НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Приладобудівний факультет

Кафедра комп'ютерно-інтегрованих оптичних та навігаційних систем

«На правах рукопису»
УДК 615. 471.03

«До захисту допущено»
Завідувач кафедри
_____Надія БУРАУ
«___»__грудня_2021р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Комп'ютерно-інтегровані технології та системи навігації і керування»

зі спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології на тему: «Автоматизована система збереження персональних даних користувача»

Виконав:
студент ІІ курсу, групи ПГ-01мп
Зубарський Дмитро Олександрович _____
Науковий керівник:
Доцент кафедри КІОНС
Цибульник Сергій Олексійович _____
Консультант з розроблення стартап-проекту:
Професор, д.е.н., проф.
Бояринова Катерина Олександрівна _____
Рецензент:
в.о. зав.каф. ВП, д.т.н., проф.
Антонюк Віктор Степанович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2021 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Приладобудівний факультет

Кафедра комп’ютерно-інтегрованих оптичних та навігаційних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність (спеціалізація) – 151 «Автоматизація та комп’ютерно-інтегровані технології» («Комп'ютерно-інтегровані технології та системи навігації і керування»)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Надія БУРАУ

«___»_____2021 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Зубарському Дмитру Олександровичу

1. Тема дисертації «Автоматизована система збереження персональних даних користувача», науковий керівник дисертації Цибульник Сергій Олексійович, к.т.н., доцент, затверджені наказом по університету від «___»_____ 20__ р. №_____

2. Термін подання студентом дисертації: 06 грудня 2021 року

3. Об’єкт дослідження: процес захисту персональних даних.

4. Предмет дослідження: розроблення програмної логіки функціонування підсистем ідентифікації користувача, збереження і шифрування персональних даних.

5. Перелік завдань, які потрібно розробити:

5.1 Провести огляд стану проблеми.

5.2 Провести огляд сучасних методів та засобів збереження та захисту персональних даних.

5.3 Провести огляд публікацій за темою дисертації.

5.4 Спроектувати архітектуру менеджера паролів.

5.5 Розробити основні модулі та підсистеми менеджера паролів відповідно до архітектури.

5.6 Провести тестування працездатності.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: таблиці, графіки, рисунки

7. Орієнтовний перелік публікацій: 1 стаття

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розроблення стартап-проекту			

9. Дата видачі завдання 15 вересня 2021 року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Провести огляд стану проблеми	05.10.2021	
2.	Провести огляд сучасних методів та засобів збереження та захисту персональних даних	19.10.2021	
3.	Провести огляд публікацій за темою дисертації	26.10.2021	
4.	Спроектувати архітектуру менеджера паролів	09.11.2021	
5.	Розробити основні модулі та підсистеми менеджера паролів відповідно до архітектури	23.11.2021	
6.	Провести тестування працездатності	30.12.2021	
7.	Оформлення рукопису дисертації	07.12.2021	

Студент Дмитро ЗУБАРСЬКИЙ

Науковий керівник дисертації Сергій ЦИБУЛЬНИК

РЕФЕРАТ

Магістерська дисертація складає 104 сторінки, в ній містяться 36 ілюстрацій, 23 таблиці і 40 використаних джерел.

Актуальність: В еру цифрових технологій, їх використання стало звичкою, жодна наукова область чи будь-яка інша сфера не може існувати без використання новітніх технологій, гаджетів, тощо.

Цифрове забезпечення покращується кожного дня і можливості його використання стають все більш всеохоплюючими та необмеженими. Використовуючи різноманітні сервіси, програми, сайти, додатки, користувчі створюють персональні кабінети для ідентифікування особи, можливості використання потрібного забезпечення та збереження персональних даних. Для ідентифікації користувуча завжди створюються логін та пароль для входу, які зберігаються в пам’яті сервіса, який використовується для повторного використання на сайті, у програмі, тощо.

В свою чергу, користувач повинен запам’ятовувати чи записувати ці данні для використання свого аккаунту на конкретній платформі. Бувають випадки втрати паролів або навіть втрати доступу до особистого кабінету користувача.

Основними недоліками зберігання паролів є можливість їх втратити. Для зберження паролів та подальшого їх застосування використовуються різні способи збергання паролів: на комп'ютері, телефоні, у спеціальних програмах, на папері, у голові.

Одні з них більш безпечні чи практичні, інші мешн практичні, безпечніше всього використовувати спеціальні менеджери паролів, але всі вони мають свої недоліки.

Мета: Вдосконалення існуючих менеджерів паролів для забезпечення підвищеної безпеки зберігання деяких персональних даних користувачів (логінів та паролів), а також для запобігання заволодіння ними сторонніми особами.

Завдання:

1. Висвітлити проблему зберігання персональних даних.

2. Провести огляд сучасних методів та засобів збереження та захисту персональних даних.

3. Провести огляд публікацій за темою дисертації.

4. Розробити архітектуру менеджера паролів.

5. Реалізувати основні підсистеми та модулі менеджера паролів відповідно до розробленої архітектури.

6. Провести тестування працездатності розробленого менеджера паролів.

Об’єкт: Процес зберігання персональних даних (логінів та паролів).

Предмет: Підсистема ідентифікації користувача та підсистеми збереження і шифрування персональних даних.

Методи дослідження: Збір даних, аналіз, створення власної системи, тестування та аналіз власної автоматизованої системи.

Наукова новизна: Система з можливістю гнучкої генерації даних для збереження за допомогою спеціального алгоритму шифрування.

Практичне значення: Автоматизована система збереження персональних даних користувача генерує та зберігає дані користувачів для аутентифікації.

Публікації: Зубарський Д.О. Менеджери паролів / Погляд у майбутнє приладобудування : Збірник праць XIV Науково-практична конференція студентів, аспірантів та молодих вчених / 18-19 травня 2021 р. – К.:ПБФ, КПІ ім. Ігоря Сікорського. – 2021. - С. 38-41.

Ключові слова: менеджер паролів, шифрування, ключ, аутентифікація, база даних, алгоритм, логін, пароль.

ABSTRACT

The master's dissertation is 104 pages, it contains 36 illustrations 23 tables and 40 used sources.

Actuality: In the age of digital technologies, their use has become a habit, no scientific field or any other field can exist without the use of the latest technologies, gadgets, etc.

Digital security is improving every day and the possibilities of its use are becoming more comprehensive and unlimited. Using a variety of services, programs, sites, applications, users create personal accounts to identify the person, the ability to use the necessary software and store personal data. To identify the user, a login and login password are always created, which are stored in the memory of the service, which is used for reuse on the site, in the program, etc.

In turn, the user must remember or record this data to use their account on a particular platform. There are cases of loss of passwords or even loss of access to the user's personal account.

The main disadvantages of storing passwords are the possibility of losing them. To save passwords and their subsequent use, different ways of saving passwords are used: on a computer, phone, in special programs, on paper, in the head.

Some of them are more secure or practical, others are more practical, it is safest to use special password managers, but they all have their drawbacks.

Purpose: To improve existing password managers to ensure increased security of storage of some personal data of users (logins and passwords), as well as to prevent the acquisition of third parties.

Task:

1. Highlight the problem of personal data storage.

2. Review the current methods and means of storing and protecting personal data.

3. Review publications on the topic of the dissertation.

4. Develop a password manager architecture.

5. Implement the main subsystems and modules of the password manager in accordance with the developed architecture.

6. Carry out performance testing of the developed password manager.

Object: The process of storing personal data (logins and passwords).

Subject: Subsystem of user identification and subsystem of storage and encryption of personal data.

Research methods: Data collection, analysis, creation of own system, testing and analysis of own automated system.

Scientific novelty: A system with the ability to flexibly generate data for storage using a special encryption algorithm.

Practical significance: The automated system for storing personal data of the user generates and saves user data for authentication.

Publications: Zubarskyi D.O. Password managers / A look into the future of instrument making: Proceedings of the XIV Scientific and Practical Conference of Students, Postgraduates and Young Scientists / May 18-19, 2021 - K .: PBF, KPI. Igor Sikorsky. - 2021. - P. 38-41.

Keywords: password manager, encryption, key, authentication, database, algorithm, login, password.

Пояснювальна записка
до магістерської дисертації
на тему: «Автоматизована система збереження персональних даних користувача»

Київ – 2021 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

PIN - Personal Identification Number

OTP - One Time Password

SMS – Short Message Service

ЕЦП – Електронний Цифровий Підпис

IT – Information Technology

2FA –Two-Factor Authentication

QR – Quick Response

SD – Secure Digital

DVD – Digital Versatile Disc

API – Application Programming Interface

DES – Data Encryption Standard

RSA – Rivest, Shamir, Adleman

AES – Advanced Encryption Standard

PGP – Pretty Good Privacy

ПК – Персональний Комп’ютер

ЦП – Центральний Процесор

OS – Operating System

JVM – Java Virtual Machine

AOP – Aspect-Oriented Programming

XML – Extensible Markup Language

MVC – Model View Controller

MVP – Model View Presenter

MVVM – Model View View Model

SWOT – Strengths, Weaknesses, Opportunities, Threats

PHP - Personal Home Page

ВСТУП

В еру цифрових технологій, їх використання стало звичкою. Жодна наукова область чи будь-яка інша сфера не може існувати без використання новітніх технологій, гаджетів, тощо.

Цифрове забезпечення покращується кожного дня і можливості його використання стають більш всеохоплюючими та необмеженими. Використовуючи різноманітні сервіси, програми, сайти, додатки, користувачі створюють персональні кабінети для ідентифікування особи, можливості використання потрібного забезпечення та збереження персональних даних. Для ідентифікації користувача завжди створюються логін та пароль для входу, які зберігаються в пам’яті сервіса і в подальшому можуть бути використані повторно.

У свою чергу, користувач повинен запам’ятовувати чи записувати ці данні для використання свого акаунту на конкретній платформі. Відомо багато випадків втрати паролів або навіть втрати доступу до особистого кабінету користувача.

РОЗДІЛ 1
ОГЛЯД СТАНУ ПРОБЛЕМИ

1.1 Проблема збереження персональних даних.

Основними недоліками зберігання паролів є можливість їх втратити. Для зберження паролів та подальшого їх застосування дуже часто використовуються різні способи збергання паролів: на комп'ютері, телефоні, у спеціальних програмах, на папері, у пам’яті. Одні з них більш безпечні чи практичні, інші навпаки. Одним із сучасних засобів зберігання паролів є спеціальні менеджери паролів, але всі вони мають свої недоліки.

Як ви можете втратити всі паролі в диспетчері паролів одночасно?

Ви можете забути основний пароль, який захищає інші паролі. Після того, як ви замінили свої паролі на випадкові паролі та покладалися на те, що менеджер паролів вводить їх для вас, ви навряд чи зможете запам’ятати багато з них. Якщо ви втратите майстер-пароль, який менеджер паролів використовує для захисту інших паролів, ви можете втратити все. Є варіанти відновлення, але жоден не є ідеальним, про що я розповім пізніше.

Атака на ваш менеджер паролів може виявити всі ваші паролі. Це включає атаки на будь -який пристрій, на якому ви зберігаєте керовані паролі. Навіть якщо ви заблокували менеджер паролів, зловмисник зможе потрапити до нього, коли ви наступного разу розблокуєте його на цьому пристрої. Багато менеджерів паролів пропонують «заблокований» режим, в якому ви можете вимкнути доступ пристрою до ваших паролів, але більшість уразливі до атаки, яка може розблокувати ваші паролі в той момент, коли шкідлива програма потрапить на ваш пристрій.

Якщо ваш особистий ноутбук заражений шкідливим програмним забезпеченням, і ви використовуєте на ньому свій менеджер паролів, зловмисне програмне забезпечення може прочитати кожен ваш пароль. Якщо

ви використовуєте свій менеджер паролів на робочому комп'ютері, будь -хто, хто має адміністративний доступ до цього комп'ютера, може зламати ваші паролі в вашому менеджері паролів – навіть для сайтів, на які ви ніколи не входите з роботи. Якщо ви використовуєте свій менеджер паролів на планшеті і залишаєте його незаблокованим, будь-хто з планшетом може отримати доступ до всіх ваших паролів. Якщо ваш телефон вкрали, якщо злодій може розблокувати ваш телефон, і якщо у вас немає налаштованого менеджера паролів, який вимагає розблокування при кожному використанні, злодій тепер має доступ до всіх ваших паролів.

На відміну від цього, якщо ви не використовуєте менеджер паролів і ваш пристрій заражено шкідливим програмним забезпеченням, зловмисник може вкрасти введені вами паролі, але не ті, які ви не використовуєте. Ви можете вирішити, що деякі паролі можна вводити на пристроях із меншим рівнем безпеки, але їх слід вводити лише на пристроях з більш високим рівнем безпеки.

Менеджер паролів – це ще одна програма, встановлена на ваших пристроях, яка може бути скомпрометована. Усе програмне забезпечення містить помилки, і, незважаючи на те, що вони розроблені для задоволення потреб безпеки, менеджери паролів далеко не застраховані від помилок безпеки [1].

1.2 Види персональних даних

Деякі дані та інформація, що зберігаються на комп'ютері, є особистими і їх потрібно зберігати конфіденційними. Люди хочуть, щоб їхні банківські реквізити та медична документація були конфіденційними і були подалі від будь-кого. Якщо той, хто не має права бачити ці дані, може отримати доступ без дозволу, це несанкціонований доступ. Закон про захист даних встановлює правила, що запобігають цьому [2].

Людина має фізичну, фізіологічну, соціальну та культурну ідентичність [3]. Персональні дані діляться на дві категорії: загальні та особливі (чутливі).

До загальної категорії можна віднести: прізвище та ім’я; дату та місце народження; громадянство; сімейний стан; псевдонім; дані, записані в посвідченні водія; економічне і фінансове становище; дані про майно; банківські дані; підпис; дані з актів цивільного стану; номер пенсійної справи; адресу місця проживання; дипломи про освіту, професійну підготовку тощо.

Особлива категорія охоплює інформацію про: расове, етнічне та національне походження; політичні, релігійні та світоглядні переконання; членство в політичних партіях та/або організаціях, професійних спілках, релігійних організаціях чи громадських організаціях світоглядної спрямованості; стан здоров’я (медичні дані); статеве життя; біометричні дані; генетичні дані; притягнення до адміністративної чи кримінальної відповідальності; застосування до особи заходів у рамках досудового розслідування.

1.3 Пароль, як засіб ідентифікації користувача

Пароль [4] - це рядок символів, що використовується для перевірки ідентичності користувача під час процесу автентифікації. Паролі зазвичай використовуються в парі з іменем користувача; вони розроблені таким чином, щоб бути відомими лише користувачеві і дозволяють цьому користувачеві отримати доступ до пристрою, програми або веб-сайту. Паролі можуть мати різну довжину і містити букви, цифри та спеціальні символи.

Пароль іноді називають парольною фразою, коли в паролі використовується більше одного слова, або кодом, коли в паролі використовуються лише цифри, такі як персональний ідентифікаційний номер (PIN).

Пароль - це проста програма автентифікації виклик-відповідь, яка використовує усний, письмовий або набраний код для задоволення запиту на виклик. Порядок і різноманітність символів часто визначають складність або надійність даного пароля. Ось чому системи безпеки часто вимагають від користувачів створювати паролі, які використовують принаймні одну велику літеру, цифру та символ. Щоб пароль був ефективним механізмом безпеки, його дані повинні зберігатися в таємниці. В іншому випадку неавторизовані користувачі можуть отримати доступ до файлів та цінних паперів, які намагаються захистити.

1.3.1 Статичний пароль

Статичний пароль - пароль, який постійно застосовується для ідентифікації та отримання доступу до персональних даних. Зберігати такий пароль треба ретельніше. Підбір багаторазового паролю потребує особливої уваги, тому що від його надійності залежить безпека конфіденційної інформації. Багаторазові паролі програють одноразовим через те, що мають менший захист та можуть бути застосовані декілька разів.

Очевидні слабкі місця систем статичного пароля з незашифрованою передачею [3]. Простого списку деяких найбільш очевидних прикладів достатньо, щоб показати легкість, з якою зловмисник може отримати пароль. Ці методи включають:

- підслуховування: "прослуховуючи" мережу, зловмисник отримує пароль користувача, а потім може встановити особу жертви, заповнивши форму автентифікації;

- атака за словником: це використовується для отримання "слабких" паролів. Це паролі, які користувачі можуть легко запам'ятати, і з цієї причини вони широко використовуються. Ця форма нападу враховує незначні зміни в

режимах словника, такі як додавання цифри або великої літери посередині слова;

- атака соціальної інженерії: зловмисник намагається вгадати пароль на основі особистої інформації користувача (дата народження, імена дітей чи домашніх тварин, улюблені види спорту тощо);

- фішинг: цей метод полягає у надсиланні шахрайських електронних листів, що імітують визнану установу чи компанію, наприклад банк, із запрошенням користувачеві надати дані для входу через веб-сайт зловмисника (який також імітує веб-сайт компанії), нібито для повторної активації або розблокування особистого облікового запису.

Певною кількістю веб-сайтів були прийняті незначні контрзаходи, намагаючись обмежити вплив деяких із цих атак. Наразі словникові атаки зараз важче здійснювати через обмеження кількості спроб автентифікації, які можуть бути зроблені за певний період; однак це також незручно для жертви спроби нападу, чий рахунок на деякий час буде заблокований.

Крім того, багато веб-сайтів використовують більш-менш точне вимірювання ентропії пароля користувача, тобто простоту, з якою пароль можна отримати за допомогою грубої сили. Паролі, які вважаються слабкими, зазвичай не приймаються. Однак користувачам нелегко запам’ятовувати складні паролі. З цієї причини користувачі часто зберігають паролі у своєму браузері, що є незахищеним середовищем. Це також може викликати інші проблеми: наприклад, коли користувач змінює обладнання, він може втратити доступ до одного або кількох облікових записів, для яких він не зміг запам’ятати пароль.

Загалом, єдиною перевагою статичної автентифікації паролем є передбачувана простота реалізації. Це правда лише частково; хоча це технічно просто реалізувати, певний ступінь складності передається користувачеві, який відповідає за безпеку своїх паролів. Це створює реальну об’єктивну складність, особливо при управлінні кількома паролями.

1.3.2 Динамічний пароль

Динамічний пароль [5] - пароль або кодове слово, яке може використовуватися тільки один раз. Такий пароль є дійсним тільки для одного сеансу аутентифікації - перевірки приналежності користувача до осіб з дозволом на перегляд конфіденційної інформації за допомогою кодового слова. Функціонування одноразового пароля може обмежуватися певним заздалегідь проміжком часу (як правило, 24 години). Основна перевага такого виду пароля полягає в тому, що його неможливо використовувати повторно, а тому зловмисник, навіть якщо дізнався пароль, не може отримати доступ до інформації під захистом. Недоліком застосування одноразового пароля є потреба в частій зміні та запам'ятовуванні пароля.

Крок у правильному напрямку - це перехід від статичного пароля до динамічного.

Постійна зміна пароля [6] - це дуже велика проблема, але, на щастя, це не є динамічним паролем, і насправді, можливо, ви вже використовуєте динамічні паролі сьогодні. Уявіть, що ви отримуєте доступ до своєї банківської програми, і вона надсилає вам код на телефон, і вам потрібно ввести цей код із телефону, щоб отримати доступ до банківської програми. Це вже динамічний пароль. Вони називаються паролями одноразового доступу або одноразовими паролями, і це загальновживаний тип динамічного пароля - машинно генерований випадковий рядок, який використовується лише один раз для автентифікації.

Принцип роботи динамічних паролів ґрунтується на методі автентифікації, він надсилає вам код, який працює лише один раз, закінчується протягом короткого періоду часу і ускладнює доступ хакерів до вашого облікового запису. Поширеними прикладами автентифікаторів, які використовують OTP, є Google Authenticator та Microsoft Authenticator, які

дають вам 6-значний код, та у вас є одна хвилина для доступу до входу, наприклад SalesForce.

Динамічні паролі зручні тим, що їх не потрібно запам’ятовувати, а оскільки пароль ніколи не збігається, вони служать основною перешкодою для хакерів, які, можливо, прагнуть зламати облікові записи користувачів. Любителям статичних паролів пора зіткнутися з фактами - статичний пароль зникне.

1.3.3 BankID

BankID [7] – спосіб верифікації громадян через українські банки для надання адміністративних та інших послуг через Інтернет.

BankID вирішує проблему верифікації користувача через Інтернет. Щоб надати громадянину довідку, потрібно переконатися в тому, що це дійсно конкретний громадянин. Якщо громадянин вибере верифікацію через BankID, він вводить логін/пароль свого інтернет-банку, проходить другий етап авторизації (наприклад, введення SMS-пароля) і таким чином підтверджує свою особу.

1.3.4 Електро́нний цифрови́й пі́дпис

Електронний цифровий підпис [8] (або скорочено – ЕЦП) за правовим статусом прирівняний до власноручного підпису або печатки.

ЕЦП – це дані в електронній формі, отримані за результатами криптографічного перетворення, які додаються до інших даних або документів і забезпечують їх цілісність та ідентифікацію автора.

За допомогою послуг ЕЦП можна підписувати електронні документи, користуватися електронними послугами, реєструватися на державних

порталах тощо. Документи, підписані за допомогою ЕЦП, мають таку саму юридичну силу, як і звичайні.

