ШИФР «VAS»

«ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДВОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ ПРОГРАМНОГО ЗАСОБУ НА ОСНОВІ ГОЛОСУ»

ЗМІСТ

ВСТУП

Актуальність даного дослідження полягає в тому, що сьогодні основним способом персоніфікації користувача є вказівка його мережевого імені та пароля. Небезпеки, пов'язані з використанням пароля, добре відомі: паролі забувають, зберігають в невідповідному місці, нарешті, вони можуть бути втрачені. Як повідомляють групи інформаційних технологій багатьох компаній, більша частина дзвінків в службу підтримки пов'язана із забутими або такими, що втратили силу паролями [1]. Окрім звичайних символьних паролів, використовується біометричний метод автентифікації, що полягає в ідентифікації людини за унікальними, властивими тільки їй біологічними ознаками.

Впровадження систем автентифікації в життя суспільства є незаперечним фактом. Актуальність розвитку технологій ідентифікації особи обумовлена збільшенням числа об’єктів і потоків інформації, які необхідно захищати від несанкціонованого доступу, а саме: криміналістика, системи контролю доступу, системи ідентифікації особи, інформаційна безпека, тощо [2].

При впровадженні масових систем двофакторної автентифікації сучасні сфери діяльності матимуть змогу надавати суспільству можливість використання належного захисту та створювати безпечні умови від порушень зловмисників.

Метою роботи є підвищення захищеності програмного засобу від несанкціонованого доступу шляхом двофакторної автентифікації на основі голосу.

Для досягнення поставленої мети необхідно виконати наступні завдання:

– проаналізувати методи та засоби автентифікації програмних додатків;
– розробити архітектуру системи двофакторної автентифікації;
– розробити алгоритми роботи модулів системи;
– створити програмний засіб на основі автентифікації особи за голосом;
– виконати тестування програмного засобу.

Об’єктом дослідження є процеси автентифікації програмних засобів.

Предметом дослідження є методи та засоби автентифікації програмного засобу.

Методи дослідження. Для реалізації поставлених завдань були використано методи аналізу та порівняння для дослідження систем автентифікації та, зокрема, голосової, метод синтезу для розробки структури інформаційної технології, методи проектування програмного забезпечення для програмної реалізації інформаційної технології, методи розпізнавання для голосової ідентифікації відбитка користувача, метод оцінки ефективності роботи біометричних систем.

Наукова новизна. Запропоновано інформаційну технологію автентифікації програмного додатку, яка полягає у використанні символьного паролю у якості першого фактору та голосового відбитку - якості другого, що відрізняється застосуванням інтелектуальних методів розпізнавання на базі хмарних технологій. Це дозволяє забезпечити високу точність ідентифікації та підвищення захищеності спеціалізованого прикладного програмного застосунку.

Практична цінність. Розроблено програмний засіб для автентифікації на основі голосу, який ефективно забезпечує захист від несанкціонованого доступу до прикладного додатку за рахунок використання двох факторів автентифікації.

Розроблене програмне забезпечення впроваджено на підприємстві з розробки програмного забезпечення ТОВ «РОДЖМАКС ДІДЖИТАЛ».

За результатами науково-дослідної роботи оформлено та отримано авторське свідоцтво на твір «Комп’ютерна програма. Програмний засіб для автентифікації на основі голосу «VAS».

Результати роботи доповідалися на міжнародній науковій конференції «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення» та опубліковані у збірнику тез (випуск 39, 11 червня 2019 року) та Всеукраїнській НПК Молодь в науці: дослідження, проблеми, перспективи (Вінниця, 2020). Результати доповідей опубліковані у збірниках матеріалів конференцій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Види, способи та механізм автентифікації

Автентифікація – шлях встановлення вірогідності інформації, пред'явленої користувачем у разі звернення його до системи та відкриття йому доступу, якщо він має на це право; електронний процес, який дає змогу підтвердити електронну ідентифікацію фізичної, юридичної особи, інформаційної, телекомунікаційної, інформаційно-телекомунікаційної системи, а також походження та цілісність електронних даних [1].

Головним механізмом та одним із способів автентифікації в інформаційній системі являється попередня ідентифікація на основі користувацького ідентифікатора – «логіна» і пароля – певної конфіденційної інформації, знання якої передбачає володіння певним ресурсом в мережі.

Отримавши введений користувачем логін і пароль, комп’ютер порівнює їх зі значенням, яке зберігається в спеціальній захищеній базі даних і, у випадку успішної автентифікації проводить авторизацію з подальшим допуском користувача до роботи в системі.

Традиційну автентифікацію за допомогою пароля називають ще однофакторною або слабкою. За наявності певних ресурсів перехоплення або підбір пароля є справою часу. Не останню роль в цьому грає людський чинник – чим стійкішим до взлому методом підбору є пароль, тим важче його запам'ятати людині і тим вище ймовірність що він буде додатково записаний, що підвищить ймовірність його перехоплення або викрадення. І навпаки – легкі для запам'ятовування паролі (наприклад часто вживані слова або фрази, як приклад, дати народження, імена близьких, назви моніторів чи найближчого обладнання) в плані стійкості до злому є дуже не вдалими. Як вихід, впроваджуються одноразові паролі, проте їхнє перехоплення також можливе [2].

Інші способи автентифікації включають:

– двофакторну автентифікацію – додає додатковий рівень захисту процесу автентифікації. 2FA вимагає, щоб користувач надав другий фактор автентифікації на додаток до пароля. Системи 2FA часто вимагають, щоб користувач вводив код

перевірки, отриманий за допомогою текстового повідомлення на попередньо зареєстрованому мобільному телефоні, або код, створений за допомогою програми автентифікації. Це перевіряє доступ до кількох служб, а також попереджає власника облікового запису про шкідливі спроби доступу до свого облікового запису.

– багатофакторну автентифікацію – вимагає від користувачів автентифікації більш ніж один фактор автентифікації, включаючи біометричний фактор, такий як відбиток пальця або розпізнавання обличчя, та фактор володіння, як маркер безпеки, створений програмою автентифікації.

– одноразовий пароль – це автоматично створений числовий або буквено-цифровий рядок символів, що автентифікує користувача. Цей пароль дійсний лише для одного сеансу входу або транзакції, і зазвичай використовується для нових користувачів або для користувачів, які втратили свої паролі, і їм надається одноразовий пароль для входу в систему і перехід до нового пароля.

– трифакторну автентифікацію – вимагає використання, як правило, фактору знань (пароль) в поєднанні з фактором володіння (маркери безпеки) і коефіцієнтом належності (біометричний).

– біометричну автентифікацію – хоча деякі системи автентифікації можуть залежати виключно від біометричної ідентифікації, біометрія зазвичай використовується як другий або третій фактор автентифікації. Більш поширені типи біометричної автентифікації включають сканування відбитків пальців, сканування обличчя або сітківки та розпізнавання голосу.

– мобільну автентифікацію – це процес перевірки користувача через їхні пристрої або перевірку самих пристроїв. Це дозволяє користувачам увійти в безпечні місця та ресурси з будь-якого місця. Процес автентифікації мобільного зв'язку включає багатофакторну автентифікацію, яка може включати одноразові паролі, введення PIN-кодів, біометричну автентифікацію або перевірку QR-коду.

– безперервну автентифікацію – при постійній автентифікації, замість того, щоб користувач входив або виходив, додаток компанії постійно обчислює "оцінку

автентичності", яка визначає, наскільки власником облікового запису є особа, яка використовує пристрій [3, 4].

Серед вище наведених способів автентифікації перспективним є саме біометрія, адже автентифікація даного типу використовує інформацію, що безпосередньо пов’язана із людиною, а так як людина – унікальна істота, відповідно, що її елементи тіла – унікальні. Перевага біометрії полягає в тому, що унікальні людські якості складно підробити, а залишити фальшивий відбиток пальця за допомогою власного, або зробити райдужну оболонку свого ока схожою на чиюсь є довгим та витратним заняттям. На відміну від паперових ідентифікаторів (паспорт, водійські права, посвідчення особи), від пароля або персонального ідентифікаційного номера, біометричні характеристики не можуть бути забуті або втрачені.

Вибираючи для системи той чи інший фактор або спосіб автентифікації, необхідно, перш за все, відштовхуватися від необхідного ступеня захищеності, вартості побудови системи та забезпечення мобільності суб'єкта і пам’ятати, що стійка автентифікація не забезпечує захист від активних атак, в ході яких маскується зловмисник, що може оперативно перехопити інформацію.

