Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія авторизації користувача з використанням клавіатурного почерку»

Виконала: студентка 2-го курсу, групи 1КН-21м спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
_____ Король Я.О.
(прізвище та ініціали)

Керівник: к.т.н., ст. викл.
_____ Озеранський В.С.
(прізвище та ініціали)
«15» 12 2022 р.

Опонент: к.т.н., доцент каф. КСУ
_____ Юхимчук М. С.
(прізвище та ініціали)
«15» 12 2022 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)
«16» 12 2022 р.

Вінниця ВНТУ - 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти ІІ-й (магістерський)
Галузь знань – 12 – Інформаційні технології
Спеціальність – 122 – Комп'ютерні науки
Освітньо-професійна програма – Системи штучного інтелекту

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.

(підпис)
"14" 09 2022 року

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

Король Яні Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи: «Інформаційна технологія авторизації користувача з використанням клавіатурного почерку»
керівник роботи к.т.н., ст. викл. Озеранський В. С.
затверджені наказом вищого навчального закладу від «14» 09 2022 року № 203
2. Строк подання студентом роботи 18 листопада 2022 року
3. Вхідні дані: мова програмування – об'єктно-орієнтована; автентифікація користувача – двофакторна; точність визначення користувачів – не менше 80%; час обробки запиту – не більше 0,3 с.
4. Зміст текстової частини: вступ, обгрунтування доцільності розробки інформаційної технології авторизації користувача з використанням клавіатурного почерку, моделювання інформаційної технології авторизації користувача з використанням клавіатурного почерку, структурна огранізація та особливості програмної реалізації інформаційної технології авторизації користувача з використанням клавіатурного почерку, економічна частина, висновки, перелік використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): схема етапів інформаційної технології авторизації користувача з використанням клавіатурного почерку; схема архітектури системи виявлення підміни користувача; схема алгоритму авторизації користувача з використанням клавіатурного почерку; схема загального алгоритму інформаційної технології авторизації користувача з використанням клавіатурного почерку; UML-діаграма прецедентів для програми авторизації користувача з використанням клавіатурного почерку; UML-діаграма класів інформаційної технології авторизації користувача з використанням клавіатурного почерку; приклад роботи програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконан прийня
1-3	Озеранський В. С. к.т.н., ст. викладач каф. КН	14.09.22	14.12.22
4	Нікіфорова Л. О. к.е.н. доцент каф. ЕПВМ	19.09.22	12.12.2

7. Дата видачі завдання 14.09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Прим
1	Обґрунтування доцільності розробки інформаційної технології авторизації користувача з використанням клавіатурного почерку	14.09.2022р – – 1.10.2022р	
2	Моделювання інформаційної технології авторизації користувача з використанням клавіатурного почерку	2.10.2022р – -16.10.2022р	
3	Структурна організація та особливості програмної реалізації інформаційної технології	17.10.2022р – – 07.11.2022р	
4	Підготовка економічної частини	08.11.2022р – 21.11.2022р	
5	Апробація та/або впровадження результатів дослідження	23.11.2022р – 01.12.2022р	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.12.2022р – – 14.12.2022р	

Студент _____ (підпис) Король Я.О.

Керівник роботи _____ (підпис) Озеранський В.С.

АНОТАЦІЯ

УДК 621.374.415

Король Я. О. Інформаційна технологія авторизації користувача з використанням клавіатурного почерку. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма – Комп'ютерні науки. Вінниця: ВНТУ, 2022. 131 с.

На укр. мові. Бібліогр.: 42 назв; рис.: 20; табл. 8.

У магістерській кваліфікаційній роботі розроблено інформаційну технологію авторизації користувача з використанням клавіатурного почерку. Вона дозволяє підвищити точність авторизації користувачів у системі. У загальній частині роботи розглянуто особливості розробки програмного забезпечення для авторизації користувачів, а також обґрунтована доцільність їх розробки. Розроблено алгоритм авторизації користувача з використанням клавіатурного почерку. Програмне забезпечення розроблено мовою програмування Java, в середовищі IntelliJ IDEA, приведено тестування та аналіз результатів роботи.

Графічна частина складається з 7 плакатів із результатами моделювання.

У розділі економічної частини здійснено оцінювання комерційного потенціалу розробки інформаційної технології авторизації користувача з використанням клавіатурного почерку, проведено оцінювання комерційного потенціалу розробки, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, які склали 840 322,38 грн, розраховано період окупності – 0,92 роки.

Ключові слова: інформаційна технологія, авторизація користувача, клавіатурний почерк.

ABSTRACT

Korol Ya. O. Information technology of user authorization with the use of keyboard handwriting. Master's degree in specialty 122 – Computer Science, educational program – Computer Science. Vinnytsia: VNTU, 2022. 131 p.

In Ukrainian language. Bibliogr .: 42 titles; fig .: 20; table 8.

In the master's qualification work the information technology of user authorization with the use of keyboard handwriting is developed. It allows you to expand the functionality of the user authorization program in the system. In the general part of the work the peculiarities of software development for user authorization are considered, as well as the expediency of their development is substantiated. An algorithm for user authorization using keyboard handwriting has been developed. The software is developed in the Java programming language, in the IntelliJ IDEA environment, testing and analysis of work results are provided.

The graphic part consists of 7 posters with simulation results.

In the section of the economic part the estimation of commercial potential of development of information technology of authorization of the user with use of keyboard handwriting is carried out, the estimation of commercial potential of development is carried out, the cost of research and implementation of results, which amounted to 840 322,38 UAH, the payback period – 0,92 years.

Keywords: information technology, user authorization, keyboard handwriting.

ЗМІСТ

ВСТУП

Актуальність теми дослідження. На сьогоднішній день у світі повсюди застосовуються автоматизовані системи, які досягли дуже великих масштабів і обіймають майже всі етапи життя людей, компаній і підприємств. У цілому це система, що включає персонал та набір засобів для автоматизації його діяльності та втілює інформаційні технології для виконання встановлених функцій. Однією з найпроблематичніших галузей реалізації автоматизованих інформаційних систем вважається робота зі структурами документообігу.

Такі системи можуть підвищити ефективність роботи працівників, у разі гарної організації взаємодії між людьми та автоматизованими інформаційними системами, проте разом з цим існує багато недоліків, які необхідно виправити, щоб уникнути помилок, але перш за все, потрібно переконатися що ця система працює з тими, хто має дозвіл авторизації та необхідні навички. Якщо системою користується некомпетентна особа або зловмисник, можна завдати шкоди різного масштабу, наприклад, оприлюднення приватних даних, завдання шкоди компанії, шкоди репутації, що може призвести до багатьох проблем, залежно від масштабів системи. З цією метою поряд із розширенням та популяризацією інформаційних технологій розробляються системи інформаційної безпеки, що включають системи авторизації для запобігання доступу до даних сторонніх і злочинців, «хакерів», що постійно модернізовують свої навички та винаходять все нові способи для обходу системи авторизації. Через постійну конкуренцію між «хакерами» та фахівцями з інформаційної безпеки захист постійно покращується, а доступ до інформації, що не призначена для незнайомців, що видають себе за когось іншого стає менш реальним. Щоб прийняти рішення, чи потрібно надати особі доступ, або заборонити третій стороні отримувати інформацію, яка їй не належить. Для цього всі системи використовують авторизацію та автентифікацію [1].

На жаль, сучасні програми, що містять розпізнавання почерку на клавіатурі мають проблеми через нестатичну біометричну характеристику людини, що

може змінюватися в залежності від психоемоційного та фізичного стану користувача. Тому важливо розробити нову систему авторизації користувачів з використанням клавіатури, яка допомогла б зробити ідентифікацію та автентифікацію користувачів кращою та точнішою.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження є підвищення точності авторизації користувача з використанням методу периферійного введення особистих даних, що дозволить розширити функціональні можливості програмних засобів.

Об'єкт дослідження – це процес авторизації користувача з використанням клавіатурного почерку.

Предмет дослідження – програмні засоби авторизації користувача з використанням клавіатурного почерку.

Для досягнення поставленої мети необхідно розв'язати такі **задачі**:

- проаналізувати предметну область авторизації користувача з використанням клавіатурного почерку;
- провести аналіз сучасних систем авторизації користувача;
- розробити структуру інформаційної технології авторизації користувача з використанням клавіатурного почерку;
- розробити математичну модель інформаційної технології авторизації користувача з використанням клавіатурного почерку;
- здійснити програмну реалізацію інформаційної технології авторизації користувача з використанням клавіатурного почерку;
- провести тестування роботу програми авторизації користувача з використанням клавіатурного почерку.

Методи дослідження. У роботі використано такі методи наукових досліджень: метод біометричної ідентифікації; методи прихованого моніторигу; метод розпізнавання клавіатурного почерку користувача, методи визначення клавіатурного почерку на основі оцінки тривалості утримання клавіш; методи об’єктно-орієнтованого програмування для автоматизації розрахунків.

Наукова новизна одержаних результатів полягає в наступному: удосконалено інформаційну технологію авторизації користувача з використанням клавіатурного почерку, яка відрізняється від існуючих підвищенням точності визначення користувача за допомогою використання характеристик, властивих конкретному користувачу.

Практичне значення одержаних результатів полягає у такому:

1. Удосконалено алгоритм авторизації користувача з використанням клавіатурного почерку.

2. Здійснено програмну реалізацію інформаційної технології авторизації користувача з використанням клавіатурного почерку.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок здобувача. Результати даної магістерської кваліфікаційної роботи отримані самостійно. У публікації в співавторстві здобувачу належить дослідження перспектив інформаційної технології авторизації користувача з використанням клавіатурного почерку [1].

Апробація результатів роботи. Результати досліджень було апробовано на «Науково-технічна конференція підрозділів Вінницького національного технічного університету (2022)».

Публікації. За основними результатами досліджень опубліковано тези доповіді на конференції [1].

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОРИЗАЦІЇ КОРИСТУВАЧА З ВИКОРИСТАННЯМ КЛАВІАТУРНОГО ПОЧЕРКУ

1.1 Аналіз загроз інформації в автоматизованих системах електронного документообігу

При широкому використанні автоматизованих систем і систем електронного документообігу зросла можливість несанкціонованого доступу в злочинних цілях. Незахищені системи зв'язку, включаючи автоматизовані системи, залишаються сьогодні особливо вразливими. Інформація, яка там поширюється, може бути незаконно модифікована, викрадена або видалена.

Останнім часом у ЗМІ з’явилося багато повідомлень про злочинний вплив на автоматизовані системи обробки, зберігання та передачі інформації, переважно в банківській сфері. Відповідно до оперативної інформації Спеціальної державної служби з питань захисту державних інформаційних ресурсів у проміжку часу з 23 по 29 травня 2022 року системою кіберзахисту державних інформаційних ресурсів та об’єктів моніторингу критичної інфраструктури зафіксовано 45 973 підозрілі дії, що є на 16% менше, ніж минулого тижня. Більша частина підозрілих подій пов’язана зі спробами сканування мережі (55%), знаходженням нестандартних протоколів чи подій (27%), виявленням троянських програм у мережі (7%), веб-атаками (6%), і спроби отримати права адміністратора (4%). Система безпечного доступу державних органів до Інтернету заблокувала 7664 різні види атак, що на 49% менше, ніж минулого тижня [2]. Більшість з них (98%) – це мережеві атаки на рівні програми. Також заблоковано п’ять DDoS-атак, більшість з яких – на сайтах Офісу Президента України [3]. Урядова група реагування на надзвичайні комп’ютерні ситуації, CERT-UA, зафіксувала 463 кіберінциденти та відреагувала на них.

Для боротьби з кіберзлочинністю або зменшення її шкоди потрібно вводити заходи та засоби для гарантування захисту інформації від неправомірного виходу та несанкціонованого доступу до неї. Також існує потреба у знаннях основних законодавчих положень в цій сфері, організаційних, програмних та інших заходах щодо забезпечення інформаційної безпеки. Розвиток засобів, методів і форм автоматизації обробки інформації та узагальнення персональних комп'ютерів робить інформацію більш вразливою. Для всіх аспектів інформаційної безпеки головним компонентом вважається аналіз можливих операцій щодо порушення роботи автоматизованих систем.

Основними факторами, які підвищують його вразливість, є [4]:

- збільшення обсягу інформації, що збирається, зберігається та обробляється електронно-обчислювальними машинами;

- концентрація в базах даних інформації різних функцій та різного відношення;

- розширення кількості користувачів, які мають прямий доступ до ресурсів комп'ютерної системи та наборів даних;

- перевантаженість режимів роботи технічних засобів комп'ютера;

- обмін інформацією в локальних і глобальних мережах, зокрема на великі дистанції.

Нещодавно відбувся перехід від традиційної форми подання документів до електронної. Це сталося завдяки перевагам користування електронного документообігу, що значно спрощує роботу зі створення, зберігання та надсилання важливої інформації. Однак залишаються невирішеними деякі питання щодо забезпечення надійного обміну електронними документами, боротьби з розголошенням, отриманням несанкціонованого доступу до документів електронного формату. Впровадження систем електронного документообігу створює нові потенційні загрози у сфері інформаційної безпеки.

Завдання, які покладаються на системи електронного документообігу, полягають у реєстрації документів та їх поточного стану в спеціальній базі даних, результатом чого є заповнення спеціальної реєстраційної картки документа. База

даних не містить оригінальних документів, а лише відображає їх поточне розташування та статус, включно з атрибутами моніторингу продуктивності. Крім обробки та пошуку документів в базі даних, система повинна забезпечувати формування звітів, що надають можливість надходження інформації про використання документа та іншу основну інформацію. Автоматизація системи документообігу дає змогу:

- підвищити виконавську дисципліну – це робиться завдяки покращенню контролю реалізації доручень за документами;

- продуктивно систематизувати повідомлення та сповіщення, що дає можливість попередити всіх агентів про наближення терміну;

- використовуючи підсумкові звіти та журнали, легко побудувати повну картину роботи окремих співробітників і компанії в цілому;

- сформувати персональні шляхи документів, що дозволяє швидко здійснити найбільш оптимальний маршрут їх переміщення в організації.

- автоматична генерація номерів, поточної дати, використання довідників і словників скорочує час реєстрації та дозволяє уникнути помилок, пов'язаних із заповненням реквізитів документа. Засоби системи дозволяють миттєво шукати документи та замовлення (за їх змістом і будь-яким поєднанням реквізитів).

Існує кілька істотних перешкод у проведені автоматизації документообігу [5]:

- низька обізнаність фахівців, що займаються розробкою автоматизованих інформаційних систем, які встановлюють систему, про діяльність державного органу, який вони автоматизують;

- низька обізнаність IT-компаній, які встановлюють систему, про специфіку документообігу структур даних компаній;

- наявність недостатньої співпраці між підрядниками та безпосередніми користувачами;

- відсутність в команді, що встановлюють системи, спеціалістів із системного документообігу – фахівців, які знають особливості генерації, руху, типової відповідності інформаційних запитів споживачів, а також знання інформаційно-комунікаційної інтегрованої системи та мережі;

Широке застосування інформаційних технологій у більшості сфер життєдіяльності людей робить завдання захисту інформації, інформаційних ресурсів, каналів передачі даних від злочинних дій злочинців досить актуальним.

З розвитком технологій електронних платежів і документообігу існує висока загроза втручання сторонніх осіб, що мають на меті завдати шкоду бізнесу, що призведе до великих збитків. Не дарма захист даних у комп'ютерних мережах постає однією з найвагоміших проблем.

Застосування систем електронного документообігу (СЕД) може дати значний економічний ефект [6]. Але при впровадженні СЕД не слід нехтувати безпекою системи. Одна з найвагоміших вимог до кожної СЕД є гарантування безпеки електронного документообміну.

Основні загрози безпеки СЕД називають:

- загрозами цілісності інформації;
- загрозами конфіденційності;
- загрозами продуктивності системи;
- загрозами доступності.

Важливе значення має регулювання (регламент) документообігу, що пов'язано з електронними документами.

Регулювання документообігу – це сукупність визначених законом, нормативно-правовими актами чи договорами правил інформаційного застосування суб'єктами інформаційних відносин. Регулювання документообігу ідентифікує ролі та права суб'єктів при створенні, володінні, використанні та розпорядженні документами, порядку створення та запису інформації на носіях.

Посилаючись на статтю 6 Закону України «Про електронні документи та електронний документообіг» відзначимо, що електронний підпис є неодмінною умовою електронного документа (ЕД), що знайшов застосування у процесі ідентифікації автора та/або особи, що підписує ЕД іншими суб'єктами електронного документообігу [7].

Електронний підпис – інформація в електронній формі, що додається до решти електронних даних або логічно пов'язана з ними та призначається для

ідентифікації особи, що підписує ці дані [8]. Однак електронний цифровий підпис (ЕЦП) – це вид електронного підпису, одержаного шляхом криптографічного перетворення набору електронних даних, який логічно додається до цього набору або об'єднується з ним і дає змогу підтвердити його цілісність та ідентифікувати підписанта. На електронний цифровий підпис накладається особистий ключ і перевіряється відкритим ключем.

Захист ЕЦП від відтворення або фальсифікації заснований на використанні криптографічних методів у відповідних технологіях. Так, у разі використання алгоритму генерації та перевірки електронного цифрового підпису з довжиною ключа 264 біти, час, необхідний для її можливого «злому» за допомогою сучасного криптоаналізу за допомогою комп'ютера з частотою процесора 3 ГГц, експерти оцінюють майже в 1000 років [9].

Автентифікація – це процес встановлення того, що користувач має інформацію в системі про надані ним облікові дані.

Ідентифікація – процедура розпізнавання користувача в системі, частіше за все, за допомогою заздалегідь визначеного імені (ID) або іншої відомої інформації про нього, що сприймається системою.

З метою організації захисту інформації системи електронного документообігу необхідно приділяти додаткову увагу забезпеченню доступності публічної інформації та блокуванню несанкціонованого доступу. Головне в організації захисту СЕД не технічні засоби, а лояльність користувачів. Як тільки документ потрапляє до користувача, конфіденційність цього документа по відношенню до користувача порушується. Технічні заходи, в принципі, не можуть запобігти витоку документа цим користувачем. Він може знайти багато способів копіювання інформації, від збереження на зовнішніх носіях до банальної фотографії. Реєстрація дій користувача є важливим моментом захисту в електронному документообігу. Його правильна реалізація в системі дозволить відстежити всі протиправні дії та знайти винного, а в разі хірургічного втручання навіть припинити спробу протиправних чи шкідливих дій [10].

Створення захисту електронного документообігу повинно бути комплексним. Потрібно оцінити можливі загрози та ризики СЕД та розміри можливих втрат через загрози. Захист СЕД не обмежується захистом документів та обмеженням доступу до них. Важливими є питання захисту системного обладнання, персональних комп'ютерів, принтерів та інших пристроїв; захист мережевого середовища, в якому працює система, захист каналів передачі даних і мережевого обладнання, можна виділити СЕД до спеціального сегменту мережі. На кожному рівні захисту важливу роль відіграє комплекс організаційних заходів. Погана організація може звести нанівець усі технічні заходи, якими б досконалими вони не були. У будь-якому випадку слід впровадити елементарний і не менш ефективний засіб – доступ до системи документообігу повинен здійснюватися через систему паролів з обмеженим рівнем доступу. Фізичний доступ до приміщень, де встановлено систему документообігу, повинен відповідати правилам внутрішнього розпорядку та бути зарезервованим для третіх осіб.

1.2 Можливості авторизації користувача в сучасних інформаційних системах

Одним з основних факторів, що визначає стан безпеки системи інформаційної інфраструктури, є ефективність системи управління та забезпечення доступу користувачів і захисту інформації, що в ній зберігається. Необхідні серйозні заходи щодо захисту секретної та цінної інформації від несанкціонованого доступу, щоб уникнути значної матеріальної та нематеріальної шкоди.

Основним завданням у проблемі захисту інформації в інформаційних системах від обмеженого доступу є завдання розмежування функціональних повноважень [11].

Завдання полягає в тому, щоб не дати зловмиснику прочитати або змінити збережену інформацію. Дії щодо захисту інформації від несанкціонованого доступу включають:

- запобігти проникненню зловмисника на IP на основі засобів розпізнавання користувача;

- створення спеціального програмного забезпечення для захисту інформації;

- використання спеціальних засобів захисту інформації від несанкціонованого доступу.

Поняття авторизації та автентифікації різні, проте вони йдуть рука об руку і використовуються разом, тому більшість користувачів їх не розрізняють.

Авторизація набула актуальності в процесі посилення впливу людського фактору на працездатність автоматизованих систем та просування дистанційних сервісів, через Інтернет, що потребує ідентифікації користувача, зацікавленого у використанні відповідних сервісів.

Авторизація – це процес доведення прав на виконання деяких операцій – зміни даних. Вона необхідна для задоволення безпеки при реалізації різних дій, для розмежування прав користувачів і для захисту від зловмисників [12].

Авторизація знайшла застосування на сайтах, в банкоматах, інтернет-магазинах, державних установах. Як правило, користувач має ввести своє ім’я користувача та пароль. При правильно введених даних користувачеві надається право увійти в систему для виконання дозволених маніпуляцій. Якщо виникає помилка, з’єднання з системою не встановлюється.

Помилка авторизації – неправильний логін, пароль та інші дані. При введенні кодового слова користувач повинен звернути увагу на правильний порядок символів, регістр та розкладку, встановлену на клавіатурі. У разі помилкової авторизації система блокує доступ користувача і може провести наступні операції:

- усунути факт несанкціонованого доступу;

- подати звуковий або світловий сигнал, відображення повідомлень на екрані;

- обмежити доступ на деякий період часу;

- запропонувати повторне введення коду;

- здійснити відновлення паролю;

- блокувати облікового запису.

Існують різні типи дозволів, які поділяються на три класи:

Дискреційний контроль доступу – доступ до об'єктів, даних або функцій, наданий явно зазначеним сутності, користувачам чи групам користувачів. До прикладу, «користувач_1» має право читати «файл_1», але не має права змінювати його. Кожен об'єкт пов'язаний з суб'єктом – власником, який встановлює права доступу до об'єкта. У системі також є виділений об’єкт, суперкористувач, який має право визначати права доступу для всіх об’єктів, і будь-який об’єкт може передати свої права іншим користувачам. Такий доступ застосовують в сучасних операційних системах, де для авторизації використовуються права доступу та списки контролю доступу (ACL) [13].

Мандатний контроль доступу – це поділ інформації за ступенем конфіденційності та користувачів за рівнями доступу до цієї інформації. Основною перевагою обов'язкового доступу є обмеження прав користувача об'єкта. Права суб'єктів на створені ними об'єкти залежатимуть від рівня їхнього доступу. Відповідно, вони не зможуть видавати їх випадково або спеціально неавторизованим користувачам [14].

Рольовий контроль доступу – це розробка політики вибіркового доступу, де доступ до об’єктів системи формується з урахуванням особливостей їх використання залежно від ролі суб’єктів у певний момент часу. Ролі дозволяють користувачам зрозуміти правила розмежування доступу. Роль поєднує в собі властивості вибіркового контролю доступу, визначаючи об'єкти на основі суб'єктів. Коли ви змінюєте ролі, змінюється доступ до файлової групи. Цей тип доступу більш гнучкий, ніж попередні, і може їх моделювати.

1.3 Дослідження існуючих методів, що застосовуються для поставленої задачі

Розглянемо методи, які застосовують для авторизації користувача з використанням клавіатурного почерку.

Метод на основі нейронної мережі вперше був використаний Обайдатом і Маккіароло для автентифікації та ідентифікації користувачів за часом між натисканнями клавіш [15]. Вони досягли 96,8% точності ідентифікації оператора за допомогою нейронної мережі суми роботи. Йонг і його колеги запропонували використовувати динамічну нейронну мережу. За результатами аналізу даних дослідження було зроблено висновок, що головною перевагою нейронних мереж є те, що вони можуть обробляти декілька параметрів рукописного введення. Виділено основні недоліки використання нейронних мереж для розпізнавання клавіатури: процедури навчання та автентифікації займають тривалий час, наявність ситуацій, коли нейронна мережа не може навчатися через особливості вхідної вибірки. Оскільки цей метод використовується лише як «чорний ящик», неможливо визначити достатньо для отримання шаблону почерку та подальшої успішної операції обсягу та складу вхідного зразка. Крім того, якщо ви додаєте нову модель користувача в систему, вам доведеться перенавчати всю нейронну мережу.

Для методу розпізнавання візерунків Giot та його колеги запропонували використовувати метод опорного вектора для розпізнавання клавіатури. Вони досягли рівня ідентифікації 95% [16]. Основною перевагою цього методу виявляється висока точність, що обумовлено ігноруванням помилок і помилок у вимірюванні часу утримання ключа. Недоліком цього методу є те, що ідентифікація проводиться не на всіх шаблонах, а лише на тій частині шаблонів, яка знаходиться на межах.

Реветт та інші, використовуючи генетичні алгоритми, отримали ймовірність помилки FAR 0,43% і FRR 4,75% [17]. Виявлено головну перевагу використання генетичних алгоритмів – вони можуть легко взаємодіяти з

великими базами даних та обробляти багатовимірні, недиференційні, безперервні та непараметричні дані. Основними недоліками генетичних алгоритмів є висока: висока складність, що обмежує діапазон, низька стійкість до відхилень часових характеристик почерку, цей метод не гарантує знаходження оптимального рішення.

У деяких випадках використовується метод автентифікації пароля з розширеним методом біометричної автентифікації, заснованим на параметричному навчанні класифікатора [18]. Вектор біометричних характеристик визначається шляхом обробки даних, отриманих від користувача з використанням розрахунку математичних очікувань параметрів отриманих векторів і подальшого розрахунку елементів матриці коваріації. Потім визначаються дискримінантна функція та оптимальний коефіцієнт Стьюдента, які визначають значення допуску, отримане експериментальним методом для N користувачів, які тестують метод.

Також використовується біометричний метод автентифікації користувача за допомогою клавіатури на основі декомпозиції Хаара та близькості Хеммінга [19]. Кількість доданків у розкладанні рядів Фур'є визначає похибку цього методу, на основі якої на етапі експериментального дослідження пропонується вибрати максимальну довжину пароля. Розкладання вектора на ряд Фур'є не є оптимальним з ряду причин, тому пропонується використовувати функції Хаара, які утворюють повну періодичну ортонормовану систему непарних функцій, які мають чутливість як до локальних, так і до глобальних. Після вибірки з часом отриманої функції користувача та її перетворення на основі декомпозиції Хаара визначається вектор бажаних біометричних параметрів. Запропонований метод розрахований на максимальну довжину парольної фрази для зменшення помилок введення, є гнучким, а точність цього методу визначається кількістю коефіцієнтів розширення, які залежать від довжини парольної фрази.

Існує метод гістограмного розпізнавання клавіатурного почерку [20]. Запропоновано сформувати вектор біометричних ознак на основі тривалості подій очікування та паузи шляхом перетворення їх у r-вимірний вектор

біометричних параметрів. Для навчальної вибірки визначається діапазон варіації кожного з компонентів і повертається до початку координат. Вони в свою чергу визначають координати r-вимірної зменшеної області векторного розподілу. Після розподілу області на субрегіон будується оцінка щільності векторів і порівнюється з щільністю векторного розподілу навчальної вибірки за законом на основі формування відношення правдоподібності. До переваг цього методу можна віднести його простоту і зрозумілий фізичний сенс, недоліки – відсутність оптимального способу поділу території на субрегіони, неможливість досягти конвергенції за рахунок збільшення розміру набору навчання, необхідність введення парольної фрази до створити необхідну кількість біометричних векторів даних для автентифікації. Найбільш цікавими є методи та математичні моделі моніторингу прихованої клавіатури під час тривалого використання клавіатури.

Однією з останніх розробок у біометричній автентифікації натискання клавіші є метод, заснований на використанні теорії ймовірності та математичної статистики для оцінки часу очікування та затримки ключа як оператора характеристичного введення [21]. Розкритий метод розпізнавання рукописного тексту на клавіатурі шляхом введення вільного тексту на основі механізму аналізу даних клавіатури. Цей метод реалізований в алгоритмі розпізнавання рукописного введення клавіатури на основі часу утримання клавіші та часу введення послідовностей букв (N-грам), які часто використовуються в рукописному введенні. Система автентифікації, розроблена для оператора системи інформаційної інфраструктури, має точність 98%, якщо в системі зареєстровано 100 операторів. Також доведено, що за допомогою методу визначення почерку за допомогою клавіатури на основі вмісту клавіш для запису часу стає можливим визначити почерк клавіатури з вільного тексту. Метод не враховує інтервал часу між натисканнями клавіш, що іноді підвищує точність визначення через перекриття сусідніх клавіш.

1.4 Аналіз відомих програмних реалізацій для авторизації користувача з використанням клавіатурного почерку

Одним із найвідоміших комерційних продуктів у сфері авторизації користувача з використанням клавіатурного почерку є технологія BioPassword, розроблена Net Nanny Software. Ця технологія заснована на запатентованому алгоритмі, який дозволяє створити індивідуальний шаблон на основі аналізу часових інтервалів після натискання і відпускання клавіш. Програма дає приголомшливі результати: навіть ввівши пароль з невеликої кількості символів, можна з високою ймовірністю визначити користувача.

BioPassword має довгу історію. Розвиток технології розпізнавання клавіатури активно розвивався в ряді дослідницьких центрів США в 1980-х рр. У 1993 р. корпорація Treasure Corporation (яка пізніше стала Net Nanny Software International, Inc.) придбала права на технології, придбала патенти, комерціалізувати та розробити технологію і перетворили її на товарний продукт.

Зараз Net Nanny Software, Inc. виробляє програмне забезпечення під брендом BioPassword. Перша комерційна реалізація цієї технології включена в продукт BioPassword LogOnd для Windows NTv. BioPassword LogOn для Windows NT – це біометрична система, спеціально розроблена для серверної платформи Windows NT і підвищує надійність доступу паролем до мережі. Windows NT пропонує високий рівень захисту, який, однак, передбачає, що користувач зберігає своє ім'я користувача та пароль у таємниці, але на практиці пароль іноді стає відомим хакеру. BioPassword пов’язує «портрет» клавіатури користувача з логіном і паролем, тому якщо пароль і логін збігаються з перевіреними, а написання на клавіатурі не відрізняється, система відмовляє . BioPassword LogOn для Windows NT використовує стандартну клавіатуру, не вимагаючи жодного додаткового обладнання або процедури авторизації [22].

Оскільки рішення засноване на програмному забезпеченні, його легко реалізувати онлайн. Для мережевих адміністраторів це програмне забезпечення дозволяє збільшити термін служби логіна та пароля, крім того, не потрібно часто

змінювати пароль. В принципі, технологію BioPassword можна використовувати в будь-якій системі, яка вимагає керування записом з клавіатури.

