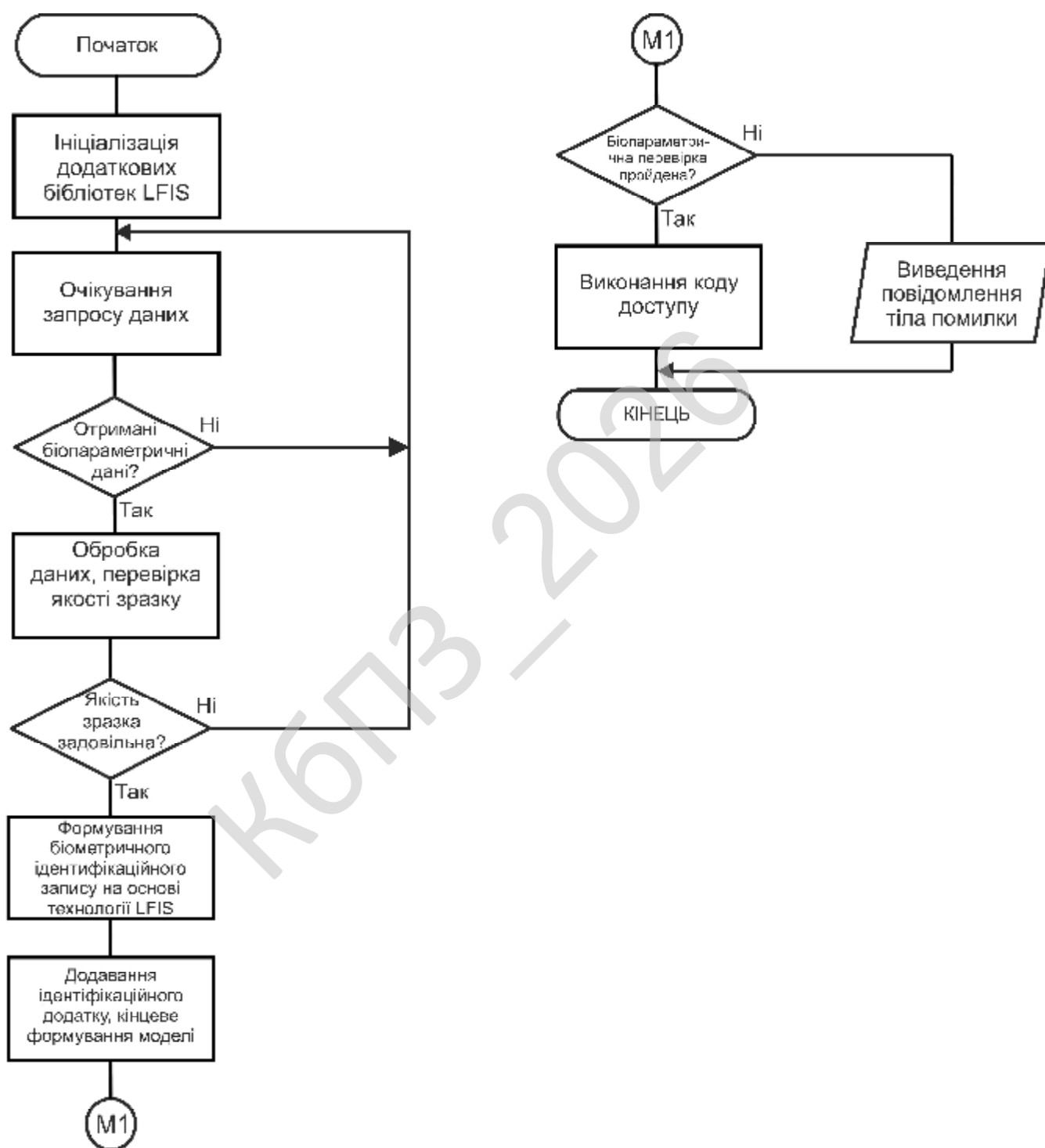


Рисунок 4.2 – Блок-схема роботи підпрограми

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.

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.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

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.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

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 відрізняти 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

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

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 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 зафарбованим ромбиком.

Діаграма компонент в UML це діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компонування, а деякі під час роботи програми.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

Компоненти об'єднуються разом використовуючи структурні зв'язки (assembly connector) щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія – «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

При використанні діаграми компонент щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

Система керування базами даних (СКБД, Database Management System, DBMS) – набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних. Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

Першим поколінням СКБД прийнято вважати ієрархічні й мережеві системи. Ці системи отримали широке поширення в 1970-х роках, а першою комерційною системою цього типу була система IMS компанії IBM.

У 1980-х роках ці системи були витіснені системами другого покоління – повсюдно використовуваними і донині реляційними СКБД. У цих системах використовувалися непроцедурні мови управління даними (SQL) і передбачався значний ступінь незалежності даних. Реляційні системи внесли значні удосконалення в управління даними: графічний користувацький інтерфейс (GUI), клієнт-серверні застосунки, розподілені бази даних, паралельний пошук даних та інтелектуальний аналіз даних.