Електронний підпис [9] - це законний спосіб отримати згоду або схвалення електронних документів або бланків. Одним з найбільш використовуваних визначень електронного підпису є "електронний символ або процес, приєднаний до запису або логічно пов'язаний з ним ... прийнятий особою з наміром підписати запис". Електронні підписи можна використовувати для заміни рукописних підписів практично в кожному особистому або бізнес-процесі. Приклади включають договори, форми заяв, нові форми найму на роботу, угоди про нерозголошення інформації, документи про залучення постачальників та пропозиції, дозвіл на зміну та форми реєстрації державних пільг.

1.3.5 Біометричний пароль

Біометрія [10] – це ідентифікація людини за унікальними, властивими тільки їй біологічними ознаками. Сьогодні експлуатується вже більше десятка різних біометричних ознак. Причому для найпоширеніших з них (відбитки пальців і райдужна оболонка ока) існує безліч різних за принципом дії сканерів. Так що користувачам, що вирішили використати біометричну ідентифікацію, є із чого вибрати.

Головною перевагою біометричних технологій є найвища надійність. І дійсно, усі знають, що двох людей з однаковими відбитками пальців у природі просто не існує. Правда, сьогодні вже відомо кілька способів обману дактилоскопічних сканерів. Наприклад, потрібні відбитки пальців можуть бути перенесені на плівку або до пристрою може бути прикладена велика фотографія пальця зареєстрованого користувача. Втім, треба зізнатися, що сучасні пристрої вже не попадаються на такі прості виверти. Так що

зловмисникам доводиться видумувати все нові й нові способи обману біометричних сканерів.

Департаменти міграційної служби завжди були переповнені величезними чергами, і в 2015 році вони збільшилися майже вдвічі[11]. Причиною цього стало впровадження в систему нового виду документів - біометричного паспорта.

Навколо цього документу одразу розгорілися гучні дискусії, тому що тільки почали говорити про безвізовий режим, але ні про які контракти на міжнародній арені не йшлося. Таким чином, Інтернет був наповнений запитами, такими як "який біометричний паспорт в Україні" та іншими, які впливають на предмет необхідних документів.

За роки видачі цього документа у світі багато чого змінилося, експерименти та протести проти цієї системи відбувалися в різних країнах, змінилися процедури видачі документів.

Біометричні паспорти видаються у 101 країні світу;

54 країни містять дані за такими індексами, як обличчя та відбитки пальців;

Майже половина (47 країн) використовують виключно обличчя як біометричний параметр ідентифікації особи, серед яких 10 країн мають у плані використання відбитків пальців як другого індексу;

Біометричний паспорт, що зберігає інформацію про сітківку очей, не використовується ні в одній країні;

У 23 країнах використовуються повністю автоматизовані системи прикордонного контролю за паспортами з чіпами (Автоматизований прикордонний контроль).

Сьогодні біометричний закордонний паспорт є додатковим засобом підвищення безпеки, а також спрощує життя мандрівників та людей, які відвідують інші країни з метою роботи.

1.3.6 Двохфакторна аутентифікація

Сьогодні в галузі кібербезпеки виокремлюють ряд рішень для малого та середнього бізнесу, які допомагають забезпечити додаткові рівні безпеки корпоративної мережі [12]. Проте важливо захищати не тільки IT-системи, а й конфіденційні дані, які є головним ресурсом будь-якого підприємства. Одним із першочергових кроків для безпеки важливої інформації має бути посилення захисту облікових записів на додаток до стандартного пароля.

Зокрема набуває популярності двофакторна аутентифікація, яка є ідеальним рішенням для захисту різних онлайн-сервісів від несанкціонованого доступу. Правильне впровадження та використання технології — найкраще доповнення до створення надійних паролів. Оскільки більшість користувачів не звикла створювати та запам'ятовувати велику кількість складних комбінацій, використання двофакторної аутентифікації дозволить підвищити рівень безпеки облікових записів в компанії.

Недостатній захист даних, зокрема повідомлень електронної пошти, клієнтських баз даних, бібліотек документів, може серйозно вплинути на репутацію організації. Саме тому необхідно використовувати різні варіанти двофакторної аутентифікації для додаткового захисту облікових записів співробітників. Зазвичай, використовується автоматичне надсилання SMS-повідомлень або окрема програма, яка генерує коди доступу. Після стандартного пароля користувачеві потрібно також ввести ще код, а в деяких системах використовується програма для введення коду.

Помилка Heartbleed 2014 року [13] відкрила мільйони входів в Інтернет шахраям завдяки одному дрібничковому коду, і наші кошмари безпеки з тих пір тільки погіршилися.

Що робити середньому користувачеві Інтернету? Вам обов’язково потрібно регулярно змінювати паролі. Вони є досить смішним методом автентифікації, і їх можна легко отримати різними методами.

Вам дійсно потрібен другий спосіб перевірити себе. Ось чому багато Інтернет-сервісів, деякі з яких відчули, що їх зламали, пропонують двофакторну автентифікацію. Іноді вона називається 2FA або використовується як взаємозамінний з термінами "двоетапний" та "перевірка" залежно від маркетингу.

Як каже провідний аналітик з питань безпеки компанії PCMag Ніл Дж. Рубенкінг, «є три загальновизнані фактори для автентифікації: те, що ви знаєте (наприклад, пароль), те, що у вас є (наприклад, мобільний телефон), і те, що ви є (наприклад, ваш відбиток пальця). Двофакторний означає, що система використовує два з цих варіантів ".

Біометричні сканери для відбитків пальців, сітківки очей або обличчя знаходяться на підйомі завдяки таким інноваціям, як Apple Face ID і Windows Hello. Але в більшості випадків додаткова автентифікація - це просто числовий код; кілька цифр, надісланих на ваш телефон, які можна використовувати лише один раз.

Ви можете отримати цей код за допомогою текстового повідомлення або спеціалізованої програми для смартфонів під назвою "автентифікатор". Після з’єднання з вашими обліковими записами додаток відображає набір кодів, що постійно обертається, щоб використовувати їх у разі потреби - навіть не вимагає підключення до Інтернету. Є декілька, включаючи деякі від великих компаній, таких як Microsoft та Google, з додатками для обох основних мобільних платформ. Ці два досить базові. Інші, такі як Twilio Authy, Duo Mobile та LastPass Authenticator, роблять те саме, по суті, деякі з них керують паролями та іншими функціями. І навпаки, більшість популярних менеджерів паролів (наприклад, LastPass) за замовчуванням пропонують аутентифікацію 2FA.

Коди, надані програмами автентифікації, синхронізуються у ваших облікових записах, тож ви можете відсканувати QR-код на телефоні та

отримати свій шестизначний код доступу у своєму веб-переглядачі, якщо він підтримується.

Треба врахувати, що налаштування 2FA може фактично зламати доступ до деяких старих служб. У таких випадках вам доведеться покладатися на паролі додатків - пароль, який ви генеруєте на головному веб-сайті для використання з певною програмою (наприклад, Xbox Live). Паролі додатків відображатимуться як опція для Facebook, Twitter, Microsoft, Yahoo, Evernote та інших - усі вони використовуються як сторонні входи або мають функції, до яких ви можете отримати доступ із інших служб.

Пам’ятайте про це, наскільки важко це все звучить: бути безпечним непросто. Шахраї розраховують на те, що ви невпевнено захищаєте себе. Впровадження 2FA означатиме, що для кожного входу на новий пристрій потрібно трохи більше часу, але це варто того, щоб уникнути серйозних крадіжок, будь то ваша особа, дані чи гроші.

1.4 Методи зберігання персональних даних

Зберігання персональних даних [14]. Зберігання персональних даних може бути у різних формах (і “паперовій” і цифровій). У цифровій формі підприємство чи підприємець може зберігати відомості локально, використовуючи свої власні потужності і технічні засоби, або віддалено у хмарних сховищах. При цьому, якщо це окремо не врегульовано в угоді з власником відповідних потужностей, то ризик втрати або витоку інформації несе підприємство, яке розміщає персональні дані з метою зберігання. З огляду на європейське законодавство велике значення має юрисдикція (країна) фактичного місцезнаходження потужностей (серверів), на яких зберігаються персональні дані клієнтів і користувачів. Що ж до технічних аспектів захисту персональних даних, то з юридичної точки зору з цим все просто. Програмні і технічні заходи безпеки мають бути актуальними, дієвими та відповідати

потенційним загрозам, характеру і масштабам наявної бази персональних даних.

1.4.1 Сховище на персональному комп’ютері

Незважаючи на безліч зовнішніх рішень для цифрових файлів, деякі люди все ще зберігають свої фотографії, відео та файли вмісту на персональному комп’ютері. Єдина проблема [14] цього методу полягає в тому, що ваш комп’ютер може швидко забитися тисячами файлів. Це уповільнює ваше цінне апаратне забезпечення (комп’ютер).

Якщо ви хочете знайти цифровий файл, ви, ймовірно, очікуєте, що цей файл з’явиться на вашому екрані в одну мить. Тим не менш - кожен, хто зберігає багато фотографій на комп’ютері, знає, що на їх пошук можуть знадобитися хвилини, іноді години - навіть якщо ви зберігаєте їх на робочому столі. Просто не так зручно зберігати речі таким чином. Найголовніше, що просто зберігання цих цифрових файлів на робочому столі робить їх уразливими до вірусів, пошкоджень або крадіжок. Люди, які покладаються на це, також зазвичай не мають резервного плану.

1.4.2 Холодне зберігання

Відсутність резервного копіювання змусила багатьох досліджувати холодне зберігання. Це включає в себе окремі диски, такі як флеш - накопичувачі та SD-карти, і DVD. Одна з переваг - це економічний спосіб зберігання великих файлів відносно безпечним способом. Файли можна упорядковувати, щоб інформацію можна було легко знаходити та ділитися - після її підключення до настільного комп’ютера або ноутбука. Однак є і зворотна сторона. Дані фрагментовані та недоступні - чи матимете ви доступ до машини з дисководом, коли прийде час запускати DVD? Ви можете не

знати, на якому накопичувачі файли з днем народження, або пам’ятати, на якому диску є фотографії бабусі. Крім того, його можна легко втратити або пошкодити. Також, зовнішні накопичувачі часто потребують технічного обслуговування через кілька років, як і будь -яке інше обладнання, що вимагає додаткових витрат на відновлення файлів, якщо вони пошкоджені та виникає потреба перенести всю цю інформацію на новий пристрій.

1.4.3 Сховище соціальних медіа

Оскільки популярність профілів у соціальних мережах зростала, люди незабаром зрозуміли, що вони служать способом зберігання фотографій та відеозаписів, до яких вони завжди можуть отримати доступ. Таким чином, ці цифрові файли не займають місце на мобільних пристроях або комп’ютерах. Крім того, ці сайти в соціальних мережах дали можливість створювати альбоми для легкого доступу та обміну ними.

Проте проблеми є. По-перше, зображення зберігаються у низькій роздільній здатності. Це не дає вам найкращої версії вашого цифрового життя. Інші рішення для зберігання зберігають зображення та відео у високій роздільній здатності. Крім того, зображення в соціальних мережах можна ненавмисно видалити і назавжди втратити. У багатьох випадках компанії соціальних медіа не дають жодних гарантій щодо того, що зберігається на їхній платформі чи ні.

Ще страшніше те, що сайти соціальних медіа визнають, що активно видобувають ваші особисті дані. Наприклад, Facebook заявив, що їм "належать" ваші фотографії. Навіть якби ви видалили свій обліковий запис, вони сказали, що дані все одно будуть десь на їх серверах. Тепер є способи змінити це до певної міри. Однак, поки ви зберігаєте фотографії та відео у соціальних мережах, ви повинні вважати, що ваші особисті дані ніколи не можуть бути справді вашими.

1.4.4 Хмарне сховище

Цей варіант цифрового зберігання стає все більш популярним завдяки своїм численним перевагам, незважаючи на деякі недоліки. Такі компанії, як Dropbox, з'явилися, щоб запропонувати варіант для цифрових файлів, які вирішують існуючі проблеми зі зберіганням. Незабаром з’явились Google Drive, Box та численні інші хмарні платформи.

Хмарне сховище усунуло частину необхідності зберігати файли на комп’ютерах, оскільки пропонує організований спосіб пошуку, обміну та контролю вашої інформації. Дані не активно видобуваються, як сховище соціальних мереж, а також не втрачаються і не пошкоджуються. Крім того, для більшої зручності є негайна синхронізація та доступ з будь -якого пристрою. Деякі навіть містять історію переглядів, щоб ви могли повернутися до старішої версії цифрового файлу.

Однак навіть це рішення для зберігання має свої недоліки. По-перше, багато сервісів хмарного зберігання дозволяють зберігати не дуже великі обсяги даних. Потім, коли ви переглядаєте варіанти підписки, ви розумієте, що це може стати досить дорогим. Тоді що станеться, якщо ви скасуєте підписку - куди йдуть ваші дані?

По-друге, багато експертів із безпеки висловили стурбованість щодо справжньої безпеки за допомогою варіантів зберігання в хмарі. Наприклад, InformationWeek зазначив, що існують деякі ключові загрози, включаючи порушення даних, втрату даних, викрадення трафіку, небезпечні API та інше. ОГС пояснила, що з кожним роком виникають нові проблеми безпеки, оскільки хакери розуміють, як вони можуть отримати ці дані, перш за все, під час передачі даних хмарному провайдеру.

1.4.5 Локальне та хмарне сховища

Локальне зберігання [15] змушує хакерів вдаватися до складних підходів на основі шкідливого програмного забезпечення, таких як: використання кейлогерів, троянських програм та інших передових інструментів. Оскільки пароль зберігається на пристрої користувача, користувач повністю контролює його безпеку.

Ліцензії менеджера паролів найчастіше можна використовувати лише на одному пристрої, тобто для кожного пристрою, необхідного для синхронізації паролів, потрібно придбати декілька ліцензій. Якщо пристрій втрачено та / або викрадено, усі паролі також втрачаються.

Хмарне зберігання [15] покращує доступність та зручність для користувачів. Оскільки зашифровані паролі зберігаються на хмарних серверах, користувачі можуть отримати до них доступ із будь-якої кількості пристроїв та відносно легко синхронізувати паролі між пристроями без необхідних додаткових кроків. Ці служби зберігають зашифровані копії сховища на власних серверах, а також виступають гарантами, що всі пристрої того користувача завжди синхронізовані та шифрують передачі між пристроями та серверами. Хмарне сховище також дозволяє відновити паролі, якщо користувач втрачає пристрій.

Недоліком хмарного сховища [15] є те, що користувач не може забезпечити безпеку даних. Ризик, хоч і невеликий, полягає в тому, що одна із хмарних служб може бути зламана, а паролі викрадені. Якщо менеджер паролів робить свою роботу правильно, він зберігає всі паролі у зашифрованому форматі, а головний пароль зберігає лише як "хеш", що є результатом незворотного математичного процесу.

Користувачі стикаються із загрозами безпеки, незалежно від того, яке сховище паролів використовується: хмарне чи локальне. Також необхідно відмітити відмітити, що немає єдиного варіанту, який підходить для всіх [16].

1.4.6 Менеджери паролів

Менеджери паролів допомагають генерувати унікальні та надійні паролі, зберігати їх в одному безпечному (зашифрованому) місці та використовувати їх, лише запам'ятовуючи один головний пароль. Головний пароль розблоковує зашифроване сховище, що надає вам доступ до кожного з ваших паролів.

Найважливіше рішення для використання – це вирішити де зберігати паролі: локально на власних комп’ютерах та мобільних пристроях. Або в хмарі на чужих серверах.

1.5 Шифрування даних

Шифрування - це цікава технологія, яка працює шляхом зховування даних, тому її неможливо прочитати будь-якій людині. Розглянемо, як це працює з програмним забезпеченням для електронної пошти.

Якщо необхідно надіслати приватне повідомлення, відбувається його шифрування за допомогою будь-якої зі спеціалізованих програм. Ось зашифроване повідомлення [16]:

wUwDPglyJu9LOnkBAf4vxSpQgQZltcz7LWwEquhdm5kSQIkQlZtfxtSTsmaw
q6gVH8SimlC3W6TDOhhL2FdgvdIC7sDv7G1Z7pCNzFLp0lgB9ACm8r5RZOi
N5ske9cBVjlVfgmQ9VpFzSwzLLODhCU7 /
2THg2iDrW3NGQZfz3SSWviwCe7GmNIvp5jEkGPCGcla4Fgdp /
xuyewPk6NDlBewftLtHJVf= PAb3

На перший погляд зашифроване повідомлення буквально стає невпорядкованим хаосом випадкових символів. Але маючи секретний пароль цей текст можна розшифрувати та знайти оригінальне повідомлення [16].

“Заходьте за хот-догами та содовою”!

Незалежно від того, чи це транзит, як електронна пошта, чи зберігання на жорсткому диску, шифрування працює, щоб уникнути стороннього втручання в дані - навіть якщо зловмисники якось отримають доступ до мережі або системи.

1.5.1 Ключі шифрування

Ключ [17] — параметр криптографічної системи, який використовується для:

- шифрування і/або дешифрування повідомлення при шифруванні;
- накладення та перевірки коду автентифікації повідомлень або електронного цифрового підпису.

1.5.2 Симетричні та асиметричні алгоритми

У асиметричних криптосистемах ключі зазвичай створюються парами [17]: ключ шифрування e, та ключ дешифрування d. Якщо знаючи один можна легко отримати інший, та навпаки (наприклад вони збігаються), то криптосистема називається шифруванням з симетричними ключами. Якщо ж з одного можна отримати інший, але навпаки дуже важко, то така система називається шифруванням з несиметричними ключами.

Приклади ключів:

Відкритий ключ — ключ, котрий дозволяється передавати по відкритому каналу зв'язку, а таємний ключ — мусить зберігатися таємно, або передаватися з використанням закритого каналу зв'язку.

Сеансовий ключ — ключ, що використовується під час сеансу обміну повідомленнями для захисту каналу зв'язку.

1.5.3 Огляд алгоритмів шифрування

Технологія шифрування поставляється у багатьох формах, при цьому розмір та надійність ключів, як правило, є найбільшими між різними видами шифрування алгоритмів [15]. Розглянемо деякі з них:

1. Алгоритм Triple DES [15].

Triple DES був розроблений, щоб замінити оригінальний алгоритм Data Encryption Standard (DES), який з часом хакери навчилися перемагати з відносною легкістю. Колись Triple DES був рекомендованим стандартом та найбільш широко використовуваним симетричним алгоритмом у галузі.

Triple DES використовує три окремі клавіші по 56 біт кожна. Загальна довжина ключа складає 168 біт, але експерти стверджують, що 112-бітна.

Незважаючи на те, що компанія Triple DES повільно припиняє свою діяльність, все ще вдається зробити надійне рішення апаратного шифрування для фінансових послуг та інших галузей.

2. Алгоритм RSA [15].

RSA - це алгоритм шифрування з відкритим ключем і стандарт для шифрування даних, що надсилаються через Інтернет. Це також є одним із методів, що використовуються в програмах PGP та GPG.

На відміну від Triple DES, RSA вважається асиметричним алгоритмом завдяки використанню пари ключів. Отримано ключ, який використовується для шифрування повідомлення, та приватний ключ для його розшифровки. Результатом шифрування RSA є список незрозумілих символів, що займає у зловмисників досить багато часу для розшифровки.

3. Алгоритм Blowfish [15].

Blowfish - ще один алгоритм, призначений замінити DES. Цей симетричний шифр розбиває повідомлення на блоки по 64 біти і шифрує їх окремо.

Blowfish відомий як своєю надзвичайною швидкістю, так і загальною ефективністю, оскільки багато хто стверджує, що він ніколи не був переможений [15]. Цей алгоритм є безкоштовним та знаходиться у відкритому доступі.

Blowfish можна знайти в категоріях програмного забезпечення, починаючи від платформ електронної комерції для забезпечення платежів і закінчуючи інструментами управління паролями, де він використовувався для захисту паролів. Це, безумовно, один із найбільш гнучких та доступних методів шифрування.

4. Алгоритм Twofish [15].

Експерт з питань комп'ютерної безпеки Брюс Шнайєр - головний розробник Blowfish та його наступника Twofish. Ключі, що використовуються в цьому алгоритмі, можуть мати довжину до 256 біт, і як симетричний прийом потрібен лише один ключ.

Twofish вважається [15] одним з найшвидших алгоритмів та ідеально підходить для використання як в апаратному, так і в програмному середовищі. Як і Blowfish, Twofish знаходиться у вільному доступі.

5. Алгоритм AES [15].

Стандарт розширеного шифрування (AES) - це алгоритм, якому довіряють, як стандарту, уряд США та численні організації.

Незважаючи на те, що він надзвичайно ефективний у 128-бітовій формі, AES також використовує ключі 192 та 256 біт для цільового шифрування.

AES, в основному, вважається непроникним для всіх атак, за винятком грубої сили, яка намагається розшифрувати повідомлення, використовуючи всі можливі комбінації в 128, 192 або 256-бітному шифрі. Проте експерти з питань безпеки вважають [15], що AES врешті-решт буде визнано фактичним стандартом шифрування даних у приватному секторі.