Ще одним рішенням щодо захисту інформації щодо несанкціонованого доступу за допомогою авторизації користувача розглянемо менеджер паролів LastPass. LastPass – це незалежний додаток для автентифікації, доступний для пристроїв Android та iOS [23]. Ця програма забезпечує ефективну двофакторну автентифікацію з найбезпечнішим сервісом. Крім того, за допомогою цієї програми ви можете захистити необмежену кількість облікових записів. Якщо у вас уже є обліковий запис LastPass, установити та активувати цю програму буде легкою справою.

Важливі особливості програми включають полегшення керування обліковим записом завдяки LastPass; можливість створення резервної копії своїх облікових записів автентифікації у своєму LastPass Vault; простій функції перевірки в один дотик, здатній заощадить ваш час; автоматичному налаштуванню процесу автентифікації за допомогою сканера QR-коду; додаванню кількох облікових записів; надання 6-значного коду, що займає лише 30 секунд.

Плюсом програми є те, що користувачі отримують функції для всіх інших сторонніх служб і програм із підтримкою TOTP. Однак мінусом є його погана сумісність з Apple Watch.

Іншими програмами автентифікації на основі пасивної біометрії розглянемо програми TypingDNA, BehavioSec, Banking with Keystroke Authentication та BioCatch-Auth.

TypingDNA – це вбудований механізм, заснований на елементах штучного інтелекту, здатний розрізняти два шаблони введення з безпрецедентною точністю [24]. Надається інтерфейс для використання автентифікації на основі аналізу натискання клавіш, який забезпечує: безпеку під час входу в систему або примусового скидання пароля; виявлення порушників; впровадження онлайн-біометричної автентифікації для аналізу поведінки користувачів; виконувати

багатофакторну автентифікацію; здійснювати ідентифікацію користувача; запобігати діям шахраїв.

BehavioSec – це продукт шведської компанії, яка спеціалізується на системах довгострокової автентифікації [25]. Це програмне забезпечення відстежує діяльність користувача, щоб переконатися, що власник працює за комп’ютером. Використовується не тільки аналіз динаміки натискання клавіш, а й використання комп'ютерної мишки (тачпада).

Banking with Keystroke Authentication – це відкрита реалізація пасивної біометрії, представлена стандартною сторінкою банківського сайту та аналізом динаміки натискань клавіш [26]. З огляду на той факт, що характеристики введення різних людей унікальні, користувачі автентифікуються на основі вилучення шаблону стилю введення пароля.

BioCatch-Auth – це програмне рішення, розроблене на основі підходу пасивної біометрії [27]. Платформа створює профілі поведінки користувачів, щоб розпізнавати широкий спектр різних загроз кібербезпеці, включаючи зловмисне програмне забезпечення, троянські програми віддаленого доступу (RAT) і роботу роботів (ботів).

BioCatch-Auth надає такі функції:

– перевірка особи – аналіз різних вимірів методів введення інформації (вільне використання додатків, можливості перегляду) і порівняння даних для ідентифікації використання вкрадених або штучних облікових даних під час заповнення онлайн-заявок;

– довгострокова автентифікація – вибір 20 унікальних характеристик з бази даних метрик профілю користувача для аналізу поведінки під час сесії;

– запобігання шахрайству – за допомогою підходу «прихованого спостереження» можливі загрози ідентифікуються та сповіщення про виявлені підозри відбуваються в режимі реального часу з мінімальною кількістю помилкових тривог.

Аналіз відомих програмних рішень для авторизації користувача показав досить високі результати за показником точності встановлення автентичності

пред’явленого ідентифікатора, однак також виявлено недоліки методів, що застосовуються в розглянутих програмах. Серед таких недоліків виділено наступні: залежність від змін психоемоційного стану користувача, що визначає наявність збоїв у роботі системи в майбутньому; відсутність можливості перевірити процес встановлення автентичності користувача на основі заявленого ним ідентифікатора; відсутність можливості встановити факт зміни користувача на робочому місці.

1.5 Висновок до розділу 1

У даному розділі розкрито аналіз загроз в автоматизованих системах та досліджено можливості авторизації користувача в сучасних інформаційних системах. Досліджено різні методи вирішення поставленої задачі та виділено їх основні переваги та недоліки. Проведено аналіз відомих програмних реалізацій для авторизації користувача з використанням клавіатурного почерку, що дозволили виявити недостатню точність авторизації користувача з використанням клавіатурного почерку та доводить актуальність досягнення поставленої мети.

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОРИЗАЦІЇ КОРИСТУВАЧА З ВИКОРИСТАННЯМ КЛАВІАТУРНОГО ПОЧЕРКУ

2.1 Аналіз клавіатурного почерку в процесах автентифікації, авторизації та ідентифікації користувача

Використання методів поведінкової біометрії на основі клавіатурного письма, на відміну від методів фізіологічної біометрії, не вимагає придбання додаткових пристроїв. Щоб отримати зразок написання на клавіатурі, потрібна лише звичайна стандартна клавіатура. Це робить цей метод недорогим і непомітним для користувача і може застосовуватися приховано, що дозволить покращити існуючі комп'ютерні системи для забезпечення інформаційної безпеки.

Слід зазначити, що в основі досліджень клавіатурного почерку лежить праця користувачів телеграфу. У середині 19 століття, коли телеграф був широко поширений, було помічено, що телеграфісти можуть ідентифікувати інших операторів за їх швидкістю набору. Метод «Кулак відправника» використовувався під час Другої світової війни для ідентифікації відправника повідомлення за частотою, темпом і часом натискання телеграфних клавіш. Браян і Хартер провели серію експериментів на тридцяти семи операторах телеграфу з різним ступенем навичок друку. Вони зазначили, що телеграфісти вміли впізнавати інших операторів за стилем набору [28].

На початку 1980-х років Національний науковий фонд і Національне бюро стандартів у Сполучених Штатах провели дослідження, які виявили, що зразки рукописних клавіатур містять унікальні особливості, які можна ідентифікувати. Шаффер показав, що нанесення ударів є запрограмованим навиком і що ударні рухи організовуються до того, як вони фактично виконуються.

У працях Іванова А.І. [29] зазначається, що при наборі тексту на клавіатурі однією рукою задіяно близько 50 м'язів пальців і передпліч і близько 20 м'язів плеча і плечового пояса, тобто при наборі двома руками людина задіює близько

140 м'язів. Таким чином, почерк людини є поведінковою характеристикою, яка розвивається з часом і тому не може бути змінена, втрачена чи забута. У будь-якій поведінковій біометричній характеристиці можна спостерігати значні зміни характерних ознак. Однак вони повинні надавати достатньо інформації, щоб ідентифікувати та визначити автентичність особи за зразком почерку. Купер був першим дослідником друкарських машинок, який розділив процес друку на чотири етапи: сприйняття тексту людиною; збережіть в пам'яті; людина перетворює збережені в пам'яті символи в команди, що передаються м'язам; прямий ввід із процедурою зворотного зв’язку, необхідною для перевірки точності введення.

Salthouse удосконалив і доопрацював запропоновану модель процесу введення тексту Купера [30]. Butsch визначив, що «інтервал копіювання», тобто обсяг тексту, який зберігається в пам’яті користувача під час набору тексту, залежить від досвіду та навичок користувача. Використання пам'яті як короткочасного буфера перед друком було доведено експериментами, проведеними Томасом і Джонсом. У своєму дослідженні Купер продемонстрував, що користувачі розбивають текст на передбачувані малі групи через обмеження розміру буфера. У своїх експериментах Вервей і Дронкерт довели, що процеси сприйняття прочитаного тексту і рухи м'язів при друкуванні відбуваються одночасно. У дослідженні, проведеному Шаффером, було запропоновано існування внутрішнього регулярного ритму набору певних символьних послідовностей. Він припускав, що письмо на клавіатурі не є постійною характеристикою людини, а постійно змінюється. Він також зазначив, що процес друку базується на знанні переходів і рухів між клавішами.

Інтервали між послідовними натисканнями клавіш для досвідчених користувачів, як показали спостереження, були коротшими, ніж для досвідчених користувачів. Р. С. Гейнс, В. Лісовскі, С. Дж. Пресс і Н. Шапіро [31] показали, що швидкість руху пальців у досвідчених друкарок вдвічі вище, ніж у недосвідчених. Острі, досліджуючи синхронізацію рухів руки при наборі,

показав, що організація руху пов'язана з поточним психофізіологічним станом людини.

На основі аналізу перерахованих вище досліджень пропонується розглядати письмо на клавіатурі як індивідуальне та унікальне для кожної людини. Румельхарт і Норман зробили комп'ютерне моделювання дій користувача, що працює з клавіатурою. Вони змоделювали часовий інтервал між натисканнями клавіш і помилками повторного натискання клавіш. Моделі Купера, Солтхауса, Румельхарта та Нормана є першими моделями, розробленими для аналізу властивостей і характеристик написання користувача на клавіатурі.

2.2 Обґрунтування вибору методу авторизації користувача з використанням клавіатурного почерку

При наборі тексту користувачем, можна відзначити три події: коли клавішу натиснуто, утримується та коли клавішу відпущено. Часові характеристики почерку розраховуються на основі вибірки моментів натискання та утримання клавіш. Виділено такі характеристики клавіатурного запису, які використовуються в системах ідентифікації та автентифікації оператора:

- інтервал часу між натисканнями клавіш, тобто час між натисканням клавіші та моментом натискання наступної клавіші або час між моментом відпускання клавіші та моментом натискання наступної. Використання цієї характеристики вимагає тривалого збору статистичних даних. Підраховано, що для збору даних про час, що минув між натисканням 33 клавіш, що відповідають буквам українського алфавіту, з вибіркою 30 елементів необхідно 32670 одиниць вхідних даних. Таким чином, ця характеристика почерку не може бути використана для виявлення того, що видає себе за легального користувача.

- час введення біграм (наприклад, «АА», «ОЕ», «ЯЯ»), триграм (наприклад, «ААА», «МАМ», «ЭЮЯ»), тобто час між натисканням першої клавіша і момент відпускання останньої клавіші N - грам. Методика аналізу цієї характеристики,

як і попередньої, вимагає збору великої кількості статистичних даних. Це виправдовує неадекватність цієї характеристики для виявлення підміни користувачів.

- час утримання клавіші, тобто час між натисканням і відпусканням клавіші. Кількість відліків статистичних даних часу утримування відповідає кількості натискань клавіш користувачем, тобто 33 відлікам часу утримання клавіші відповідають 33 літери російського алфавіту. Така характеристика почерку дозволяє організувати процедуру постійного негласного спостереження за клавіатурою особи користувача з метою виявлення підміни.

- сила натискання на клавішу під час набору тексту. Потрібна установка додаткових датчиків (по одному на кожну клавішу) на клавіатурі. Це призводить до збільшення складності розробки клавіатур, збільшення їх вартості та збільшення ймовірності пошкодження клавіатури.

- швидкість, тобто кількість символів, введених користувачем за певний проміжок часу, і швидкість введення. Встановлено, що ця характеристика почерку залежить від психофізіологічного стану користувача і тому часто змінюється. Це виправдовує складність застосування цих характеристик у процесі виявлення заміни законного користувача [32].

Функція опису процесу набору тексту користувачем матиме загальний вигляд:

$$\vartheta(t) = \gamma(t) + \theta(t) + \lambda(t), \qquad (2.1)$$

де $\gamma(t)$ – складова, яка характеризує підсвідомі процеси мислення при введенні тексту; $\theta(t)$ – складова, яка характеризує свідомі процеси мислення; $\lambda(t)$ – складова, що характеризує механічні характеристики клавіатури, які впливають на процес введення тексту.

Основним завданням системи біометричної ідентифікації користувача за характеристиками почерку клавіатури є завдання виділення та подальшої ідентифікації компоненти $\gamma(t)$ функції $\vartheta(t)$, яка визначає вихідні дані для

ідентифікації системи після вимірювання часових характеристик користувача. почерк клавіатури. Для цього з вихідної функції $\vartheta(t)$ необхідно виділити компоненти $\theta(t)$ і $\lambda(t)$. Зрозуміло, що через неможливість побудови механічної моделі рухів людини під час набору тексту єдиним прийнятним рішенням є збір статистичних даних про використання клавіатури великою кількістю користувачів.

Ймовірність автентифікації користувача на основі часу утримання клавіші порівняно з довжиною ключової фрази є набагато більш стабільною характеристикою операторської клавіатури, ніж час між натисканнями клавіш, який збільшується зі збільшенням довжини ключової фрази. Це пояснюється тим, що процес натискання клавіші на клавіатурі є справді підсвідомим процесом мислення.

Характер цієї функції практично не змінюється для широкого кола операторів, незалежно від їх кваліфікації та досвіду роботи з клавіатурою. З цього випливає, що складова $\gamma(t)$ найточніше характеризує час утримання клавіші під час набору тексту користувачем. Час утримання клавіші розраховується за формулою (2.2):

$$T_i^y = T_i^v - T_i^n, \tag{2.2}$$

де T_i^y – час утримання клавіші; T_i^v – час відпускання клавіші; T_i^n – час натискання клавіші.

За результатами аналізу почеркописних характеристик клавіатури запропоновано використовувати час утримання клавіші в процесі виявлення підміни легітимного користувача. У моделі почерку користувача пропонується зберігати середні значення часу утримання ключа.

Запропоновано спосіб розпізнавання рукописного тексту на клавіатурі за часом утримання клавіші та інтервалом між натисканнями. Це дозволить отримати шаблон рукописного тексту, який не залежить від набраного тексту та порядку введення символів. Таким чином стає можливим визначити почерк

користувача ключа системи за допомогою вільного контрольного тексту. Це відкриває можливість застосування методу в задачах постійного моніторингу прихованої клавіатури з метою виявлення підміни авторизованого легального користувача, визначення відхилення психофізіологічного стану користувача ключової системи від нормального.

2.3 Розробка математичної моделі клавіатурного почерку

Аналіз запису на клавіатурі базується на припущенні, що запис на клавіатурі представлений у вигляді середніх значень подій на клавіатурі [33]. Існує три типи подій клавіатури в системах Microsoft Windows:

Подія KeyDown, що відбувається одноразово. Спрацьовує, коли натискається фізична клавіша. Подія нижнього рівня – реагує на натискання будь-якої клавіші на клавіатурі. Повертає код натиснутої клавіші.

Подія KeyUp, яка виникає один раз після того, як користувач відпускає фізичний ключ. У всьому іншому подія схожа на KeyDown.

Подія KeyPress, яка може відбуватися кілька разів, коли користувач утримує клавішу. Ця подія відбувається, коли натискається клавіша, що призводить до введення символу.

У системах розпізнавання клавіатурного почерку статистичними даними є часові значення подій на клавіатурі. Обраною характеристикою запису з клавіатури є час утримання клавіші, який є інтервалом часу між подіями KeyDown(A) і KeyUp(A), де A – одна з клавіш на клавіатурі. У рамках використання цього методу необхідно зібрати статистику, що складається з вибірки тимчасових значень, де елементом вибірки буде час утримання ключа.

У ймовірнісному статистичному формулюванні необхідно побудувати усереднені статистичні моделі на основі вибірок, представлених системі в режимі навчання. При цьому слід враховувати, що на характеристики клавіатурного письма людини впливає багато факторів: програмні та апаратні затримки, які також є випадковими величинами, рух нервових імпульсів по

нейронах, час відгуку людини м'язи на сигнал, який посилає мозок, та ін. Таким чином, на написання на клавіатурі впливає багато незалежних випадкових змінних. Дія їх складу описується формулою Гаусса. Таким чином, щоб зменшити вплив випадкових похибок, необхідно кілька разів вимірювати досліджувану величину.

За допомогою формули Гаусса здійснимо обробку результатів вимірювання характеристик клавіатурного почерку. Нехай вимірюється час утримання певного ключа, позначеного X. Здійснивши необхідні вимірювання отримаємо вибірку значень величини (2.3):

$$X_1, X_2, X_3 \dots X_N \tag{2.3}$$

Отриманий ряд значень X характеризуватиме вибірку часу утримання клавіші. За даною вибіркою оцінюється результат вимірювання. Величину, що позначає оцінку, позначають $\bar{X}$. Для подальших обчислень необзідно оцінити також помилку вимірювання, що позначається ΔX. Після цього результат вимірювання матиме вигляд:

$$\mu = \bar{X} \pm \Delta X. \tag{2.4}$$

Отже, для вибірки $X_1, X_2, X_3 \dots X_N$ необхідно визначити оцінку результату вимірювання $\bar{X}$, помилку ΔX та надійність P. Це здійснюється завдяки використанню математичної статистики. Оцінку результатів вимірювань $\bar{X}$ визначимо як:

$$\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N} \tag{2.5}$$

де N – кількість вимірювань.

Тобто для вибірки, що має N вимірювань, найбільш ймовірно значення вимірюваної величини буде дорівнювати її середньому арифметичному значенню. Середньоквадратична помилка визначатиметься наступним чином:

$$S_{\bar{X}} = \sqrt{\frac{\sum(\bar{X}-X_i)^2}{N(N-1)}} = \frac{S}{\sqrt{N}} \quad (2.6)$$

При збільшенні кількості вимірювань зростає точність оцінки.

За допомогою шаблону клавіатурного почерку здійснюється автентифікація та ідентифікація користувача. Для цього порівнюється поточний зразок почерку та збереженого шаблону. Їх порівняння здійснюється за допомогою використання евклідової відстані:

$$M = \sqrt{\sum_{i=1}^{V}(A_i - B_i)^2}, \quad (2.7)$$

де M – значення евклідової відстані, V – число вибірок часу утримання клавіші, яке дорівнює кількості аналізованих клавіш, A_i – час утримання клавіші поточного зразка почерку, B_i – час утримання клавіші збереженого шаблону.

Користувач буде успішно ідентифікований або його ідентифікація буде підтверджена, якщо розраховані значення евклідової відстані нижче порогу доступу, визначеного в системі. Поріг доступу вибирається виходячи з вимог системи, що розробляється. Основними вимогами до систем захисту інформації є ймовірності виникнення помилок першого та другого типу.

2.4 Проектування структури інформаційної технології авторизації користувача з використанням клавіатурного почерку

Структура інформаційної технології авторизації користувача з використанням клавіатурного почерку характеризується етапами, наведеними на рисунку 2.1.

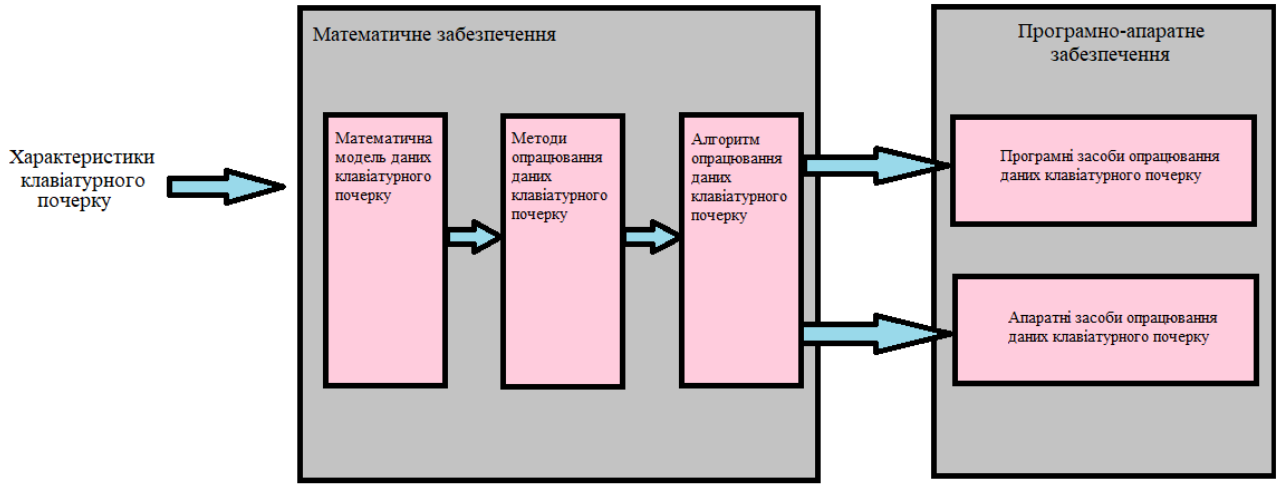


Рисунок 2.1 – Схема етапів інформаційної технології авторизації користувача з використанням клавіатурного почерку

Першим критичним кроком у проектуванні систем автентифікації на основі клавіатурного запису є створення їх математичних моделей, які б адекватно відображали важливі, з точки зору завдань дослідження, аспекти часової структури даних, введених з допомогою клавіатури. Адже саме математична модель значною мірою визначає потенціал і ефективність створюваних інформаційних технологій, визначає структуру програмно-апаратних компонентів проектованої інформаційної системи. Від якості математичної моделі даних суттєво залежить точність і надійність методів їх обробки системою біометричної автентифікації, рівень інформативності та репрезентативність автентифікаційних та ідентифікаційних характеристик, а також достовірність прийнятих рішень.

На основі аналізу запропоновано наступну архітектуру системи виявлення підміни користувача (рис. 2.2), яка забезпечує процес отримання шаблону запису та порівняння записів.

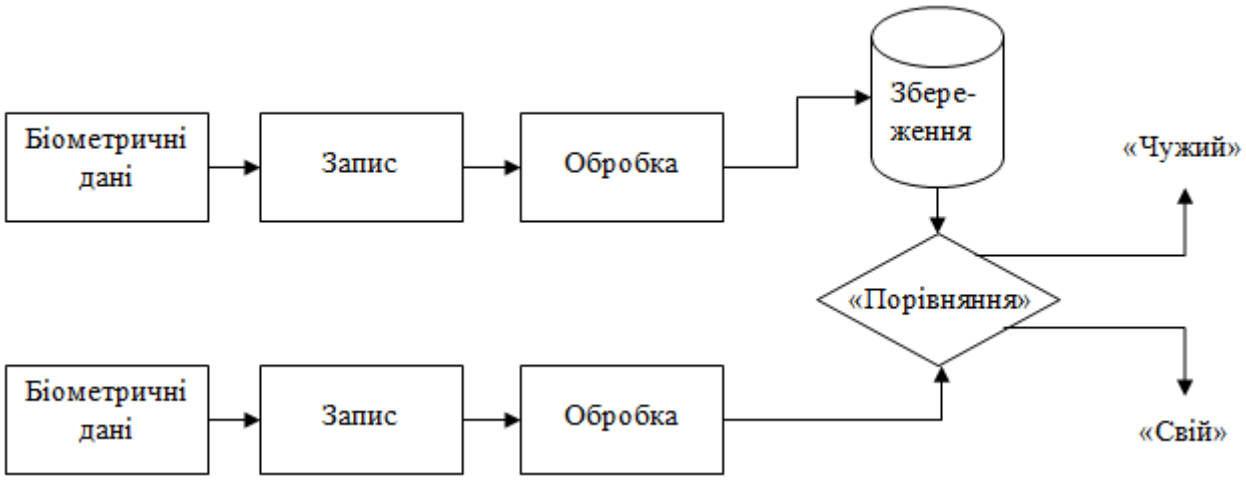


Рисунок 2.2 – Схема архітектури системи виявлення підміни користувача

Як видно на рисунку, система складається з кількох блоків:

- Блок прийому та запису клавіатурного почерку користувача. Він відповідає за отримання часових позначок подій натискання та відпускання клавіш і зазначає, до якої клавіші належать ці події. Для отримання часу настання подій використовується таймер. Цей блок також містить фільтр для довгих натискань клавіш. Важливим моментом є точність визначення моменту виникнення подій клавіатури. Тому системи повинні бути стійкими до помилок вибірки часу.

- Блок обробки вибірки подій клавіатури відповідає за обчислення середніх значень часу утримання клавіш.

- Блок зберігання рукописного тексту клавіатури дозволяє зберігати значення шаблону рукописного тексту користувача в базі даних шаблонів.

- Поточний блок порівняння почерку клавіатури та моделі використовується для порівняння почерків і прийняття рішення щодо автентифікації та ідентифікації користувача на основі результатів порівняння.

2.5 Розробка алгоритму авторизації користувача

На рисунку 2.3 зображено схему алгоритму авторизації з самостійним вводом логіна і пароля.

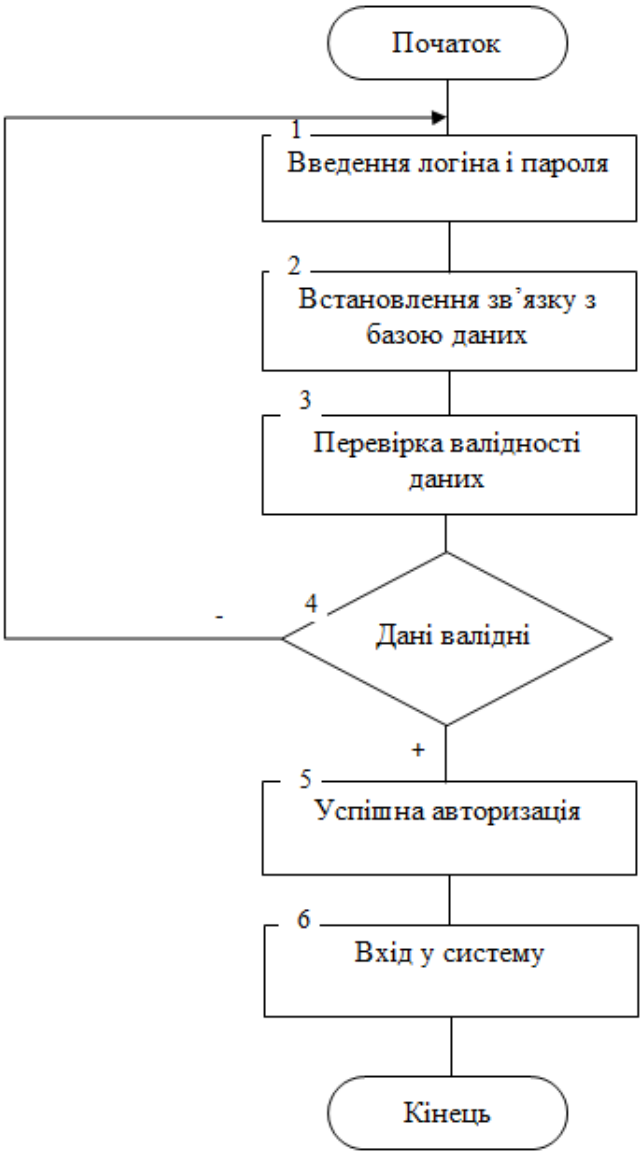


Рисунок 2.3 – Схема алгоритму авторизації користувача з з самостійним вводом логіна і пароля

Схему алгоритму авторизації користувача з використанням клавіатурного почерку наведено на рисунку 2.4.

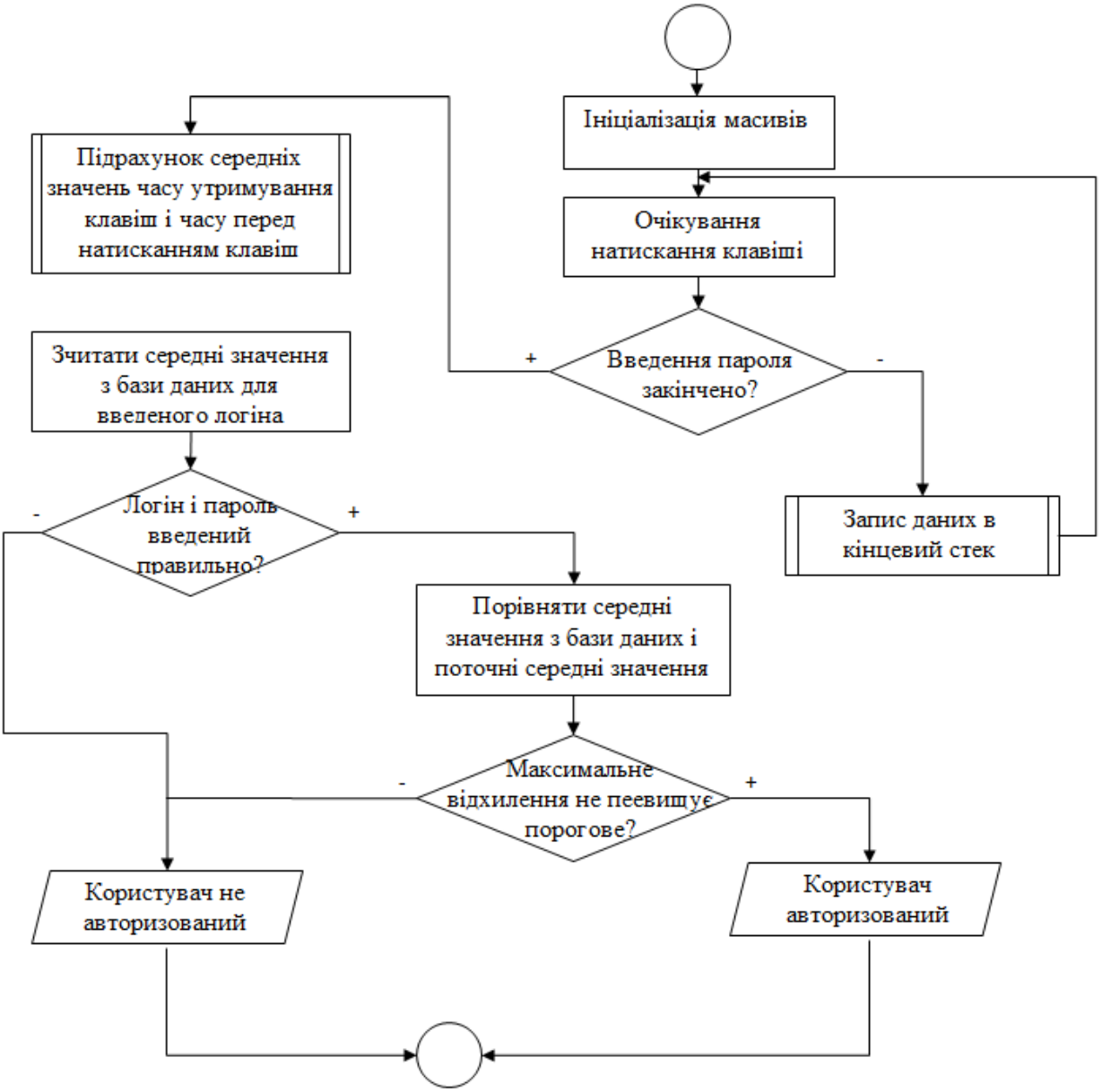


Рисунок 2.4 – Схема алгоритму авторизації користувача з використанням клавіатурного почерку

На рисунку 2.5 зображено схему загального алгоритму інформаційної технології авторизації користувача з використанням клавіатурного почерку.

