МІНІСТЕРСТОВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп‘ютерної та програмної інженерії
Кафедра комп’ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ А.С. Савченко
«___» _____ 20___ р

ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СПУПЕНЯ «БАКАЛАВР»

Тема: «Двофакторна система аутентифікації корпоративного середовища університету»

Виконавець: студентка УС-411 Горовая Наталія Миколаївна
(студент, група, прізвище, ім'я, по батькові)

Керівник: к. т. н., доцент Райчев Ігор Едуардович
(науковій ступень, вчене звання, прізвище, ім'я, по батькові)

Консультант: доцент Куклінський М.В._____
(П.І.Б.) (підпис)

Нормоконтролер: ст. викл. Шевченко О.П._____
(П.І.Б.) (підпис)

КИЇВ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп‘ютерної та програмної інженерії

Кафедра комп’ютерних інформаційних технологій

Освітній ступінь: Бакалавр

Спеціальність, спеціалізація: 12 “Інформаційні технології”, 122 “Комп’ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ
Завідувач кафедри

А.С. Савченко
“_____”_____ 2021 р.

ЗАВДАННЯ

на виконання дипломного проекту студента

Горовая Наталія Миколаївна

(прізвище, ім’я, по батькові)

1.Тема проекту: «Двофакторна система аутентифікації корпоративного середовища університету» затверджена наказом ректора № 636/ст. від 22.04.2021р.

2. Термін виконання роботи: з 11.05.2021 по 12.06.2021р.

3. Вихідні дані до роботи: розробка програми двофакторної аутентифікації за допомогою QR-коду.

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): вступ, аналітичний огляд і постановка завдання, розгляд завдання аутентифікації користувачів, дослідження технологій та засобів, розробка програмного продукту генерування QR-коду, оцінка якості технології, висновки.

5. Перелік обов’язкового графічного матеріалу: загальний перелік існуючих систем та обробка інформації створеним програмним продуктом. Використання структури проблематики питань аутентифікації.

КАЛЕНДАРНИЙ ПЛАН

	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Аналіз літератури та джерел за темою дипломного проекту.	11.05.2021р. – 12.05.2021р.	
2	Розробка та затвердження плану дипломного проекту.	13.05.2021р.	
3	Проведення консультації з науковим керівником щодо створення першого розділу.	14.05.2021р.	
4	Аналітичний огляд і постановка задачі.	15.05.2021р. – 18.05.2021р.	
5	Порівняльний аналіз існуючих способів аутентифікації.	19.05.2021р. – 22.05.2021р.	
6	Тестування варіантів заміни існуючих програм.	23.06.2021р. – 27.05.2021р.	
6	Створення програми для генерування QR-коду.	28.05.2021р. – 04.06.2021р.	
7	Висновки та оформлення пояснювальної записки дипломного проекту.	05.06.2021р. – 08.06.2021р.	
8	Підписання необхідних документів у встановленому порядку.	09.06.2021р. – 10.06.2021р.	
9	Підготовка до захисту та попередній захист дипломного проекту на випусковій кафедрі дипломного проекту	11.06.2021р. – 12.06.2021р.	

Студент **(*Горовая Н.М.*)**

Керівник дипломної роботи **(*Райчев І.Е.)***

РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Двофакторна система аутентифікації корпоративного середовища університету» містить: 57 сторінок, 7 рисунків, 3 таблиці, 19 літературних джерел.

Об’єкт дослідження: електронне корпоративне середовище університету.

Предмет дослідження: двофакторна система аутентифікації користувачів корпоративного середовища університету за допомогою QR-коду.

Мета роботи: покращення безпеки корпоративного середовища університету.

Методи дослідження, технічні та програмні засоби: розробка програмних бібліотек, порівняльний аналіз, обробка літературних джерел.

Отримані результати та їх новизна: запропоновано методи двофакторної аутентифікації, створено генератор QR-коду, запропоновано методи використання QR-коду в навчальному процесі.

АУТЕНТИФІКАЦІЯ, КОРПОРАТИВНЕ СЕРЕДОВИЩЕ, ДВОФАКТОРНА АУТЕНТИФІКАЦІЯ, QR-КОД.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

QR	–	Quick Response Code
HTTP	–	HyperText Transfer Protocol
IP	–	Internet Protocol
SMS	–	short message service
ПК	–	Персональний комп’ютер
NFC	–	Near field communication
AOL	–	America Online

ВСТУП

У сучасному інформаційному суспільстві з кожним днем стрімко збільшується рівень розвитку інформаційних і комунікаційних технологій. За останні роки їх інтенсивне використання і глобальне поширення, а також необмежений доступ суспільства до інтернету привели до багаторазового збільшення інформації обсягу інформації. У зв'язку з цим виникає необхідність подання інформації в компактному, простому у використанні, комфортному і візуально приємне для користувача вигляді. Це допоможе користувачеві серед величезного обсягу відомостей швидко і зручно знайти необхідну інформацію, витративши при цьому мінімум часу і зусиль.

Життя сучасної людини вже не можна уявити без різних пристроїв, таких як мобільний телефон або планшетний комп'ютер, вони стали невід'ємним атрибутом кожного. Тому виникла необхідність в новій формі подання інформації, яка буде відповідати великому списку вимог сучасного інформаційного суспільства. Так виникли QR-коди або «quick response», що перекладається як швидкий відгук. Такі матричні коди являють собою двовимірні штрих-коди, розшифровка яких проводиться в двох вимірах, по горизонталі та по вертикалі, що дозволяє «закодувати» великий обсяг інформації.

В даний час це один з найпопулярніших інструментів мобільного комерції. Спочатку QR-коди використовувалися тільки в промисловості, на сьогоднішній день вони активно застосовуються в споживчому середовищі (реклама, онлайн покупки), у фінансовій сфері та економіці (банківські термінали), в сфері авіа та залізничних перевезень (інформація з квитків), в освіті (формування розкладу), в культурній сфері (музеї), а також в медицині.

Таким чином, QR-код дозволяє будь-якій зацікавленій людині, наприклад, миттєво зайти на сайт, присвячений компанії, продукту, історичного об'єкту і отримати вичерпні дані.

У зв'язку з цим можна стверджувати, що створення програмного додатку, який буде зашифровувати інформацію і генерувати QR- код на сьогоднішній день є актуальною темою. Додаток, розроблений в ході дослідження, дозволить, наприклад,

простим користувачам не вводити довгі посилання на необхідний сайт вручну, а просто відсканувавши зображення, автоматично потрапити на web-сторінку, тобто скоротити обсяг виконаних користувачем дій, полегшити його працю і мінімізувати витрачений час, не вимагаючи від самого користувача спеціальних знань або навичок.

Об'єкт дослідження: аутентифікація за допомогою QR-коду.

Предмет дослідження: алгоритми генерації та використання QR-коду.

Мета: дослідити можливість двофакторної аутентифікації за допомогою QR-коду в корпорорativному середовищі університету.

Для реалізації мети були поставлені такі завдання:

1. проаналізувати наукову літературу в області QR-кодів з метою вивчення структури QR-кодів;

2. проаналізувати диніснуючі способи аутентифікації користувача;

3. розробити програму, що реалізує генерацію QR-коду;

4. запрпонувати можливі способи використання QR-коду в корпоративному середовищі цніверситету.

РОЗДІЛ 1
БЕЗПЕКА КОРПОРАТИВНОГО СЕРЕДОВИЩА

На сьогоднішній день, студенти та викладачі взаємодіють через корпоративне середовище університету у зв’язку з переходом на дистанційну форму навчання, але оскільки такий метод навчання є відносно новим, то є доцільним розглянути питання безпеки використання та поширення матеріалів у корпоративному середовищі університету.

1.1. Принципи роботи корпоративного середовища університету

У сучасних умовах основними чинниками успішної діяльності вищих навчальних закладів є, поряд з їх конкурентоспроможністю і рентабельністю, гнучкість, адаптивність і готовність до постійного розвитку. Корпоративна культура та корпоративне середовище є тією невід'ємною частиною інноваційного потенціалу, яка створює різного роду передумови для здійснення інноваційної діяльності. Ефективність організації вимагає, щоб корпоративна культура, її стратегія, оточення (зовнішнє середовище) і технологія (внутрішнє середовище) були приведені у відповідність. Корпоративне середовище (середовище взаємодії) є єдиним корпоративним простором, що визначає умови корпоративної діяльності, і забезпечує узгоджене інформаційну взаємодію учасників навчального процесу, а також взаємодія з зовнішніми інформаційними просторами. Корпоративне середовище вищого навчального закладу можна назвати новою управлінської реальністю, для якої характерні система зв'язків, дій, відносин і взаємин в організації, які здійснюються в рамках конкретної трудової діяльності

Якісне корпоративне середовище в трудовому колективі є умовою для ефективної корпоративної культури і менеджменту організації, спрямованого на підвищення ефективності роботи вищого навчального
закладу, виступає також як організаційний простір, в якому розвивається діяльність трудового колективу і прийняття управлінських рішень.

З теоретико-методологічної точки зору - це складна сукупність як явищ і відносин, так і умов, діяльності в рамках даної організації як системи. Корпоративне середовище – це те, що забезпечує взаємну відповідність частин у процесах побудови, функціонування та розвитку системи (організації) як цілісного утворення. Таким чином, корпоративне середовище стає відображенням корпоративної дійсності, під якою розуміється вся сукупність корпоративних відносин, що складаються між учасниками організаційно-технологічної взаємодії в організації[1].

Корпоративне середовище вищого навчального закладу є єдиним простором для всіх учасників освітнього процесу. Сучасне корпоративне середовище вищого навчального закладу є основною умовою для успішної реалізації місії і стратегії університету.

1.2. Ризики безпеки корпоративного середовища університету

Фахівці з інформаційної безпеки часто стикаються з питанням про посилення захисту того чи іншого сервісу. При цьому питання ідентифікації користувача з подальшим проходженням аутентифікації грає важливу роль. Сьогодні звичайний пароль не захистить від хакерів. Фішингові атаки або шкідливі програми на комп'ютері з легкістю скомпрометують облікові дані. Для посилення захисту акаунтів найчастіше застосовується двофакторна аутентифція, яка використовує поєднання звичного кодового слова з ще однією додатковою перевіркою. Однак нові технології, при всій їх незаперечній користі, привносять в діяльність корпоративного середовища і нові ризики, - наприклад, таким новим ризиком є несанкціонований доступ (НСД) до інформаційних систем та інформації, що належить організації. Також, в ряді випадків, відбувається трансформація старих ризиків в нові, іноді більш масштабні.

