МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

МАЛЕНКО Дмитро Анатолійович

Безпека авторизації в мобільному телефоні за допомогою ключа NFC/ Security of Mobile Phone Authorization Using NFC Key

спеціальність: 125 – Кібербезпека

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -21
Д.А. Маленко

Науковий керівник
к.т.н., доцент Н.Г.Яцків

Кваліфікаційну роботу
Допущено до захисту:

«____» _____ ___2023 р.

Завідувач кафедри

_____ **В.В.Яцків**

ТЕРНОПІЛЬ - 2023

АНОТАЦІЯ

Кваліфікаційна робота на тему «Безпека авторизації в мобільному телефоні за допомогою ключа NFC» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека» освітньо-професійної програми «Кібербезпека» написана обсягом 96 сторінок і містить 86 ілюстрацій, 3 таблиці, 2 додатки та 30 джерел за переліком посилань.

Метою кваліфікаційної роботи є згрупування наявних даних, а також отримання додаткової інформації про фізичні та практичні нюанси реалізації методу авторизації користувачів з використанням технології NFC.

Методи досліджень. Для оцінки працездатності додатку використовувалася налагоджувальна інформація з додатку, яка виводилася за допомогою бібліотеки Log в консолі під час роботи додатку. Для тестування використовувалися наступні показники: кількість символів у паролі, швидкість авторизації користувача, дані опитування, проведеного після тестування методів авторизації користувачів, як зазначено вище.

Результати дослідження: були виявлені переваги та недоліки всіх представлених методів авторизації користувачів.

Результати роботи можуть бути застосовані при розробці власного додатку з перспективою використання технології NFC.

Ключові слова: АВТОРИЗАЦІЯ, БЕЗПЕКА, NFC.

ABSTRACT

Qualification work on "Secure authorization in mobile phones with the help of an NFC key " for the degree of "Master" in the specialty 125 "Cybersecurity" educational and professional program "Cybersecurity" is written in 96 pages and contains 86 illustrations, 3 tables, 2 appendices and 30 source according to the list of links.

The purpose of the qualification work is to group the available data, as well as to obtain additional information about the physical and practical nuances of implementing the user authorization method using NFC technology.

Research methods. To evaluate the application's performance, we used debugging information from the application, which was displayed using the Log library in the console while the application was running. The following indicators were used for testing: the number of characters in the password, user authorization speed, and survey data from the user authorization methods testing, as described above.

Research results: the advantages and disadvantages of all the presented methods of user authorization were identified.

The results of the work can be used to develop your own application with the prospect of using NFC technology.

Keywords: AUTHORIZATION, SECURITY, NFC.

ЗМІСТ

ВСТУП

Актуальність роботи. NFC (Near Field Communication) - це високочастотна технологія бездротового зв'язку малого радіусу дії, яка дозволяє безконтактно обмінюватися даними між мобільними телефонами, смарт-картами, платіжними терміналами, системами контролю доступу та іншими пристроями.

Підтримка смартфонів почалася з випуском Android 4.0 "Ice Cream Sandwich" у 2011 році, а також з випуском 2014 року. Apple iPhone 6 і смартфонів на базі WindowsPhone 8.1.

Смартфон з підтримкою NFC може бути як смарт-карткою, що використовується для оплати, так і зчитувачем, за допомогою якого можна переказувати гроші між телефоном і карткою або двома телефонами, а також обмінювати реальні банківські картки з підтримкою ISO / IEC 14443 [1] (стандарт безконтактних карток) на віртуальні.

Оскільки чіп NFC здатний передавати дані в обох напрямках і не вимагає автентифікації пристрою, його можна використовувати як просту і більш зручну заміну Bluetooth. Використовуючи NFC, ви можете обмінюватися посиланнями, паролями, контактами та іншими даними між смартфонами. Підтримання даних користувача в актуальному стані та можливість доступу до них з будь-якого пристрою є основоположним принципом забезпечення зручності взаємодії користувача з IT-системами. Кількість пристроїв для особистого користування більше не обмежується телефоном і комп'ютером. Для забезпечення швидкої аутентифікації може бути використана технологія NFC, яка дозволяє не тільки швидко проходити аутентифікацію, але й використовувати складні паролі, які не потрібно запам'ятовувати.

У цій роботі розглядаються різні методи автентифікації, кожен з яких має свої переваги та недоліки. Відзначено відмінності між ними. В якості рішення для забезпечення безпеки автентифікації було обрано надійну технологію NFC, яка все ще розвивається. Розглянуто її безпеку як каналу передачі даних, а також як методу авторизації, на який можна покластися при розробці програмного забезпечення.

Мета і завдання дослідження. Метою цієї роботи є розробка нового метод автентифікації користувачів та порівняти його з відомими методами автентифікації на основі дослідження, проведеного на групі користувачів. В рамках цієї роботи було розроблено конкретний додаток, який використовує традиційні методи автентифікації та метод автентифікації на основі NFC і дозволяє користувачеві визначити переваги та недоліки представлених методів на основі певних критеріїв.

Об'єкт дослідження – процеси збору, аналізу та порівняння отриманих даних в результаті авторизації різними методами;

Предмет дослідження – метод авторизації за допомогою технології NFC та порівняння його з іншими популярними методами.

Методи досліджень. Для розв'язання поставлених задач у даній кваліфікаційній роботі використано: методи авторизації та метод авторизації за допомогою технології NFC.

Наукова новизна одержаних результатів. Визначені плюси та мінуси використання технології NFC в якості методу авторизації на мобільних пристроях.

Практичне значення отриманих результатів. Створено мобільний додаток та серверну частину, за допомогою котрих було досліджено якісні параметри методу авторизації за допомогою технології NFC у порівнянні з іншими популярними методами авторизації.

Публікації та апробація КР.

1. Маленко Д.А. Метод автентифікації користувачів за допомогою стандарту NFC. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ - 2023), Тернопіль, 2023. – С.187-189.

2. Маленко Д.А. Основний принцип роботи NFC-пристроїв та їхня безпека. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. – С. 114-116.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз запатентованих методів з використанням NFC

Пітер Конрад Тисовскі у своїй статті "Система ближнього зв'язку (NFC), що забезпечує аутентифікацію географічного положення за допомогою міток NFC і пов'язані з нею методи" представив альтернативний метод аутентифікації за допомогою міток, що містять геодані [2]. Система передбачає наявність телефону, який може отримувати інформацію про місцезнаходження не тільки з GPS. Такий підхід підходить для великих і малих міст з недостатнім покриттям мобільного зв'язку. При такому підході користувач може бути авторизований за допомогою методу автентифікації на основі NFC лише за умови, що він перебуває в довіреній зоні, де встановлені ключові мітки.

У статті "Wearable Authentication Device" Джеймса Ван Боша (James A Van Bosch) та Павла Шостака (Pavel A. Shostak) представлено концепцію носимого пристрою, який має вдосконалене з'єднання з NFC-зчитувачем [3]. Його компактність дозволяє конструкції вписатися в розміри каблучки. Принцип роботи полягає в розташуванні двох антен замість однієї в приймальному пристрої. Таким чином, пристрій покривається вдвічі більшою зоною прийому сигналу, за умови, що антена знаходиться на протилежній стороні від початкової. Це дає можливість швидко встановити контакт замість того, щоб вибирати місце розташування антени за замовчуванням.

У статті Юріса Клоновса, Крістоффера К'єльдгаарда Петерсена, Хеннінга Олесена, Аллана Хаммершоя під назвою "Підтвердження особи на ходу: розробка мобільної системи біометричної автентифікації на основі ЕЕГ" йдеться про використання електроенцефалографа та NFC-міток [4]. Основна увага приділяється електроенцефалографічному підпису. Система передбачає біометричне розпізнавання людини на основі її мозкової активності. Згідно зі статтею, активність мозку кожної людини є унікальною. Щоб уникнути випадкової аутентифікації клієнта, до схеми аутентифікації вирішили додати технологію NFC і розпізнавання обличчя. Система знаходиться на ранніх стадіях розробки і була

розроблена з урахуванням технологічного прогресу, коли частини системи можуть бути зроблені з більш компактних пристроїв.

Цікавий підхід застосовано в статті Мухаммада Касима Саїда та Коліна Д. Уолтера "Аутентифікація міток у режимі реального часу" (Muhammad Qasim Saeed and Colin D. Уолтера "Аутентифікація за допомогою NFC-мітки в автономному режимі". Суть полягає в тому, щоб використовувати NFC-мітки, коли у вас немає підключення до інтернету. Цілісність даних мітки забезпечується цифровим підписом. Звичайні протоколи не забезпечують захист від використання копій таких міток. У цій статті пропонується варіант запобігання використанню копій тегів. Рішення базується на протоколі "запит-відповідь" з використанням криптографії з відкритим ключем та PKI [5]. Запропонований варіант сильно відрізняється від оригінального підходу. Ці підходи стосуються автентифікації тегів NDEF.

У статті "Мобільний пристрій бездротового зв'язку з функціями розблокування та зміни даних мітки бездротового зв'язку ближнього радіуса дії (NFC) та пов'язані з ними методи" розповідається про розблокування пристрою за допомогою NFC-мітки. Цей метод розблокування пристрою включає два варіанти зчитування. Перший варіант - просте зчитування і розблокування. Другий варіант - розблокування пристрою шляхом перезапису ключа за розкладом [6].

У статті "Автентифікація RFID/NFC на основі еліптичної кривої з використанням датчика температури для захисту від ретрансляційних атак" обговорюється, як уникнути перехоплення інформації зовнішніми слухачами. Також розглянуто недоліки представленого методу автентифікації. Основна ідея статті полягає в тому, що температура поверхні мітки змінюється, що впливає на сигнал і погіршує можливість його перехоплення. Через це змінюється час передачі даних на відстань і сигнал спотворюється [7].

Стаття Філіпа Хьюінсона "Двофакторна автентифікація користувачів з використанням зв'язку ближнього поля" описує двосторонню перевірку особи. Унікальний ключ зберігається в пам'яті телефону. При першому підключенні мітки до зчитувача автентифікація проходить успішно, в мітці зберігається окремий

ключ, який ідентифікує мітку. При наступному підключенні мітки відбувається порівняння ключа, який ідентифікує телефон, і ключа, який ідентифікує мітку. Наявність двох ключів для перевірки підвищує безпеку розблокування. Надійність стає ще більшою завдяки динамічній зміні ключа мітки [8].

Стандартним протоколом для передачі даних на NFC-мітку є NDEF. У статті "Безпечний та легкий протокол аутентифікації для сервісів на основі NFC-міток" описано потенційну можливість пошкодження даних на мітці з даним протоколом та його наслідки. В якості рішення пропонуються додаткові протоколи, які є більш вигідними при використанні недорогих міток з меншим об'ємом пам'яті. Додатковими перевагами запропонованих протоколів є більш безпечна реєстрація та зберігання даних, що запобігає підробці, DoS, модифікації даних та фішинговим атакам[9].

Цікава реалізація антени NFC для телефонів запропонована в статті "Дизайн антени NFC для низькопроникного феромагнітного матеріалу", авторами якої є Б'юнгдже Лі, Б'єнгкван Кім, Френсіс Харакевич, Б'єнггві Мун, Хюнву Лі. У статті описується нова конструкція антени, заснована на зміні внутрішньої петлі та структури обмотки. Використано феритно-полімерний композит, який є кращим і ефективнішим за дорогі феритові елементи. Запропонована антена має на 23% більший діапазон зчитування і на 65% кращу модуляцію навантаження порівняно з існуючими на ринку аналогами [10].

У статті "Автоматичне регулювання імпедансу антени для зв'язку ближнього поля (NFC)" запропоновано новий підхід до зчитування різних типів пристроїв. У телефоні модуль NFC влаштований інакше, ніж у банківській картці або NFC-мітці. Для забезпечення повної сумісності пристрій, який він зчитує, повинен містити кілька типів антен. Як рішення було запропоновано використовувати одну антену з автоматичним налаштуванням антени на основі цифрового підстроювання конденсатора. У цьому випадку антена підлаштовується під передавальний пристрій, ефективно ідентифікує його та обмінюється інформацією[11].

1.2 Технології

1.2.1 Android

Android – операційна система для смартфонів, планшетів, електронних книг, цифрових плеєрів, розумних годинників, ігрових консолей, нетбуків, окулярів Google Glass, телевізорів, систем автоматичного наведення та інших пристроїв. Операційна система базується на ядрі Linux та власній реалізації віртуальної машини Java від Google. Спочатку вона була розроблена компанією AndroidInc., яка була придбана Google у 2005 році. Після цього Google ініціював створення OpenHandsetAlliance (OHA), який зараз займається підтримкою та подальшим розвитком платформи [12].

Android – популярна операційна система для мобільних пристроїв - телефонів і планшетів. Ця система має багато відмінних рис, які роблять її впізнаваною і привабливою для великої кількості користувачів у всьому світі. Операційна система Android невибаглива і може працювати в різних конфігураціях. Саме тому більшість світових виробників оснащують свої пристрої саме цією операційною системою, оскільки інші програмні продукти розробляються для конкретних пристроїв, які відповідають певній специфікації. Така гнучкість Android пояснюється тим, що система побудована на ядрі Linux, яке має відкритий вихідний код, що надає необмежені можливості для розробників. Операційна система дозволяє встановлювати додатки з офіційного репозиторію Google, який надає найбільшу в світі базу даних програм. Це пов'язано з тим, що будь-який розробник може самостійно написати будь-яку програму для пристрою і завантажити її в інтернет-магазин Google Play. Ця можливість реалізується також завдяки відкритості операційної системи. Варто зазначити, що додатки для Android-пристроїв можна встановлювати як безпосередньо з телефону чи планшета, так і з комп'ютера, завантаживши файл .apk та встановивши його на пристрій. Відмінною особливістю Android є інтеграція з сервісами Google - Gmail, голосовим пошуком тощо. Операційна система швидко реагує на натискання користувача, встановлює

та завантажує необхідні програми та файли зі швидкістю, яка не поступається іншим сучасним мобільним операційним системам.

1.2.2 Android Studio

Android Studio – інтегроване середовище розробки (IDE) для роботи з платформою Android, анонсоване 16 травня 2013 року. На конференції Google I/O [13].

IDE було вільно доступне з версії 0.1, випущеної в травні 2013 року, а потім перейшло в режим бета-тестування, починаючи з версії 0.8, яка була випущена в червні 2014 року. Перша стабільна версія 1.0 вийшла в грудні 2014 року, після чого було припинено підтримку плагіна Android Development Tools (ADT) для Eclipse.

Android Studio, заснована на програмному забезпеченні JetBrains IntelliJ IDEA, є офіційним інструментом для розробки додатків для Android. Це середовище розробки доступне для Windows, OS X та Linux. 17 травня 2017 року. Під час щорічної конференції Google I/O компанія Google оголосила про підтримку мови Kotlin, яка використовується в Android Studio, як офіційної мови програмування для платформи Android, на додаток до Java і C++.

1.2.3 PhpStorm

PhpStorm – комерційне, крос-платформне інтегроване середовище розробки для PHP. Розроблене компанією JetBrains на основі платформи IntelliJ IDEA.

PhpStorm – це інтелектуальний редактор PHP, HTML і JavaScript з аналізом коду в реальному часі, запобіганням помилкам у коді та інструментами автоматичного рефакторингу для PHP і JavaScript. Завершення коду в PhpStorm підтримує специфікації PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1 і 7.2 (сучасні і традиційні проекти), включаючи генератори, процедури, простори імен, закриття, типи і синтаксис коротких масивів.

Існує повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

PhpStorm базується на платформі IntelliJ IDEA, написаній на мові Java. Користувачі можуть розширити функціональність середовища розробки, встановивши плагіни, призначені для платформи IntelliJ, або написавши власні плагіни [14].

1.2.4 Java

Java – паралельна, заснована на класах, об'єктно-орієнтована мова програмування загального призначення. Створена робочою групою під керівництвом Джеймса Гослінга з компанії Sun Microsystems. Java - це мова для створення вихідних програм, які компілюються в байт-код - форму, що виконується віртуальною машиною. Мова характеризується сильною типізацією. Її основні концепції запозичені з Smalltalk (віртуальна машина, управління пам'яттю) та C++ (значна частина синтаксису та ключових слів) [15].

1.2.5 PHP

PHP - інтерпретована скриптова мова програмування, призначена для генерації веб-сторінок і створення веб-додатків у реальному часі.

PHP найчастіше використовується для написання сценаріїв на стороні веб-сервера, але також може застосовуватися для обробки командного рядка і навіть для написання програм, що працюють у графічному режимі (наприклад, з бібліотекою GTK+, використовуючи розширення PHP-GTK). Реалізація PHP разом з веб-сервером Apache і сервером баз даних MySQL називається платформою AMP (в середовищі Linux - LAMP, в Windows - WAMP) [16].

1.3 Сучасні методи авторизації

Сьогодні існує багато різновидів методів авторизації, і їхня кількість постійно зростає. З'являються нові методи автентифікації для користувачів, про

які раніше не думали або важко було уявити, що їх можна реалізувати в мобільному світі.

Кожен метод має свої переваги та недоліки. Кожен метод був обраний з урахуванням безпеки, зручності, доступності розробки, універсальності та популярності додатку.

Початкові запропоновані методи авторизації:

1. Біометричне розпізнавання
2. Класичне гасло
3. Пароль замасковано
4. Авторизація за допомогою QR-коду
5. Типова авторизація
6. Авторизація за допомогою кнопок соціальних мереж
7. Авторизація за допомогою SMS-сервісу
8. Автентифікація Bluetooth
9. Голосова авторизація

Авторизація за допомогою біометричного розпізнавання означає не лише використання різних датчиків і сканерів, а й розпізнавання різних частин тіла. Ключем до успішної автентифікації є параметри частин тіла, які мають унікальні характеристики, комбінація яких повторюється з дуже низькою ймовірністю. Чим рідше повторюється комбінація параметрів, тим надійнішим є метод біометричної автентифікації. Методи біометричної автентифікації можуть спеціалізуватися на різних частинах тіла. Найчастіше біометричні методи автентифікації базуються на рисах обличчя, сітківці ока, геометрії руки, відбитках пальців або декількох пальців. Приклади використання показані на рисунках 1.1, 1.2, 1.3.

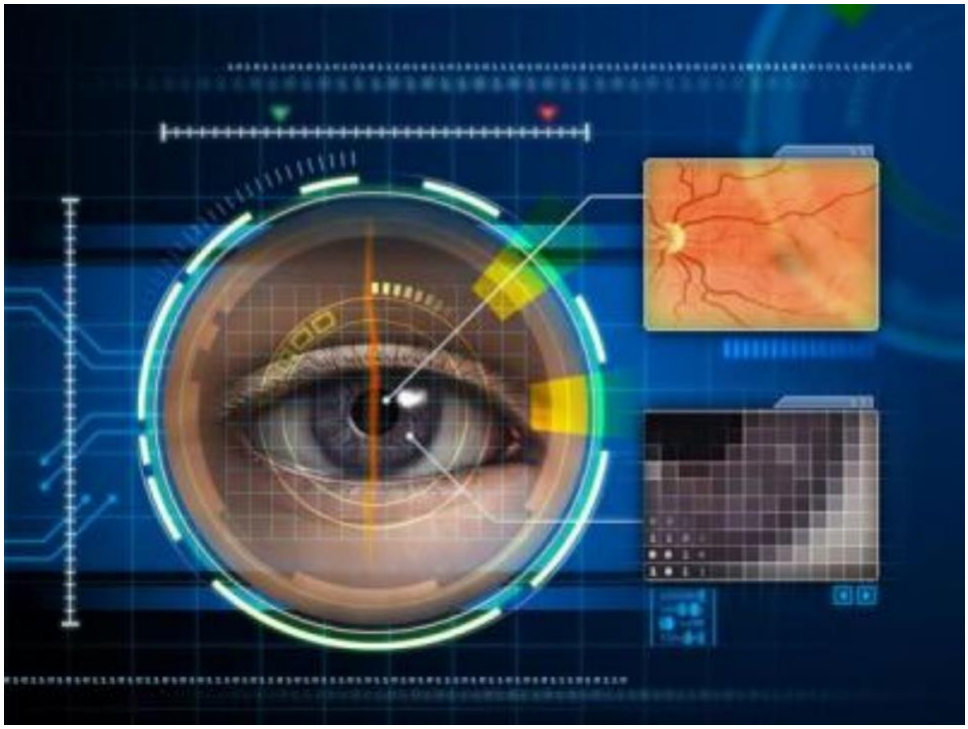


Рисунок 1.1 – Метод автентифікації на основі сітківки ока [19]

Рисунок 1.2 – Метод автентифікації за допомогою розблокування за відбитком пальця [20].

Рисунок 1.3 – Метод автентифікації за допомогою біометричного розпізнавання обличчя [21]

Розпізнавання відбитків пальців на сьогоднішній день є одним з найпопулярніших методів автентифікації. Щоб скористатися цим методом автентифікації, просто прикладіть палець до зчитувача відбитків пальців, і через деякий час пристрій буде розблоковано, а користувач увійде в систему з високим ступенем точності. Біометричний метод автентифікації з використанням розпізнавання відбитків пальців вважається настільки надійним, що його використовують як основний метод автентифікації користувачів у банківських додатках, які вимагають високого рівня безпеки.

Також біометричне розпізнавання залежить від датчика, який використовується як інструмент для зчитування ключової інформації. Наприклад, біометричне розпізнавання обличчя може мати кілька варіацій цього методу автентифікації, оскільки використовуються різні датчики, які мають різні підходи до обробки інформації. Наприклад, різні методи біометричної автентифікації за обличчям можуть використовувати камеру, термочутливу камеру або тривимірний датчик.

Вхід за допомогою класичного пароля - один з перших методів автентифікації, який почав використовуватися для загального користування. До сьогодні цей метод авторизації користувачів залишається одним з

найпопулярніших методів авторизації користувачів. Приклад цього методу авторизації користувача показано на рисунку 1.4.

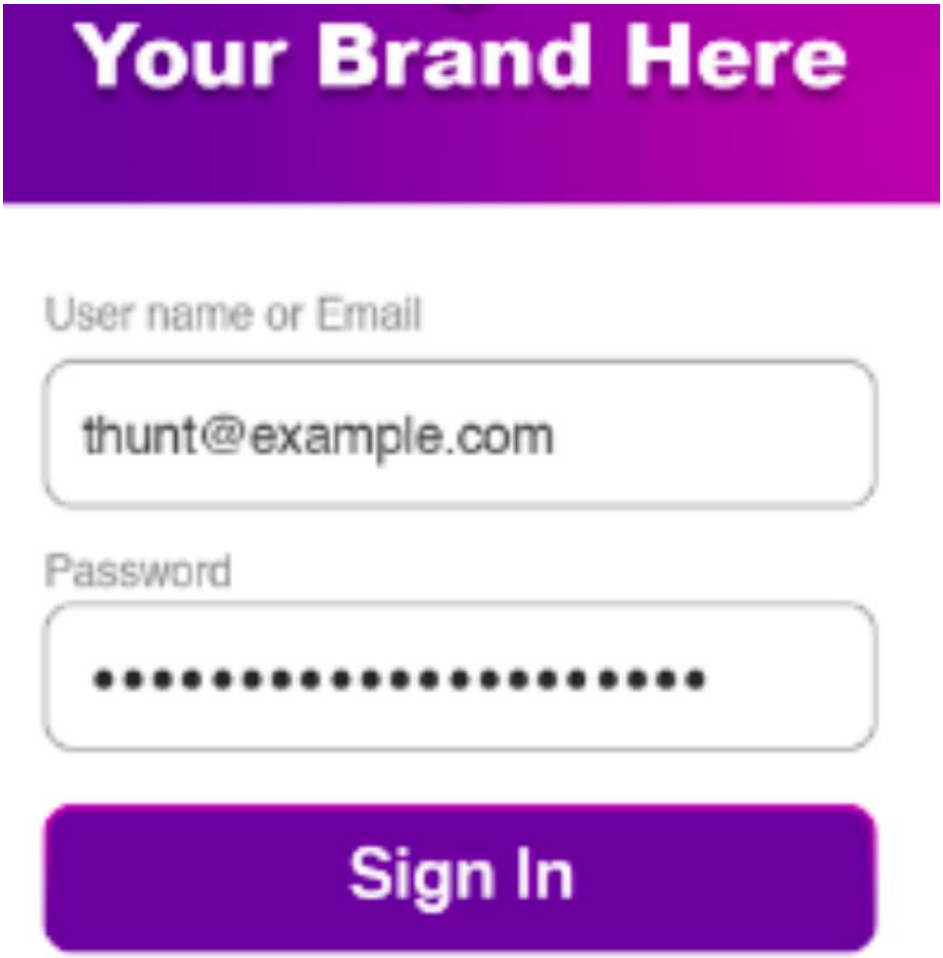


Рисунок 1.4 – Класичний метод парольної автентифікації [22]

Завдяки своїй простоті, метод автентифікації за допомогою класичного пароля є безпечним як для користувачів, так і для розробників. Цей метод автентифікації простий і складається з двох полів. В одному полі потрібно ввести логін. Зазвичай це адреса електронної пошти, номер телефону або унікальний логін, призначений користувачем, який вводиться в поле логіна. У другому полі вводиться пароль. При введенні пароля можна використовувати паролі різної довжини та складності. В результаті ця проста комбінація забезпечує необхідний рівень захисту для систем, які не потребують високого рівня безпеки.

