

КОРОБЕЙНИКОВА ТЕТЯНА

Національний університет «Львівська політехніка»

<https://orcid.org/0000-0003-2487-8742>e-mail: tetianakorobeinikova@gmail.com

ПРАКТИКА АСЕМБЛЮВАННЯ ТА РЕАСЕМБЛЮВАННЯ ДОКУМЕНТІВ У СПЕЦІАЛІЗОВАНІЙ АВТОМАТИЗОВАНІЙ СИСТЕМІ ДОКУМЕНТООБІГУ

У роботі представлено практичну реалізацію серверного рішення для ефективного створення та автоматизованого генерування документів у спеціалізованій автоматизованій системі документообігу (САСД). У роботі розглянуто групу алгоритмів на стороні сервера, які забезпечують формування надійних підписів для документів та уникнення дублювання, використовуючи JWT токени та хешування SHA256.

Ключові слова: система документообігу, процеси асемблювання та реасемблювання документів, підписування документів, електронно-цифровий підпис, JWT токен, хешування.

KOROBEGINIKOVA TETIANA

Lviv National Technical University

PRACTICE OF DOCUMENT ASSEMBLY AND REASSEMBLY IN A SPECIALIZED AUTOMATED DOCUMENT CIRCULATION SYSTEM

This work is dedicated to solving the problem of document assembly and reassembly during document circulation and focuses on the development of server-side algorithms. The work examines mechanisms that allow for reliable and unique digital signatures for documents by forming JWT tokens for actor identification and hashing documents using SHA256. Various organizations, especially educational ones, receive numerous requests in the form of documents that are manually formed, leading to an increase in bureaucratic errors. Thus, there is a need to analyze and improve methods and means of automated processing and generation of documents. The main focus is on the processes of document assembly and reassembly. The aim of the work is to develop and implement document circulation algorithms for creating and automated generation of documents in a specialized automated document circulation system (ADCS). This is achieved through the application of server procedures for document assembly and reassembly. ADCS is a document circulation system, the purpose of which is to automate the process of document processing and manual processes associated with them. The reassembly process is the breakdown of a document into separate parts, using special markers and regular expressions for further analysis of these parts. The assembly process involves converting a document from a program state to a physical one for further processing. During assembly, a document is formed using available program objects and data. Increasing the efficiency of document processing is achieved by leveling the shortcomings of the manual approach to document processing. A group of algorithms for the proposed ADCS is designed to improve the experience of using the services of institutions with high bureaucratic load.

Keywords: document circulation system, processes of document assembly and reassembly, document signing, digital signature, jwt token, hashing.

Постановка проблеми

У даній роботі було використано системний підхід для вирішення проблеми асемблювання та реасемблювання документів у спеціалізованій системі документообігу. Основна увага зосереджена на розробці алгоритмів на стороні сервера та обробці документів на рівні кожного департаменту. Тут розглянуті ключові механізми, які дозволяють забезпечити надійні та унікальні електронно-цифрові підпису (ЕЦП) для документів шляхом формування JWT токенів для ідентифікації акторів та хешування документів за допомогою SHA256.

Аналіз останніх джерел

Щороку різні організації, зокрема, освіти, отримують безліч запитів у формі документів [1-5]. Кожен документ має відповідати певним вимогам і формується вручну. На основі цих документів складаються протоколи засідань та витяги з них. Такий підхід призводить до збільшення бюрократичних помилок [6-9], які можуть бути не лише граматичними. Таким чином, існує потреба в аналізі та вдосконаленні методів та засобів автоматизованої обробки та генерації документів. Основна увага зосереджується на процесах асемблювання та реасемблювання документів. Отож, виникає необхідність у розробці САСД, що автоматизує процес створення витягів з протоколів {УЕ

засідань структурних підрозділів та обміну документами між ними. Запропонована розробка дозволить редагувати, зберігати та передавати документи між структурними підрозділами, спрощуючи процес їх подання та розгляду на всіх рівнях бюрократичної ієрархії та забезпечить можливість застосування ЕЦП для документів [10-13].

Метою роботи є: розробка та впровадження алгоритмів документообігу для ефективного створення та автоматизованого генерування документів у спеціалізованій автоматизованій системі документообігу (САСД). Це досягається через застосування серверних процедур асемблювання та реасемблювання документів.

