МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Безпека комп'ютерних систем і мереж»

Група: 4КБ-02

Дипломний проект

здобувача освіти денної форми навчання
КБ.02.15.000.ДП

ПОЛТОРАКІНА
ДАНІЇЛА ВОЛОДИМИРОВИЧА

м. Одеса
2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **123 «Комп'ютерна інженерія»**

Освітньо-професійна програма: **«Безпека комп'ютерних систем і мереж»**

Група: **4КБ-02**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

Розробка моделі захисту SmartHome за допомогою біометричної аутентифікації

Проектний матеріал складається з пояснювальної записки на 72 сторінках та графічного (презентаційного) матеріалу на 12 аркушах (слайдах)

Дипломник _____ (Полторакін Д.В.)

Керівник _____ (Залапін О.І.)

Консультанти:

з економічного розділу _____ (Канський М.Ю.)

з розділу охорони праці та техніки безпеки _____ (Чорновол Н.І.)

з нормоконтролю _____ (Петрашова В.І.)

старший консультант _____ (Кривченко Ю.В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю.В.)

Завідувач відділення _____ (Краснокутьса К.Г.)

Захист «28» червня 2025 р. Протокол ЕК № 7

Оцінка ЕК 4 (добре) / 75 б.

Секретар ЕК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення *комп'ютерних систем* Комісія *КТ та ПІ*

Спеціальність *123 «Комп'ютерна інженерія»*

Освітньо-професійна програма *«Безпека комп'ютерних систем і мереж»*

ЗАТВЕРДЖУЮ:

Заст. дир. З НВР *Беркань І.В.*

"19" 05 2025 р.

ЗАВДАННЯ

на дипломний проект

Здобувачеві освіти *Полторакіна Даніїла Володимировича*

(прізвище, ім'я, по батькові)

1. Тема проекту *Розробка моделі захисту системи SmartHome за допомогою біометричної аутентифікації*

затверджена наказом по коледжу від "14" 11 2024 р. № 246

2. Термін здачі закінченого проекту

3. Вихідні данні до проекту *1. Реалізувати модель захисту системи SmartHome за допомогою біометричної аутентифікації; 2. Забезпечити програмними засобами генерацію AES ключа; 3. Реалізувати серверну частину та клієнтський інтерфейс програмними засобами; 4. Розробити програмними засобами механізм двохфакторної аутентифікації.*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Огляд існуючих систем SmartHome; Аналіз існуючих рішень аутентифікації; Архітектура системи SmartHome Lock; Розробка загальної моделі роботи програми; Реалізація серверної частини; Розробка клієнтського інтерфейсу; Реалізація інтерактивного меню.

5. Перелік графічного(презентаційного) матеріалу(з точним зазначенням обов'язкових креслень, кількості слайдів)

Блок-схема роботи SmartHome Lock; Відображення інтерфейсу SmartHome Lock; Взаємодія з Інтерфейсом Telegram Bot для додавання користувача створеного за допомогою штучного інтелекту; Тестування аторизації; Екстрене відкривання дверей через BACKUP_CODES; Схема алгоритму роботи Telegram-бота SmartHome Lock; Логування подій програмної моделі; Сценарії використання SmartHome Lock; Процкс верефікації користувачів

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	*Залапін О.І.*		
Економічний розділ	*Канський М.Ю.*		
Розділ охорони праці	*Чорновол Н.І.*		
Нормоконтроль	*Петрашова В.І.*		
Старший консультант	*Кривченко Ю.В.*		

7. Дата видачі завдання _____

Керівник *Залапін О.І.* _____
(підпис)

Завдання прийняв до виконання *Полторакін Д.В.* _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	*Вступ. Постановка задачі проектування*	18.05.2025	виконано
2	*Аналіз технічного завдання та пошук літератури*	21.05.2025	виконано
3	*Аналіз структури системи SmartHome*	23.05.2025	виконано
4	*Аналіз рішень на базі штучного інтелекту*	24.05.2025	виконано
6	*Розробка загальної моделі SmartHome Lock*	28.05.2025	виконано
7	*Розробка механізму захисту від атак*	29.05.2025	виконано
8	*Реалізація клієнтського інтерфейсу*	02.06.2025	виконано
9	*Розробка серверної частини*	04.06.2025	виконано
10	*Тестування роботи програмної моделі*	06.06.2025	виконано
11	*Виконання економічних розрахунків*	09.06.2025	виконано
12	*Розробка питань з охорони праці та техніки безпеки*	12.06.2025	виконано
13	*Підготовка мультимедійної презентації*	14.06.2025	виконано
14	*Підготовка до малого захисту*	16.06.2025	виконано

Дипломник _____
(підпис)

Керівник _____
(підпис)

Формат	Зона	Поз.	Позначення	Назва	Кіл.	Примітка
				Документація		
			КБ 02. 15 000. 00 ДП	Дипломний проект		
А4			КБ 02. 15 000. 00 ДП ПЗ	Пояснювальна записка	1	

Зм.	Арк.	№ докум.	Підпис	Дата	КБ 02. 15 000. 00 ДП		
Розробив		Полторакін Д.В.		16.06.2025	Розробка моделі захисту SmartHome за допомогою біометричної аутентифікації	Літ.	Аркуш 4 / Аркушів 72
Перевірив		Залапін О.І.		16.06.2025		НДП	
Н. контр.		Петрашова В.І.		23.06		ВСП "ОТФК ОНТУ" гр.4КБ-02	
Затверд.		Кривченко Ю.В.		26.06			

ЗМІСТ

ВСТУП

Сучасний розвиток технологій SmartHome вже давно перестав бути лише теоретичною концепцією й активно впроваджується у повякденне життя. Крім того, розумні замки залишаються невід'ємною частиною такої системи, оскільки дають можливість контролювати доступ до приміщень, підвищуючи рівень комфорту користувачів. Проте підвищення функціональних можливостей супроводжується зростанням ризиків, пов'язаних із кібербезпекою. У зв'язку з цим, питання створення надійної моделі захисту системи за допомогою біометричної аутентифікації, набуває особливої актуальності в сучасних умовах.

Метою дипломного проєкта є розробка моделі SmartHome з системою контролю доступу та використанням біометричних методів автентифікації. Основна ідея полягає у впровадженні альтернативи паролям, які часто виявляються ненадійними, шляхом застосування унікальних фізичних характеристик користувача відбитків пальців чи розпізнавання обличчя для підтвердження особи. Система орієнтована на підвищення рівня безпеки та зручності у використанні, що є актуальним завданням в умовах зростаючих кіберзагроз.

Основним завданням цього дипломного проекту є розробка надійної та ефективної моделі контролю доступу із застосуванням біометричних методів автентифікації. Така модель має забезпечити високий рівень безпеки та зручність для користувачів, мінімізуючи ризики несанкціонованого доступу.

Розробка та впровадження подібної моделі відповідає сучасним вимогам щодо захисту інформації та матеріальних цінностей, після чого біометричні технології можуть підвищити як рівень безпеки, так і зручність використання у повсюдному житті. Даний проект спрямований на створення практично значимої моделі, яка забезпечує ефективний контроль доступу на основі біометричної аутентифікації.

1 ОСНОВНИЙ РОЗДІЛ

1.1 Огляд існуючих систем SmartHome

Розумний будинок сьогодні є реальністю, а не тільки концепцією з наукової фантастики. Йдеться про інтеграцію цифрових технологій у повсякденне життя для автоматизації та оптимізації керування домашніми системами. Основна мета - надати користувачу можливість контролювати освітлення, опалення, вентиляцію, безпеку та інші аспекти житла через смартфон, планшет або голосового асистента.

Головна мета таких систем — підвищення комфорту, безпеки та ефективності використання ресурсів у побуті. Автоматизація дозволяє спростити управління численними приладами та зменшити витрати на електроенергію чи воду завдяки оптимізованій роботі обладнання. Це досягається використанням датчиків, мережевих пристроїв та спеціалізованого програмного забезпечення, яке об'єднує різні елементи системи в єдину інфраструктуру.

Суттєвою перевагою розумного будинку є можливість віддаленого контролю: власник може керувати системою навіть перебуваючи поза межами дому, наприклад, за допомогою мобільного додатка. Можна дистанційно вимикати прилади, контролювати доступ до приміщень або застосовувати сценарії на основі зовнішніх умов, таких як прогноз погоди. Автоматизація рутинних завдань, наприклад, ранковий запуск чайника чи відкриття воріт голосовою командою, робить повсякденне життя значно зручнішим.

Ці системи сприяють підвищенню рівня безпеки, можуть інтегрувати сигналізацію, датчики руху, системи контролю доступу тощо. Технології розумного будинку передбачають взаємодію великої кількості пристроїв, які спільно забезпечують не лише комфорт, а й економію ресурсів, а також адаптацію до індивідуальних потреб користувача.

Сучасні SmartHome (рис.1.1) можуть оснащуватися як дротовими, так і бездротовими системами — іноді навіть їхньою комбінацією. Встановлення бездротової системи автоматизації є досить зручним і не потребує значних засобів, що дозволяє оперативно інтегрувати функції на кшталт інтелектуального освіти, клімат-контролю чи системи безпеки. Втім, ціна такого рішення може сягати

кілька тисяч доларів, що забезпечує рівень комфорту та функціональності для користувачів.Крім того, бездротові системи мають певні технічні вимоги: стабільне та потужне покриття Wi-Fi у всій частині, що може потребувати додаткових інвестицій у мережеву інфраструктуру — наприклад, у ретранслятори або додаткові точки доступу. Варто зазначити, що бездротові рішення переважно обирають для невеликих приватних будинків чи орендованої нерухомості, де проведення проводів є менш доцільним через обмежену площу або тимчасовість проживання.Дротові системи, у своєму разі, вважаються значно надійнішими та менш вразливими до зовнішніх вручань. присутність може позитивно вплинути на ринкову вартість їх житла. без винятку, дротові системи легко масштабуються, що робить їх оптимальним вибором при проектуванні новобудов або проведенні капітального ремонту. Таким чином, рішення щодо типу системи часто залежить від конкретного об'єкта та довгострокових планів власника.

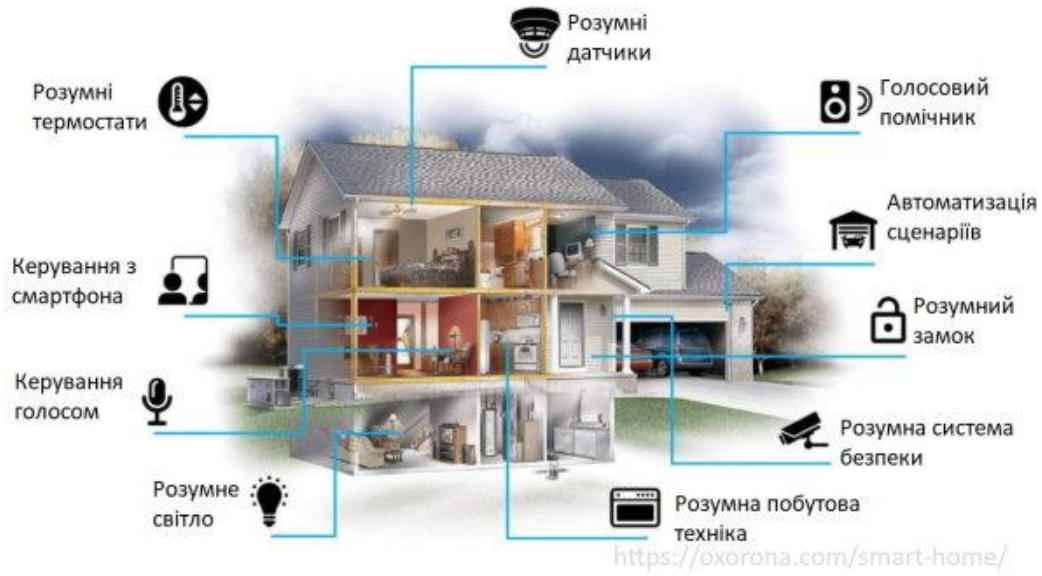


Рисунок 1.1 Демонстрація технології SmartHome

Значущим недоліком, який варто враховувати, є висока вартість оснащення житла дротовою розумною системою. Такі рішення можуть обійтися власникам у десятки тисяч доларів, а також вимагають додаткового простору для розміщення мережевого обладнання, включаючи кабелі Ethernet.У випадку бездротових систем пристрої в оселі взаємодіють через стандарти бездротового зв’язку, такі як Wi-Fi, Bluetooth чи Thread, з можливістю керування через смартфон, планшет або

комп’ютер.Вибір технології бездротового зв’язку безпосередньо впливає на функціональність системи. Наприклад, Wi-Fi є поширеним варіантом завдяки простоті налаштування та здатності підтримувати сервіси з високою пропускною здатністю, зокрема потокове відео. Водночас пристрої з Wi-Fi зазвичай дорожчі та споживають більше енергії.Bluetooth — це також розповсюджена й проста у використанні технологія, яка вирізняється низьким енергоспоживанням. Проте її радіус дії обмежений, а максимальна кількість підключених пристроїв невелика. Thread орієнтована на застосування з низьким енергоспоживанням і пропускною здатністю, характерні для автоматизації розумного будинку. Вона дозволяє підключати велику кількість пристроїв і має більший радіус дії, ніж Bluetooth. Проте пристрої Thread можуть бути дорогими і для підключення до Інтернету потребують окремого хабу.Узагальнюючи: Wi-Fi найкраще підходить для задач із високою пропускною здатністю, Bluetooth — для короткодистанційної комунікації між пристроями, а Thread — для автоматизації, що не потребує великої потужності чи швидкості передачі даних.

1.1.1 Типи загроз і вразливостей SmartHome

Несанкціонований доступ до мережі – якщо нехтувати елементарними правилами кібергігієни, наприклад, використовувати надто прості або інші паролі для різних пристроїв, ризик проникнення в домашню мережу суттєво зростає. Зловмисники активно сканують мережі на об’єкті слабких місць, і маршрутизатор чи “розумний” пристрій залишився із заводським паролем, це фактично якщо запрошення для несанкціонованого доступу. Особливо вразливі старі моделі пристроїв, які давно не отримували оновлення безпеки.

Атаки Man-in-the-Middle (MitM) – цей тип атаки передбачає перехоплення даних у процесі їх передачі між пристроями користувачів та зовнішніми сервісами, наприклад, хмарними сховищами. Зловмісник може отримати доступ до конфіденційної інформації, змінювати або підміняти дані. деякі подібні атаки реалізуються у відкритих або незахищених мережах Wi-Fi, де відсутнє шифрування або автентифікація.

DDoS-атаки – Інтернет-речі (IoT), що створюють нові точки входу для хакерів. Здавалося б, звичайна камера відеонагляду чи “розумний” термостат можуть бути зламані та підключені до ботнету, який організовує масовані DDoS-атаки на великих сервісах. Власник пристрою навіть часто не підозрює, що його техніка стала частиною глобальної атаки, а слідки для цільової системи можуть бути катастрофічними – від доступу до повного падіння сервісу.

Шкідливе програмне забезпечення (трояни, боти) – через підроблені додатки, заражені або навіть оновлення через рибальські листи, пристрої можуть бути інфіковані троянами чи ботами. Таке ПЗ може автоматично фіксувати натискання клавіш, викрадати паролі, ушкоджувати шкідливі дії або автоматично перетворювати пристрій на елемент ботнету. Особливо небезпечні ті випадки, якщо користувач не помічає жодних змін у роботі пристрою, шкідливий софт діє.

Фішинг та соціальна інженерія – тут основна загроза не в технологіях, а в людському факторі. Зловмісники створюють рибальські сайти, підроблені листи, місяться або навіть дзвінки, щоб обманним шляхом отримати доступ до облікових записів, паролів чи іншої критично важливої інформації. Соціальна інженерія працює через маніпуляцію довірою, і навіть найсучасніші технічні засоби не захищають, якщо користувач самостійно передає своїм стороннім особам.

У підсумку сучасні кіберзагрози мають комплексний характер і вимагають не тільки технічних засобів захисту, а й підвищення обізнаності користувачів щодо основних принципів інформаційної безпеки.

Фізичні загрози — це не просто якась абстракція, а цілком реальні ситуації, з якими можна зіштовхнутися будь-коли. В першу чергу, йдеться про крадіжку даних із пристроїв. Наприклад, якщо хтось отримає доступ до камер спостереження та заволодіє відеозаписами, це може призвести до серйозного порушення приватності та навіть шантажу. В сучасних умовах, коли відеоспостереження стало майже повсюдним, ризики тільки зростають.

Ще одна важлива загроза — підробка RFID або NFC-ключів. Ці технології широко використовують для доступу до приміщень, автомобілів, навіть банківських карток. Якщо нападник зуміє скопіювати або підробити ключ, то всі

складні електронні замки стають, по суті, марними. Це створює додатковий виклик для розробників систем безпеки, адже потрібно постійно вдосконалювати механізми захисту.

Окрема категорія — вимкнення або пошкодження пристроїв. Наприклад, якщо зловмисник просто знеструмить сигналізацію чи камеру, то вся система охорони моментально втрачає ефективність. Наслідки можуть бути серйозними — від крадіжок до втрати важливої інформації. Саме тому сучасні системи безпеки намагаються впроваджувати резервні джерела живлення та сигнали тривоги у випадку відключення.

Загалом, фізичні загрози не варто недооцінювати. Вони можуть здаватися простішими, ніж складні кібератаки, але їх наслідки часто бувають не менш руйнівними.

Вразливість системи розумного дому — це не просто технічна проблема, а справжній виклик для безпеки користувачів. Розгляньмо детальніше ключові аспекти.

По-перше, застосування слабких паролів для замовчування наприклад, “admin:admin”, залишається дивовижно розширеною практикою. Це відкриває прямий шлях для несанкціонованого доступу до пристроїв і даних користувача. Досвід показує, що ігнорування елементарних правил кібергігієни стає причиною численних інцидентів.

Другий критичний момент — демонстрація шифрування під час передачі даних. Відео та аудіо, які передаються у відкритому вигляді, можуть бути перехоплені чи підслухані будь-ким, хто має мінімальні знання у сфері мережевої безпеки. Це створює ризик як для приватності, так і для цілісності даних.

Не менше є питання оновлення програмного забезпечення. Використання старих версій прошивок із відомими вразливостями розширене явище, обумовлене як несправністю користувачів, так і відсутністю автоматичного оновлення системи зі сторони виробників. Це робить пристрій легкою мішенню для нападу.

Відкриті порти та незахищені API — ще одна болюча тема. Вони не зможуть зловмисникам отримати віддалений доступ до системи, використовуючи

стандартні засоби сканування мережі. Така недбалість у налаштуваннях може призвести до повного контролю над розумним домом сторонніми особами.

Окремої уваги заслуговує вразливість сторонніх сервісів, зокрема керування хмарними платформами. Недостатньо захищені або неправильно налаштовані хмарні сервіси здатні стати додатковим каналом для компрометації персональних даних та управління пристроями.