Маскований пароль складається з декількох полів, в які вводиться один символ. Кожне поле відповідає за певний символ пароля. Деякі поля є неактивними, і введення даних в них при введенні пароля не призведе до успіху. Цей метод

автентифікації не є найпопулярнішим або найзручнішим, проте він забезпечує високий рівень безпеки і в основному використовується в банківських додатках. На рисунку 1.5 показано, як виглядає цей метод.

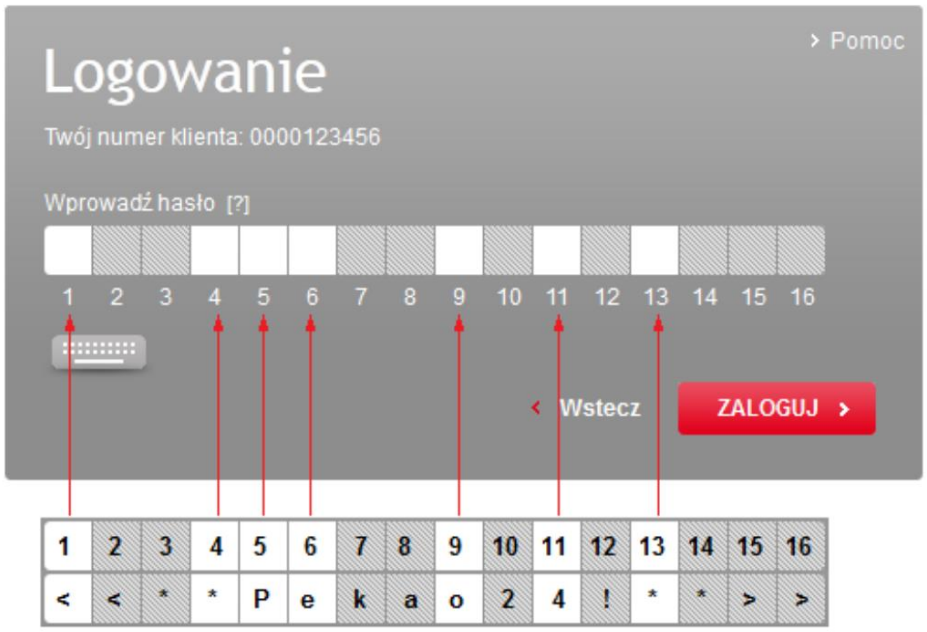


Рисунок 1.5 – Метод автентифікації за допомогою маскованого пароля [23]

Розблокування за допомогою QR-коду відбувається за допомогою модуля камери. Камера наводиться на закодоване зображення, а інформація, закодована в зображенні, перетворюється пристроєм в цифрові дані. Цей метод має низький рівень безпеки, оскільки закодоване зображення може бути сфотографоване, а потім роздруковане або збережене в пам'яті іншого пристрою. Цей метод автентифікації використовується дуже рідко, оскільки існує велика ймовірність того, що зображення з ключем буде підглянуто. Приклад реалізації цього методу показано на рисунку 1.6.

Рисунок 1.6 – Метод автентифікації за допомогою QR-коду [24]

Один банк в Україні адаптував метод авторизації з використанням QR-коду як одноразового ключа для доступу до функцій банкомату, а також для доступу до свого профілю на сайті за допомогою телефону. Використання методу авторизації за допомогою QR-коду як одноразового ключа означає взаємодію між платформами.

Для авторизації користувача за допомогою візерунка необхідно намалювати на екрані певне число, обмежене кількістю та розташуванням точок на екрані, які були раніше встановлені користувачем як пароль. Особливостями графічного методу розблокування паролем є швидке введення пароля та простота використання. При використанні складної комбінації складно підібрати пароль з першого разу. Оскільки графічний метод розблокування паролем має середній рівень безпеки, високий рівень зручності і користується популярністю серед сучасних методів розблокування. Зовнішній вигляд цього методу показано на рисунку 1.7.

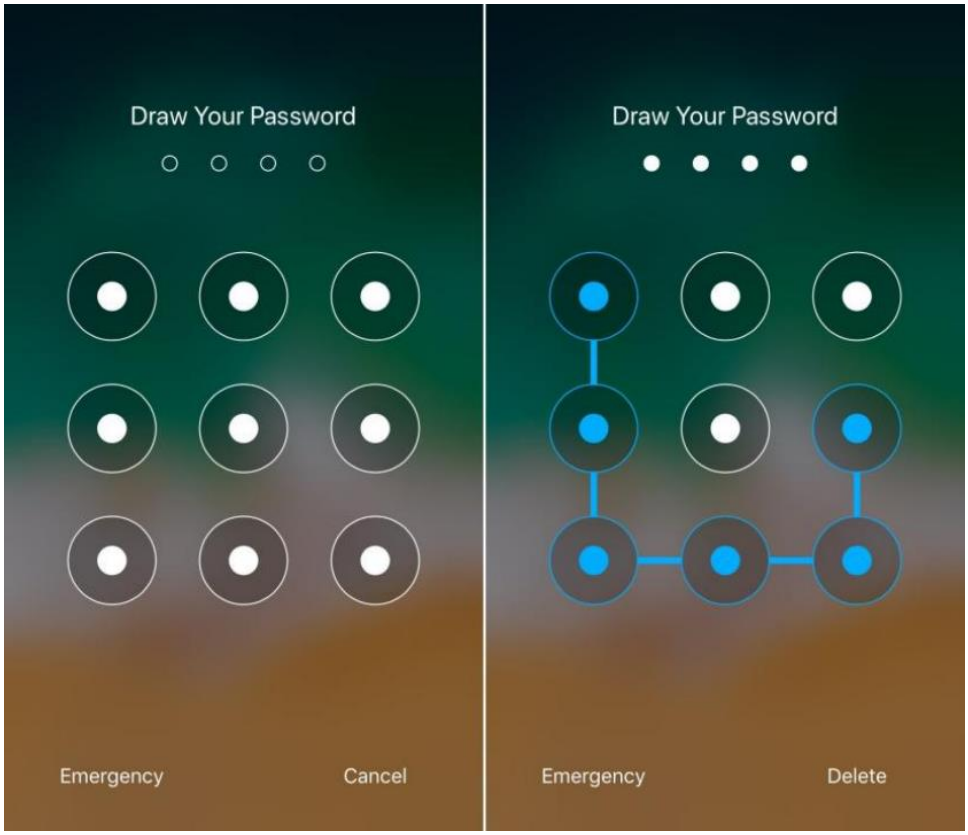


Рисунок 1.7 – Метод аутентифікації за формулою [25]

Метод автентифікації за шаблоном також реалізовано як один з методів розблокування екрану в Android.

Метод автентифікації за допомогою кнопок соціальних мереж дуже ефективний, якщо вам потрібно швидко створити обліковий запис і зберегти особисту інформацію користувача. Використовуючи цей метод, ви можете легко зареєструватися і так само швидко увійти в систему. Якщо обліковий запис зберігається на пристрої, буде запропоновано право на читання даних облікового запису для додатку. Якщо користувач раніше не був зареєстрований у соціальній мережі, додаток спочатку попросить користувача увійти в соціальну мережу. На рисунку 1.8 показано, як виглядає цей спосіб.

Рисунок 1.8 – Метод автентифікації за допомогою кнопок соціальної мережі [26]

Метод SMS-авторизації передбачає використання телефонного номера користувача. На вказаний номер надходить одноразове SMS-повідомлення. Довжина та складність надісланого коду залежить від послуги, для якої надсилається SMS-повідомлення. Найчастіше це набір цифр, оскільки паролі щоразу генеруються з нуля і ймовірність підібрати один з них дуже низька.

На рисунку 1.9 показано, як працює цей метод.

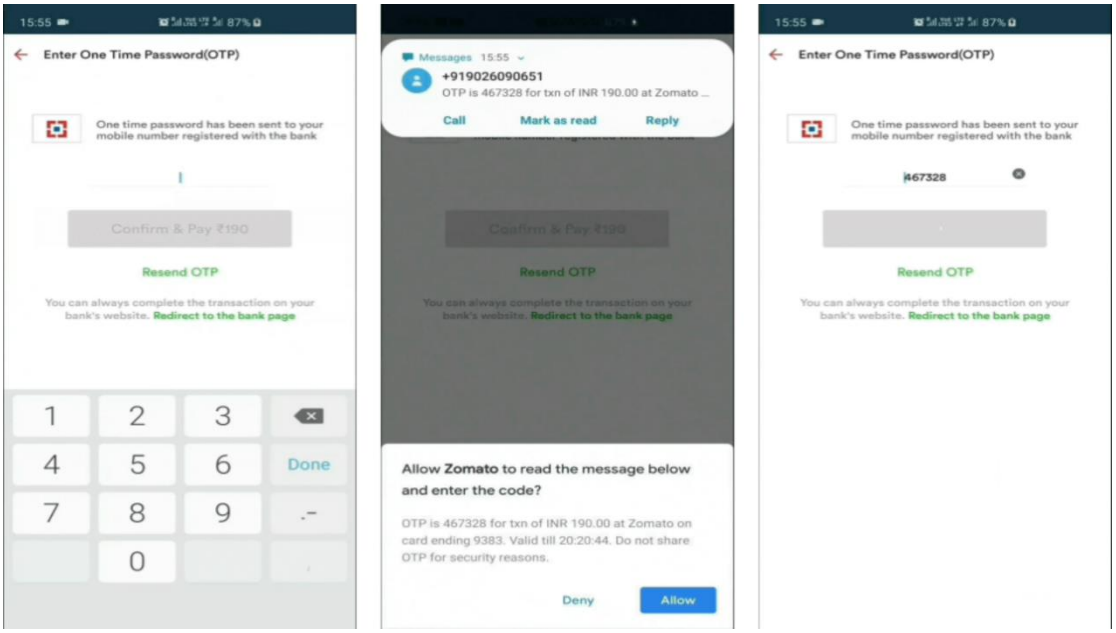


Рисунок 1.9 – Метод аутентифікації за допомогою кодів, отриманих з SMS [27]

Також є можливість для програми самостійно прочитати SMS і ввести код у поле для введення пароля. Для цього необхідно погодитися на надання відповідних прав доступу. З одного боку, такий спосіб авторизації забезпечує швидкий і безпечний вхід, з іншого боку, авторизація в додатку постійно залежить від номера, до якого прив'язаний обліковий запис. Більш поширеним методом авторизації є комбінація двох методів авторизації. Спочатку користувач вводить пароль, а потім отримує SMS-повідомлення на пристрій, щоб процес авторизації можна було завершити введенням одноразового перевірочного пароля.

Метод автентифікації Bluetooth передбачає використання одного додаткового пристрою, який виконує роль ключа. Якщо пристрій підключено, виконується автентифікація користувача. Ключем є MAC-адреса підключеного пристрою. Цей метод авторизації також вбудовано в один із методів розблокування екрана в Android. На рисунку 1.10 показано вигляд конфігурації цього методу.

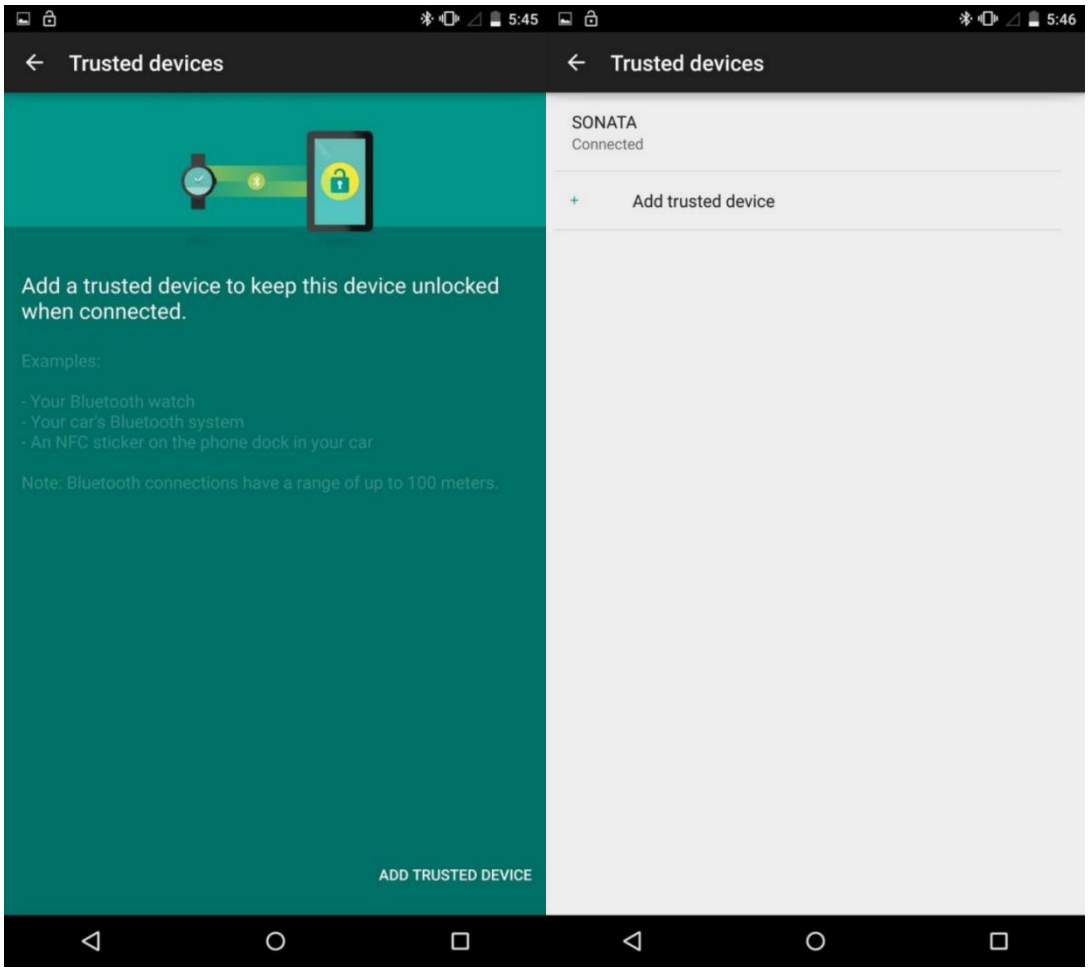


Рисунок 1.10 – Метод автентифікації Bluetooth [28]

Як видно з рисунків, ключем може бути як один, так і декілька пристроїв. Крім того, смартфон може не тільки включати в себе метод Bluetooth-аутентифікації, але і бути ключем. Такий підхід дуже часто використовується сьогодні в системах розумного будинку, наприклад, для відкриття вхідних дверей. Принцип роботи в цьому випадку може полягати в розблокуванні дверей, коли смартфон знаходиться в зоні виявлення системи безпеки, або в розблокуванні дверей натисканням кнопки в додатку смартфона. На рисунку 1.11 показано приклад застосунку для цього методу.

Рисунок 1.11 – Розблокування пристрою розумного будинку телефоном за допомогою Bluetooth [29]

Голосова автентифікація дозволяє користувачеві розблокувати пристрій, вимовивши певну фразу. Під час автентифікації за допомогою голосового методу голос спочатку записується, а потім аналізується, і із записаного голосу виділяються певні параметри, які порівнюються із заданими параметрами. Метод

голосової автентифікації, як і метод автентифікації через Bluetooth, є частиною системи розблокування смарт-екрану Android. На рисунку 1.12 показано, як працює цей метод.

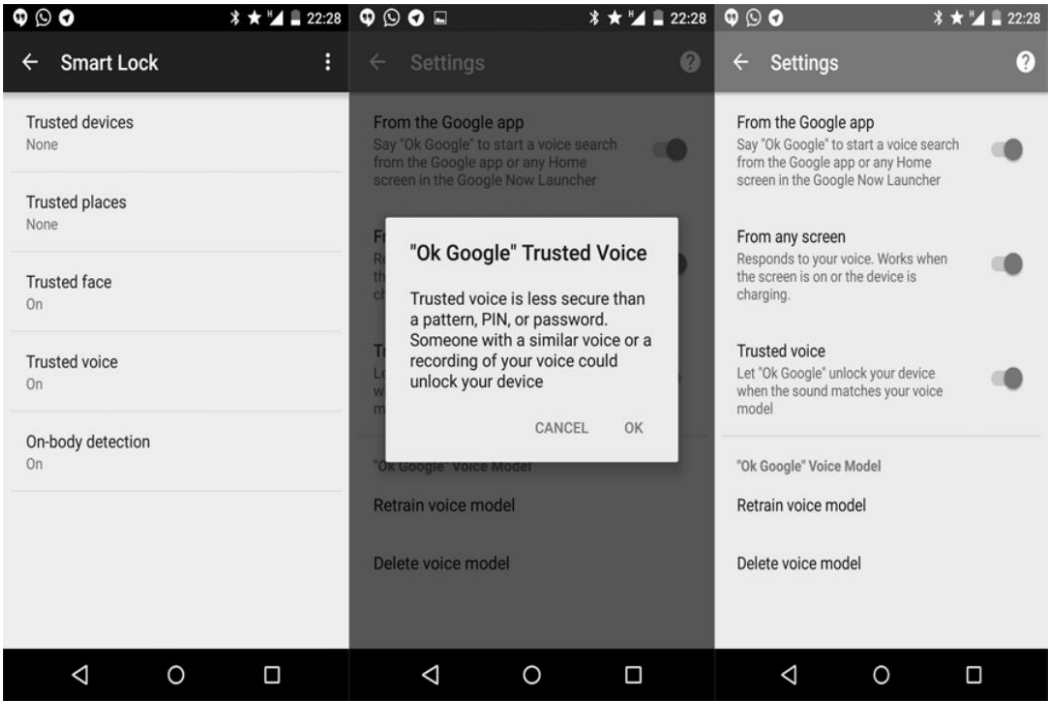


Рисунок 1.12 – Метод розблокування автентифікації за допомогою розпізнавання голосу [30]

У цій дипломній роботі представлено новий метод автентифікації з використанням технології NFC. Метод автентифікації користувачів за допомогою технології NFC не є новим, і в даний час мало використовується для розробки додатків для Android. Представлений метод автентифікації користувачів буде описано більш детально в наступному розділі.

2 МЕТОД АВТЕНТИФІКАЦІЇ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ NFC

У цьому розділі представлено метод автентифікації з використанням технології NFC. Метод автентифікації користувачів за допомогою технології NFC не є новим, і в даний час мало використовується для розробки додатків для Android. Запропонований метод аутентифікації користувача буде описано більш детально в цьому розділі.

2.1 Переваги методу автентифікації з використанням технології NFC

Метод автентифікації за допомогою технології NFC виділяється своїми багатообіцяючими можливостями. Незважаючи на те, що стандарт NFC був представлений кілька років тому, технологія все ще розвивається. Постійно з'являються все більш цікаві, оптимізовані та ефективні рішення як в технічному напрямку, так і в напрямку розробки програмного забезпечення. Рівень безпеки при використанні методу аутентифікації користувача за допомогою технології NFC знаходиться на високому рівні при дотриманні простих правил безпеки зберігання ключів. Цей метод використовується в різних системах, які вимагають поєднання швидкості авторизації та безпеки. В даний час метод аутентифікації за допомогою технології NFC використовується в системах безпеки, системах "розумного будинку", при оплаті телефоном як карткою і в різних інших системах, пов'язаних з різними сферами життя.

2.2 Проблеми безпеки пов'язані з технологією NFC

2.2.1 Прослуховування

Оскільки NFC є бездротовим стандартом, захист від прослуховування є важливою проблемою. Коли два пристрої взаємодіють за допомогою NFC, вони використовують радіохвилі для обміну повідомленнями або даними.

Злодій може використовувати антену для прийому переданих сигналів. Для перехоплення радіочастотного сигналу злодію потрібне обладнання, а також обладнання для декодування радіочастотного сигналу. Передача даних через інтерфейс NFC зазвичай відбувається між двома пристроями, розташованими близько один до одного. Зазвичай відстань не перевищує 10 см. Основне питання полягає в тому, наскільки близько повинен знаходитися зловмисник, щоб витягти корисну інформацію з радіосигналу? Точної відповіді на це питання немає, оскільки відстань, необхідна для атаки, залежить від багатьох факторів [16], таких як:

- характеристика радіочастотного поля передавального пристрою (геометрія антени, екранування, навколишнє середовище);
- характеристики атаки антени;
- якість атакуючого приймача;
- якість атакуючого декодера;
- рівень сигналу NFC-пристрою;
- ·бар'єри на землі (стіни, метал, рівень шуму).

Тому будь-яке точне число, що відповідає потрібній зловмиснику відстані, матиме сенс лише для певного набору значень вищезгаданих характеристик.

Крім того, важливо, чи пристрій надсилає дані в активному або пасивному режимі. Це означає, що відправник генерує власне поле (активний режим) або використовує радіочастотне поле, згенероване іншим пристроєм (пасивний режим). У різних режимах використовуються різні методи передачі даних. Прослуховувати пасивні пристрої набагато складніше. Наприклад, коли пристрій надсилає дані в активному режимі, зловмисник може перебувати на відстані близько 10 метрів. Якщо пристрій перебуває в пасивному режимі, ця відстань скорочується до 1 метра.

Сама технологія NFC не застрахована від прослуховування. Слід зазначити, що дані, передані в пасивному режимі, буде набагато складніше підслухати. У той же час, передавати дані лише в пасивному режимі практично неможливо, а самого

лише пасивного режиму недостатньо для того, щоб повною мірою скористатися перевагами технології NFC.

Єдиним реальним рішенням проти прослуховування є захист каналу передачі даних за допомогою шифрування.

2.2.2 Пошкодження даних

Замість того, щоб просто прослуховувати сигнал, зловмисник може спробувати змінити дані, що передаються через інтерфейс NFC. У найпростішому випадку зловмисник може просто розірвати з'єднання, щоб приймач не зміг зрозуміти дані, надіслані іншим пристроєм. Спотворення даних може бути досягнуто шляхом надсилання фактичних частот спектру даних у правильний час. Правильний час можна розрахувати, якщо зловмисник розуміє схеми модуляції та кодування, що використовуються.

NFC-сумісні пристрої здатні протистояти цій атаці, оскільки вони можуть перевіряти радіочастотні поля на наявність шумів під час передачі даних. Якщо NFC-сумісний пристрій зробить це, він зможе виявити атаку.

Потужність, необхідна для пошкодження даних, набагато більша, ніж потужність, необхідна для виявлення NFC-пристрою. Таким чином, будь-яка така атака може бути виявлена.

2.2.3 Зміна даних

При модифікації даних зловмисник хоче не пошкодити дані, а замінити їх непомітно для користувача. Можливість такої атаки сильно залежить від сили амплітуди модуляції.

Однак, завдяки модифікованому коду Міллера [17], при кодуванні двох послідовних одиниць інформації зловмисник може змінити другу одиницю на нуль, заповнивши паузу, яка кодує другу одиницю. Декодер не побачить паузи в другому біті і декодує його як нуль, оскільки він передує одиниці. Можливість атаки значною мірою залежить від динамічного діапазону вхідного сигналу приймача.

Дуже ймовірно, що високий рівень модифікованого сигналу перевищить можливий діапазон вхідного сигналу.

Слід зазначити, однак, що на практиці дуже важко досягти такої зміни бітів під час передачі.

Захист від модифікації даних може бути досягнутий різними способами. Використовуючи швидкість передачі 106 кбіт/с в активному режимі, зловмисникові практично неможливо змінити всі дані, що передаються на радіочастоті, як описано в цьому розділі вище. Це означає, що для захисту необхідно, щоб обидва пристрої перебували в активному режимі. Хоча це можливо, але є один суттєвий недолік - активний режим є найбільш вразливим для прослуховування. Крім того, такий захист від модифікації не є ідеальним, оскільки навіть при швидкості 106 кбіт/с деякі біти можуть бути змінені [18]. Тому наступні два варіанти можуть бути кращими.

Пристрої NFC можуть перевіряти радіочастотне поле під час передачі. Це означає, що пристрій-передавач може постійно перевіряти наявність такої атаки і може зупинити передачу даних у разі виявлення атаки.

