Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Школьняка Артема Руслановича

на здобуття ступеня вищої освіти Бакалавра

Система ідентифікації персоналу за фотографією обличчя
на основі сервісу Nyckel

Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітня програма Кібербезпека

Шифр КРБКБ.220255.22.02.36 ПЗ

Виконав студент 4 курсу група КБ-22-2 Артем ШКОЛЬНЯК
Керівник канд. техн. наук, доцент Володимир ПЕТРУШАК
Нормоконтролер д-р філософії Наталія ПЕТЛЯК

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

8 06 2026 р.

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки
Юрій КЛЬОЦ
20 січня 2026 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Школьняку Артему Руслановичу

1 Тема роботи Система ідентифікації персоналу за фотографією обличчя на основі сервісу Nyckel

Керівник роботи

Затверджено наказом ректора університету від 20 січня 2026 № 7

2 Строк подання студентом кваліфікаційної роботи на кафедру

3 Вихідні дані до роботи програмний додаток для ідентифікації персоналу за використанням фотографії обличчя, розробка фізичної складової для роботи системи, рекомендації щодо встановлення та налаштування системи

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз та опис існуючої проблеми. Загрози при використанні різних систем ідентифікації. Опис біометричних методів за принципом роботи. Вибір системи класифікації. Налаштування системи ідентифікації. Проектування фізичного модулю системи. Розробка програмного додатку. Рекомендації щодо впровадження системи

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
Електронна структурна схема, Алгоритм роботи системи ідентифікації, Схема підключення фізичного модулю

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7 Дата видачі завдання 20 січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Січень-Лютий	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Лютий	
Постановка задачі	Березень	
Визначення загальних принципів рішення задачі	Березень	
Деталізація принципів рішення задачі	Квітень	
Розробка проєктних рішень	Квітень	
Апробація проєктних рішень	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Червень	
Захист КР	Червень	

Студент _____ Артем ШКОЛЬНЯК

Керівник кваліфікаційної роботи _____ Володимир ПЕТРУШАК

АНОТАЦІЯ

Тема кваліфікаційної роботи: Система ідентифікації персоналу за фотографією обличчя на основі сервісу Nyckel

Автор роботи: Школьняк Артем Русланович.

Керівник роботи: Петрушак Володимир Степанович.

Пояснювальна записка: 65с., 5 додатків, 23 рисунок, 40 джерел.

Графічна частина: 3 плакати.

Ключові слова: СИСТЕМА ІДЕНТИФІКАЦІЇ ПЕРСОНАЛУ, РОЗМЕЖУВАННЯ ДОСТУПУ, БІОМЕТРИЧНА ІДЕНТИФІКАЦІЯ.

Кваліфікаційну роботу бакалавріату з спеціальності “Кібербезпека” присвячено розробці системи ідентифікації персоналу за використанням фотографії обличчя на базі сервісу Nyckel. В рамках виконання роботи проведено дослідження сучасних рішень у сфері систем ідентифікації персоналу та контролювання доступу, зокрема описано різні методи, їх переваги та недоліки. Опираючись на описані методи було обрано сервіс для роботи та розроблено фізичну складову системи ідентифікації з використанням ультразвукового датчику наближення та сканера радіочастотних міток. Результатом роботи є розроблена система, яка довела свою працеспроможність в реальних умовах та забезпечує відповідний рівень точності в поєднанні з захистом та надійністю.

25.05.2026

ABSTRACT

Theme of the qualification work: Personnel identification system using a facial photo based on the Nyckel service

Author of the work: Shkolnyak Artem Ruslanovich.

Supervisor: Petrushak Volodymyr Stepanovych.

Explanatory note: 65p., 5 appendices, 23 figures, 40 sources.

Graphic part: 3 posters.

Keywords: PERSONNEL IDENTIFICATION SYSTEM, ACCESS DELIMITATION, BIOMETRIC IDENTIFICATION.

The qualification work of the bachelor's degree in the specialty "Cybersecurity" is devoted to the development of a personnel identification system using a facial photo based on the Nyckel service. As part of the work, a study of modern solutions in the field of personnel identification systems and access control was conducted, in particular, various methods, their advantages and disadvantages are described. Based on the described methods, a service was selected for work and the physical component of the identification system was developed using an ultrasonic proximity sensor and a radio frequency tag scanner. The result of the work is a developed system that has proven its operability in real conditions and provides an appropriate level of accuracy combined with protection and reliability.

25.05.2026

ЗМІСТ

ВСТУП

Створення системи контролювання та управління доступом є необхідним фактором для забезпечення безпечного функціонування та розвитку багатьох компаній, особливо які займаються, або планують, інформаційною діяльністю різних галузей – від створення та підтримки сайтів до проектування та розробки програмного забезпечення різних видів використання: антивіруси, програми керування базами даних тощо. Оскільки робота компанії в напрямку, який пов’язаний, з інформаційною діяльністю досить конкурентний (хто перший запропонує інноваційне рішення старих проблем той і займає лідируючі позиції на ринку програмного забезпечення та його підтримки тощо), тому створення безпечного контуру доступу та його контроль є необхідним рішенням.

Основною метою написання даної роботи є розробка, проектування та створення системи розпізнавання обличь за використанням фотографії за допомогою сервісу Nyckel. В наш час, коли забезпечення захисту доступу та цілісності інформації є важливим фактором для забезпечення безперебійного функціонування великої кількості підприємств та установ, проектування системи розпізнавання обличь є доволі важливим фактором в забезпеченні безпеки та створення додаткової системи захисту від несанкціонованого проникнення в офіс компанії.

Використання цієї системи безпеки надає змогу створення більш безпечного, як для матеріальної складової так і інтелектуальної власності компанії, доступу, оскільки ця система може фіксувати час проходу до певних приміщень з використанням створеної системи доступу. Це наддасть змогу відслідковувати дії співробітників на протязі дня та допоможе в створенні більш точних та детальних робочих звітів та обліку годин праці: який співробітник прийшов в певний час та пішов, скільки годин в день/тиждень він був на роботі, скільки співробітників було одночасно на робочому місці та скільки фактично працювали з них з урахуванням виконаної роботи. Такі допоміжні можливості системи обліку наддадуть змогу більш детально відслідкувати робочі години та

більш точно розслідувати потенційно можливі випадки та інциденти, пов’язані з витоком даних, інтелектуальною ,або матеріальною, крадіжкою.

Окрім описаного вище, в використанні системи розпізнавання персоналу є також більш зручна функція для ефективного управління персоналом. Крім доступу до офісу, або його певних сегментів, через використання система ідентифікації може надати можливість відслідковувати місце перебування персоналу відносно робочого місця – чи знаходиться певний співробітник на своєму робочому місці чи ні та чим він фактично зайнятий на протязі робочого дня.

Опираючись на вище зазначений функціонал та можливості, які надає використання системи ідентифікації, можна затвердити, що написання дипломної роботи на обрану тему є важливим та дозволить розширити функціонал вже існуючих систем ідентифікації персоналу та надає змогу підтримувати в межах робочого колективу певні зобов’язання. Оскільки в системі є змога логування дій, це дозволить відстежувати початок робочого дня та його час, що посилить дисципліну.

Таким чином можна підвести, що створення системи ідентифікації та її впровадження є додатком інструментом створення точного рівня ідентифікації для надання доступу до офісного приміщення та допоможе в організаційних моментах в середині підприємства.

1 ДОСЛІДЖЕННЯ ІСНУЮЧИХ СИСТЕМ КОНТРОЛЮВАННЯ ТА УПРАВЛІННЯ ДОСТУПОМ, ОПИС ПЕРЕВАГ ТА НЕДОЛІКІВ, ПОСТАНОВКА ЗАДАЧІ

1.1 Основні відомості про системи контролювання та управління доступом і галузі їх використання

Для успішного та ефективного управління керівництва підприємством або установи з найманими робітниками важливо мати методи або системи ідентифікації. При управлінні невеликою компанією, наприклад сімейний бізнес, невелика мануфактура або мала IT-компанія, не завжди встановлення покращеної системи контролювання та управління доступом [1] (СКУД) може бути доцільним. Але за умови зросту числа співробітників або цінного майна компанії – як інтелектуального так і фізичного – гостро постає питання в формуванні посиленої системи захисту власності. Раніше в цьому могли допомагати окремі замки з ключами тільки в малому та довіреному колу осіб (старший персонал, власник тощо), наймання охоронця і тому подібне. Але через стрімкий розвиток технологій, налагодження загальних протоколів та їх поширення, створення СКУД з різною степеню масштабування стає все більш і більш доступним методом захисту власності. Крім цього, сучасні системи ідентифікації надають функціонал з відстеження дій користувачів, час проходу через пропускний пункт, що потенційно дозволяє зменшити фінансові витрати на прохідних(наприклад заводу чи великого складу), оскільки пропадає необхідність фізичної присутності для проходу та запису часу певного співробітника. В наш час це вже робиться автоматично, що дозволяє уникнути певних зловживань та людського фактору.

Ринок представлених систем ідентифікації персоналу досить різноманітний та пропонує різні методи та варіанти роботи. Це дозволяє покращити автоматизацію процесів, та убезпечує захист від небажаного або не поміченого “візиту”, оскільки він може потенційно нести на собі ризики, пов’язані з випадками безпеки (матеріальна крадіжка, шпіонаж тощо).

1.2 Огляд та аналіз існуючих систем та рішень

В загальному існує два види контролерів, які використовуються при проектуванні та створені системи ідентифікації – автономні та мережеві контролери відповідно [2]. Нижче більш детально описано особливості експлуатації контролерів, в чому складаються їх переваги та недоліки.

Автономні контролери – вид контролер системи управління доступом, який поєднує в одному корпусі мікрокомп’ютер-контролер та сам сканер [2]. До переваг використання саме цього рішення є те, що це є більш зручним в інсталяції та дешевим варіантом системи [3]. Проте системи такого рівня мають суттєві недоліки, пов’язані з їх налаштуванням, експлуатацією та майбутніми можливостями щодо масштабування. Основним недоліком є те, що таки пристрої повністю автономні та не мають можливості віддаленого підключення через ПК, саме тому налаштування проходить біля сканеру – додавання нового користувача, видалення старих співробітників тощо. Також, як було зазначено вище, такі контролери автономні і не мають змоги функціонувати в парі з іншими пристроями мережі. Це є суттєвим мінусом, оскільки це додає відповідні адміністративні складності з додаванням нових співробітників чи видалення доступу для старих. Крім того, варто зазначити що складність масштабування через це зростає, оскільки це додає певне адміністративне навантаження на компанію та не додає можливості централізовано відслідковувати дисципліну в середині робочого колективу, оскільки відсутнє загальне логування проходу з прив’язкою до актуального часу. Саме тому можна підвести підсумок, що такі системи ідентифікації персоналу є доступним рішенням [3], в порівнянні з мережевими контролерами, але через певний ряд недоліків є доволі незручним в експлуатації в рамках великих підприємств та установ. Доволі часто подібні варіанти встановлюють на вхідних дверях для під’їздів, невеличких складах або магазинчиках, де масштабування не розглядається та адміністративні проблеми, пов’язані з цими сканерами, не завдають великих перепон для функціонування;

Нижче буде приведено приклад автономного контролера та описано його фунціонал та цінову групу. На рисунку 1.1 приведено приклад автономного

контролера з кодовою панеллю та вбудованим зчитувачем радіочастотних міток:

Рисунок 1.1 – Автономний контролер

Цей сканер являє собою продукт компанії Trinix, моделі TRK-200EI. Це автономний сканер, який являє собою готове рішення для використання для налаштування та створення системи контролю та управління доступом. Він містить “на борту” сенсорну кодову панель та зчитувач радіохвильових міток з частотою в 125 кГц. Ціна такого сканеру на даний момент становить близько 10 доларів США, або 415 гривень.

Сканер, який наведено в прикладі, має власну пам'ять вмістом до 2000 користувачів, підтримує ідентифікацію за трьома методами: використання спеціалізованої мітки, картки + PIN-коду та використання мітки або PIN-коду. Корпус контролера виконаний з пластику, а робоча напруга знаходиться в діапазоні 12-24 Вольти постійного струму [4].

Сканер такого типу, як і було зазначено вище, є гарним варіантом для невеликих приміщень з одним входом, наприклад – невеликий склад продовольчих товарів. Через свої розміри (114x75x16 мм) та описані вище

характеристики це є ідеальним варіантом для використання за умови, коли встановлення більш розвинутої системи не є доцільним.

Мережеві контролери в свою чергу є більш розвинутими рішенням, в порівнянні з автономними контролерами [2]. Це окремий прилад, який займається зчитуванням даних про вхід персоналу через використання спеціальних сканерів. Він представляє з себе окремий невеликий блок керування з обмеженою кількістю двосторонніх точок входу/виходу з використанням системи проходу зчитувач/зчитувач (1, 2, 4) [5]. Підключення та налаштування цього пристрою відбувається за допомогою використання комп'ютера з відповідним доступом та встановленим програмним забезпеченням для цього. Крім цього, деякі мережеві контролери підтримують біометричні сканера, що поширює варіативність для використання сканерів різних типів ідентифікації. Варто зазначити, що контролер має більший функціонал та дозволяє проводити більш гнучкі налаштування системи [2], наприклад – встановлення робочих годин (щоб після певного часу фізичний доступ до приміщень був обмежений або відсутній), налаштування вихідних в компанії, розмежування доступу через різні рівні прав у співробітників. Для прикладу, це дозволить обмежити доступ співробітників до складу чи архіву без використання сторонніх систем. Через певну кількість точок входу/виходу, можливе використання одного контролеру для компанії (основний вхід та пожежний, наприклад), що зменшує потенційно адміністративне навантаження в процесі налаштування та адміністрування правил входу/виходу через можливість віддаленого налаштування та програмування ідентифікаторів (карточки, брелки, сканування відбитків пальця чи ідентифікації через обличчя тощо) [2]. Проте у такого типу контролеру є і певні недоліки, а саме основний це ціна – в порівнянні з використанням автономних контролерів [5], що робить використання саме цих недоречним при встановленні та експлуатації в під'їздах, невеликих офісних, чи промислових, приміщень тощо.

Нижче буде більш детально розглянуто контролер такого типу, його характеристики та можливості. На рисунку 1.2 приведено малюнок описуваного контролеру:

Рисунок 1.2 – Мережевий контролер Dahua

На зображенні вище приведено мережевий контролер компанії Dahua, моделі DHI-ASC2202B-S. Він використовується, як було зазначено вище, для створення мережевої системи контролювання доступом з налаштуваннями через комп’ютер з адміністративними правами та встановленим спеціалізованим обладнанням. Це мультифункціональний контролер з проводовим з’єднанням з мережею з двома точками входу, або однією точною входу-виходу – це можуть бути 2 двері, або 1 турнікет. Він підтримує використання сканерів різних методів ідентифікації: за використанням паролю, радіохвильової мітки, відбитку пальця або множинної автентифікації за використанням комбінацій, описаних вище, методів. Такий функціонал надає змогу створення більш безпечної системи ідентифікації та розширює фунціонал використання контролерів такого типу, в порівнянні з автономними.

На даний момент цей контролер системи доступу коштує близько 125 доларів США, або 5100 гривень. Сканер має вбудовано пам'ять на 100 тисяч користувачів, 3 тисячі унікальних відбитків пальців, 100 тисяч карток доступу та 500 тисяч записів, що робить його гарним рішенням при використанні в на

прохідній заводу з великою кількістю співробітників [5]. Контролер має 2 виходи на датчики дверей, 2 замки та 2 тривоги. Окрім зазначеного вище, сканер підтримує живлення через використання блоку живлення 12 Вольт постійного струму або використання POE (Power on Ethernet) – технологію забезпечення живлення пристроїв за допомогою використання кабелю Ethernet, при тому, що дані, які передаються кабелем, не викривлюються. Це дозволяє використовувати один кабель замість монтажу декількох [6].

Контролер також обладнаний системою захисту та тривоги за умови вторгнення, примусу, вводу неправильного паролю тривоги, повторного проходу при умові використання на турнікету або реєстрації входу та виходу, що унеможливлює використання дублікату ідентифікатора тощо. На контролері можна проводити великий перелік налаштувань для контролю та управління доступом: до 128 налаштування періодів, до 128 святкових днів, чорних та білих списків та розмежування доступу за присвоєння користувачеві відповідного рівня доступу – звичайний, патруль, гість, адміністратор тощо.

Використання мережевого контролеру є доцільним при використанні в великих підприємствах, таких як великі склади, заводах, так і в межах муніципальних установ – шкіл, університетів тощо. Оскільки контролер є мережевим, його розміри, майже, не мають значення, оскільки він може знаходитись далеко від замків та дверей. Але, незважаючи на всі його переваги, використання такого типу може бути не доцільним за умови великого збільшення вартості обладнання, в порівнянні з використанням автономних аналогів.

В свою чергу також існує кілька основних процесів ідентифікації персоналу не зважаючи на тип обраного контролера – автономних чи мережевих варіантів. В загальному метод ідентифікації поділяють на 2 основні категорії – ідентифікація через використання специфічного ідентифікатора (спеціальний брелок, мітка, пристрій тощо) та біометрична ідентифікація (сканування відбитку пальця, обличчя або сітківки ока) Нижче приведено більш змістовний опис особливостей, притаманних кожному з методів ідентифікації, їх особливості, плюси та мінуси:

– ідентифікація на основі використання специфічного ідентифікатора – ці

системи є доволі поширеними через ціну ідентифікаторів та сканерів. Їх, умовно, можна поділити на 2 типи – контактні та безконтактні [9]. До контактних належать такі підвиди;

– магнітні картки це метод ідентифікації оснований на використанні ідентифікації через картки з магнітною стрічкою з цифровим кодом. Проте останнім часом цей метод втрачає свою актуальність через відносну ненадійність карток (стрічка розмагнічується або пошкоджуєтеся що перетворює її в шматочок пластику) [9];

– система ідентифікатори “Touch memory” – цей ідентифікатор є більш надійним через свою простоту, надійність та дешевизну в порівнянні з магнітними картками. Ці ключі містять пасивні системи з записаним кодом ідентифікації, який під час контакту з зчитувачем передається на контролер. Такі типи ключів знайшли велику популярність в домофонах та охоронних системах в під’їздах багатоквартирних будинках [10].