Але вже до кінця 1980-х років існуюча тоді реляційна модель перестала задовольняти розробників в силу низки обмежень. Відповіддю на зростаючу складність програм баз даних стали два нових напрямки розвитку СКБД: об'єктно-орієнтовані СКБД і об'єктно-реляційні СКБД.

У 1991 був утворений консорціум ODMG, основною метою якого стало вироблення промислового стандарту об'єктно-орієнтованих баз даних. Між 1993 та 2001 роками ODMG опублікувала п'ять ревізій своїх специфікацій. Остання версія стандарту має індекс 3.0, після чого група розпустилася. До кінця 1990-х існувало близько десяти компаній, що виробляли комерційні продукти, що позиціонуються на ринку як ООСКБД. Найбільш відомими системами даного класу стали Objectivity, Versant виробництва однойменних компаній, а також СКБД Jasmine, випущена компанією CA. Незважаючи на переваги, що дозволяють ефективніше вирішувати певний ряд завдань, об'єктно-орієнтовані системи так і не змогли завоювати значущу частку ринку СКБД, залишившись «нішевим» продуктом.

Постачальниками традиційних реляційних СКБД також була проведена значна робота з об'єднання об'єктно-орієнтованих і реляційних систем. Розробники постаралися розширити мову SQL, щоб включити в неї концепції об'єктно-орієнтованого підходу, зберігаючи переваги реляційної моделі (об'єктні розширення мови SQL були зафіксовані в стандарті SQL:1999).

Основний принцип – це еволюційний розвиток можливостей СКБД без поломки попередніх підходів та зі збереженням наступності з системами попереднього покоління.

Поняття СКБД третього покоління, якими, власне кажучи, і є об'єктно-реляційні СКБД, з'явилося після опублікування групою відомих фахівців в області баз даних «Маніфесту систем баз даних третього покоління». Основні принципи СКБД третього покоління, позначені в маніфесті:

Крім традиційних послуг з управління даними, СКБД третього покоління повинні забезпечити підтримку розвиненіших структур об'єктів і правил. Розвинутіша структура об'єктів характеризує засоби, необхідні для зберігання і маніпулювання нетрадиційними елементами даних (тексти, просторові дані, мультимедіа). СКБД третього покоління повинні включити в себе СКБД другого покоління. Системи другого покоління внесли вирішальний вклад у двох

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

областях – непроцедурний доступ за допомогою мови запитів SQL і незалежність даних. Ці досягнення обов'язково повинні враховуватися в системах третього покоління.

СКБД третього покоління повинні бути відкриті для інших підсистем. Це включає оснащення різноманітними інструментами підтримки прийняття рішень, доступом з багатьох мов програмування, інтерфейсами до існуючих популярних систем і бізнес-застосунків, можливістю запуску програм з бази даних на іншій машині і розподілені СКБД. Весь набір інструментів і СКБД має ефективно функціонувати на різноманітних апаратних платформах з різними операційними системами.

Крім того, СКБД, що розраховує на широку сферу застосування, повинна бути оснащена мовою четвертого покоління (4GL).

У середині 1990 років було лише кілька дослідних прототипів СКБД, які поєднали найкращі риси реляційних і об'єктно-орієнтованих СКБД. Першим комерційним продуктом, якому були властиві об'єктно-реляційні риси, став Universal Server компанії Informix(згодом була поглинена IBM). В даний час більшість цих ідей вже втілено в реальних комерційних рішеннях, в тому числі і в продуктах основних постачальників СКБД (Oracle Database і IBM DB2).

Розвиток індустрії систем керування базами даних базується на значних фундаментальних наукових дослідженнях. Найчастіше, між самими дослідженнями та їхньою конкретною реалізацією в прикладних рішеннях минають роки, а іноді й десятиліття. Роботу в області управління даними проводять як університетські дослідницькі групи (MIT, Berkeley), так і центри розробок основних постачальників СКБД (Oracle, IBM, Microsoft). Інвестування в управління даними – це довгострокове, і разом з тим, вигідне вкладення коштів. В даний час дослідники мають у своєму розпорядженні засоби, що дозволяють ефективно реалізувати найскладніші запити, що маніпулюють терабайтами й петабайтами різних даних.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Основними тенденціями, які дали привід для проведення різних масштабних досліджень в області баз даних стали:

Експонентний ріст даних. Обсяг даних, у тому числі синтетичних, що генеруються автоматизованими системами, значно зріс. Збільшилося і число прикладних областей, в яких вимагається обробка великих обсягів даних. До таких областей тепер відносяться не тільки традиційні корпоративні програми, пошук у веб, але також і наукові дослідження, обробка природних мов, аналіз соціальних мереж тощо.

Значне ускладнення структур використовуваних даних. Прості види даних у вигляді чисел і символьних рядків стали доповнятися численною мультимедійною інформацією, просторовими, процедурними даними та великою кількістю інших складних форматів.

Широке поширення дешевих високопродуктивних апаратних засобів. Щорічно ми спостерігаємо зростання обчислювальних можливостей мікропроцесорів, збільшення ємності і зниження вартості доступних і зручних в експлуатації пристроїв дискової оперативної пам'яті.

