Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Пояснювальна записка

до магістерської кваліфікаційної роботи
магістр
(освітньо-кваліфікаційний рівень)

на тему: Розробка методу та засобів системи ідентифікації користувачів

Виконав: студент II курсу
групи 2ПІ-18 м
спеціальності
121 – Інженерія програмного забезпечення
(шифр і назва напряму підготовки, спеціальності)
Мисько Ю.О.
(прізвище та ініціали)
Керівник: к.т.н., доц. каф. ПЗ Черноволик Г. О.
(прізвище та ініціали)
Рецензент: к.т.н., доц. каф. КН Арсенюк І. Р.
(прізвище та ініціали)

ВНТУ – 2019

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Освітньо-кваліфікаційний рівень – магістр
Спеціальність 121 – Інженерія програмного забезпечення

УЗГОДЖЕНО
Директор ТОВ «Екзістек»
Феферман О. Д. _____
«____» _____2019 року

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
_____ Романюк О. Н.
«____» _____2019 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Мисько Юлії Олегівні

1. Тема роботи — розробка методу та засобів системи ідентифікації користувачів.

Керівник роботи: Черноволик Галина Олександрівна, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від "__"_____2019 року №___

2. Строк подання студентом роботи
_____

3. Вихідні дані до роботи: Операційна система — Linux; середовище розробки — Android Studio; мови програмування — Java, Kotlin; смартфон — з ОС Android версії 6.0 і вище, зі сканером відбитків пальців та підтримкою технології Wi-Fi.

4. Зміст розрахунково-пояснювальної записки: аналіз методів ідентифікації, аналіз методів біометричної ідентифікації, аналіз аналогів, розробка методу вибору ідентифікації користувача за відбитками пальців через смартфон, розробка методики розмежування доступу на основі отриманих біометричних даних, проектування бази даних користувачів, розробка інструкції користувача, тестування роботи програмного додатку, економічна частина.

5. Перелік графічного матеріалу: тема, автор, науковий керівник магістерської кваліфікаційної роботи; мета, об'єкт та предмет дослідження; аналіз предметної області; порівняльний аналіз аналогів; структура програмного додатку; алгоритм програмного додатку; обґрунтування вибору програмного середовища розробки; обґрунтування вибору мови програмування; результати роботи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Черноволик Г. О., к.т.н., доцент кафедри ПЗ		
4	Бальзан М.В., к.е.н., доцент кафедри ЕПВМ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області та постановка задач дослідження		Вик.
2	Розробка оптимального методу із використанням біометричної ідентифікації за відбитками пальців для розв'язку поставлених задач		Вик.
3	Програмна реалізація та тестування програмного засобу ідентифікації користувачів		Вик.
4	Економічна частина		Вик.

Студент _____ **Мисько Ю.О.**

(підпис) (прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи _____ **Черноволик Г.О.**

(підпис) (прізвище та ініціали)

АНОТАЦІЯ

У магістерській кваліфікаційній роботі розглянуто сучасні методи біометричної ідентифікації користувачів комп’ютерних систем, призначені для забезпечення захисту конфіденційної інформації. Встановлено недоліки та переваги кожного методу, наведено показники якості ідентифікації та обґрунтовано перспективні напрями досліджень. Для захисту інформаційних комп’ютерних систем від небажаного доступу з боку неавторизованих користувачів у роботі запропоновано систему контролю та управління доступом, в якій його розмежування досягається шляхом ідентифікації користувача за відбитками пальців через смартфон.

Ключові слова: ідентифікація користувачів, біометрична ідентифікація, відбитки пальців, сканер відбитків пальців у смартфоні, розмежування доступу.

ABSTRACT

The master's qualification examines the modern methods of biometric identification of users of computer systems, designed to protect the confidential information. The disadvantages and advantages of each method are identified, the indicators of quality of identification are given and the perspective directions of researches are grounded. To protect computer information systems from unauthorized access by unauthorized users, a work is offered in the system of access control and management, which is achieved by identifying the user by fingerprint through a smartphone.

Keywords: user identification, biometric identification, fingerprints, smartphone fingerprint scanner, access control.

ЗМІСТ

ВСТУП

Обґрунтування вибору теми дослідження. Із появою та еволюцією нових інформаційних технологій з’явилася проблема інформаційної безпеки, зв'язана із потребою безпечного збереження і конфіденційності даних, що обробляється та зберігається в комп’ютерних системах. Врегулюванню цих проблем приділяється все більша увага, удосконалюються реалізовані методи захисту інформаційних систем, постійно розробляються нові методи, що дають змогу збільшувати надійність і стійкість систем, призначених для вирішення такого типу задач.

Завдання інформаційної безпеки набуває ще більшої значущості через зростання злочинності в сфері використання комп’ютерної інформації.

Доступ користувачів до різних класів інформації повинен визначатися ідентифікацією, тобто процесом розпізнавання параметрів, що однозначно визначають особу користувача. Останнім часом все більше набувають популярності системи на основі біометричних методів розмежування та контролю доступу. Сформувався специфічний ринок біометричних пристроїв і відповідних програмних продуктів, оскільки вони дозволять вирішувати важливі завдання в області інформаційної безпеки.

Необхідність ідентифікації особистості людини зумовлена активною інформатизацією сучасного суспільства та збільшенням потоків конфіденційної інформації. Аналіз сучасних систем контролю доступу свідчить про очевидний рух у бік біометричних методів завдяки їх зручності, надійності та достовірності. Порівняно з технологіями розпізнавання за сітківкою і райдужною оболонкою ока [17-18] сканування відбитків пальців є дешевшим і зручнішим; порівняно з технологіями розпізнавання за рукописним і клавіатурним почерком, технології ідентифікації за відбитками пальців на кілька порядків кращі за статистичними показниками помилок першого і другого роду; порівняно з технологіями розпізнавання за «геометрією» обличчя [16] – відбитки пальців [13-15] не змінюються, отже це не погіршує

статистичну надійність методу, крім того геометричний метод потребує дуже дорогого обладнання, а також зміна міміки обличчя і перешкоди на ньому погіршують статистичну надійність методу.

Отже, дослідження у напрямку захисту інформації шляхом ідентифікації відбитків пальців через смартфон є актуальним.

Оскільки комп'ютерні системи тепер прямо вбудовані в інформаційні структури сучасного суспільства, засоби захисту мають враховувати сучасні форми представлення інформації (системи захисту повинні забезпечувати безпеку на рівні інформаційних ресурсів, а не відокремлених документів, файлів чи повідомлень). Для того, щоб бути запитуваним сучасним ринком інформаційних систем, засоби безпеки не повинні конфліктувати з існуючими додатками та сформованими технологіями обробки даних, а, навпаки, мають стати невіддільною частиною цих засобів і технологій.

Відсутність у багатьох керівників підприємств і компаній чіткого уявлення з питань захисту інформації призводить до того, що їм складно повною мірою оцінити необхідність створення надійної системи захисту інформації на своєму підприємстві і тим більше складно буває визначити конкретні дії, необхідні для захисту тих чи інших конфіденційних відомостей.

Впровадження біометрії полегшує діяльність працівників усіх зазначених груп, одночасно забезпечуючи істотне зростання рівня інформаційної безпеки.

Розроблювана система біометричної ідентифікації буде досить ефективною, оскільки виконуватиме наступні функції:

- уніфікація процесів доступу до інформаційних ресурсів на основі єдиного ідентифікатора — відбитка пальця;
- захист інформаційних ресурсів від несанкціонованого доступу і/або доступу, здійснюваного з порушенням встановлених правил;
- автоматизація та централізація управління користувачами, їх обліковими записами та відповідними повноваженнями в операційних системах і різних прикладних продуктах;

- прискорення доступу легальних користувачів до ресурсів корпоративної мережі із забезпеченням максимальної простоти та прозорості цього процесу;
- скорочення непродуктивних втрат, викликаних помилками при введенні логіна/пароля, блокуванням облікових записів і т.п.;
- оптимізація діяльності системних адміністраторів за рахунок зменшення числа звернень, пов'язаних із помилками користувачів при ідентифікації, аутентифікації й авторизації;
- швидка реєстрація нових користувачів і оперативна зміна облікових записів співробітників.

Усі наявні продукти на ринку, аналогічні цьому, мають свої недоліки. Ідеального не існує, тому є необхідність розробки нових програмних засобів ідентифікації користувача в захищеній комп'ютерній системі з подальшою авторизацією.

Враховуючи усі проблеми та незручності, із якими зіштовхуються користувачі комп'ютерних систем, адміністратори і, як наслідок, самі підприємці, які несуть чималі втрати, у зв'язку з незахищеністю системи буде розроблено систему ідентифікації користувачів, яка вирішить їх.

Отже, розроблювана система поєднуватиме у собі зручність, захищеність, контроль, мобільність, зрозумілість, швидкість, економність.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Метою роботи є підвищення ефективності захисту конфіденційних даних у комп'ютерних системах від несанкціонованих дій шляхом розмежування доступу за рахунок ідентифікації користувачів за відбитками пальців через смартфон, а також пошук компромісу між надійністю, доступною ціною та зручністю у використанні й адмініструванні засобів ідентифікації та аутентифікації.

Основними завданнями дослідження є:

- визначити сутність та значення проблеми захисту інформації в процесі ідентифікації користувачів;

- проаналізувати недоліки та переваги існуючих підходів до ідентифікації користувачів;

- обґрунтувати використання підходу ідентифікації користувачів за відбитками пальців через смартфон як оптимальний;

- описати принцип роботи розроблених засобів системи;

- розробити метод ідентифікації користувачів за відбитками пальців через смартфон для авторизації в комп'ютерній системі;

- розробити методику розмежування доступу на основі отриманих біометричних даних;

- спроектувати базу даних користувачів;

- розробити інструкцію користувача.

Об'єкт дослідження — процес захисту інформації в комп'ютерних системах.

Предмет дослідження — методи ідентифікації користувача.

Методи дослідження. У процесі дослідження застосовувалися: методи теорії алгоритмів і прикладної теорії інформації для розробки алгоритмів і програмного забезпечення; комп'ютерне моделювання для аналізу та перевірки достовірності отриманих теоретичних положень; метод виділення зображення відбитків пальців для подальшого використання при ідентифікації користувачів.

Наукова новизна отриманих результатів.

1) Подальшого розвитку набув метод ідентифікації користувачів за відбитками, що відрізняється від існуючих подальшим розмежуванням доступу до комп'ютерної системи та дозволяє підвищити ефективність системи захисту даних і розширити її функціональні можливості.

2) Розроблено методику розмежування доступу, яка відрізняється від існуючих дистанційним блокуванням/розблокуванням захищених комп'ютерних систем, що дозволяє зменшити час входу в систему.

Практична цінність одержаних результатів. Практичне значення результатів дослідження полягає у розробленні програмного модулю ідентифікації користувача через смартфон із подальшою авторизацією та розмежуванням доступу у захищеній комп'ютерній системі.

Впровадження. Результати досліджень використовуються в компанії «ТОВ Екзістек» для підвищення захищеності конфіденційної інформації в комп'ютерних системах, для більшої мобільності, спрощеного контролю доступу та обліку робочого часу працівників.

Особистий внесок здобувача. Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто.

Апробація. Результати роботи було представлено на щорічній конференції НТКП ВНТУ (м. Вінниця, 2018) та на Міжнародній науково-практичній Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ» (9-10 грудня 2019 року).

Публікації. Основні результати досліджень було опубліковано у тезах доповіді [35-36].

Структура та обсяг роботи. Магістерська кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури, що містить 36 найменувань, 4 додатки. Робота містить 53 ілюстрації, 11 таблиць.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

Одна з найбільш актуальних задач розвитку інформаційних технологій на сучасному етапі — забезпечення надійного захисту інформації. Важливою, але ще не повністю вирішеною проблемою захисту інформації є ефективна процедура ідентифікації користувача, який отримує доступ до конфіденційної інформації.

1.1 Ідентифікація в системах захисту інформації та поняття розмежування доступу

Ідентифікація — це процес розпізнавання користувача у системі за наперед визначеним ідентифікатором або іншою попередньою інформацією про нього, що приймається системою [1].

Ідентифікація об’єкта − це встановлення особистості користувача. Це необхідно для того, щоб система в подальшому змогла прийняти рішення щодо видачі людині дозволу для роботи на комп’ютері, доступу до закритої інформації тощо. Тому, ідентифікація є одним із основних понять в інформаційній безпеці. Ідентифікація дозволяє суб’єкту (користувачу, процесу), що діють від імені визначеного користувача або іншого апаратно-програмного компоненту назвати себе (повідомити своє ім’я). За допомогою аутентифікації інша сторона пересвідчується, що суб’єкт дійсно той, за кого він себе видає [1].

Надійна ідентифікація ускладнена не лише мережевими загрозами, але й цілим рядом інших причин. По-перше, майже про всі складові автентифікації можна дізнатися, вкрасти їх або підробити. По-друге, є протиставлення між надійністю аутентифікації та зручностями користувача і системного адміністратора. Тому, із міркувань безпеки, користувачу потрібно з певною частотою повторно вводити аутентифікаційну інформацію (оскільки його робоче місце могла зайняти інша людина), а це не тільки клопітно, але й

підвищує ризик того, що хтось може підглянути за процесом введенням даних. По-третє, чим надійнішим є засіб захисту, тим він і дорожчий.

Сучасні засоби ідентифікації/аутентифікації мають підтримувати концепцію єдиного входу у мережу. А він, у першу чергу, вимога зручності для споживачів. Якщо в корпоративній мережі велика кількість інформаційних сервісів, що допускають незалежне звернення, то багатократна ідентифікація/аутентифікація стає занадто обтяжливою. Не можна запевняти, що єдиний вхід у мережу став нормою, тому метою магістерської роботи є пошук компромісу між надійністю, доступною ціною та зручністю у використанні й адмініструванні засобів ідентифікації й аутентифікації [2].

Система ідентифікації є одним із ключових елементів інфраструктури захисту від несанкціонованого доступу до будь-якої інформаційної системи.

Під несанкціонованим доступом до інформації розуміється доступ до інформації, що порушує встановлені правила розмежування доступу і здійснюваний з використанням штатних засобів обчислювальної техніки або автоматизованих систем. НСД може носити випадковий або навмисний характер.

Задачею систем ідентифікації є визначення та верифікація набору повноважень суб’єкта при доступі до інформаційної системи.

Ідентифікація та аутентифікація нерозривно зв’язані між собою, оскільки спосіб перевірки визначає, яким чином і що користувач повинен пред’явити системі, щоб отримати доступ.

На теперішній час існує декілька способів ідентифікації користувачів (рис. 1.1). У кожного з них є свої переваги та недоліки, завдяки чому деякі технології підходять для використання в одних комп’ютерних системах, решта − в інших. Однак у багатьох випадках немає строго певного рішення. А тому як розробникам програмного забезпечення, так і користувачам доводиться самостійно вирішувати, який спосіб ідентифікації реалізовувати у власних інформаційних комп’ютерних системах.

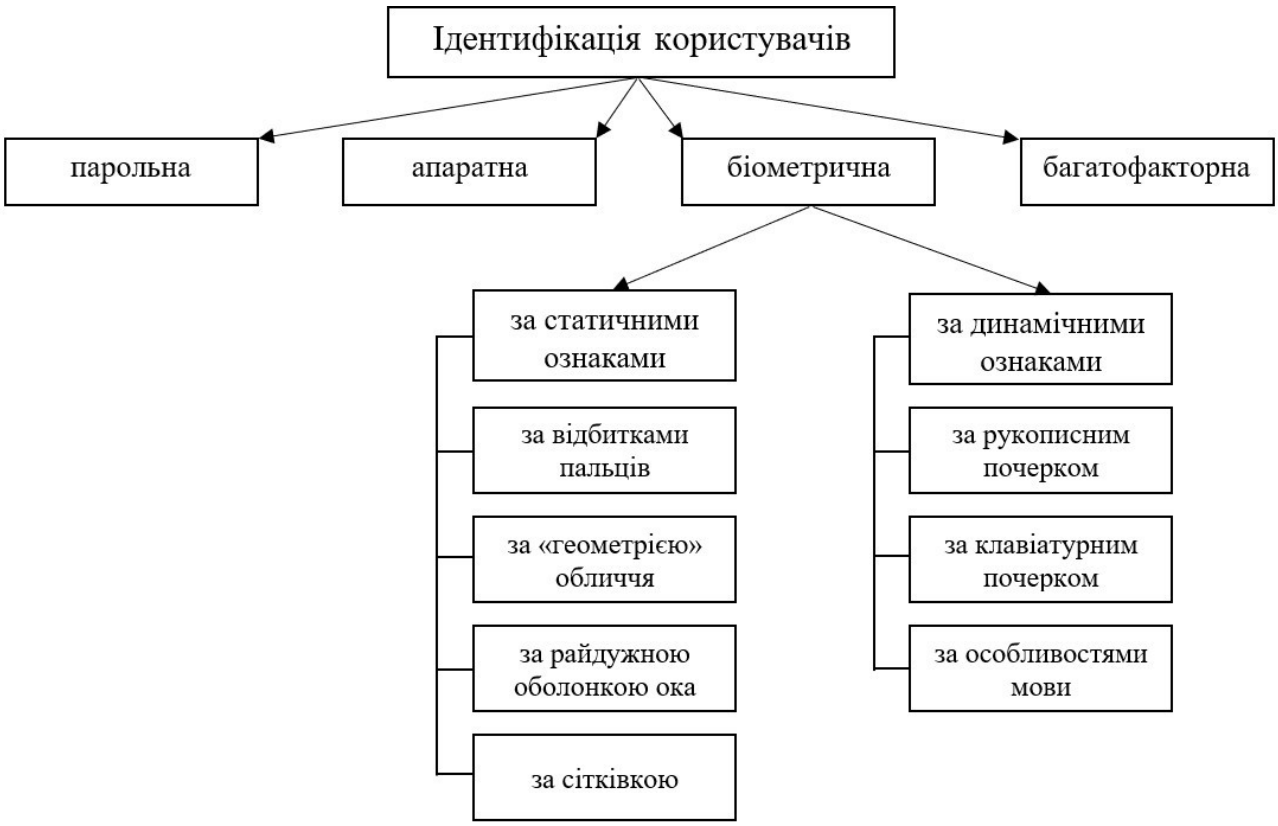


Рисунок 1.1 – Схематичне зображення способів ідентифікації користувачів

У зв’язку зі збільшенням загроз для комп’ютерної інформації, все більше уваги приділяється задачам вдосконалення існуючих та розробці нових засобів захисту інформаційних комп’ютерних систем від небажаного доступу з боку неавторизованих користувачів. Одним із напрямків для досягнення цієї мети є системи контролю та управління доступом, в яких розмежування доступу досягається ідентифікацією користувачів, тобто процесом розпізнавання параметрів, що однозначно визначають користувача [3].

У будь-якій системі аутентифікації як правило можна відзначити декілька основних елементів (рис. 1.2).

Коректність рішення розпізнавання та перевірки дійсності впливає на те, чи буде надано конкретному користувачеві доступ до ресурсів системи, тобто чи буде він авторизований.

Рисунок 1.2 – Складові системи аутентифікації

Авторизація являє собою процедуру наділення суб’єкта певними правами доступу до ресурсів системи після успішного проходження ним процедури ідентифікації. Для кожного суб’єкта в системі визначається набір прав, які він може використовувати при зверненні до її ресурсів.

Процес керування доступом суб’єктів до ресурсів системи називається адмініструванням. У розробленій системі цей процес містить такі складові, зображені на рисунку 1.3.

Логічне керування доступом особливо вагоме на рівні операційної системи.

Активне використання комп’ютерних систем для розв’язання широкого діапазону практичних завдань потребує підвищеної уваги з боку розробників відносно питань інформаційної безпеки. Однією з точок зору, стосовно цього питання, є проблема розмежування доступу до інформаційних і функціональних ресурсів для забезпечення безпеки.

Рисунок 1.3 – Елементи процесу адміністрування

Система розмежування доступу контролює доступ суб’єктів до об’єктів. Об’єкт доступу — це елемент ОС, доступ до якого перевіряється і може бути обмежений. Важливе значення має метод доступу — дія, яку здійснює суб’єкт над об’єктом. Варто розрізняти терміни «методу» та «право доступу». Право доступу — це право виконувати доступ до об’єкта з використанням деякого методу чи їх групи. Суб’єкти можуть мати певні привілеї — право доступу, що використовує деякий метод доступу, який надається певному суб’єкту на всі об’єкти ОС, що підтримують цей метод [4].

В ОС об’єктами бувають не лише об’єкт-користувачі, але й процеси та потоки, суб’єктом же виступає суперпозиція перерахованих елементів. По факту, в усіх захищених системах, процес здійснення доступу відбувається від імені певного користувача.

Головними об’єктами, які вимагають гарантії безпеки та керуються операційною системою є ті, що містять інформацію, що знаходиться в процесі оброблення та зберігання.

До типових привілеїв, які підтримує ОС, належать такі: привілей власника об'єкта (у разі довірчого керування доступом власник завжди може змінити права доступу до об'єкта), привілей адміністратора (у різних ОС він

може мати доволі різний рівень — аж до повних і необмежених прав), привілей здійснювати доступ до системи тощо.

Політику керування доступом на рівні ОС, як правило, реалізують втіленням дискреційного керування доступом. Оскільки об’єктів доступу в операційній системі дуже багато, застосувати в ній матрицю доступу в чистому вигляді неможливо. Натомість в ОС використовують списки керування доступом, які пов'язують з кожним об'єктом доступу. У деяких ОС реалізовано розширення дискреційного керування доступом, за якого звичайні правила дискреційної політики певним чином модифікуються: по-перше, можливість доступу однозначно визначається не трійкою користувач-об'єкт-метод, а четвіркою користувач-процес-об'єкт-метод, а по-друге, для кожного суб'єкта визначається список програм, які він може запускати. Середовище з такою політикою керування доступом називають ізольованим (або замкненим) програмним середовищем [4].

У спеціально призначених для побудови ІКС операційних системах, які використовують для оброблення конфіденційної інформації, окрім дискреційного керування доступом, є ще й мандатне керування доступом. Системи розмежування доступу в таких ОС реалізують певний набір моделей, переважно модель Белла — ЛаПадула та її розширення.

Варто відзначити важливу особливість систем ідентифікації з розмежуванням доступу. У більшості їх реалізацій права доступу суб'єкта до об'єкта перевіряються лише в момент відкривання об'єкта для доступу з використанням певного методу або набору методів. Після того, як права доступу успішно перевірено, створюються умови для доступу суб'єкта до об'єкта. Після цього суб'єкт вільно працює з об'єктом протягом визначеного часу або допоки його не буде закрито.

Але є системи, в яких права доступу перевіряються щоразу, коли здійснюється спроба доступу. Необхідність контролювати кожну спробу доступу пов'язана з тим, що можливість доступу визначається тим, в якому стані знаходиться процес у поточний момент. Стан процесу може бути змінено

під час його функціонування залежно від того, наприклад, до яких об'єктів і за якими методами цим процесом уже було здійснено доступ.

Загальні ПРД мають бути конкретизовані на рівні вибору необхідних функціональних послуг захисту (профіль захищеності) та впровадження організаційних заходів захисту. Частиною ПРД є керування доступом (КД) (рис. 1.4).

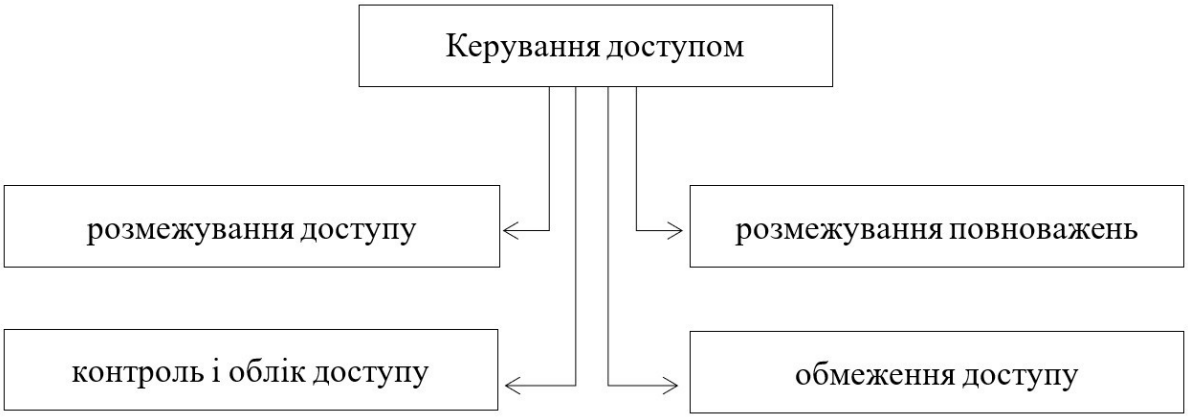


Рисунок 1.4 — Елементи системи керування доступом

В ІС повинно бути реалізоване адміністративне керування доступом. Тільки адміністратори СЗІ мають право включати і вилучати користувачів та об'єкти.

У програмному засобі встановлюються наступні обмеження на роботу користувачів:

- обмеження періоду часу, в ході якого користувач може входити в мережу;
- визначення адрес робочих станцій, з якими дозволено входити в мережу;
- обмеження кількості спроб входу в мережу з неправильним паролем.

Розмежування доступу полягає в організації доступу до інформації користувачів відповідно до їхніх функціональних особливостей і повноважень.

У даний час існує безліч різних підходів, що визначають можливості учасників інформаційної взаємодії в доступі до тих чи інших видів даних (ресурсів).

Водночас основними з них є:

- технології систем дискреційного розмежування доступу;

- технології систем мандатного розмежування доступу;

- технології рольового розмежування доступу;

- суб'єктно-орієнтована технологія ізольованого програмного середовища.

Кожен із цих напрямків є описом набору правил, на підставі аналізу яких приймається рішення про доступ. Разом із тим, реалізація механізмів розмежування доступу до ресурсів кожної конкретної комп'ютерної системи, як правило, ґрунтується на більш складних, ніж базові, технологіях, які враховують окремі особливості такої системи, середовища її оточення та положення політики її інформаційної безпеки. У кожному з цих випадків базові моделі або їх комбінації деталізуються цілим рядом додаткових обмежень і правил.

Рольове розмежування доступу є складовою багатьох сучасних комп'ютерних систем. Як правило, рольове розмежування доступу застосовується в системах захисту СУБД або в елементах мережевих операційних систем. В основі всіх математичних моделей цього напряму лежить базова модель рольового розмежування доступу, яка визначає найзагальніші принципи побудови ролей. Основними елементами базової моделі рольового розмежування доступу є: множина користувачів $\{U\}$, множина ролей $\{R\}$, множина прав доступу на об'єкти комп'ютерної системи $\{P\}$, множина сесій користувача $\{S\}$, функція, що визначає для кожної ролі множину прав доступу p PA: R 2 →, при цьому для кожного p ∈ існує r ∈ така, при якій p PA(r) ∈ , функція, що визначає для кожного користувача множину ролей, на які він може бути авторизований, а саме p UA: U 2 → і т.п. [5]. У базовій моделі рольового розмежування доступу існує низка обмежень, які дозволяють її використовувати в реальних комп'ютерних системах. Так, у базовій моделі рольового розмежування доступу відсутні механізми, що дозволяють одній сесії активізувати іншу (усі сесії активізуються користувачем). Ще одним важливим механізмом цього напряму моделювання є обмеження, що накладаються на множину ролей, на які може бути

авторизований користувач або на які він авторизується в період однієї сесії. Однією з відмінних особливостей цього напряму моделювання стала побудова системи адміністрування рольового розмежування доступу.

1.2 Класифікація методів ідентифікації

У даний час існують три основні підходи до ідентифікації користувачів інформаційних систем:

1) користувачу відоме щось, що він може повідомити системі і що дозволяє однозначно його ідентифікувати (наприклад, пароль, PIN-код, ключ і т.п.);

2) користувач може виконати деяку унікальну процедуру (наприклад, використати всілякі карти, магнітні брелоки і т.п.);

3) вимірювання і використовування унікальних характеристик користувача (біометричних).

Найбільш поширені в даний час методи ідентифікації засновані на використанні паролів. Недоліки цього підходу добре відомі, адже пароль може бути скомпрометований багатьма способами.

Методи, які відносяться до другого підходу, також досить поширені. Фізичні об'єкти (носії інформації) можуть бути втрачені, вкрадені, передані іншій особі, дубльовані. З цим зв'язані основні недоліки методів даного класу.

Методи, які використовують для ідентифікації унікальні характеристики користувача (біометричні методи), вільні від перерахованих недоліків, тому є найперспективнішими, крім того, вони активно розвиваються останнім часом. Перевага біометричних систем ідентифікації, в порівнянні з традиційними, полягає в тому, що ідентифікується не зовнішній предмет, що належить людині, а сама людина.

Біометричні характеристики є невід’ємною частиною людини, оскільки їх неможливо загубити, передати іншій людині чи забути, а підроблювання будь-якої з них досить складне і, як наслідок, коштовне.

Парольна ідентифікація. Парольна ідентифікація найбільш проста як у реалізації, так і у використанні. Її суть зводиться до того, що кожен зареєстрований користувач системи має набір персональних реквізитів (зазвичай використовуються пари: логін-пароль). Далі, при кожній спробі входу користувач повинен вказати цю інформацію. Оскільки вона унікальна для кожного користувача, то на підставі її система й робить висновок про особистість та ідентифікує.

Головною перевагою парольної ідентифікації є відносна простота і звичність (але «відносна», оскільки для більшого захисту пароль повинен бути достатньо складним і об’ємним, а при частому блокуванні програми чи системи втрачається на його введення багато часу та й у цілому цей процес набридає активним користувачам). Паролі давно вбудовані в операційні системи та інші сервіси. При правильному використанні вони можуть забезпечити прийнятний для багатьох організацій рівень безпеки. Проте, по сукупності характеристик їх варто визнати найслабшим засобом перевірки достовірності. Саме слабкий рівень парольного захисту є однією з основних причин уразливості комп'ютерних систем до спроб несанкціонованого доступу [6].

Щодо недоліків, то, на жаль, їх багато. І, мабуть, найголовніший − величезна залежність надійності ідентифікації від самих користувачів, а точніше, від обраних ними паролів. Справа в тому, що більшість людей використовують ненадійні ключові слова, які легко підбираються. До них відносяться занадто короткі паролі, загальновідомі сполучення символів і т.п.

Для детальнішого розгляду принципів побудови парольних систем варто зрозуміти декілька основних визначень.

Ідентифікатор користувача − деяка унікальна кількість інформації, що дозволяє розрізняти індивідуальних користувачів парольної системи. Часто ідентифікатор також називають ім'ям користувача або ім'ям облікового запису користувача.

Пароль користувача − деяка секретна кількість інформації, відома лише користувачу та парольній системі, яку може запам'ятати користувач і

пред'явити для проходження процедури аутентифікації. Одноразовий пароль дає можливість користувачу однократно пройти аутентифікацію. Багаторазовий пароль може бути використаний для перевірки достовірності повторно.

Обліковий запис користувача – сукупність його ідентифікатора та пароля.

База даних користувачів парольної системи містить облікові записи всіх користувачів даної парольної системи.

Парольна система є «переднім краєм оборони» всієї системи безпеки. Деякі її елементи (зокрема ті, що реалізують інтерфейс користувача) можуть бути розташовані в місцях, відкритих для доступу потенційному зловмиснику. Тому парольна система стає одним із перших об'єктів атаки при вторгненні зловмисника в захищену систему [6].

Основні типи загроз безпеки парольних систем наведено у таблиці 1.1.

Таблиця 1.1 — Типи загроз безпеки парольних систем

Розголошування параметрів облікового запису	підбір в інтерактивному режимі
	підглядання
	навмисна передача пароля його власником іншій особі
	захоплення бази даних парольної системи
	перехоплення переданої по мережі інформації про пароль
	зберігання пароля в доступному місці

Продовження таблиці 1.1

Втручання у функціонування компонентів парольної системи	впровадження програмних закладок
	виявлення і використовування помилок, допущених на стадії розробки
	виведення з ладу парольної системи

Деякі з перерахованих типів загроз пов'язані з наявністю так званого людського фактору, що виявляється в тому, що користувач може:

- використовувати стандартний пароль, вказаний у документації, не змінюючи його;

- вибрати пароль, який легко запам'ятати і також легко підібрати;

- записати пароль, який складно запам'ятати, і покласти запис в доступному місці;

- ввести пароль так, що його зможуть побачити сторонні;

- передати пароль іншій особі навмисно або під впливом помилки.

На додаток до вище сказаного необхідно наголосити на існуванні «парадоксу людського фактору». Полягає він у тому, що користувач нерідко прагне виступати скоріш супротивником парольної системи, як, втім, і будь-якої системи безпеки, функціонування якої впливає на його робочі умови, ніж союзником системи захисту, тим самим послаблюючи її.

У більшості випадків аутентифікація відбувається в розподілених системах і пов'язана з передачею по мережі інформації про параметри облікових записів користувачів. Якщо інформація, що передається по мережі в процесі аутентифікації, не захищена належним чином, виникає загроза її перехоплення зловмисником і використання для порушення захисту парольної системи.