Загалом, нехтування базовими принципами кібербезпеки, викликає належну увагу до оновлень і налаштувань, а також сліпу довіри до зовнішніх платформ — це ті чинники, які роблять сучасні SmartHome-системи вразливими до атаки і загрожують як приватність, так і безпека користувачів.

Організаційні проблеми починаються з того, що виникає двофакторна автентифікація (2FA) — це не просто дрібна помилка, а реальна загроза для безпеки всієї системи. Досити одного вдалого фішингу чи банального підбору пароля, і доступ до конфіденційної інформації чи способами керування можна знайти в чужих руках. Розширена думка, що мене це не потрібно, закінчується неприємними наслідками для власників SmartHome.

Ще одна проблема — неправильна настройка мережі. Відсутність сегментації означає, що всі пристрої підключені до однієї мережі, і якщо зламати хоча б один пристрій, можна отримати доступ до всіх інших. Слабкий пароль на Wi-Fi наче відкриті двері для зловмисників; про складність пароля часто забувають, і дарма.

Окремо варто використовувати використання неперевірених додатків для керування SmartHome. Багато користувачів завантажують перше-ліпше програмне забезпечення, не замислюючись, хто його розробник і чи є воно безпечним. Такий підхід створює додаткові ризики, оскільки подібні добавки можуть створити шкідливий код або мати вразливість, про які навіть не здогадуються самі користувачі.

Сукупність цих організаційних недоліків не лише забезпечує ймовірність кіберінцидентів, а й серйозно ускладнює реагування на них. Тому ігнорування базових принципів інформаційної безпеки у сфері SmartHome — це не тільки

недалекоглядність, а й показовий шлях до суттєвих матеріальних та репутаційних втрат.

Втрата конфіденційності — це не просто неприємність, а реальна загроза. Коли сторонні особи підтримують доступ до камер чи мікрофонів, йдеться не лише про втрату приватності, а й про порушення основних прав людини. У сучасних реаліях інформація — це ресурс, і контроль над ним відкриває безліч можливостей для зловмисників.

Далі, крадіжка особистих даних — вже класика жанру. Але що особливо небезпечно, так це те, що через популяризацію голосових помічників, користувачі часто навіть не вважають, скільки інформації вони “віддають” пристроям. Логіни, банківські реквізити, інші чутливі дані — усе це стає помітною здобиччю для кіберзлочинців. А відновити втрачену довіру до системи потім ой як непросто.

Фізичний злам — ще одна болюча тема. Якщо через вразливість “розумного дому” можна просто дистанційно відчинити двері або вимкнути сигналізацію, мова вже йде про безпеку самого життя та майна користувача. Тобто ризики залишаються не тільки віртуальними, а досить матеріальними.

Ну і, нарешті, використання ваших пристроїв, скажемо, холодильника чи телевізора, як плацдарму для атаки на інші системи. Ця ідея “Інтернету речей” іноді грає з нами злий жарт: навіть побутова техніка може стати частиною масштабної кібератаки, якщо не підбати про захист.

Тож, підсумовуючи: сучасні порушення безпеки — це не лише про вік даних чи зламаний обліковий запис, а про комплекс загроз, які можуть зачепити всю сферу життя. Ігнорувати ці ризики — дуже необачна стратегія, особливо в епоху, коли технології буквально керують нашим буденним життям.

Заходи захисту пристроїв та мереж – штука серйозна, ігнорувати її не варто навіть у домашніх умовах, не кажучи вже про корпоративний рівень.

Перше й вихідне – регулярне оновлення прошивок і програмного забезпечення. Виробники постійно випускають патчі для закриття вразливостей. Якщо відкласти оновлення «потім», файл можна залишити відкритим для відомого нападу.

Паролі – це не просто формальність, а реально перший рубіж оборони. Використовуйте складні комбінації, комусь можна здатися параноїдальним, але двофакторна автентифікація (2FA) – це must-have. Вона мінімізує ризик несанкціонованого доступу навіть якщо хтось побачить пароль.

Мережу слід сегментувати. Якщо в домі чи офісі є IoT-пристрої, виділіть їм окрему VLAN. Це не тільки рівень безпеки, а й дозволяє краще контролювати трафік і швидко ізолювати виявлено скомпрометований сегмент.

Особливо важливо вимікати зайві функції. Віддалений доступ, UPnP, автоматичне перенаправлення портів – все це зручно, але часто небезпечні речі. Якщо ви не впевнені, що щось потрібно – краще вимкнути.

VPN – універсальний інструмент захисту від зовнішніх загроз. Крім того, якщо підключаєтеся до власної мережі з кав'ярні, VPN шифрує дані й захищає від перехоплення.

Дуже часто користувачі бездумно надають додаткові дозволи. Слідкуйте, до чого має доступ кожен додаток. Наприклад, холодильнику точно не потрібна ваша геолокація чи контакти.

Шифрування трафіку – основа сучасної безпеки. Використовуйте WPA3 для Wi-Fi, TLS для передачі даних. Це ускладнити життя зломісникам навіть у випадку перехоплення пакету.

Підсумовуючи: кібербезпека – це не параноя, а елементарна гігієна у світі, де пристрої спілкуються між собою більше, ніж деякі люди. Не відкладайте захист "на завтра" – завтра, можливо, вже буде пізно.

1.1.2 Сучасні технології біометричної ідентифікації

Основні поняття у сфері біометричної ідентифікації охоплюють низку важливих принципів, без яких побудова надійної та ефективної системи неможлива:

– універсальність передбачається, що будь-яка особа хоче мати одну відповідну біометричну характеристику, наприклад, відбитки пальців або форму обличчя, яку можна об'єктивно виміряти. Без універсальності система втрачає сенс, тому не зможе охопити все населення.

– унікальність Біометрична ознака шкіри повинна бути достатньою

відмінністю, щоб дозволити ідентифікувати окрему особу серед усіх інших. Наприклад, відбитки пальців навіть у однояйцевих близнюків відрізняються, що підкреслює потенціал біометрії як надійного серйозного розпізнавання.

– стійкість біометричні характеристики повинні залишатися лише незмінними протягом життя людини. Це важливо для забезпечення довгострокової надійності системи. Наприклад, райдужна рослина ока змінюється мінімально з вікном, що робить її перспективною для ідентифікації.

– вимогливість в ідеалі процес отримання біометричних даних повинен бути швидким, зручним і не завдавати дискомфорту користувачу. Надмірно складні або інвазивні методи знижують практичність використання таких систем у повсюдному житті.

– продуктивність комплексний показник, що включає точність, швидкість і надійність системи роботи. Високопродуктивна система має мінімізувати кількість гібних позитивних і негативних результатів, а також забезпечити стабільну роботу навіть при високих навантаженнях

– прийнятність відображає рівень довіри до біометричної технології з боку користувачів та суспільства. Сюди відносяться питання захисту персональних даних, етичні та юридичні аспекти застосування біометрії.

Верифікація - (порівняння “один до одного”). У цьому режимі система перевіряє, чи співпадає біометричні дані користувача із зразком, прив’язаним до конкретного цифрового запису чи ідентифікатора. загальноприйнятий як додатковий рівень захисту при вході за допомогою пароля, смарт-карти або мітки.

Ідентифікація - (порівняння “один до багатьох”). Тут біометричний зразок складається з усією базою даних, і система робиться, чи є серед зареєстрованих осіб власник цих даних. Це особливо актуально для доступа до об'єктів з високими вимогами безпеки, контролю на кордонах тощо.

1.1.3 Аналіз існуючих рішень автентифікації

У сфері цифрової безпеки та автентифікації користувачів найбільшим популярним технічним є FaceID розроблений корпорацією Apple. FaceID став передовим рішенням, яке виходить за межі традиційних методів і встановлює нові

стандарти зручності і можливостей застосування автентифікації особистості. Розроблена технологія була представлена компанією у вересні 2017 року з випуском нової моделі телефону. Розробка технології це результат кропотливої роботи передових інженерів корпорації, єкспертами з комп'ютерного зору та фахівців з апаратного обладнення.

Технологія була створена як відповідь на численні обмеження та вразливості, що застосовуються традиційним методам автентифікації. На той момент команда інженерів комплексно підійшла до проблеми, здійснивши масштабний аналіз і переосмислення існуючих підходів. Робота не обмежилася поверхневими вдосконаленнями, їх було розроблено абсолютно нову архітектуру, яка стала основою для революційних змін у сфері захисту чутливої інформації. Варто підкреслити, що використання сучасних досягнень у сфері штучного інтелекту та комп'ютерного зору стало ключовим елементом цієї технології. Саме за допомогою цих інструментів можна забезпечити безперервний та надійний контроль автентичності користувачів, що значно ускладнює можливість несанкціонованого доступу. У поєднанні з багаторівневим аналізом поведінки система здатна лише ідентифікувати особу з високою точністю, а й адаптуватися до використання нових загроз та сценаріїв.

безсумнівно, цей підхід відкриває принципово нові можливості для інтеграції захисту на різних рівнях цифрової взаємодії. Це дозволило розширити горизонти використання технології — від фінансового сектору до медицини, де безпека персональних даних є критично важливою. В результаті розроблене рішення стало не просто відповіддю на актуальні виклики, а й визначило новий стандарт для майбутніх систем аутентифікації на глобальному рівні.

Іноваційний підход до розробки FaceID відкрив з множину цікавих рішень за допомогою по'єднання різноманітних знань і досліджень в цій сфері. Розробка поставила численні виклики, вимагаючи конвергенції різноманітних знань. Команда інженерів ретельно розробила передові алгоритми, здатні обробляти та аналізувати риси обличчя з неперевершеною точністю. Технології визначання глибини, включаючи скаладний набір датчиків, були інтегровані для створення

точних тривимірних карт обличчя, забезпечуючи надійність проти потенційних спроб спуфінгу.

Apple вкотре продемонструвала свій підхід до розвитку технологій, об'єднавши команду справжніх профі у сфері програмного забезпечення, апаратної інженерії та машинного навчання. Вони не просто виконали стандартне завдання, а поставили собі за мету створити систему розпізнавання обличчя, яка відповідала б найвищим вимогам сучасної кібербезпеки. Експерти ретельно аналізували різні аспекти цієї технології, враховуючи як технічні бар'єри, так і соціальні фактори впливу. Інженери зосередили особливу увагу на адаптивності алгоритмів розпізнавання, щоб система могла точно працювати з різноманітними структурами облич, кольором шкіри, змінами зачіски чи наявністю бороди. Було враховано навіть сезонні зміни зовнішності користувачів. Подолання таких викликів вимагало не лише глибоких технічних знань, а й креативного підходу до побудови та навчання нейронних мереж. В результаті було створено систему, здатну забезпечити надійну ідентифікацію у реальних умовах, не втрачаючи точності навіть при зміні освітлення чи кута огляду. Окремо слід відзначити підхід Apple до питань конфіденційності та етики. Компанія врахувала сучасні тенденції щодо захисту персональних даних, впровадивши багаторівневі методи шифрування та мінімізуючи ризики несанкціонованого доступу. Експерти провели аналіз потенційних загроз приватності, щоб впевнитися в надійності захисту інформації користувачів, і таким чином задовольнити навіть найвибагливіших критиків. Результатом цієї масштабної роботи став не просто новий продукт, а цілий стандарт у сфері біометричної аутентифікації. Технологія інтегрована у повсякденний досвід користувачів, і вже сприймається як невід'ємний елемент сучасних пристроїв. Інноваційний підхід та системна увага до деталей дозволили Apple не тільки вирішити поточні технічні завдання, але й сформувати новий рівень довіри до цифрової безпеки у суспільстві.

З моменту появи FaceID (рис.1.2) став одним із корисніших інструментів серед користувачів Iphone. Його інтеграція стала невід'ємною частиною сучасних айфонів і інших пристроїв від компанії Apple. Простота розблокування,

автентифікація в сторонніх застосунках, банківських переводах одним лише поглядом у поєднанні з високим рівнем безпеки, який забезпечує ця технологія, вивели FaceID до цента уваги і дали цій технологій величезний зріст популярності серед користувачів.

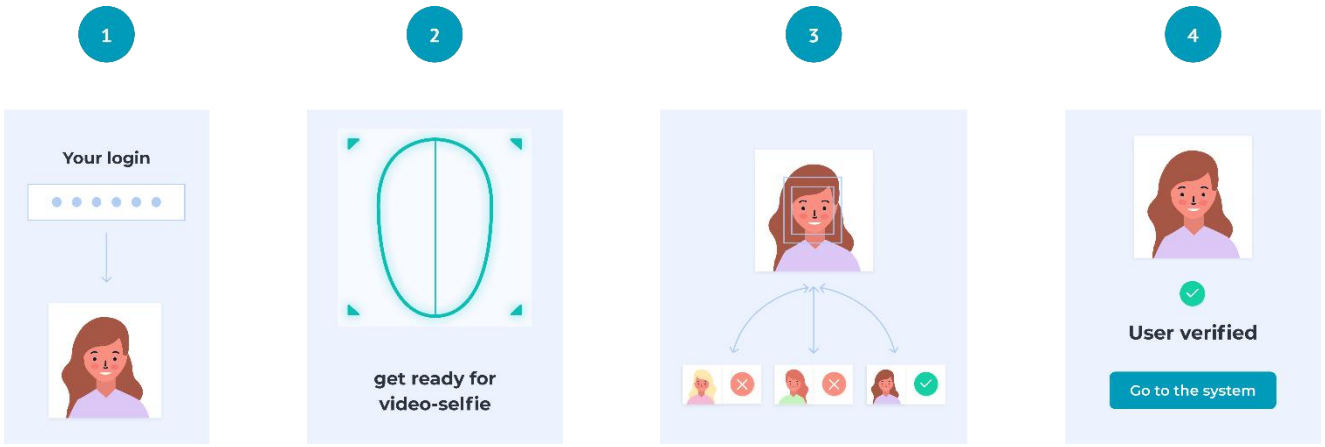


Рисунок 1.2 Огляд функціональних можливостей FaceID

FaceID став справжнім проривом у сучасній цифровій безпеці. Його поява фактично зняла з користувачів купу обтяжливих процедур, пов'язаних із введенням PIN-кодів, паролів або використанням сканерів відбитків пальців. Якщо раніше люди постійно забували свої паролі або взагалі не ставили захист на пристрої, то тепер процес ідентифікації став майже непомітним для користувача. Це вирішило одразу дві проблеми: і підвищило захищеність даних, і зробило користування пристроями максимально зручним. Варто зазначити, що завдяки складним алгоритмам і сучасним протоколам безпеки, рівень надійності FaceID значно перевищує стандарти, які встановили попередні форми аутентифікації. Технологія використовує тривимірне сканування обличчя, що дозволяє розпізнавати навіть найдрібніші зміни у зовнішності користувача. Наприклад, якщо ви змінили зачіску, відпустили бороду чи одягли окуляри, система все одно розпізнає вас. Це особливо важливо в умовах сучасного динамічного життя, коли зовнішність може змінюватися досить часто. Окрему увагу заслуговує те, як FaceID вплинув на ставлення людей до цифрової безпеки. Простота й інтуїтивність використання спонукає навіть тих, хто раніше ігнорував налаштування захисту,

активувати цю функцію. В результаті, дедалі більше користувачів почали відповідальніше ставитися до захисту своєї особистої інформації. Завдяки глибокій інтеграції з операційною системою та постійним оновленням безпекових протоколів, FaceID залишається одним із найбільш надійних інструментів для захисту даних у сучасному світі, де ризики несанкціонованого доступу лише зростають. Система не просто зберігає баланс між зручністю та безпекою, а й піднімає ці показники на новий рівень. У підсумку, FaceID став не лише технологічною новинкою, а й важливим етапом еволюції цифрової ідентифікації, який змінив підхід до захисту особистої інформації в умовах зростаючої цифрової взаємодії.

У ході розгляду архітектури FaceID (рис1.3) було виявлено що це — це цілий комплекс високотехнологічних рішень комп'ютерного зору, що базуються на сучасних досягненнях глибокого навчання. Науковий підхід тут не просто формальність, а справжній фундамент: у центрі уваги знаходяться згорткові нейронні мережі, які довели свою ефективність у задачах розпізнавання образів. Їх застосування дозволяє системі FaceID не просто фіксувати наявність обличчя, а й аналізувати його з зовнішньою точністю.Спочатку система отримує зображення, яке проходить через серію попередніх обробок. На цьому етапі відбувається виявлення області обличчя, де використовується алгоритм виділення рис із використанням згорткових нейронних мереж. Ці мережі навчені на величезних наборах даних, які охоплюють різноманітність людського одягу, освітлення, кути огляду, наявність аксесуарів чи міміки. Саме така різноманітність поточних даних дозволяє досягти високої точності під час ідентифікації в реальних умовах.Другий етап відбувається в детальному аналізі ключових рис обличчя. Алгоритми глибокого навчання виділяють характерні точки, наприклад, розташування очей, контур носа та форму рота. Ці особливості аналізуються з високою точністю, що дозволяє зберегти навіть мінімальні відмінності між усіма людьми. Система не обмежується лише базовими параметрами, а враховує цілу сукупність мікроскопічних деталей, які формують унікальний “відбіток” обличчя. Далі формується математична модель, яка є своїм рідним цифровим відображенням

унікальних рис конкретної особи. На цьому етапі генерується шаблон, який містить загальну інформацію про розташування ключових точок і характерних елементів зображення. Цей шаблон зберігається в зашифрованому вигляді в ізольованому апаратному компоненті, що гарантує високий рівень захисту персональних даних.