Основні поняття та основа для реалізації

САСД – це така система документообігу, первинною ціллю якої є автоматизація процесу обробки документів та всіх мануальних процесів пов'язаних з ними.

Процес реасемблювання – це процес розбиття документа на окремі частини, використовуючи спеціальні маркери та регулярні вирази для подальшого аналізу цих частин. Цей процес передбачає

перетворення документа на програмні об'єкти для подальшого використання. Процес реасемблювання показано на рисунку 1, а. Тут система аналізує наданий протокол і автоматично створює витяги для нього.

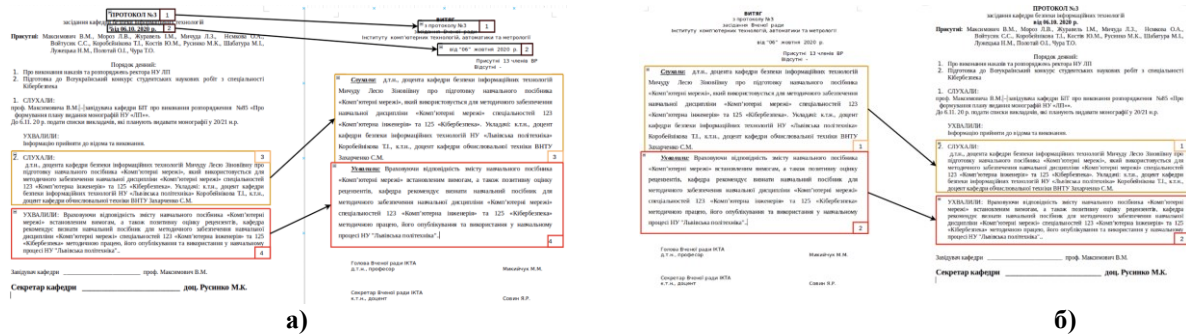


Рис. 1. Візуалізація процесів реасемблювання та асемблювання

Процес асемблювання – це процес, протилежний реасемблюванню і полягає в перетворенні документа з програмного стану в фізичний для подальшої обробки. Зазвичай цей процес передбачає використання порожнього (незаповненого) шаблону документа, який програма заповнює даними в автоматичному режимі. Під час асемблювання формується документ за допомогою доступних програмі об'єктів та даних. Процес асемблювання показано на рисунку 1,б.

Підвищення ефективності обробки документів відбувається шляхом нівелювання недоліків мануального підходу до обробки документів, натомість передбачається використання хмарних технологій та спеціалізованих засобів парсингу для організування автоматичного генерування нових документів та швидкого отримання інформації, що може бути оброблена програмно.

Призначення САСД та постановка типової задачі

У більшості випадків в процесі обробки паперових документів люди мають справу із помилками. Таким чином, документ стає предметом, що вимагає постійної уваги та супроводу з боку особи, що його ініціює [17-22]. Одним з таких недоліків є різноманітність інстанцій. Група алгоритмів для запропонованої САСД покликана покращити досвід користування послугами закладів з високою бюрократичною завантаженістю. Окрім створення та збереження документів в хмарі запропонована САСД буде надавати такі можливості:

- закріплення документу за певним користувачем;
- автоматизоване створення витягів з протоколів шляхом реасемблювання документу;
- автоматизоване створення протоколів на основі БД, що поповнюються самими користувачами;
- повністю автоматизований процес розгляду, підписання документу та його обробка (при потребі, і – генерація) в інстанціях всіх рівнів без участі користувача.

Спрощена постановка тривіальної задачі: є потреба для видавництва Львівської Політехніки надати витяги з протоколу засідання вченої ради університету, інституту та засідання кафедри. Можна це реалізувати особисто пройшовши по всім інстанціям, а можна автоматизувати цей процес та використати переваги ІТ. Отож, опишемо типову задачу документообігу між структурними одиницями у вузі.