До безконтактних методів ідентифікації можна віднести використання RFID-міток. Ця технологія чимось подібна до Touch memory, проте вона базується на принципі радіочастотної ідентифікації. Ідентифікація за використанням цієї технології надає змогу в проведені безконтактної ідентифікації мітки на відстані до 300м [9]. Для ідентифікації необхідно просто піднести ідентифікатор до зчитувача на певну відстань. Це відбувається через те, що зчитувач постійно генерує власне електромагнітне поле, потрапляючи в яке, картка живиться від нього та передає свій унікальний номер на сканер. Саме тому для проходження ідентифікації треба просто піднести ідентифікатор в межі дії цього поля. Такі мітки діляться на 2 категорії: низькочастотні (125 кГц) та високочастотні (13,56 МГЦ) [11]. Їх ключова різниця, в рамках огляду на ступінь захисту, полягає в тому, що низькочастотні картки піддаються легшому копіюванню, що, потенційно, може ставити під загрозу систему через її компрометацію. В свою чергу, високочастотні карти є більш захищеним рішенням, оскільки вона мають власний мікропроцесор та захищену пам’ять, що надає змогу шифрувати дані [11]. Через це їх копіювання стає більш важким процесом, але є недолік –

високочастотні картки є більш дорожчим варіантом, в порівнянні з низькочастотним аналогом. Також варто зазначити, попри всі переваги – ціну сканерів та карток, їх надійність та зручність користування – метод радіочастотної ідентифікації має певний недолік, а саме те, що ідентифікатор співробітника може втратитись або викрастись зловмисниками, саме тому цей метод є доцільним та зручним, але тільки за умови усвідомлення ризиків його використання.

Нижче приведено приклад мережевого радіочастотного сканера. Також було розглянуто його технічні характеристики, такі як дальність ідентифікації міток, розглянуто його ціну в контексті мережевих сканерів, особливості використання та переваги саме цього вибору для системи ідентифікації. На рисунку 1.3 приведено зображення зовнішнього вигляду сканера для мережевого контролера.

Рисунок 1.3 – Сканер RFID-міток

Наведений вище приклад сканеру представляє собою модель компанії Trinix TRR-1103EW. Це зовнішній сканер, обладнаний двоколірною індикацією (червоний та зелений колір), який підтримує роботу з високочастотними ідентифікаторами Em-Marine, що працюють на частоті 125 кГц. Це пристрій близької ідентифікації та працює на відстані до 5 сантиметрів, також він має низьке споживання енергії – менше 30 мА.

На момент написання, цей сканер коштував приблизно 12 доларів США, або

520 гривень [12]. Він є ідеальним варіантом, в співвідношенні ціни до функціоналу, оскільки підходить як для встановлення на зовнішній частині приміщення, так і в середині, має світлову індикацію та надає змогу в повному обсязі забезпечити відповідний рівень захисту для прибудинкової території, входу в офісне приміщення і тому подіне:

Ідентифікація через відбиток пальця – ця технологія в наш час є доволі поширеною через зниження ціни на сканери. Ідентифікація, при використанні такого методу, базується на унікальності відбитків пальців людей, оскільки відбиток є унікальною рисою кожної людини, оскільки їх важко підробити, а малюнок, майже, однаковий все життя [13]. Такі пристрої поділяються за методом сканування на чотири види: оптичні сканери (роблять фотографію пальця для порівняння), ємнісні (або CMOS) сканери (використовують конденсатори під струмом для формування зображення відбитку пальця, що підвищує їх точність), ультразвукові сканери (використовують ультразвук для проходження в епідермальні шари шкіри) та термічні сканери (базуються на різниці температур на поверхні шкіри між гребнями та долинами відбитку пальця) [13].

Такий вид сканерів так само може від’єднуватись до мережевого контролера або бути автономним. Цей спосіб ідентифікації персоналу має великі переваги, в порівнянні з використанням радіочастотної ідентифікації, оскільки підробка ідентифікатора, майже, неможлива, як і його втрата (випав в тролейбусі, розмагнітився, вкрали тощо), як і те, що відпадає необхідність в систематичній до купівлі необхідних карток та брелків (не зважаючи на їх ціну, на протязі довгого проміжку часу це може становити певні фінансові витрати). Проте такий тип сканерів досить чутливий до порізів шкіри, температури конкретного пальця. Також сканери такого типу не є гігієнічними саме через те, що палець необхідно прикладати до певної площадки сканування.

Підсумовуючи, в наш час це доволі популярна технологія через свій степінь захисту та можливість ідентифікації високого рівня, через унікальність ідентифікатора кожного суб’єкта та неможливість його крадіжки, підміни або втрати (або майже). Проте ця система є більш дорогим варіантом, ніж

радіочастотна ідентифікація, оскільки в ній задіяні більш складні технологічні рішення та виконання [14].

Наступний на огляд буде сканер відбитку пальця. Це буде мережевий сканер, для роботи якого необхідний контролер. Огляд саме мережевих сканерів дозволить дізнатись зрозуміти їх фактичні можливості та фунціонал. Приклад сканеру ідентифікації персоналу через використання відбитка пальця та опис його характеристик приведено нижче, на малюнку 1.4 продемонстровано зовнішній вид сканера.

Рисунок 1.4 – Сканер відбитку пальця Dahua

Цей сканер представляє з себе елемент системи контролю та управління доступом на базі використання мережевого контролера. Великий відсоток сканерів біометрії, а саме – ідентифікації за відбитком пальця, підтримують декілька методів ідентифікації. Представлений варіант підтримує зчитування низькочастотних та високочастотних, крім використання відбитку. Це надає змогу комбінувати методи ідентифікації між собою або створення запасних, на випадок втрати ідентифікатора. Цей сканер коштує приблизно 110 доларів США, або 4600 гривень, що є дорожчим варіантом, в порівнянні з сканерами RFID-міток. Він має дальність зчитування ідентифікаторів в 5 сантиметрів та обладнаний класом захисту IP 65, що створює можливість його встановлення як на вулиці, так і в

приміщені. Окрім вище зазначеного, сканер має звукову та світлову індикацію, сторожовий таймер (виявлення та контроль ненормального робочого стану обладнання). Він споживає вже більше, в порівнянні з минулим прикладом, тобто близько 1.7 Вт [15].

Сканера такого типу є доволі розповсюдженими через свій ступінь захисту, можливість інсталяції в різних умовах та місцях через наявність захисту від вологи та пилу та додаткового методі ідентифікації, що надає змогу не знімати рукавички в зимовий час та використовувати картку або брелок.

– Ідентифікація на базі розпізнавання обличь. Цей метод базується на використанні біометричної ідентифікації, як і розпізнавання по відбитку пальця, але він використовує спеціальні протоколи для розпізнавання обличчя. Ідентифікація відбувається за фотографією або відеокадром і порівнюються з наявною базою обличь [16]. Як і відбиток пальця, ця технологія стає більш доступною та популярною через зменшення ціни на сканери та створення більш простої системи в користуванні для впізнавання. Принцип взаємодії комп'ютерних систем з біологічними даними доволі важкий в реалізації, проте новітні технології базуються на чотирьох кроках ідентифікації: 1 – виявлення лиця для сегментації його з фону зображення, 2 – вирівнювання обличчя з урахуванням пози лиця, розміру зображення та факторів сканування (освітлення об'єкту, освітлення фону тощо), 3 – локалізація рис лиця, 4 – ідентифікація обличчя. Певний набір фізіологічних особливостей, таких як положення та пропорції очей, носа та роту точно визначаються для співставлення лиць з базою даних [11].

Методи ідентифікації діляться на кілька різновидів за методом роботи:

– традиційний метод ідентифікації базується за допомогою ідентифікацією рис обличчя, опираючись на певні орієнтири або особливості обличь. До прикладу – алгоритм аналізує відносне положення та розмір очей, носу, базових рис обличчя для співставлення з вже існуючою базою. Ця модель досить успішна, на свій час, основана на методі співвідношення шаблонів [16];

– 3-D ідентифікація це технологія розпізнавання обличь, яка базується на

використанні трьохмірних датчиків для ідентифікації особливостей поверхні обличчя. Основним плюсом цього методу є те, що його робота не залежить від рівня освітленості та використання цього методу дозволяє більш точно ідентифікувати обличчя. Також можлива ідентифікація з діапазону кута огляду до профілю. Проте цей метод ідентифікації є більш дорогим, в порівнянні з традиційним [16];

– теплові камери як метод ідентифікації базується на використанні теплових камер. Цей метод дозволяє уникнути проблеми при використанні окулярів або масок, оскільки він ідентифікує форму голови. Проте в наш час ця технологія доволі “сира”, оскільки ці камери не можуть надати надійну ідентифікацію на тепловому знімку. Але цей метод є, відносно, перспективним в використанні з іншими методами ідентифікації [16].

В системах контролювання та управління доступом використовується традиційний метод, трьохмірних. Також може бути використання їх комбінації, в більш розвинутих моделях сканерів. Метод ідентифікації через використання обличчя є доволі точним в своїй роботі та, на відміну від використання відбитку пальця, більш захищеним та гігієнічним в роботі, оскільки не відбувається прямого контакту з поверхнями зчитувача.

Сучасні сканери надають великий процент точності в роботі, в порівнянні з сканерами відбитку пальця. Так само вони мають велику стійкість до використання муляжів та підробок, оскільки використання фото для ідентифікації становить загрозу для системи безпеки. Також вони можуть мати фунціонал розпізнавання обличь в масках або окулярах, що робить коло їх використання більшим, в порівнянні з минулими версіями сканерів.

Але вони мають суттєвий недолік – високу ціну на сканер в порівнянні з радіохвильовими сканерами, або сканерами відбитку пальця. Однак слід зазначити, що будь-яка ціна може бути виправданою, опираючись на необхідний ступінь захисту та зручність використання. Такі сканери можуть стояти на вхід до режимних об’єктів, складів з цінною та чутливою інформацією тощо.

Приклад сканера з підтримкою ідентифікації через розпізнавання обличчя

приведено нижче. На зображені 1.5 приведено сам зовнішній вигляд сканеру.

Рисунок 1.5 – Сканер розпізнавання обличчя Dahua

В прикладі приведено сканер компанії Dahua, моделі DHI-ASI3223A. Цей сканер являє собою частину системи з ідентифікації персоналу за використання технологій розпізнавання обличь. Цей сканер надає можливість його використання як окремого пристрою, так і підключити до мережі контролю і управління доступом. Він містить скляний сенсорний екран та 2-мегапіксельну ширококутну подвійну камеру з можливості інфрачервоної підсвітки для розпізнавання при умові недостачі світла. Ціна терміналу такого типу становить близько 200 доларів США, або 8200 гривень.

Як і сканер відбитку пальця, він містить декілька способів ідентифікації – це використання розпізнавання обличчя та ввід паролю на екрані. Це надає змогу комбінувати методи розпізнавання. Сканер має підтримку до 2 тисяч користувачів, 50 адміністраторів та 300 тисяч записів. Пристрій має велику ступінь точності та швидкий час порівняння – близько 0.2с при ідеальних умовах.

Також, при приєднанні цього терміналу до інтернету, він надає можливість дистанційного керування, виконання відеодзвінків та надсилання сповіщень про тривогу [17].

Підсумовуючи вище сказане, використання терміналу ідентифікації є доцільним в межах приміщення та є ідеальним варіантом для використання в лікарнях, доступу до медичних складів або офісів, де питання гігієни стоїть більш гостро, в порівнянні з іншими варіантами. Такий термінал є гарний вибором як окремий пристрій, але через його високу ціну та певні особливості (він не має можливості розпізнавання людей в масках або окулярах) це робить його досить дорогим варіантом в межах системи ідентифікації для звичайних установ та підприємств без виконання суворих вимог до гігієни або відповідного рівня ідентифікації персоналу.

Але сучасні технології надають змогу зменшити ціну на побудову системи ідентифікації з використання розпізнавання персоналу через використання обличчя до прийнятної. Саме така система була створена в межах виконання дипломної роботи. Функціонал системи та її робота буде базуватись на використанні традиційного методу ідентифікації на основі побудови математичних моделей обличчя, обчислення унікальний векторів, притаманних кожному обличчю, та порівнянні їх з вже існуючими в наявній базі обличь.

Саме тому процес ідентифікації персоналу на базі використання не традиційних сканерів, а з опорою на новітні технології та використанням на базі вже існуючих пристроїв є гарним рішенням для тестів системи ідентифікації та подальшої розробки.

1.3 Постановка задачі

Для розробки системи ідентифікації персоналу на базі розпізнавання обличь було використано сервіс “Nyckel”. Він базується на традиційному методі ідентифікації, оскільки цей метод є, відносно, легким в налаштуванні та створені повноцінної системи ідентифікації персоналу за використанням фотографії обличчя. Саме він був обраний через свою точність в ідентифікації та є ідеальним

варіантом для створення відносно дешевих сканерів для власних цілей.

Цей варіант для системи розпізнавання обличь було обрано через певний ряд його переваг, а саме – ціну на використання (розповсюджується на безоплатній основні) та простоту налаштування. До прикладу – конкурент від Microsoft Azure пропонує готові рішення за використанням Face API [18], проте використання цього рішення в рамках побудови системи ідентифікація персоналу за використанням фотографії може нести певні фінансові витрати – від процесу налаштування цього методу ідентифікації до його тестів роботи та впровадження може піти багато часу та ресурсів компанії, як людських так і фінансових.

Створювана система, в рамках цієї дипломної роботи, має розпізнавати на фотографії людину та ідентифікувати, опираючись на налаштовану створену базу обличь. Крім описаного вище, система має підтримувати масштабування бази даних з обличчями, саме тому вона має розпізнавати 10 різних людей та створювати журнал логування з часом ідентифікації та ім’ям ідентифікованої особи. Система має також фізичну реалізацію через використання зовнішньої плати-мікроконтролера, яка буде визначати наявність певної особи перед зчитувачем та робити контрольний знімок для перевірки. Після проведення тестування та класифікації фотографії особи, має відбутись певна дія. Наприклад – фізично буде загоратись світлодіод для позначання результату ідентифікації або надаватись доступ до інформаційного ресурсу.

Програма для взаємодії з сервісом має мати інтерфейс для кінцевих користувачів, журнал логування буде доступний через інтерфейс за умови підтвердження повноважень рівня доступу як “адміністратор системи”, як і налаштування системи. Це наддасть змогу для зручного використання та відслідковування часу ідентифікації, без загрози зміни в журналах користувачами в цілях редагування робочих годин як своїх, так і чужих. При використанні системи такого типу в підприємствах комерційного типу це наддасть інструмент для зручного формування журналу з годинами робочого часу для формування зарплати та наддасть змогу для перевірки фактичного часу роботи відповідного користувача або робітника.

Система базується на використанні веб-камери ноутбука для можливості створення фотографії обличчя персоналу та проведення процесу ідентифікації робітників. Однак слід зазначити, що сервіс “Nyckel” надає можливість використовувати його як бібліотеку при виконанні певної програми та можливість використання різних камери для ідентифікації, які будуть під’єднані до контролеру, який являє собою комп’ютер охорони. Після процесу ідентифікації та отримання відповіді від серверу, має бути реалізована можливість для обробки та використання даних перевірки не лише для ідентифікації. В контексті використання в рамках СКУД це може бути надання доступу до певного приміщення після ідентифікації, згідно з встановленими правилами розмежування доступу персоналу, за умови наявності обличчя серед існуючої бази та приналежності до відповідного рівня повноважень для ідентифікованого співробітника.

2 СТВОРЕННЯ І НАВЧАННЯ МОДЕЛІ СЕРВІСУ NYCKEL ДЛЯ КЛАСИФІКАЦІЇ ОБ'ЄКТІВ

2.1 Основні відомості про існуючу модель ідентифікації

Створення системи ідентифікації персоналу в рамках написання дипломної роботи буде базуватись на використанні можливостей сервісу Nyckel. Цей сервіс пропонує доступ до власної моделі нейронної мережа. Це стартап, який з’явився в 2021 році та за власники якого мають великий досвід в програмуванні та сфері проектування та реалізації нейронних мереж та працювали до цього в таких світових компаніях, як Oracle та Microsoft [19].

Основний продукт, який представляє сервіс, є модель розпізнавання та класифікація об’єктів за умови проведення невеликого навчання. Принцип роботи сервісу оснований на використанні автоматичного машинного навчання (Machine learning) та розпізнавання об’єктів. Машинне навчання це метод навчання сервісу штучного інтелекту, який оснований на алгоритмі “вивчення” готових шаблонів даних для створення бази та подальшої класифікації та ідентифікації нової інформації. Такий метод навчання моделі дозволяє формувати власні шаблони роботи без чіткого пропису інструкцій, що дозволяє програмувати більш чіткі та великі моделі з різними областями використання без необхідності програмувати детальні патерни та взаємозв’язки. Метод машинного навчання в даний час є домінуючим в області штучного інтелекту, оскільки надає гарний результат при роботі з новою інформацією [20]. Принцип навчання є доволі затратним та важким, оскільки нейромережі необхідно проаналізувати великий обсяг інформації та зробити певні заключення про інформацію або об’єкт, який вона аналізує.

Сервіс Nyckel представляє можливість полегшеного створення та налаштування готових моделей нейромережі з різними можливостями – від створення логотипів до аналізу резюме нових співробітників на вакансію. Але є і можливість власної моделі з своїми налаштуваннями та можливостями, такими як розпізнавання тексту, предмету на фото і тому подібне [21].

Опираючись на вище описане, можна зробити заключення що машинне

навчання програми це доволі важкий процес, але сервіс надає змогу для полегшення навчання моделі. Для цього необхідно загрузити певну кількість штучних даних для навчання первинної моделі обробки та ідентифікації об’єктів – тексту на фото, класифікація через використання фото, пошук певної інформації в тексті тощо. Також у сервісу є можливість використання API для полегшення використання та інтеграції сервісу в різні області користування – від використання для полегшення робочих задач, так і автоматизоване використання для ідентифікації персоналу через фотографію. Це дозволяє використовувати сервіс як надбудову для вже існуючих систем без використання, або створення, складних методів обходу технічних проблем. Платформа також забезпечує безпеку інформації та даних користувачів, що робить її гарним вибором при використанні та інтеграції в існуючі системи ідентифікації компаній та підприємств, або проектування та впровадження нових систем ідентифікації. Сервіс робить все для зручного використання та приємного досвіду користування з боку клієнтів, саме тому при потребі в масштабуванні моделі клієнта це не є важкою проблемою через можливість використання хмарного середовища, що знижує поріг входу для використання та полегшує процес масштабування та роботи з процесами та надає змогу сфокусувати уваги на розробці моделі без створення проблем [22].