Дослідимо біометричну автентифікацію та візьмемо за основу її основні види, що використовуються у сучасних системах захисту.

1.2 Методи та засоби автентифікації на основі голосу

Розпізнавання голосу є технологією, яка дозволяє користувачеві застосовувати свій голос у якості вхідних даних. Розпізнавання голосу може використовуватися для диктування тексту комп'ютера або для подачі команд комп'ютера (наприклад, для відкриття програмних додатків, розгортання меню або збереження роботи).

Розглянемо декілька методів ідентифікації людини за голосом та оберемо оптимальний варіант для подальшої розробки програмного застосунку:

1) Кепстральний аналіз

У роботах шляхом розпізнавання голосу найбільш популярний метод кепстрального перетворення спектра мовних сигналів [5, 6].

Схема методу така: на інтервалі часу в 10-20 мс обчислюється поточний спектр потужності, а потім застосовується зворотне перетворення Фур'є від логарифма цього спектру (кепстра) і знаходяться коефіцієнти:

$$c_n = \frac{1}{\theta}\int_0^{\theta} |S(j,\omega,t)^2| \exp^{-jn\omega\,\Omega} d\omega, \Omega = 2\frac{2\pi}{\theta}$$

де θ – верхня частота в спектрі мовного сигналу, $|S(j, \omega, t)|^2$ – спектр потужності.

2) Метод GMM – Gauss Mixture Modules

Метод GMM випливає з теореми про те, що будь-яка функція щільності ймовірності може бути представлена як зважена сума нормальних розподілів.

Дуже часто в системах з цією моделлю використовується діагональна коваріаційна матриця. Вона може використовуватися для всіх компонент моделі або навіть для всіх моделей. Щоб знайти матрицю коваріації, ваги, вектори середніх часто використовують ЕМ-алгоритм. На вході маємо навчальну послідовність векторів X = $\{x1, \ldots, X_T\}$.

Параметри моделі ініціалізуються початковими значеннями і потім на кожній ітерації алгоритму відбувається переоцінка параметрів. Для визначення початкових параметрів зазвичай використовують алгоритм кластеризації такий, як алгоритм К-середніх [7].

Після того як відбулося розбиття безлічі навчальних векторів на М кластерів, параметри моделі можуть бути визначені так: початкові значення μ_j збігаються з центрами кластерів, матриці коваріації розраховуються на основі потрапляння в даний кластер векторів, ваги компонентів визначаються часткою векторів даного кластера серед загальної кількості навчальних векторів.

Кроки повторюються, доки не буде досягнуто сходження параметрів. Векторне квантування є найпростішою моделлю в системах розпізнавання, незалежних від контексту [8].

3) SVM – Support Vector Machine

Метод опорних векторів (SVM або MOB) будує гіперплощину або набір гіперплощин у просторі високої або нескінченної вимірності, які можна використовувати для класифікації, регресії та інших задач. Інтуїтивно, добре розділення досягається гіперплощиною, яка має найбільшу відстань до найближчих точок тренувальних даних будь-якого з класів (так зване функційне розділення), оскільки в загальному випадку що більшим є розділення, то нижчою є похибка узагальнення класифікатора.

4) На основі штучних нейронних мережей

Нейронна мережа – це мережа або контур нейронів, або в сучасному розумінні, штучна нейронна мережа, що складається з штучних нейронів або вузлів. Ці штучні мережі можуть бути використані для моделювання прогнозування, адаптивного контролю та застосувань, де вони можуть бути навчені за допомогою набору даних. Самонавчання, яке є результатом досвіду, може відбуватися в мережах, які можуть виводити висновки з складного і, здавалося б, не пов'язаного між собою набору інформації [9].

Переваги методу нейронних мереж проявляються в узагальнюючій здатності нейронних мереж дозволяти ідентифікувати шуми в сигналах, а також у високій швидкості розпізнавання, але перед цим треба витратити час на навчання мережі. На відміну від інших методів метод нейронних мереж не потребує великого обсягу пам’яті та має малу обчислювальну трудомісткість.

1.3 Формалізація вимог та постановка задачі

Виконавши дослідження вище викладених даних, можна зробити висновок, що одним з найбільш поширених засобів автентифікації при захисті від несанкціонованого доступу є використання двофакторної автентифікації.

Висвітлено головні переваги автентифікації на основі двох факторів, зокрема – варіативність комбінацій факторів при створенні системи автентифікації (пароль-відбиток пальця, пароль-голос, тощо) та незалежність факторів один від одного, а саме при досягненні зловмисником доступу до інформації про один із факторів системи, потрібно також витратити час на визначення другого фактору.

Головні недоліки двофакторної автентифікації – складність системи автентифікації і час її проходження та відсутність належних сервісів, що дозволили б швидко відновити доступ при втраті інформації про фактор, пов’язаний із характеристикою людини;

Програмний засіб створюється для безкоштовного використання звичайними користувачами, тому з проаналізованих засобів і підходів до захисту можна сформувати вимоги для покращення програмного засобу для захисту від несанкціонованого доступу методом двофакторної автентифікації.

Класична автентифікація та автентифікація на основі голосу є визначальними факторами в системі автентифікації. Класичну автентифікацію було обрано через зручність та тривіальність при створенні облікового запису. Автентифікація на основі голосу характеризується простотою в застосуванні. Переваги даного методу полягають у використанні лише мікрофону і звукової плати та складності відтворення голосу зловмисниками через постійну зміну даної характеристики через різні фактори (здоров’я людини, навколишнє середовище, тощо). В той же час, постійна зміна голосу є і визначальним недоліком, адже не всі системи використовують шумові фільтри при автентифікації користувача на відкритому просторі.

Для мінімізації впливу недоліків на програмний засіб, буде інтегровано спеціалізований хмарний сервіс, що використовує згорткові нейронні мережі для розпізнавання голосу.

На основі аналізу даних, розглянутих вище визначено наступні завдання:

– спроектувати архітектуру системи двофакторної автентифікації на основі голосу;

– делегувати задачі на окремі підсистеми роботи із створенням облікового запису, її керування, роботи з проміжними даними та їх візуалізацією;

– розробити алгоритми роботи для системи автентифікації у загальному форматі та окремо для кожної підсистеми;

– дослідити алгоритм роботи хмарного сервісу та виконати його інтеграцію у систему двофакторної автентифікації на основі голосу

– виконати підключення до хмарного сервісу, що виконує розпізнавання голосових повідомлень та на їх основі проводить процес автентифікації до користувача;

– обрати мову програмування та інструменти розробки;

– реалізувати роботу із графічним інтерфейсом програми, локальною базою даних та виконати підключення до бази даних хмарного сервісу;

– захистити локальну базу даних, використовуючи один із методів блокового симетричного шифрування;

– створити програмний засіб на основі спроектованої архітектури системи та розроблених алгоритмів роботи;

– виконати тестування програмного засобу.

Програмний засіб повинен включати в себе наступні модулі захисту, зокрема – двофакторну автентифікацію, захист бази даних.

Вимоги до модулю двофакторної автентифікації:

– для авторизації потрібно використовувати логін, пароль, «голосовий відбиток» та секретне слово;

– в разі невдалої спроби автентифікації потрібно завершити інтерактивний режим роботи із програмою і повернутися до головного меню програми;

– для голосової автентифікації потрібно використовувати хмарний сервіс із функцією розпізнавання голосу, що використовує нейронні мережі;

– при розпізнаванні голосу програмний засіб повинен виводити проміжні результати достовірності автентифікації користувача;

– при введенні інформації, зокрема логіна чи пароля, користувач повинен отримати повідомлення про наявність чи відсутність логіна та його відповідність до пароля облікового запису;

Вимоги до модулю захисту бази даних:

– у якості бази даних повинна використовуватись SQLite;

– у базі даних потрібно створити одну таблицю, що зберігає інформацію про логін, пароль користувача, його «голосовий відбиток» та секретне слово, і зашифрувати її за допомогою бібліотеки SQLCipher.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

2.1 Структура інформаційної технології

Інформаційна технологія автентифікації включає в себе комплекс методів і задач та складається з процесів, що є незалежними частинами. Інформаційна технологія реалізована у вигляді системи автентифікації на основі голосу, архітектура якої наведена на рисунку 2.1 [10].