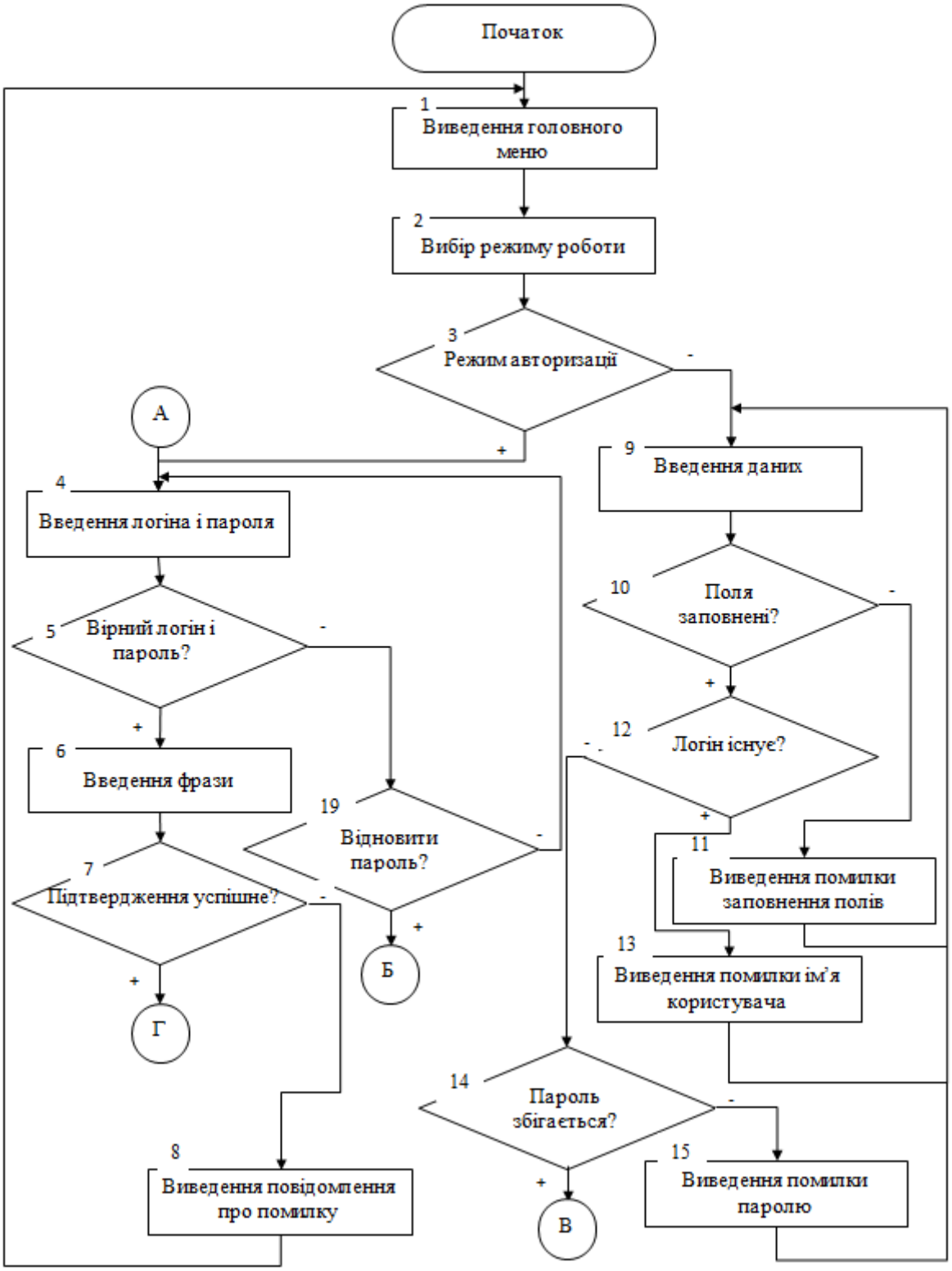


Рисунок 2.5 – Схема загального алгоритму інформаційної технології авторизації користувача з використанням клавіатурного почерку

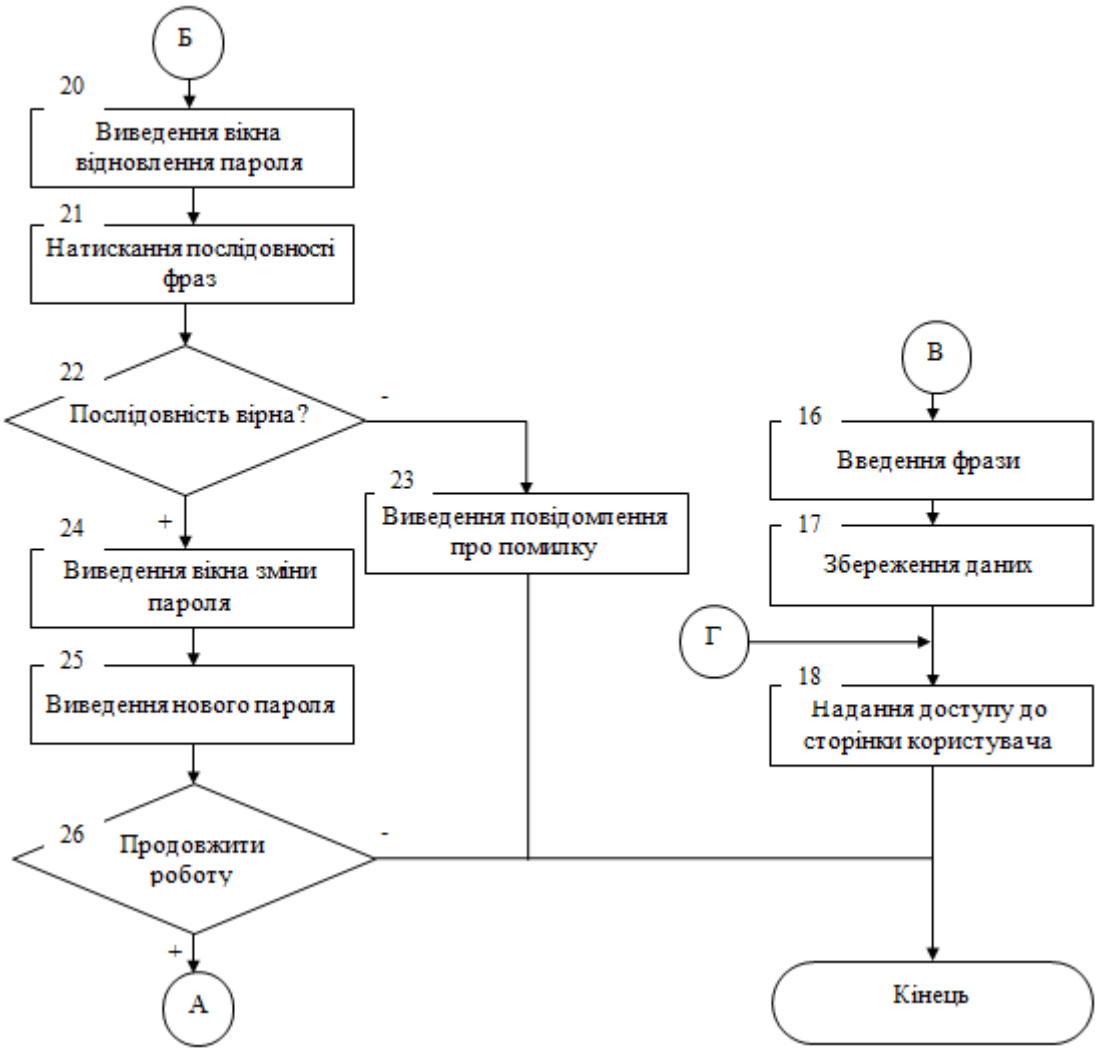


Рисунок 2.5 – Продовження

Даний алгоритм складається з наступних кроків:

Крок 1. Виведення головного меню інформаційної технології.

Крок 2. Вибір режиму роботи.

Крок 3. Перевірка вибору режиму авторизації. У разі вибору іншого режиму перехід до кроку 9.

Крок 4. Введення логіна і пароля.

Крок 5. Перевірка на відповідність логіна і пароля. Якщо перевірка не пройдена, перехід до кроку 19.

Крок 6. Введення фрази для визначення відповідності клавіатурного почерку.

Крок 7. Перевірка проходження підтвердження. При успішному підтвердженні перехід до кроку 18.

Крок 8. Виведення повідомлення про помилку підтвердження. Перехід до кроку 1.

Крок 9. Введення даних – логіна, пароля, повторне введення пароля.

Крок 10. Перевірка заповненості полів. Якщо поля заповнені – перехід до кроку 12.

Крок 11. Виведення повідомлення про помилку заповнення полів. Перехід до кроку 9.

Крок 12. Перевірка логіна на наявність ім’я користувача в базі даних. Якщо такого логіну не знайдено, перехід до кроку 14.

Крок 13. Виведення повідомлення про помилку ім’я користувача. Перехід до кроку 9.

Крок 14. Перевірка на відповідність пароля та повторно введеного пароля. Якщо паролі збіраються – перехід до кроку 16.

Крок 15. Виведення повідомлення про помилку пароля, перехід до кроку 9.

Крок 16. Введення фрази для зчитування клавіатурного почерку.

Крок 17. Збереження даних нового користувача.

Крок 18. Надання доступу до сторінки користувача. Прехід до кроку 27.

Крок 19. Перевірка на необхідність відновлення пароля. Якщо користувач не бажає відновити пароль – перехід до кроку 4.

Крок 20. Виведення вікна відновлення пароля.

Крок 21. Натискання послідовності фраз для підтвердження користувача.

Крок 22. Перевірка послідовності фраз. Якщо послідовність правильна – перехід до кроку 24.

Крок 23. Виведення повідомлення про помилку.

Крок 24. Виведення вікна зміни пароля.

Крок 25. Введення нового пароля та його збереження.

Крок 26. Перевірка на необхідність продовжити роботу. Якщо є необхідність – перехід до кроку 4.

Крок 27. Завершення діалогу з програмою.

На рисунку 2.6 зображено UML-діаграму прецендентів для програми авторизації користувача з використанням клавіатурного почерку.

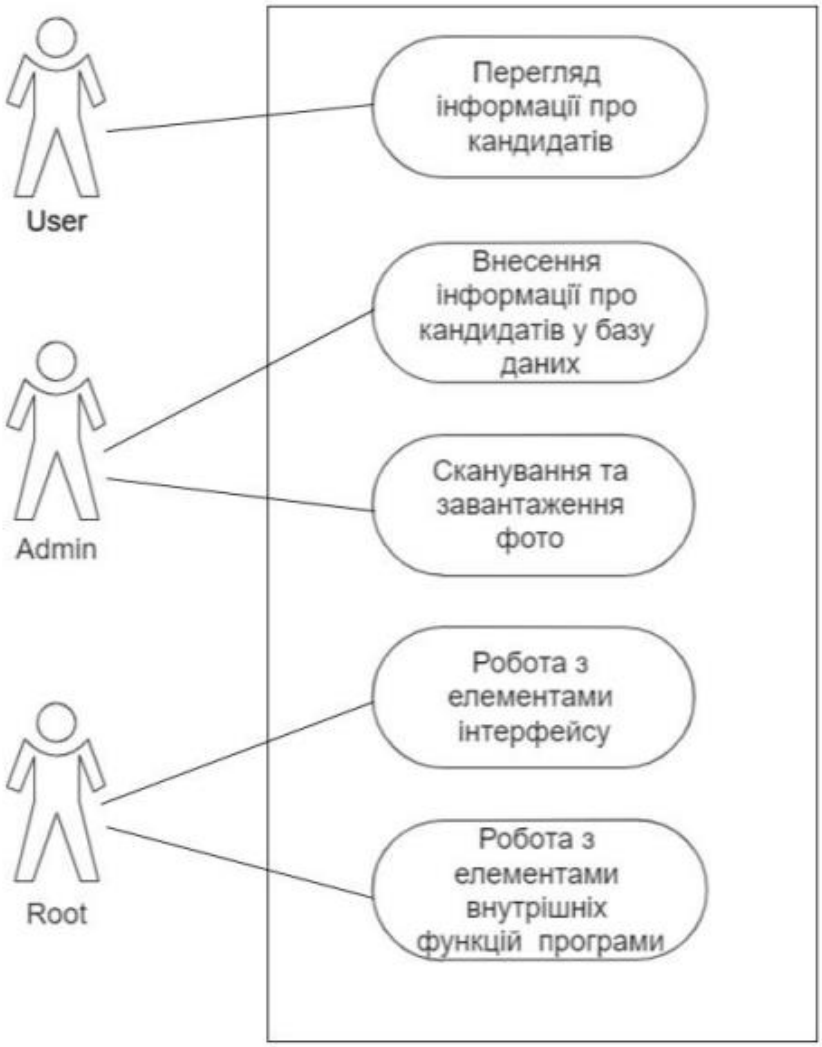


Рисунок 2.6 – UML-діаграма прецендентів для програми авторизації користувача з використанням клавіатурного почерку

Отже, в рамках даного підрозділу розроблено алгоритм авторизації користувача з використанням клавіатурного почерку, а також розглянуто UML-діаграму прецедентів програмного забезпечення.

2.6 Висновок до розділу 2

Проаналізувавши використання клавіатурного почерку в процесах автентифікації, авторизації та ідентифікації користувача було запропоновано розглядати клавіатурний почерк як індивідуальний та унікальний для кожної людини, що дозволить проводити автентифікацію за допомогою біометричних характеристик. Визначено основні події, що впливають на формування індивідуального клавіатурного почерку кожного користувача. Розроблено математичну модель авторизації користувача за допомогою клавіатурного почерку, використовуючи порівняння шаблону та поточного введення за допомогою використання евклідової відстані. Спроектовано структуру інформаційної технології авторизації користувача з використанням клавіатурного почерку, наведено основні етапи інформаційної технології, розроблено архітектуру системи виявлення підміни користувача. Розроблено алгоритм авторизації користувача з використанням клавіатурного почерку, а також розглянуто UML-діаграми прецедентів та розгортування програмного забезпечення.

3 СТРУКТУРНА ОРГАНІЗАЦІЯ ТА ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОРИЗАЦІЇ КОРИСТУВАЧА З ВИКОРИСТАННЯМ КЛАВІАТУРНОГО ПОЧЕРКУ

3.1 Обґрунтування вибору мови програмування

Для розробки інформаційної технології авторизації користувача з використанням клавіатурного почерку необхідно вибрати об’єктно-орієнтовану мову програмування, яка матиме значні переваги для досягнення максимальної ефективності цього завдання. Розглянемо такі мови, як C # та Python та Java.

Спочатку розглянемо мову програмування C #.

C# – це об'єктно-орієнтована мова програмування, що має безпечну систему типів для платформи .NET [34]. Мова була розроблена Андерсом Галесбергом, Скоттом Вілтамутом і Пітером Голдом під егідою Microsoft. Синтаксис мови C# схожий до мов програмування C++ і Java. C# має сувору статичну типізацію, підтримує поліморфізм, перевантаження операторів, посилання на функції-члени класу, атрибути, події, властивості, винятки та коментарі у форматі XML. C#, який дуже відрізняється від своїх попередників – C++, Delphi, Modules і Smalltalk, – виключає певні шаблони, що зумовлюють проблеми при проектуванні програмного забезпечення: так, C# не підтримує спадкування кількох класів (на відміну від C++) або визначення типу (на відміну від Haskell).

Розглядаючи C#, слід виділити наступні переваги:

- використовує проміжний код, що забезпечує запуск на будь-якому комп’ютері, незважаючи на апаратне та програмне забезпечення системи;
- повністю об'єктно-орієнтований – код можна писати тільки для класів;
- має велику кількість доповнень, що полегшують процес написання коду;
- має можливість автоматичного видалення непотрібної пам'яті;
- підтримка політики .NET.

- програма C# може складатися з одного або декількох файлів, які включають вихідний код мовою C#. Всі ці файли мають розширення .cs.

Недоліки включають поганий графічний інтерфейс платформи X; C# є внутрішньою частиною платформи .NET, а отже сервер, що запускає програму, повинен бути на базі Windows; C# має невисоку гнучкість, так як переважно залежить від платформи .Net.

Наступною розглянемо об’єктно-орієнтовану мову програмування Python.

Python – це високорівнева інтерпретована інтерактивна та об’єктно-орієнтована сценарна мова, розроблена Гвідо ван Россумом [35]. Синтаксис Python надзвичайно простий для написання та читання коду – часто здається, ніби англійська мова застосовується замість синтаксису мови програмування, так як Python, навпроти інших високо використовуваних мов програмування, використовує англійські ключові слова замість пунктуації як оператори.

Python є інтерпретованою мовою програмування, а це означає, що дана мова опрацьовується одночасно з запуском. Програму, написану мовою, про яку йде мова, не потрібно компілювати перед її виконанням.

Python – це інтерактивна мова програмування, яка дозволяє взаємодіяти з командним рядком Python у реальному часі та таким чином створювати програму безпосередньо при розробці.

Об’єктна орієнтація Python означає, що він підтримує об’єктно-орієнтований вид програмування та підхід до програмування, що інкапсулює код усередині об’єктів. У цілому, в Python все виступає об’єктами, включно з винятками, інструментами для роботи з примітивними типами даних (float, integer, …) і навіть функціями. Їх називають «об’єктами першого класу», що означає можливість передавачі їх як аргументів іншим функціям, зберігання в змінній або навіть в окремих структурах даних.

Python є непоганою платформою для розробки систем переважно в будь-якій сфері, від інтерактивних веб-сайтів та ігор до складних наукових задач, включно з обробкою довільної текстової інформації.

Ця мова програмування має багато особливостей, включаючи такі [36]:

- easy-to-read – код, написаний з використанням стандартних ключових слів Python, дуже просто читається, що дозволяє розумти всю логіку, яку розробник вкладав у код;

- код Python можна запускати на безлічі комп'ютерного обладнання, при чому інтерфейс буде однаковим на кожній з платформ;

- інтерактивний режим Python забезпечує тестування та налагодження окремих ділянок коду онлайн;

- «ядро» Python, при необхідності, можна розширити при додаванні самостворених низькорівневих модулів, що дозволяє більше контролювати роботу створених розробником алгоритмів;

- Python реалізує драйвери для більшості відомих баз даних, включаючи SQL-орієнтовані MySQL, PostgreSQL, MSSQL, SQLite і NoSQL, а також MongoDB, DynamoDB, Redis, Neo4j та інші;

- Python має високу масштабованість, що означає, що мова забезпечує непогану структуру та підтримку програмних додатків з великою вагою.

Більш технічними моментами виділимо наступні ознаки:

- Python надає доступ до структур даних дуже високого рівня та підтримує динамічну перевірку типів даних. Вірне застосування цих функцій дає змогу істотно зменшити час, потрібний для реалізації різних архітектурних рішень;

- Python можна застосовувати як мову сценаріїв або скомпілювати в байт-код для розробки великих програм.

- Python неважко інтегрувати з іншими мовами програмування під час написання програм, такими як C/C++ (Cython), Java (Jython), C#/.NET (IronPython), Common Lisp (CLPython) та багатьма іншими.

Наступною розглянемо об'єктно-орієнтовану мову програмування Java.

Java також може бути універсально корисною мовою програмування високого рівня, спочатку розробленою для портативних гаджетів і приставок. У 1995 році Java створила програми в Інтернеті. Сьогодні Java широко використовується для створення веб-додатків і портативних додатків. Java згадується як один із перших улюблених і, отже, найбільш використовуваних

діалектів програмування останнім часом. Ця мова існує вже 20 років. Кілька експертів і експертів погоджуються, що Java є одним із найкращих життєздатних діалектів програмування, які коли-небудь створювали. Це найпоширеніша мова програмування, відома людині, і очікується, що вона продовжуватиме використовуватися через всюдисущу атмосферу Інтернету [37].

Перевагою Java є її зрозумілість – програмування, верстка, компіляція, дослідження та навчання не складніші, ніж додаткові діалекти програмування. Java може заплутати більше, ніж C++; тому Java використовує запрограмовану частину пам'яті та збирач сміття.

Ще однією перевагою є об’єктна орієнтація, яка дозволяє створювати стандартні проекти та багаторазовий код. Незалежний від стадії код Java - працює на будь-якій машині, не турбуючись про унікальне програмування, але JVM має бути доступною на машині.

Java – це розподілена мова. Це розподілена мова, оскільки вона надає інструмент для обміну інформацією та проектами між кількома ПК, покращуючи представлення та контроль інфраструктури. RMI (Remote Method Invocation) – це те, що полегшує розподілену обробку в Java. Безпека Java полягає в тому, що вона не має унікального покажчика. Крім того, саме адміністратор безпеки характеризує запис класу [38].

Наступна перевага – розподіл пам'яті. У Java пам'ять складається з двох розділів, один з яких зберігається, а інший є стеком. Кожного разу, коли ми викликаємо змінну, JVM виділяє пам’ять зі стека або купи. Це дозволяє без проблем створювати резервні копії та відновлювати інформацію.

Багатопотоковість також є плюсом – це потенціал програми для виконання багатьох завдань одночасно. востаннє з'явилося, щоб представити ідеї багатопоточності в Java. Java забезпечує автоматичне збирання сміття - Java має заплановану пам'ять для процесорів, керованих віртуальною машиною Java (JVM). У жодному разі ці статті більше не використовуються програмами, і вони ні з чим не пов’язані.

Враховуючи переваги та недоліки розглянутих мов програмування, для програмної реалізації інформаційної технології авторизації користувача з використанням клавіатурного почерку було вибрано об’єктно-орієнтовану мову програмування Java.

3.2 Обґрунтування вибору бібліотек та фреймворків для розробки інформаційної технології авторизації користувача з використанням клавіатурного почерку

Як згадувалось раніше, для реалізації інформаційної технології авторизації користувача з використанням клавіатурного почерку було вибрано об’єктно-орієнтовану мову програмування Java. Найбільш зручне та сучасне середовище розробки даною мовою – IntelliJ IDEA [39].

Перш за все розглянемо інструменти, за допомогою яких зручно реалізувати графічний інтерфейс користувача. Жодна сучасна орієнтована на користувача програма не може обійтися без зручного, зрозумілого, в ідеалі – красивого інтерфейсу. Найпершою технологією для створення графічного інтерфейсу користувача була технологія AWT, Abstract Window Toolkit. Пакет java.awt, мабуть, зазнав найбільших змін із розробкою випусків Java, однак залишається конкурентоспроможним і досі. Серед особливостей технології – дерево компонентів, модель повідомлень для гнучкої обробки дій користувача, робота з кольорами, шрифтами, графічна реалізація примітивів тощо[40].

Swing – це бібліотека Java Foundation Classes (JFC) і розширення Abstract Window Toolkit (AWT). Swing пропонує значно покращену функціональність порівняно з AWT, нові компоненти, розширені функції компонентів, чудову обробку подій із підтримкою перетягування.

Swing має приблизно в чотири рази більше компонентів інтерфейсу користувача, ніж AWT, і є частиною стандартного дистрибутива Java. Відповідно до сьогоднішніх вимог графічного інтерфейсу додатків, AWT є обмеженою реалізацією, яка не зовсім здатна забезпечити компоненти, необхідні

для розробки складних графічних інтерфейсів, необхідних для сучасних комерційних програм. Набір компонентів AWT має досить багато помилок і справді займає багато системних ресурсів порівняно з еквівалентними ресурсами Swing [41].

Найпростіші компоненти Swing мають такі можливості, що виходять далеко за рамки компонентів AWT:

- поворотні кнопки та мітки можуть відображати зображення замість або на додаток до тексту;

- межі навколо більшості компонентів Swing можна легко змінити. Наприклад: легко встановити межу в 1 піксель навколо зовнішньої сторони мітки Swing;

- компоненти Swing не обов’язково мають бути прямокутними. Кнопки, наприклад, можуть бути круглими;

- тепер новітні технології Assertive, такі як програми зчитування екрана, можуть легко отримувати інформацію з компонентів Swing. Наприклад: інструмент зчитування з екрана може легко захопити текст, який відображається на кнопці Swing або ярлику.

Дана бібліотека включає набір абстрактних класів, що забезпечують просту та ефективну розробку. Клас Frame призначений для створення повнофункціональних вікон додатків із рядком заголовка, рамкою, кнопками закриття, згортання та розгортання вікна. Оскільки Frame зазвичай є головним вікном програми, він створюється невидимим, щоб розробники могли налаштувати всі параметри, додати всі контейнери та вкладені компоненти, а потім переглянути його в підготовленому вигляді. Конструктор приймає один текстовий параметр – заголовок кадру.

Наведемо приклад компонентів для задання параметрів головного вікна програми:

```
public void createWindow() {
    frame = new JFrame();
    frame.setTitle("Меню");
    frame.setSize(300, 200);
```

```
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setLocationRelativeTo(null);
```

Компонент AWT має прямокутну форму, тому його розмір також описується двома цілочисельними параметрами – шириною (width) і висотою (height). Для опису розміру існує спеціальний клас Dimension (size), який визначає два публічних поля int width і height, а також допоміжні методи.

Розробники можуть встановити розмір компонента за допомогою методу setSize, який може приймати кілька цілих чисел як аргументи, або Dimension. Можна дізнатися поточні розміри за допомогою методу getSize(), який повертає Dimension, або за допомогою методів getWidth() і getHeight().

Задання шрифту реалізується за допомогою класу Font, який включає наступні параметри: ім’я шрифту, розмір і стиль. Приклад задання шрифту в інформаційній технології авторизації користувача з використанням клавіатурного почерку наведено в наступній стрічці:

```
Font font1 = new Font("TimesRoman", Font.BOLD, 14);
Font font0 = new Font("TimesRoman", Font.BOLD, 10);
```

У програмному забезпеченні об’єкт доступу до даних (DAO) – це абстрактний інтерфейс до певного типу бази даних або механізму зберігання. Деякі функції надаються незалежно від використовуваного механізму зберігання, без узгодження конкретно з цим механізмом зберігання. Цей шаблон проектування застосовується до багатьох мов програмування, більшості програмного забезпечення, що вимагає зберігання інформації, і більшості баз даних. Але цей шаблон традиційно асоціюється з програмами Java Enterprise Edition, які взаємодіють з реляційними базами даних через інтерфейс JDBC, оскільки він з’явився в рекомендаціях Sun Microsystems [42].

При проектуванні інформаційної системи виявляються певні шари, які відповідають за взаємодію різних модулів системи. Підключення до бази даних є одним з найважливіших компонентів програми. Завжди є частина коду, модуль, який відповідає за надсилання запитів до бази даних і обробку отриманих з неї

відповідей. У цьому випадку визначення об’єкта доступу до даних описує його як прошарок між базою даних і системою. DAO абстрагує системні сутності та відображає їх у базі даних, визначає загальні методи використання з’єднання, його отримання, закриття та (або) повернення до Connection Pool.

В інформаційній технології авторизації користувача з використанням клавіатурного почерку також використовується пакет Java.util. Він містить структуру колекцій, застарілі класи колекцій, модель подій, засоби дати й часу, інтернаціоналізацію та різні корисні класи (токенізатор рядків, генератор випадкових чисел і бітовий масив). За генерацію випадкових чисел в пакеті відповідає клас Random [43]. Основні методи класу:

- nextBoolean – випадкове значення типу boolean;
- nextDouble – випадкове значення для типу даних double;
- nextFloat – випадкове значення для типу даних float;
- nextInt – випадкове значення для типу даних int;
- nextLong – випадкове значення для типу даних long.

Іншими важливими класами пакету, що використовуються в інформаційній технології авторизації користувача з використанням клавіатурного почерку є:

ArrayList, що дозволяє реалізацію інтерфейсу List зі змінним розміром масиву;

HashSet: цей клас реалізує інтерфейс Set, який підтримується хеш-таблицею (фактично екземпляром HashMap).

Set: містить методи, успадковані від інтерфейсу колекції, і додає функцію, яка обмежує вставлення повторюваних елементів.

3.3 Опис класів і функцій інформаційної технології авторизації користувача з використанням клавіатурного почерку

Розроблена UML-діаграма класів інформаційної технології авторизації користувача з використанням клавіатурного почерку, фрагмент якої зображено на рисунку 3.1. Повна діаграма наведена в додатку В.

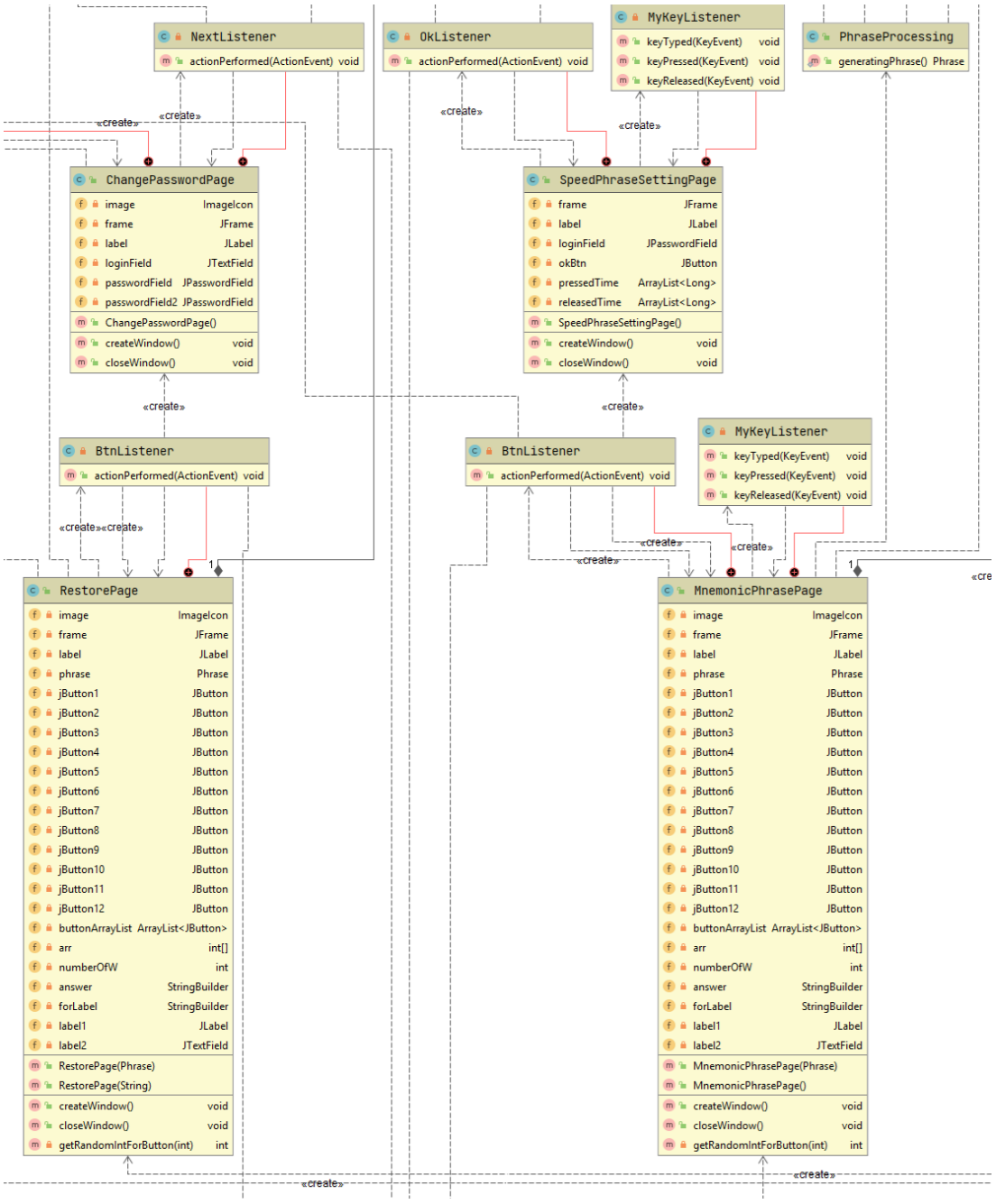


Рисунок 3.1 – Фрагмент UML-діаграми класів інформаційної технології авторизації користувача з використанням клавіатурного почерку

В інформаційній технології розроблено наступні класи: ChangePasswordPage, LoginPage, MainMenu, MnemonicPhrasePage, PhraseProcessing, RegisterPage, RestorePage, SpeedPhraseCheckPage, SpeedPhraseSettingPage, UserHomepage.