В порівнянні з конкурентами, Nyckel проводить автоматичний аналіз декількох базових моделей штучного інтелекту для пошуку оптимального варіанту моделі для виконання поставленої задачі. Це надає змогу для більш швидкого вибору системи для навчання, оскільки сервіс виконує це самостійно та робить його використання більш доступним для широкого кола користувачів, які не володіють спеціалізованими навичками чи відповідною освітою. Почати використовувати сервіс можна як за допомогою веб-інтерфейсу так і за використанням програмного коду. Через автоматичний вибір основи для побудови програми, точність результату класифікації зростає, в порівнянні з представленими конкурентами на ринку [23].

Ключова перевага використання моделей, по типу Nyckel полягає в

швидкості їх налаштування, в порівнянні з використанням звичайних моделей автоматичного машинного навчання. Сервіс використовує вже навчені базові моделі, час на навчання яких займав великий проміжок часу, бо програма має передивитись декілька мільйонів фотографій або текстів. Коли користувач сервісу створює власну модель, Nyckel має знайти лише певні ознаки для використання в проведені кінцевих налаштувань невеликого шару штучного інтелекту для використання в певній моделі, створеній користувач. Саме через використання принципу до налаштування системи, а не повного її пере налаштування “з нуля”, використання саме цього сервісу є доцільним та універсальним під різноманітні задачі. Так само такий підхід дозволяє використовувати менший об’єм вхідних даних для налаштування (від 5 до 10 прикладів в порівнянні з налаштуванням “з нуля”) та пришвидшити процес інтеграції системи ідентифікації, що додає великий плюс до використання саме цього сервісу [24].

2.2 Процес створення та налаштування моделі класифікації

Для створення системи класифікації на базі використання сервісу Nyckel треба перейти на сайт сервісу та створити власну власний проект. Як було описано вище – можна використовувати як вже готові рішення так і створити власне рішення, яке буде оптимальний варіантом для побудови системи.

Після процесу реєстрації (або входу в систему), надається можливість для створення власної моделі системи на базі сервісу. В процесі налаштування та тренування системи не має виникати складності, але, якщо вони є, на сайті Nyckel є підтримка та блог, де описано процес створення та навчання системи розпізнавання згідно параметрів користувача.

Нижче буде детально описано процес створення та адаптації моделі класифікації для використання під зазначені вище цілі, а саме – класифікації об’єктів. На рисунку 2.1 приведено інтерфейс сервісу та створення власної моделі без використання шаблонів.

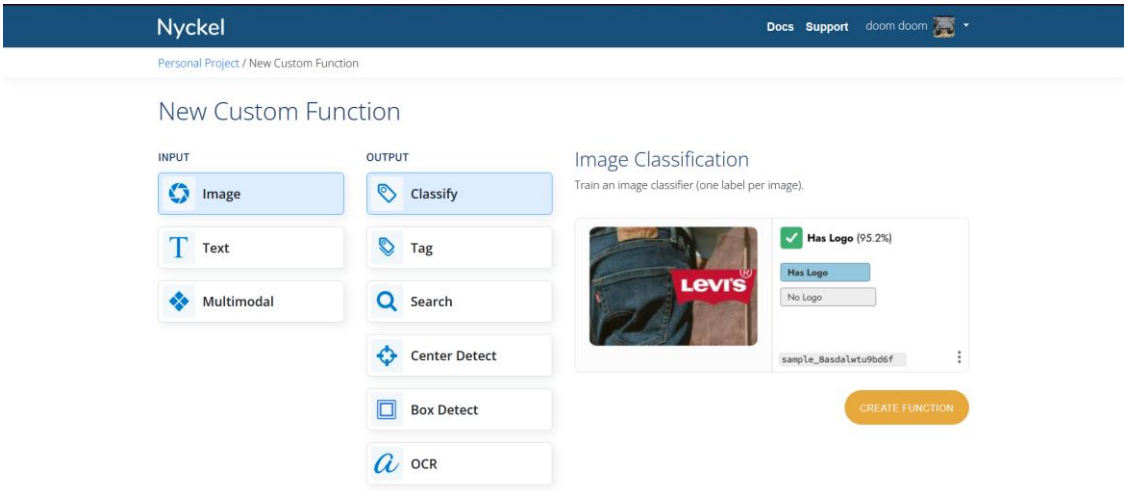


Рисунок 2.1 – Інтерфейс системи та в редакторі функцій

На рисунку приведено вище продемонстровані можливості сервісу. Він надає змогу для розпізнавання об’єктів на фото, а саме – їх класифікації (Classify) за обраними параметрами; призначення міток (Tag), наприклад: колір предмету, або який це предмет та його колір; пошуку (Search) зображень в галереї по параметрам; пошук об’єктів на фото (Center Detect) та відображення їх, наприклад – пошук зарядного пристрою телефону на фото кімнати; пошук предметів, їх розмірів та центрів (Box Detect); розпізнавання тексту на фото (OCR), наприклад розпізнавання погано читабельного тексту. Також система може взаємодіяти з різними типами інформації, як фото, текст або багато функціональна модель з поєднанням різних типів. В нашому випадку необхідно використовувати модель з розпізнавання фотографії. Після створення функції з обраними вихідними даними користувач може приступати до етапу навчання та налаштування сервісу.

Ідеальним варіантом для класифікації об’єктів, згідно умови, є використання методу класифікації об’єктів. Після вибору методу роботи з вхідними даними необхідно створити класи для призначення. Наприклад, співробітник Школьняк А.Р. та чужак. Створення класу “чужак” необхідно для збільшення точності роботи системи класифікації, в цей клас завантажуються кілька десятків зображень людей з інтернету, що дозволить більш точно

налаштувати систему. На рисунку 2.2, приведеному нижче, продемонстровано сторінку створення класів та вже створені класи для використання в ідентифікації. Цей процес необхідний для використання класів в подальшому навчанні системи.

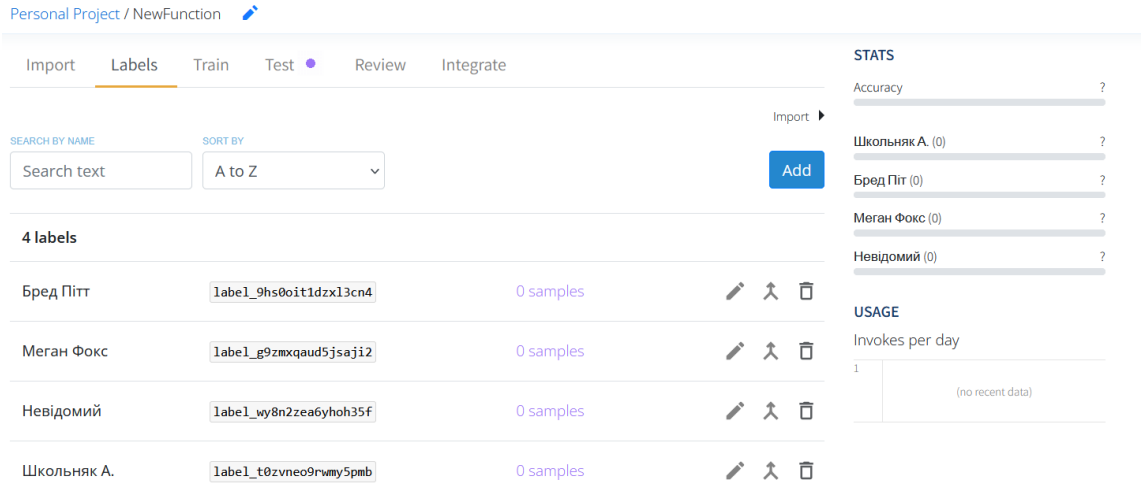


Рисунок 2.2 – Створені класи для навчання системи

Для створення цих класів треба було перейти в вкладку “Labels” та натиснути кнопку створення нового класу (синя кнопка “Add”). Далі треба дати назву для класу та, за бажанням, надати опис для класу. На рисунку, приведеному вище, для функції створено 4 класи: “Школьняк А.”, “Бред Пітт”, “Меган Фокс” та “Невідомий”. Зазначені класи будуть використовуватись в подальшому процесу навчання системи та призначенню відповідних критеріїв класифікації. Класи можна створити як на веб-сайті сервісу, так і використовувати вже готові за допомогою функції завантаження класів. Файл з готовими класами має бути в розширені “.csv”, щоб система змогла його успішно використати. Після завантаження файлу, класи з нього будуть відображені в списку. Цей надає змогу для створення резервних копії системи та швидкого розгортання її в разі зміни пристрою використання або втрати його в разі техногенних випадків, таких як апаратний або технічний відказ системи. Це є доволі зручним рішенням для можливості розширення. Також на сторінці з класами системи можна виконати певні дії над ними, а саме – видалити, перейменувати або поєднати – це зручний метод при випадковому створенню класів, закріплених за однією людиною. Зліва

на зображені, біля поля з класами є діаграма на якій є відсотки налаштувань системи для використання з певним класом. Це дозволяє в процесі роботи оцінювати ступінь налаштування та навчання системи для фактичного використання. Зручна діаграма показує ступінь навчання для кожного з класів, що надає змогу пришвидшити процес та більш детально його зрозуміти – наприклад скільки ще треба часу на налаштування та скільки ще фотографій треба завантажити для роботи. Під діаграмою з процентами налаштування системи є поле з статистикою кількості використання сервісу в день

Наступним кроком в процесі підготовки до роботи є процес навчання класифікації для створення унікальних математичних розрахунків, притаманних кожному обличчю. Для цього необхідно перейти в вкладу завантаження вхідної інформації (“Import”). Після відкриття відповідного вікна з налаштуванням, треба додати власні фото для проведення процесу налаштування сервісу. Сервіс підтримує використання різних типів фотографії, що прибирає потребу в зміні формату вже наявних фотографій.

Для успішного проведення етапу налаштування та навчання сервісу необхідно використовувати від 5 до 10 фотографій. Для забезпечення кращого результату варто використовувати фото при різному рівні освітлені, ракурсі та фоні. Також варто урахувати той факт, що люди можуть змінювати свої зачіски, для чоловіків актуальна проблематика з відрощуванням додаткових волосяних покровів на обличчі – борода, вуси або загальна небритість –, що може ускладнювати процес та зменшати точність ідентифікації. Для жінок це може бути, наприклад, зміна зачіски або наявність/відсутність макіяжу на обличчі та його стилю. Це наддасть змогу програмі більш точно проаналізувати риси обличчя та вираховувати відповідні математичні моделі для автоматичної класифікації. Для прикладу буде проведено заповнення та налаштування людини для класу “Меган Фокс”, для цього необхідно загрузити відповідну кількість фотографій на сайт та провести навчання. На рисунку 2.3 приведено результат завантаження відповідних фотографій для подальшого використання у налаштуванні через вкладку “Import”.

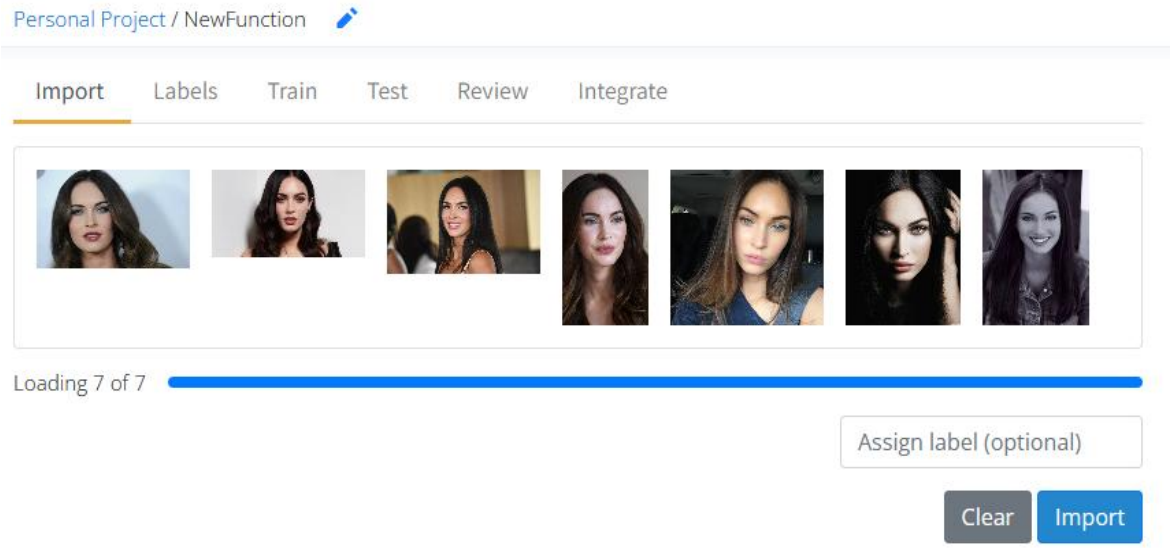


Рисунок 2.3 – Процес завантаження нових фотографій на сервіс

Після вибору відповідних фотографій та їх завантаження на веб-сайт, необхідно додати їх до галереї застосунку. Цей процес виконується за допомогою натискання на клавішу “Import”, після цього фотографії будуть додані до галереї поточного проекту.

Після цього треба перейти до вкладки “Train” та провести ручну ідентифікацію шляхом призначення відповідного класу до кожного з зображень. Таким чином відбується ручний процес налаштування системи з опором на дані, які вона отримала до цього. Це надає змогу сервісу вирахувати особливі риси, притаманні кожній фотографії певної людини, та побудувати на цій основні системи для класифікації. Для первинного налаштування та початку тестів необхідно завантажити декілька фотографій та провести ручну модерацію, як і було зазначено вище, але для покращення точності результату варто збільшити кількість фотографій, щоб система змогла краще класифікувати об’єкт на фото вже при використанні по прямому призначенню. Тобто для успішної роботи системи класифікації певного об'єкта, можна використати або мінімальну кількість фотографій для початку, або провести більш точне налаштування та навчання розпізнавання саме відповідного об’єкту і використати більшу кількість прикладів фотографії об'єкта. В контексті створення система ідентифікації

персоналу за використанням фотографії треба зробити відповідну кількість знімки, схожих на ракурс, який має робити камера в системі ідентифікації та завантажити їх на сайт. Оскільки ідентифікації, як було зазначено в першому розділі, відбувається за використанням веб-камери ноутбука, необхідно зробити кількість знімків з такого ракурсу, який схожий на ракурс камери. Саме тому в прикладі з додаванням відповідних знімків було обрано лише знімки обличчя, а не всієї фігури, оскільки точність результату від використання знімків такого типу зменшується. На рисунку 2.4 приведено процес ручної модерації галереї з завантаженими знімками для навчання системи.

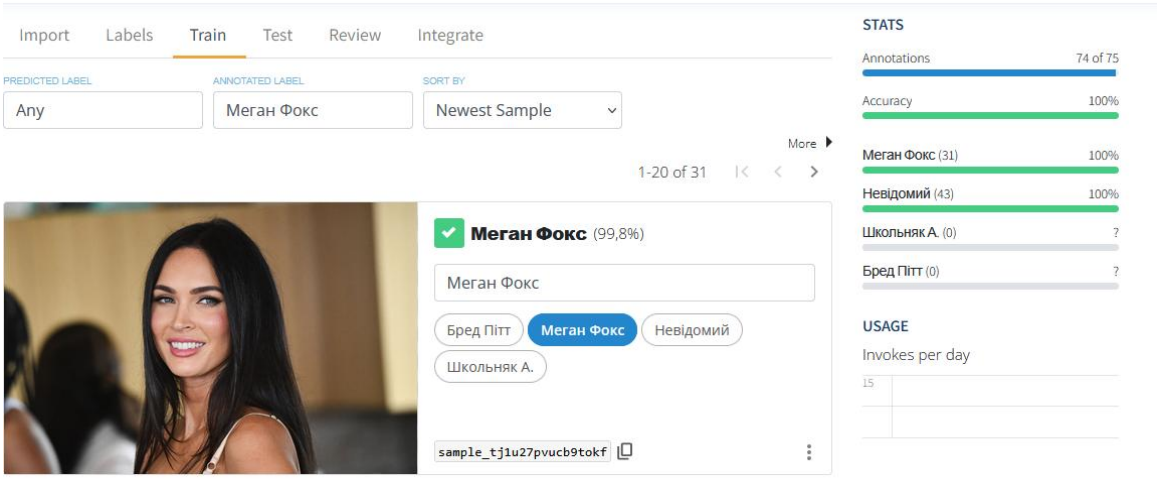


Рисунок 2.4 – Процес налаштування системи

Для переходу до тестування варто провести навчання системи згідно заданих класів, тобто – заповнити їх, щоб система могла порівнювати та класифікувати особу на фотографії. В контексті перших пробних тестувань варто також заповнити клас “Невідомий”, щоб збільшити точність класифікованих співробітників. Для налаштування будуть використовуватись стокові фотографії обличь людей з інтернету з великою кількістю вибірки для зменшення відсотку похибки при роботі системи ідентифікації. Після заповнення класу, що відповідає за незнайомців (можна сказати чужаків), буде виконаний первинний тест роботи системи ідентифікації та, додатково, заповнено інші класи, що відповідають за певних осіб. Це наддасть змогу провести повний тест системи, зрозуміти методи її

роботи та точність виконання класифікації людей по фотографіях. Також в системі ідентифікації варто використовувати лише самітні фотографії особи, щоб зменшити відсоток похибок при налаштуванні та навчанні системи. Опираючись на цю особливість, в виборці фотографії для навчання та налаштування системи немає сторонніх осіб на фото.

На рисунку 2.4 буде приведено процес навчання системи для розпізнавання чужаків. Для цього буде використано фотографії обличь людей з інтернету.