Несанкціонований доступ – це доступ до інформації з порушенням посадових повноважень співробітника, доступ до закритої для публічного доступу інформації з переліку осіб, які не мають дозволу на доступ до цієї інформації. Також несанкціонованим доступом в окремих випадках називають отримання доступу до інформації особами, які мають право на доступ до цієї інформації в обсязі, що перевищує необхідний для виконання службових обов'язків.

Доступ до корпоративного середовища університету мають лише ті, хто має корпоративну пошту, а саме, пошту з доменом @stud.nau.edu.ua(для студентів) та @npp.nau.edu.ua(для викладачів). Крім поштової адреси, на даний момент, жодного способу автентифікації в корпоративному середовищі університету не застосовується. Таким чином, доступ до корпоративного середовища зберігається у відрахованих студентів та випускників університету, оскільки поштова адреса та пароль залишаються у них і завдяки цьому, колишні студенти продовжують мати доступ до корпоративного середовища.

1.3. Аутентифікація користувача

Стосовно до сторін взаємодії аутентифікація означає перевірку однієї із сторін того, що сторона, яка взаємодіє - саме та, за яку вона себе видає. Часто аутентифікацію сторін називають також ідентифікацією.

Основним засобом для проведення ідентифікації є протоколи ідентифікації, що дозволяють здійснювати ідентифікацію кожної з сторін, які беруть участь у взаємодії. Розрізняють протоколи односторонньою і взаємної ідентифікації. Протокол це розподілений алгоритм, який визначає послідовність дій кожної з сторін. В процесі виконання протоколу ідентифікації кожна зі сторін не передає ніякої інформації про свій таємний ключ, а зберігає його у себе і використовує для формування відповідних повідомлень на запити, що надходять при виконанні протоколу.

1.3.1. Поняття аутентифікації

Нарешті, стосовно самої інформації аутентифікація означає перевірку того, що інформація, що передається по каналу, є справжньою за змістом, джерела, часу створення, часу пересилання і т.д.

Перевірка справжності змісту інформації зводиться, по суті, до перевірки її незмінності (з моменту створення) в процесі передачі або зберігання, тобто перевірці цілісності. Аутентифікація джерела даних означає підтвердження того, що вихідний документ був створений саме заявленим джерелом.

Зауважимо, що якщо сторони довіряють одна одній і мають загальний секретний ключ, то аутентифікацію сторін можна забезпечити застосуванням коду аутентифікації. Дійсно, кожне успішно декодоване одержувачем повідомлення може бути створено тільки відправником, так як тільки він знає їх загальний секретний ключ.

Для тих сторін, які не довіряють одна одній рішення подібних задач з використанням загального секретного ключа стає неможливим, тому при аутентифікації джерела даних потрібен механізм цифрового підпису.

В цілому, аутентифікація джерела даних виконує ту ж роль, що і протокол ідентифікації. Відмінність полягає лише в тому, що в першому випадку є деяка передана інформація, авторство якої потрібно встановити, а в другому потрібно просто встановити сторону, з якої здійснюється взаємодія.

Аутентифікація - це процес перевірки автентичності чого-небудь. Термін найчастіше використовується в середовищі інформаційних технологій. Прикладом аутентифікації може бути порівняння пароля, введенного користувачем, з паролем, який збережений в базі даних сервера. Подібна перевірка може бути як односторонньою, так і взаємної - все залежить від способу захисту і політики безпеки сервісу[2].

Методи аутентифікації поділяються залежно від типу ресурсу, структури і тонкощів організації мережі, віддаленості об'єкта і технології, яка використовується

в процесі розпізнавання. На підставі ступеня конфіденційності можна виділити декілька рівнів аутентифікації:

• доступ до інформації, витік якої не несе значущих наслідків для користувача і інтернет-ресурсу - в такий ситуації досить застосування багаторазового паролю;

• розкриття або пропажа даних приведуть до істотного збитку - необхідна більш сувора аутентифікація: одноразові паролі, додаткова перевірка при спробі доступу до решти розділів ресурсу;

• доступ до систем конфіденційних даних передбачає використання взаємної аутентифікації і багатофакторних методів повірки.

Всі методи аутентифікації можна розділити на односторонню (перевірка здійснюється тільки однією стороною) і взаємну (в перевірці даних беруть участь обидві сторони). Також виокремлюють однофакторний і криптографічний спосіб. Найбільш популярним прикладом застосування однофакторних систем є паролі. Залежно від рівня організації і ступеня конфіденційності даних, вони можуть бути багаторазовими (менш захищений варіант) і одноразовими.

Всі способи аутентифікації можна розташувати по зростанню їх складності.

Базова аутентифікація. При застосуванні цього виду аутентифікації логін користувача і його пароль входять до складу веб-запиту. Будь-який перехоплювач пакета інформації легко впізнає засекречені дані. Даний спосіб не рекомендується використовувати навіть в ситуаціях, коли засекречені дані не несуть суттєвої інформації ні для користувача, ні для інтернет-ресурсу. Це пов'язано з тим, що більшість людей використовують в мережі один і той же пароль для всіх сервісів, якими вони користуються. За даними Sophos (компанія-виробник засобів информацион-ної безпеки), 41% інтернет-користувачів застосовують одні й ті ж дані для реєстрації для різних платформах, будь то банківська сторінка або форум, присвячений їхньому улюбленому хобі.

Дайджест-аутентифікація. Вид аутентифікації, який означає передачу паролів користувача в хешованому стані. На перший погляд може здатися, що рівень захисту в даному випадку зовсім трохи відрізняється від базової перевірки. Насправді це не так: до кожного паролю додається довільний рядок, що складається з символів (хеш),

яка генерується окремо на кожен новий веб-запит. Постійне оновлення хешу не дає зловмиснику можливості розшифрувати пакет даних - кожне нове підключення утворює інше значення пароля. На основі даного методу аутентифікації працює більшість інтернет-браузерів (Mozilla, Google Chrome, Opera).

HTTPS. Цей протокол дає можливість шифрування не тільки логіна і пароля користувача, але і всіх інших даних, що передаються між інтернет-клієнтом і сервером. Використовується для введення особистої інформації:

• адреси;

• контактних даних;

• інформації про кредитну картки;

• банківських даних.

У протоколу є один істотний недолік - він значно уповільнює швидкість з'єднання.

Аутентифікація з пред'явленням цифрового сертифіката. Такий спосіб означає використання протоколів із запитом і відповіддю на нього. Сторінка аутентифікації направляє до користувача певний набір символів («адресу»). Відповіддю є запит сервера, який підписаний за допомогою персонального ключа. Аутентифікація по відкритому ключу застосовується в якості захисного механізму в наступних протоколах:

• SSL;

• Kerberos;

• RADIUS.

Аутентифікація з використанням Cookies. Cookie - невеликий масив даних, який відправляється інтернет-сервером і зберігається на ПК користувача. Браузер при кожній спробі підключення до даного ресурсу посилає Cookies як одну із складових частин HTTP-запиту. Дана технологія, крім аутентифікації, використовується для:

• збереження індивідуальних налаштувань і переваг;

• спостереження за станом сеансу;

• збору статистичних даних про користувачів (частота відвідувань, унікальні відвідування і т.д.).

Як засіб аутентифікації Cookies використовуються для систем безпеки чатів, форумів і різних інтернет-ігор. Cookies мають низький ступінь захисту - якщо сесія погано фільтрується, то викрасти їх не складає труднощів. Тому застосовується додатково прив'язка за IP-адресою, з якого користувач увійшов в систему.

Децентралізована аутентифікація. Виділяють кілька основних протоколів, що працюють за принципом децентралізованої аутентифікації:

• OpenID. Протокол дозволяє завести один пароль для декількох інтернет-ресурсів. Безпека здійснюється за рахунок цифрового підпису повідомлень на основі алгоритму Діффі-Хеллмана. Недоліками є вразливість перед фішинговими атаками і атакою «людина посередині». На основі OpenID зараз працюють такі всесвітньо відомі компанії як Google, Yandex, BBC, PayPal, Microsoft та інші.

• OpenAuth. Працює за схожим алгоритмом з OpenID. Дозволяє використовувати сервіси AOL і будь-які інші, надбудовані поверх них. При цьому у користувача не виникає необхідності створювати новий обліковий запис на кожному сайті. Інформація про сесії зберігається не в cookies, а самі cookies аутентифікації відмежовані специфікованим доменом.

• OAuth. Дає можливість одному веб-ресурсу отримати доступ до призначених для користувача даних на іншому веб-ресурсі. Застосовується в системі Twitter і в сервісах Apple.

Відстеження процесу аутентифікації користувачем. Безпека призначених для користувача даних багато в чому залежить від поведінки самого користувача. Багато веб-ресурси відстежуючи-ють підозрілу активність і повідомляють про це власника облікового запису. Наприклад, Google фіксує IP-адреси, з яких здійснювався вхід в систему і надає користувачу можливості наведені нижче:

• переключитися на передачу інформації тільки через HTTPS;

• включити функцію відстеження підозрілих сесій: в даному випадку Google буде надсилати вам повідомлення про активність облікового запису в підозрілий час, велику кількість вихідного спаму, видалення старих повідомлень і т.д .;

• відстежувати списків третіх сторін, які мають доступ до тих же сервісів Google, що і користувач.

Включивши функцію аудиту сеансів користувача, ви отримуєте доступ до такої інформації:

• час входу і тривалість сеансу;

• Ім'я користувача;

• вид сеансу (з використанням реєстрації або без нього);

• успішність в аутентифікації або неможливість здійснити перевірку;

• точка, з якої виконувався вхід в систему.

Багато веб-сервіси дозволяють відслідковувати процес аутентифції за допомогою якогось одного значення, наприклад, IP-адреси (НЕ-об'єктивно, якщо у вас динамічний IP).

Багатофакторна аутентифікація. Багатофакторна аутентифікація має на увазі пред'явлення більш ніж одного «доказу» способу аутентифікації для доступу до даних. Такими «доказами» можуть бути:

• певне знання - інформація, якою володіє користувач (пін-код, пароль);

• володіння - предмет, який є у суб'єкта (флеш-пам'ять, електронний пропуск, магнітна банківська картка, токен та ін.);

• властивість - якість, властиве виключно суб'єкту - сюди відносять дані біометрії і персональні відзнаки: форма обличчя, індивідуальні особливості райдужної і сітчастої оболоуги ока, відбитків пальців та ін.