В класі MainMenu за допомогою інструментів Swing та AWT реалізовано графічний інтерфейс користувача. В даному класі використовуються наступні компоненти:

- JLabel – об’єктний компонент для розміщення тексту в контейнері;
- JFrame – створення повнофункціональних вікон додатків із рядком заголовка, рамкою, кнопками закриття, згортання та розгортання вікна;
- ImageIcon – реалізація інтерфейсу Icon, який малює піктограми із зображень.

У класі LoginPage використовуються ті ж самі компоненти, що і в MainMenu, однак він доповнений ще кількома об’єктами:

- об’єкт JTextField – це текстовий компонент, який дозволяє редагувати один рядок тексту;
- об'єкт JPasswordField – текстовий компонент, спеціалізований для введення пароля.

За допомогою класу LoginPage реалізовано вікно вводу логіна та пароля, передбачена можливість відновлення пароля, а також виведення діалогового вікна, що сповіщає про помилку при невірно введеному логіна або пароля. На рисунку 3.2 наведено діаграму діяльності інформаційної технології авторизації користувача з використанням клавіатурного почерку при введенні логіна та пароля.

Реалізація вікна реєстрації відбувається в класі RegisterPage. В даному класі використовуються компоненти JLabel, JFrame, ImageIcon, JTextField, JPasswordField, розглянуті в попередніх класах. За допомогою класу RegisterPage реалізується реєстрація користувача в системі. При введенні користувачем даних логіна, пароля та повторному введенні пароля спочатку реалізована перевірка заповненості полів, потім звіряється текст пароля з повторно введеним паролем,

а також перевірка наявності зареєстрованого користувача з введеним логіном. У разі виконання однієї з вимог виводиться відповідна помилка.

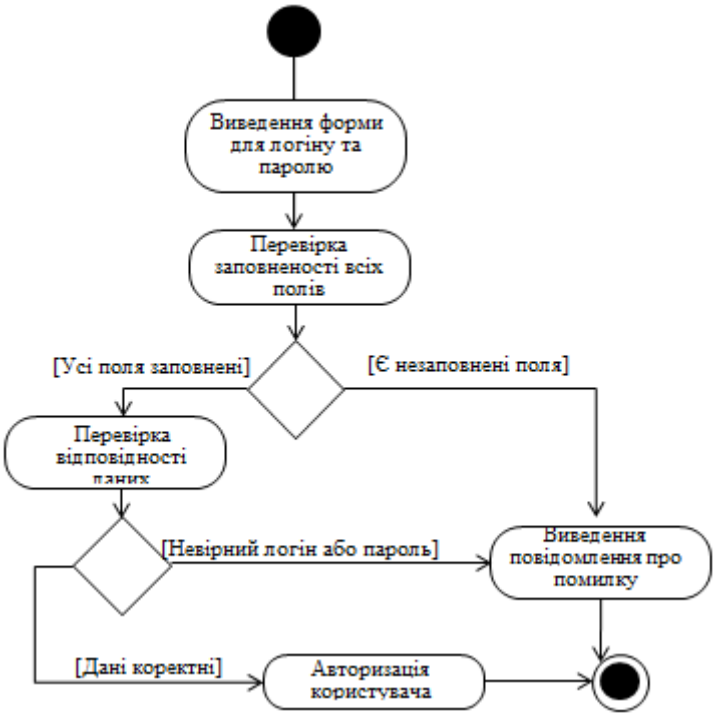


Рисунок 3.2 – Діаграма діяльності інформаційної технології при введенні логіна та пароля

Клас RestorePage реалізує створення сторінки для відновлення пароля. В класі реалізується доступ до бази даних, з якої імпортуються фрази для підтвердження користувача. В ньому присутні компоненти для створення графічного інтерфейсу користувача, а також використовуються компоненти пакету Java.util – ArrayList, що дозволяє реалізацію інтерфейсу List зі змінним розміром масиву та Random, що відповідає за генерацію випадкових чисел.

Реалізація зміни пароля відбувається у класі ChangePasswordPage. В класі присутні компоненти JLabel, JFrame, ImageIcon, JTextField, JpasswordField.

Клас PhraseProcessing відповідає за генерацію слів. В ньому присутні компоненти пакету Java.util: Random, ArrayList, HashSet та Set.

Класи SpeedPhraseCheckPage, SpeedPhraseSettingPage містять основну логіку програми. Так, клас SpeedPhraseSettingPage зчитує швидкість введення фрази користувачем та встановлює значення для алгоритму авторизації

користувача за допомогою клавіатурного почерку. Натомість SpeedPhraseCheckPage при авторизації користувача запитує зареєстрованого користувача фразу, за якою відбувається перевірка клавіатурного почерку та приймається рішення щодо надання доступу в програму.

Діаграма діяльності інформаційної технології авторизації користувача з використанням клавіатурного почерку при перевірці клавіатурного почерку зображена на рисунку 3.3.

Рисунок 3.3 – Діаграма діяльності інформаційної технології при перевірці клавіатурного почерку

Клас MnemonicPhrasePage реалізує вікно, в якому випадковим чином розставлені фрази, послідовність введення яких слугує підтвердженням для користувача, що бажає відновити пароль.

В класі використовується об’єкт доступу до даних (DAO), бібліотека Swing та компоненти пакету Java.util. Створений масив з 12 кнопок, приймають випадкове значення, якому відповідає певна фраза. За допомогою цього послідовність фраз кожен раз змінюється. Конкретний користувач при реєстрації вибирає послідовність фраз, що зберігатиметься в базі та буде

ідентифікувати користувача при зміні фрази, за якою здійснюється перевірка клавіатурного почерку та підтверджується зміна пароля.

Клас UserHomepage реалізує сторінку користувача, доступ до якої був наданий при авторизації.

Отже, в рамках даного підрозділу розроблено UML-діаграма інформаційної технології авторизації користувача з використанням клавіатурного почерку, описано основні компоненти розроблених класів, наведено діаграму діяльності інформаційної технології авторизації користувача з використанням клавіатурного почерку при перевірці клавіатурного почерку та діаграму діяльності інформаційної технології при введенні логіна та пароля.

3.4 Тестування інформаційної технології авторизації користувача з використанням клавіатурного почерку

Розроблена інформаційна технологія авторизації користувача з використанням клавіатурного почерку була протестована, що підтвердило коректність її роботи. Тестування можна характеризувати як процес експериментального аналізу функціональності досліджуваної інформаційної технології. Характер тестування залежить від знань про завдання, поставлені перед розробкою та про структурний склад і характер зв’язків між компонентами інформаційної технології.

Провівши 500 запусків програми було доведено підвищення точності авторизації користувача з використанням клавіатурного почерку.

Початок роботи з інформаційною технологією здійснюється шляхом виведення вікна початкової активності – головного меню програми. Вікно має назву «Меню» та містить кнопки, що відповідають режиму роботи (рис. 3.4). Наявні два режими – авторизації та реєстрації.

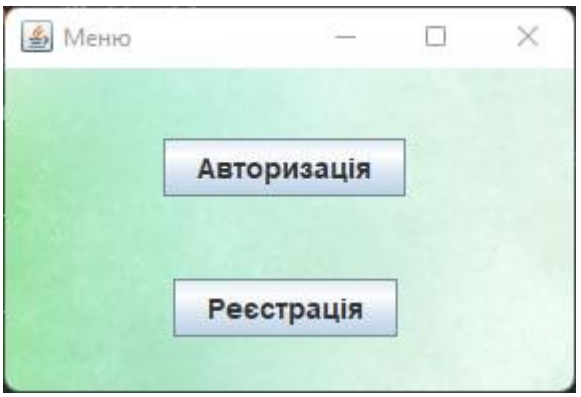


Рисунок 3.4 – Загальний вигляд інтерфейсного вікна початкової активності

Функціональні кнопки меню здійснюють перехід до наступних вікон: вікна авторизації та вікна реєстрації відповідно до побажань користувача. Якщо користувач вибрав режим «Авторизація», виводиться вікно з полями для введення логіна та пароля, кнопками «Авторизуватися» та «Відновити пароль» (рис. 3.5).

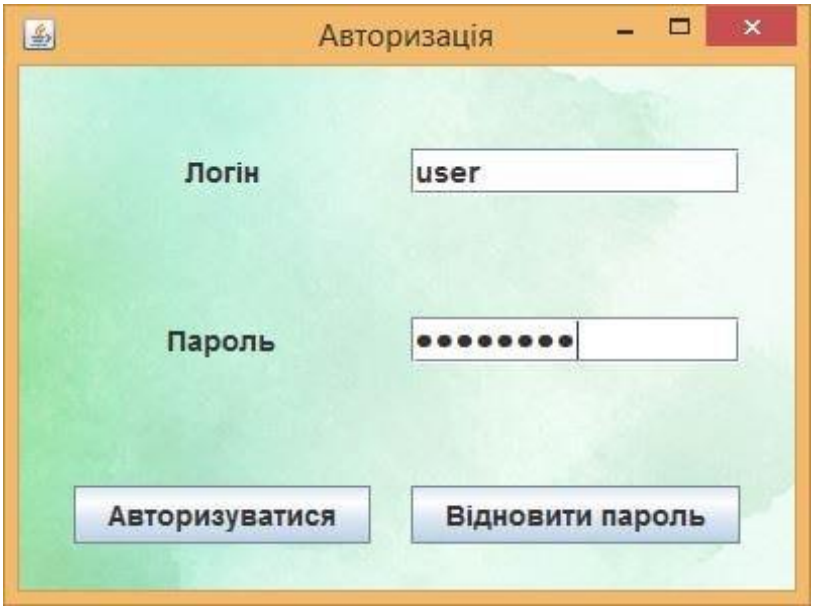


Рисунок 3.5 – Загальний вигляд інтерфейсного вікна авторизації

Ввівши логін і пароль правильно, користувач натискає кнопку авторизуватися, після чого відбувається перевірка клавіатурного почерку за допомогою спеціальної фрази (рис. 3.6) та вхід до системи, якщо перевірка пройдена успішно.

Рисунок 3.6 – Загальний вигляд інтерфейсного вікна для введення фрази

Після успішної перевірки клавіатурного почерку виводиться діалогове вікно, зображене на рисунку 3.7, при невдалій – вікно на рисунку 3.8.

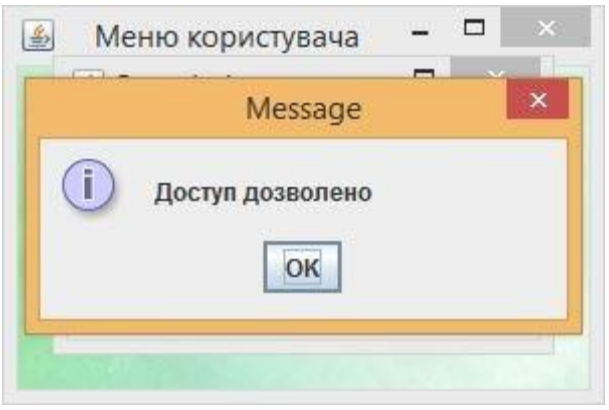


Рисунок 3.7 – Загальний вигляд діалогового вікна надання доступу

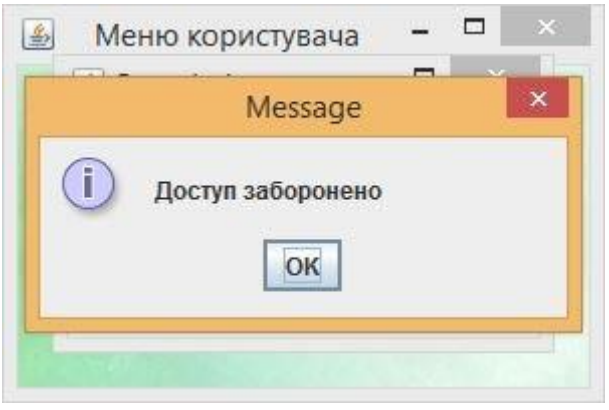


Рисунок 3.8 – Загальний вигляд діалогового вікна заборони доступу

Якщо користувач бажає відновити пароль, здійснюється перехід до вікна перевірки фрази (рис. 3.9). Якщо користувач ввів послідовність слів невірно, виводиться повідомлення про помилку (рис. 3.10).

Рисунок 3.9 – Загальний вигляд інтерфейсного вікна для відновлення пароля

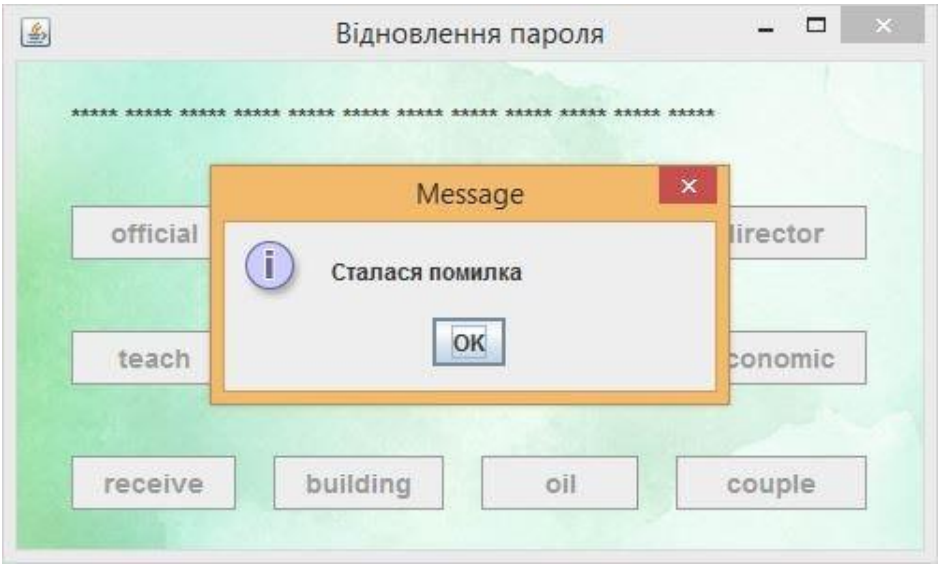


Рисунок 3.10 – Загальний вигляд інтерфейсного вікна виведення помилки

Після проходження перевірки виводиться вікно зміни пароля (рис. 3.11). Вікно містить поля для введення нового пароля, для повторення пароля, кнопки «Назад» та «Далі».

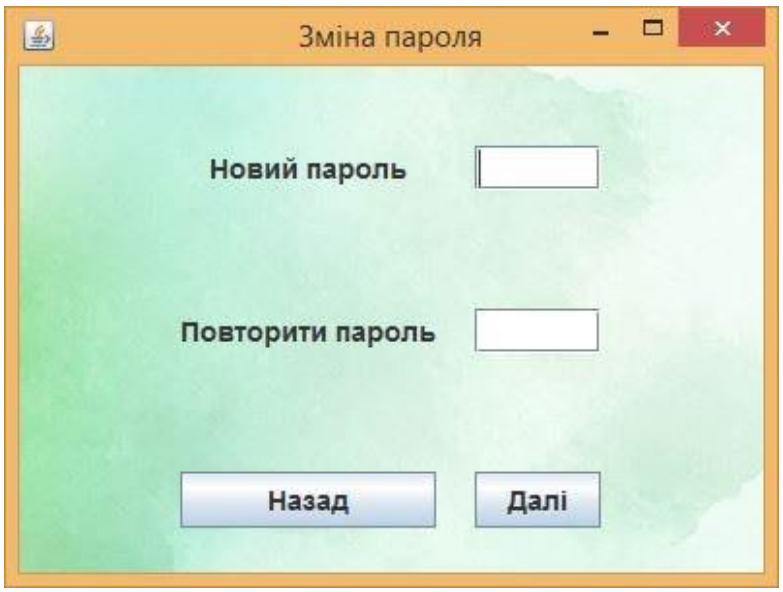


Рисунок 3.11 – Загальний вигляд інтерфейсного вікна зміни пароля

Для нового користувача існує можливість реєстрації в системі. Для цього у вікні головного меню (рис. 3.4) обирається пункт «Реєстрація», після чого здійснюється перехід до вікна реєстрації (рис. 3.12).

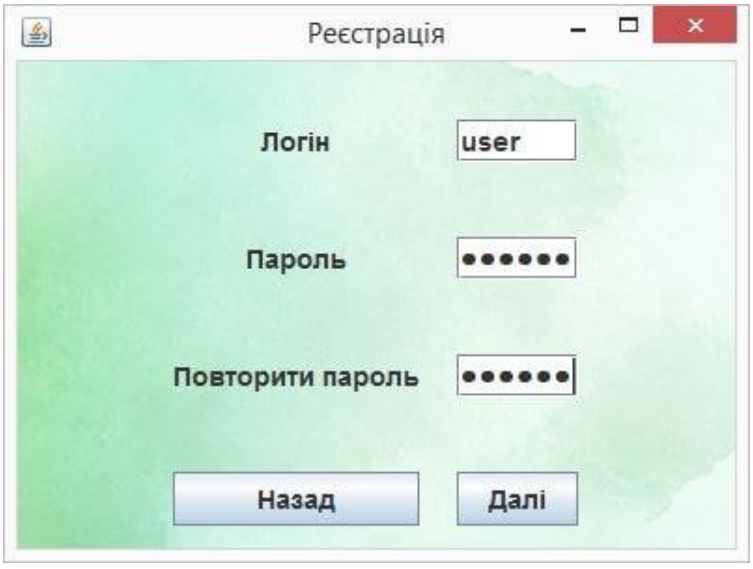


Рисунок 3.12 – Загальний вигляд інтерфейсного вікна реєстрації

Вікно реєстрації користувача містить поля для введення логіна, пароля та повторення пароля. Також тут присутні кнопки «Назад», для повернення до головного меню, та «Далі». При реєстрації користувача здійснюється перевірка на заповненість всіх полів, відповідність повторно введеного паролю

початковому, наявність введеного логіну в базі. Відповідно до наведених помилок виводиться діалогове вікно з конкретною помилкою (рис. 3.13 – 3.14).

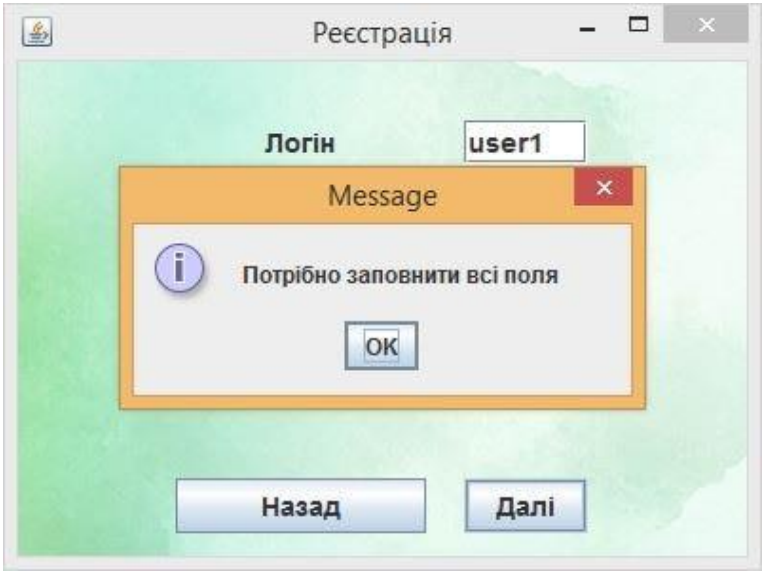


Рисунок 3.13 – Загальний вигляд інтерфейсного вікна виведення помилки заповнення полів

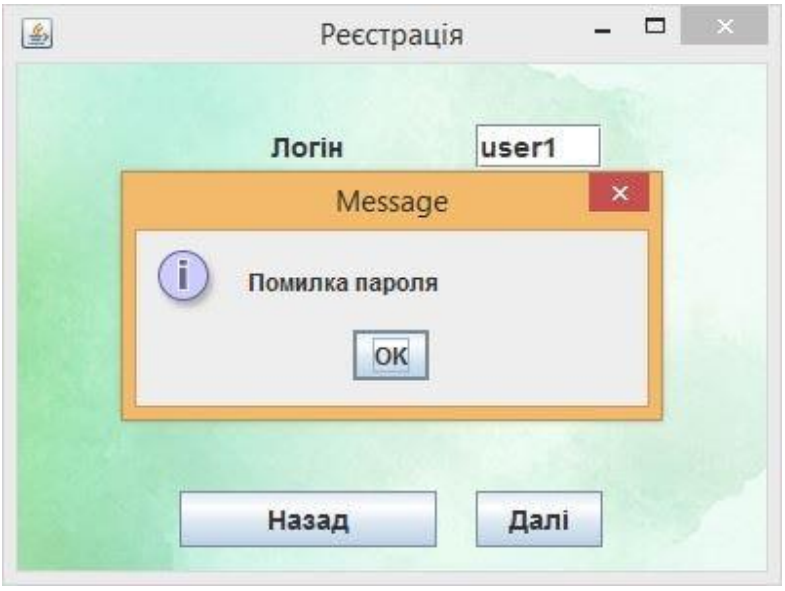


Рисунок 3.14 – Загальний вигляд інтерфейсного вікна виведення помилки пароля

В таблиці 3.1 наведено порівняльні результати роботи розробленого алгоритму авторизації користувача з використанням клавіатурного почерку

використовуючи порівняння шаблону та поточного введення за допомогою використання евклідової відстані зі звичайним алгоритмом.

Таблиця 3.1 – Порівняльний аналіз ефективності роботи програми за точністю визначення користувача

Програма	Точність авторизації користувача
Розроблений алгоритм	86,3%
Звичайний алгоритм	82,9%

Таким чином, із таблиці 3.1 видно, що розроблена інформаційна технологія має на 3,4% вищу точність авторизації користувача з використанням клавіатурного почерку.

Результати тестування роботи програмного додатку відповідають результатам, які очікувались.

3.5 Висновок до розділу 3

У третьому розділі було здійснено обґрунтуваня вибору мови програмування для реалізації інформаційної технології авторизації користувача з використанням клавіатурного почерку. Інформаційну технологію розроблено на об’єктно-орієнтованій мові програмування Java, враховуючи її кросплатформність, об’єктно-орієнтованість та простоту розробки.

Розроблено та описано основні класи і функції, програмно реалізовано інформаційну технологію авторизації користувача з використанням клавіатурного почерку.

Проведено тестування роботи інформаційної технології авторизації користувача з використанням клавіатурного почерку та проаналізовано її результати. Тестування програми підтвердило очікувані результати та правильність роботи програми, довівши підвищення точності визначення користувача.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу інформаційної технології авторизації користувача з використанням клавіатурного почерку, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Аналогом може бути система «ESET Protect Entry», вартість ліцензії якої складає 794 грн/місяць, тобто за три роки вартість використання даної програми складе 28584 грн..

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів, якими можуть бути провідні викладачі випускової або спорідненої кафедри чи інші відомі фахівці. Не рекомендується залучати експертами керівника магістерської кваліфікаційної роботи та завідувача відповідної випускової кафедри.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за п’ятибальною шкалою)					
	0	1	2	3	4
1	*2*	*3*	*4*	*5*	*6*
Технічна здійсненість концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження таблиці 4.1

1	*2*	*3*	*4*	*5*	*6*
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижча за ціни аналогів	Ціна продукту значно нижча за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні не значні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

Продовження таблиці 4.1

1	*2*	*3*	*4*	*5*	*6*
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5ти років. Термін окупності інвестицій більший 5ти років	Термін реалізації ідеї менший 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менший 3-х років. Термін окупності інвестицій менший 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що потребує значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту потребує незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 4.2.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	3	3	4
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	3	4	3

Продовження таблиці 4.2

4. Ринкові переваги (технічні властивості)	4	3	4
5. Ринкові переваги (експлуатаційні витрати)	3	4	4
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	3	4	2
9. Практична здійсненність (наявність фінансів)	3	4	4
10. Практична здійсненність (необхідність нових матеріалів)	4	3	3
11. Практична здійсненність (термін реалізації)	3	3	3
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	*38*	*40*	*39*
Середньоарифметична сума балів $СБ_c$	$\text{СБ}_\text{с} = \frac{\sum_i^3 \text{СБ}_i}{3} = 39$		

За результатами розрахунків, наведених в таблиці 4.2, робиться висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використовують рекомендації, наведені в табл. 4.3.

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вищий середнього
21…30	Середній
11…20	Нижчий середнього
0…10	Низький

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є вищим середнього, що досягається за рахунок підвищенням точності авторизації користувача за допомогою використання характеристик, властивих конкретному користувачу.

Також, використання розробленої програми є значно дешевшим для кінцевого споживача. Якщо для аналога затрати складають 600 грн щомісячно, то для розроблюваної програми щомісячні витрати плануються на рівні 300-400 грн.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної, дослідноконструкторської, конструкторсько-технологічної роботи, створенням дослідного зразка і здійсненням виробничих випробувань, під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуються за такими *статтями:*

- витрати на оплату праці;
- відрахування на соціальні заходи;
- матеріали;
- паливо та енергія для науково-виробничих цілей;
- витрати на службові відрядження;
- спецустаткування для наукових (експериментальних) робіт;
- програмне забезпечення для наукових (експериментальних) робіт;
- витрати на роботи, які виконують сторонні підприємства, установи і організації;
- інші витрати;
- накладні (загальновиробничі) витрати.

4.2.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, лаборантам, робітникам, студентам, аспірантам та іншим

працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховують відповідно до посадових окладів працівників, за формулою:

$$З_о = \sum_i = 1 \frac{k \times M_{ni} \times t_i}{T_p}, \quad (4.1)$$

де k – кількість посад дослідників, залучених до процесу досліджень; M_{ni} – місячний посадовий оклад конкретного дослідника, грн; t_i – кількість днів роботи конкретного дослідника, дн.; T_p – середня кількість робочих днів в місяці, $T_p=21…23$ дні. Для дослідження оберемо $T_p=22$. Проведені розрахунки бажано звести до таблиці.

Таблиця 4.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Кількість днів роботи	Витрати на заробітну плату, грн
Керівник проекту	40 000	1 1818,2	40	72 727
Науковий співробітник	25 000	1136,4	40	45 455
Всього				118 182

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата дослідників та робітників

Додаткова заробітна плата розраховується як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_д = (З_о + З_р) \times \frac{Н_{дод}}{100\%}, \quad (4.2)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати.

$З_д$ = (118182 * 12 % / 100 %) = 14 181,82 (грн.)

4.2.2 Відрахування на соціальні заходи

До статті «Відрахування на соціальні заходи» належать відрахування внеску на загальнообов'язкове державне соціальне страхування та для здійснення заходів щодо соціального захисту населення (*ЄСВ – єдиний соціальний внесок*).

Нарахування на заробітну плату дослідників та робітників розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Н_з = (З_о + З_р + З_{дод}) \times \frac{Н_{зп}}{100\%}, \quad (4.3)$$

де $Н_{зп}$ – норма нарахування на заробітну плату.

$Н_з$ = (118 181 + 14 181,82) · 22 % / 100 % = 29 119,82 (грн.)

4.2.3 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних

для проведення досліджень, також витрати на їх проектування, формування та встановлення.

До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10…12% від вартості програмного забезпечення.

Балансову вартість програмного забезпечення розраховують за формулою:

$$В_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot С_{прг.i} \cdot К_i \quad (4.4)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу цього виду, грн; $С_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.; $К_i$ – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо ($К_i$ = 1,10…1,12), в даному дослідженні використаємо $К_i$ = 1,10; k – кількість найменувань програмних засобів.

Отримані результати необхідно звести до таблиці 4.5.

Таблиця 4.5 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Intellij Idea 2018	2	20000	40000
			40000

4.2.4 Амортизація обладнання, програмних засобів та приміщень

До статті «Амортизація обладнання, програмних засобів та приміщень» відносять амортизаційні відрахування по кожному виду обладнання, устаткування та інших приладів і пристроїв, а також програмного забезпечення для проведення науково-дослідної роботи, *за його наявності* в дослідній організації або на підприємстві.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$А_{обл} = \frac{Ц_б}{Т_в} \times \frac{Т_{вик}}{12}, \quad (4.5)$$

$$А_{обл} = \frac{56210}{2} \times \frac{1{,}82}{12} = 4262{,}59$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн; $t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців; $Т_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Проведені розрахунки необхідно звести до таблиці 4.6.