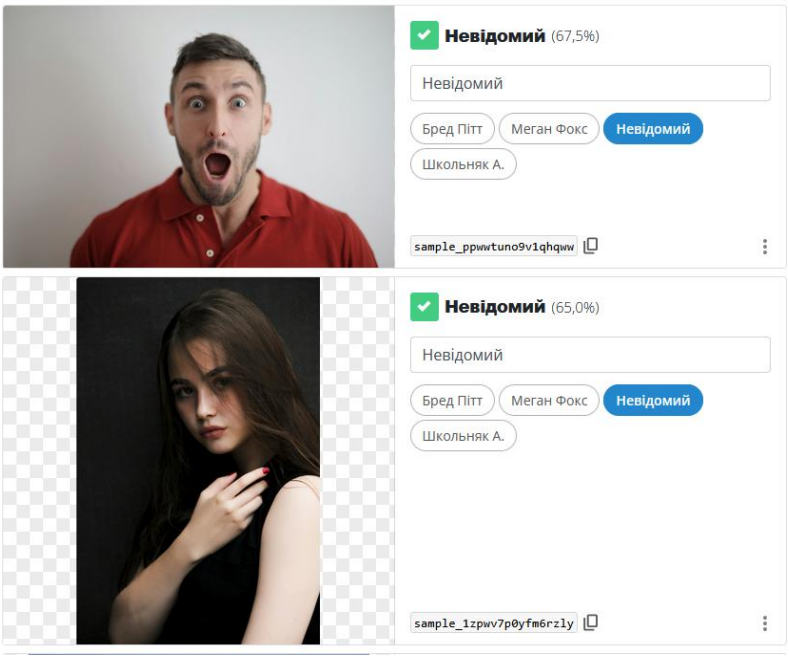


Рисунок 2.4 – Заповнення класу “Невідомий”

Для порівняння різних обличь та збільшення точності для заповнення даного класу було використано близько 40 різних фотографій, щоб відкалібрувати систему для надання максимальної точності. Наступним етапом, після заповнення класу, який відповідає за співробітника, в нашому випадку цим класом виступив клас “Меган Фокс”, та заповнення класу, який відповідає за чужаків та називається “Невідомий”, можна переходити до етапу проведення тестів роботи системи ідентифікації особи за використанням фотографії [25]. Процес налаштування системи і є складовою системи машинного навчання штучного інтелекту для подальшого використання. Проте, через описані вище особливості системи та її переваги, для початку роботи з системою вже не треба “проганяти”

через неї тисячі фотографій, що є зручним для налаштування в межах великих підприємств.

Для проведення первинних тестів налаштування системи класифікації варто перейти до вкладки “Test” на веб-сторінці сервісу. Після цього відкриється вікно з меню для додавання фотографії та можливістю отримати результат роботи. На рисунку 2.5 продемонстровано результат виконання роботи системи. На прикладі буде проведено класифікацію особи, яка належить до налаштованого класу.

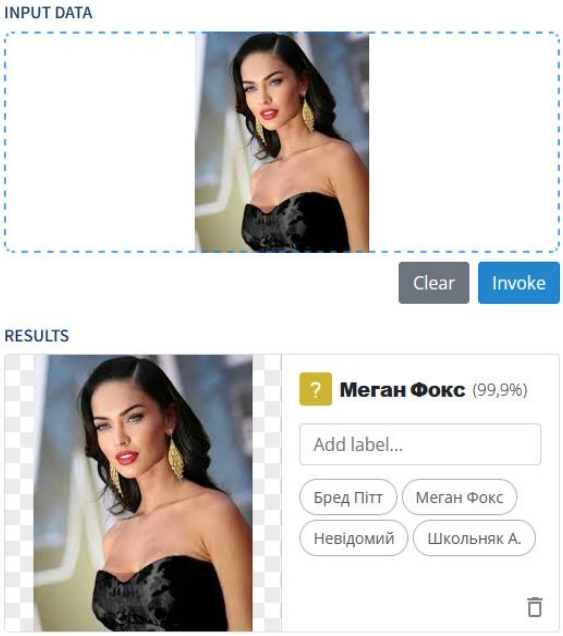


Рисунок 2.5 – Тест роботи системи класифікування

На зображенні вище продемонстровано роботу системи класифікації. Вона чітко опізнала об’єкт на фото та присвоїла йому відповідний клас в системі. Для додаткової точності треба обрати правильний варіант. Система класифікувала особу з вірогідністю в 99%.

Ці відсотки відповідають за здогадки відносно категорії об’єкту на фото, за якими штучний інтелект робить розмежування. Це надає змогу зрозуміти наскільки системи впевнена в виборі відповідного класу. Для більш зрозумілого сприйняття користувачем, використовується відсоткова система, оскільки точність варіюється від 0 до 1, де 0 – повна відсутність зв’язків та подібностей, а 1 – повна точність збігу. В основному, через незначні зміни, система може надавати

дуже близький результат до 1, але не давати повну гарантію. Точність цих даних для використання в моделі залежить від кількості прикладів для ручного налаштування та навчання моделі, саме тому при недостатній кількості початкових даних, система в майбутньому може працювати не коректно [26]. Але, навіть враховуючи все вище описане, система може працювати коректно при детальному налаштуванні та навчанні і може надати велику точність для розпізнавання об’єктів (в цьому випадку – співробітників по фото) та використовуватись як окрема та повноцінна система ідентифікації. Саме через описані вище особливості, використання та навчання систем такого типу, хоч і є легким, в порівнянні з конкурентами, все рівно вимагає детальних налаштувань та розумінню принципів роботи.

2.3 Розробка тестового додатку для тестів роботи системи

Система класифікації об’єктів на базі використання сервісу Nyckel також, крім вище зазначеного, надає змогу для інтеграції сервісу в вже існуючи або новостворені програми та додатки. Завдяки підтримки REST API [27], розробники мають можливість для автоматизації процесів в реальному часі. REST API це метод обміну інформацією між сервером та кінцевим користувачем, за умови надсилання на сервер запиту з повною інформацією, яку він має обробити. Процес використання цього методу необхідно забезпечити наявністю повноти інформації в запиті від клієнта до серверу. Виконати цю умову для робити можна, наприклад, за допомогою використання користувацького інтерфейсу взаємодії, такого як форми на веб-сайті [28]. Він базується на виконанні простих запитів протоколу HTTP в роботі, що трохи пришвидшує роботу в порівнянні з використанням HTTPS, проте позбавляє захисту в роботі додатку [29].

Це є дуже корисним при використанні системи для автоматичної модерації або створенні системи класифікації, де швидкість та точність визначення категорій має суттєвий вплив на безпеку в використанні платформи. Окрім цього,

можливості сервісу надають змогу для використання його в виконанні рутинних бізнес-задач без необхідності довгого та дорогого навчання. До переліку таких задач, наприклад, може відноситись обробка типових документів (резюме нових співробітників), система класифікації (для ідентифікації співробітників, класифікації товару), розпізнавання тексту (використання скан-копії або фотографії документу для створення нового в текстовому додатку без спотворення через копіювання копії документу) тощо.

Сервіс надає підтримку використання запитів на мові програмування Python, що є зручним методом розробки додатків. Саме на базі цієї мови програмування буде базуватись створення додатку для системи ідентифікації персоналу за використанням фото. Програмний додаток буде представляти з себе програмне вікно з кнопкою для додавання фотографії, що складається з кнопки для перевірки подібності та поле з виводом результату класифікації. Для забезпечення зручності використання програмного додатку, він буде створений за використання бібліотеки Tkinter, яка надає фунціонал для легкого створення програми з зручним вікном для користування [30]. Ця бібліотека є стандартним інтерфейсом Python, що надає забезпечення легкості використання та наявність великого кола підтримки в разі виникнення нетипової помилки. Використання цієї бібліотеки для створення візуалізації програми дозволяє збільшити зручність використання та перевести додаток від консольного варіанту до створення зручного в користуванні додатку з наявністю робочих кнопок, зміни фонів та різних видів відображення інформації. Крім створення візуальних складових додатку, програма надає методи для створення функційних кнопок та текстових полів.

Розробку програми можна умовно поділити на кілька етапів – створення графічного інтерфейсу, налаштування використання REST API та прописування функцій для використання кнопок та створення запитів до сервісу Nyckel. Для початку було проведено створення графічної складової, а саме – створення вікна програми, графічних елементів та їх визначення їх розташування для зручності користування. У графічному вікні з розмірами в 350 на 400 пікселей було

створено 2 кнопки – кнопка додавання фотографії та кнопка перевірки. При натисканні на кнопку додавання фотографії відкривається вікно файлового провідника, в якому треба вибрати фотографію для тестування. Кнопка, яка відповідає за відсилання запиту на перевірку, в цей час не функціональна. Це виконано для того, щоб уникнути випадкових запитів до системи та забезпечити захист від виконання пустих запитів, що може призвести до помилки. Після додавання фотографії для перевірки в рамках тесту системи, кнопка створення запиту на перевірку змінює свій стан та стає функціональною. Після чого на неї можна натиснути та провести перевірку. Після цього нижче буде виведено повідомлення з результатом тестування – з успішним проходження перевірки або помилці в виконанні процесу класифікації.

Розробка цього додатку ставить собі на меті тестування процесу надсилання запитів типу REST API до системи класифікації за використанням фотографії на базі використання сервісу Nyckel. Для проведення тестування буде використовуватись фотографія обличчя випадкової людини з мережі інтернет. Для тестування буде використовуватись модель, яка було налаштовано та навчено в попередньому підрозділі роботи.

На зображенні 2.6 приведено графічний інтерфейс створеного програмного додатку з уже пройденим процесом класифікації. Як і було зазначено вище – класифікувати система має особу, яка належить до списку невідомих. Відповідність проценту сканування не було відображено для уникнення графічного перевантаження відображуваної відповіді.

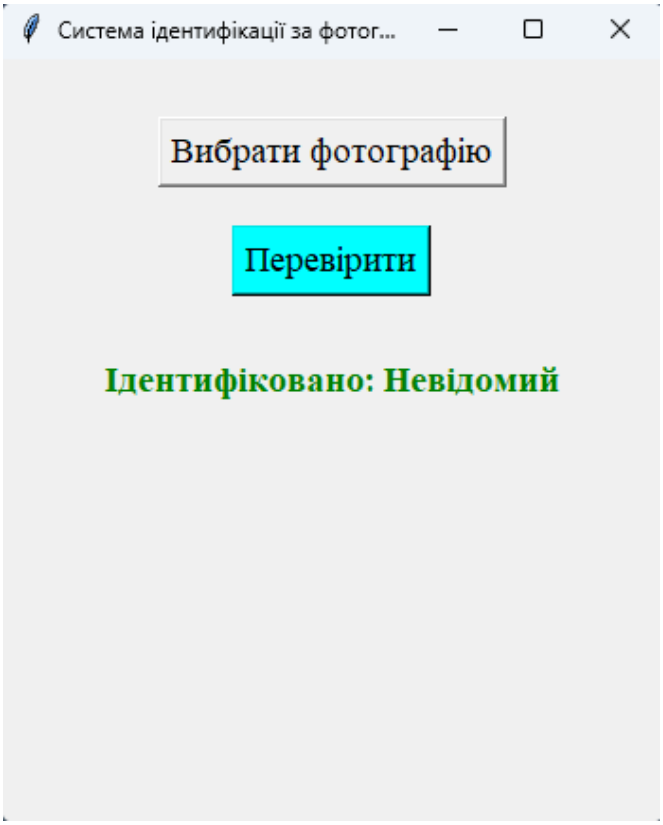


Рисунок 2.6 – Екран тестового програмного додатку

Як ми бачимо на приведеному рисунку, програма створила запит до система та отримала правильну відповідь на відповідний запит. Сервіс класифікації ідентифікувала особу за фотографією та відніс її до правильного класу – “Невідомий”. Таким чином було створено просту програму для реалізації системи ідентифікації з використанням інтегрування запитів до сервісу Nyckel в межах програмного коду.

Виконання перевірки на працеспроможність такого типу показала факт того, що система дійсно дозволяє інтегрувати її в вже наявні системи та програми без використання важких реалізацій, опираючись лише на описані джерела для цього. Порядок дій та їх опис, для використання різних методів інтеграцій в наявні та новостворені системи, описано на сторінці з документацією сервісу.

Для створення запитів типу REST API необхідно, крім програмної реалізації, підготувати певні дії з боку змінних. На рисунку 2.7

```
API_URL = "https://www.nyckel.com/v1/functions/8rknftc5tcmmlo9x/invoke"
TOKEN_URL = 'https://www.nyckel.com/connect/token'


CLIENT_ID = 'a902k4daceetn9z3kh5nsdqm3e1be4z6'
CLIENT_SECRET = 'h************************************************************dhkt'
```

Рисунок 2.7 – Код для використання API сервісу Nyckel

Так як і було описано вище – на зображенні 2.7 продемонстровано код для створення запитів типу API. Далі слід розібрати детального який рядок коду для чого використовується та його призначення. На першому рядку приведеного коду створено статичну зміну під назвою “API_URL”. Ця змінна необхідна для виклику відповідної налаштованої функції [27]. Цей рядок з змінною є необхідним при створенні запиту з передачею зображення. Ця зміна містить посилання на сервіс Nyckel та його сторінці, який відповідає за функції. В цьому рядку йде певний набір значень який відповідає ідентифікатору функції класифікації, яка була створена та налаштована вище.

Наступним рядком є рядок оголошення змінної “TOKEN_URL”. Цей рядок відповідає за підключення програми до сервісу. Він відповідає за підключення програми до сервісу та підтвердження повноважень для використання функції. Без нього неможливо використовувати функцію, оскільки сервіс не надає можливості цього зробити через питання конфіденційності та безпеки. Можливість використання та впровадження чужих функцій класифікації без відповідного контролю зі сторони власника була б вразливістю в плані безпеки та використання недоцільних даних для вже налаштованої та працюючої системи. Для прикладу – для налаштованої системи класифікації хатніх тварин (наприклад система класифікації тварин в притулках відповідно кольору та приналежності до виду) було б передані дані, які не відповідають налаштованій системі. Через це може бути зменшення точності результату, а через певний час зміна вже налаштованої бази та вихід з роботи всієї системи класифікації. Окрім зазначеного, подорожчання використання системи через зайві запити та їх обробка і виконання.

Після цього йде рядок програми з вмістом відомостей про особистий ідентифікатор для акаунту. Це надає змогу для процесу автентифікації користувача на сайті при використанні запиту до функції через API. Без використання таких цих рядків не можливо буде провести автентифікацію особи та надати їй можливість для використання функції системи. Автентифікація для використання системи має 2 поля – перше поле відповідає за ідентифікаційний номер аккаунта в системі та є відкритою інформацією. Друге поле містить вже більший ряд комбінації, в порівнянні з “CLIENT_ID” та має назву “CLIENT_SECRET”. В цьому полі міститься комбінація для ідентифікації аккаунту для використання в системі. Це є другим ідентифікатором для використання системи. Вона є чутливою, оскільки ця комбінація може надати змогу для використання системи зловмисниками заради наживи або компрометації системи класифікації.

Для прикладу – якщо зловмисник отримає важелі для відправки запитів для використання системи класифікації, яка використовується для модерації форуму, він може запустити в цю систему свої запити, які будуть змінювати точність формування та не відповідати заданим критеріям. Це, в свою чергу, може викликати помилки в точності роботи та збільшити час на обробку нової інформації, або повністю скомпрометувати систему, що змусить на деякий час прибрати модерацію на веб-ресурсі та повернутись до ручної. В цей же час може відбуватись атака спамом на ресурс або запускатись хибні повідомлення з фішинговими посиланнями. Більш детальний процес роботи системи, функції та методи для класифікації будуть описані нижче при проектуванні та створені програмного додатку для системи ідентифікації персоналу.

2.4 Висновки до розділу

Під час формування цього розділу було проведено процес налаштування та навчання системи класифікації за використання сервісу Nyckel. Було описано ключовий принцип в виборі основного сервісу для моделі класифікації. Як і було зазначено в підрозділі 2.1, система, яку надає сервіс, базується на використання полегшеного процесу машинного навчання. Це надає можливості для швидкого налаштування та впровадження системи без наявності глибоких знань в архітектуру системи, ручного навчання системи на основі тисяч шаблонів з ручною модерацією. Попри всі описані вище переваги, ця модель також має велику точність роботи, як і професійно створені моделі штучного інтелекту. Також в підрозділі 2.3 було сформовано тестовий програмний додаток для тестування можливостей роботи системи ідентифікації через використання API.

Процес налаштування та навчання системи був повністю описаний та розподілений на кілька частин: вибір моделі роботи (з зображенням, з текстом або табулярними значеннями) та метод обробки результату для вхідної інформації; детальний опис процесу створення класів для роботи системи, їх налаштування та заповнення прикладами через ручну модерацію завантаженого контенту; процес тестування точності роботи системи класифікації за допомогою вбудованих інструментів на веб-сайті сервісу та нових фотографій; створення тестового додатку для впровадження в нього функціоналу для відправки запитів по типу API з вмістом інформації для тестування на базі вже налаштованої системи.

В межах написання розділу також було обрано вхідні дані для обробки штучним інтелектом, а саме – фотографію та обрано тип даних, які мають повертатись після обробки інформації. В нашому випадку це використання фотографій та повертання значень класифікатору, що є цілком доцільним в межах формування цієї роботи та повністю задовольняє вимоги. Також були сформовані класи для створення моделі класифікації та проведено їх налаштування згідно з заданими характеристиками та відповідністю через ручну модерацію.

Варто зазначити, що сервіс надає можливість для використання його, як

частини, в сторонніх програмах. Виконується це за допомогою запитів типу REST API. Також, в рамках цього розділу, було описано механізм зворотного зв’язку від системи. Після проведення первинного навчання було проведено тестування системи з приведенням прикладів. Сервіс Nyckel надає змогу відслідковувати рівень точності системи, виводячи інформацію в відсотках. При запиті на сервер, програма отримує відповідь з певним набором інформації, а саме – приналежність до певного класу та відсоток точності роботи для кожного запиту, що є зручним інструментом для використання в інтеграції в програмних додатках. Опираючись на значення, які повертає система, є можливість для розробки методу порогового значення ідентифікації. Для прикладу – якщо рівень точності менший за 80%, результат не приймається як дійсний та доступ до системи заблоковано.