1.3.2. Двофакторна аутентифікація

Однією з різновидів багатофакторної аутентифікації є двофакторна (також називається двоетапною або подвійною). Такий спосіб означає перевірку даних користувача на підставі двох відмінних один від одного компонентів.

Прикладом двоетапної аутентифікації є сервіси від Google і Microsoft. При спробі авторизації з нового пристрою, крім логіна і пароля, необхідно також ввести код, що складається з шести (Google) або восьми (Microsoft) знаків. Отримати його можна одним з перерахованих способів:

• SMS-повідомлення на мобільний телефон;

• голосовий виклик на телефон;

• реєстр одноразових кодів;

• програма-аутентифікатор для мобільного або ПК.

Основними перевагами подвійною аутентифікації є зручність (смартфон завжди під рукою) і безпеку (постійна зміна коду підтвердження). Даний метод також має деякі недоліки. Проблеми з мобільною мережею можуть стати перешкодою для отримання коду підтвердження, а саме смс-повідомлення може бути перехоплено зловмисниками. Існує також деяка затримка при отриманні SMS - вона пов'язана з процедурою перевірки автентичності.

Двофакторна аутентифікація використовується сервісами Facebook, Web Money, Yandex, Microsoft, Google і іншими. Всі вони використовують свої власні програми-аутентифікатори, кожна з яких підпорядковується певним стандартам.

1.4. Постановка задачі

Під інформаційною безпекою розуміється захищеність інформації та підтримує інфраструктури від випадкових або навмисних впливів природного або штучного характеру, що можуть призвести нанесенням шкоди власникам або користувачам інформації і підтримуючої інфраструктури.

На практиці найважливішими є три аспекти інформаційної безпеки:

- доступність (можливість за розумний час отримати необхідну інформаційну послугу);
- цілісність (актуальність і несуперечність інформації, її захищеність від руйнування і несанкціонованого зміни);
- конфіденційність (захист від несанкціонованого ознайомлення).

Формування режиму інформаційної безпеки - проблема комплексна. Заходи для її рішення можна поділити на чотири рівні:

➢ законодавчий (закони, нормативні акти, стандарти і т.п.);

адміністративний (дії загального характеру, що починаються керівництвом організації);

- процедурний (конкретні міри безпеки, що мають справу з людьми);
- програмно-технічний (конкретні технічні заходи).

При формуванні режиму інформаційної безпеки слід враховувати сучасний стан інформаційних технологій. Майже всі організації чекають від інформаційних систем в першу чергу корисною функціональності. Комп'ютерні системи купуються не заради захисту даних; навпаки, захист даних будується заради вигідного використання комп'ютерних систем. Для отримання корисної функціональності природно звернутися до найбільш сучасним рішенням в області інформаційних технологій. Значить, кажучи про захист, слід мати на увазі перш за все сучасні апаратні та програмні платформи.

Висновки до розділу 1

Отож, основним напрямком розвитку ринку двофакторної аутентифікації залишаються ризики несанкціонованого доступу до даних. Двофакторну автентифікацію дозволяють застосовувати для входу в операційні системи Windows і Linux, в хмарні сервіси та в корпоративне середовище. За допомогою двофакторної аутентифікації можна нейтралізувати досить велику частку загроз, і вибір на користь посиленою аутентіфікації виправдовує себе і є зручним та досить простим способ в застосуванні для організації доступу до корпоративних ресурсів, до порталів і хмарних сервісів.

РОЗДІЛ 2
СПОСОБИ ЗДІЙСНЕННЯ ДВОФАКТОРНОЇ АУТЕНТИФІКАЦІЇ

2.1. Огляд способів двофаторної аутентифікації

На даний момент існує багато різноманітних способів посвідчення особи з допомогою двофакторної аутентифікації:

- Одноразові паролі на папері (роздруковується набір кодів);
- Відправка тимчасового коду на адресу електронної пошти;
- Відправка одноразового пароля по SMS;
- OTP-токени (апаратні генератори одноразових паролів);
- Додатки для двофакторної аутентифікації (додатки класу Authenticator).

2.1.1. Друковані паролі

Одноразові паролі на папері, як правило, використовують онлайн-банки. Можна, наприклад, отримати в офісі кредитної організації картку зі покриттям, яке стирається, поверх полів з кодами або просто роздрукувати набір паролей в банкоматі, а система дистанційного обслуговування (ДБО) при аутентифікації на сайті буде просити ввести одноразовий набір символів під певним номером. Відправка пароля по електронній пошті - це максимально спрощений варіант захисту. Для аутентифікації користувача потрібно знати всього лише його адреса (який дуже часто використовується в якості логіна). Відповідно, рівень захисту таким методом - нижче, ніж в іншими способами, особливо якщо використовується один пароль для входу в пошту і на цільовий ресурс

2.1.2. Одноразові коди

Ще одним способом двофакторної аутентифікації є відправка одноразового пароля по SMS.

Однак коди в SMS-повідомленнях - теж не дуже надійний варіант. По-перше, такий пароль можна підглянути в повідомленні на екрані блокування смартфона. По-друге, повідомлення може бути перехоплено шкідливою програмою. Поширений також вид шахрайства, званий «SIM swapping»: шляхом обману або змови в салоні стільникового зв'язку можна отримати нову SIM-карту з потрібним номером, і SMS-повідомлення будуть приходити на неї, а телефон жертви навіть не зможе підключитися до мережі. Є й інші способи компрометації одноразових паролів по SMS. Більш надійним є апаратний OTP-токен (генератор одноразових паролів). Він являє собою пристрій у вигляді брелка з дисплеєм і кнопкою. У пам'яті пристрою заздалегідь програмується певна кількість паролів. При натисканні на кнопку коди відображаються на дисплеї.

2.1.3. Додатки для двофакторної аутентифікації

У нинішній час на тлі масовості смартфонів під управлінням операційних систем Android і iOS популярність стали набирати спеціалізовані програми для двофакторної аутентифікації, особливо поширеними стали додатки, які використовують біометричні дані.

Ідентифікація за відбитками пальців - найпоширеніша біометрична технологія аутентіфікаціі користувачів. Метод використовує унікальність малюнка папілярних візерунків на пальцях людей. Відбиток, отриманий за допомогою сканера, перетворюється в цифровий код, а потім порівнюється з раніше введеними наборами еталонів. Переваги використання автентифікації за відбитками пальців - легкість у використанні, зручність і надійність. Універсальність цієї технології дозволяє застосовувати її в будь-яких сферах і для вирішення будь-яких і найрізноманітніших завдань, де необхідна достовірна і досить точна ідентифікація користувачів.

Для отримання відомостей про відбитки пальців застосовуються спеціальні сканери. Щоб отримати чітке електронне подання відбитків пальців, використовують досить специфічні методи, так як відбиток пальця занадто малий, і дуже важко отримати добре помітні папілярні візерунки.

Зазвичай застосовуються три основні типи сканерів відбитків пальців: ємнісні, прокатні, оптичні. Найпоширеніші і широко використовувані це оптичні сканери, але вони мають один серйозний недолік. Оптичні сканери нестійкі до муляжів і мертвим пальцях, а це значить, що вони не настільки ефективні, як інші типи сканерів. Так само в деяких джерелах сканери відбитка пальців ділять на 3 класи за їхніми фізичними принципам: оптичні, кремнієві, ультразвукові.

Аутентифікація по геометрії обличчя - це досить поширений спосіб ідентифікації. Технічна реалізація представляє собою складну математичну задачу. Широке застосування мультимедійних технологій, за допомогою яких можна побачити достатню кількість відеокамер на вокзалах, аеропортах, площах, вулицях, дорогах і інших місцях скупчення людей, стало вирішальним у розвитку цього напрямку. Для побудови тривимірної моделі людського обличчя, виділяють контури очей, брів, губ, носа, і інших різних елементів особи, потім обчислюють відстань між ними, і за допомогою нього будують тривимірну модель.

Щоб визначення унікального шаблону, що відповіднає певній людині, потрібно від 12 до 40 характерних елементів. Шаблон повинен враховувати безліч варіацій зображення на випадки повороту особи, нахилу, зміни освітленості, зміни виразу. Діапазон таких варіантів варіюється в залежності від цілей застосування даного способу (для ідентифікації, віддаленого пошуку на великих територіях і т. Д.). Деякі алгоритми дозволяють компенсувати наявність у людини окулярів, капелюхів, вусів і бороди.

Також можна здійснювати двофакторну аутентифікацію за допомогою QR-коду, але цей вид аутентифыкації ми розглянемо далі більш детально.

В цілому двофакторна аутентифікація − це технологія контролю доступу в два етапи, коли крім введення логіна і пароля до аккаунту користувача просять підтвердити свою особистість додатковим способом, наприклад ввести одноразовий

пароль, отриманий по електронній пошті, в SMS-повідомленні на мобіль -ний телефон, за допомогою генератора (апаратного або програмного) і т. п. Введення додаткового рівня безпеки забезпечує більш ефективний захист від несанкціонованого доступу.

2.2. Застосування QR-коду для двофакторної аутентифікації

Можна, звичайно, використовувати відбитки пальців, USB-токени, одноразові паролі, NFC-теги, але для всього цього потрібна модифікація серверної частини або апаратні модифікації клієнта та ін. В разі зчитування QR-коду прямо в поля введення сервера отримує все ті ж логін і пароль. Камера і сучасний браузер є на більшості сучасних ноутбуків, планшетів, ПК. де це можна використовувати.

2.2.1. Історія створення QR-коду

QR код або Quick Response Code, - код швидкого реагування - увійшов в сучасний побут в кінці 2000-х років. QR-сканер є практично в кожному мобільному: товари, послуги, локації, веб-адреси.

Японія здавна славиться різного роду винаходами, покликаними зробити бізнес-процеси і виробничі завдання якомога більш компактними та ефективними. На початку 1990-х років перед інженерами великої машинобудівної компанії Denso стояла нетривіальне завдання: створити уніфікований штрих-код (або щось подібне) для маркування деталей та компонентного сканування. На той момент всередині компанії були прийняті більше 10 кодів різного призначення, і співробітники заводу скаржилися на те, що робота з кодами вимагає великої концентрації, а самі коди містять надзвичайно мало корисної інформації. При цьому не можна сказати, що проблема ця носила приватний характер: по всій країні компанії розробляли власні варіанти штрих-кодів, намагаючись вкласти в них якомога більше даних. Масахиро Хара, співробітник відділу розробки Denso Wave, в 1992 році взявся за вирішення цього завдання[3]. Нові коди повинні були відповідати наступним вимогам:

• обсяг інформації, яку можливо зберігати в коді, повинен істотно зрости;

• процес зчитування повинен бути якомога більш точним і швидким;

• самі коди повинні бути стійкі до плям від масла, бруду та інших пошкоджень;

• пристрій, що зчитує має бути простим і дешевим.