Таблиця 4.6 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Комп’ютер (Ноутбук Lenovo IdeaPad Gaming 3 16IAH7 49999 грн, монітор Lenovo 24" S24E-20 6211грн)	56210	2	1,82	4262,59
Приміщення	900000	20	1,82	6825
Офісне обладнання (Крісло 4 199грн, стіл 2 199грн)	6398	5	1,82	194,07
Ліцензійна ОС, та спеціалізовані ліцензійні нематеріальні ресурси (Intellij Idea)	20000	10	1,82	303,33
Всього				11584,99

4.2.5 Паливо та енергія для науково-виробничих цілей

До статті «Паливо та енергія для науково-виробничих цілей» належать витрати на придбання у сторонніх підприємств, установ і організацій будь-якого палива, що витрачається з технологічною метою на проведення досліджень. Стаття формується у разі виконання енергоємних наукових досліджень за методом прямого внесення витрат і досягає значної питомої ваги у собівартості досліджень.

Витрати на силову електроенергію (B_e) розраховують за формулою:

$$B_e = \sum \frac{W_{yi} \times t_i \times Ц_e \times К_{впі}}{\eta_{i=1}}, \quad (4.6)$$

де W_{yi} – встановлена потужність обладнання на певному етапі розробки, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год; $Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії для юридичних осіб $Ц_e$ = 6,10); $К_{впі}$ – коефіцієнт, що враховує використання потужності, $К_{впі} < 1$; η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Проведені розрахунки необхідно звести до таблиці.

Таблиця 4.6 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук (Lenovo IdeaPad Gaming 3 16IAH7)	0.096	320	168,65
Монітор (Lenovo 24"S24E-20)	0.036	320	63,24
Освітлення	0.024	320	42,16
Всього			274,06

4.2.6 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «*Інші витрати*» розраховуються як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \times \frac{Н_{ів}}{100\%}, \quad (4.7)$$

де $H_{ів}$ – норма нарахування за статтею «*Інші витрати*».

$I_в$ = 118182 * 60% / 100% = 70 909,2 (грн.)

4.2.7 Накладні

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов’язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «*Накладні (загальновиробничі) витрати*» розраховуються як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{нзв} = (З_о + З_р) \times \frac{Н_{нзв}}{100\%}, \quad (4.8)$$

де $Н_{нзв}$ – норма нарахування за статтею «*Накладні (загальновиробничі) витрати*».

$Н_{нзв}$ = 118 182 * 115% / 100% = 135 909,3 (грн.)

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$В_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}. \quad (4.9)$$

$В_{заг}$ = 72 727 + 45 455 + 14 181,82 + 29 119,82 + 40 000 + 11584,99 + 274,06 + 70 909,2 + 135 909,3 = 420161,19 грн.

Загальні витрати *ЗВ* на завершення науково-дослідної (науковотехнічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{В_{заг}}{\eta}, \quad (4.10)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науководослідної роботи. Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η =0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7; впровадження, то η=0,9.

ЗВ = 378 384,47 / 0,5 = 840322,38 грн.

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 4-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб’єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов’язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу.

При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);

- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

4.3.1 Розробка чи суттєве вдосконалення інформаційної технології для використання масовим споживачем

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta П_i = (\pm \Delta Ц_0 \cdot N + Ц_0 \cdot \pm \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \qquad (4.11)$$

де $\pm\Delta Ц_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу; N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки; $Ц_o$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $Ц_o = Ц_б \pm \Delta Ц_o$; $Цб$ – вартість програмного продукту у році до впровадження результатів розробки; ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик; λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0{,}8333$. p – коефіцієнт, який враховує рентабельність продукту; ϑ – ставка податку на прибуток, у 2022 році ϑ =18%.

Припустимо, що при прогнозованій ціні 7200 грн. за програмний продукт, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 500 грн. Кількість

проданих підписок на програмний продукт також збільшиться: протягом першого року – на 1000 шт., протягом другого року – на 2000 шт., протягом третього року на 4000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$\Delta П_1$ = (0*500 + (7200 + 500)*1000)* 0,8333* 0,3) * (1 - 0,18) = 1 578 436,86 грн.

$\Delta П_2$ = (0*500 + (7200 + 500)*(1000 + 2000)* 0,8333* 0,3) * (1 - 0,18) = 4 735 310,58 грн.

$\Delta П_3$ = (0*500 + (7200 + 500)*(1000 + 2000 + 4000)* 0,8333* 0,3) * (1 - 0,18) = 11 049 058 грн.

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 17 362805,44 грн.

Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-дослідної розробки:

$$ПП = \sum_1^T \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.12)$$

де $\Delta П_i$ збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної роботи, грн; Т – період часу, протягом якою виявляються результати впровадженої науково-дослідної роботи, роки; τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ = 0,05…0,15; t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

ПП= (1 578 436,86 / (1 + 0,1)^1) + (4 735 310,58 / (1 + 0,1)^2) + (11 049 058 / (1 + 0,1)^3) = 1 434 942,6 + 3 913 479,82 + 8 301 320,83 = 13 649 743,25 грн.

Далі розраховують величину початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \times \text{ЗВ}, \quad (4.13)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}$ = 2...5, але може бути і більшим; ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

PV = 2 * 840 322,38 = 1 680 644,76 грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - \text{PV}, \quad (4.14)$$

$$E_{\text{абс}} = 13\,649\,743{,}25 - 1\,680\,644{,}76 = 11\,969\,098{,}49 \text{ грн.}$$

Оскільки $E_{\text{абс}} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар’єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$. Для цього використаємо формулу:

$$E_{\text{в}} = \sqrt[\text{Т}_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \tag{4.15}$$

$\text{Т}_{\text{ж}}$ життєвий цикл наукової розробки, роки.

$$E_{\text{в}} = \sqrt[3]{(1 + 11\,969\,098{,}49 \div 1\,680\,644{,}76)} - 1 = 1{,}01$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \tag{4.16}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d = (0,23…0,25); f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,5).

$$\tau_{min} = 0{,}25 + 0{,}05 = 0{,}29.$$

Так як $\text{Е}_{\text{в}} > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$\text{Т}_{\text{ок}} = \frac{1}{\text{Е}_{\text{в}}}, \tag{4.17}$$

$$\text{Т}_{\text{ок}} = 1 / 1{,}01 = 0{,}92 \text{ р.}$$

Оскільки < 3-х років, а саме термін окупності рівний 0,92 роки, то фінансування даної наукової розробки є доцільним.

4.4 Висновок до розділу 4

Економічна частина даної роботи містить розрахунок витрат на розробку інформаційної технології авторизації користувача з використанням клавіатурного почерку, сума яких складає 840322,38 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є високо конкурентоспроможним.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи реалізовано інформаційну технологію авторизації користувача з використанням клавіатурного почерку. Для досягнення мети були поставлені такі завдання: проаналізувати предметну область авторизації користувача з використанням клавіатурного почерку; провести аналіз сучасних систем авторизації користувача; розробити структуру інформаційної технології авторизації користувача з використанням клавіатурного почерку; розробити математичну модель системи; здійснити програмну реалізацію інформаційної технології авторизації користувача з використанням клавіатурного почерку; провести тестування роботи інформаційної технології авторизації користувача з використанням клавіатурного почерку, які виконані в повному обсязі.

Проведено аналіз предметної області та обґрунтовано доцільність розробки інформаційної технології авторизації користувача з використанням клавіатурного почерку. Наведено можливості авторизації користувача в сучасних інформаційних системах та проаналізовані загрози в авторамтизованих системах.

Проведено аналіз сучасних систем авторизації користувача. Виявлено недостатню точність наявних систем авторизації користувача з використанням клавіатурного почерку, що доводить актуальність досягнення поставленої мети.

Було розроблено структуру інформаційної технології авторизації користувача з використанням клавіатурного почерку та спроектовано схему алгоритму роботи розроблюваної програми. наведено основні етапи інформаційної технології, розроблено архітектуру системи виявлення підміни користувача. Розроблено алгоритм авторизації користувача з використанням клавіатурного почерку, а також розглянуто UML-діаграми прецедентів та розгортування програмного забезпечення.

На основі аналізу існуючих моделей та методів, що застосовуються для вирішення поставленої задачі було обґрунтовано використання в інформаційній

технології авторизації користувача за допомогою клавіатурного почерку, використовуючи порівняння шаблону та поточного введення за допомогою використання евклідової відстані, які увійшли в розроблену математичну модель авторизації користувача з використанням клавіатурного почерку, що дає змогу підвищити точність захисту.

Здійснено проектування та розробку інформаційної технології авторизації користувача за допомогою клавіатурного почерку. Обґрунтовано вибір мови програмування. Розробка інформаційної технології велася на об'єктно-орієнтованій мові програмування Java в середовищі розробки IntelliJ IDEA. Здійснено програмну реалізацію інформаційної технології авторизації користувача за допомогою клавіатурного почерку, детально описано класи та функції, які використовуються в інформаційній технології.

Проведено тестування розробленої програми та підтверджено її працездатність. Точність авторизації користувача з використанням клавіатурного почерку зросла на 3,4%, що означає доведення поставленої мети.

Здійснено економічне обґрунтування доцільності розробки інформаційної технології авторизації користувача за допомогою клавіатурного почерку. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, знайдено термін окупності витрат для виробника та економічний ефект для споживача при використанні такої розробки. В результаті аналізу розрахунків можна зробити висновок, що розробка у виробництві та використання дешевша за аналог і є висококонкурентоспроможною. Період окупності складе близько 0,92 роки. Загальні витрати становлять 840322,38 грн.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Король Я.О., Озеранський В.С., Перевозніков С.І. «Аналіз та реалізація засобів автентифікації користувача» в Матеріали конференції «L Науково-технічна конференція підрозділів Вінницького національного технічного університету (2021)», Вінниця, 2021. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki2021/paper/ view/ 12689/ 10667. Дата звернення: листопад 2022.

2. Державна служба спеціального зв’язку та захисту інформації України. URL: https://cip.gov.ua/ua (дата звернення: 21.08.2022).

3. Сайт президента зазнав DDos-атаки. URL: https://www.slovoidilo.ua/2022/06/25/novyna/polityka/sajt-prezydenta-zaznav-dos-ataky (дата звернення: 21.08.2022).

4. Чала Л. Е. Порівняльний метод автентифікації користувачів комп'ютерних систем за клавіатурним почерком, Харків, 2008. 400с. URL: https://goo.gl/9JuhYu. (дата звернення: 29.08.2022).

5. Копняк К. В., Корсунець Т. А. Автоматизація документообігу як складова підвищення ефективності діяльності підприємства // Економіка. Фінанси. Менеджмент: актуальні питання науки і практики. Вінниця, 2017, №11. С. 57-68.

6. Писаренко В. П. Організаційно-правові засади електронного документування в органах влади : монографія. Полтава: ПУЕТ, 2012. 250 с.

7. Колеснікова Я.В. Огляд програмних продуктів для автоматизації кадрового діловодства // Вісник Луганського національного університету ім. Т. Шевченка. 2010. №17. С. 230-237.

8. Закон України «Про електронні документи та електронний документообіг» (Відомості Верховної Ради України (ВВР), 2003, №36, ст.275). Верховна Рада України. 2003. URL: https://xn--80aagahqwyibe8an.com/ukrajiny-zakony/zakon-ukrajini-pro-elektronni-dokumenti-2003-786.html (дата звернення: 03.09.2022).

9. Що таке електронний цифровий підпис (ЕЦП)? Урядовий портал. URL: https://www.kmu.gov.ua/usi-pitannya-po-e-poslugam/sho-tak-elektronnij-cifrovij-pidpis-ecp (дата звернення: 05.09.2022).

10. Поліщук В. В. Програмні технології захисту інформації : конспект лекцій. Ужгород : УжНУ, 2018. 80 с.

11. Лагун А. Е. Криптографічні системи та протоколи : нав. посібник. Львів : Видавництво Львівської політехніки, 2013. 96 с.

12. Горбенко І. Д. Гриненко Т. О. Захист інформації в інформаційнотелекомунікаційних системах : навч. посіб. Ч.1. Криптографічний захист інформації. Харків : ХНУРЕ, 2004. 368 с.

13. Лупенко С. А., Шаблій Н. Р., Лупенко А. М. Компаративний аналіз моделей, методів та засобів автентифікації особи в інформаційних системах за її клавіатурним почерком // конференції Національного університету «Львівська політехніка», Львів, 2014. URL: https://ena.lpnu.ua/. (дата звернення: 15.09.2022).

14. Кухаренко С. В. Комплексні системи контролю та управління доступом : метод. вказівки. Одеса : НУ «ОЮА», 2020. 32 с.

15. Дранишников Л. В. Інтелектуальні методи в управлінні : навч. посіб. Кам'янське: ДДТУ, 2018. 416 с.

16. The Definitive Guide to Authentication URL: https://www.strongdm.com/authentication (дата звернення: 21.09.2022).

17. Особливості аналізу бізнес-середовища. URL: https://cutt.ly/Xnhi5QV. (дата звернення: 21.09.2021).

18. Класифікація веб-каталог URL: https://www.neworldencyclopedia. org/entry/Web_directory. (дата звернення: 21.09.2022).

19. R.V. Yampolskiy and V. Govindaraju. Behavioral biometric: a survey and classification. International Journal of Biometrics, 2008, 1(1): 81-113.

20. Лупенко С. А. Розвиток теорії моделювання та обробки циклічних сигналів в інформаційних системах: дис. д-ра техн. наук. Львів, 2010. 515 с.

21. Чала Л. Е. Методи динамічної ідентифікації користувачів розподілених інформаційних систем: автореф. «Автоматизовані системи управління та прогресивні інформаційні технології» Харків, 2006. 20 с.

22. Obaidat, Mohammad S., Issa Traore, and Isaac Woungang. Biometric-Based Physical and Cybersecurity Systems. Springer International Publishing, 2019.

23. LastPass URL: https://www.lastpass.com/ (дата звернення: 18.10.2022).

24. TypingDNA URL: https://www.typingdna.com/ (дата звернення: 18.10.2022)

25. BehavioSec URL: https://www.behaviosec.com/ (дата звернення: 18.10.2022)

26. Banking System with Keystroke authentication URL: https://github.com/saqsham/Banking-Systemwith-Keystroke-authentication (дата звернення: 18.10.2022)

27. BioCatch-Auth URL: https://www.biocatch.com/ (дата звернення: 18.10.2022)

28. Pirlo G, Cuccovillo V, Diaz-Cabrera M, Impedovo D, Mignone P (2015) Multidomain verification of dynamic signatures using local stability analysis. IEEE Trans Hum Mach Syst 45(6): 805–810.

29. Aaker, David A. and Day George S. Marketing Research. 4th ed. NewYork: John Wiley and Sons, 1990. Chapter 22 «Forecasting».

30. K. Revett, H. Jahankhani, S. T. de Magalhes, and H. M. D. Santos. A survey of user authentication based on mouse dynamics // Proceedings of 4th International Conference on Global E-Security, pages 210-219, London, June 2008.

31. Іванов В.Г., Мазниченко Н.І. Ідентифікація користувачів інформаційних систем: аналіз і прогнозування підходів. Системний аналіз. Інформатика. Управління (САІУ-2012) : матеріали III Міжнар. наук.-практ. конф., м. Запоріжжя, 14-16 березня 2012 р. Запоріжжя : КПУ, 2012. С.127-128.

32. P. Bours and C. J. Fullu. A login system using mouse dynamics // Fifth International Conference of Intelligent Information Hiding and Multimedia Signal Processing, 2009:1072-1077.

33. Foo Kune, Denis and Kim, Yongdae. “Timing attacks on pin input devices”. In: Proceedings of the 17th ACM conference on Computer and communications security. ACM. 2010, pp. 678–680.

34. Коноваленко І.В., Марущак П.О., Савків В.Б. Програмування мовою C# 7.0 : навч. посіб. Тернопіль : ТНТУ імені Івана Пулюя, 2017. 300 с.

35. Настенко Д.В., Нестерко А. Б.Основи об’єктноорієнтованого програмування на мові C# : навч. посіб. Київ: НТУУ «КПІ», 2016. 76 с.

36. Костюченко А. О. Основи програмування мовою Python : навч. посіб. Чернігів : НУЧК ім. Т. Г. Шевченка.

37. М.Ф. Копитко, К. С. Іванків. Основи програмування мовою Java: конспект лекцій. Львів: ЛНУ ім. Івана Франка, 2002. 83с.

38. Каліниченко А. В. Порівняння інтегрованих середовищ розробки додатків JAVA із відкритим кодом: ECLIPSE та INTELLIJ IDE. Львів, 2013. 82 с.

39. Spring Framework. URL: https://spring.io/projects/spring-framework (дата звернення: 29.09.2022).

40. И. Портянкин. Swing: Эффективоные пользовательские интерфейсы. – М.: Лори 2011. – 591 с.

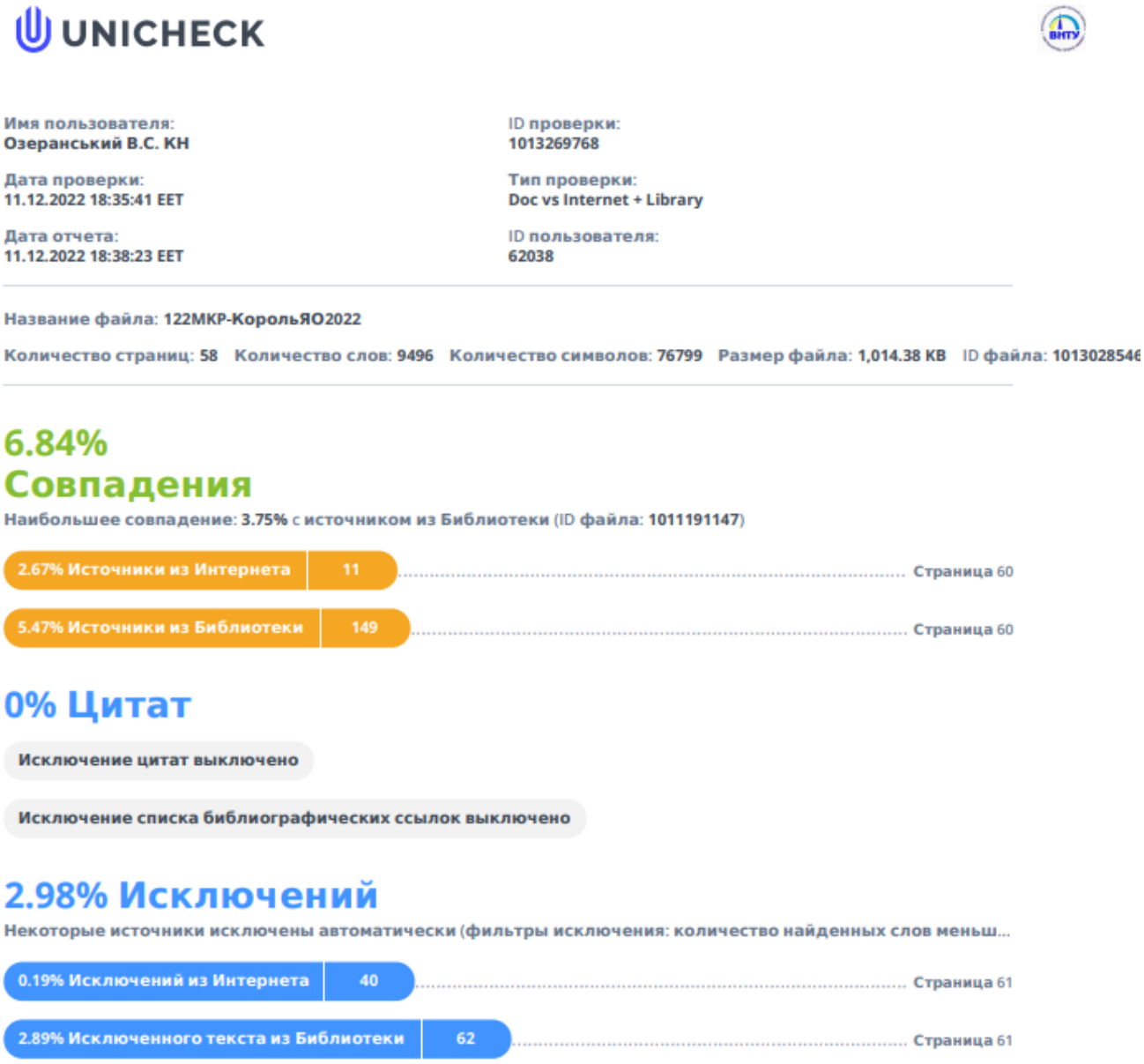
41. Java JDBC Tutorial URL: https://www.javatpoint.com/java-jdbc (дата звернення: 30.09.2022).

42. Пакет Java.util URL: https://wm-help.net/lib/b/book/3683783285/37 (дата звернення: 10.10.2022)

ДОДАТКИ

Додаток А (обов'язковий)

Результат перевірки на плагіат в онлайн-системі UNICHECK



UNICHECK

Имя пользователя:
Озеранський В.С. КН

ID проверки:
1013269768

Дата проверки:
11.12.2022 18:35:41 EET

Тип проверки:
Doc vs Internet + Library

Дата отчета:
11.12.2022 18:38:23 EET

ID пользователя:
62038

Название файла: 122МКР-КорольЯО2022

Количество страниц: 58 Количество слов: 9496 Количество символов: 76799 Размер файла: 1,014.38 KB ID файла: 1013028546

6.84%
Совпадения

Наибольшее совпадение: 3.75% с источником из Библиотеки (ID файла: 1011191147)

2.67% Источники из Интернета 11 ..... Страница 60

5.47% Источники из Библиотеки 149 ..... Страница 60

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

2.98% Исключений

Некоторые источники исключены автоматически (фильтры исключения: количество найденных слов меньш...

0.19% Исключений из Интернета 40 ..... Страница 61

2.89% Исключенного текста из Библиотеки 62 ..... Страница 61

Додаток Б (обов'язковий)

Лістинг програми

Клас MainMenu

```
package GUI;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MainMenu {
    private ImageIcon image;
    private JFrame frame;
    private JLabel label;


    public MainMenu() {
        image = new ImageIcon(getClass().getResource("1.jpg"));
    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Меню");
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 14);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);
        label.setIcon(image);

        JButton loginBtn = new JButton();
        loginBtn.setText("Аворизація");
        loginBtn.setFont(font1);
        label.add(loginBtn, new GridBagConstraints(0, 0, 2, 1, 0.0, 0.9, GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

        loginBtn.addActionListener(new LoginListener());

        JButton registerBtn = new JButton();
        registerBtn.setText("Реєстрація");
        registerBtn.setFont(font1);
        label.add(registerBtn, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(20, 10, 20, 10), 0, 0));

        registerBtn.addActionListener(new RegisterListener());

        frame.add(label);
        frame.setVisible(true);
```

```
    }

    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);

    }
    private class LoginListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new LoginPage().createWindow();
        }
    }
    private class RegisterListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new RegisterPage().createWindow();
        }
    }
}
```

Клас LoginPage

```
package GUI;

import DB.entity.User;
import DB.UserDAO;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class LoginPage {
    private ImageIcon image;
    private JFrame frame;
    private JLabel label;

    private JTextField loginField;
    private JPasswordField passwordField;

    public LoginPage() {
        image = new ImageIcon(getClass().getResource("1.jpg"));
    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Авторизація");
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 14);
```

```
    label = new JLabel();
    label.setLayout(new GridBagLayout());
    label.setVisible(true);
    label.setBackground(Color.BLUE);
    label.setIcon(image);

    JLabel loginLabel = new JLabel();
    loginLabel.setText("Логін");
    loginLabel.setFont(font1);
    label.add(loginLabel, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(30, 10, 10, 10), 0, 0));

    JLabel passwordLabel = new JLabel();
    passwordLabel.setText("Пароль");
    passwordLabel.setFont(font1);
    label.add(passwordLabel, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(30, 10, 10, 10), 0, 0));

    loginField = new JTextField();
    loginField.setFont(font1);
    label.add(loginField, new GridBagConstraints(1, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    passwordField = new JPasswordField();
    passwordField.setFont(font1);
    label.add(passwordField, new GridBagConstraints(1, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));


    JButton loginBtn = new JButton();
    loginBtn.setText("Авторизуватися");
    loginBtn.setFont(font1);
    label.add(loginBtn, new GridBagConstraints(0, 2, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    loginBtn.addActionListener(new LoginListener());

    JButton RestoreBtn = new JButton();
    RestoreBtn.setText("Відновити пароль");
    RestoreBtn.setFont(font1);
    label.add(RestoreBtn, new GridBagConstraints(1, 2, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    RestoreBtn.addActionListener(new RestoreListener());

    frame.add(label);
    frame.setVisible(true);

}
public void closeWindow() {
    frame.setVisible(false);
    frame.remove(label);

}
private class LoginListener implements ActionListener {

    @Override
```

```
        public void actionPerformed(ActionEvent JCom) {
            User authUser = new UserDAO().getUserByLoginAndPassword(loginField.getText(),
passwordField.getText());
            if (authUser!=null){
                Main.setCurrentUser(authUser);
                closeWindow();
                new UserHomepage().createWindow();
            }else{
                JOptionPane.showMessageDialog(null, "Невірний логін або пароль");
                passwordField.setText("");
            }
        }
    }
    private class RestoreListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            User user = new User();
            user.setUsername(loginField.getText());
            Main.setCurrentUser(user);
            closeWindow();
            new RestorePage(loginField.getText()).createWindow();
        }
    }

}
```

Клас MnemonicPhrasePage

```
package GUI;

import DB.PhraseDAO;
import DB.entity.Phrase;
import procesing.PhraseProcessing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.util.*;

public class MnemonicPhrasePage {
    private ImageIcon image;
    private JFrame frame;
    private JLabel label;
    private Phrase phrase;

    private JButton jButton1, jButton2, jButton3, jButton4, jButton5, jButton6, jButton7, jButton8, jButton9,
jButton10, jButton11, jButton12;
    private ArrayList<JButton> buttonArrayList = new ArrayList<>();
    private int[] arr = new int[12];
    private int numberOfW = 0;
    private StringBuilder answer = new StringBuilder();
    private StringBuilder forLabel = new StringBuilder();
```

```
    private JLabel label1;
    private JTextField label2;

    public MnemonicPhrasePage(Phrase phrase) {
        this.phrase=phrase;
        image = new ImageIcon(getClass().getResource("1.jpg"));

    }
    public MnemonicPhrasePage() {
        this.phrase=PhraseProcessing.generatingPhrase();
        image = new ImageIcon(getClass().getResource("1.jpg"));

    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Підтвердження");
        frame.setSize(510, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 13);
        Font font0 = new Font("TimesRoman", Font.BOLD, 10);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);
        label.setIcon(image);
//        frame.addKeyListener(new MyKeyListener());



        String[] pr = phrase.getPhrase().split(" ");

//        StringBuilder sb = new StringBuilder();
//        for (int i = 0; i < phrase.getPhrase().length(); i++) {
//            sb.append("*");
//        }


        label1 = new JLabel();
        label1.setFont(font1);
        label.add(label1, new GridBagConstraints(0, 0, 4, 1, 0.0, 0.9, GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));

        label2 = new JTextField();
        label2.setFont(font0);
        label2.setText("");
        label.add(label2, new GridBagConstraints(0, 1, 4, 1, 0.0, 0.9, GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 5, 10, 5), 0, 0));
        label2.addKeyListener(new MyKeyListener());

        jButton1 = new JButton();
        jButton1.setText(pr[0]);
        jButton1.setFont(font1);
        buttonArrayList.add(jButton1);
```

```
jButton2 = new JButton();
jButton2.setText(pr[1]);
jButton2.setFont(font1);
buttonArrayList.add(jButton2);

jButton3 = new JButton();
jButton3.setText(pr[2]);
jButton3.setFont(font1);
buttonArrayList.add(jButton3);

jButton4 = new JButton();
jButton4.setText(pr[3]);
jButton4.setFont(font1);
buttonArrayList.add(jButton4);

jButton5 = new JButton();
jButton5.setText(pr[4]);
jButton5.setFont(font1);
buttonArrayList.add(jButton5);

jButton6 = new JButton();
jButton6.setText(pr[5]);
jButton6.setFont(font1);
buttonArrayList.add(jButton6);

jButton7 = new JButton();
jButton7.setText(pr[6]);
jButton7.setFont(font1);
buttonArrayList.add(jButton7);

jButton8 = new JButton();
jButton8.setText(pr[7]);
jButton8.setFont(font1);
buttonArrayList.add(jButton8);

jButton9 = new JButton();
jButton9.setText(pr[8]);
jButton9.setFont(font1);
buttonArrayList.add(jButton9);

jButton10 = new JButton();
jButton10.setText(pr[9]);
jButton10.setFont(font1);
buttonArrayList.add(jButton10);

jButton11 = new JButton();
jButton11.setText(pr[10]);
jButton11.setFont(font1);
buttonArrayList.add(jButton11);

jButton12 = new JButton();
jButton12.setText(pr[11]);
jButton12.setFont(font1);
buttonArrayList.add(jButton12);
```

```
        int n = getRandomIntForButton(0);
        int a1 = 12;
        label.add(buttonArrayList.get(n), new GridBagConstraints(0, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(1);
        label.add(buttonArrayList.get(n), new GridBagConstraints(1, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(2);
        label.add(buttonArrayList.get(n), new GridBagConstraints(2, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(3);
        label.add(buttonArrayList.get(n), new GridBagConstraints(3, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());

        n = getRandomIntForButton(4);
        label.add(buttonArrayList.get(n), new GridBagConstraints(0, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(5);
        label.add(buttonArrayList.get(n), new GridBagConstraints(1, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(6);
        label.add(buttonArrayList.get(n), new GridBagConstraints(2, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(7);
        label.add(buttonArrayList.get(n), new GridBagConstraints(3, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());

        n = getRandomIntForButton(8);
        label.add(buttonArrayList.get(n), new GridBagConstraints(0, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(9);
        label.add(buttonArrayList.get(n), new GridBagConstraints(1, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(10);
        label.add(buttonArrayList.get(n), new GridBagConstraints(2, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
```

```
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(11);
        label.add(buttonArrayList.get(n), new GridBagConstraints(3, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, a1, 10, a1), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());


        frame.add(label);
        frame.setVisible(true);

    }

    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);
    }

    private int getRandomIntForButton(int n){
        int rez = -1;
        Random random = new Random();
        boolean present = true;
        while(present){
            rez = random.nextInt(12);
            boolean b = false;
            int i =0;
            while (i<n && !b){
                if (arr[i++]==rez){
                    b = true;
                }
            }
            present=b;
        }
        arr[n]=rez;
        System.out.println(rez);
        return rez;
    }

    private class MyKeyListener implements KeyListener{

        @Override
        public void keyTyped(KeyEvent e) {

        }

        @Override
        public void keyPressed(KeyEvent e) {
            if (e.getKeyCode()==KeyEvent.VK_UP){
                label2.setText(phrase.getPhrase());
            }
        }

        @Override
        public void keyReleased(KeyEvent e) {
            if (e.getKeyCode()==KeyEvent.VK_UP){
                label2.setText("");
            }
```

```
        }
    }


    private class BtnListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            numberOfW++;
            String button = ((JButton) JCom.getSource()).getText();
            answer.append(button).append(" ");
            forLabel.append("*****").append(" ");
            label1.setText(forLabel.toString());
            ((JButton)JCom.getSource()).setEnabled(false);
            if (numberOfW==12){
                if (answer.toString().equals(phrase.getPhrase())){
                    new PhraseDAO().addNewPhrase(phrase);
                    Main.getCurrentUser().setPhrase(phrase);
                    new SpeedPhraseSettingPage().createWindow();
//                    JOptionPane.showMessageDialog(null, "Все гуд");
                }else{
                    JOptionPane.showMessageDialog(null, "Сталася помилка");
                    closeWindow();
                    new MnemonicPhrasePage(phrase).createWindow();
                }
            }
        }
    }
}
```