Отже, можна сказати, що під час формування та процесу написання цього розділу було оглянуто основні можливості використання сервісу Nyckel. Описано процес створення та налаштування функції класифікування з орієнтацією на використання обличь в рамках процесу ідентифікації персоналу. Також було розроблено тестовий додаток та описано основні вимоги для впровадження сервісу в рамках використання в ньому. Використання саме цього рішення є надійним фундаментом для використання в подальшому в реальних умовах для автоматизації процесу ідентифікації та надання доступу. Це дозволяє зменшити адміністративні витрати підприємства та збільшити точність отримуваної інформації.

3 РОЗРОБКА ПРОГРАМНОГО ТА ТЕХНІЧНОГО РІШЕННЯ ЗАБЕЗПЕЧЕННЯ ІДЕНТИФІКАЦІЇ

3.1 Розробка технічної складової системи

Для технічної реалізації фізичного модулю для забезпеченню роботи системи ідентифікації персоналу за використанням хмарного сервісу машинного навчання Nyckel, було спроектовано та розроблено модель фізичного пристрою. Фізична модель базується на використанні кількох основних модулів: модуль з ультразвуковим датчиком відстані HC-SR04, який дозволить робити запити до програми лише при наявності людини в межах роботи, що зменшить навантаження на програмну складову системи. Іншим модулем у фізичному виконання буде модуль радіочастотної ідентифікації RC552. Цей модуль є додатковим рівнем захисту системи від несанкціонованого проникнення та дозволить більш зручно оперувати системою. Радіочастотна ідентифікація буде необхідна для доступу до адміністративного облікового запису. Це необхідно для створення системи розмежування доступу та покращення захисту системи. Без використання розпізнавання обличчя та наявності RFID-брелка, не можна буде переглянути журнал логування дій в системі та ввести зміни до неї.

Основною метою розробки та впровадження описаної вище фізичної моделі для роботи системи є реалізація взаємодії між сервісом машинного навчання та кінцевим виконавчим пристроєм, на якому буде виконуватись основна дія з обробки інформації та прийняття відповідних рішень. Для обробки інформації з результатами тестування від сервісу Nyckel буде використовуватись комп’ютер з інстальованим програмним додатком.

Логіка реалізації функціоналу наступна: людина підходить до фізичного датчику, її фіксує ультразвуковий датчик відстані та відправляє запит на створення фотографії до Веб-камери та формування запиту до сервісу Nyckel. Після цього програма отримує відповідь від серверу, де сервіс проводить перевірку та співставлення особи з вже існуючою базою класифікації персоналу. Наступним кроком програма робить висновки щодо надання доступу відповідно

до точності результату ідентифікації особи. Для отримання адміністративних повноважень необхідно пройти ідентифікації та авторизацію. Цей процес виконується через програмний додаток, далі треба прикласти брелок до зчитувача. Система перевіряє відповідність брелка з вже існуючим в базі та надає рішення щодо надання доступу.

На рисунку 3.1 приведено копію електричної структурної схеми системи. Повна схема знаходиться за Додатком А.

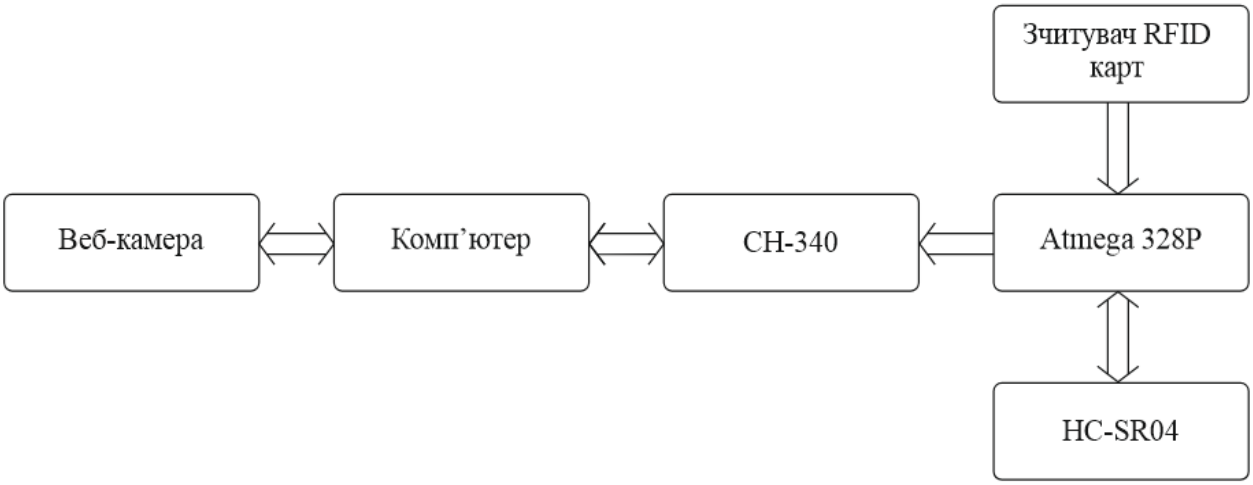


Рисунок 3.1 – Електрична структурна схема

На схемі, яка приведена вище, продемонстровано електричну структурну схему роботи фізичного додатку в поєднанні з ПК з підєднаною веб-камерою. У приведеній схемі ПК виконує роль основного обчислюваного елемента, який відправляє запит на створення фотографії з веб-камери, взаємодіє з вторинним мікроконтролером Atmega 328P, який відповідає за взаємодію з фізичними датчиками. Також на ПК інстальовано програмне забезпечення та створюються журнали логування дій для відслідковування взаємодій співробітників з системою ідентифікації. Мікроконтролер на базі процесора Atmega 328P взаємодіє на локальному рівні з датчиками та проводить їх періодичне опитування для отримання актуальних даних. Такий підхід гарантує мінімальні затримки в роботі та обислиннях, уникненню створення зайвих запитів до сервісу ідентифікації Nyckel (наприклад кожні 5 секунд з новим фото) та прибирає можливості для

хибного спрацювання системи. Для більшого розуміння та обґрунтування вибору саме таких компонентів для реалізації, нижче буде приведено опис комплектуючих технічної складової системи ідентифікації персоналу та зазначено їхні характеристики, які стали важливим фактором в обранні саме такого рішення для створення:

– Мікроконтролер на базі Atmega 328P. Цей контролер містить 8-бітни процесор з тактовою частотою в 16 МГц, який є головним обчислювальним механізмом. Також він оснащений 32 КБ Flash пам’яті, 2 КБ SRAM. Він має 14 цифрових входів/ виходів та 8 аналогових [31]. В рамках конкретного фізичного виконання контролер відповідає за отримання сирих даних від датчиків та їх первинну обробку. Це дозволяє зменшити навантаження на ПК та полегшити процес взаємодії його з різними датчиками. На рисунку 3.2 приведено зображення цього контролера.

Рисунок 3.2 – Мікроконтролер на базі Atmega 328P

– Мікросхема CH340 (USB-UART). Це мікросхема входить до контролеру та відповідає за інтеграцію взаємодіє мікроконтролера через шину USB з ПК. Мікросхема створює в ОС ПК віртуальний COM-порт, забезпечуючи стабільне та захищене з’єднання без втрати важливої інформації. Саме через використання цієї мікросхеми відбувається створення з’єднання та подальша взаємодія між

комп’ютером, який виступає в ролі сервера, та мікроконтролера Atmega 328P, який працює з датчиками.

– Модуль радіочастотної ідентифікації RC522. Він представляє з себе плату, яка під’єднується до мікроконтролера та відповідає за сканування RFID брелків. Він відносить до високочастотних датчиків та працює на частоті 13,56 МГц (Mifare). З’єднання з Atmega 328P відбувається через використання шини SPI. Дальність зчитування досягає 5 см, а робоча напруга становить 3.3V [32]. Використання високочастотного сканера надає додатковий фунціонал для забезпечення безпечного контуру в середині системи, оскільки карти такого типу доволі важко клонувати. Нижче на рисунку 3.3 приведено зовнішній вигляд модулю.

Рисунок 3.3 – Модуль RFID RC522

– Ультразвуковий датчик HC-SR04. Це модуль для безконтактного вимірювання відстані (від 2 см до 4 м) з погрішністю в 3 мм. Датчик працює за принципом ехолокації та генерує імпульси на частоті 40 кГц, які не чує людське вухо. Датчик працює від живлення в 5 В [33]. В фізичній моделі реалізації системи датчик буде використовуватись для отримання інформації про наявність особи для ідентифікації перед веб-камерою. Після цього буде запущено скрипт для роботи та створено запит до сервісу. На рисунку 3.4 приведено зовнішній

вигляд датчику.

Рисунок 3.4 – Ультразвуковий датчик HC-SR04

Вся схема фізичного виконання буде доволі компактна та розміщена на макетній платі на 400 отворів. На рисунку 3.5 приведено схему фізичного підключення компонентів. Повна схема приведена в Додатку Б.

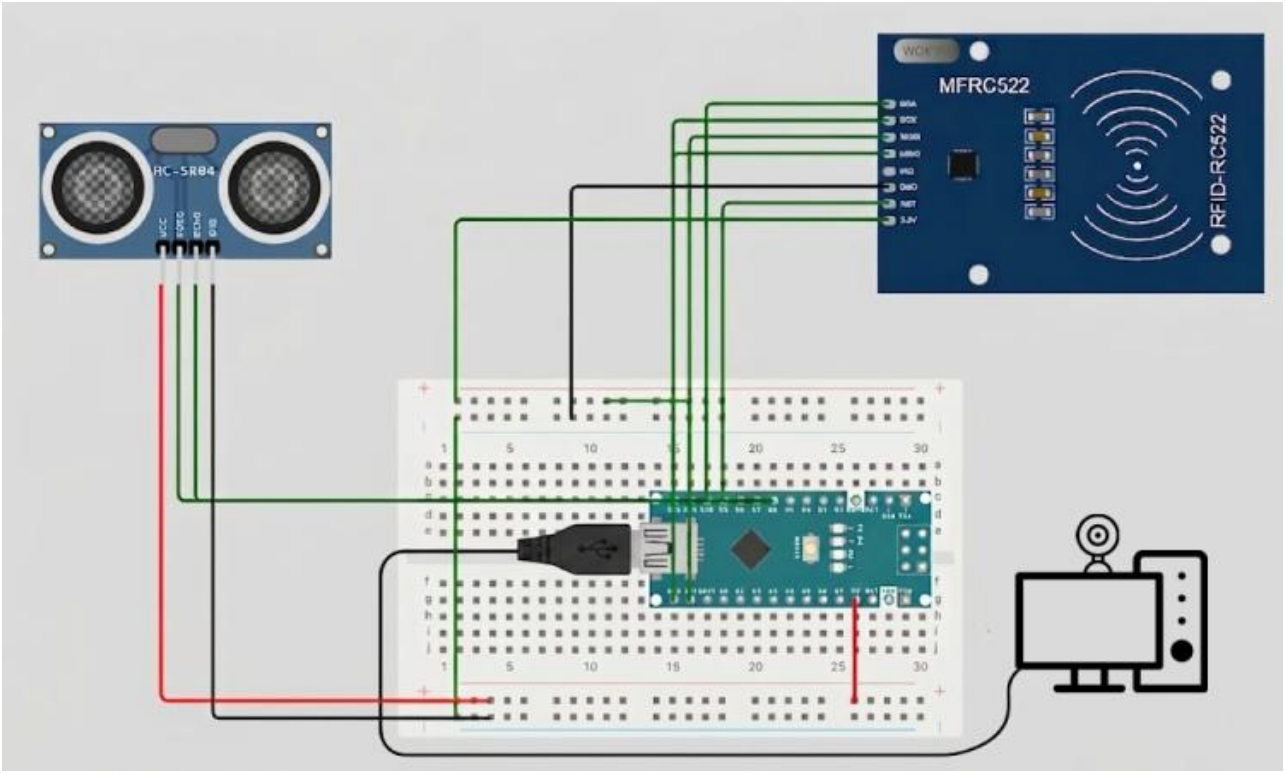


Рисунок 3.5 – Зібрана схема фізичного модулю

На приведеному вище рисунку продемонстровано схему з’єднання компонентів фізичного модулю системи між собою за використанням макетної плати. Живлення компонентів відбувається від центрального контролера на базі

Atmega 328P. Макетна плата надала змогу для створення двох окремих контурів живлення компонентів з напругою в 3.3 В та 5 В відповідно. Мінусом в системі живлення виступає загальний мінус на платі мікроконтролера.

Написання логіки роботи для фізичного модулю виконується на мові програмування C++. Конфігурація системи завантажується на через мікросхему CH340.

Повний код для конфігурації наведеної вище зібраної фізичної складової приведено в Додатку Д. Проте варто розглянути окремі функції, які відповідають за функціонал системи. Нижче, на рисунку 3.6, розглянуто функціонал конфігурації для знаходження відстані та виведення поточних даних в терміналі.

```
long duration = pulseIn(ECHO_PIN, HIGH);
long distance = duration * 0.034 / 2;

if (distance > 0 && distance < 400) {
  if (millis() - lastDistPrintTime > 500) {
    Serial.print("DIST:");
    Serial.println(distance);
    lastDistPrintTime = millis();
  }
}
```

Рисунок 3.6 – Функція знаходження відстані

Ця частина програми відповідає за обчислення відстані від ультразвукового датчика відстані HC-SR04 до об’єкта перед ним та виведення інформації про відстань в монітор порту.

Першочергово функція “pulseIn(ECHO_PIN, HIGH);” вираховує тривалість сигналу високого рівня на вказаному піні підключення. За час заміру відбувається запуск ультразвукового імпульсу, його відбиття та повернення до сенсора. Після цього виконується сталий математичний розрахунок для знаходження відстані за формулою “distance = duration * 0.034/2”. Такі цифри зумовлені поширенням швидкості звуку в повітрі (340 метрів в секунду). Результат обчислення треба

ділити на 2 оскільки хвиля виконує заміри до об’єкта та повернення до сенсора.

Далі в коді є умова для “відсікання” хибних сигналів, оскільки дальність роботи сенсора 4м, або 400 см. Остання частина функції використовує “millis()” для створення таймера, щоб запобігти надто великій кількості запитів до сенсора та уникнути зайвого навантаження як на сенсор так і на сам мікропроцесор плати для обробки інформації.

3.2 Розробка логіки роботи програмного додатку

Крім фізичної складової системи, для її функціонування необхідно ще і програмний додаток, який буде керувати отримуваною інформацією. Він отримуватиме сигнали про знаходження людини біля датчику, що є сигналом для створення фотографії через веб-камеру та відправки запиту до сервісу машинного навчання Nyckel.

Програмний додаток буде представляти з себе повноцінну систему для роботи з документацією для компанії та адміністрування цієї системи. В цій програмі буде декілька рівні доступу до системи та повноважень в ній, а саме:

– адміністратор безпеки – особа з цим профілем системи розмежування доступу, має повноваження переглядати журнал логування подій за останній день, перегляду існуючих працівників та їх приналежність до певних груп в системі (бухгалтерія, кадровий відділ, відділ закупівель тощо);

– головний адміністратор – це є найвищим за рівнем наданих прав доступу в системі розмежування доступу, який може переглядати журнал логування подій за весь час, перегляду існуючого персоналу з можливістю видалення та/або зміни його приналежності до відділів в середині компанії та можливістю налаштування робочих годин та вихідних днів;

– користувач – це профіль з мінімальними правами доступу в системі, окрім документації з приналежністю до свого відділу (бухгалтерія, кадровий відділ, відділ закупівель тощо). Цей профіль відповідає за можливість до роботи у

користувачів з різних відділів тільки з файлами власної зони відповідальності без можливості перегляду документації з інших відділів.

Розмежування системи за такими групами наддає змогу для створення безпечної робочої моделі та забезпечує систему розмежування доступу та створенню необхідних умов мінімізації ризиків, пов’язаних з зловживанням отриманої інформації, роботи з нею та її компрометації (зміна груп користувачів та доступів, зміна файлів бухгалтерії тощо). Далі буде описано алгоритм роботи ідентифікації та авторизації програмного додатку з обґрунтуванням вибору саме цих рішень.

Програмний додаток забезпечує двохфакторну автентифікації для створення додаткового рівня захисту та відкидання помилкових спрацювань при випадкових умовах, що унеможливлює отримання небажаного та/або випадкового доступу до системи Для проходження ідентифікації з подальшою авторизацією в системі необхідно пройти ідентифікацію за розпізнаванням фотографії обличчя та ввести унікальний пароль для кожного облікового запису. Для отримання доступу до адміністративного аккаунта також додатково необхідно прикласти радіочастотний брелок до сканера. Це необхідно для забезпечення захисту системи та отримання необхідного рівня доступу оскільки користувач з адміністративним доступом може скомпрометувати діяльність в системі та призупинити роботу, а створення додаткових методів для ідентифікації надає можливість для зменшення ризику.

На рисунку 3.7, який буде приведено нижче, продемонстровано схему алгоритму, повністю приведеного в Додатку В, яка відповідає за процес ідентифікації, авторизації та автентифікації користувача в системі перед можливістю роботи в ній [34].