За легендою, натхнення прийшло до Масахиро за обідом, під час гри в шахи. Що ж, дуже схоже на правду: QR-код дійсно нагадує дошку для гри з розставленими на ній камінчиками, а ігрова ситуація - це ті ж закодовані дані.

Тепер дані могли кодуватися не тільки по горизонталі, але і по вертикалі. Їх обсяг зріс до 7 000 знаків, включаючи не тільки латиницю, але і ієрогліфи. Сам код сканувався під будь-яким нахилом завдяки квадратах в трьох кутах коду, які функціонують як детектори положення.

За словами Масахиро, «квадратики», з яких складається код, також були вибрані не випадково: з'ясувалося, що патерни з квадратів практично не зустрічаються в бізнес-документах і в маркуванні в цілому. Таким чином, ймовірність помилки при зчитуванні, викликаної «паразитними» даними, зводиться до нуля. З метою додаткової страховки Масахиро запропонував використовувати певний розмір відступів між інформаційної частиною патерну і його межами. Кінцевий формат коду визначається наступним співвідношенням габаритів чорних і білих пікселів: 1: 1: 3: 1: 1. Пристрій, що зчитує здатне розпізнати код, розташований під будь-яким кутом, орієнтуючись лише по цьому співвідношенню. До 1994 року новий формат коду (Quick Response Code) був повсюдно впроваджений на заводах виробничого ланцюжка автоконцерну Toyota, але швидко «перетік» з цехів в інші бізнес-сфери. Масахиро Хара згадує, що аж до презентації нового формату коду він не був упевнений, що його дітище приживеться в компанії. Так, швидкість зчитування даних і надійність формату не викликали сумнівів, проте 2D-сканери могли стати серйозною перешкодою на шляху впровадження технології. Проте код був сприйнятий і главами, і рядовими співробітниками корпорації дуже тепло. Протягом наступного місяця вдалося успішно впровадити його у власну Kanban-програму Toyota.

Простота і зручність коду сподобалися людям і за межами Японії. До середини-кінця 2000-х років про японському винахід дізнався весь світ. При цьому фраза-

позначення «QR code» є зареєстрованим товарним знаком, що належить компанії-винахіднику, проте використання самих кодів не обкладається відрахуваннями.

Говорити про версії QR-кодів можна досить довго: існують і «маленькі» версії (21x21px), і більші - 177x177px. Ось основні прийняті в усьому світі кодування даних:

• цифрова кодування, до 7089 цифр;

• алфавітно-цифрова кодування, до 4296 символів (або до 2953 символів з підтримкою кирилиці);

• байтовая кодування, до 2953 байт;

• кодування «кандзі», до 1817 ієрогліфів.

Крім того, існують і більш «екзотичні», практично не прижилися формати: наприклад, для опису графічної і аудіоінформації.

Крім того, була розроблена функція виправлення помилок для зчитувальних пристроїв. Якщо частина зображення була пошкоджена, система могла прочитати і обробити всю закодовану інформацію.

Код складається з чорних квадратів, розташованих у квадратної сітці і вміщує в себе три кілобайти двійкового коду.

У двотисячних QR-код став експериментальним способом оплати, але спочатку операція займала занадто багато часу - сканер зчитував інформацію цілих 20 секунд. У 2003 році китайська компанія Inspiry винайшла швидкий механізм зчитування QR-кодів. Пізніше вона ж випустила перший портативний сканер, а технологія стала по-справжньому популярною.

Вибухове зростання технології припав на період масової появи планшетів, комунікаторів і смартфонів. Виробники гаджетів навчили камери розпізнавати QR-коди, і технологія стала поширюватися в усьому світі.

У Китаї технологія стала популярною разом з появою месенджера WeChat. У кожного користувача соцмережі є унікальний QR-код, а в додаток вбудована програма-сканер. Власники аккаунта використовують QR-коди для розміщення в рекламі, пошуку інформації, обміну контактами, авторизації та реєстрації в сервісах, підключення до Wi-Fi в публічних місцях - майже для будь-яких дій.

Внутрішня платіжна система соцмережі WeChat Pay - один з найпопулярніших фінансових інструментів в Китаї, де всі транзакції відбуваються за допомогою QR. Через систему користувачі здійснюють платежі і перекази на суму близько 2 млрд доларів щорічно. Це приблизно третина всіх мобільних платежів в країні.

WeChat став потужним інструментом маркетингу і монетизації для компаній: у кожного бренду в соцмережі з'явилася своя публічна сторінка з індивідуальним QR-кодом. Жителі швидко звикли до технології всередині середовища месенджера - тепер QR-код в країні використовують всюди[4].

Starbucks однією з перших помітила перспективну технологію і розробила на її основі програму лояльності. Компанія розмістила QR-коди в журналах, буклетах і на рекламних щитах. Користувачі могли відсканувати код і отримати посилання. Вона вела на сайт компанії, де можна було дізнатися про найближчі кав'ярнях, оцінити кави, отримати знижку або подивитися відео про типи обсмаження кави.

У США одним з піонерів напрямку став PayPal: в 2015 році він придбав сервіс, який допомагав здійснювати оплату за допомогою QR-кодів. Трохи пізніше співтовариство найбільших підприємств роздрібної торгівлі США - Walmart, Best Buy, Kmart і 7-Eleven і інші - розробили власну систему платежів за допомогою QR-кодів Current C5.

Технологію активно розвивали і західні соцмережі. Наприклад, Instagram запустив сервіс Nametag - картки, схожі на QR-код, що працюють за тим же принципом. Власник будь-якого профілю міг створити таку картку для швидкого переходу на сторінку.

Потім QR почали використовувати і в роздрібній торгівлі. Наприклад, компанії UGG і Sennheiser розробили можливість перевірки оригінальності товару по QR-коду, який розміщували всередині упаковки. При скануванні покупець отримував інформацію про покупку і визначав справжність товару. Якщо товар підроблений, то покупець міг повернути його в магазин.

2.3.2. Еволюція QR-коду

Примітка: QR-коди нижче представлені виключно для демонстрації технології. Ми не несемо відповідальності за їх зміст - сканувати на свій страх і ризик.

Гра з кольором. Сучасні смартфони і зчитувальні пристрої куди менше залежать від чіткості зображення коду, ніж їх попередники. Завдяки цьому стало можливо розфарбовувати QR-коди в різні кольори і інтегрувати їх в дизайн продуктів.

Рис. 2.1. Кольоровий QR

Гра з формою. «Закруглені» QR-коди, коди, в які вписані зображення і логотипи - також не новина. Щоб пристрої могли розібрати нестандартні з точки зору форми коди, застосовується технологія надлишкового кодування.

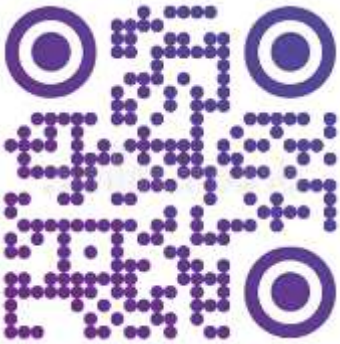

Рис. 2.2. «Круглий» QR

Окремо тут стоять «дизайнерські» варіанти QR-кодів: це не просто вписування квадратика із зображенням в поле коду, а цілий витвір мистецтва.

Рис. 2.3. «Дизайнерський» QR

Коди з нестандартною орієнтацією в просторі також не нові.

Рис. 2.4. Нестандартно розташовані QR

Крім того, періодично можна зустріти анімовані QR-коди, однак це, скоріше, модна дивина, ніж виправдане використання технології.

Непряме застосування QR-коду. Листівки, футболки, біжутерія з пам'ятними шифровками - навіть сюди дісталися QR-коди.

Рис. 2.5. Підвіска у вигляді QR-коду

Відкриті виставки і вуличні бібліотеки використовують QR-коди: найчастіше в них шифруються основні дані про художників, письменників і музикантів, а за скороченою посиланням можна завантажити копію того чи іншого твору.

Рітейл не залишився в стороні: в QR-кодах шифруються коди купонів і номери карт для програм лояльності. Втім, тут у QR-коду є сильні конкуренти, наприклад, NFC-рішення.

Рис. 2.6. Картка-меню в QR-коді

QR-коди все ще популярні серед рекламодавців: у кодах шифруються адреси сайтів і посилання на скачування додатків - своєрідна спроба перевести користувачів з оффлайну в онлайн.

2.3.3. Порівняння QR-коду зі штрих-кодом

QR-код - це прямий спадкоємець штрих-коду. Ось тільки в основу другого лягла технологія азбуки Морзе, що використовувалася для автоматизації різного товару і техніки. І десятиліттями штрих-код був єдиним нормальним варіантом маркування. Звичні смуги і цифри вже давно стали загальноприйнятим явищем для будь-якого сучасника. Однак можливості штрих-коду обмежені.

Ключова відмінність QR-коду від традиційного штрихкоду - він розпізнається сканером як двовимірне зображення. Для нормалізації зображення при зчитуванні і зниження ймовірності помилки код містить кілька великих квадратів в одному з кутів, а також безліч дрібніших синхронізуючих точок, розосереджених по всій площі коду. Кумедний момент: специфікація QR-коду описує тільки сам принцип побудови коду, але не формат даних, зашифрованих в ньому. Це створює ціле поле для експериментів, які не закінчуються до цього дня.

Лінійний код може вмістити в себе від 20-ти до 30-ти символів, чого часом недостатньо. Японські фахівці поставили перед собою мету - розширити можливості штрих-коду, але з класичним підходом це було неможливо. І на арені з'являються двомірні (матричні) коди, серед яких головним по праву став QR-код.

Традиційні штрих-коди, які ми часто скануємо в супермаркетах, щоб дізнатися ціну продукту, обмежені тільки 20 буквено-цифровими символами по горизонталі. Вони грунтуються на американському стандарті відстеження товарів UPC (Universal Product Code = універсальний код товару), розробленому в 1973 році.

Його недоліки: якщо штрих-код пошкоджений, інформація недоступна. І він не може шифрувати ієрогліфи.