1.5.4 Програми шифрування даних

1. Система TrueCrypt [18].

Цей софт, мабуть, один з найвідоміших шифрувальників. TrueCrypt дозволяє створювати зашифровані контейнери на фізичних носіях, захищати флешки, розділи і цілі жорсткі диски від несанкціонованого доступу.

2. Система PGP Desktop [18].

Це програма-комбайн для максимального захисту інформації на комп'ютері. PGP Desktop вміє шифрувати файли і каталоги, в тому числі і в локальній мережі, захищати поштові вкладення і повідомлення, створювати закріптованние віртуальні диски, безповоротно видаляти дані шляхом багатопрохідної перезапису.

3. Система Folder Lock [18].

Folder Lock - найбільш доброзичливий до користувача софт. Програма дозволяє приховувати з видимості папки, шифрувати файли і дані на флешках, зберігати паролі та іншу інформацію в захищеному сховище, може безслідно затирати документи і вільне місце на дисках, має вбудований захист від злому.

4. Система Dekart Private Disk [18].

Дана програма призначена виключно для того, щоб створювати зашифровані образи дисків. В налаштуваннях можна вказати, які програми, що містяться в образі, будуть запускатися при монтуванні або демонтування, а також включити брандмауер, що відслідковує додатки, які намагаються отримати доступ до диска.

5. Система R-Crypto [18].

Ще один софт для роботи з закріптованнимі контейнерами, які виступають в якості віртуальних носіїв даних. Контейнери R-Crypto можна підключати як флешки або звичайні жорсткі диски і відключати їх від системи при виконанні умов, зазначених в налаштуваннях.

6. Система Crypt4Free [18].

Crypt4Free - програма для роботи з файловою системою. Вона дозволяє шифрувати звичайні документи і архіви, прикріплені до листів файли і навіть інформацію в буфері обміну. До складу програми також входить генератор складних паролів.

7. Система RCF EnCoder / DeCoder [18].

Цей маленький шифрувальник дає можливість захищати каталоги і що містяться в них документи за допомогою створюваних ключів. Основною особливістю RCF EnCoder / DeCoder є можливість шифрування текстового вмісту файлів, а також те, що він поставляється тільки в портативної версії.

Це був невеликий список відомих, і не дуже, програм для шифрування файлів і папок на жорстких дисках комп'ютера і знімних носіях. Всі вони мають різні функції, але виконують одну задачу - приховувати інформацію користувача від сторонніх очей.

1.5.5 Перспективи розвитку шифрування

Кібератаки постійно розвиваються, тому фахівці з питань безпеки повинні залишатися зайнятими в лабораторії, створюючи нові схеми, щоб тримати їх на відстані. Експерти сподіваються, що новий метод, який називається Honey Encryption, стримуватиме хакерів, подаючи фальшиві дані для кожного неправильного вгадування коду ключа. Цей унікальний підхід не тільки сповільнює зловмисників, але й потенційно ховає правильний ключ у копиці помилкових надій. Потім з’являються нові методи, такі як розподіл квантових ключів, який розділяє ключі, вбудовані в фотони, над волоконно-оптичним, які можуть мати життєздатність зараз і через багато років у майбутньому.

Незалежно від того, чи це захист ваших повідомлень електронної пошти чи збережених даних, слід включити певний тип шифрування у вашій лінійці інструментів безпеки.

1.6 Огляд робіт за темою дисертації

Відновлення контролю користувача для менеджерів паролів за допомогою автоматичного виправлення (підтримані операції з безпеки, VaultIME. 2018-05-01).

VaultIME [19] пропонує підштовхувати більшу кількість користувачів до прийняття менеджерів паролів, пропонуючи їм відчутну перевагу, лише незначним чином втручаючись у їхній поточний досвід використання. Замість "автоматичного заповнення" полів паролів, вони пропонують "автоматично виправляти" паролі на випадок незначних помилок. Працюючи як додаток на мобільних телефонах, VaultIME запам’ятовує паролі користувачів на основі кожного додатка та виправляє помилково введені паролі в наборі, що допускає друкарські помилки.

Менеджери паролів [20] - це все про довіру та прозорість (MDPI AG, 2020-10-01).

Пароль вважається першою лінією захисту при захисті облікових записів в Інтернеті, однак існують проблеми, коли люди погано використовують власні паролі, наприклад, повторне використання паролів, бо їх важко запам’ятати. Менеджери паролів видаються перспективним рішенням, яке допоможе людям обробляти свої паролі. Однак популярність менеджерів паролів низька, хоча вони широко доступні, і проведено багато досліджень для користувачів менеджерів паролів. Висновки показують, що зручність використання не є основною проблемою, швидше за все відсутність довіри та прозорості є основними причинами низького рівня прийняття менеджерів паролів. Користувачі менеджерів паролів турбуються про довіру та безпеку, тоді як є декілька проблем із користувацькими інтерфейсами та функціями менеджерів паролів.

Інформація (розробка та впровадження алгоритму 4DES (Universitas Syiah Kuala, 2017-01-01) [21], що необхідна для життя будь-якої людини, оскільки за

відсутності відповідної інформації будь-яку справу неможливо зробити належним чином. Проблема безпеки є одним із найважливіших аспектів файлу, що містить конфіденційну інформацію. Це дослідження мало на меті оцінити аналіз продуктивності та створити прототип системи безпеки файлів медичних записів за допомогою алгоритму 4DES. Алгоритм 4DES - це більш надійний і здатний захищати інформацію належним чином варіант алгоритму 3DES. Він цілком може забезпечити надійність файлів.

Методи та технології захисту інформаційно-комунікаційних систем були висвітлені у огляді елементів [22] електронної ідентифікації (електронна ідентифікація, підпис та безпека інформаційних систем (ACTA MONTANISTICA SLOVACA, 2002-12-01)), таких як статичний пароль, динамічний пароль та єдиний вхід. До цієї категорії належать також біометричні та динамічні характеристики перевіреної людини. Широко поширеною є автентифікація на основі володіння елементами ідентифікації, такими як різні картки та калькулятори автентифікації.

Оцінка поточної практики створення та використання паролів як механізму контролю доступу до інформації(SOUTH AFRICAN JOURNAL OF INFORMATION MANAGEMENT, 2007-11-01)

Найважливіше питання управління інформацією [23] в організації: забезпечення належного контролю та застосування, щоб забезпечити доступ людей до інформації. Паролі широко використовуються як основний механізм контролю для ідентифікації користувачів, які хочуть отримати доступ до систем, додатків, файлів даних, мережевих серверів або особистої інформації. Даний механізм контролю забезпечує належий контроль та застосування даних і забезпечує доступ до інформації.

Обмін ключами і захищені протоколи зміни пароля(SYMMETRY, 2017-07-01). Нові протоколи обміну ключами (PAKE) та захищена зміна пароля (PPC) без будь-яких симетричних криптосистем чи відкритих ключів, були показані авторами у статті [24]. Безпека запропонованих протоколів базується

на обчислювальному припущенні Діффі-Хеллмана у випадковій моделі оракула. Запропонована схема може протистояти як підробці сервера, так і атакам відмови в обслуговуванні.

Багатофакторна автентифікація для постачальників символічних посвідчень(JOURNAL OF INTERNET SERVICES AND APPLICATIONS, 2020-12-01). Стаття [25] відображає рішення для автентифікації користувачів у декількох адміністративних доменах. Академічні федерації, такі як Бразильська федерація, є прикладами цієї моделі на практиці. Більшість установ, які беруть участь в академічних федераціях, використовують аутентифікацію на основі паролів для своїх користувачів, а зловмиснику потрібно лише знайти один пароль, щоб персоніфікувати користувача у всіх федеративних постачальниках послуг.

Багатофакторна автентифікація з’являється як рішення для підвищення надійності процесу автентифікації. Її використання допомогає використовувати онлайн сервіси безпечно та дозволяє зберігати конфіденціальні дані у мережі.

1.6.1 Недоліки існуючих менеджерів паролів

Більшість менеджерів мають не дуже простий інтерфейс, викоритовувати їх буває складно. Бувають випадки, коли доведеться увійти в програму, скопіювати логін/пропуск, а потім повернутися на попередній веб-сайт/програму, щоб ввести свої дані. Не завжди можна перейти на сайт через плагін менеджера без переходу в налаштування. Це може бути трохи незрозумілим. Іноді це спрацьовує, іноді ні, і незрозуміло чому.

Іноді виникають помилки під час пошуку певного логіна/пропуску, коли на цьому сайті є кілька збережених облікових записів. Було б дуже корисно мати функцію перегляду, де можливо легко бачити, хто має доступ до чого.

Було б добре, якби функція створення пароля не займала стільки кліків у самій системі. Гарно були би додати спосіб групової зміни паролів. Це важко зробити, але було б так зручно. Деякі веб-сторінки погано взаємодіють із багатьма менеджерами. Іноді доводиться копіювати ім’я користувача та пароля із плагіна браузера. Коли забагато адрес у пошуку, плагін не дає достатньо місця, щоб побачити їх усі, і прокручування цього списку заважає.

Посилання на перелік паролів часто перекриває вміст у полі введення веб-сторінки. Багато незручних функцій менеджерів можна було б покращити.

1.7 Мета та завдання досліджень

Вдосконалення існуючих менеджерів паролів для забезпечення підвищеної безпеки зберігання деяких персональних даних користувачів (логінів та паролів), а також для запобігання заволодіння ними сторонніми особами.

Для досягнення мети необхідно вирішити наступні завдання:

1. Висвітлити проблему зберігання персональних даних.

2. Провести огляд сучасних методів та засобів збереження та захисту персональних даних.

3. Провести огляд публікацій за темою дисертації.

4. Розробити архітектуру менеджера паролів.

5. Реалізувати основні підсистеми та модулі менеджера паролів відповідно до розробленої архітектури.

6. Провести тестування працездатності розробленого менеджера паролів.

РОЗДІЛ 2
ОГЛЯД ТЕХНОЛОГІЙ

2.1 Опис інструментарію

Менеджери паролів можуть бути використані як додаток у веб-браузері так і бути встановлені на ПК як окрема програма.

Операційна система [26] - це найважливіше програмне забезпечення, яке працює на комп'ютері. Він керує пам’яттю і процесами комп’ютера, а також усім його програмним та апаратним забезпеченням.

Робота операційної системи :

Операційна система (ОС) вашого комп’ютера керує всім програмним та апаратним забезпеченням комп’ютера. У більшості випадків одночасно працює кілька різних комп’ютерних програм, і всі вони потребують доступу до центрального процесора (ЦП) та пам’яті вашого комп’ютера. Операційна система координує все це, щоб переконатися, що кожна програма отримує те, що їй потрібно.

2.1.1 Типи операційних систем

Операційні системи зазвичай поставляються попередньо завантаженими на будь-який комп’ютер, який ви купуєте. Більшість людей використовують операційну систему, яка постачається разом із комп’ютером, але можна оновити або навіть змінити операційну систему. Три найпоширеніші операційні системи для персональних комп’ютерів – це Microsoft Windows, macOS та Linux.

1. Система Microsoft Windows.

Microsoft створила операційну систему Windows в середині 1980-х років. Було багато різних версій Windows, але найновішими є Windows 11 (див.

рис.1) (випущена в 2021 році), Windows 10 (випущена в 2015 році), Windows 8 (2012), Windows 7 (2009) і Windows Vista (2007). Windows поставляється попередньо завантаженою на більшість нових ПК, що допомагає зробити її найпопулярнішою операційною системою у світі.

Рис.1 Робочий стіл Windows 11.

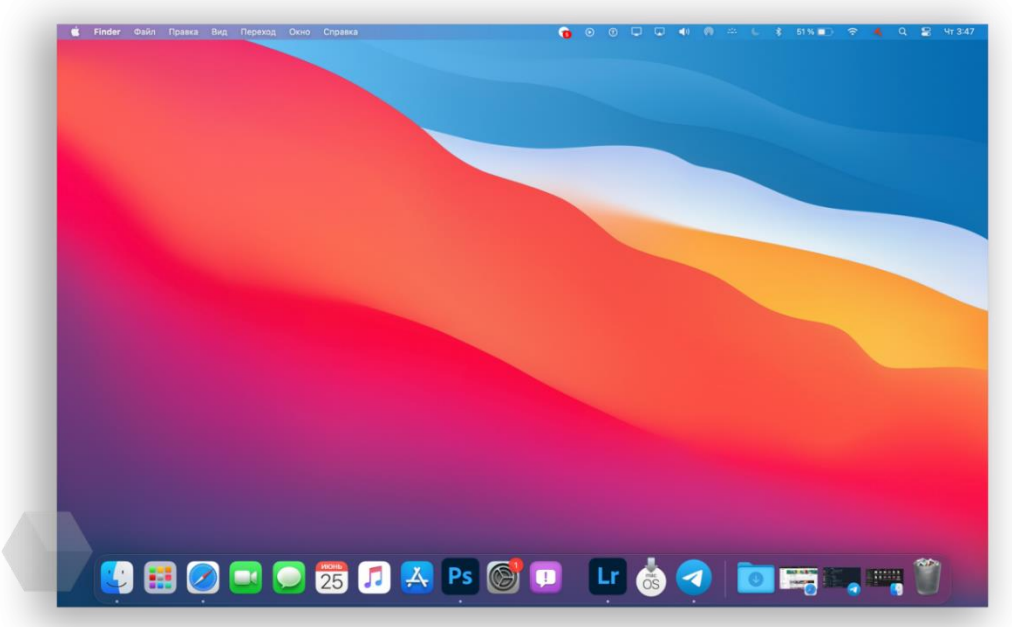


Рис.2 Інтерфейс MacOS.

MacOS (раніше називалася OS X) (див. рис.2) — це лінійка операційних систем, створених Apple. Він попередньо завантажений на всі комп’ютери Macintosh або Mac.

За даними StatCounter Global Stats, користувачі macOS становлять менше 10% глобальних операційних систем, що набагато менше, ніж відсоток користувачів Windows (більше 80%). Однією з причин цього є те, що комп’ютери Apple, як правило, дорожчі. Однак багато людей віддають перевагу зовнішньому вигляду macOS, а не Windows.

Linux (див. рис.3) — це сімейство операційних систем з відкритим кодом, що означає, що їх може змінювати та поширювати будь-хто в усьому світі. Це відрізняється від програмного забезпечення, такого як Windows, яке може бути змінене лише компанією, що володіє ним. Переваги Linux полягають у тому, що він безкоштовний, і є багато різних дистрибутивів або версій, з яких ви можете вибрати.

За даними StatCounter Global Stats, користувачі Linux становлять менше 2% глобальних операційних систем. Однак більшість серверів працюють під управлінням Linux, оскільки його відносно легко налаштувати.

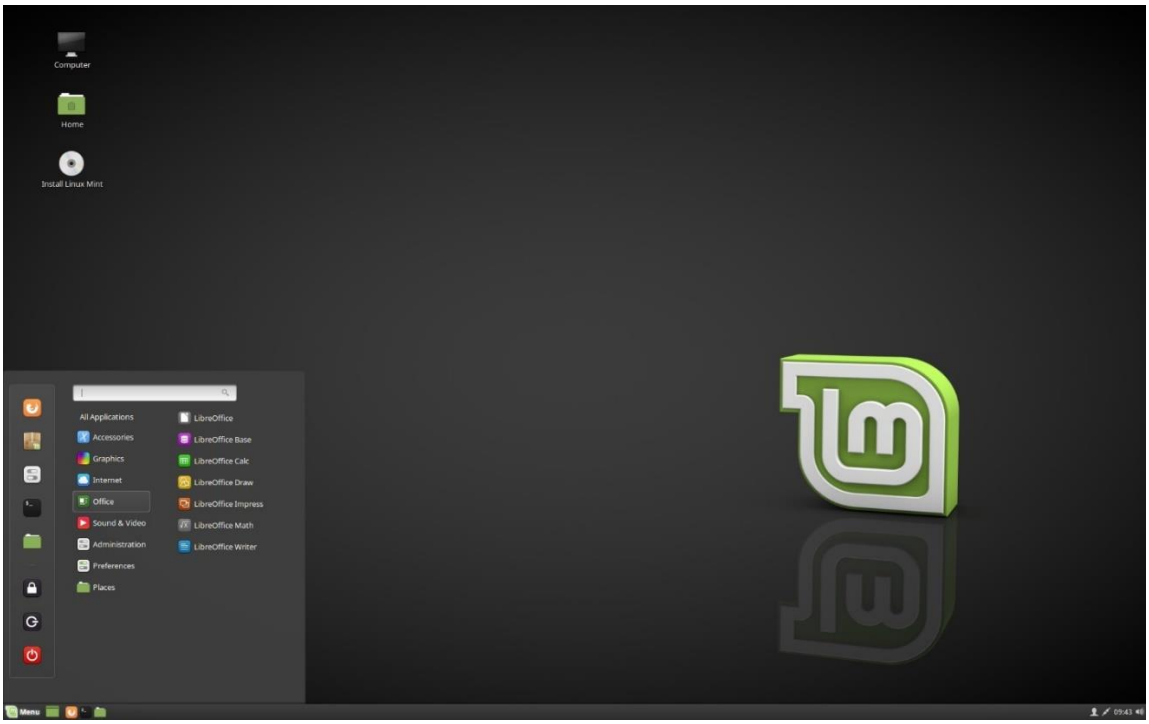


Рис.3 Інтерфейс Linux.

Використання веб-версії є більш легким та швидшим під-час работи у браузері, перегляд та копіювання паролів, логінія та адрес сайтів.

Великої різниці у використанні саме веб-версій менеджерів паролів немає. За простотою та поширіністю використання оберемо Windows.

2.1.2 10 найкращих мов для розробки веб-додатків

1) Мова Java [27].

Рис.4 Логотип Java.

Надійні функції безпеки та функціональні можливості роблять Java ідеальною для веб-сайтів. Логотип Java див. рис.4.

Згідно з опитуванням, Java є другою за популярністю мовою програмування після Python, за нею йде JavaScript. Протягом десятиліть Java була, безперечно, найпопулярнішим вибором, оскільки вона забезпечує — вертикальну та горизонтальну — масштабованість.

Java приносить більшу цінність веб-розробці через його кросплатформну природу. Наприклад, Java RUNTIME ENVIRONMENT переводить коди в машинний код, сумісний з рідною ОС, будь то Linux, Windows або macOS. Більшість підприємств віддає перевагу Java через її надійні функції безпеки.

Java має чудові фреймворки для розробки веб-додатків, зокрема Spring, Hibernate, Play, GWT, JSF. Крім того, Java може створювати складні веб-програми з високою продуктивністю.

Spring завжди був найпопулярнішим і найнадійнішим фреймворком для будь-якої компанії з розробки веб-додатків Java.

Найпопулярніші веб-програми: Opera Mini (веб-браузер), Thin-Free Office, Twitter, Spotify, Nimbuzz, Murex.

Підходить для: великих додатків корпоративного класу, додатків великих даних (Hadoop і Apache), урядових веб-сайтів, наукових проектів.

2) Мова PHP [27].

Рис.5 Логотип PHP.

PHP є однією з бажаних мов для розробки веб-додатків у 2021 році. Це динамічна мова сценаріїв на стороні сервера, що робить його розумним вибором для створення повнофункціональних служб веб-розробки PHP. WordPress є найпопулярнішим програмним додатком, створеним на PHP.

За даними W3techs, PHP поширений серед 79,1% веб-сайтів з такими гігантами, як Wikipedia, Facebook і Tumblr. Логотип PHP див. рис.5.

Мова з відкритим вихідним кодом з (вбудованими інструментами та модулями) робить його ідеальним для статичних і динамічних веб-сайтів. Завдяки кільком бібліотекам фреймворків та інструментам автоматизації розробка стає швидшою та простішою.

Крім того, його керівництво використовувало для розробки високоякісних веб-додатків за менший час, що полегшило їх налагодження мовою програмування.

Він має потужні фреймворки, такі як Laravel, що дозволяє програмістам PHP швидко створювати веб-додатки, використовуючи архітектурний шаблон сучасного контролера перегляду.

Найпопулярніші веб-програми: Facebook, Yahoo, Wikipedia, WordPress, Tumblr, MailChimp, Flickr.

Підходить для: систем управління навчанням, веб-сайтів електронної комерції, систем керування вмістом

HTML є основною потребою розробки веб-додатків для визначення структури та макета веб-сторінок. Однак за сучасними стандартами підприємства цього недостатньо.

3) Мова Python [27].

Рис.6 Логотип Python.

Жодна інша мова не отримала такої популярності, як Python. Це одна з улюблених мов для розробки веб-додатків, і її послідовне зростання продовжується серед спільноти розробників Python. Логотип Python див. рис.6.

З деяких переконливих причин простота кодів робить Python — першим вибором для початківців. Це значно економить час розробників, що робить його ідеальним для розробки користувацьких веб-додатків. Будь-яка фірма з розробки веб-сайтів, яка шукає швидкого виконання проектів із поєднанням новітніх технологій, віддає перевагу Python.

Python допомагає розробникам створювати серверну частину програми.

Як і JavaScript, Python також має широку підтримку спільноти з фреймворками, бібліотеками та інструментами, які сприяють швидкому розробці веб-додатків.

