МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Марченко Олександр Олександрович

УДК 004.94

КВАЛІФІКАЦІЙНА РОБОТА

ІНФОРМАЦІЙНА СИСТЕМА ЗАБЕЗПЕЧЕННЯ РОБОТИ РЕКЛАМНОГО АГЕНТСТВА

126 «Інформаційні системи та технології»

Подається на здобуття освітнього ступеня бакалавр

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____
(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Дрейс Юрій Олександрович,
кандидат технічних наук. доцент

Житомир – 2023

Висновок кафедри _____

за результатами попереднього захисту: _____

Протокол засідання кафедри _____

№ _____ від «_____» _____ 20____ р.

Завідувач кафедри _____

_____ _____ _____

(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)

«_____» _____ 20____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)

(прізвище ,ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

_____ _____ _____

(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)

АНОТАЦІЯ

Марченко О.О. Інформаційна система забезпечення роботи рекламного агентства. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 126 – Інформаційні системи та технології. – Поліський національний університет, Житомир, 2023.

Дипломна робота присвячена розробці інформаційної системи забезпечення роботи рекламного агентства. Зважаючи на швидкий розвиток інформаційних технологій та широке використання глобальної мережі Інтернет особливої актуальності набуває потреба захисту інформації, що циркулює в інформаційній системі рекламного агентства та захисту процесів її обробки. Саме цим питанням приділена основна увага у роботі.

Запропоновано концептуальну та об’єктно-орієнтовану модель програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи рекламного агентства та шляхи її реалізації.

Результати дослідження можуть бути використані у роботі рекламного агентства, що сприятиме зростанню ефективності його роботи, забезпеченню конфіденційності даних клієнтів, надійному захисту інформації, що циркулює в інформаційній системі рекламного агентства та захисту процесів її обробки, і, як наслідок, зростанню конкурентоспроможності агентства на ринку послуг.

Ключові слова: інформаційна система, захист інформації, рекламне агентство, бази даних, програмний компонент, концептуальна модель, безпека.

SUMMARY

Marchenko O.O. Information system for ensuring the work of an advertising agency. - Qualification work on manuscript rights.

Qualification work for obtaining a bachelor's degree in specialty 126 - Information systems and technologies. – Polis National University, Zhytomyr, 2023.

The thesis is devoted to the development of an information system for ensuring the work of an advertising agency. Taking into account the rapid development of information technologies and the wide use of the global Internet, the need to protect information circulating in the information system of an advertising agency and to protect the processes of its processing becomes particularly relevant. These issues are the main focus of the work.

A conceptual and object-oriented model of the software component of the database protection subsystem of the web-oriented information system of the advertising agency and ways of its implementation are proposed.

The results of the research can be used in the work of the advertising agency, which will contribute to the growth of the efficiency of its work, ensuring the confidentiality of customer data, reliable protection of information circulating in the information system of the advertising agency and the protection of its processing processes, and, as a result, the growth of the agency's competitiveness in the service market .

Keywords: information system, information protection, advertising agency, databases, software component, conceptual model, security.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Основні умовні позначення та їх розшифрування:

IV – вектор ініціалізації

БД – база даних

ІС – інформаційна система

ОС – операційна система

ПЗ – програмне забезпечення

ВСТУП

Сьогодні на ринку послуг функціонує велика кількість рекламних агентств, що створює сильну конкуренцію між ними. Тому їх конкурентоспроможність залежить не тільки від широти асортименту послуг, що пропонуються, але й від ефективності роботи персоналу, швидкості обробки інформації. Сучасні рекламні агентства накопичують величезні обсяги даних. Необхідність використання великих обсягів інформації при вирішенні того чи іншого завдання спонукає до створення інформаційних систем для даного виду діяльності. Створення інформаційної системи забезпечення роботи рекламного агентства сприятиме оптимізації робочого процесу менеджерів агентства, зростанню ефективності роботи агентства та його конкурентоспроможності на ринку.

Водночас з розвитком інформаційних технологій та використанням глобальної мережі Інтернет особливої актуальності набуває потреба захисту інформації, що циркулює в інформаційній системі рекламного агентства та захисту процесів її обробки. Адже комп'ютерні системи і мережі зазнають тисяч різних атак як ззовні, так і зсередини. Для належного захисту інформації від атак, що здійснюють зловмисники, використовуються різні методи, способи та засоби, зокрема, й програмні. Одним із способів надійної передачі інформації незахищеними каналами зв'язку – є використання криптографічних засобів захисту інформації, тобто шифрування даних за допомогою симетричних та асиметричних алгоритмів шифрування. В даному випадку розглядається шифрування за допомогою симетричного блокового алгоритму AES-128 та надійна передача даних до БД Web-орієнтованої інформаційної системи.

Тому одним з основних завдань підсистеми захисту бази даних Web-орієнтованої інформаційної системи – створення та функціонування процесів надійного та якісного шифрування даних, а також безпечної їх передачі на відповідні БД Web-орієнтованої інформаційної системи. Таким чином,

реалізування цієї функції програмним компонентом підсистеми захисту бази даних Web-орієнтованої інформаційної системи є актуальним.

Об’єктом дослідження є інформаційна система забезпечення роботи рекламного агентства. Предметом дослідження є розробка підсистеми захисту бази даних web-орієнтованої інформаційної системи забезпечення роботи рекламного агентства.

Результати дослідження були апробовані на всеукраїнській та міжнародній конференціях:

Всеукраїнської науково-практичної конференція здобувачів вищої освіти і молодих вчених «Інформаційні технології та моделювання систем», 30 березня 2023 р., Житомир. URL: https://drive.google.com/file/d/1qBxHc7bTla--u6FbuPtHKucXxggF18ro/view?usp=share_link

VIII Міжнародна науково-практична конференція SCIENCE AND TECHNOLOGY: PROBLEMS, PROSPECTS AND INNOVATIONS, 11-13.05.2023, Осака, Японія.

РОЗДІЛ 1 АНАЛІЗ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ РОБОТИ РЕКЛАМНОГО АГЕНТСТВА ТА ПІДСИСТЕМИ ЇЇ ЗАХИСТУ

1.1 Аналіз предметної області дослідження інформаційної системи рекламного агентства та підсистеми її захисту

Інформаційна система являє собою організаційно-технічну систему, що об’єднує комп’ютерну систему (апаратне та програмне забезпечення), фізичне середовище (зовнішнє середовище та контрольована зона), персонал (керівництво, адміністратори і користувачі) і оброблювальну інформацію. Тоді як однією з її складових є інформаційна система [1 - 4].

Комп’ютерна система – інформаційно-технічний комплекс (сукупність апаратного та програмного забезпечення), який призначений для обробки, модифікації, вводу та виводу інформації [5 - 8].

У кожній інформаційній системі повинні бути забезпечені механізми (методи, засоби та способи) захисту інформації, що циркулює в системі, та процесів її обробки. Для здійснення належного захисту потрібно регулярно проводити пошук найбільш вразливих місць, які спроможні викликати порушення цілісності інформації, її доступності та конфіденційності. Цей пошук проводиться з метою постійного покращення та оновлення програмного забезпечення. Одним з основних методів захисту інформаційних ресурсів, для запобігання порушення конфіденційності, цілісності та доступності інформації – це використання криптографічних методів захисту та створених на їх основі програмного забезпечення [10].

Криптографія – це мистецтво, наука та технологія забезпечення секретності інформації. Воно вивчає способи та методи захисту інформації від змін і неавторизованого втручання при передаванні, обробленні та зберіганні [10]. Тобто, основна ціль криптографії – передача захищеного повідомлення від одного пункту до іншого, таким чином, щоб зміст повідомлення був відомий та зрозумілий лише відправнику та отримувачу повідомлення.

Криптографія передбачає два головні процеси – шифрування та розшифрування. Шифрування у системах обробки інформації означає криптографічне перетворення даних, яке має здійснюватися у певній (посимвольній) послідовності задля отримання шифрованого тексту [11 - 12]. Відповідно розшифрування постає як процес санкціонованого трансформації зашифрованих даних у такі, що придатні для читання [13 - 14]. Тобто, шифрування призначене для приховування інформації, що передається, від втручання сторонніх осіб, яким вона не призначається для ознайомлення чи модифікації. А розшифрування призначене для отримання користувачем відкритого тексту з шифртексту.

Також є можливість отримання відкритого тексту з шифртексту користувачем, для якого цей текст був непризначений, тобто зловмисником. Процес несанкціонованого отримання інформації з зашифрованих даних називається дешифруванням [13 - 14]. При цьому ключ дешифрування зазвичай невідомий.

Дія шифру визначається алгоритмами та ключем, який невідомий третій стороні і забезпечує секретність процесу і змісту передачі повідомлення. Важливим запобіжником від зламу алгоритмів шифрування є змінні ключі. Оскільки дуже часто для шифрування та дешифрування використовуються шифри без дотримання додаткових процедур (аутентифікації, перевірки цілісності).

Основними методами шифрування, на основі яких будуються алгоритми шифрування, є симетричне та асиметричне шифрування.

Суть симетричного шифрування полягає у використанні для шифрування і дешифрування одного і того ж секретного ключа. На основі даного методу розроблено ефективні (швидкі й надійні) методи шифрування.

Переваги симетричного шифрування: велика пропускна здатність шифрів; невеликий розмір ключів; можливість комбінування шифрів; можливість застосування шифрів у побудові різноманітних криптографічних

механізмів (псевдовипадкові генератори чисел, хеш-функції тощо). із симетричним ключем можна комбінувати для отримання сильніших шифрів.

Недоліком симетричного шифрування вважається необхідність тримати ключ у секреті на обох кінцях у випадку зв'язку між двома особами та необхідність частої зміни ключів.

Слабким місцем симетричного шифрування є необхідність передачі ключа, в процесі якої можливе його перехоплення сторонніми особами, а знання секретного ключа дасть можливість розшифрувати інформацію тому, кому вона не призначена.

На противагу симетричному, асиметричне шифрування передбачає наявність пари ключів: відкритого (англ. public key) та закритого (англ. private key). Перший ключ публічний, що означає доступ до нього усіх, кому потрібно зашифровувати інформацію. Другий ключ є приватним, тобто доступний доступним лише тому хто має право на розшифровування інформації. Поряд з цією перевагою, він має недолік порівняно із секретним ключем симетричного шифрування - це великий розмір [13 - 14].