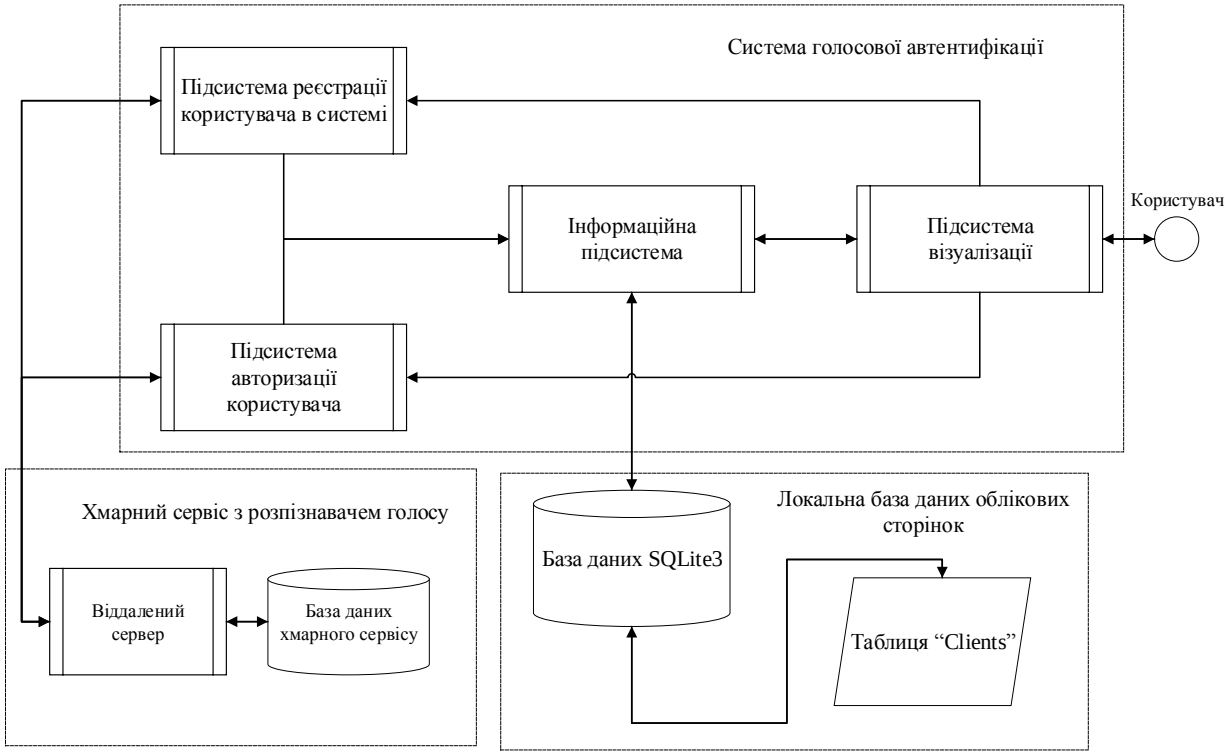


Рисунок 2.1 – Архітектура системи

Архітектура системи автентифікації на основі голосу складається з декількох процесів, що відрізняються за своїм функціональним призначенням. Виходячи з аналізу предметної області, виділяються наступні елементи розроблюваної системи:

– процес реєстрації користувача в системі;
– процес візуалізації;
– процес отримання інформації;
– процес керування [10].

Підсистеми знаходяться в блоці «Система голосової автентифікації». Для виконання поставлених завдань, в архітектуру також додані блоки з

використанням баз даних, як локальної (блок «Локальна база даних облікових сторінок»), так і віддаленої (блок «Віддалена стороння база даних з розпізнавачем голосу»).

Система починає роботу з підсистеми візуалізації, запускаючи графічний інтерфейс та ініціалізуючи дані, пов’язані з коректною роботою програми, взаємодіючи з інформаційною системою і виконуючи підключення до бази даних, в якій зберігаються дані до облікових сторінок користувачів [10].

Для виконання поставлених задач програма звертається до двох процесів в архітектурі – «Підсистема реєстрації користувача в системі» та «Підсистема керування», кожна з них безпосередньо звертається до інформаційної підсистеми, щоб отримати інформацію, що вводить користувач, як за допомогою клавіатури, так і за допомогою мікрофону, в залежності від того, яка підсистема виконує свої завдання звернення до бази даних відбувається або з метою читання даних, або з метою запису (append) нового облікового запису. При цьому, зазначається, що персональний комп’ютер користувача завчасно має встановлену звукову карту, мікрофон та підключення до інтернету.

При записі «голосового відбитку», програма звертається до віддаленого серверу, що має функцію розпізнавання голосу (далі – розпізнавач), який використовує нейронні мережі для ідентифікації голосу користувача та перевірки на її автентичність до входження в систему, перевірки на відповідність актуального голосу користувача та створеного голосового відбитку – голосового еталону [10].

Зчитані голосові повідомлення з мікрофона передаються до сервера з розпізнавачем та за допомогою вбудованих алгоритмів підраховує у відсотковому співвідношенні ймовірність факту вимови саме того слова чи фрази, яке передав через мікрофон користувач.

Детально розглянемо кожен елемент архітектури системи і засоби їх реалізації:

1) Процес реєстрації користувачів, призначений для реєстрації користувачів в систему. Додавання нового користувача реалізується за допомогою діалогових

вікон, що надається підсистемою візуалізації, зокрема, діалогові вікна типу «заповнення бланків» з елементами діалогу типу «меню» з можливістю прийняття рішення, натиснувши на кнопку «ОК» чи відміни операції, натискаючи на кнопку «Cancel». Як було описано вище, запис голосового відбитку є важливим процесом при проходженні автентифікації, адже для запису необхідно мати якісні апаратні засоби у вигляді мікрофону та звукової карти, а також високошвидкісний інтернет для передачі цілісних голосових даних [10]. Для реєстрації користувача необхідно створити логін, пароль на основі обов'язкових умов до створення, повторити його введення, перевіряючи його на відповідність попередньо створеному, записати голосовий відбиток та записати голосом секретне слово. Процес реєстрації користувачів забезпечує введення даних, що відображають атрибути реєстрації та її параметри. Атрибутами є текстова та голосова реєстрація R_t та R_v. Параметрами відповідають елементи, необхідні для успішного створення облікового запису, а саме логін, пароль, голос користувача та секретне слово – L, P, V, W. Виходячи з цього, маємо таку модель процесу реєстрації користувачів:

$$\mathrm{I}_1 = \langle\langle R_t, R_v\rangle, \langle L, P, V, W\rangle\rangle$$

2) Процес керування користувача виконує операції автентифікації користувача до облікового запису. Математична модель процесу керування майже не відрізняється від процесу реєстрації за винятком додаткового атрибуту достовірності автентифікації користувача N. Відповідно до цього, маємо наступну модель процесу:

$$\mathrm{I}_2 = \langle\langle R_t, R_v\rangle, \langle L, P, V, W\rangle, \langle N\rangle\rangle$$

3) Даний етап є необхідним для захисту від несанкціонованого доступу та копіювання. Автентифікація користувача, наявного в базі даних проводиться наступним чином – перевіряється чи є цей користувач в системі, якщо є, то програма виконає запит на отримання відповідного пароля від власника даного логіна. Після введення пароля, програма дасть можливість автентифікувати користувача за його голосом, чекаючи введення даних з мікрофона [10]. Протягом мовлення програма відправлятиме дані до віддаленого сервера із розпізнавачем, що використовує нейронні мережі для перевірки ідентичності отриманого голосу

та створеного голосового еталону, що збережений в базі даних. Як тільки звуковий зразок отримано, алгоритми програми виводять інформацію про ідентичність голосів у відсотковому співвідношенні та визначають результати для подальшого проходження автентифікації [10].