Другим і, ймовірно, найкращим рішенням було б захистити канал за допомогою хешування повідомлень. Однак передбачається, що зловмиснику вкрай важко модифікувати дані, і тому в дисертації не буде розглянуто реалізацію захисту від модифікації даних.

2.2.4 Людина посередині

Визначення того, чи потрібна автентифікація пристрою "точка-точка". Чи можна прослухати сигнал, чи можна змінити сигнал за певних обставин, чи можна атакувати людину в середині і чи потрібен захист від цієї атаки.

У класичній схемі атаки "Людина посередині" є дві групи, які хочуть спілкуватися одна з одною - це Аліса і Боб, і третя сторона - зловмисник - Єва. На схемі це показано на Рисунку 2.1.

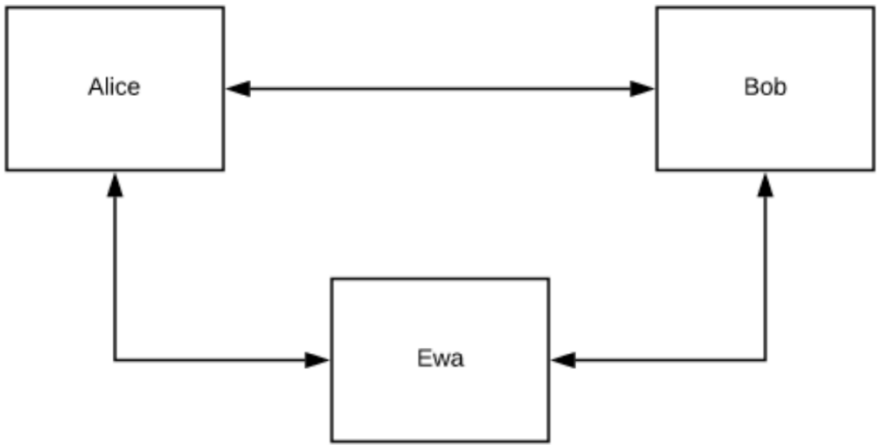


Рисунок 2.1 – Шаблон атаки "Людина посередині"

Аліса і Боб не повинні розуміти, що вони не розмовляють один з одним, а приймають і передають дані третій стороні. Це класична небезпека аутентифікації в таких протоколах, як протокол Діффі-Хеллмана: Аліса і Боб хочуть домовитися про секретний ключ, який вони будуть використовувати для безпечного каналу. Але Єва, перебуваючи посередині, може встановити свій власний ключ для Аліси і ще один ключ для Боба. І коли Аліса і Боб пізніше використовують свої ключі для захисту даних, Єва може легко прослуховувати дані, що передаються, а також змінювати їх. Для того, щоб така атака могла бути реалізована на практиці з використанням, наприклад, технології NFC, пристрій-перехоплювач повинен знаходитися між пристроєм, що зчитує сигнал, і пристроєм, що приймає сигнал.

Як наслідок, атака "людина посередині" практично неможлива в контексті NFC. Але для більшої впевненості рекомендується використовувати активно-пасивний режим зв'язку, щоб радіочастотне поле постійно генерувалося одним з реальних учасників.

Крім того, активні учасники повинні прослуховувати радіочастотне поле, яке створюється під час надсилання даних, щоб виявити будь-яке вторгнення в це поле, спричинене зловмисником.

2.3 Удосконалений метод автентифікації

Новизна методу автентифікації користувачів на основі NFC полягає не в самій технології, а в її застосуванні. Вже сьогодні технологія NFC використовується на багатьох підприємствах як додатковий захід безпеки та для контролю присутності співробітників. Однак для ідентифікації співробітників лише деякі компанії використовують телефон, який завжди знаходиться поруч з власником, а не NFC-мітку або картку. Реалізація автентифікації користувачів за допомогою технології NFC відкриває додаткові можливості як для розробників, так і для користувачів методу автентифікації. Яскравим прикладом є реалізація, коли співробітники не тільки реєструються в компанії, але й бачать власну присутність завдяки простоті поєднання функціональності сучасних технологій.

Опис методу автентифікації. Метод автентифікації користувача за допомогою технології NFC складається лише з одного компонента Детальніші міркування можна побачити на рисунку 2.2.

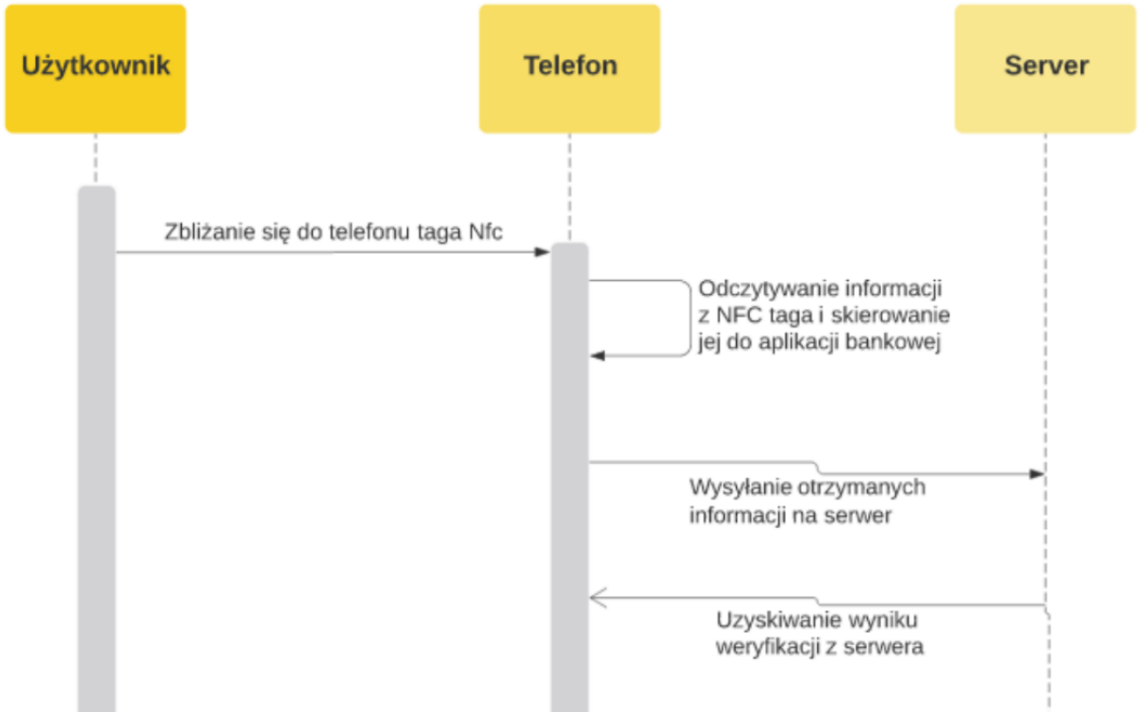


Рисунок 2.2 – Логіка отримання даних з NFC-мітки

Цей компонент є зображенням, яке інформує користувача про те, що наразі використовується саме цей метод автентифікації користувача. Як зображення вибирається іконка з логотипом, що позначає NFC. Вся логіка методу автентифікації користувача за допомогою технології NFC міститься в коді, і все, що потрібно від користувача - це піднести до телефону приймаючу мітку. Логіка полягає в тому, що після успішної перевірки входу в систему пристрій зчитує дані з пред'явленої мітки і відправляє їх в додаток. Коли додаток працює, дані перехоплюються і надсилаються на сервер.

3 ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ АВТЕНТИФІКАЦІЇ

3.1 Метод дослідження

Для цієї дипломної роботи було розроблено додаток, в якому були реалізовані всі методи аутентифікації з попереднього розділу. Під час тестування програма виводить тестові дані у вікно налагодження програми.

Деякі з представлених методів є поширеними і знайомими багатьом користувачам Android-пристроїв. Деякі методи представлені через їх потенційно високий рівень безпеки. Рівень безпеки буде визначено в подальшому при аналізі результатів дослідження методів автентифікації користувачів.

Дослідження було проведено з метою визначення зручності, переваг та рівня безпеки методів автентифікації користувачів. Дослідження включає як статистичні дані, отримані в результаті опитування користувачів, так і аналітичні дані, отримані під час користувацького тестування додатку.

3.1.1 Структура проекту для дослідження

Додаток, що використовується для проведення дослідження, складається з серверної частини та мобільного додатку для пристроїв на платформі Android. Серверна частина проекту виконує функцію зберігання даних користувача в базі даних і забезпечує доступ до цих даних шляхом виконання запиту. Серверна частина розміщується на віртуальному сервері, який повинен бути запущений на комп'ютері для повноцінного функціонування проекту. Серверна частина також може бути розміщена і доступна 24/7.

Мобільний додаток надсилає запити до сервера та відображає отримані з сервера дані у зручному для користувача вигляді. Мобільний додаток працює тільки на пристроях на платформі Android і може бути запущений на декількох пристроях одночасно.

Серверна частина була написана на PHP. Також було використано середовище Laravel, засноване на мові PHP. Основна роль фреймворку Laravel -

забезпечення стабільного з'єднання з базою даних та коректного виконання запитів. База даних проекту зберігається на локальному сервері.

База даних генерується відповідно до параметрів, описаних у створених класах, які будуть представлені далі. Існує основна схема бази даних та допоміжна схема. Основна схема бази даних складається з таблиць, які використовуються безпосередньо для отримання даних користувача. Деталі можна побачити на рисунку 3.1.

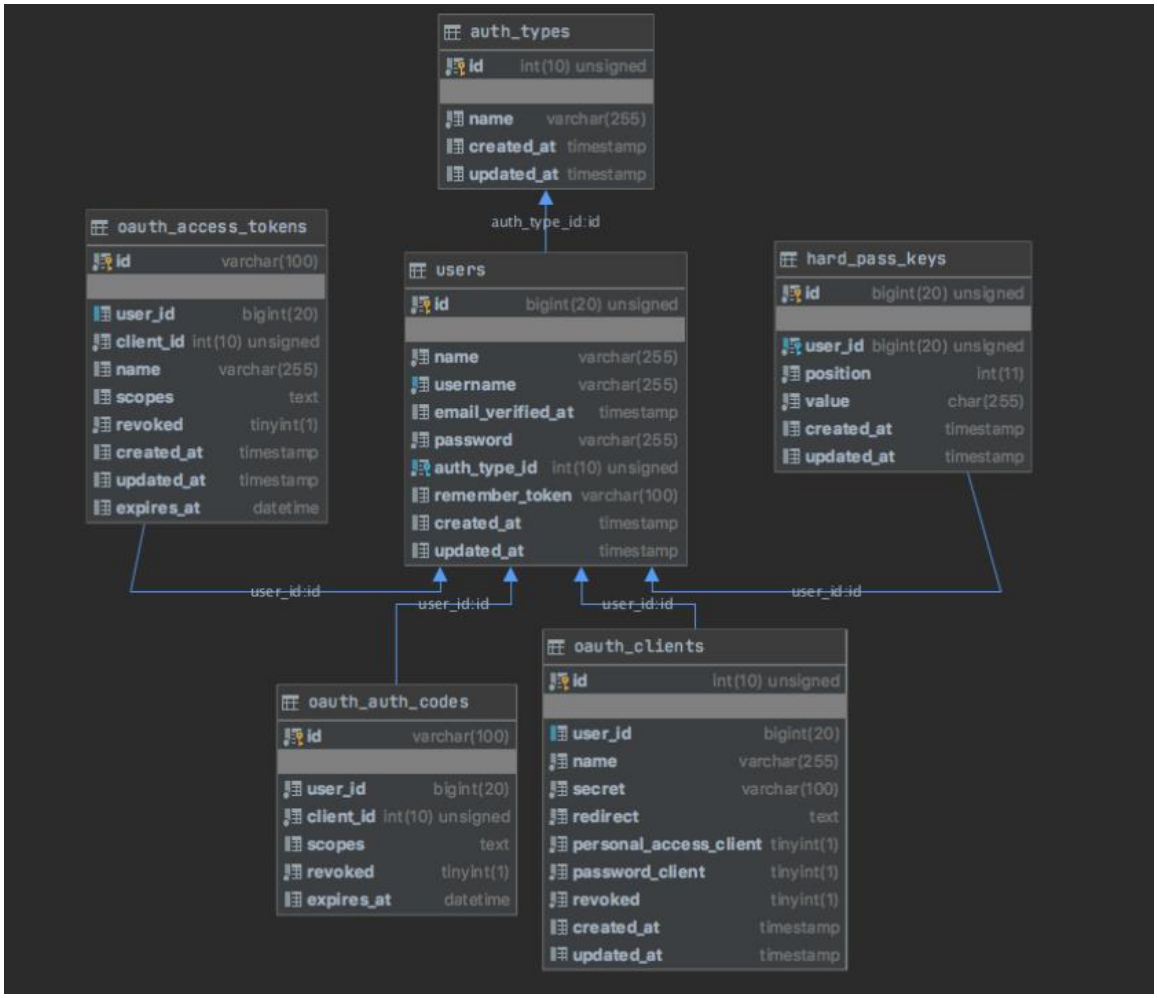


Рисунок 3.1 – Базова схема бази даних

Схема допоміжної бази даних зберігає історію змін, внесених під час створення та модифікації основної бази даних. Більше можна побачити на рисунку 3.2.

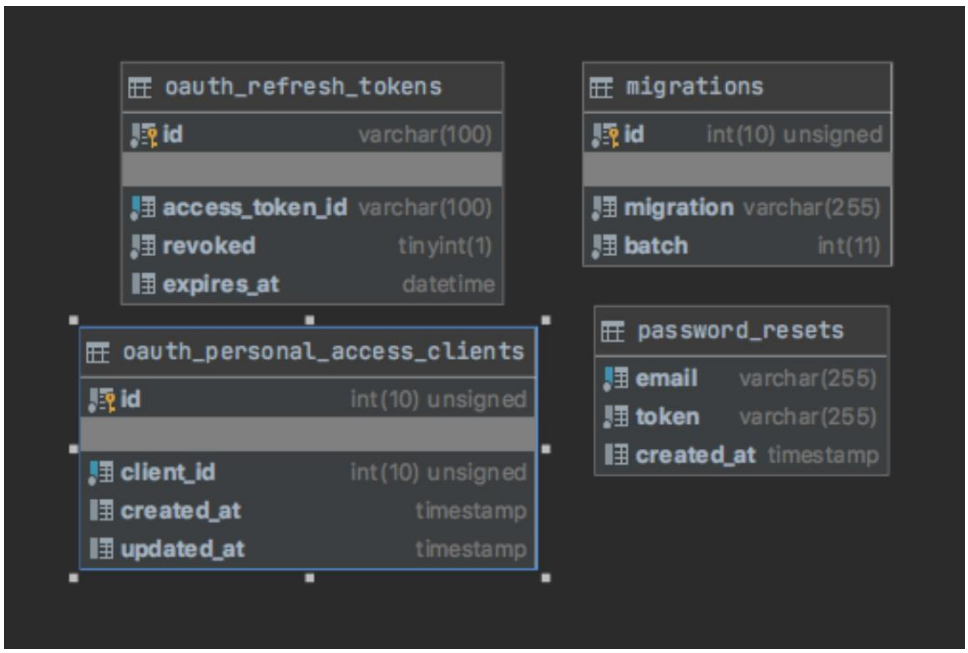


Рисунок 3.2 – Схема допоміжної бази даних

Фреймворк Laravel дозволяє створювати складні схеми таблиць бази даних не стандартним способом (не за допомогою SQL-запитів), а шляхом написання коду в класах, успадкованих від класу Migrations, за допомогою спеціального методу класу Schema::create. Більш детально про це можна дізнатися з рисунку 3.3.

```
Schema::create( table: 'hard_pass_keys', function (Blueprint $table) {
    $table->bigIncrements( column: 'id');
    $table->bigInteger( column: 'user_id')->unsigned()->index();
    $table->foreign( columns: 'user_id')->references( columns: 'id')->
            on( table: 'users')->onDelete( action: 'cascade');
    $table->integer( column: 'position');
    $table->char( column: 'value');
    $table->timestamps();
});
```

Рисунок 3.3 – Створення таблиці бази даних з допомогою методу класу Schema::create

Кожна сутність схеми бази даних зберігається в окремому класі в каталозі migrations. Каталог заводів містить класи для генерації даних для таблиць бази даних. Більше можна побачити на рисунку 3.4.

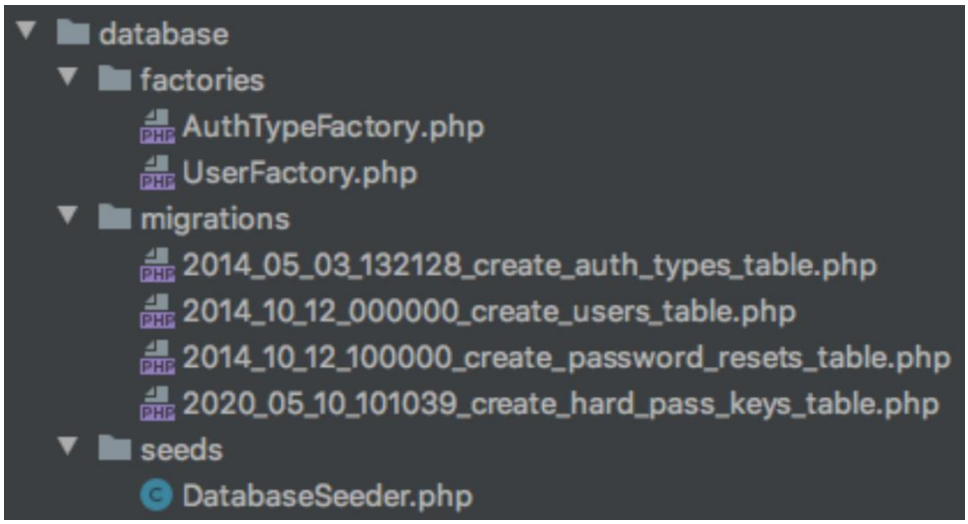


Рисунок 3.4 – Зміст довідників заводів та міграцій

Створення даних для таблиць бази даних так само просте, як і створення самих таблиць. Більше інформації наведено у рисунку 3.5.

```
$factory->define( class: User::class, function (Faker $faker) {

    static $password;

    return [
        'name' => $faker->name,
        'username' => $faker->unique()->safeEmail,
        'email_verified_at' => now(),
        'password' => $password ?: $password = bcrypt( value: 'secret'), // password
        'remember_token' => Str::random( length: 10),
        'auth_type_id' => function () {
            return \App\AuthType::all()->random();
        },
    ];
```

Рисунок 3.5 – Приклад методу генерації даних для таблиць бази даних

Для доступу до даних використовуються посилання. Посилання визначаються за допомогою шляхів, вказаних у коді в спеціальному файлі api.php. Нижче наведено кілька прикладів того, як виглядає адреса призначення посилання. Більше можна побачити у рисунку 3.6.

Рисунок 3.6 – Шляхи до методів контролера

Нехай кожне таке посилання посилається на певний фрагмент коду, який міститься в класах контролерів. Класи контролерів об'єднують декілька методів для більш зручного та читабельного коду. Нижче ви можете побачити код на прикладі шляху реєстрації. Деталі можна побачити у рисунку 3.7.

```php
public function register(Request $request)
{
    $request->validate([
        'name' => 'required',
        'username' => 'required|unique:users',
        'password' => 'required|min:6',
        'auth_type_id'=>'required'
    ]);

    $user = User::create([
        'name' => $request->name,
        'username' => $request->username,
        'password' => bcrypt($request->password),
        'auth_type_id'=>$request->auth_type_id
    ]);

    if (AuthType::idIsHardPass($user->auth_type_id)) {
        $i = 0;
        foreach (str_split($request->password) as $symbol) {
            $i++;
            HardPassKey::create([
                'user_id' => $user->id,
                'position' => $i,
                'value' => $symbol
            ]);
        }
    }
    return response()->json([
        'result'=>'ok'
    ]);
}
```

Рисунок 3.7 – Приклад вигляду вмісту методу

У кожному методі з вхідними даними поля попередньо перевіряються на відповідність заданим параметрам. Якщо всі надіслані дані відповідають заданим параметрам, виконується примусове створення об'єкта, який розміщується в таблиці бази даних відповідно до заданих параметрів. Об'єктом може бути один рядок з таблиці бази даних або декілька рядків (колекція). Після створення об'єкта

із заданими параметрами об'єкт обробляється і результат його обробки надсилається клієнту у вигляді JSON-рядка.

У проекті реалізовано методи для авторизації, реєстрації та отримання інформації про конкретні параметри, що містяться у декількох класах. Більше можна побачити на рисунку 3.8.

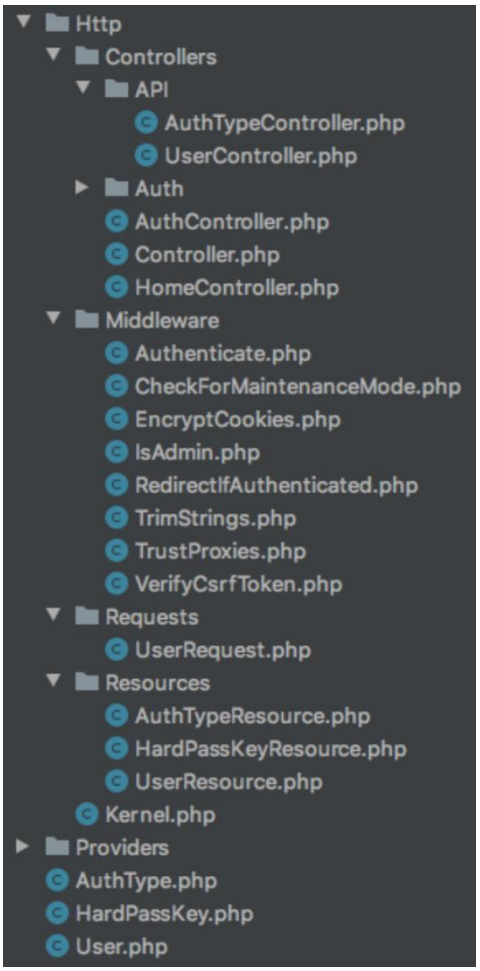


Рисунок 3.8 – Знімок екрана з класами проектів

Незважаючи на простоту використання шляхів запитів, на серверній стороні проекту передбачена валідація, щоб уникнути помилок при виконанні запитів на рівні бази даних. Також реалізована безпечна реєстрація, при якій пароль зашифрований в таблицях, і навіть при доступі до таблиць бази даних доступ до паролю все одно обмежений.

Для успішної реєстрації користувача достатньо ввести ім'я, логін, пароль та ідентифікатор обраного методу авторизації. Якщо поля введені правильно і не

суперечать умовам перевірки полів, реєстрація буде успішною. Поля grant_type, client_id та client_secret є унікальними для проекту і правильні дані можна отримати з допоміжних таблиць бази даних. Деталі можна побачити на рисунку 3.9.

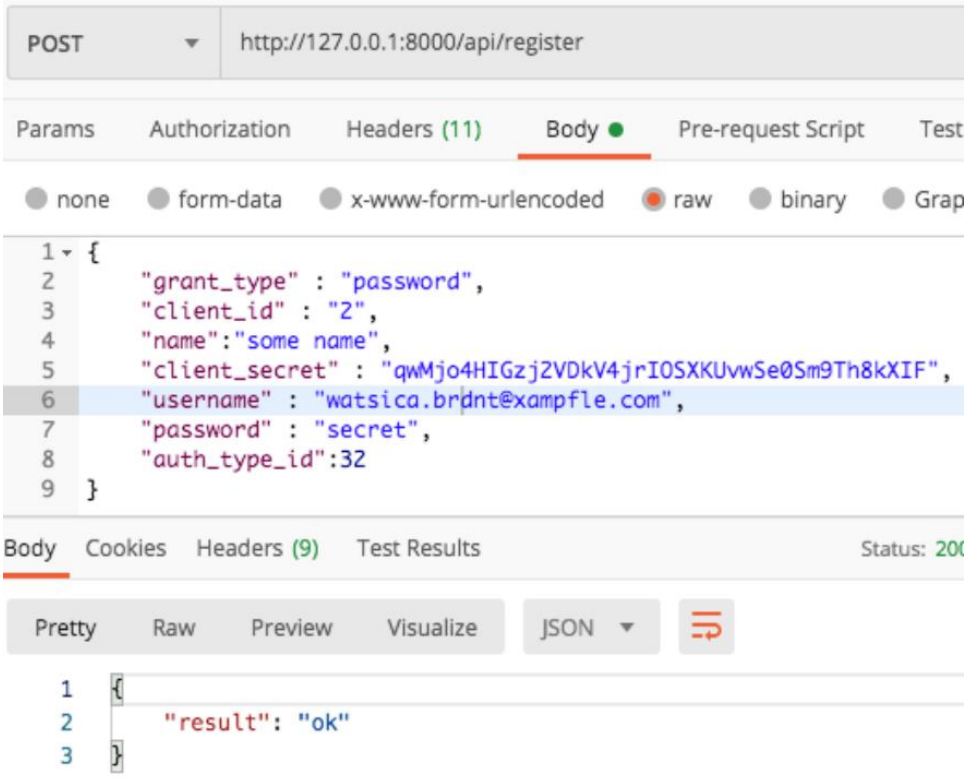


Рисунок 3.9 – Створення нового користувача

Якщо ви введете неправильні дані або спробуєте використати ім'я користувача, яке вже використовується, з'явиться відповідна помилка з кодом відповіді сервера 400. Більше інформації наведено на рисунку 3.10.