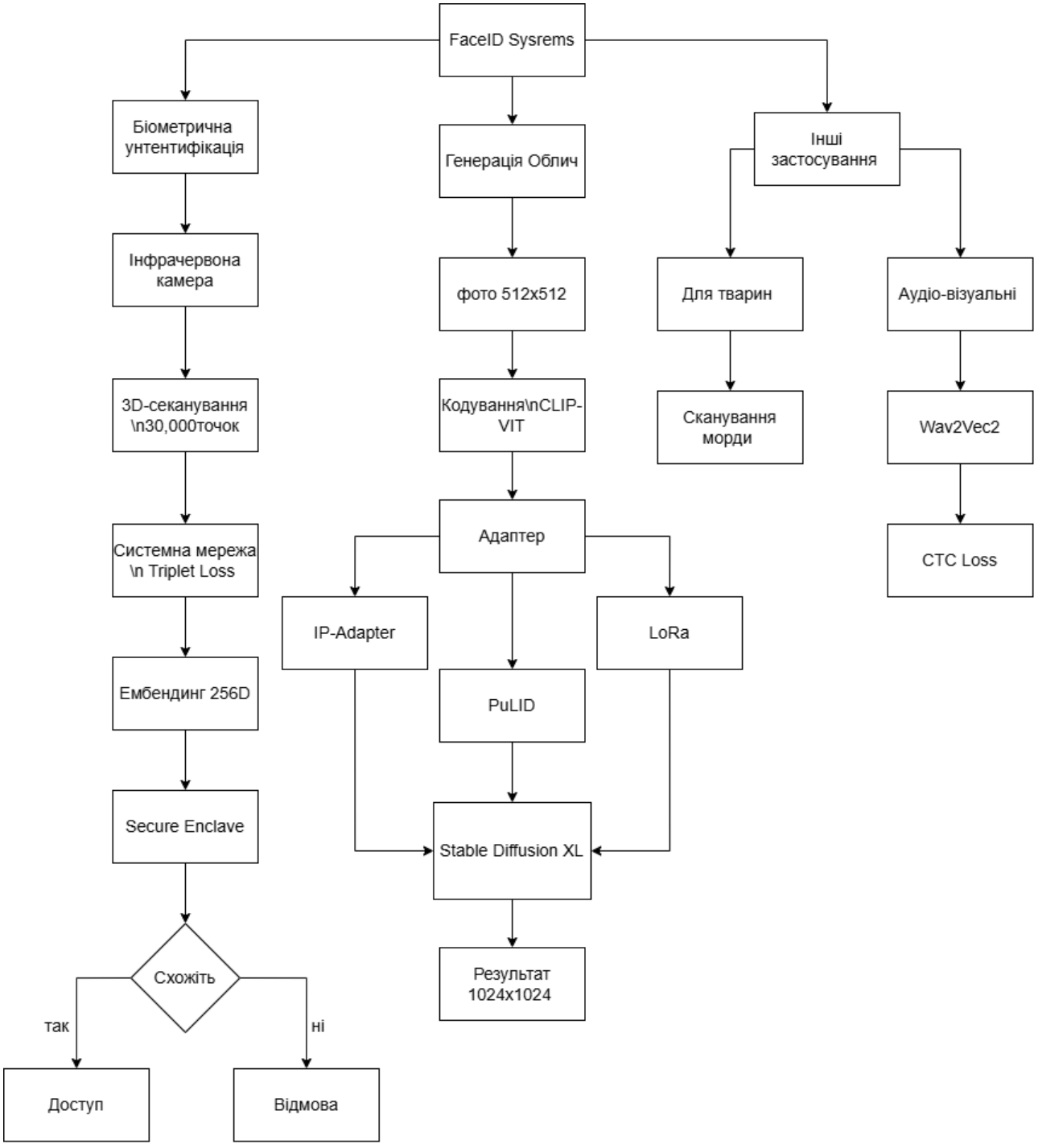


Рисунок 1.3 Загальна діаграма архітектури FaceID

Під час автентифікації система отримує нове зображення, проводить його обробку за аналогічною схемою, генерує новий шаблон та порівнює його з еталонним. Для порівняння застосовуються алгоритми, які враховують допустимі похибки, пов'язані зі зміною освітлення, курсом чи виразом обличчя. Усі обчислення та зберігання даних відбуваються в зашифрованій пам'яті, доступ до якої має лише апаратна частина пристрою. Це забезпечує додатковий рівень безпеки та мінімізує ризики несанкціонованого доступу. Завдяки такому підходу FaceID демонструє високу надійність і стійкість до різних спроб обходу систем. Наукова глибина реалізації дозволяє поєднати точність, швидкість та безпеку в єдиному розрахунку, що відповідає сучасним вимогам до біометричних систем ідентифікації.

З технічної точки зору, система FaceID є справжнім шедевром інженерної думки, де кожен елемент апаратної частини працює у тісній зв'язці з іншими для досягнення максимальної точності та безпеки. Якщо розглядати це детально, то фронтальна камера так званої глибини насправді є цілою платформою з низкою сенсорів, які діють синхронно. Основну роль тут відіграє інфрачервона камера, яка має здатність фіксувати риси обличчя користувача навіть при майже повній відсутності зовнішнього освітлення. Це стало можливим завдяки спеціальному інфрачервоному підсвічуванню, котре рівномірно розподіляє світловий потік по поверхні обличчя. Завдяки цьому жодна деталь не залишається в тіні, що критично для точного розпізнавання. Особливу увагу заслуговує точковий проектор, який формує структуру з тисяч невидимих інфрачервоних точок на обличчі користувача. Ці точки створюють складний патерн, який потім аналізується для побудови тривимірної мапи рельєфу обличчя. Важливо, що цей патерн не є випадковим, а ретельно прорахований для максимальної ефективності виявлення особливостей рельєфу, включаючи такі нюанси як глибина очних западин, форма носа, контур вилиць. Далі інфрачервона камера фіксує зображення цього патерну, і система починає аналіз деформації точок. На основі цих деформацій формується високоточна тривимірна карта обличчя користувача, яка враховує навіть найдрібніші особливості. Не менш важливим є те, що всі ці дані обробляються

окремим апаратним модулем, який ізольований від решти системи пристрою. Це рішення підвищує рівень безпеки, оскільки шаблони обличчя не потрапляють у загальний доступ і не можуть бути використані сторонніми особами навіть у разі спроби злому. Ще одним технологічним досягненням є застосування принципу часу польоту (Time-of-Flight), коли система вимірює, скільки часу потрібно інфрачервоному світлу, щоб дістатися від проектора до поверхні обличчя та повернутися назад. Це дозволяє ще точніше визначати рельєф і глибину різних ділянок обличчя, що в сукупності з інформацією від точкового патерну створює надзвичайно деталізовану тривимірну модель.

Підсумовуючи, FaceID сьогодні можна вважати справжнім проривом на перетині комп'ютерного зору та штучного інтелекту. Йдеться не лише про набір технічних рішень, а про цілісну систему, що кардинально змінила підхід до ідентифікації користувача. Втілення таких складних алгоритмів у масових пристроях, як смартфони, стало можливим завдяки багатопрофільній роботі великої команди фахівців, які не просто вдосконалили розпізнавання облич, а й інтегрували ці технології у повсякденне життя мільйонів людей. Особливу увагу заслуговує те, наскільки FaceID виявився зручним для користувачів: для розблокування пристрою достатньо одного погляду, без жодних додаткових дій. Це суттєво підвищує комфорт та швидкість доступу до інформації, водночас забезпечуючи високий рівень захисту персональних даних. З точки зору безпеки, система базується на унікальних біометричних параметрах кожної людини, що майже унеможливлює несанкціонований доступ.Варто відзначити, що впровадження FaceID стало своєрідним стандартом для всієї індустрії. Після появи цієї функції інші компанії активно працюють над створенням власних аналогів, намагаючись впровадити подібні системи у свої пристрої. Це стимулює подальший розвиток галузі та підвищує загальний рівень захищеності цифрових продуктів на ринку. Ще одна важлива деталь полягає у тому, що для більшості користувачів FaceID фактично став синонімом біометричної автентифікації. Люди відчувають довіру до цієї технології, оскільки вона поєднує у собі простоту використання та надійність, яку важко знайти у попередніх методах. Досвід взаємодії з FaceID для

багатьох став прикладом того, як сучасні технології можуть не лише полегшувати повсякденні завдання, а й забезпечувати відчуття безпеки.

Загалом, FaceID не просто витіснив старі підходи до автентифікації на своїй платформі, а фактично змінив уявлення про те, як має працювати захист даних у цифрову епоху. Його вплив на індустрію, користувацький досвід і загальні стандарти безпеки складно переоцінити.

1.1.4 Аналіз застосованих рішень на базі штучного інтелекту

У наш час збір та обробка персональних даних співробітників компаніями вже давно перестали бути чимось дивовижним чи сенсаційним — це стало буденністю, до якої суспільство поступово звикло. Індивідуальність кожної людини все частіше редукується до набору чисел, статистики, соціальних рейтингів та інших аналітичних показників. Наразі роботодавці не просто фіксують присутність чи продуктивність працівників, вони використовують веб-камери для відстеження уваги, аналізуючи навіть рухи очей. Такий підхід відкриває новий рівень детальності в контролі за поведінкою співробітників, що, з одного боку, може підвищувати ефективність, а з іншого — породжує низку етичних питань щодо приватності.

В сучасному технологічно розвиненому світі питання захисту персональної інформації напряму пов'язане із здатністю серверів забезпечувати надійне шифрування. Людина, реєструючись на нових сервісах або платформах, фактично погоджується дати доступ до своїх даних заради комфорту, швидкості чи нових можливостей. Системи в свою чергу активно збирають і аналізують інформацію про дії користувача в мережі, формують персоналізовані пропозиції, які максимально відповідають його інтересам. Таким чином, користувач стає не лише споживачем інформації, а й цінним джерелом даних для самої системи, що дозволяє їй вдосконалюватися та адаптуватися.

Яскравим прикладом впровадження біометричних технологій у повсякденне життя є система FaceID від компанії Apple. За допомогою спеціальних сенсорів і камер смартфон створює унікальну 3D-модель обличчя користувача, яка служить ключем для біометричної аутентифікації. Це не тільки підвищує рівень безпеки, а

й спрощує доступ до пристроїв та сервісів, інтегрованих в екосистему компанії. Водночас це демонструє, наскільки тісно сучасна людина пов’язана із своїми біометричними даними та наскільки важливим є питання їх надійного захисту.

Застосування систем розпізнавання облич на основі штучного інтелекту стрімко поширюється у різних сферах — від забезпечення безпеки до оптимізації доступу до приміщень чи послуг. Головною умовою ефективної роботи таких систем є розробка зручних, інтуїтивно зрозумілих інтерфейсів, що забезпечують якісну взаємодію між апаратною частиною, камерами та програмним забезпеченням. Сучасний рівень розвитку штучного інтелекту дозволяє не лише розпізнавати риси обличчя в реальному часі, а й аналізувати великі потоки відеоданих, забезпечуючи оперативну і точну ідентифікацію.

Важливим аспектом підвищення ефективності таких систем є використання граничних обчислень, коли попередня обробка даних здійснюється безпосередньо на пристрої edge device, підключеному до камери. Це дозволяє зменшити навантаження на центральні сервери, прискорити процес розпізнавання та зробити систему більш гнучкою і масштабованою.

Детальні наукові дослідження в галузі створення інтерфейсів для систем розпізнавання облич на основі штучного інтелекту мають фундаментальне значення для розвитку сучасних інформаційних технологій. Вдосконалення методів передачі та обробки даних з камер, оптимізація алгоритмів машинного навчання та підвищення точності розпізнавання — усе це формує основу для впровадження безпечних, надійних і зручних систем ідентифікації, які поступово стають невід’ємною частиною нашого повсякденного життя.

1.2 Архітектура системи SmartHome Lock

1.2.1 Розробка загальної моделі роботи програми

Система SmartHome Lock (рис.1.5) побудована на модульній архітектурі, де кожен компонент має чітко визначені функції та не перетинається з іншими зайвими завданнями. Організація нагадує добре структуровану лабораторну установку, де кожен прилад відповідає своєму етапу експерименту.

Клієнтський додаток, реалізований як скрипт на Python під назвою client.py , виступає інтерфейсом для кінцевого користувача. Його основна роль передбачає забезпечення взаємодії між оператором та системою за допомогою консольного інтерфейсу. Додаток наступних команд, які вводять користувача, після чого надсилає їх до Telegram Bot API (рис.1.4). Для виконання HTTP-запитів використовується сучасна асинхронна бібліотека aiohttp, що дозволяє досягти високої продуктивності та зменшити затримку при обробці запитів. Відповіді, які надходять від Telegram Bot API, обробляються в реальному часі, що надає користувачу можливість оперативно реагувати на зміни стану системи.

Центральною ланкою виступає Telegram Bot, який реалізований у вигляді окремого Python-скрипту з іменем bot.py . Компонент можна порівняти із сервером, який після цієї команди від клієнтського додаток аналізує їх та виконує відповідні дії. До завдань входити керування станом замка, тобто визначення, яке має бути закрито відкритим або закритим у певний момент часу. Для забезпечення високого рівня безпеки Telegram Bot також реалізує механізм авторизації користувачів на основі біометричних даних. Крім того, використовується бібліотека DeepFace для розпізнавання обличчя, що дозволяє автоматично ідентифікувати користувача за зображенням.

Зберігання біометричних даних організовано у вигляді файлової системи, де шкірне відоме системне обличчя представлено окремим графічним файлом. Усі ці зображення розміщені в каталозі unknown_faces, шлях до якого можна змінити через параметр Config.KNOWN_FACES_DIR у конфігураційному файлі. Для ідентифікації власника кожного файлу надається ім'я у форматі user_id.jpg. Такий підхід дозволяє швидко розмістити потрібне зображення під час авторизації та безпеки відповідності між користувачами та його біометричними даними.

Загалом будівельна система SmartHome Lock забезпечує чіткий розподіл завдань між компонентами, що забезпечує її надійність, масштабованість і зручність використання.

```
123456789:ABCdefGHIJKlmNoPQRsTUVwxyZ
```

Рисунок 1.4 Приклад вигляду Telegram Bot API

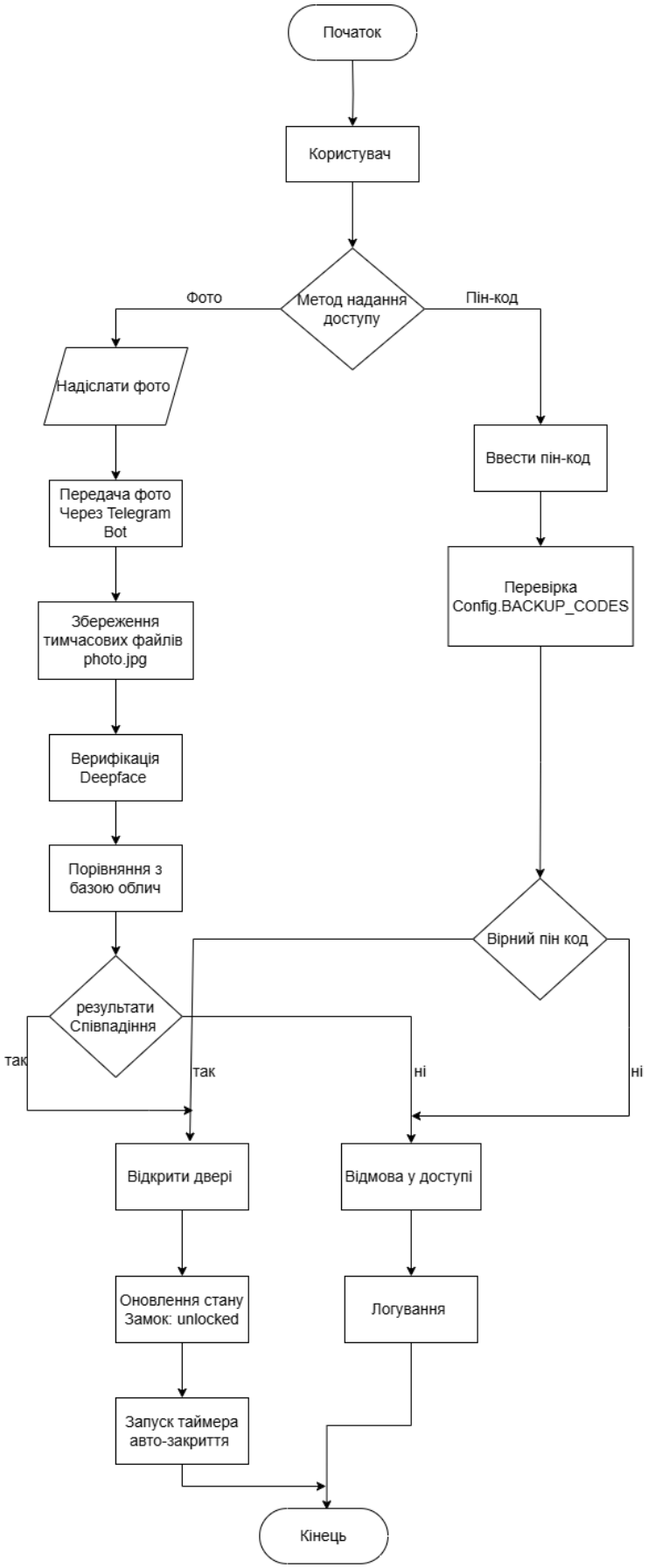


Рисунок 1.5 Блок-схема роботи SmartHome Lock

1.2.2 Вибір технологій для реалізації

Вибір мови програмування зупинився на Python ве. Цей вибір обумовлений передусім високою гнучкістю та простою інтеграцією як із Telegram API, так і з сучасними бібліотеками машинного навчання. Python має багатий набір бібліотек, які полегшують розробку, скорочують час на реалізацію функцій та гарантують підтримку актуальних стандартів безпеки.

Щодо бібліотек, серед основних використання python telegram bot. Саме ця бібліотека дозволяє працювати з Telegram Bot API в асинхронному режимі, забезпечуючи високу продуктивність і масштабованість рішення. Друга ключова бібліотека — DeepFace, яка є загальною з найпотужніших у сфері розпізнавання облич. Вона підтримує низку популярних моделей для ідентифікації, зокрема Facenet та VGG Face, що дозволяє гнучко оптимально вибрати під завдання архітектури для досягнення високої точності.

Для здійснення мережевих запитів використовується aiohttp , переваги можуть уможливити асинхронні HTTP-запити, що особливо важливо у взаємодії з клієнтськими пристроями або зовнішніми службами через API. Це суттєво створює швидку систему та дозволяє ефективно обробляти великий обсяг даних.

Ще одним елементом у сучасній розробці є використання бібліотеки dotenv. Вона значно спрощує роботу з конфігураційними змінними, зберігаючи їх в окремому файлі формату .env. Такий підхід лише покращує організацію проєкту, а й суттєво завершує безпеку системи. Зберігання чутливих даних, як-от токенів доступу чи криптографічних ключів, поза основним кодом дозволяє уникнути їх випадкового потрапляння у відкриті репозиторії або сторонні руки. Це особливо актуально для командної роботи, коли кілька розробників мають доступ до коду — так кожен може мати власний файл змінного середовища, не ризикуючи розкрити приватну інформацію. Щодо шифрування даних, варто відзначити, що застосування стандарту AES 256 є загальноприйнятою практикою у сфері інформаційної безпеки. Алгоритм симетричного шифрування визнаний одним із найбільш надійних і широко використовуваних для захисту конфіденційної інформації як у промисловості, так і в наукових дослідженнях.

1.3. Реалізація серверної частини

1.3.1. Алгоритм роботи бота

Алгоритм роботи бота(рис.1.7) працює наступним чином. На початковому етапі, коли користувач надсилає команду /start, бот реагує автоматично. Він надсилає інструкції, що містять основну інформацію щодо подальших дій, які користувач може виконати.