На засіданні кафедри секретар веде протокол засідання кафедри (ПЗК), з ПЗК роблять витяги ПЗК, система буде містити стандартну форму ПЗК та витягу з ПЗК. З даних ПЗК система автоматично буде формувати витяги по всім питанням. Далі буде можливість зробити ЕЦП для сформованих документів, а після цього – скерувати в який саме підрозділ в подальшому йде витяг ПЗК (наприклад, на вчену раду (ВР) інституту). Якщо витяг не потрібен – буде передбачено процес неформування витягу. Також буде можливість зберігання витягу і звертання до нього коли він передбачений, але зберігається на кафедрі. В результаті користувачу буде доступний підписаний витяг ПЗК секретарем, а також можливість надіслати його на цифровий підпис завідувачу кафедри. Знову ж, користувачеві стає доступним підписаний витяг з ПЗК секретарем та завкафедрою, а секретарю – можливість підтвердити надсилання до секретаря ВР інституту (СВРІ). Після цього, у СВРІ з'являється можливість підтвердити прийняття витягу, у випадку підтвердження – система автоматично включає дані з витягу ПЗК у протокол засіданні ВР інституту (ЗВРІ), у результаті чого формується проект протоколу ЗВРІ, у який, за потреби, СВРІ може вносити зміни, додавати нові пункти. Після цього, у СВРІ з'являється можливість його підписати. Далі користувачу стає доступним підписаний витяг ПЗВРІ секретарем, який автоматично надсилається директору інституту, у якого з'являється можливість його підписати. У результаті – користувач має підписаний витяг з ПЗВРІ секретарем та директором, а СВРІ має можливість скерувати витяг до секретаря ВР університету (СВРУ), якщо є така потреба. Далі СВРУ може прийняти витяг, у випадку прийняття – система автоматично включає дані з витягу СВРІ у протокол ЗВРУ та формує проект протоколу ЗВРУ, який, також, у разі потреби, СВРУ може редагувати чи додавати нові пункти. Після завершення внесення змін, СВРУ має можливість його підписати. Знову ж користувачу стає доступним підписаний витяг ПЗВРУ секретарем, який також надсилається ректору для підписання. Як результат – користувач має підписаний витяг з ПЗВРУ секретарем та ректором. Останній крок – підписаний витяг з ПЗВРУ – надсилається голові видавництва ЛП.

Розробка алгоритмів роботи САСД

Розглянемо алгоритм роботи САСД поетапно на прикладі обробки документу на рівні кожного з департаментів: 1) на рівні кафедри – структурного підрозділу 1-го рівня; 2) на рівні вченої ради інституту – структурного підрозділу 2-го рівня; 3) на рівні вченої ради університету – структурного підрозділу 3-го рівня.