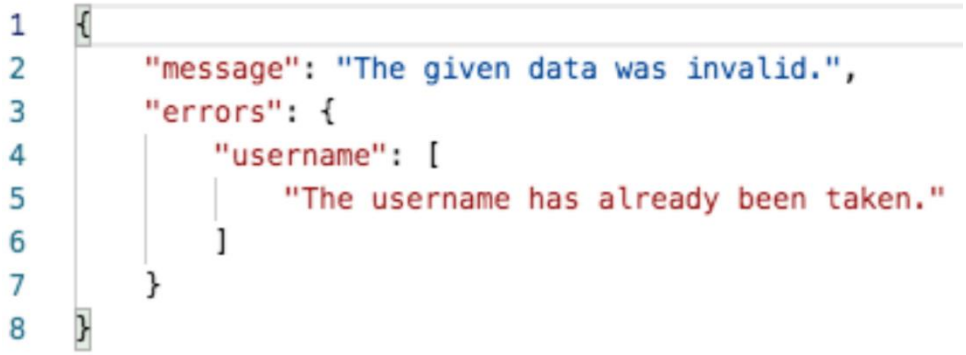


Рисунок 3.10 – Відповідь сервера на невірні дані для створення нового користувача

Для входу за допомогою простого пароля, шаблону та методу з використанням технології NFC, просто використовуйте своє ім'я користувача та пароль, щоб задати питання серверу. Використовуються дані для входу, обрані при реєстрації. Поля "grant_type", "client_id", "client_secret" використовуються для отримання доступу до функції Laravel. Додаткова автентифікація в Laravel забезпечує більшу безпеку. Поля "ім'я користувача" і "пароль" використовуються для входу в систему. Деталі можна побачити на рисунку 3.11.

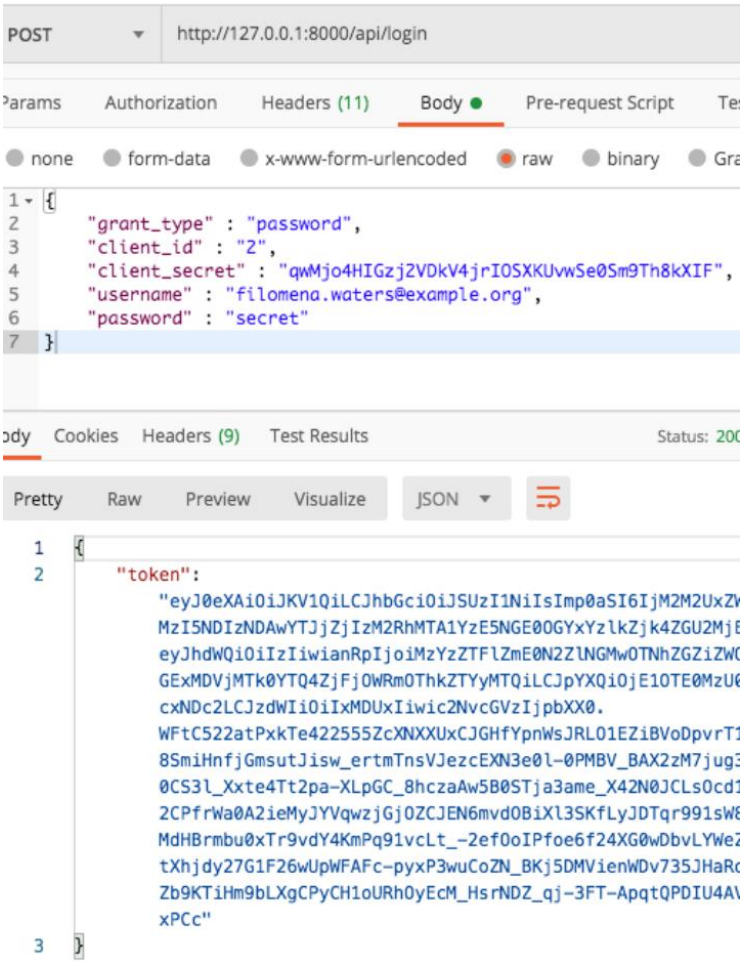


Рисунок 3.11 – Приклад автентифікації користувача за допомогою класичного пароля, методу шаблону та авторизація користувача за допомогою технології NFC

У відповідь користувач отримує токен доступу, який може бути використаний для доступу до решти запитів проекту в межах встановлених прав облікового запису.

Для входу за допомогою методу прихованого пароля необхідно надіслати на сервер логін користувача та масив об'єктів, кожен елемент якого вказує на позицію літери в паролі та її значення. Більше можна побачити на рисунку 3.12.

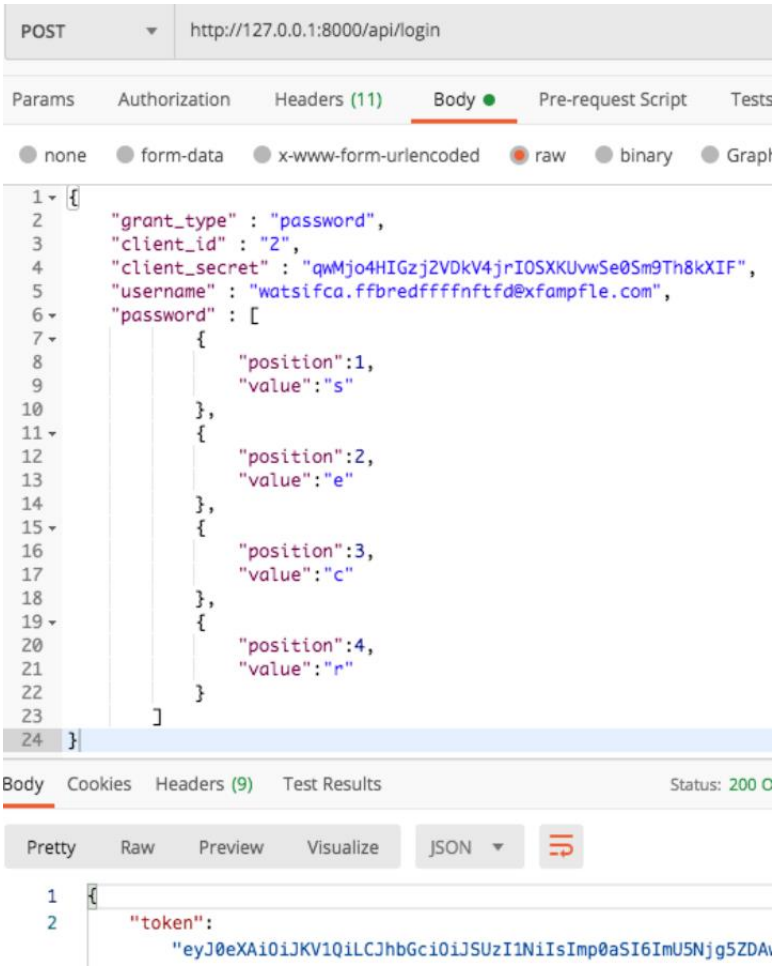


Рисунок 3.12 – Метод автентифікації за допомогою прихованого пароля

Клієнтська частина програми складається з додатку на базі мобільної платформи Android.

Проект складається з двох основних папок java та src. Тека java містить класи та інтерфейси, розбиті на пакети і написані на мові java.

Тека src містить зображення та файли у форматі XML. Файли, записані у форматі XML, можуть містити константи різних типів даних: від чисел, які використовуються для визначення розмірів елементів екрана, до рядків, які можуть одночасно зберігати переклади для кількох мов. Ієрархію файлів показано на рисункц 3.13.

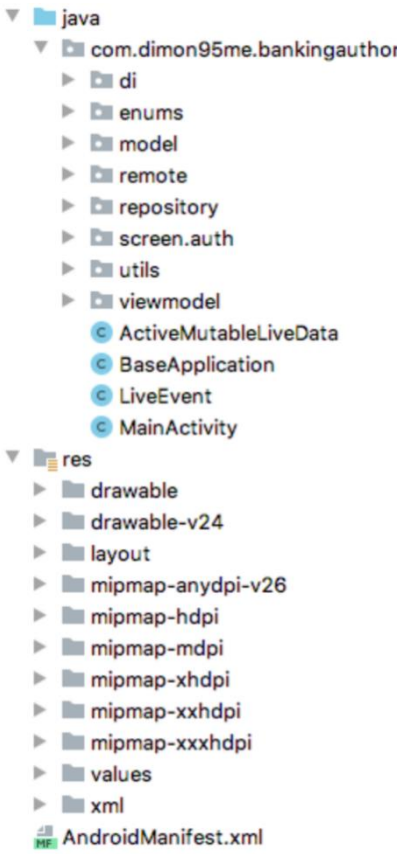


Рисунок 3.13 – Ієрархія файлів проекту

Тека src / layout містить файли, написані у форматі XML, які містять елементи екрану (рис. 3.14). Елементи екрану знаходяться у фіксованому положенні відповідно до написаного коду. Нижче наведено приклад файлу, в якому одночасно відкрито відображення елементів екрану та код файлу.

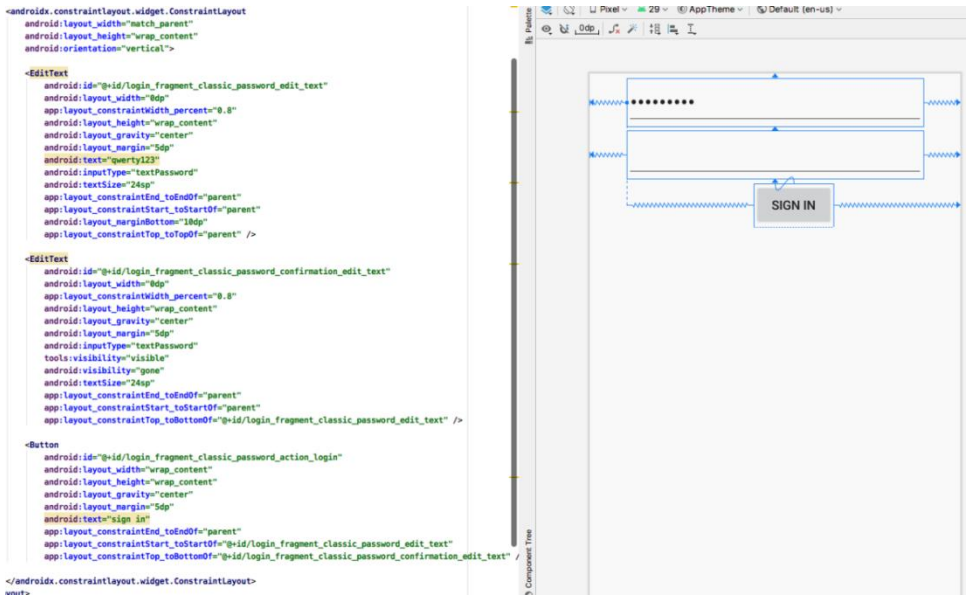


Рисунок 3.14 – Перегляд вмісту XML-файлу

Для того, щоб встановити з'єднання з сервером, додаток повинен бути підключений до Інтернету. За підключення додатку до сервера відповідає бібліотека Retrofit. Надана бібліотека дозволяє додатку обробляти отримані JSON-запити та надсилати запити до сервера у спрощеному вигляді. Код наведено у рисунку 3.15.

```
@Provides
@Singleton
static Retrofit provideRetrofit(Gson gson) {
    return new Retrofit.Builder()
            .baseUrl("https://55d2486d3cb5.ngrok.io/")
            .addConverterFactory(GsonConverterFactory.create(gson))
            .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
            .build();
}
```

Рисунок 3.15 – Метод виклику Retrofit

Викликавши об'єкт, отриманий за допомогою представленої функції, показаної у рисунку 3.15, можна скористатися функціональністю бібліотеки Retrofit.

Кожен з методів авторизації користувача має свої особливості, тому для кожного з методів авторизації створено окремий клас, в якому відбувається звернення до сервера. Передача пароля для перевірки на сервер відбувається двома способами, залежно від обраного методу авторизації користувача. Методи авторизації за допомогою класичного паролю, графічного ключа, методу з використанням технології NFC надсилають на сервер рядок, викликаючи метод, показаний у рисунку 3.16.

```
@Override
public void authorize(String password) {
    authViewModel.authorize(binding.etUsername.getText().toString(), password);
}
```

Рисунок 3.16 – Метод авторизації через рядок

Метод автентифікації за допомогою прихованого пароля використовує окремий метод для надсилання пароля на сервер, оскільки він надсилає масив символів. Його реалізацію показано у рисунку 3.17.

```
@Override
public void authorize(List<HardPassKey> hardPassKeys) {;
    authViewModel.authorize(binding.etUsername.getText().toString(), hardPassKeys);
}
```

Рисунок 3.17 – Метод аутентифікації через масив символів

Об'єкт authViewModel є посередником між елементами інтерфейсу користувача програми та бібліотекою Retrofit і використовує перевантажені методи для авторизації. Більш детальну інформацію можна знайти у рисунку 3.18.

```
public void authorize(String username, String password) {
    authRepository.authorize(responseLoginMutableLiveData, username, password);
}

public void authorize(String username, List<HardPassKey> hardPassKeys) {
    authRepository.authorize(responseLoginMutableLiveData, username, hardPassKeys);
}
```

Рисунок 3.18 – Методи автентифікації у authViewModel

Об'єкт authRepository містить методи send та receive при взаємодії з сервером. Приклад відправки паролю на сервер у вигляді рядка показано у рисунку 3.19.

```
public void authorize(MutableLiveData<ResponseLogin> responseLoginMutableLiveData, String username, String password) {
    disposable.add(
            authService.login(CreateJsonParamUtils.createLogin(username, password))
                    .subscribeOn(Schedulers.io())
                    .observeOn(AndroidSchedulers.mainThread())
                    .subscribeWith(
                            new DisposableSingleObserver<ResponseLogin>() {
                                @Override
                                public void onSuccess(ResponseLogin responseLogin) {
                                    responseLoginMutableLiveData.postValue(responseLogin);
                                }

                                @Override
                                public void onError(Throwable e) {
                                    if (errorCodeMutableLiveData != null)
                                        errorCodeMutableLiveData.postValue(ErrorCode.INVALID_USERNAME_OR_PASSWORD);
                                }
                            }
                    )
    );
}
```

Рисунок 3.19 – Метод зв'язку з сервером

Об'єкт responseLoginMutableLiveData знаходиться в authViewModel і успадковується від спеціального класу MutableLiveData, який має схожі властивості з паттерном Observer. При отриманні відповіді від сервера об'єкт відправляється за допомогою спеціального методу postValue(). Клас, який підписаний на responseLoginMutableLiveData, отримує повідомлення про отримання нових даних і викликає раніше визначений метод. Більш детальну інформацію можна знайти у рисунку 3.20.

```
authViewModel.getResponseLoginMutableLiveData().observe( owner: this, responseLogin ->
{
    Toast.makeText(getContext(), text: "You are logged in", Toast.LENGTH_SHORT).show();
});
```

Рисунок 3.20 – Реакція підписаного класу на нові дані

Об'єкт authService містить методи для зв'язку з сервером, які визначають шлях для виконання запиту та вміст запиту, що надсилається на сервер. Приклади наведено у рисунку 3.21.

```
@POST("api/register")
Single<ResponseRegister> register(@Body JsonObject object);

@GET("api/auth_types")
Single<List<AuthType>> getAllAuthTypes();

@POST("api/login")
Single<ResponseLogin> login(@Body JsonObject object);
```

Рисунок 3.21 – Методи зв'язку з сервером через шлях доступу

Статичний клас CreateJsonParamUtils містить методи, які перетворюють вказані дані у JSON-рядок для завантаження на сервер. Приклад одного з таких методів наведено у рисунок 3.22.

```
public static JsonObject createLogin(String username, List<HardPassKey> hardPassKeys) {
    JsonObject param = new JsonObject();

    param.addProperty(GRANT_TYPE, PASSWORD);
    param.addProperty(CLIENT_ID, CLIENT_ID_VALUE);
    param.addProperty(CLIENT_SECRET, CLIENT_SECRET_VALUE);

    param.addProperty(USERNAME, username);
    JsonArray hardPassword = new JsonArray();
    for (HardPassKey hardPassKey : hardPassKeys) {
        JsonObject key = new JsonObject();
        key.addProperty(POSITION, hardPassKey.getPosition());
        key.addProperty(VALUE, hardPassKey.getValue());
        hardPassword.add(key);
    }
    param.add(PASSWORD, hardPassword);


    return param;
}
```

Рисунок 3.22 – Метод перетворення даних у JSON-рядок

Використання такої схеми завантаження на сервер реалізовано для того, щоб дизайн можна було масштабувати, а запити надсилати в окремому потоці. Якщо виконувати все в потоці UI, то можна помітити, що додаток працює повільно, оскільки основний потік, який відповідає за відображення елементів на екрані, перевантажений. Щоб вирішити проблему використання потоку UI, виконуйте операції доступу до сервера в окремих потоках.

Методи, що викликаються під час аутентифікації, також використовуються для реєстрації, оскільки вони виконують ту ж саму задачу - передають введений пароль на сервер.

При підтвердженні введеного класичного пароля викликається метод, показаний у рисунку 3.23.

```
public String getPassword() {
    String response = null;
    if (binding.loginFragmentClassicPasswordEditText.getText().toString().length() < 8) {
        Toast.makeText(getContext(),  text: "Password is too short", Toast.LENGTH_SHORT).show();
    } else if (!binding.loginFragmentClassicPasswordEditText.getText().toString()
            .equals(binding.loginFragmentClassicPasswordConfirmationEditText.getText().toString())) {
        Toast.makeText(getContext(),  text: "Password fields are not the same", Toast.LENGTH_SHORT).show();
    } else
        response = binding.loginFragmentClassicPasswordEditText.getText().toString();
    return response;
}
```

Рисунок 3.23 – Метод отримання класичного пароля

Коли користувач забирає палець з екрана під час введення пароля методом автентифікації за зразком, викликається метод, який перевіряє введений пароль на

відповідність і, у разі успіху, повертає закодований рядок на сервер. Докладнішу інформацію наведено у рисунку 3.24.

```
@Override
public void onComplete(List<PatternLockView.Dot> pattern) {
    if (pattern.size() > 5) {
        String supposedPassword = (ShaUtil.sha256(pattern.toString())).substring(10, 18);
        pattern.clear();
        switch (mode) {
            case LOGIN:
                    actionsListener.authorize(supposedPassword);
                break;
            case REGISTRATION:
                if (firstEntry == null || firstEntry.isEmpty()||(firstEntry!=null
                        &&secondEntry!=null&&firstEntry.equals(secondEntry))) {
                    firstEntry = supposedPassword;
                    Toast.makeText(getContext(), text: "Re-enter password for verify", Toast.LENGTH_SHORT).show();
                } else if (firstEntry.equals(supposedPassword)) {
                    secondEntry = supposedPassword;
                    Toast.makeText(getContext(), text: "Password accepted", Toast.LENGTH_SHORT).show();
                } else {
                    firstEntry = secondEntry = null;
                    Toast.makeText(getContext(), text: "Passwords are not equals", Toast.LENGTH_SHORT).show();
                    Toast.makeText(getContext(), text: "Enter password two times again", Toast.LENGTH_SHORT).show();
                }
                break;
        }
        Log.d(TAG, msg: "onComplete: " + (ShaUtil.sha256(pattern.toString())).substring(10, 18));
    } else {
        pattern.clear();
        Toast.makeText(getContext(), text: "You are need more than 5 dots", Toast.LENGTH_SHORT).show();
    }
}
```

Рисунок 3.24 – Метод отримання класичного пароля

Коли користувач прикріплює NFC-мітку, викликається метод, який автоматично надсилає пароль для перевірки. Метод показано у рисунку 3.25.

```
public void onNfcDetected(Ndef ndef, Intent intent) {
    switch (mode) {
        case LOGIN:
            actionsListener.authorize(readFromNFC(ndef, intent).substring(1));
            break;
        case REGISTRATION:
            write(createTextMessage(), intent);
            break;
    }
}
```

Рисунок 3.25 – Метод взаємодії мітки NFC

У випадку реєстрації на сервер викликається метод write, через який надсилається рядок з випадковим набором символів. Описаний метод наведено у рисунку 3.26.

```
private String createNfcPassword() {
    Random generator = new Random();
    StringBuilder passwordStringBuilder = new StringBuilder();
    int randomLength = 8 + generator.nextInt( bound: 25);
    char randomChar;
    for (int i = 0; i < randomLength; i++) {
        randomChar = (char) (generator.nextInt( bound: 96) + 32);
        passwordStringBuilder.append(randomChar);
    }
    return passwordStringBuilder.toString();
}
```

Рисунок 3.26 – Метод генерації пароля для NFC-мітки

Після підтвердження замаскованого пароля викликається метод, який повертає масив символів з певним порядковим номером, який вказує на їх позицію у паролі. Докладнішу інформацію наведено у рисунку 3.27.

```
private void onLoginClick(View view) {
    List<HardPassKey> hardPassKeys = new ArrayList<>();
    for (EditText editText : editTexts) {
        if (editText.getText().toString().isEmpty()) {
            Toast.makeText(getContext(), text: "Fill all fields first", Toast.LENGTH_SHORT).show();
            return;
        }
    }
    for (int i = 0; i < editTexts.size(); i++) {
        if (editTexts.get(i).isEnabled()) {
            hardPassKeys.add(new HardPassKey( position: i + 1, editTexts.get(i).getText().toString().charAt(0)));
        }
    }
    actionsListener.authorize(hardPassKeys);
}
```

Рисунку 3.27 – Метод отримання замаскованого пароля

Програма виконує реєстрацію нових користувачів та їхню авторизацію на сервері. Коли відкривається вікно реєстрації, перше, що завантажує сервер, це список методів авторизації. Деталі можна побачити на рисунку 3.28.

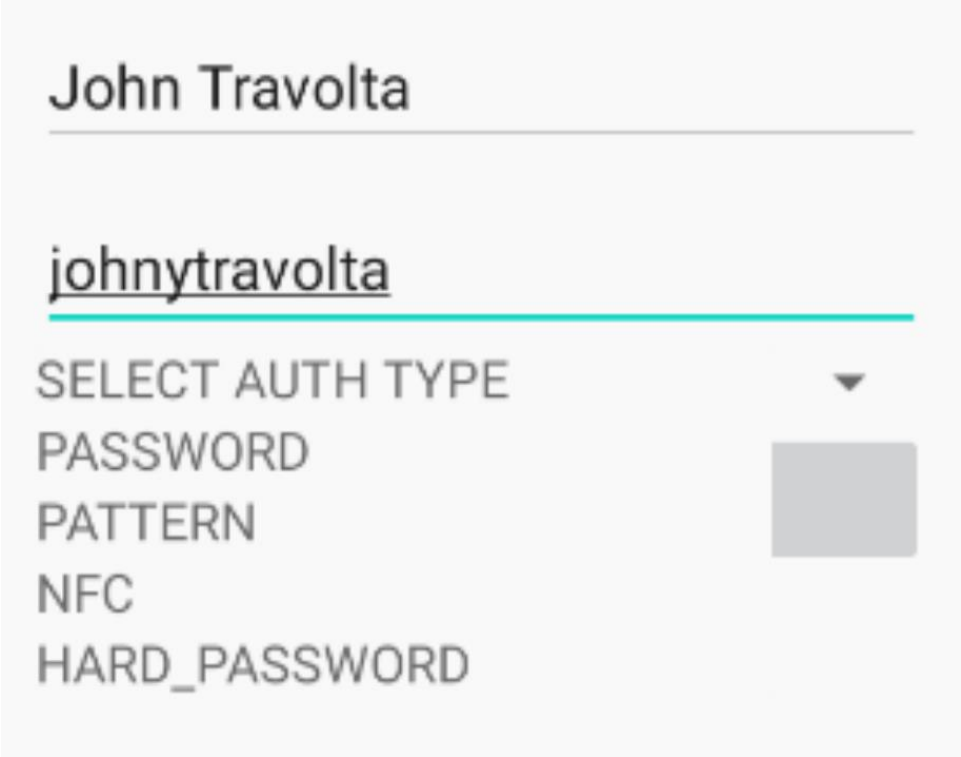


Рисунок 3.28 – Перелік методів авторизації

Після вибору методу авторизації з'явиться відповідний розділ обраного методу реєстрації, в якому можна буде ввести пароль.