Апаратна ідентифікація. Цей принцип ідентифікації ґрунтується на визначенні особистості користувача за певним предметом, ключем, що перебуває в його ексклюзивному користуванні. Мова йде про спеціальні електронні ключі. На даний момент найбільше поширення одержали два типи пристроїв. До першого відносяться всілякі карти. Їх досить багато, і працюють

вони за різними принципами. Так, наприклад, досить зручні у використанні безконтактні карти, що дають змогу користувачам проходити ідентифікацію як у комп’ютерних системах, так й у системах доступу в приміщення. Найбільш надійними вважаються смарт-карти — аналоги звичних багатьом людям банківських карт. Крім того, є й більш дешеві, але менш стійкі до злому карти: магнітні, зі штрих-кодом і т.п. [7].

Іншим типом ключів, які можуть використовуватися для апаратної ідентифікації, є так звані токени. Ці пристрої мають власну захищену пам’ять і підключаються безпосередньо до одного з портів комп’ютера (USB, LPT).

eToken – персональний засіб аутентифікації і зберігання даних, що апаратно підтримує роботу з цифровими сертифікатами і електронними цифровими підписами (ЕЦП). eToken може бути виконаний у вигляді USB-ключа або стандартної смарткарти. eToken підтримує роботу й інтегрується зі всіма основними системами та додатками, що використовують технології смарт-карт або PKI (Public Key lnfrastructure).

Основне призначення:

- двофакторна аутентифікація користувачів при доступі до захищених ресурсів (комп'ютерів, мереж, додатків);

- безпечне зберігання закритих ключів цифрових сертифікатів, профілів користувачів, налаштувань додатків і т.п. в незалежній пам'яті ключа;

- апаратне виконання криптографічних операцій в довіреному середовищі (генерація ключів шифрування, симетричне й асиметричне шифрування, розрахунок хеш-функції, формування ЕЦП).

Головною перевагою застосування апаратної ідентифікації є досить висока надійність. У пам’яті токенів можуть зберігатися ключі, підібрати які досить складно. Крім того, у них реалізовано чимало різних захисних механізмів. А вбудований мікропроцесор дозволяє електронному ключу не лише приймати участь у процесі ідентифікації користувача, але й виконувати деякі інші корисні функції. Кожен апаратний (електронний) ідентифікатор є

фізичним пристроєм, зазвичай невеликих розмірів для зручності його носіння з собою [7].

Найбільш серйозною небезпекою у випадку використання апаратної ідентифікації є можливість крадіжки зловмисниками токенів або карт у зареєстрованих користувачів. Також вони можуть бути втрачені, передані іншій особі, дубльовані. Другий мінус розглянутої технології – ціна. Взагалі, останнім часом вартість як самих електронних ключів, так і програмного забезпечення, що може працювати з ними, дещо знизилася. Проте, для введення в експлуатацію системи такої ідентифікації однаково будуть потрібні неабиякі вкладення, оскільки кожного зареєстрованого користувача потрібно забезпечити персональними токенами. Крім того, згодом деякі типи ключів можуть зношуватися, або бути загублені й т.п. Тобто апаратна ідентифікація вимагає значних експлуатаційних витрат.

Біометрична ідентифікація. Біометрична ідентифікація – це спосіб ідентифікації особи за окремими специфічними біометричними ознаками (ідентифікаторами), властивими конкретній людині. Сучасний рівень розвитку комп'ютерних технологій дає змогу використовувати подібні ознаки як основу для ідентифікації людини й ухвалення рішення про можливість або неможливість доступу до інформаційних комп’ютерних систем.

Усі біометричні системи працюють практично за однаковою схемою. По-перше, система запам'ятовує зразок біометричної характеристики (це називається процесом запису). Під час запису деякі біометричні системи можуть попросити зробити декілька зразків для того, аби скласти найточніше зображення біометричної характеристики. Потім одержана інформація обробляється та перетворюється в математичний код. Крім того, система може попросити провести ще деякі дії для того, щоб «приписати» біометричний зразок до певної людини. В такому разі, знову робиться зразок біометричної характеристики і порівнюється з представленим зразком [8].

Ідентифікація по будь-якій біометричній системі складається з чотирьох стадій:

1) запис − фізичний або поведінковий зразок, що запам'ятовується системою;

2) виділення − унікальна інформація витягується із отриманого зразка і складається біометричний зразок;

3) порівняння − збережений зразок порівнюється з представленим;

4) збіг/неспівпадання − система визначає, чи співпадають біометричні зразки, і виносить рішення.

На сьогоднішній день експлуатується більше десяти видів біометричних ознак і цей напрямок продовжує дуже активно розвиватися. Тому абсолютно зрозумілим є рішення використовувати їх в області інформаційних технологій.

Головною перевагою біометричних технологій є найвища надійність. І дійсно, відомо, що двох людей з однаковими відбитками пальців у природі просто не існує. Сучасні пристрої дуже стійкі по відношенню щодо будь-якої фальсифікації, тому зловмисникам доводиться вигадувати все нові й нові способи обману біометричних сканерів. Зручність використання з мінімальними затратами часу — ще одна з переваг цього способу [9].

Основний недолік біометричної ідентифікації заключається в тому, що є ймовірність помилкового реагування системи: зареєстрованим користувачам забороняється доступ до ресурсів системи, але це не суттєво, оскільки головна мета — не допустити до користування системою осіб, що не повинні мати до неї доступ.

Багатофакторна ідентифікація. Досі було розглянуто три види (або підходи) однофакторної ідентифікації користувачів інформаційних систем. Тобто в розглянутих системах для визначення особи користувача використовувався лише один фактор. Проте, поступово все більшого поширення одержує багатофакторна ідентифікація, коли для визначення особистості застосовується відразу декілька параметрів. Причому комбінуватися вони можуть у довільному порядку. Втім, у переважній

більшості випадків, використовується тільки одна пара: парольний захист і токен. У цьому випадку користувач може не боятися підбору його пароля зловмисником (без електронного ключа він працювати не буде), а також крадіжки токена (він не буде працювати без пароля). Але, у деяких системах застосовуються максимально надійні процедури ідентифікації. У них одночасно використовуються паролі, токени й біометричні характеристики людини [10].

До переваг багатофакторної ідентифікації можна віднести її здатність захистити інформацію, як від внутрішніх загроз, так і від зовнішніх вторгнень. Певною слабкістю можна вважати необхідність використання додаткових програмно-апаратних комплексів, пристроїв зберігання та зчитування даних.

Більшість продуктів із функцією багатофакторної автентифікації вимагають від користувача клієнтське програмне забезпечення, для того, щоб система багатофакторної аутентифікації запрацювала. Деякі розробники створили окремі настановні пакети для входу в мережу, ідентифікаційних даних веб-доступу VPN-підключення. Щоб використовувати з цими продуктами токен або смарт-карту, потрібно встановити на PC чотири або п'ять пакетів спеціального програмного забезпечення. Це можуть бути пакети, які використовуються для здійснення контролю версії або це можуть бути пакети для перевірки конфліктів з бізнес-додатками. Якщо доступ може бути проведений з використанням веб-сторінок, то тоді можна обійтися без непередбачених витрат. З іншими програмними рішеннями багатофакторної аутентифікації, такими як «віртуальні» токени або деякі апаратні токени, жодне не може бути встановлено безпосередніми користувачами [10].

Існують різні форми реалізації багатофакторної ідентифікації, оскільки вона не стандартизована. Отже, проблема полягає в її здатності до взаємодії. Існує багато процесів і аспектів, які необхідно враховувати при виборі, розробці, тестуванні, впровадженні та підтримці цілісної системи управління ідентифікацією безпеки, включаючи всі релевантні механізми аутентифікації і супутніх технологій.

Багатофакторна аутентифікація має ряд недоліків, які перешкоджають її поширенню. Зокрема людині, яка не розуміється конкретно в цій області, складно стежити за розвитком апаратних токенів або USB-штекерів. Багато користувачів не можуть самостійно встановити сертифіковане програмне забезпечення, оскільки не володіють відповідними технічними навичками. Загалом, багатофакторні рішення вимагають додаткових витрат на встановлення та оплату експлуатаційних витрат. Більшість апаратних комплексів, заснованих на токенах, запатентовані, тому деякі розробники стягують з користувачів щорічну плату. З точки зору логістики, розмістити апаратні токени важко, оскільки вони можуть бути пошкоджені або втрачені. Крім витрат на встановлення багатофакторної аутентифікації значну суму також становить оплата технічного обслуговування.

Якщо ж багатофакторна ідентифікація здійснюється через мобільний пристрій ряд недоліків значно зростає:

- мобільний телефон повинен завжди бути у мережі під час аутентифікації, інакше повідомлення з паролем просто не дійде.
- користувач поширює дані мобільного номеру, які можуть використовуватись для розсилки спаму та реклами;
- текстові повідомлення (SMS), які, потрапляючи на мобільний телефон, можуть бути перехоплені;
- текстові повідомлення приходять з деякою затримкою, оскільки деякий час йде на перевірку.

Сучасні смартфони використовуються як для одержання пошти, так і для отримання SMS. Як правило, електронна пошта на мобільному телефоні завжди увімкнена. Таким чином, усі акаунти, для яких пошта є ключем, можуть бути зламані (перший фактор), мобільний пристрій (другий фактор) — смартфон змішує два чинника в один.

1.3 Класифікація методів біометричної ідентифікації

Серед біометричних механізмів ідентифікації можна виділити такі [11]:

1) по статичних ознаках − те, що практично не змінюється з часом, починаючи від народження людини (фізіологічні характеристики);

2) по динамічних ознаках − поведінкові характеристики, тобто ті, що побудовані на особливостях, характерних для підсвідомих рухів у процесі відтворення якої-небудь дії. Динамічні ознаки можуть змінюватися з часом, але не різко, а поступово.

Серед статичних методів ідентифікації користувачів найвідомішими є:

1. Ідентифікація за відбитками пальців. В основу цього методу покладена унікальність малюнка папілярних візерунків на пальцях. Ідентифікація побудована таким чином: за допомогою сканера одержують зображення відбитку, потім це зображення по складному алгоритму перетворюється на спеціальний цифровий код. Далі цей код порівнюється з еталонними кодами, які зберігаються в базі даних.

2. Ідентифікація по сітківці ока. У даному випадку сканується малюнок кровоносних судин очного дна. Зрозуміло, що цей малюнок спостерігається лише за певних умов, тому при скануванні людина дивиться на видалене світлове джерело і спеціальна камера сканує його очне дно.

3. Ідентифікація по райдужній оболонці ока. Малюнок райдужної оболонки ока − унікальний для кожної людини. В цьому методі важлива не лише спеціальна камера, але й надійне програмне забезпечення. Адже саме за допомогою програмного забезпечення із зображення виділяється потрібний малюнок.

4. Ідентифікація за формою кисті руки. Цей метод ґрунтується на розпізнаванні геометричних особливостей кисті руки. Спеціальний сканер формує тривимірний малюнок кисті. При аналізі цього малюнка виконуються вимірювання, за допомогою яких формується відповідний цифровий код.

5. Ідентифікація за формою обличчя. Цей метод дещо схожий до попереднього. Тут так само будується тривимірний образ обличчя. Спеціальне програмне забезпечення виділяє з цього образу контури очей, губ та інших частин обличчя. Далі проводяться точні вимірювання між отриманими контурами. Саме за цими даними будується цифровий код.

Серед динамічних методів можна назвати такі:

1. Ідентифікація по голосу. В даний час існує безліч програм для розпізнавання голосу. У цьому методі ідентифікації важливі частотні характеристики голосу людини. Саме по них і будується цифрова модель.

2. Ідентифікація по почерку. При ідентифікації цим методом зазвичай досліджується підпис людини. Перевіряються такі динамічні характеристики, як: графічні параметри, сила натиску на поверхню, швидкість нанесення підпису. За цими характеристиками і будується цифровий код.

3. Ідентифікація за клавіатурним почерком. Даний метод аналогічний ідентифікації по почерку, але замість того, щоб ставити автограф, людині необхідно надрукувати кодове слово. Цифровий код будується по динаміці набору певного слова.

Основні характеристики перерахованих вище методів біометричної ідентифікації наведено у таблиці 1.2.

Таблиця 1.2 — Основні характеристики методів біометричної ідентифікації

Метод отримання біометричних параметрів	Ймовірність відмови у доступі, %	Ймовірність помилкової ідентифікації «чужого» (без використання муляжу), %	Ймовірність помилкової ідентифікації «чужого» (з використанням муляжу), %	Збереження таємниці образу у процесі ідентифікації абонента	Вартість технічної реалізації в грошовому еквіваленті, у.о.
Геометрична будова руки	0,2…4	0,2….1	10…75	неможливо приховати	Від 600 до 3000
Відбитки пальців	2...6	0,0001	10...70	неможливо приховати	Від 60 до 600

Продовження таблиці 1.2

Особливості малюнка сітківки ока	0,4	6...10	_____	неможливо приховати	приблизно 4000
Райдужна оболонка ока	0,2…2	0,0001	_____	неможливо приховати	Від 500 до 6000
Портрет обличчя	1…9	_____	_____	неможливо приховати	55000
Рукописний почерк	0,5…5	0,5…5	0,5...5	8-10…10-40	_____
Клавіатурний та комп'ютерний почерк	3…9	3...9	_____	6-10…10-12	_____
Характеристики і особливості мови	0,5...5	0,5...5	25...90 (запис)	10-16…10-30	1...60

При всьому теоретичному різноманітті можливих біометричних методів, тих, що застосовуються на практиці для ідентифікації користувачів комп'ютерних систем серед них небагато. Основних методів чотири – розпізнавання за відбитками пальців, за зображенням обличчя (двомірним або тривимірним), за райдужною оболонкою та сітківкою ока.

Існують два статистичні показники, що визначають якість, точність біометричних технологій [12]:

FAR (False Acceptance Rate) – вірогідність помилкового розпізнавання, тобто вірогідність того, що система визнає «чужого» за «свого».

FRR (False Rejection Rate) – вірогідність помилкового нерозпізнавання, тобто того, що система не розпізнає знайомого їй суб'єкта. Будь-яку біометричну систему можна налаштувати на різний ступінь «пильності», тобто на різне значення вірогідності помилкового розпізнавання FAR. При цьому, чим нижчий FAR, тобто чим пильніша система, тим вища вірогідність помилкового нерозпізнавання FRR (система менш чутлива). Ідеальні характеристики системи – це такі показники помилки та відмови ідентифікації,

коли одночасно при великій надійності ідентифікації (помилка 0,0001%) досягається відмова ідентифікації всього долі відсотка.

Розпізнавання за відбитками пальців. На отриманому зі сканера зображенні відбитків пальців (залежно від якості) можна виділити деякі характерні ознаки, які надалі можна використовувати в цілях ідентифікації.

На найпростішому технічному рівні, наприклад, якщо роздільна здатність отриманого зі сканеру зображення складає 300 − 500 dpi, на поверхні зображення пальця можна виділити досить велику кількість дрібних деталей, за допомогою яких можна їх класифікувати, але, як правило, в системах ідентифікації використовують всього два типи деталей візерунку (особливих точок) [13]:

- кінцеві точки − точки, в яких «виразно» закінчуються папілярні лінії;
- точки розгалуження − точки в яких папілярні лінії роздвоюються.

На зображенні поверхні пальця з роздільною здатністю близько 1000 dpi можна виявити деталі внутрішньої будови самих папілярних ліній, зокрема, пори потових залоз. Їх розташування можна використовувати для ідентифікації. Проте цей метод мало поширений внаслідок складності здобуття в не лабораторних умовах зображень такої якості.

При автоматизованому розпізнаванні відбитків пальців, на відміну від традиційної дактилоскопії, виникає значно менше проблем, пов'язаних із різними зовнішніми чинниками, що впливають на процес розпізнавання. Якість отриманого зі сканера зображення папілярного візерунку пальця є одним із основних критеріїв, від якого залежить вибір алгоритму формування згортки відбитку пальця і, зрештою, ідентифікації людини.

Сканування відбитків пальців − це найстаріша методика з усіх існуючих, але водночас вона вважається однією з найперспективніших. Кожна людина має унікальні, незмінні відбитки пальців, що доведено криміналістичною наукою та підтверджено експертною практикою. Відбиток пальця умовно складається з рельєфних ліній − папілярного візерунка, будова якого зумовлена рядами гребінцевих виступів шкіри, поділених борозенками. Ці лінії

утворюють складні шкіряні зображення – дуги, петлі, завитки, яким у цілому притаманні такі властивості, як: індивідуальність, стійкість, відновлюваність [14].

Процес розпізнавання наведено на рисунку 1.5.

Рисунок 1.5 – Процес дактилоскопічного розпізнавання

Біометричні системи ідентифікації особи за відбитком пальця в цілому мають дуже низькі коефіцієнти відмови в доступі до об'єкта (система не буде розпізнавати достовірність відбитків пальців руки зареєстрованого користувача), за деякої можливості помилкового або підробленого доступу до об'єкта (можливість того, що система помилково «ідентифікує» відбиток пальця користувача, не зареєстрованого в даній системі) [14].

У даний час активно розробляються алгоритми, стійкі до «шуму» в зображеннях відбитків пальця, що дозволяє досягти збільшення точності й швидкості розпізнання об'єктів у реальному часі.

Переваги та недоліки методу

Переваги методу. Висока достовірність – статистичні показники методу краще показників способів ідентифікації за обличчям, голосом, розписом. Низька вартість пристроїв, які сканують зображення відбитка пальця. Досить проста процедура сканування відбитку. Оскільки смартфони із вбудованими сканерами відбитків пальців стали дуже розповсюдженими, то їх власникам не потрібно купляти дороге додаткове обладнання, достатньо лише встановити розроблене програмне забезпечення. Мінімальна втрата часу на проведення ідентифікації користувачів [15].

Недоліки: папілярний візерунок відбитку пальця дуже легко пошкоджується дрібними подряпинами, порізами. Більшість сканерів погано відносяться до сухої шкіри і не пропускають людей похилого віку.

Розпізнавання за «геометрією» обличчя. В основі технології – створення двомірних або тримірних «карт» людських рис – система запам'ятовує й опізнає контури носу та губ, форму брів, відстань між окремими рисами.

Розробники систем біометричного аналізу – компанія Biolink, називають розпізнання за обличчям другою за поширеністю та популярністю біометричною технологією, але «упізнання» за геометрією обличчя – задача трудомістка, адже на сприйняття машини впливає освітлення, кут нахилу голови, наявність макіяжу.

Процес відеосканування обличчя засновано на створенні «живого» шаблона оригіналу в реальному часі та порівнянні його з файлом шаблона-зразка. Ступінь достовірності об'єктів зображення, що потребують перевірки, являє собою певний рівень ймовірності, який може бути не однаковим для різних типів задач, або ж бути залежним від персоналу, ПК, часу й інших факторів. При розпізнаванні обличь із великої відстані є суттєва залежність між якістю та результатом ідентифікації.

Існують три основні методи розпізнавання обличчя. Вони включають аналіз зображень з метою встановлення відмінних характеристик обличчя [16]:

- аналіз «відмінних рис обличчя» – найрозповсюдженіший, адаптований до змін міміки;

- аналіз на основі «нейронних мереж» – побудований на порівнянні «особливих точок», здатних ідентифікувати обличчя у важких умовах;

- метод «автоматичної обробки зображення обличчя» – визначення відстані та відношення відстані між установленими особливими рисами обличчя людини.

Останній метод не такий ефективний як інші, але він може бути використаний для ідентифікації зображення в погано освітлених приміщеннях.

Основними проблемними питаннями, які суттєво впливають на ефективність роботи зазначених біометричних систем є зміна освітлення, різні варіації положення обличчя під час руху, складність виділення інформативно-значимої частини обличчя, у тому числі обличчя-портрета, та несприятливий фон, який ускладнює ідентифікацію обличчя. Частково ці питання вдається вирішити шляхом автоматичного виділення на обличчі особливих точок та вимірювання відстані між ними. Таким чином, на обличчі виділяються контури очей, брів, носа, підборіддя, вух. Відстані між характерними точками цих контурів утворюють своєрідний і компактний еталон конкретної особи, який легко піддається масштабному вимірюванню.

Система розпізнавання обличчя ділиться на два напрямки: 2-D розпізнавання і 3-D розпізнавання. У кожного з них є переваги та недоліки, проте багато що залежить ще й від області застосування і вимог, пред'явлених до конкретного алгоритму.

2-D розпізнавання особи − один із самих статистично неефективних методів біометрії (наведено на рисунку 1.6). З'явився він досить давно і застосовувався, в основному, в криміналістиці, що і сприяло його розвитку. Унаслідок з'явилися комп'ютерні інтерпретації методу, тому він і став більш надійним, але поступається іншим біометричним методам ідентифікації особистості. В даний час, через погані статистичні показники він застосовується в мультимодальній або, як її ще називають, перехресній біометрії, або в соціальних мережах [16].

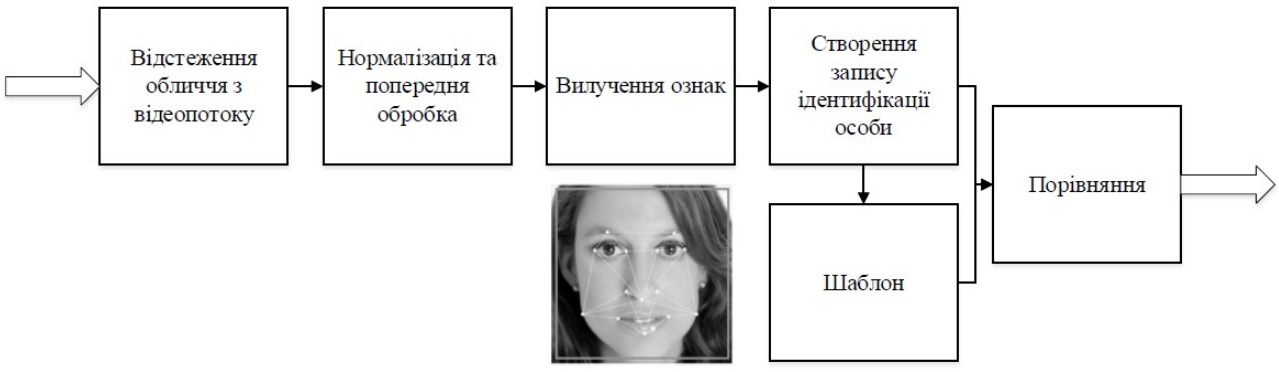


Рисунок 1.6 − Процес 2-D розпізнавання особи

Переваги та недоліки методу

Переваги методу. При 2-D розпізнаванні, на відміну від більшості біометричних методів, не потрібне дороге обладнання. При відповідному обладнанні можливість розпізнавання на значних відстанях від камери.

Недоліки. Низька статистична достовірність. Пред’являються вимоги до освітлення. Для багатьох алгоритмів недопустимі будь-які зовнішні перешкоди, наприклад, окуляри, борода, деякі елементи зачіски. Обов’язковим є фронтальне зображення особи з невеликими відхиленнями. Багато алгоритмів не враховують можливі зміни міміки обличчя, тобто вираз має бути нейтральним.

3-D розпізнавання обличчя. У даний час існує безліч методів за 3-D розпізнаванням особи. Методи неможливо порівняти один з одним, так як вони використовують різні сканери та бази.

Перехідним від 2-D до 3-D методом є метод, який реалізує накопичення інформації про особу. Він використовує всього одну камеру. При занесенні суб’єкта у базу, він повертає голову і алгоритм з’єднує зображення воєдино, створюючи 3d шаблон (наведено на рисунку 1.7).

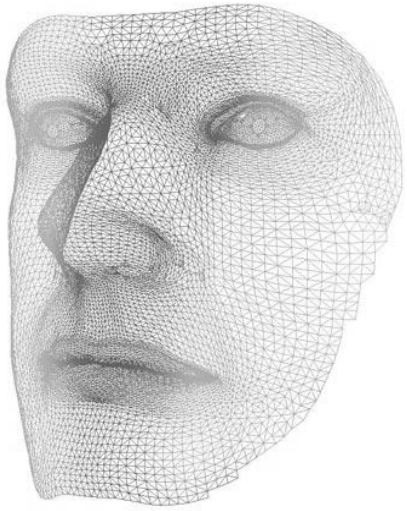


Рисунок 1.7 – 3-D шаблон для розпізнавання особи

Найбільш класичним методом є метод проектування шаблону. Він полягає в тому, що на об’єкт (особу) проектується сітка. Далі камера робить

знімки з швидкістю десяти кадрів у секунду, після чого отримані зображення обробляються спеціальною програмою. На першому етапі обробки видаляються зображення, на яких обличчя не видно взагалі або присутні сторонні предмети, що заважають ідентифікації. За отриманими знімками відновлюється 3-D модель, на якій виділяються і видаляються непотрібні перешкоди (зачіска, борода, вуса й окуляри). Потім проводиться аналіз моделі − виділяються антропометричні особливості, які в підсумку й записуються в унікальний код, заносячи його в базу даних. Час захоплення і обробки зображення становить 1-2 секунди для кращих моделей.

Переваги та недоліки методу

Переваги методу. Відсутність необхідності контактувати зі скануючим пристроєм. Низька чутливість до зовнішніх факторів, як на самій людині (поява окулярів, бороди, зміна зачіски), так і в її оточенні (освітленість, поворот голови).

Недоліки методу. Дуже дороге обладнання. Зміни міміки обличчя та перешкоди на обличчі погіршують статистичну надійність методу. Метод ще недостатньо добре розроблений, що ускладнює його широке застосування.

Ідентифікація за райдужною оболонкою ока. Малюнок райдужки випадковий, а чим більше ступінь випадковості, тим більше ймовірність того, що конкретний малюнок буде унікальним.

Ідея використовувати текстуру райдужки для ідентифікації особи була запропонована в 70-80 х роках минулого століття. А результати досліджень було вперше опубліковано Даугманом на конференції у 1992 році. На сьогоднішній момент робота Даугмана є основоположною працею в даній області. У 1994 році система ідентифікації особистості за райдужною оболонкою ока на основі досліджень Даугмана була запатентована [17].

Для того, щоб зафіксувати малюнок на райдужці потрібна фотокамера з високим розширенням. Отримане зображення збільшується і перетворюється в унікальний код, який присвоюється людині.

Малюнок райдужки, який остаточно формується на другому році життя дитини, практично не змінюється протягом життя, якщо людина не отримує травм і не страждає від серйозних офтальмологічних патологій.

Перевагою методу є простота у скануванні. Людині не обов'язково зосереджено дивитися в одну точку, бо пляма на сітківці знаходиться прямо на поверхні очного яблука і легко зчитується на відстані, що не перевищує 1 м. Використовувати даний метод зручно в банківських організаціях або громадському транспорті. Зацікавились технологією і виробники смартфонів – у 2015 році в Японії до продажу надійшла перша модель із сканером райдужної оболонки.

Деякі захворювання спричиняють на райдужній оболонці появу характерних пігментних плям та зміну кольору очей. Для ослаблення впливу стану здоров'я на результати ідентифікації особи в технічних системах використовують лише чорно-білі зображення високої роздільної здатності. Ціни цих систем коливаються в інтервалі від 500$ до 6500$.

Однак ця система має і певні недоліки, зумовлені фізіологічними змінами організму людини. З роками в розміщенні плям на райдужній оболонці людини можливі суттєві зміни, наприклад, райдужна оболонка ока дитини з роками може змінюватися настільки, що біометрична система просто не зможе її розпізнати. Крім того, похибка негативної ідентифікації може з'явитися при невеликих травмах ока чи навіть у результаті безсоння або великих навантажень на очі. Зміни такого роду незначні, але система ідентифікації може в таких випадках не розпізнати райдужну оболонку ока [17].

Побудова коду проводиться в три етапи:

- виділення «кола» райдужки із загального зображення;

- попередня обробка отриманого зображення – наприклад видалення шуму, поліпшення зображення, у тому числі вирівнювання гістограми, усунення відблиску. Помилка на даному етапі може вплинути на подальше розпізнавання;

- складання коду. Попередньо оброблене зображення фільтрується способом, що залежить від конкретного методу. За результатами фільтрації складається представлення у вигляді коду.

Якщо для захоплення зображення не було використано спеціальної апаратури, може знадобитися попереднє видалення небажаних ефектів, таких як відблиск всередині зіниці від спалаху або іншого яскравого джерела світла, якщо ці ефекти заважають коректній роботі алгоритму виділення райдужки.

Переваги та недоліки методу

Переваги методу. Статистична надійність алгоритму. Захоплення зображення райдужної оболонки можна проводити на відстані від декількох сантиметрів до декількох метрів, при цьому фізичний контакт людини з пристроєм не відбувається. Райдужна оболонка захищена від пошкоджень, а значить не буде змінюватися у часі. Так само, можливо використовувати високу кількість методів, щоб уникнути підробки.

Недоліки методу. Ціна системи, заснованої на райдужній оболонці вище ціни системи, заснованої на розпізнаванні пальця або на розпізнаванні обличчя. Низька доступність готових рішень. Здебільшого продаються дорогі системи, встановлювані великими компаніями, такими як Iridian або LG.

Ідентифікація за сітківкою ока. Просканувати сітківку − внутрішню оболонку очного яблука, що реагує на світло, важче: для цього до кровоносних судин задньої стінки ока, через зіницю посилають низько інтенсивні інфрачервоні світлові промені. Подібний метод встановлення особистості вважається високоефективним та активно використовується на державних і військових об’єктах.

І знову виробники мобільних гаджетів не залишились осторонь. Цілий ряд компаній працює над створенням комбінованих технологій ідентифікації за сітківкою та райдужкою.

Ймовірність пропуску незареєстрованого користувача (помилки першого роду) при скануванні сітківки ока складає 0,0001%. Водночас ймовірність помилки другого роду досить висока − близько 0,1%. Це пояснюється тим, що

спочатку дані системи були розроблені на замовлення військових, де до помилок першого роду пред’являють найжорсткіші обмеження. При цьому передбачається, що користувачі можуть повторити процедуру автентифікації декілька разів.

Ця біометрична система ідентифікації має низьку пропускну здатність і досить дорога в використанні, проте ступінь її надійності перевищує інші в рази. Процес розпізнавання особи за сітківкою ока зображено на рисунку 1.8.

Рисунок 1.8 − Процес розпізнавання особи за сітківкою ока

На відміну від змінюваної, залежно від освітлення, тиску та наявності в організмі лікарських засобів, райдужної оболонки, сітківка ока залишається нерухомою і не змінюється з часом. Помилки можливі лише у разі хвороби ока, наприклад, катаракти, але, такі випадки − лише винятки. А венозний малюнок окового дна − ще один незмінний «контур захисту», що є унікальним для кожної людини. Ідентифікація за сітківкою ока заслужила собі почесне місце в системах безпеки державних секретних служб.

Сутність процедури такого методу ідентифікації − в скануванні сітківки ока за допомогою вузького пучка світла інфрачервоного діапазону, спрямованого через зіницю на очне дно. В останніх моделях сканерів замість інфрачервоного світла використовується лазер м’якої дії. Людина повинна наблизити обличчя до сканера, зафіксувати його положення та направити

погляд на спеціальну мітку на дисплеї сканера. У цьому, мабуть, єдиний недолік подібних систем − необхідність зберігати нерухомість протягом досить тривалого часу. Тобто, сканери сітківки ока не підходять для установки на в’їздах і прохідних підприємства: це просто зупинить виробничий процес. Подібні системи передбачені, в першу чергу, для безпеки приватних кабінетів, сейфів і сховищ [18].

Переваги та недоліки методу

Переваги. Високий рівень статистичної надійності. Через низьку поширеність систем мала ймовірність розробки способу їх «обману».

Недоліки. Складна при використанні система з високим часом обробки. Висока вартість системи. Відсутність широкого ринку попиту і, як наслідок, недостатня інтенсивність розвитку методу.

1.4 Аналіз аналогів та сучасного стану досліджуваної предметної області

Широке застосування комп'ютерних технологій в автоматизованих системах обробки інформації та управління призвело до загострення проблеми захисту інформації, що циркулює в комп'ютерних системах, від несанкціонованого доступу.

Широке використання комп'ютерних технологій в системах автоматизованої обробки та управління посилило проблему захисту інформації, що циркулює в комп'ютерних системах, від несанкціонованого доступу.

Захист інформації в комп'ютерних системах має низку специфічних особливостей, пов'язаних із тим, що інформація не є жорстко пов'язаною з носієм, може легко та швидко копіюватися та передаватися по каналах зв'язку. Відомо дуже велику кількість загроз інформації, які можуть бути реалізовані як з боку зовнішніх порушників, так і з боку внутрішніх порушників.

Комп'ютери забезпечують збереження інформації, її обробку та надання споживачам, реалізуючи в такий спосіб інформаційні технології. Однак саме

найвищий ступінь автоматизації, до якого прагне сучасне суспільство, ставить його в залежність від ступеню безпеки використовуваних ним інформаційних технологій. Як показує аналіз останніх років, постійно винаходяться нові й нові види та форми обробки інформації і паралельно винаходяться все нові й нові види та форми її захисту.

Наукові та технічні передумови ситуації із забезпеченням інформаційних технологій [19]:

1. Збільшення обсягів інформації, що накопичується, зберігається та обробляється за допомогою ЕОМ та інших засобів обчислювальної техніки (ЗОТ). Це відноситься не лише різкого та буквального збільшення самих обсягів, а й розширення арсеналу методів, способів і можливостей її зосередження та збереження.