Ще одним позитивним моментом асиметричного шифрування є те, що відкритий ключ не потребує процесу передачі і збереження її секретності, оскільки він є публічним. Водночас асиметричне шифрування забирає більше часу (у 3-4 рази) порівняно із симетричним та потребує більше обчислювальної потужності для забезпечення процесів шифрування та розшифрування інформації.

Задля подолання недоліків та використання переваг кожного з методів, їх часто поєднують. Зокрема для забезпечення ефективного шифрування з передаванням секретного ключа, використаного відправником, інформацію спочатку симетрично шифрують випадковим ключем, а потім цей ключ зашифровують відкритим асиметричним ключем одержувача, лише після цього повідомлення і ключ відправляють у мережу.

Програмний компонент підсистеми захисту бази даних Web-орієнтованої ін-формаційної системи буде невеликим некомерційним проектом, що

працювати в межах локальної мережі та буде здійснювати обмін інформацією в межах цієї мережі. Для даної задачі буде достаньо використання симетричного методу шифрування, на основі одного ключа шифрування та розшифрування.

Симетричні алгоритми шифрування поділяються на потокові та блочні. Перші послідовно обробляють текст повідомлення, а другі працюють з блоками фіксованого розміру. Найбільш поширеними алгоритмами симетричного шифрування – є потокові з довжиною ключа 128 та 256 біт. Одним із таких алгоритмів є алгоритм AES (Advanced Encryption Standard), який буде використовуватися в процесі розробки програмного компоненту.

1.2 Вимоги до програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи

Програмний компонент використовуватиметься для захисту бази даних Web-орієнтованої інформаційної системи, а саме шифрування інформаційних ресурсів на стороні клієнта та їх подальшу передачу до відповідних БД.

З огляду на це до програмного компоненту повинні пред’являтися ряд вимог до його структури та функціонування. Такими вимогами є:

1) програмний компонент повинен бути класифікований та побудована за наступними вимогами:
   - за способом організації – основі архітектури клієнт-сервер;
   - за сферою діяльності – програмний продукт захисту бази даних Web-орієнтованої інформаційної системи.

2) програмний компонент повинен бути побудований мовою програмування високого рівня на основі об’єктно-орієнтованого підходу;

3) програмний компонент повинен мати можливість роботи як в межах локальної мережі, так і підтримувати вихід в глобальну мережу Інтернет;

4) програмний компонент повинен мати зручний графічний інтерфейс для користувачів;

5) програмний компонент повинен підтримувати «локалізацію»;

6) програмний компонент повинен підтримуватися найбільш популярними операційними системи персональних комп’ютерів. Зокрема:

– Microsoft Windows (Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10);

– Mac OS X;

– Unix-подібні операційні системи (Linux, FreBSD).

Висновок до розділу 1

Визначено основні характеристики інформаційної системи. Проаналізовано процес захисту інформаційних систем за допомогою криптографічних методів. Серед них було вибрано алгоритм симетричного блокового шифрування AES-128, на основі якого буде створюватися програмний компонент.

За результатами аналізу розроблено вимоги до програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи. Зокрема, сформовано вимоги до його структури та функціонування.

РОЗДІЛ 2. РОЗРОБКА ПІДСИСТЕМИ ЗАХИСТУ БАЗИ ДАНИХ WEB-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ РОБОТИ РЕКЛАМНОГО АГЕНТСТВА

2.1 Концептуальна модель програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи

Для того, щоб створити функціональну модель, яка відображатиме структуру і функції системи, а також потоки інформації і матеріальних об'єктів, що зв’язують ці функції, можна застосувати методологію IDEF0 [15]. У методології IDEF0 модель постає як штучний об’єкт, як образ системи і її компонентів, кожний з яких пояснює певний функціонал системи, що пред’являється до системи та порядок його використання та взаємодії з іншими компонентами цієї ж системи. Для більш кращого розуміння всіх процесів, що відбуваються в системі, над кожним компонентом проводиться декомпозиція (розбивання на декілька менших функціональних частин) доти, доки ця процедура є доцільною.

Компоненти системи, або функції, представляються за допомогою прямокутників. Взаємодії між функціями відбуваються за допомогою інтерфейсів, представлених стрілками. Вхідні стрілки – це початкова (вхідна) інформація, що перетворюється функцією і на її виході, за допомогою вихідних стрілок, отримується результат перетворення (вихідна інформація). Далі, цей результат стає початковою інформацією, для наступної функції, що взаємодіє з попередньою.

Метою моделювання є створення підсистеми захисту бази даних Web-орієнтованої інформаційної системи забезпечення роботи рекламного агентства шляхом шифрування певних файлів та їх передача у відповідні БД в зашифрованому вигляді. А також, можливість розшифрування файлів та отримання вихідних (відкритих файлів).

Розпочнемо з побудови функціональної моделі (див. рис. 2.1). Відповідно до опису системи основною діяльністю є підсистема захисту, що складається

з підсистеми шифрування та підсистеми розшифрування файлів, що присутні в комп'ютерній системі.

Вхідними даними для підсистеми захисту (підсистеми шифрування та підсистеми розшифрування) є відкритий файл, що потрібно зашифрувати та розшифрувати. Даний файл вибирається користувачем системи за допомогою файлового менеджеру ОС. Вихідними даними для цієї підсистеми є зашифровані та розшифровані файли.

З огляду на це функціональна модель програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи матиме такий вигляд як на рис. 2.1.

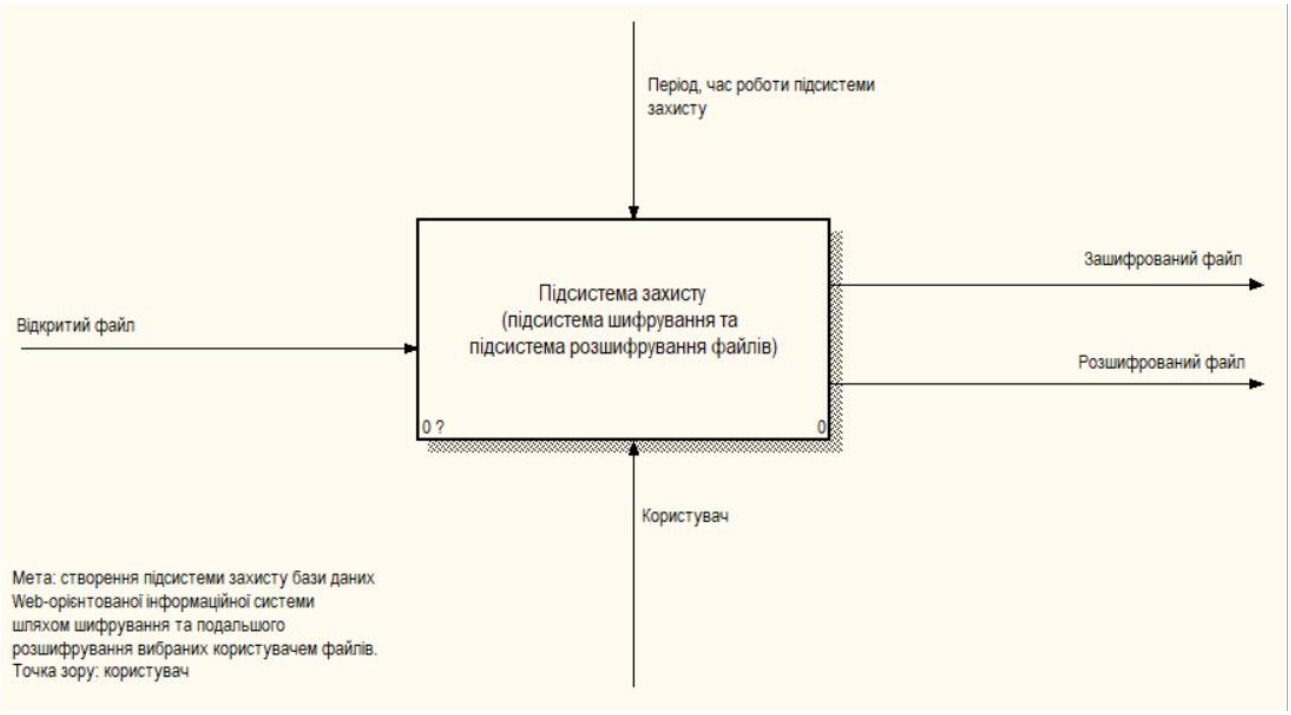

Рисунок 2.1 – Функціональна модель програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи

Проведемо декомпозицію функціональної моделі (див. рис. 2.2), описавши послідовність функцій даної підсистеми захисту: шифрування файлу та розшифрування файлу.

Таким чином, програмний компонент представляється двома підсистемами, кожна з яких реалізує окрему функцію. Отримані підсистеми можна розбивати на менші частини, до тих пір поки не досягнуть належної точності, яку вимагають від даних декомпозицій. Так як даний компонент не складний, то проводити подальшу декомпозицію немає потреби.