При реєстрації користувача з використанням класичного пароля і пароля з маскою використовується однакова ділянка екрана. Це пов'язано з тим, що пароль вводиться з клавіатури під час автентифікації. Більше можна побачити на рисунку 3.29.

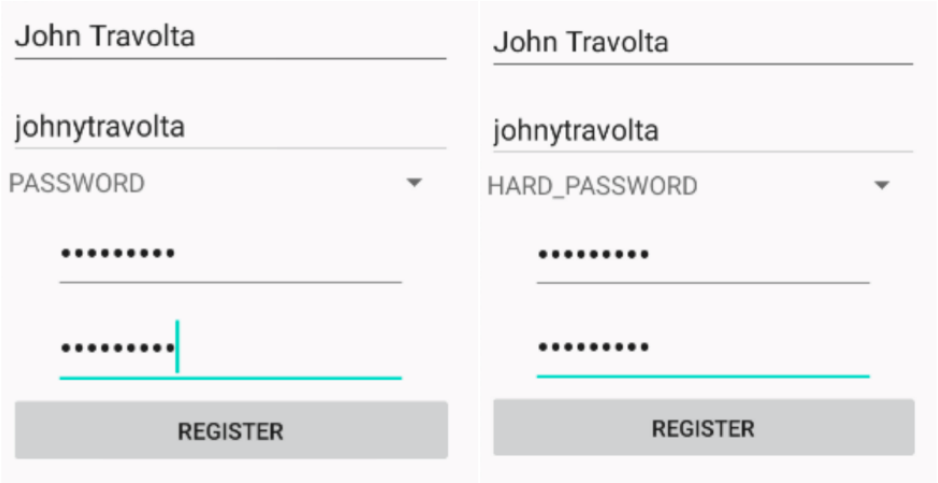


Рисунок 3.29 – Реєстрація користувача зі звичайним та маскованим паролем

При реєстрації користувача за шаблоном відображається розділ для введення пароля. Після одноразового введення пароля програма попросить вас повторно ввести пароль, щоб уникнути реєстрації користувача з невідомим паролем. Якщо з двох спроб було введено два різних паролі, програма попросить вас ввести той самий пароль двічі. Деталі можна побачити на рисунку 3.30.

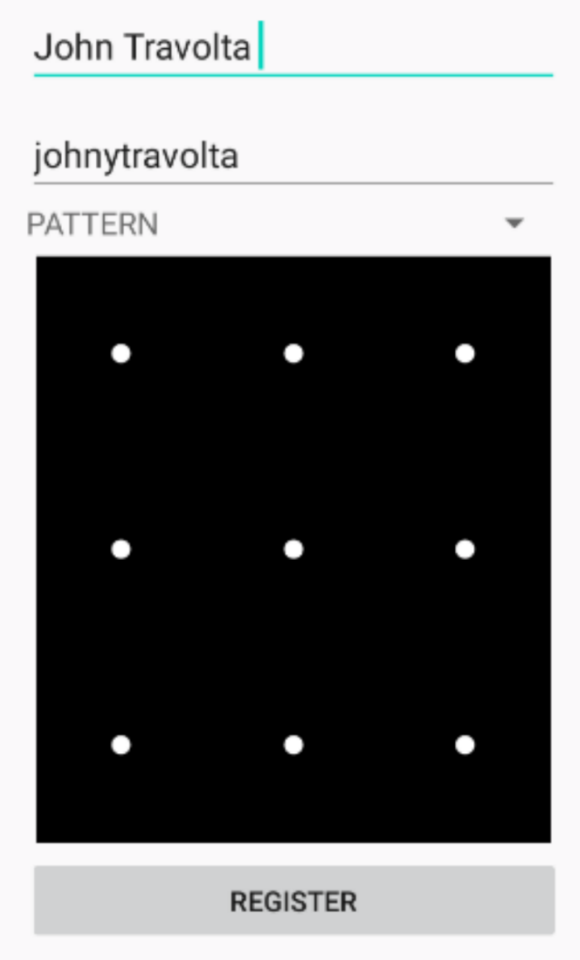
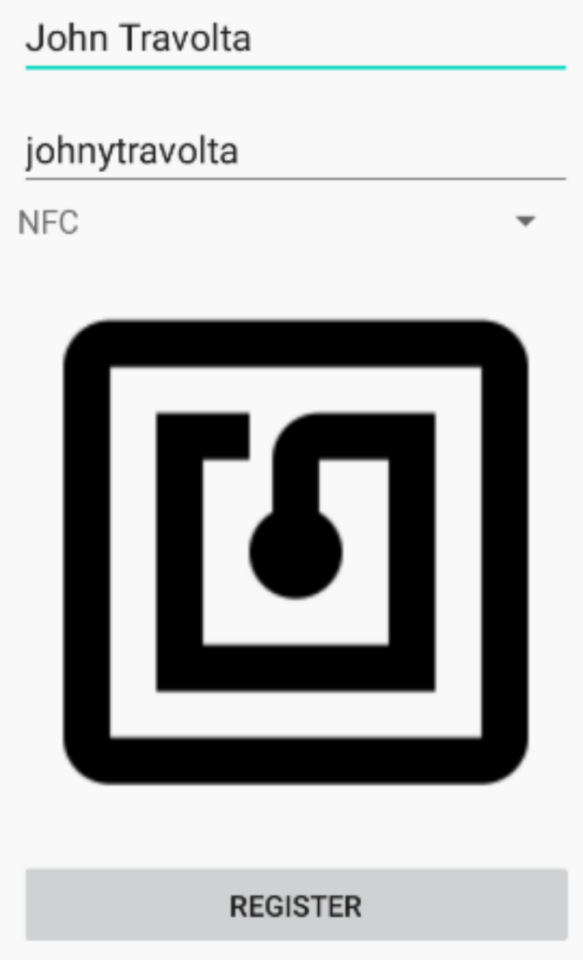


Рисунок 3.30 – Реєстрація користувача за допомогою шаблону

При виборі методу реєстрації за допомогою технології NFC користувачеві відображається фрагмент, який підключається до вбудованої технології зчитування NFC-міток. Після того, як NFC-мітка прикладена до телефону, додаток видаляє попередню інформацію з мітки і записує на її місце пароль. Користувач отримує відповідне повідомлення про те, що новий пароль був успішно записаний на NFC-мітку, і може продовжити реєстрацію. Більше можна побачити на рисунку 3.31.



c

Рисунок 3.31 – Реєстрація користувача за допомогою технології NFC

Якщо користувач введе неправильні дані, існуючий в базі даних логін або не введе пароль відповідно до правил введення пароля для кожного з нових методів реєстрації користувачів, він отримає повідомлення про те, що не всі дані були заповнені правильно.

Після успішної реєстрації додаток переходить до вікна авторизації. Після введення логіну на сервер надсилається запит на перевірку, чи існує користувач з таким логіном. Якщо користувача з таким логіном не існує, з'являється відповідне повідомлення з пропозицією перевірити логін на правильність написання. Якщо сервер стверджує, що користувач з таким логіном існує, повертається відповідь від сервера, яка містить інформацію про метод аутентифікації користувача. У випадку прихованого пароля також повертається інформація про те, скільки символів містить пароль. Деталі можна побачити на рисунку 3.32.

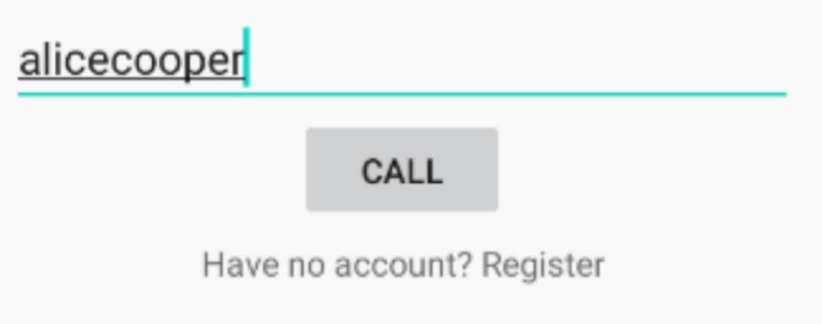


Рисунок 3.32 – Форма для перевірки наявності користувача за заданим логіном

При автентифікації за допомогою класичного методу пароля користувачеві буде запропоновано ввести пароль для перевірки відповідності на сервері. Більше можна побачити на рисунку 3.33.

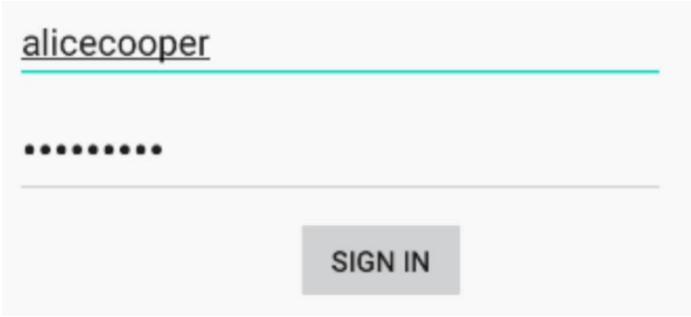


Рисунок 3.33 – Класичний метод автентифікації за допомогою пароля

При авторизації за допомогою методу пароля користувачеві пропонується ввести той самий пароль, що і при реєстрації. Після введення пароля інформація з введеного шаблону автоматично відправляється на сервер для перевірки пароля на збіг. Деталі можна побачити на рисунку 3.34.

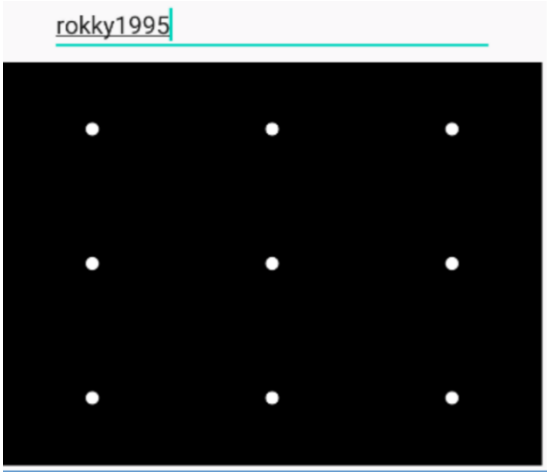


Рисунок 3.34 – Метод автентифікації за формулою

Під час автентифікації за допомогою методу NFC користувачеві пропонується прикласти NFC-мітку з попередньо збереженим паролем. Після зчитування пароля з NFC-мітки пароль автоматично надсилається на сервер для перевірки сумісності. Більше можна побачити на рисунку 3.35.

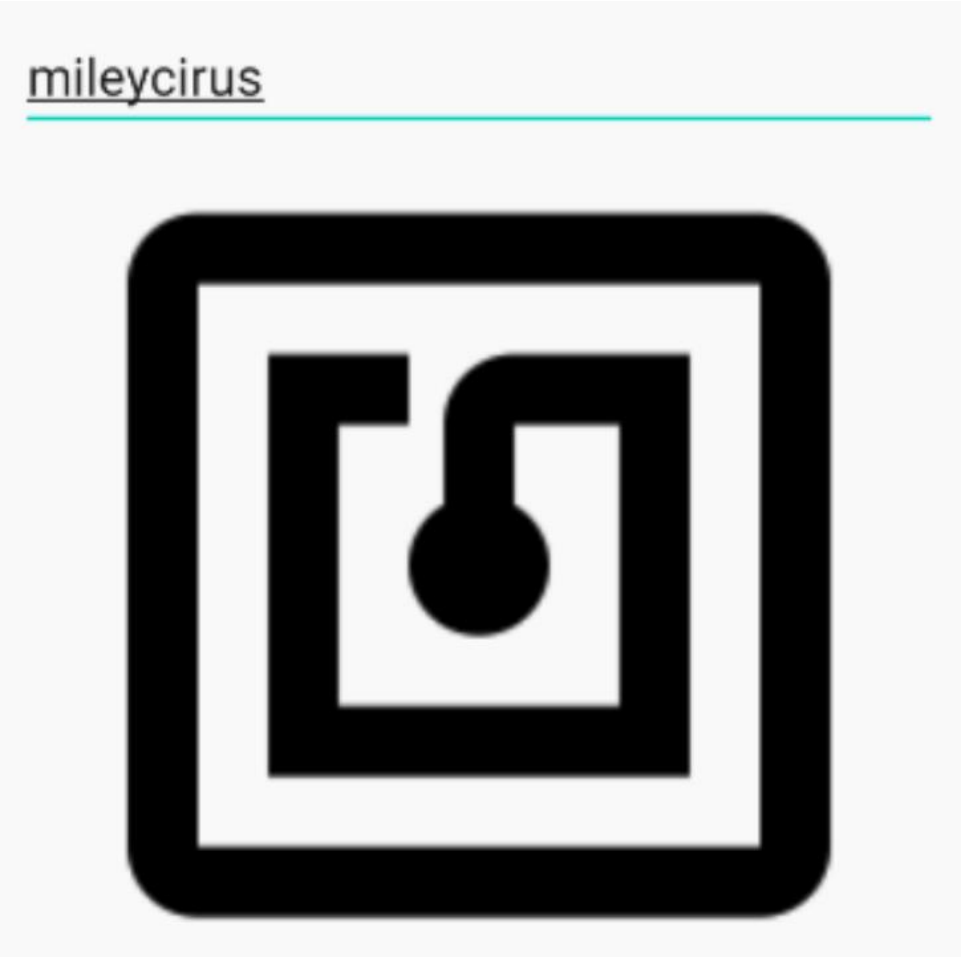


Рисунок 3.35 – Метод автентифікації за допомогою технології NFC

При автентифікації за допомогою методу прихованого пароля користувачеві показується кількість полів, що дорівнює кількості символів у його паролі. Поля з сірими номерами позначені як порядкові і не підлягають зміні вмісту. Користувачеві пропонується заповнити змінні поля. За замовчуванням, в кожне поле можна ввести лише один символ. Після введення символу курсор автоматично переходить до наступного поля, яке потрібно змінити. Детальнішу інформацію показано на рисунку 3.36.

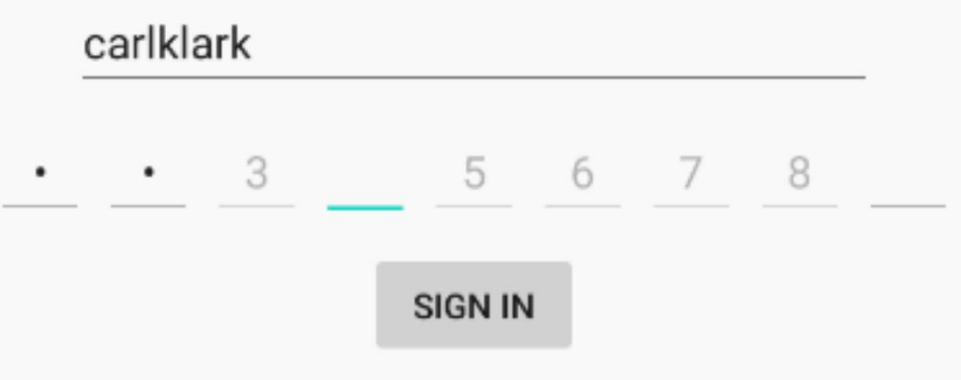


Рисунок 3.36 – Автентифікація за допомогою методу прихованого пароля

Після успішної авторизації будь-яким із способів користувач отримує повідомлення про те, що авторизація пройшла успішно. Більше можна побачити на рисунку 3.37.

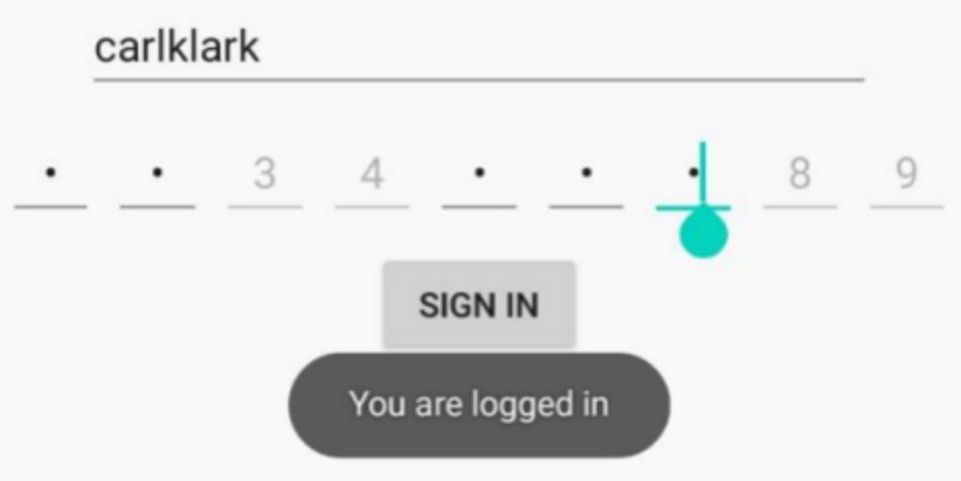


Рисунок 3.37 – Успішний вхід в систему

Найменшою одиницею, якою можна виміряти проміжок часу між запуском і виконанням операції, є наносекунда. Для більш зручного зчитування результатів було вирішено вимірювати виконання операції в мікросекундах, оскільки ця одиниця показує результати без додаткових нулів, що допомагає швидше зрозуміти отриману інформацію. Функція, яка зчитує поточний час у мікросекундах, є системною функцією (рисунок 3.38).

```
System.currentTimeMillis();
```

Рисунок 3.38 – Метод отримання замаскованого пароля

Початок відліку авторизації починається з моменту натискання кнопки "Далі", і ця інформація фіксується у вікні терміналу. Кінець відліку авторизації настає, коли дані для входу і дані для перевірки пароля повертаються з сервера і відображаються у вікні налагодження програми. Час, витрачений на авторизацію, обчислюється в мілісекундах, оскільки період авторизації займає відносно небагато часу і найвищий рівень точності для субтестів у цій ситуації буде пріоритетним. Для кожного методу авторизації користувача статистика успішних спроб авторизації зберігається окремо.

Група користувачів, яка тестувала додаток, складається з 25 осіб. Користувачі - чоловіки та дівчата віком від 17 до 25 років. Кожен користувач мав попередній досвід використання класичного методу автентифікації за допомогою пароля та методу автентифікації за допомогою графічного шаблону. Лише 16 з 25 мали попередній досвід використання методу автентифікації за допомогою прихованого пароля. Жоден з користувачів не мав попереднього досвіду авторизації за допомогою технології NFC. Користувачі використовували унікальні комбінації імені користувача та пароля для кожного методу автентифікації для відповідних методів автентифікації користувачів. Наприкінці тесту кожен користувач пройшов опитування, в якому відповів на запитання:

1. За шкалою від 1 до 10, наскільки складно було зрозуміти принцип роботи з методом автентифікації користувача?
2. Наскільки тривалим здавався процес авторизації за шкалою від 1 до 10?
3. За шкалою від 1 до 10, наскільки вам подобається цей метод авторизації?
4. Чи використовуєте ви цей метод у своєму житті?

3.1.2 Тестова платформа

При проведенні тестів важливо забезпечити максимально рівні умови тестування, щоб зовнішні фактори не впливали на результати тестування методу автентифікації користувача. В якості тестового пристрою ми використовували

телефон Motorola Droid Turbo з підтримкою NFC, 5,2 -дюймовим екраном з роздільною здатністю 1440 x 2560 пікселів.

3.2 Тестування обраних методів авторизації

У цьому розділі описано та проаналізовано методи авторизації, обрані для дослідження.

З представлених методів автентифікації користувачів для дослідження були обрані наступні методи:

· Класичний метод автентифікації за допомогою пароля

· Метод авторизації за формулою

· Метод автентифікації за допомогою прихованого пароля

3.2.1 Класичний метод автентифікації за допомогою пароля

Візуальну реалізацію цього методу автентифікації користувача можна побачити на рисунку 3.38.

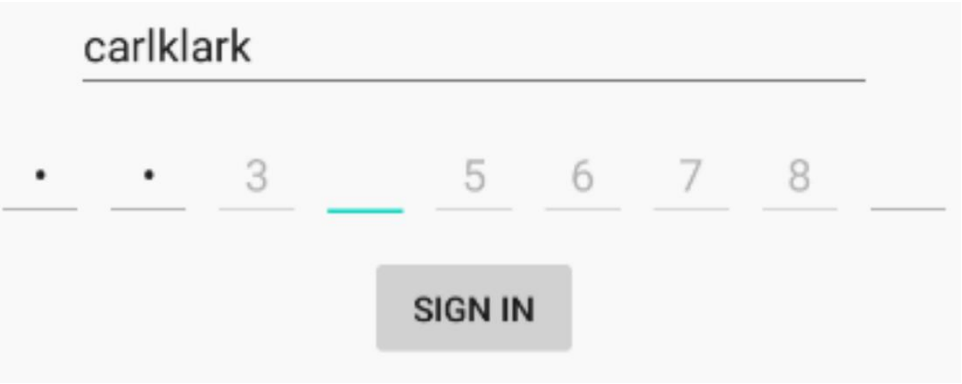


Рисунок 3.38 – Форма автентифікації з використанням класичного пароля

Нижче наведено таблицю, що містить дані тесту автентифікації користувача за допомогою введення класичного пароля. Різні дані зведено в одну таблицю для більш зручного читання та аналізу результатів. У таблиці 3.1 наведено результати тестування цього методу.

Таблиця 3.1 – Результати тестування методу автентифікації з використанням класичного паролю

Номер користувача	Пароль	Довжина пароля	Кількість помилок	Середній час авторизації [мс]	Кількість спроб входу в систему
1	Dark5+sick	10	1	10651	19
2	45679Wuwu_	10	3	8808	15
3	5672Ryry	8	1	8022	16
4	Bubu-67892	10	2	9085	16
5	6789Cdef-	9	0	5444	11
6	After_67893	11	4	8305	13
7	56784Ququ!	10	3	6069	11
y8	34567Bibi!	10	0	9375	14
9	School_3456	12	4	7170	13
10	Want!23452	10	3	6171	12
11	Open-45675	10	1	8231	16
12	7897Ruxe-	9	0	8586	17
13	4569Usual-	10	1	8945	16
14	56782Bus!	9	1	6313	13
15	7894Fly-	8	2	10643	19
16	3458Winter-	11	1	10689	18
17	7897Opq-	8	0	9497	17
18	4564Free-	9	2	6812	15
19	23452Become!	12	4	7697	16
20	Ghij!45672	10	1	11313	17
21	Sleep-12341	11	1	10027	16
22	Plane!7892	10	4	11901	24
23	Love!7897	9	2	5652	10
24	Engine!3452	11	0	9870	18
25	Set_3452	8	4	9546	15

У таблиці наведено результати автентифікації користувача за допомогою класичного пароля. Таблиця містить узагальнені дані зі спроб автентифікації користувачів. Паролі користувачів мають довжину від 8 до 12 символів. Більш детальну інформацію показано на рисунку 3.39.