2. Сучасні комп'ютери за останні роки отримали величезну обчислювальну потужність, але стали набагато простішими в експлуатації. Це означає, що користуватися ними стало набагато простіше і, як наслідок, все більша кількість нових користувачів одержує доступ до комп'ютерів. Середня кваліфікація користувачів знижується, що значною мірою полегшує задачу ЗЛ, оскільки в результаті такої «персоналізації» ЗОТ більшість користувачів мають власні робочі станції і самі здійснюють їх адміністрування. Більшість із них не здатні постійно підтримувати безпеку своїх систем на високому рівні, оскільки це вимагає відповідних знань, навичок, а також часу та коштів.

3. Прогрес у сфері апаратних засобів супроводжується ще більш бурхливим розвитком програмного забезпечення (ПЗ). Як показує практика, більшість розповсюджених сучасних програмних засобів (у першу чергу – операційних систем (ОС)), незважаючи на великі зусилля розробників у цьому напрямку, не відповідають навіть мінімальним вимогам безпеки. Головним чином це виражається в наявності вад в організації засобів, що відповідають за безпеку.

4. У сучасних умовах надзвичайно важливим є обґрунтування вимог безпеки, створення нормативної бази, яка не ускладнює задачі розробників, а, навпаки, встановлює обов'язковий рівень безпеки.

Унаслідок сукупної дії перерахованих факторів перед розробниками сучасних інформаційних систем, призначених для обробки важливої інформації, постають такі завдання, що вимагають негайного й ефективного вирішення [20]:

1. Забезпечення безпеки інформаційних ресурсів. Оскільки комп'ютерні системи тепер прямо інтегровані в інформаційні структури сучасного суспільства, засоби захисту повинні враховувати сучасні форми представлення інформації (системи захисту повинні забезпечувати безпеку на рівні інформаційних ресурсів, а не окремих документів, файлів чи повідомлень).

2. Інтеграція захисту інформації в процес автоматизації її обробки як обов'язковий елемент. Для того, щоб бути затребуваними сучасним ринком інформаційних систем, засоби безпеки не повинні вступати в конфлікт з існуючими додатками та сформованими технологіями обробки інформації, а, навпаки, мають стати невід'ємною частиною цих засобів і технологій.

3. Досвід експлуатації існуючих систем показав, що сьогодні від систем захисту вимагаються зовсім нові функції.

У сучасних інформаційних системах інформаційна безпека має забезпечувати передусім цілісність та доступність інформації, а не лише її конфіденційність.

Основна частина інформаційних загроз (понад 90%) реалізується всередині самої системи — це стосується тривіальної компетентності, некваліфікованого та недбалого персоналу системи.

Велика кількість користувачів вважає, що їхня інформація недостатньо цінна, тому для її захисту не потрібно докладати жодних зусиль. Така точка зору може бути дуже небезпечною, оскільки оцінка важливості або цінності інформації, особливо власної, є достатньо складним і багато в чому суб'єктивним процесом.

Засоби захисту від несанкціонованого доступу (НСД), що реалізовані в КС, слід розглядати як підсистему захисту від НСД у складі СЗІ. Характеристики фізичного середовища, персоналу, оброблюваної інформації, організаційної підсистеми істотно впливають на вимоги до функцій захисту, що реалізуються КС.

Обчислювальна система — поєднання апаратного та програмного забезпечення, призначеного для обробки інформації. Кожен із компонентів ОС може бути розроблений і проданий як самостійний продукт, а також може реалізовувати певні функції захисту інформації.

КС означає комплекс апаратних, програмних, програмно-апаратних засобів та окремих організаційних заходів, що впливають на інформаційну безпеку.

Як КС можуть виступати: ЕОМ загального призначення або персональна ЕОМ; ОС; прикладна або інструментальна програма (пакет програм); підсистема захисту від НСД, що являє собою надбудову над ОС; локальна обчислювальна мережа; мережева ОС; ОС автоматизованої системи; в найбільш загальному випадку − сама АС або її частина [20].

Конфіденційність, цілісність, доступність та спостережність — основоположні терміни, що характеризують значущість інформації, яку потрібно захищати.

Термін конфіденційність визначається як властивість інформації, яка полягає в тому, що вона не може бути доступною для ознайомлення користувачам і/або процесам, що не мають на це відповідних повноважень.

Цілісність інформації − це властивість, яка полягає в тому, що вона не може бути доступною для модифікації користувачам і/або процесам, що не мають на це відповідних повноважень. Цілісність інформації може бути фізичною і/або логічною.

Доступність інформації − це властивість, що полягає в можливості її користування на вимогу користувача, який має відповідні повноваження.

Спостереженість − це властивість інформації, яка полягає в тому, що процес її обробки повинен постійно перебувати під контролем.

Рівень захищеності інформації в системі − це певна міра (наприклад, ймовірнісна) можливості виникнення на якому-небудь етапі життєдіяльності системи такої події, наслідком якої можуть бути небажані впливи на інформацію, тобто порушення хоча б одного із зазначених ФВЗІ.

Головне спрямування, що характеризує розвиток сучасних інформаційних технологій — це досить швидше зростання числа комп'ютерних злочинів, пов'язаних із ними крадіжок конфіденційної та іншої інформації, а також, як наслідок, і матеріальних втрат.

На теперішній час, неможливо визначити точну кількість загальних втрат через комп’ютерні злочини, пов’язані з неправомірним доступом до конфіденційної інформації. Здебільшого, це через те, що «постраждалі» підприємства та компанії не розголошують про подібні випадки, щоб не спонукати ще більших посягань на розроблювані продукти і дані. Але крім того, не завжди можливо точно оцінити в грошовому еквіваленті втрати від розкрадання інформації.

Будь-яке сучасне підприємство, незалежно від виду діяльності та форми власності, не в змозі успішно розвиватися та вести господарську діяльність без створення на ньому умов для надійного функціонування СЗ власної інформації.

Відсутність у багатьох керівників підприємств і компаній чіткого уявлення з питань захисту інформації призводить до того, що їм складно повною мірою оцінити необхідність створення надійної системи захисту інформації на своєму підприємстві і тим більше складно буває визначити конкретні дії, необхідні для захисту тих чи інших конфіденційних відомостей.

Необхідно чітко визначати коло співробітників, які допускаються до конфіденційної інформації та повноваження співробітників з доступу до неї.

У даний час широко застосовуються спеціалізовані програмні та програмно-апаратні засоби захисту інформації, які дозволяють максимально

автоматизувати процедури доступу до інформації та забезпечити при цьому необхідний ступінь її захисту.

Серед аналогів розроблюваної системи можна відзначити декілька основних:

1) за допомогою сканування відбитків пальців на смартфоні можна розблоковувати лише сам смартфон та додатки на ньому;

2) для розблокування ОС на комп'ютері вимагається введення логіну та паролю до облікового запису користувача — це не забезпечує повного захисту системи, витрачається чимало часу на запам'ятовування та введення надійного паролю, а також його періодичну зміну;

3) звичайно, у наш час вже з'явилися ноутбуки з вбудованим сканером відбитків пальців для спрощеного та надійного входу в захищену систему, але для того, щоб користуватися таким варіантом, потрібно придбати нову, досить коштовну техніку, а якщо це потрібно використовувати на підприємстві, де працює велика кількість людей, то запровадження вартуватиме чималих грошових вкладень.

Розроблювана система істотно відрізняється від розглянутих аналогів, адже:

- спрощує процедуру входу в захищену комп'ютерну систему;

- забезпечує неможливість підміни ідентифікаційних даних або оволодіння ними обманним шляхом;

- знижує навантаження на користувачів та адміністраторів;

- забезпечує мінімізацію витрат на засоби ідентифікації та розмежування/ управління доступом;

- дає змогу адміністратору контролювати дії з обліковими записами користувачів та вести облік робочого часу.

Ефективність захисту інформації досягається використанням системи захисту інформації. СЗІ — це технічний, програмний засіб або матеріал, що використовується для ефективного захисту інформації. На ринку є велика

різноманітність засобів захисту інформації, які за напрямками захисту поділяються на декілька груп [21]:

- засоби, що забезпечують розмежування доступу до інформації в автоматизованих системах;
- засоби, що забезпечують захист інформації під час її передачі каналами зв'язку;
- засоби, що гарантують захист від витоку інформації за різними фізичними параметрами, які з’являються при роботі з технічними засобами автоматизованих систем;
- засоби, що забезпечують захист від дій програм-вірусів;
- матеріали, що задовольняють рівень безпеки зберігання, перенесення носіїв інформації та їх захист від копіювання.

Призначення засобів розмежування доступу в автоматизованих системах — це, передусім, їх застосування відносно локальних і мережевих інформаційних ресурсів.

СЗІ цієї групи забезпечують:

- ідентифікацію та аутентифікацію користувачів системи;
- розмежування доступу зареєстрованих користувачів до інформаційних ресурсів;
- реєстрацію дій користувачів;
- контроль цілісності інформаційних ресурсів [21].

У ролі ідентифікаторів користувачів використовуються здебільшого позначення на кшталт набору символів, а для автентифікації — переважно паролі. Введення цих значень відбувається за запитом системи захисту інформації.

Використання окремих біологічних параметрів як ідентифікаторів характеризується, з однієї сторони, більш високим рівнем конфіденційності, але з іншої — досить високою вартістю подібних систем. Розроблена у магістерській роботі система вирішує проблему вартості подібних систем. Доступ зареєстрованих користувачів до інформаційних ресурсів має бути

обмеженим відповідно до повноважень користувача. Облікові дані користувачів встановлюються із використанням спеціальних налаштувань.

Розроблена система захисту інформації передбачає ведення спеціального журналу, в якому реєструються певні події, що стосуються дій користувачів, наприклад, дата та час початку поточної сесії, минулі сесії цього користувача, спроби несанкціонованого доступу до ресурсів, що захищаються, завершення поточної сесії.

Попри функціональну подібність ЗЗІ цієї групи, все ж вони відрізняються середовищем, у якому працюють (операційне середовище, апаратна платформа, автономні комп’ютери чи комп’ютерні мережі), складністю налаштування та керування параметрами, типами використовуваних ідентифікаторів, списком подій, які треба зареєструвати, а також вартістю засобів.

Загальновизнаним фактом є констатація ненадійності, незручності та неефективності застарілої парольної системи ідентифікації.

Досить популярними є рішення, що використовують для ідентифікації різноманітні матеріальні носії - смарт-карти, токени, «таблетки» Touch Memory. У деяких із цих засобів захист ідентифікаційної інформації забезпечується шифруванням, генерацією одноразових паролів, електронним підписом і цифровими сертифікатами.

Якщо застосовувати розглянуті варіанти не самостійно, а у комплексі із засобами біометричної ідентифікації, то це забезпечить значно ефективніший результат. Хоча й у першому випадку незмінним залишається те, що біометрія — це точність, зручність та швидкість ідентифікації.

Зазвичай заходи щодо захисту інформації пов'язані з додатковим навантаженням і новими обов'язками, покладеними на користувачів, адміністраторів, співробітників служби безпеки. Біометрія є винятком: її впровадження полегшує діяльність працівників усіх зазначених груп, одночасно забезпечуючи істотне зростання рівня інформаційної безпеки [22].

Розроблена система біометричної ідентифікації досить ефективна, оскільки виконує наступні функції:

- уніфікація процесів доступу до інформаційних ресурсів на основі єдиного ідентифікатора — відбитка пальця;
- захист інформаційних ресурсів від несанкціонованого доступу і / або доступу, здійснюваного з порушенням встановлених правил;
- автоматизація та централізація управління користувачами, їх обліковими записами та відповідними повноваженнями в операційних системах і різних прикладних продуктах;
- прискорення доступу легальних користувачів до ресурсів корпоративної мережі із забезпеченням максимальної простоти та прозорості цього процесу;
- скорочення непродуктивних втрат, викликаних помилками при введенні логіна / пароля, блокуванням облікових записів і т.п.;
- оптимізація діяльності системних адміністраторів за рахунок зменшення числа звернень, пов'язаних з помилками користувачів при ідентифікації, аутентифікації і авторизації;
- швидка реєстрація нових користувачів і оперативна зміна облікових записів співробітників, які перейшли в інший підрозділ, звільнених і т.п.

Усі наявні продукти на ринку аналогічні цьому і мають свої недоліки. Ідеального не існує, тому є необхідність розробки нових програмних засобів ідентифікації з подальшою авторизацією користувача захищеної системи.

1.5 Висновки до розділу 1

Біометрія – це ідентифікація особи за унікальними біологічними ознаками, що однозначно її вирізняють з-поміж усіх інших. Використання біометричних ознак у сфері інформаційної безпеки є абсолютно зрозумілим рішенням, оскільки цей напрямок дуже активно розвивається.

Завдяки незаперечним перевагам, системи біометричної ідентифікації досить високо позиціонуються на ринку інформаційної безпеки. Цей тип систем вирізняється не лише швидкістю обробки даних та стабільністю авторизаційних даних, але й прийнятною вартістю, що однозначно схиляє до їх впровадження усе більшу кількість підприємців.

Було проведено аналіз особливостей побудови існуючих компонентів безпеки комп'ютерних систем, наведено класифікацію методів ідентифікації, показано ефективність побудови компонентів безпеки на основі біометричних даних, наведено класифікацію компонентів захисту комп'ютерних систем та мереж за класифікаційною ознакою, що визначає рівень очікуваного ефекту захищеності.

Проаналізувавши теперішній стан у сфері досліджень ефективності елементів безпеки комп'ютерних систем з використанням біометричних ознак було зроблено висновок про актуальність та доцільність подальших досліджень цього напрямку.

На основі детального розгляду основних типів загроз по відношенню до інформаційної безпеки та існуючих засобів ідентифікації користувачів комп'ютерних систем, зроблено висновок, що у подальшому, зі зростанням продуктивності комп'ютерів, використання систем біометричної ідентифікації буде все необхіднішим, оскільки це дасть змогу значно підвищити рівень надійності систем ідентифікації.

2 РОЗРОБКА ОПТИМАЛЬНОГО МЕТОДУ ІЗ ВИКОРИСТАННЯМ БІОМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ ЗА ВІДБИТКАМИ ПАЛЬЦІВ ДЛЯ РОЗВ'ЯЗКУ ПОСТАВЛЕНИХ ЗАДАЧ

Через інтенсивний розвиток комп'ютерної техніки, а також збільшення кількості користувачів, усе важливішими стають питання контролю доступу до їх ресурсів. Такий контроль реалізується завдяки заданим адміністратором обмеженням доступу — можливості виконати певні дії користувачів відносно ресурсів системи.

Основу системи інформаційної безпеки утворюють засоби ідентифікації користувачів і управління їх доступом до корпоративних інформаційних ресурсів. Завдання, що будуть вирішені при впровадженні розробленої системи ідентифікації:

- однозначність розпізнавання користувача по унікальних, властивих йому одному ознаках;
- неможливість розкрадання, втрати, підміни ідентифікаційних ознак та / або оволодіння ними обманним шляхом;
- запобігання можливому обміну ідентифікаторами та відповідними повноваженнями користувачів;
- реалізація принципу «неможливості відмови» користувача від транзакцій, здійснених із застосуванням ідентифікаторів, що відповідають перерахованим вище критеріям;
- ефективна інтеграція засобів ідентифікації та управління доступом в інформаційну інфраструктуру компанії зі збереженням безперервності поточних бізнес-процесів;
- зниження навантаження на користувачів та адміністраторів;
- мінімізація витрат, пов'язаних із використанням засобів ідентифікації та управління доступом;
- відмовостійкість і масштабованість формованої системи ідентифікації та управління доступом.

2.1 Розробка методу вибору ідентифікації користувача за відбитками пальців через смартфон

Серед усіх наявних біометричних технологій було обрано сканування та розпізнання відбитків пальців для вирішення основної задачі магістерської роботи, оскільки таке рішення є найоптимальнішим для створення засобів систем інформаційної безпеки.

Порівняльні переваги даного виду біометричної аутентифікації наступні [23]:

- у порівнянні з технологіями розпізнання за сітківкою та райдужною оболонкою ока, сканування відбитків пальців дешевше та зручніше, оскільки для входу в різні програмні системи та виконання в них критичних операцій, що вимагають строгої ідентифікації користувача, простіше кілька разів прикласти палець до сканера, ніж декілька разів правильно приставляти око до камери;

- у порівнянні з технологіями розпізнання за формою та розташуванням вен на лицьовій стороні долоні, сканери відбитків пальців менш громіздкі, а процес безпосереднього зчитування ідентифікуючих ознак — зручніший;

- у порівнянні з технологіями розпізнання за формою та термограмою обличчя — усі, що перераховані вище, тобто незручність використання в повсякденних діях і громіздкість устаткування;

- у порівнянні з технологіями розпізнання за рукописним і клавіатурним почерком – у технології ідентифікації за відбитками пальців на кілька порядків кращі статистичні показники помилок першого та другого роду, це ж відноситься і до систем розпізнання за голосом.

Сканер відбитків пальців – це вид технології безпеки на основі біометричних ознак, що використовується для розпізнавання за відбитками пальців, завдяки поєднанню апаратних і програмних методів. За допомогою нього відбувається процес ідентифікації та перевірки справжності отриманих відбитків, в результаті чого приймається рішення щодо надання чи не надання доступу до смартфону, додатків на ньому чи інших місць, які потребують

захисту від небажаного втручання. Сканер відбитків пальців було обрано з ряду причин: вартість — плати для сканування відбитків досить дешеві (оскільки вони прості у виготовленні), зручність використання та швидкість — потрібно лише на долі секунди доторкнутися до сканера і відразу ж приступати до роботи зі смартфоном.

Ідентифікація за відбитками пальців – на сьогоднішній день є найпоширенішою БТ. За даними International Biometric Group, частка систем розпізнавання за відбитками пальців складає 48% від усіх використовуваних у світі БС [23].

Як і в інших біометричних методах, у методі з використанням відбитків пальців для ідентифікації особи користувача можна виділити два основних кроки: реєстрацію та розпізнавання (ідентифікацію/аутентифікацію).

Процес реєстрації користувача складається з трьох основних етапів:

1) введення отриманого зображення папілярних ліній;
2) виділення індивідуальних елементів малюнка;
3) формування шаблону;
4) збереження зразка.

Завдяки новим напрацюванням, програмне забезпечення, яке використовується у пристроях, що використовують розпізнавання за відбитками пальців, дає змогу аналізувати лише найпростіші елементи малюнку папілярних ліній, не враховуючи розташування пор на шкірі. Це робить процес перевірки набагато швидшим, на відміну він повного аналізу всього відбитку.

Усі існуючі на теперішній час типи сканерів працюють однаковим чином: при спробі розблокувати смартфон, на сканері зчитується відбиток пальця користувача, порівнюється із тим, що вже занесений у пристрій, якщо співпадіння буде знайдено — смартфон розблокується і з ним можна буде працювати, а у протилежному випадку — користувач побачить інформативне повідомлення про помилку.

Збереження даних у спеціальній області виключає можливість вилучення відбитків із бази даних.

Сканер відбитку пальців в Android-пристроях − вже звична річ. Але раніше функціональність сканерів обмежувалася лише можливістю розблокування смартфона, а оскільки Android офіційно не підтримував роботу зі сканерами, то розробникам доводилося постійно шукати шляхи обходу. Після того, як Google випустили нову версію Android — 6.0 Marshmallow, одним із достоїнств якої стала нативна підтримка сканерів відбитка та Fingerprint API, нарешті з’явилася можливість використовувати усі наявні переваги сканеру в розроблюваних додатках [24].

Враховуючи усі проблеми та незручності, із якими зіштовхуються користувачі комп'ютерних систем, адміністратори і, як наслідок, самі підприємці, які несуть чималі втрати, у зв'язку з незахищеністю системи було розроблено систему ідентифікації користувачів, яка вирішує їх.

Дана розробка забезпечує зручність у використанні та адмініструванні. До прикладу, користувачеві комп'ютерної системи, який працює з конфіденційними даними у якійсь фірмі/організації/установі, потрібно завжди блокувати систему, коли він покидає своє робоче місце, хай навіть на декілька хвилин (оскільки йому потрібно захищати не конкретно якусь програму/файл, а всю інформацію, яку містить у собі система), а потім втрачати час на те, щоб ввести довгий, складний пароль до облікового запису (в якому можна ще й помилитися, та й взагалі, усі недоліки парольного захисту було описано у першому розділі). А якщо блокувати систему потрібно по декілька разів на годину? Користувач просто не буде цього робити, що може призвести до чималих втрат.

За допомогою розробленої системи усе робиться доволі просто: для початку користувач реєструється через смартфон у додатку, за допомогою сканера відбитків пальців підтверджує свою особу, після чого адміністратору відправляється запит на додавання і він вирішує чи надавати доступ певному користувачеві. Після успішної реєстрації замість того, щоб проводити безліч

зайвих маніпуляцій, вводів паролів і т. п. достатньо лише зайти у додаток на своєму смартфоні, обрати свій обліковий запис і прикласти палець до сканеру відбитків — доступ до комп'ютерної системи буде надано, після успішної ідентифікації.

Адміністратору виводиться уся інформації про вдалі чи невдалі спроби входу, час, із якого акаунту їх було здійснено, що значно спрощує процес адміністрування та економить час. Адміністратор повинен налаштовувати підключення до конкретної захищеної комп’ютерної системи в адмін-панелі. Для цього потрібно лише вказати у формі відповідну IP-адресу та порт, до якого буде відбуватися під’єднання.

IP означає інтернет-протокол і описує набір стандартів та вимог для створення й передачі пакетів даних по мережах. Інтернет-протокол є частиною інтернет-рівня набору і призначений для роботи в динамічній мережі (IP повинен працювати без центрального каталогу або монітору і не може покладатися на конкретні посилання чи існуючи вузли.

Зв’язок між смартфоном та захищеною системою відбуватиметься при умові, що обидва об’єкти знаходяться в одній локальній мережі. Це може бути як підключення по Wi-Fi, так і в режимі «Спільна точка доступу», коли зі смартфону роздається трафік, водночас як на комп’ютері відбувається підключення до цієї точки. Адміністратор у своєму кабінеті має можливість вказати до якого системи буде відбуватися під’єднання, вказуючи IP-адресу та порт (рис. 2.1).

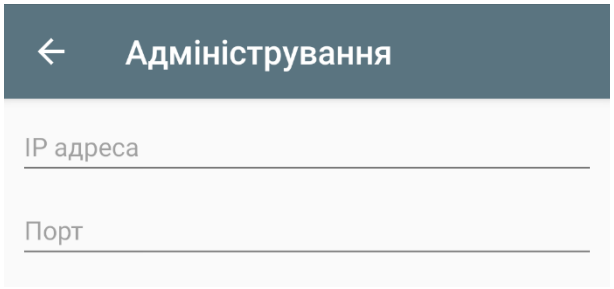


Рисунок 2.1 — Вікно для адміністрування під’єднань

Для підприємця/роботодавця теж є свої «плюси», а саме — мінімізація витрат. Адже система буде захищена, про втрати цінної інформації не варто

хвилюватися, а ніякого додаткового коштовного обладнання купувати не потрібно, оскільки смартфони з вбудованими сканерами відбитків пальців стали дуже розповсюдженими і є у переважної більшості користувачів. А на деяких підприємствах і взагалі є «робочі» телефони, які можна використовувати для цього. Не потрібно додатково «наймати» фахівців з технічної реалізації або витрачати значні кошти та час на навчання наявних фахівців. Система є інтуїтивно зрозумілою, тому не виникне ніяких проблем ні з боку адміна, ні з боку користувача.

Для наглядності розроблено блок-схему алгоритму роботи додатку (рис. 2.2).

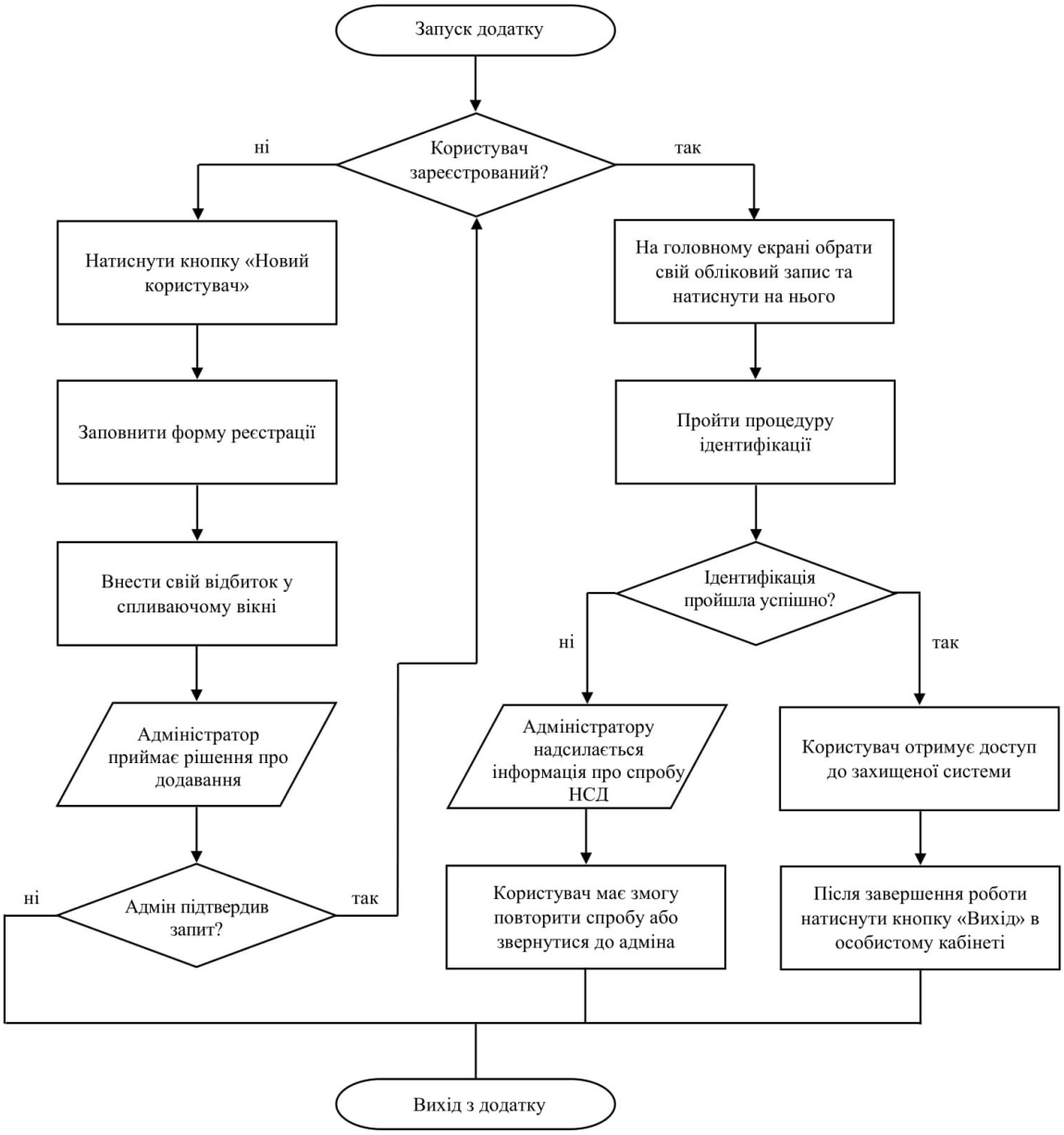


Рисунок 2.2 — Блок-схема алгоритму роботи програми

Отже, розроблена система поєднує у собі:

1) зручність;

2) захищеність;

3) контроль;

4) мобільність;

5) зрозумілість;

6) швидкість;

7) економність.

Завдяки появі сканерів відбитків пальців на пристроях Android, з’явилася альтернатива традиційним методам ідентифікації користувачів. Перехід на використання відбитків пальців для ідентифікації користувачів набирає обертів, оскільки включає захист, що менш нав'язливий, ніж ім'я користувача та пароль.

2.2 Розробка методики розмежування доступу на основі отриманих біометричних даних

У комп'ютерних системах список контролю доступу містить список дозволів та користувачів, до яких застосовуються ці дозволи. Такі дані можуть бути переглянуті тільки певними людьми та контролюються доступом. Ця технологія дозволяє адміністратору зберігати інформацію та встановлювати привілеї стосовно того, до якої інформації можна отримати доступ, хто може отримати доступ до неї та в який час вона може бути доступна.

Access Control List або ACL — це насамперед універсальний та потужний механізм фільтрації. За допомогою нього можна визначати на кого навішувати відповідні політики, а на кого ні, хто буде приймати участь у деяких процесах, а хто ні, хто має право виконувати певні операції, а хто ні і т.п. [25].

Процедура надання доступу відбувається наступним чином: суб’єкт відправляє запит на виконання певної операції, після чого система перевіряє, які операції (чи їх перелік) доступний конкретно для цього суб’єкта, та виносить рішення щодо надання йому доступу на виконання цієї дії.

При спробі користувача отримати доступ до захищеного ресурсу, оцінка здійснюється у наступному порядку:

1. Відповідність ідентифікатора адміністратора за допомогою записів користувача. Оцінка зупиняється на вході користувача в систему. Дозволи, які надаються є правами доступу в запису відповідності адміністратора.

2. Якщо немає відповідних записів адміністратора, надаються дозволи відповідних користувачів, якщо вони існують.

3. Якщо немає відповідних записів користувача або групи, і немає будь-яких інших записів, то користувач не має прав.

4. Коли неідентифікований користувач намагається отримати доступ до захищеного ресурсу, оцінка здійснюється таким чином: якщо ACL не містить запис для неідентифікованого, то доступ заборонено.

5. Якщо ACL не містить запис для будь-якого іншого, то доступ заборонено (рис. 2.3).

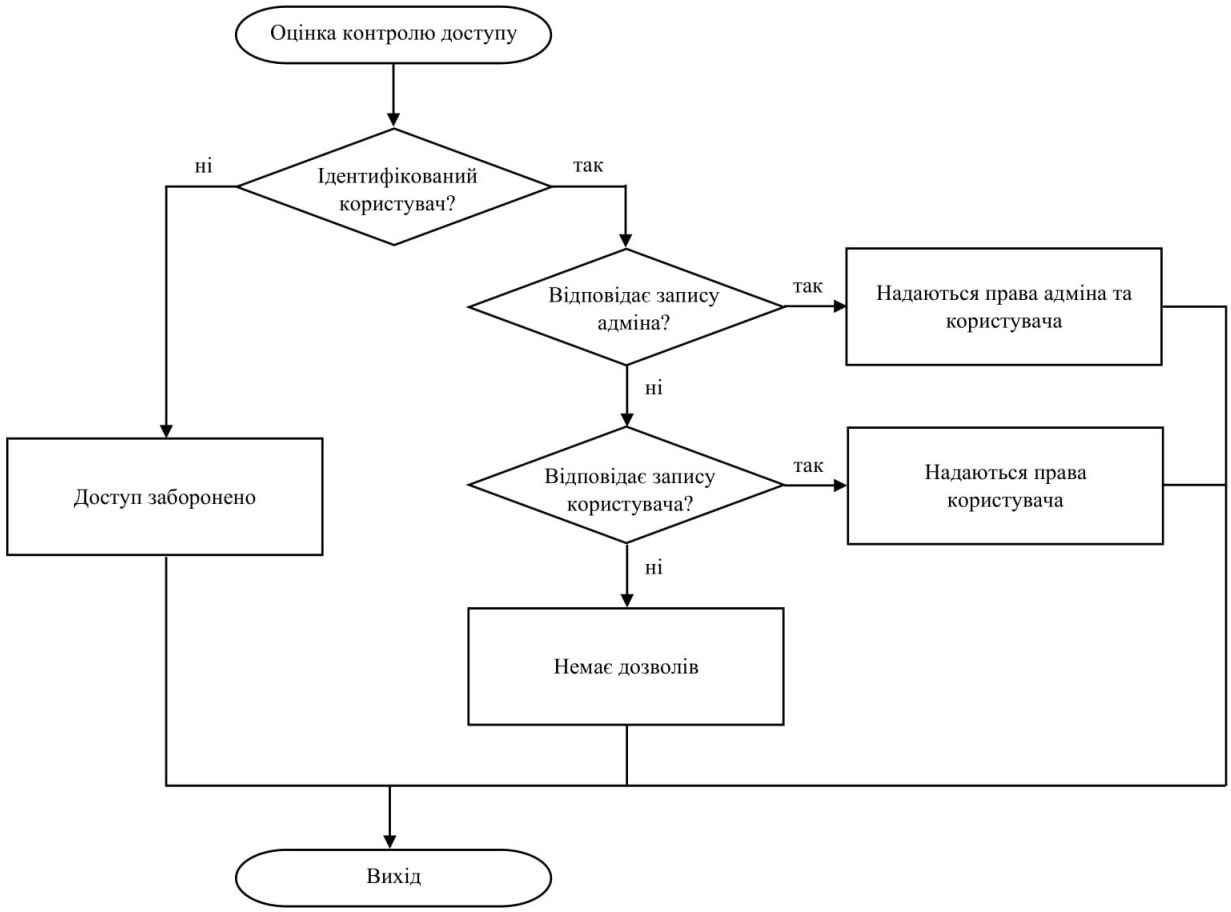


Рисунок 2.3 — Оцінка контролю доступу

Рольовий доступ, що був використаний в основі розробленої системи, має ряд переваг:

- нейтральність по відношенню до конкретних видів прав, а також способів їх перевірки;
- легкість в адмініструванні завдяки покращенню керованості підсистемою розмежування доступу за рахунок встановлення між ролями зв'язків (окрім того, ролей повинно бути набагато менше, ніж самих користувачів).