При позитивному результаті користувач повинен вимовити секретне слово, а згодом при його розпізнаванні – отримає повідомлення з успішним входженням в систему, інакше програма виведе повідомлення про помилку при автентифікації / відмову в доступі до програми. Результатами роботи даної підсистеми є візуальне відображення до екрану користувача. Суть її роботи полягає в ініціалізації графічних компонентів програми, зокрема відображенні головного вікна програми та кнопок з якими взаємодіятиме користувач, у вбудовуванні зображень та іконок, створенні діалогових вікон після натиснень на кнопки, створення моделі спектральної характеристики у реальному часі при записі голосу та впровадження інтерактивного режиму у кнопки та діалогові вікна [10]. Процес ідентифікації залежить від кількох факторів. Першим є використовувана система S. Для кожної системи характерні свої налаштування Con_S та формат введення/виведення даних F_S. Для диференціації визначених об’єктів характерний різний набір алгоритмів Alg. Тому, маємо наступну модель:

$$\mathrm{I}_3 = \langle\langle S, Con_s, F_s\rangle, Alg\rangle$$

4) Процес отримання інформації необхідний для зберігання даних на всіх етапах роботи підсистеми, забезпечення взаємозв'язку між іншими процесами, а також для збору і зберігання необхідних допоміжних даних, таких як параметри користувачів та зразки голосу користувача [11]. Даний процес організований у вигляді бази даних і має необхідний набір програмних засобів для доступу, пошуку, зміни і корекції даних, що зберігаються. Процес отримання інформації включає в себе базу даних із таблицею “Clients”, в якій зберігаються логіни, їх паролі, зразки голосу у форматі .wav та секретні слова до кожного логіну. Процес візуалізації даних пропонується описати за допомогою простору об’єктів інтерфейсу UI_{obj}. Відображення процесу H_i у відповідний їм об’єкт інтерфейсу UI^i_{obj}. Сукупність таких алгоритмів утворює простір об’єктів інтерфейсу Z_B.

Виходячи з цього, пропонується така модель процесу візуалізації інтерфейсу програмного засобу:

$$I_4 = \langle UI_{obj}, H, Z_B \rangle$$

2.2 Розробка алгоритмів функціонування системи

Програмний засіб виконуватиме дві основні функції – створення облікового запису та автентифікація у систему. При виконанні обох функцій, користувач повинен вводити свої дані, зокрема логін, пароль у вікна запиту, записувати «голосовий відбиток» та вимовити секретне слово для подальшої автентифікації в систему.

Програма, в свою чергу, шифруватиме локальні дані і передаватиме їх у базу даних системи автентифікації, передавати «голосовий відбиток» до хмарного сервісу, що виконує завдання розпізнавання голосового повідомлення, виконуватиме операцію читання з бази даних для підтвердження усіх етапів автентифікації та гарантувати ідентифікацію «голосового відбитку» з бази даних та утвореного користувачем [11, 12].

Також необхідно визначитись, яким чином хмарний сервіс буде слугувати програмному засобу, щоб мати можливість розробити алгоритм запису голосового еталону. Алгоритм ідентифікації користувача за голосом на хмарному сервісі виглядає наступним чином.

Хмарний сервіс складається з окремих модулів, зокрема з модулю роботи з розпізнаванням голосового повідомлення, модулю роботи з аудіо-файлами, модулю роботи з базою даних та модулю аналізу даних за допомогою нейронної мережі.

На рисунку 2.2 позначено схему процесів інформаційної технології.

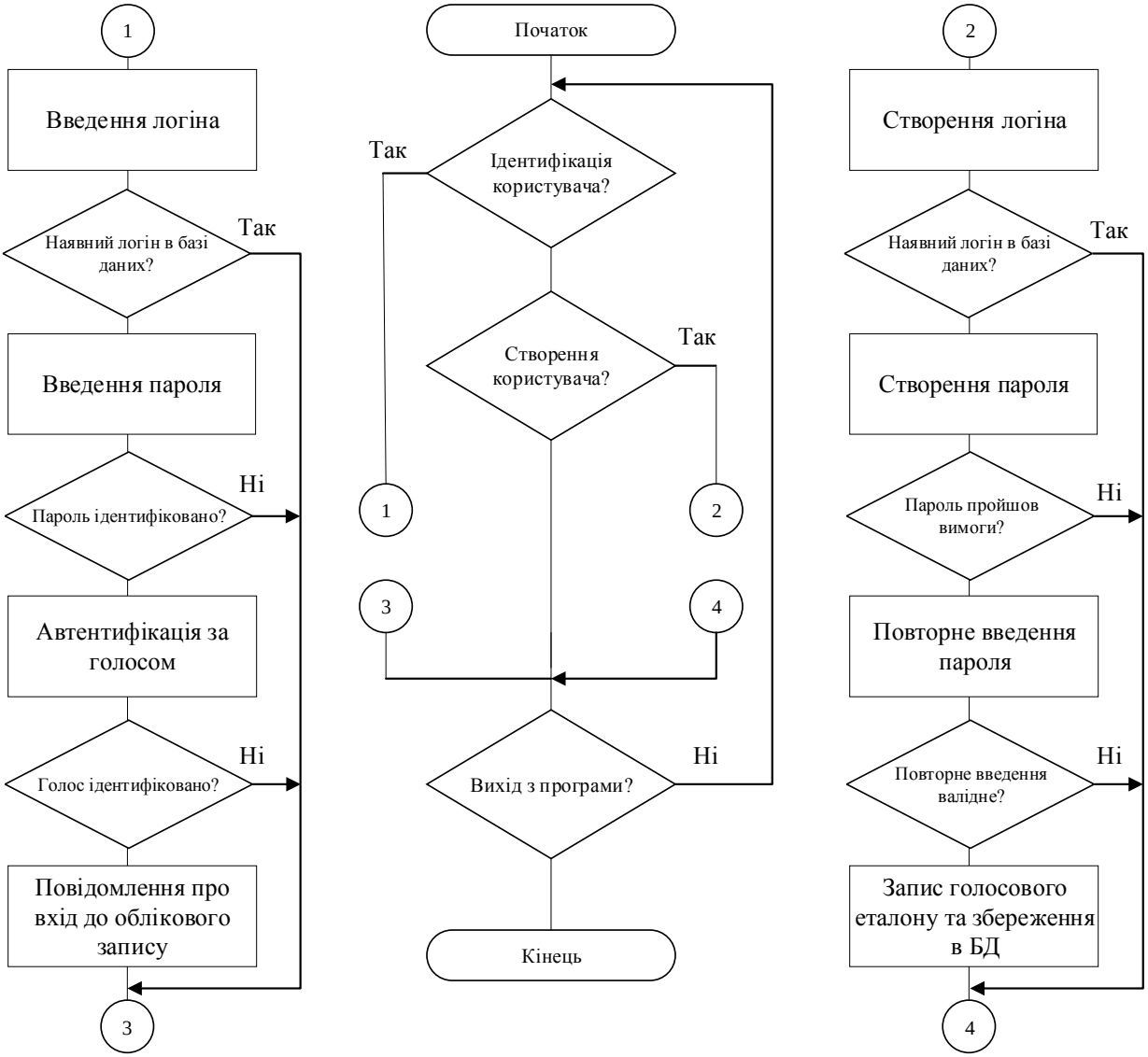


Рисунок 2.2 – Схема процесів інформаційної технології

За допомогою модуля роботи з базою даних хмарний сервіс надає системі автентифікації доступ до читання та запису голосових повідомлень користувачів, збережених при створенні облікового запису.

Алгоритм роботи з розпізнаванням голосового повідомлення розроблено для розпізнавання слів користувача та передачі інформації до подальшого аналізу даних для автентифікації аудіо-файлу.

Алгоритм роботи з файлом виконує завдання, пов’язані із записом файлу у форматі .wav та подальшим його використанням при автентифікації та розпізнаванні голосу.

Алгоритм аналізу даних за допомогою нейронної мережі використовується бібліотекою speech_recognition та виконує основні операції обчислення, в результаті якого отримується результат класифікації.

Алгоритм запису «голосового відбитку» працюватиме наступним чином:

1) система автентифікації, завчасно ініціалізуючи функції роботи із системою розпізнавання на хмарному сервісі та активуючи мікрофон повідомляє користувача про початок запису «голосового відбитку»;

2) користувач ознайомлюється із інструкцією у повідомленні, що надсилає програма;

3) користувач промовляє протягом певного часу будь-які слова чи фрази, що прийдуть першими до думки, при цьому виконується запис аудіо-файла з голосом користувача, при цьому хмарний сервіс працює одночасно із програмою, розпізнаючи кожне слово і записуючи дану інформацію у свою базу даних;

4) користувач вимовляє «стоп», щоб завершити запис аудіо-файла;

5) програма повідомляє користувача про необхідність вимовити секретне слово, для подальшої ідентифікації в систему;

6) користувач говорить це слово;

7) програма повідомляє користувача про завершення запису і запис інформації зі створеного облікового запису у локальну базу даних.

Робота з базою даних закладається у виконанні двох дій – записом інформації про новий обліковий запис у БД та читанням інформації для порівняння при автентифікації користувача [13].