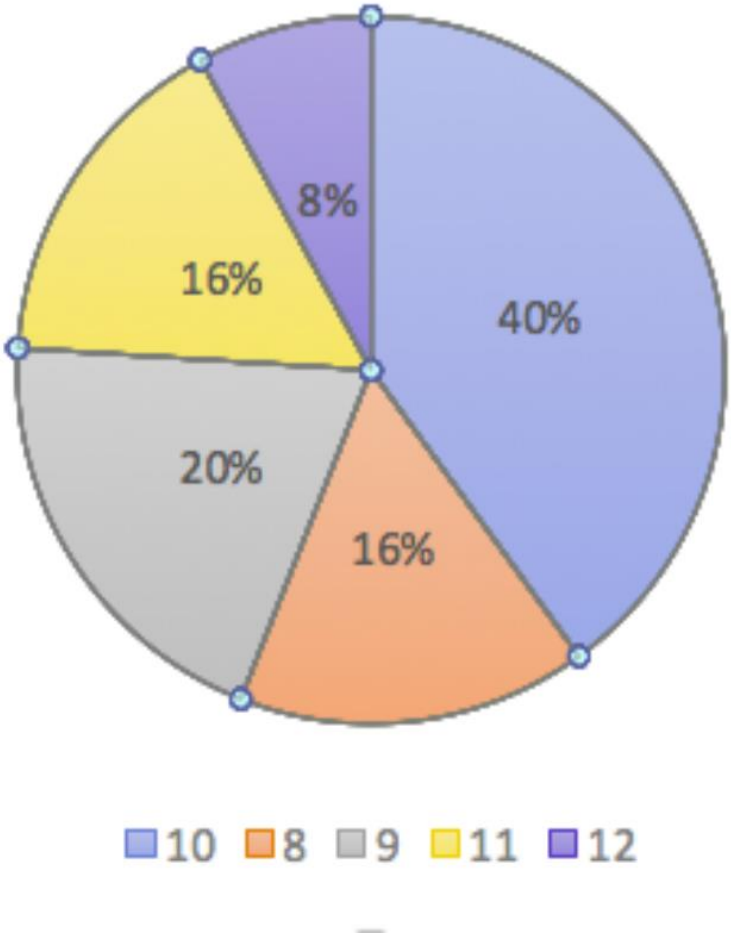


Рисунок 3.39– Довжина пароля користувача

На рисунку показано відсоток випадків використання паролів певної довжини, які були придумані користувачами під час реєстрації. Користувачі, які тестували додаток, частіше обирали середній пароль з 10 символів. Схоже, що користувачі інтуїтивно обирали пароль, який не був ані занадто довгим, ані занадто коротким, щоб збалансувати швидкість введення та безпеку під час автентифікації.

Ще одне дослідження полягає у вимірюванні довжини пароля (рис. 3.40) та кількості помилок під час спроби входу (рис. 3.41). Було помічено, що у випадках з паролями максимальної довжини частіше відбуваються спроби авторизації. Крім того, у випадках з паролями мінімальної довжини частіше трапляється найменша кількість помилкових авторизацій або їх немає взагалі.

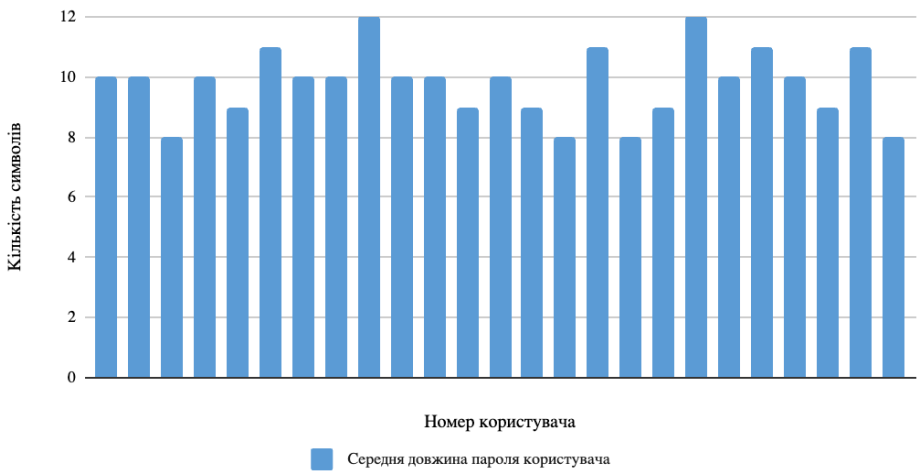


Рисунок 3.40 – Середня довжина пароля користувача

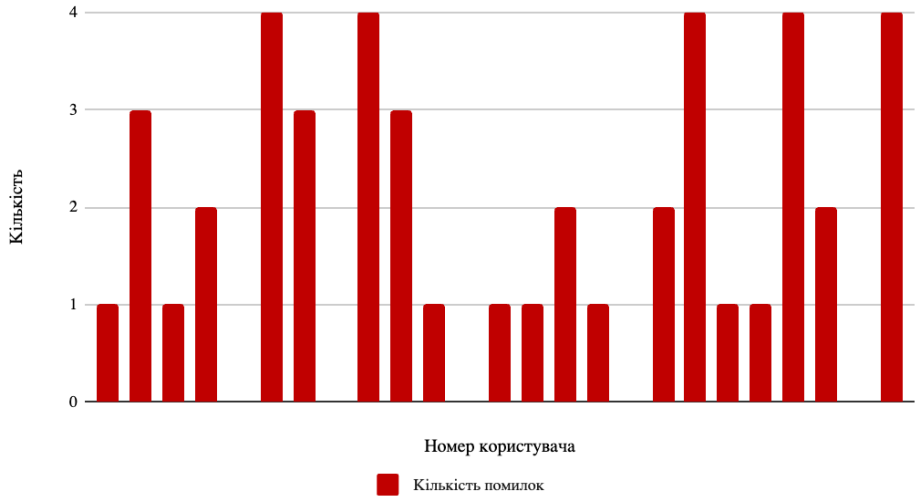


Рисунок 3.41 – Кількість помилок автентифікації користувачів

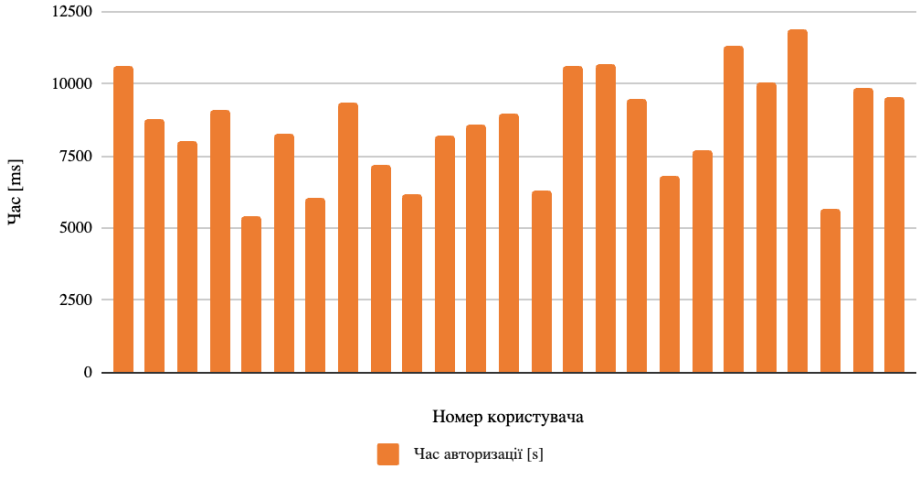


Рисунок 3.42 – Середній час авторизації користувачів

Дослідження взаємозв'язку між довжиною пароля та часом, витраченим на автентифікацію (рис. 3.42), показало, що в середньому кількість символів у паролі приблизно дорівнює тій самій кількості секунд, яка необхідна для завершення процесу автентифікації. У цьому випадку важливу роль відіграє людський фактор. Очевидно, що користувач вводить символи в середньому набагато швидше, ніж за секунду. При цьому необхідно також враховувати час, необхідний для концентрації уваги на введенні пароля, а також час, необхідний для зміщення фокусу з поля введення і клавіатури на кнопку для підтвердження виконаної дії.

Таблиця 3.2 – Найшвидший та найповільніший дозвіл

	Час [с]
Найповільніша авторизація	12,631
Найшвидша авторизація	3,928

Згідно з дослідженням, представленим у Таблиці 3.2, мінімальний час, витрачений на автентифікацію за допомогою введення класичного пароля, становив 3,9 секунди. Максимальний час, витрачений на автентифікацію при введенні класичного пароля, склав майже 12,6 секунди. Більш детально це показано на рисунку 3.43.

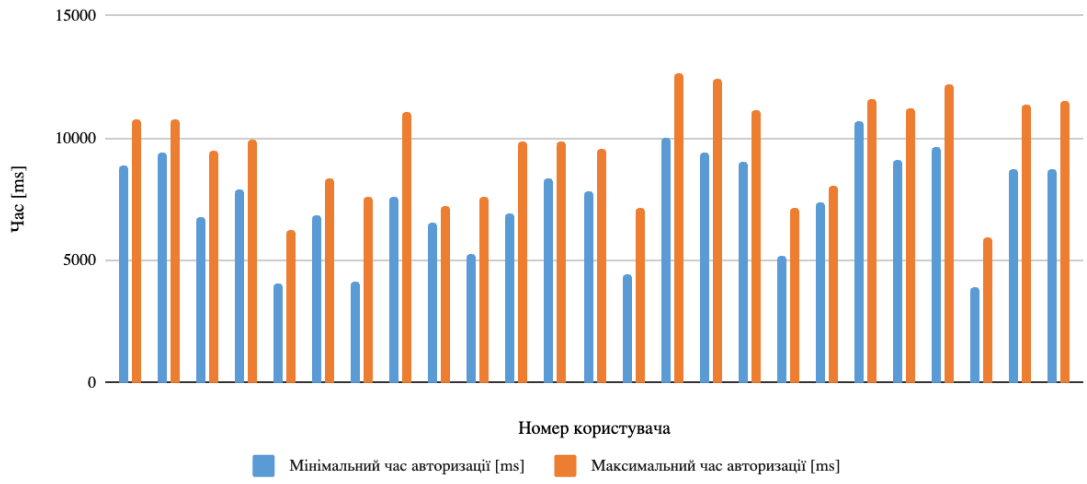


Рисунок 3.43 – Мінімальний та максимальний час авторизації користувача

Опитування користувачів показало очікувані результати. Класичний метод автентифікації за допомогою пароля є одним з найстаріших і найпопулярніших методів автентифікації користувачів. Результати опитування показали, що абсолютно всі користувачі використовують цей метод авторизації. Середня оцінка того, наскільки користувачам подобається класичний метод авторизації за допомогою пароля, склала 6,4 бала. Середня оцінка цього методу з точки зору тривалості авторизації склала 1,9 бала, що є дуже хорошим показником. Через складність розуміння того, як працювати з методом авторизації за допомогою класичного пароля, користувачі оцінили цей метод в середньому на 1,7 бала. Це дуже хороший показник, оскільки всі користувачі були раніше знайомі з цим методом авторизації користувачів. На запитання про використання цього методу в житті 100% користувачів відповіли позитивно, оскільки цей метод автентифікації користувачів використовується всіма ними з самого початку його використання користувачами Інтернету та локальних мереж. На рисунку 3.44 показані оцінки користувачів.

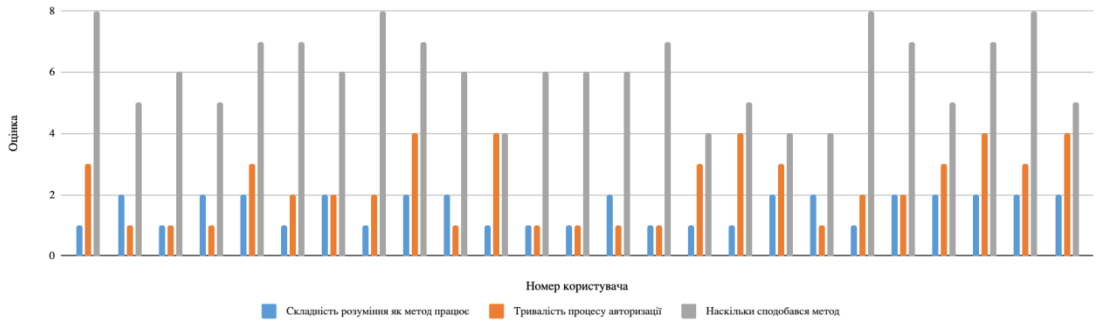


Рисунок 3.44 – Результати опитування користувачів

За результатами опитування, наведеними на рисунку, стало зрозуміло, що ні в кого не виникло труднощів з розумінням роботи з цим методом авторизації. Це пов'язано з тим, що спосіб авторизації користувачів за допомогою класичного пароля та робота з ним знайома кожному користувачеві. Тривалість процесу авторизації здалася користувачам нижчою за середню. Результати виглядають так, тому що, хоча на написання пароля потрібен час, за багато років використання

цього методу авторизації користувачі звикли працювати з ним швидко і автоматично.

3.2.2 Метод авторизації графічним ключем

Візуальну реалізацію цього методу автентифікації користувача показано на рисунку 5.46.

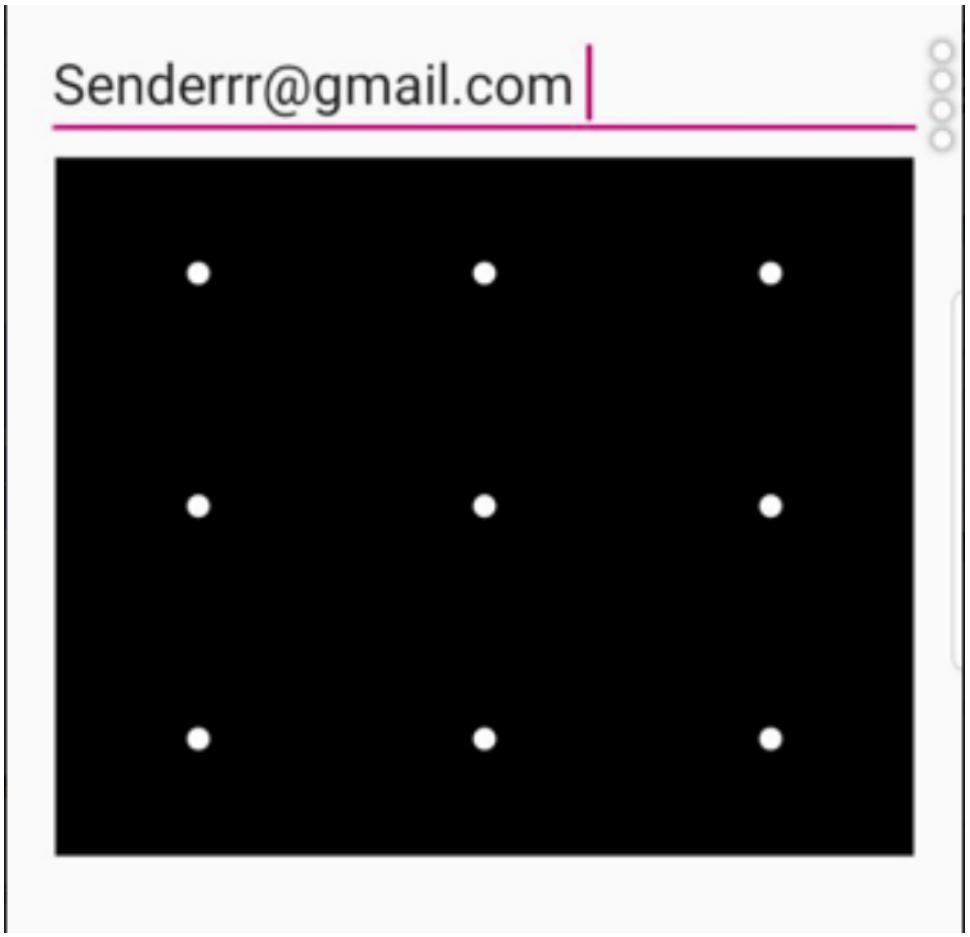


Рисунок 3.45 – Форма авторизації за графічним ключем

При введенні пароля методом автентифікації користувача за графічним ключем в паролі можна виділити такі особливості, як кількість точок, що використовуються при введенні користувача, і довжина пароля, що зберігається в базі даних. Довжина паролів, що зберігаються в базі даних для цього методу, однакова і складається з восьми символів. Така кількість символів виходить після шифрування даних про введений пароль. Кількість точок, які використовуються при введенні пароля, серед користувачів варіюється від 6 до 9. Судячи з результатів опитування, найбільша кількість користувачів віддає перевагу паролю, що

складається з 7 точок. Такий вибір зробили 10 користувачів з 25. Більш детальні дані наведені на рис. 3.46.

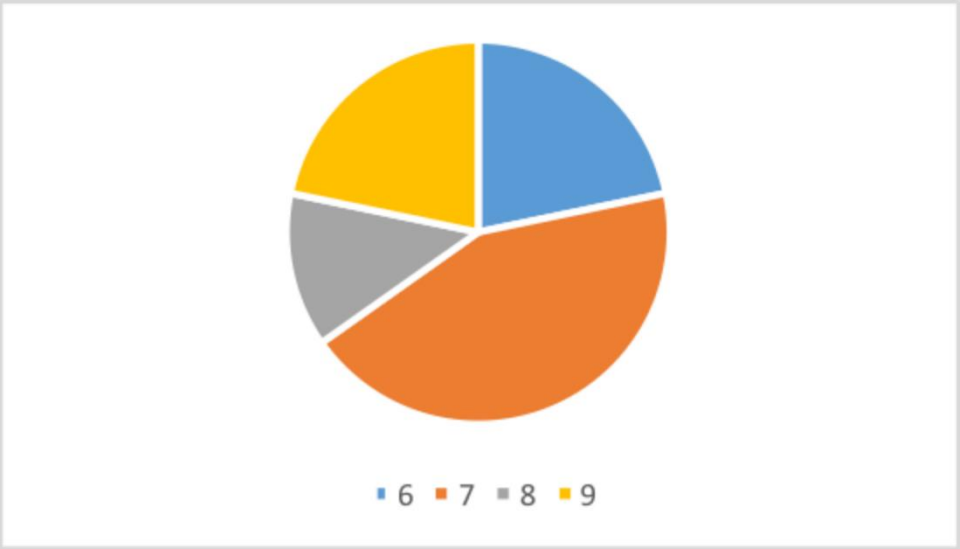


Рисунок 3.46 – Довжина пароля користувача

Судячи з даних опитування, користувачі зацікавлені у більшій безпеці, оскільки пароль, що складається з шести символів, використовувався рідше. Кількість помилок при аутентифікації, допущених користувачами при використанні цього методу, у всіх випадках не перевищує п'яти. У більшості випадків користувачі робили менше трьох помилок при спробі входу в систему. Більш детальна інформація наведена на рисунку 3.47.

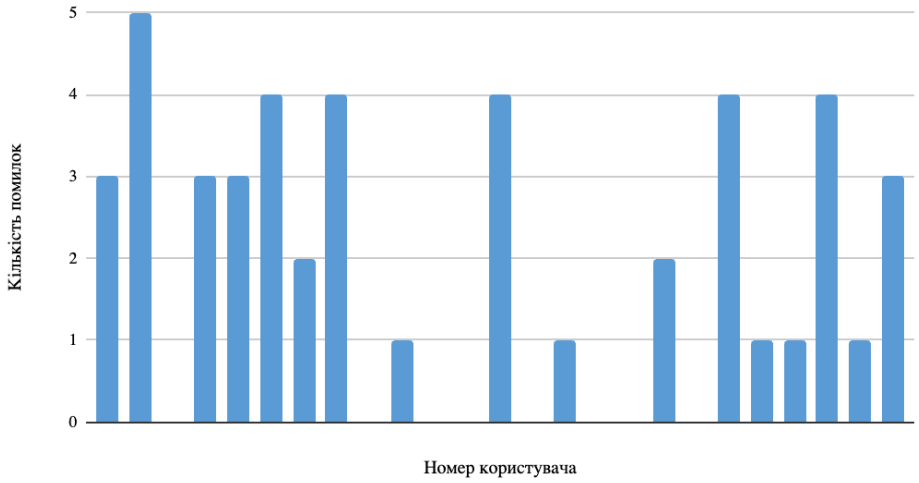


Рисунок 3.47 – Кількість помилок користувачів

Зручність і швидкість використання цього методу авторизації означає, що користувачі можуть швидко входити в систему. Середній час авторизації

користувача становить 2,5 секунди. Найшвидша авторизація зайняла 1,2 секунди. Найдовша авторизація зайняла 4 секунди. Більш детальна інформація наведена на рисунку 3.48.

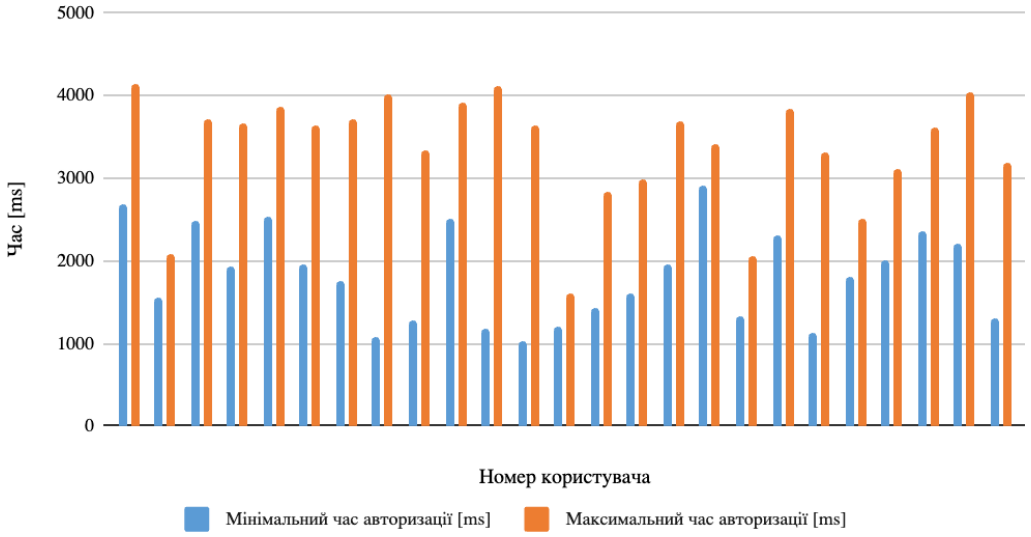


Рисунок 3.48 – Мінімальний та максимальний час авторизації користувача

Результати дослідження показали, що цей спосіб автентифікації користувача за всіма показниками потребує найменшої кількості часу для автентифікації користувача в досліджуваному додатку. Більш детальні дані наведено на рисунку 3.49.

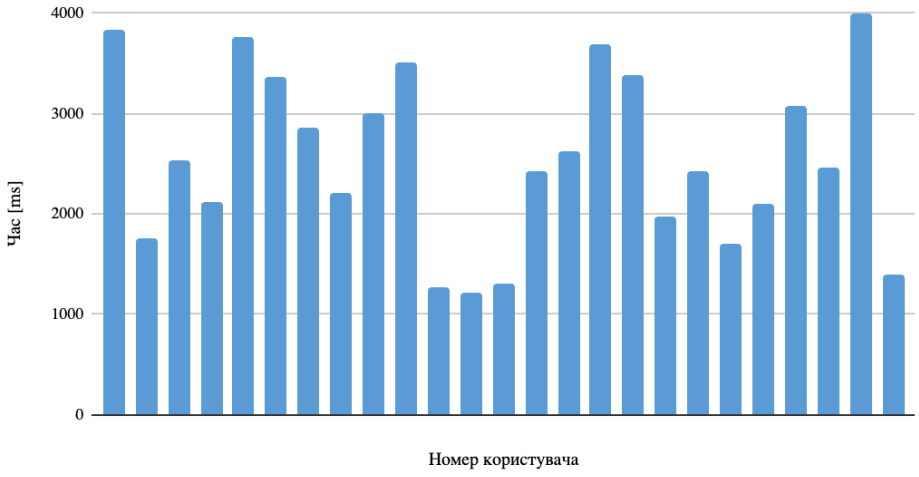


Рисунок 3.49 – Середній час авторизації користувачів

Як бачимо, середній час авторизації користувачів є неоднорідним. Це пов'язано з тим, що різні користувачі мають індивідуальний рівень готовності та

набір звичок для роботи з цим методом. Водночас можна побачити, що всі дані тримаються відносно близько до значення 2500 - сумарного середнього значення швидкості авторизації користувачів, які скористалися цим методом. Більш детальна інформація наведена на рисунку 3.29.

Багато користувачів відзначили, що цей метод є дуже швидким, впізнаваним і зручним. Результати опитування показують, що 68% користувачів використовують цей метод автентифікації. Більш детальна інформація наведена на рисунку 3.50.

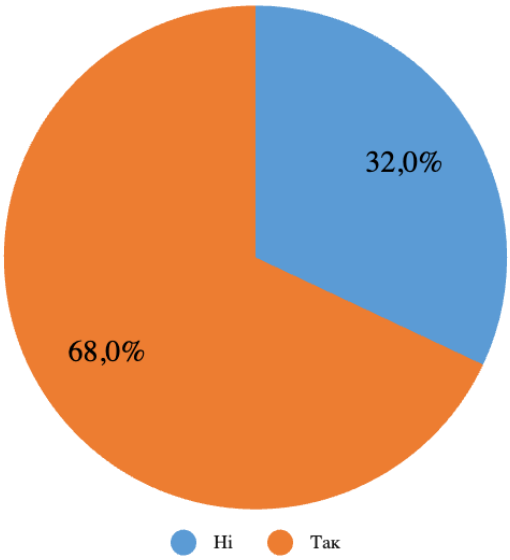


Рисунок 3.50 – Чи отримують користувачі користь від методу

На питання про простоту концепції принципу роботи користувачі відповіли в середньому 2 бали. Такий результат свідчить про те, що освоєння цього способу автентифікації користувачів не вимагало майже ніяких зусиль. Дуже велику роль у такому позитивному результаті відіграє той факт, що цей спосіб автентифікації користувача знайомий багатьом користувачам через можливість аналогічним чином встановлювати екран блокування телефону.