Клас PhraseProcessing

```
package procesing;

import DB.entity.Phrase;
import DB.entity.Word;
import DB.WordDAO;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.Random;
import java.util.Set;

public class PhraseProcessing {

    public static Phrase generatingPhrase(){
        Set<Word> wordSet = new HashSet<>();
        ArrayList<Word> wordArrayList = new ArrayList<>();
        Random random = new Random();
        while (wordSet.size()<12){
            int r = random.nextInt(75)+1;
            Word word = new WordDAO().getWordById(r);
            boolean b = wordSet.add(word);
            if (b) {
                wordArrayList.add(word);
            }
        }
```

```
        Phrase phrase = new Phrase();
        phrase.setWordArrayList(wordArrayList);
        StringBuilder sb = new StringBuilder();
        for (Word word: wordSet) {
            sb.append(word.getWord()).append(" ");
        }
        phrase.setPhrase(sb.toString());
        return phrase;
    }
}
```

Клас ChangePasswordPage

```
package GUI;

import DB.UserDAO;
import DB.entity.User;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ChangePasswordPage {
    private ImageIcon image;
    private JFrame frame;
    private JLabel label;

    private JTextField loginField;
    private JPasswordField passwordField, passwordField2;

    public ChangePasswordPage() {
        image = new ImageIcon(getClass().getResource("1.jpg"));
    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Зміна пароля");
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 14);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);
        label.setIcon(image);

        JLabel passwordLabel = new JLabel();
        passwordLabel.setText("Новий пароль");
        passwordLabel.setFont(font1);
        label.add(passwordLabel, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
                GridBagConstraints.CENTER, new Insets(30, 10, 10, 10), 0, 0));

        JLabel passwordLabel2 = new JLabel();
        passwordLabel2.setText("Повторити пароль");
```

```
    passwordLabel2.setFont(font1);
    label.add(passwordLabel2, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(30, 10, 10, 10), 0, 0));


    passwordField = new JPasswordField();
    passwordField.setFont(font1);
    label.add(passwordField, new GridBagConstraints(1, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    passwordField2 = new JPasswordField();
    passwordField2.setFont(font1);
    label.add(passwordField2, new GridBagConstraints(1, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    JButton backBtn = new JButton();
    backBtn.setText("Назад");
    backBtn.setFont(font1);
    label.add(backBtn, new GridBagConstraints(0, 3, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    backBtn.addActionListener(new BackListener());

    JButton nextBtn = new JButton();
    nextBtn.setText("Далі");
    nextBtn.setFont(font1);
    label.add(nextBtn, new GridBagConstraints(1, 3, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    nextBtn.addActionListener(new NextListener());

    frame.add(label);
    frame.setVisible(true);

}

public void closeWindow() {
    frame.setVisible(false);
    frame.remove(label);

}

private class NextListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent JCom) {
        if (passwordField.getText().equals("") || passwordField2.getText().equals("")){
            JOptionPane.showMessageDialog(null, "Потрібно заповнити всі поля");
            passwordField.setText("");
            passwordField2.setText("");
            return;
        }

        if (!passwordField.getText().equals(passwordField2.getText())){
            JOptionPane.showMessageDialog(null, "Помилка пароля");
            passwordField.setText("");
            passwordField2.setText("");
```

```
            return;
        }

        Main.getCurrentUser().setPassword(passwordField.getText());
        new UserDAO().updateUserPassword(Main.getCurrentUser());
        Main.setCurrentUser(new
UserDAO().getUserByLoginAndPassword(Main.getCurrentUser().getUsername(),
Main.getCurrentUser().getPassword()));
        closeWindow();
        new UserHomepage().createWindow();
      }
   }

   private class BackListener implements ActionListener {

      @Override
      public void actionPerformed(ActionEvent JCom) {
         closeWindow();
         new MainMenu().createWindow();
      }
   }
}
```

Клас RegisterPage

```
package GUI;

import DB.UserDAO;
import DB.entity.User;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class RegisterPage {
   private ImageIcon image;
   private JFrame frame;
   private JLabel label;

   private JTextField loginField;
   private JPasswordField passwordField, passwordField2;

   public RegisterPage() {
      image = new ImageIcon(getClass().getResource("1.jpg"));
   }

   public void createWindow() {
      frame = new JFrame();
      frame.setTitle("Реєстрація");
      frame.setSize(400, 300);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setLocationRelativeTo(null);

      Font font1 = new Font("TimesRoman", Font.BOLD, 14);
      label = new JLabel();
      label.setLayout(new GridBagLayout());
```

```
    label.setVisible(true);
    label.setBackground(Color.BLUE);
    label.setIcon(image);

    JLabel loginLabel = new JLabel();
    loginLabel.setText("Логін");
    loginLabel.setFont(font1);
    label.add(loginLabel, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(30, 10, 10, 10), 0, 0));

    JLabel passwordLabel = new JLabel();
    passwordLabel.setText("Пароль");
    passwordLabel.setFont(font1);
    label.add(passwordLabel, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(30, 10, 10, 10), 0, 0));

    JLabel passwordLabel2 = new JLabel();
    passwordLabel2.setText("Повторити пароль");
    passwordLabel2.setFont(font1);
    label.add(passwordLabel2, new GridBagConstraints(0, 2, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.CENTER, new Insets(30, 10, 10, 10), 0, 0));


    loginField = new JTextField();
    loginField.setFont(font1);
    label.add(loginField, new GridBagConstraints(1, 0, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    passwordField = new JPasswordField();
    passwordField.setFont(font1);
    label.add(passwordField, new GridBagConstraints(1, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    passwordField2 = new JPasswordField();
    passwordField2.setFont(font1);
    label.add(passwordField2, new GridBagConstraints(1, 2, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    JButton backBtn = new JButton();
    backBtn.setText("Назад");
    backBtn.setFont(font1);
    label.add(backBtn, new GridBagConstraints(0, 3, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    backBtn.addActionListener(new BackListener());

    JButton nextBtn = new JButton();
    nextBtn.setText("Далі");
    nextBtn.setFont(font1);
    label.add(nextBtn, new GridBagConstraints(1, 3, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

    nextBtn.addActionListener(new NextListener());
    frame.add(label);
    frame.setVisible(true);

}
```

```
    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);

    }

    private class NextListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            if (loginField.getText().equals("") || passwordField.getText().equals("") ||
passwordField2.getText().equals("")){
                JOptionPane.showMessageDialog(null, "Потрібно заповнити всі поля");
                passwordField.setText("");
                passwordField2.setText("");
                return;
            }
//            System.out.println(passwordField.getText());
//            System.out.println(passwordField2.getText());
            if (!passwordField.getText().equals(passwordField2.getText())){
                JOptionPane.showMessageDialog(null, "Помилка пароля");
                passwordField.setText("");
                passwordField2.setText("");
                return;
            }
            if (new UserDAO().isUserPresent(loginField.getText())){
                JOptionPane.showMessageDialog(null, "Недопустиме ім'я корстувача");
                passwordField.setText("");
                passwordField2.setText("");
                return;
            }
            User user = new User();
            user.setUsername(loginField.getText());
            user.setPassword(passwordField.getText());
            Main.setCurrentUser(user);
            closeWindow();
            new MnemonicPhrasePage().createWindow();
        }
    }

    private class BackListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            closeWindow();
            new MainMenu().createWindow();
        }
    }

}
```

Клас RestorePage

```
package GUI;

import DB.PhraseDAO;
```

```
import DB.entity.Phrase;
import procesing.PhraseProcessing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.util.ArrayList;
import java.util.Random;

public class RestorePage {
    private ImageIcon image;
    private JFrame frame;
    private JLabel label;
    private Phrase phrase;

    private JButton jButton1, jButton2, jButton3, jButton4, jButton5, jButton6, jButton7, jButton8, jButton9,
jButton10, jButton11, jButton12;
    private ArrayList<JButton> buttonArrayList = new ArrayList<>();
    private int[] arr = new int[12];
    private int numberOfW = 0;
    private StringBuilder answer = new StringBuilder();
    private StringBuilder forLabel = new StringBuilder();

    private JLabel label1;
    private JTextField label2;

    public RestorePage(Phrase phrase) {
        this.phrase=phrase;
        image = new ImageIcon(getClass().getResource("1.jpg"));

    }
    public RestorePage(String login) {
        this.phrase= new PhraseDAO().getPhraseByLogin(login);
        image = new ImageIcon(getClass().getResource("1.jpg"));

    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Відновлення пароля");
        frame.setSize(500, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 14);
        Font font0 = new Font("TimesRoman", Font.BOLD, 11);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);
        label.setIcon(image);

        String[] pr = phrase.getPhrase().split(" ");
```

```
label1 = new JLabel();
label1.setFont(font1);
label.add(label1, new GridBagConstraints(0, 0, 4, 1, 0.0, 0.9, GridBagConstraints.CENTER,
      GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));


jButton1 = new JButton();
jButton1.setText(pr[0]);
jButton1.setFont(font1);
buttonArrayList.add(jButton1);

jButton2 = new JButton();
jButton2.setText(pr[1]);
jButton2.setFont(font1);
buttonArrayList.add(jButton2);

jButton3 = new JButton();
jButton3.setText(pr[2]);
jButton3.setFont(font1);
buttonArrayList.add(jButton3);

jButton4 = new JButton();
jButton4.setText(pr[3]);
jButton4.setFont(font1);
buttonArrayList.add(jButton4);

jButton5 = new JButton();
jButton5.setText(pr[4]);
jButton5.setFont(font1);
buttonArrayList.add(jButton5);

jButton6 = new JButton();
jButton6.setText(pr[5]);
jButton6.setFont(font1);
buttonArrayList.add(jButton6);

jButton7 = new JButton();
jButton7.setText(pr[6]);
jButton7.setFont(font1);
buttonArrayList.add(jButton7);

jButton8 = new JButton();
jButton8.setText(pr[7]);
jButton8.setFont(font1);
buttonArrayList.add(jButton8);

jButton9 = new JButton();
jButton9.setText(pr[8]);
jButton9.setFont(font1);
buttonArrayList.add(jButton9);

jButton10 = new JButton();
jButton10.setText(pr[9]);
jButton10.setFont(font1);
buttonArrayList.add(jButton10);

jButton11 = new JButton();
```

```
        jButton11.setText(pr[10]);
        jButton11.setFont(font1);
        buttonArrayList.add(jButton11);

        jButton12 = new JButton();
        jButton12.setText(pr[11]);
        jButton12.setFont(font1);
        buttonArrayList.add(jButton12);


        int n = getRandomIntForButton(0);
        label.add(buttonArrayList.get(n), new GridBagConstraints(0, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(1);
        label.add(buttonArrayList.get(n), new GridBagConstraints(1, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(2);
        label.add(buttonArrayList.get(n), new GridBagConstraints(2, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(3);
        label.add(buttonArrayList.get(n), new GridBagConstraints(3, 2, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());

        n = getRandomIntForButton(4);
        label.add(buttonArrayList.get(n), new GridBagConstraints(0, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(5);
        label.add(buttonArrayList.get(n), new GridBagConstraints(1, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(6);
        label.add(buttonArrayList.get(n), new GridBagConstraints(2, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(7);
        label.add(buttonArrayList.get(n), new GridBagConstraints(3, 3, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());

        n = getRandomIntForButton(8);
        label.add(buttonArrayList.get(n), new GridBagConstraints(0, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
```

```
        n = getRandomIntForButton(9);
        label.add(buttonArrayList.get(n), new GridBagConstraints(1, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(10);
        label.add(buttonArrayList.get(n), new GridBagConstraints(2, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());
        n = getRandomIntForButton(11);
        label.add(buttonArrayList.get(n), new GridBagConstraints(3, 4, 1, 1, 0.0, 0.9,
GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(10, 10, 10, 10), 0, 0));
        buttonArrayList.get(n).addActionListener(new BtnListener());


        frame.add(label);
        frame.setVisible(true);

    }

    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);
    }

    private int getRandomIntForButton(int n){
        int rez = -1;
        Random random = new Random();
        boolean present = true;
        while(present){
            rez = random.nextInt(12);
            boolean b = false;
            int i =0;
            while (i<n && !b){
                if (arr[i++]==rez){
                    b = true;
                }
            }
            present=b;
        }
        arr[n]=rez;
        System.out.println(rez);
        return rez;
    }

    private class BtnListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            numberOfW++;
            String button = ((JButton) JCom.getSource()).getText();
            answer.append(button).append(" ");
            forLabel.append("*****").append(" ");
            label1.setText(forLabel.toString());
            ((JButton)JCom.getSource()).setEnabled(false);
```

```
            if (numberOfW==12){
                if (answer.toString().equals(phrase.getPhrase())){
                    closeWindow();
                    Main.getCurrentUser().setPhrase(phrase);
                    new ChangePasswordPage().createWindow();
//                    JOptionPane.showMessageDialog(null, "Все гуд");
                }else{
                    JOptionPane.showMessageDialog(null, "Сталася помилка");
                    closeWindow();
                    new RestorePage(phrase).createWindow();
                }
            }
        }
    }
}
```

Клас SpeedPhraseCheckPage

```
package GUI;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.util.ArrayList;

public class SpeedPhraseCheckPage {
    private JFrame frame;
    private JLabel label;

    private JPasswordField loginField;
    private JButton okBtn;
    private ArrayList<Long> pressedTime = new ArrayList<>();
    private ArrayList<Long> releasedTime = new ArrayList<>();


    public SpeedPhraseCheckPage() {

    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Speed phrase check");
        frame.setSize(250, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 14);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);


        loginField = new JPasswordField(60);
```

```
    loginField.setFont(font1);
    label.add(loginField, new GridBagConstraints(0, 0, 3, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

    loginField.addKeyListener(new MyKeyListener());

    okBtn = new JButton();
    okBtn.setText("          Далі          ");
    okBtn.setFont(font1);
    label.add(okBtn, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
            GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

    okBtn.addActionListener(new OkListener());

    frame.add(label);
    frame.setVisible(true);

}

public void closeWindow() {
    frame.setVisible(false);
    frame.remove(label);

}

private class MyKeyListener extends KeyAdapter {


    @Override
    public void keyTyped(KeyEvent e) {
        super.keyTyped(e);
    }

    @Override
    public void keyPressed(KeyEvent e) {
        if (e.getKeyChar() == KeyEvent.VK_ENTER) {
            okBtn.doClick();
        } else {
            pressedTime.add(System.currentTimeMillis());
        }
    }

    @Override
    public void keyReleased(KeyEvent e) {
        if (e.getKeyCode() != KeyEvent.VK_ENTER)
            releasedTime.add(System.currentTimeMillis());
    }
}

private class OkListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent JCom) {
        long time1 = releasedTime.get(releasedTime.size() - 1) - pressedTime.get(0);
        int myTime1 = (int) (time1 / releasedTime.size());

        long time2 = 0;
```

```
            for (int i = 0; i < releasedTime.size(); i++) {
                time2 = time2 + releasedTime.get(i) - pressedTime.get(i);
            }
            int myTime2 = (int) (time2 / releasedTime.size());
            System.out.println(myTime1);
            System.out.println(Main.getCurrentUser().getHand1());
            System.out.println(myTime2);
            System.out.println(Main.getCurrentUser().getHand2());
            if                    (((Main.getCurrentUser().getHand1()*0.65<myTime1)                    &&
(Main.getCurrentUser().getHand1()*1.3>myTime1))&&
                    ((Main.getCurrentUser().getHand2()*0.65<myTime2)                    &&
(Main.getCurrentUser().getHand2()*1.3>myTime2))){
                JOptionPane.showMessageDialog(null, "Доступ дозволено");
                closeWindow();
            }else{
                JOptionPane.showMessageDialog(null, "Доступ заборонено");
                closeWindow();
                new SpeedPhraseCheckPage().createWindow();
            }

        }
    }
}
```

Клас SpeedPhraseSettingPage

```
package GUI;

import DB.UserDAO;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.util.ArrayList;

public class SpeedPhraseSettingPage {
    private JFrame frame;
    private JLabel label;

    private JPasswordField loginField;
    private JButton okBtn;
    private ArrayList<Long> pressedTime = new ArrayList<>();
    private ArrayList<Long> releasedTime = new ArrayList<>();


    public SpeedPhraseSettingPage() {

    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Speed phrase");
        frame.setSize(250, 150);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    frame.setLocationRelativeTo(null);

    Font font1 = new Font("TimesRoman", Font.BOLD, 14);
    label = new JLabel();
    label.setLayout(new GridBagLayout());
    label.setVisible(true);
    label.setBackground(Color.BLUE);


    loginField = new JPasswordField(60);
    loginField.setFont(font1);
    label.add(loginField, new GridBagConstraints(0, 0, 3, 1, 0.0, 0.9, GridBagConstraints.CENTER,
          GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

    loginField.addKeyListener(new MyKeyListener());

    okBtn = new JButton();
    okBtn.setText("          Далі          ");
    okBtn.setFont(font1);
    label.add(okBtn, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
          GridBagConstraints.HORIZONTAL, new Insets(20, 10, 10, 10), 0, 0));

    okBtn.addActionListener(new OkListener());

    frame.add(label);
    frame.setVisible(true);

}

public void closeWindow() {
    frame.setVisible(false);
    frame.remove(label);

}

private class MyKeyListener extends KeyAdapter {


    @Override
    public void keyTyped(KeyEvent e) {
        super.keyTyped(e);
    }

    @Override
    public void keyPressed(KeyEvent e) {
        if (e.getKeyChar() == KeyEvent.VK_ENTER) {
            okBtn.doClick();
        } else {
            pressedTime.add(System.currentTimeMillis());
        }
    }

    @Override
    public void keyReleased(KeyEvent e) {
        if (e.getKeyCode() != KeyEvent.VK_ENTER)
            releasedTime.add(System.currentTimeMillis());
    }
```

```
    }

    private class OkListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            long time1 = releasedTime.get(releasedTime.size() - 1) - pressedTime.get(0);
            int myTime1 = (int) (time1 / releasedTime.size());

            Main.getCurrentUser().setHand1(myTime1);

            long time2 = 0;
            for (int i = 0; i < releasedTime.size(); i++) {
                time2 = time2 + releasedTime.get(i) - pressedTime.get(i);
            }
            int myTime2 = (int) (time2 / releasedTime.size());
            Main.getCurrentUser().setHand2(myTime2);
            Main.getCurrentUser().setSpeed(loginField.getText());
            new UserDAO().addNewUser(Main.getCurrentUser());
            new UserHomepage().createWindow();

        }
    }
}
```

Клас UserHomepage

```
package GUI;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class UserHomepage {
    private ImageIcon image;
    private JFrame frame;
    private JLabel label;

    public UserHomepage() {
        image = new ImageIcon(getClass().getResource("1.jpg"));
    }

    public void createWindow() {
        frame = new JFrame();
        frame.setTitle("Меню користувача");
        frame.setSize(300, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);

        Font font1 = new Font("TimesRoman", Font.BOLD, 14);
        label = new JLabel();
        label.setLayout(new GridBagLayout());
        label.setVisible(true);
        label.setBackground(Color.BLUE);
        label.setIcon(image);
```

```
        JButton actionOne = new JButton();
        actionOne.setText("Дія 1");
        actionOne.setFont(font1);
        label.add(actionOne, new GridBagConstraints(0, 0, 2, 1, 0.0, 0.9, GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(30, 10, 10, 10), 0, 0));

        actionOne.addActionListener(new Action1Listener());

        JButton actionTwo = new JButton();
        actionTwo.setText("Дія 2");
        actionTwo.setFont(font1);
        label.add(actionTwo, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.9, GridBagConstraints.CENTER,
                GridBagConstraints.HORIZONTAL, new Insets(20, 10, 20, 10), 0, 0));

        actionTwo.addActionListener(new Action2Listener());

        frame.add(label);
        frame.setVisible(true);

    }

    public void closeWindow() {
        frame.setVisible(false);
        frame.remove(label);

    }

    private class Action1Listener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            new SpeedPhraseCheckPage().createWindow();
        }
    }
    private class Action2Listener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent JCom) {
            new SpeedPhraseCheckPage().createWindow();
        }
    }
}
```

Додаток В (обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АВТОРИЗАЦІЇ КОРИСТУВАЧА З ВИКОРИСТАННЯМ КЛАВІАТУРНОГО ПОЧЕРКУ»

Виконала: студентка 2-го курсу,
групи 1КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)

Король Я. О.
(прізвище та ініціали)

Керівник: к.т.н., ст. викл.
Озеранський В.С.
(прізвище та ініціали)
«____» _____2022 р.

Вінниця ВНТУ - 2022 рік

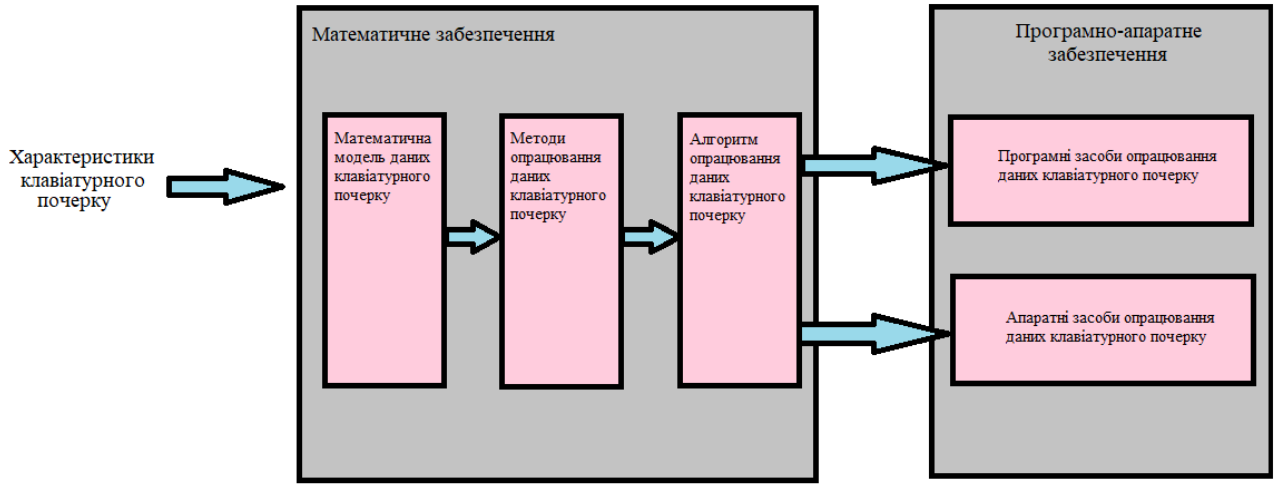


Рисунок В.1 – Схема етапів інформаційної технології авторизації користувача з використанням клавіатурного почерку

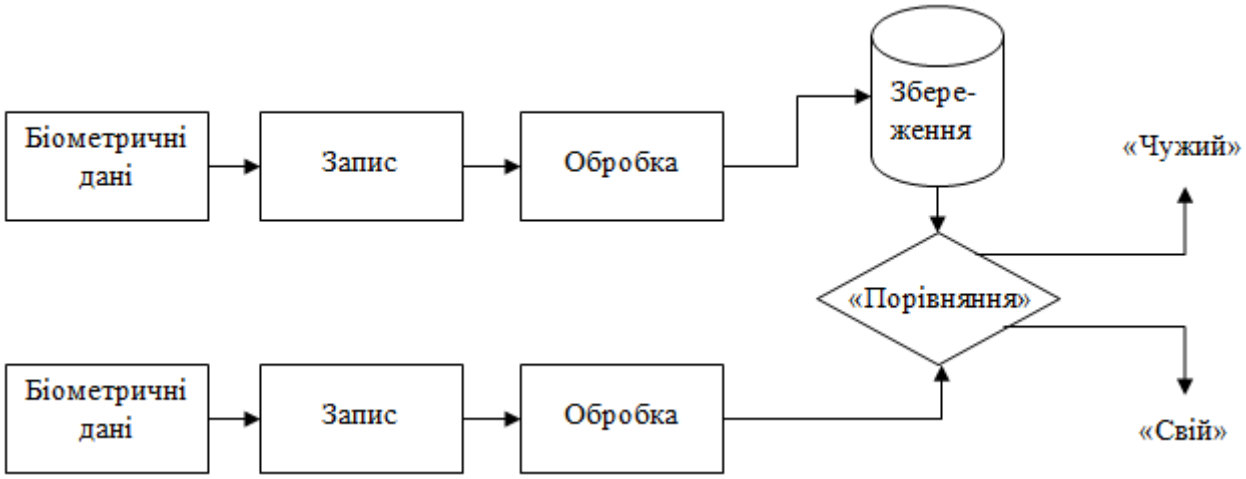


Рисунок В.2 – Схема архітектури системи виявлення підміни користувача

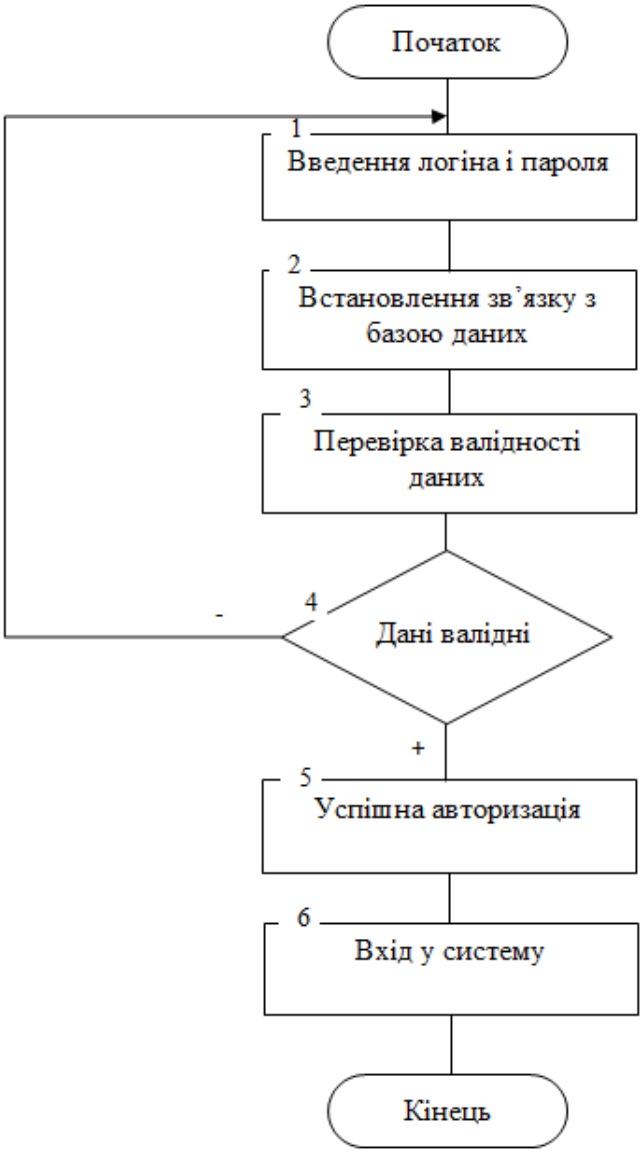


Рисунок В.3 – Схема алгоритму авторизації користувача з з самостійним вводом логіна і пароля

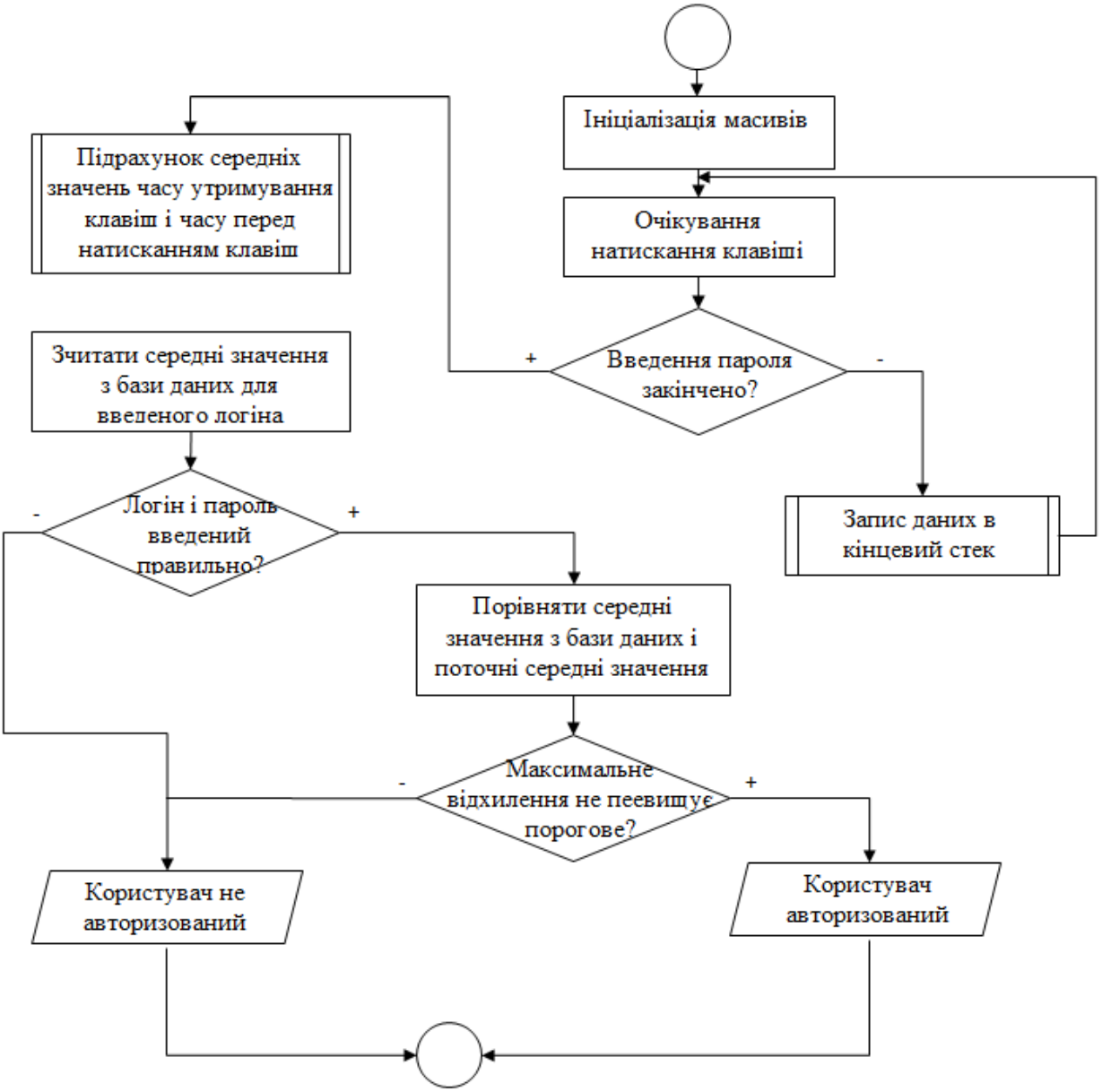


Рисунок В.4 – Схема алгоритму авторизації користувача з використанням клавіатурного почерку