Django допомагає створювати повні веб-програми. Однак для більшого контролю ви також можете використовувати Flask, мінімалістичний фреймворк веб-розробки.

Найпопулярніші фреймворки: Django, Pyramid, Bottle, Flask, Cheery Py, WebApp2 і TurboGears.

Популярні веб-програми: Dropbox, Pinterest, Google, Uber, Instagram, Instacart.

Підходить для: машинного навчання штучного інтелекту, науки про дані, відеоігор, написання системних скриптів

4) Мова JavaScript [27].

Рис.7 Логотип JavaScript.

Текстова мова програмування, яка використовується як на клієнтській, так і на серверній стороні, що робить веб-сторінки інтерактивними. Код JavaScript простий, гнучкий, його легко писати та налагоджувати. Більше того,

ви можете використовувати його в сценаріях, написаних іншими мовами програмування. Логотип Python див. рис.7.

Розробка додатків JavaScript може перетворювати статичні веб-сайти у веб-програми, додаючи функції меню, анімацію та взаємодії при наведенні курсора.

Детальна документація Google збільшує швидкість розробки веб-додатків.

Універсальність і потужність JavaScript можуть працювати у браузерах і серверах за допомогою NodeJS і доступних фреймворків і бібліотек для розробки веб-додатків.

Згідно з результатами опитування розробників Stackoverflow 2020, 67,7% професійних розробників люблять програмувати на JavaScript.

Можна використовувати ReactJS і Angular для інтерфейсу, NodeJS для бекенда і React Native для розробки кросплатформних (android та iOS) мобільних додатків.

Найпопулярніші веб-програми: Netflix, Uber, LinkedIn, Candy Crush.

Підходить для: розробки на стороні сервера, розробки ігор, створення веб-серверів і серверних додатків.

5) Мова Swift [27].

Рис.8 Логотип Swift.

У 2014 році Apple вперше анонсувала програмне забезпечення для розробки додатків для iOS загального призначення. Потужна інтуїтивно зрозуміла мова програмування, яка використовується для розробки додатків iOS і macOS. Логотип Swift див. рис.8.

Це нова мова програмування, яка використовується для створення додатків iOS і macOS.

Переваги міжплатформного використання, динамічна бібліотека, безпроблемне обслуговування коду, швидша розробку та багато іншого.

Підходить для: мобільних і настільних програм, хмарних сервісів.

6) Мова Kotlin [27].

Рис.9 Логотип Kotlin.

Він ідеально підходить для розробки на стороні сервера та клієнта, що схиляло розробників до того, що його популярність зросла після того, як Google оголосила його для розробки додатків для Android. Універсальна мова програмування статичного типу підтримує служби розробки Kotlin, оскільки є одночасно об'єктно-орієнтованим і функціональним.

Kotlin є високорівневим і суто об’єктно-орієнтованим, має на увазі, що «кожне значення є об’єктом», і немає доступних примітивних типів даних. Він має надійне і динамічне введення тексту та автоматичне збирання сміття (форма для керування пам'яттю). Логотип Kotlin див. рис.9.

Підходить для: Android і веб-додатку, вбудованих систем.

7) Мова Ruby [27].

Рис.10 Логотип Ruby.

Динамічна мова програмування з відкритим вихідним кодом, швидка та безпечна природа робить її ідеальною для програм соціальних мереж, електронної комерції. Крім того, оскільки він є зрілим, він пропонує стабільність і зручність для сервісів розробки Ruby on Rail для стартапів.

Ruby набрав обертів після запуску Ruby on Rails. Це також підвищує продуктивність і допомагає набагато швидше доставляти програми для запуску. Логотип Ruby див. рис.10.

Крім того, його фреймворк (Ruby on Rails) є великим хітом в індустрії веб-розробки на стороні сервера.

Найпопулярніші веб-програми: Twitter, GitHub, Groupon і AirBnb.

Підходить для: електронної комерції, розробки веб- програм.

8) Мова TypeScript [27].

Рис.11 Логотип TypeScript.

По-перше, це сучасна мова програмування для веб-розробки.

Наднабір JavaScript, що робить його важливим вибором для розробки інтерфейсних програм для браузера. Як C++ для C.

Ця суворо типізована об’єктно-орієнтована мова називається набором інструментів або додаткових функцій JavaScript. Він вводить статичну інформацію в код JavaScript. Це робить схильний до помилок код простим і легшим для читання, а також легше налагоджувати.

JavaScript є динамічним і багатим на гнучкість. Однак гнучкість кодів робить його схильним до помилок. TypeScript вводить статичну інформацію в код JavaScript. Це робить коди менш схильними до помилок, їх легко читати, розуміти та налагоджувати розробку веб-додатків.

Підходить для: програм JavaScript, JavaScript IDE, статичної перевірки.

Однак протягом майже десятиліття JavaScript був твердим оплотом у мовах програмування для веб-розробки. Логотип TypeScript див. рис.11.

9) Мова Perl [27].

Рис.12 Логотип Perl.

Perl — це мова програмування, яка розвивалася та привернула увагу веб-розробників. Раніше єдиною метою була робота з текстом, але тепер вона розширилася до веб-додатків, системного адміністрування, потреб,

мережевого програмування, адміністрування мережі, розробка графічного інтерфейсу тощо. Логотип Perl див. рис.12.

Маніпулювання текстом і швидкий цикл розробки роблять Perl ідеальним для розробки веб-програм і адміністрування мережі. Це дає переваги, оскільки є ефективною мовою і швидко створює програми.

Таким чином, це ідеальна мова для підприємств, якщо працює конкретна концепція. Однак Perl не такий поширений, як інші мови програмування.

Підходить для: розробки графічного інтерфейсу, мережевого програмування, системного адміністрування

10) Мова C# [27].

Рис.13 Логотип C#.

Він входить до 10 найкращих мов програмування для розробки корпоративних додатків, незважаючи на те, що є мовою програмування для початківців. Логотип C# див. рис.13.

Популярний, оскільки підходить для об’єктно-орієнтованої розробки.

Як і C++ і C Sharp — це об’єктно-орієнтована мова загального призначення, заснована на C. Спочатку вона була розроблена компанією Microsoft як частина її платформи .NET для створення додатків Windows.

Найпопулярніші програми для Windows: Skype, Photoshop, Microsoft Office, SQL Server 2012.

Підходить для: додатків Windows, розробки веб-додатків, створення 2D і 3D ігор.

Розмір і складність — менший розмір модулів і бібліотек добре підходить для невеликих веб-програм. Однак, якщо є більші веб-проекти з широкою функціональністю (наприклад, сайти соціальних мереж, електронна комерція), потрібна гнучкість у виборі.

Розширюваність — бізнес спрямований на розширення горизонтів. Якщо треба збільшити трафік на веб-сайтах, сервери веб-програм повинні витримувати велике навантаження без затримок. Кілька вибраних технологій підвищать загальну масштабованість.

Час, гроші, ресурси — швидке виконання та швидша розробка веб-додатків створюють жорстку конкуренцію конкурентам.

Після цього треба розглянути розвиток з нуля; ці фактори не повинні визначати, яку мову програмування треба вибрати:

1. Попередні веб-програми. Незалежно від того, наскільки успішно виконувалися попередні проекти, їхні технологічні стеки не повинні співвідноситися з мовою програмування, яку ви вибрали для свого проекту. Усі проекти мають специфічні вимоги та є унікальними.

2. Особисті переваги — потрібно дотримуватися останніх тенденцій і технологій.

3. Проекти конкурентів. Навчатися на досвіді інших – розумна річ. Але немає сенсу завжди наслідувати те, що роблять інші.

Перевагою технології є її універсальність. Будь-який технологічний стек або мова програмування керує ринком, поки щось не зруйнує його. Веб-розробка також не є винятком.

Будь-яка компанія з розробки веб-додатків надасть підприємствам широкий вибір розробників. Жорсткого правила не існує. Однак, вибираючи будь-яку мову програмування, слід дотримуватися основ, або перелічуючи вимоги, звернувшись до експертних думок щодо кращих практик.

Вибираючи мову програмування можно звернути увагу, що Java проста у використанні, написанні, компіляції, налагодженні та навчанні, ніж альтернативні мови програмування. Java менш складна, ніж C++, в результаті Java використовує автоматичне виділення пам'яті та збирання сміття.

Код Java працює на будь-якій машині, не потрібно встановлювати спеціальне програмне забезпечення, але JVM має бути присутнім на машині.

Java має менеджер безпеки, який визначає доступ до класів.

У Java пам'ять ділиться на дві частини, одна - купа, а інша - стек. Кожного разу, коли ми оголошуємо змінну, JVM надає пам’ять із стека або з простору купи. Це допомагає зберегти інформацію та легко її відновити.

Програма може виконувати багато завдань одночасно.

Java споживає пам'ять і значно повільніше, ніж мови, скомпільовані нативно, такі як C або C++.

У Java управління пам’яттю здійснюється за допомогою збирання сміття, і щоразу, коли запущений збірник сміття, це впливає на продуктивність програми. Це пов’язано з тим, що всі інші потоки в потрібно зупинити, щоб потік збирача сміття запрацював.

Крім того, Java перевершує різні мови з точки зору можливостей роботи, які вона може запропонувати. На додаток до різних плюсів і мінусів Java, вона була на 1-й позиції в індексі TIOBE протягом останніх трьох років.

Для веб-розробки на яві написано кілька фреймворків. Spring — це платформа Java, яка значно спрощує створення програм. Однією з найбільших причин його популярності є те, що він підтримує багато. Розробникам не потрібно використовувати сервер або будь-який інший корпоративний контейнер. Це робить весь фреймворк надзвичайно сценаріїв легким, що є значною перевагою при розробці веб-додатків.

Гнучкі конфігурації — розробники мають можливість вибрати анотації на основі XML або Java для цілей налаштування. Наявність такої опції значно спрощує роботу розробників.

Модуль AOP — розробники можуть мати різні блоки компіляції або окремий завантажувач класів.

Тестування легше — ін’єкція Spring Dependency допомагає розробникам вставляти тестові дані.

Є багато шаблонів архітектур програмних систем для Інтернету.

Враховуючи це різноманіття, програмісти здатні розробляти складне та високофункціональне програмне забезпечення.

2.2 Вибір архітектурного шаблону.

Асортимент архітектурних зразків поділяється на багато груп. Однак є три найпопулярнішими: MVC, MVP, MVVM. Шаблони, згадані вище, зазвичай використовуються у веб-додатках, мобільних і настільних додатках.

Шаблон MVC — це оригінальне архітектурне рішення для мобільних пристроїв Android та IOS, також часто реалізується у веб-додатках.

Інший шаблон — MVP (Model-View-Presenter) не часто використовується у веб-розробці. Цей шаблон має реалізацію для Windows Forms, Silverlight і ASP.NET Web Forms в середовищі .NET і в різні фреймворки Java, PHP.

MVVM є найновішим шаблоном серед розглянутих вище. Він розділяє додаток на три компоненти: Model-View-ViewModel.

Шаблон MVVM використовується Windows Presentation Foundation, Silverlight і реалізовано у фреймворках Java для Android.

Традиційно веб-розробники використовують шаблон MVC, настільні програми надають перевагу MVP.

У сфері веб-розробки, Model-View-Controller [28] є одним із найбільш обговорюваних шаблонів дизайну у світі веб-програмування сьогодні. Архітектура MVC спочатку була включена до двох основних фреймворків веб-розробки – Struts і Ruby on Rails.

Шаблон проектування в розробці програмного забезпечення — це техніка для вирішення поширеної проблеми при розробці програмного забезпечення.

Проектування моделі визначає, який тип архітектури ви використовуєте для вирішення проблеми або проектування моделі.

Проекти моделей, засновані на архітектурі MVC, відповідають шаблону проектування MVC і відокремлюють логіку програми від інтерфейсу користувача при розробці програмного забезпечення. Як випливає з назви, шаблон MVC має три шари:

Модель – представляє бізнес-рівень програми.

Перегляд – визначає презентацію програми.

Контролер – керує потоком програми.

У контексті програмування на Java модель складається з простих класів Java, View відображає дані, а контролер складається із сервлетів. Це поділ призводить до того, що запити користувачів обробляються таким чином:

Браузер на клієнті надсилає запит на сторінку контролеру, присутнім на сервері.

Контролер виконує дію виклику моделі, тим самим одержуючи дані, необхідні йому у відповідь на запит.

Потім контролер передає отримані дані у представлення.

Подання відображається та надсилається назад клієнту для відображення браузером.

Поділ програмного додатка на ці три окремі компоненти є гарною ідеєю з ряду причин.

Архітектура MVC пропонує багато переваг для програміста при розробці додатків, серед яких:

Декілька розробників можуть працювати з трьома шарами (модель, перегляд і контролер) одночасно.

Пропонує покращену масштабованість, що доповнює здатність програми розвиватися.

Оскільки компоненти мають низьку залежність один від одного, їх легко обслуговувати.

Модель може повторно використовуватися кількома представленнями, що забезпечує можливість повторного використання коду.

Застосування MVC робить додаток більш виразним і легким для розуміння.

Розширення та тестування програми стає легким.

Тому MVC є найпопулярнішим шаблоном дизайну у світі веб-програмування.

2.2.1 Реалізація MVC за допомогою Java

1. Шар моделі.

У шаблоні проектування MVC модель — це рівень даних, який визначає бізнес-логіку системи, а також представляє стан програми. Об’єкти моделі отримують і зберігають стан моделі в базі даних. Через цей рівень ми застосовуємо правила до даних, які в кінцевому підсумку представляють концепції, якими керує наша програма.

2. Шар перегляду.

Цей рівень шаблону проектування MVC представляє вихід програми або інтерфейсу користувача. Він відображає дані, отримані з шару моделі контролером, і представляє дані користувачеві.

3. Рівень контролера.

Контролер — це як інтерфейс між моделлю та переглядом. Він отримує запити користувача від рівня перегляду та обробляє їх, включаючи необхідні перевірки. Потім запити відправляються в модель для обробки даних. Після їх

обробки дані знову відправляються назад на контролер, а потім відображаються.

Архітектура MVC забезпечує абсолютно новий рівень модульності коду, що робить його набагато більш читабельним і зручним для обслуговування.

2.3 Архітектура автоматизованої системи

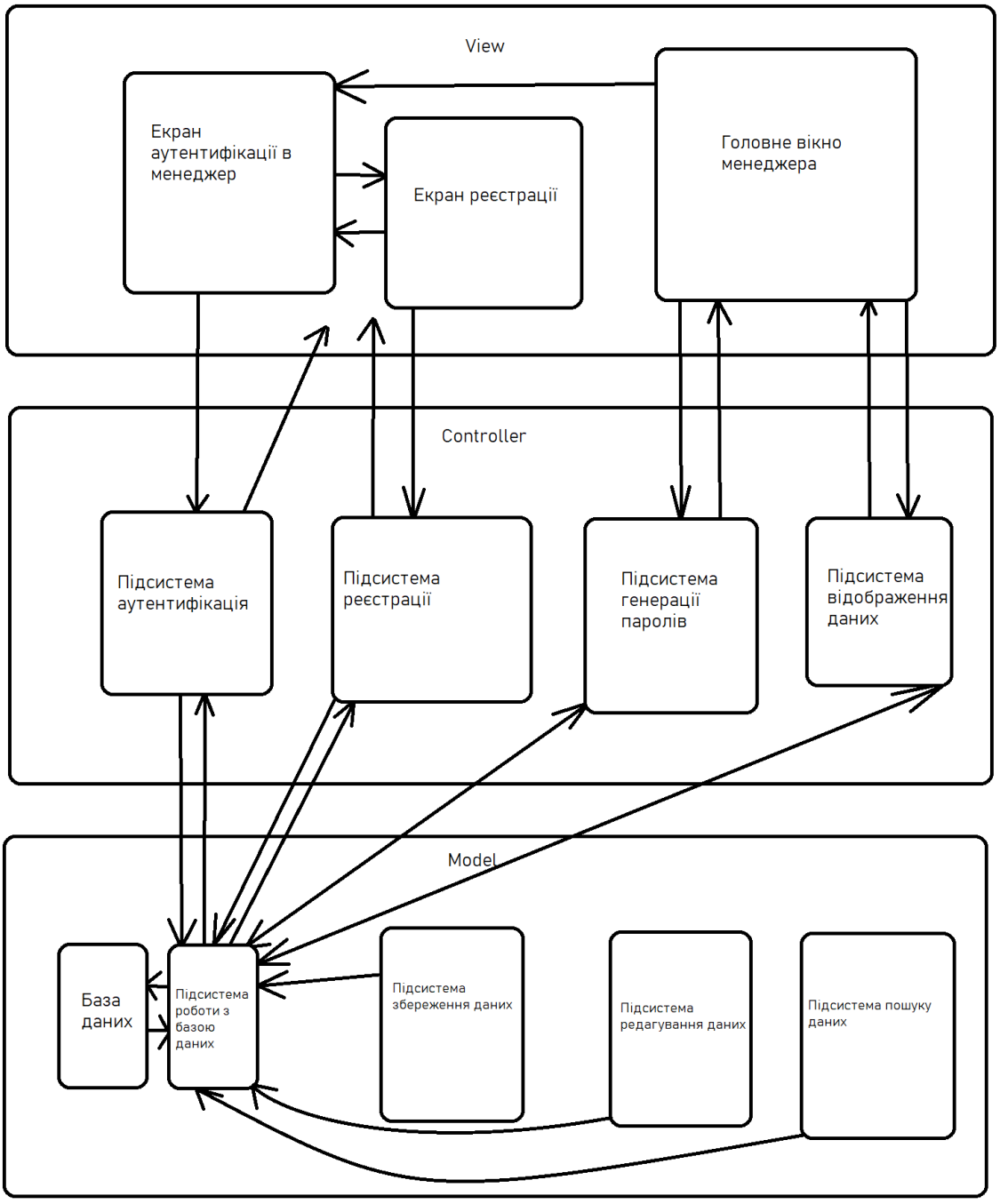


Рис. 14 Архітектура автоматизованої системи.

Архітектура автоматизованої системи складається з модулей Model, Controller та View можемо побачити на рис.14.

Модуль View відповідає за відображення даних користувача та складається з екрану аутентифікації в менеджер, екрану реєстрації та головного вікна менеджера.

Модуль Controller відповідає за бізнес-логіку системи та складається з підсистеми аутентифікації, підсистеми реєстрації, підсистеми генерації паролів та підсистеми відображення даних.

Модуль Model відповідає за внутрішню роботу системи та складається з бази даних, підсистеми роботи з базою даних, підсистеми збереження даних, підсистеми редагування даних та підсистеми пошуку даних.

Висновки

Після аналізу існуючих автоматизованих систем збереження персональних даних користувача та визначення конкретних проблем, були вибрані вид системи у вигляді веб-додатку, операційна системаWindows, мова програмування Java, фреймворк Spring та архітектурний шаблон MVC для створення необхідної автоматизованої системи збереження персональних даних користувача вдосконалюючи існуючий функціонал відповідними алгоритмами.

РОЗДІЛ 3
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Діаграми бази даних та класів.

На даній діаграмі (див. рис.15) відображені таблиці бази даних Manager. У базы даних Manager зберігаються дві таблиці. Таблиця Data використовується для збереження персональних даних користувачів під час реєстрації. У таблицю Users зберігаються дані користувачів, які вони використовують для аутентифікації на інші сайти, логін відповідає логіну збереженому у таблицю Data, дані прив’язуються до персонального логіну користувача.

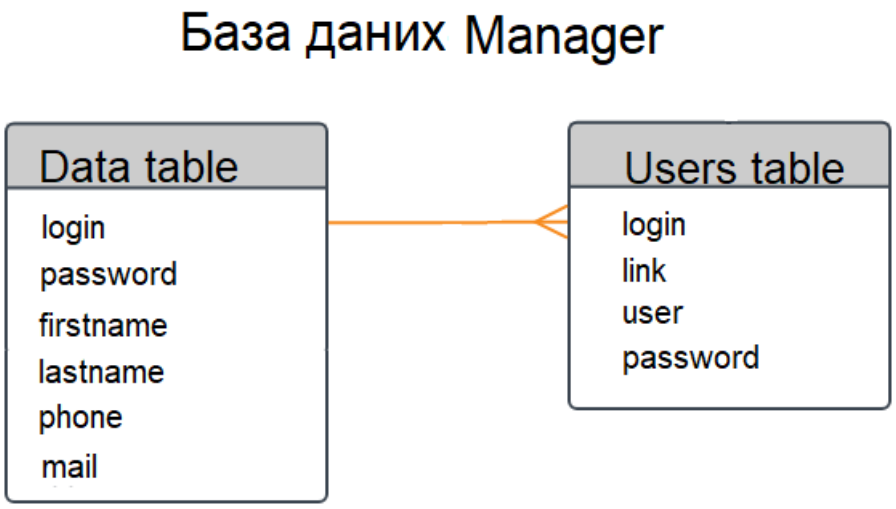


Рис. 15 Діаграма бази даних.

На діаграмі (див. рис.16) відображені класи та методи системи та взаємодії між ними.

Використання та функціонал класів та методів автоматизованої системи описані у наступному підрозділі.