Рольове керування доступом містить ряд складових [5]:

- користувач та сеанс роботи користувача;
- роль (зазвичай визначається відповідно до організаційної структури);
- об'єкт (сутність, доступ до якої розмежовується);
- операція (залежить від об'єкта);
- право доступу (дозвіл виконувати певні операції над певними об'єктами).

У всіх випадках, використання рольової моделі дозволяє значно підвищити ефективність адміністрування складних автоматизованих систем. Для кожної ролі призначаються права доступу, а згодом і самі користувачі. Ролей може бути багато, так само як і користувачів, що відносяться до кожної із них. Існує випадок, як у розроблюваній системі, коли в одного суб’єкта може бути декілька ролей, у даному випадку — користувач та адміністратор, внаслідок чого відбувається об’єднання прав.

Для програмного модулю ідентифікації користувача за відбитками пальців через смартфон з подальшою авторизацією було обрано рольове управління доступом через низку переваг описаних раніше.

2.3 Проектування бази даних користувачів

База даних може бути заснована на одній моделі чи на комбінації декількох.

Є три основні типи моделей даних, саме: реляційна, ієрархічна та мережева.

Реляційна модель зберігання даних побудована на взаємовідношенні складових частин. Під час створення складних інформаційних моделей, реляційна модель складає сукупність взаємопов’язаних таблиць, водночас як у найпростішому випадку — двовимірний масив чи двовимірну таблицю [26]. Будь-який стовпець такої таблиці називається полем, а будь-який рядок — записом.

Реляційна модель бази даних має такі властивості:

- один елемент таблиці — один елемент даних;
- усі стовпці в таблиці однакового типу;
- кожний елемент має своє унікальне ім’я;
- в таблиці немає однакових рядків;
- порядок рядків у таблиці може характеризуватися кількістю полів та записів, типом даних, а також бути абсолютно довільним.

Дії, що можна виконувати над цією моделлю БД:

- сортування даних (наприклад, за алфавітом);
- вибірка даних за групами (наприклад, класами);
- пошук записів (наприклад, за прізвищами) і т.п.

Зазвичай, реляційна модель містить у собі декілька окремих таблиць, що пов’язані між собою відповідними ключами. Ключ — це поле, що 100-відсотково ідентифікує конкретний запис. У даний час цей тим моделі є найбільш застосовуваним для зберігання даних та дуже зручним у використанні [26].

У свою чергу ієрархічна модель включає в себе такі параметри, як вузли (сукупність властивостей даних, що описують певний об’єкт), зв’язки (визначають взаємозалежність двох вузлів) та рівні (визначають сукупність підпорядкованості вузлів). Тобто, по своїй суті це певна сукупність елементів,

розташованих у порядку їх виконання: від загального до часткового. Надалі, ці елементи створюють граф.

Мережева модель даних являється розширенням ієрархічного підходу, складається з множини записів, що можуть бути власниками або членами групових відносин, а зв’язок між різнорівневими елементами є довільним [27].

Проектування методом сутність-зв’язок. Модель «сутність-зв'язок» є зручним інструментом для єдиного представлення незалежних від ПЗ (яке його втілює) даних.

Головними поняттями ER-моделі є сутність, зв'язок і атрибут.

Проектування методом сутність-зв’язок складається з таких етапів [28]:

- визначення сутностей;
- визначення зв`язків;
- визначення атрибутів;
- визначення ключів сутностей;
- визначення ступеня зв`язку;
- визначення класу належності.

Для розробки бази даних користувачів програмного засобу ідентифікації користувача через смартфон за відбитками пальців було обрано ER-модель, оскільки для даної розробки вона є оптимальним рішенням.

Для початку потрібно визначити сутності, тобто ті типи об’єктів, про які надалі має накопичуватися інформація.

Отже, для вирішення поставленої задачі визначено такі сутності:

- КОРИСТУВАЧ;
- ВІДБИТОК;
- РОЛЬ;
- РІВЕНЬ_ДОСТУПУ.

Другим кроком є визначення зв’язків між обраними сутностями:

- КОРИСТУВАЧ – має – РІВЕНЬ ДОСТУПУ;
- ВІДБИТОК – визначає - КОРИСТУВАЧ;
- РОЛЬ – призначена – КОРИСТУВАЧ;

- РОЛЬ – впливає – РІВЕНЬ_ДОСТУПУ;

На третьому кроці визначаються атрибути (властивості) обраних сутностей:

– КОРИСТУВАЧ (Код_користувача, Повне_ім’я, Телефон, Код_доступу, Код_відбитку, Код_ролі, Пароль);

– РІВЕНЬ_ДОСТУПУ (Код_доступу, Вид_доступу, Права);

– ВІДБИТОК (Код_відбитку, Зображення_відбитку);

– РОЛЬ (Код_ролі, Назва_ролі, Опис_ролі).

Наступний, четвертий етап, характеризується визначенням ключів отриманих сутностей, що дозволить однозначно ідентифікувати екземпляр сутності.

Ключами для даних сутностей будуть:

– КОРИСТУВАЧ (<Код_користувача>, Повне_ім’я, Телефон, Код_доступу, Код_відбитку, Код_ролі, Пароль);

– РІВЕНЬ_ДОСТУПУ (<Код_доступу>, Вид_доступу, Права);

– ВІДБИТОК (<Код_відбитку>, Зображення_відбитку);

– РОЛЬ (<Код_ролі>, Назва_ролі, Опис_ролі).

На п’ятому та шостому етапах проектування необхідно визначити ступінь зв’язку між сутностями, який демонструє кількість прямих зв’язків кожного екземпляру сутності з екземпляром тієї, з якою він пов’язаний, а також клас належності, що буває двох видів: обов’язковий та необов’язковий. Основна характеристика, що відрізняє види класів належності між собою це те, що у першому випадку участь екземплярів даної сутності є обов’язковою, в другому — ні. Усі описані зв’язки бінарні, оскільки пов’язують між собою лише дві сутності.

Для кожного бінарного зв’язку отримуються попередні відношення за такими правилами [28]:

1) якщо ступінь бінарних зв'язків 1:1 і клас належності э обов'язковим для обох сутей, гарантується однократне появлення кожного значення сутей в будь-

якому екземплярі відношень. Тобто, у відношенні ніколи не буде ні порожньої інформації, ні груп надлишкових даних, що повторюються;

2) якщо ступінь бінарного зв'язку 1:1 і клас належності однієї суті є обов'язковим, а іншої — необов'язковим, то необхідна побудова двох відношень. При цьому ключ суті повинен служити первинним ключем для відповідного відношення;

3) якщо ступінь бінарного зв'язку 1:1 і клас належності обох сутей не є обов'язковим, то необхідно використовувати три відношення: по одному для кожної суті, ключі котрих служать первинні ключі відповідних відношень, і одне для зв'язку. Первинним ключем цього відношення може бути будь-яка з двох сутей. Серед своїх атрибутів відношення, що виділяється для зв'язку, повинно мати по одному ключу кожної суті.

4) якщо ступінь бінарного зв'язку 1:N, і клас належності n-зв'язної суті є обов'язковим, то досить використати по одному відношенню на кожну суть, при умові, що ключ кожної суті служить як первинний ключ для відповідного відношення. Додатково, ключ 1-зв'язаної суті повинен бути добавлений як атрибут у відношення, що відводиться n-зв’язній суті.

5) якщо ступінь бінарного зв'язку 1:N і клас належності n-зв’язної суті є необов'язковими, то необхідно формування трьох відношень: по одному для кожної суті, причому ключ кожної суті служить як первинний ключ для відповідного відношення, і одного відношення для зв'язку. Зв'язок повинен мати серед своїх атрибутів ключ суті кожної з зв'язних сутей.

6) якщо ступінь бінарного зв'язку M:N, то для зберігання даних потрібні три відношення: по одному для кожної суті, причому ключ кожної суті служить як первинний ключ для відповідного відношення, та одного відношення для зв'язку. Зв'язок повинен мати серед своїх атрибутів і ключ суті кожної зі зв'язних сутей.

Зв`язок між сутностями може бути представлений у вигляді діаграми ER-екземплярів для кожного зв`язку. Діаграми представлені на рисунках нижче.

Визначення ступеня зв’язку класу належності сутностей «КОРИСТУВАЧ та РІВЕНЬ_ДОСТУПУ» показано на рисунку 2.4.

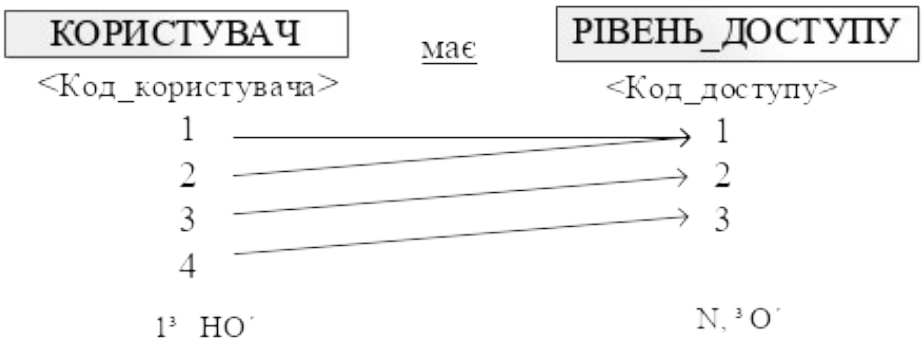


Рисунок 2.4 – Ступінь зв’язку та клас належності сутностей КОРИСТУВАЧ та РІВЕНЬ_ДОСТУПУ

Визначення ступеня зв’язку класу належності сутностей ВІДБИТОК та КОРИСТУВАЧ показано на рисунку 2.5.

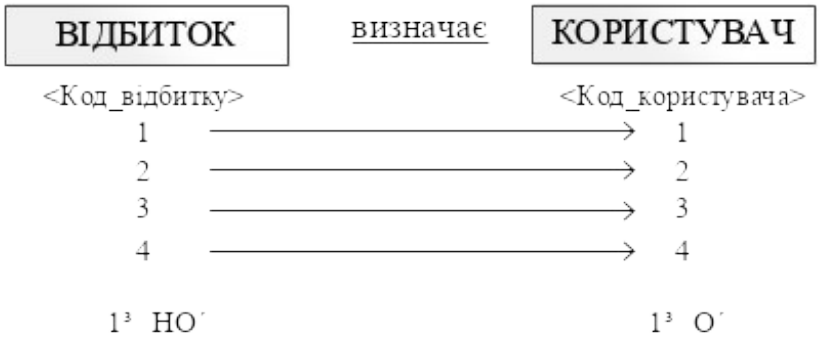


Рисунок 2.5 – Ступінь зв’язку та клас належності сутностей ВІДБИТОК та КОРИСТУВАЧ

Визначення ступеня зв’язку класу належності сутностей РОЛЬ та КОРИСТУВАЧ показано на рисунку 2.6.

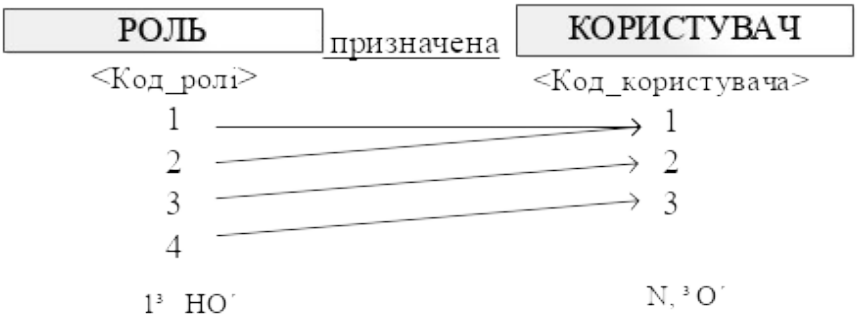


Рисунок 2.6 – Ступінь зв’язку та клас належності сутностей РОЛЬ та КОРИСТУВАЧ

Визначення ступеня зв’язку класу належності сутностей РОЛЬ та РІВЕНЬ_ДОСТУПУ показано на рисунку 2.7.

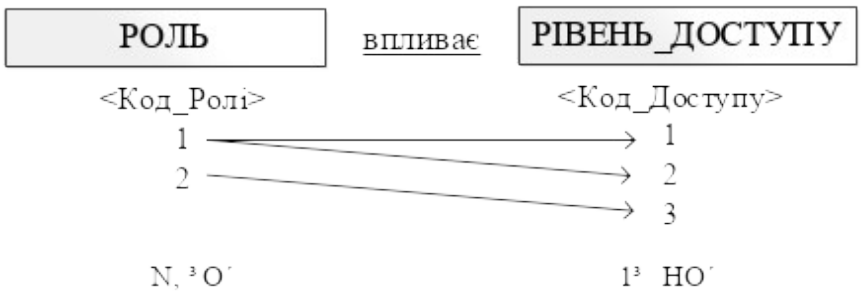


Рисунок 2.7 – Ступінь зв’язку та клас належності сутностей РОЛЬ та РІВЕНЬ_ДОСТУПУ

Проектування ER-моделі. Модель сутність-зв'язок — це модель даних, що дає змогу описувати концептуальні схеми предметної області і використовується при високорівневому проектуванні баз даних. Вона відображає логіку обміну інформацією у системі та відповідає реляційній моделі даних, тому дані для кожного типу об'єктів розміщуються окремо, між ними встановлюються зв'язки відповідного типу.

Отже, сутність — це об’єкт, що може бути ідентифікований якимось певним чином, який відрізнятиме його від інших об’єктів. Зв’язок — це об’єднання, встановлене між декількома сутностями.

Результатом буде ER-модель зображена на рисунку 2.8.

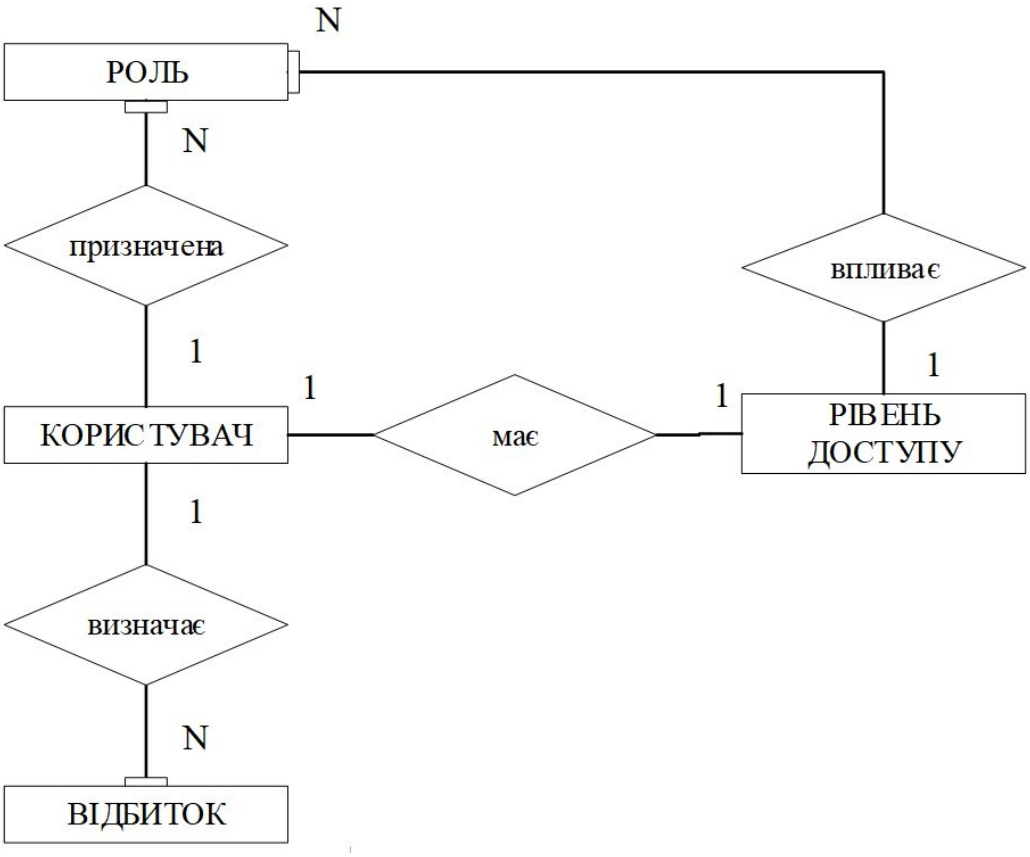


Рисунок 2.8 – ER-модель

Нормалізація БД. Нормалізація — поетапний оборотній процес заміни поточного відношення (або набору відношень) іншою сукупністю відношень, які вирізняються простішою та регулярнішою структурою. Апарат нормалізації відношень був розроблений Кодом, котрий виділив три нормальних форми представлення відношень. Кінцева мета нормалізації заключається в отриманні структури бази даних без надмірності інформації. Це важливо не так для того, аби зекономити використовувану пам’ять, а щоб усунути можливу спірність даних, що зберігаються.

Відповідно до методу нормалізації відношень першим етапом проектування є визначення інформаційних об’єктів.

Для системи ідентифікації користувачів було визначено такі інформаційні об’єкти: КОРИСТУВАЧ, РІВЕНЬ_ДОСТУПУ, ВІДБИТОК, РОЛЬ.

Визначено інформацію, необхідну при роботі з кожним обраним інформаційним об’єктом:

– КОРИСТУВАЧ (<Код_користувача>, Повне_ім’я, Телефон, Код_доступу, Код_відбитку, Код_ролі, Пароль);

– РІВЕНЬ_ДОСТУПУ (<Код_доступу>, Вид_доступу, Права);

– ВІДБИТОК (<Код_відбитку>, Зображення_відбитку);

– РОЛЬ (<Код_ролі>, Назва_ролі, Опис_ролі).

На основі проведеного аналізу, для реалізації поставленої задачі, потрібно скласти універсальне відношення, що матиме вигляд:

R: (Код_користувача, Повне_ім’я, Телефон, Код_доступу, Код_відбитку, Код_ролі, Пароль, Код_доступу, Вид_доступу, Права, Код_відбитку, Зображення_відбитку, Код_ролі, Назва_ролі, Опис_ролі).

Первинним ключем обрано: <Код_користувача, Код_відбитку>.

Якщо відношення знаходиться у першій нормальній формі, то усі не ключові атрибути є функціонально залежними від ключа первинного ключа.

1НФ:

R: <Код_користувача, Код_відбитку>, Повне_ім’я, Телефон, Код_доступу, Код_відбитку, Код_ролі, Пароль, Код_відбитку, Зображення_відбитку.

Якщо ж не ключовий атрибут залежить лише від частини обраного ключа, то це — часткова залежність. Відношення знаходиться в другій нормальній формі якщо: відношення у першій нормальній формі, а кожен не ключовий атрибут повно залежить від складеного ключа.

Щоб привести відношення до 2НФ, потрібно побудувати проекцію без атрибутів, що знаходяться в частковій транзитивній залежності від складеного ключа. Після чого — будується проекція на частину складеного ключа й атрибути, що від неї залежать.

Отже, із цих правил випливають такі відношення в 2НФ:

R1: <Код_користувача>, Повне_ім’я, Телефон, Код_доступу, Код_відбитку, Код_ролі, Пароль.

R2: <Код_доступу>, Вид_доступу, Права.

6) Відношення знаходиться в третій нормальній формі якщо: відношення у другій нормальній формі, транзитивні залежності не ключових атрибутів від ключа — відсутні.

У конкретно цьому варіанті транзитивна залежність існує у відношенні:

R3: <Код_відбитку>, Зображення_відбитку.

R4: <Код_ролі>, Назва_ролі, Опис_ролі

R5: <Код_доступу>, Вид_доступу.

R6: <Код_ролі>, Суть_ролі.

У результаті проектування за методом нормалізації кінцевими відношеннями будуть: R1, R2, R3, R4, R5, R6.

За створення системи безпеки відповідає СУБД. Мова SQL є основоположенням системи безпеки у реляційній системі керування базами даних: за допомогою інструкцій SQL визначаються вимоги, які висуваються до системи захисту інформації в БД.

Із захистом даних в SQL пов'язані три основні концепції [29]:

1. Діючими особами в базі даних є користувачі. Усі дії з даними СУБД робить від імені якогось користувача.

2. Об'єкти бази даних є тими елементами, чий захист може здійснюватися за допомогою SQL. Зазвичай забезпечується захист таблиць, але й інші об'єкти, такі як форми, прикладні програми й цілі бази даних, також можуть бути захищені.

3. Привілеї — це права користувача на проведення тих чи інших дій над певним об'єктом бази даних.

Для введення елементів системи безпеки застосовується інструкція, за допомогою якої тим чи іншим користувачам надаються певні привілеї на використання тих чи інших об'єктів. В інструкції задається комбінація ідентифікатора користувача, об'єкта і привілеїв.

Кожному користувачеві реляційної бази даних присвоюється ідентифікатор — відбиток пальця, що однозначно визначає користувача для СУБД. Ці ідентифікатори є основою системи безпеки.

2.4 Висновки до розділу 2

Актуальність розвитку в ІТ біометричної ідентифікації зумовлена збільшенням кількості об’єктів і потоків інформації, які необхідно оберігати від неправомірного втручання.

Статичні методи біометричної ідентифікації ґрунтуються на фізіологічній (статичній) характеристиці людини, тобто унікальній характеристиці, даній йому від народження і невід’ємною від нього. До переваг статичних методів відносяться малі витрати зусиль користувачів, а також мала залежність від їх психологічного стану.

Динамічні методи біометричної ідентифікації ґрунтуються на поведінковій (динамічній) характеристиці людини, тобто побудовані на особливостях, характерних для підсвідомих рухів у процесі відтворення якої-небудь дії. До переваг динамічної біометрії відносяться низька вартість, оскільки вона може бути побудована лише за допомогою програмного забезпечення (клавіатура, звукова карта, планшет, мобільний телефон − уже входять до складу апаратної частини), а також швидка зміна образу особистості за рахунок швидкої зміни відтвореного слова або фрази.

Застосування біометричних засобів не лише суттєво полегшує процедуру ідентифікації користувачів різних програмних засобів, а ще й підвищує рівень надійності систем безпеки. Користувачам смартфонів, планшетів та ноутбуків якнайкраще підійде саме ідентифікація за відбитками пальців, для підтвердження відповідності особи користувача.

Проаналізувавши переваги та недоліки методу ідентифікації на основі біометричних даних із використанням відбитків пальців та алгоритм захисту інформації в комп’ютерних мережах вирішено, що захист необхідно

виконувати на базі системного підходу, оскільки він зможе забезпечити необхідний рівень захищеності.

Було спроектовано базу даних користувачів за допомогою моделі «сутність-зв’язок». Проектування містило такі етапи визначення: сутностей, зв’язків, атрибутів, ключів сутностей, визначення ступеня зв`язку, класу належності.

Контроль за доступом до інформації здійснює сервер бази даних. При виконанні запиту користувача SQL-сервер отримує відбиток пальців, визначає ім'я користувача і за внутрішньою інформацією визначає, чи може ця особа виконати цей запит. Якщо користувач має таке право, то сервер обробляє цей запит, в іншому випадку — користувач отримує інформативне повідомлення про помилку. Яким чином зберігається інформація про привілеї — це внутрішня справа SQL-сервера.

Отже, у результаті дослідження було обґрунтовано біометричний метод за відбитками пальців як оптимальний підхід для захисту інформації в комп’ютерних мережах через смартфон.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАСОБУ ІДЕНТИФІКАЦІЇ КОРИСТУВАЧІВ

Останнім часом збільшується кількість загроз відносно комп'ютерної інформації, тому все більше уваги приділяється покращенню існуючих та розробці нових програмних засобів захисту інформаційних комп'ютерних систем від несанкціонованого доступу з боку неавторизованих користувачів. Незважаючи на активну діяльність протягом останніх років у напрямку розробки та вдосконалення методів ідентифікації користувачів для керування доступом до ресурсів інформаційних систем, надійність та стійкість існуючих засобів недостатня для потреб сьогодення. Отже, проблема розробки програмного модулю ідентифікації користувачів за відбитками пальців через смартфон є актуальною.

3.1 Обґрунтування вибору мови програмування

Існує кілька можливих підходів до вибору мови програмування. Для розробки мобільного додатку було обрано та використано типізовану мову програмування Kotlin.

Працюючи з Kotlin, генерується набагато менше коду, отже обслуговувати, підтримувати та тестувати доведеться мінімальний обсяг написаного. У результаті скоротиться час виходу продуктів на ринок, баг-фікси стануть швидшими і рідшими.

Переваги Kotlin [30]:

1) Покращена продуктивність. Послідовний та інтуїтивний синтаксис Kotlin гарантує підвищення продуктивності команд розробників. Для написання та розгортання програми потрібно значно менше часу й менше рядків коду. У результаті — готовий додаток можна отримати швидше.

2) 100% сумісності із Java. Методи Java можна викликати з Kotlin, що є суттєвою перевагою не лише для розробників, але й для компаній, що мають велику кодову базу на Java.

3) Легкість підтримки. Android-розробникам досить легко підтримувати код на Kotlin, адже ця мова підтримується багатьма середовищами розробки, в тому числі Android studio та декілька інших SDK. Розробники мають можливість працювати з будь-яким звичним для них набором інструментів.

4) Надійність. Остання версія Kotlin має зворотну сумісність з усіма попередніми версіями. Це означає, що Android-розробникам не потрібно працювати з різними версіями.

5) Легкість вивчення. Kotlin вивчати легше, ніж Java, оскільки він не вимагає особливих знань в сфері розробки мобільних додатків.

6) Дає змогу писати модульні програми.

7) Підтримка Android Studio. Android Studio забезпечує розширеною підтримкою Kotlin та інструментами адаптації. Також, розробники мають можливість працювати одночасно на Kotlin і на Java.

Разом із тим, для розробки другої частини системи, а саме — PC App і деякого налаштування між цими двома частинами, було використано мову програмування Java. Java є ключовим елементом в Android − мобільній операційній системі з відкритим вихідним кодом [31]. Хоча Android, побудований на ядрі Linux, написаний в основному на C, Android SDK використовує мову Java в якості основи для додатків Android. Мова байт-коду, підтримувана Android SDK, несумісна з байт-кодом Java і працює на власній віртуальній машині, оптимізованій для пристроїв із малою пам'яттю, таких як смартфони та планшетні комп'ютери. Залежно від версії Android байт-код інтерпретується або віртуальною машиною Dalvik, або скомпільований у власний код за допомогою Android Runtime.

Основна ціль створення цієї мови заключалася в тому, щоб зробити її переносною та безпечною мовою програмування. Крім цього, також є деякі

дивовижні функції, які мають велике значення в гласності цієї мови. Представлені нижче функції прості та зрозумілі.

Переваги Java:

1. Об'єктно-орієнтована: на Java все є об'єктом. Java можна легко розширити, оскільки вона заснована на об'єктній моделі. Програмне забезпечення можна організувати як комбінацію різних типів об'єктів, що включає дані та поведінку.

Об'єктно-орієнтоване програмування (ООП) – це методологія, яка спрощує розробку та супровід програмного забезпечення, надаючи деякі правила.

2. Незалежна від платформи: на відміну від багатьох інших мов програмування, включаючи C і C++, Java компілюється не в платформі конкретної машини, а в незалежному байт-коді. Цей байтовий код поширюється по мережі й інтерпретується JVM (віртуальною машиною), на якій працює.

Платформа – це апаратне або програмне середовище, в якому запускається програма. Існує два типи платформ, заснованих на програмного та апаратному забезпеченні. Java представляє програмну платформу, яка відрізняється від більшості інших платформ тим, що це програмна платформа, яка працює на вершині інших апаратних платформ. Вона складається з двох компонентів: середовище виконання та API (інтерфейс прикладного програмування).

3. Проста: Java призначена для простого вивчення. У ній видалено більшість заплутаних і/або рідко використовуваних функцій (явні вказівники, перезавантаження оператора). Не треба видаляти непотрібні об’єкти, оскільки є автоматична збірка сміття.

4. Безпечна: із захищеною функцією. Java дозволяє розробляти безвірусні системи без втручання користувача. Методи аутентифікації засновані на шифруванні з відкритим ключем. Програми Java запускаються в ізольованому програмному середовищі віртуальної машини. Classloader додає безпеку,

відділяючи пакет від класів локальної файлової системи від тих, які імпортуються з мережевих джерел. Bytecode Verifier перевіряє фрагменти коду для незаконного коду, який може порушувати права доступу до об’єктів. Security Manager визначає, які ресурси може отримувати клас, наприклад читання та запис на локальний диск.

5. Архітектурно-нейтральна: Java-компілятор генерує нейтральний по архітектурі об'єктний формат файлу, що робить скомпільований код виконуваним на багатьох процесорах із системою часу виконання Java.

У С-програмуванні тип даних int займає 2 байта пам'яті для 32-бітної архітектури і 4 байта пам'яті для 64-бітної архітектури. Але в Java він займає 4 байта пам'яті і для 32-, і для 64-бітних архітектур.

6. Портативна: будучи нейтральним по архітектурі та незалежним від реалізації аспектом специфікації, робить Java переносною, що полегшує перенесення байт-коду Java на будь-яку платформу.

7. Надійна: Java робить усе можливе, щоб унеможливити ситуації, пов'язані з помилками, опираючись переважно на перевірку часу виконання компіляції. Вона використовує потужне керування пам’яттю.

8. Багатопотокова: з цією функцією Java можна писати програми, що виконуватимуть одночасно безліч завдань. Ця конструктивна особливість дає змогу розробникам створювати інтерактивні додатки, які можуть працювати плавно. Основною перевагою є те, що вона має спільну область пам’яті, а не займає пам’ять для кожного потоку окремо.

9. Інтерпретована: код байта Java перетвориться «на льоту» у власні машинні інструкції та ніде не зберігається. Процес розробки є більш швидким і аналітичним, оскільки зв'язування є інкрементним і легким процесом.

10. Високопродуктивна: завдяки використанню компіляторів Just-In-Time Java забезпечує високу продуктивність.

11. Розподілена: Java призначена для розподіленого середовища Інтернету. Це полегшує створення розподілених додатків в Java, із використанням RMI та EJB.

12. Динамічна: Java вважається більш динамічною, ніж C або C ++, оскільки призначена для адаптації до змінного середовища. Java-програми можуть містити інформацію про час виконання, яка використовується для перевірки та дозволу доступу до об'єктів під час виконання.

Оскільки Java спочатку призначалася для мобільних пристроїв, що обмінюватимуться даними через мережу, вона була створена для забезпечення високого рівня безпеки. Додатки та програми, написані на цій мові, вирізняються високим ступенем надійності, оскільки в мові повністю відсутні механізми, які часто призводять до помилок у роботі програми, наявний строгий контроль для запобігання різних ситуацій. Мова повністю безпечна, а тому є настільки застосовуваною в сфері програмування та розробки ефективних програм. Завдяки можливості реалізовуватися різними методами, виходять різноманітні типи високофункціональних додатків.

Отже, мови Kotlin та Java є оптимальним засобом для реалізації нашого програмного продукту.

3.2 Обгрунтування вибору середовища розробки

Android Studio — офіційне інтегроване середовищем розробки (IDE) для розробки додатків для Android, заснованих на IntelliJ IDEA. На додаток до потужного редактору та інструментам розробника IntelliJ, Android Studio пропонує ще більше можливостей, що підвищують продуктивність при створенні додатків для Android, таких як:

- гнучка система збірки на основі Gradle;
- швидкий і багатофункціональний емулятор;
- єдине середовище, у якому можна розробляти для всіх Android-пристроїв;

- миттєвий запуск для внесення змін у робочому додатку без створення нового APK;

- шаблони коду й інтеграція GitHub, щоб допомогти створювати спільні функції додатку та імпортувати приклад коду;

- обширні інструменти та засоби тестування;

- інструменти Lint для відстеження продуктивності, зручності використання, сумісності версій і т.п.;

- підтримка C ++ та NDK;

- вбудована підтримка хмарної платформи Google, що дозволяє легко інтегрувати Google Cloud Messaging і App Engine.

SDK розшифровується як комплект для розробки програмного забезпечення або, інакше кажучи, dekkit. Це набір програмних засобів та програм, які розробники використовують для створення додатків для певних платформ [32]. Інструменти SDK складаються з цілого ряду речей, що включає: бібліотеки, документацію, зразки коду, процеси та керівництва, які розробники можуть використовувати та інтегрувати у власні програми.

Є декілька чітких переваг щодо забезпечення завантаження SDK для розроблюваної програми:

- Більш швидка інтеграція — коротші цикли продажів. Якщо потрібно закрити чим більше угод, мобільний SDK прискорює їх. devkit допоможе скоротити цикл продажів, оскільки це значно спростить інтеграцію з існуючим набором клієнтів.

- Ефективний розвиток — швидше розгортання. Якщо взяти до уваги той факт, що середній додаток для Android використовуватиме близько 18,2 сторонніх пакетів), стає зрозуміло, що жоден розробник програмного забезпечення не має часу кодувати кожен окремий інструмент з нуля. Можна почати з перевірки інструментарію SDK для Android SD, щоб знайи код, який працював би на пристрої Android. Це прискорить розгортання, не потребуючи кодування з порожнього аркуша. Таку ж ефективність можна надавати іншим розробникам, коли створюється SDK для свого продукту, який вони можуть використовувати для з’єднання мого продукту зі своїми програмами.