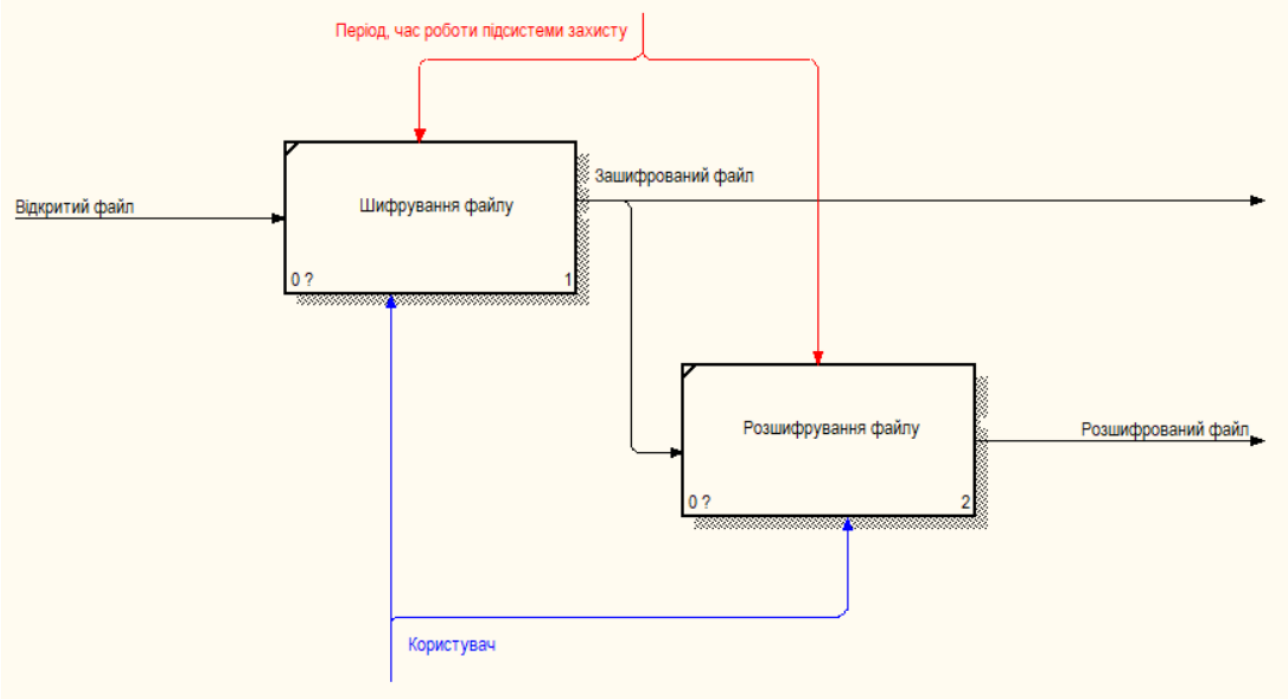

Рисунок 2.2 – Декомпозиція функціональної моделі програмного компоненту

Робота програмного компоненту починається з підсистеми «Шифрування файлу». В даній підсистемі на вхід подається відкритий файл. В результаті роботи даної підсистеми отримуємо зашифрований файл. Після цього робота підсистеми «Шифрування файлу» завершується та починається виконання функції «Розшифрування файлу».

У підсистему «Розшифрування файлу» на вхід подається зашифрований користувачем файл, який вибирається з відповідного списку раніше зашифрованих файлів. В результаті, на виході, отримуємо розшифрований файл. Після цього робота підсистеми «Розшифрування файлу» завершується.

Підсистеми «Шифрування файлу» та «Розшифрування файлу» працюють за допомогою симетричного блокового алгоритму AES-128

Для встановлення зв’язків між етапами підсистеми захисту бази даних Web-орієнтованої інформаційної системи обрано методологію IDEF3 як способу опису процесів з використанням структурованого методу. Якщо більшість технологій моделювання процесів мають жорсткі синтаксичні або семантичні обмеження, що робить опис неповних або нецілісність систем незручним, IDEF3 таких обмежень не має [14]. Також, IDEF3-моделювання органічно доповнює традиційне моделювання з використанням стандарту методології IDEF0 шляхом задання послідовності одиниць робіт означеної діяльності.

Далі, згідно з наступним етапом концептуального моделювання, в нотації IDEF3 [16], побудуємо процесну модель програмного компоненту (див. рис. 2.3). Початковим процесом функціонування програмного компоненту, що моделюється, є його запуск користувачем та увімкнення функції «Шифрування файлу», яка використовує алгоритм блокового шифрування AES-128.

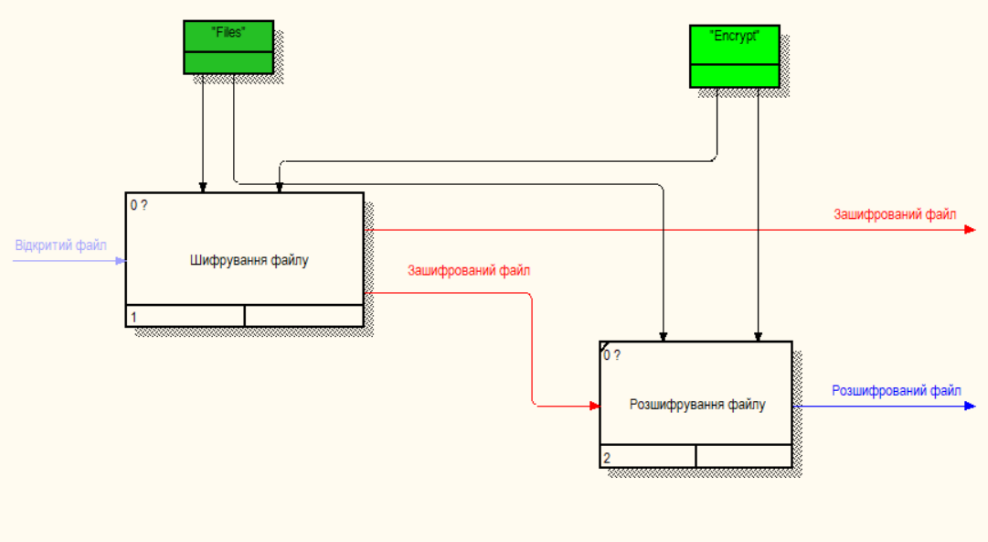

Рисунок 2.3 – Процесна модель програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи

Після закінчення процедури шифрування – відбувається запис ID зашифрованого файлу, назви файлу, зашифрованих байтів (зашифрованого

файлу) та дати зашифрування в БД «Files». А в БД «Encrypt» – записується ID зашифрованого файлу, псевдорандомний пароль, ключ шифрування та вектор ініціалізації IV.

Наступний процес функціонування програмного компоненту, що моделюється, – це увімкнення функції «Розшифрування файлу». Вона відбувається за рахунок вибору одного із зашифрованих файлів, зі списку доступних. В результаті виконання даного процесу – отримуємо розшифрований файл. Для розшифрування файлу потрібно використовувати дані, що записані в БД «Files» та «Encrypt».

Як результат, користувач має змогу захистити свій файл шляхом його шифрування та передачі останнього у відповідні БД. А також отримати вихідний файл з даних БД шляхом розшифрування.

Розглянемо діаграму декомпозиції даної процесної моделі програмного компоненту. Елементи даної діаграми пов’язані послідовністю потоків даних – їх отримання, обробкою та передачею. Тобто, головними частинами діаграми є сховища даних.

Для коректної роботи даної системи необхідно всього дві БД, а точніше файли – це «Files» і «Encrypt». В БД «Files» містяться ID зашифрованого файлу, назва файлу, зашифровані байтів файлу (зашифрований файл) та дата зашифрування. В БД «Encrypt» – ID зашифрованого файлу, псевдорандомний пароль, ключ шифрування та вектор ініціалізації IV (більш детально про ці дані описано в четвертому розділі). Дані БД потрібні для коректної роботи програмного компоненту (див. рис. 2.4).

Потік даних в програмному компоненті, що моделюється за допомогою діаграми нотації DFD (див. рис. 2.4), починається з передачі відкритого файлу від користувача до «Підсистеми шифрування». Дана підсистема створює псевдорандомний пароль для шифрування файлу. На основі даного паролю створюється ключ шифрування та вектор ініціалізації IV. Завдяки паролю, ключу та IV відбувається процедура шифрування файлу (на основі

симетричного блокового алгоритму AES-128). В результаті роботи даної системи отримуємо відкритий файл, який передається користувачу.

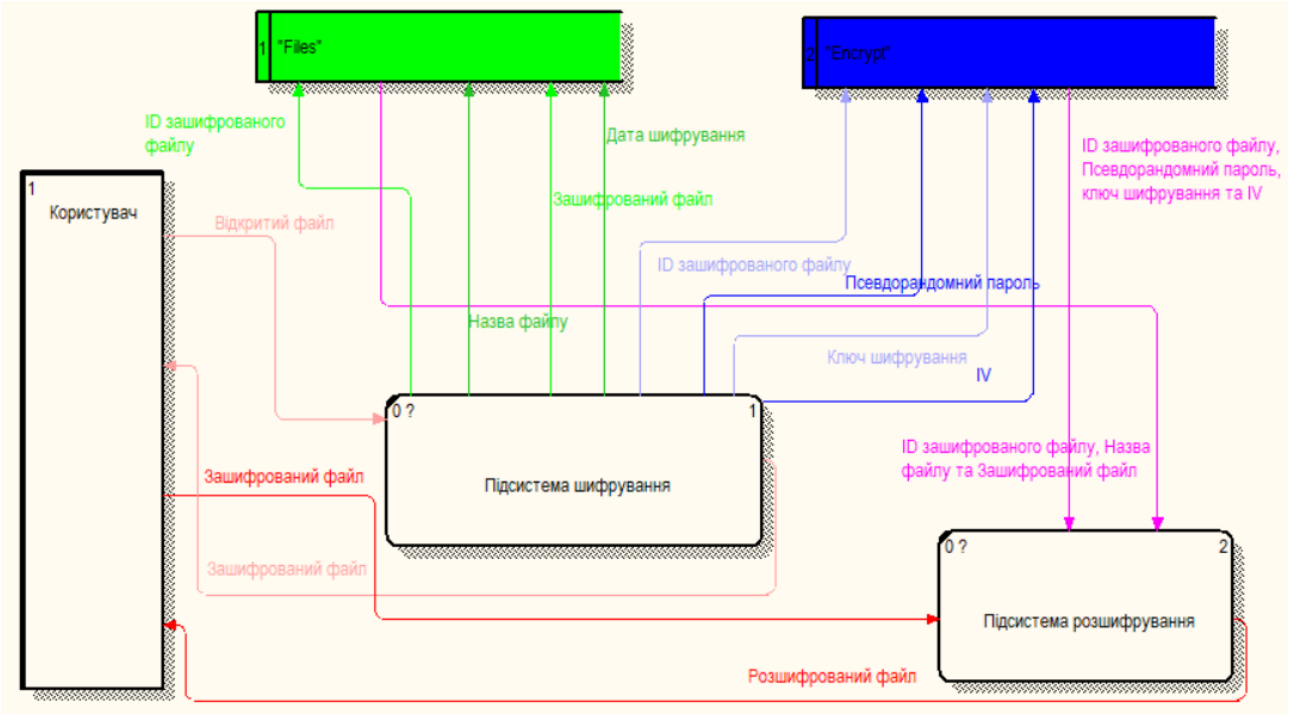

Рисунок 2.4 – Декомпозиція моделі потоків даних

Далі, відбувається робота «Підсистеми розшифрування». В дану підсистему на вхід подаються вибраний користувачем зашифрований файл (зі списку зашифрованих), а в результаті роботи цієї підсистеми отримуємо розшифрований (відкритий) файл, який система передає користувачу на ознайомлення.