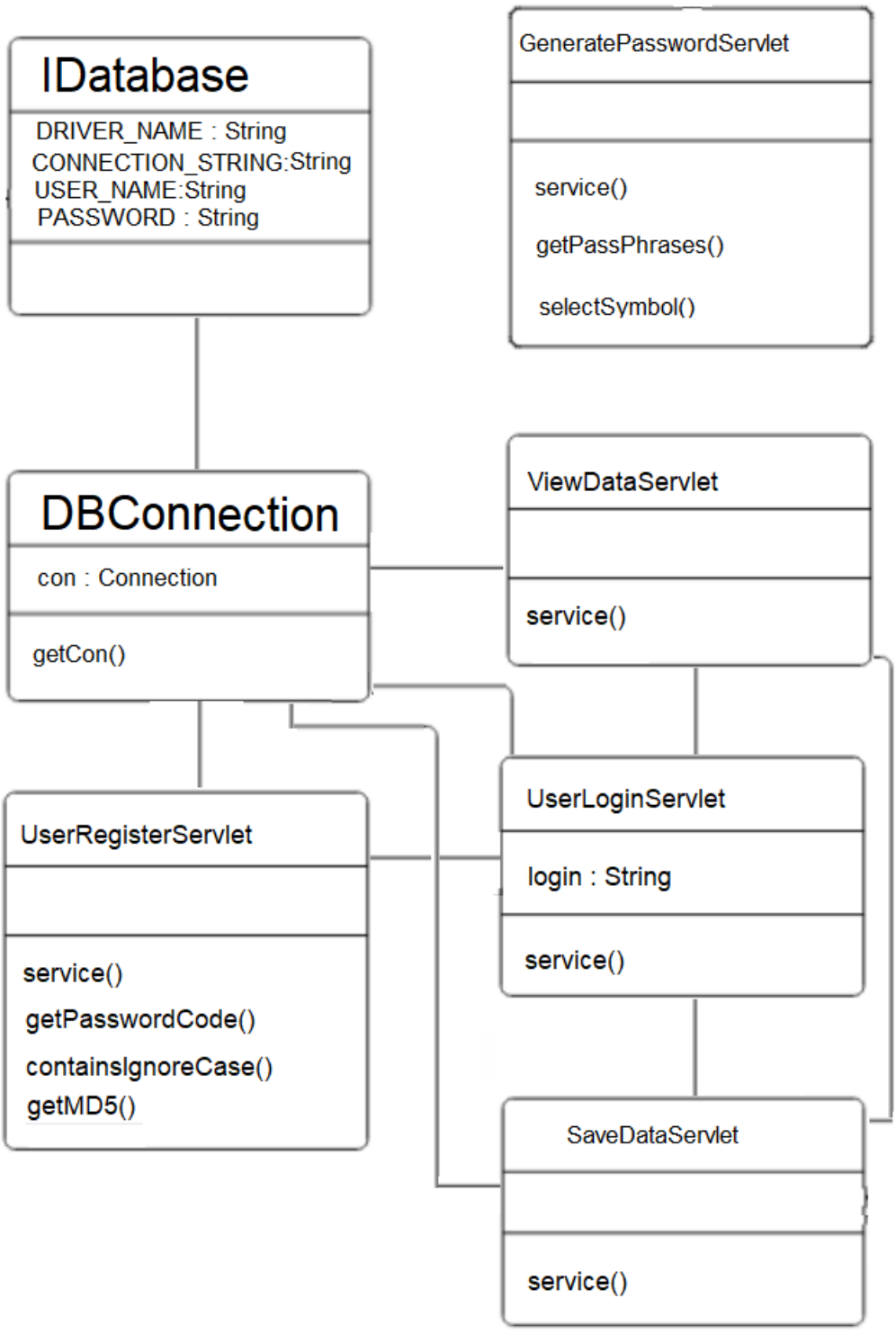


Рис. 16 Діаграма класів.

3.2 Опис класів та методів.

Клас IDatabase (див. рис.17) зберігає константи, які потрібні для підключення до бази даних. Дані константи використовуються в інших класах для отримання доступу до бази даних у якій зберігаються таблиці з даними, які використовуються у менеджері паролів.

IDatabase.java

```java
package constants;

public class IDatabase {
    public static final String DRIVER_NAME = "com.mysql.cj.jdbc.Driver";
    public static final String CONNECTION_STRING = "jdbc:mysql://127.0.0.1:3306/manager?serverTimezone=UTC";
    public static final String USER_NAME = "root";
    public static final String PASSWORD = "root";
}
```

Рис. 17 Клас IDatabase

DBConnection.java

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {
    private static Connection con;

    private DBConnection() {
    }

    static {
        try {
            Class.forName(IDatabase.DRIVER_NAME);
        } catch (Exception e) {
            e.printStackTrace();
        }

        try {
            con = DriverManager.getConnection(IDatabase.CONNECTION_STRING, IDatabase.USER_NAME, IDatabase.PASSWORD);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static Connection getCon() { return con; }
}
```

Рис.18 Клас DBConnection

Клас DBConnection (див. рис. 18) створений для підключення до бази даних. У даному класі використовуються константи з класу IDatabase. Для підключення до потрібної бази даних треба мати правильний логін та пароль від бази даних.

Метод getCon() повертає connection, який використовується для підключення до бази даних. Декілька інших класів використовують цей метод для отримання даних з бази даних або збереження даних.

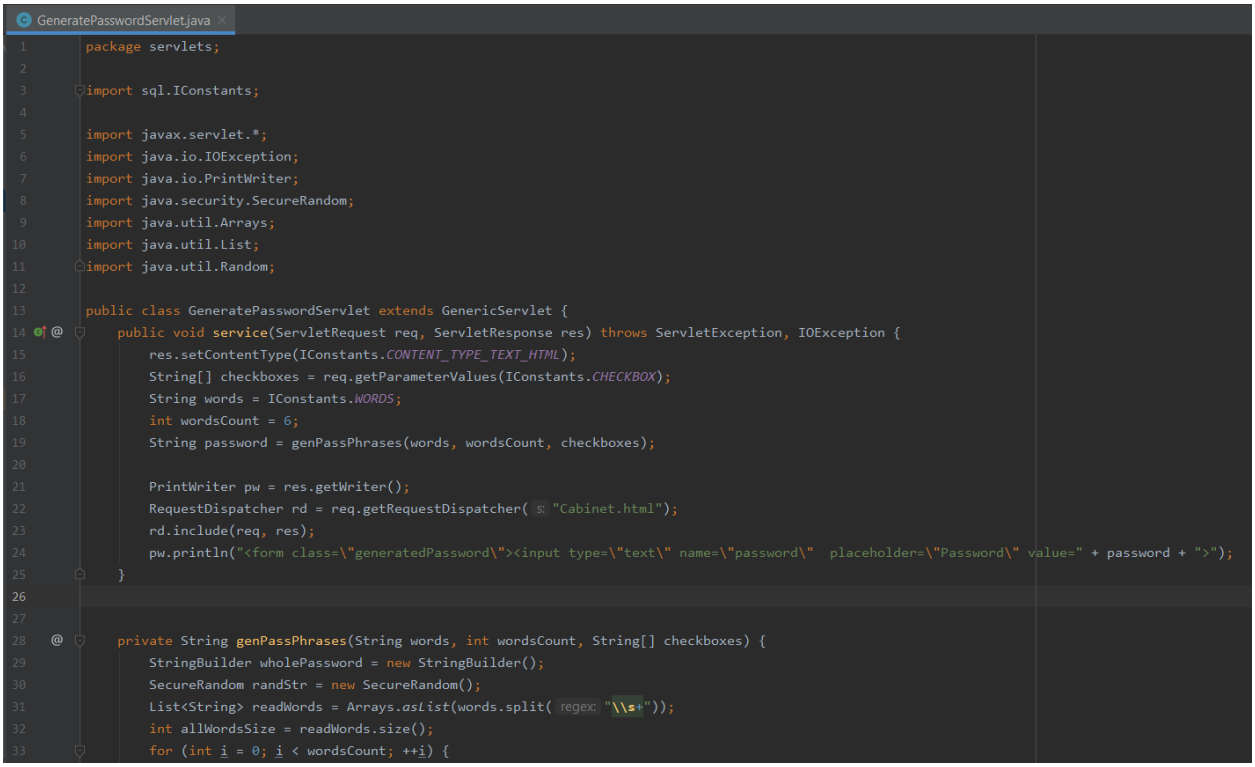


Рис. 19 Клас GeneratePasswordServlet

Клас GeneratePasswordServlet (див. рис.19 та рис.20) розширює клас GenericServlet. Даний клас використовується для генерації паролів, для збереження їх у базі даних та подальшого використання під-час аутентифікації. При генерації пароля за заданими характеристиками він відображається у полі генерації пароля для доступу до конкретного сайту. Паролі генеруються з окремих слів. Така генерація є більш захищеною та займає найбільше часу у хакерів для злому. Користувач має можливість

додати великі літери або спеціальні символи у генерацію пароля вибравши відповідну опцію.

Метод service() є основним методом класу, він отримує дані з додатку та відправляє дані у додаток. Він використовує метод getPassPhrases(), який створює пароль з різних слів та може додати великі літери чи спеціальні символи, за запитом користувача.

selectSymbol() випадково обирає потрібний символ для пароля.

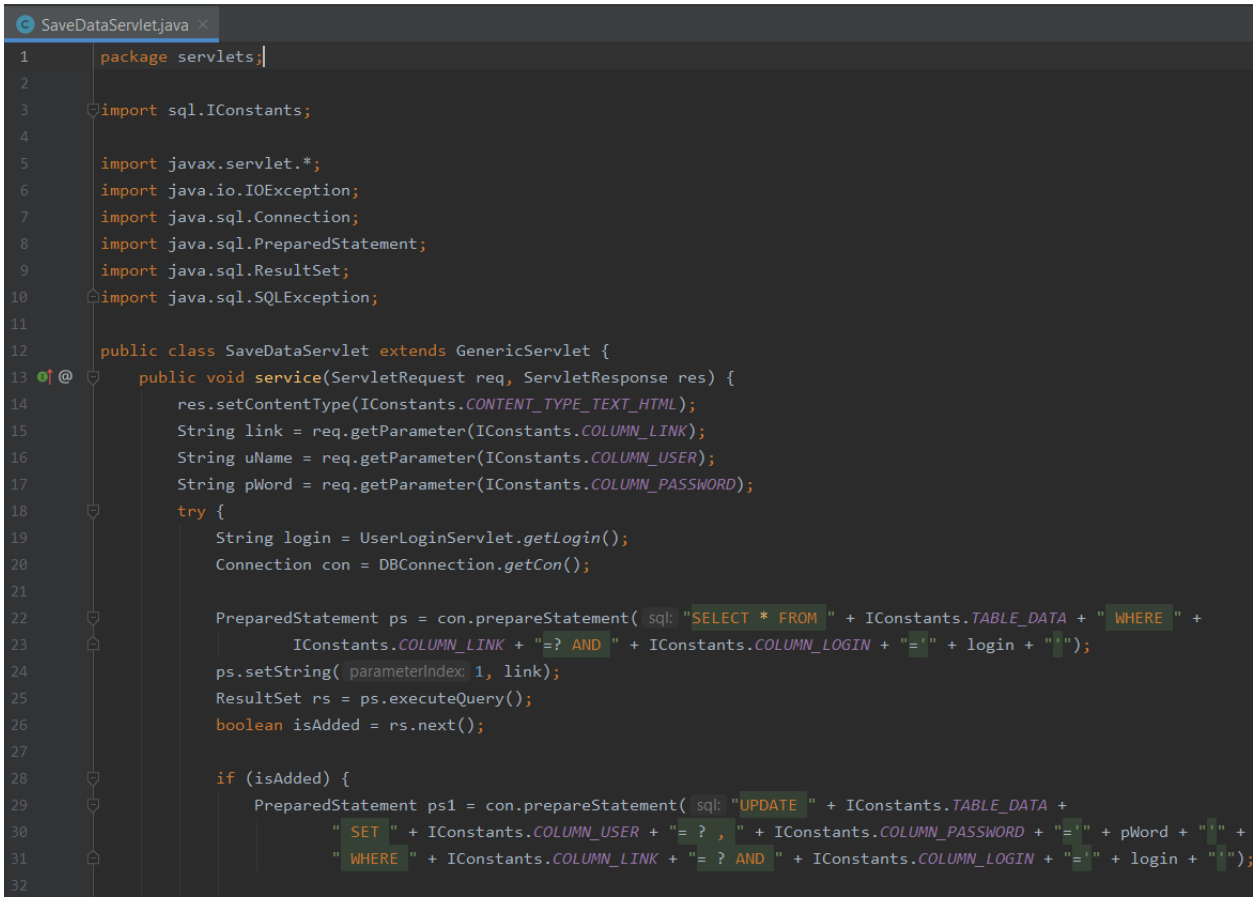
```
            if (!readWords.isEmpty()) {
                int word = randStr.nextInt(allWordsSize);
                wholePassword.append(readWords.get(word));
                if (Arrays.toString(checkboxes).contains("Use special characters")) {
                    String specialSymbol = selectSymbol( s: "!\"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~");
                    wholePassword.append(specialSymbol);
                }
                if (Arrays.toString(checkboxes).contains("Use Uppercase")) {
                    String letter = selectSymbol(wholePassword.toString());
                    wholePassword.append(letter.toUpperCase());
                }
            }
        }
        return wholePassword.toString();
    }

    private String selectSymbol(String s) {
        Random random = new Random();
        int index = random.nextInt(s.length());
        return s.substring(index, index + 1);
    }
}
```

Рис. 20 Клас GeneratePasswordServlet

Клас SaveDataServlet (див. рис.21 та рис.22) використовується для збереження в базу даних інформації, яка може використовуватися для аутентифікації на сайти, які використовує користувач. Зберігається посилання на сайт, логін та пароль користувача за ім’ям користувача.

Користувач має можливість змінювати вже збережені дані у базі даних.

Метод service() виконує усі основні функції класу. Він отримує дані з додатку, виконує потрібні дії та може повернути потрібну інформації назад до програмного забезпечення.



```java
package servlets;

import sql.IConstants;

import javax.servlet.*;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class SaveDataServlet extends GenericServlet {
    public void service(ServletRequest req, ServletResponse res) {
        res.setContentType(IConstants.CONTENT_TYPE_TEXT_HTML);
        String link = req.getParameter(IConstants.COLUMN_LINK);
        String uName = req.getParameter(IConstants.COLUMN_USER);
        String pWord = req.getParameter(IConstants.COLUMN_PASSWORD);
        try {
            String login = UserLoginServlet.getLogin();
            Connection con = DBConnection.getCon();

            PreparedStatement ps = con.prepareStatement( sql: "SELECT * FROM " + IConstants.TABLE_DATA + " WHERE " +
                    IConstants.COLUMN_LINK + "=? AND " + IConstants.COLUMN_LOGIN + "='" + login + "'");
            ps.setString( parameterIndex: 1, link);
            ResultSet rs = ps.executeQuery();
            boolean isAdded = rs.next();

            if (isAdded) {
                PreparedStatement ps1 = con.prepareStatement( sql: "UPDATE " + IConstants.TABLE_DATA +
                        " SET " + IConstants.COLUMN_USER + "= ? , " + IConstants.COLUMN_PASSWORD + "='" + pWord + "'" +
                        " WHERE " + IConstants.COLUMN_LINK + "= ? AND " + IConstants.COLUMN_LOGIN + "='" + login + "'");
```

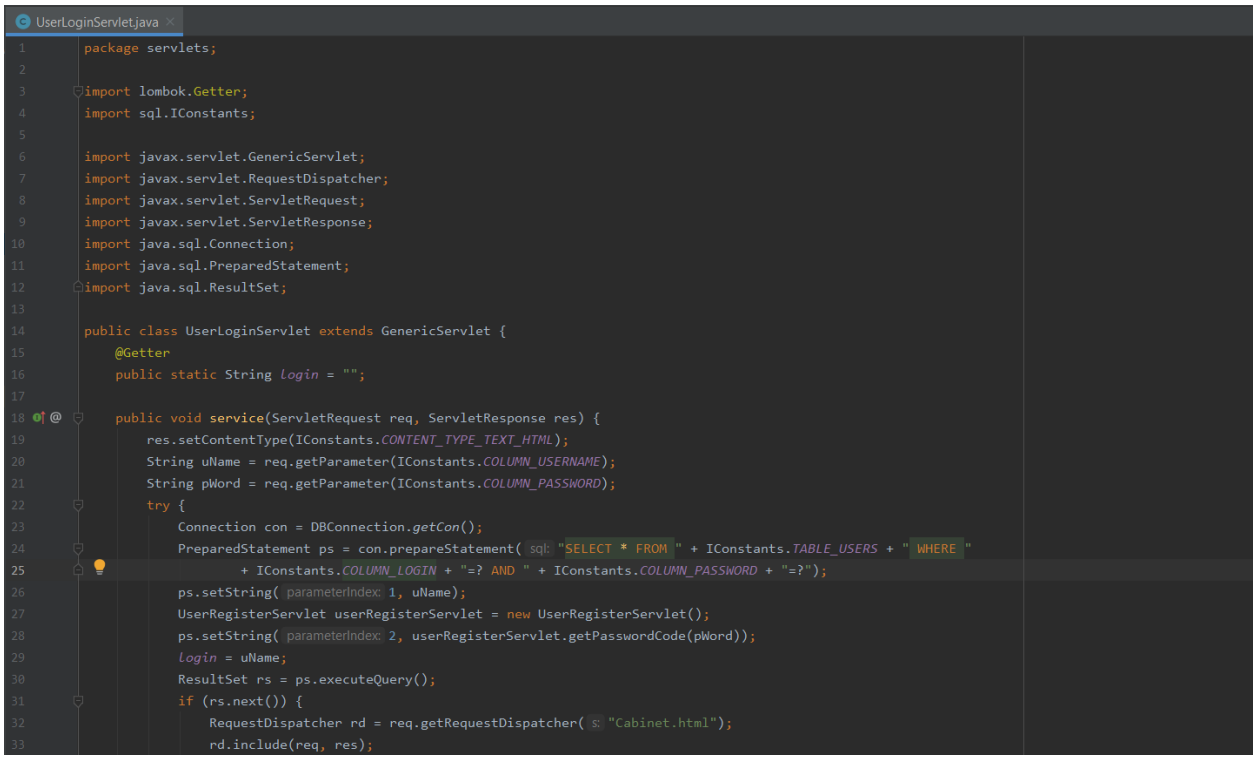
Рис.21 Клас SaveDataServlet

```java
                ps1.setString( parameterIndex: 1, uName);
                ps1.setString( parameterIndex: 2, link);
                int rs1 = ps1.executeUpdate();
                if (rs1 == 1) {
                    RequestDispatcher rd = req.getRequestDispatcher( s: "Cabinet.html");
                    rd.include(req, res);
                }
            } else {
                PreparedStatement ps2 = con.prepareStatement( sql: "INSERT INTO " + IConstants.TABLE_DATA + " VALUES ('" +
                        login + "',?,?,?)");

                ps2.setString( parameterIndex: 1, link);
                ps2.setString( parameterIndex: 2, uName);
                ps2.setString( parameterIndex: 3, pWord);
                int rs2 = ps2.executeUpdate();
                if (rs2 == 1) {
                    RequestDispatcher rd = req.getRequestDispatcher( s: "Cabinet.html");
                    rd.include(req, res);
                }
            }
        } catch (ServletException | IOException | SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Рис.22 Клас SaveDataServlet

Клас UserLoginServlet (див. рис. 23 та рис.24) використовується для аутентифікації користувача у менеджер паролів.