На відміну від звичайного штрих-коду QR-код має низку позитивних якостей:

• Збільшення обсягу закодованої інформації в кілька разів;

• Інформація не дублюється символами, зрозумілими людині;

• На вибір є кілька варіантів виконання.

Фактично складно назвати QR-код чимось концептуально новим. Все-таки технологія вкрай близька до класичного штрих-коду. Однак різниця є. QR-код - це свого роду сполучна ланка між реальністю і віртуальним світом, як би дивно це не звучало. Можливості, які відкрили QR-коди, дійсно набагато ширше, ніж були раніше. Будь-який сучасний телефон або планшет може без проблем зчитувати інформацію з QR-коду за частки секунди. І інформація ця може бути найрізноманітнішою: дані про продукцію, посилання на офіційний сайт, зашифрований код, який бере участь в акції, і навіть коротка розповідь. Зашифрувати можна практично все, і користувач зможе без особливих проблем вважати дану інформацію, використовуючи свій кишеньковий гаджет.

2.3.4. Алгоритм генерування QR-коду

Щоб генерувати QR-коди потрібно розуміти, як вони працюють.. Принцип використання QR-кодів полягає в тому, що роздрукований або намальований код поміщається на об'єкт, після чого він може бути зчитаний і розшифрований за допомогою пристрою, який має функціонуючу камеру і встановлене програмне забезпечення, яке декодує сам QR-код.

Рис. 2.7. Приклад QR-коду

Назва QR-код розшифровується з англійської мови як Quick Response code, тобто код швидкого відгуку. Цей стандарт кодів прийшов на зміну звичайного штрих-коду, який отримав величезну популярність завдяки чудовим функціональним характеристикам, точністю інформації, що міститься в ньому і швидкості її зчитування. Але головним недоліком звичайного штрих-коду в порівнянні з QR-кодом є невеликий допустимий обсяг збережених даних, а так само обмеження на типи даних, які можуть зберігатися в штрих-коді.

Інформація в QR-коді розташовується в двох напрямках - горизонтально і вертикально. Завдяки такому розташуванню даних в QR-коді, він здатний зберігати у багато разів більше інформації, ніж його попередники, включаючи різні типи даних: цифри, букви, ієрогліфи, символи і т.д. Максимальній обсяг інформації різних типів даних, який поміщається в один QR-код, представлено в таблиці 1:

Таблиця 1

Типи даних і максимальний можливий обсяг в QR-кодах

Тип даних	**Максимальний обсяг (символ)**	**Можливі символи**
Числові дані	7089	9,8,7,6,5,4,3,2,1,0
Символьні дані	4296	A–Z, $, %, *, +, -, ., /, space, :
Бінарна інформація, байт	2953	JIS X 0201
Ієрогліфи	1817	JIS-X/0208

Ще однією перевагою QR-коду є його здатність відновлювати інформацію, що міститься в ньому, навіть якщо певна частина символів на зображенні QR-коду було пошкоджено або не розпізнано. Це стало можливим завдяки системі корекції помилок на базі кодів Ріда-Соломона.

Максимальна кількість кодових слів, яке може бути відновлено, становить до 30%. У відповідності зі специфікацією QR-код має 4 ступеня корекції помилок: L -

7%, M - 15%, Q - 25%, H - 30%. Чим вище ступінь корекції помилок, тим менше даних можна зашифрувати і помістити в QR-код.

Сам QR-код складається з певного набору міток і безпосередньо пікселів, які представляють собою закодоване повідомлення, збережене в QR-коді.

На будь-якому QR-коді обов'язково повинні бути присутніми наступні види міток (представлені на рисунку 8):

1. Позиціонування. Область необхідна для детектування коду;

2. Номер версії. Визначає яка версія коду використовується (від 1 до 40);

3. Синхронізація. Дублюється в двох напрямках, і дозволяють знизити ймовірність виникнення помилок при зчитуванні, системної інформації (наприклад, версія, тип даних і т.п.);

4. Формат. Необхідні для визначення типи даних закодованих в коді.

7. Вирівнювання. Використовуються для кращого позиціонування коду під час обробки (при версії QR-коду вище 1);

8. Рівень виправлення помилки. Дозволяють визначити, який рівень захищеності від перешкод був використаний на етапі кодування для правильного вибору способу виявлення можливих помилок в коді.

Рис. 8 Мітки та дані на QR-коді

У всіх існуючих програмах, які зчитують і декодують QR-коди, реалізований простий алгоритм виявлення QR-коду на зображенні, отриманому з камери. Потім реалізована стандартна процедура декодування інформації з QR-коду. Однак цей алгоритм розпізнавання вимагає дуже чіткого позиціонування спеціально виділеної області на пристрої, за допомогою якого відбувається розпізнання і визначеного розташування QR-коду в просторі. Після того, як спеціально визначена область на пристрої чітко збіглася з границями QR-коду, відбувається пошук трьох міток позиціонування, про які йшлося раніше.

Ці мітки розташовані строго в певних місцях виділеної області зображення. Недолік такого підходу полягає в тому, що QR-код може перебувати в будь-якій області на зображенні, і користувачеві необхідно самому попередньо фокусуватися на необхідному QR-коді, і стежити за тим, щоб область для зйомки QR-коду збігалася з самим QR-кодом, який потрібно розкодувати.

Програми для двофакторної аутентифікації влаштовані дуже просто. Ось що треба зробити користувачу: встановити на смартфон сам додаток, зайти в налаштування безпеки сервісу, який в числі варіантів двофакторної аутентифікаціі пропонує використовувати такі програми, і вибрати відповідну опцію, відсканувати QR-код, який відобразиться в сервісі, за допомогою додатку для двофакторної аутентифікації. Після цього програма починає періодично (наприклад, кожні 30 секунд) створювати новий одноразовий код. Паролі формуються на основі ключа, який відомий тільки їй і серверу, а також поточного часу, округленого до 30 секунд. Оскільки обидві складові однакові і у клієнта, і у сервісу, коди генеруються синхронно. Даний алгоритм називається OATH TOTP (Time-based One-Time Password), і в переважній більшості випадків використовується саме він.

QR код - це монохромна картинка, на якій деякі пристрої (наприклад смартфон зі спеціальним додатком) розпізнають текст. Цим текстом може бути не тільки проста фраза, але і, хоч це і не входить в офіційну специфікацію, посилання, номер телефону або візитна картка. Такі коди найчастіше використовують, щоб закодувати посилання і роздрукувати її на плакаті або візитці.

Процес генерації QR коду ділиться на кілька чітких кроків:

1. Кодування даних;

2. Додавання службової інформації та заповнення;

3. Поділ інформації на блоки;

4. Створення байтів корекції;

5. Об'єднання блоків;

6. Розміщення інформації на QR коді.

Кодування даних. QR код підтримує кілька способів кодування даних, в залежності від того, які символи використовуються: цифрове, буквено-цифрове, кандзі (китайсько-японські ієрогліфи) і побайтово кодування. Цифрове кодування має на увазі використання тільки цифр від 0 до 9, буквено цифрове - прописні букви латинського алфавіту, цифри і символи $% * + -. /: І пробіл. Спочатку треба створити порожню послідовність біт, яка далі буде заповнюватися.

Цифрове кодування. Цей тип кодування вимагає 10 біт на 3 символи. Вся послідовність символів розбивається на групи по 3 цифри, і кожна група (тризначне число) переводиться в 10-бітове двійкове число і додається до послідовності біт. Якщо загальна кількість символів не кратне 3, то якщо в кінці залишається 2 символи, отримане двозначне число кодується 7 битами, а якщо 1 символ, то 4 бітами. Наприклад, є рядок «12345678», який треба закодувати. Ми розбиваємо її на числа: 123, 456 і 78, потім переводимо кожне з них в двійковий вигляд: 0001111011, 0111001000 і 1001110, і об'єднуємо це в один потік: 000111101101110010001001110.

Буквено-цифрове кодування. У цьому випадку на 2 символи потрібно 11 біт інформації. Вхідний потік символів розділяється на групи по 2, в групі кожен символ кодується згідно таблиці внизу, значення першого символу в групі множиться на 45 і додається до значення другого символу. Отримане число переводиться в 11-бітове двійкове число і додається до послідовності біт. Якщо в останній групі 1 символ, то його значення відразу кодується 6-бітовим числом і додається до послідовності біт.

Таблиця 2

Значення символів в буквено-цифровому кодуванні.

Значення	Символ	Значення	Символ	Значення	Символ	Значення	Символ
0	0	12	C	24	O	36	Пробіл
1	1	13	D	25	P	37	$
2	2	14	E	26	Q	38	%
3	3	15	F	27	R	39	*
4	4	16	G	28	S	40	+
5	5	17	H	29	T	41	-
6	6	18	I	30	U	42	.
7	7	19	J	31	V	43	/
8	8	20	K	32	W	44	:
9	9	21	L	33	X		
10	A	22	M	34	Y		
11	B	23	N	35	Z		

Наприклад, рядок «HELLO» кодується таким чином. Розбиваємо на групи: HE, LL, O; знаходимо відповідне значення символам в кожній групі: (17, 14), (21, 21), (24); знаходимо значення для кожної групи: 17 * 45 + 14 = 779, 21 * 45 + 21 = 966, 24 = 24; переводимо кожне значення в двійковий вигляд: 779 = 01100001011, 966 = 01111000110, 24 = 011 000; і об'єднуємо все це в одну послідовність біт: +011000010110111100011001l000.

Побайтове кодування. Це універсальний спосіб кодування, яким можна закодувати будь-які символи. Єдиним недоліком методу є відносно низька щільність інформації. В цьому випадку текст кодується в будь-якому кодуванні (рекомендований в UTF-8) і отримана сукупність електронних даних береться в незмінному вигляді.

Додавання службової інформації. На цьому етапі треба визначитися з рівнем корекції: чим вищий цей рівень, тим вище допустимий рівень пошкодження зображення і тим менше інформації при рівному розмірі. Всього є 4 рівня корекції: L (допустимо максимум 7% пошкоджень), M (15%), Q (25%) і H (30%). Найчастіше використовується рівень M.

Ще одна властивість QR коду - його версія (чим вона більша, тим більше розмір). Всього існує 40 версій. Номер версії залежить від кількості інформації і від рівня корекції.