- Чудовий продукт + SDK — збільшений обсяг. Якщо розроблюваний продукт є цінним і супровідний пакет SDK дозволяє отримати велику сумісність, можна збільшити шанси на те, що інші інструменти захочуть інтегруватися з цим продуктом, що призведе до більшого попиту.

- Керування системою — пом'якшені ризики. За допомогою SDK забезпечується кращий контроль над елементами інтерфейсу користувача, які відображаються в інших додатках. Це дозволяє диктувати не тільки те, як продукт інтегрується з іншими програмами, а й те, як він виглядає. Усі найважливіші функції захищені від підробки та можливих збоїв з вини користувача.

Найбільш важливим інструментом є емулятор мобільного пристрою, проте до SDK входять й інші інструменти (для налагодження, упаковки та інсталяції додатків на емулятор).

Структура проекту. Кожний проект в Android Studio містить один або кілька модулів з файлами вихідного коду та файлами ресурсів. Типи модулів включають:

- модулі додатків для Android;
- бібліотечні модулі;
- модулі Google App Engine.

За замовчуванням Android Studio відображає файли проектів у поданні проекту Android, яке організовано модулями для швидкого доступу до вихідних файлів.

Усі файли збірки видно на верхньому рівні під Gradle Scripts, і кожний модуль додатку містить наступні папки:

1) manifestests: містить AndroidManifest.xml файл.

2) java: містить файли вихідного коду Java, включаючи тестовий код JUnit.

3) res: містить усі ресурси без коду, такі як XML-макети, строки користувацького інтерфейсу та растрові зображення.

Головне вікно Android Studio складається з наступних логічних областей:

- панель інструментів, яка дозволяє виконувати широкий спектр дій, в тому числі запуск додатку та інструментів Android;

- панель навігації, яка дозволяє переміщуватися по проекту і відкривати файли для редагування. Вона забезпечує компактніший вигляд структури, видимої у вікні проекту;

- у вікні редактора можна створювати та редагувати код. В залежності від поточного типу файлу редактор може змінитися;

- вікно панелі інструментів працює зовні вікна IDE і містить кнопки, які дозволяють розгорнути або згорнути окремі вікна інструментів;

- у вікні інструментів надається доступ до конкретних завдань (управління проектами, пошук, контроль версій і багато чому іншому);

- рядок стану відображає стан вашого проекту і саме IDE, а також будь-які попередження або повідомлення.

Вікна інструментів. Замість використання встановлених перспектив Android Studio слідує вашому контексту і автоматично створює відповідні вікна інструментів під час роботи. За замовчуванням найчастіше використовувані вікна інструментів прикріплюються до панелі інструментів вікна по краях.

Завершення коду. Android Studio має три типи завершення коду, до яких ви можете отримати доступ:

1. Базова комплектація – відображає основні пропозиції для змінних, типів, методів, виразів і т.п. Якщо викликати базове завершення двічі поспіль, ви побачите більше результатів, включаючи приватних членів і не імпортованих статичних членів.

2. Інтелектуальне завершення – відображає відповідні параметри на основі контексту. Розумне завершення знає про очікуваний тип і потоки даних. Якщо викликати Smart Completion двічі поспіль, то можна побачити включаючи ланцюжок.

3. Завершення заяви – завершує поточний оператор, додаючи відсутні дужки, дужки, фігурні дужки, форматування і т.п.

Стиль і форматування. При редагуванні Android Studio автоматично застосовує форматування і стилі, які зазначено в параметрах стилю коду, який можна налаштувати за допомогою мови програмування, включаючи вказівку умовностей для вкладок і відступів, прогалин, обгортання, фігурних дужок і порожніх рядків.

Основи управління версіями. Android Studio підтримує різні системи управління версіями (VCS), включаючи Git, GitHub, CVS, Mercurial, Subversion і Google Cloud Source Repositories. Після імпорту додатку в Android Studio використовуйте параметри меню Android Studio VCS, щоб включити підтримку VCS для системи управління необхідної версією, створити репозиторій, імпортувати нові файли в систему управління версіями та виконати інші операції.

Система збірки. Android Studio використовує Gradle в якості основи для системи збирання, з більш специфічними для Android можливостями, наданими плагіном Android для Gradle. Ця система збірки працює як інтегрований інструмент з меню Android Studio і незалежна від командного рядка. Функції системи збірки можна використовувати, щоб зробити наступне:

- налаштувати та/або розширити процес складання;

- створити кілька APK для свого додатку, використовуючи різні функції, той же проект і модулі;

- повторно використовувати код та ресурси в наборах джерел.

Використовуючи гнучкість Gradle, можна досягти всього цього, не змінюючи основні вихідні файли програми. Названо файли збірки Android Studio build.gradle. Це прості текстові файли, які використовують синтаксис Groovy для налаштування збірки з елементами, які надаються плагіном Android для Gradle. Кожен проект має один файл збірки верхнього рівня для всього проекту й окремі файли збірки на рівні модуля для кожного модуля. При імпорті існуючого проекту Android Studio автоматично генерує необхідні файли збірки.

Підтримка декількох APK. Дозволяє ефективно створювати кілька APK на основі щільності екрану або ABI. Наприклад, можна створювати окремі APK додатки для щільності екрану hdpi і mdpi, все ще розглядаючи їх як один варіант і дозволяючи їм ділитися параметрами APK, javac, dx і ProGuard.

Скорочення ресурсів. Скорочення ресурсів в Android Studio автоматично видаляє невживані ресурси з ваших залежних додатків і бібліотек. Наприклад, якщо додаток використовує служби Google Play для доступу до функцій Google Drive, і в даний час ви не використовуєте Google Sign-In, то скорочення ресурсів може видалити різні доступні ресурси для SignInButton кнопок.

Управління залежностями. Залежності для проекту вказані у назві в build.gradleфайле. Gradle піклується про їх пошук і робить доступними у збірці. Можна оголошувати залежності модуля, віддалені виконавчі та локальні виконавчі залежності в build.gradle файлі. Android Studio налаштовує проекти для використання центрального сховища Maven за замовчуванням (ця конфігурація включена у файл збірки верхнього рівня для проекту).

Інструменти налагодження і профілів. Android Studio допомагає налагоджувати і поліпшувати продуктивність, в тому числі вбудовані засоби налагодження і аналізу продуктивності.

Вбудоване налагодження. Використовується для поліпшення наскрізних кодів в поданні налагоджувальника із вбудованою перевіркою посилань, виразів і значень змінних. Вбудована налагоджувальна інформація включає:

- вбудовані значення змінних;
- об’єкти, які посилаються на обраний об'єкт;
- значення, що повертаються методу;
- лямбда і операторні вирази;
- значення підказки.

Профілювання продуктивності. Можна легше здійснювати контроль за використанням пам'яті додатку та процесора, знаходити звільнені об'єкти, знаходити витоки пам'яті, оптимізувати продуктивність графіки й аналізувати

мережеві запити. Потрібно відкрити додаток Android Profiler та запустити на пристрої або емуляторі.

Збірка сміття. При профілюванні використання пам'яті в Android Studio, можна одночасно ініціювати складання сміття і вивантажувати купу Java в моментальний знімок купи в двійковому файлі формату HPROF для Android. Засіб перегляду HPROF відображає класи, екземпляри кожного класу і дерево посилань, щоб допомогти відстежувати використання пам'яті та знаходити її витоки.

Профайлер пам'яті. Можна використовувати Memory Profiler для відстеження виділення пам'яті та перегляду об'єктів, які виділяються при виконанні певних дій. Знання цих розподілів дозволяє оптимізувати продуктивність додатку та використання пам'яті, налаштувавши виклики методів, пов'язані з цими діями.

Доступ до файлів даних. Інструменти Android SDK, такі як Systrace та logcat, генерують продуктивність та інформацію про налагодження для детального аналізу додатків.

Перевірка коду. Кожного разу при компілюванні програми Android Studio автоматично запускає налаштовані перевірки Lint та інших IDE, щоб допомогти легко ідентифікувати та усунути проблеми зі структурною якістю коду. Інструмент Lint перевіряє вихідні файли проекту Android на можливі помилки та покращує оптимізацію для правильності, безпеки, продуктивності, зручності використання, доступності та інтернаціоналізації. На додаток до перевірок Lint, Android Studio також виконує перевірки коду IntelliJ та перевіряє анотації для оптимізації робочого процесу кодування.

Анотації в Android Studio. Android Studio підтримує анотації для змінних, параметрів і значень, щоб допомогти зловити помилки, такі як виключення нульового покажчика та конфлікти типів ресурсів. Android SDK Manager упаковує бібліотеку Support-Annotations у репозиторий підтримки Android для використання з Android Studio. Android Studio перевіряє налаштовані анотації під час перевірки коду.

Журнальні повідомлення. При створенні та запуску додатку за допомогою Android Studio, можна переглядати повідомлення виходу adb і повідомлення журналу пристрою у вікні Logcat.

3.3 Android Fingerprint API

API дозволяє користувачеві аутентифікуватися за допомогою свого відбитку. Для роботи з сенсором API пропонує FingerprintManager, що є досить простим в освоєнні.

Щоб почати отримувати профіт від нового API, насамперед потрібно додати permission в маніфесті:

```
<uses-permission android:name="android.permission.USE_FINGERPRINT"/>
```

Використовувати Fingerprint API можна тільки на пристроях, які його підтримують: відповідно, це пристрої Android 6+ із сенсором.

Сумісність можна легко перевірити за допомогою методу:

```
public static boolean checkFingerprintCompatibility(@NonNull Context context) {
    return FingerprintManagerCompat.from(context).isHardwareDetected();
}
```

FingerprintManagerCompat – це зручна обгортка для звичайного FingerprintManager'a, яка спрощує перевірку пристрою на сумісність, інкапсулюючи в собі перевірку версії API. В даному випадку, isHardwareDetected () поверне false, якщо API нижче 23.

Перед початком розробки додатків для Android корисно зрозуміти загальний підхід платформи до управління зміною API. Також важливо зрозуміти Android API Level (Ідентифікатор рівня API) і його роль у забезпеченні сумісності вашого застосування з пристроями, на яких воно буде встановлюватися.

Рівень API – цілочисельне значення, яке однозначно визначає версію API платформи Android. Платформа забезпечує структури API, які додатки можуть

використовувати для взаємодії з системою Android. Будь-яка наступна версія Android може містити оновлення API.

Оновлення API-структури розроблені так, щоб новий API залишався сумісним з більш ранніми версіями API. Таким чином, більшість змін в API є сукупною і вводить нові функціональні можливості або виправляє попередні. Оскільки частина API постійно оновлюється, застарілі API не рекомендуються до використання, але не видаляються з міркувань сумісності з наявними додатками.

Рівень API, який використовує додаток для Android, визначається цілочисловим ідентифікатором, який вказується у файлі конфігурації кожного Android-додатки. У таблиці 3.1 наведено відповідність рівня API і версії платформи Android.

Таблиця 3.1 — Відповідність версії платформи та рівня API

Версія платформи	Назва версії	Рівень API
Android 10	Q	29
Android 9	Pie	28
Android 8	Oreo	26
Android 6.0	Marshmallow	23
Android 5.0	Lollipop	21
Android 4.3	Jelly bean	19
Android 1.6	Donut	4
Android 1.5	Cake	3
Android 1.0	Apple Pie	1

Далі, потрібно зрозуміти, чи готовий сенсор до використання. Для цього визначимо enum станів:

```
public enum SensorState {
   NOT_SUPPORTED,
   NOT_BLOCKED, // якщо пристрій не захищено піном, малюнком або паролем
   NO_FINGERPRINTS, // якщо на пристрої нема відбитків
   READY
}
```

І скористаємося методом:

```
public static SensorState checkSensorState(@NonNull Context context) {
   if (checkFingerprintCompatibility(context)) {
      KeyguardManager =
            (KeyguardManager) context.getSystemService(KEYGUARD_SERVICE);
      if (!keyguardManager.isKeyguardSecure()) {
         return SensorState.NOT_BLOCKED;
      }
      FingerprintManagerCompat fingerprintManager =
FingerprintManagerCompat.from(context);
      if (!fingerprintManager.hasEnrolledFingerprints()) {
         return SensorState.NO_FINGERPRINTS;
      }
      return SensorState.READY;
   } else {
      return SensorState.NOT_SUPPORTED;
   }
}
```

Код використовується досить тривіальний. Невелике непорозуміння може викликати момент, коли ми перевіряємо чи заблоковано пристрій. Нам потрібна ця перевірка, так як, хоч Android і не дозволяє додавати відбитки у незахищений пристрій, деякі виробники це обходять, тому підстрахуватися не завадить.

Різні стани можна використовувати для того, щоб дати користувачеві зрозуміти, що відбувається та направити його на шлях істинний.

Підготовка. Отже, не зациклюючись на перевірці пін-коду на валідність, прикинемо наступну спрощену логіку дій:

- користувач вводить пін-код, якщо SensorState.READY − зберігаємо пін-код і запускаємо MainActivity;

- перезапускаємо додаток, якщо SensorState.READY − зчитуємо відбиток, дістаємо пін-код, імітуємо його введення, запускаємо MainActivity.

Схема була б досить простою, якби не одне але: Google наполегливо рекомендує не зберігати приватні дані користувача у відкритому вигляді. Тому потрібен механізм шифрування та розшифрування для, відповідно, збереження та використання.

Для шифрування та розшифрування потрібно:

- захищене сховище для ключів;

- криптографічний ключ;
- шифрувальник.

Сховище. Для роботи з відбитками система надає нам свій кейстор – «AndroidKeyStore» і гарантує захист від несанкціонованого доступу.

Скористаємося ним:

```
private static KeyStore sKeyStore;
private static boolean getKeyStore() {
   try {
      sKeyStore = KeyStore.getInstance("AndroidKeyStore");
      sKeyStore.load(null);
      return true;
   } catch (KeyStoreException | IOException | NoSuchAlgorithmException |
CertificateException e) {
      e.printStackTrace();
   }
   return false;
}
```

Слід зрозуміти, що кейстор зберігає тільки криптографічні ключі. Паролі, пін і інші приватні дані там зберігати не можна.

Ключ. На вибір у нас два варіанти ключів: симетричний ключ і пара з публічного і приватного ключа. З міркувань UX скористаємося парою. Це дозволить відокремити введення відбитка від шифрування пін-коду.

Ключі будемо діставати з кейстора, але спочатку потрібно їх туди покласти. Для створення ключа потрібно скористатися генератором.

```
private static KeyPairGenerator sKeyPairGenerator;
private static boolean getKeyPairGenerator() {
   try {
      sKeyPairGenerator =
KeyPairGenerator.getInstance(KeyProperties.KEY_ALGORITHM_RSA,
"AndroidKeyStore");
      return true;
   } catch (NoSuchAlgorithmException | NoSuchProviderException e) {
      e.printStackTrace();
   }
   return false;
}
```

При ініціалізації вказуємо, в який кейстор підуть згенеровані ключі і для якого алгоритму призначений цей ключ.

Сама ж генерація відбувається наступним чином:

```
private static boolean generateNewKey() {
   if (getKeyPairGenerator()) {
      try {
         sKeyPairGenerator.initialize(new KeyGenParameterSpec.Builder(KEY_ALIAS,
            KeyProperties.PURPOSE_ENCRYPT | KeyProperties.PURPOSE_DECRYPT)
               .setDigests(KeyProperties.DIGEST_SHA256,
KeyProperties.DIGEST_SHA512)
               .setEncryptionPaddings(KeyProperties.ENCRYPTION_PADDING_RSA_OAE
P)
               .setUserAuthenticationRequired(true)
               .build());
         sKeyPairGenerator.generateKeyPair();
         return true;
      } catch (InvalidAlgorithmParameterException e) {
         e.printStackTrace();
      }
   }
   return false;
}
```

Тут необхідно звернути увагу на два місця:

KEY_ALIAS — це псевдонім ключа, по якому витягатимемо його з кейстора, звичайний psfs.

.setUserAuthenticationRequired(true) — цей прапор вказує, що кожен раз, коли нам потрібно буде скористатися ключем, потрібно буде підтвердити себе, в нашому випадку − за допомогою відбитка.

Перевіряти наявність ключа будемо наступним чином:

```
private static boolean isKeyReady() {
   try {
      return sKeyStore.containsAlias(KEY_ALIAS) || generateNewKey();
   } catch (KeyStoreException e) {
      e.printStackTrace();
   }
   return false;
}
```

Шифрувальник. Шифруванням та дешифруванням в Java займається об’єкт Cipher. Для цього потрібно ініціалізувати його:

```
private static Cipher sCipher;
private static boolean getCipher() {
   try {
      sCipher = Cipher.getInstance("RSA/ECB/OAEPWithSHA-256AndMGF1Padding");
```

```
            return true;
        } catch (NoSuchAlgorithmException | NoSuchPaddingException e) {
            e.printStackTrace();
        }
        return false;
    }
```

Після того, як його проініціалізовано та отримано, необхідно підготувати його до роботи. При генерації ключа вказано, що будемо використовувати його тільки для шифрування та розшифрування. Відповідно, Cipher також буде для цих цілей:

```
    private static boolean initCipher(int mode) {
        try {
            sKeyStore.load(null);
            switch (mode) {
                case Cipher.ENCRYPT_MODE:
                    initEncodeCipher(mode);
                    break;
                case Cipher.DECRYPT_MODE:
                    initDecodeCipher(mode);
                    break;
                default:
                    return false; //this cipher is only for encode\decode
                }
            return true;
        } catch (KeyPermanentlyInvalidatedException exception) {
            deleteInvalidKey();
        } catch (KeyStoreException | CertificateException | UnrecoverableKeyException |
    IOException |
                    NoSuchAlgorithmException | InvalidKeyException | InvalidKeySpecException |
    InvalidAlgorithmParameterException e) {
            e.printStackTrace();
        }
        return false;
    }
```

де initDecodeCipher() та initEncodeCiper() наступні:

```
    private static void initDecodeCipher(int mode) throws KeyStoreException,
    NoSuchAlgorithmException, UnrecoverableKeyException, InvalidKeyException {
        PrivateKey key = (PrivateKey) sKeyStore.getKey(KEY_ALIAS, null);
        sCipher.init(mode, key);
    }

    private static void initEncodeCipher(int mode) throws KeyStoreException,
    InvalidKeySpecException, NoSuchAlgorithmException, InvalidKeyException,
    InvalidAlgorithmParameterException {
        PublicKey key = sKeyStore.getCertificate(KEY_ALIAS).getPublicKey();
```

```
    PublicKey unrestricted =
KeyFactory.getInstance(key.getAlgorithm()).generatePublic(new
X509EncodedKeySpec(key.getEncoded()));
    OAEPParameterSpec spec = new OAEPParameterSpec("SHA-256", "MGF1",
MGF1ParameterSpec.SHA1, PSource.PSpecified.DEFAULT);
    sCipher.init(mode, unrestricted, spec);
}
```

Помітно, що зашифровуючий Cipher трохи складніше форматувати. Суть у тому, що публічний ключ вимагає підтвердження користувача.

Момент з KeyPermanentlyInvalidatedException — якщо з якоїсь причини ключ не можна використовувати, вибиває цей виняток. Можливі причини – додавання нового відбитка до існуючого, зміна або повне видалення блокування. Тоді ключ більше немає сенсу зберігати і він видаляється:

```
public static void deleteInvalidKey() {
    if (getKeyStore()) {
        try {
            sKeyStore.deleteEntry(KEY_ALIAS);
        } catch (KeyStoreException e) {
            e.printStackTrace();
        }
    }
}
```

Шифрування та розшифрування. Використовується метод, який зашифровує строку аргумент. У результаті чого отримуємо Base64-строку, яку можна спокійно зберігати в преференсах додатку.

Метод розшифрування на вхід отримує не тільки зашифровану строку, але й об’єкт Cipher. Для того, щоб нарешті використовувати сенсор, необхідно скористуватися методом FingerprintManagerCompat.

Хендлер і прапори зараз не потрібні, сигнал використовується, щоб скасувати режим зчитування відбитків (при згортанні додатка, наприклад), зворотні виклики повертають результат конкретного зчитування, а ось над криптооб’єктом зупинимося детальніше.

CryptoObject у даному випадку використовується як обгортка для Cipher'a. Криптооб’єкт створюється з розшифровуючого Cipher. Якщо цей Cipher прямо зараз відправляє в метод decode (), то вилетить виключення,

повідомляючи про те, що ми намагаємося використати ключ без підтвердження. Строго кажучи, ми створюємо криптооб'єкт і відправляємо його на вхід authenticate() як раз для отримання цього підтвердження. Якщо getCryptoObject () повернуло нуль, то це означає, що при ініціалізації Chiper'a відбулося KeyPermanentlyInvalidatedException. Тут вже нічого не зробиш, крім як дати користувачеві знати, що вхід по відбитку недоступний і йому прийдеться знов ввести пін-код.

Результати зчитування сенсора отримуються в методах зворотніх викликів:

```
@Override
public void onAuthenticationHelp(int helpCode, CharSequence helpString) {
   //брудні пальці, недостатньо сильний зажим
}
@Override
public void onAuthenticationFailed() {
   //відбиток зчитався, але не розпізнався
}
@Override
public void onAuthenticationError(int errorCode, CharSequence errString) {
   //декілька невдалих спроб зчитування (5)
   //після цього сенсор стане недоступним на деякий час (30 сек)
}
@Override
public void onAuthenticationSucceeded(@NonNull
FingerprintManagerCompat.AuthenticationResult result) {
   //все прошло успішно
}
```

У випадку успішного розпізнавання ми отримуємо AuthenticationResult, з якого ми можемо достати об'єкт Cipher із вже підтвердженим ключем:

```
result.getCryptoObject().getCipher()
```

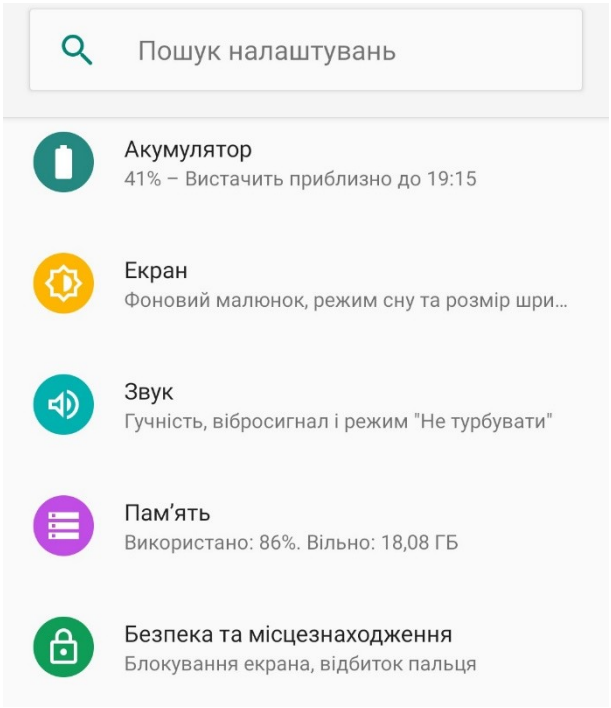
Тепер можна відправити його на вхід в decode(), отримати пін-код, зімітувати його введення та показати користувачу його дані.

3.4 Розробка інструкції користувача

3.4.1 Реєстрація нового користувача

Розглянемо як відбувається налаштування вбудованого у смартфон сканеру відбитків пальців на прикладі Xiaomi Mi A2.

1) Зайшовши в меню «Налаштування» (рис. 3.1) необхідно зі списку вибрати вкладку «Безпека та місцезнаходження».



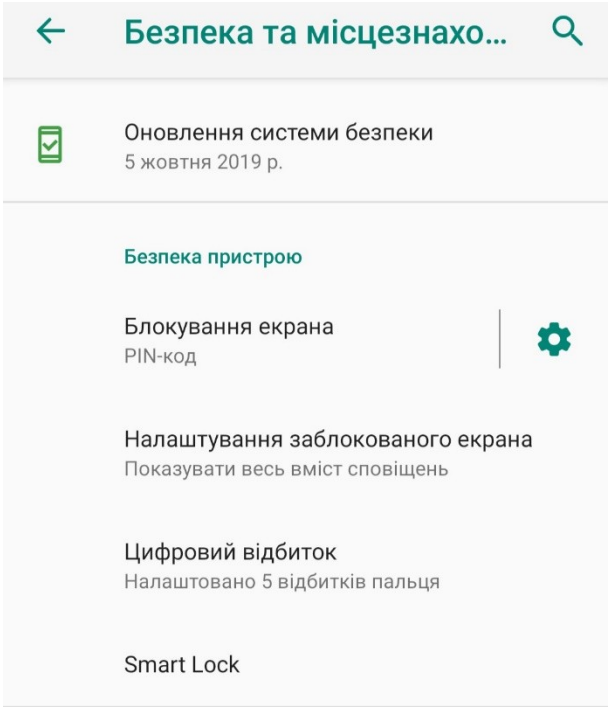


Рисунок 3.1 – Вигляд меню «Налаштування» та Рисунок 3.2 – Вигляд вкладки «Безпека та місцезнаходження»

2) Далі натискаємо на вкладку «Цифровий відбиток» (рис. 3.2).

3) У новому відкритому меню (рис. 3.3) звертаємо увагу на останній пункт у вікні – «Додати цифровий відбиток». Але варто зазначити, що даний смартфон зберігає лише до 5 відбитків пальців (рис. 3.4). Якщо виникла необхідність додати інший відбиток, то потрібно звільнити для нього місце, видаливши один із вже існуючих.

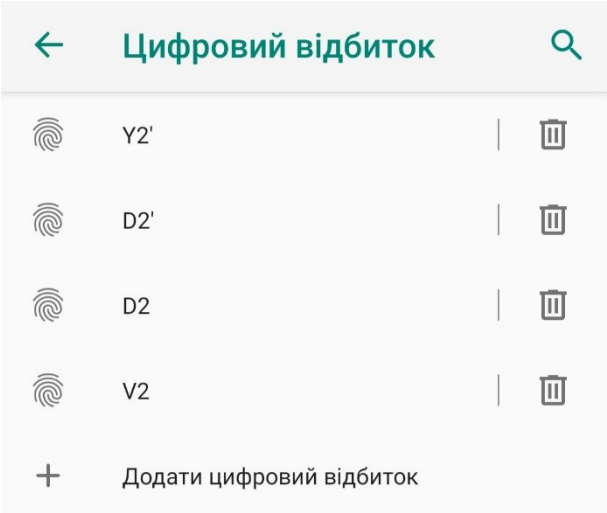


Рисунок 3.3 – Вигляд вкладки «Керувати відбитками пальців»

Рисунок 3.4 – Вигляд інформативного вікна перевищення кількості відбитків

4) На екрані смартфону з’являється повідомлення із подальшими вказівками для користувача (рис. 3.5).

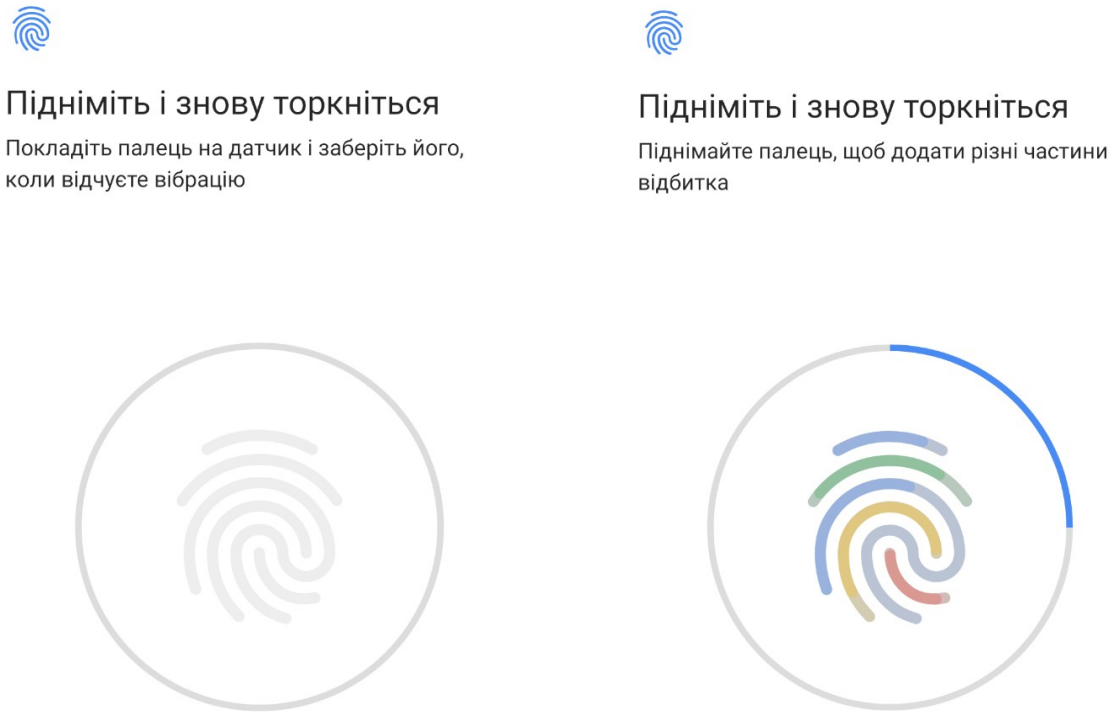


Рисунок 3.5 – Вигляд вікна з подальшими вказівками та Рисунок 3.6 – Вигляд вікна сканування

5) Багатократно прикладаємо будь-який палець (один і той самий) до поверхні сканеру вашого смартфону. Про успішність кожного етапу сканування вам повідомить анімація з заповненням кружка (рис. 3.6) та легка вібрація пристрою. Ці дії потрібно повторювати до тих пір, поки не заповняться усі пусті лінії в кружку.

6) Якщо все зроблено правильно – ви отримуєте сповіщення про успішне налаштування (рис. 3.7). Після чого, ваш пристрій готовий до подальшої роботи.

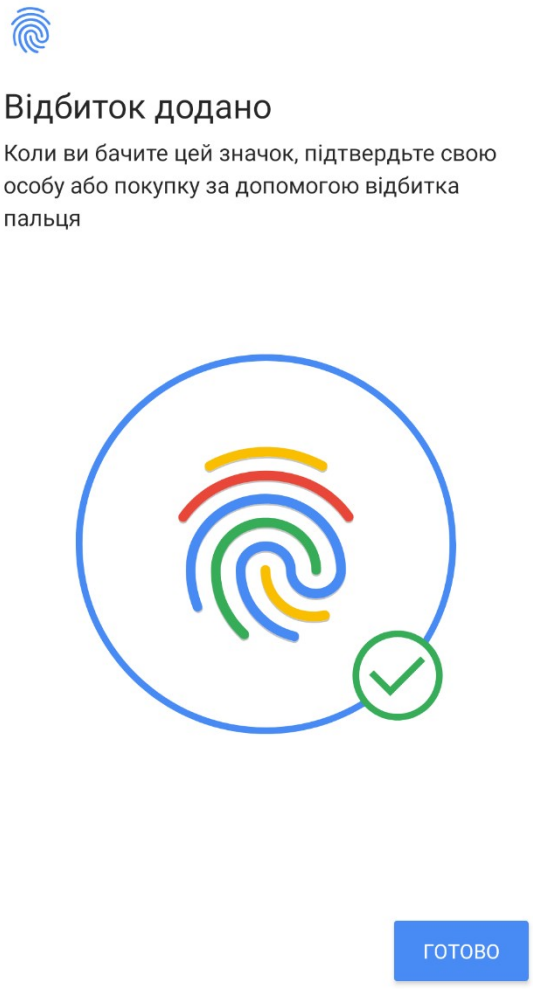


Рисунок 3.7 – Вигляд сповіщення про успішне налаштування

7) Наступним кроком буде встановлення додатку для ідентифікації користувачів за відбитками пальців через смартфон (рис. 3.8).

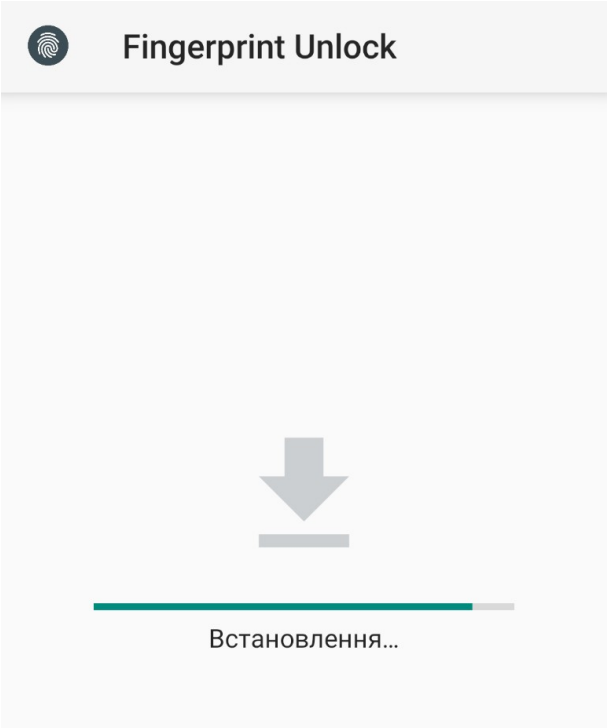


Рисунок 3.8 – Вигляд вікна встановлення додатку

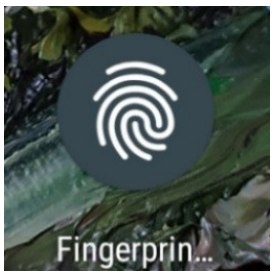


Рисунок 3.9 – Вигляд іконки встановленого додатку

Тепер стосовно розробленого додатку. Вигляд головного екрану відразу ж після встановлення додатку зображено на рисунку 3.10.