Тривалість процесу авторизації, згідно з досвідом користувачів, не зайняла багато часу. Оскільки більшість користувачів стикалися з цим методом авторизації, він допоміг їм швидше і точніше вводити ключ. Як наслідок, користувачі використовували сформовану звичку користуватися цим методом авторизації, і

середній час авторизації скоротився. Середня оцінка користувачів за питання про тривалість авторизації склала 1,92 бала.

На запитання, наскільки користувачам сподобався цей метод авторизації, були отримані різні відповіді. Деяким користувачам сподобався цей метод, навіть якщо вони не були знайомі з ним. Деякі користувачі оцінили метод середньо, незважаючи на те, що мали достатній досвід використання цього методу авторизації користувачів раніше. Пояснюючи, чому метод викликав нейтральну реакцію, незважаючи на те, що він був добре знайомий, користувачі відповіли, що метод досить старий і, хоча він зручний, вони насправді хотіли б спробувати щось нове. Більш детальні дані наведені на рисунку 3.51.

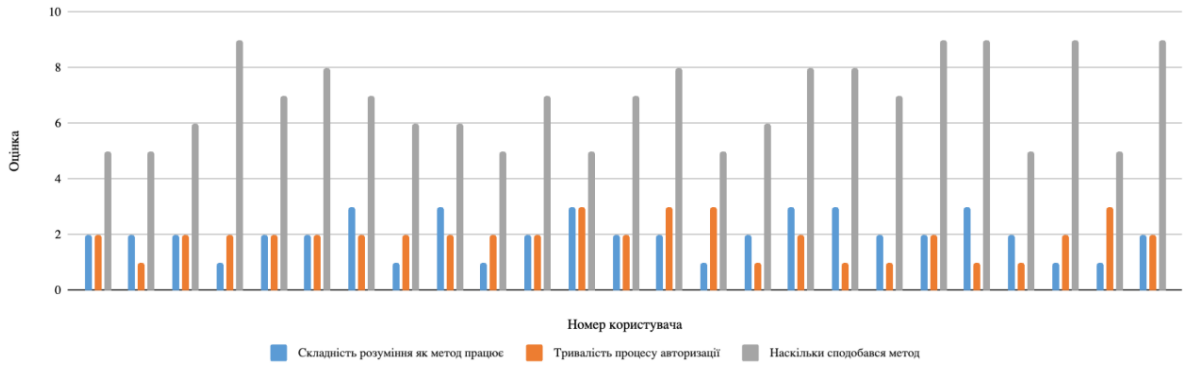


Рисунок 3.51– Результати опитування користувачів щодо досвіду використання методу авторизації шаблонів

Дослідження цього методу автентифікації користувачів показало не лише очікувані результати. Вдалося також з'ясувати, що, незважаючи на простоту використання і звичність, користувачі також хочуть спробувати щось нове.

3.2.3 Метод автентифікації за допомогою пароля з маскою

Візуальну реалізацію цього методу автентифікації користувача показано на рисунку 3.52.

Рисунок 3.52 – Форма автентифікації за допомогою прихованого пароля

У представленому методі користувачі використовують ті ж самі паролі, які були встановлені при реєстрації для класичного методу парольної автентифікації користувачів. При спробі входу користувачі допустили різну кількість помилок. Більш детальні дані наведено на рисунку 3.53.

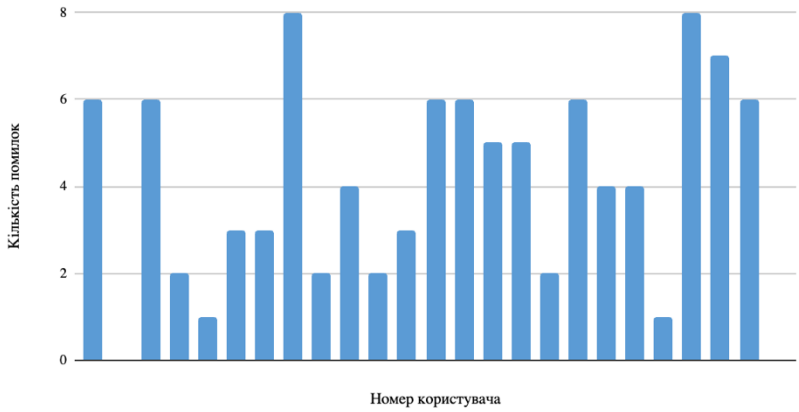


Рисунок 3.53 – Кількість помилок користувачів

Коли пароль вводиться при автентифікації за допомогою пароля з маскою, використовуються лише перші 8 символів пароля. Виходячи з цього, можна зробити висновок, що довжина пароля не впливає на швидкість автентифікації. Кількість помилок, показаних на рисунку, залежить від того, наскільки швидко і наскільки уважно користувачі вводили пароль.

Середній час авторизації для всіх спроб становить 12,2 секунди. Якщо розглядати цей метод у порівнянні з іншими методами авторизації користувачів, то метод авторизації за допомогою прихованого пароля в середньому займає більше часу, ніж будь-який з інших представлених методів. Найменший час авторизації

склав 5,5 секунд. Найбільше часу на авторизацію було витрачено 18,7 секунд. Більш детальна інформація наведена на рисунку 3.54.

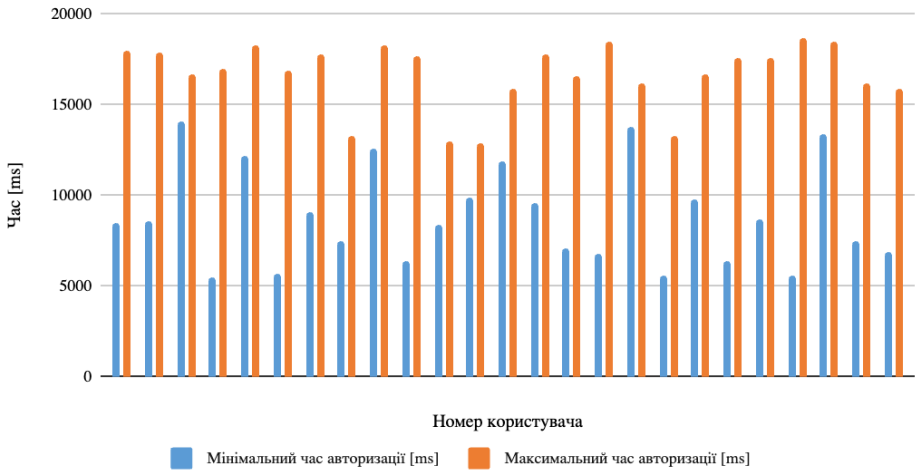


Рисунок 3.54 – Мінімальний та максимальний час автентифікації користувача з використанням пароля з маскою

Під час опитування реакція користувачів була більш незадовільною, ніж у порівнянні з іншими методами автентифікації. Варто зазначити, що у випадку більш складного введення паролю порівняно з іншими розглянутими методами автентифікації користувачів, ця складність врівноважується значно вищим рівнем безпеки. Користувачі оцінили питання складності концепції принципу роботи від 1 до 6 балів. Після знайомства з цим методом автентифікації деякі користувачі не звикли до більш складної форми введення загального пароля. Більш детальна інформація наведена на рисунку 3.55.

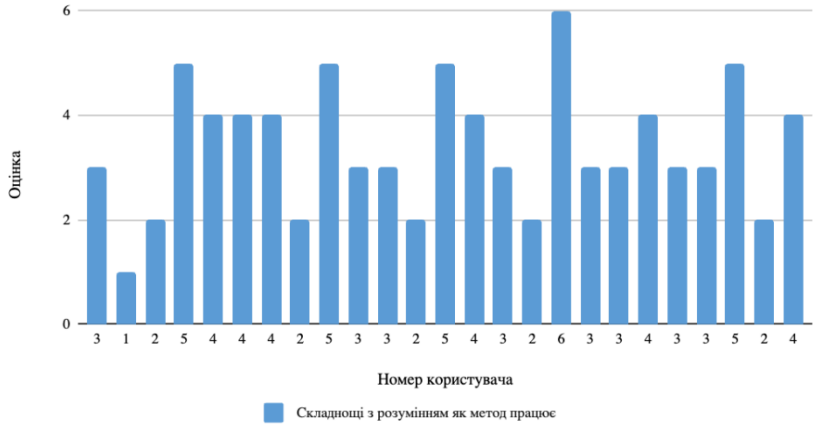


Рисунок 3.55 – Труднощі з розумінням того, як працює метод автентифікації за допомогою прихованого пароля

Як бачимо, мало хто з користувачів цього методу авторизації вважав його дуже складним. Середня оцінка складності розуміння того, як працює цей метод, склала 3,4 бали. На запитання про тривалість процесу авторизації користувачі відповіли, що оцінили б цей метод від 3 до 5 балів. Цей результат показує, що користувачі також вважають, що представлений метод авторизації користувачів займає більше часу, ніж інші методи авторизації користувачів. Більш детальна інформація наведена на рисунку 3.56.

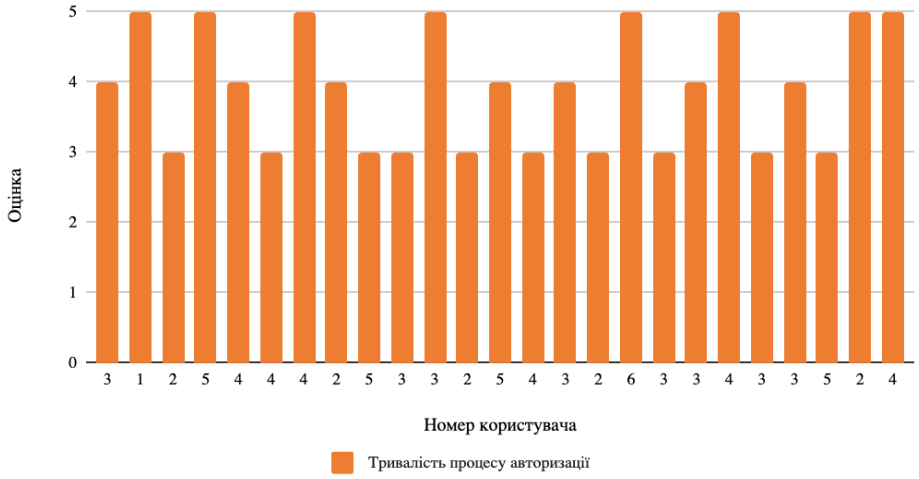


Рисунок 3.56– Оцінка тривалості процесу автентифікації для методу автентифікації за прихованим паролем

На запитання, наскільки користувачам сподобався представлений метод, були отримані неоднозначні дані. Комусь метод сподобався, комусь - дуже мало. Деякі користувачі зазначили, що хоча цей метод автентифікації користувачів займає більше часу, вони добре розуміють, що це компенсується вищим рівнем безпеки. Детальніше див. рисунок 3.57.

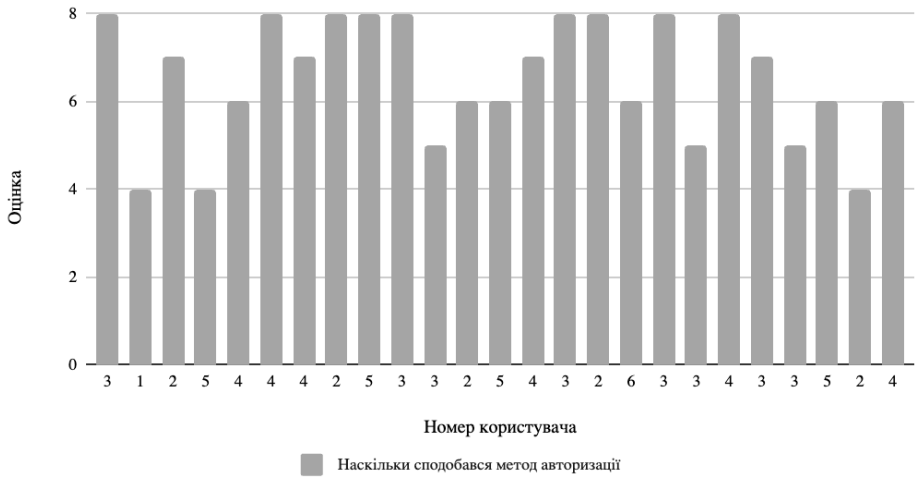


Рисунок 3.57– Наскільки сподобався користувачам метод автентифікації за допомогою прихованого пароля

Користувачі оцінили цей метод від 4 до 8 балів. Крім того, середня оцінка всіх користувачів склала 6,5 балів.

На запитання, чи використовують користувачі цей метод у своєму житті, лише 6 користувачів відповіли позитивно. Це можна пояснити тим, що цей спосіб автентифікації користувачів використовується рідко і не у всіх системах. Більш детальна інформація з цього питання наведена на рисунку 3.58.

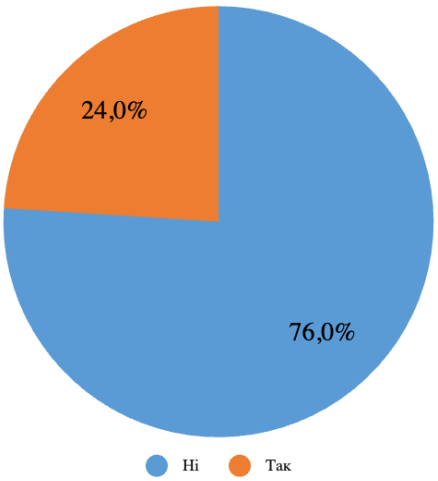


Рисунок 3.58– Чи використовують користувачі метод автентифікації за допомогою прихованого пароля

Можна підсумувати, що цей метод виявився найповільнішим, хоча його варто розглянути, оскільки рівень безпеки, підтримуваний цим методом

автентифікації користувачів, дає йому перевагу в цій категорії над іншими методами автентифікації користувачів.

3.2.4 Тестування методу автентифікації за допомогою технології NFC

Візуальну реалізацію цього методу автентифікації користувача показано на рисунку 3.59.

Рисунок 3.59 – Форма автентифікації за допомогою технології NFC

Цей спосіб має кілька особливостей. При реєстрації користувача не потрібно вводити або придумувати пароль. Замість цього генерується замаскований пароль з випадковою довжиною від 16 до 32 символів. Пароль для методу автентифікації користувача NFC використовує різні випадкові символи. Пароль генерується з використанням літер англійського алфавіту, цифр і малих літер. Кількість неправильних спроб авторизації дорівнює нулю, оскільки пароль не вводиться користувачем особисто, а зчитується з NFC-мітки. Деталі показано на рисунку 3.60.

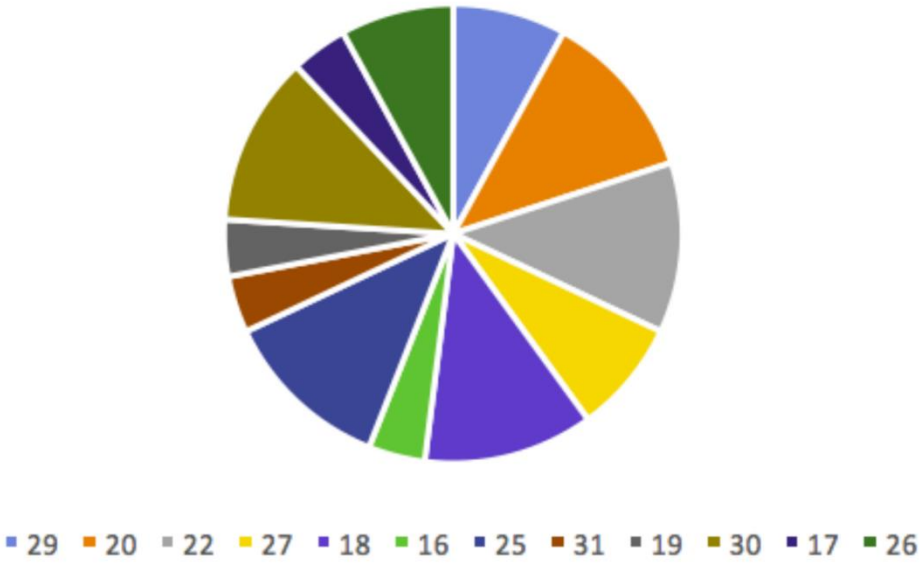


Рисунок 3.60 – Довжина пароля користувача

Із зображень на діаграмі можна зрозуміти, що кількість символів у паролі не має ніякої закономірності і є абсолютно випадковою в заданих межах.

Дослідження показало, що всі спроби автентифікації були успішними і що ймовірність успішної автентифікації становить 100%. Цей факт настільки значущий, тому що людина не бере участі в процесі зберігання пароля в пам'яті і, відповідно, в безпосередньому введенні пароля. Всі дані про пароль зберігаються в NFC-мітці, а телефон зчитує і обробляє ці дані. Таким чином, виходить, що користувачеві потрібно лише ввести логін і прикласти NFC-мітку до телефону. Більш детальна інформація про це наведена в Таблиці 3.3.

Таблиця 3.3 – Результати тестування методу автентифікації з використанням технології NFC

Номер користувача	Пароль	Довжина пароля
1	t:ZVq,!qG_3oZ6EBq4468))_q5aqH	29
2	88+UTNB7=glOxi{.Q+7v	20
3	35,Hdi5mZ*LQw}#0iR№s31	22
4	c#m&N5pFW~cWA93655q(zVS}[7i	27

Продовження таблиці 3.3

5	Yy7]6$Wk7G5^2$scvZ	18
6	79:N_(vP9oqJK8#g	16
7	1lqVZ4$rBx8r#L#!@I4?1C3rl	25
8	5;7q38C{Lo]i-lvt3y,2**f9DCV3GLF	31
9	q3Y4%6$$81(kLSuztBQ	19
10	FeCfS!&AZ30?d-#qV8vOLy20-2?x8z	30
11	cvZK-Rn27&C]5jz8%	17
12	W=dJCEgfV1?a46B}Sc5(8u#a9O8&$g	30
13	#oy-T&AvC4J-ok67U=31~9zQu	25
14	lHPk0&2ny0a%~Tb]Z2~rkXD4R81;}	29
15	Vj@W5E8[;22ivE*$rIJe	20
16	996h8T|CwE$j#u=rT6	18
17	IHc3~IR6%12:e?4G4dLez3%(kWh	27
18	X[C!3N!8LyxzoX71rk-10.H(s	25
19	o6wGy%m]S}s0#88In,{i5N3OIE	26
20	J~BaSCOx*?ktk]v56I-B)Ks48№5c79	30
21	e?DzM01Wtg54:4K.]p	18
22	P|qdc7O77ZbN?7;Do!Y:	20
23	vbuXI(K891V5}u5!&Z%aV!Oy6p	26
24	1fXf]=4dL:vaR46=I1O-i8	22
25	q@F?NKs1Jp3b+h2$4Q0,p2	22

Складні паролі, що генеруються для користувачів, які обирають метод автентифікації NFC, відрізняються за довжиною. З одного боку, якщо розглядати паролі користувачів окремо, то деякі паролі слабші за інші. З іншого боку, якщо при створенні пароля збільшується кількість варіантів, які можна вибрати, то з кожним збільшенням кількості символів в паролі додається багато різних варіацій, що ускладнює злом пароля.

Згідно з дослідженням, середній час авторизації для всіх користувачів склав 2,7 секунди. Ця характеристика дає можливість перевірити, чи дійсно авторизація є швидкою. На рисунку показано середні значення для кожного користувача. Більш детальна інформація представлена на рисунку 3.61.

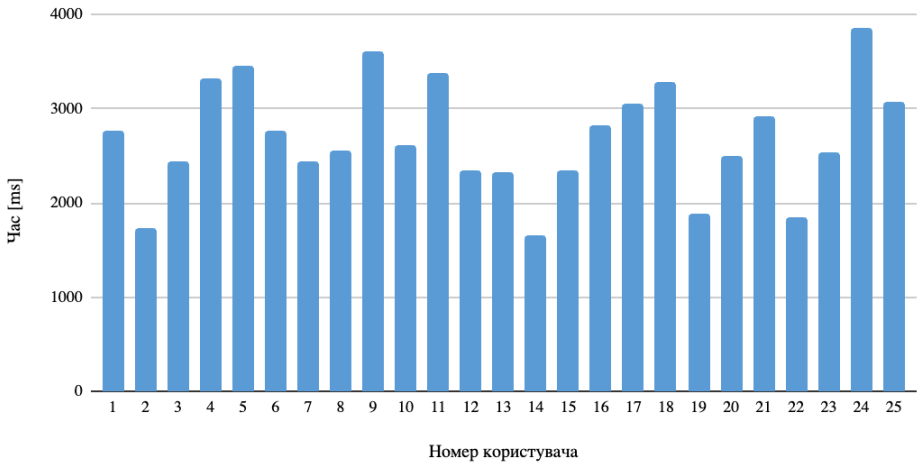


Рисунок 3.61 – Середній час автентифікації користувачів при використанні методу NFC-автентифікації

Мінімальний час авторизації серед усіх спроб всіх користувачів становив 1,5 секунди, що є дуже швидкою авторизацією, враховуючи складність і довжину пароля. Максимальний час, витрачений усіма користувачами, становить 4,9 секунди. Більш детальні дані щодо мінімальної та максимальної кількості користувачів наведені на рисунку. Враховуючи, що не всі дані, які вказують на максимальний час авторизації з дослідження, близькі до загального максимального значення, можна ще раз підтвердити, що існує швидкий метод авторизації користувачів за допомогою технології NFC. Більш детальна інформація представлена на рисунку 3.62.

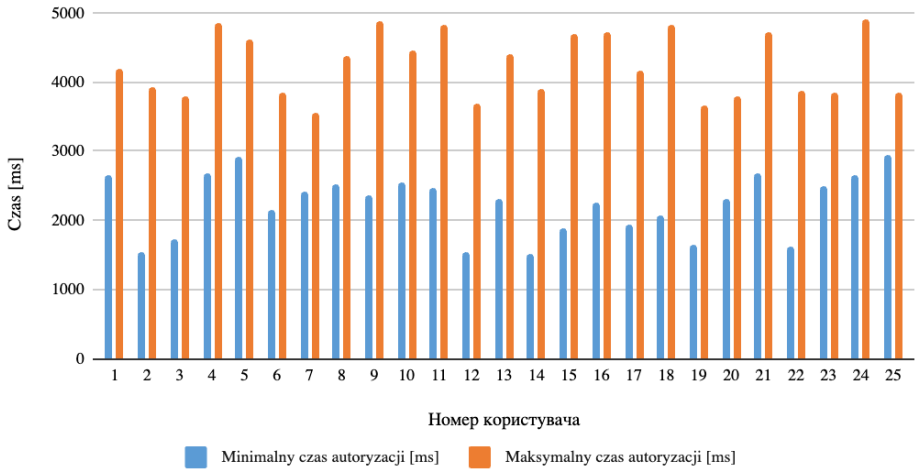


Рисунок 3.62 – Мінімальний та максимальний час автентифікації користувача при використанні методу автентифікації NFC

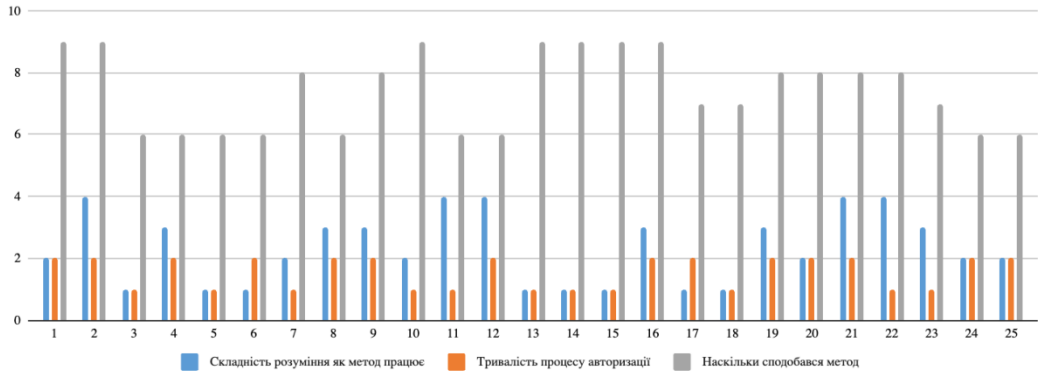


Рисунок 3.63 – Результати опитування користувачів з використанням методу автентифікації NFC

На рис. 3.63 наведено дані, отримані в результаті опитування. З даних видно, що складність під час автентифікації не здалася користувачам складною і за шкалою обирається лише між значеннями 1 та 2, що означає, що вона була дуже простою у використанні. Розуміння роботи методу автентифікації користувача за допомогою технології NFC для користувачів коливається від 1 до 4. Це пов'язано з тим, що даний метод автентифікації в контексті даного додатку змусив деяких користувачів коротко замислитися над тим, як працювати з цим методом. У той же час середній бал за складність розуміння того, як працює цей метод авторизації, склав 2,3 і свідчить про те, що більшість користувачів розібралися досить легко. Користувачам настільки сподобався цей метод, що його оцінили від 6 до 9 балів,

середня оцінка склала 7,4 бала. Результат останнього запитання в опитуванні користувачів показав, що жоден з користувачів не використовує цей метод авторизації у своєму житті.