Далі реалізується процедура розпізнавання обличчя. Користувач надсилає своє фото, яке бот зберігає у тимчасовий файл, що має унікальну назву, сформовану з ідентифікатора користувача. Цей підхід дозволяє уникнути плутаніни між різними сесіями та забезпечити індивідуальність обробки кожного зображення. Після збереження файлу бот використовує функцію DeepFace.verify для порівняння отриманого фото з базою відомих облич. Саме цей етап є ключовим для забезпечення безпеки та ідентифікації особи. Якщо співпадіння підтверджено і обличчя розпізнано вірно, система ініціює відкриття дверей на проміжок часу, заданого в параметрі Config.AUTO_LOCK_DELAY. Тривалість відкриття дверей збільшиться цим параметром і після завершення одного часу двері знову автоматично блокуються. Таким чином, автоматизація цього процесу забезпечує як зручність, так і додатковий рівень безпеки.

Щодо додавання нового обличчя, ця функція доступна тільки за допомогою адміністратора, ідентифікованого за ADMIN_USER_ID. Для запуску процедури адміністратор використовує команду /add_face. У цей момент є ConversationHandler, який організовує діалог і контролює запуск наступних етапів взаємодії: спочатку ініціалізується стан ADD_FACE, далі відбувається підтвердження через CONFIRM_FACE. Система поступово проводить адміністратора через усі необхідні кроки, що мінімізує ризик помилок при додаванні нових осіб до бази даних, алгоритм по використанню простоту використання для кінцевого цього користувача з високим рівнем безпеки та контролю з боку адміністратора. Інтеграція сучасних методів розпізнавання облич та структурована система доступу дозволяє ефективно вирішуватизавдання ідентифікації та контролю доступу.

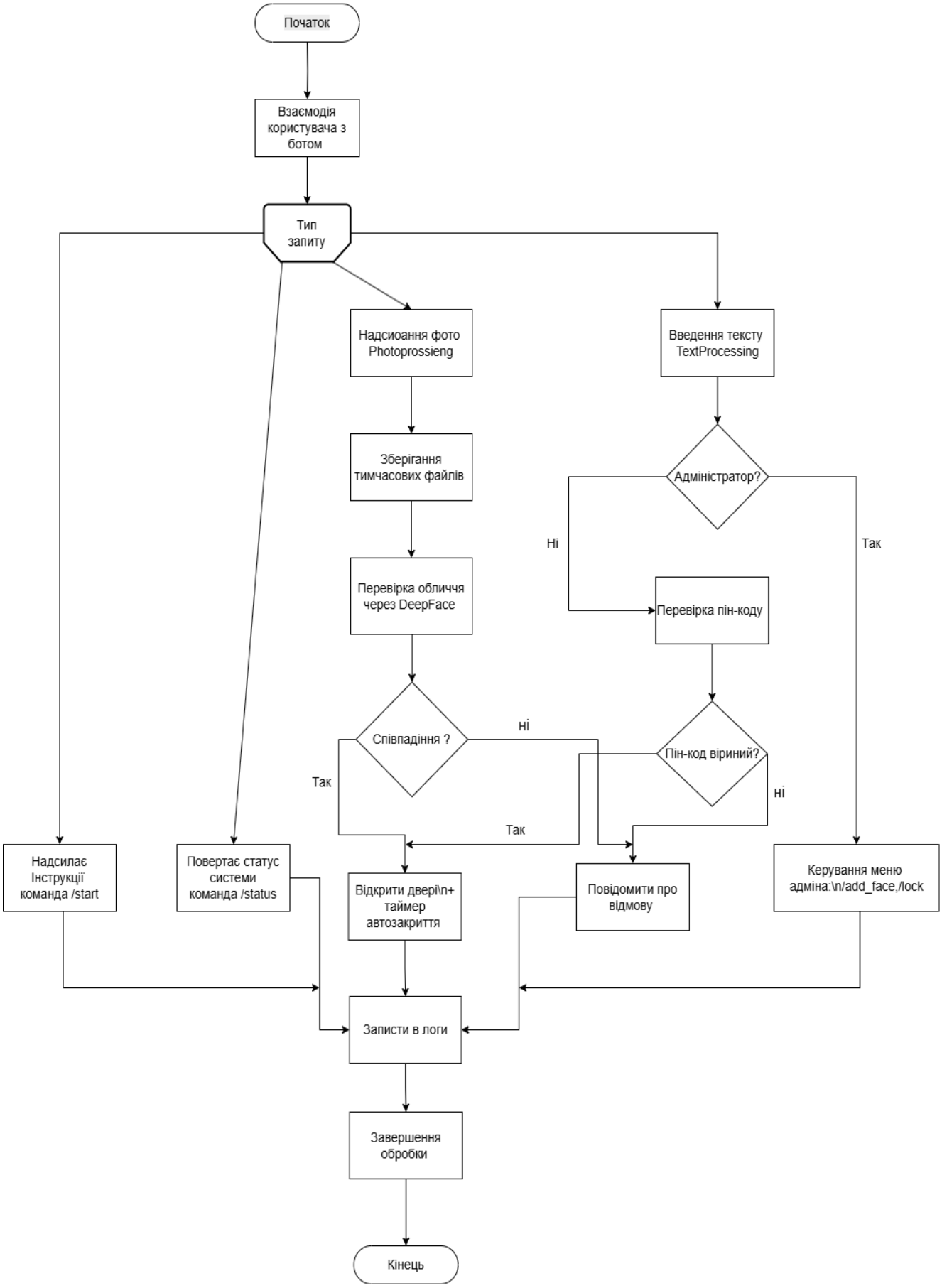


Рисунок 1.7 Блок-схема алгоритму роботи Telegram-бота SmartHome Lock

1.3.2. Аналіз шифрування даних

Шифрування організовано досить надійно. Для створення ключа AES не використовується звичайний рядок, а спочатку функція SHA-256(рис.1.8). Джерелом служить ENCRYPTION_KEY, який обробляється цим алгоритмом для отримання хешу, а вже цей хеш і стає ключем для симетричного шифрування. Такий підхід неможливий для використання простих або слабких паролів, хоча навіть такі значення на вході дають абсолютно різні хеші. У коді цей процес реалізований через метод Config.get_encryption_key, який автоматизує всі ці перетворення.

Щодо безпеки зображень із зображеннями обличчя, тут застосовав додатковий рівень безпеки. Фотографії не передаються у відкритому вигляді в Інтернеті чи хмарі, вони зберігаються лише локально на пристрої, що значно знижує ризик несанкціонованого доступу до даних. Якщо виникає потреба передати таку фотографію через Telegram, тут працює власний захищений протокол MTProto. Завдяки цьому протоколу інформація під час передачі шифрується, і перехопити її або розшифрувати дані фактично неможливо без спеціальних знань та ресурсів. У результаті, навіть якщо хтось намагається втрутитися в процес передачі, він отримує лише зашифровану інформацію, до якої не зможе отримати доступ.

В цілому така модель забезпечення інформаційної безпеки відповідає сучасним вимогам і дозволяє мінімізувати результати загрози при роботі з чутливими даними, такими як фотографії обличчя. Варто відзначити, що використання комбінації симетричного шифрування, хешування та протоколів передачі даних значно забезпечує загальний рівень захисту системи.

```
def get_encryption_key():
    key = os.getenv("ENCRYPTION_KEY", "default_key_1234567890")
    if len(key) < 32:
        key = sha256(key.encode()).hexdigest()
    return key[:32].ljust( _width: 32,  _fillchar: '0').encode()
```

Рисунок 1.8 Реалізація генерації AES-ключа

1.3.3. Механізми захисту від атак

Механізми захисту від атак — насправді це не просто якісь сухі технічні налаштування, а цілісна система запобігання несанкціонованому доступу. Перше що помічає користувач є обмеження кількості спроб введення пін-коду. Якщо користувач тричі вводить невірний код, система автоматично припиняє приймати подальші спроби. Це зменшує ймовірність перебору комбінацій зловмисниками та ускладнює brute-force атаки.

Другий рівень захисту пов'язаний із часовими обмеженнями. Після трьох невдалих спроб система блокує можливість наступного введення пароля на п'ять хвилин. Така затримка суттєво гальмує потенційного зловмисника, змушує чекати й фактично унеможливлює масові автоматизовані атаки. Це як поставити сигналізацію не лише на двері, а ще й на вікна.

Крім того, для підвищення надійності передбачено використання резервних кодів. Наприклад, якщо користувач забув основний пін-код або сталася якась форс-мажорна ситуація, він може скористатися спеціальними резервними комбінаціями. Перелік таких кодів формується заздалегідь і зберігається окремо, що дозволяє забезпечити доступ до системи навіть у випадках непередбачуваних збоїв.

Загалом, подібна багаторівнева система захисту є стандартом для сучасних безпекових рішень. Вона базується на поєднанні обмеження спроб, часових затримок та альтернативних способів доступу, що в комплексі значно ускладнює несанкціонований вхід для сторонніх осіб.

1.4. Розробка клієнтського інтерфейсу

1.4.1. Реалізація інтерактивного меню

Меню інтерактивне(рис.1.9), реалізовано в методі show_menu класу SmartHomeClient(рис1.10). Користувач бачить два базові варіанти. Перший, відкрити двері за допомогою пін-коду. Тут все просто: система запитує у користувача пін-код, який складається з чотирьох цифр. Це стандартна процедура, знайома багатьом із банкоматів або систем безпеки.

Другий варіант, ідентифікація через розпізнавання обличчя. Якщо

користувач обирає цю опцію, система надає інструкцію, як пройти авторизацію. Може бути, потрібно зробити фотографію в реальному часі або завантажити вже наявне зображення. Далі система складає це фото з даними у базі осіб, яким дозволено вхід. Результат або відкритий доступ, або користувач отримує відповідне повідомлення про невдалу авторизацію.

Варто зазначити, що такий підхід дає можливість підвищити рівень безпеки та зручності. Користувач самостійно обирає спосіб, який йому більше підходить. Обидва методи мають свої переваги: пін-код простий у використанні, а розпізнавання обличчя знижує ризик несанкціонованого доступу.

Загалом, меню дозволяє ефективно організувати процес входу та зробити його максимально адаптивним до різних потреб користувача.

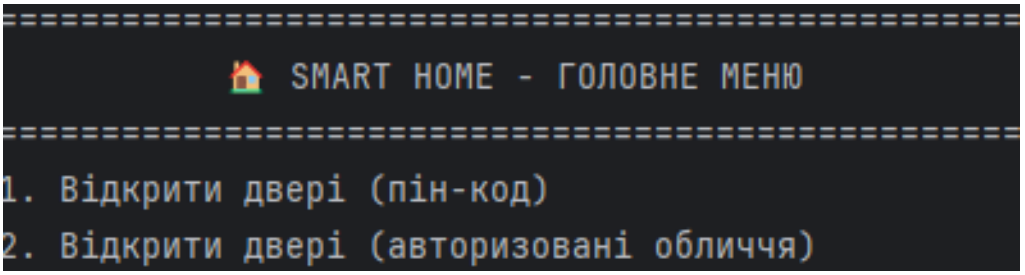


Рисунок 1.9 Первинна реалізація інтерактивного меню в терміналі

```
print("\n" + "=" * 50)
print("🏠 SMART HOME - ГОЛОВНЕ МЕНЮ".center(50))
print("=" * 50)
print("1. Відкрити двері (пін-код)")
print("2. Відкрити двері (авторизовані обличчя)")
```

Рисунок 1.10 Демонстрація інтерактивного меню в коді

1.4.2. Логування подій

Сервер збереження журналів у файлі smart_home.log, для чого підтримується детальний рівень реєстрації — DEBUG або INFO, (рис 1.11) залежно від налаштування системи. Це означає, що до журналу потрапляють як технічні події, пов’язані з роботою пристроїв або системних компонентів, так і основні дії користувачів, наприклад, запуск сценаріїв чи зміну налаштувань. Такий комплексний підхід дозволяє отримати максимально повну інформацію про стан

системи в будь-який момент часу.

Записи в журналі мають чітко структурований формат: кожен рядок містить дату і час джерела, події, події (наприклад, конкретний пристрій або програмний модуль), рівень важливості (DEBUG, INFO тощо) і короткий опис суті подій. Завдяки цьому забезпечується не лише зручність пошуку й аналізу інформації, а й можливість автоматизації обробки журналу за допомогою спеціалізованих інструментів.

Ведення таких детальних журналів здійснює ключову роль у системі моніторингу роботи розумний дім. З іншого боку, це дозволяє виявити і швидко локалізувати причини можливих збоїв або неочікуваних ситуацій, що особливо важливо для підтримки стабільності системи. З іншого боку, на основі накопичених логів можна проводити глибокий аналіз системи — виявляти закономірності в поведінці пристроїв і користувачів, оптимізувати сценарії автоматизації, а також прогнозувати деякі проблемні місця в роботі.

Різного серверного журналу, клієнтська частина також має власний механізм фіксації подій — історію команд (command_history). Тут зберігається інформація про точний час виконання кожної команди, її статус (успішна, з помилкою тощо), а також додаткові параметри, які можуть бути корисними для подальшого аналізу. Цей підхід забезпечує користувачу прозорість власних дій у системі: можна в будь-який момент переглянути, які саме команди виконувалися, порівняти сценарії управління та вирішувати малоефективні або проблемні дії. Збереження історії команди сприяє не тільки зручності користування, але й підвищенню загального рівня безпеки — адже користувач може оперативно відреагувати на відчуття або несанкціоновані дії. Крім того, аналізуючи командний журнал, можна визначити тенденції системи у використанні, адаптувати під себе автоматизацію і, зрештою досягти максимальної ефективності та комфорту від використання розумного дому.

Загалом, впровадження багаторівневої системи журналювання подій і дій користувачів є невід’ємною частиною сучасних розумних систем, що забезпечує як технічний контроль і підтримку так і підґрунтя для подальшого вдосконалення.

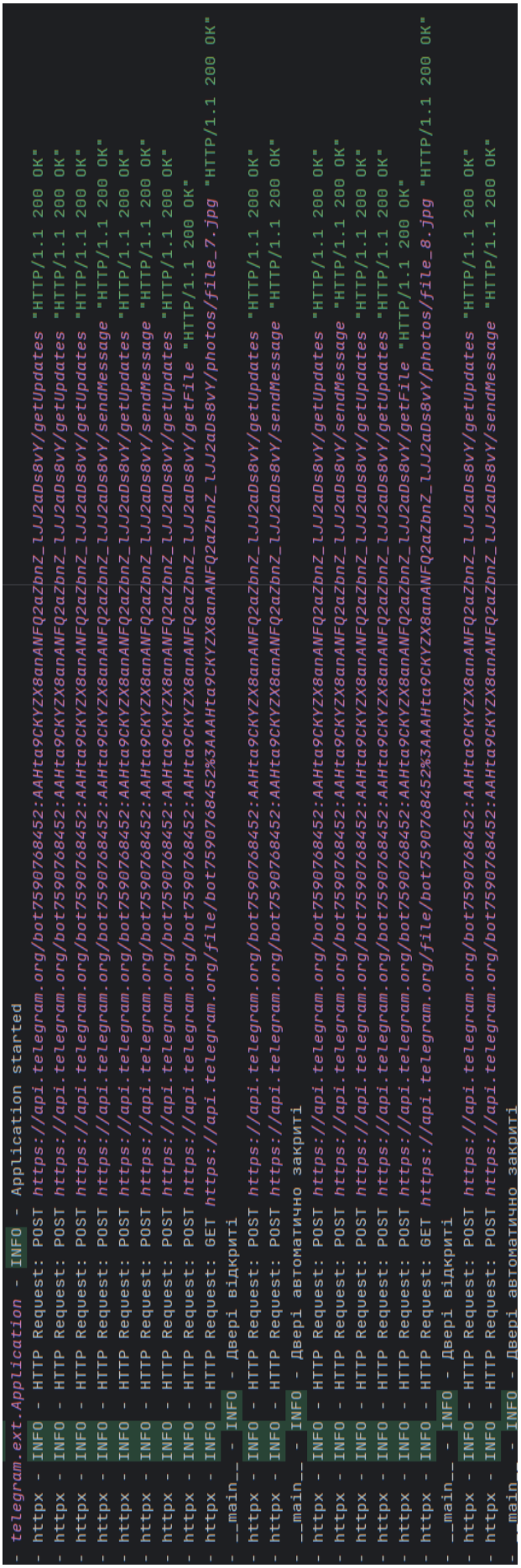


Рисунок 1.11 Логування подій в системі SmartHome Lock

1.5. Система безпеки

У цій системі безпеки використовується двофакторна аутентифікація, що складається з двох наступних етапів. Спочатку користувач вводить пін-код, який є першим бар'єром для несанкціонованого доступу. Якщо система виконує розпізнавання обличчя, що забезпечує значний рівень захисту особи, навіть у разі компрометації пін-коду не можна отримати доступ без фізичної присутності власника. Такий підхід дозволяє значно зменшити ймовірність несанкціонованого проникнення до системи.

Механізм автозакриття реалізовано через параметр AUTO_LOCK_DELAY, який для замовчування встановлено на десять секунд. Тому, навіть якщо користувач випадково залишає двері відкритими, система автоматично ініціює їх блокування після короткої затримки. Такий підхід суттєво знижує ризик того, що двері залишаються незамкненими, а отже, мінімізує ймовірність несанкціонованого доступу до приміщення. До речі, це не лише зробити безпеку, а й зручно з практичної точки зору — зайвий раз не потрібно хвилюватися, замкнути чи ні двері, особливо коли поспішаєш.

Що ж захищає системи захисту від атак типу DDoS, тут реалізовано багаторівневий підхід. По-перше, система встановлює обмеження на кількість запитів до Telegram API, що дозволяє зберегти стабільність сервісу навіть у випадку спроби масового перезавантаження. Такий міра є надзвичайно важливою, оскільки атаки подібного типу можуть призвести до повної недоступності систем для легітимних користувачів. По-друге, сам Telegram має вбудовані механізми захисту — це і обмеження на частоту запитів, і алгоритми відстеження негативної активності ботів та потенційних зловмисників. У сукупності цих заходів забезпечується надійна працездатність системи навіть під час пікових навантажень та підвищеної активності зі сторони негативних атакуючих. Такий багатошаровий захист сьогодні є стандартом для безпечних автоматизованих систем, особливо якщо йдеться про інтеграцію з популярними зовнішніми платформами, як-от Telegram.