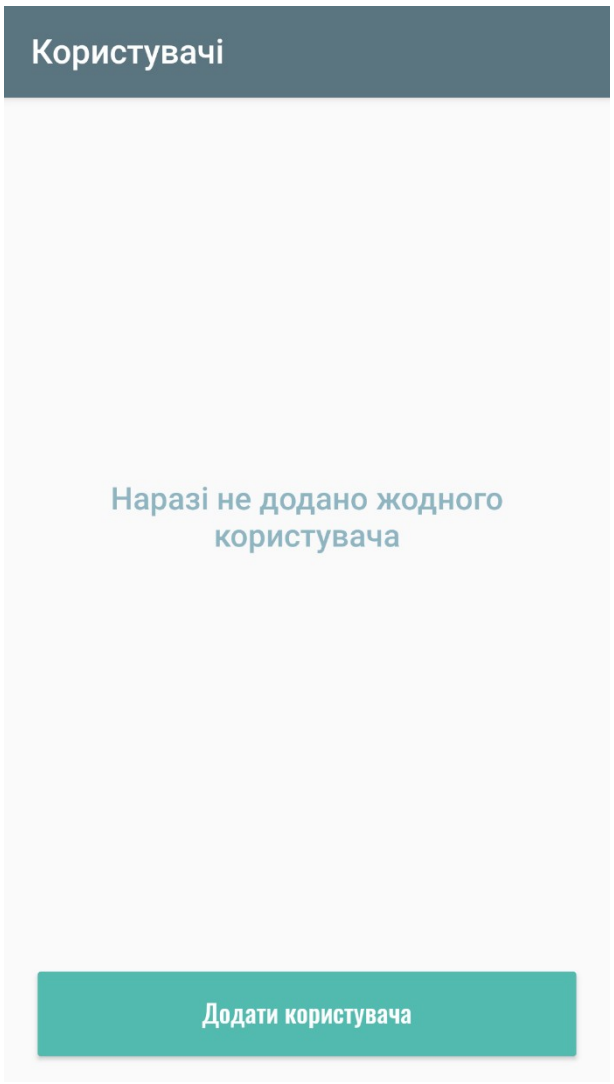


Рисунок 3.10 – Вигляд головного екрану додатку

На головному екрані після натискання на кнопку «Додати користувача» (рис. 3.11) з’являється форма реєстрації нового користувача (рис. 3.12). За замовчуванням перший зареєстрований користувач у системі призначається на роль «Адміністратор» (рис. 3.13), а усім наступним користувачам (що буде показано у цій частині інструкції) призначається роль «Користувач». Заповнивши усі поля форми (Логін, Повне ім’я, Пароль, Телефон та, за бажанням, Фото) валідними даними та натиснувши повторно кнопку «Додати користувача» відкривається діалог зчитування пальця (рис. 3.14). Необхідно звернути увагу на те, що усі поля обов’язкові до заповнення, інакше — користувач не зможе продовжити процедуру реєстрації (рис. 3.15).

Рисунок 3.11 – Вигляд кнопки «Додати користувача»

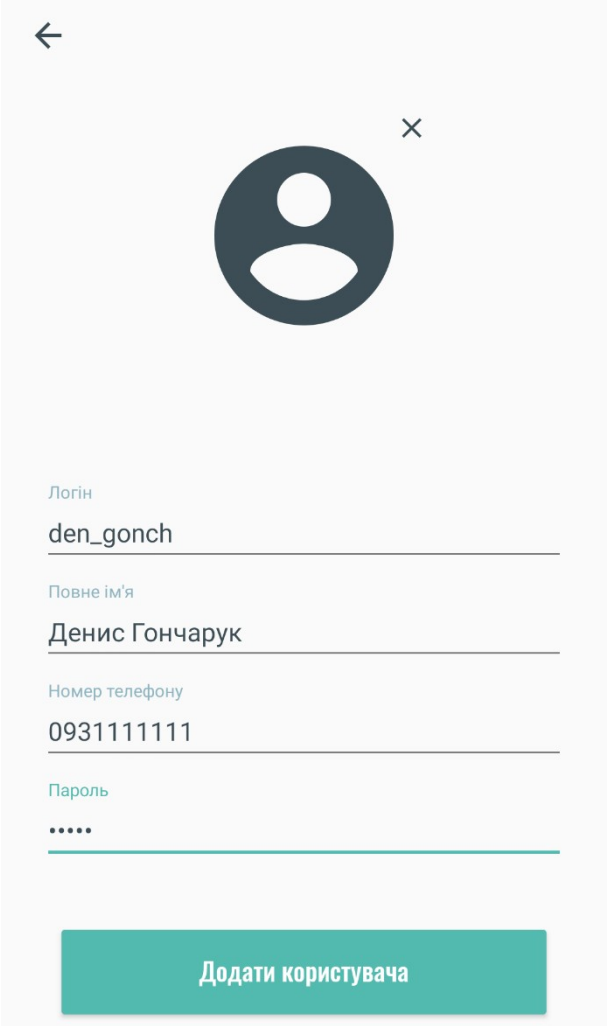


Рисунок 3.12 – Вигляд форми реєстрації нового користувача

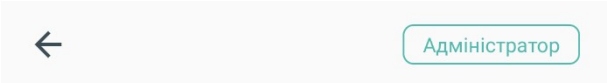


Рисунок 3.13 – Вигляд позначки, що вказує роль, яка буде призначена після реєстрації

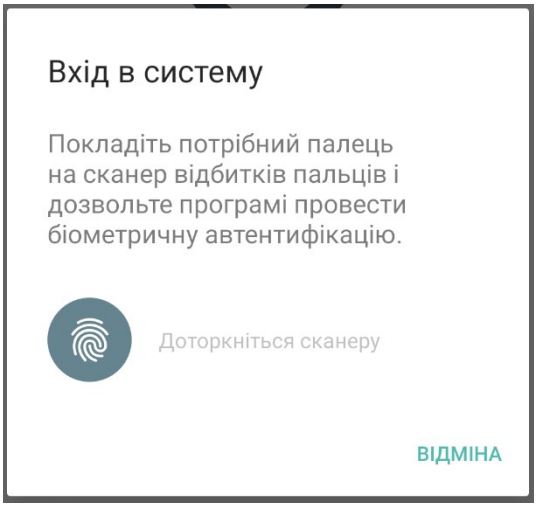


Рисунок 3.14 – Вигляд діалогу зчитування відбитку пальця

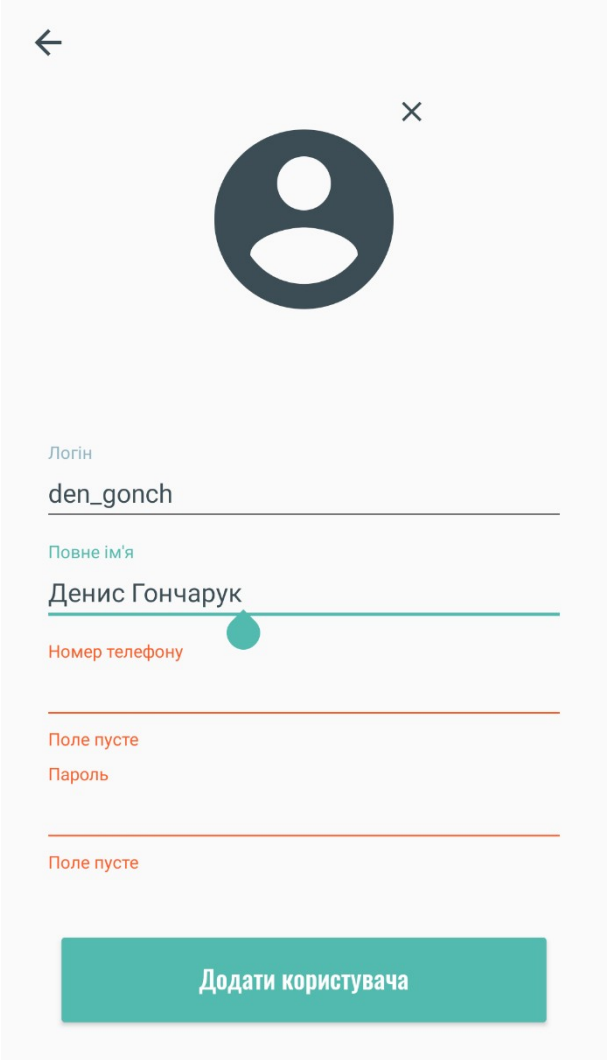


Рисунок 3.15 – Вигляд форми реєстрації нового користувача з незаповненими обов’язковими полями

Після успішної ідентифікації особи адміністратору відправляється запит на додавання цього користувача. Поки його не буде підтверджено, обліковий запис користувача є неактивним (рис. 3.16), а при спробі входу користувача буде додатково сповіщено про причину та подальші дії (рис. 3.17)

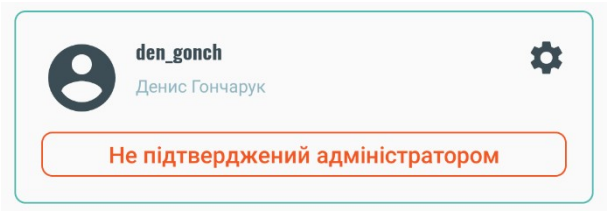


Рисунок 3.16 – Вигляд облікового запису відразу ж після реєстрації користувача

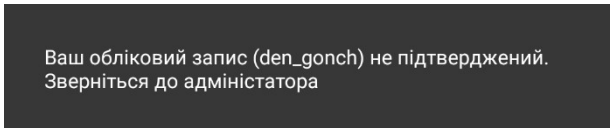


Рисунок 3.17 – Вигляд інформативного повідомлення про стан облікового запису

Після підтвердження адміністратором запиту на додавання, на головному екрані з’являється обліковий запис (рис. 3.18), який, у разі необхідності, за допомогою відповідної кнопки (рис. 3.19) та після проходження ідентифікації, можна редагувати (якщо змінилися якісь дані, або було скоєно помилку під час введення якихось із них), а також повністю видаляти з бази. Зберегти зміни можна зберегти за допомогою однойменної кнопки (рис. 3.20).

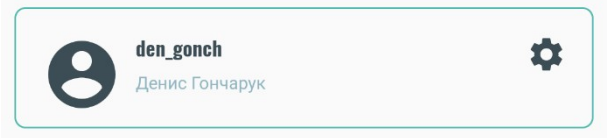


Рисунок 3.18 – Вигляд облікового запису користувача на головному екрані

Рисунок 3.19 – Вигляд кнопки редагування облікового запису

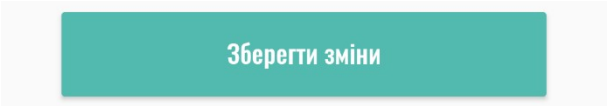


Рисунок 3.20 – Вигляд кнопки «Зберегти зміни» в меню редагування

3.4.2 Інструкція з використання розробленого програмного продукту для зареєстрованого користувача

Тепер перейдімо власне до використання самого програмного продукту у роботі.

1 крок. Користувач заходить у додаток на смартфоні. Відразу ж відкривається головний екран, на якому можна побачити список усіх користувачів (рис. 3.21).

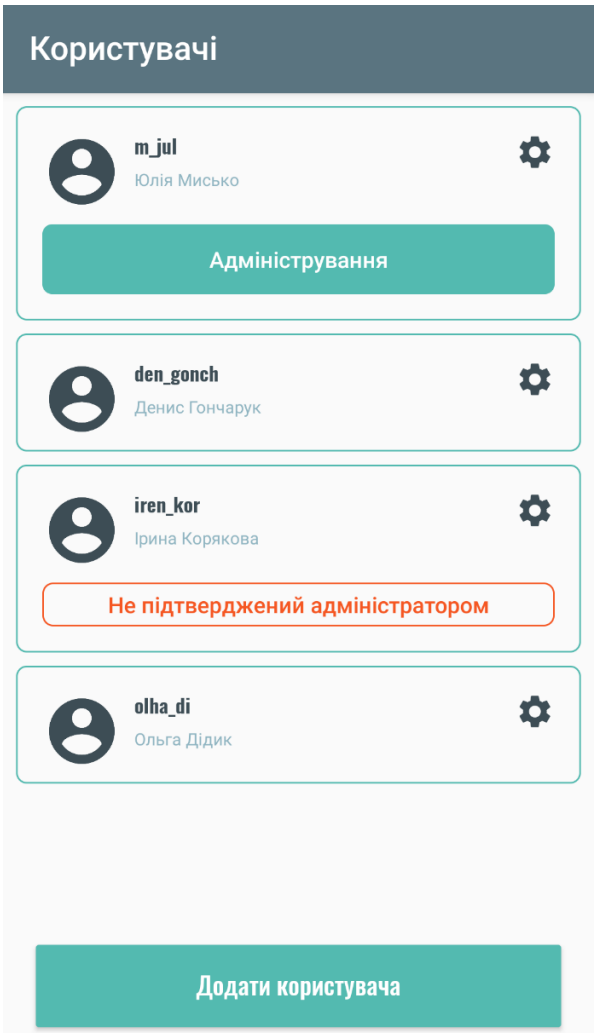


Рисунок 3.21 – Вигляд головного меню зі списком усіх зареєстрованих користувачів

2 крок. Користувач обирає зі списку свій обліковий запис та натискає на нього (рис. 3.22). Йому буде запропоновано прикласти свій палець (який використовувався при реєстрації) до сканеру відбитків пальців на смартфоні для ідентифікації особи (рис. 3.23). Якщо ідентифікація пройшла успішно (рис. 3.24), то користувач матиме доступ до захищеної системи (рис. 3.25), а також перейде в особистий кабінет (рис. 3.26), де зможе бачити та контролювати усі входи в свій обліковий запис (коли було розпочато/закінчено сесію та невдалі входи у свій обліковий запис). Якщо ж користувача не буде ідентифіковано системою, то він побачить повідомлення, зображене на рисунку 3.27, в системі буде зроблено відповідний запис, а самого користувача не допущено до системи.

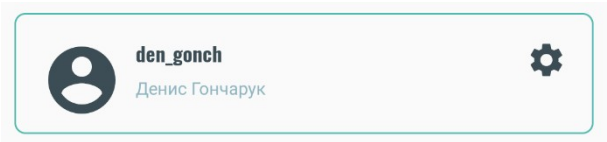


Рисунок 3.22 – Вигляд облікового запису користувача на головному екрані

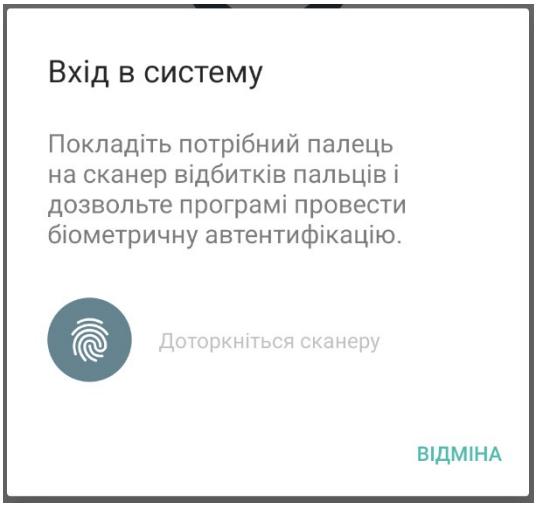


Рисунок 3.23 – Вигляд діалогу зчитування відбитку пальця

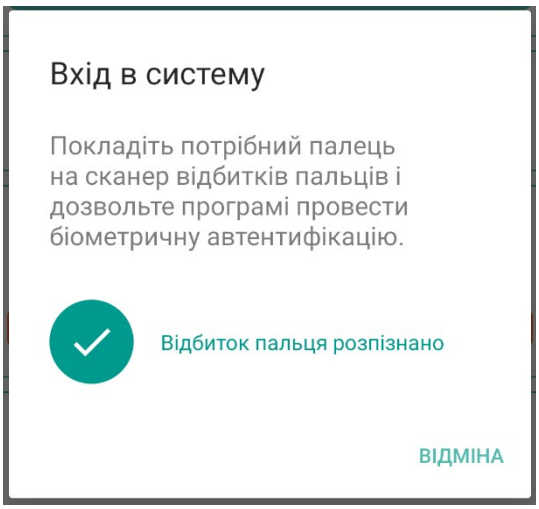


Рисунок 3.24 – Вигляд діалогу зчитування відбитку пальця після успішної ідентифікації користувача

а)

б)

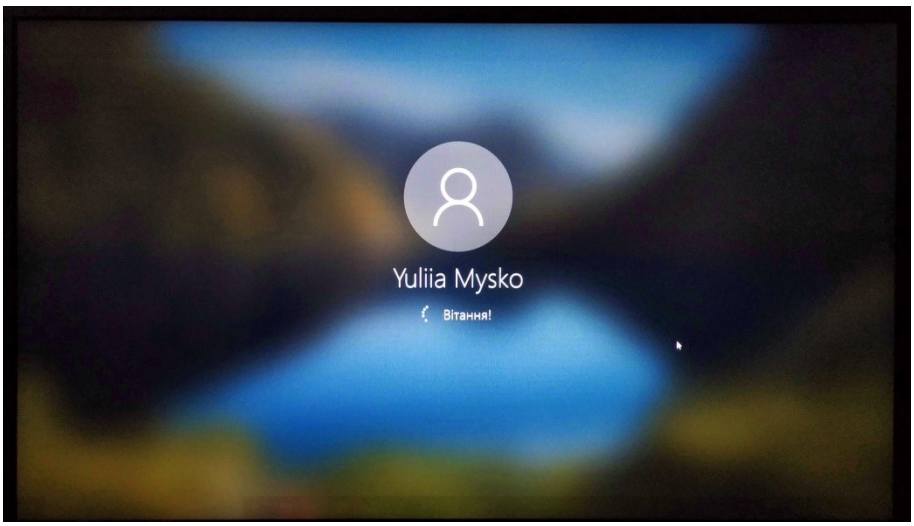


Рисунок 3.25 – Вигляд системи на ПК (а — заблокована система, б —розблокована система)

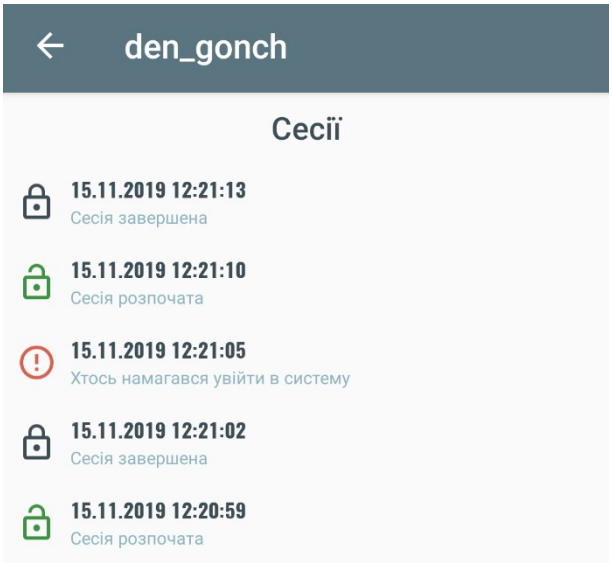


Рисунок 3.26 – Вигляд особистого кабінету користувача

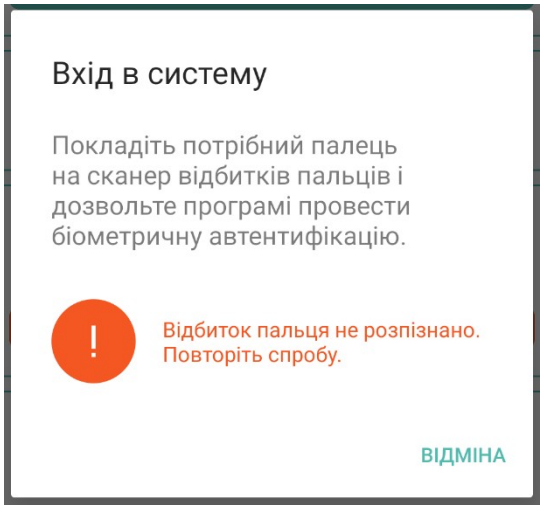


Рисунок 3.27 – Вигляд діалогу зчитування відбитку пальця після невдалої ідентифікації користувача

3 крок. Після завершення роботи з захищеною системою, користувачеві потрібно лише натиснути відповідну кнопку в правому верхньому куті (рис. 3.28) в особистому кабінеті і система відразу ж заблокується (рис. 3.29).

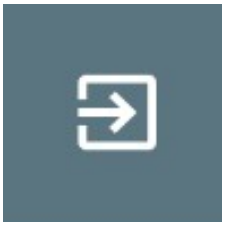


Рисунок 3.28 – Вигляд кнопки виходу із системи

а)

б)

Рисунок 3.29 − Вигляд системи на ПК (а — розблокована система, б —заблокована система)

3.4.3 Інструкція з використання розробленого програмного продукту для адміністратора системи

На роль адміністратора призначається перший зареєстрований у системі користувач.

Адміністратор також заповнює форму реєстрації у системі (рис. 3.30), проходить процедуру ідентифікації, але, на відміну від звичайних користувачів, не потребує жодного підтвердження.

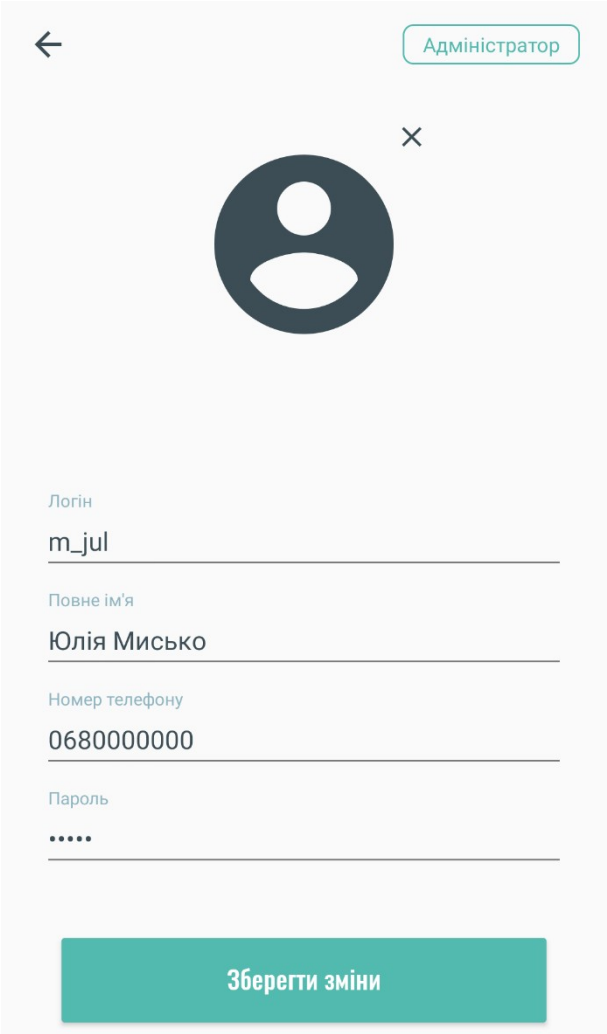


Рисунок 3.30 – Вигляд заповненої форми реєстрації адміністратора

Вигляд облікового запису адміністратора зображено на рисунку 3.31.

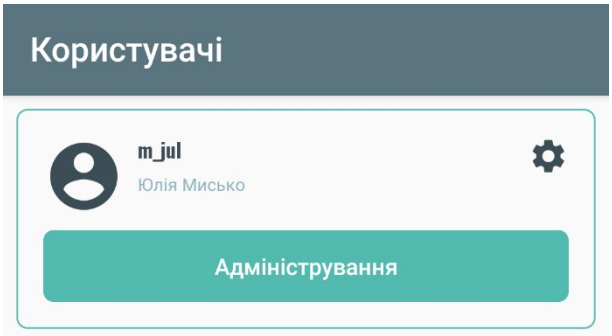


Рисунок 3.31 – Вигляд облікового запису адміністратора на головному екрані

Після натискання на кнопку «Адміністрування» адміністратор бачить список облікових записів усіх користувачів системи, з кожним із яких він може виконати ряд дій (рис. 3.32), а саме: подивитися список та час усіх сесій обраного користувача (рис. 3.33) (що також дає змогу контролювати затрачений

робочий час працівника протягом робочого дня та невдалі входи в систему, задля забезпечення безпеки останньої від несанкціонованого доступу), передавати право адміністрування іншому користувачеві у разі необхідності (рис. 3.34), а також видаляти користувача із системи.

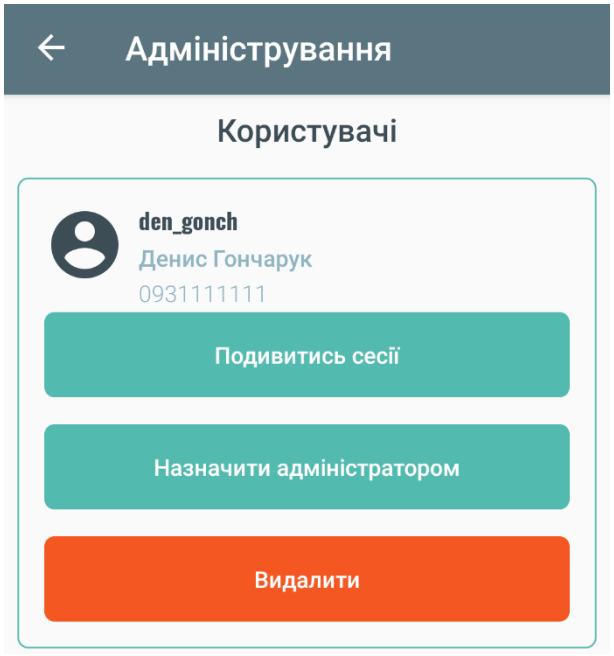


Рисунок 3.32 – Вигляд облікового запису користувача в адмінці

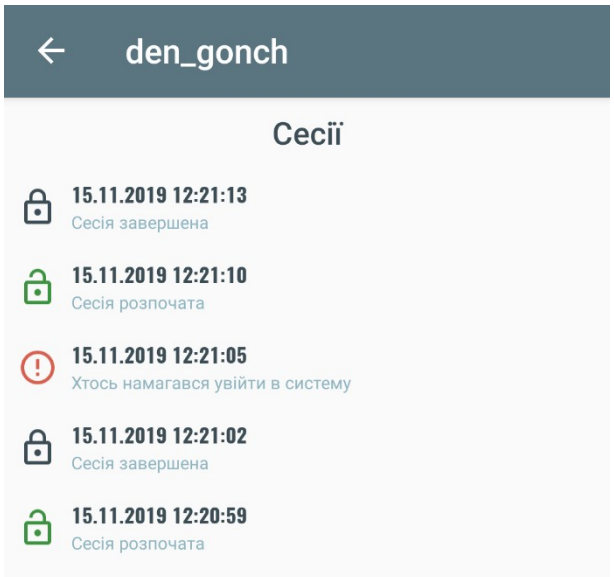


Рисунок 3.33 – Вигляд списку логів обраного користувача в адмінці

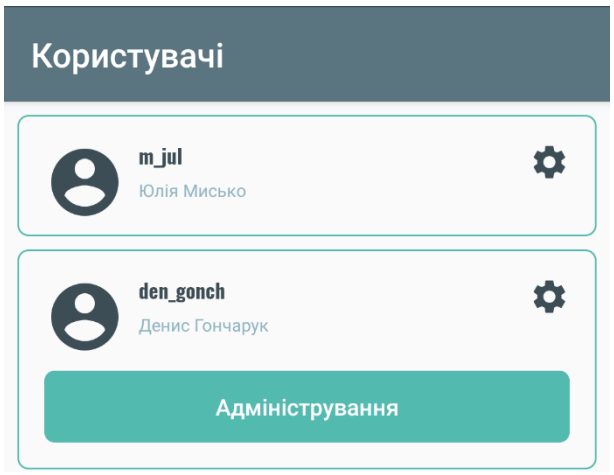


Рисунок 3.34 – Вигляд списку користувачів після передачі права адміністрування

Адміністратор контролює облік нових користувачів. У процесі реєстрації користувача адміністратору в особистий кабінет приходить запит на додавання або відхилення цієї реєстрації (рис. 3.35).

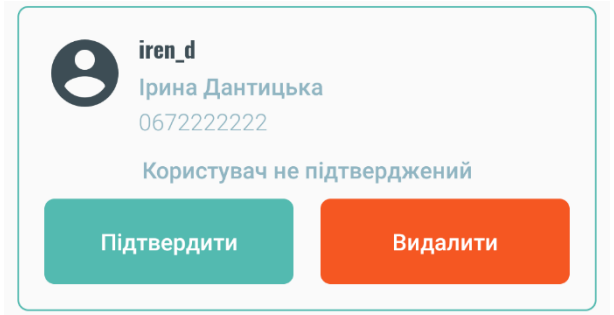


Рисунок 3.35 – Вигляд запиту на додавання нового користувача

Адміністратор може зайти в систему як звичайний користувач та переглядати свої логи (рис. 3.36).

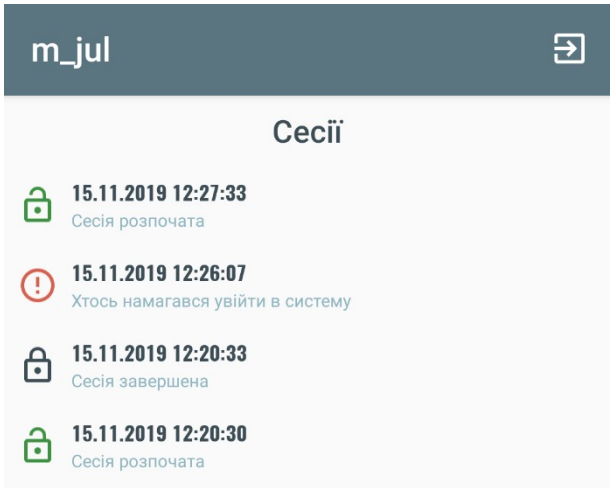


Рисунок 3.36 – Вигляд списку логів адміністратора

Для цього необхідно натиснути в будь-якій білій області свого блоку (все, окрім кнопки «Адміністрування»).

3.5 Тестування роботи програмного додатку

Чек-лист — це список, який містить ряд необхідних перевірок під час тестування програмного продукту. Відзначаючи пункти списку, можна дізнатися про поточний стан виконаної роботи і про якість продукту [33]. Працюючи над проектом по чек-листу, виключена ймовірність повторної перевірки за тими ж кейсами, а також підвищується якість тестування, оскільки ймовірність залишити без уваги якийсь функціонал суттєво знижується.

Для перевірки розробленого додатку було розроблено чек-лист, за яким проведено подальше тестування:

1) установка додатку:

- установка додатку на карту пам’яті;

- установка додатку з .apk;

- відсутність «крешів»;

2) функції на пристроях:

- перевірка відображення в горизонтальному та вертикальному положеннях;

3) графічні елементи:

- розмір елементу дозволяє користувачу попасти на нього;

- швидкість відгуку елементу;

- перевірка того, що всі написи відображаються в рамках відповідних форм/кнопок і не накладаються на дані полів;

4) реакція додатку на зовнішні переривання:

- реакція додатку на вхідні/вихідні дзвінки;

- реакція додатку на заряджання пристрою;

- реакція додатку на блокування/розблокування екрану;

- сумісність із іншими додатками;

- робота у фоновому режимі;

- робота додатку в режимі розділення екрану;

5) зворотній зв’язок із користувачем:

- реакція кнопок на натискання;

- наявність повідомлень при спробі видалити обліковий запис користувача;

- наявність екрану/повідомлення після закінчення процесу ідентифікації;

- наявність інформативного повідомлення при спробі входу в непідтверджений обліковий запис;

- можливість редагування профілю користувача;

- працездатність функції перепризначення адміністратора;

- інтуїтивна зрозумілість використання;

6) ресурси телефону:

- працездатність після оновлення ОС;

- витоки пам’яті;

- зберігання даних у кеш додатку;

- нестача місця для встановлення або роботи додатку;

- перевірка роботи батареї;

7) різноманітні розширення екрану та версії ОС:

- додаток встановлюється на підтримувані пристрої;

- додаток не повинен встановлюватися на непідтримувані пристрої;

- версії ОС.

Також, для детальнішої перевірки правильності роботи додатку було розроблено тест-кейси (табл. 3.2-3.4).

Тестовий випадок (Test Case) — це мінімальний елемент тестування (усього одна верифікація/перевірка); кроки, певні умови та значення, потрібні для перевірки реалізації функції, що тестується або її частини; опис певних дій і умов, необхідних для виявлення бага [33].

За очікуваним результатом тест-кейси поділяються на позитивні та негативні:

- позитивний тест кейс використовує лише коректні дані та перевіряє, що додаток правильно виконав функцію, що викликається;

- негативний тест кейс оперує як коректними, так і некоректними даними (щонайменше один некоректний параметр) та перевіряє виняткові ситуації, а також те, чи функція, яка викликається додатком, не виконується при спрацьовуванні валідатора.