2.2 Об’єктно-орієнтована модель програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи

Створення, опис та деталізація об’єктно-орієнтованої моделі будь-якого програмного застосунку починається зі специфікування вимог до нього. Тобто, на початку даного моделювання потрібно вияснити сферу застосування програмного застосунку, його функціональне призначення та функціональні вимоги. Дану проблему дозволяє вирішити використання моделей в нотації мови UML, а саме – діаграми варіантів використання [18].

Діаграма варіантів використання – діаграма, на якій зображуються варіанти використання проектованої системи, укладені в кордон (границю) суб'єкта, і зовнішні ектори, а також взаємовідношення між екторами і варіантами використання [18]. Тобто, дана діаграма описує функціональне призначення системи в найзагальнішому вигляді з точки зору всіх її користувачів і зацікавлених осіб. Дану діаграму можна уявити у вигляді системи типу «чорна скриня» – в неї подаються вхідні дані/параметри, які система обробляє та перетворює в певний результат (вихідні дані/параметри). Причому, її не цікавить технічна та фізична сторона реалізації виконання варіантів використання і взаємодія акторів з системою.

Вимоги до програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи специфіковано діаграмою варіантів використання (див. рис. 2.5) [18]. На даній діаграмі відбувається взаємодія між ектором системи та варіантами використання.

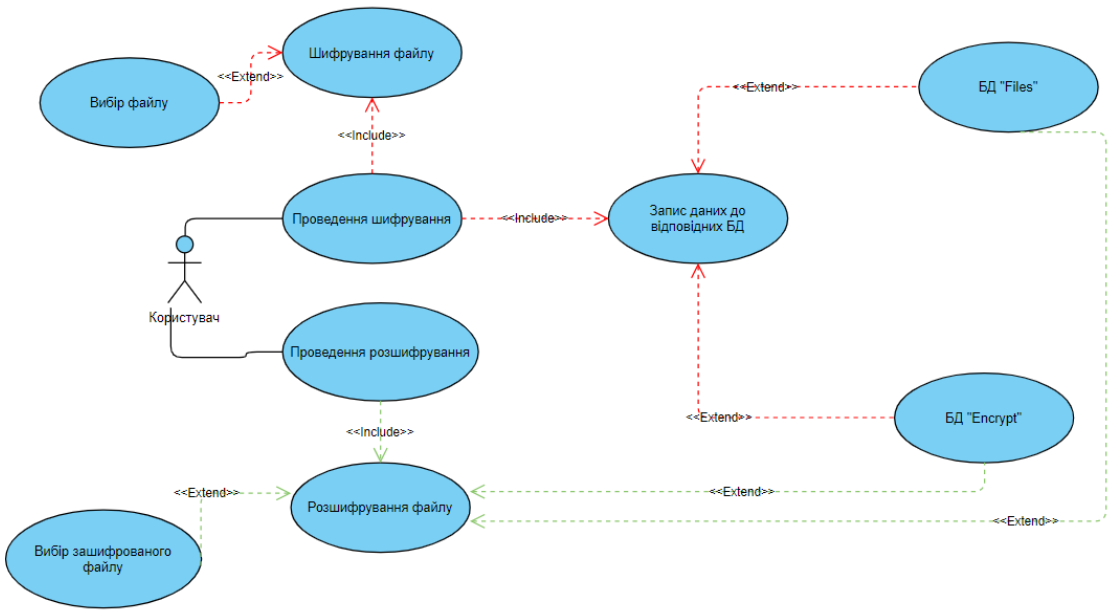

Рисунок 3.1 – Специфікування вимог до програмного компоненту діаграмою варіантів використання

Ектором в даній системі виступає користувач, тобто будь-яка фізична чи юридична особа, яка використовує програмний компонент з метою захисту

файлу шляхом його шифрування та подальшою відправкою зашифрованих даних у відповідні БД, що знаходять на локальному сервері.

Головними варіантами використання виступають «Проведення шифрування» та «Проведення розшифрування», робота яких починається безпосередньо після запуску програми.

У створенні логічної структури програмного компоненту слід враховувати особливість логічного уявлення, яке полягає в оперуванні поняттями, що мають віртуальний характер і не мають матеріальної сутності.

Логічна структура програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи рекламного агентства відображена діаграмою діяльності (див. рис. 2.6) [18].

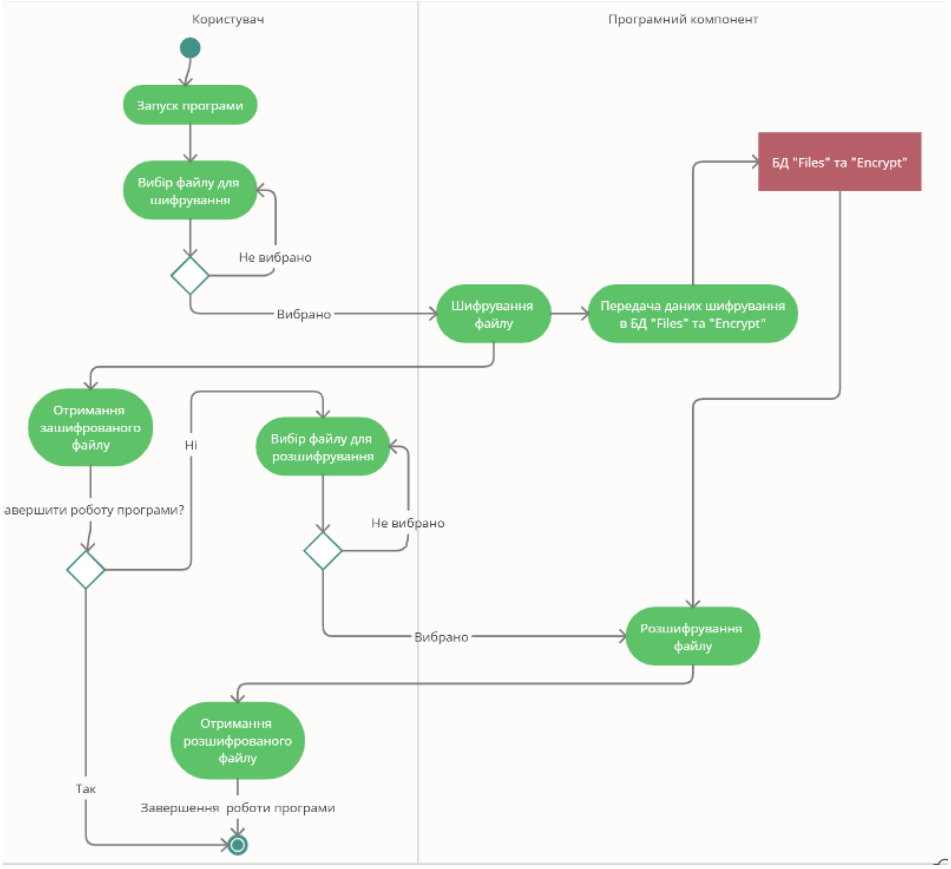

Рисунок 2.6 – Логічна структура програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи систему

На даній діаграмі початок діяльності починається з користувача, який своїми діями здійснює запуск програмного компоненту підсистеми захисту, а потім – «Вибір файлу шифрування» за допомогою файлового менеджера ОС. Далі користувач стикається з розгалуженням, а саме – з необхідністю вибрати файл. При виборі файлу – відбувається виконання процесу «Шифрування файлу» за допомогою симетричного блокового алгоритму AES-128. Інформація про успішне шифрування передається користувачу у вигляді списку усіх зашифрованих ним файлів, а дані шифрування передаються до БД «Files» та «Encrypt». За ці дії відповідають процеси «Отримання зашифрованого файлу» та «Передача даних шифрування в БД «Files» та «Encrypt» відповідно.

Наступними діями користавача, при продовжені роботи програми, є – вибір файлу для розшифрування Дані дії здійснюються за допомогою процесу «Вибір IP-адреси для сканування». Далі, на стороні програмного компоненту, здійснюється процедура «Розшифрування файлу» - за допомогою симетричного блокового алгоритму AES-128. Як результат, розшифрований файл передається користувачу, в процедуру «Отримання розшифрованого файлу». За допомогою останньої процедури відбувається відкриття розшифрованого файлу відповідним програмним забезпеченням.

Для деталізації програмних компонентів і взаємозв'язків між ними, а також артефактів, які їх реалізують, і вузлів, на яких вони розгортаються і виконуються, призначене фізичне моделювання в нотації мови UML. Для моделювання фізичної структури використовуються діаграми компонентів та діаграми розгортання.

За допомогою діаграми компонентів можна описати послідовність компонентів, що взаємодіють між собою в програмному засобі (див. рис. 2.7) [18]. Всі дії, що здійснює користувач в даній програмі, виконуються за допомогою функціоналу, який описаний в таких файлах як «EncryptWorker», «EncryptedFile», «DataManager».

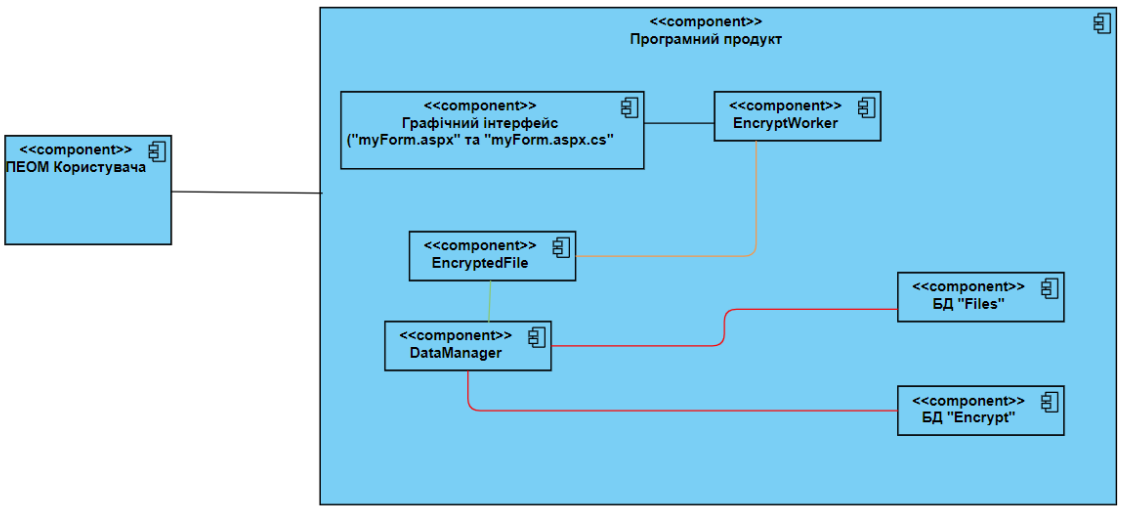

Рисунок 2.7 – Відображення діаграма компонентів програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи

Тобто, файли що відповідають за графічний інтерфейс, а саме – «myForm.aspx» та «myForm.aspx.cs» – взаємодіють з файлами «EncryptWorker», «EncryptedFile» та «DataManager». В свою чергу, останній файл забезпечує взаємодію з БД «Files» та «Encrypt».