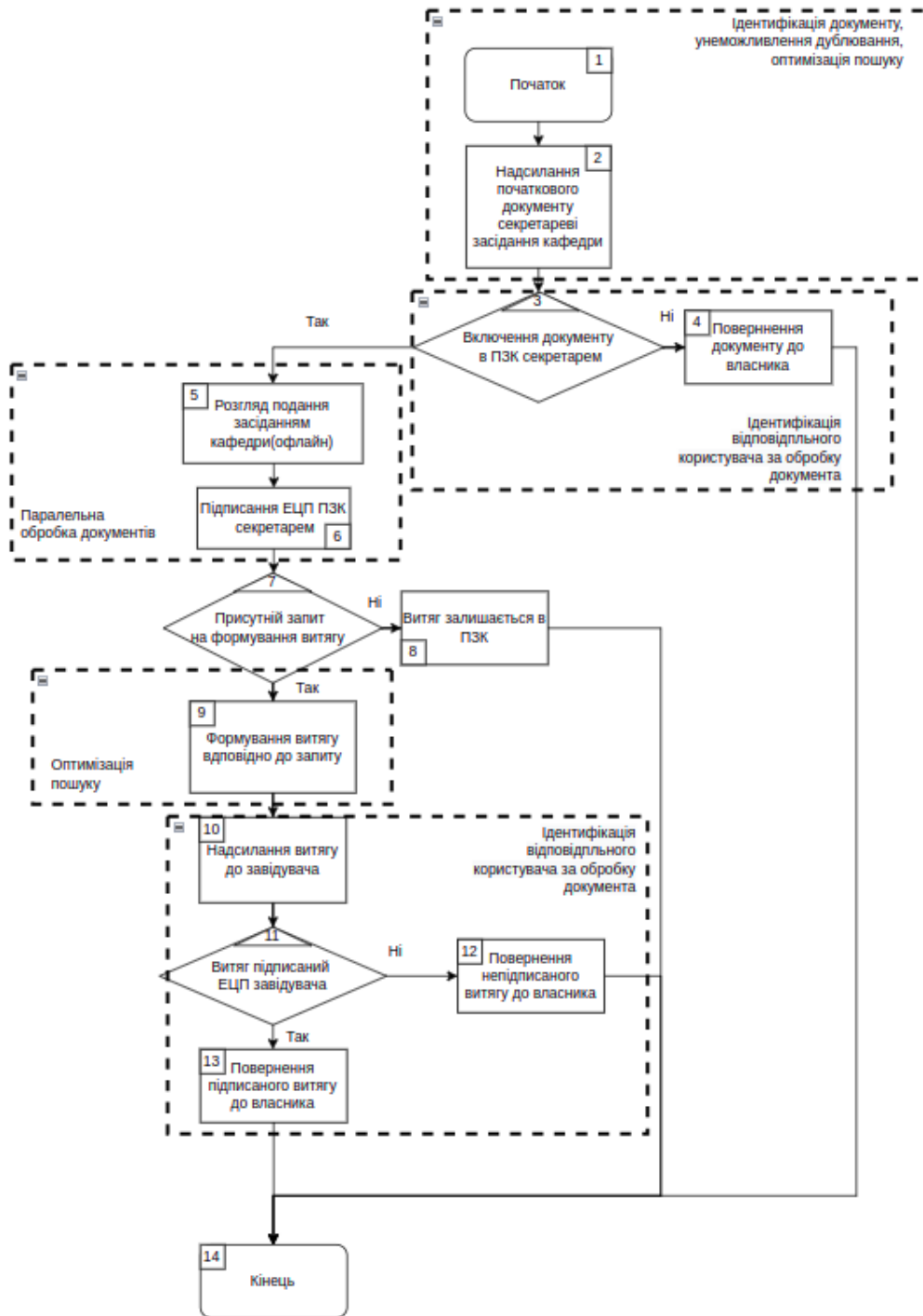


Рис. 2. Алгоритм обробки документів на рівні засідання кафедри

Обробка документів на рівні кафедри. Обробка документів на рівні кафедри передбачає врахування операцій асемблювання та реасемблювання, отримання підписів акторів структурного підрозділу першого рівня: секретаря та (рис. 2).

- 1) Початок алгоритму.
- 2) Користувач створює файл, а система завантажує його у хмару і реєструє запис в БД. Створений документ синхронізується (для перегляду на інших пристроях користувача) і надсилається секретареві обраного департаменту. Новоствореному документу надається помітка “НАДІСЛАНО”.

- 3) Секретар отримує сповіщення про отриманий файл і включає даний документ до протоколу засідання кафедри. У той же час файлу включеному в протокол надається помітка “ЗАТВЕРДЖЕНО”. В іншому разі відхилений файл повертається до автора документу з поміткою “ВІДХИЛЕНО”.
- 4) У разі якщо секретар не включив файл до протоколу засідання кафедри відхилений файл повертається до автора документу з поміткою “ВІДХИЛЕНО”.
- 5) Далі відбувається оффлайн засідання кафедри на якому розглядаються документи включені секретарем до протоколу.
- 6) Секретар створює його фінальну версію (з рішенням засідання стосовно кожного з документів) використовуючи інтерфейс програми. Для цього секретарю необхідно надати інформацію стосовно дати проведеного засідання, присутніх членів, порядку денного, матеріали розглянутих питань і рішень по кожному з питань. Після чого надані матеріали будуть автоматично сформовані у кінцеву версію протоколу. Після формування кінцевої версії протоколу, протокол засідання кафедри (ПЗК) підписується електронно-цифровим підписом (ЕЦП) секретаря кафедри. Далі протокол сканується на предмет визначення документів необхідних для витягу для кожного із надсилаючих користувачів.
- 7) САСД визначає файли, що мають бути сформовані у витяг.
- 8) У разі якщо надсилаючий користувач не запросив витяг надісланого ним документу – документ залишається в ПЗК з поміткою “ЗАКРИТО”.
- 9) У разі якщо документ необхідно оформити у витяг з протоколу вибираються всі матеріали, що стосуються даного витягу і формуються в окремий файл. Сформованому файлу надається помітка “ПІДПИСАНО”.
- 10) Далі всі витяги, що вимагають підпису завідувача кафедри надсилаються завідувачеві, про що останній отримує відповідне сповіщення. Перш ніж надіслати такі витяги програма перевіряє кожен з них на предмет присутності підпису секретаря засідання кафедри для кожного з них в БД.
- 11) Завідувач кафедри переглядає документи надіслані йому задля отримання підпису і підписує потрібні. На цьому етапі програма викликає окремий сервіс для генерації підпису завідувача, прив’язки його до відповідного документу і створення запису в БД.
- 12) Після успішного підписання документи автоматично надсилаються секретареві вченої ради інституту і їм надається мітка “НАДІСЛАНО”.
- 13) У разі якщо документ не було підписано завідувачем кафедри сформований витяг надсилається до його автора, а мітка документу змінюється на мітку “ВІДХИЛЕНО”.
- 14) Кінець алгоритму