Залишається визначити які дії система буде виконувати для успішної автентифікації користувача. Алгоритм порівняння «голосових відбитків» відрізняється способом доступу до бази даних хмарного сервісу, замість запису аудіо-файлу система виконуватиме запит на читання необхідного голосового повідомлення, закріпленого за користувачем, логін якого буде прописано перед виконанням даного алгоритму.

Спочатку система автентифікації виведе повідомлення з інструкціями для користувача, в якому описано що потрібно робити для успішної автентифікації до

облікового запису. Користувач, в свою чергу, вимовляє слова та фрази для порівняння його голосу із «голосовим відбитком», програма відправляє голосові дані до розпізнавача на хмарному сервісі. Після процесу розпізнавання та ідентифікації ознак голосового повідомлення, хмарний сервіс повертає інформацію про проміжні дані щодо відсотку достовірності автентифікації до системи. Програма виводить користувачеві повідомлення про статус його проходження автентифікації та ймовірність її успішності. Для проходження автентифікації, відсоток достовірності автентифікації повинна перевищити 85% включно. Дії повторюються ще пару разів, до тих пір, поки хмарний сервіс остаточно не ідентифікує користувача. Після остаточної ідентифікації, користувач вимовляє секретне слово і якщо воно відповідає наявному в базі даних, програма повідомляє користувача про успішну автентифікацію до системи.

Локальна база даних складається з однієї таблиці з назвою Clients, що складається з п’яти полів:

1) id – номер ідентифікатору користувача;
2) login – унікальне ім’я користувача в системі;
3) password – пароль до входу в систему;
4) voice_data – «голосовий відбиток» користувача;
5) secret_word – секретне слово користувача, завдяки якому ймовірність пройти автентифікацію збільшується.

База даних на хмарному сервісі зберігає «голосові відбитки» користувачів і використовує їх для порівняння отриманих аудіо-файлів від користувачів, що хочуть пройти автентифікацію. Інформація в базі даних обробляється операціями шифрування для захисту цілісності та конфіденційності [13].

Таким чином, спроектована система автентифікації на основі голосу, взаємодіючи із хмарним сервісом виконує поставлені на неї завдання.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

3.1 Розробка програмного засобу

Для запуску програми необхідно виконати файл voice_auth_system.exe, що знаходиться в корені директорії voice_auth_system. При запуску програми, користувач побачить головне меню з зображенням та трьома кнопками (рис. 3.1).

Рисунок 3.1 – Головне меню програми

Можливість змінювати розмір вікна – вимкнена, три кнопки – активні, при натисненні на кожну з них виконується певний комплекс дій. Основна робота з кнопками полягає у наявності інтерактивного режиму, а саме у роботі з діалоговими вікнами.

При натисненні на кнопку «Create Account…», користувач почне процедуру створення облікового запису. Дана процедура складається с кількох пунктів:

1) Користувач отримує вікно з повідомленням про створення логіну

2) Користувач прописує логін та натискує «ОК» або «Cancel», якщо бажає відмінити процедуру.

3) Програма отримує інформацію з текстового поля та порівнює її із логінами, що знаходяться в базі даних програми.

4) При наявності логіна в базі даних, програма виведе попередження про необхідність створити новий логін, інакше виведеться повідомлення про успішне створення логіну.

5) Користувач отримує вікно з повідомленням про створення логіну.

6) Користувач прописує пароль до логіну та натискує «ОК» або «Cancel», якщо бажає відмінити процедуру.

7) Програма отримує інформацію з текстового поля та перевіряє її на виконання обов'язкових умов – пароль повинен бути не менше 12 символів, мати один спеціальний символ, одну цифру та одну велику літеру.

8) При виконанні умов виведеться повідомлення про успішне введення пароля (рис. 3.2.а), інакше – про невідповідність обов'язковим вимогам до створення пароля (рис. 3.2.б).

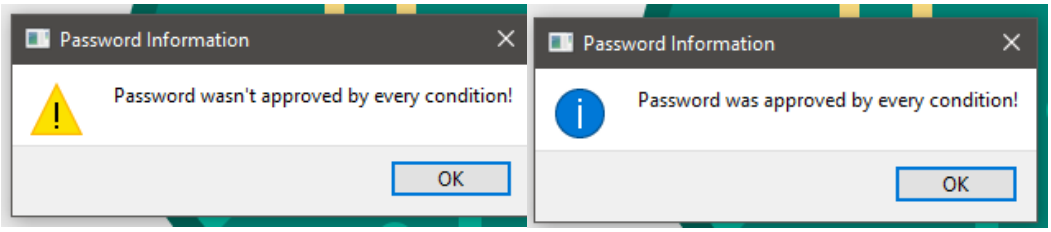


Рисунок 3.2 – Повідомлення про результати створення пароля:
а) попередження про невідповідність умовам паролю, б) повідомлення про успішне створення паролю

9) Користувач повторно прописує пароль для підтвердження першого етапу автентифікації.

10) При повторному введенні пароля, користувач може отримати повідомлення про невідповідність або про успішне введення пароля.

11) Програма виведе повідомлення про перехід до режиму запису голосу. Активується режим запису аудіо-файлу і виконується збереження у директорії sqlite/audio_files/<login>. Щоб завершити запис, користувач повинен сказати «стоп» та промовити секретне слово, яке буде використовуватись при подальшій автентифікації.

12) Програма чекає вхідних даних з мікрофона користувача. При відсутності інтернета голосовий розпізнавач отримає голосові дані, але не обробить їх, тому користувач повинен переконатись, що в системі увімкнений інтернет. Коли користувач говорить «стоп», програма виведе повідомлення про необхідність вимовлення секретного слова та подальшого завершення створення аккаунту.

13) Користувач вимовляє секретне слово, розпізнавач отримує інформацію з мікрофону, програма виводить повідомлення про успішний запис голосу та запису секретного слова.

14) Програма записує логін, пароль, аудіо-файл та секретне слово у таблицю clients в базі даних python_sqlite.db;

15) Програма виводить повідомлення про успішне створення облікового запису (рис. 3.3).

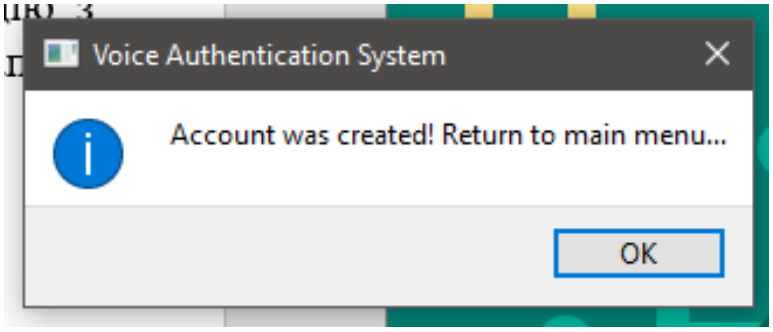


Рисунок 3.3 – Повідомлення про успішне створення облікового запису

Таким чином створюється обліковий запис в програмному засобі. Усі локальні дані переносяться до бази даних, аудіо-файл, що було записано користувачем при процесі розпізнавання даних, перетворюється у файл бінарного формату також надсилається до бази даних.

При натисненні на кнопку «Log In…», користувач почне процедуру автентифікації. Дана процедура складається с кількох пунктів:

1) Користувач отримує вікно з повідомленням про введення логіну.

2) Користувач вводить логін, якщо він наявний в базі даних – програма виведе повідомлення про наявність логіну, інакше – про її відсутність.

3) Користувач отримує вікно з повідомленням про введення паролю.

4) Користувач вводить пароль, якщо він закріплений за логіном, який автентифікується – програма виведе повідомлення про його відповідність, інакше – про його не відповідність.

5) Програма виведе повідомлення про перехід до режиму запису голосу.

6) Користувач вимовляє певні речення для програми, яка слухає його голос.

Через мікрофон програма отримує голосові дані і за допомогою бібліотеки speech_recognition, що використовує нейронні мережі визначатиме автентичність голосу для відповідного логіну.

Щоб пройти автентифікацію користувач тричі повинен отримати повідомлення про відповідність його голосу до голосового еталону.

Повідомлення дасть інформацію про ймовірність автентичності голосу (рис. 3.4).