Користувач отримує доступ до даних менеджера паролів, які може використовувати у подальшому для аутентифікації на сайти, для яких дані він і зберігає.



```
UserLoginServlet.java
package servlets;

import lombok.Getter;
import sql.IConstants;

import javax.servlet.GenericServlet;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class UserLoginServlet extends GenericServlet {
    @Getter
    public static String login = "";

    public void service(ServletRequest req, ServletResponse res) {
        res.setContentType(IConstants.CONTENT_TYPE_TEXT_HTML);
        String uName = req.getParameter(IConstants.COLUMN_USERNAME);
        String pWord = req.getParameter(IConstants.COLUMN_PASSWORD);
        try {
            Connection con = DBConnection.getCon();
            PreparedStatement ps = con.prepareStatement( sql: "SELECT * FROM " + IConstants.TABLE_USERS + " WHERE "
                    + IConstants.COLUMN_LOGIN + "=? AND " + IConstants.COLUMN_PASSWORD + "=?");
            ps.setString( parameterIndex: 1, uName);
            UserRegisterServlet userRegisterServlet = new UserRegisterServlet();
            ps.setString( parameterIndex: 2, userRegisterServlet.getPasswordCode(pWord));
            login = uName;
            ResultSet rs = ps.executeQuery();
            if (rs.next()) {
                RequestDispatcher rd = req.getRequestDispatcher( s: "Cabinet.html");
                rd.include(req, res);
```

Рис. 23 Клас UserLoginServlet

```
            } else {
                RequestDispatcher rd = req.getRequestDispatcher( s: "index.html");
                rd.include(req, res);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Рис. 24 Клас UserLoginServlet

Клас UserRegisterServlet (див. рис. 25-28) використовується для реєстрації користувачів у менеджері паролів.

Для використання менеджера паролів спершу треба зареєструватися. Користувач вводить свої дані, які зберігаються у базу даних та в подальшому будуть використані для аутентифікації у менеджер паролів.

Дані користувача зберігаються у таблицю в звичайному вигляді, крім пароля. Перед збереженням пароля спочатку використовується алгоритм шифрування, який буде описаний пізніше. Після шифрування пароль зберігається у таблицю даних разом з іншими даними.

Окрім основного метода service() у даному класі використовується метод getPasswordCode(), його використовує не тільки цей клас.

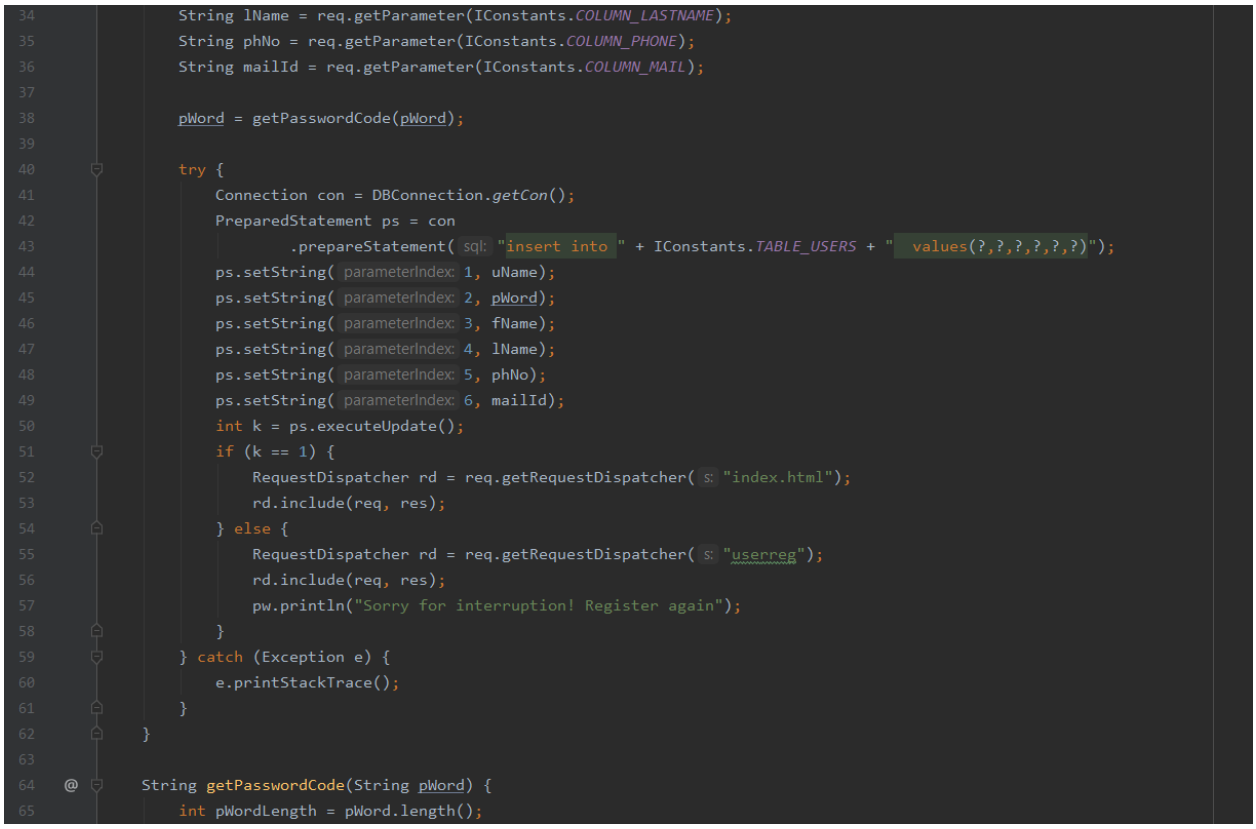
У цьому методі відбувається шифрування паролів, отримуючи сирий пароль даний метод з використанням шифруючого методу getMD5() перетворює незахищені дані у більш секретний пароль аутентифікації.

Також тут використовується метод containsIgnoreCase(), який перевіряє дані типу текст на наявність тієї чи іншої частини тексту не зважаючи на регістр літер.

UserRegisterServlet.java

```
package servlets;

import lombok.SneakyThrows;
import sql.IConstants;

import javax.servlet.GenericServlet;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import java.io.PrintWriter;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.util.Arrays;
import java.util.List;

import static sql.IConstants.ALPHABET;

        public class UserRegisterServlet extends GenericServlet {

    public UserRegisterServlet() {
    }

    @SneakyThrows
    public void service(ServletRequest req, ServletResponse res) {

        PrintWriter pw = res.getWriter();
        res.setContentType(IConstants.CONTENT_TYPE_TEXT_HTML);
        String uName = req.getParameter(IConstants.COLUMN_USERNAME);
        String pWord = req.getParameter(IConstants.COLUMN_PASSWORD);
        String fName = req.getParameter(IConstants.COLUMN_FIRSTNAME);
```

Рис. 25 Клас UserRegisterServlet



```
34          String lName = req.getParameter(IConstants.COLUMN_LASTNAME);
35          String phNo = req.getParameter(IConstants.COLUMN_PHONE);
36          String mailId = req.getParameter(IConstants.COLUMN_MAIL);
37
38          pWord = getPasswordCode(pWord);
39
40          try {
41              Connection con = DBConnection.getCon();
42              PreparedStatement ps = con
43                      .prepareStatement( sql: "insert into " + IConstants.TABLE_USERS + "  values(?,?,?,?,?,?)");
44              ps.setString( parameterIndex: 1, uName);
45              ps.setString( parameterIndex: 2, pWord);
46              ps.setString( parameterIndex: 3, fName);
47              ps.setString( parameterIndex: 4, lName);
48              ps.setString( parameterIndex: 5, phNo);
49              ps.setString( parameterIndex: 6, mailId);
50              int k = ps.executeUpdate();
51              if (k == 1) {
52                  RequestDispatcher rd = req.getRequestDispatcher( s: "index.html");
53                  rd.include(req, res);
54              } else {
55                  RequestDispatcher rd = req.getRequestDispatcher( s: "userreg");
56                  rd.include(req, res);
57                  pw.println("Sorry for interruption! Register again");
58              }
59          } catch (Exception e) {
60              e.printStackTrace();
61          }
62      }
63
64  @   String getPasswordCode(String pWord) {
65          int pWordLength = pWord.length();
```

Рис. 26 Клас UserRegisterServlet

```
66          String specialLetter = pWord.substring(pWordLength - 1);
67          String rawPassword = pWord.substring(0, pWordLength - 1);
68
69          pWord = getMD5(rawPassword);
70
71          if (containsIgnoreCase(ALPHABET, specialLetter)) {
72              String lowerCaseLetter = specialLetter.toLowerCase();
73              int specialLetterNumber = 0;
74
75              for (int i = 0; i < lowerCaseLetter.length(); i++) {
76                  specialLetterNumber = ALPHABET.indexOf(lowerCaseLetter.charAt(i)) + 1;
77              }
78
79              List<String> lettersList = Arrays.asList(pWord.split( regex: "(?<=.)"));
80              StringBuilder newCode = new StringBuilder();
81
82              int finalSpecialLetterNumber = specialLetterNumber;
83              lettersList.forEach(letter -> {
84                  if (containsIgnoreCase(ALPHABET, letter)) {
85                      String newLetter = Character.toString(ALPHABET.charAt(ALPHABET.indexOf(letter.toLowerCase()) + finalSpecialLetterNumber));
86                      newCode.append(newLetter);
87                  } else newCode.append(letter);
88              });
89              pWord = String.valueOf(newCode);
90          }
91          return getMD5(pWord);
92      }
93
94  @   boolean containsIgnoreCase(String haystack, String needle) {
95          return haystack.toLowerCase().contains(needle.toLowerCase());
96      }
97
```

Рис. 27 Клас UserRegisterServlet

```
    String getMD5(String input) {
        try {
            MessageDigest md = MessageDigest.getInstance("MD5");
            byte[] messageDigest = md.digest(input.getBytes());
            BigInteger no = new BigInteger( signum: 1, messageDigest);

            StringBuilder hashText = new StringBuilder(no.toString( radix: 16));
            while (hashText.length() < 32) {
                hashText.insert( offset: 0,  str: "0");
            }
            return hashText.toString();
        } catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Рис. 28 Клас UserRegisterServlet

ViewDataServlet.java

```
package servlets;

import sql.IConstants;

import javax.servlet.GenericServlet;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class ViewDataServlet extends GenericServlet {
    public void service(ServletRequest req, ServletResponse res) throws IOException {
        PrintWriter pw = res.getWriter();
        res.setContentType("text/html");
        try {
            Connection con = DBConnection.getCon();
            PreparedStatement ps = con.prepareStatement( sql: "Select * from " + IConstants.TABLE_DATA + " WHERE "
                    + IConstants.COLUMN_LOGIN + "='" + UserLoginServlet.getLogin() + "'");
            ResultSet rs = ps.executeQuery();
            RequestDispatcher rd = req.getRequestDispatcher( s: "Cabinet.html");
            rd.include(req, res);

            pw.println("<div class=\"tab\">Available Sites</div>");
            pw.println("<div class=\"tab\">\r\n" +
                    "       <table>\r\n" +
                    "           <tr>\r\n" +
                    "               \r\n" +
                    "               <th>Link</th>\r\n" +
                    "               <th>Login</th>\r\n" +
```

Рис. 29 Клас ViewDataServlet

```
                    "              <th>Password</th><br/>\r\n" +
                    "          </tr>");
            while (rs.next()) {
                String link = rs.getString( columnIndex: 2);
                String login = rs.getString( columnIndex: 3);
                String password = rs.getString( columnIndex: 4);
                pw.println("<tr><td>" + link + "</td>");
                pw.println("<td>" + login + "</td>");
                pw.println("<td>" + password + "</td></tr>");
            }
            pw.println("</table>\r\n" +
                    "   </div>");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Рис. 30 Клас ViewDataServlet

Клас ViewDataServlet (див. рис.29 та рис.30) використовується для відображення даних менеджера паролі індивідуально кожного користувача після аутентифікації. Тут можна побачити усі сайти які використовує користувач, паролі та логіни для них. Завдяки цьому класу можна побачити збережені дані у таблицю даних для подальшого їх використання та змінювати їх. Цей клас створює таблицю у якій саме і відображається усі збережені дані.

IConstants.java

```
package sql;