Додавання службових полів. До цього моменту вже повинен бути обраний рівень корекції і визначена версія. Тепер треба перед послідовністю біт, отриманої в попередньому пункті, додати на початку два поля: спосіб кодування і кількість даних. Спосіб кодування - поле довжиною 4 біта, яке має наступні значення: 0001 для цифрового кодування, 0010 для буквено-цифрового і 0100 для побайтового. Кількість даних - це кількість кодованих символів, а для побайтового - кількість байт (а не біт в отриманої послідовності), представлене у вигляді довічного числа певної довжини (визначається за таблицею 2).

Таблиця 3

Довжина поля кількості даних.

	Версія 1-9	Версія 10-26	Версія 27-40
Цифрове	10 біт	12 біт	14 біт
Буквенно-цифрове	9 біт	11 біт	13 біт
Побайтове	8 біт	16 біт	16 біт

Наприклад, дана рядок довжиною 100 байт, закодована побайтово, рівень корекції - M. Довжина послідовності біт цього рядка - 800 біт. Скориставшись таблицею 2 можна визначити, що оптимальніше за все буде використовувати 6-у версію. Довжина поля, що визначає кількість даних в нашому випадку - 8 біт (таблиця

3). Поле, що спосіб кодування має вигляд 0100, поле кількості даних - 01100100 (100 в двійковому вигляді). У підсумку вийде послідовність біт 010001100100 – вихідна послідовність.

Якщо довжина отриманої послідовності біт виявилася більше допустимої для обраної версії, то версію треба збільшити на одну і виконати додавання службових полів заново.

Специфікація допускає використання змішаного кодування. Це означає, що кілька груп даних можна закодувати різними способами і об'єднати їх в одну послідовність. Це робиться в такий спосіб: <спосіб кодування даних 1> <кількість даних 1> <дані 1> <спосіб кодування даних 2> <кількість даних 2> <дані 2> і так далі.

Заповнення. На даному етапі є послідовність біт даних, кількість біт в якій напевно не кратно 8. Треба доповнити її нулями так, щоб її довжина стала кратна 8. Тепер нашу послідовність біт можна розбити на групи по 8 біт і представити у вигляді послідовності байт. Якщо кількість біт в поточній послідовності байт менше того, яке потрібно для обраної версії, то її треба доповнити.

Приклад. Є послідовність: послідовність біт, довжина якої кратна 8 10101011101; доповнюємо її нулями, щоб її довжина стала кратна 8: послідовність біт, довжина якої кратна 8 10101011101 00000; тепер припустимо, що її довжина - 104 біта, а для обраної версії необхідно 128 біт, тоді для заповнення потрібно додати 24 «заповнюють» біта (3 байта): послідовність біт, довжина якої кратна 8 10101011101 00000 11101100 00010001 11101100.

Поділ інформації на блоки. Сукупність електронних даних, отримана на попередньому етапі, (далі дані) поділяється на обределённое для версії і рівня корекції кількість блоків. Якщо кількість блоків дорівнює одному, то цей етап можна пропустити.

Визначення кількість байт в кожному блоці. Для цього треба розділити всю кількість байт (можна визначити кількість байт в даних або розділити число з таблиці 2 на вісім) на кількість блоків даних. Якщо це число не ціле, то треба визначити залишок від ділення. Цей залишок визначає скільки блоків з усіх доповнені (такі

блоки, кількість байт в яких більше на один ніж в інших). Всупереч сподіванням, доповненими блоками повинні бути не перші блоки, а останні.

Наприклад, для версії 9 і рівня корекції M кількості даних - 182 байта, кількість блоків - 5. ділячи кількість байт даних на кількість блоків, отримуємо 36 байт і 2 байта в залишку. Це означає, що блоки даних матимуть такі розміри: 36, 36, 36, 37, 37 (байт). Якби залишку не було, що всі 5 блоків мали б розмір 36 байт.

Заповнення блоків. Блок заповнюється байтами з даних повністю. Коли поточний блок повністю заповнюється, черга переходить до наступного. Байтів даних повинно вистачити рівно на всі блоки, ні більше і ні менше.

Створення байтів корекції. Наступний алгоритм застосовується до кожного блоку даних (якщо блок даних один, то просто до даних). Цей алгоритм заснований на алгоритмі Ріда-Соломона. Перше що треба зробити – визначити скільки байтів корекції треба створити. За кількістю байтів корекції визначається так званий генеруючий многочлен. Многочленом він називається, тому що оригінальний метод використовує многочлен з тими ж коефіцієнтами.

Перед виконанням циклу треба підготувати масив, довжина якого дорівнює максимуму з кількості байтів в поточному блоці і кількості байтів корекції, і заповнити його початок байтами з поточного блоку, а кінець нулями.

Цикл, описаний в цьому списку, повторюється стільки раз, скільки байтів даних міститься в поточному блоці.

1. Беремо перший елемент масиву, зберігаємо його значення у змінній A і видаляємо його з масиву (всі наступні значення зсуваються на одну клітинку вліво, останній елемент заповнюється нулем).

2. Якщо A дорівнює нулю, то пропустити наступні дії (до кінця списку) і перейти до наступної ітерації циклу.

3. Знаходимо відповідне числу A число в таблиці 8, заносимо його в змінну Б.

4. Далі для N перших елементів, де N - кількість байтів корекції, i - лічильник циклу:

• До i-му значенню генеруючого многочлена треба додати значення Б і записати цю суму в змінну В (сам многочлен не зраджувати).

• Якщо B більше 254, треба використовувати її залишок при діленні на 255 (саме 255, а не 256).

• Знайти відповідне У значення в таблиці 7 і зробити опеацію побітового складання по модулю 2 (XOR, у багатьох мовах програмування оператор ^) з і-м значенням підготовленого масиву і записати отримане значення в і-ю осередок підготовленого масиву.

Перші N байтів підготовленого масиву після цього циклу - і є байти корекції. Для кожного блоку даних вийде відповідний блок байтів корекції.

Об'єднання блоків. Є кілька блоків даних і стільки ж блоків байтів корекції, їх треба об'єднати в один потік байт. Робиться це в такий спосіб: з кожного блоку даних по черзі береться один байт інформації, коли черга доходить до останнього блоку, з нього береться байт і черга переходить до першого блоку. Так триває до тих пір, поки в кожному блоці не закінчаться байти. Якщо в поточному блоці вже немає байт, то він пропускається (таке відбувається, коли звичайні блоки вже порожні, а в доповнених ще є по одному байту). Аналогічним чином треба зробити з блоками байтів корекції. Вони беруться в тому ж порядку, що і відповідні блоки даних.

У результаті повинно вийти щось подібне: <1-й байт 1-го блоку даних> <1-й байт 2-го блоку даних> ... <1-й байт n-го блоку даних> <2-й байт 1-го блоку даних> ... <(m - 1) -й байт 1-го блоку даних> ... <(m - 1) -й байт n-го блоку даних> <m-й байт k-го блоку даних> ... <m-й байт n-го блоку даних> <1-й байт 1-го блоку байтів корекції> <1-й байт 2-го блоку байтів корекції> ... <1-й байт n -го блоку байтів корекції> <2-й байт 1-го блоку байтів корекції> ... <l-й байт 1-го блоку байтів корекції> ... <l-й байт n-го блоку байтів корекції>. Тут n - кількість блоків даних, m - кількість байтів на блок даних у звичайних блоків, l - кількість байтів корекції, k - кількість блоків даних мінус кількість доповнених блоків даних (тих, у яких на 1 байт більше).

2.3.5. Перспективи аутентифікації за допомогою QR-коду

Використання QR-коду є зручним способом аутентифікації оскільки:

- є безліч додатків, що зберігають паролі в базі даних. Це добре, але ці програми

показують пароль у вигляді тексту і потрібно ввести його руками. Можна відображати збережений пароль у вигляді QR-коду

- QR-код можна роздрукувати на папірці, пластику і прикріпити його як брелок.

- QR-код можна генерувати при реєстрації де-небудь і зберегти його, як резервний спосіб входу

- можна генерувати досить складні одноразові паролі і кодувати їх в QR.

Висновки до розділу 2

Отже, в наш час QR-коди можна побачити практично скрізь, тому що багато компаній і користувачів вже змогли переконатися в їх ефективності та універсальності. Легко пізнавані чорно-білі квадрати допомагають людям вирішити різні завдання в багатьох сферах життєдіяльності.

Працюючи над темою, було виявлено тенденцію збільшення використання QR-кодів в освіті, запропонувано можливі способи практичного використання двовимірних цифрових кодів в навчальному процесі і в прикладній діяльності освітньої установи.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ДВОФАКТОРНОЇ АУТЕНТИФІКАЦІЇ ЗА ДОПОМОГОЮ QR-КОДУ

3.1. Програмна реалізація генератора QR-коду

Перед тим як братися за розробку будь-якої програми, важливо знати, які саме функції виконуватиме цю програму. Для того щоб чітко визначити всі функції майбутньої розробки, потрібно побудувати функціональну модель. Головною особливістю таких моделей є те, що за допомогою блоксхем можна змоделювати додаток, визначити всі завдання, оптимізувати додаток і головне розподілити тимчасові рамки розробки програми. У функціональних моделях відображаються всі елементи проекту та зв'язку між ними [17]. Функціональне моделювання використовує графічний мова опису процесів, тому будь-яка функціональна модель буде виглядати як сукупність упорядкованих діаграм. Для опису функціональних моделей в основному використовуються дві методології:

• діаграма потоків даних (DFD);

• метод функціонального моделювання (IDEF0).

В ході дослідження була обрана динамічна бібліотека для генерації QR-кодів messagingtoolkit-qrcode. Для опису функціональної моделі алгоритму QR-коду використана діаграма функціонального моделювання IDEF0(Рис.9)

Рис. 3.1. Функціональна модель бібліотеки

На основі обраної бібліотеки необхідно розробити додаток для генерації QR-кодів. Нижче наведена структура розробленого додатка, для її моделювання була використана методологія функціонального моделювання IDEF0.