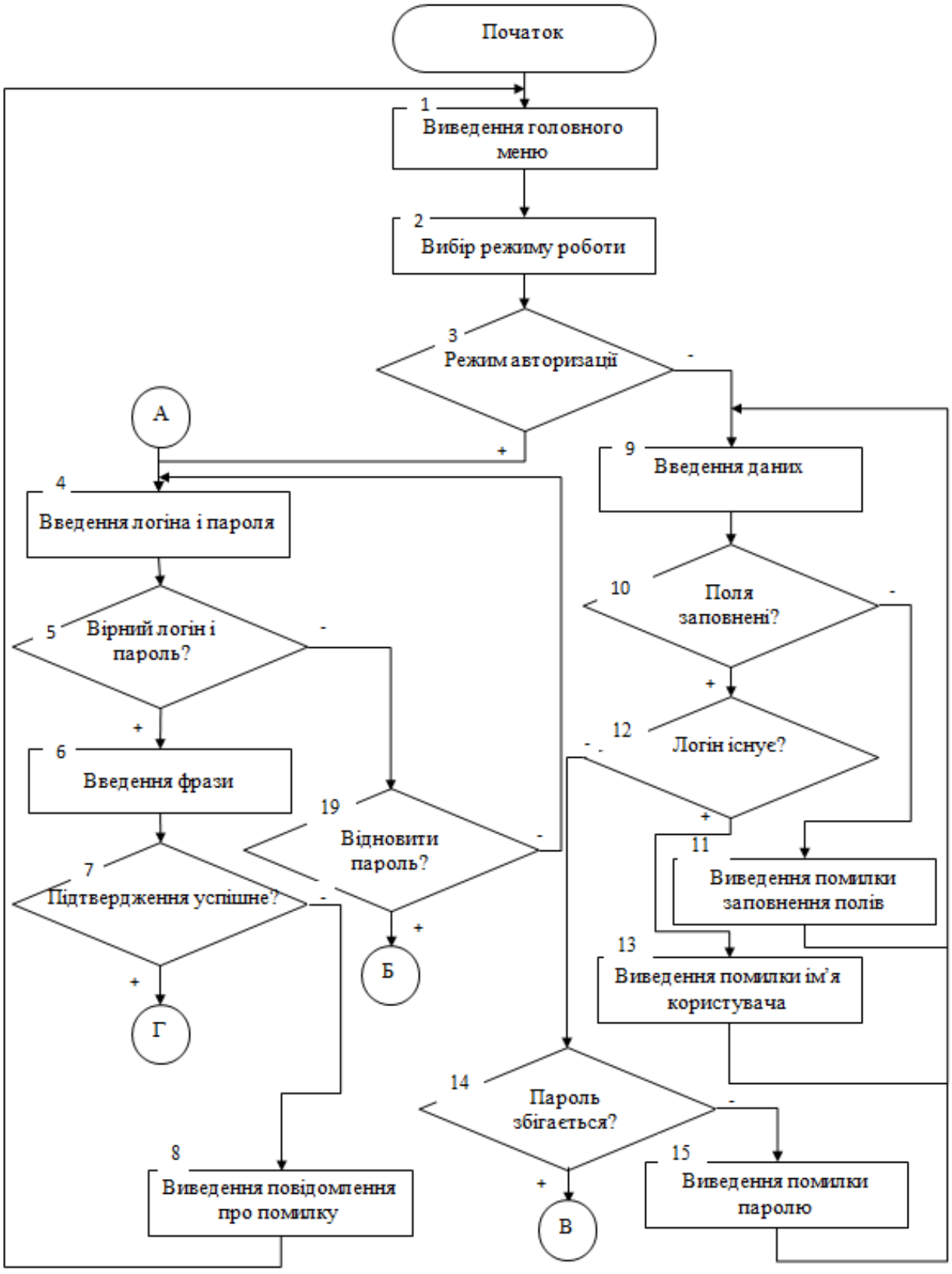


Рисунок В.5 – Схема загального алгоритму інформаційної технології авторизаціїкористувача з використаням клавіатурного почерку

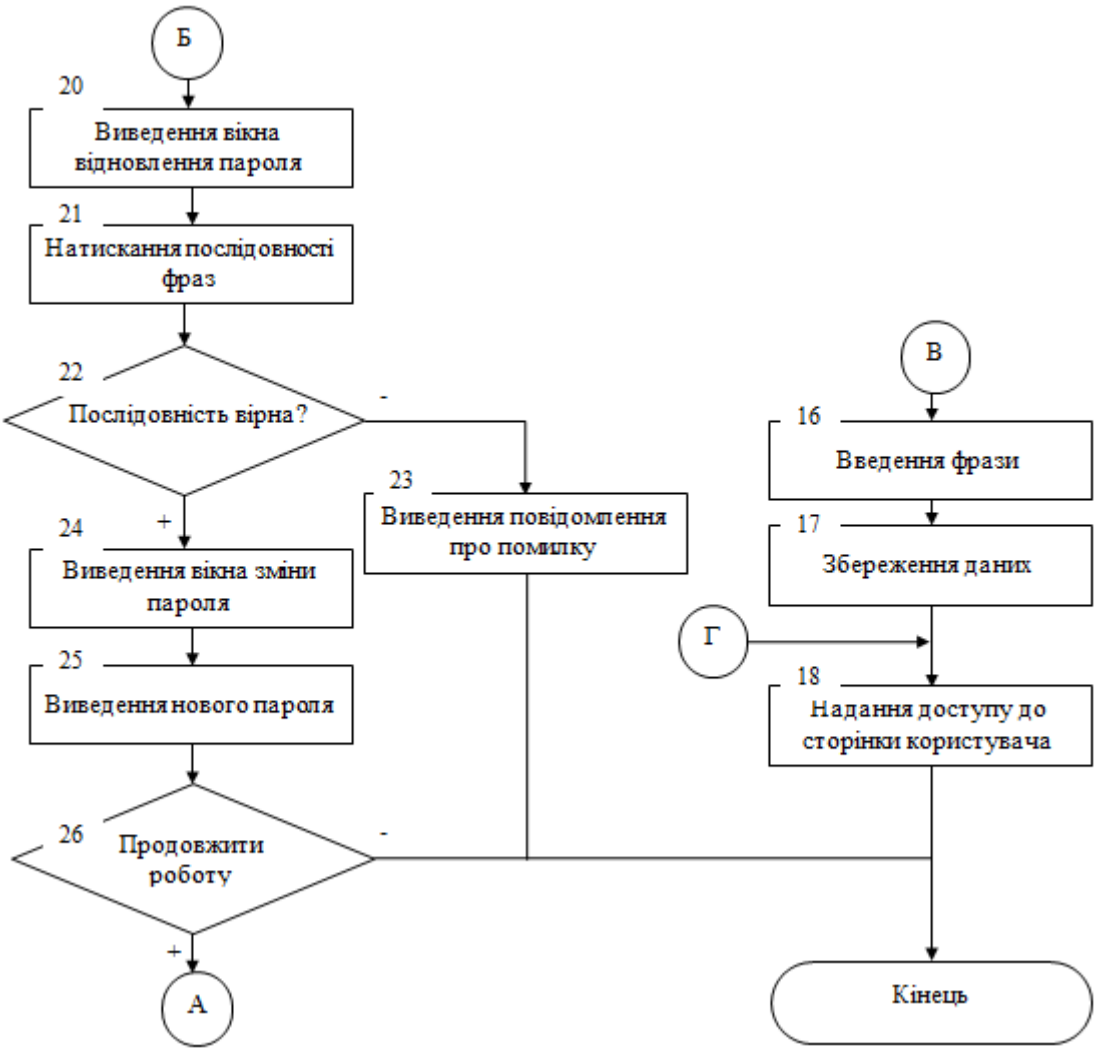


Рисунок В.5 – Продовження

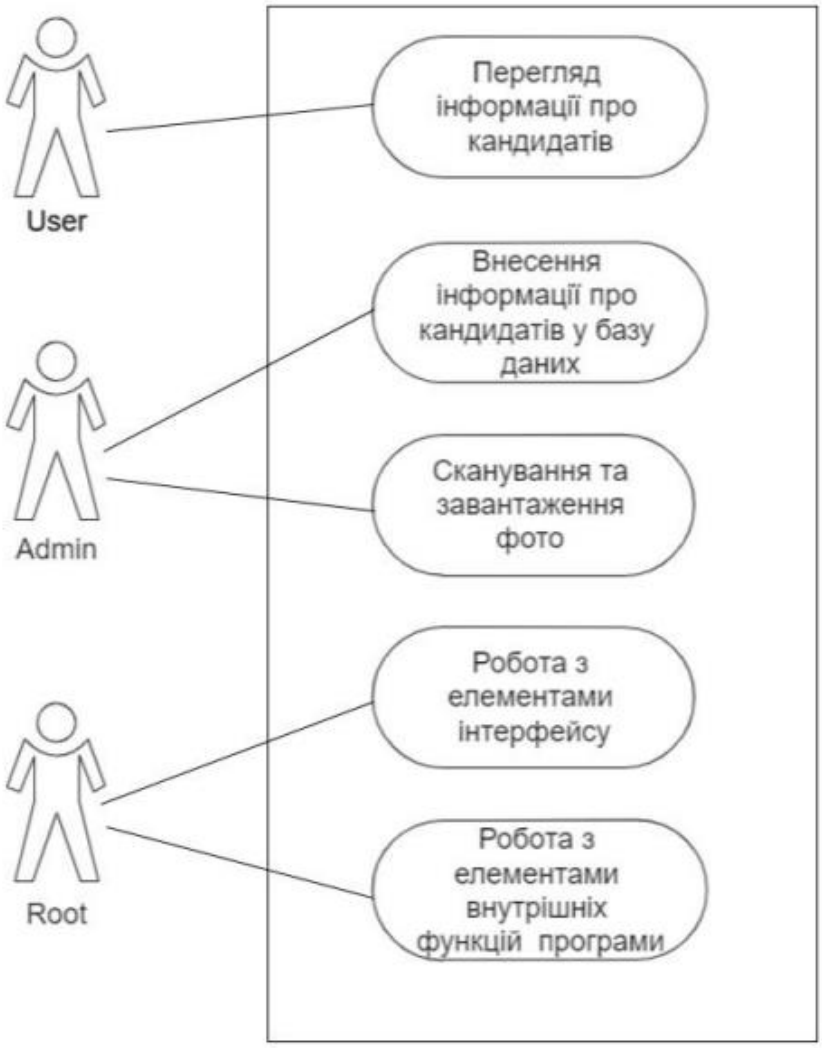


Рисунок В.6 – UML-діаграма прецедентів для програми авторизації користувача з використанням клавіатурного почерку

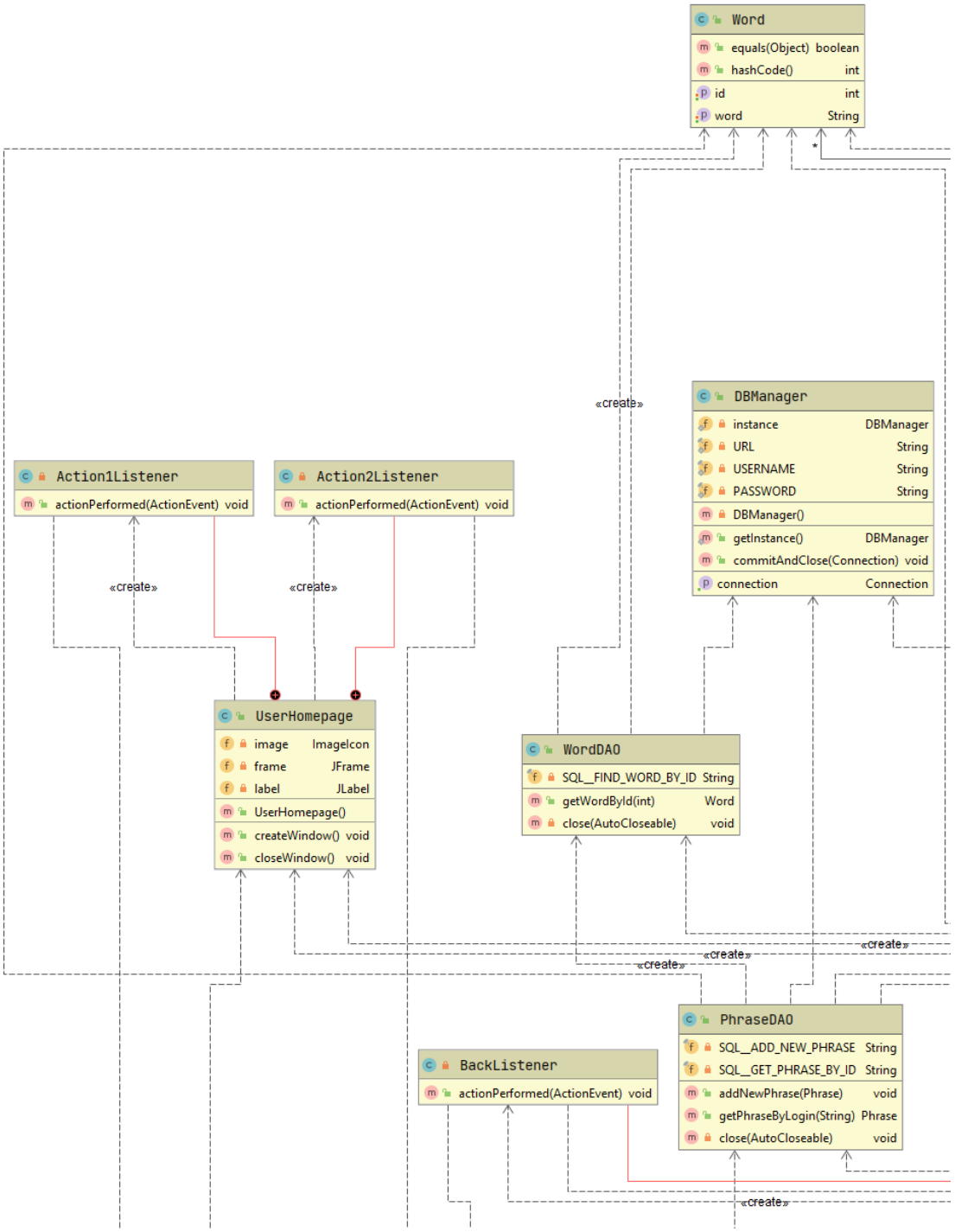


Рисунок В.7 – UML-діаграма класів інформаційної технології авторизації користувача з використанням клавіатурного почерку