Активний розвиток засобів комунікації та «всесвітньої павутини» World Wide Web. WWW стає єдиним інформаційним середовищем, що пронизує весь світ і об'єднує величезне число користувачів та електронних пристроїв.

Поява нових важливих областей застосування СКБД. У першу чергу, це пов'язано з інтелектуальним аналізом даних, сховищами даних, а останнім часом – з паралельними обчисленнями і хмарними технологіями.

Основні характеристики СКБД

1. Контроль за надлишковістю даних.
2. Несуперечливість даних.
3. Підтримка цілісності бази даних (коректність та несуперечливість).
4. Цілісність описується за допомогою обмежень.
5. Незалежність прикладних програм від даних.
6. Спільне використання даних.

7. Підвищений рівень безпеки.

Можливості СКБД

1. Дозволяється створювати БД (здійснюється за допомогою мови визначення даних DDL (Data Definition Language)).

2. Дозволяється додавання, оновлення, видалення та читання інформації з БД (за допомогою мови маніпулювання даними DML, яку часто називають мовою запитів)

3. Можна надавати контрольований доступ до БД за допомогою: Системи забезпечення захисту, яка запобігає несанкціонованому доступу до БД; Системи керування паралельною роботою прикладних програм, яка контролює процеси спільного доступу до БД; Система відновлення – дозволяє відновлювати БД до попереднього несуперечливого стану, що був порушений в результаті збою апаратного або програмного забезпечення.

Основні компоненти середовища СКБД

1. Апаратне забезпечення.

2. Програмне забезпечення.

3. Дані.

4. Процедури – інструкції та правила, які повинні враховуватись при проектуванні та використанні БД.

5. Користувачі: адміністратори даних (керування даними, проектування БД, розробка алгоритмів, процедур) та БД (фізичне проектування, відповідальність за безпеку та цілісність даних); розробники БД; прикладні програмісти; кінцеві користувачі.

Архітектура СКБД

Існує трирівнева система організації СКБД ANSI-SPARC, при якій існує незалежний рівень для ізоляції програми від особливостей представлення даних на нижчому рівні.

Рівні:

1. Зовнішній – представлення БД з точки зору користувача.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

2. Концептуальний – узагальнене представлення БД, описує які дані зберігаються в БД і зв'язки між ними. Підтримує зовнішні представлення, підтримується внутрішнім рівнем.

3. Внутрішній – фізичне представлення БД в комп'ютері.

Логічна незалежність – повна захищеність зовнішніх моделей від змін, що вносяться в концептуальну модель.

Фізична незалежність – захищеність концептуальної моделі від змін, які вносяться у внутрішню модель.

Було використано MongoDB – документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць.

MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії AGPLv3.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій.

У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

простим додаванням нових машин. При розробці автори виходили з необхідності спеціалізації баз даних, завдяки чому їм вдалося відійти від принципу «один розмір під усе». За рахунок мінімізації семантики для роботи з транзакціями з'являється можливість вирішення цілого ряду проблем, пов'язаних з нестачею продуктивності, причому горизонтальне масштабування стає простішим. Використовувана модель документів зберігання даних (JSON/BSON) простіше кодується, простіше управляється (у тому числі за рахунок застосування так званого безсхемного стилю (schemaless style), а внутрішнє угруповання релевантних даних забезпечує додатковий вииграш в швидкодії.

Нереляційний підхід досить зручний для створення баз даних, у яких горизонтальне масштабування означає розгортання на множині машин. Можливість забезпечувати найкращу продуктивність повинна існувати паралельно з підтримкою більшої функціональності, ніж це дозволяє використання пар «ключ-значення» (у чистому вигляді).

Технологія баз даних має працювати скрізь, починаючи з серверів користувача та віртуальних машин і закінчуючи хмарними технологіями.

MongoDB, на думку розробників, має заповнити розрив між простими сховищами даних типу «ключ-значення» (швидкими і легко масштабованими) і великими СКБД (зі структурними схемами і потужними запитами).

Основні можливості MongoDB:

1. Документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних).
2. Досить гнучка мова для формування запитів.
3. Динамічні запити.
4. Повна підтримка індексів.
5. Профілювання запитів.
7. Швидкі оновлення «на місці».
8. Ефективне зберігання бінарних даних великих обсягів, наприклад, фото та відео.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

9. Журналювання операцій, що модифікують дані в БД.

10. Підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг.

11. Може працювати відповідно до парадигми MapReduce.

СКБД управляє наборами JSON-подібних документів, що зберігаються в бінарному форматі BSON. Зберігання і пошук файлів в MongoDB відбувається завдяки викликам протоколу GridFS.

Подібно до інших документо-орієнтованих СКБД (CouchDB тощо), MongoDB не є реляційною СКБД.

Є докладна і якісна документація, велике число прикладів і драйверів для популярних мов Java, C++, C#, PHP, Python, Perl, Ruby.

При випуску одразу було заявлено, що реліз MongoDB 1.0 готовий до використання у виробництві як одиничний хост, так і у зв'язках master/slave.