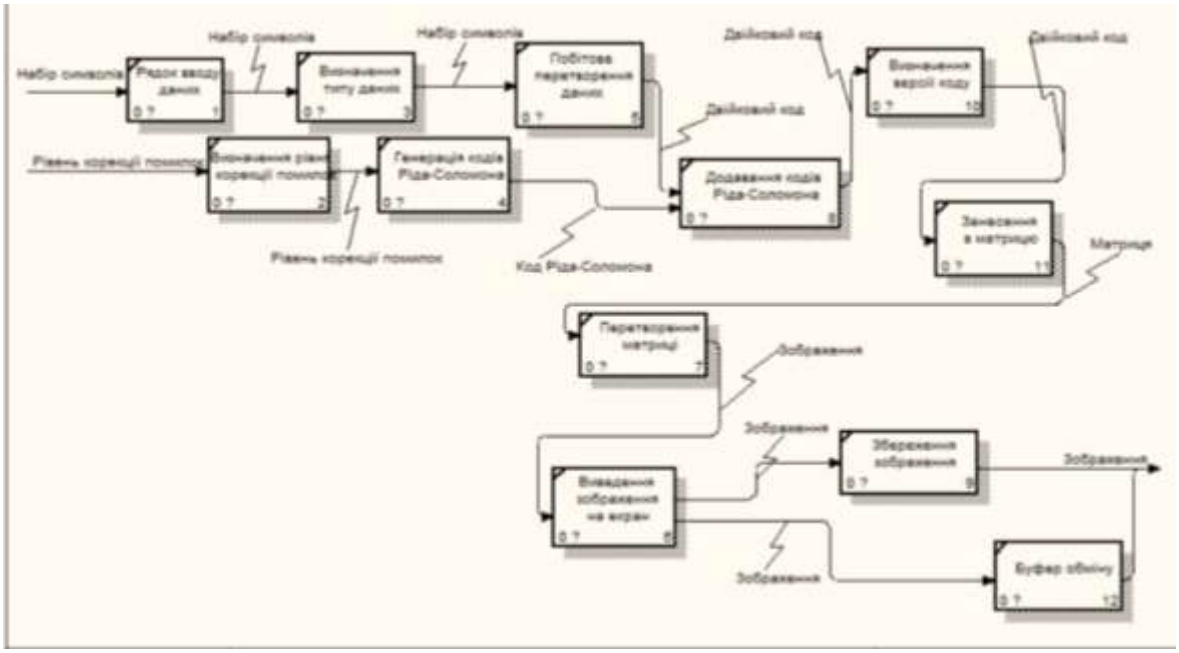

Рис. 3.2 Функціональна модель додатку

Розробка програми в середовищі програмування Visual Studio 2008:

Для додатку генерації QR-коду нам потрібно:

• Графічний елемент для відображення QR-коду (pictureBox);

• Кнопки «Згенерувати», «Зберегти», «Скопіювати», «Завантажити», «Розпізнати» (button);

• Бібліотека для генерації QR-коду (messagingtoolkit-qrcod).

Для початку потрібно прикріпити бібліотеку до проекту, це робиться в браузері рішень. Далі потрібно її форматувати, для цього потрібно перейти в редактор коду і в розділі «*Using*» пишемо рядки:

```
using MessagingToolkit.QRCode.Codec;
using MessagingToolkit.QRCode.Codec.Data;
```

Далі переміщаємо компоненти і розставляємо їх на свої місця.

Для роботи з динамічної бібліотекою будуть використовуватися команди:

• згенерувати;

• розкодувати.

Оброблювач кнопки «Згенерувати» виглядає наступним чином [18]:

```
 // якщо рядок введення даних не порожній
 if (textBox1.Text! = "")
{// створюємо змінну qrtext
// і присвоюємо їй рядок введення даних
 string qrtext = textBox1.Text ;
// створення об'єкта encoder класу QRCodeEncoder
 QRCodeEncoder encoder = new QRCodeEncoder ();
// кодируем qrtext в змінну qrcode класу Bitmap
Bitmap qrcode = encoder.Encode (qrtext);
 // відправляємо змінну qrcode в pictureBox1.
 pictureBox1.Image = qrcode as Image;
}
else
{// якщо рядок введення даних порожня то
// виводимо системне повідомлення про помилку
MessageBox.Show ( "Можливо, ви не заповнили поле для кодування.", "Помилка");
```

```
}
```

А обробник кнопки «Розкодувати» виглядає так [2]:

```
// якщо поле виводу QR-коду не порожнє то
 if (pictureBox1.Image! = null)
{// Створюємо об'єкт decoder класу QRCodeDecoder
QRCodeDecoder decoder = new QRCodeDecoder ();
// виводимо системне повідомлення з розшифрованих кодом
// в повідомленні розшифровуємо decoder з pictureBox
MessageBox.Show (decoder.decode (new
QRCodeBitmapImage (pictureBox1.Image as Bitmap)), ("Виведення"));
 // відправляємо розшифроване повідомлення в буфер обміну
Clipboard.SetData (DataFormats.Text, decoder.decode (new
QRCodeBitmapImage (pictureBox1.Image as Bitmap)));
 }
else
{// якщо pictureBox порожній, то виводимо повідомлення про помилку
 MessageBox.Show ( "Можливо, нічого розшифровувати.",
"Помилка");
}
```

Кнопки «Зберегти» і «Завантажити» використовують стандартні діалогові команди:

• SaveFileDialog

• OpenFileDialog

Кнопка "Копіювати код» має більш складну структуру [2]:

```
// знаходимо pictureBox щодо екрану
 Rectangle rect = pictureBox1.ClientRectangle;
```

```
// створюємо зображення потрібних розмірів
Bitmap bmp = new Bitmap (rect.Width, rect.Height);
// перемальовували вміст pictureBox на Bitmap
pictureBox1.DrawToBitmap (bmp, rect);
// задаємо розміри області, що копіюється
Rectangle copyRect = new Rectangle (0, 0, 180, 180);
// створюємо зображення за розміром заданої області
Bitmap bmpCopy = new Bitmap (copyRect.Width, copyRect.Height);
// отримуємо об'єкт Graphics
using (Graphics g = Graphics.FromImage (bmpCopy))
// перемальовували область з початкової картинки
g.DrawImage (bmp, 0, 0, copyRect, GraphicsUnit.Pixel);
// копіюємо в буфер обміну виділену область
Clipboard.SetImage (bmpCopy);
```

У ході дослідження було поставлено завдання розробити додаток для генерації QR-кодів на основі динамічної бібліотеки messagingtoolkitqrcode. Додаток розроблявся на об'єктно-орієнтованій мові програмування Visual C# в середовищі програмування Visual Studio 2008.

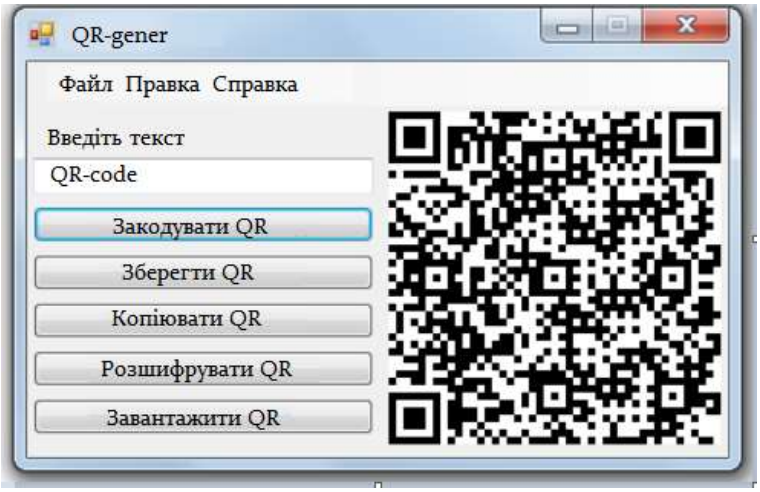

Рис. 3.3. Додаток генерації QR-коду

Розроблений додаток підтримує числове і буквено-числове кодування. Також при такому кодуванні можливо закодувати URL-адресу, номер телефону, e-mail адресу, візитку. Додаток ставить найбільший рівень корекції помилок, що забезпечує найбільшу безпеку даних в QR-коді. Так само додаток саме визначає версію за допомогою бібліотеки.

У додатку є можливість управління QR-кодом за допомогою кнопок:

• Створення QR-коду;

• Збереження QR-коду;

• Копіювання QR-коду;

• Розшифровки QR-коду;

• Завантаження QR-коду;

Для генерації QR-коду досить ввести будь-яке повідомлення (на латиниці або цифрами) в поле введення тексту і натиснути «Закодувати QR-код». Якщо поле введення інформації пусте, то програма виведе повідомлення про помилку, а точніше, вона каже що поле повідомлення порожнє.

Для збереження закодованого QR-коду необхідно натиснути на кнопку «Зберегти QR-код». Після натискання на кнопку відкриється діалогове вікно, де потрібно буде вибрати місце, куди зберегти згенерований QR-код і ввести ім'я для зображення QR-коду.

Якщо створений QR-код необхідно скопіювати в інший додаток, то можна натиснути на кнопку «Скопіювати QR-код», після чого перейти в іншу програму, куди потрібно скопіювати код і вставити скопійоване зображення за допомогою клавіш «Ctrl» + «V» або за допомогою спеціальної команди в додатку-одержувачі. QR-код не вставиться в додаток, якщо програма не приймає графічну інформацію.

Якщо потрібно розкодувати інформацію з QR-коду, то спочатку потрібно натиснути на кнопку «Завантажити QR-код», після чого відкриється діалогове вікно 35 для вибору зображення QR-коду. Коли QR-код завантажиться в додаток потрібно натиснути на кнопку «Розшифрувати QR-код» і додаток виведе повідомлення з розкодованої з QR-коду інформацією і додасть скопійовану інформацію в буфер

обміну, після чого цю інформацію можна вставити в будь-який додаток, що приймає текстову інформацію (наприклад web-оглядач).

Якщо поле виведення QR-коду пусте, то додаток виведе повідомлення про помилку. Для більшої зручності, команди з головної форми додатка продубльовані в пункті «Файл». Якщо потрібно очистити поле введення даних, то потрібно натиснути спершу кнопку «Правка», а потім кнопку «Очистити поле введення».

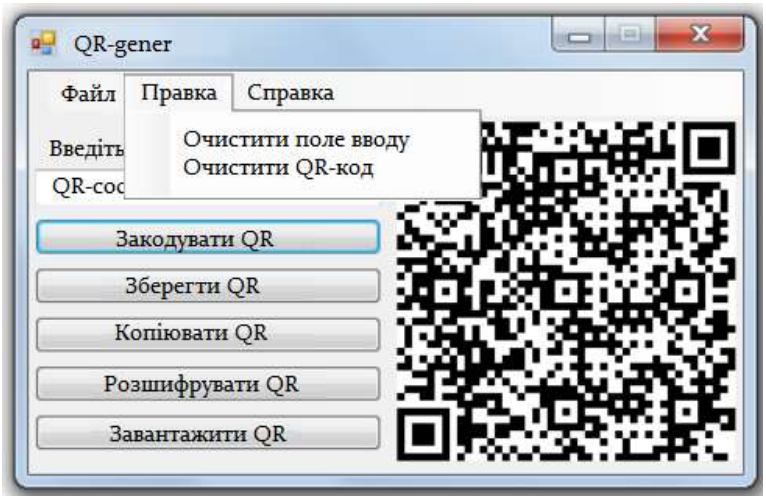

Рис. 3.4. Додаток для генерації QR-коду

Аналогічно можна очистити поле виведення QR-коду: «Правка» / «Очистити QR-код».