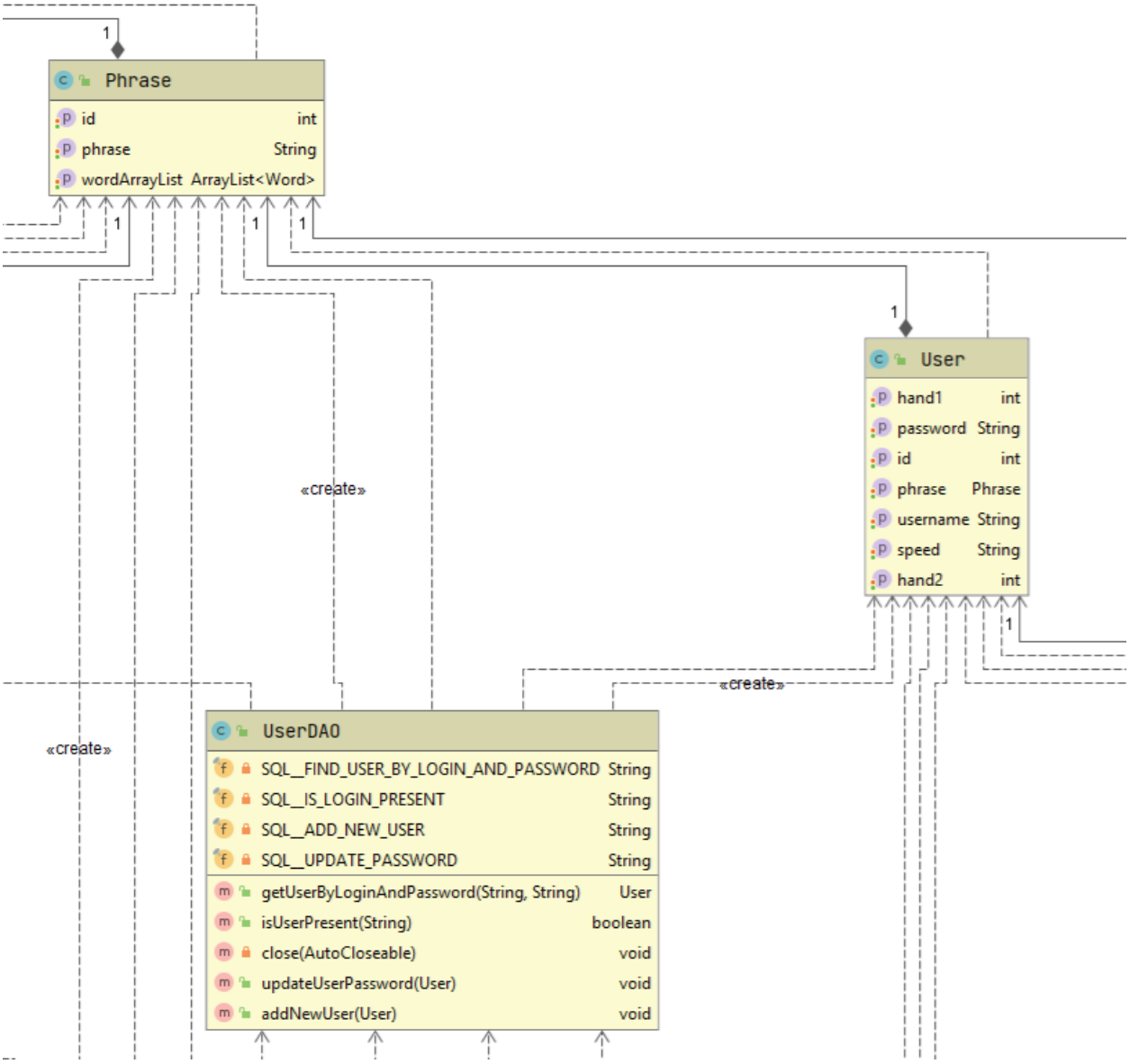


Рисунок В.7 – Продовження

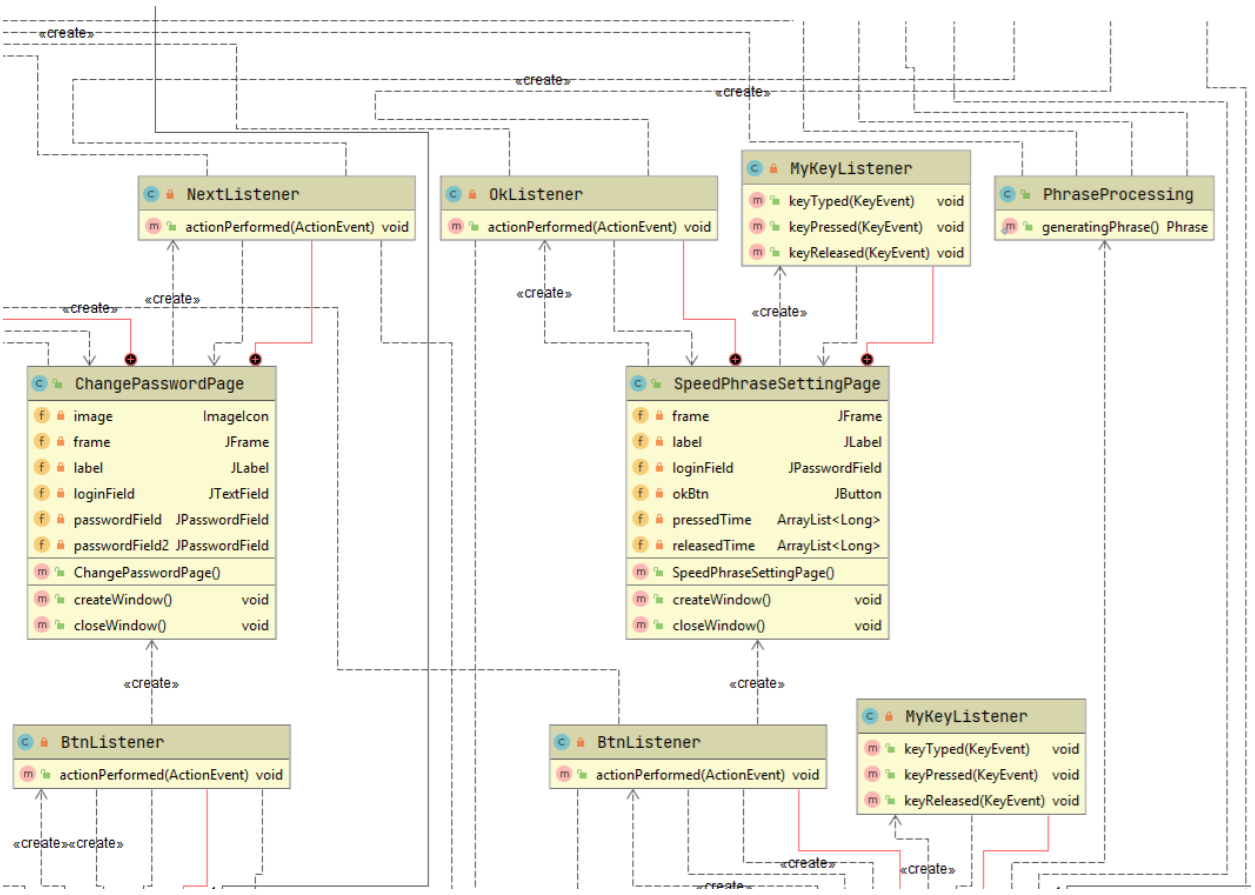


Рисунок В.7 – Продовження

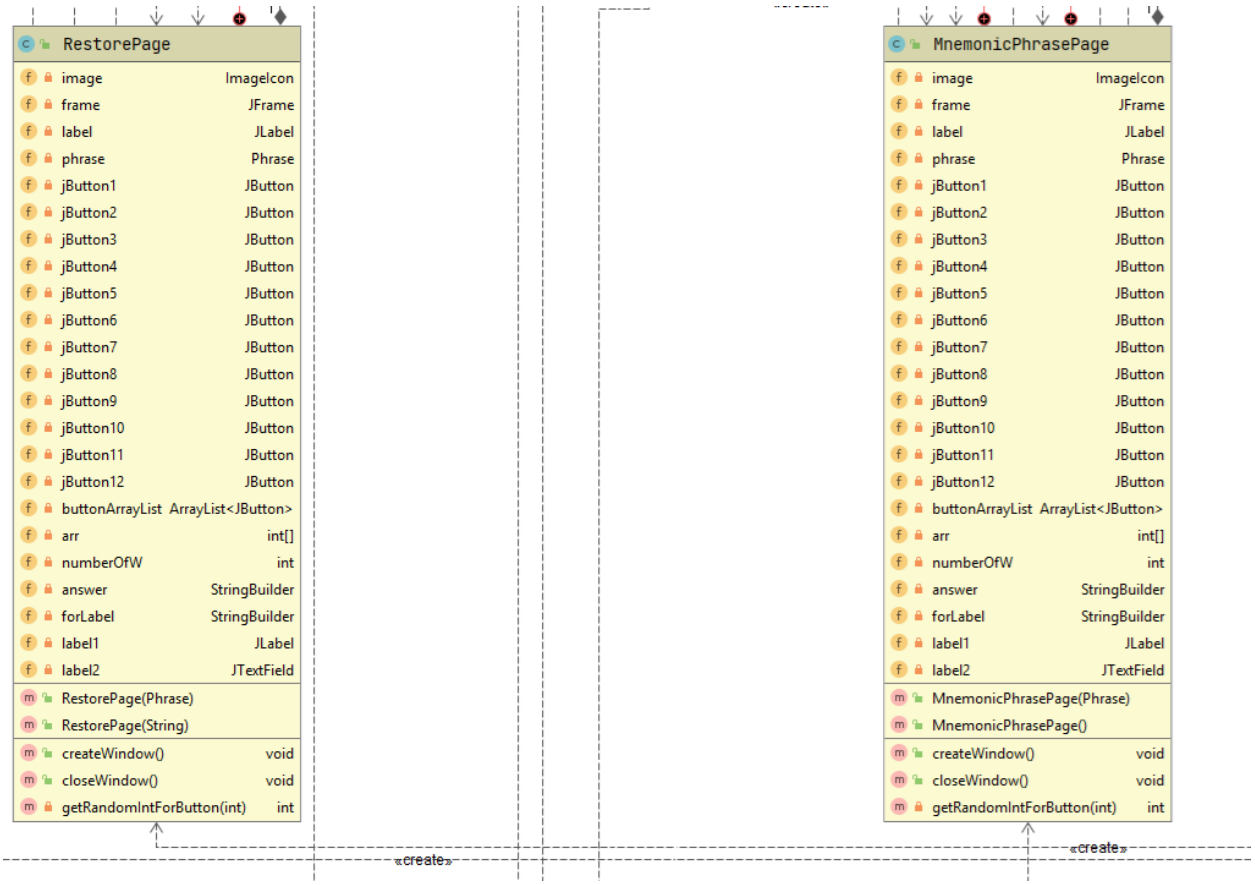


Рисунок В.7 – Продовження

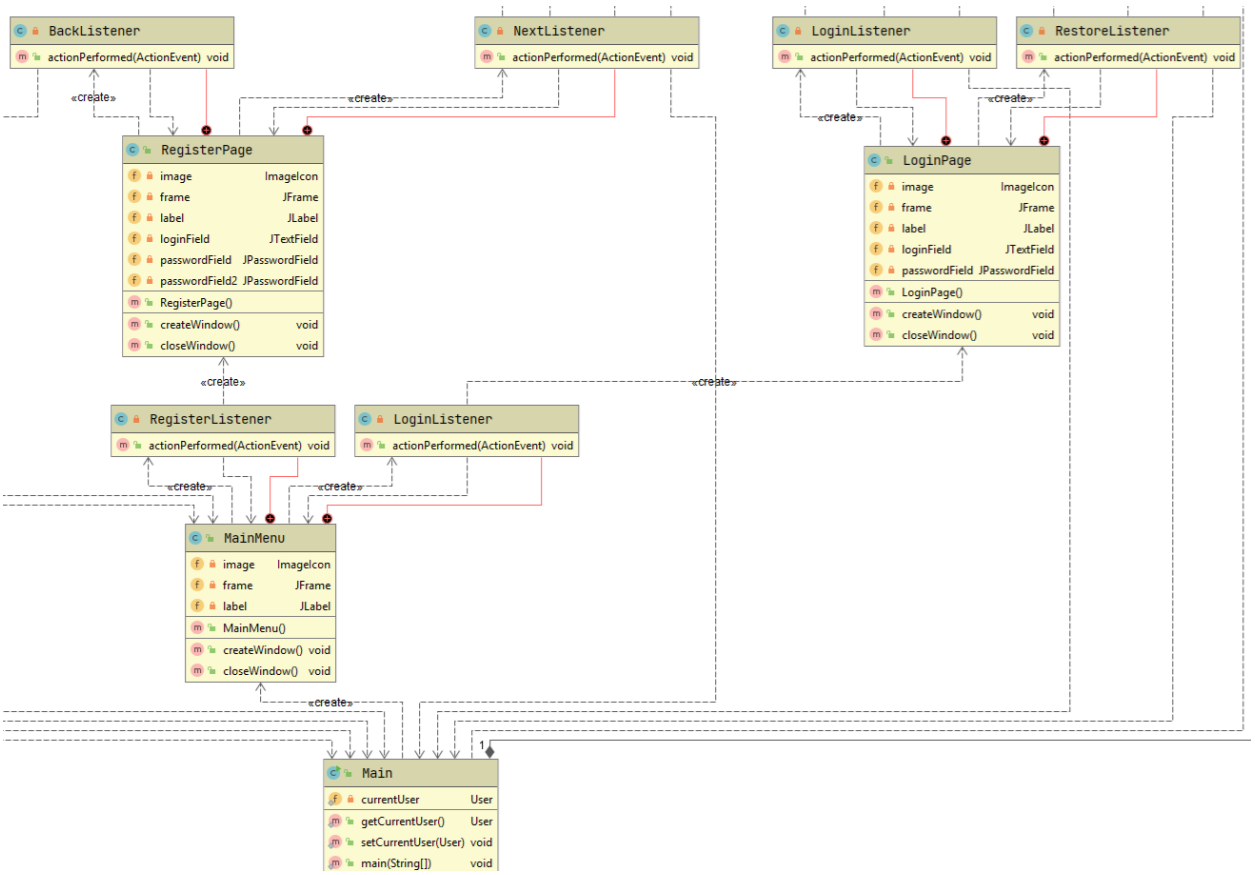


Рисунок В.7 – Продовження

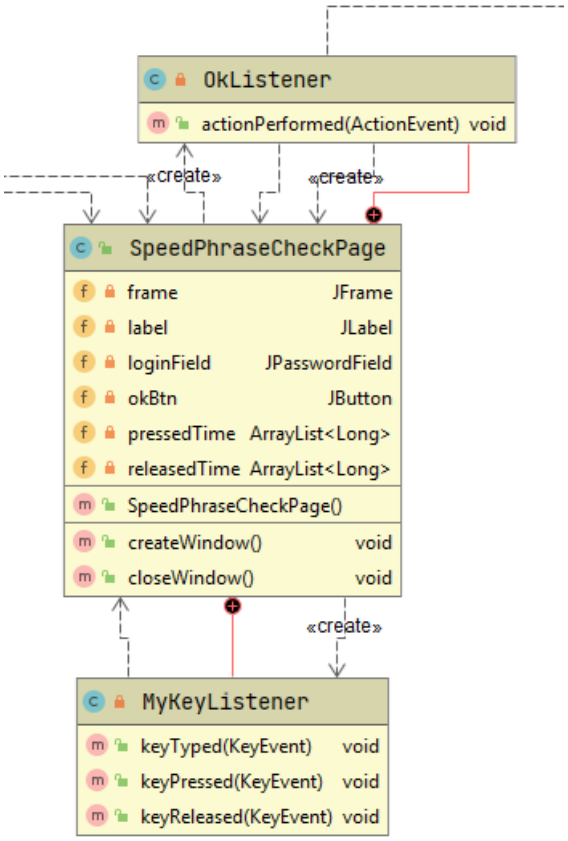


Рисунок В.7 – Продовження

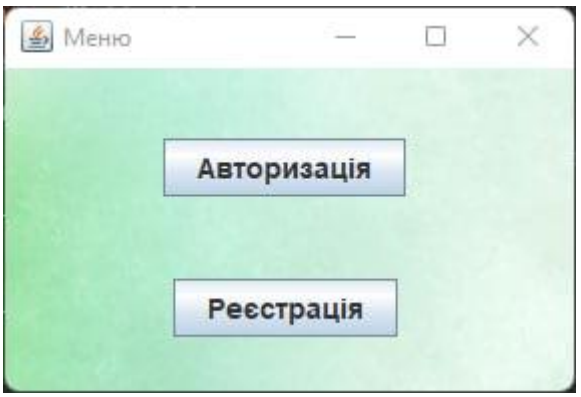


Рисунок В.8 – Загальний вигляд інтерфейсного вікна початкової активності

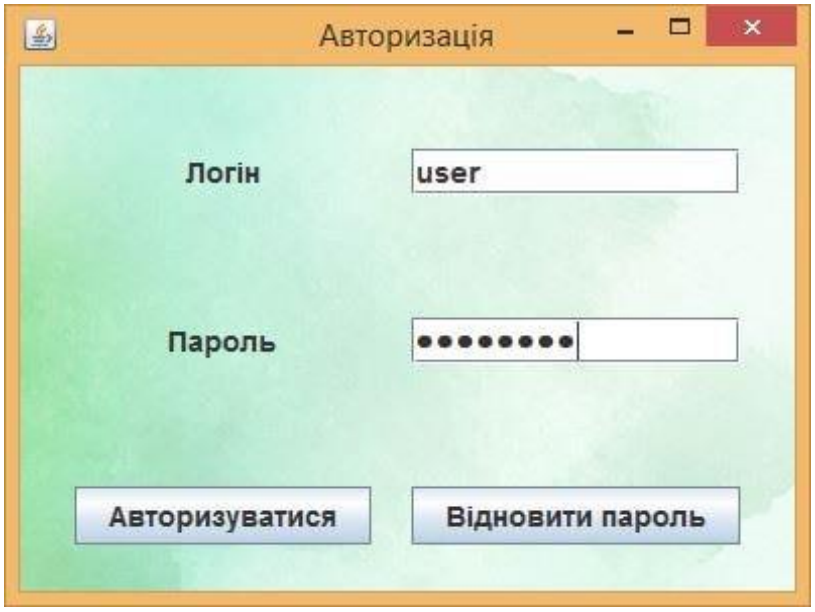


Рисунок В.9 – Загальний вигляд інтерфейсного вікна авторизації

Рисунок В.10 – Загальний вигляд інтерфейсного вікна для введення фрази

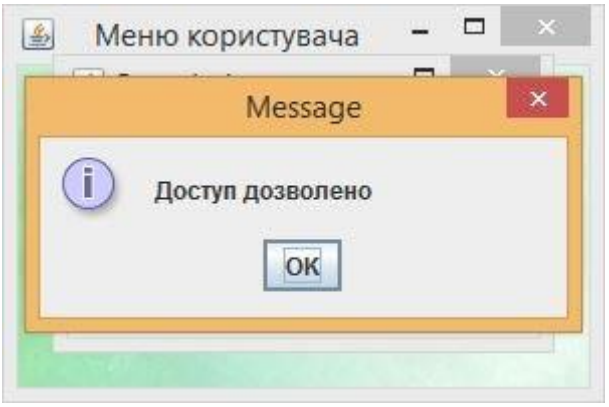


Рисунок В.11 – Загальний вигляд діалогового вікна надання доступу

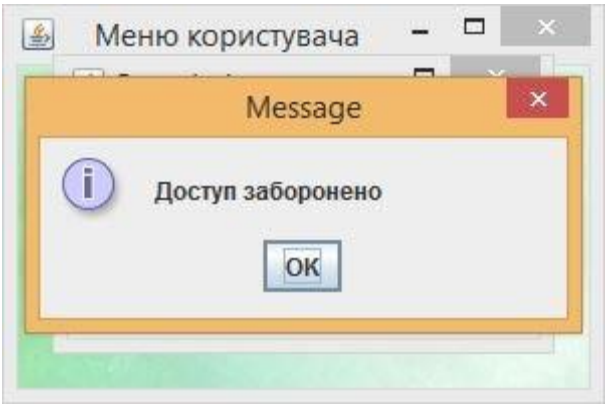


Рисунок В.12 – Загальний вигляд діалогового вікна заборони доступу

Рисунок В.13 – Загальний вигляд інтерфейсного вікна для відновлення пароля

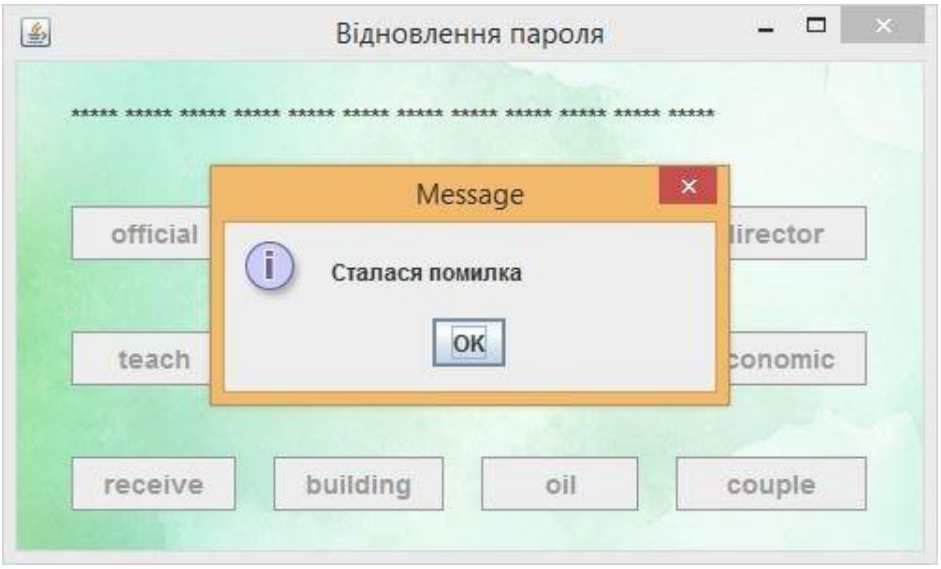


Рисунок В.14 – Загальний вигляд інтерфейсного вікна виведення помилки

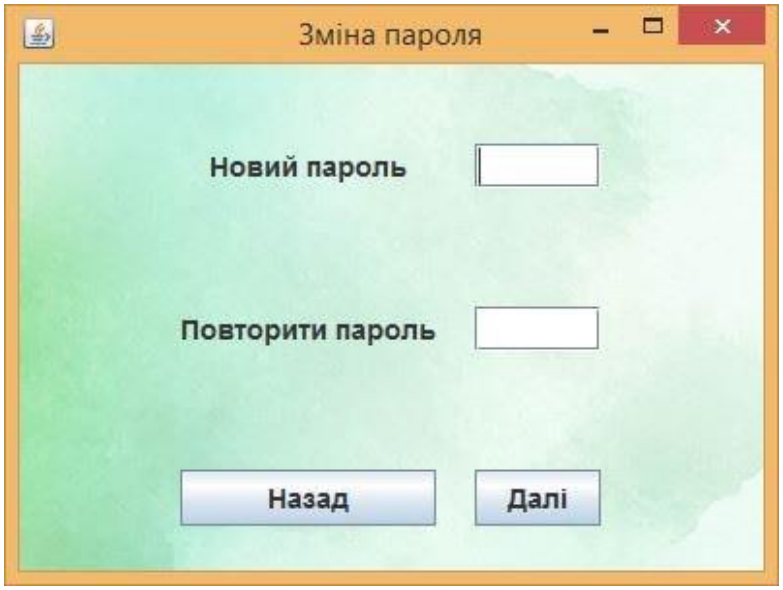


Рисунок В.15 – Загальний вигляд інтерфейсного вікна зміни пароля

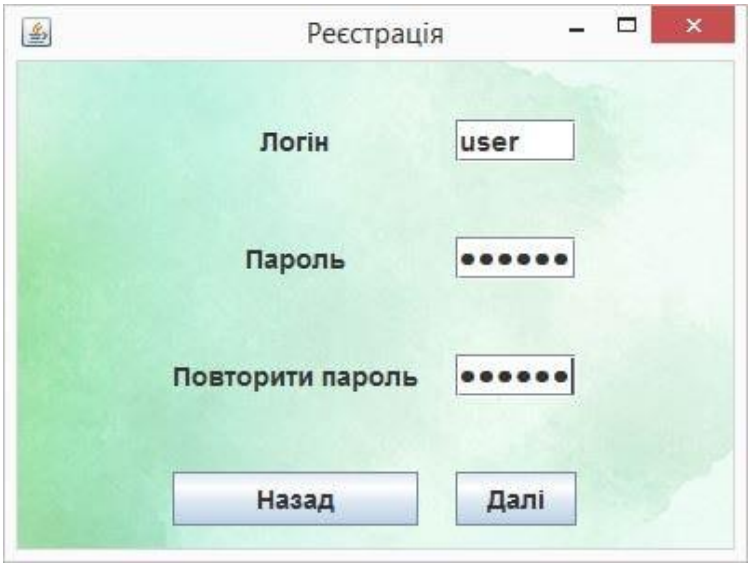


Рисунок В.16 – Загальний вигляд інтерфейсного вікна реєстрації

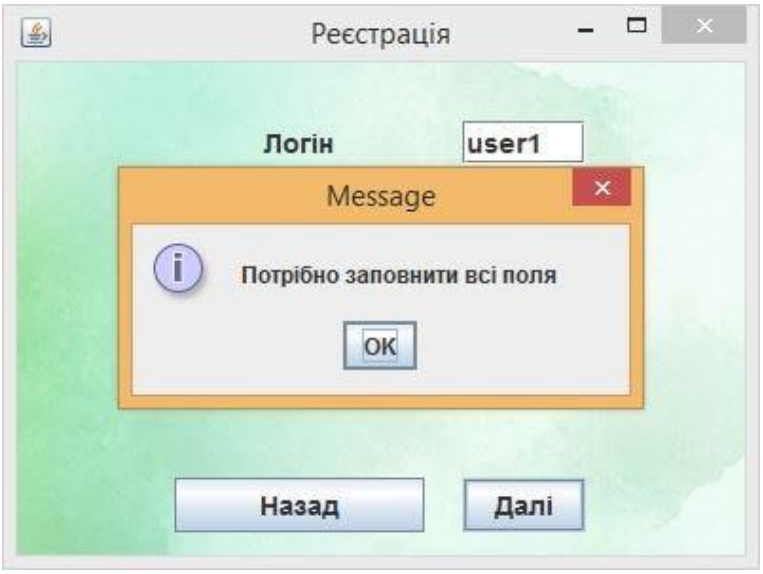


Рисунок В.17 – Загальний вигляд інтерфейсного вікна виведення помилки заповнення полів

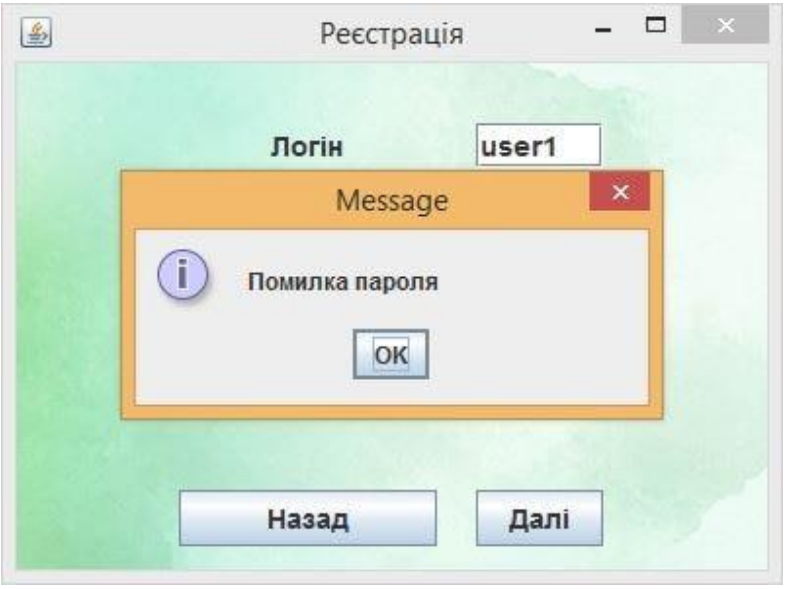


Рисунок В.18 – Загальний вигляд інтерфейсного вікна виведення помилки пароля

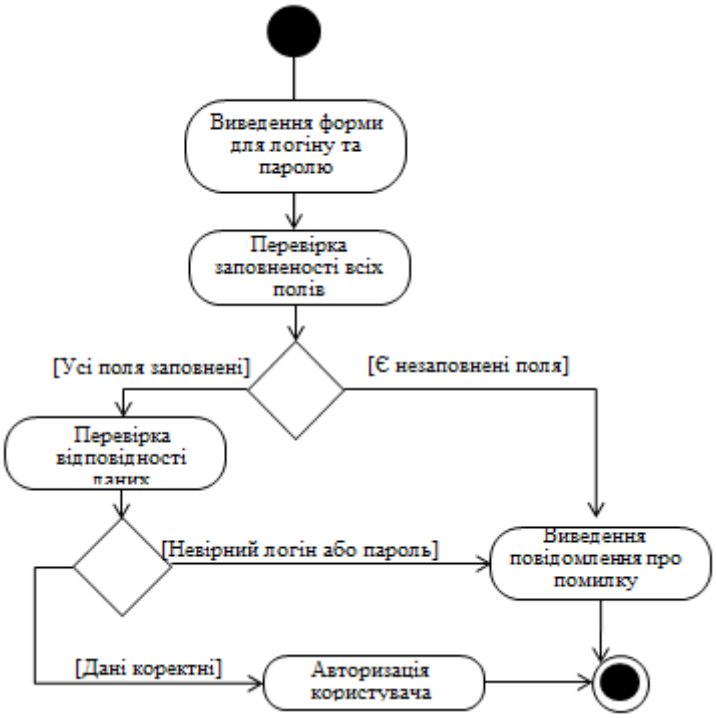


Рисунок В.19 – Діаграма діяльності інформаційної технології при введенні логіна та пароля

Рисунок В.20 – Діаграма діяльності інформаційної технології при перевірці клавіатурного почерку

Додаток Г (довідниковий)

Інструкція користувача

Початок роботи з інформаційною технологією здійснюється шляхом виведення вікна початкової активності – головного меню програми (рис. Г.1). Наявні два режими – авторизації та реєстрації.

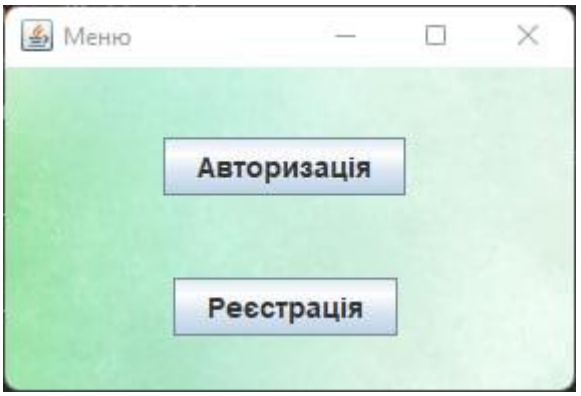


Рисунок Г.1 – Загальний вигляд інтерфейсного вікна початкової активності

Якщо користувач вибрав режим «Авторизація», виводиться вікно з полями для введення логіна та пароля, кнопками «Авторизуватися» та «Відновити пароль» (рис. Г.2).

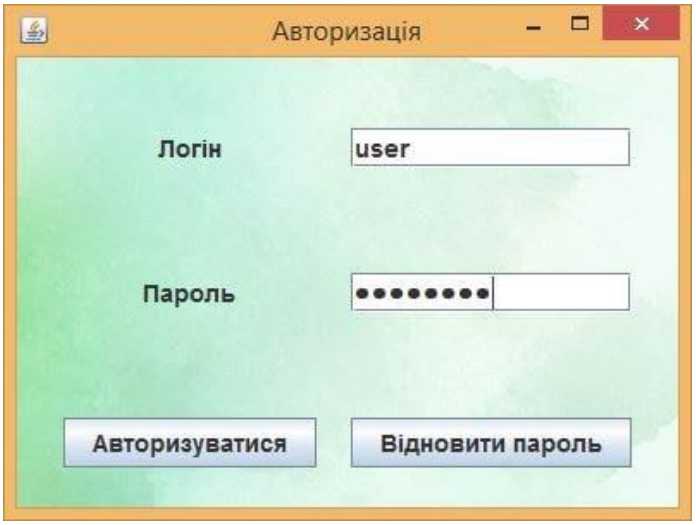


Рисунок Г.2 – Загальний вигляд інтерфейсного вікна авторизації

Ввівши логін і пароль правильно, користувач натискає кнопку авторизуватися, після чого відбувається перевірка клавіатурного почерку за допомогою спеціальної фрази (рис.Г.3) та вхід до системи, якщо перевірка пройдена успішно.

Рисунок Г.3 – Загальний вигляд інтерфейсного вікна для введення фрази

Після успішної перевірки клавіатурного почерку виводиться діалогове вікно, зображене на рисунку Г.4, при невдалій – вікно на рисунку Г.5.

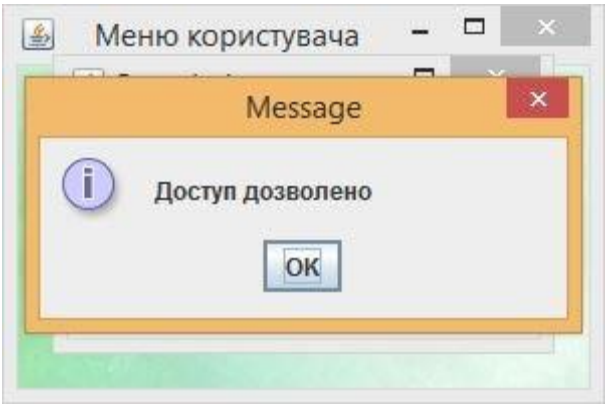


Рисунок Г.4 – Загальний вигляд діалогового вікна надання доступу

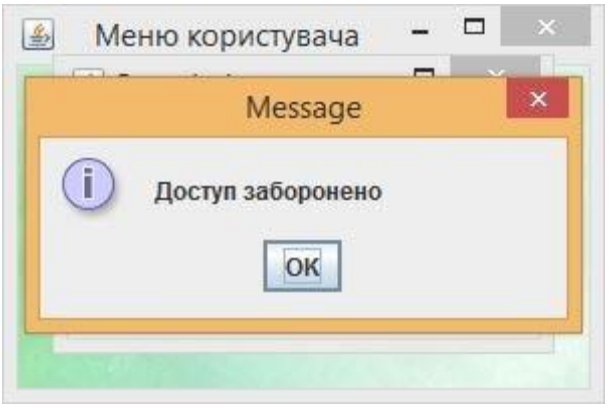


Рисунок Г.5 – Загальний вигляд діалогового вікна заборони доступу

Якщо користувач бажає відновити пароль, здійснюється перехід до вікна перевірки фрази (рис. Г.6). Якщо користувач ввів послідовність слів невірно, виводиться повідомлення про помилку (рис. Г.7).

Рисунок Г.6 – Загальний вигляд інтерфейсного вікна для відновлення пароля

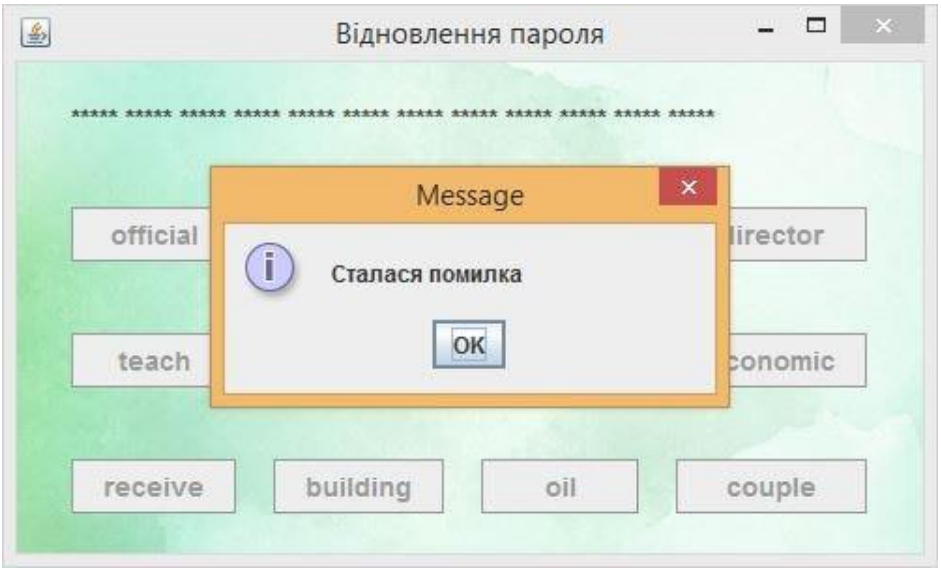


Рисунок Г.7 – Загальний вигляд інтерфейсного вікна про виведення помилки

Після проходження перевірки виводиться вікно зміни пароля (рис. Г.8). Вікно містить поля для введення нового пароля, для повторення пароля, кнопки «Назад» та «Далі».

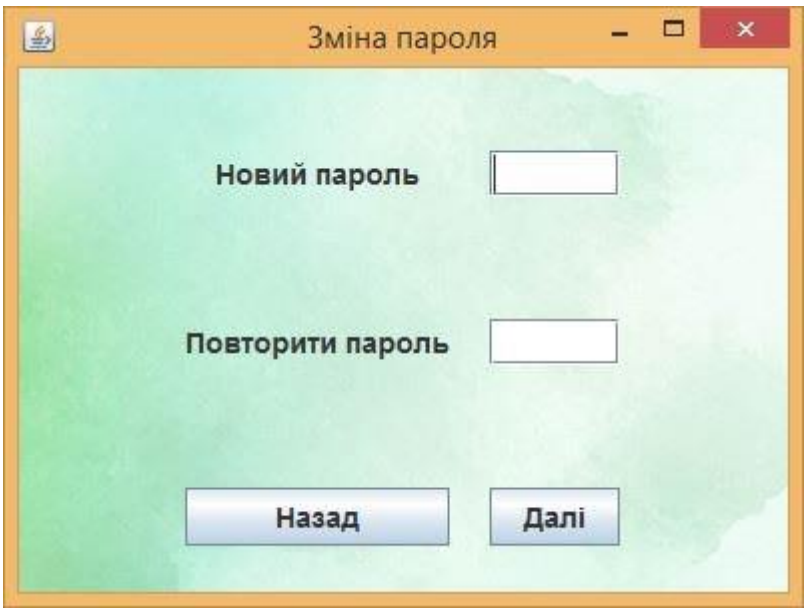


Рисунок Г.8 – Загальний вигляд інтерфейсного вікна зміни пароля

Для нового користувача існує можливість реєстрації в системі. Для цього у вікні головного меню (рис. Г.1) обирається пункт «Реєстрація», після чого здійснюється перехід до вікна реєстрації (рис. Г.9).

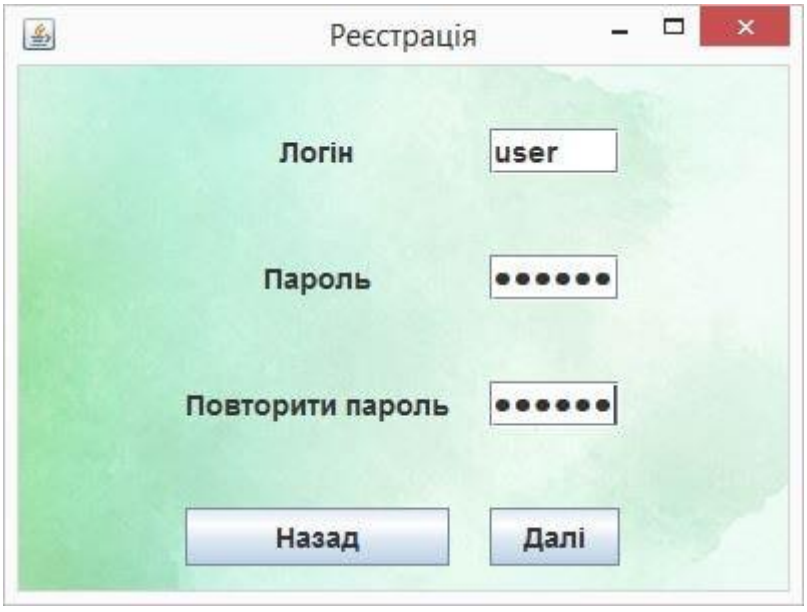


Рисунок Г.9 – Загальний вигляд інтерфейсного вікна реєстрації

Вікно реєстрації користувача містить поля для введення логіна, пароля та повторення пароля. Також тут присутні кнопки «Назад», для повернення до головного меню, та «Далі». При реєстрації користувача здійснюється перевірка

на заповненість всіх полів, відповідність повторно введеного паролю початковому, наявність введеного логіну в базі. Відповідно до наведених помилок виводиться діалогове вікно з конкретною помилкою (рис. Г.10 – Г.11).

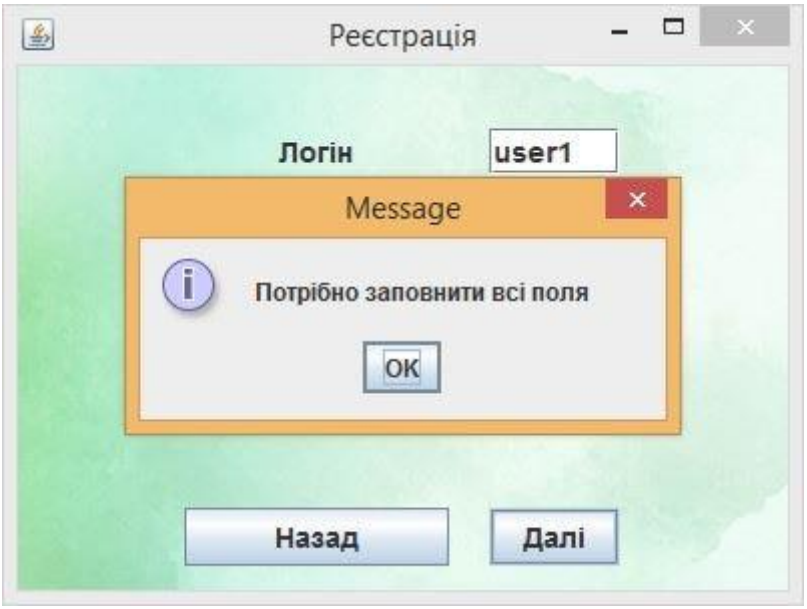


Рисунок Г.10 – Загальний вигляд інтерфейсного вікна виведення помилки заповнення полів

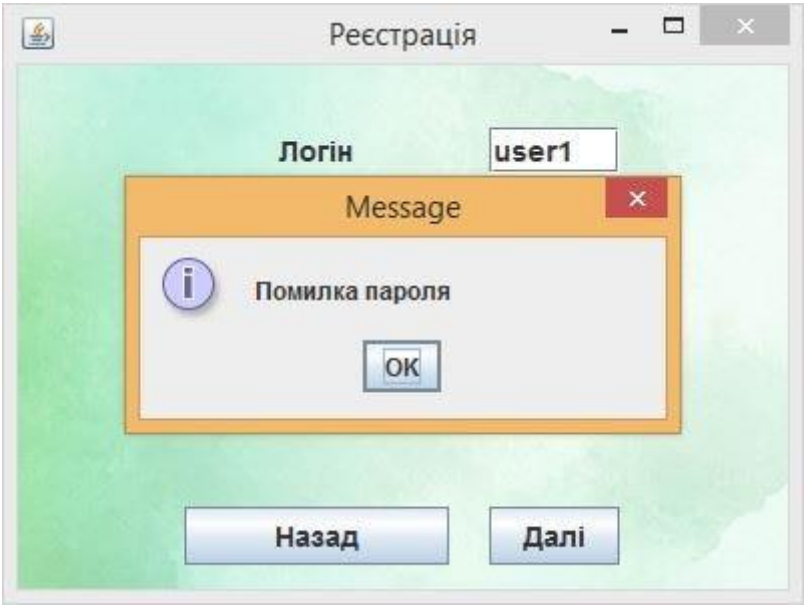


Рисунок Г.11 – Загальний вигляд інтерфейсного вікна виведення помилки пароля

Додаток Д.
Довідка про впровадження

ТОВ «Айбекс Айті»

Україна, 21050, м. Вінниця , вул. Театральна 39

1 грудня 2022 року

Довідка надана студентці Вінницького національного технічного університету групи 1КН-21м Король Яні Олександрівні про те, що результати магістерської кваліфікаційної роботи «Інформаційна технологія авторизації користувача з використанням клавіатурного почерку» використані в програмному забезпеченні для виконання робіт, що пов'язані з автентифікацією і авторизацію особистості, що розробляється компанією ТОВ «Айбекс Айті». За результатами дослідження програмний продукт планується до впровадження.

Бойко Р. В.

Директор
ТОВ «Айбекс Айті»

Товариство з обмеженою відповідальністю «АЙБЕКС АЙТІ» 42342127 Україна м.Вінниця