1.6. Тестування системної моделі SmartHome Lock

Адміністратор отримує повний доступ до функціональної системної моделі. Мається на увазі , що йому відкриті всі можливості управління та налаштування. Для прикладу, реалізовано меню в терміналі на Python, (рис. 1.12) яке дозволяє адміністратору взаємодіяти із системою в інтерактивному режимі. Такий підхід забезпечує зручність і ефективність роботи з моделлю. (рис. 1.12)

```
==================================================
        🏠 SMART HOME Lock - ГОЛОВНЕ МЕНЮ
==================================================
1. Відкрити двері (пін-код)
2. Відкрити двері (авторизовані обличчя)
3. Перевірити статус системи
4. Історія команд
5. Вийти
==================================================
🛠 Ваш вибір:
```

Рисунок 1.12 Відображення інтерфейсу SmartHome Lock в терміналі

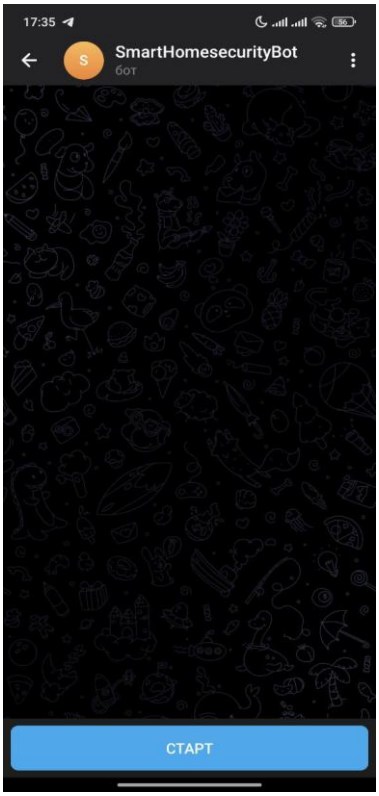


Рисунок 1.13 Відображення інтерфейсу Telegram Bot на мобільному пристрої

Рисунок 1.14 Відображення інтерфейсу Telegram Bot у десктопному застосунку

У цій програмній моделі (рис. 1.15) тільки користувач із правами адміністратора має можливість додавати інших користувачів за допомогою команди (/add_face). На (рис. 1.16–1.18) представлено, як відбувається процес авторизації . Варто відзначити, що наступні приклади обличь створені штучним інтелектом і не є існуючими у реальному просторі.

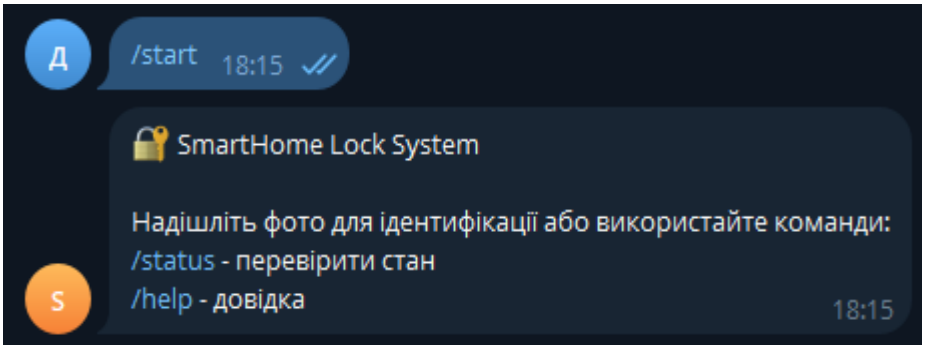


Рисунок 1.15 Інтерфейс Telegram Bot при старті роботи

Рисунок 1.16 Перше додавання користувача створеного за допомогою штучного інтелекту

Варто відзначити, що наступні приклади обличь створені штучним інтелектом і не є існуючими у реальному просторі.

Рисунок 1.17 Друге додавання користувача створеного за допомогою штучного інтелекту

Рисунок 1.18 Третє додавання користувача створеного за допомогою штучного інтелекту

Як видно на (рис 1.16–1.18), всі умовні користувачі, створені за допомогою штучного інтелекту, успішно пройшли тестування без жодних помилок чи збоїв у процедурі. Це підтверджує стабільність роботи системи на етапі імітаційного тестування та демонструє працездатність алгоритмів верифікації користувачів навіть у випадках, коли дані були згенеровані штучно.

Далі, у процесі тестування авторизації (рис. 1.19–1.20) також застосовували цих умовних користувачів, створених штучним інтелектом. Це дозволило оцінити, наскільки система SmartHome Lock здатна коректно ідентифікувати користувачів у різних сценаріях, включаючи можливі неточності вхідних даних. У моделі реалізовано біометричну аутентифікацію: користувач або завантажує готове фото, або робить селфі у реальному часі. Далі, система проводить порівняння отриманого зображення з даними, що зберігаються у локальній закритій базі, після чого формує відповідь щодо успішності чи невдалості авторизації.

Рисунок 1.19 Перше тестування авторизації за допомогою користувачів створених за допомогою штучного інтелекту

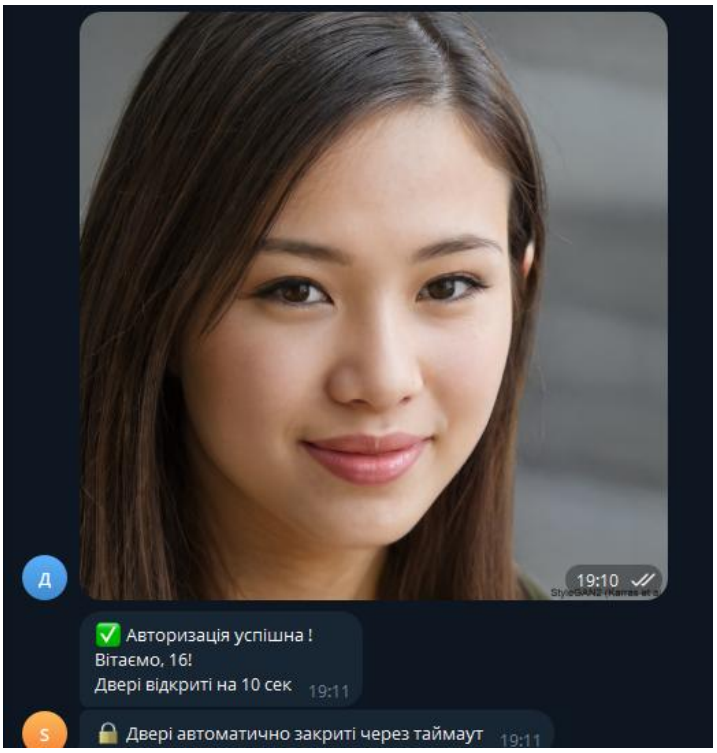


Рисунок 1.20 Друге тестування авторизації за допомогою користувачів створених за допомогою штучного інтелекту

Рисунок 1.21 Третє тестування авторизації за допомогою користувачів створених за допомогою штучного інтелекту

На (рис 1.19–1.21) можна чітко спостерігати, що всі умовні користувачі, створені за допомогою штучного інтелекту, успішно пройшли процедуру

авторизації. Жодних технічних збоїв чи помилок при цьому не виникло, що працює про стабільність роботи системи під час тестування на штучно згенерованих обличчях. Цей важливий етап, оскільки моделювання можливих сценаріїв дозволяє переконатися в критичних вразливостях на ранніх стадіях.

Далі під час роботи системи перевірки реальних даних (рис. 1.22-1.23), тобто при авторизації за допомогою справжнього обличчя людини, результат залишився позитивним: процедура завершилася успішно, і жодних помилок не було зафіксовано. Це передбачає, що розроблений алгоритм коректно функціонує не тільки в лабораторних умовах, а й при взаємодії з живими користувачами саме це і демонструє ключовий показник якості.

Ще одним аспектом є збереження моделі обличчя в (рис. 1.24) локальній папці операція також відбулася без ускладнень: модель була записана правильно, і система не видала жодних попереджень чи помилок. Така стабільність на всіх етапах (від авторизації до збереження моделі) про комплексну надійність розробленого програмного забезпечення підтверджує його готовність до подальшого впровадження та масштабування.

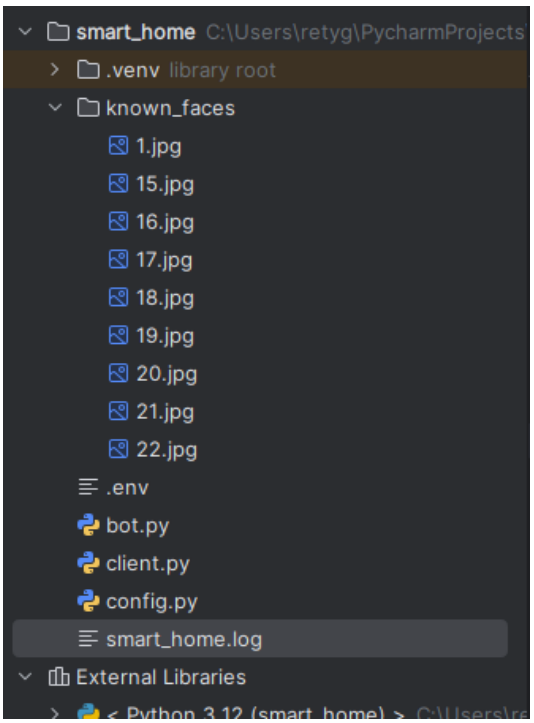


Рисунок 1.24 Локальна папка для збереження моделей обличь

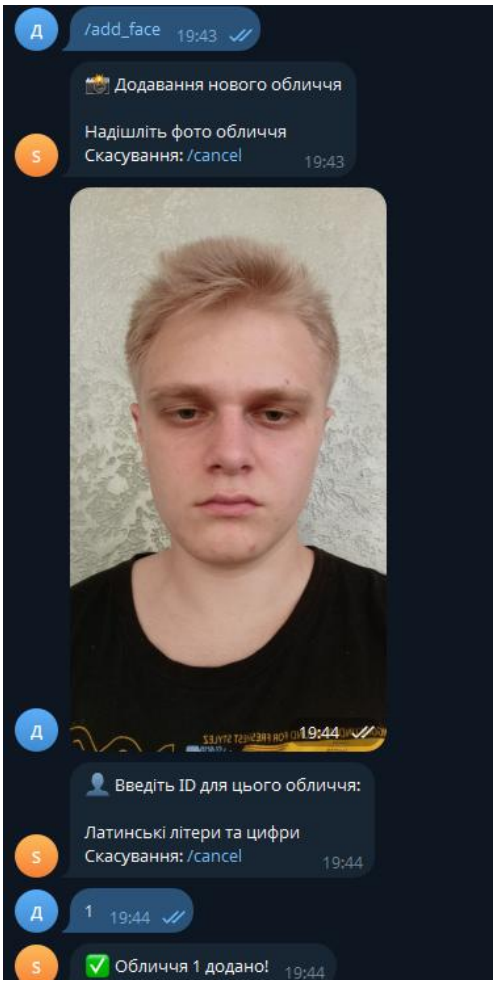


Рисунок 1.23 Додавання користувача

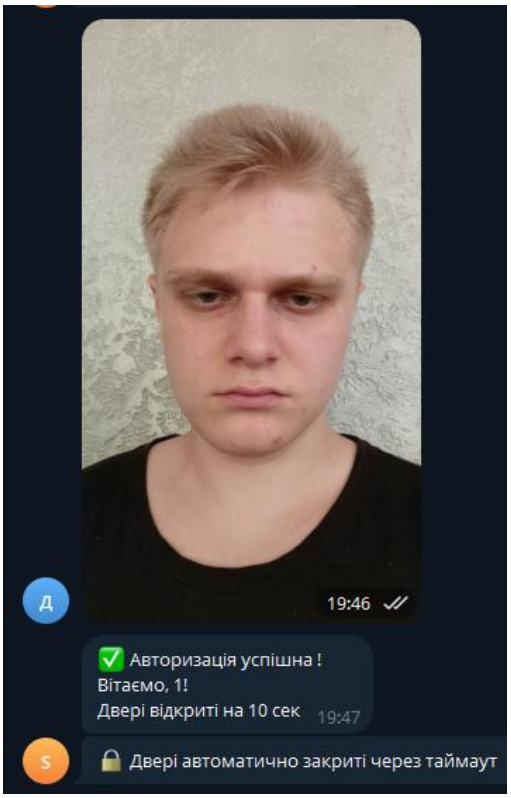


Рисунок 1.24 Авторизація користувача

Під час проведення тестування функції відкриття дверей із використанням BACKUP_CODES результат виявився позитивним: жодних збоїв чи непопереджених проблем не спостерігалося. Система працює чітко, як і було задумано, що підтверджується на (рис. 1.25) особливо мене приємно здивувала стабільність роботи — навіть під час кількох повторних спроб все працювало без збоїв. Можна зробити висновок, що реалізований механізм резервного доступу дійсно надійний і може використовуватися як ефективний спосіб у випадку втрати основного коду чи поломки основного способу доступу.(рис. 1.26)

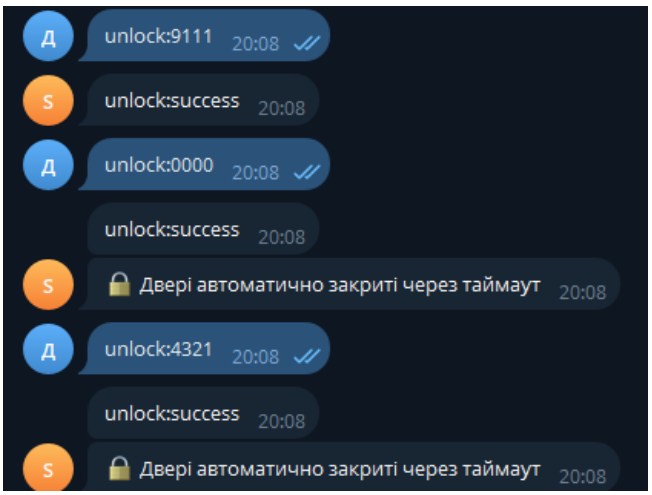


Рисунок 1.25 Відкривання дверей через BACKUP_CODES

Рисунок 1.26 Приклад помилки при неправильному пін-коду BACKUP_CODES

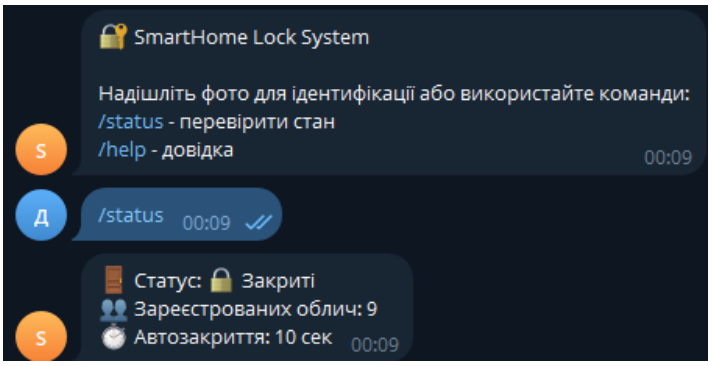


Рисунок 1.27 Перевірка статусу дверей

Рисунок 1.28 Логування подій під час тестування системної моделі SmartHome Lock

Таблиця 1.1 Сценарії використання SmartHome Lock

Сценарії	**Очікуваний результат**	**Критерій успіху**
Користувач надсилає вірне фото	Двері відкриваються на 10 сек	Статус locked:True у відповіді
Невідоме обличчя	Повідомлення "Доступ заборонено"	Код відповіді 200, правильний текст
Введення резервного коду	Миттєве відкриття дверей	Лог-файл містить запис про успіх
5 невдалих спроб пін-коду	Тимчасова блокування (5 хв)	Наступні запити відхиляються
Спробу відкриття під час блокування	Повідомлення "Система тимчасово заблокована"	Відсутність запитів до DeepFace під час блокування
Адмін-команда /add_face	Додавання нового обличчя до бази	Збільшення len(lock_system.known_faces) на 1

Коли користувач надсилає вірне фото, система повинна відкрити двері на 10 секунд. У відповідь сервер повертає статус locked:True, що підтверджує успішне розпізнавання обличчя та виконання команди.

У випадку, якщо система отримує зображення невідомого обличчя, вона повинна надіслати повідомлення «Доступ заборонено». При цьому відповідь сервера матиме статус 200 (HTTP-запит) і міститиме текст повідомлення.

1.7. Перспективи розвитку моделі

Перспективи подальшого розвитку моделі виглядають досить багатообіцяючими та заслуговують на серйозну увагу.

По-перше, значно розширити функціонал можна шляхом впровадження додаткових методів аутентифікації. Окрім стандартних паролів, цілком можливо використовувати біометричні дані, наприклад відбитки пальців, а також RFID-картки. Такі підходи забезпечують вищий рівень безпеки, що особливо актуально в умовах зростання кіберзагроз. Варто зазначити, що інтеграція декількох методів аутентифікації дозволяє створити багаторівневу систему захисту, яка суттєво ускладнює несанкціонований доступ.

По-друге, ще одним напрямом розвитку є інтеграція моделі з екосистемами Інтернету речей. Підключення до Home Assistant через MQTT відкриває доступ до широкого спектра смарт-пристроїв і дозволяє створити цілісну систему керування домашньою автоматикою. Такий підхід не лише підвищує рівень комфорту користувача, а й сприяє більш ефективному розподілу ресурсів і оптимізації енергоспоживання.

Не менш важливим є вдосконалення інтерфейсу користувача. Відмова від суто консольного управління на користь графічного інтерфейсу, створеного на основі сучасних бібліотек, таких як PyQt, значно покращує взаємодію з системою. Візуалізація даних та інтуїтивно зрозумілий дизайн інтерфейсу сприяють зростанню популярності продукту серед ширшого кола користувачів, у тому числі тих, хто не має спеціальних технічних навичок.

Щодо архітектурних рішень, особливої уваги заслуговує перехід до розподіленої структури. Використання мікросервісів дозволяє розділити функціонал системи на незалежні компоненти. Наприклад, окремий сервіс для розпізнавання облич може працювати автономно, що спрощує масштабування системи та її обслуговування. Така модульність не лише підвищує надійність, а й дозволяє швидко впроваджувати нові функції без ризику для стабільності всієї моделі.

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

В даному дипломному проекті розроблено програмний продукт (ПП) – програмну модель SmartHome Lock, яка забезпечує безпечний контроль доступу до приміщення за допомогою біометричної автентифікації, розпізнавання обличчя та резервних пін-кодів.

Програмна модель створювалась з урахуванням сучасних методів розробки, що забезпечує її ефективність і робить використання максимально зручним.

Система SmartHome Lock відрізняється високою точністю верифікації, захистом від несанкціонованого доступу та автоматичним керуванням станом замка. У розділі детально пояснюється економічний аспект розробки, зокрема витрати коштів, ресурсів і часу, які вимагають створення програми.

2.2 Розрахунок ціни програмного продукту нормативним методом

2.2.1 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного забезпечення серйозно залежить від складності самого проекту, обсягу робіт, трудомісткості окремих етапів, часу, кількості обмежених ресурсів та рівня підготовки команди розробників. Для попередньої оцінки часто застосовують метод структурного порівняння. Він обґрунтовується на аналізі вже реалізованих подібних проектів із близьким визначенням обсягу необхідного коду.