Код цього релізу досить стабільний і перевірений в промисловій експлуатації протягом 1,5 років. MongoDB рекомендується розгортати мінімум на двох серверах використовуючи реплікацію Master/Slave.

Це забезпечує наявність актуальних даних при виході з ладу однієї з СКБД. MongoDB – продукт досить молодий, і відтак у ньому зустрічаються помилки, з'являються нові можливості тощо. Характерний високий темп розробки (проект пишуть не тільки волонтери, а й компанія людей на повній зайнятості). Компанія-розробник надає платні підтримку, хостинг, консультації.

Приклади запитів

Запити можуть витягати дані з вбудованих об'єктів та масивів. Якщо в колекцію users вставлений такий об'єкт:

```
{
  "username" : "bob",
  "address" : {
    "street" : "123 Main Street",
    "city" : "Springfield",
    "state" : "NY"
  }
}
```

Ми можемо запитати цей документ (і всі документи з адресою в Нью-Йорку) за допомогою:

```
> db.users.find({"address.state" : "NY"})
```

Можна також запитати елементи масиву:

```
> db.food.insert({"fruit" : ["peach", "plum", "pear"]})
```

```
> db.food.find({"fruit" : "pear"})
```

Jira – була використана комерційна система відслідковування помилок, призначена для організації взаємодії з користувачами, хоча в деяких випадках використовується і для управління проектами. Розроблено компанією Atlassian, є одним з двох її основних продуктів (поряд з вікі-системою Confluence). Має веб-інтерфейс.

Назва системи отримано шляхом усічення слова «Gojira» – Японського імені монстра Годзилла, що, в свою чергу, є відсиланням до назви конкуруючого продукту – Bugzilla; створювалася в якості заміни Bugzilla і багато в чому повторює її архітектуру. Система дозволяє працювати з декількома проектами. Для кожного з проектів створює і веде схеми безпеки і схеми оповіщення.

До версії 3.13.5 (включно) розрізнялися редакції Enterprise, Professional і Standard, після – Залишилася тільки редакція Enterprise (для великих організацій).

Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах.

Основний елемент обліку в системі – завдання (ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад – Фотографіями, скриншотами) або коментарями. Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настраюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором

статусів, одним або декількома видами уявлення (screens). Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, керування вимогами, керування ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce.

Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи українську, англійську, японську, німецьку, французьку, іспанську.

Для сторонніх розробників надаються кошти розробки розширень системи – плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатки. Ціноутворення залежить від максимального числа користувачів, при цьому близько \$50 за користувача для локального і \$7 на місяць за користувача для віддаленого доступу є типовими цінами.

Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника.

Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

- проект використовує ліцензії, схвалені Open Source Initiative;
- Вихідний код проекту доступний для скачування;
- у проекту є публічно доступна веб-сайт;

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

– програмне забезпечення від Atlassian є на веб-сайті проекту.

Хоча я реалізовував програму сам, було використано підходи Scrum для саморозвитку та пришвидшенню розробки, розглянемо цей метод. Scrum – підхід управління проектами для гнучкої розробки програмного забезпечення. Скрам чітко робить акцент на якісному контролі процесу розробки.

Підхід вперше описали Гіротака Takeучі та Ікуджіро Нонака в статті The New New Product Development Game (Гарвардський Діловий Огляд, січ–лют 1986). Вони відзначили, що проекти, над якими працюють невеликі, крос–функціональні команди, зазвичай систематично продукують кращі результати, і пояснили це, як «підхід регбі». У 1991 році Деґрейс та Шталь у книжці Злі проблеми, справедливі рішення послалися на цей підхід, як на Scrum (штовханина; сутичка навколо м'яча (у регбі)), спортивний термін, згаданий в статті Takeучі і Нонака. Кен Швабер на початку 1990–х використовував підхід який привів Scrum в його компанію.

Вперше метод Scrum було представлено на загальний огляд задокументованим, чітко сформульованим та описаним спільно Сазерлендом та Швабером на OOPSLA'96 в Остіні. Швабер та Сазерленд протягом наступних років працювали разом щоб обробити та описати весь їхній досвід та найкращі практичні зразки для індустрії в одне ціле, в ту методологію, що відома сьогодні як Scrum. Швабер об'єднав зусилля з Майком Бідлом в 2001, щоб детально описати метод в книжці Agile Software Development with SCRUM. Не зважаючи на те, що для Scrum нарікли долю управління проектами з розробки ПЗ, він може також використовуватися в роботі команд обслуговувань програмного забезпечення (software maintenance teams), або як підхід управління розробкою і супроводом програм: Scrum of Scrums.

Scrum – це кістяк процесу, який включає набір методів і попередньо визначених ролей. Головні дійові особи – ScrumMaster, той хто опікується процесами, веде їх і працює як керівник проекту, Власник Продукту, людина, що представляє інтереси кінцевих користувачів та інших зацікавлених в продукті

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

сторін, та Команду, яка включає розробників. Протягом кожного спринту, 15-30 денного періоду (тривалість визначається командою), працівники створюють функціональний ріст програмного забезпечення.