Обробка документів на рівні вченої ради інституту. Обробка документів на рівні кафедри передбачає врахування операцій асемблювання та реасемблювання, отримання підписів акторів структурного підрозділу другого рівня: секретаря та директора інституту. Блок-схема алгоритму обробки документів на рівні кафедри є на рис. 3 і містить такі кроки:

- 1) Початок алгоритму
- 2) На етапі надсилання підписаного витягу секретареві вченої ради інституту (СВРІ), секретар отримує сповіщення про надісланий йому документ. Перш ніж надіслати такі витяги програма перевіряє кожен з них на предмет присутності підпису завідувача кафедри для кожного з них в БД.
- 3) Секретар включає надісланий йому документ до протоколу засідання вченої ради інституту (ПЗВРІ). У той же час файлу включеному в протокол надається помітка “ЗАТВЕРДЖЕНО”. В іншому разі відхилений файл повертається до автора документу з поміткою “ВІДХИЛЕНО”.
- 4) У разі якщо секретар не включив файл до протоколу засідання кафедри відхилений файл повертається до автора документу з поміткою «ВІДХИЛЕНО».
- 5) Далі відбувається оффлайн засідання вченої ради інституту на якому розглядаються документи включені секретарем до протоколу.
- 6) На етапі підписання протоколу засідання вченої ради інституту (ПЗВРІ) секретар створює його фінальну версію (з рішенням засідання стосовно кожного з документів) використовуючи інтерфейс програми. Для цього секретарю необхідно надати інформацію стосовно дати проведеного засідання, присутніх членів засідання, порядку денного, матеріали розглянутих питань і рішень винесених по кожному з питань окремо. Після чого надані матеріали будуть автоматично сформовані у кінцеву версію протоколу. Після формування кінцевої версії протоколу, протокол підписується електронно-цифровим підписом (ЕЦП) секретаря, Далі протокол сканується на предмет визначення документів необхідних для витягу для кожного із надсилаючих користувачів.
- 7) На цьому етапі система визначає файли, що мають бути сформовані у витяг.
- 8) У разі якщо надсилаючий користувач не запросив витяг надісланого ним документу – документ залишається в ПЗВРІ з поміткою “ЗАКРИТО”.
- 9) У разі якщо документ необхідно оформити у витяг з протоколу вибираються всі матеріали, що

стосуються даного витягу і формуються в окремий файл. Сформованому файлу надається помітка “ПІДПИСАНО”.

- 10) Далі всі витяги, що вимагають підпису директора інституту надсилаються директорові, про що останній отримує відповідне сповіщення. Перш ніж надіслати такі витяги програма перевіряє кожен з них на предмет присутності підпису секретаря засідання вченої ради інституту для кожного з них в БД.