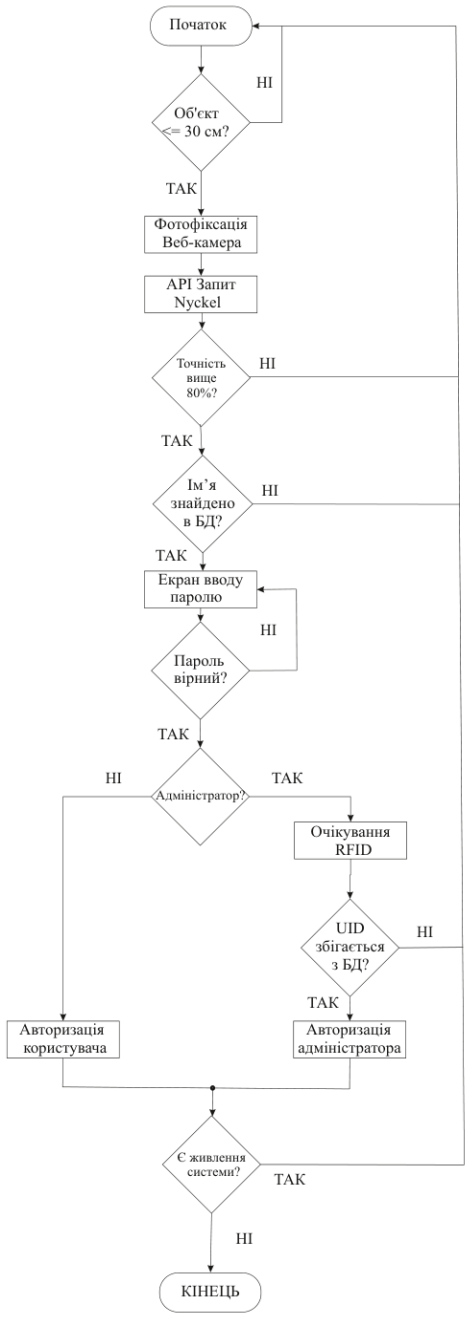


Рисунок 3.7 – Логічна схема дій для ідентифікації користувача

Описаний на схемі процес ідентифікації персоналу відповідає алгоритму та послідовності виконання операцій щодо надання можливостей для ідентифікації. Ця система забезпечує добрий рівень захисту системи від небажаного проникнення. Також додатковий ввід паролю для авторизації забезпечує надання небажаного доступу до системи, коли в цьому немає необхідності. Це є доволі зручним рішенням, оскільки це і задовольняє вимоги в додатковому захисті системи так і не створює великих складностей в реалізації та використанні

кінцевими користувачами.

Для більш детального розуміння схеми варто більш детально зупинитись на процесі та різниці між ідентифікацією, авторизацією та автентифікацією

Ідентифікація це процес під час отримання доступу до системи, який відповідає за доступ до певного облікового запису. Він полягає в співставленні когось або чогось з вже існуючою базою [35]. В рамках програмного додатку, цей процес буде виконуватись автоматично через використання веб-камери та сервісу Nyckel.

Авторизація це процес керування рівнями доступу до ресурсу, а саме – бази даних в середині додатку [36]. В залежності від введеного логіну та паролю користувачу буде надаватись відповідний доступ – як і адміністративний так і до баз даних (бухгалтерія, відділ кадрів тощо).

Автентифікація це процес який відбувається між ідентифікацією та авторизацією. Під час цього процесу відбувається встановлення відповідностей певного ідентифікатора до інформації [37]. В створеному програмному додатку це буде розмежування доступу до журналу логів чи до баз даних.

3.3 Реалізація програмного додатку та опис роботи системи

Програмний додаток буде розроблений за використанням мови програмування Python. Саме такий вибір був зумовлений простою взаємодією та можливістю використання REST API сервісу Nyckel. Ця мова програмування надає широкий спектр для розробки, використовуючи базові бібліотеки, як і було описано вище.

Взаємодія фізичного модулю буде відбуватись за допомогою мікросхеми CH340 яка використовує віртуального COM порту та технології UART. Це старий тип підключення та взаємодії пристроїв між собою. Він з’явився як тип підключення периферійних пристроїв до основної машини. Зараз він є популярним в сфері використання комп’ютерної діагностики автомобільної

електроніки, підключені невеликих плат [38]. Проте через свою простоту він не надає потрібної швидкості для передачі інформації, саме тому він втратив свою масовість та актуальність в загальних цілях [39].

Програма буде представляти з себе повноцінний додаток. Першим екраном взаємодії буде вікно для вводу даних для авторизації в системі. Методом для проходження ідентифікації буде розроблена та впроваджена система для ідентифікації за використання фотографії на базі сервісу Nyckel. Це є зручним методом та дозволить людині вводити лише пароль для авторизації в системі. Для адміністративних облікових записів є певний виняток, бо для доступу їм треба ще прикласти брелок для доступу.

Для початку роботи треба запустити програмний додаток, після чого буде відкрито початковий екран, на якому робиться фото з Веб-Камери пристрою. Для авторизації було використано модуль HC-SR04 який збирає інформацію про відстань людини від пристрою. Для роботи треба підійти на відстань в 40 см до датчику. Після цього буде автоматично виконано фото для перевірки та створено запит до сервісу машинного навчання Nyckel. Таким чином відбувається ідентифікація особи. Після етапу проходження ідентифікації та отримання позитивної відповіді, треба ввести унікальний пароль для входу в систему. Після правильного введення паролю буде надано доступ до документів, які відповідають базі даних.

Як і було зазначено вище – в системі буде кілька рівнів доступу, що наддасть змогу для розмежування системи та повноважень в ній. Для входу під звичайними обліковими записами користувачам необхідно просто ввести пароль, таким чином підтвердити ідентифіковану особу. Для входу адміністратору безпеки варто, крім зазначеного вище вводу паролю, прикласти відповідний брелок для отримання доступу до системи. Брелок містить радіочастотний датчик. Це є додатковим рішенням для створення захищеної системи та уникнення можливості для розголошення технічної інформації, такої як список логів користувачів для відслідковування часу приходу на роботу кожного з них та зміні груп користувачів. Запити до сервісу Nyckel формуються за таким самим

принципом, як і було описано вище при створенні тестової програми. Для проходження до етапу вводу пароля треба пройти тест з точністю в 80 процентів, або більше. Весь код програмного додатку приведено в Додатку Ж. Якщо тест було пройдено на менший результат – треба пройти його повторно.

При вході в аккаунти з правами адміністрування системи та зміни прав користувачів треба пройти ідентифікацію за допомогою зображення обличчя, після цього треба ввести пароль. Після успішного введення паролю буде виведено екран, на якому описані дії при авторизації, а саме – прикласти RFID брелок до зчитувача. На рисунку 3.8 приведено функцію, яка відповідає за цей етап роботи програми.

```
if user:
    self.current_user = {
        'id': user[0], 'username': user[1], 'full_name': user[2],
        'role': user[3], 'department': user[4], 'rfid_uid': user[5]
    }
    if self.current_user['rfid_uid']:
        self.show_rfid_screen()
    else:
        self.log_action(self.current_user['username'], "Успішний вхід (FaceID + Pass)")

        if self.current_user['role'] == 'SUPER_ADMIN':
            self.show_super_admin_screen()
        else:
            self.show_user_screen()
else:
    self.log_action(self.identified_username, "Невдала спроба входу (невірний пароль)")
    self.error_label.configure(text="Невірний пароль")
```

Рисунок 3.8 – Перевірка рівня доступу

Як і було зазначено вище – ця частина відповідає за перевірку рівня доступу до системи. Система робить перевірку та надає доступ до певного функціоналу, екрану програми. Для входу в адміністративний аккаунт треба додатково провести сканування мітки Rfid.

На першому етапі програма аналізує результат від запиту до бази даних програми з обліковими записами. Для цього процесу створюється локальний масив для збереження даних з бази даних з ім’ям, посадою, ідентифікатором та, за

потреби, унікальним номером радіочастотної мітки. У разі успішної перевірки всіх факторів здійснюється автоматичне перенаправлення до відповідного екрану для виконання роботи. Усі дії з входу фіксуються в системному журналі логування входів.

При введені неправильних даних алгоритм блокує доступ до системи, показує повідомлення про це та реєструє інцидент в журналі. Це дозволяє полегнити процес розслідування та аудиту системи.

Це є необхідним кроком, як і було описано вище. Після успішного зчитування ідентифікатора з брелка буде надано доступ до системи адміністрування.

На рисунку 3.9, який приведено нижче продемонстровано екран програми при вході з адміністративним доступом до системи та можливості для дій в ньому.

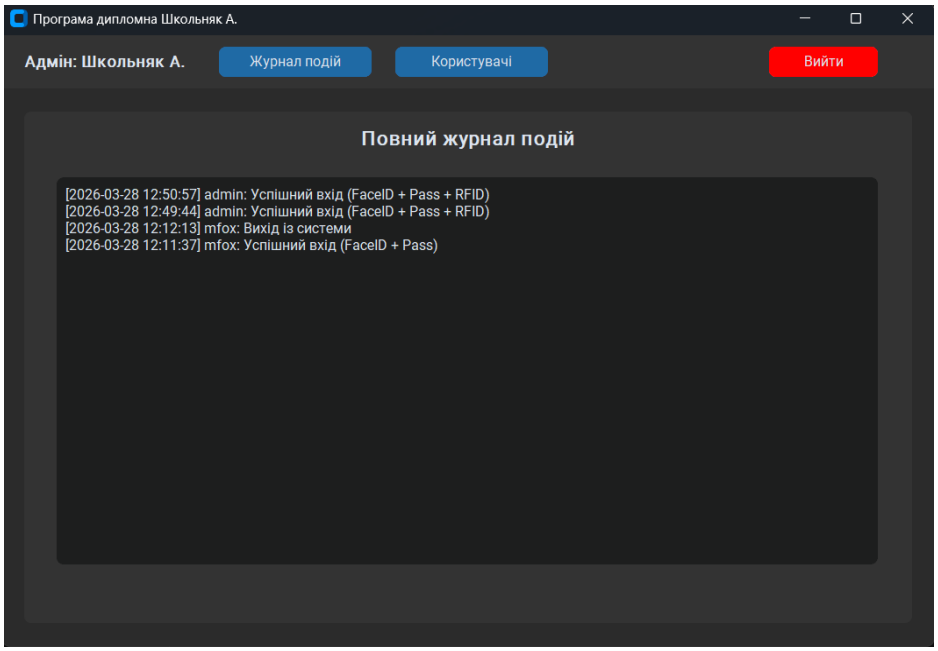


Рисунок 3.9 – Головний екран програми

На зображені, яке приведено вище можна бачити головний екран програмного додатку під обліковим записом адміністратора. На ньому ми можемо вивід повного журналу дій в системі, а саме – час входу в систему, обліковий запис, який увійшов до неї. Окрім того – метод ідентифікації та підтвердження особи та результат спроби входу: успішний або не успішний. Такий зміст журналу

логування подій надає змогу для відслідковування дій у системі та взаємодії її з персоналом та співробітниками. Таким чином можна формувати політику компанії для ідентифікації в початку робочого дня для отримання доступу до бази даних компанії. Це є зручним рішенням для формування журналу роботи кожного співробітника, відслідковування його часу роботи та фактичних дій в системі. В той самий час витік даних залишається можливим, але складність цього процесу значно покращена в сторону захисту персональної та технічної інформації компанії.

Наступною вкладкою облікового запису з адміністративними повноваженнями це вкладка “Користувачі”. На рисунку 3.10 приведено вміст описаної вкладки програмного додатку.

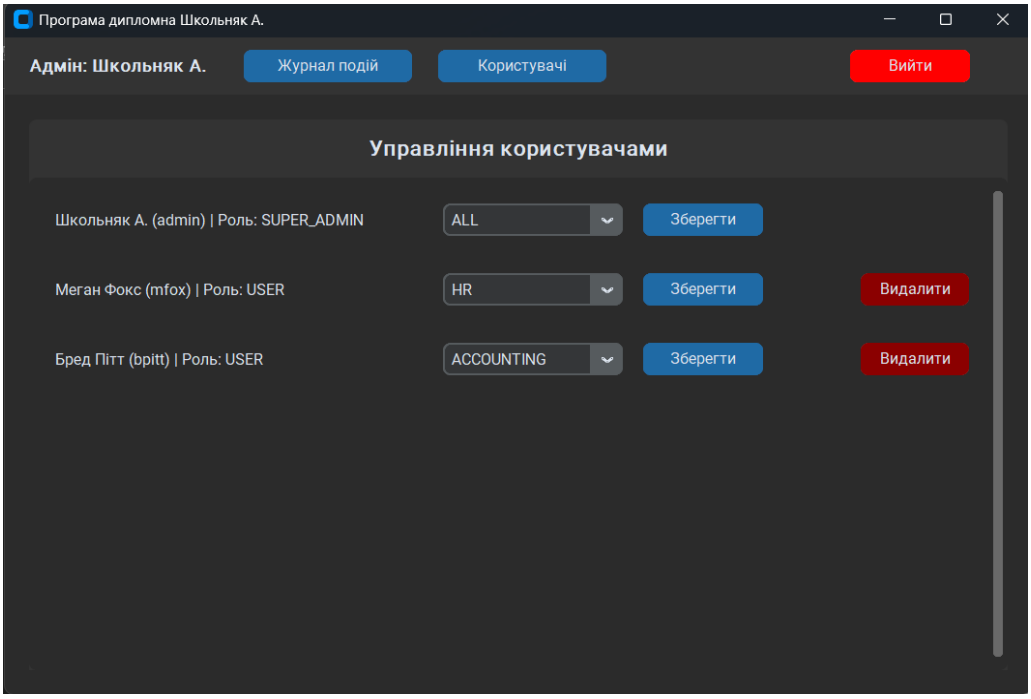


Рисунок 3.11 – Сторінка керування користувачами

На сторінці, яка була наведена вище, виведено загальний список користувачів системи, тобто – користувачів програми. Тут приведено їх імя, в скобочках прописано ідентифікатор (логін) та роль. Програма під адміністративним доступом надає функціонал для зміни приналежності до певних груп користувачів, таких як бухгалтерія, кадровий відділ та відділ закупівель та

функціонал для видалення співробітника для блокування доступу до системи. Фунціонал для зміни приналежності користувачів до різних груп зменшує навантаження на адміністратора система, оскільки він має фунціонал виконати ці дії без окремих запитів до бази даних до системи обліку, опираючись лише на готові команди. Дані компанії зберігаються на локальній базі даних, що забезпечує їх структуризацію та зручність для користування. Для створення бази даних було обрано SQLlite. Цей інструмент є вбудованим до Python та не вимагає додаткової інсталяції [40], а через високу популярність та наявність великого кола співробітників має простоту в налаштуваннях та вирішення типових проблем при формуванні запитів.

3.4 Опис рекомендацій щодо впровадження системи

В рамках цього розділу було описано процес створення системи ідентифікації персоналу за використанням зображення обличчя. Ця система є автоматичною та напівавтономною. Це надає змогу для її використання та впровадження в реальних умовах на умові використання в рамках системи ідентифікації персоналу на підприємстві або установі різних розмірів. А використання журналу логування входу та додаткової ідентифікації для користувачів з адміністративним доступом надає змогу для розширення можливостей системи та прав без втрати відповідного рівня безпеки у системі.

Через створені методи захисту інформації, які були описані в минулих розділах, система задовольняє потреби в розмежування доступу та блокування отримання доступу до певних користувачів. Також створення двох окремих рівнів доступу до адміністративних прав задовольняє потребу системи в обмеженні влади в руках однієї людини, оскільки це може викликати небажані наслідки в майбутньому.

Фізичний модуль можна помістити, наприклад, на пропускному пунктів з невеликою модифікацією програми, щоб прохід був дозволений та подано

відповідний сигнал на турнікет або магнітний замок з створенням запису до системи. Це є можливим через використання мікропроцесору на базі Atmega 328P. Також ця система може бути частиною процесу ідентифікації при використанні в компаніях з однією робочою станцією для доступу та роботи.

Через компактність фізичного модулю та його можливість для використання в різних типах корпусу – як металевих так і пластикових, великих чи компактних – це є універсальним методом для створення системи ідентифікації особи та подальшому впроваджені в компанії в різних сферах доступу та методах використання.

3.5 Висновки до розділу

Під час формування цього розділу було описано, розроблено та створено комплексу для ідентифікації персоналу за використанням сервісу Nyckel. Система містить в собі як і програмну частину в вигляді програмного додатку для керування так і фізичне виконання в вигляді повноцінного модулю для забезпечення потреб програмного додатку.

Було описано вибір мови програмування та функціонування програмного додатку, в поєднанні з роботою фізичного модулю. Завдяки використанню вбудованих бібліотек для мови програмування Python було створено кінцевий додаток, зручний для використання користувачами системи та реалізовано певний перелік систем доступу та захисті даних з розділенням між підрозділами згідно описаних груп.

Важливим етапом в забезпеченні моделі безпеки, такої як розмежованих груп доступу є використання вбудованої бази даних SQLlite. Збереження інформації в такому вигляді дозволяє збільшити рівень захисту системи та зробити більш структуровану систему доступу та зберігання інформації, в порівняні з використанням звичайних текстових документів для збереження. Також це дозволить дотримуватись розроблених та оголошених політик щодо

доступів до інформації в системі.

Сама система ідентифікації базується на використанні сервісу машинного навчання Nyckel. Такий варіант виконання надає змогу для зменшення навантаження на поточну робочу станцію та зменшення фінансових витрат для утримання, розробки та навчання власних систем ідентифікації та підтримки їх роботи в перспективі. та більш гнучкі можливості для використання “заліза”, що надає змогу для задовільнені потреб в використанні бюджетних рішень.

Підсумовуючи – створений, в рамках цього розділу, комплекс системи ідентифікації персоналу відповідає сучасним нормам та вимогами до систем ідентифікації такого класу. Поєднуючи в собі доволі надійні механізми та системи для фізичної ідентифікації на базі використання контролера з мікропроцесором Atmega 328P та новітніх методів класифікації та ідентифікації, що надало змогу для створення гнучкої та масштабованої системи. Ця система готова до використання “в полі”, до прикладу як – офіс компанії, яка працює в інформаційній галузі. Фізичне виконання модулю для роботи системи є простим та надійним рішенням, яке дозволить працювати системі доволі довгий час без потреби в обслуговувані або ремонті, а використання окремих модулів, а не готового рішення, покращує ремонтостійкість системи та можливості для модернізації з змінами потреб та вимог, доводячи її, майже, до безкінечної.

ВИСНОВКИ

Під час написання пояснювальної записки до дипломної роботи з заданої теми було спроектовано та розроблено схему ідентифікації персоналу за використанням сервісу машинного навчання Nyckel. Цей проект наглядно демонструє процес створення власної системи контролювання доступу та ідентифікації персоналу з можливістю багатофакторної автентифікації та різних видів доступу. Створений комплекс технічних рішень об'єднує в собі засоби фізичного виявлення людини в полі роботи об'єкта, взаємодії через REST API з хмарним рішенням з розпізнавання та ідентифікації персоналу та програмний засіб для керування правами доступу до інформаційних ресурсів підприємства.