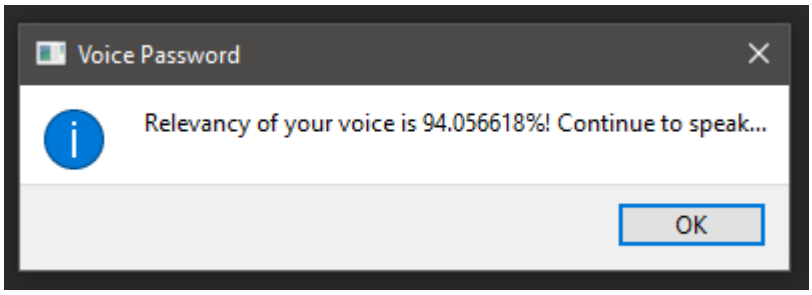


Рисунок 3.4 – Повідомлення про відповідність голосу на 94% до його голосового еталону

7) Після трьох таких повідомлень, програма виведе повідомлення із запитом до вимови секретного слова, що закріплений за логіном.

8) Користувач вимовляє секретне слово та отримує інформацію про успішну автентифікацію голосу (рис. 3.5.а), а також доступ до облікового запису (рис. 3.5.б).

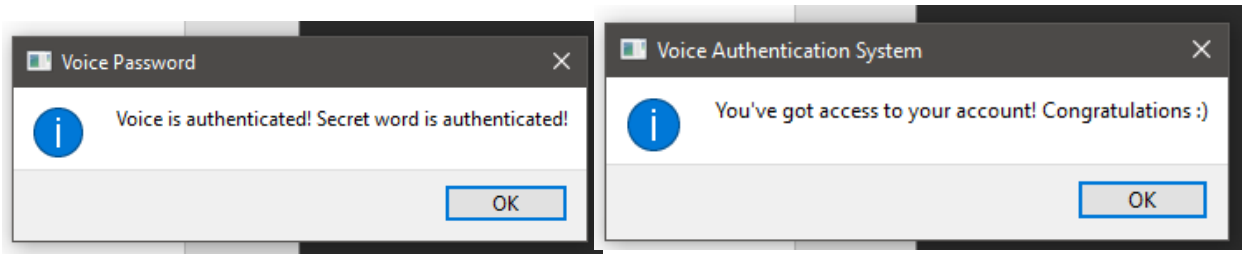


Рисунок 3.5 – Повідомлення про результати автентифікації за голосом: а) повідомлення про успішну голосову автентифікацію, б) отримання доступу до облікової сторінки

Таким чином виконується процес автентифікації користувача до облікового запису. Дані, що були використані у процесі зберігаються у базі даних в зашифрованому вигляді, як і передбачено програмним засобом.

3.2 Тестування програмного засобу

Для тестування були обрані два критерії, які обов'язково повинні бути забезпечені для нормальної роботи програмного забезпечення.

Перший критерій – відсутність помилок при роботі програми.

Другий критерій і найважливіший, забезпечення всіх покладених на програму вимог, тобто мінімально можливу кількість помилкових спрацьовувань алгоритму порівняння зразків голосу користувача з еталонними зразками, збереженими в базі даних.

Дослідження якості ідентифікації помилок першого роду (FAR – «чужих» користувачів, що пройшли поріг) та другого роду (FRR – «своїх», що не пройшли поріг) сприятиме до визначення порогового значення достовірності автентифікації для користувача. Хмарний сервіс має підготовлений набір даних, що складається із приблизно трьох тисяч людських голосів (жіночих та чоловічих) для подальшого використання при розпізнаванні голосу.

FRR є – добутком відношення числа відмов у доступі користувачам до загального числа спроб отримання доступу клієнтами і 100%. FAR є – добутком відношення числа надання доступу зловмисникам до загального числа спроб отримання доступу зловмисниками і 100%.

Підрахуємо значення FAR та FRR експериментальним чином. Для прикладу, визначимо чи користувач при автентифікації коректно вимовив секретне слово, і воно є аналогічним наявному в базі даних.

Чим більше буде проведено тестів, тим точніші дані будуть отримані – візьмемо тестовий обліковий запис із секретним словом «лампа». Бібліотека speech_recognition, окрім розпізнавання голосу та виведення достовірності автентифікації у систему автентифікації, також повертає достовірність кожного вимовленого слова у відсотковому представленні.

В цілому, проведено 25 тестувань з наступними результатами – 19 з 25 разів, хмарний сервіс розпізнав слово «лампа», з середнім значенням достовірності у 94-95%. Серед схожих слів найчастіше розглядались слова «рампа» – 90%, «лама» – 88%, «ламб» – 88%, «лава» – 88%, «ламбада» – 86%,

“Трамп» – 80%. При невдалих спробах пройти автентифікацію, хмарний сервіс розпізнавав слова «рампа» – 86-94%, «лама» – 81-90% та «лава» – 80-87%. Також одна невдала спроба сталась через затримку при розпізнаванні в результаті втрати підключення до хмарного сервісу.

Так як, FRR є значенням кількості «своїх», що не пройшли поріг, відповідно проведеному тестування його значення = 6 / 25 * 100% = 24%, що є задовільним результатом, адже важливішим фактором для системи автентифікації на основі голосу є відсутність можливості отримання доступу зловмисником.

Виконаємо аналогічне тестування із секретним словом «лампа», але при цьому навмисно будемо вимовляти інше слово, схоже на секретне слово. При експерименті використовувались слова «рампа», «лава» та «лама». Серед 25 тестів, 22 тести повернули значення достовірності до слів, що були вимовлені при експерименті, та лише двічі хмарний сервіс повернув найбільше значення достовірності до слова «лінь».

FAR – є значенням допуску «чужих», що пройшли поріг, відповідно результатам тестування, значення FAR = 3 / 25 * 100 = 12%. Даний відсоток є не поганим результатом, адже зловмиснику доведеться витратити чимало часу, щоб досягнути результату при проходженні систем автентифікації.

Значення EER (точку перетину FAR та FRR) дорівнює:

$$(FAR + FRR) / 2 = 18\%$$

Необхідно пам’ятати, що при зменшенні значення FAR – ймовірність автентифікації користувача зменшується, якщо зменшується FRR, то у зловмисника збільшується шанс пройти автентифікацію до облікового запису авторизованого користувача.

Тестування проводилось при різних умовах введення голосу від людини, зокрема коли людина була здоровою, хворою, знаходилась в спокійних умовах без шуму, знаходилась в умовах шумного простору. Для успішної автентифікації людина повинна мати ймовірність ідентифікації більше 85%, інакше автентифікації не буде пройдено. Середній час створення облікового запису – 2,5 хвилини, автентифікації – 2 хвилини.

В таблиці 3.1 описано усі варіації тестування та ймовірності ідентифікації людини.

Таблиця 3.1 – Тестування процесу автентифікації людини за голосом при різних умовах

№	Людина	Умови	Достовірність автентифікації, %			Середній час при створенні облікового запису, хв	Середній час при автентифікації користувача, хв
1	Чоловік №1	Здоровий при тиші	95.65%	96.22%	93.42%	01:33	01:09
2	Чоловік №2	Здоровий при шумі	90.13%	91.7%	91.1%	02:13	01:42
3	Чоловік №3	Хворий при тиші	86.45%	85.43%	88.88%	02:38	02:11
4	Чоловік №4	Хворий при шумі	84.1%	83.65%	84.92%	03:29	03:28
5	Жінка №1	Здорова при тиші	96.4%	97.41%	96.83%	01:47	01:17
6	Жінка №2	Здорова при шумі	91.86%	90.1%	93.2%	02:41	02:09
7	Жінка №3	Хвора при тиші	88.03%	86.52%	87.13%	02:45	02:06
8	Жінка №4	Хвора при шумі	84.9%	85.32%	84.8%	03:59	02:55

Результати показують, що найкращим варіантом для розпізнавання голосу є відсутність шуму та чудовий стан здоров’я людини.

Наступну сходинку із середнім значенням достовірності автентифікації 90% посідає варіант, коли людина здорова, але навколишнє середовище є шумним. Вбудований фільтр допомагає зменшити потенційну похибку при ідентифікації, очищаючи шуми від голосової інформації.

Коли людина хворіє і знаходиться в спокійному стані сервер із розпізнавачем ідентифікує голос в середньому на 87%. Даний результат є задовільним, але і близьким до мінімального порогового результату у 85%.

Найменшого результату розпізнавач досягає при поганому самопочутті людини та шумному просторі – в дуже рідкому випадку, людина зможе перевищити поріг ідентифікації, але як показує практика, для успішної автентифікації краще знайти місце тихіше або спробувати пізніше, але будучи у здоровому стані.