З розрахунку отриманого значення встановлюється норма часу на розробку аналогічного програмного забезпечення. Враховуються умови виконання проекту, це значення встановлюється відповідним коефіцієнтом (Кк = 0,7÷0,8). Таким чином, розрахункова трудомісткість становить: $T^{a}р = 262 \times 0,7 = 183,4$ год/люд.

Трудомісткість запуску для кожного етапу окремо, орієнтуючись на показники трудових витрат для подібних програмних продуктів. При цьому враховують складність реалізації, рівень новизни проекту

- Розробка технічного завдання

$$T_{ТЗ} = T^a p \times L_1 \times K_H \quad (2.1)$$

- Розробка технічного проекту

$$T_{ТП} = T^a p \times L_2 \times K_H \quad (2.2)$$

- Розробка робочого проекту

$$T_{РП} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

Li – питома вага і-го етапу розробки (див. табл. 2.1);

Кн – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.2);

Кт – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм Обрані варіанти виділено синім кольором. (див. табл. 2.3).

Таблиця 2.1 Значення питомих коефіцієнтів трудомісткості стадії

Код стадії	Ступінь новизни		
	А	Б	**В**
ТЗ (L_1)	0,15	0,12	**0,12**
ТП (L_2)	0,16	0,15	**0,11**
РП (L_3)	0,55	0,58	**0,61**

Таблиця 2.2 Значення поправочного коефіцієнта

Код ступеня новизни	Ступінь новизни	Значення $К_н$
А	Принципово нові ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
В	**ПП маючий аналог**	**0,7**

Оскільки розроблювана система є програмним забезпеченням, що має існуючі аналоги, для мого ПЗ встановлено код ступеня новизни “В”, а коефіцієнт $К_н = 0,7$.

Відповідно до таблиці 2.2, знаючи код ступеня новизни, тепер можна визначити значення питомих коефіцієнтів трудомісткості за таблицею 2.1.

L_1=0,12 L_2=0,11 L_3=0,61

Таблиця 2.3 Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, %	Значення $К_т$
60 і вище	0,6
40-60	0,7
20-40	**0,8**
До 20	0,9

У розробленому програмному продукті використовується від 20 до 40 відсотків існуючих функцій, це значить, що $К_т$= 0,8 Тепер потрібно розрахувати трудомісткість по кожному етапу окремо:

- Трудомісткість технічного завдання

$$Т_{тз}=T^a p *L_1*К_н= 183,4*0,12*0,8 = 17,3 \quad \text{(люд/годин)}$$

- Трудомісткість розробки технічного проекту

$$Т_{тп}=T^a p *L_2*К_н = 183,4*0,11*0,8 = 17,6 \quad \text{(люд/годин)}$$

- Трудомісткість розробки робочого проекту

$$Т_{рп}=T^a p *L_3*К_н*К_т= 183,4*0,61*0,8 = 89,1 \quad \text{(люд/годин)}$$

Таблиця 2.4 Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, години.		
1	Розробка ПП	Контроль керівника	Нормоконтроль
1.ТЗ	$Т_{Ртз}$=**17,3**	$Т_{кк}=0,7*N_{тз}=0,7*2=$**1,4**	$Т_{нк}=0,15*N_{тз}=0,15*2=$**0.3**
2.Розробка ТП	$Т_{Ртп}$=**17,6**	$Т_{кк}=0,7*N_{тп}=0,7*40=$**28**	$Т_{нк}=0,15*N_{тп}=0,15*40=$**6**
3.Розробка РП	$Т_{Ррп}$=**89,1**	$Т_{кк}=0,7*N_{рп}=0,7*10=$**7**	$Т_{нк}=0,15*N_{рп}=0,15*10=$**1.5**
4.Розробка пояснювальної записки	$Т_{рпз}=1,5*N_{пз}$ = 1,5*20=**30**	$Т_{кк}=0,7*N_{тз}=0,7*20=$**14**	$Т_{нк}=0,15*N_{пз}=0,15*20=$**3**
Усього, в т.ч.:	$Т_{пп}$ =**215,2**		
- на розробку	$\Sigma Т_р$=**154**		
- контроль керівника		$\Sigma Т_{кк}$=**50,4**	
-нормоконтроль			$\Sigma Т_{нк}$=**10,8**

Для подальших розрахунків необхідно розрахувати кількість аркушів, витраченого на кожен етап. Технічне завдання: $N_{тз}$= 2 сторінки, технічний проєкт :$N_{тп}$= 40 сторінок, робочий проект – $N_{рп}$= 10 сторінок , пояснювальна записка $N_{пз}$= 20 сторінок – технічне завдання, розробка технічного проекту, розробка робочого проекту, пояснювальна записка відповідно. Розрахунок зведений у таблицю 2.4

На основі таблиці 2.4 розрахуємо тривалість розробки в роках:

$T_{пп}$=T/(8.0*0.73*360)= 183,4/(8.0*0.73*360)= 0,087 (р),

де:

8,0 – тривалість робочого дня;

0,73 – коефіцієнт перекладу в календарні дні

2.2.2 Розрахунок ціни програмного продукту

У цьому розділі детально розглянуто розрахунок вартості програмного продукту, зокрема: основної заробітної плати, матеріальних витрат, витрат на машино-години та загальних витрат. Відомості щодо заробітних плат наведені у таблиці 2.5 Важливо зазначити, що з 1 січня 2025 року мінімальна місячна заробітна плата в Україні складає 8000 грн, а погодинна ставка — 46 грн.

Таблиця 2.5 Розрахунок основної заробітної плати виконавців.

Найменування робіт	Трудомісткість робіт, роб.години	Годинна тарифна ставка,грн..	Розрахунок, грн.
1.Розробка ПП	$\Sigma T_р$=**154**	46	**7084**
2.Контроль керівника	$\Sigma T_{кк}$=**50,4**	46	**2318,40**
3.Нормоконт-роль	$\Sigma T_{кк}$=**10,8**	46	**496,80**
Усього ($З_о$)	-	-	Σ Зо= **9899,20**

Виконаємо розрахунок матеріальних витрат, необхідних для створення програмного продукту. Підсумкові дані наведено в таблиці 2.6.

Таблиця 2.6 Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість, шт	Ціна одиниці, грн.	Вартість, грн.
Папір А1	аркуш			
Папір А4		**72**	4	**288**
Разом	-	-	4	$В_{м\,і}$ = -
Транспортно – заготівельні Витрати 10%				$Втр\_з = 0{,}1 \times Вм1$= 0,1*288=**28,8**
Усього				$В_м=В_{м1}+В_{тр\_з=}$ **316,8**

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.7.

Таблиця 2.7 Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	**316,8**	$В_м$ (див. табл. 2.8)
2. Основна заробітна плата	**9899,20**	$З_о$ (див. табл. 2.7)
3.Додаткова заробітна плата	**989,92**	$Зд = 0{,}1 \times Зо$=0,1*9899,20
4.Відрахування до єдиного фонду соціального внеску	**2395,60**	$Вє.с.в. = 0{,}22 \times (Зо + Зд)$ =0,22*(9899,20+989,92)
5. Накладні витрати	**5939,52**	$Внак. = 0{,}6 \times Зо$=0,6*9899,20
6. Повна собівартість	**19540,04**	$С_{пов}= В_м +З_о + З_д +$Вє.с.в. + Внак. 316,80+9899,20+989,92+2395,60+5939,52

Розмір прибутку, що включається в ціну, визначається наступною формулою:

$П=(С_{пов}*Р)/100$ = (*10)/100 = (19540,04*10)/100=**1954** грн

Де Р– плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$Ц_о=С_{пов}+П$= 19540,04 +1954,00 = **21494,04** грн

Податок на додану вартість визначається по наступній формулі:

ПДВ = 0.2 * $Ц_о$ = 21494,04*0,2 = **4298,81** грн

Ціна реалізації програмного продукту на основі формули, становитиме:

$Ц_р=Ц_о+$ПДВ= 21494,04 +4298,81 *0,2 = **25792,85** грн

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

У цьому розділі йдеться про основні питання організації безпечних та комфортних умов праці для розробників програмного забезпечення. сподіваюсь на те, що ця професія остаточно не асоціюється з виробничими ризиками, все з певними аспектами, які можуть негативно впливати на здоров'я працівників у разі їхнього ігнорування.

Особливо важливо звертати увагу на ергономіку робочого місця. Йдеться про правильну організацію простору, розміщення комп'ютера, якісне освітлення та дотримання правил електробезпеки. Тривала робота за комп'ютером може спричинити перевтому очей, напруження опорно-рухового апарату і навіть зниження загальної працездатності. Тому всі ці фактори обов'язково потрібно використовувати під час організації робочого процесу.

Також розглядаються вимоги до мікроклімату в офісі, рівня шуму, освітлення, а також заходи пожежної та електробезпеки. Дотримання відповідних стандартів не лише відповідає законодавчим та галузевим нормам, а й є необхідними умовами для збереження здоров’я працівників та підвищення ефективності їх діяльності.

3.1 Шкідливі фактори, що впливають на розробника

Робота програміста супроводжується низькою несприятливими діячами, які можуть негативно вплинути на здоров'я людини. завдяки, тривалому перебування перед екранним монітором значного навантаження на органи зору. Це може збільшити втому очі, тимчасове або стійке зниження гостроти зору.

Ще однією проблемою є статична поза в процесі роботи. Тривале сидіння за комп'ютером завдяки напруженню м'язів спини, шиї та рук. З часом це може стати причиною захворювання опорно-рухового апарату. Для профілактики роботи робити перерви та дотримуватися принципів ергономічної організації робочого місця.

Варто також звернути увагу на дотримання правил електробезпеки під час роботи з комп'ютерною технікою. Це пов'язано з ризиком пошкодження електричним струмом у разі несправності обладнання. Важливими є також

показники мікроклімату – температура, вологість, рівень шуму. Відхилення цих параметрів може знизити працездатність працівника й негативно вплинути на його самопочуття.

Дотримання рекомендацій щодо захисту здоров’я, ергономіки та безпеки є необхідним умовою ефективної та безпечної діяльності програміста.

3.2 Вимоги з гігієни у приміщенні

3.2.1 Вимоги до робочого приміщення

Якщо відштовхуватись від додаткового вищого тексту, можна виділити такі вимоги.

По-перше, площа робочого місця на одній людині повинна становити щонайменше шість квадратних метрів. Це необхідно для комфортного розміщення обладнання та вільного пересування співробітника.

По-друге, висота стелі має бути не менше трьох цілих двох десяти метрів, що забезпечує належну циркуляцію повітря у всі часи. Також обов'язково потрібно встановити вентиляційну систему для надання свіжого повітря та видалення тепла від комп'ютерної техніки.

Дотримання цих санітарних і гігієнічних норм сприяє безпеці та комфортним умовам праці під час виконання професійних обов'язків.

3.2.2 Вимоги до рівня шуму в приміщенні

Під час роботи за комп'ютером шум не є основним шкідливим фактором, але його рівень все одно має вплив на самопочуття та працездатність користувача. До джерела шуму можна віднести системний блок вентиляторів принтерів та іншого обладнання.

Тривалий вплив підвищеного рівня шуму через це знижує концентрацію уваги та призводить до зниження продуктивності.

відповідно до санітарних норм допустимий рівень шуму на робочому місці користувача персонального комп'ютера не повинен перевищувати 50 дБ. Такий рівень забезпечений безпечним і комфортним для тривалої роботи.

Для впливу на зменшення шуму використовувати комп’ютерне обладнання

з низьким рівнем шуму розміщувати його на певному місці від робочого місця, а також використовувати у всіх матеріалах, які поглинають шум.

3.2.3 Вимоги до освітлення в приміщенні

Освітлення робочого місця дійсно впливає на стан зору та рівень комфорту під час роботи. При недостатній або підвищеній освіченості зростає ризик втоми очей, знижується працездатність і навіть може погіршитися загальне самопочуття.

Відповідно до чинних нормативів, для роботи з комп'ютером освітленість поверхні робочого столу повинна становити не менше 300 люксів. Важливо забезпечити рівномірний розподіл світла, уникати різких тіней і відблисків на екрані монітора, і це допоможе припинити завантаження на органи зору.

Рекомендують комбінувати природне та штучне освітлення. Для штучних джерел бажано використовувати люмінесцентні або світлодіодні світильники з нейтральною кольоровою температурою в межах від 4000 до 5000 Кельвінів. Робоче місце варто організувати так, щоб перші сонячні промені не вийшли на монітор це підвищений комфорт при роботі. Важливо забезпечити рівномірний розподіл світла, уникати різких тіней і відблисків на екрані монітора.

3.2.4 Вимоги з електробезпеки

Робоче місце має III клас небезпеки за ПУЕ. Живлення 230 В виконується по системі TN-S із виділеним захисним провідником PE. Основні заходи:

− Захист від ураження струмом — диференційні автомати типу A, $I\Delta n = 30$ мА, час спрацювання ≤ 40 мс.

− Вирівнювання потенціалів — всі металеві частини столів та стелажів приєднані до шини PE.

− Захист від перенапруг — комбіновані SPD T1+T2 зі струмом розряду 10 кА встановлені на вводі, локальні фільтри T3 на робочих розетках.

− Організація електромережі — групові лінії виконані кабелем NYM 3×2,5 мм² Cu у ПВХ-каналі, номінал автоматів — C-16 A. Запас по струму (коеф. завантаження 0,6) запобігає перегріву. Інструктаж з електробезпеки проводиться щороку, журнали обліку — згідно з НПАОП 40.1-1.21-98. Засоби захисту — діелектричні

килимки біля електрощитів, ізолюючі рукавички та вимірювальні кліщі — перевіряються повіркою кожні шість місяців. Робоче місце належить до класу I за ПУЕ, електроприймачі — категорії III надійності, мережа 220 В — система TN-S із PE-провідником 2,5 мм² Cu, щит оснащено Δ-автоматами 30 мА для захисту від витоків; пожежну безпеку забезпечено автоматами C-16 A та термоіндикаторами, що зменшують ризик замикань і перегріву. ДСТУ EN ISO 9241-5 вимагає відстань «око-екран» 500–700 мм, кут погляду 15–20°, висоту столу 720–750 мм; крісла мають підтримку попереку, регулювання сидіння 420–530 мм і 3D-підлокітники, а 8-годинні зміни організовано за циклом 50+10 хв (НПАОП 0.00-1.28-10). Кондиціонери щомісяця чистять фільтри G-3 і дезінфікують 3%-м розчином перекису водню, тримаючи концентрацію мікроорганізмів

3.2.5 Вимоги до мікроклімату

Робота програміста відноситься до категорії 1а згідно ДСанПіН, що означає мінімальні фізичні навантаження. Оптимальна температура повітря взимку становить 22–24 градуси, у теплий період – 23–25 градусів. Відносна вологість повітря повинна бути у межах 40–60 відсотків, а швидкість руху повітря не перевищувати 0,1 метра за секунду.

Підвищення температури повітря понад 28 градусів призводить до зниження швидкості психічних реакцій на 9–12 відсотків. Якщо вологість падає нижче 30 відсотків, це підвищує ризик розвитку офтальмологічних захворювань, викликає сухість очей і зменшує працездатність.

Система вентиляції має забезпечувати повітрообмін не менше 60 кубічних метрів на годину на одне робоче місце. Це дозволяє підтримувати якість повітря на безпечному рівні. У міжсезоння доцільно застосовувати локальні зволожувачі повітря з фільтрами-адсорберами для регуляції вологості. Вони сприяють покращенню мікроклімату у приміщенні, збереженню здоров’я працівників та підвищенню ефективності праці.

Система опалення вентиляції та кондиціонування включає приточно-витяжну установку продуктивністю 180 кубічних метрів на годину з рекуператором і касетними фільтрами класу G4 та F7.

3.3 Пожежна безпека

Організація робочого місця забезпечення суворе дотримання правил пожежної безпеки. Крім того, якщо основна робота встановлюється за комп'ютером, використання електрообладнання завжди супроводжує ризик виникнення зараження, особливо у випадках несправностей або неправильного використання.

Основні вимоги пожежної безпеки включають наступне.

Первинні засоби пожежогасіння повинні знаходитися у легкодоступних місцях. Найбільш доцільно обмін вуглекислотними або порошковими вогнегасниками, залишаються ефективними для гасіння електроустановок, що знаходяться під напругою. Необхідно встановити пожежну сигналізацію та чітко вказати маршрути евакуації. Персонал повинен бути ознайомлений з планом евакуації та порядком дій у надзвичайних ситуаціях. Використання легкозаймистих матеріалів поблизу комп'ютерної техніки недопустиме, зокрема це застосування паперу чи горючих рідин.

Керівники повинні регулярно проводити інструктажі з пожежної безпеки та перевіряти готовність приміщення до можливої високої температури.

Дотримання цих вимог значно знижує ризики виникнення пожежі та забезпечує безпеку працівників.

Організаційні заходи та документація у сфері пожежної безпеки передбачають видачу відповідного наказу про призначення відповідальної особи, яка щорічно проводить вступний первинний та повторний інструктаж. Інструкція (ІН ПБ 2025 03) дається раз на три роки для актуалізації вимог перегляду. Журнал (ПБ 6) веде визначену особу також отримати щомісячний контроль маси вогнегасників. Договір з ДСНС щодо обслуговування пожежної сигналізації та системи протидімного захисту чинним до 2027 року.

ВИСНОВКИ

У рамках дипломного проєкту була розроблена система SmartHome Lock яка на мою думку, є сучасним та ефективним рішенням для контролю доступу. Головна ідея полягає у поєднанні біометричної аутентифікації, розпізнавання облич на базі нейромережі DeepFace з високим рівнем кібербезпеки. Програмна модель реалізує три ключові функції: точне розпізнавання облич, резервний механізм доступу через захищені пін-коди та автоматизоване керування станом замка, включаючи автозакриття.

Головна перевага розробленої системи — універсальність та доступність. На відміну від дорогих комерційних рішень, моя розробка не потребує спеціального обладнання: усе працює на звичайних камерах або фотографіях , а керування здійснюється через зручний Telegram-бот. Інтеграція в існуючі системи "розумного дому" також не викликає складнощів.

Технологічні особливості. Простота використання користувачу достатньо зробити селфі або ввести пін-код. Усі налаштування виконуються через інтуїтивне меню. Надійність захисту базується на шифруванні критичних даних за допомогою AES-256 та захисті від brute-force атак і детальному логування всіх подій. Адаптивність надає можливість розширення функціоналу (наприклад, додавання нових способів аутентифікації), підтримка різних апаратних платформ (від Raspberry Pi до промислових контролерів).

У порівнянні з традиційними системами на основі паролів або карток доступу, запропоноване рішення знижує ризик втрати або крадіжки носія, зменшує експлуатаційні витрати та підвищує зручність для користувача.

Щодо застосування для домашнього користування це може бути захист житлових приміщень та інтеграція з іншими пристроями Smart Home. У комерційному сегменті — контроль доступу до офісних приміщень, керування відвідувачами у готелях чи коворкінгах.