3.2. Способи використання QR-коду в навчальному процесі

Таким чином, обробивши теоретичну інформацію про безпеку корпоративного середовища, зрозумівши принципи роботи QR-коду та створивши програмний продукт для його генерації можна запропонувати наступні шляхи впровадження та використання QR-коду в корпоративному середовищі університу:

- Можна згенерувати та присвоїти для кожного студента його QR-код, який може бути аналогом залікової книжки та містити інформацію про успішність студента;

- Можливо також використання QR-коду для отримання доступу до занять в середовищі Classroom, що допоможе захистити дані, які знаходяться на платформі від доступу сторонніх осіб;
- Окремо можна виділити можливість надання інтерактивної інформації студентам у вигляді QR-коду, а також для надання доступу до лекційних матеріалів.

Висновки до розділу 3

На основі обраної динамічної бібліотеки було створено додаток, що дозволяє генерувати QR-код і розпізнавати його з завантаженої картинки. У додатку реалізовано два способи виведення згенерованого QR-коду: зберегти як картинку і скопіювати в буферобміну. Додаток розроблено в середовищі Visual Studio.

Середовище розробки було обрано в силу універсальності додатків, які розробляються на ній. В якості мови програмування, на якій було розроблено додаток генерації QR-коду, було обрано Visual C #, тому, що ця мова набирає все більшої популярності серед розробників.

ВИСНОВКИ

У проведеному дослідженні була проаналізована наукова та методична література в області безпеки корпоративного середовища та QR-кодів. З проведеного аналізу літератури були зроблені висновки про принципи роботи та ризики безпеки корпоративного середовища, та про структуру QR-коду, про його матриці і способи його кодування. Матриця QR-коду, його структура, а так само всі її складові частини були детально вивчені. Таким чином, обробивши теоретичну інформацію про безпеку корпоративного середовища, зрозумівши принципи роботи QR-коду та створивши програмний продукт для його генерації було запропоновано наступні шляхи впровадження та використання QR-коду в корпоративному середовищі університу: згенерувати та присвоїти для кожного студента його QR-код, який може бути аналогом залікової книжки та містити інформацію про успішність студента; використання QR-коду для отримання доступу до занять в середовищі Classroom, що допоможе захистити дані, які знаходяться на платформі від доступу сторонніх осіб. можливість надання інтерактивної інформації студентам у вигляді QR-коду, а також для надання доступу до лекційних матеріалів.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pidru4niki − [Електронний ресурс] Режим доступу: https://pidru4niki.com (дата звернення 19.05.2021р.) − Назва з екрану.

2. Msdn.microsoft − [Електронний ресурс] Режим доступу: https://msdn.microsoft.com (дата звернення 19.05.2021р.) − Назва з екрану.

3. PHP QR Code − [Електронний ресурс] Режим доступу: http://phpqrcode.sourceforge.net (дата звернення 19.05.2021р.) − Назва з екрану.

4. Qrcc − [Електронний ресурс] Режим доступу: http://qrcc.ru/generator.php (дата звернення 19.05.2021р.) − Назва з екрану.

5. Qrcode − [Електронний ресурс] Режим доступу: http://qrcode.com.ua (дата звернення 19.05.2021р.) − Назва з екрану.

6. Qrcode.kaywa − [Електронний ресурс] Режим доступу: http://qrcode.kaywa.com (дата звернення 20.05.2021р.) − Назва з екрану.

7. Qrcoder − [Електронний ресурс] Режим доступу: http://qrcoder.ru (дата звернення 20.05.2021р.) − Назва з екрану.

8. Qr-coder − [Електронний ресурс] Режим доступу: http://qr-coder.ru (дата звернення 20.05.2021р.) − Назва з екрану.

9. Qreambee − [Електронний ресурс] Режим доступу: http://qreambee.ru (дата звернення 21.05.2021р.) − Назва з екрану.

10. QREncode − [Електронний ресурс] Режим доступу: https://packages.altlinux.org/ru/Sisyphus/srpms/qrencode (дата звернення 21.05.2021р.) − Назва з екрану.

11. QR-код. Добавление служебной информации − [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/QR-код (дата звернення 22.05.2021р.) − Назва з екрану.

12. QR-код. Этап размещения информации на поле кода − [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/QR-код(дата звернення 22.05.2021р.) − Назва з екрану.

13. QR-код. Кодирование − [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/QR-код (дата звернення 22.05.2021р.) − Назва з екрану.

14. Messaging made easy − [Електронний ресурс] Режим доступу: http://platform.twit88.com (дата звернення 22.05.2021р.) − Назва з екрану.

15. ZXing (Zebra Crossing) − [Електронний ресурс] Режим доступу: https://github.com/zxing/zxing/wiki/Getting-Started-Developing (дата звернення 22.05.2021р.) − Назва з екрану.

16. ГОСТ Р ИСО/МЭК 18004-2015 Спецификация символики штрихового кода QR-code − Москва: Стандартинформ, 2015. − 113 с.

17. МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ IDEF0 / Москва: ГОССТАНДАРТ РОССИИ, 2000. 75 с. – Електрон. аналог друк. вид.: режим доступу: https://nsu.ru/smk/files/idef.pdf (дата звернення 22.05.2021р.) − Назва з екрану.

18. О.М. Котов Язык C#. Краткое описание и введение в технологии программирования. − Екатеринбург: Издательство Уральского университета, 2014. − 208 с.

19. Таблица Unicode − [Електронний ресурс] Режим доступу: https://geektimes.ru/post/256932/ (дата звернення 22.05.2021р.) – Назва з екрану.

ДОДАТКИ

Додаток А

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MessagingToolkit.QRCode.Codec;
using MessagingToolkit.QRCode.Codec.Data;
namespace QR_Cod
{
public partial class Form1 : Form
{
public Form1()
{ InitializeComponent(); }
String result = System.String.Empty;
private void button1_Click(object sender, EventArgs e)
{
if (textBox1.Text != "")
{
string qrtext = textBox1.Text;
//зчитуємо текст з TextBox'y
QRCodeEncoder encoder = new QRCodeEncoder();
//створюємо об’єкт класу QRCodeEncoder
Bitmap qrcode = encoder.Encode(qrtext); 44
/* кодуємо слово, отримане з TextBox'y (qrtext) в
```

```
змінну qrcode. класа Bitmap(клас, який використовується для
роботи з зображеннями)*/
pictureBox1.Image = qrcode as Image;
// pictureBox виводить qrcode як зображення.
}
else
{
MessageBox.Show("Можливо, ви не заповнили поле для
кодування.", "Помилка");
}
}
private void button2_Click(object sender, EventArgs e)
{
SaveFileDialog save = new SaveFileDialog();
/*save буде запитувати у користувача місце, де він хоче зберегти файл. */
save.Filter = "PNG|*.png|JPEG|*.jpg|GIF|*.gif|BMP|*.bmp";
/*створюємо фільтр, який визначає, в яких форматах ми
можемо зберегти нашу інформацію. В данному випадку обираємо
формати зображень. Записується так:
"назва_формату|*.розширення_формату")*/
if (save.ShowDialog() == System.Windows.Forms.DialogResult.OK)
//якщо користувач натискає кнопку "Зберегти"
{
pictureBox1.Image.Save(save.FileName);
/*зображення з pictureBox'а зберігається під іменем,
яке введе користувач*/
}
}
```

```
private void button3_Click(object sender, EventArgs e)
{
OpenFileDialog load = new OpenFileDialog();
/*load буде запитувати у користувача місце, з якого він
хоче завантажити файл.*/
if (load.ShowDialog() == System.Windows.Forms.DialogResult.OK)
//якщо користувач натискає кнопку "Відкрити".
{
pictureBox1.ImageLocation = load.FileName;
/*в pictureBox'і відкривається файл, який був за адресою,
що вказав користувач.*/
}
}
private void button4_Click(object sender, EventArgs e)
{
if (pictureBox1.Image != null)
{
QRCodeDecoder decoder = new QRCodeDecoder();
// створюємо "розкодоване зображення"
MessageBox.Show(decoder.decode(new
QRCodeBitmapImage(pictureBox1.Image as Bitmap)));
/*в MessageBox'і программа запише розкодоване
повідомлення з зображення, яке попередньо буде
переведено з pictureBox'у в клас Bitmap, щоб ми мали змогу працювати з цим
зображенням.*/
Clipboard.SetData (DataFormats.Text, decoder.decode(new
QRCodeBitmapImage(pictureBox1.Image as Bitmap)));
}
```

```
else
{
MessageBox.Show("Можливо, нічого розшифровувати",
"Помилка");
}
}
private void button5_Click(object sender, EventArgs e)
{
// Область PictureBox
Rectangle rect = pictureBox1.ClientRectangle;
// Створюємо зображення потрібного розміру
Bitmap bmp = new Bitmap(rect.Width, rect.Height);
// Малюємо вміст PictureBox на Bitmap
pictureBox1.DrawToBitmap(bmp, rect);
// Задаємо розмір області, яку будемо копіювати
Rectangle copyRect = new Rectangle(0, 0, 180, 180);
// Створюємо зображення за розміром обраної області
Bitmap bmpCopy = new Bitmap(copyRect.Width, copyRect.Height);
// Получаем объект Graphics
using (Graphics g = Graphics.FromImage(bmpCopy))
// Перемальовуємо область з обраного зображення
// bmp – звідки копіюємо
// 0, 0 - координати лівого верхнього кута на новому зображенні
// copyRect - область из которой срисовываем
// Одиниці вимірювання для перемальовки
g.DrawImage(bmp, 0, 0, copyRect, GraphicsUnit.Pixel);
// Кладемо в буфер обміну "вирізаний" шматок
Clipboard.SetImage(bmpCopy);
```

```
}
private void очиститиПолеВводуToolStripMenuItem_Click(object sender,
EventArgs e)
{
textBox1.Text = "";
}
private void очиститиQRкодToolStripMenuItem_Click(object sender,
EventArgs e)
{
pictureBox1.Image = null;
}
}
}
```