Фізична частина створеної системи базується на використанні мікроконтролера, який виконує взаємодію з ПК та датчиками для отримання актуальної інформації. Умова для створення фотографії людини для подальшої перевірки виконується завдяки інтеграції в систему ультразвукового датчика відстані, що дозволяє створювати запити лише при наближені людини до певної відстані від модулю. Такий підхід забезпечить зниження побічного навантаження на систему та дозволить запобігти хибним спрацюванням. Додатковий контроль захисту системи забезпечує наявність в ній датчику радіочастотної ідентифікації, який також інтегровано в кінцеву систему роботи. Фізична взаємодія між датчиками та основним комп'ютером виконується через стабільне з'єднання при використанні протоколу послідовної передачі даних.

Програмна частина системи ідентифікації реалізована за використанням мови програмування Python та має простий та зрозумілий інтерфейс роботи. Для зберігання інформації про систему та її складові (документи, файли) було реалізовано вбудований модуль створення баз даних, що дозволить покращити систему безпеки інформації та забезпечить її цілісність та конфіденційність. Доступ до сегментів бази даних розділено на різні доступи для уникнення компрометації та несанкціонованої зміни інформації, наприклад – персонал з кадрового відділу не має доступу до файлів бухгалтерського відділу. Також в

програмний додаток було впроваджено систему для фіксування дій логування в журналі подій, що надає змогу для більш зручного та швидкого розслідування інцидентів різного типу.

Ключовою особливістю системи є використання сервісу машинного навчання Nyckel, що одночасно полегшило процес налаштування та введення в роботу системи, без необхідності використання значних ресурсів – як фінансових так і людських – для побудови власної системи ідентифікації. Програма працює за наступним принципом: при знаходженні людини перед камерою, ультразвуковий датчик це помічає та посилає запит для створення фотографії та формує запит до сервісу ідентифікації, після чого вже отримує результат та обробляє його. Зроблені фотографії не зберігаються на кінцевому пристрої, що підвищує рівень конфіденційності співробітників. А використання встановленого порогового значення для рівня ідентифікації дозволить уникнути хибних ідентифікації та робить систему більш стійкою до колізій.

В кінцевому висновку можна сказати, що результат проведеної роботи в рамках дипломної роботи має високу практичну цінність. Розроблена система ідентифікації є повністю функціональним прототипом, який готовий до повноцінної роботи та можливостей подальшого масштабування та впровадження. Використання популярної технічної бази для реалізації фізичної частини у поєднанні з новітніми рішеннями класифікації робить цю систему доволі бюджетною та ремонтостійким продуктом в умовах сучасного використання в корпоративному сегменті. Створена система в рамках роботи є досить практичним, компактним та захищеним рішенням, яка забезпечує високу точність роботи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Система контролю і управління доступом. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Система_контролю_і_управління_доступом (Дата звернення: 03.03.2026).

2. Як вибрати контролер для СКУД (системи контролю керування доступом). Forter. URL: https://www.forter.com.ua/yak-vibrati-kontroler-dlya-skud-sistemi-kontrolyu-keruvannya-dostupom/ (Дата звернення: 03.03.2026).

3. Автономні контролери. Nadzor.ua. URL: https://nadzor.ua/uk/kontrol-dostupa/kontrollery/avtonomnye-kontrollery (Дата звернення: 03.03.2026).

4. Зчитувач Trinix TRK-200EI. Nadzor.ua. URL: https://nadzor.ua/uk/product/scityvatel-trinix-trk-200ei (Дата звернення: 05.03.2026).

5. Мережевий контролер Dahua DHI-ASC2202B-S. Nadzor.ua. URL: https://nadzor.ua/uk/product/setevoj-kontroller-dahua-dhi-asc2202b-s (Дата звернення: 05.03.2026).

6. Микитишин А. Г. Комп'ютерні мережі. Книга 1 : навчальний посібник / А. Г. Микитишин, М. М. Митник, П. Д. Стухляк. – Львів : Магнолія 2006, 2024. – 256 с. (Дата звернення: 05.03.2026).

7. Мережеві контролери. Nadzor.ua. URL: https://nadzor.ua/uk/kontrol-dostupa/kontrollery/setevye-kontrollery (Дата звернення: 03.03.2026).

8. Finkenzeller K. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication / Klaus Finkenzeller. – 3rd ed. – Chichester : John Wiley & Sons, 2010. – 480 с. (Дата звернення: 03.03.2026).

9. Котенко А. М. Системи контролю та управління доступом на об’єкти інформаційної діяльності : посібник / А. М. Котенко. – Київ : ДУІКТ, 2024. – 79 с. (Дата звернення: 03.03.2026).

10. Ключи Touch Memory. Worldvision. URL: https://worldvision.com.ua/kluchi-touch-memory/ (Дата звернення: 03.03.2026).

11. Як обрати ідеальний зчитувач для СКУД: код, картка чи біометрія.

Shop Security. URL: https://shop-security.com.ua/news/yak-obraty-idealnyj-zchytuvach-dlya-skud-kod-kartka-chy-biometriya (Дата звернення: 03.03.2026).

12. Зчитувач TRR-1103EW. Nadzor.ua. URL: https://nadzor.ua/uk/product/scityvatel-trr-1103ew (Дата звернення: 05.03.2026).

13. Fingerprint scanner. Wikipedia. URL: https://en.wikipedia.org/wiki/Fingerprint_scanner (Дата звернення: 04.03.2026).

14. Біометричні СКД. Elvis. URL: https://www.elvis.com.ua/kontrol-dostupu/biometrichni-skd/?ocf=F1S0V32 (Дата звернення: 04.03.2026).

15. Зчитувач Dahua DHI-ASR2102A. Nadzor.ua. URL: https://nadzor.ua/uk/product/scityvatel-dahua-dhi-asr2102a (Дата звернення: 05.03.2026).

16. Шапіро Л. Комп'ютерний зір / Л. Шапіро, Дж. Стокман ; пер. з англ. – Київ : Промінь, 2006. – 512 с. (Дата звернення: 04.03.2026).

17. Зчитувач Dahua DHI-ASR2102A. Nadzor.ua. URL: https://nadzor.ua/uk/product/scityvatel-dahua-dhi-asr2102a (Дата звернення: 05.03.2026).

18. Pricing – Face API. Microsoft Azure. URL: https://azure.microsoft.com/en-us/pricing/details/cognitive-services/face-api/ (Дата звернення: 06.03.2026).

19. About Nyckel. Nyckel. URL: https://www.nyckel.com/about/ (Дата звернення: 06.03.2026).

20. What is Machine Learning?. IBM. URL: https://www.ibm.com/think/topics/machine-learning (Дата звернення: 06.03.2026).

21. Pretrained Models. Nyckel. URL: https://www.nyckel.com/console/pretrained (Дата звернення: 06.03.2026).

22. Nyckel Moge.ai. URL: https://moge.ai/product/nyckel (Дата звернення: 06.03.2026).

23. Keras Getting Started. Nyckel Blog. URL: https://www.nyckel.com/blog/keras-getting-started/ (Дата звернення: 06.03.2026).

24. Nyckel Redefining AutoML. Nyckel Blog. URL: https://www.nyckel.com/blog/nyckel-redefining-automl/ (Дата звернення: 06.03.2026).

25. Image Classification Quickstart. Nyckel Docs. URL: https://www.nyckel.com/docs/image-classification-quickstart (Дата звернення: 09.03.2026).

26. Image Classification. Nyckel Blog. URL: https://www.nyckel.com/blog/image-classification/ (Дата звернення: 09.03.2026).

27. Nyckel Documentation. Nyckel Docs. URL: https://www.nyckel.com/docs (Дата звернення: 09.03.2026).

28. Таненбаум Е. Комп'ютерні мережі / Ендрю Таненбаум, Девід Уезерролл. – 5-те вид. – Київ : Діалектика, 2019. – 960 с. (Дата звернення: 09.03.2026).

29. Оліфер В. Г. Комп'ютерні мережі. Принципи, технології, протоколи : підручник / В. Г. Оліфер, Н. А. Оліфер. – 5-те вид. – Київ : БХВ, 2016. – 960 с. (Дата звернення: 09.03.2026)

30. tkinter – Python interface to Tcl/Tk. Python Documentation. URL: https://docs.python.org/uk/3.9/library/tkinter.html (Дата звернення: 10.03.2026).

31. Відладочна плата Arduino Nano Atmega328P Type-C. RoboStore. URL: https://robostore.com.ua/otladochnaia-plata-arduino-nano-atmega328p-type-c/ (Дата звернення: 26.03.2026).

32. Модуль RFID з карткою доступу для Arduino RC522. RoboStore. URL: https://robostore.com.ua/modul-rfid-z-kartkoiu-dostupu-dlia-arduino-rc522/ (Дата звернення: 26.03.2026).

33. Ультразвуковой датчик відстані HC-SR04. RoboStore. — URL: https://robostore.com.ua/moduli-i-datchiki/datchiki-zvuka/ultrazvukovoj-datchik-rasstoyaniya-hc-sr04/ (Дата звернення: 26.03.2026).

34. Authentication, Authorization, and Identification. QATestLab Blog. URL: https://training.qatestlab.com/blog/technical-articles/authentication-authorization-and-identification/ (Дата звернення: 26.03.2026).

35. Identification. Wikipedia. URL: https://en.wikipedia.org/wiki/Identification. — Дата звернення: 26.03.2026.

36. Authorization. Wikipedia. URL:

https://en.wikipedia.org/wiki/Authorization (Дата звернення: 26.03.2026).

37. Authentication. Wikipedia. URL: https://en.wikipedia.org/wiki/Authentication (Дата звернення: 26.03.2026).

38. Що таке COM-порт. GSM-Hub. URL: https://gsmhub.com.ua/glossary/com-port (Дата звернення: 28.03.2026).

39. Бенько Т. Г. Мікроконтролери : навчально-методичний посібник / Т. Г. Бенько. – Івано-Франківськ : Прикарпатський національний університет ім. Василя Стефаника, 2023. – 151 с. (Дата звернення: 28.03.2026).

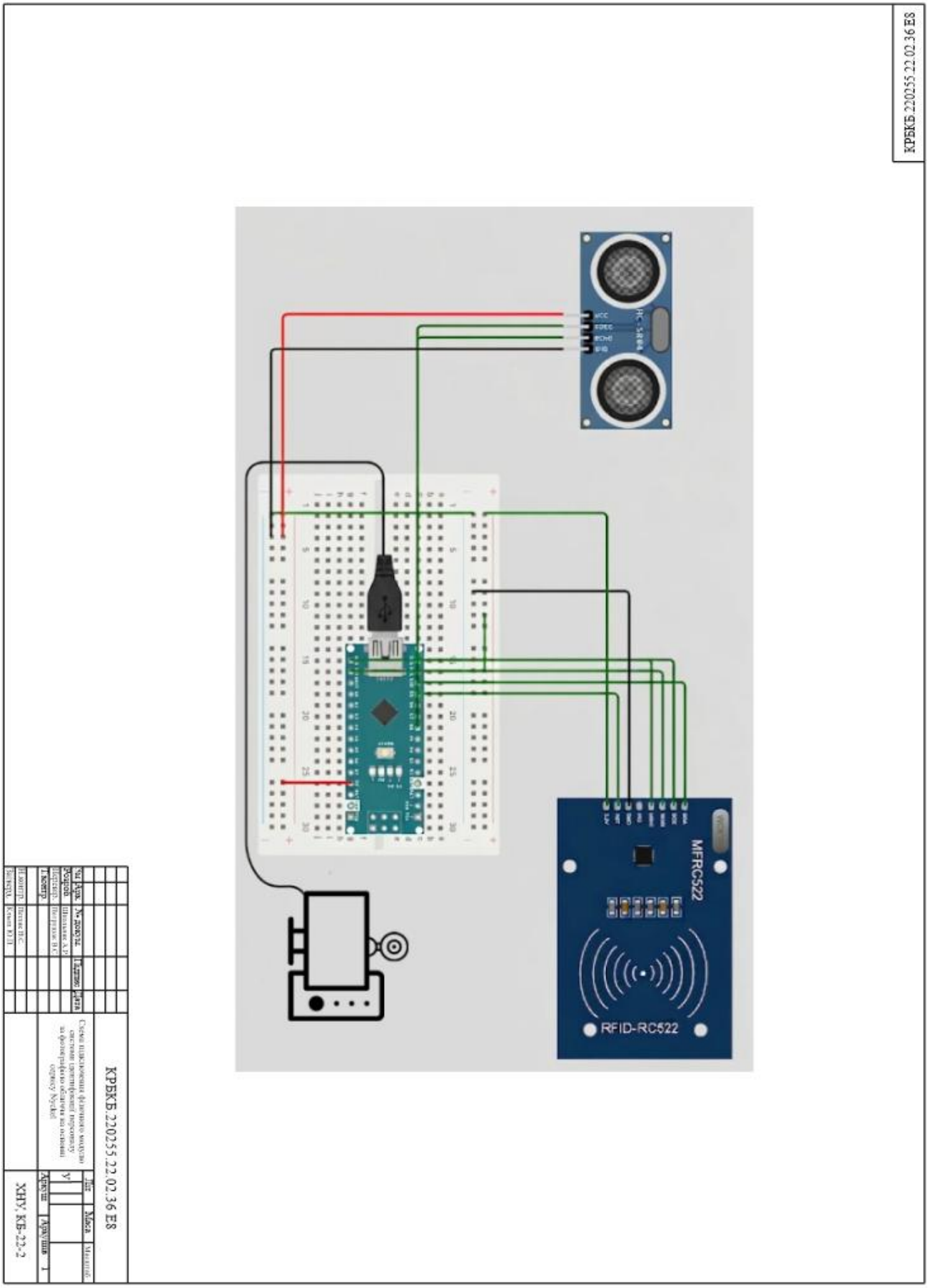
40. Лутц М. Програмування на Python. Том 1. Вступ до Python, основи мови, системне програмування, графічний інтерфейс / Марк Лутц. – 4-те вид. – Харків : Фоліо, 2020. – 992 с. (Дата звернення: 28.03.2026)

Додаток А

Копія графічної частини

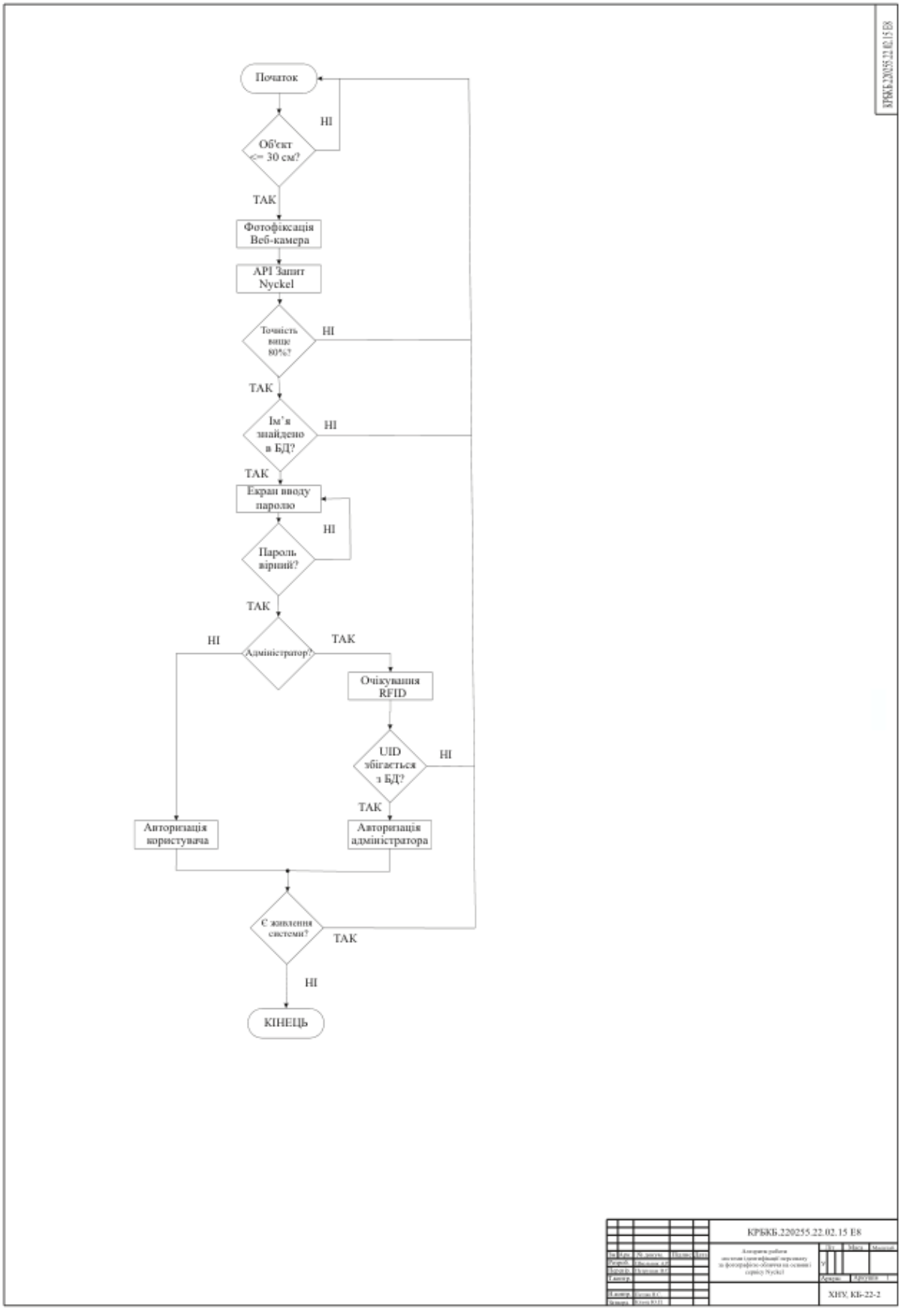
Додаток Б

Копія графічної частини



Додаток В

Копія графічної частини



Додаток Д

Код для фізичного модулю

```
#include <SPI.h>
#include <MFRC522.h>

#define TRIG_PIN 2
#define ECHO_PIN 3
#define RST_PIN 9
#define SS_PIN 10

MFRC522 mfrc522(SS_PIN, RST_PIN);

unsigned long lastDetectTime = 0;
unsigned long lastDistPrintTime = 0;
const int detectCooldown = 5000;

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  SPI.begin();
  mfrc522.PCD_Init();
}

void loop() {
  checkDistance();
  checkRFID();
  delay(50);
}
```

```
void checkDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long duration = pulseIn(ECHO_PIN, HIGH);
  long distance = duration * 0.034 / 2;

  if (distance > 0 && distance < 400) {
    if (millis() - lastDistPrintTime > 500) {
      Serial.print("DIST:");
      Serial.println(distance);
      lastDistPrintTime = millis();
    }
  }

  if (distance > 0 && distance <= 30) {
    if (millis() - lastDetectTime > detectCooldown) {
      Serial.println("PERSON_DETECTED");
      lastDetectTime = millis();
    }
  }
}

void checkRFID() {
   if (!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial()) {
    return;
  }
```

```
  Serial.print("RFID:");
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
  }
  Serial.println();
  mfrc522.PICC_HaltA();
}
```