Таким чином, SmartHome Lock — це оптимальне рішення для сучасних потреб безпеки, яке поєднує інноваційні технології із простотою використання. На мою думку, система може бути корисною як для приватних осіб, так і для бізне

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Конахович Г.Ф. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних: навчальна література. Центр навч. літ., 2018. 560 с.
2. Саммерфілд М. Python 3. Підручник програміста. – Харків: Фактор, 2021. – 600 с.
3. Основи охорони праці. навчально-методичний посібник для студентів вищих закладів педагогічного напрямку [Укладачі: В.І. Кошель,Г.П. Сав'юк, Б.С. Дзундза] – Івано-Франківськ: НАІР, 2020. –182 с.
4. Охорона.ком – Системи безпеки для дому та бізнесу [Електронний ресурс]. – Режим доступу: https://oxorona.com/ (дата звернення 11.04.2025)
5. Лутц М. Вивчаємо Python. Том 1. – Київ: Діалектика, 2020. – 720 с.
6. Мовчан В.І. Захист інформації в комп'ютерних системах і мережах. – Київ: Кондор, 2020. – 256 с.
7. Зіміна І.В. Основи біометричної автентифікації. – Львів: ЛНУ імені Івана Франка, 2023. – 198 с.
8. Biometric Standards – National Institute of Standards and Technology (NIST) [Електронний ресурс]. – Режим доступу: nist.gov/itl/iad/biometrics (дата звернення: 26.04.2025)
9. ISO/IEC 19794-2:2011. Information technology – Biometric data interchange formats – Part 2: Finger minutiae data [Електронний ресурс]. – Режим доступу: https://www.iso.org/standard/52926.html (дата звернення: 13.05.2025)
10. FIDO Alliance. Biometric Requirements v1.0 [Електронний ресурс]. – Режим доступу:https://fidoalliance.org/specs/biometric (дата звернення: 18.05.2025)
11. ISO/IEC 24760-1:2019. IT Security and Privacy – A framework for identity management – Part 1: Terminology and concepts [Електронний ресурс]. – Режим доступу: https://www.iso.org/standard/77582.html (дата звернення: 27.05.2025)
12. This Person Does Not Exist [Електронний ресурс]. – Режим доступу: https://thispersondoesnotexist.com/ (дата звернення: 28.05.2025).

ДОДАТОК А. Лістинг коду програмної моделі SmartHome Lock

```
import os
import logging
import shutil
import asyncio
from telegram import Update
from telegram.ext import (
    Application,
    CommandHandler,
    MessageHandler,
    CallbackContext,
    ConversationHandler,
    filters
)
from deepface import DeepFace
from config import Config
# Налаштування логування
logging.basicConfig(
    format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
    level=logging.DEBUG if Config.DEBUG_MODE else logging.INFO,
    filename=Config.LOG_FILE,
    encoding='utf-8'
)
logger = logging.getLogger(__name__)
# Стани для ConversationHandler
ADD_FACE, CONFIRM_FACE = range(2)
class FaceLockSystem:
    def __init__(self):
        self.known_faces = {}
        self.locked = True
        self.lock_timer = None
        self.load_known_faces()
    def load_known_faces(self):
        """Завантаження відомих облич з бази"""
        self.known_faces = {}
        try:
            for file in os.listdir(Config.KNOWN_FACES_DIR):
                if file.lower().endswith(('.jpg', '.jpeg', '.png')):
                    user_id = os.path.splitext(file)[0]
                    self.known_faces[user_id] =
os.path.join(Config.KNOWN_FACES_DIR, file)
            logger.info(f"Завантажено {len(self.known_faces)} облич")
        except Exception as e:
            logger.error(f"Помилка завантаження облич: {e}")
```

```
    async def verify_face(self, img_path: str) -> tuple:
        """Перевірка обличчя"""
        try:
            if not self.known_faces:
                return False, None
            for user_id, known_img in self.known_faces.items():
                result = DeepFace.verify(
                    img1_path=known_img,
                    img2_path=img_path,
                    model_name=Config.FACE_MODEL,
                    detector_backend="retinaface",
                    enforce_detection=True
                )
                if result["verified"]:
                    return True, user_id
            return False, None
        except Exception as e:
            logger.error(f"Помилка перевірки обличчя: {e}")
            return False, None
    async def add_face(self, user_id: str, img_path: str) -> bool:
        """Додавання нового обличчя"""
        try:
            filename = f"{user_id}.jpg"
            dst_path = os.path.join(Config.KNOWN_FACES_DIR, filename)
            shutil.copy2(img_path, dst_path)
            self.known_faces[user_id] = dst_path
            return True
        except Exception as e:
            logger.error(f"Помилка додавання обличчя: {e}")
            return False
    async def unlock_door(self, context: CallbackContext = None):
        """Відкриття дверей"""
        self.locked = False
        logger.info("Двері відкриті")
        if self.lock_timer:
            self.lock_timer.cancel()
 self.lock_timer = asyncio.create_task(self.auto_lock_door(context))
async def auto_lock_door(self, context: CallbackContext = None):
        """Автоматичне закриття дверей"""
        await asyncio.sleep(Config.AUTO_LOCK_DELAY)
        self.locked = True
        logger.info("Двері автоматично закриті")
        if context:
            await context.bot.send_message(
                chat_id=Config.CHAT_ID,
```

```
                text="🔒 Двері автоматично закриті через таймаут"
            )
    async def lock_door(self):
        """Примусове закриття дверей"""
        self.locked = True
        if self.lock_timer:
            self.lock_timer.cancel()
            self.lock_timer = None
        logger.info("Двері закриті вручну")
    async def get_status(self):
        """Отримання статусу системи"""
        return {
            "locked": self.locked,
            "faces_count": len(self.known_faces),
            "auto_lock_delay": Config.AUTO_LOCK_DELAY
        }
# Глобальний об'єкт системи
lock_system = FaceLockSystem()
async def start(update: Update, context: CallbackContext):
    """Обробка команди /start"""
    await update.message.reply_text(
        "🔐 SmartHome Lock System\n\n"
        "Надішліть фото для ідентифікації або використайте команди:\n"
        "/status - перевірити стан\n"
        "/help - довідка"
    )
async def help_command(update: Update, context: CallbackContext):
    """Обробка команди /help"""
    await update.message.reply_text(
        "ℹ️ Довідка:\n\n"
        "1. Надішліть фото - для ідентифікації\n"
        "2. /status - стан системи\n"
        "3. /help - ця довідка"
    )
async def status_command(update: Update, context: CallbackContext):
    """Обробка команди /status"""
    status = await lock_system.get_status()
    await update.message.reply_text(
        f"🚪 Статус: {'🔒 Закриті' if status['locked'] else '🔓 Відкриті'}\n"
        f"👥 Зареєстрованих облич: {status['faces_count']}\n"
        f"⏱️ Автозакриття: {status['auto_lock_delay']} сек"
    )
async def handle_photo(update: Update, context: CallbackContext):
    """Обробка фото для ідентифікації"""
    user = update.message.from_user
    temp_path = f"temp_{user.id}.jpg"
```

```
    try:
        photo_file = await update.message.photo[-1].get_file()
        await photo_file.download_to_drive(temp_path)
        verified, user_id = await lock_system.verify_face(temp_path)
        if verified:
            await lock_system.unlock_door(context)
            await update.message.reply_text(
                f"✅ Авторизація успішна !\nВітаємо, {user_id}!\n"
                f"Двері відкриті на {Config.AUTO_LOCK_DELAY} сек"
            )
        else:
            await update.message.reply_text("❌ Доступ заборонено! Обличчя не
розпізнано")
    except Exception as e:
        logger.error(f"Помилка обробки фото: {e}")
        await update.message.reply_text("⚠ Помилка обробки фото")
    finally:
        if os.path.exists(temp_path):
            os.remove(temp_path)
async def add_face_start(update: Update, context: CallbackContext) -> int:
    """Початок додавання обличчя"""
    if update.message.from_user.id != Config.ADMIN_USER_ID:
        await update.message.reply_text("⛔ Недостатньо прав!")
        return ConversationHandler.END
    await update.message.reply_text(
        "📸 Додавання нового обличчя\n\n"
        "Надішліть фото обличчя\n"
        "Скасування: /cancel"
    )
    return ADD_FACE
async def add_face_photo(update: Update, context: CallbackContext) -> int:
    """Обробка фото для додавання"""
    user = update.message.from_user
    temp_path = f"new_face_{user.id}.jpg"

    try:
        photo_file = await update.message.photo[-1].get_file()
        await photo_file.download_to_drive(temp_path)
        context.user_data['temp_face_path'] = temp_path

        await update.message.reply_text(
            "👤 Введіть ID для цього обличчя:\n\n"
            "Латинські літери та цифри\n"
            "Скасування: /cancel"
        )
        return CONFIRM_FACE
    except Exception as e:
```

```
        logger.error(f"Помилка додавання фото: {e}")
        await update.message.reply_text("⚠ Помилка завантаження фото")
        return ConversationHandler.END

async def add_face_confirm(update: Update, context: CallbackContext) -> int:
    """Підтвердження додавання обличчя"""
    try:
        face_id = update.message.text.strip()
        temp_path = context.user_data.get('temp_face_path')

        if not face_id.isalnum():
            await update.message.reply_text("❌ Невірний формат ID! Лише
латинські літери та цифри")
            return CONFIRM_FACE
        if await lock_system.add_face(face_id, temp_path):
            await update.message.reply_text(f"✅ Обличчя {face_id} додано!")
        else:
            await update.message.reply_text("❌ Помилка додавання")

        return ConversationHandler.END
    except Exception as e:
        logger.error(f"Помилка підтвердження: {e}")
        return ConversationHandler.END
    finally:
        if 'temp_face_path' in context.user_data:
            if os.path.exists(context.user_data['temp_face_path']):
                os.remove(context.user_data['temp_face_path'])
            del context.user_data['temp_face_path']
async def cancel(update: Update, context: CallbackContext) -> int:
    """Скасування операції"""
    await update.message.reply_text("❌ Операцію скасовано")
    return ConversationHandler.END
async def handle_client_message(update: Update, context: CallbackContext):
    """Обробка повідомлень від клієнта"""
    try:
        text = update.message.text.lower()
        if text == "status":
            status = await lock_system.get_status()
            await update.message.reply_text(

        f"status:locked:{status['locked']}:"
        f"faces:{status['faces_count']}:"
        f"delay:{status['auto_lock_delay']}"
    )
elif text.startswith("unlock:"):
    pin = text.split(":")[1]
    if pin in Config.BACKUP_ CODES:
```

ДОДАТОК Б. Слайди мультимедійної презентації

РОЗРОБКА МОДЕЛІ ЗАХИСТУ SMARTHOME ЗА ДОПОМОГОЮ БІОМЕТРИЧНОЇ АУТЕНТИФІКАЦІЇ

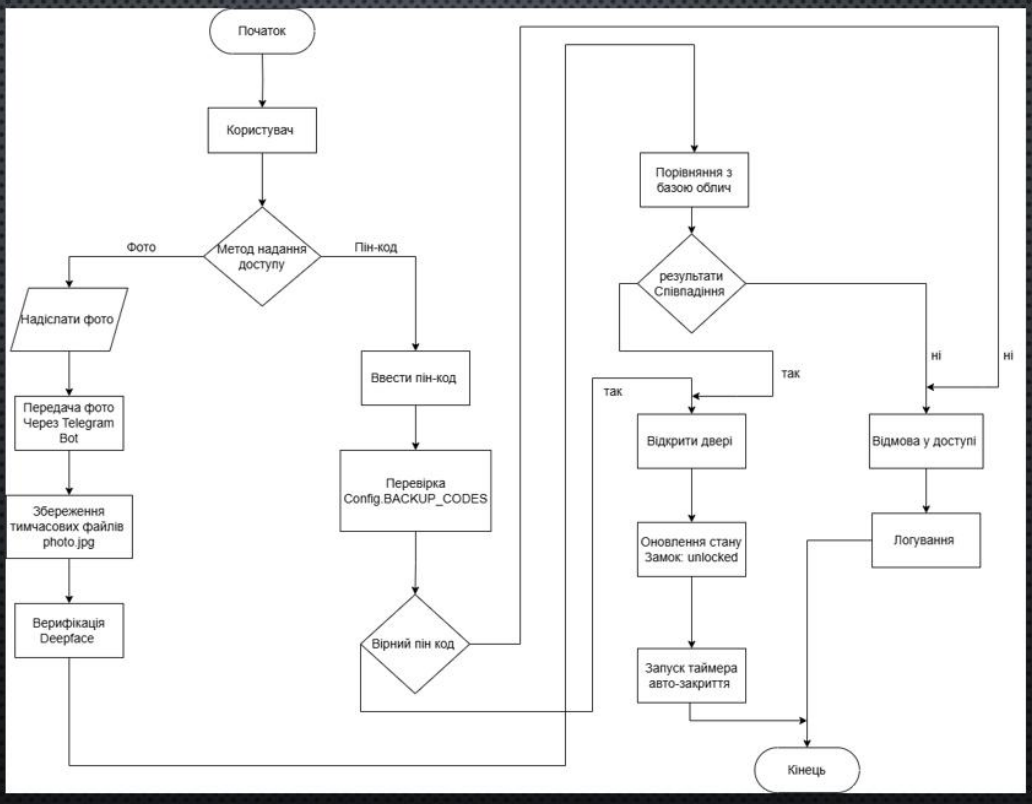
Виконав студент групи 4КБ-02

Полторакін Данїїл

ВСТУП ТА АКТУАЛЬНІСТЬ ТЕМИ

Біометрична аутентифікація

Блок-схема роботи SmartHome Lock

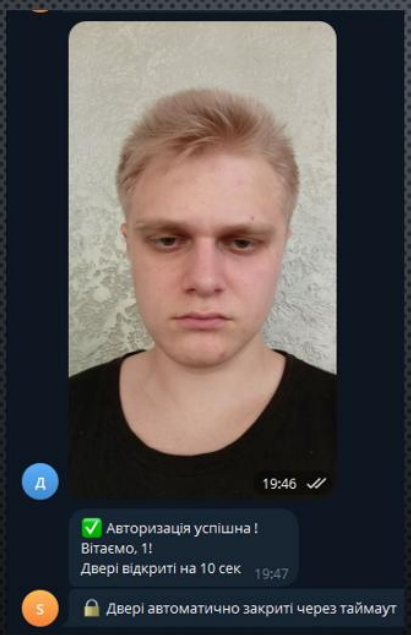


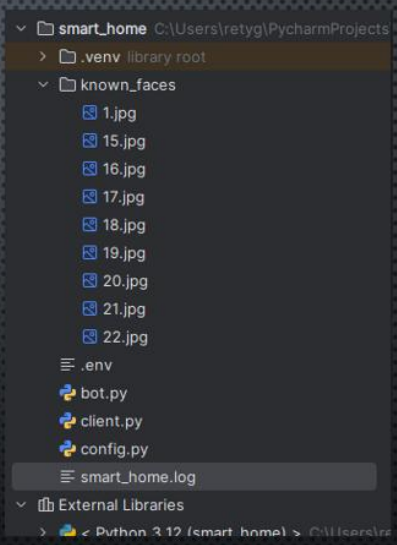
Відображення інтерфейсу SmartHome Lock в терміналі

```
================================================
        🏠 SMART HOME Lock - ГОЛОВНЕ МЕНЮ
================================================
1. Відкрити двері (пін-код)
2. Відкрити двері (авторизовані обличчя)
3. Перевірити статус системи
4. Історія команд
5. Вийти
================================================
🛠 Ваш вибір:
```

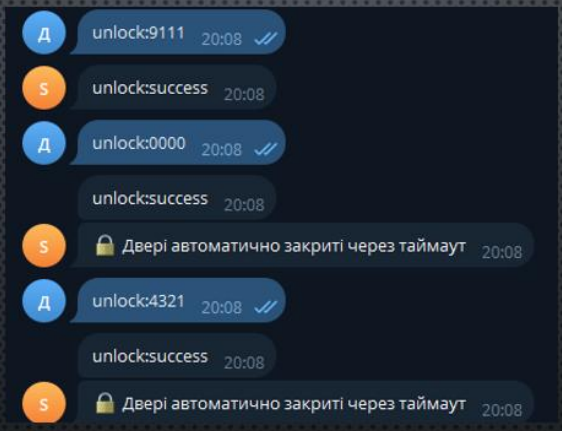
Взаємодія з Інтерфейсом Telegram Bot для додавання користувача створеного за допомогою штучного інтелекту