Також згідно статистики, при ідентифікації кожної статі, серед 16 проведених тестів – 15 успішних тестів на рахунок жінок та 13 – у чоловіків, при тихому навколишньому середовищі та здоровому самопочутті людини жінки мають більшу достовірність автентифікації, ніж чоловіки (середнє значення достовірності: 97% проти 94,5%), аналогічно жінки мають більший відсоток при не бажаних умовах тестування (середній значення достовірності: 85% проти 84%). У решті випадків спостерігається гендерна рівність (з невеликою перевагою жіночої статі).

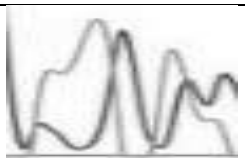
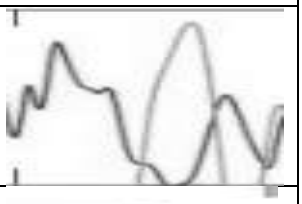
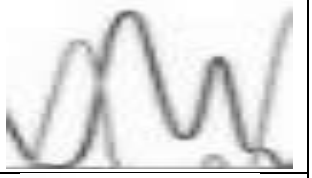
Час реєстрації облікового запису в середньому займав від 1,5 до 3 хвилин, найбільше часу займає процес запису «голосового відбитку» через стан здоров’я людини і шуми (чи їх відсутність) та надіслання його до хмарного сервісу, що займається розпізнаванням голосу. Чоловіки при цьому, впорались за коротший час, ніж жінки (дельта між виконанням – приблизно 30 секунд).

На автентифікацію користувача було витрачено менше часу, ніж на створення облікового запису, через просту форму класичної автентифікації та швидку перевірку хмарним сервісом ідентифікації користувача на основі його голосу. При часу автентифікації між чоловіками і жінками визначилась гендерна рівність (з невеликою перевагою чоловічої статі).

Для збільшення значення достовірності автентифікації користувач повинен забезпечити умови відсутності шумів, не дивлячись на наявність в хмарному сервісі засобу фільтрації, а також мати гарний стан здоров’я.

В таблиці 3.2 додані – спектральні характеристики протестованих голосів. Темними лініями зображені голоси користувачів, світлими – шуми.

Таблиця 3.2 – Спектральні характеристики людини

№	Людина	Умови	Спектральна характеристика
2	Чоловік №2	Здоровий при шумі	
3	Чоловік №3	Хворий при тиші	
4	Чоловік №4	Хворий при шумі	
5	Жінка №1	Здорова при тиші	
6	Жінка №2	Здорова при шумі	
7	Жінка №3	Хвора при тиші	
8	Жінка №4	Хвора при шумі	

Згідно статистики зображеної вище, якщо людина хворіє та знаходиться в шумних умовах – ймовірність ідентифікуватись не доходить до порогового значення, що означає відсутність можливості пройти автентифікацію.

Тестування проводилося на застарілій звуковій карті, що інтегрована в материнську плату. Картка з високим рівнем шуму та ігноруванням високих і низьких частот, а також зі слабким мікрофоном, який не забезпечує необхідний рівень запису. З хорошою звуковою картою, можна домогтися значно кращих результатів. Помилок в програмі в ході тестування виявлено не було.

ВИСНОВКИ

У процесі дослідження спроектовано програмний засіб для автентифікації на основі голосу та виконано поставлені завдання.

В результаті аналізу обґрунтовано вибір двофакторної автентифікації для реалізації захисту програмного засобу, її переваги та недоліки, проаналізовано механізм автентифікації з двома факторами (класична автентифікація та ідентифікація голосу) та її параметри, розглянуто сфери спеціалізації використання голосової автентифікації.

Розроблено архітектуру системи двофакторної автентифікації, що складається із підсистеми взаємодії із користувачем, бази даних та хмарного сервісу розпізнавання «голосового відбитку», зокрема – інформаційної підсистеми, підсистеми візуалізації, підсистеми реєстрації користувачів в систему та підсистеми керування. Досліджено архітектуру хмарного сервісу та його механізм взаємодії із системою автентифікації, описано роль рекурентної нейронної мережі у хмарній системі при виконанні поставлених завдань та взаємодії із базою даних.

Розроблено алгоритми роботи системи автентифікації на основі голосу, зокрема визначено роботу кожної підсистеми та її взаємодії із користувачем, реалізовано режими роботи програми при збереженні, читанні, записі та передачі даних до локальної бази даних та інтегровано роботу із хмарним сервісом.

На основі розробленої архітектури системи двофакторної автентифікації та її алгоритмів роботи створено програмний засіб, що вбудовується у програмне забезпечення, є простим для користувача та не вимагає додаткових маніпуляцій для встановлення та використання програмного забезпечення.

Проведено тестування програмного засобу, визначено похибку при тестуванні помилок першого та другого роду, достовірність автентифікації користувача при різних варіантах порівняння «голосового відбитку». Виявлено відсоток похибки при автентифікації користувача, середній час, що необхідний для створення облікового запису та автентифікації у систему.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Термін «Автентифікація», ВРУ [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/rada/term/263 – Назва з екрану.
2. Автентифікація в кібербезпеці [Електронний ресурс]. – Режим доступу: https://searchsecurity.techtarget.com/definition/authentication – Назва з екрану.
3. Иванов А. И. Биометрическая идентификация личности по динамике подсознательных движений – Пенза: Издательство ПГУ. – 2000. – 186 с. – ISBN 5-7260-0356-X.
4. Рыбченко Д. Е. Анализ клавиатурного почерка аппаратом нечетких множеств / Рыбченко Д. Е., Иванов А. И. // Технические средства конфиденциальной связи. – Пенза: ПНИЭИ. – 1996. – Выпуск №1. – с. 116-119.
5. Тасов К. Л. Метод ідентифікації людини по голосу / Тасов К. Л., Дятлов Р. А. – [Електронний ресурс]. – Режим доступу: http://engjournal.ru/articles/1103/1103.pdf – Назва з екрану.
6. Davis S. Comparison of parametric representations. IEEE Transactions on Acoustics, Speech and Signal Processing / Davis S., Mermelstein P. // Haskins Laboratory. 1980. – Т. 28. – с. 357-366.
7. Первушин Є. А. Огляд основних методів розпізнавання дикторів [Електронний ресурс]. – Режим доступу: http://msm.univer.omsk.su/jrns/jrn24/pervushinOverview.pdf – Назва з екрану.
8. Press W. H. Computing Recipes – The Art of Science Computing/ [Press W. H., Teukolsky S. A., Vetterling S. A. та ін.] // New York Est.: Cambridge University. – 2007. – 1256 p. – ISBN 978-0-521-88068-8
9. Purves D. Neuroscience, 5th edition // Sinauer Est. – 2011. – 507 p. – ISBN 978-0-878-93695-3.
10. Куперштейн Л., Лукічов В., Айвазян С. Система двофакторної автентифікації на основі голосу. [Електронний ресурс]. – Режим доступу: http://www.konferenciaonline.org.ua/arhiv-konferenciy/arhiv-konferenciy11-06-2019 – Назва з екрану.
11. Панченко Д. Розпізнавання мови на основі штучних нейронних мереж [Електронний ресурс]. – Режим доступу: https://moluch.ru/conf/tech/archive/3/712/ – Назва з екрану.
12. Нейронна мережа зустрічного розповсюдження та її принципи функціонування і навчання. [Електронний ресурс]. – Режим доступу: https://studwood.ru/1860384/informatika/neyronna_merezha_zustrichnogo_rozpovsyudzhennya – Назва з екрану.
13. Yao J. A Case Study on Using Neural Networks to Perform Technical Forecasting of / Yao J., Tan C. // Massie University. – Neurocomputing, vol. 34. – 2000. – p. 79-98.