Набір можливостей, які імплементуються кожного спринту, приходять з етапу, що має назву product backlog (документація запитів на виконання робіт), який має найвищу пріоритетність за рівнем вимог до роботи, що повинна бути виконана.

Запити на виконання робіт (backlog items), що визначені протягом наради з планування спринту (sprint planning meeting), переміщуються в етап спринту. Протягом цієї наради Власник Продукту інформує про завдання, які він хоче, аби були виконані. Тоді Команда визначає, скільки з бажаного вони можуть зробити, щоб завершити необхідні частини протягом наступного спринту. Протягом спринту команда виконує визначений фіксований список завдань (т.з. backlog items). Впродовж цього періоду ніхто не має права змінювати перелік запитів на виконання робіт, що слід розуміти, як заморожування вимог (requirements) протягом спринту.

Product backlog – це документ, який має список вимог до функціональності, які упорядковані згідно зі ступенем важливості. Product backlog представляє список того, що повинно бути реалізовано. Елементи цього списку називається «історіями» (user story) або елементами backlog-у (backlog items). Product backlog відкритий для редагування усім учасникам Scrum-процесу.

Обов'язкові поля:

1. ID – унікальний ідентифікатор, порядковий номер, який використовується для ідентифікації історій у разі їх перейменування.

2. Назва (Name) – стислий опис історії. Він повинен бути однозначним, щоб і розробники і product owner могли зрозуміти, про що йдеться і відрізнити одну історію від іншої.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

3. Важливість (Importance) – ступінь важливості даної історії на погляд product owner 'а. Зазвичай являє собою натуральне число, іноді для цієї цілі використовуються числа Фібоначчі. Чим більше значення, тим більше пріоритет.

4. Попередня оцінка (initial estimate) – початкова оцінка об'єму робіт, необхідного для реалізації історії порівняно з іншими історіями. Вимірюється у story point'ах. Приблизно відповідає числу «ідеальних людино-днів».

5. Як продемонструвати (how to demo) – стисле пояснення того, як завершена задача буде продемонстрована у кінці спринта. Дане поле може являти собою код автоматизованого приймального тесту.

Додаткові поля. Іноді, також, використовуються додаткові поля у product backlog, в основному для того, щоб допомогти product owner'у визначитися з його пріоритетами.

Категорія (track). Наприклад, «панель управління» чи «оптимізація». За допомогою цього поля product owner може легко вибрати усі пункти категорії «оптимізація» і задати їм низький пріоритет.

Компоненти (components) – указує, які компоненти (наприклад, база даних, сервер, клієнт) будуть зачеплені при реалізації історії. Дане поле складається з групи checkbox'ів, які відмічаються, якщо відповідні компоненти потребують змін.

Ініціатор запиту (requestor). Product owner може захотіти зберігати інформацію про усіх замовників, зацікавлених у даній задачі. Це потрібно для того, щоб тримати їх у курсі діла про хід виконання робіт.

ID у системі обліку помилок (bug tracking ID) – якщо ви використовуєте окрему систему обліку помилок, тоді у описі історії корисно зберігати посилання на всі дефекти, які до неї відносяться.

Sprint backlog – містить функціональність, обрану Product Owner із Product Backlog. Всі функції розбиті по задачах, кожна з яких оцінюється командою. Кожен день команда оцінює об'єм роботи, який необхідно провести для завершення задачі.

Burndown chart – показує, скільки вже виконано і скільки ще залишається зробити.

Планування спринта (Sprint Planning Meeting)

Проходить на початку нової ітерації Спринта:

- Із Product Backlog обираються задачі, зобов'язання по виконанню яких за спринт приймає на себе команда;
- На основі обраних задач створюється Sprint Backlog. Кожна задача оцінюється у ідеальних людино-годинах;
- Рішення задачі не повинно займати більше 12 годин або одного дня. При необхідності задача розбивається на підзадачі;
- Обговорюється та визначається, яким чином буде реалізовано цей об'єм робіт;
- Тривалість наради обмежена зверху 4–8 годинами в залежності від тривалості ітерації, досвіду команди тощо;
- (перша частина наради) Беруть участь Product Owner + Команда: обирають задачі із Product Backlog;
- (друга частина наради) Бере участь лише команда: обговорюють технічні деталі реалізації, наповнюють Sprint Backlog.

Щоденна нарада (Daily Scrum Meeting)

Відбувається кожен день протягом спринта. Є «пульсом» ходу спринта. Нараді властиві наступні обмеження:

- починається точно вчасно;
- всі можуть спостерігати, але говорять тільки обрані;
- триває не більш ніж 15 хвилин;
- проводиться в одному і тому ж місці протягом одного спринта.

Протягом наради кожен член команди відповідає на 3 запитання:

- Що зроблено з моменту попередньої щоденної наради?;
- Що буде зроблено з моменту поточної наради до наступної?;

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

– Які проблеми заважають досягненню цілей спринта? (Над рішенням цих проблем працює ScrumMaster. Зазвичай це рішення проходить за рамками щоденної наради і у складі осіб, що безпосередньо займаються даною перешкодою.)