Додаток Ж

Код програмного додатку

```
import sqlite3
import datetime
import customtkinter as ctk
from tkinter import messagebox
import serial
import threading
import cv2
import requests
import time

COM_PORT = 'COM7'
BAUD_RATE = 9600

NYCKEL_CLIENT_ID = "a902k4daceetn9z3kh5nsdqm3e1be4z6"
NYCKEL_CLIENT_SECRET =
"hdqu5cnmvk02sgg1dkshxtlxidy521v8f8c9i5y7tis7couny0ytch9ostbrdhkt"
NYCKEL_FUNCTION_ID = "8rknftc5tcmmlo9x"

def init_db():
    conn = sqlite3.connect('company_system.db')
    cursor = conn.cursor()

    cursor.execute('''
        CREATE TABLE IF NOT EXISTS users (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            username TEXT UNIQUE NOT NULL,
            password TEXT NOT NULL,
            full_name TEXT NOT NULL,
            role TEXT NOT NULL,
            department TEXT,
            rfid_uid TEXT
        )
    ''')
    cursor.execute('CREATE TABLE IF NOT EXISTS logs (id INTEGER PRIMARY
KEY AUTOINCREMENT, timestamp DATETIME DEFAULT
CURRENT_TIMESTAMP, username TEXT, action TEXT)')
    cursor.execute('CREATE TABLE IF NOT EXISTS accounting_data (id
INTEGER PRIMARY KEY, document_name TEXT)')
    cursor.execute('CREATE TABLE IF NOT EXISTS hr_data (id INTEGER
PRIMARY KEY, document_name TEXT)')
    cursor.execute('CREATE TABLE IF NOT EXISTS procurement_data (id
INTEGER PRIMARY KEY, document_name TEXT)')
```

```
    cursor.execute('SELECT COUNT(*) FROM users')
    if cursor.fetchone()[0] == 0:
        cursor.execute("INSERT INTO users (username, password, full_name, role,
department, rfid_uid) VALUES ('admin', 'admin', 'Школьняк А.', 'SUPER_ADMIN',
'ALL', ' 65 B1 44 00')")
        cursor.execute("INSERT INTO users (username, password, full_name, role,
department, rfid_uid) VALUES ('mfox', '1234', 'Меган Фокс', 'USER', 'HR',
NULL)")
        cursor.execute("INSERT INTO users (username, password, full_name, role,
department, rfid_uid) VALUES ('bpitt', '1234', 'Бред Пітт', 'USER',
'ACCOUNTING', NULL)")
        cursor.execute("INSERT INTO users (username, password, full_name, role,
department, rfid_uid) VALUES ('jwick', '1234', 'Джон Уік', 'USER',
'PROCUREMENT', NULL)")

        cursor.execute("INSERT INTO hr_data (document_name) VALUES
('Графік_відпусток.docx')")
        cursor.execute("INSERT INTO accounting_data (document_name) VALUES
('Фінансовий_звіт_2026.xlsx')")
        cursor.execute("INSERT INTO procurement_data (document_name) VALUES
('Договори.pdf')")
    conn.commit()
    conn.close()

class AccessControlApp(ctk.CTk):
    def __init__(self):
        super().__init__()
        self.title("Програма дипломна Школьняк А.")
        self.geometry("850x550")
        ctk.set_appearance_mode("dark")
        ctk.set_default_color_theme("blue")

        self.conn = sqlite3.connect('company_system.db')
        self.cursor = self.conn.cursor()
        self.current_user = None
        self.current_frame = None
        self.identified_username = None
        self.waiting_for_rfid = False
        self.serial_port = None
        self.connect_arduino()

        self.show_waiting_screen()

    def connect_arduino(self):
```

```
        try:
            self.serial_port = serial.Serial(COM_PORT, BAUD_RATE, timeout=1)
            threading.Thread(target=self.read_serial_loop, daemon=True).start()
            print("[+] Arduino підключено.")
        except Exception as e:
            print(f"[-] Помилка підключення Arduino (перевірте COM_PORT): {e}")

    def read_serial_loop(self):
        while True:
            if self.serial_port and self.serial_port.in_waiting > 0:
                try:
                    line = self.serial_port.readline().decode('utf-8').strip()
                    if line.startswith("DIST:"):
                        distance = line.split(":")[1]
                        print(f"[Сенсор] Поточна відстань: {distance} см")
                    elif line == "PERSON_DETECTED":
                        print("\n[ТРИГЕР] Людина в зоні камери! Запуск
розпізнавання...")
                        if self.identified_username is None and not self.waiting_for_rfid:
                            self.after(0, self.start_face_recognition)
                        else:
                            print("[-] Система зайнята іншим користувачем. Чекаємо.")
                    elif line.startswith("RFID:"):
                        uid = line.split(":")[1]
                        self.after(0, lambda: self.verify_rfid(uid))

                except Exception as e:
                    print(f"[-] Помилка читання порту: {e}")

            time.sleep(0.05)

    def log_action(self, username, action):
        self.cursor.execute("INSERT INTO logs (username, action) VALUES (?, ?)",
(username, action))
        self.conn.commit()

    def clear_screen(self):
        if self.current_frame is not None:
            self.current_frame.destroy()

    def logout(self):
        if self.current_user:
            self.log_action(self.current_user['username'], "Вихід із системи")
        self.current_user = None
        self.identified_username = None
```

```
        self.waiting_for_rfid = False
        self.show_waiting_screen()


    def show_waiting_screen(self):
        self.clear_screen()
        self.current_frame = ctk.CTkFrame(self)
        self.current_frame.pack(pady=50, padx=50, fill="both", expand=True)

        self.status_label = ctk.CTkLabel(self.current_frame, text="Очікування
співробітника...\nПідійдіть до датчика", font=("Roboto", 24, "bold"))
        self.status_label.pack(pady=100)

        btn_manual = ctk.CTkButton(self.current_frame, text="[Тест] Імітувати
виявлення", fg_color="gray", command=self.start_face_recognition)
        btn_manual.pack(side="bottom", pady=20)

    def start_face_recognition(self):
        self.status_label.configure(text="Виявлено рух!\nРобимо знімок...")
        self.update()

        cap = cv2.VideoCapture(0)
        ret, frame = cap.read()
        cap.release()

        if ret:
            self.status_label.configure(text="Фото зроблено.\nВідправка до Nyckel
API...")
            self.update()
            recognized_user = self.recognize_face_via_nyckel(frame)

            if recognized_user:
                self.identified_username = recognized_user
                self.show_password_screen()
            else:
                self.status_label.configure(text="Обличчя не розпізнано (або точність <
80%).\nВідійдіть і спробуйте знову.")
                self.update()
                self.after(3000, self.show_waiting_screen)
        else:
            self.status_label.configure(text="Помилка камери.")
    def recognize_face_via_nyckel(self, frame):
        try:
            token_resp = requests.post(
                "https://www.nyckel.com/connect/token",
```

```
            data={
                "client_id": NYCKEL_CLIENT_ID,
                "client_secret": NYCKEL_CLIENT_SECRET,
                "grant_type": "client_credentials"
            }
        )
        token_resp.raise_for_status()
        access_token = token_resp.json()["access_token"]
        success, buffer = cv2.imencode('.jpg', frame)
        if not success:
            print("[-] Помилка кодування зображення в пам'яті")
            return None

        image_bytes = buffer.tobytes()

        resp = requests.post(

f"https://www.nyckel.com/v1/functions/{NYCKEL_FUNCTION_ID}/invoke",
            headers={"Authorization": f"Bearer {access_token}"},
            files={"data": ("image.jpg", image_bytes, "image/jpeg")}
        )
        resp.raise_for_status()
        result = resp.json()

        predicted_user = result.get('labelName')
        confidence = result.get('confidence', 0.0)

        print(f"[Nyckel] Нейромережа розпізнала: {predicted_user} з точністю
{confidence*100:.1f}%")

        if confidence > 0.80:
            return predicted_user
        else:
            print(f"[-] Відмова. Точність ({confidence*100:.1f}%) не перевищує
заданий поріг.")
            return None

    except requests.exceptions.RequestException as e:
        print(f"Помилка: {e}")
        return None
    except Exception as e:
        print(f"Неочікувана помилка: {e}")
        return None

def show_password_screen(self):
```

```
        nyckel_name = self.identified_username
        print(f"[DEBUG] Nyckel розпізнав: '{nyckel_name}'. Шукаємо в базі за полем full_name...")

        self.clear_screen()
        self.current_frame = ctk.CTkFrame(self)
        self.current_frame.pack(pady=50, padx=50, fill="both", expand=True)
        self.cursor.execute("SELECT username, full_name, role FROM users WHERE full_name=?", (nyckel_name,))
        user_data = self.cursor.fetchone()

        if not user_data:
            print(f"[-] ПОМИЛКА: Співробітника з ім'ям '{nyckel_name}' немає в базі даних!")

            error_msg = f"Помилка ідентифікації.\nNyckel розпізнав вас як: '{nyckel_name}',\nале такого імені немає в базі."
            label = ctk.CTkLabel(self.current_frame, text=error_msg, font=("Roboto", 20, "bold"), text_color="red")
            label.pack(pady=100)

            self.identified_username = None
            self.after(5000, self.show_waiting_screen)
            return

        db_username, full_name, role = user_data
        self.identified_username = db_username

        label = ctk.CTkLabel(self.current_frame, text=f"Розпізнано: {full_name}\nВведіть пароль", font=("Roboto", 24, "bold"))
        label.pack(pady=30)

        self.entry_password = ctk.CTkEntry(self.current_frame, placeholder_text="Пароль", show="*", width=250)
        self.entry_password.pack(pady=10)

        btn_login = ctk.CTkButton(self.current_frame, text="Підтвердити", width=250, command=self.process_password)
        btn_login.pack(pady=20)

        self.error_label = ctk.CTkLabel(self.current_frame, text="", text_color="red")
        self.error_label.pack()

    def process_password(self):
        password = self.entry_password.get()
```

```
        self.cursor.execute("SELECT id, username, full_name, role, department,
rfid_uid FROM users WHERE username=? AND password=?",
                            (self.identified_username, password))
        user = self.cursor.fetchone()

        if user:
            self.current_user = {
                'id': user[0], 'username': user[1], 'full_name': user[2],
                'role': user[3], 'department': user[4], 'rfid_uid': user[5]
            }

            if self.current_user['rfid_uid']:
                self.show_rfid_screen()
            else:
                self.log_action(self.current_user['username'], "Успішний вхід (FaceID +
Pass)")
                if self.current_user['role'] == 'SUPER_ADMIN':
                    self.show_super_admin_screen()
                else:
                    self.show_user_screen()
        else:
            self.log_action(self.identified_username, "Невдала спроба входу (невірний
пароль)")
            self.error_label.configure(text="Невірний пароль")


    def show_rfid_screen(self):
        self.clear_screen()
        self.waiting_for_rfid = True
        self.current_frame = ctk.CTkFrame(self)
        self.current_frame.pack(pady=50, padx=50, fill="both", expand=True)

        label = ctk.CTkLabel(self.current_frame,
text="АДМІНІСТРАТОР\nПрикладіть картку доступу (RFID)", font=("Roboto",
24, "bold"), text_color="orange")
        label.pack(pady=50)

        btn_cancel = ctk.CTkButton(self.current_frame, text="Скасувати",
fg_color="gray", command=self.logout)
        btn_cancel.pack(pady=20)

        btn_manual = ctk.CTkButton(self.current_frame, text="[Тест] Імітувати RFID
адміна", fg_color="darkred",
                                   command=lambda: self.verify_rfid(" DE AD BE EF"))
```

```
        btn_manual.pack(side="bottom", pady=20)

    def verify_rfid(self, scanned_uid):
        if not self.waiting_for_rfid: return

        expected_uid = self.current_user['rfid_uid']

        if scanned_uid == expected_uid:
            self.waiting_for_rfid = False
            self.log_action(self.current_user['username'], "Успішний вхід (FaceID +
Pass + RFID)")
            self.show_super_admin_screen()
        else:
            self.log_action(self.current_user['username'], f"Помилка RFID. Відмова в
доступі. UID: {scanned_uid}")
            messagebox.showerror("Доступ заборонено", "Невірна картка доступу!")
            self.logout()

    def show_super_admin_screen(self):
        self.clear_screen()
        self.current_frame = ctk.CTkFrame(self)
        self.current_frame.pack(fill="both", expand=True)

        menu_frame = ctk.CTkFrame(self.current_frame, height=50, corner_radius=0)
        menu_frame.pack(fill="x", side="top")

        lbl_user = ctk.CTkLabel(menu_frame, text=f"Адмін:
{self.current_user['full_name']}", font=("Roboto", 14, "bold"))
        lbl_user.pack(side="left", padx=20, pady=10)

        btn_logs = ctk.CTkButton(menu_frame, text="Журнал подій",
command=self.render_logs_view)
        btn_logs.pack(side="left", padx=10)

        btn_users = ctk.CTkButton(menu_frame, text="Користувачі",
command=self.render_users_view)
        btn_users.pack(side="left", padx=10)

        btn_logout = ctk.CTkButton(menu_frame, text="Вийти", fg_color="red",
width=100, command=self.logout)
        btn_logout.pack(side="right", padx=50)

        self.content_frame = ctk.CTkFrame(self.current_frame)
        self.content_frame.pack(fill="both", expand=True, padx=20, pady=20)
        self.render_logs_view()
```

```
    def render_logs_view(self):
        for widget in self.content_frame.winfo_children(): widget.destroy()
        ctk.CTkLabel(self.content_frame, text="Повний журнал подій",
font=("Roboto", 18, "bold")).pack(pady=10)
        log_textbox = ctk.CTkTextbox(self.content_frame, width=750, height=350)
        log_textbox.pack(pady=10)
        self.cursor.execute("SELECT timestamp, username, action FROM logs ORDER
BY timestamp DESC")
        for row in self.cursor.fetchall(): log_textbox.insert("end", f"[{row[0]}]
{row[1]}: {row[2]}\n")
        log_textbox.configure(state="disabled")

    def render_users_view(self):
        for widget in self.content_frame.winfo_children(): widget.destroy()
        ctk.CTkLabel(self.content_frame, text="Управління користувачами",
font=("Roboto", 18, "bold")).pack(pady=10)
        scroll_frame = ctk.CTkScrollableFrame(self.content_frame, width=750,
height=350)
        scroll_frame.pack(fill="both", expand=True)
        self.cursor.execute("SELECT username, full_name, role, department FROM
users")
        for user in self.cursor.fetchall():
            u_name, f_name, role, dept = user
            row_frame = ctk.CTkFrame(scroll_frame)
            row_frame.pack(fill="x", pady=5, padx=5)
            ctk.CTkLabel(row_frame, text=f"{f_name} ({u_name}) | Роль: {role}",
width=300, anchor="w").pack(side="left", padx=10, pady=10)
            combo = ctk.CTkComboBox(row_frame, values=["ACCOUNTING", "HR",
"PROCUREMENT", "ALL", "None"], width=150)
            combo.set(dept if dept else "None")
            combo.pack(side="left", padx=10)
            ctk.CTkButton(row_frame, text="Зберегти", width=100, command=lambda
u=u_name, c=combo: self.update_user_dept(u, c.get())).pack(side="left", padx=5)
            if u_name != self.current_user['username']:
                ctk.CTkButton(row_frame, text="Видалити", fg_color="darkred",
width=90, command=lambda u=u_name: self.delete_user(u)).pack(side="right",
padx=10)

    def update_user_dept(self, target_username, new_dept):
        if new_dept == "None": new_dept = None
        self.cursor.execute("UPDATE users SET department=? WHERE username=?",
(new_dept, target_username))
        self.conn.commit()
        messagebox.showinfo("Успіх", "Оновлено!")
```

```
    def delete_user(self, target_username):
        if messagebox.askyesno("Підтвердження", "Видалити?"):
            self.cursor.execute("DELETE FROM users WHERE username=?",
(target_username,))
            self.conn.commit()
            self.render_users_view()

    def show_user_screen(self):
        self.clear_screen()
        self.current_frame = ctk.CTkFrame(self)
        self.current_frame.pack(pady=20, padx=20, fill="both", expand=True)
        dept = self.current_user['department']
        ctk.CTkLabel(self.current_frame, text=f"Робоча область: {dept} |
{self.current_user['full_name']}", font=("Roboto", 20, "bold")).pack(pady=10)
        btn_logout = ctk.CTkButton(self.current_frame, text="Вийти", fg_color="red",
command=self.logout)
        btn_logout.place(relx=0.85, rely=0.02)
        files_textbox = ctk.CTkTextbox(self.current_frame, width=750, height=300)
        files_textbox.pack(pady=20)
        try:
            self.cursor.execute(f"SELECT document_name FROM {dept.lower()}_data")
            for row in self.cursor.fetchall(): files_textbox.insert("end", f"  {row[0]}\n")
        except: files_textbox.insert("end", "[-] Немає доступу.")
        files_textbox.configure(state="disabled")

if __name__ == "__main__":
    init_db()
    app = AccessControlApp()
    app.mainloop()
```