Діаграма розгортання призначена для представлення загальної конфігурації або топології розподіленої програмної системи і містить зображення розміщення різних артефактів по окремих вузлах системи [18]. Також, діаграма розгортання показує наявність фізичних з'єднань або маршрутів для передачі інформації між апаратними та програмними пристроями, які забезпечують функціонування системи в різних режимах.

Фізичну структуру програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи, що відображена діаграмою розгортання, представлено на рис. 2.8 [18]. Дана діаграма пояснює фізичну та логічну взаємодію між артефактами (кодом) даного компоненту.

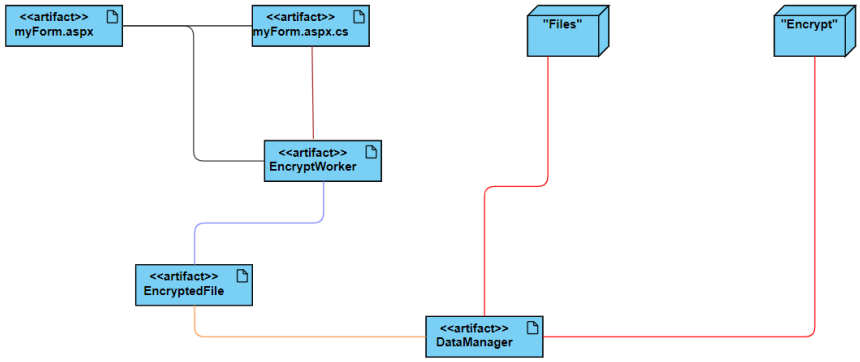

Рисунок 2.8 – Відображення фізичної структури програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи

Весь графічний інтерфейс відбувається в двох файлах – «myForm.aspx» та «myForm.aspx.cs». А у файлах «EncryptWorker», «EncryptedFile» та «DataManager» знаходиться програмний код, що відповідає за взаємодію функціональної частини програмного компоненту Також на даній діаграмі відображені дві БД: «Files» та «Encrypt».

Висновки до розділу 2

Створено концептуальну модель програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи рекламного агентства, використання якої дозволяє формалізувати його застосування щодо функцій, процесів та потоків даних, а також визначити варіанти використання програмного компоненту.

Процес моделювання розпочато з визначення контексту, тобто найбільш абстрактного рівня опису програмного компоненту в цілому, у контекст якого входили визначення мети та точки зору на концептуальну модель. На функціональному рівні концептуальної моделі відображено програмний компонент тестування на проникнення в комп’ютерну систему. Для кожної з функцій сформовано вхідні та вихідні дані. Тоді як на процесному рівні концептуальної моделі відображено послідовність одиниць роботи при роботі підсистеми захисту. На рівні потоків даних концептуальної моделі

відображено потоки даних між визначеними на функціональному рівні функціями з урахуванням інформаційних і зовнішні об’єктів.

Створено структуру програмного компоненту підсистеми захисту бази даних Web-орієнтованої інформаційної системи рекламного агентства на принципах формалізування його роботи щодо функцій, процесів та потоків даних, що дозволить забезпечити його функціональну придатність.

Зокрема, специфіковано вимоги до програмного продукту, які описані та промодельовані за допомогою діаграми варіантів використання. Також було створено логічну та фізичні структури програмного компоненту. Логічна структура описана та промодельована за допомогою діаграми діяльності. Тоді як фізична структура враховує: компоненти і зв’язки між ними (описано та промодельовано за допомогою діаграми компонентів), вузли і зв’язки між ними (описано та промодельовано за допомогою діаграми розгортання).

РОЗДІЛ 3. РЕАЛІЗАЦІЯ КОМПОНЕНТУ ПІДСИСТЕМИ ЗАХИСТУ БАЗИ ДАНИХ WEB-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Опис інтерфейсу та роботи з додатком

Створений програмний компонент підсистеми захисту бази даних web-орієнтованої інформаційної системи – це веб-застосунок (у вигляді веб-сайту) в якому клієнтом виступає веб-браузер, а сервером – локальний сервер IIS Express, що вбудований в Microsoft Visual Studio). Даний програмний компонент написаний мовою програмування C#, за допомогою технології ASP.NET, з використання середовища розробки Microsoft Visual Studio 2017 та СУБД Microsoft SQL Server 2012 на базі ОС Windows 10. Графічний вигляд даного додатку зображено на рис. 3.1.

Як видно з рис. 3.1, розроблений компонент складається з двох функціональних частин. Перша функціональна частина – це вибір файлу з файлової системи ОС Windows, шифрування даного файлу та передача назви (імені) зашифрованого файлу в поле зі списком зашифрованих файлів.

Рисунок 3.1 – Графічний вигляд програмного компоненту

Друга функціональна частина – це вибір зашифрованого файлу, з поля зі списком зашифрованих файлів, та його розшифрування з метою отримання вихідного відкритого тексту.

Робота даного програмного компоненту починається з натискання клавіші «Browse…», що розташована зверху посередині під пошуковим полем (поле з URL-адресою). Після натискання даної клавіші відбувається відкриття файлового менеджеру. Даний процес відображено на рис. 3.2.

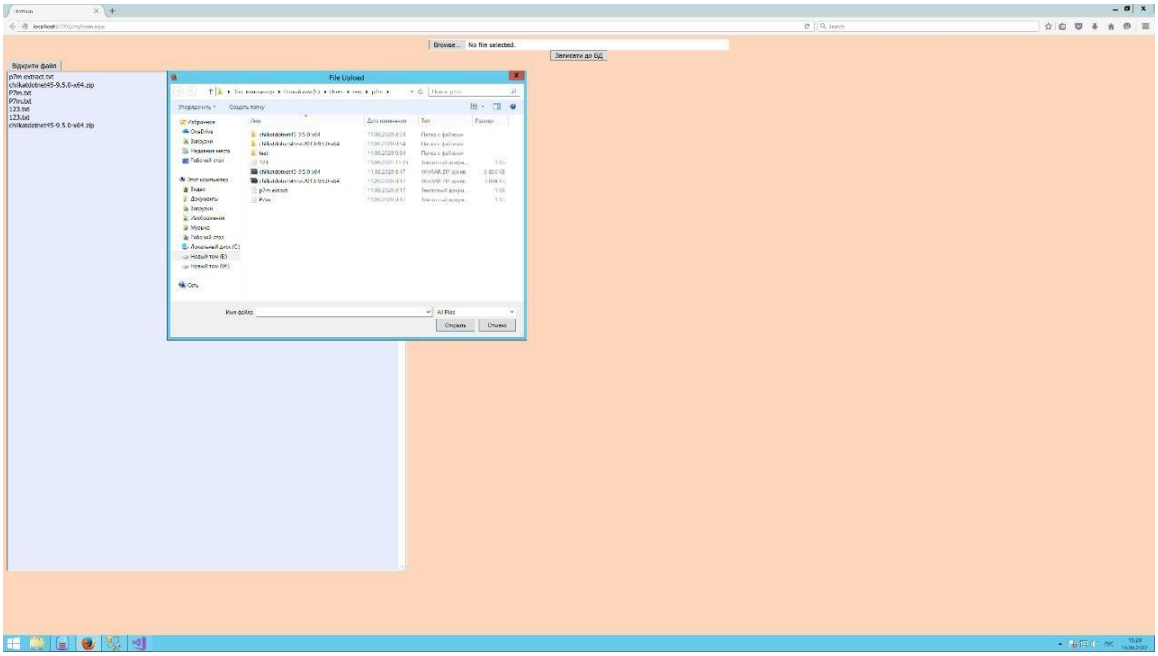

Рисунок 3.2 – Вибір файлу для подальшого шифрування

В файловому менеджері потрібно знайти потрібний користувачу файл (для подальшого шифрування) та натиснути клавішу «Відкрити», що знаходиться у файловому менеджері. Після даної процедури в текстовому полі, що знаходиться праворуч від клавіші «Browse…», буде відображено вибраний користувачем файл. Даний процес відображено на рис. 3.3.