Демонстрація (Sprint Review Meeting):

- Проходить у кінці ітерації (спринта).
- Команда демонструє внесок функціональності до продукту всім зацікавленим особам.
- Залучається максимальна кількість глядачів.
- Усі члени команди беруть участь у демонстрації (одна людина на демонстрацію або кожен показує, що зробив за спринт).
- Обмежена 4-ма годинами в залежності від тривалості ітерації і змін у продукті.

Ретроспектива (Sprint Retrospective):

- Члени команди висловлюють свою думку про минулий спринт.
- Відповідають на два основних запитання: Що було зроблено добре у минулому спринті?; Що потрібно покращити в наступному?.
- Виконують покращення процесу розробки (вирішують питання та фіксують вдалі рішення).
- Обмежена 1–3ма годинами.

Розглянемо визначення API. Це прикладний програмний інтерфейс (Application Programming Interface, API) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

Спрощено – це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

API є абстрактним поняттям – програмне забезпечення, що пропонує деякий API, часто називають реалізацією (implementation) даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

Високорівневі API часто програють у гнучкості. Виконання деяких функцій нижчого рівня стає набагато складнішим, або навіть неможливим.

В об'єктно-орієнтованих мовах, прикладний програмний інтерфейс зазвичай включає в себе опис набору визначень класу, з набором форм поведінки, пов'язаних з цими класами. Це абстрактне поняття пов'язане з реальними функціями, які надані або надаватимуться, класами, які реалізуються в методах класу.

Прикладний програмний інтерфейс в даному випадку можна розглядати як сукупність всіх методів, які публічно доступні в класах (зазвичай званий інтерфейс класу). Це означає, що прикладний програмний інтерфейс вказує методи, за допомогою яких взаємодіє з об'єктами, отриманими з визначень класів і обробляє їх.

У більш загальному плані можна визначити Прикладний Програмний Інтерфейс як сукупність усіх видів об'єктів, які можна вивести з визначення класу, і пов'язаних з ними можливих варіантів поведінки.

Наприклад: клас, що представляє Stack, може просто виставити публічно два методи Push() (для додавання нового елемента в стек) і Pop() (для вилучення останнього пункту, ідеально розташований на вершині стека).

У цьому випадку Прикладний Програмний Інтерфейс може бути інтерпретованим як два методи pop() і push(), або, більш широко, використовується варіант, коли можна використовувати елемент типу Stack, який реалізує поведінку стека, надаючи йому можливість для додавання / видалення елементів з вершини. Друга інтерпретація видається більш доречною в дусі об'єктно-орієнтованого підходу.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Якість документації, пов'язаної з Прикладним Програмним Інтерфейсом, є часто ключовим фактором, що визначає його успішність з точки зору простоти використання.

ППІ, як правило, пов'язаний із бібліотеками програмного забезпечення: ППІ описує і вказує очікувану поведінку в той час, як бібліотека є фактичною реалізацією даного набору правил. Один ППІ може мати декілька реалізацій (або жодної, будучи абстрактним) у вигляді різних бібліотек, які мають такий же інтерфейс.

Прикладний програмний інтерфейс також може бути пов'язаним з платформами програмування: платформа може бути заснована на кількох бібліотеках реалізує декілька інтерфейсів ППІ, але на відміну від звичайного використання ППІ, доступ до поведінки вбудований в платформу опосередкований шляхом розширення його змісту новими класами і вставлений в саму платформу. Крім того, загальний потік управління програми може бути під контролем абонента.

Прикладний програмний інтерфейс може бути також реалізацією протоколу.

Коли ППІ реалізує протокол, він може бути заснованим на проксі-методах віддалених викликів, що засновані на протоколі зв'язку. Роль ППІ може заключатися саме в тому, щоб приховати деталі транспортного протоколу. Наприклад: RMI є ППІ, який реалізує протокол або JRMP IIOP як RMI-IIOP.

Протоколи, як правило, розподіляються між різними технологіями і зазвичай дозволяють різним технологіям обмінюватися інформацією, діючи як абстракція між двома світами. ППІ, як правило, є специфічним для конкретної технології: звідси, інтерфейси даної мови не можуть бути використані на інших мовах, якщо виклики функції не будуть перетворені з конкретної адаптації бібліотеки. Деякі мови, серед яких такі, що працюють на віртуальних машинах (наприклад: мови, сумісні з NET CLI середовища CLR і JVM сумісних мов у віртуальній машині Java) можуть ділитися програмними інтерфейсами.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

У цьому випадку віртуальна машина дозволяє мові взаємодії завдяки спільному знаменнику віртуальної машини, що абстрагується від конкретної мови, використовувати проміжний байт-код і його мову.

При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, ППП визначається набором повідомлень запиту HTTP, також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON). У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилась (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) на більш прямі передачі репрезентативного стану (REST) стилів веб-ресурсів та ресурсів-орієнтованої архітектури (ROA).