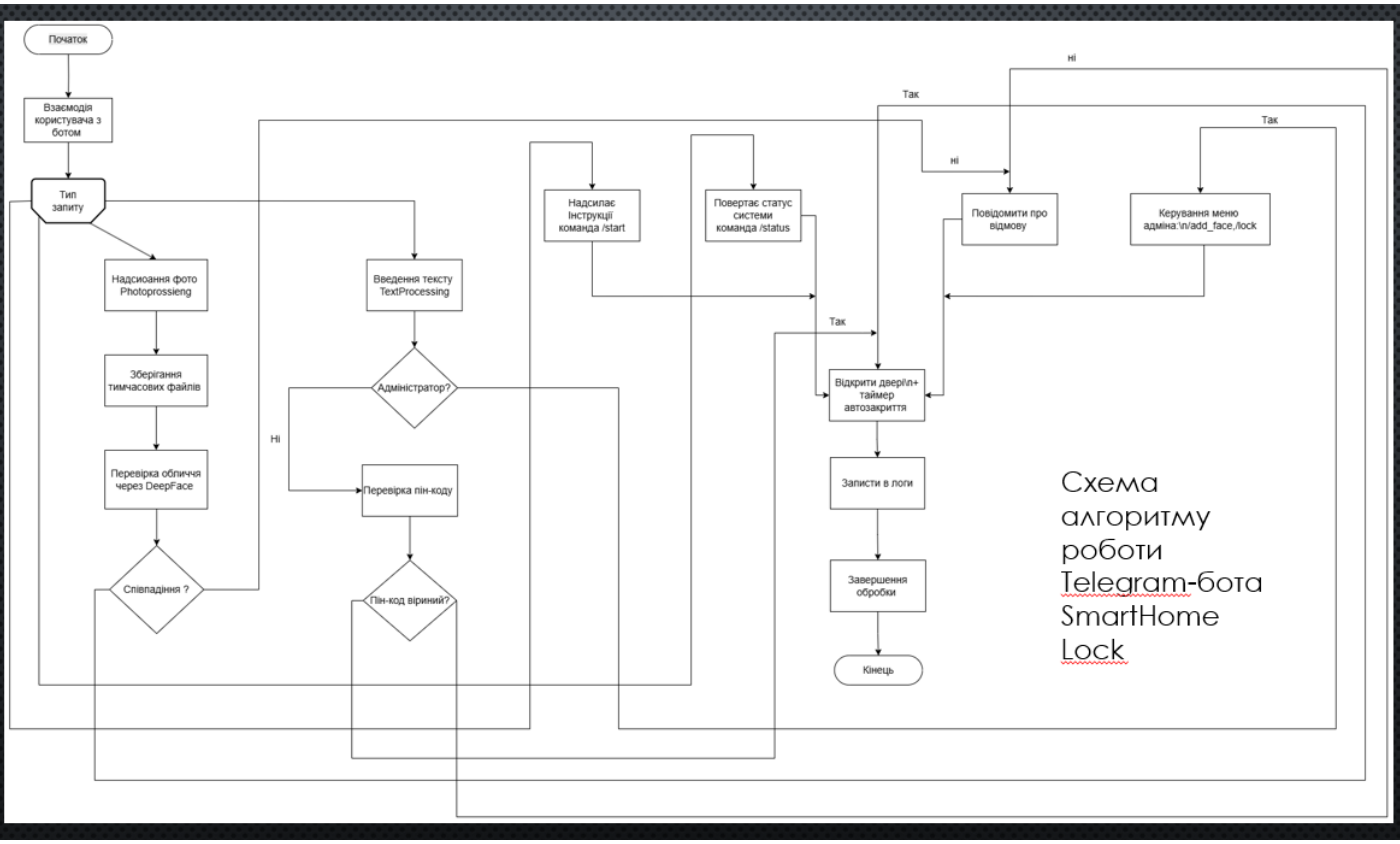
Тестування аторизації





Екстрене відкривання дверей через BACKUP_CODES





Логування подій програмної моделі

```
httpcore.http11 - DEBUG - receive_response_body.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - receive_response_body.complete
httpcore.http11 - DEBUG - response_closed.started
httpcore.http11 - DEBUG - response_closed.complete
telegram.ext.ExtBot - DEBUG - Call to Bot API endpoint `getUpdates` finished with return value `[]`
telegram.ext.ExtBot - DEBUG - No new updates found.
telegram.ext.ExtBot - DEBUG - Calling Bot API endpoint `getUpdates` with parameters `{'timeout': 10, 'offset': 557620227}`
httpcore.http11 - DEBUG - send_request_headers.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - send_request_headers.complete
httpcore.http11 - DEBUG - send_request_body.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - send_request_body.complete
httpcore.http11 - DEBUG - receive_response_headers.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - receive_response_headers.complete return_value=(b'HTTP/1.1', 200, b'OK', [(b'Server', b'nginx/1.18.0'), (b'Date', b'Tue,
httpx - INFO - HTTP Request: POST https://api.telegram.org/bot7590768452:AAHta9CKYZX8anANFQ2aZbnZ_lJJ2aDs8vY/getUpdates "HTTP/1.1 200 OK"
httpcore.http11 - DEBUG - receive_response_body.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - receive_response_body.complete
httpcore.http11 - DEBUG - response_closed.started
httpcore.http11 - DEBUG - response_closed.complete
telegram.ext.ExtBot - DEBUG - Call to Bot API endpoint `getUpdates` finished with return value `[]`
telegram.ext.ExtBot - DEBUG - No new updates found.
telegram.ext.ExtBot - DEBUG - Calling Bot API endpoint `getUpdates` with parameters `{'timeout': 10, 'offset': 557620227}`
httpcore.http11 - DEBUG - send_request_headers.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - send_request_headers.complete
httpcore.http11 - DEBUG - send_request_body.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - send_request_body.complete
httpcore.http11 - DEBUG - receive_response_headers.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - receive_response_headers.complete return_value=(b'HTTP/1.1', 200, b'OK', [(b'Server', b'nginx/1.18.0'), (b'Date', b'Tue,
httpx - INFO - HTTP Request: POST https://api.telegram.org/bot7590768452:AAHta9CKYZX8anANFQ2aZbnZ_lJJ2aDs8vY/getUpdates "HTTP/1.1 200 OK"
httpcore.http11 - DEBUG - receive_response_body.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - receive_response_body.complete
httpcore.http11 - DEBUG - response_closed.started
httpcore.http11 - DEBUG - response_closed.complete
telegram.ext.ExtBot - DEBUG - Call to Bot API endpoint `getUpdates` finished with return value `[]`
telegram.ext.ExtBot - DEBUG - No new updates found.
telegram.ext.ExtBot - DEBUG - Calling Bot API endpoint `getUpdates` with parameters `{'timeout': 10, 'offset': 557620227}`
httpcore.http11 - DEBUG - send_request_headers.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - send_request_headers.complete
httpcore.http11 - DEBUG - send_request_body.started request=<Request [b'POST']>
httpcore.http11 - DEBUG - send_request_body.complete
httpcore.http11 - DEBUG - receive_response_headers.started request=<Request [b'POST']>
```

Сценарії використання SmartHome Lock

Сценарії	Очікуваний результат	Критерій успіху
Користувач надсилає вірне фото	Двері відкриваються на 10 сек	Статус locked:True у відповіді
Невідоме обличчя	Повідомлення "Доступ заборонено"	Код відповіді 200, правильний текст
Введення резервного коду	Миттєве відкриття дверей	Лог-файл містить запис про успіх
5 невдалих спроб пін-коду	Тимчасова блокування (5 хв)	Наступні запити відхиляються
Спробу відкриття під час блокування	Повідомлення "Система тимчасово заблокована"	Відсутність запитів до DeepFace під час блокування
Адмін-команда /add_face	Додавання нового обличчя до бази	Збільшення len(lock_system.known_faces) на 1

Процес верифікації користувачів

```python
class FaceLockSystem:  1 usage
    def __init__(self):
        self.known_faces = {}
        self.locked = True
        self.lock_timer = None
        self.load_known_faces()

    def load_known_faces(self):  1 usage
        """Завантаження відомих облич з бази"""
        self.known_faces = {}
        try:
            for file in os.listdir(Config.KNOWN_FACES_DIR):
                if file.lower().endswith(('.jpg', '.jpeg', '.png')):
                    user_id = os.path.splitext(file)[0]
                    self.known_faces[user_id] = os.path.join(Config.KNOWN_FACES_DIR, file)
            logger.info(f"Завантажено {len(self.known_faces)} облич")
        except Exception as e:
            logger.error(f"Помилка завантаження облич: {e}")

    async def verify_face(self, img_path: str) -> tuple:  1 usage
        """Перевірка обличчя"""
        try:
            if not self.known_faces:
                return False, None

            for user_id, known_img in self.known_faces.items():
                result = DeepFace.verify(
                    img1_path=known_img,
                    img2_path=img_path,
                    model_name=Config.FACE_MODEL,
                    detector_backend="retinaface",
                    enforce_detection=True
                )
                if result["verified"]:
                    return True, user_id
            return False, None
        except Exception as e:
            logger.error(f"Помилка перевірки обличчя: {e}")
            return False, None
```

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти

відділення комп'ютерних систем

Полторакіна Даніїла Володимировича

(прізвище, ім`я та по батькові)

Спеціальність *123 «Комп'ютерна інженерія»*

Освітньо-професійна програма *«Безпека комп'ютерних систем і мереж»*

Керівник дипломного проекту (роботи) *Залапін Олексій Ігорович*

(прізвище, ім`я та по батькові)

Тема дипломного проекту (роботи) *Розробка моделі захисту SmartHome за допомогою біометричної аутентифікації*

Обсяг розрахунково-пояснювальної записки *72* сторінок

Обсяг графічної (презентаційної) частини *12* аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню

Представлений на рецензію дипломний проект повністю відповідає меті проектування та технічному завданню. Тематика дипломного проекту є актуальною для своєї галузі та присвячена питанням створення моделі захисту за допомогою біометричної аутентифікації.

б) характеристика виконання кожного розділу дипломного проекту (роботи)

Дипломний проект складається зі вступу, трьох розділів, висновків, переліку використаних джерел. У основному розділі розглянуті питання проблематики та розробки моделі захисту SmartHome за допомогою біометричної аутентифікації, сформовано концепцію моделі згідно до теми дипломного проекту та завданню, виконано проектування основних аспектів розроблюємого програмного забезпечення. За допомогою відповідного програмного забезпечення реалізовані всі намічені роботи з процесом розробки.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи)

Графічна частина виконана на достатньо високому рівні у вигляді презентації із використанням офісного пакету Microsoft PowerPoint. Пояснювальна записка виконана задовільно та у відповідності до норм оформлення документів із використанням офісного пакету Microsoft Word. Загальна якість виконання – задовільна, академічного плагіату у роботі не виявлено.

г) перелік позитивних якостей дипломного проекту (роботи) *Детально описано процес виконання розробки моделі системи SmartHome; Виконано проектування елементів програмного забезпечення з поясненнями на схемах та за допомогою коду; Створено архітектуру системи, розроблено модель взаємодії між клієнтом, ботом та сервером*

д) основні недоліки дипломного проекту (роботи) *Модель використовує стороннє програмне забезпечення для візуалізації результату. Замість бази даних застосовано файлову систему для зберігання зображень, що обмежує масштабованість. Резервні пін-коди не шифруються і зберігаються у відкритому вигляді.*

Оцінка розрахункової частини *Добре*

Оцінка графічної частини *Добре*

Загальна оцінка *Добре*

Прізвище, ім'я, по батькові рецензента *к.т.н. Шибаєва Наталя Олегівна*

Місце роботи і посада рецензента *Національний університет «Одеська політехніка», доцент кафедри інформаційних технологій*

Підпис:

«04» 06 2025 р.

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти

відділення комп'ютерних систем

Полторакіна Даніїла Володимировича

(прізвище, ім`я та по батькові)

Спеціальність: *123 "Комп'ютерна інженерія"*

Освітньо-професійна програма: *«Безпека комп'ютерних систем і мереж»*

Тема дипломного проекту: *Розробка моделі захисту системи SmartHome за допомогою біометричної аутентифікації*

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) *Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка містить 72 сторінки. У пояснювальній записці наведено етапи розробки програмного моделі системи SmartHome за допомогою біометричної аутентифікації. Графічна частина складається з 12 слайдів мультимедійної презентації, які також містять схеми, діаграми та алгоритми, що передбачені технічним завданням. Якість виконання пояснювальної записки та графічної частини добра, розробку виконано в повному обсязі.*

б) самостійність роботи над проектом: *Протягом всього строку дипломного проектування та переддипломної практики здобувач освіти Полторакін Д.В. поступово та послідовно виконував всі етапи розробки. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника, але з запізненням згідно графіку.*

в) теоретична підготовка випускника (випускниці): *Здобувач освіти Полторакін Д.В. під час роботи над дипломним проектом вивчив достатню кількість літературних джерел та матеріалів за даною тематикою.*

Вважаю, що теоретична підготовка дипломника задовільна і він готовий до захисту дипломного проекту

г) вміння розв'язувати виробничі та конструкторські питання *Під час дипломного проектування здобувач освіти Полторакін Д.В. мав змогу самостійно приймати окремі рішення з реалізації програмної частини для біометричної аутентифікації та показав вміння організовано працювати над поставленим завданням, розробляти структурні схеми та програмні зв'язки із застосуванням сучасних комп'ютерних програмних засобів, таких як Python та Visul Studio code, а також готувати презентаційні та звітні матеріали.*

Оцінка розрахункової частини *Задовільно*

Оцінка графічної частини *Добре*

Загальна оцінка *Добре*

Прізвище, ім'я, по батькові керівника дипломного проекту *Залапін Олексій Ігорович*

Місце роботи і посада керівника дипломного проекту *ВСП "Одеський технічний фаховий коледж ОНТУ", викладач спецдисциплін комісії комп'ютерних технологій та програмної інженерії.*

Підпис _____

«16» 06 2025 р.

ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»

Ми, що нижче підписалися,

Полторакін Даніїл Володимирович
здобувач освіти гр. 4КБ-02, та

Залапін Олексій Ігорович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка моделі захисту SmartHome за допомогою біометричної аутентифікації» (автор роботи – Полторакін Д.В., керівник роботи – Залапін О.І.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець _____ / Полторакін Д.В. /

Керівник _____ / Залапін О.І. /

«16» червня 2025 р.

ДОВІДКА

циклової комісії КТ та ПІ
про допуск до захисту дипломного проєкту
здобувача (здобувачки) освіти IV курсу
відділення комп'ютерних систем групи 4КБ-02

Полтаракіна Даніїла Володимировича

на тему *Розробка моделі захисту системи SmartHome за допомогою біометричної аутентифікації*

Висновок відповідальної особи за проведення нормоконтролю:
пояснювальна записка до дипломного проєкту виконана з некритичними порушеннями ДСТУ та оформлена відповідно до вимог Положення про дипломне проєктування

(підпис) *26.06.2025* (дата) *Петрашова В.І.* (П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного плагіату *згідно звіту про перевірку від 25.06.2025 р. значення коефіцієнту подібності в роботі становить 10,67%, коефіцієнт цитування – 2,62%.*

(підпис) *26.06.2025* (дата) *Краснокутська К.Г.* (П.І.Б.)

Попередня експертиза (малий захист) дипломного проєкту

здобувача (здобувачки) освіти *Полтаракіна Д.В.* (П.І.Б.)

проведена «*26*» *червня* 20*25* р.

Висновки *Пояснювальна записка до дипломного проєкту виконана у повному обсязі. Випускна кваліфікаційна робота (дипломний проєкт) відповідає вимогам Положення про дипломне проєктування та рекомендована до захисту.*

Голова ЦК КТ та ПІ (підпис) *Кривченко Ю.В.* (П.І.Б.)

StrikePlagiarism.com

Дата звіту 6/25/2025
Дата редагування 6/25/2025

Документ прийнятий

Звіт подібності

метадані

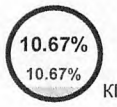
Назва організації
Odesa Technical Professional College of Odesa National University of Technology
Заголовок
Розробка моделі захисту системи SmartHome за допомогою біометричної аутентифікації

Автор
Полторакін Даніїл Володимирович
Науковий керівник / Експерт
Залапін Олексій Ігорович

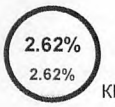
підрозділ
Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1



КЦ

25	12105	99079
Довжина фрази для коефіцієнта подібності 2	Кількість слів	Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв	ß	26
Інтервали	A→	0
Мікропробіли	·	16
Білі знаки	ß	0
Парафрази (SmartMarks)	a	75

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз — Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	78 0.64 %
2	Інтерфейс взаємодії із системою розпізнавання обличч для пристроїв доступу [redacted] 3/16/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	67 0.55 %

3	Інтерфейс взаємодії із системою розпізнавання обличч для пристроїв доступу [redacted] 3/16/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	65 0.54 %
4	https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bede670e308b/download	64 0.53 %
5	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	59 0.49 %
6	Інтерфейс взаємодії із системою розпізнавання обличч для пристроїв доступу [redacted] 3/16/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	58 0.48 %
7	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	55 0.45 %
8	https://card-file.ontu.edu.ua/bitstreams/5240e379-7721-49f0-8ee8-27140b0b473a/download	36 0.30 %
9	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	36 0.30 %
10	https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f80b18efbbc8/download	35 0.29 %

з домашньої бази даних (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)

з програми обміну базами даних (1.98 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Інтерфейс взаємодії із системою розпізнавання обличч для пристроїв доступу [redacted] 3/16/2025 **National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)**	**209 (5) 1.73 %**
2	Кваліфікаційна робота Чабан [redacted] 5/15/2025 Lviv University of Trade and Economics (Lviv University of Trade and Economics)	21 (3) 0.17 %
3	2022_62260000_Malanych_Mariana_Ihorivna_98258 [redacted] 10/26/2024 National University "Lviv Politechnika" (National University Lviv Politechnika)	10 (1) 0.08 %

з Інтернету (8.69 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	**https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content**	**371 (15) 3.06 %**
2	**https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bede670e308b/download**	**94 (2) 0.78 %**
3	**https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download**	**90 (6) 0.74 %**
4	https://benpaodewoniu.github.io/2024/12/16/telegram2/	86 (12) 0.71 %
5	**https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download**	**52 (3) 0.43 %**
6	**https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download**	**45 (3) 0.37 %**
7	https://card-file.ontu.edu.ua/server/api/core/bitstreams/c63b91ba-d04f-4715-890d-b16277695c7e/content	40 (2) 0.33 %
8	**https://card-file.ontu.edu.ua/bitstreams/5240e379-7721-49f0-8ee8-27140b0b473a/download**	**36 (1) 0.30 %**

9	https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f80b18efbbc8/download	**35 (1) 0.29 %**
10	https://card-file.ontu.edu.ua/bitstreams/aed610a6-43ef-47e0-9066-e85c89456f3e/download	**30 (1) 0.25 %**
11	https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download	26 (2) 0.21 %
12	https://ndth.org/2024/07/23/create-telegram-bot-chatgpt/	26 (2) 0.21 %
13	https://cyberleninka.ru/article/n/metod-uskorennoy-identifikatsii-otpechatkov-paltsev	24 (2) 0.20 %
14	https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download	20 (1) 0.17 %
15	https://elartu.tntu.edu.ua/bitstream/lib/41583/1/KRB_Blavitskyi_A._2023.pdf	19 (1) 0.16 %
16	https://pastebin.com/GFuFC5R4	14 (2) 0.12 %
17	https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16a8-4070-bead-5562769b7c71/download	10 (2) 0.08 %
18	https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download	9 (1) 0.07 %
19	https://card-file.ontu.edu.ua/bitstreams/21173711-5b67-4b87-b17f-6302c25e7a31/download	9 (1) 0.07 %
20	https://card-file.ontu.edu.ua/bitstreams/b1c4b329-c3e8-4b5a-a1fc-ae232ec677bd/download	9 (1) 0.07 %
21	https://card-file.ontu.edu.ua/bitstreams/e4afae26-0a7e-4a4d-afc2-94341838de2a/download	7 (1) 0.06 %

Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»
Освітньо-професійна програма: «Безпека комп'ютерних систем і мереж»
Група: 4КБ- 02

Дипломний проект здобувача освіти денної форми навчання
КБ.02.15.000. ДП

ПОЛТОРАКІНА
ДАНІЇЛА ВОЛОДИМИРОВИЧА м. Одеса
2025 р. МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»
Освітньо-професійна програма: «Безпека комп'ютерних систем і мереж»
Група: 4 КБ-02

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломного проекту на тему:
_____
_____
Проектний матеріал складається з пояснювальної записки на _____ сторінках та графічного (презентаційного) матеріалу на _____ аркушах (слайдах)

Дипломник _____ (Полторакін Д.В.)
Керівник _____ (Залапін О.І.)

Консультанти:
з економічного розділу_____ (Канський М.Ю.)
з розділу охорони праці та техніки безпеки_____ (Чорновол Н.І.) з нормоконтролю
_____(. Петрашова В.І.) старший консультант_____(Кривченко Ю. В.) До захисту допущений Голова циклової комісії _____ (Кривченко Ю.В.)
Завідувач відділення _____ (Краснокутьса К.Г.)