Таблиця 3.2 — Ідентифікація користувача з використанням валідних даних

Дія	Результат
Обрати зі списку на головному екрані відповідний обліковий запис та натиснути на нього	З’являється спливаюче вікно сканування відбитку пальця
Пройти процедуру ідентифікації	У вікні сканування з’являється повідомлення (зеленими літерами) про успішну ідентифікацію користувача в системі, відбувається розблокування захищеної системи, а в особистому кабінеті відображаються усі сесії користувача та спроби входу в поточний обліковий запис
Після роботи із захищеною системою натиснути кнопку виходу	Завершується поточна сесія користувача, робиться відповідний запис у файлі логів, вхід у систему блокується, у додатку відбувається перехід на головний екран

Таблиця 3.3 — Ідентифікація користувача з використанням невалідних даних

Дія	Результат
Пройти процедуру ідентифікації	У вікні сканування з’являється повідомлення (червоними літерами) про невдалу ідентифікацію користувача в системі, захищена система залишається заблокованою, а користувачеві пропонується ще одна спроба входу в обліковий запис

Таблиця 3.4 — Процес адміністрування з використанням валідних даних

Дія	Результат
Обрати зі списку на головному екрані обліковий запис адміністратора та натиснути на кнопку «Адміністрування»	З’являється спливаюче вікно сканування відбитку пальця з позначкою «Адміністрування»
Пройти процедуру ідентифікації	У вікні сканування з’являється повідомлення (зеленими літерами) про успішну ідентифікацію користувача в системі, відкривається екран з усіма користувачами, які пройшли/проходять процес реєстрації
Обрати користувача, якому потрібно передати права адмініструвати на натиснути кнопку «Назначити адміністратором»	З поточного адміністратора знімаються відповідні права та призначаються обраному користувачеві, який відображатиметься в списку з позначкою «Адміністрування»

Продовження таблиці 3.4

Обрати користувача, сесії якого потрібно переглянути, та натиснути кнопку «Подивитись сесії»	Відображається екран з усіма сесіями та невдалими спробами входу в обліковий запис обраного користувача
Обрати зі списку користувача, якого потрібно видалити з бази даних, та натиснути кнопку «Видалити»	Відбувається видалення користувача з бази даних
Обрати зі списку користувача, який очікує підтвердження реєстрації, та натиснути кнопку «Підтвердити»	Відбувається успішна реєстрація користувача, який з моменту підтвердження може користуватися своїм обліковим записом
Обрати зі списку користувача, який очікує підтвердження реєстрації, та натиснути кнопку «Видалити»	Відбувається відхилення реєстрації нового користувача та видалення введених ним даних
Вийти з адмінки, натиснувши стрілку «Назад»	Відбувається перехід на головний екран додатку

Отже, у результаті виконання всіх пунктів чек-листа та позитивного/негативного тест-кейсів, додаток успішно пройшов процес тестування і готовий до подальшого використання.

3.6 Висновки до розділу 3

Мова Java є оптимальним засобом для реалізації нашого програмного продукту, оскільки вона була створена для забезпечення високого рівня безпеки, додатки та програми, написані на цій мові, вирізняються високим

ступенем надійності, тому що в мові повністю відсутні механізми, які часто призводять до помилок в роботі програми. Мова повністю безпечна, а тому настільки застосовувана в сфері програмування та розробки ефективних програм. Завдяки можливості мови реалізовуватися зовсім різними методами, виходять дуже різноманітні типи додатків, які будуть володіти високою функціональністю, різними способами роботи.

Android Studio — інтегроване середовище для розробки додатків під Android, заснованим на IntelliJ IDEA. Дане середовище розробки було обрано, оскільки воно пропонує ще більше можливостей, що підвищують продуктивність при створенні додатків для Android, а саме: гнучку систему збірки на основі Gradle; швидкий і багатофункціональний емулятор; єдине середовище, у якому можна розробляти для всіх Android-пристроїв; миттєвий запуск для внесення змін у робочому додатку без створення нового APK; інструменти Lint для відстеження продуктивності, зручності використання та сумісності версій; підтримка C++, NDK та багатьох інших.

4 ЕКОНОМІЧНА ЧАСТИНА

В магістерській кваліфікаційній роботі розробляється система ідентифікації користувачів.

Для ефективного вкладення капіталу необхідна попередня підготовка техніко-економічного обґрунтування (ТЕО) інноваційного проекту. Значне місце цього етапу в інноваційному процесі обумовлюється тим, що чим більше вірогідної та грамотно оформленої інформації про підприємство (проект) одержить інвестор, тим менше на нього чекає ризику на етапі реалізації проекту.

Оскільки інновації за своїм змістом і складом наближаються до реальних інвестицій, то насамперед реальні інвестиційні проекти потребують розробки ТЕО.

Тому в економічній частині магістерської роботи буде виконано такі етапи робіт [34]:

- оцінювання комерційного потенціалу розробки;
- прогнозування витрат на виконання наукової роботи та впровадження її результатів;
- прогнозування комерційних ефектів від реалізації результатів розробки;
- розрахунок ефективності вкладених інвестицій та періоду їх окупності.

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Для проведення технологічного аудиту було залучено 3-х незалежних експертів. Такими експертами є: керівник магістерської роботи – доц каф. ПЗ Черноволик Галина Олександрівна, ктн., доц каф. ПЗ – Ракитянська Ганна Борисівна, ктн., доц каф. ПЗ – Майданюк Володимир Павлович.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями, наведеними в таблиці 4.1.

Таблиця 4.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
№	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 4.1

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх Штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому Комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.2.

Таблиця 4.2 — Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1. Черноволик Г.О.	2. Ракитянська Г.Б.	3. Майданюк В.П.
	Бали, виставлені експертами:		
1	4	4	4
2	3	1	2
3	4	4	4
4	3	3	4
5	4	4	4
6	2	2	1
7	3	3	3
8	3	3	2
9	1	1	1
10	3	3	3
11	4	4	4
12	4	4	4
Сума балів	$СБ_1 = 38$	$СБ_2 = 36$	$СБ_3 = 36$
Середньоарифметична сума балів $\overline{СБ}$	$СБ = \frac{\sum_{1}^{3} СБ_i}{3} = \frac{38+36+36}{3} = 37$		

Отже, з отриманих даних таблиці 4.2 видно, що нова розробка має рівень комерційного потенціалу вище середнього.

Реалізація розробки відбуватиметься через розповсюдження у вигляді ліцензійної угоди на певний термін дії та надання технічної підтримки, що зацікавить клієнтів.

Якість виконання даної розробки підтверджена мануальним тестуванням, не поступається подібним програмним продуктам, вирізняється новизною рішень та відповідає стандартам ISO.

Шляхом зіставлення розробленої системи ідентифікації користувачів із використанням мобільних пристроїв з показниками найближчого аналога, основні функціональні характеристики зведено до таблиці 4.3.

Таблиця 4.3 — Порівняння функціональних характеристик з аналогом

Показники	Аналог	Розробка	
1. Швидкодія	78%	95%	1,22
2. Зручність	70%	100%	1,43
3. Функціональність	64%	92%	1,44

4. Підтримка	87%	98%	1,13

Переваги розробки полягають у тому, що вона зручна та зрозуміла у користуванні, швидка, мобільна, забезпечує контроль та захищеність, економічно вигідна.

Розробка орієнтована на підвищення зручності використання користувачами захищених комп'ютерних систем на підприємствах/організаціях тощо.

Технічна готовність розробленої системи ідентифікації користувачів із використанням сканеру відбитків пальців на смартфоні уже реалізована у вигляді Android додатку, до якого розроблена інструкція, та може бути впроваджена для продажу. Якщо розглядати існуючий стан робіт із комерціалізації розробки, то можна зазначити про проведення переговорів щодо впровадження даної системи.

4.2 Прогнозування витрат на виконання науково-дослідної та конструкторсько–технологічної роботи

Прогнозування витрат на виконання науково-дослідної та конструкторсько-технологічної роботи складається з таких етапів:

- розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

- розрахунок загальних витрат на виконання даної роботи;

- прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Основна заробітна плата для розробників визначається за формулою:

$$З_о = \frac{М}{Т_р} \cdot t, \quad (4.1)$$

де М — місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

$Т_р$ — кількість робочих днів у місяці, $Т_р = 22$ дні;

t — число робочих днів роботи розробника.

Розрахунки заробітних плат для наукового керівника та розробника наведені в таблиці 4.4.

Таблиця 4.4 — Розрахунки основної заробітної плати

Найменування посади виконавця	Місячний оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на оплату праці, грн.
Науковий керівник	7000	318	3	954
Інженер-програміст	9000	409	66	26994
Усього:				27948

Розрахуємо додаткову заробітну плату:

$$З_{дод} = (0,1…0,12) \cdot З_0, \qquad (4.2)$$

$З_{дод} = 0,1 \cdot 27948 = 2795$ (грн).

Нарахування на заробітну плату $Н_{зп}$ розробників, які брали участь у виконанні даного етапу роботи, розраховується за формулою:

$$Н_{зп} = (З_0 + З_{дод}) \cdot \frac{\beta}{100}, \qquad (4.3)$$

де $З_о$ — основна заробітна плата розробників, грн.;

$З_{дод}$ — додаткова заробітна плата всіх розробників, грн.;

β — ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, %.

$$Н_{зп} = (27948 + 2795) \cdot \frac{22}{100} = 6763,5 \text{ (грн)}.$$

Амортизаційні відрахування за прямолінійним методом розраховується за такою формулою:

$$А = \frac{Ц}{Т} \cdot \frac{Т_в}{12}, \qquad (4.4)$$

де Ц — балансова вартість обладнання, грн;

Т — корисний час використання;

$Т_в$ — термін використання.

Отже, розрахуємо амортизаційні відрахування та зведемо результати до таблиці 4.5:

$$А_к = \frac{200000}{20} \cdot \frac{3}{12} = 2500 \text{ (грн).}$$

$$А_к = \frac{25000}{2} \cdot \frac{3}{12} = 3125 \text{ (грн).}$$

$$А_с = \frac{8000}{2} \cdot \frac{1}{12} = 333 \text{ (грн).}$$

Таблиця 4.5 — Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення, приміщень тощо	Балансова вартість, грн.	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
Приміщення	100000	3	2500
Комп’ютер	25000	3	3125
Смартфон	8000	1	333
Усього:			5958

Витрати на матеріали: папір — 100 грн., канцелярія — 45 грн, інтернет за 3 місяці — 450 грн. Разом ця стаття витрат складає 595 грн.

Під час створення програмного продукту використовувалось безкоштовне програмне забезпечення.

Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \qquad (4.5)$$

де В — вартість 1кВт-години електроенергії (В = 0,9 грн/кВт);

П — установлена потужність комп’ютера (П = 0,3 кВт);

Ф — фактична кількість годин роботи комп’ютера (Ф = 528 год);

K_n — коефіцієнт використання потужності ($K_n < 1$, $K_n = 0{,}8$).

Потужність комп’ютера складає 230 Вт/год = 0,23 кВт/год + потужність на освітлення, тоді П = 0,3 кВт/год.

$В_е = 0{,}9 \cdot 0{,}3 \cdot 528 \cdot 0{,}8 = 114 \text{ (грн)}.$

Розрахуємо інші витрати $В_{ін}$.

Інші витрати $І_в$ можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які виконували дану роботу, тобто:

$$В_{ін} = (1..3) \cdot З_0, \qquad (4.6)$$

Отже, розрахуємо інші витрати:

$$В_{ін} = 2 \cdot 27945 = 55890 \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даного етапу роботи — В.

$$В = 27948 + 2795 + 6763 + 5958 + 595 + 114 + 55890 = 100063 \text{ (грн)}.$$

Розрахуємо загальну вартість наукової розробки $В_{заг}$ за формулою:

$$В_{заг} = \frac{В}{\alpha}, \qquad (4.7)$$

де α — частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відносних одиницях = 1.

$$В_{заг} = \frac{100063}{1} = 100063 \text{ (грн)}.$$

Прогнозування загальних витрат на виконання та впровадження результатів виконаної наукової розробки здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta}, \qquad (4.8)$$

де β — коефіцієнт, який характеризує етап виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{100063}{0{,}9} = 111181 \text{ (грн)}.$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Магістерська робота містить економічну частину обґрунтування економічної доцільності розробки системи ідентифікації користувачів, для того, щоб виконати дану розробку потрібно 66 робочих днів. Дана розробка вважається економічно вигідною, якщо її окупність становитиме 1 рік.

Термін окупності витрат на розробку становитиме 0,43 роки, тому розробка нового методу буде економічно ефективною. Новий метод є

економічно доцільним для споживача. Отримані вище результати доводять, що цей метод є більш корисніший та необхідніший для нової розробки.

Зростання чистого продукту для даного методу можна оцінити у теперішній вартості грошей. Зростання чистого прибутку забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Моніторинг ринку показує, що кількість підриємств, які потенційно зацікавлені у використанні нашої розробки в Україні складає приблизно 400 шт. (в розрахунку на рік). Реальний попит на розробку може становити 25-50 шт. за рік. Середня ціна подібної розробки, яка виконує аналогічні функції, дорівнює приблизно 23 тис. грн. Але оскільки запропонована нами мобільна система має значно кращі технічні параметри та функціональні можливості, то це дозволяє реалізовувати нашу розробку приблизно на 20-30% дорожче.

Припустимо, що наша розробка буде користуватися підвищеним попитом на ринку протягом 3-років після впровадження.

Результати нашої розробки можуть бути впроваджені вже з 1 січня 2020 року, а її результати будуть виявлятися протягом 2020-го, 2021-го та 2022-го років. Прогноз попиту на розробку складає по роках:

- 1-й рік після впровадження (2020 р.) – приблизно 28 шт.;
- 2-й рік після впровадження (2021 р.) – приблизно 59 шт.;
- 3-й рік після впровадження (2022 р.) – приблизно 104 шт.

На 4-й рік (2023 р.) не планується отримання прибутків, оскільки високою є ймовірність, що будуть розроблені нові, більш ефективні системи подібного типу.

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta \Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta П_i = \sum_1^n \left(\Delta П_я \cdot N + П_я \cdot \Delta N\right)_i, \qquad (4.9)$$

де Δ $П_я$ — покращення основного якісного показника від впровадження результатів розробки у даному році;

N — основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

Δ N — покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$ — основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n — кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту Δ $П_i$ протягом першого року складатиме:

$$\Delta П_1 = 1 \cdot 20000 + 28 \cdot 30000 = 860000 \text{ (грн)}.$$

Протягом другого року:

$$\Delta П_2 = 1 \cdot 20000 + (28 + 31) \cdot 30000 = 1790000 \text{ (грн)}.$$

Протягом третього року:

$$\Delta П_3 = 1 \cdot 20000 + (28 + 31 + 45) \cdot 30000 = 3140000 \text{ (грн)}.$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1) Розраховано теперішню вартість інвестицій PV, що вкладаються в наукову розробку (прогнозована величина загальних витрат ЗВ на виконання та впровадження результатів НДДКР). ЗВ = PV.

2) Розраховано очікуване збільшення прибутку ΔПі, яке отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження.

3) Для спрощення подальших розрахунків побудовано вісь часу, на яку наносяться всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів. Платежі показуються у ті терміни, коли вони здійснюються.

Рисунок, що характеризує рух платежів у грн. (інвестицій та додаткових прибутків) матиме вигляд, наведений на рисунку 4.1.

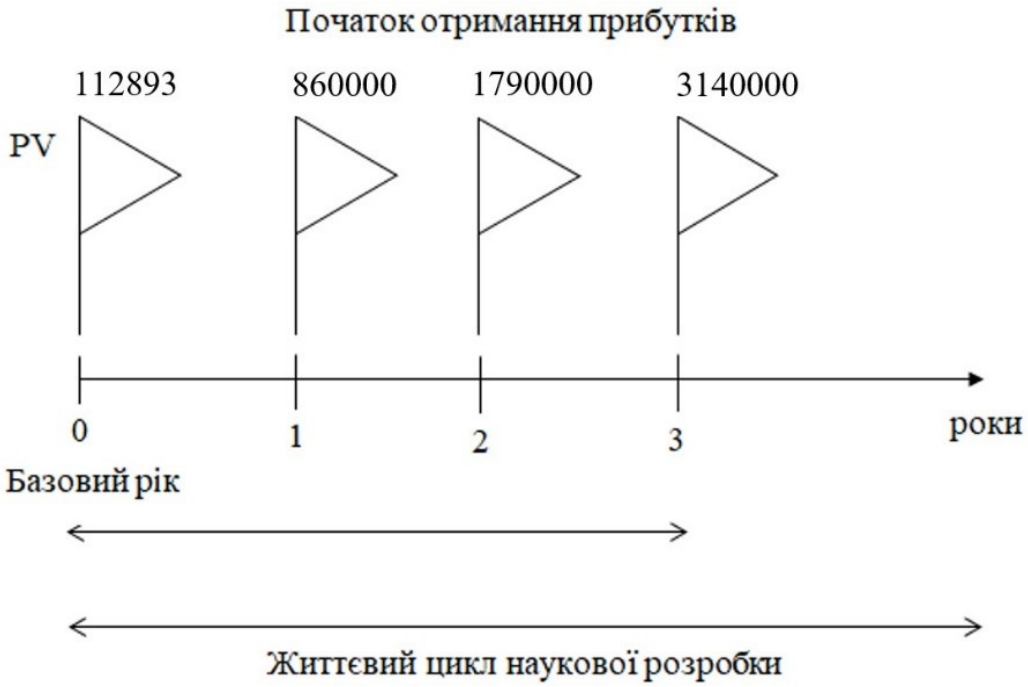


Рисунок 4.1 — Вісь часу з фіксацією платежів у грн., що мають місце під час розробки та впровадження результатів НДДКР

4) Абсолютна ефективність $E_{абс}$ вкладених інвестицій розраховується за формулою:

$$E_{абс} = (ПП - PV), \qquad (4.10)$$

де ПП — приведена вартість чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV — теперішня вартість інвестицій PV=ЗВ, грн.

Розрахуємо вартість чистих прибутків за формулою:

$$ПП = \sum_{1}^{т} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (4.11)$$

де $\Delta П_i$ — збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн.;

т — період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t — період часу (в роках) від моменту отримання чистого прибутку до точки «0».

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{860000}{(1+0,1)^2} + \frac{1790000}{(1+0,1)^3} + \frac{3140000}{(1+0,1)^4} = 4200406 \text{ (грн).}$$

Тоді розрахуємо $Е_{абс}$:

$$Е_{абс} = 4200406 - 111181 = 4089225 \text{ (грн).}$$

Оскільки $Е_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

5) Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $Е_в$ за формулою:

$$Е_в = \sqrt[Тж]{1 + \frac{Е_{абс}}{PV}} - 1, \qquad (4.12)$$

де $Е_{абс}$ — абсолютна ефективність вкладених інвестицій, грн.;

PV — теперішня вартість інвестицій PV = ЗВ, грн.;

$Т_ж$ — життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_в = \sqrt[3]{1 + \frac{4089225}{111181}} - 1 = 2{,}3 \text{ або } 230\%$$

Далі, розраховану величину $E_в$ порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$\tau = d + f, \qquad (4.13)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках (d = 0,2);

f — показник, що характеризує ризикованість вкладень, величина f = 0,1.

$\tau = 0{,}2 + 0{,}1 = 0{,}3$

Оскільки $E_в = 230\% > \tau_{мін} = 30\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

6) Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_в}, \qquad (4.14)$$

$$T_{ок} = \frac{1}{2{,}3} = 0{,}43 \text{ (років)}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

4.5 Висновки

В даному розділі було проведено оцінку комерційного потенціалу розробки системи ідентифікації користувачів, спрогнозовано витрати на виконання наукової роботи та впровадження її результатів, і комерційні ефекти від реалізації результатів розробки та розраховано ефективність вкладених інвестицій та період їх окупності.

Проаналізувавши отримані результати, можна зробити висновок, що розробка даної системи є актуальною та доцільною з точки зору економічних

розрахунків. Таким чином, відносна (щорічна) ефективність вкладених в наукову розробку інвестицій $E_в$ становить 230%. Тому, обрахувавши термін окупності даної наукової розробки, що складає 0,43 роки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

ВИСНОВКИ

Під час написання магістерської кваліфікаційної роботи було проаналізовано методи ідентифікації особи, розглянуто такий великий пласт, як біометрична ідентифікація та обрано для використання, дослідження та вдосконалення один із них — розпізнавання за відбитками пальців.

Актуальність розвитку біометричних технологій ідентифікації особи обумовлена збільшенням числа об'єктів і потоків інформації, які необхідно захищати від несанкціонованого доступу, а саме: системи контролю доступу; системи ідентифікації особи; системи електронної комерції; інформаційна безпека (доступ в мережу, вхід на ПК); облік робочого часу та реєстрація відвідувачів; аутентифікація на web-ресурсах; різні соціальні проекти, де потрібна ідентифікація людей і т.п. Використання біометричних засобів спрощує процедуру аутентифікації особи, а також підвищує надійність систем безпеки.

Переваги біометричних систем ідентифікації користувачів − незаперечні. Швидкість обробки даних, постійність авторизаційної інформації в поєднанні з доступною ціною, все це беззаперечно повинно схилити підприємців до впровадження біометричних систем ідентифікації.

Було проведено аналіз особливостей побудови існуючих компонентів безпеки комп’ютерних систем, наведено класифікацію методів ідентифікації та автентифікації, показано ефективність побудови компонентів безпеки на основі біометричних даних, наведено класифікацію компонентів захисту комп’ютерних систем та мереж за класифікаційною ознакою, що визначає рівень очікуваного ефекту захищеності.

Використання біометричних засобів спрощує процедуру ідентифікації особи, а також підвищує надійність систем безпеки. Для підтвердження особи при використанні смартфону, планшету або ноутбуку найбільш підходящою ІТ є біометрія за відбитками пальців.

Проаналізувавши переваги та недоліки методу ідентифікації та алгоритм захисту інформації в комп’ютерних мережах на основі біометричних даних з використанням відбитків пальців акцентовано, що захист комп’ютерних мереж необхідно виконувати на основі системного підходу, забезпечуючи необхідний рівень захищеності на всіх рівнях.

Було спроектовано базу даних користувачів за допомогою моделі «сутність-зв’язок». Проектування містило такі етапи визначення: сутностей, зв’язків, атрибутів, ключів сутностей, визначення ступеня зв`язку, класу належності. Контроль за доступом до інформації здійснює сервер бази даних. При виконанні запиту користувача SQL-сервер отримує відбиток пальців, визначає ім'я користувача і за внутрішньою інформацією визначає, чи може ця особа виконати цей запит.

Також було обґрунтовано вибір оптимальної мови програмування для реалізації нашого програмного продукту. Мова повністю безпечна, а тому настільки застосовувана в сфері програмування та розробки ефективних програм. Як середовище розробки було обрано Android Studio, оскільки воно пропонує безліч можливостей, що підвищують продуктивність при створенні додатків для Android.

Було проведено оцінку комерційного потенціалу розробки системи ідентифікації користувачів, спрогнозовано витрати на виконання наукової роботи та впровадження її результатів, а також комерційні ефекти від реалізації результатів розробки та розраховано ефективність вкладених інвестицій та період їх окупності. Проаналізувавши отримані результати, було зроблено висновок, що розробка даної системи є актуальною та доцільною з точки зору економічних розрахунків, а також фінансово вигідною.

Отже, було розроблено програмні засоби ідентифікації користувачів комп’ютерних систем за відбитками пальців через смартфон та інструкцію користувача до них.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Полякова А. Захист інформації в сучасних інформаційних технологіях. 2017. URL: https://www.epravda.com.ua/publications/2017/12/15/632183/ (Дата звернення: 07.10.2019).

2) Галатенко В.А. Идентификация и аутентификация, управление доступом лекция из курса «Основы информационной безопасности». Интернет Университет Информационных Технологий, 2010.

3) Постанова від 29 березня 2006 р. N 373 Київ Про затвердження Правил забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах. URL: https://zakon.rada.gov.ua/laws/show/373-2006-%D0%BF (Дата звернення: 08.10.2019).

4) Семенов С.Г. Методы и средства распределения доступа и защиты данных в компьютеризированных информационных управляющих системах критического применения / С.Г. Семенов. – Х.:НТУ «ХПИ», 2013. – 360 с.

5) Девянин П.Н. Модели безопасности компьютерных систем / П.Н. Девянин. – М.: Издательский центр «Академия», 2005. – 144 с.

6) Безмалый В. Парольная защита: прошлое, настоящее, будущее / В. Безмалый // IT спільнота України, жовтень 2014. URL: https://www.itcommunity.in.ua/2014/10/parolnaya-zashhita-proshloenastoyashhee-budushhee.html/ (Last accessed: 11.10.2019).

7) Дшхунян В. Л., Шаньгин В. Ф. Электронная идентификация. Бесконтактные электронные идентификаторы и смарт-карты. – М.: АСТ, 2004. – 696 с.

8) Мальцев А. Современные биометрические методы идентификации / Антон Мальцев, 2011. URL: https://habrahabr.ru/post/126144/ (Last accessed: 14.10.2019).

9) Кумченко Ю. О. Аналіз існуючих підходів біометричної ідентифікації та аутентифікації людини / Ю. О. Кумченко, А. І. Купін //

Системні технології. Регіональний міжвузівський збірник наукових праць. – Дніпропетровськ, 2013. Вип. 4 (87). С. 129–134.

10) Шрамко В. Н. Комбинированные системы идентификации и аутентификации // PCWeek/RE, 2004. №45. С. 30–32.

11) Кухарев Г.А. Биометрические системы: Методы и средства идентификации личности человека / Г.А. Кухарев. – СПб.: Политехника, 2001. – 240 с.

12) Кумченко Ю. О. Аналіз основних характеристик біометричних систем розпізнавання на основі поширених помилок ідентифікації / Ю. О. Кумченко, А. І. Купін // Сучасні інформаційні технології 2013: Матеріали третьої Міжнародної конференції студентів і молодих науковців (25–26 квітня 2013 р.). Одеса, 2013. С. 112–113.

13) Задорожный В. В. Идентификация по отпечаткам пальцев / В. В. Задорожный. // PC Magazine. 2004. №1.

14) Задорожный В. В. Идентификация по отпечаткам пальцев / В. В. Задорожный. // PC Magazine. 2004. №2.

15) Гуреева О. Биометрическая идентификация по отпечаткам пальцев. Технология FingerChip / Ольга Гуреева. // Компоненты и технологии, 2007. №4. С. 176–180.

16) Анализ существующих подходов к распознаванию лиц. URL: https://habrahabr.ru/company/synesis/blog/238129/ (Дата доступа: 18.10.2019).

17) Daugman, J., “Recognizing Persons by Their Iris Patterns”, In: Biometrics: Personal Identification in Networked Society, 1998, Amsterdam: Kluwer, pp. 103–121.

18) VeriEye SDK // NEUROtechnology, 2016. URL: http://www.neurotechnology.com/verieye.html (Last accessed: 19.10.2019).

19) Барабанова М.И., Кияев В.И. Информационные технологии: открытые системы, сети, безопасность в системах и сетях: Учебное пособие. - СПб.: Изд-во СПбГУЭФ, 2010. - 267 с.

20) Тихонов И. А. Информативные параметры биометрической аутентификации пользователей информационных систем, 2010. № 9. С. 26-32.

21) Цирлов В.Л. Основы информационной безопасности автоматизированных систем: краткий курс. - Феникс, 2008.

22) Дубчак О. В. Аналіз ефективності та надійності методів біометричної аутентифікації / О. В. Дубчак, Ю. О. Максімов. // Современные информационные технологии, 2011. №1.

23) Лисенко А. М. Застосування біометричних систем для ідентифікації особи / А. М. Лисенко, О. С. Мельник. // Юридичні науки. Вісник Київського національного університету імені Тараса Шевченка, 2004. №60/62. С. 87–91.

24) Biometric fingerprint scanners. URL: https://www.explainthatstuff.com/ /fingerprintscanners.html (Last accessed: 03.11.2019).

25) Семёнов Д.А., Бобров А.В. ACL: списки контроля доступа и их применение в организации сетей // Студенческий: электрон. научн. журн. 2018. №11(31). URL: https://sibac.info/journal/student/31/112744 (Дата обращения: 05.11.2019).

26) Томас Коннолли, Каролин Бегг. Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2018. - 1440 с.

27) Базы данных. Виды и типы баз данных. Структура реляционных баз данных. Проектирование баз данных. Сетевые и иерархические базы данных, 2012. URL: https://zametkinapolyah.ru/zametki-o-mysql/bazy-dannyx-vidy-i-tipy-baz-dannyx-struktura-relyacionnyx-baz-dannyx-proektirovanie-baz-dannyx-setevye-i-ierarxicheskie-bazy-dannyx.html (Дата обращения: 10.11.2019).

28) Понятие ER-модели. Понятие сущности. Атрибуты. Виды атрибутов. URL: https://www.bestprog.net/ru/2019/01/24/the-concept-of-er-model-the-concept-of-essence-and-communication-attributes-attribute-types-ru/ (Дата обращения: 14.11.2019).

29) Билл Карвин. Программирование баз данных SQL. Типичные ошибки и их устранение, 2012. – 336 с.

30) Что такое Kotlin и с чем его едят: обучающее руководство и сравнение нового языка Android-разработки с Java. URL: https://tproger.ru/translations/kotlin-vs-java-android/ (Last accessed: 16.11.2019).

31) Java. URL: https://www.theserverside.com/definition/Java (Last accessed: 18.11.2019).

32) Meet Android Studio, 2018. URL: https://developer.android.com/studio/intro/ (Last accessed: 21.11.2019).

33) Савин Р. С. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. — М.: Дело, 2007. — 312 с.

34) Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.

35) Електронні інформаційні ресурси: створення, використання, доступ: Збірник матеріалів Міжнародної науково-практичної Інтернет конференції. – Вінниця: ВНТУ, 2019. – 292 с.

36) Мисько Ю. О. Сучасні методи біометричної ідентифікації користувачів. – ВНТУ, 2018. URL: https://conferences.vntu.edu.ua/index.php/mn/mn2018/paper/view/5627.

ДОДАТКИ

Додаток А. Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп’ютерної інженерії

УЗГОДЖЕНО
Директор ТОВ «Екзістек»
Феферман О. Д. _____
«____» _____2019 року

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
_____ Романюк О. Н.
«____» _____2019 року

Технічне завдання
на магістерську кваліфікаційну роботу
«Розробка методу та засобів системи ідентифікації користувачів»
за спеціальністю 121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:
_____ к.т.н., доц. Г.О. Черноволик
"____"_____2019 р.
Виконав:
_____студент гр. 2ПІ-18м Ю.О. Мисько
"____"_____2019 р.

Вінниця — 2019 року

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка методу та засобів системи ідентифікації користувачів».

Галузь застосування — системи ідентифікації користувачів.

2. Підстава для розробки.

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ №___ ректора по ВНТУ про закріплення тем МКР.

3. Мета та призначення розробки.

Метою роботи є підвищення ефективності захисту конфіденційних даних у комп'ютерних системах від несанкціонованих дій шляхом розмежування доступу за рахунок ідентифікації користувачів за відбитками пальців через смартфон, а також пошук компромісу між надійністю, доступною ціною та зручністю у використанні й адмініструванні засобів ідентифікації та аутентифікації.

Призначення роботи — розробка методу та засобів системи ідентифікації користувачів.

3 Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

1. Конахович Г.Ф. Захист інформації в мережах передачі даних: підручник / Г.Ф. Конахович, О.Г. Корченко, О.К. Юдін. – К.: Видавництво ТОВ НВП «ІНТЕРСЕРВІС», 2009. – 714 с.

2. Лисенко А. М. Застосування біометричних систем для ідентифікації особи / А. М. Лисенко, О. С. Мельник. // Юридичні науки. Вісник Київського національного університету імені Тараса Шевченка. – 2004. – №60/62. – С. 87–91.

3. Кухарев Г.А. Биометрические системы: методы и средства идентификации личности человека / Г.А. Кухарев.– СПб.: Политехника, 2001. – 240 с.

4. НД ТЗІ 2.5-004-99 Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу. [Електронний ресурс]. – Режим доступу до ресурсу: http://tzi.com.ua/nd-tz-2.5-004-99.html.

4. Технічні вимоги

Операційна система — Linux; середовище розробки — Android Studio; мови програмування — Java, Kotlin.

Смартфон з наступною мінімальною конфігурацією:

- ОС Android — версія 6.0 і вище;
- наявність сканеру відбитків пальців;
- технології — Wi-Fi.

Таким чином, розроблена система зовсім невибаглива і може бути використана навіть на найпростіших смартфонах.

5. Конструктивні вимоги.

Конструкція пристрою повинна відповідати естетичним та ергономічним вимогам, повинна бути зручною в обслуговуванні та керуванні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

6. Перелік технічної документації, що пред'являється по закінченню робіт:

– пояснювальна записка до МКР;

– технічне завдання;

– лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз предметної області та постановка задач дослідження	
2	Розробка оптимального методу із використанням біометричної ідентифікації за відбитками пальців для розв’язку поставлених задач	
3	Програмна реалізація та тестування програмного засобу ідентифікації користувачів	
4	Економічна частина	

10. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи.

Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Додаток Б. Акт впровадження

УЗГОДЖЕНО	ЗАТВЕРДЖУЮ
Директор ТОВ «Екзістек»	Завідувач кафедри ПЗ
Феферман О. Д.	д.т.н., професор Романюк О. Н.
«____» _____20 року	«____» _____20 року

АКТ ВПРОВАДЖЕННЯ № _____
результатів науково-дослідних робіт

Замовник ТОВ «Екзістек»
(найменування організації)

Цим актом підтверджується, що результати роботи – «Розробка методу та засобів системи ідентифікації користувачів»
(найменування теми)

що виконала студентка гр. 2ПІ – 18м ВНТУ Мисько Ю. О.
(виконавці)

За договором про творчу співдружність без взаємних грошових розрахунків, на громадських засадах відповідно до теми затвердженої наказом №___ від «___»_____20____ р.
виконана з _____ по _____
(строки виконання)

впроваджено у ТОВ «Екзістек»
(найменування організації, де здійснювалося впровадження)

1. Вид впроваджених результатів: експлуатація програмного забезпечення
(експлуатація виробу, роботи, технології)

2. Характеристика масштабу впровадження: одиничне
(унікальне, одиничне, партія, масове, серійне)

3. Форма впровадження: дослідний зразок

4. Новизна результатів науково-дослідної роботи якісно нові
(піонерські, принципово нові, якісно нові, модифікації, модернізація старих розробок)

5. Впроваджені: для внутрішнього моніторингу та контролю

6. Соціальний та науково-технічний ефект: спеціальне призначення
(охорона навколишнього середовища, поліпшення й оздоровлення умов праці, удосконалення структури керування, науково-технічних напрямків, спеціальне призначення)

Від виконавця:
Студентка групи 2ПІ-18м
_____ Мисько Ю.О.

Від ТОВ «Екзістек»
Директор
_____ Феферман О.Д.

Керівник: к.т.н., доцент кафедри _____ Черноволик Г.О.

Додаток В. Лістинг

Лістинг ValidateIP.kt

```
package ua.yuliyamysko.fingerprintauth.connection

import java.util.regex.Pattern

object ValidateIP {

   private val PATTERN = Pattern.compile(
      "^(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.){3}([01]?\\d\\d?|2[0-4]\\d|25[0-5])$"
   )//to validate ip address

   fun validateIP(ip: String): Boolean {
      return PATTERN.matcher(ip).matches()
   }
   fun validatePort(portNumber: String?): Boolean {
        return if (portNumber != null && portNumber.length == 4 && portNumber.matches(".*\\
d.*".toRegex())) {
         Integer.parseInt(portNumber) > 1023
      } else
         false
   }
}
```

Лістинг MakeConnection.java

```
package ua.yuliyamysko.fingerprintauth.connection;

import android.os.AsyncTask;

import java.io.ObjectOutputStream;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.SocketAddress;

import ua.yuliyamysko.fingerprintauth.FingerprintApp;

public class MakeConnection extends AsyncTask<Void, Void, Socket> {

   String ipAddress, port;
   FingerprintApp;
   Socket clientSocket;
   CallbackReceiver;

      public MakeConnection(String ipAddress, String port, FingerprintApp, CallbackReceiver
callbackReceiver) {
      this.ipAddress = ipAddress;
      this.port = port;
      this.fingerprintApp = fingerprintApp;
      this.callbackReceiver = callbackReceiver;
```

```
        }
        @Override
        protected Socket doInBackground(Void... params) {
            try {
                int portNumber = Integer.parseInt(port);
                SocketAddress = new InetSocketAddress(ipAddress, portNumber);
                clientSocket = new Socket();
                // 5s timeout
                clientSocket.connect(socketAddress, 5000);
                                                                    fingerprintApp.setObjOutputStream(new
ObjectOutputStream(clientSocket.getOutputStream()));
            } catch(Exception e) {
                e.printStackTrace();
                clientSocket = null;
            }
            return clientSocket;
        }

        protected void onPostExecute(Socket clientSocket) {
            callbackReceiver.receiveData(clientSocket);
        }
    }
```

Лістинг User.kt

```
package ua.yuliyamysko.fingerprintauth.data.model

data class User(
    var fullName: String,
    var userName: String,
    var pass: String,
    var phone: String,
    var imageUri: String? = null,
    var isApproved: Boolean,
    var isAdmin: Boolean = false,
    var isNeedAdditional: Boolean = false,
    var enterLogs: ArrayList<EnterLog> = ArrayList(),
    val id: Long = System.currentTimeMillis()
)
```

Лістинг UserRepository.kt

```
package ua.yuliyamysko.fingerprintauth.data

import android.content.Context
import android.content.SharedPreferences
import android.preference.PreferenceManager
import com.google.gson.Gson
import com.google.gson.reflect.TypeToken
import ua.yuliyamysko.fingerprintauth.data.model.EnterLog
import ua.yuliyamysko.fingerprintauth.data.model.User
class UserRepository(context: Context) {
```

```
private val sharedPreferences: SharedPreferences =
    PreferenceManager.getDefaultSharedPreferences(context)

private val gson = Gson()

fun getIpAddress(): String{
    return sharedPreferences.getString(IP_KEY, "") ?: ""
}

fun setIpAddress(ipAddress: String) {
    sharedPreferences.edit().putString(IP_KEY, ipAddress).apply()
}

fun getPort(): String{
    return sharedPreferences.getString(PORT_KEY, "") ?: ""
}

fun setPort(port: String) {
    sharedPreferences.edit().putString(PORT_KEY, port).apply()
}

fun getUserById(userId: Long) = getAllUsers().firstOrNull { it.id == userId }

fun addUser(user: User) {
    val users = getAllUsers().toMutableList()
    users.add(user)
    sharedPreferences.edit()
        .putString(USER_LIST_KEY, gson.toJson(users))
        .apply()
}

fun editUser(user: User) {
    val users = getAllUsers().toMutableList()
    if (!users.any { it.id == user.id }) {
        return
    }
    sharedPreferences.edit()
        .putString(
            USER_LIST_KEY,
            gson.toJson(users.map { if (it.id == user.id) user else it })
        )
        .apply()
}

fun removeUser(userId: Long) {
    val users = getAllUsers().toMutableList()
    users.removeAll { it.id == userId }
    sharedPreferences.edit()
        .putString(USER_LIST_KEY, gson.toJson(users))
        .apply()
}

fun getAllUsers(): List<User> {
    val usersJson = sharedPreferences.getString(USER_LIST_KEY, "[]")
```

```
        return gson.fromJson(usersJson, object : TypeToken<ArrayList<User>>() {}.type)
    }

    fun approveUser(userId: Long) {
        val user = getUserById(userId) ?: return
        user.isApproved = true
        editUser(user)
    }

    fun setAdminRole(userId: Long) {
        val users = getAllUsers().toMutableList()
        if (!users.any { it.id == userId }) {
            return
        }
        users.forEach {
            it.isAdmin = it.id == userId
        }
        sharedPreferences.edit()
            .putString(USER_LIST_KEY, gson.toJson(users))
            .apply()
    }
    fun addLogToUser(userId: Long, type: EnterLog.Type) {
        val user = getUserById(userId) ?: return
        user.enterLogs.add(EnterLog(type))
        editUser(user)
    }
    companion object {
        const val USER_LIST_KEY = "user_list_key"
        const val IP_KEY = "ip_key"
        const val PORT_KEY = "port_key"
    }
}
```

Лістинг присвоєння значення порту для комп'ютера

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ipaddress;

import java.net.ServerSocket;

/**
 *
 * @author varun
 */
public class GetFreePort {
    private boolean isPortAvailable(int port) {
        boolean portAvailable = true;
        ServerSocket serverSocket = null;
        try {
```

```
            serverSocket = new ServerSocket(port);
        } catch(Exception e) {
            portAvailable = false;
        } finally {
            if (serverSocket != null) {
                try {
                    serverSocket.close();
                } catch(Exception e) {
                    e.printStackTrace();
                };
            }
        }
        return portAvailable;
    }

    public int getFreePort() {
        int port = 3000;
        while (true) {
            if (isPortAvailable(port) == true) {
                break;
            } else {
                port++;
            }
        }
        return port;
    }

}
```

Лістинг відображення IP адреси комп'ютера

```
package ipaddress;

import java.net.InetAddress;
import java.net.NetworkInterface;
import java.util.Enumeration;
import java.util.regex.Pattern;

/**
 *
 * @author varun
 */
public class GetMyIpAddress {

    //to validate ip address
    private static final Pattern = Pattern.compile(
            "^(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.){3}([01]?\\d\\d?|2[0-4]\\d|25[0-5])$");
    String ipAddresses[] = new String[10], temp;
    int j = 0;

    public static boolean validateIP(final String ip) {
        return PATTERN.matcher(ip).matches();
    }
```

```
public boolean validatePort(String portNumber) {
   if ((portNumber != null) && (portNumber.length() >= 4) && (portNumber.matches(".*\\d.*"))) {
      if( (Integer.parseInt(portNumber) > 1023))
         return true;
      else
         return false;
   }
   else
      return false;
}

public String[] ipAddress() {
   System.out.println("Printing only IPv4 Addresses");
   Enumeration e = null;
   String s = "";
   try {
      e = NetworkInterface.getNetworkInterfaces();
   }
   catch (Exception exception) {
      exception.printStackTrace();
   }
   while (e.hasMoreElements()) {
      NetworkInterface n = (NetworkInterface) e.nextElement();
      Enumeration ee = n.getInetAddresses();
      while (ee.hasMoreElements()) {
         InetAddress i = (InetAddress) ee.nextElement();
         temp = i.getHostAddress();
         //this temp contains IPv4 as well as IPv6 addresses
         //System.out.println(temp);
         if((temp.charAt(1) == '7' || temp.charAt(1) == '9') && (temp.charAt(2) == '2')) {
            ipAddresses[j] = temp;
            j++;
            System.out.println(temp);
         }
         else if(temp.charAt(0) == '1' && temp.charAt(1) == '0') {
            ipAddresses[j] = temp;
            j++;
            System.out.println(temp);
         }
      }
   }
   if (ipAddresses[0] == null) {
      ipAddresses[0] = "127.0.0.1";
   }
   return ipAddresses;
}

public static void main(String args[]) {
   new GetMyIpAddress().ipAddress();
}
//List of private IP Addresses
//10.0.0.1 to 10.255.255.254
```

```
    //172.16.0.1 to 172.31.255.254
    //192.168.0.1 to 192.168.255.254
}
```

Додаток Г. Ілюстративний матеріал

ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДО ЗАХИСТУ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Завідувач кафедри ПЗ, д. т. н., професор _____ О. Н. Романюк

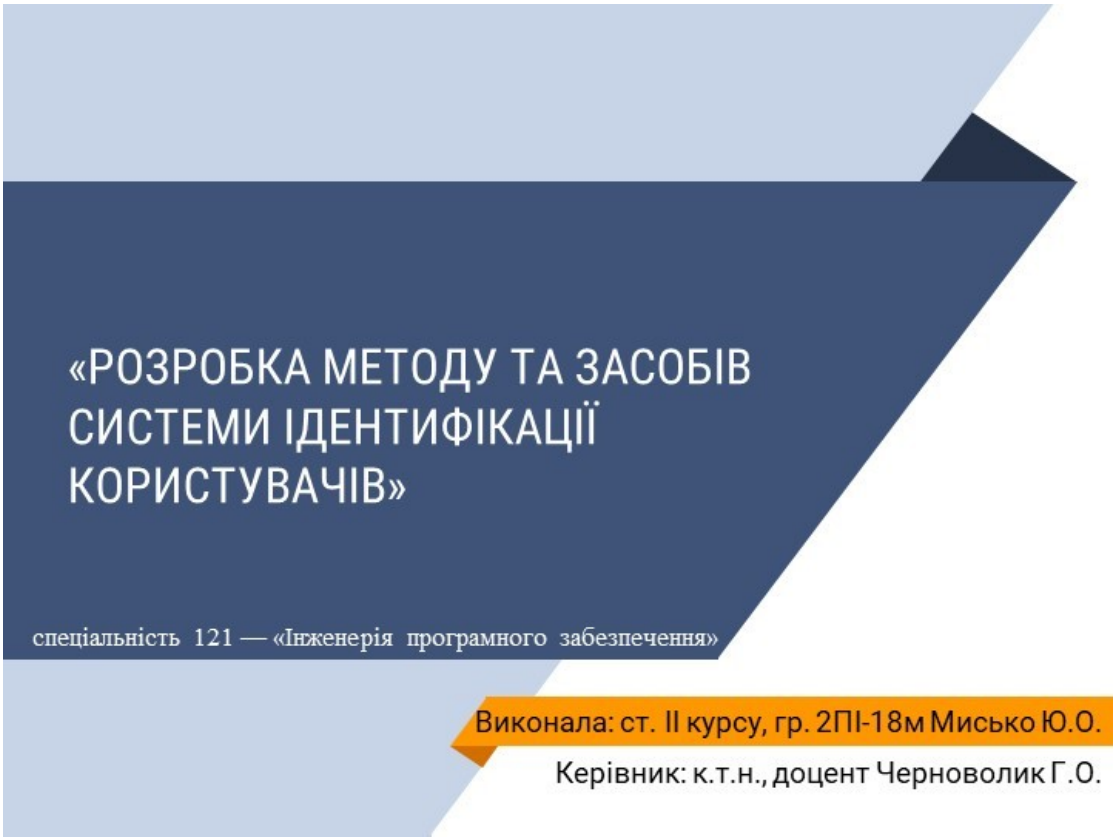
Науковий керівник, к. т. н., доцент кафедри ПЗ_____ Г. О. Черноволик

Рецензент, к. т. н., доцент кафедри КН_____ І. Р. Арсенюк

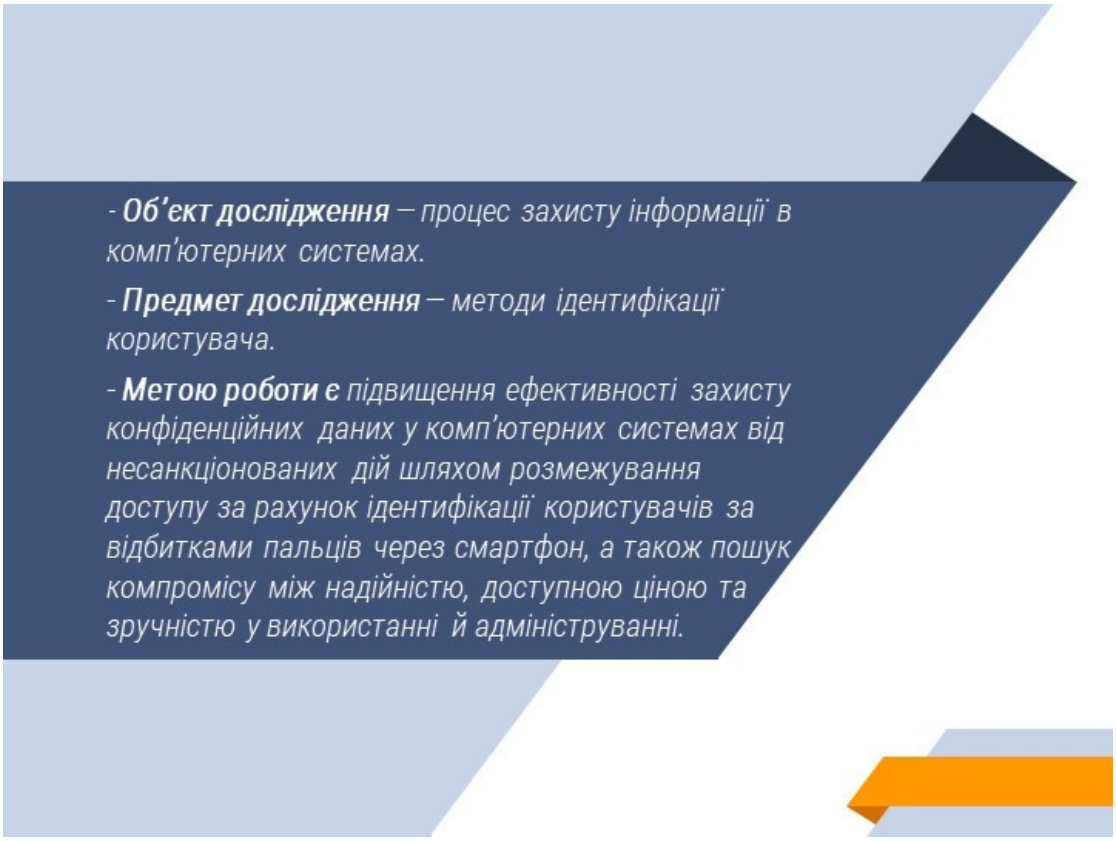
Нормоконтроль, к. т. н., доцент кафедри ПЗ _____ Г. О. Черноволик

Виконавець, студент групи 2ПІ-18м _____ Ю. О. Мисько

Слайд 1 — Титульний аркуш



Слайд 2 — Об’єкт, предмет та мета дослідження



Слайд 3 — Задачі дослідження

Задачі дослідження:

- визначити сутність та значення захисту інформації в процесі ідентифікації користувачів;
- проаналізувати недоліки та переваги існуючих підходів до ідентифікації користувачів;
- обґрунтувати використання підходу ідентифікації користувачів за відбитками пальців через смартфон як оптимальний;
- описати принцип роботи розроблених засобів системи;
- розробити метод ідентифікації користувачів за відбитками пальців через смартфон для авторизації в комп'ютерній системі;
- розробити методику розмежування доступу на основі отриманих біометричних даних;
- спроектувати базу даних користувачів;
- розробити інструкцію користувача.

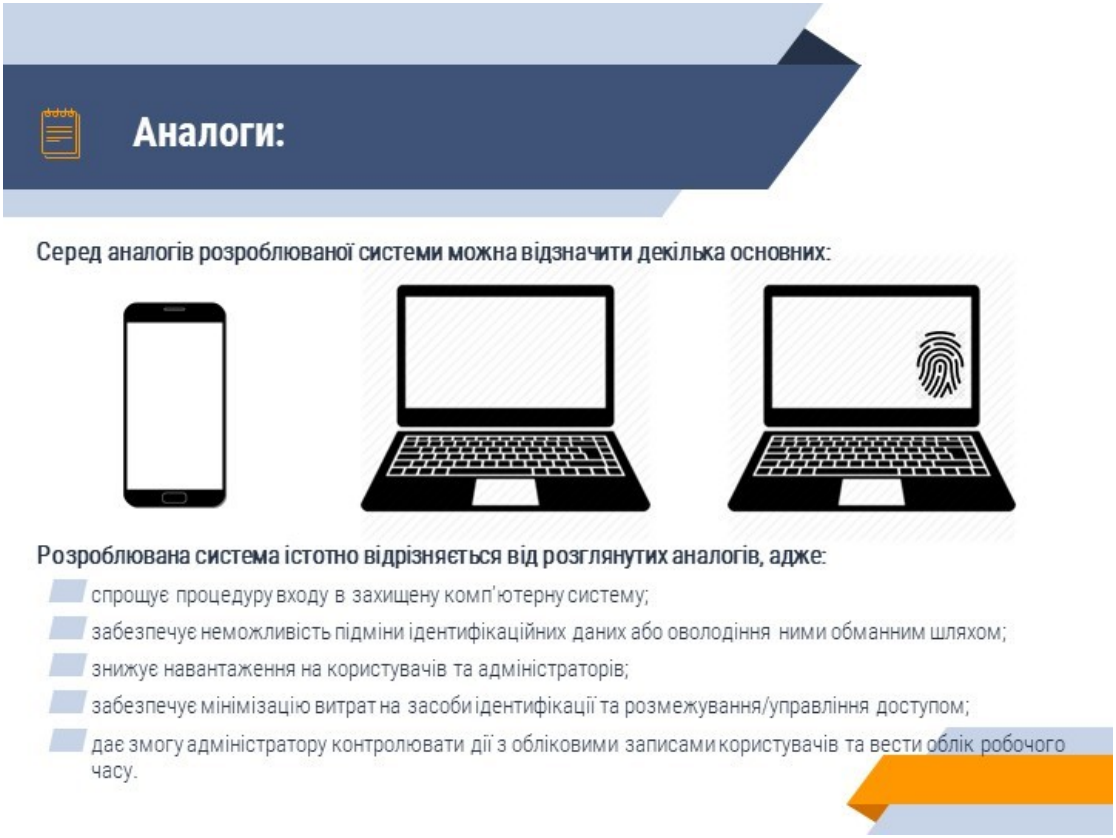
Слайд 4 — Наукова новизна

Наукова новизна:

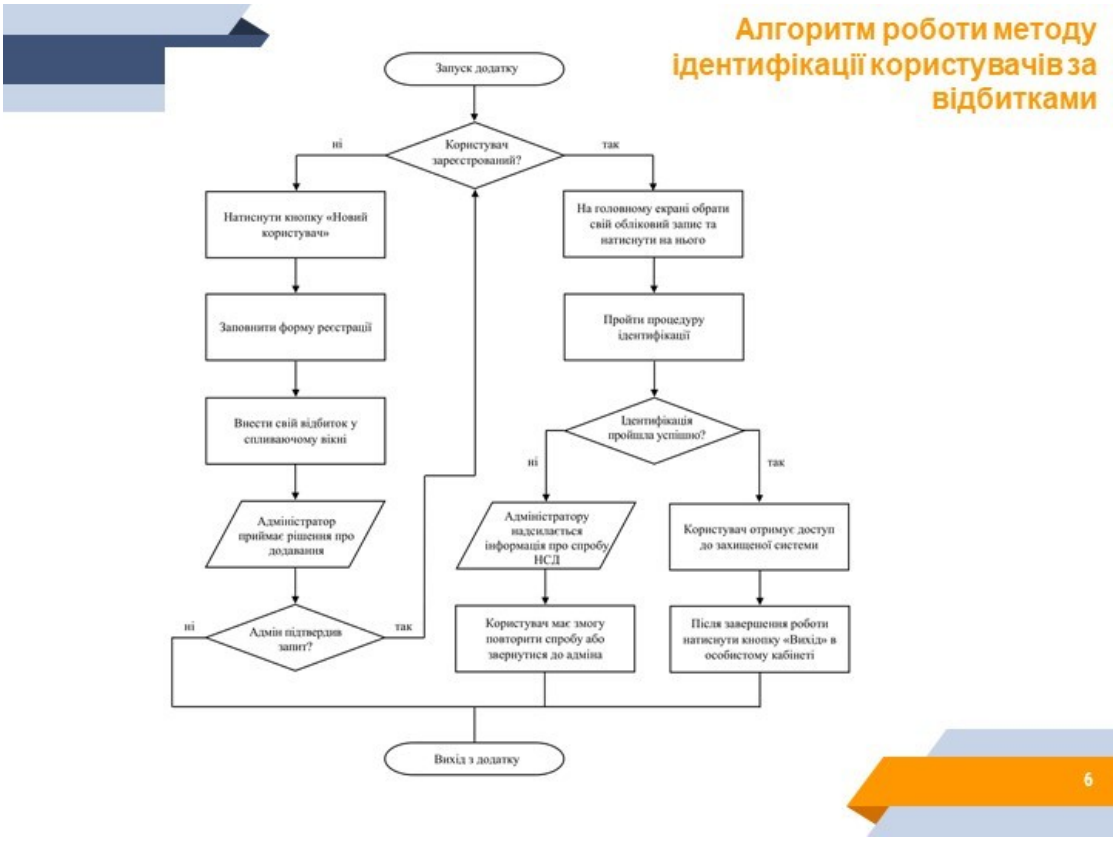
- Подальшого розвитку набув метод ідентифікації користувачів за відбитками, що відрізняється від існуючих подальшим розмежуванням доступу до комп'ютерної системи та дозволяє підвищити ефективність системи захисту даних і розширити її функціональні можливості.

- Розроблено методику розмежування доступу, яка відрізняється від існуючих дистанційним блокуванням/розблокуванням захищених комп'ютерних систем, що дозволяє зменшити час входу в систему.

Слайд 5 — Аналоги

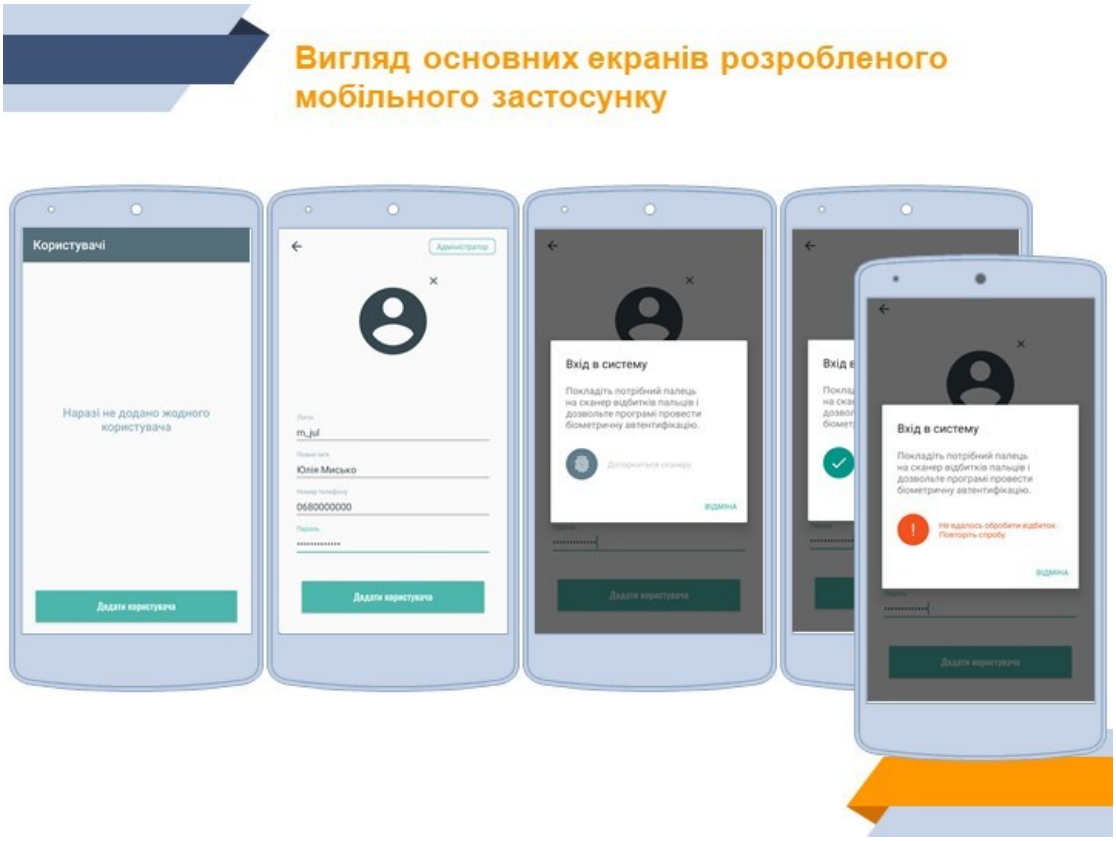


Слайд 6 — Алгоритм роботи методу ідентифікації користувачів за відбитками

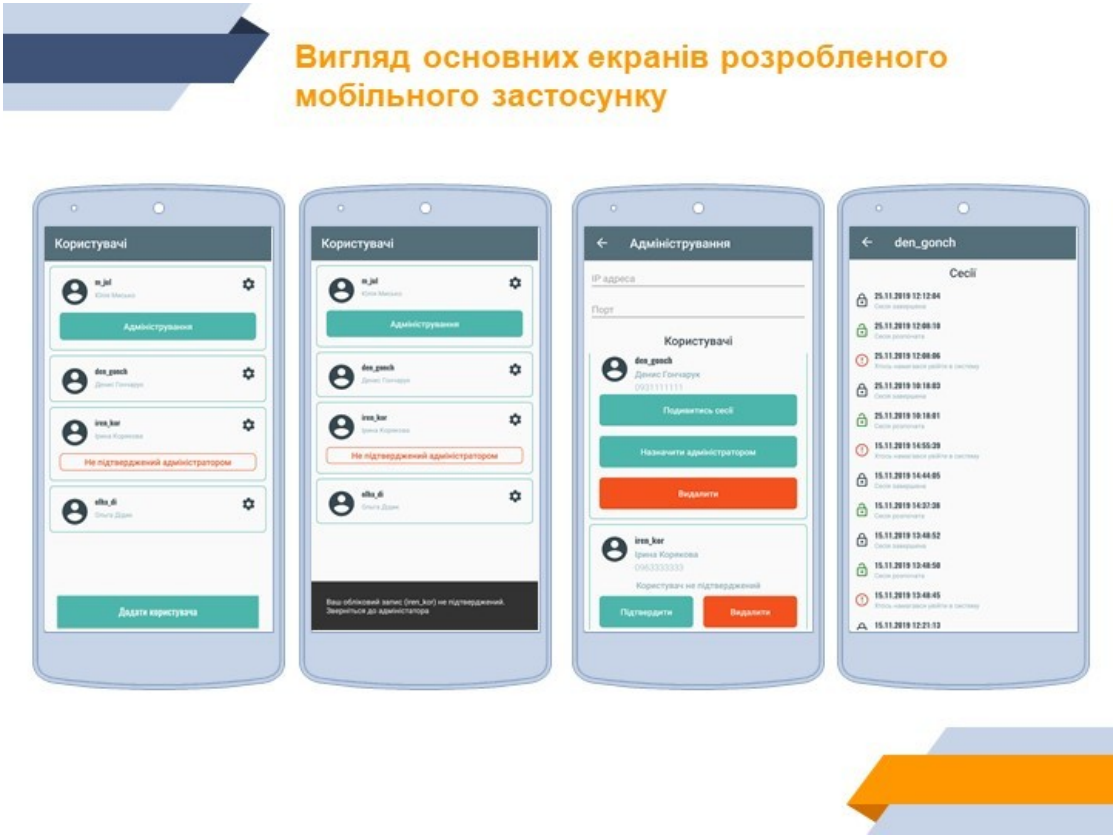


Слайд 7 — Алгоритм оцінки контролю доступу та подальше його розмежування

Слайд 8 — Вигляд основних екранів розробленого мобільного застосунку



Слайд 9 — Вигляд основних екранів розробленого мобільного застосунку



Слайд 10 — Результат роботи розробленої системи

Слайд 11 — Економічне обґрунтування

Економічне обґрунтування

- розробка має рівень комерційного потенціалу вище середнього
- розроблювання даної системи є актуальною та доцільною з точки зору економічних розрахунків
- відносна (щорічна) ефективність вкладених в наукову розробку інвестицій $E_в$ становить 230%
- термін окупності даної наукової розробки складає 0,43 роки

Отже, фінансування даної наукової розробки буде економічно доцільним.

Слайд 12 — Впровадження

Впровадження

ДОДАТОК Б Акт впровадження

УЗГОДЖЕНО
Директор ТОВ "Екзістек"
Феферман О.Д.
«2» 12 2019 року

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТЗ
д.т.н., професор Романюк О. Н.
«__» _____ 20__ року

АКТ ВПРОВАДЖЕННЯ № _____
результатів науково-дослідних робіт

Замовник ТОВ «Екзістек»
(найменування організації)
Цим актом підтверджується, що результати роботи – Розробка методу та засобів системи ідентифікації користувачів»
(найменування теми)
що виконала студентка гр. 2ПІ – 18м ВНТУ Мисько Ю.О.
(виконавці)
За договором про творчу співдружність без взаємних грошових розрахунків, на громадських засадах відповідно до теми затвердженої наказом № __ від «__» _____ 20____ р.
виконана з _____ по _____
(строки виконання)
впроваджено у ТОВ «Екзістек»
(найменування організації, де здійснювалося впровадження)
1. Вид впроваджених результатів: експлуатація програмного забезпечення
(експлуатація виробу, роботи, технології)
2. Характеристика масштабу впровадження: одиничне
(унікальне, одиничне, партія, масове, серійне)
3. Форма впровадження: дослідний зразок
4. Новизна результатів науково-дослідної роботи якісно нові
(піонерські, принципово нові, якісно нові, модифікації, модернізація старих розробок)
5. Впроваджені: для внутрішнього моніторингу та контролю.
6. Соціальний та науково-технічний ефект: спеціальне призначення
(охорона навколишнього середовища, поліпшення й оздоровлення умов праці, удосконалення структури керування, науково-технічних напрямків, спеціальне призначення)

Від виконавця:
Студентка групи 2ПІ-18м
Мисько Ю.О.

Від ТОВ «Екзістек»
Директор
Феферман О.Д.

Керівник: к.т.н., доцент кафедри
Черноволик Г.О.

Результати досліджень використовуються в компанії «ТОВ Екзістек» для підвищення захищеності конфіденційної інформації в комп'ютерних системах, для більшої мобільності, спрощеного контролю доступу та обліку робочого часу працівників.

Слайд 13 — Висновки

Висновки:

- визначено сутність та значення захисту інформації в процесі ідентифікації користувачів;
- проаналізовано недоліки та переваги існуючих підходів до ідентифікації користувачів;
- обґрунтовано використання підходу ідентифікації користувачів за відбитками пальців через смартфон як оптимальний;
- описано принцип роботи розроблених засобів системи;
- розроблено метод ідентифікації користувачів за відбитками пальців через смартфон для авторизації в комп'ютерній системі;
- розроблено методику розмежування доступу на основі отриманих біометричних даних;
- спроектовано базу даних користувачів;
- розроблено інструкцію користувача.