Частина цієї тенденції пов'язана з рухом Семантичного веб-ресурсу до Опису Платформ (RDF), Концепції розвитку веб-технологій інженерних онтологій. Прикладні програмні інтерфейси у Web, що дозволяють комбінувати декількома прикладними програмними інтерфейсами в нові додатки називають гібридними.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм Khufu. Khufu – це 64-бітовий блоковий шифр. 64-бітовий відкритий тест спочатку розщеплюється на дві 32-бітові половини, L і R . Над обома половинами й певними частинами ключа виконується операція XOR. Потім, аналогічно DES, результати проходять деяку послідовність раундів. У кожному раунді молодший значущий байт L використовується як вхід S-блоку. У кожного S-блоку 8 вхідних біт і 32 вихідних біта. Далі обраний в S-блоці 32-бітовий елемент піддається операції XOR з R . Потім L циклічно зрушується на число, кратним восьми биткам, L і R міняються місцями, і раунд завершується.

Сам S-блок не статичний, він міняється кожні вісім раундів. Нарешті, по закінченні останнього раунду, над L і R виконується операція XOR з іншими частинами ключа, і половини поєднуються, утворюючи блок шифртексту.

Хоча частини ключа використовуються для операції XOR із блоком шифрування на початку й кінці виконання алгоритму, головне призначення ключа – генерація S-блоків. Ці S-блоки секретні, по суті, це частина ключа. Повний розмір ключа алгоритму Khufu дорівнює 512 біт (64 байт), алгоритм надає спосіб генерації S-блоків по ключу.

К6ПЗ_2026

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1. З рисунку головного вікна можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Функціональних кнопок ПЗ: Провести сканування; Відміна.
- Розділу обрання біоданих сканування.
- Розділу виведення результату роботи системи.
- Функцій представлених у графічному вигляді.

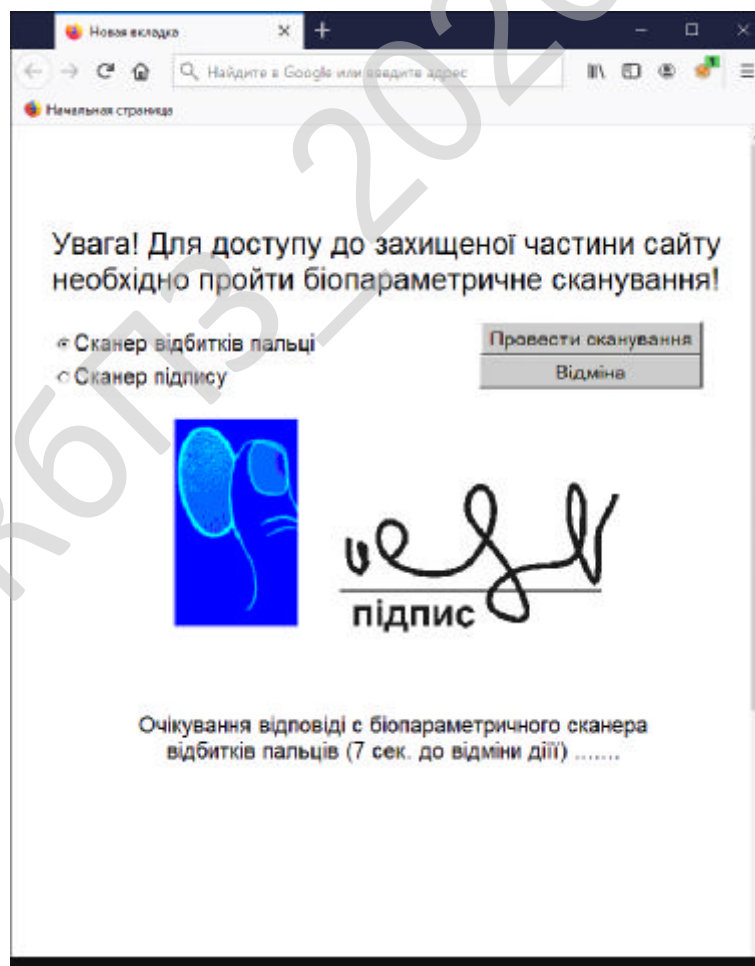


Рисунок 5.1 – Головне вікно ПЗ

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення. Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