public interface IConstants {
    String TABLE_USERS = "users";
    String TABLE_DATA = "data";
    String CONTENT_TYPE_TEXT_HTML = "text/html";
    String COLUMN_LINK = "link";
    String COLUMN_USER = "user";
    String COLUMN_USERNAME = "username";
    String COLUMN_LOGIN = "login";
    String COLUMN_PASSWORD = "password";
    String COLUMN_FIRSTNAME = "firstname";
    String COLUMN_LASTNAME = "lastname";
    String COLUMN_PHONE = "phone";
    String COLUMN_MAIL = "mail";
    String CHECKBOX = "checkbox";
    String ALPHABET = "abcdefghijklmnopqrstuvwxyz";
    String WORDS = "a\n" +
            "a&p\n" +
            "a's\n" +
            "aa\n" +
            "aaa\n" +
            "aaaa\n" +
            "aaron\n" +
            "ab\n" +
            "aba\n" +
            "ababa\n" +
            "aback\n" +
            "abase\n" +
            "abash\n" +
            "abate\n" +
            "abbas\n" +
            "abbe\n" +
```

Рис. 31 Інтерфейс IConstants

В інтерфейсі IConstants (див. рис. 31) зберігається константи, такі як назви колонок таблиці зареєстрованих даних, назви таблиць баз даних, тип контенту, назви колонок таблиць збереження персональних даних менеджера паролів, які використовується для аутентифікації на інші сайти, константи для генерації паролів та їх шифрування, а також перелік усіх слів бібліотеки.

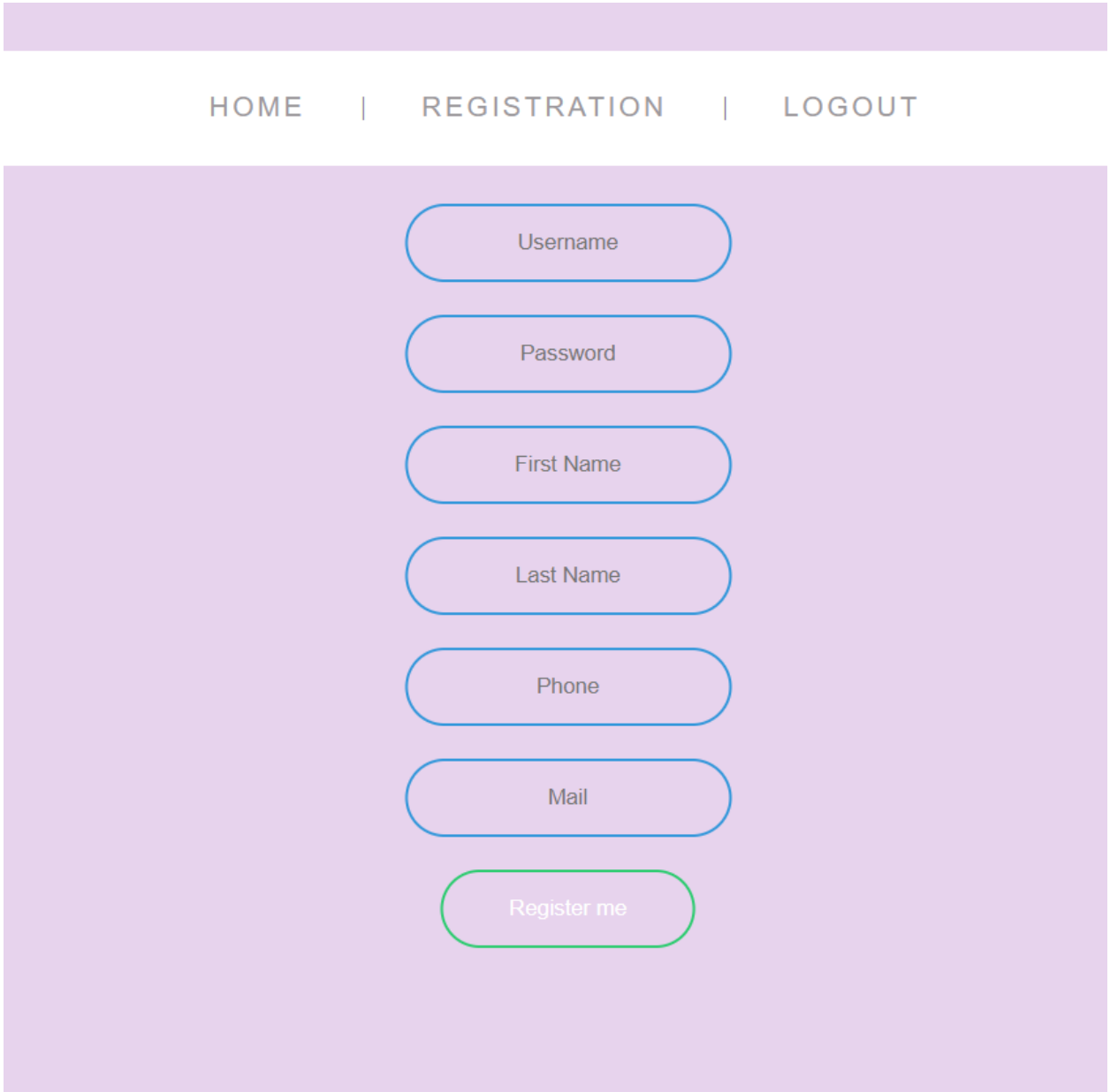


Рис. 32 Вікно реєстрації.

Перед використанням автоматизованої системи збереження персональних даних користувача користувачу потрібно пройти реєстрації для створення персонального кабінету (див.рис. 33).

Використовуючи вікно реєстрації користувач зберігає свої персональні дані до бази даних та після цього може користуватися автоматизованою системою збереження персональних даних користувача.

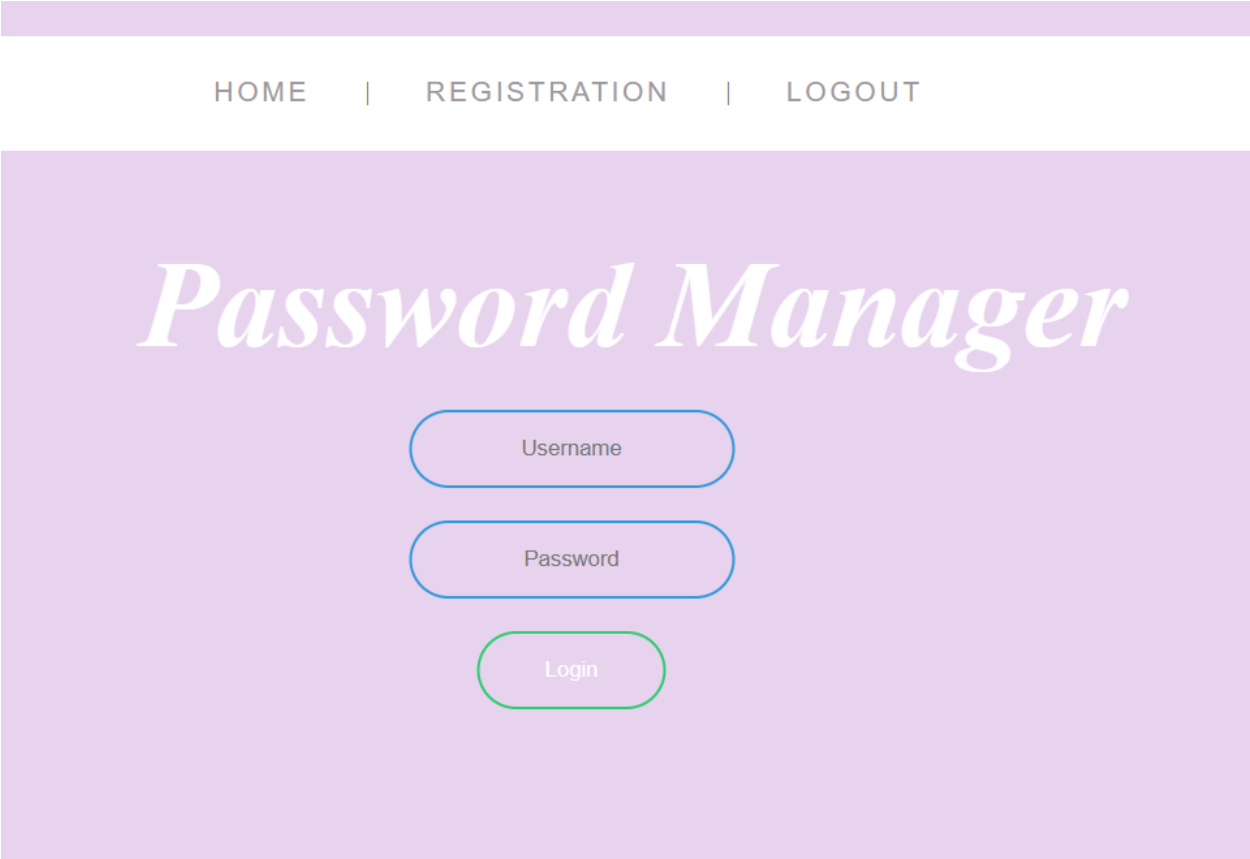


Рис. 33 Вікно аутентифікації.

Після проходження реєстрації користувачу треба пройти систему аутентифікації щоб потрапити саме у свій персональний кабінет, використовуючи науково новий алгоритм шифрування паролів користувач може не хвилюватися за безпечне зберігання своїх персональних даних та доступу до свого персонального кабінету.

Для успішної аутентифікації користувачу треба ввести свої логін та пароль, збережені у базу даних при реєстрації.

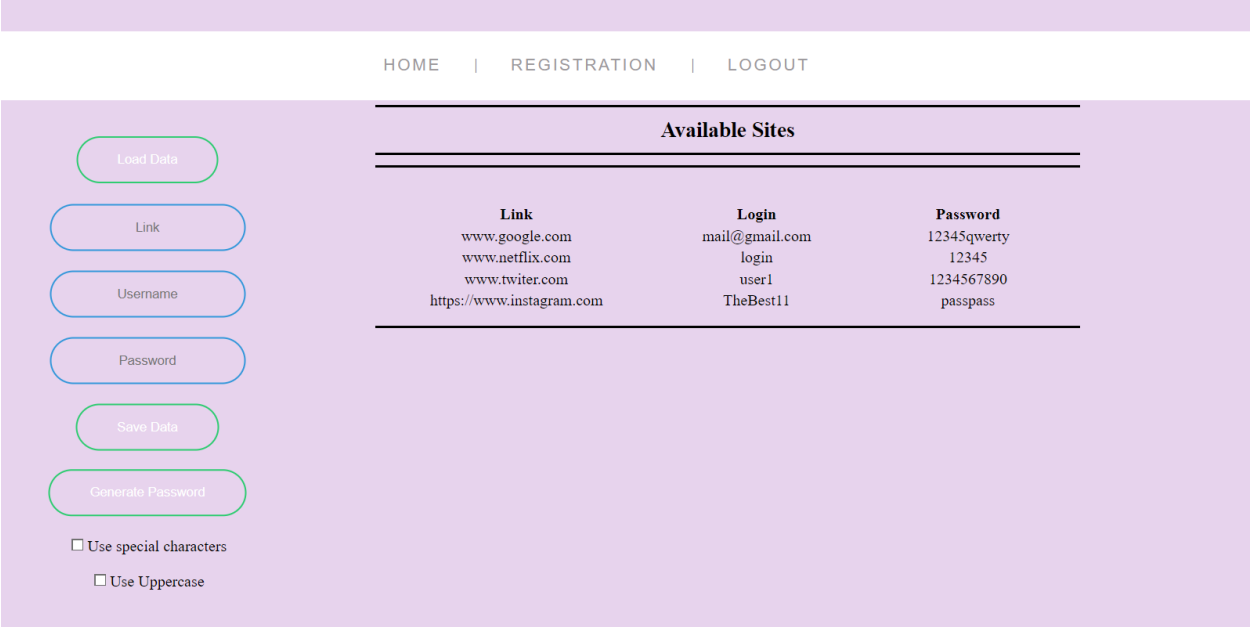


Рис. 34 Відображення збережених даних.

У персональному кабінеті (див. рис. 34) користувач може згенерувати пароль для аутентифікації на інші сайти, записати свою дані та зберегти їх у базу даних.

Користувач має можливість переглянуті персональні дані у свої персональному кабінеті.

При реєстрації користувача персональні дані зберігаються у базу даних. Пароль перед зберіганням проходить алгоритм шифрування.

При збереженні паролів у базу даних відомі сайти зазвичай використовуються хеш-шифрування паролів, щоб при крадіжці баз хакери не змогли одразу зайти в акаунти користувачів та вкрасти дані. Деякі додають у хеш сам пароль у порізаному вигляді для ускладнення злому хеша.

Алгоритм шифрування, який використовується у даному менеджері паролів доданий у підсистему реєстрації, оскільки пароль для аутентифікації

менеджер паролів є основним і при його крадіжці у хакерів буде доступ до всіх паролів. Такий спосіб може ускладнити завдання хакерам у десятки разів.

Користувач створює пароль зі спеціальною літерою наприкінці. Основна частина поля обрізається за виконується хеш – шифрування пароля, після чого кожна літера хеша змінюється на літеру алфавіту зі зміщенням у порівнянні з оригіналом на таку кількість літер, якою за рахунком в алфавіті є спеціальна літера в кінці пароля. Після цього використовується додаткове хешування.

Даний алгоритм показаний на рис.35.

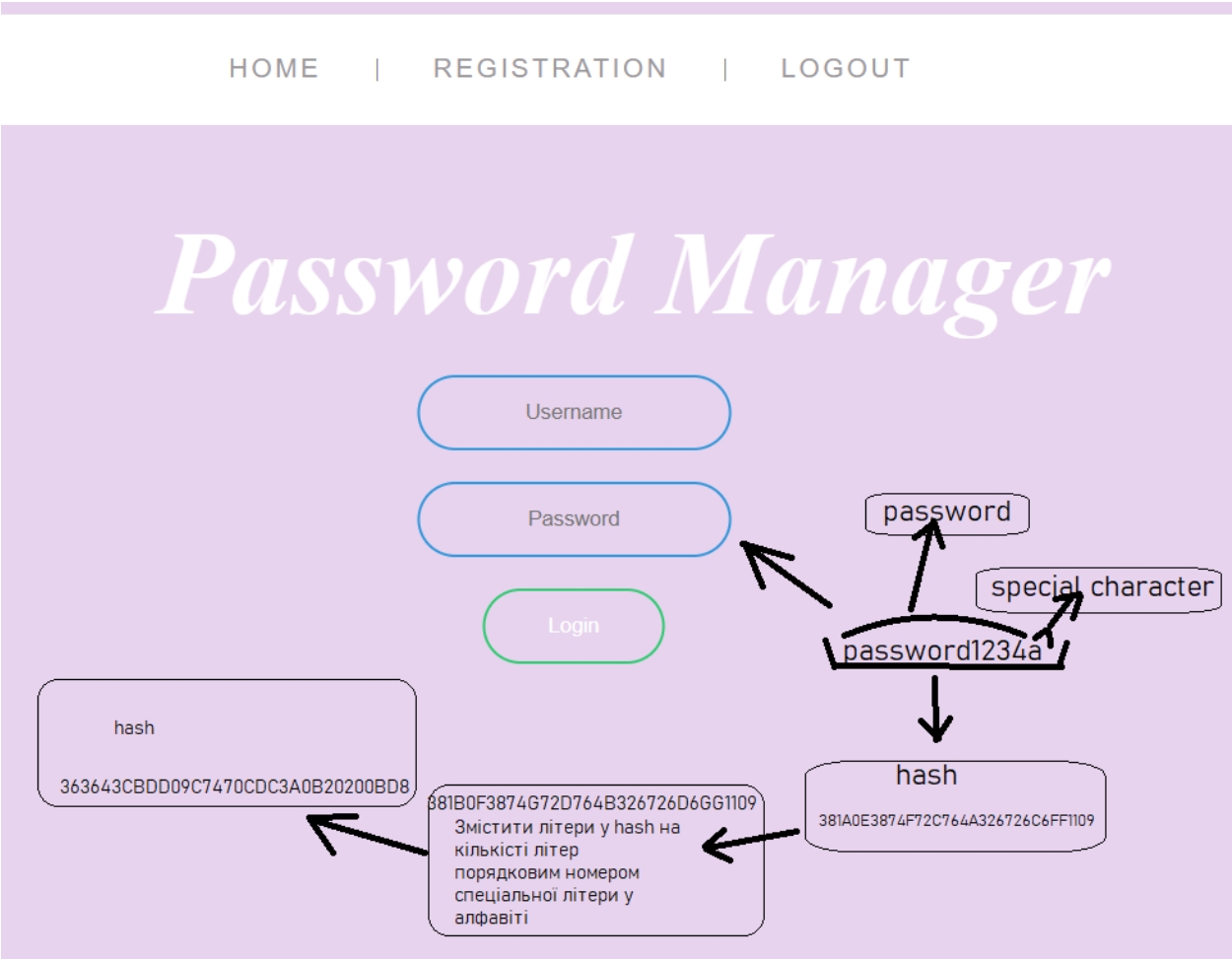


Рис.35 Алгоритм шифрування паролів.

Для реєстрації на різних сайтах користувач потребує у запам’ятовуванні персональних даних та генерації паролю. У вікні вводу та генерації даних (див. рис.36) користувач має можливість ввести та зберегти дані для аутентифікації

на сайти, а також згенерувати унікальний пароль використовуючи гнучку систему генерації паролів.

Під час реєстрації на сайтах деякі системи вимагають додати великі літери або спеціальні символи, крім безпечного генерування паролю, який ви зможете зберегти у базу даних, генератор паролів може виконати функцію додання необхідних символів до згенерованого паролю.

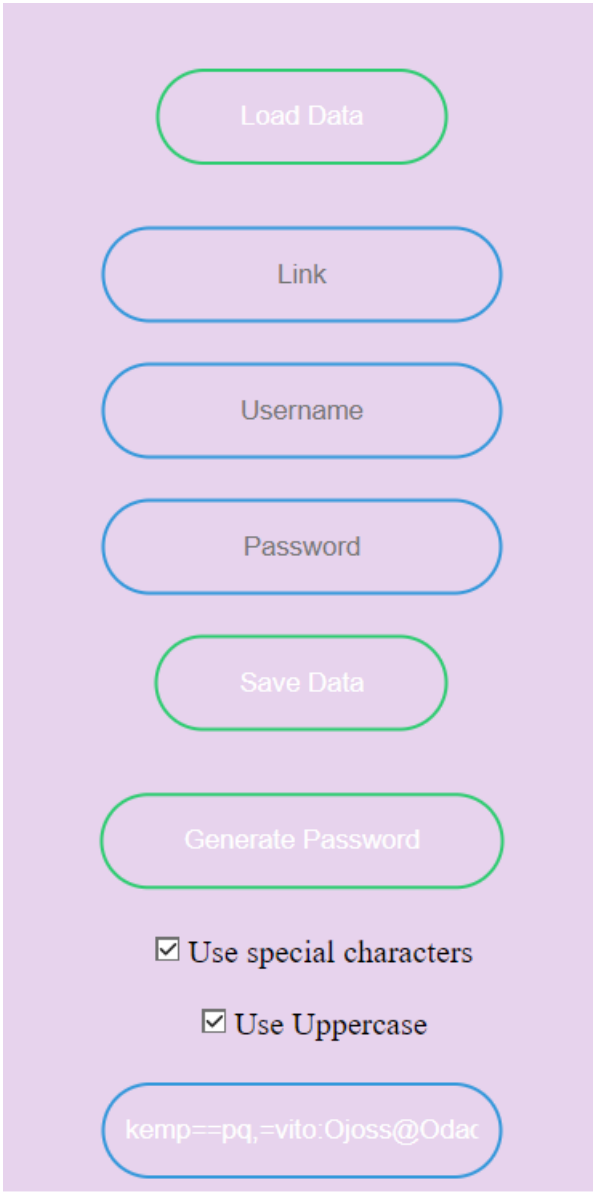


Рис. 36 Використання генерації паролів.

РОЗДІЛ 4
РОЗРОБКА СТАРТАП ПРОЕКТУ «АВТОМАТИЗОВАНА СИСТЕМА ЗБЕРЕЖЕННЯ ПЕРСОНАЛЬНИХ ДАНИХ КОРИСТУВАЧА»

4.1 Опис та технологічний аудит ідеї стартап-проекту

Основною ідеєю стартапу (див. табл.1) є вдосконалення існуючих менеджерів паролів для забезпечення підвищеної безпеки зберігання персональних даних користувачів, а також для запобігання заволодіння ними сторонніми особами.

Табл 1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Вдосконалення існуючих менеджерів паролів для підвищеної безпеки зберігання деяких персональних даних користувачів (логінів та паролів), а також для запобігання заволодіння ними	1. Для персонального застосування	Допомагає генерувати, зберігати та вводити складні паролі із зашифрованої б ази даних
	2. Для комерційного застосування	Для полегшення надання доступу до персональних кабінетів без передачі данних.

сторонніми особами.		

Не залежачи від напрямку використання продукту існує потреба у покращені безпеки менеджерів паролів, тому в основному работа з автоматизованою системою збереження персональних даних користувача не зміниться в залежності від напрямку застосування.

Розглядаючи популярні автоматизовані системи збереження персональних даних користувача можна зауважити вибірку недоліків, яка змінюється в залежності від компанії виробника (див. табл.2).

Табл 2. Визначення сильних, слабких та нейтральних характеристик
ідеї проекту.

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкурент1	Конкурент 2	Конкурент3			
1.	Шифрування	Шифрування паролів	Dashlane			Менша маштабність		Вдосконалення існуючого шифрування
2.	Генерація паролів	Різновид використання та генерації паролів	NordPass			Слабший інтерфейс		Генерація XChaCha20 нового покоління Багатофакторна автентифікація Потужна версія

								Сканер витоку даних
3.	Зберігання даниих	Обробка та збереження персональних даних до баз даних	RoboForm			Мала маштабність, та перевага конкурента на ринку використання		Самостійне розміщення або розміщення у хмарному сховищі. Вхід в один клік Легкий обмін паролями
4	Інтерфейс	Відображення отриманих даних	LastPass			Безліч розширень для браузера у конкурента. Легко налаштовується		Доступність та легке використання, Інтуітивний інтерфейс, Доступність різноманіття функціоналу

Розглядаючи недоліки відомих менеджерів паролів бачимо, що всі вони мають проблеми з безпечним шифруванням та збереженням даних, які можна покращити. Також різноманітність генерування/створення паролів для сайтів маленька та може бути покращена в залежності від потреб користувача.

Існуючі автоматизовані системи збереження персональних даних користувача виконують потрібні функції, вони являються доступними, розробка та вдосконалення відбувається записом коду системи (див. табл.3).

Табл. 3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
		Технологія 1 (технологія виготовлення товару, надання послуги)	Чи вони наявні, або ж необхідно їх розробити/доробити?	Чи вони доступні авторам проекту?
	Вдосконалення системи	Прописування коду	Потребує вдосконалення	Доступна

Попри доступність автоматизованих систем збереження персональних даних користувача та їх функціонал, деякі аспекти функціоналу, обробки та збереження інформації наданої системі потребує у покращені шляхом створення свого менеджеру паролів.

4.2 Аналіз ринкових можливостей запуску стартап-проекту

Ринок являється об’ємним та допускає збільшення гравців. Аналіз кількості гравців може змінюватись зважаючи на орієнтовність системи, від більш складного супротиву до менш складного (див. табл.4).

Табл. 4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг продаж, грн/ум.од	300 тис. ум. од.

3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Стандартизація оформлення та збереження даних
6	Середня норма рентабельності в галузі (або по ринку), %	50

Загальна оцінка ринку є сприятливою для роботи у даному направленні. Перспективи ринку зростають, обмежень немає. Лише присутня загальна специфікація для автоматизованих технічних систем. Складність конкурування навіть при наявності сильних гравців може зменшуватися при наявності гарного продукту.

Формуючи потреби ринку треба зауважити, що головною потребою у виникненні паролів є саме безпека персональних даних, менеджери паролів є системою, яка допомагає в збереженні цих даних (див. табл.5).

Табл. 5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Система з безпечнішим шифруванням	Активні користувачі гаджетами	Часте використання гаджетів	Безпечний, функціональний

Для забезпечення кращого збереження персональних даних користувачів перед збереженням даних відбувається їх шифрування, сам покращення технології шифрування у теперішньому світі технологій.

Теперішній ринок технологій передбачає високу конкуренцію та попит. Завдати шкоди прогресу виготовленого продукту, швидке вирішення проблем, які виникли може суттєво підвищити шанси на успіх (див. табл.6).

Табл. 6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Прогресс інших компаній	Покращення продукту та модифікації іновацій
2	Вихід системи з ладу	Робочий збій	Швидкий технічний ремонт
3	Викрадення системи	Публічне комерційне використання нової технології	Відстоування прав
4	Втрачання актуальності	Зменшення попиту	Вдосконалення існуючого алгоритму

Конкуруючи на ринку автоматизованих систем збереження персональних даних користувача треба мати гідний продукт та наукова новизна повинна привертати увагу, як якісне покращення. Набирати в команду треба таких спеціалістів, які б змогли оперативно та якісно відреагувати на збій.

Існуючі автоматизовані системи збереження персональних даних користувача виконують потрібні функції, вони являються доступними, розробка та вдосконалення відбувається записом коду системи (див. табл.7).

Табл. 7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Краще шифрування	Становлення стабільного алгоритму	Вдосконалення технології
2	Кращий генератор паролів	Розширені можливості для користувачів	Вдосконалення технології
3	Покращення інтерфейсу	Кращий доступ до потрібних розділів автоматизованої системи збереження персональних даних користувача	Покращення використання

Такі можливості дозволяють покращити захист та персональних даних користувачів та убезпечити від крадіжки. Оскільки всі існуючі менеджери паролів мають проблеми з безпечним шифруванням та збереженням даних, які можна покращити та зробити готовий продукт більш системним.

Вдосконалення автоматизованої системи збереження персональних даних користувача серед розповсюджених автоматизованих систем збереження персональних даних користувача може бути пов’язаним з передбаченням відкритих ризиків (див. табл.8).

Табл. 8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Олігополія	Декілька компаній виробляють конкуруючий продукт	Вдосконалення сильних сторін системи
Локальний	Початкове направлення більш вузьке	Розвиток для подальшого переходу в більш масштабні направлення
Внутрішньогалузева	Не передбачає вихід за галузь	Вдосконалене технічне направлення
Товарно-видова	Початкове направлення більш вузьке	Розвиток
Нецінова	Не пов’язане з ціновою політикою	Вдосконалення системи
Не марочна	Не пов’язано з напрямком	Немає

Забезпечення прогресу автоматизованої системи збереження персональних даних користувача серед розповсюджених автоматизованих систем збереження персональних даних користувача передбачає ріст попиту на продукт, який був розроблений при розгляді потрібних якостей.

Теперішній ринок технологій передбачає високу конкуренцію. Складніше забезпечити прогрес виготовленого продукту, але швидке

вирішення проблем, які виникли може суттєво підвищити шанси на успіх (див. табл.9).

Табл. 9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	NordPass, DashLane	LastPass, RoboForm	Стабільність	Технологічність	Кращі винаходи
Висновки:	Висока інтенсивність	Можливий потенційний вхід у ринок, але існують конкуренти	Ні	Ні	Технічні стандарти

Попри доступність автоматичних систем збереження персональних даних користувача та їх функціонал, деякі аспекти функціоналу, обробки та збереження інформації наданої системі потребує у покращені шляхом створення свого менеджеру паролів.

Табл. 10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)

1	Нове шифрування	Покращує захист паролів до десятків разів
2	Кращий інтерфейс	Полегшує користування
3	Кращий генератор паролів	Розширює вибір користувача у можливості генерації пароля автоматизованої системи збереження персональних даних користувача

Конкурентоспроможність [29] підприємства характеризує можливість та ефективність його адаптації до умов конкурентного середовища (див. табл.10).

До властивостей, якими характеризується конкурентоспроможність підприємства можна віднести: порівнюваність, просторовість, динамічність, предметність, атрибутивність, системність, об’єктивність. Дотримання названих властивостей є необхідною умовою проведення аналітичних досліджень, запорукою вирішення завдань та досягнення поставлених цілей підприємства.

Конкурентна сила підприємства визначається шляхом порівняння оцінок даного підприємства з основними конкурентами. Досить часто подібні оцінки здійснюються в ході реалізації бенчмаркінгового проекту, але в деяких випадках застосовується оцінка «абсолютної» конкурентної сили.

Табл. 11. Порівняльний аналіз сильних та слабких сторін «Автоматизованої система збереження персональних даних користувача»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з Автоматизованої система збереження

			персональних даних користувача						
			–3	–2	–1	0	+1	+2	+3
1	Нове шифрування	19		+					
2	Новий інтерфейс	10				+			
3	Зручність користування	15					+		
4.	Ціна	19			+				
5.	Надійність даних	20		+					

Оцінка абсолютної конкурентної сили [30] підприємства здійснюється за такою методикою. Спочатку вибирається перелік факторів, що визначають сильні і слабкі сторони діяльності підприємства. Потім проводиться порівняння оцінок цих факторів з оцінками найсильніших конкурентів, причому показник абсолютної конкурентної сили розраховується як алгебраїчна сума різниць між оцінками підприємства, що досліджується, і найвищими оцінками серед конкурентів (див. табл.11).

Найважливіше завдання SWOT-аналізу [31] — допомогти організації побачити та оцінити всі чинники, що впливають на прийняття рішень, а також визначити можливості розвитку (див. табл.12).

Метод використовує 4 ключові елементи: сильні та слабкі сторони, можливості та загрози. Задля зручності їх зображують у вигляді таблиці з 4-х стовпців.

Табл. 12. SWOT- аналіз стартап-проекту

Сильні сторони: Надійність, новизна, вдосконалення генерації паролів, новий алгоритм шифрування даних.	Слабкі сторони: нова компанія на ринку, відсутність досвіду, велика конкуренція.

Можливості: Розвиток та удосконалення шифрування, вихід на ринок та створення конкуренції для існуючих автоматизованих систем збереження персональних даних.	Загрози:розвиток конкурентів, вихід з ладу, складності становлення на ринку.

SWOT-аналіз допоміг зрозуміти, які чинники заважають розвитку компанії, зменшують її дохід та становлять загрозу в майбутньому. А побачивши можливості, які є на ринку, організація сфокусували свої сили та переваги для розширення власної ніші.

SWOT-аналіз — чудовий інструмент, але не панацея. Для того, щоб приймати збалансовані та розсудливі рішення, потрібне всебічне розуміння ситуації всередині компанії, знання ринку, економічних, політичних та соціальних тенденцій.

Альтернативи ринкового впровадження стартап-проекту поділяються на альтернатива (орієнтовний комплекс заходів) ринкової поведінки, ймовірність отримання ресурсів, строки реалізації (див. табл.13).

Табл. 13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Впровадження новизни	Велика (не ресурсно затратно)	1 місяць
2	Дослідження ринку	Велика (не ресурсно затратно)	3 тижні

3	Прописання стратегії розвитку компанії	Середня	1 місяць

Залежно від галузевої приналежності, ситуації на ринку [32] і сформованих особливостей управління організацією маркетингова стратегія може розроблятися на термін від 1 року до 25 років.

Найбільш часто в Україні в даний час використовується горизонт планування, рівний 1-3 років, але вже зараз можна зустріти підприємства, які розробляють стратегії на термін 5 і навіть 10 років.

4.3 Розроблення ринкової стратегії та маркетингової програми проекту

Маркетингова стратегія - це сукупність довгострокових рішень щодо способів задоволення потреб існуючих і потенційних клієнтів компанії за рахунок використання її внутрішніх ресурсів і зовнішніх можливостей.

Табл. 14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
	Активні молоді	80%	Піклуються за	Середня	Через рекламу

	люди, вік від 25 до 55 років		цілістність своїх даних		та соц мережі
	Компанії по роботі з клієнтськими даними	90%	Піклуються за цілістність даних клієнта	Середня	Через презентацію та пошук компаній
Які цільові групи обрано: Активні молоді люди, вік від 25 до 55 років					

Маркетингова стратегія компанії зазвичай закріплюється в документі, що має однойменну назву або назву "маркетингова політика". Маркетингова стратегія розробляється як складова частина загальної стратегії розвитку компанії (див. табл.14).

Вихідними даними для вибору базової стратегії служать як макро-економічні чинники [33], так і внутрішні можливості підприємства, що визначаються циклом його розвитку, а основне завдання, що вирішується при цьому, полягає у забезпеченні узгодженості між цілями і ресурсами (див. табл.15).

Табл. 15. Визначення базової стратегії розвитку

п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до	Базова стратегія розвитку*

			обраної альтернативи	
	Стратегія розширення ринку	Стратегія повного охоплення ринку	Недиференційований маркетинг	

Як результат спроб досягнення такого узгодження і виникають альтернативні варіанти. Кожному з них відповідає конкретний блок методів із забезпечення обраних напрямків діяльності, удосконалення організаційної структури і системи управління, зі стимулювання, підбору, ранжирування і підвищення кваліфікації кадрів, освоєння виробництва нових видів продукції, методів просування її на ринок і т.д.

Базова стратегія конкуренції [34]– це основа конкурентної поведінки підприємства на ринку і описує схему, що забезпечує переваги над конкурентами, є центральним моментом у стратегічній орієнтації підприємства. Ця обставина визначає необхідність детального обгрунтуван- ня даної процедури на підприємстві (див. табл.16).