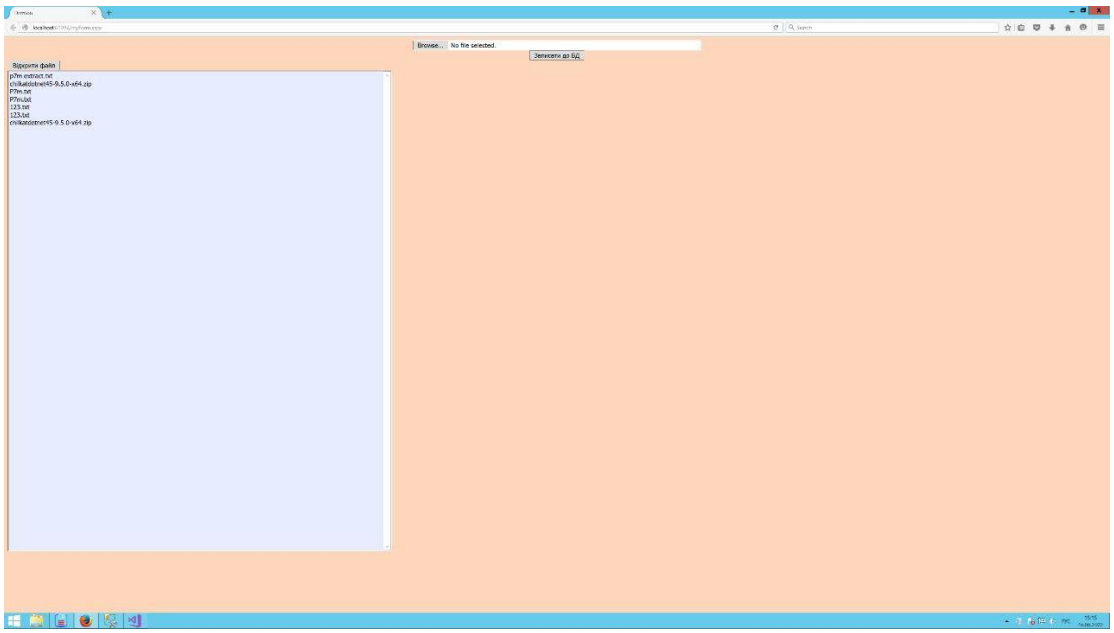

Рисунок 3.3 – Відображення вибраного користувачем файлу

Далі потрібно натиснути клавішу «Записати до БД» і при натисканні даної клавіші спрацює процедура шифрування файла. Тобто, буде виконано наступне:

1) Програмний компонент, за допомогою класу «EncryptWorker», генерує псевдорандомний пароль довжиною 14 символів;

2) Далі, з даного пароля формується ключ шифрування та вектор ініціалізації, за допомогою класу «EncryptWorker».;

3) Для подальшого шифрування використовується ключ шифрування та вектор ініціалізації IV. Тобто, за допомогою класу «EncryptWorker», відбувається шифрування – байти відкритого файлу (байтовий потік) буде зашифровано за допомогою алгоритму AES-128. Це симетричний алгоритм блокового шифрування з розміром блоку для шифруванн 128 біт та ключом шифрування 128 біт. Для цього з даного пароля формується ключ шифрування. Якщо коротко – то байти тексту (байтовий потік) розбивається на блоки розміром 128 біт та шифруються за допомогою ключа шифрування та IV.

4) Далі відбувається запис (збереження) зашифрованих байтів файлу, псевдорандомного паролю, ключа шифрування та IV в класі екземплярі класу «EncryptedFile».

5) Після цього, за допомогою класу «DataManager», відбувається робота с БД. Тобто, клас робить два SQL-запити до класу «EncryptedFile» з метою запису відповідних даних до відповідних БД. Перший запит – записує ID зашифрованого файлу, назву файлу, зашифровані байти (зашифрований файл) та дату зашифрування в БД «Files». Другий запит – записує ID зашифрованого файлу, псевдорандомний пароль, ключ шифрування та IV до БД «Encrypt»;

6) Як результат – вибраний файл зашифрований, дані шифрування записані в відповідні БД. А також відбувається відображення зашифрованого файлу в інтерфейсі програмного компоненту (в полі відображення списку зашифрованих файлів). За всі візуальні відображення відповідає клас «myForm». Даний процес, що описаний в пунктах 1-6, відображено на рис. 3.4.

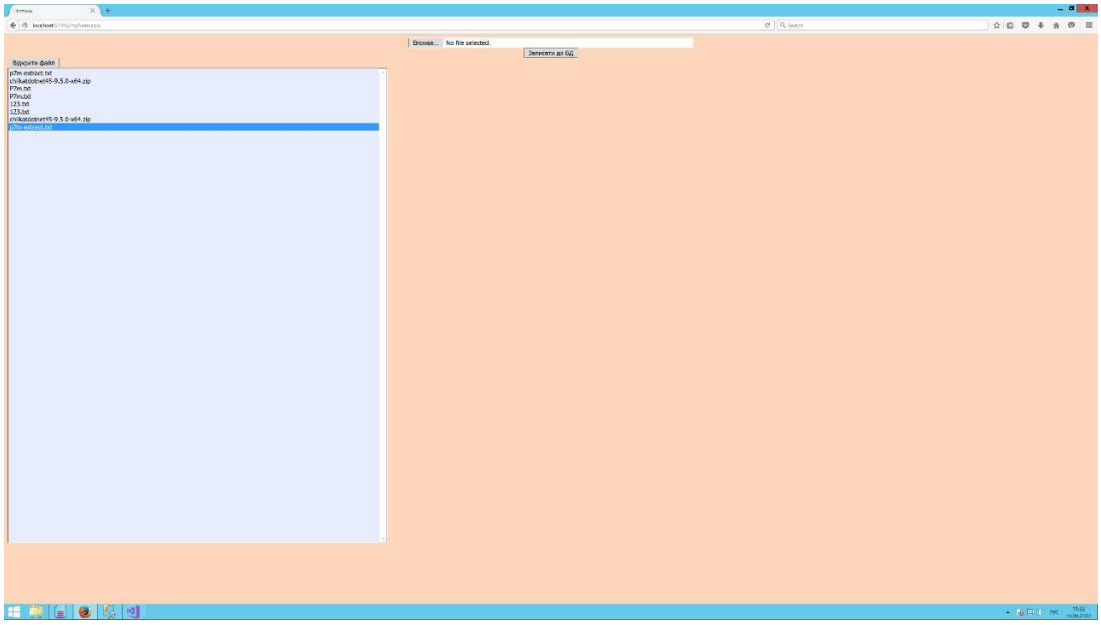

Рисунок 3.4 – Відображення зашифрованого файлу в полі відображення списку зашифрованих файлів

Далі розглянемо другу функціональна частину програмного компоненту – це вибір зашифрованого файлу з поля зі списком зашифрованих файлів, та

його розшифрування з метою отримання вихідного відкритого тексту. Для здійснення цієї процедури вибираємо зашифрований файл, який бажаємо розшифрувати та натискаємо клавішу «Відкрити файл». При натисканні даної клавіші спрацює процедура розшифрування файла. Таким чином, буде виконано наступне (що відображено на рис. 3.5.):

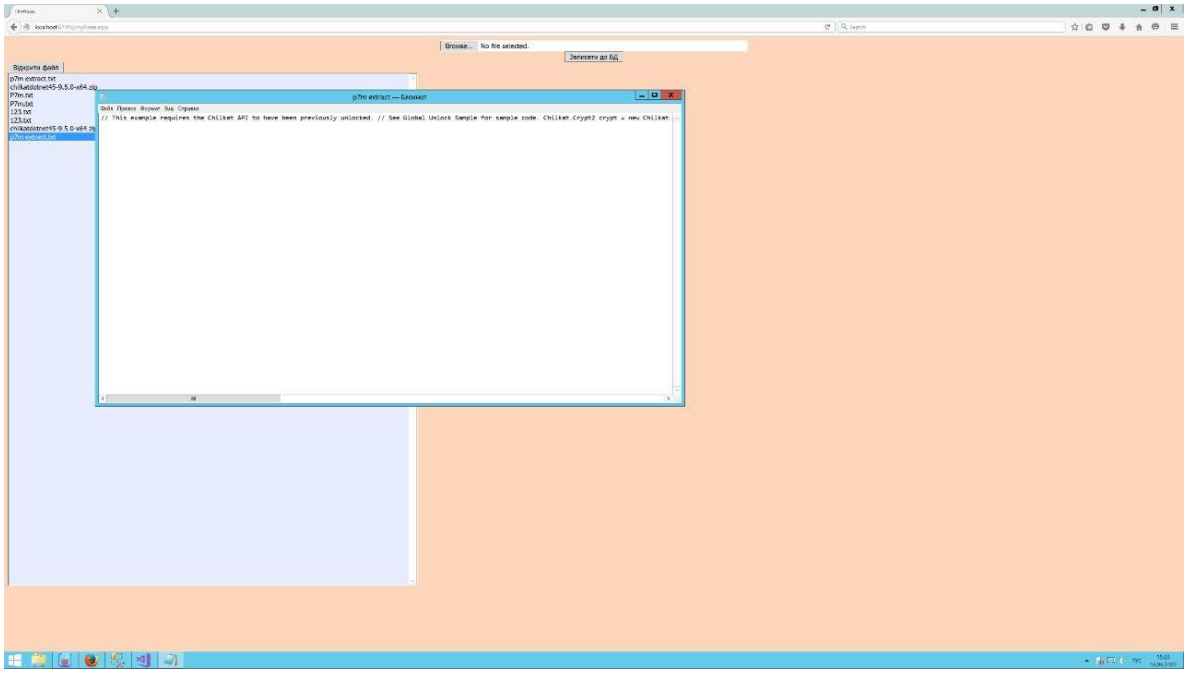

Рисунок 3.5 – Відображення та відкриття розшифрованого (відкритого вихідного) файлу

1) За допомогою класу «DataManager» здіснюється запит до двох БД. Перший запит – до БД «Files». За допомогою цього запиту здійснюється пошук файлу по його назві (імені) в даній БД. Далі, відбувається співставлення даної назви (імені) з зашифрованими байтами файлу і після чого зашифровані байти, у вигляді SQL-запиту, передаються в екземпляр класу «EncryptedFile». Другий запит – до БД «Encrypt». За допомогою цього SQL-запиту, по ID файлу, здійснюється пошук псевдорандомного паролю, ключа шифрування та IV в даній БД. Отримані дані записуються в екземпляр класу «EncryptedFile».

2) Далі, отримані та записані дані, з екземпляру класу «EncryptedFile» передаються в клас «EncryptWorker», а вже в останньому класі відбувається

процедура розшифрування файлу (за допомогою псевдорандомного паролю, ключа шифрування та IV);

3) Розшифрований файл відкривається відповідним програмним забезпеченням і відображається користувачу. Наприклад, текстовим редактором «Блокнот», якщо це файл з розширенням .txt. За всі візуальні відображення відповідає клас «myForm».

Завершення роботи програми здійснюється при закритті браузеру або даної вкладки з програмним компонентом (див. рис. 3.6).