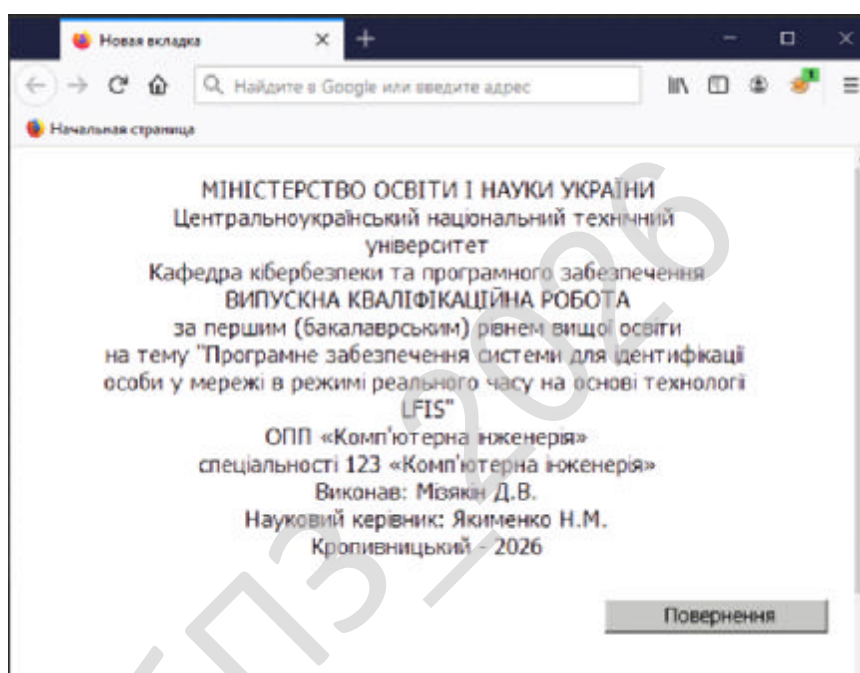


Рисунок 5.2 – Авторське право

Процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення. Таким чином у результаті вищерозглянутого можна стверджувати що розроблено інтерфейс системи у відповідності з вибраною метою роботи.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.
- Досліджена система для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.
- На основі отриманих результатів досліджень створена програмна реалізація системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS.

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 HTML, ASP, Jscript, Visual C++.. Дана мова програмування дозволяє найбільш ефективно обробляти дані

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

призначені для системи для ідентифікації особи у мережі в режимі реального часу на основі технології LFIS. 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 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 Khufu.

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.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p
2. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
3. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
4. Смірнов О., Заріцький В., Буравченко К., Коноплицька-Слободенюк О., Константинова Л., Якименко Н., Смірнов С. «Оптимізація розпізнавання облич за допомогою CUDA прискореної бібліотеки dlib». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, Том 4, № 32, С. 573-582.
5. Kuznetsov O., Frontoni E., Arnesano M., Smirnov O., Khruskov B. «Biometric Authentication in TinyML: Opportunities and Challenges». *Tiny Machine Learning: Design Principles and Applications*, 2026, pp. 587-633.
6. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинський В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр “Кафедра”, 2025.– 320 с.
7. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.
8. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

9. Kuznetsov O., Frontoni E., Kuznetsova Ye., Smirnov O., Chevardin V. «Achieving Enhanced Security in Biometric Authentication: A Rigorous Analysis of Code-Based Fuzzy Extractor». *CEUR Workshop Proceedings*, Volume 3624, 2023, pp. 330-339.

10. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

11. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings*, Volume 3504, 2023, pp. 1-11.

12. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

13. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення*, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

14. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення*, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

15. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

16. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

17. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: *Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

18. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

19. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

20. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

21. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

22. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Книшук А.В. «Вступ до кібербезпеки»: навчальний посібник – Кропивницький: ЦНТУ – 2022. – 968 с.

23. Теорія та практика сучасного інформаційно-психологічного протидіювання: навчальний посібник / [В.М. Петрик, С.О. Гнатюк, М.М. Присяжнюк та ін.]; за заг. ред. С.О. Гнатюка, В.М. Петрика та О.А Смірнова. – Полтава, 2022. – 334 с.

24. Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., Smirnova, T. «Biometric authentication using convolutional neural networks». *Lecture Notes in Networks and Systems*. Volume 152, 2021, Pages 85-98.

25. 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

26. 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.

27. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020*, Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

28. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings Volume 2805*, 2020, Pages 44-58.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

29. 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.
30. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.
31. 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.
32. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.
33. 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.
34. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.
35. 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.
36. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I., Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 166-171.

37. 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.
38. 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.
39. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.
40. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43.
41. 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.
42. 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.
43. 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.

44. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019.

45. 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.

46. Smirnov, O., Kuznetsov, A., Kiian, A., Gorbenko, Y., Cherep, O., Bexhter L. «Code-based Pseudorandom Generator for the Post-Quantum Period», *2019 IEEE International Conference on Advanced Trends in Information Theory (IEEE ATIT 2019)*. 18.12.19-20.12.19 Kyiv Ukraine. P. 204 – 209.

47. Smirnov, O., Kuznetsov, A., Nariezhnii, O., Stelnyk, S., Kokhanovska, T., Kuznetsova T., «Side Channel Attack on a Quantum Random Number Generator», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18 - 21 September 2019. P.713-718.

48. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P. 707-712.

49. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.701-706.

50. Smirnov, O., Hu, Z., Vasiliu, Y., Sydorenko, V., Polishchuk, Y., «Abstract Model of Eavesdropper and Overview on Attacks in Quantum Cryptography Systems», *10th IEEE International Conference on Intelligent Data Acquisition and*

Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.399-405.

51. 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.*

52. Smirnov, O., Kuznetsov, A., Kiian, A., Babenko, B., Zhosan, H., Prokopovych-Tkachenko, D., «Soft Decoding Method for Turbo-Productive Codes», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT 2019, Lviv, Ukraine, 2-6 July, 2019, P. 129-134.*

К6ПЗ-2026

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71