Табл. 16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*

		конкурентів?		
	Ні	Буде шукати нових	Буде копіювати основі характеристики, але удосконалювати суть.	

Жодне підприємство не може досягти переваги над конкурентами за всіма комерційними характеристиками. Тому необхідний вибір пріо-ритетів і розробка стратегії, яка найбільше відповідає ринковій ситуації й найкращим чином використовує сильні сторони діяльності під-приємства. Ця обставина визначає необхідність детального обгрунту-вання даної процедури.

Загалом, під позиціонуванням [35] розуміють процес створення ринкового іміджу товару на основі виявлених мотивацій споживачів з метою формування в свідомості цільових споживачів унікального сприйняття товару, відмінного від товарів конкурентів (див. табл.17).

Табл. 17. Визначення стратегії позиціонування

п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
	Надійність даних	Розширення ринку	Новизна захисту даних	Надійність. Безпека. Доступність

	Простота використання	Розширення ринку	Новизна захисту даних	Надійність. Безпека. Доступність
	Доступність	Розширення ринку	Новизна захисту даних	Надійність. Безпека. Доступність

Реалізація стратегії [36] позиціонування супроводжується вибором конкретної стратегії конкурентної поведінки (пасивної чи агресивної) і розробкою відповідного комплексу маркетингу, що може супроводжуватися модифікацією самого товару.

Ключові переваги над конкурентами є вдосконалення існуючих менеджерів паролів для забезпечення підвищеної безпеки зберігання персональних даних користувачів, а також для запобігання заволодіння ними сторонніми особами (див. табл.18).

Табл. 18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Захист даних	Якісний захист.	Краще шифрування
2	Легке використання	Доступний та інтуітивний інтерфейс	Легкість та швидкість використання
3	Розширений вибір генерування паролів	Розширене генерування паролів для	Опціональність генерування використованих паролів

		автоматизованої системи збереження персональних даних користувача	

Ключові переваги перед конкурентами: доступність – менеджер парлів легко підключити в зручній версії, пробний період безкоштовний – клієнти зможуть спробувати продукт безкоштовно (буде доступні не всі можливості продукту).

На різних рівнях продукт буде виглядати інакше. На етапі розробки виявилась сутність продукту – зберегти та захистити дані (див. табл.19).

Табл. 19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Зберігання та захист даних		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1.Додаток 2. Десктоп версія	не матеріальна система, яка може зберігатися у хмарному сховищі	Технічна властивість Даного продукту має яскраво виражений вдосконалений механізм та робочу автоматизовану систему зберігання персональних даних користувачів.

		та бути скачаною з будь якого ПК.	
	Якість: стандарти технічної системи.		
	Вдосконалення використання автоматизованої системи збереження персональних даних користувача		
	Технічна система «Password manager»		
III. Товар із підкріпленням	До продажу		
	Після продажу		
За рахунок чого потенційний товар буде захищено від копіювання: за рахунок безпеки інформації системи, додаткового шифрування та правової захищеності проекту.			

Товар у реальному виконанні буде виглядати як повноцінний додаток або десктоп додаток, буде проходити тестування на якість та надійність. За рахунок безпеки інформації системи потенційний товар буде захищено від копіювання.

Нижню межу ціни [37] на програмний продукт визначають декілька факторів (див. табл.20):

- витрати на виробництво програмного продукту, якщо продукт створюється власними зусиллями, тобто без залучення чужих інструментальних засобів;
- упущена вигода, пов’язана з:
- відмовою від самостійних дій на ринку в разі передачі продукту посередникам для подальшого розповсюдження;

Табл. 20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	140 грн/місяць	180 грн/місяць	20 000 – 30 000 грн	200-300 грн/місяць

Якщо при визначенні верхньої межі ціни на звичайний товар головним фактором є попит, то верхня межа ціни на програмний продукт установлюється складніше, бо можуть використовуватись декілька показників.

Це пов’язане з такими факторами:

- виробник виходить з розрахунку оцінки приросту прибутку, який матиме користувач в результаті застосування даного продукту;

- якщо виробник пропонує не унікальний, а поширений на ринку товар, то ціна конкурента є верхньою межею ціни;

- для користувача не виключається можливість самостійного розроблення програмного продукту при залученні сторонніх фахівців. Граничною ціною для користувача виступатимуть власні витрати на розроблення та застосування.

Основний принцип ціноутворення полягає в тому, що ціна продукту не може бути вищою за ціну подібного продукту або продукту-замінника. Отже, індивідуальні витрати, виведені з регулювання і визначення ціни і ціноутворення, покладено тільки на покупця.

Канал збуту [38] — шлях товару від безпосереднього виробника до споживача, включаючи гуртових та роздрібних торговців. Цей шлях забезпечують відповідні організації, люди. Розрізняють канали збуту нульового рівня (виробник сам продає товар кінцевому споживачеві, прямі

канали розподілу), однорівневий (між виробником і покупцем існує роздрібний продавець) та дворівневий (між виробником і покупцем послідовно існують гуртовий та роздрібний продавці) канали.

Табл. 21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Пріорітетом виступає якість	Тотальний контроль якості продукту	Нульового рівня	Ексклюзивний розподіл
2	Новизна	Використання нових кращих технологій	Нульового рівня	Ексклюзивний розподіл
3	Швидкість	Покращення існуючих аспектів	Нульового рівня	Ексклюзивний розподіл

Багатоканальні системи збуту, при яких торгівля ведеться і через власну збутову мережу і через незалежних посередників, тобто, допускається застосування прямих і непрямих систем збуту (див. табл.21).

Ма́ркетингова комунікáція або просування [39] — один з чотирьох елементів маркетинг-міксу.

Який являє собою інтегрований набір засобів комунікації, що застосовується для передачі повідомлень від виробника або продавця продукції до його цільових ринків (див. табл.22).

Табл. 22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Пошук надійного менеджера паролів.	Чат підтримки, соцмережі, інтернет	Доступність. Пробний період безкоштовний.	Упевнити в надійності продукту	Ввічливе, інформативне
2	Пошук функціонального менеджера паролів.	Різновид доступних опцій використання	Практичність.	Показати функціонал	Інформативне, зацікавлююче

Маркетингові комунікації виконують цілий ряд функцій, що дозволяє споживачам бути поінформованими про товари і послуги, а виробникам про потреби споживачів. Сьогодні діяльність окремих людей, груп і організацій безпосередньо залежить від їх інформованості і здатності ефективно використовувати наявну інформацію.

4.4 Організація реалізації стартап-проекту

Ролі у команді:

–- Product owner [40];

–- Scrum master;

–- Команда розробки (Development team).

Product owner (PO) є сполучною ланкою між командою розробки та замовником.

Завдання PO – максимальне збільшення цінності продукту, що розробляється і роботи команди. Одним з основних інструментів PO є Product

Backlog. Product Backlog містить необхідні для виконання робочі завдання (такі

як Story, Bug, Task і ін.), відсортовані в порядку пріоритету (терміновості).

Scrum master (SM) є «службовцем лідером» (англ. Servant-leader).

Завдання Scrum Master – допомогти команді максимізувати її ефективність за

допомогою усунення перешкод, допомоги, навчанні та мотивації команді, допомоги PO.

Команда розробки (Development team, DT) складається з фахівців, які виробляють безпосередню роботу над виробленим продуктом. Згідно The Scrum Guide (документу, що є офіційним описом Scrum від його авторів), DT

повинні володіти такими якостями і характеристиками :

– бути самоорганізованими. Ніхто (включаючи SM і PO) не може

вказувати команді як їм перетворити Product Backlog в працюючий продукт;

– бути багатофункціональними, володіти всіма необхідними навичками для випуску працюючого продукту;

– за виконувану роботу відповідає вся команда, а не індивідуальні члени команди.

Рекомендований розмір команди – 7 (плюс-мінус 2 людини). Згідно

ідеологам Скрам, команди більшого розміру вимагають занадто великих

ресурсів на комунікації, в той час як команди меншого розміру підвищують

ризики (за рахунок можливої відсутності необхідних навичок) і зменшують

розмір роботи, який команда може виконати в одиницю часу.

Табл. 23 Календарний план-графік реалізації проекту та витрати

№	Зміст етапу	Вартість етапу
1	Розробка системи	20 тис. грн.
2	Підготовка команди для системної роботи	60 тис. грн.
3	Вихід на ринок	100 тис. грн.
4	Розвиток у сучасному ринку	100 тис. грн.

Розвиток системи та проекту передбачає чисту вартість проекту – Net Present Value рівною більше нуля (див. табл. 23). Та буде працювати на зростання Індекс прибутковості – Profitability Index.

Висновки

Після проведення аналізу можна побачити, що автоматизовані системи збереження персональних даних користувача виконують потрібні функції, вони являються доступними, розробка та вдосконалення відбувається записом коду системи.

Попри доступність автоматизованих систем збереження персональних даних користувача та їх функціонал, деякі аспекти функціоналу, обробки та збереження інформації наданої системі потребує у покращені шляхом створення свого менеджеру паролів.

Ринок являється об’ємним та допускає збільшення гравців. Аналіз кількості гравців може змінюватись зважаючи на орієнтовність системи, від більш складного супротиву до менш складного.

Формуючи потреби ринку треба зауважити, що головною потребою у виникненні паролів є саме безпека персональних даних, менеджери паролів є системою, яка допомогає в збереженні цих даних.

Для забезпечення кращого збереження персональних даних користувачів перед збереженням даних відбувається їх шифрування, сам покращення технології шифрування

у теперішньому світі технологій.

Конкуруючи на ринку автоматизованих систем збереження персональних даних користувача треба мати гідний продукт та наукова новизна повинна привертати увагу, як якісне покращення. Набирати в команду треба таких спеціалістів, які б змогли оперативно та якісно відреагувати на збій.

Існуючі автоматизовані системи збереження персональних даних користувача виконують потрібні функції, вони являються доступними, розробка та вдосконалення відбувається записом коду системи.

Такі можливості дозволяють покращити захист та персональних даних користувачів та убезпечити від крадіжки. Оскільки всі існуючі менеджери паролів мають проблеми з безпечним шифруванням та збереженням даних, які можна покращити та зробити готовий продукт більш системним.

Вдосконалення автоматизованої системи збереження персональних даних користувача серед розповсюджених автоматизованих систем

збереження персональних даних користувача може бути пов’язаним з передбаченням відкритих ризиків.

Забезпечення прогресу автоматизованої системи збереження персональних даних користувача серед розповсюджених автоматизованих систем збереження персональних даних користувача передбачає ріст попиту на продукт, який був розроблений при розгляді потрібних якостей.

Розглядаючи недоліки відомих менеджерів паролів бачимо, що всі вони мають проблеми з безпечним шифруванням та збереженням даних, які можна покращити. Також різноманітність генерування/створення паролів для сайтів маленька та може бути покращена в залежності від потреб користувача. Тому вироблення даної системи є актуальним та може вирішити загально-важливі проблеми на ринку застосування автоматизованої системи збереження персональних даних користувача.

ВИСНОВКИ

Після висвітлення проблеми зберігання персональних даних, проведення огляду сучасних методів, засобів збереження та захисту персональних даних.

Проведений огляд публікацій за темою дисертації, була розроблена архітектура менеджера паролів.

За якою були реалізовані основні підсистеми та модулі менеджера паролів відповідно до розробленої архітектури.

Та за результатами тестування працездатності розробленого менеджера паролів, були виявлені покращення у безпечності збереження персональних даних користувачів. Науково новий алгоритм шифрування ускладює злом персональних даних користувачів у десятки разів, а гнучкий інтерфейс та генератор паролів дозволяє задовільнити потреби користувачів дотрумуючись реєстраційних норм веб-сайтів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Перед використанням менеджерів паролів. https://stuartschechter.medium.com/before-you-use-a-password-manager-9f5949ccf168

2. Акт захисту даних https://www.bbc.co.uk/bitesize/guides/z6kj6sg/revision/4

3. Статичний пароль. https://www.sciencedirect.com/topics/computer-science/static-password

4. Що таке пароль? https://searchsecurity.techtarget.com/definition/password

5. Питання у сфері захисту персональних даних. https://ecpl.com.ua/news/top-10-pytan-u-sferi-zakhystu-personal-nykh-danykh/

6. Динамічні паролі. https://blog.bio-key.com/dynamic-password-enforcing-authentication-otp

7. BankID https://bankid.org.ua

8. ЕЦП. Урядовий портал. Єдиний веб-портал органів виконавчої влади України. https://www.kmu.gov.ua/usi-pitannya-po-e-poslugam/sho-tak-elektronnij-cifrovij-pidpis-ecp

9. Що таке Є-підпис? https://www.adobe.com/sign/electronic-signatures.html

10. Біометрична ідентифікація. https://sites.google.com/site/identifikaciataautentifikacia/ponatta-pro-identifikaciu/biometricna-identifikacia

11. Що таке біометричний паспорт? https://jurklee.ua/en/blog/chto-takoe-biometricheskij-pasport/

12. Двофакторна аутентифікація https://eset.ua/ua/blog/view/22/dvukhfaktornaya-autentifikatsiya-chto-eto-i-kak-rabotayet

13. Двофакторна автентифікація: хто її має і як її налаштувати. https://www.pcmag.com/how-to/two-factor-authentication-who-has-it-and-how-to-set-it-up

14. Найпопулярніші способи зберігання цифрових файлів. https://www.myamberlife.com/learn/top-ways-to-store-your-digital-files/

15. Контел Бредфорд. 5 загальних алгоритмів шифрування та незламність майбутнього. https://blog.storagecraft.com/5-common-encryption-algorithms/

16. Університет Карнегі-Меллон. Управління інформаційної безпеки. https://www.cmu.edu/iso/governance/guidance/password-managers.html

17. Ключ. https://uk.wikipedia.org/wiki/Ключ_(криптографія)

18. Програми для шифрування папок і файлів. https://uk.soringpcrepair.com/program-to-encrypt-folders-and-files/

19. Ле Гуан. VaultIME. Пенсильванський державний університет, Державний коледж, Пенсільванія, США, 2018. №14 с. 1 – 12.

20. Фахад Алодьяни. Менеджери паролів - це все про довіру та прозорість. Школа комп'ютерних наук та інформатики, Кардифський університет, Куінс-білдінг, Кардіфф, Великобританія, 2020. № 189 с. 189.
https://www.mdpi.com/1999-5903/12/11/189

21. Ени Янти, Теуку Юліар Ариф, Ризал Мунаді. Розробка та впровадження алгоритму 4DES. Куала шиїтський університет, 2017.
№3 с. 73 – 82.
http://www.jurnal.unsyiah.ac.id/JRE/article/view/3271

22. Горовеак Павло. Електронна ідентифікація, підпис та безпека інформаційних систем. Техніческий університет Кошице, 2002. №4 с. 239 – 242. https://actamont.tuke.sk/pdf/2002/n4/6horovcak.pdf

23. Вессель П. Л.,Стінкамп Л. П. Журнал інформаційного менеджементу. 2007. №2.
https://sajim.co.za/index.php/sajim/article/view/27

24. Тін-Йі Чанг, Мін-Шіянг Хванг, Чу-Чен Янг. Інститут електронного навчання, Національний університет освіти Чанхуа, Тайвань. 2017.

№ 8 с. 134. https://www.mdpi.com/2073-8994/9/8/134

25. Емерсон Р. М., Мішель С. В., Самуель Б. Л., Карлос Е. С., Габріела К. С., Ширлей А.Ш., Бруно Б. Л.. Журнал Інтернет-послуг та додатків. 2020. №1 с. 1 – 21. https://jisajournal.springeropen.com/articles/10.1186/s13174-020-00128-1

26. Що таке операційна система?

https://edu.gcfglobal.org/en/computerbasics/understanding-operating-systems/1/

27. 10 найкращих мов для розробки веб-додатків.

https://becominghuman.ai/top-10-best-web-application-development-languages-8204aad91bc4

28. Яка реалізація архітектури MVC в Java?

https://www.edureka.co/blog/mvc-architecture-in-java/

29. КОНКУРЕНТОСПРОМОЖНІСТЬ ПІДПРИЄМСТВА ТА ФАКТОРИ, ЩО НА НЕЇ ВПЛИВАЮТЬ http://www.economy.nayka.com.ua/?op=1&z=3390

30. Аналіз сильних і слабких сторін підприємства http://ukr.vipreshebnik.ru/strateg/4196-analiz-silnikh-i-slabkikh-storin-pidpriemstva.html

31. SWOT-аналіз: кому, коли й навіщо потрібен https://bakertilly.ua/news/id44448

32. Основи маркетингової стратегії https://leosvit.com/art/osnovy-marketyngovoyi-strategiyi

33. Менеджмент: розробка базової стратегії управління. https://osvita.ua/vnz/reports/management/13978/

34. БАЗОВА СТРАТЕГІЯ КОНКУРЕНЦІЇ ТА ЇЇ ВПЛИВ НА ФОРМУВАННЯ РИНКОВИХ ЦІН https://buklib.net/books/23179/

35. Тарасенко О.М. Солнцев С.О. АНАЛІЗ АКТУАЛЬНОСТІ СТРАТЕГІЇ ПОЗИЦІОНУВАННЯ https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjX3uiK89T0AhWMyosKHceaDCkQFnoECAMQAw&url=http%3A%2F%2Fev.fmm.kpi.ua%2Farticle%2Fview%2F80360%2F75999&usg=AOvVaw1BJ7zlc2kMLIyyOhWSeCuC

36. Стратегія позиціонування https://megamarketing.com.ua/stratehiia-pozytsionuvannia/

37. Установлення нижньої і верхньої меж ціни https://studentbooks.com.ua/content/view/816/44/1/8/

38. Канал збуту https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjPnNaM9dT0AhWFvYsKHZXSDvoQFnoECAUQAQ&url=https%3A%2F%2Fuk.wikipedia.org%2Fwiki%2F%25D0%259A%25D0%25B0%25D0%25BD%25D0%25B0%25D0%25BB_%25D0%25B7%25D0%25B1%25D1%2583%25D1%2582%25D1%2583&usg=AOvVaw3NNkFfq1KdBW8dmHOJ3mSc

39. Ма́ркетингова комунікáція https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwikr8ir9dT0AhVtw4sKHauQBT4QFnoECAgQAw&url=https%3A%2F%2Fuk.wikipedia.org%2Fwiki%2F%25D0%259C%25D0%25B0%25D1%2580%25D0%25BA%25D0%25B5%25D1%2582%25D0%25B8%25D0%25BD%25D0%25B3%25D0%25BE%25D0%25B2%25D0%25B0_%25D0%25BA%25D0%25BE%25D0%25BC%25D1%2583%25D0%25BD%25D1%2596%25D0%25BA%25D0%25B0%25D1%2586%25D1%2596%25D1%258F&usg=AOvVaw1pJ4yrkawhUbip4UKyKBdK

40. Гавриш, О. А., Бояринова К. О., Копішинська К. О. Розробка стартап-проектів. Конспект лекцій : навчальний посібник для студентів

спеціальностей 151 – «Автоматизація та комп’ютерно-інтегровані технології» та 152 – «Метрологія та інформаційно-вимірювальна техніка»; КПІ ім. Ігоря Сікорського. Електронні текстові данні (1 файл: 2,88 Мбайт). Київ : КПІ ім. Ігоря Сікорського, 2019. 188 с

41. Зубарський Д.О. Менеджери паролів / Погляд у майбутнє приладобудування : Збірник праць XIV Науково-практична конференція студентів, аспірантів та молодих вчених / 18-19 травня 2021 р. – К.:ПБФ, КПІ ім. Ігоря Сікорського. – 2021. - С. 38-41.