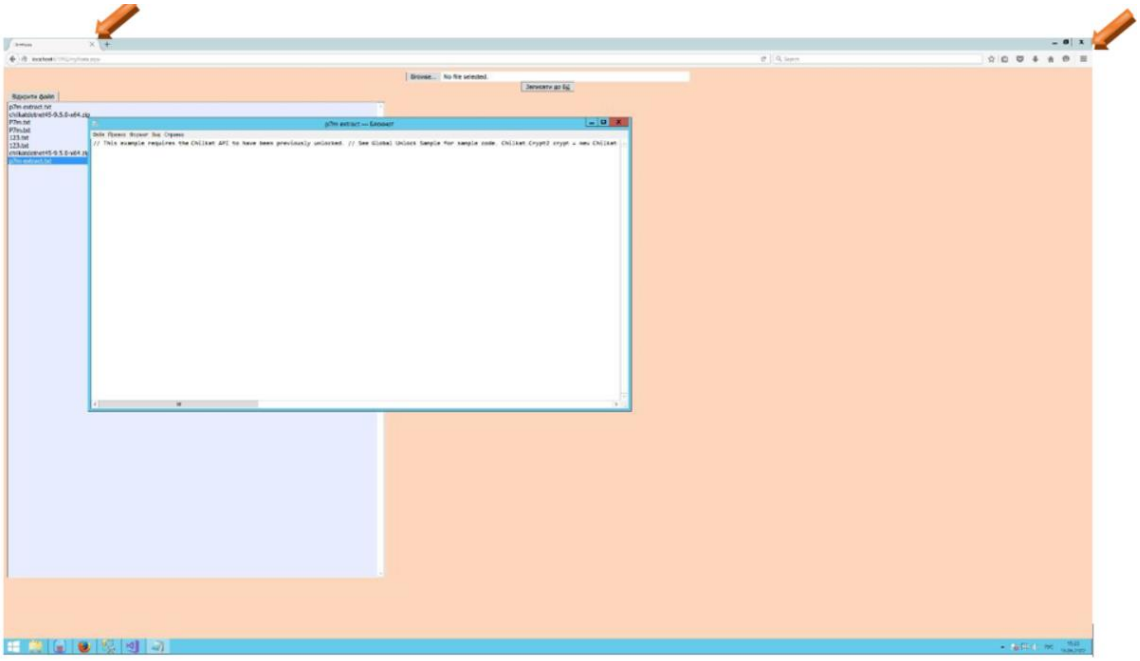

Рисунок 3.6 – Завершення роботи програмного компоненту

3.2 Реалізація програмного компоненту та його тестування

Для створення програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи було вибрано мову програмування об’єктно-орієнтованого типу C#. Дану мову програмування було вибрано виходячи з умови поставленого завдання та функціональних вимог до програмного компоненту [3, 18, 19].

C# (вимовляється Сі-шарп) — об'єктно-орієнтована мова програмування високого рівня з безпечною системою типізації для платформи .NET [19]. На вибір даної мови програмування також вплинуло те, що «мова C#» – це ключ до сучасного об’єктно-орієнтованого програмування. C# - це легка та зрозуміла мова, що створена для розробки високопродуктивного програмного забезпечення і надзвичайно популярна серед програмістів [19]. Головна перевага даної мови програмування в тому, що вона кросплатформна – тобто за її допомогою можливо створювати програмні продукти не тільки під платформу Windows, а й під Linux, Mac OS, Android, iOS. Дана мова забезпечує концептуальний фундамент, на який спираються інші мови програмування і багато сучасних засобів обробки даних. Мову використовують для розробки програмного забезпечення для бізнесу, написання як серверних та клієнтських програм, так і для розробки відеоігор та мобільних програм.

Також однією з головних причин вибору даної мови програмування вплинуло те, що вона містить засоби розробки програм для широкого спектру задач, від низькорівневих утиліт і драйверів до вельми складних програмних комплексів.

Перевагами, що вплинули на вибір мови C#, є [19]:

1. Доступність. Для C # існує величезна кількість навчальної літератури, що перекладена на різні мови світу. Мова має низький поріг входження та з легкістю підійде новачкам, які тільки починають опановувати процес розробки програмного забезпечення.
2. Простота в використанні. C# – це об'єктно-орієнтована, проста і водночас потужна мова програмування, яка дозволяє розробникам створювати багатофункціональні програми.
3. Сучасна мова програмування. C# – це мова компілюваного типу, тому вона має всі переваги таких мов. Також ця мова поєднує найкращі ідеї сучасних мов програмування – таких як Java, C++, Visual Basic.

4. Швидкість та якість розробки програмного забезпечення. Через велику різноманітність синтаксичних конструкцій та можливості працювати з платформою .Net, C# дозволяє швидше, ніж на будь-якій іншій мові, розробляти програмні рішення.

5. Кросплатформеність. Мову C# можливо використовувати для розробки програмного забезпечення для платформ Windows, Linux, Mac OS, Android, iOS.

6. Обчислювальна продуктивність. Мова забезпечує програмісту контроль над усіма елементами структури та етапами виконання програми. Надійність та елегантність. Елегантність C# досягається за рахунок великої різноманітності синтаксичних конструкцій. А велику надійності було досягнуто через роботу CLR машини, адже на відміну від інших компілятор CLR запускає розроблений додаток на віртуальному процесорі. Тому у разі виникнення будь-яких помилок, це ніяк не вплине на роботу інших програм у системі. Проте це також означає, що для запуску програми потрібен додатковий час. Відповідно програми написані мовою програмування C# надійніші, але менш швидкі (ніж такі ж програми написані на C++).

Також, окрім мови програмування, здійснювався вибір середовища розробки програми, операційної системи на базі якої вона розроблялася та технологій її створення. Також, при розробці даного програмного компоненту, потрібно було вибрати надійну та практичну СУБД.

Операційна система. Програмний компонент підсистеми захисту бази даних web-орієнтованої інформаційної системи створювався на базі ОС сімейства Windows, а саме – ОС Windows 10. Даний вибір пов’язаний з тим, що дане сімейство ОС частково створене за допомогою мови програмування C#, а також виходячи з того, що більшість середовищ створення програмних продуктів мовою C# розроблені лише для даного сімейства ОС.

Середовище розробки. Середовищем для розробки програмного компоненту тестування на проникнення в комп'ютерну систему було вибрано Microsoft Visual Studio 2017.

Технології програмування. Технологією програмування, що використовувалася в процесі створення програмного компоненту, була ASP.NET, що надає можливість розробляти вебзастосунки і вебсервіси для Microsoft Windows на мові C#.

СУБД. При розробці даного програмного компоненту було використано Microsoft SQL Server 2012 (коротко – MS SQL Server 2012) – потужна та надійна безкоштовна система управління даними, що забезпечує функціональне та надійне сховище даних для веб-сайтів та настільних додатків [20].

Для аналізу якості будь-якого програмного застосунку, а також створеного програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи, необхідно визначити характеристики та підхарактеристики, яким даний програмний продукт повинен відповідати [21].

Існують чотири основні характеристики та їх підхарактеристики:

1. **Функціональність** – це характеристика, що передбачає спроможність ПЗ виконувати в заданому середовищі чітку послідовність дій з метою задоволення споживчих властивостей, які були замовлені користувачем, та у відповідності з вимогами обробки і загальносистемних засобів. Атрибутами функціональності ПЗ постають:
   - **1.1. Функціональна повнота**: вказує на ступінь достатності основних функцій для вирішення поставлених завдань.
   - **1.2. Правильність**: вказує на рівень забезпечення досягнень правильних результатів.
   - **1.3. Захищеність**: вказує на можливість запобігання несанкціонованим доступам до програм і даних.
2. **Зручність застосування** – це комплекс атрибутів, що характеризується умовами взаємодії користувача з ПЗ. Атрибутами зручності застосування

ПЗ постають:

2.1. Зрозумілість: доступність для розпізнавання логічних концепцій ПЗ та умов їх застосування;

2.2. Легкість навчання: доступність для вивчення умов використання.

3. **Ефективність** – це атрибут, що характеризується ступенем відповідності витрачених ресурсів середовища та досягнутого рівня якості.
4. **Час реакції** – час відгуку, опрацювання, виконання функцій.

У табл. 3.1 наведено конфліктуючі між собою підхарактеристики з визначених. Позначка «+» – означає, що підхарактеристики конфліктують між собою, позначка «-» - підхарактеристики не конфліктують між собою.

Таблиця 3.1 – Результат попарного зіставлення підхарактеристик, за якими тестується програмний компонент

Характеристика		Функціональність			Зручність застосування		Ефектив-ність
		Функціональна повнота	Правильність	Захищеність	Зрозумілість	Легкість навчання	Час реакції
Функціональність	Функціональна повнота	-	-	-	-	-	0,3
	Правильність	+	-	-	-	-	0,5
	Захищеність	+	-	-	+	+	0,5
Зручність	Зрозумілість	+	+	+	-	-	0,9
	Легкість навчання	+	-	+	-	-	0,8

Ефективність	Час реакції	0,3	0,5	0,5	0,9	0,8	0,7

У таблиці 3.2. наведено метрики та вагові коефіцієнти для означених характеристик.

Таблиця 3.2 – Метрики та вагові коефіцієнти характеристик

Характеристика	Підхарактеристика	Метрика	Вага
Функціональність	Функціональна повнота	Число реалізованих функцій від загального числа заявлених функцій	0.8
	Правильність	Число результатів від загального числа очікуваних результатів	0.8
	Захищеність	Число захищеності програмного забезпечення	0.5
Зручність застосування	Зрозумілість	Число зрозумілості використання програмного забезпечення	1
	Легкість навчання	Число легкості навчання програмного забезпечення	0.9
Ефективність	Час реакції	Число часу відгуку програми на дії користувача	0,7

Також було визначено наступні характеристики експлуатаційної якості:

– **результативність** оцінює якість виконання програмою своїх функцій та отримання результату, який задовольнятиме користувача;

– **продуктивність** – співвідношення витрачених ресурсів та ефективності вирішення завдань;

– **задоволеність** – рівень відповідності отриманих результатів очікуваним..