Додаток А
Текст програми

```
# Voice Authentication System
# Built-In Modules
import os
import sys
# Array Print, Regular Expressions
from pprint import pprint
import re
# Voice and Frequency Settings
import numpy as np
import pyaudio
import matplotlib.pyplot as plt
# Speech Recognition
import speech_recognition as sr
# SQLite3
import sqlite3
from sqlite3 import Error
# PyQt5
from PyQt5.QtWidgets import \
    (QApplication, QPushButton, QMessageBox, QWidget, QLabel, QVBoxLayout,
QInputDialog)
from PyQt5.QtGui import QIcon, QPixmap

class GraphicalUserInterface(QWidget):
    def __init__(self):
        """ Конструктор за замовчуванням """
        # Виклик батьківського конструктору QWidget
        super().__init__()
        # Ініціалізація розпізнавача
        self.r = sr.Recognizer()
        # Створення класу з використанням бази даних
        self.vas_database = SQLDatabase()
        # Створення класу з використанням спектру голоса
        self.voice_spectrum = VoiceSpectrum()
        # Ініціалізація графічного інтерфейсу
        self.init_ui()
    def init_ui(self):
        """ Ініціалізація графічного інтерфейсу """
        # Встановлення іконку додатку.
        self.setWindowIcon(QIcon('images/vas.png'))
        # Створення макету горизонтального блока
        vertical_box = QVBoxLayout(self)
        # Додати зображення в макет
        program_image = QPixmap('images/vas_in_program.png')
        label_with_image = QLabel(self)
        label_with_image.setPixmap(program_image)
        # MessageBox
        self.msg = QMessageBox()
        # Створення кнопок для макету
        b_create_account = QPushButton("Create Account...")
        b_log_in = QPushButton("Log In...")
        b_exit_program = QPushButton("Quit...")
        # Додавання елементів в макет
        vertical_box.addWidget(label_with_image)
        vertical_box.addWidget(b_create_account)
        vertical_box.addWidget(b_log_in)
        vertical_box.addWidget(b_exit_program)
        # Встановити макет
        self.setLayout(vertical_box)
        # Додаємо обробники подій
        b_create_account.clicked.connect(self.create_account)
```

```
        b_log_in.clicked.connect(self.log_in)
        b_exit_program.clicked.connect(self.close)
        # Налаштовуємо розташування і налаштування вікна
        self.setFixedSize(630, 430)
        self.setWindowTitle('Voice Authentication System')
        self.show()
    def create_account(self):
        """ Подія активації діалогового вікна """
        # Створити логін
        self.login = self.create_login()
        # Якщо натиснуто кнопку "Cancel"...
        if self.login == "Canceled":
            print("Canceled! Return to main menu...")
            return
        else:
            # Якщо логін не пустий...
            if self.login != "":
                self.msg.setIcon(QMessageBox.Information)
                self.msg.setWindowTitle("Login Information")
                self.msg.setText("Login was created!")
                self.msg.setStandardButtons(QMessageBox.Ok)
                retval = self.msg.exec_()
                # вивести повідомлення про його створення.
                print("Login was created! Create your password...")
            elif self.login == "":
                self.msg.setIcon(QMessageBox.Warning)
                self.msg.setWindowTitle("Login Information")
                self.msg.setText("Login is in database! Change your login...")
                self.msg.setStandardButtons(QMessageBox.Ok)
                retval = self.msg.exec_()
                # оголосити про наявність логіна в базі даних та повернутися в
головне меню.
                print("Login is in database! Return to main menu...")
                return
        # Створити пароль
        self.password = self.create_password()
        # Якщо пароль не пустий...
        # Якщо натиснуто кнопку "Cancel"...
        if self.password == "Canceled":
            print("Canceled! Return to main menu...")
            return
        else:
            if self.password != "":
                # вивести повідомлення про його створення.
                self.msg.setIcon(QMessageBox.Information)
                self.msg.setWindowTitle("Password Information")
                self.msg.setText("Password was approved by every condition!")
                self.msg.setStandardButtons(QMessageBox.Ok)
                retval = self.msg.exec_()
                print("Password was created!")
            elif self.password == "":
                # оголосити про невідповідний пароль і повернутися в головне меню.
                self.msg.setIcon(QMessageBox.Warning)
                self.msg.setWindowTitle("Password Information")
                self.msg.setText("Password wasn't approved by every condition!")
                self.msg.setStandardButtons(QMessageBox.Ok)
                retval = self.msg.exec_()
                print("Password isn't recommended for saving! Return to main
menu...")
                return
        # Повторити введення паролю
        self.repeat_password = self.check_password(self.password)
        # Якщо натиснуто кнопку "Cancel"...
        if self.repeat_password == "Canceled":
```

```
            print("Canceled! Return to main menu...")
            return
        else:
            # Якщо пароль переписаний правильно...
            if self.repeat_password != "":
                # вивести повідомлення про його підтвердження.
                self.msg.setIcon(QMessageBox.Information)
                self.msg.setWindowTitle("Password Information")
                self.msg.setText("Repeated password was checked and it's
relevant!")
                self.msg.setStandardButtons(QMessageBox.Ok)
                retval = self.msg.exec_()
                print("Password was checked! Create your voice password...")
            elif self.repeat_password == "":
                # оголосити про невідповідність та повернутись в головне меню.
                self.msg.setIcon(QMessageBox.Warning)
                self.msg.setWindowTitle("Password Information")
                self.msg.setText("Repeated password was checked and it isn't
relevant!")
                self.msg.setStandardButtons(QMessageBox.Ok)
                retval = self.msg.exec_()
                print("Password isn't equal with previous one! Return to main
menu...")
                return
        self.secret_word, self.voice_password = self.create_voice_password()
        project = (self.login, self.password, self.voice_password,
self.secret_word)
        self.vas_database.insert_data(self.vas_database.conn, project)
        self.msg.setIcon(QMessageBox.Information)
        self.msg.setWindowTitle("Voice Authentication System")
        self.msg.setText("Account was created! Return to main menu...")
        self.msg.setStandardButtons(QMessageBox.Ok)
        retval = self.msg.exec_()
        print("Account was created! Return to main menu...")
        return
    def create_login(self):
        """ Створення логіну """
        # Прочитати стовпчик таблиці з логінами в БД
        self.sql_check_info = '''SELECT login FROM clients'''
        connection = self.vas_database.conn
        # Перетворити логіни в нормальний формат
        connection.row_factory = lambda cursor, row: row[0]
        # Виконати запит по читанню даних
        c = connection.cursor()
        # Прочитати усі логіни і зберегти у змінну
        login_from_db = c.execute(self.sql_check_info).fetchall()
        # Введення логіну
        created_login, button_result = QInputDialog.getText(self, 'Create
Account', 'Create login:')
        if button_result:
            # Перебор логінів з бази даних
            for check_login in login_from_db:
                # Якщо створений логін є в базі даних
                if created_login == check_login:
                    return ""
            return created_login
        else:
            return "Canceled"
    def create_password(self):
        """ Створення паролю """
        created_password, button_result = QInputDialog.getText(self, 'Create
Password', 'Create password:')
        if button_result:
            special_symbols, caps_symbols, numbers = 0, 0, 0
```

```
            if len(created_password) < 12:
                print("Little amount of symbols")
            for symbol in created_password:
                if str(symbol).isupper() and caps_symbols == 0:
                    caps_symbols += 1
                if str(symbol).isdigit() and numbers == 0:
                    numbers += 1
                if str(symbol) in "!@#$%^&*();\"\':,./\\!№;%:?*()" and
special_symbols == 0:
                    special_symbols += 1
            if caps_symbols > 0 and numbers > 0 and special_symbols > 0:
                print("Password is recommended for saving!")
                return created_password
            return ""
        else:
            return "Canceled"
    def check_password(self, password):
        repeat_password, button_result = QInputDialog.getText(self, 'Repeat
Password', 'Repeat password:')

        if button_result:
            if password == repeat_password:
                return repeat_password
            else:
                return ""
        else:
            return "Canceled"
    def create_voice_password(self):
        """ Створення голосового відбитку (еталону) """
        self.msg.setIcon(QMessageBox.Information)
        self.msg.setWindowTitle("Voice Password")
        self.msg.setText("Record your voice! To finish recording - say \"стоп\"
and after it - say secret word!")
        self.msg.setStandardButtons(QMessageBox.Ok)
        retval = self.msg.exec_()
        # Включення мікрофону
        with sr.Microphone() as source:
            self.r.adjust_for_ambient_noise(source)
            print("Say something")
            exit_program = False
            path_to_audio = r'sqlite/audio_files/' + self.login
            if not os.path.exists(path_to_audio):
                os.makedirs(path_to_audio)
            audio = self.r.listen(source, timeout=100, phrase_time_limit=None)
            with open(path_to_audio + "/" + "voice.wav", "wb") as f:
                f.write(audio.get_wav_data())
            while True:
                audio = self.r.listen(source, timeout=100, phrase_time_limit=None)
                try:
                    text = self.r.recognize_google(audio, show_all=True,
language="ru-RU")
                    pprint(text)
                except:
                    print("Didn't work.")
                try:
                    for list_of_recognizing in text.get("alternative"):
                        pprint(list_of_recognizing)
                        print(type(list_of_recognizing))
```