3.3 Результати порівняння

Дослідження продемонструвало переваги та недоліки кожного з розглянутих методів автентифікації користувачів та дозволило кількісно оцінити параметри методів автентифікації на основі практичного дослідження. При детальному розгляді картини довжини паролів, що використовуються в представлених методах, можна побачити, що під час дослідження, використовуючи метод автентифікації на основі NFC, паролі однозначно генерувалися довше, ніж паролі, придумані користувачами для інших методів автентифікації. Більш детальна інформація наведена на рисунку 3.64.

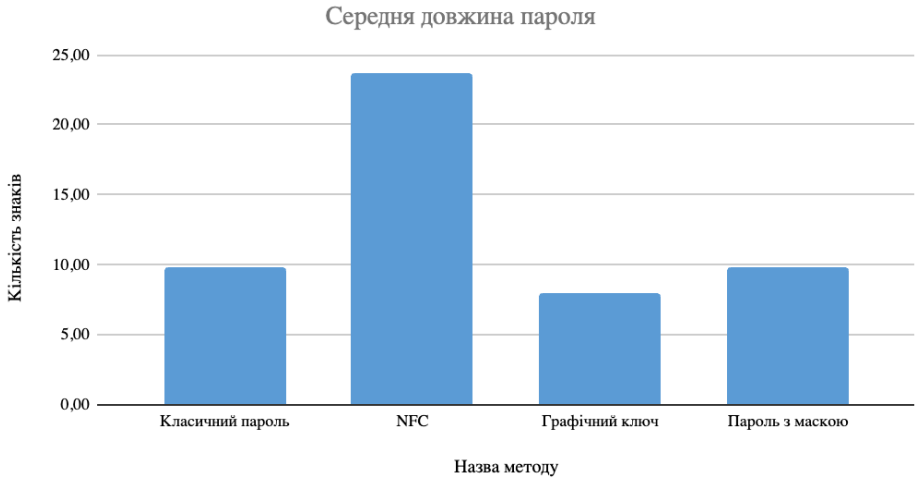


Рисунок 3.64 – Середня довжина пароля

Оскільки пароль зберігається в пам'яті NFC-мітки, а не в пам'яті користувача, метод автентифікації на основі NFC є провідним для автентифікації користувача.

Середню кількість помилкових спроб автентифікації користувача показано на рисунку. Більш детальна інформація наведена на рисунку 3.65.

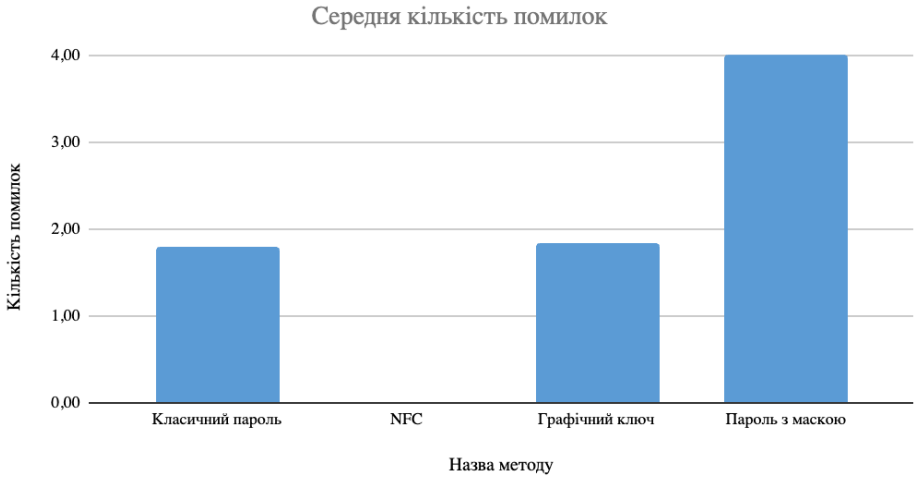


Рисунок 3.65 – Середня кількість помилок

Як бачимо, метод автентифікації за допомогою технології NFC знову ж таки веде до інших методів автентифікації. Завдяки мінімальній кількості дій, які повинен виконати користувач при використанні методу авторизації за допомогою технології NFC, всі операції введення виконуються за допомогою вбудованого в пристрій модуля NFC. Найбільша кількість помилок авторизації пов'язана з методом авторизації за допомогою прихованого пароля.

За середньою швидкістю авторизації, як показано на рисунку 8.3, найкращі результати має метод авторизації за допомогою графічного ключа. У зв'язку з тим, що більшість користувачів раніше використовували цей метод авторизації, у користувачів виробилися певні звички та рефлекси, пов'язані з цим методом авторизації. Більш детальна інформація показана на рисунку 3.66.

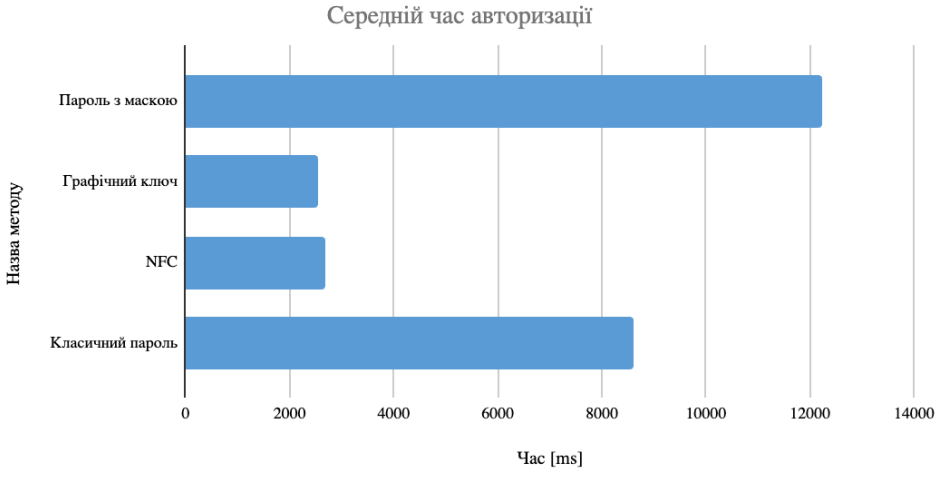


Рисунок 3.66 – Середній час авторизації

Метод автентифікації користувачів на основі NFC показав результати, гідні другого місця за середнім часом автентифікації. Методи парольної автентифікації є відносно складнішими для введення пароля і тому займають трохи більше часу.

Як показано на рисунку 3.67, найбільший час авторизації серед усіх спроб авторизації має метод авторизації за допомогою прихованого пароля. Найкоротший час авторизації серед максимальних показників за методами авторизації користувачів належить методу авторизації за допомогою графічного ключа.

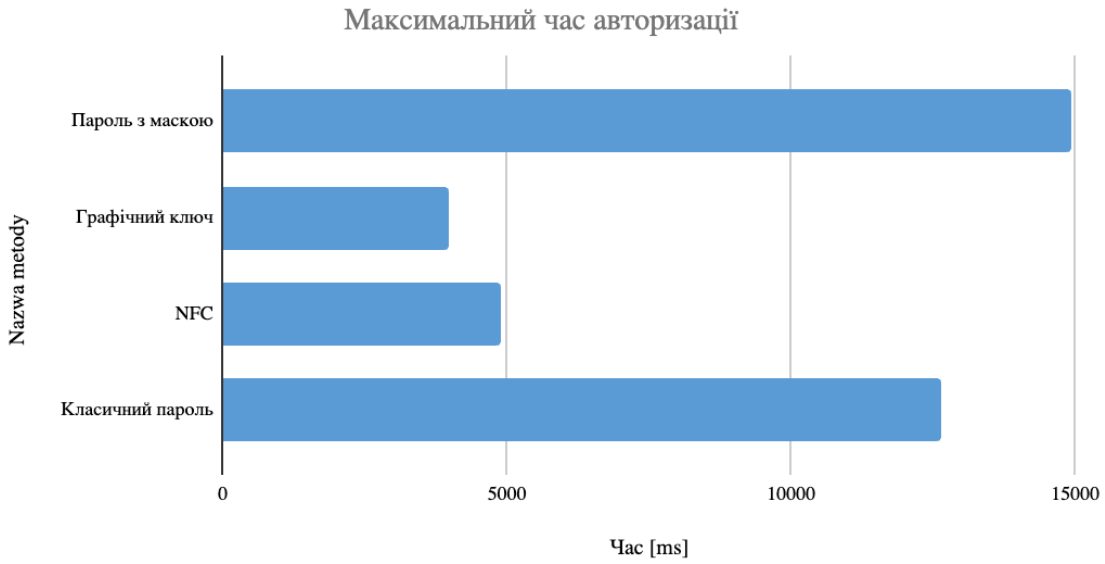


Рисунок 3.67 – Максимальний час авторизації

Мінімальний час авторизації серед методів авторизації належить методу авторизації користувачів за допомогою графічного ключа. На другому місці - метод авторизації користувачів за допомогою технології NFC. Більш детальна інформація наведена на рисунку 3.68.

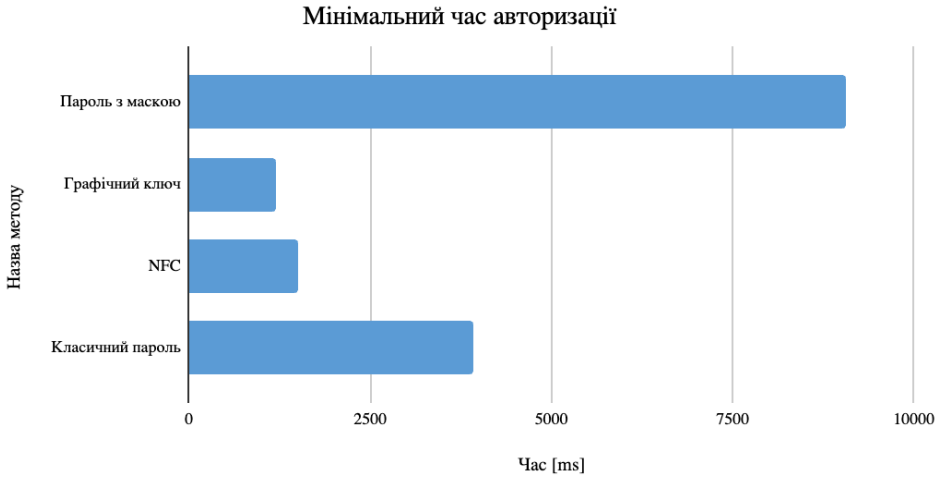


Рисунок 3.68 – Мінімальний час авторизації

Як можна побачити, дивлячись на показники [час авторизації], методи авторизації користувачів, ранжовані за швидкістю авторизації користувачів, займають однакові місця в кожному з досліджень.

Потім будуть представлені результати опитування користувачів.

Щоб відповісти на питання, наскільки складно користувачам було зрозуміти, як працює метод, були отримані середні оцінки користувачів, які використовували методи авторизації, показані на рисунку 3.69.

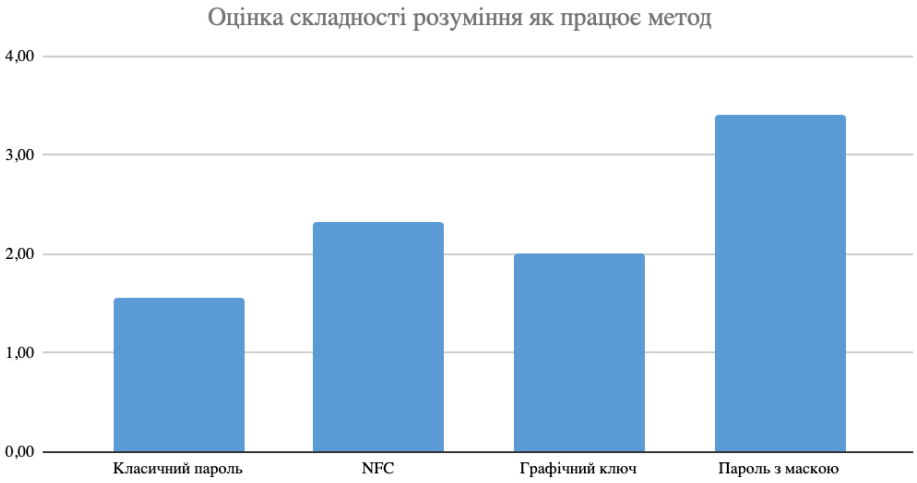


Рисунок 3.69 – Середня оцінка складності розуміння того, як працює метод

На запитання, скільки часу, на думку користувачів, зайняв процес авторизації, були отримані середні оцінки для методів авторизації користувачів, які детально показані на рисунку 3.70.

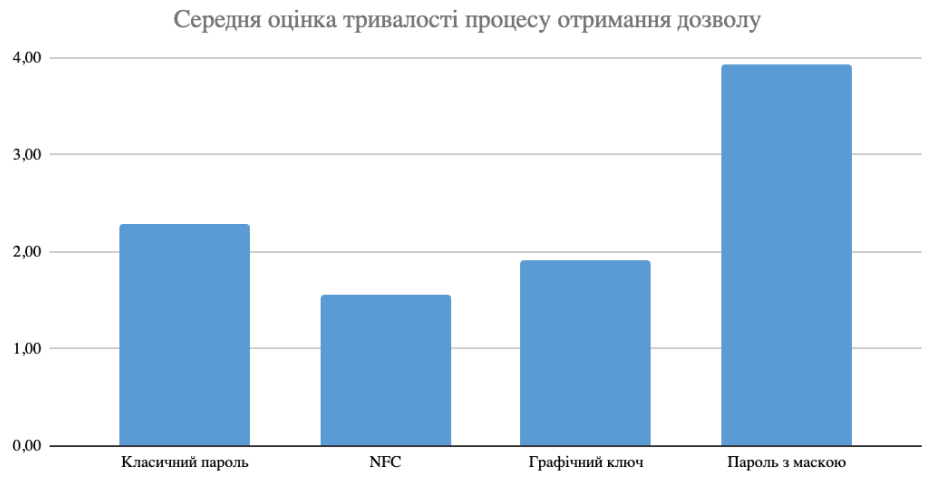


Рисунок 3.70 – Середня оцінка тривалості процесу отримання дозволу

Відповідаючи на запитання, чи користуються користувачі наведеними методами авторизації у своєму житті, було підраховано позитивні відповіді користувачів для кожного методу. Більш детальна інформація представлена на рисунку 3.71.

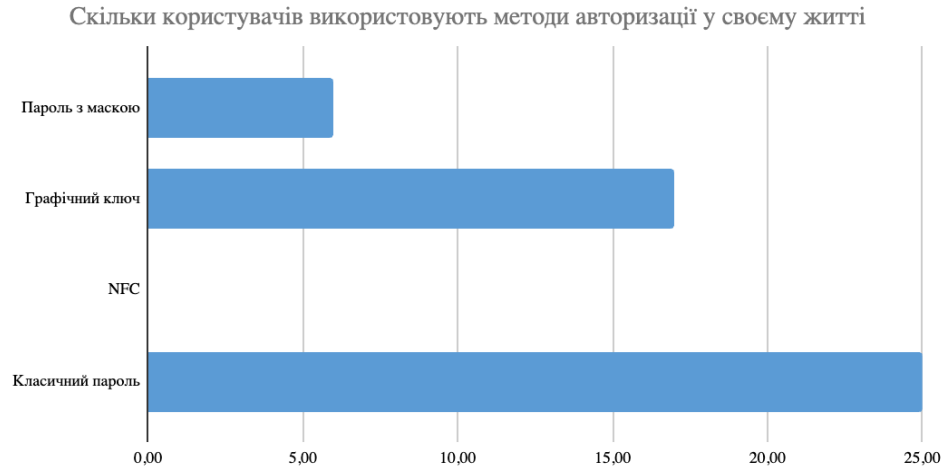


Рисунок 3.71 – Кількість користувачів, які застосовують методи у своєму житті

Як бачимо, 25 з 25 користувачів використовують класичний пароль, оскільки цей метод є дуже популярним. При цьому, жоден з їхніх користувачів в реальному житті не використовує метод автентифікації користувачів за допомогою технології NFC. Також дуже мала кількість користувачів, а саме 6 з 25, використовують метод авторизації за допомогою прихованого пароля. Метод авторизації за допомогою графічного ключа використовують 17 з 25 користувачів.

На запитання про те, наскільки їм сподобалися методи під час опитування, були отримані середні оцінки для методів, які можна побачити на рисунку 3.72.

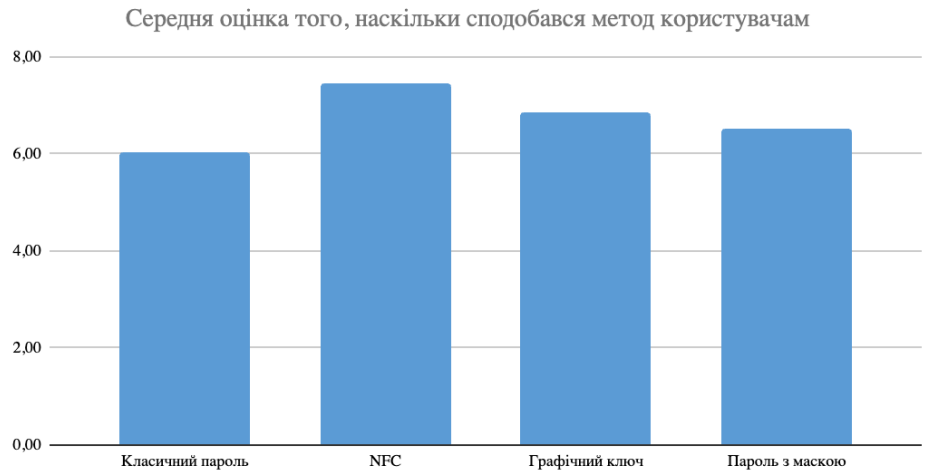


Рисунок 3.72 – Середня оцінка того, наскільки користувачам сподобався той чи інший метод

Найбільше користувачам сподобався метод авторизації користувача за допомогою технології NFC. На другому місці - метод авторизації користувача за допомогою графічного ключа. Останні місця займають методи, що використовують введення пароля для авторизації користувачів.

ВИСНОВКИ

В кваліфікаційній роботі вирішено питання актуальності використання технології NFC у методах авторизації користувачів. При цьому було отримано наступні результати.

Досліджено наукові роботи, пов’язані з технологією NFC, що дало можливість виявити переваги та недоліки технології NFC для подальшого дослідження при порівнянні з іншими методами авторизації користувачів.

Досліджено існуючі методи автентифікації користувачів, що дозволило побачити які з них на даний момент є придатними та забезпечують достатній рівень безпеки.

Досліджено переваги та недоліки методів автентифікації користувачів, що дозволило виявити спільні риси методів автентифікації, а також особливості, які дозволяють побачити унікальні характеристики кожного методу автентифікації.

Досліджено можливості нового методу авторизації користувачів з використанням технології NFC, що дозволило реалізувати цйе метод авторизації користувачів у розробленому додатку на базі Android.

Розроблено тестову платформу, що дало можливість якісно порівняти методи авторизації та провести як точні виміри за допомогою телефона, так і визначити соціальну адаптованість користувачів до розглянутих методів авторизації користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ISO/IEC 14443 [Електронний ресурс]. – Режим доступу: https://pl.wikipedia.org/wiki/ISO/IEC_14443

2. Near-field communication (NFC) system providing NFC tag geographic position authentication and related methods [Електронний ресурс]. – Режим доступу: https://patents.google.com/patent/US8831514B2/en

3. Wearable authentication: Trends and opportunities [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/305076814_Wearable_authentication_Trends_and_opportunities

4. ID Proof on the Go: Development of a Mobile EEG-Based Biometric Authentication System [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/abstract/document/6449282/

5. Off-line NFC Tag Authentication [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/abstract/document/6470914/

6. Mobile wireless communications device providing near field communication (nfc) unlock and tag data change features and related methods [Електронний ресурс]. – Режим доступу: https://patents.google.com/patent/US20140361872A1/en

7. Elliptic curve-based RFID/NFC authentication with temperature sensor input for relay attacks [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/science/article/pii/S0167923613002509.

8. Two-factor authentication through near field communication [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/7568941

9. Secure and Lightweight Authentication Protocol for NFC Tag Based Services [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/7153937

10. NFC Antenna Design for Low-Permeability Ferromagnetic Material [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/6698315

11. Automated antenna impedance adjustment for Near Field Communication (NFC) [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/6578295

12. Android – Wikipedia https://pl.wikipedia.org/wiki/Android dostęp Wrzesień [Електронний ресурс]. – Режим доступу: https://pl.wikipedia.org/wiki/Android

13. Android Studio [Електронний ресурс]. – Режим доступу: https://pl.wikipedia.org/wiki/Android_Studio

14. PhpStorm – Wikipedia https://en.wikipedia.org/wiki/PhpStorm dostęp Wrzesień [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/PhpStorm

14. Java [Електронний ресурс]. – Режим доступу: https://pl.wikipedia.org/wiki/Java,

15. PHP [Електронний ресурс]. – Режим доступу: https://pl.wikipedia.org/wiki/PHP,

16. Near Field Communication (NFC) Technology [Електронний ресурс]. – Режим доступу: https://resources.infosecinstitute.com/near-field-communication-nfc-technology-vulnerabilities-and-principal-attack-schema/

17. NFC Modulation and Coding-NRZ coding [Електронний ресурс]. – Режим доступу: https://www.rfwireless-world.com/Tutorials/NFC-Modulation-and-NFC-Coding.html

18. NFC Security [Електронний ресурс]. – Режим доступу: https://www.electronics-notes.com/articles/connectivity/nfc-near-field-communication/security.php

19. Єдина біометрична система: як працює цифрова автентифікація [Електронний ресурс]. – Режим доступу: https://www.anti-malware.ru/analytics/Technology_Analysis/single-biometric-system-how-it-works-digital-authentication

20. Xiaomi demos improved in-screen fingerprint reader; could debut on Xiaomi Mi 9 [Електронний ресурс]. – Режим доступу: https://www.bgr.in/news/xiaomi-mi-9-improved-in-screen-fingerprint-demo-mwc-2019-launch-samsung-galaxy-s10-761029/

21. 10 best phones with facial recognition: iPhone X [Електронний ресурс]. – Режим доступу: https://www.cnet.com/news/10-best-phones-with-facial-recognition-iphone-x-note-9-galaxy-s9-lg-g7

22. Amazon Cognito [Електронний ресурс]. – Режим доступу: https://aws.amazon.com/ru/cognito/

23. BankPekao [Електронний ресурс]. – Режим доступу: https://www.pekao.com.pl/klient-indywidualny/bankowosc-elektroniczna/serwis-internetowy-pekao24.html

24. «ПриватБанк» реалізував зняття готівки у банкоматах за QR-кодом [Електронний ресурс]. – Режим доступу: https://itc.ua/news/privatbank-realizoval-snyatie-nalichnyih-v-bankomatah-po-qr-kodu

25. IOS authorization methods [Електронний ресурс]. – Режим доступу: https://mtrqah.blogspot.com/2019/02/ios9101141-2.html

26. Design Pattern - Sign in / Sign up [Електронний ресурс]. – Режим доступу: https://www.pinterest.ru/carolineyee/design-pattern-sign-in-sign-up/

28. Automatic SMS Verification with SMS User Consent [Електронний ресурс]. – Режим доступу: https://www.theandroidsoul.com/how-to-automatically-unlock-your-android-phone-using-smart-lock-functions/

29. How to access your Schlage Sense™ Smart Deadbolt from anywhere [Електронний ресурс]. – Режим доступу: https://www.schlage.com/blog/categories/2017/10/schlage-sense-smart-deadbolt-remote-access.html

30. Використання Google Smart Lock на пристрої Android [Електронний ресурс]. – Режим доступу: https://solutics.ru/android/ispolzovanie-google-smart-lock-na-vashem-ustrojstve-android/