Взаємозв’язок характеристик зовнішньої та експлуатаційної якості відображено на рис. 3.7

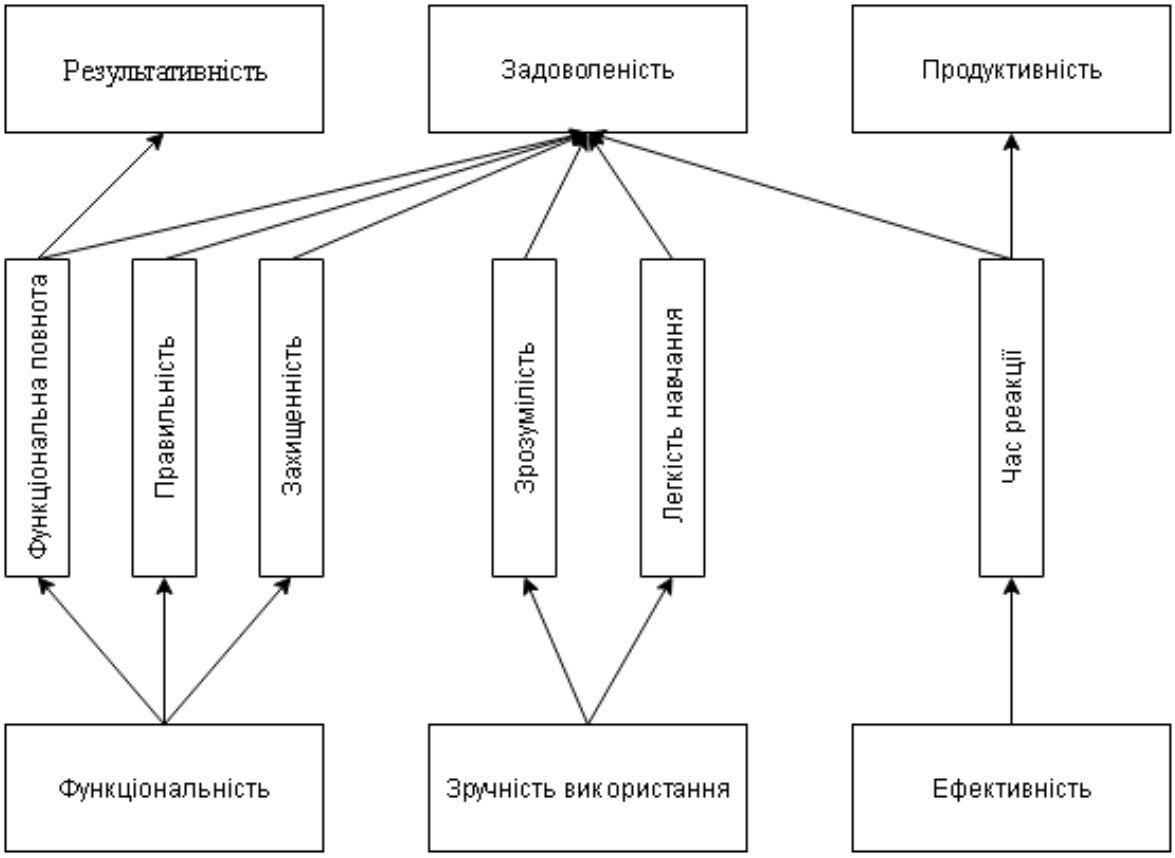

Рисунок 3.7 – Взаємозв’язок зовнішньої та експлуатаційної якості

Тестування програмного забезпечення передбачає перевірку відповідності заявлених до продукту вимог до реально реалізованої функціональності. Воно відбувається шляхом спостереження за роботою продукту в штучно створених ситуаціях та при обмеженому наборі тестів. Отже, тестування забезпечує контроль якості тих характеристик ПЗ, які проявляються в процесі його роботи [21]. Крім того, тестування надає наочне порівняння фактично існуючого ПЗ і очікуваного для виявлення відмінностей.

Для перевірки невідповідності між реальною поведінкою функцій, що були реалізовані в рамках ПЗ, і очікуваною поведінкою, визначеною вимогами до ПЗ використовується функціональне тестування.

Під час тестування програмного компоненту системи захисту інформації бази даних Web-орієнтованої інформаційної системи необхідно перевірити функціональну відповідність наступних функцій [21]:

– належну та правильну процедуру роботи підсистеми захисту бази даних web-орієнтованої інформаційної системи, а саме реалізація системи для шифрування та дешифрування;

– перевірка правильної роботи функції попереднього вибору файлу для шифрування та розшифрування;

– збереження інформації про результати шифрування файлу в БД «Files», де зберігаються байти зашифрованого файлу, а також запис пароля до зашифрованого файлу, ключа шифрування та вектора ініціалізації (IV) до БД «Encrypt».

Застосунок пройде тестування, якщо результати виконання функцій будуть відповідати очікуваним; а дані у сформованих файлах коректні та коректно записані у відповідні БД.

При тестуванні програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи було перевірено його функціональність. В табл. 3.3 наведено інформацію про тестування основних варіантів використання та їх результати зображені на рис. 3.3.

Таблиця 4.3 – Перевірка запису інформації

Мета тесту	Перевірка можливості належного шифрування та розшифрування файлів та передача даних ширфування у відповідні БД
Початковий стан	Відкрита програма підсистеми захисту бази даних web-орієнтованої інформаційної системи
Вхідні дані	Відкритий файл, над яким буде здійснюватися процедура шифрування, а також подальшого

	розшифрування
Схема проведення тесту	Запустити функцію шифрування. Після отримання зашифрованого файлу – запустити фунцію розшифрування
Очікуваний результат	Отримання зашифрованого файлу, а також, в процесі розшифрування, отримання відкритого (розшифрованого) файл
Стан програмного продукту після проведення випробувань	Список файлів, що над якими здійснювався процес шифрування, успішно отримано. Над будь-яким файлом, з даного списку, можливо здійснити операцію розшифрування з метою отримання відкритого тексту

Висновки за розділом 3

Розроблено програмний компонент підсистеми захисту бази даних web-орієнтованої інформаційної системи, використання якого дозволяє зашифровувати файли та надійно їх зберігати, а також надає змогу отримувати до даних файлів шляхом їх розшифрування.

Зокрема, описано процес реалізування програмного компоненту, що складався з вибору мови програмування, середовища створення програмного компоненту та СУБД, операційної системи, на базі якої створювався компоненту, а також технології його створення. Також в даному розділі проведено тестування програмного компоненту, а також описано використані при цьому методики.

На завершальному етапі описано використання програмного компоненту, що супроводжувався графічними прикладами у вигляді ілюстративного матеріалу виконання програми.

ВИСНОВКИ

Вирішено актуальне завдання моделювання програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи забезпечення роботи рекламного агентства для його програмної реалізації. Це стало можливим завдяки отриманню таких окремих результатів:

1. Проаналізовано процес захисту на основі використання процедур шифрування на основі симетричного блокового алгоритму AES-128. З огляду на це розроблено вимоги до програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи.

2. Створено концептуальну модель програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи, використання якої дозволяє формалізувати його застосування щодо функцій, процесів та потоків даних, а також сформувати та обґрунтувати варіанти використання програмного компоненту.

3. Створено структуру програмного компоненту підсистеми захисту бази даних web-орієнтованої інформаційної системи на основі формалізування його роботи щодо функцій, процесів та потоків даних, використання якої забезпечує їх функціональну придатність.

4. Розроблено програмний компонент підсистеми захисту бази даних web-орієнтованої інформаційної системи на основі його структури, використання якого дозволяє захисти файли при передачі їх до відповідних БД шляхом шифрування, а також отримання відкритих файлів за допомогою процедури розшифрування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методи та засоби мультимедійних інформаційних систем: навч. посіб. / Т. М. Басюк, П. І. Жежнич; Нац. ун-т «Львів. політехніка». Львів: Вид-во Львів. політехніки, 2015. 426 с.
2. Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. Х.: ХНАМГ, 2010. 222 с.
3. Бази даних та інформаційні системи. URL: https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/16313/1/%D0%91%D0%B0%D0%B7%D0%B8%20%D0%B4%D0%B0%D0%BD%D0%B8%D1%85%20%D1%82%D0%B0%20%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D1%96%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B8.%20%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%97.pdf
4. Інформатика: Комп'ютерна техніка. Комп'ютерні технології. Київ : Академія, 2002. 704 с.
5. Персональний комп’ютер. URL: https://step.org.ua/konspekt/ibmpc/tema1
6. Руденко В.Д., Макарчук О.М., Патланжоглу М.О. Практичний курс інформатики. Київ : Фенікс, 1996. С. 418.
7. Ярмуш О.В., Редько М.М. Інформатика і комп’ютерна техніка. Київ : Вища освіта, 2006. С. 359.
8. Гуржій А.М., Поворознюк Н.І., Самсонов В.В. Інформатика та інформаційні технології. Харків : ООО «Компанія СМІТ», 2003. С. 352.
9. Інформатика. Комп’ютерна техніка. Комп’ютерні технології. Київ : Каравела, 2011. С. 592.
10. Криптографія. URL: https://esu.com.ua/search_articles.php?id=1576 (дата звернення: 05.2022).
11. Корченко О. Г. Прикладна криптологія: системи шифрування: підручник / О. Г. Корченко, В. П. Сіденко, Ю. О. Дрейс. К.: ДУТ, 2014. 448 с.
12. Вербіцький О. В. Вступ до криптології. Л. : ВНТЛ, 1998. 248 с.

13. Корченко О.Г. Охорона конфіденційної інформації підприємства: навчальний посібник / О.Г. Корченко, Ю.О. Дрейс. Житомир: ЖВІ НАУ, 2011. 172 с.
14. Грайворонський, М. В. Безпека інформаційно-комунікаційних систем. підручник / М. В. Грайворонський, О. М. Новіков. Київ : Видавнича група BHV, 2009. 698 с.
15. Методологія функціонального моделювання IDEF0. URL: https://nsu.ru/smk/files/idef.pdf. (дата звернення: 05.2022).
16. Методологія IDEF3. URL: https://itteach.ru/bpwin/metodologiya-idef3. – (дата звернення: 05.2022).
17. DFD – Вікіпедія. URL: https://ru.wikipedia.org/wiki/DFD. (дата звернення: 05.2022).
18. Простий посібник зі схем UML і моделювання баз даних. URL: https://www.microsoft.com/uk-ua/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling
19. Мова програмування C/C++. Основні поняття. Типи даних. URL: https://www.eolymp.com/uk/blogs/posts/26
20. Microsoft.com. URL: https://www.microsoft.com/ru-ru/download/details.aspx?id=35579 (дата звернення: 05.2022).
21. Канер С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / С. Канер, Д. Фолк, Е. Нгуен. К. : Издательство «ДиаСофт», 2010. 544 с.
22. Ініціалізаційний вектор. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Ініціалізаційний_вектор
23. Марченко О.О. Інформаційна система забезпечення роботи рекламного агентства. Збірник праць учасників Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених «Інформаційні технології та моделювання систем», 30 березня 2023 р., Житомир. URL: https://drive.google.com/file/d/1qBxHc7bTla--u6FbuPtHKucXxggF18ro/view?usp=share_link