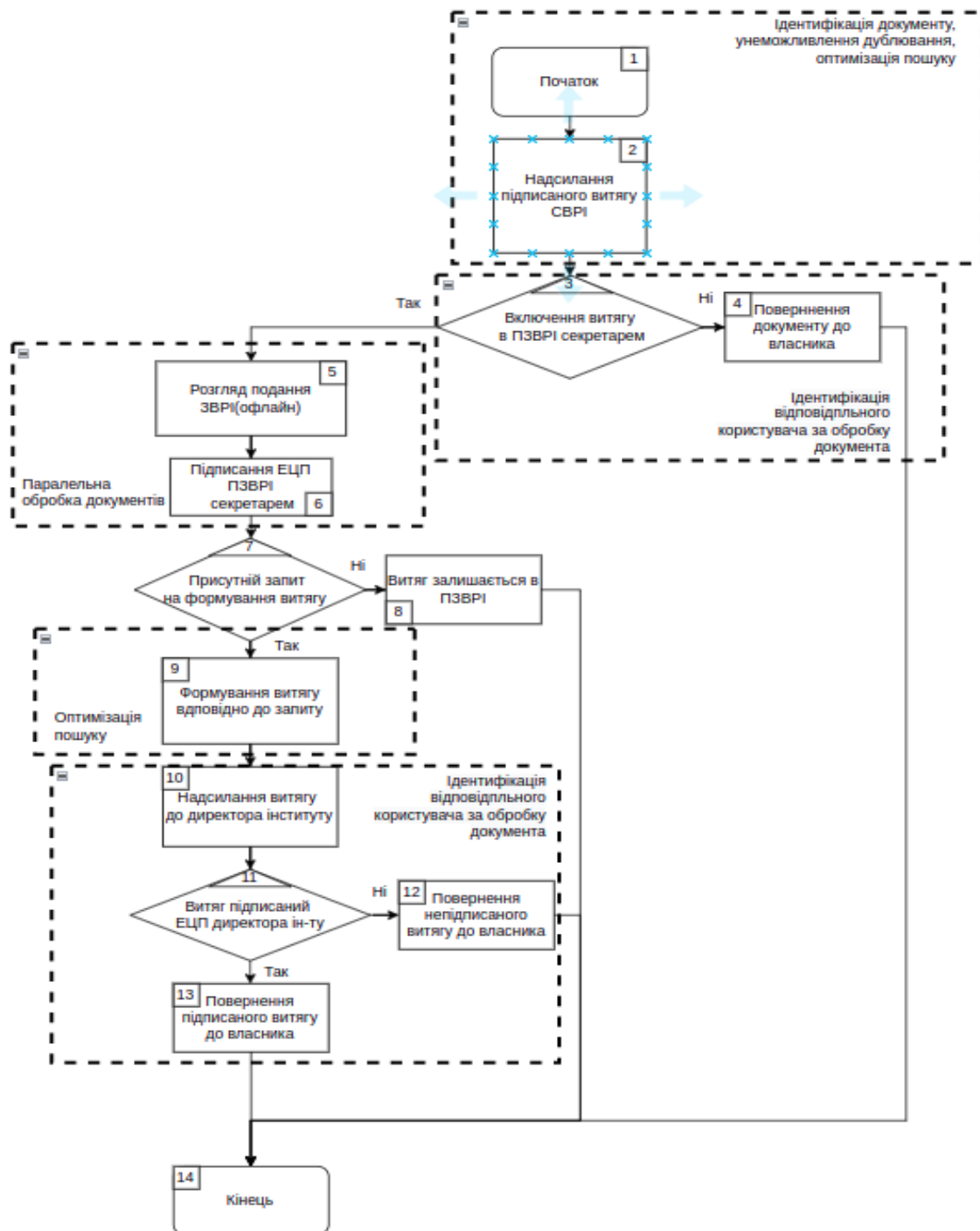


Рис. 3. Алгоритм обробки документів на рівні засідання вченої ради інституту

- 11) Директор інституту переглядає документи надіслані йому задля отримання підпису і підписує потрібні. На цьому етапі програма викликає окремий сервіс для генерації підпису директора, прив'язки його до відповідного документу і створення запису в БД.
- 12) Після успішного підписання документи автоматично надсилаються секретареві вченої ради університету і їм надається мітка “НАДІСЛАНО”.
- 13) У разі якщо документ не було підписано директором інституту сформований витяг надсилається до його автора, а мітка документу змінюється на мітку “ВІДХИЛЕНО”.
- 14) Кінець алгоритму.

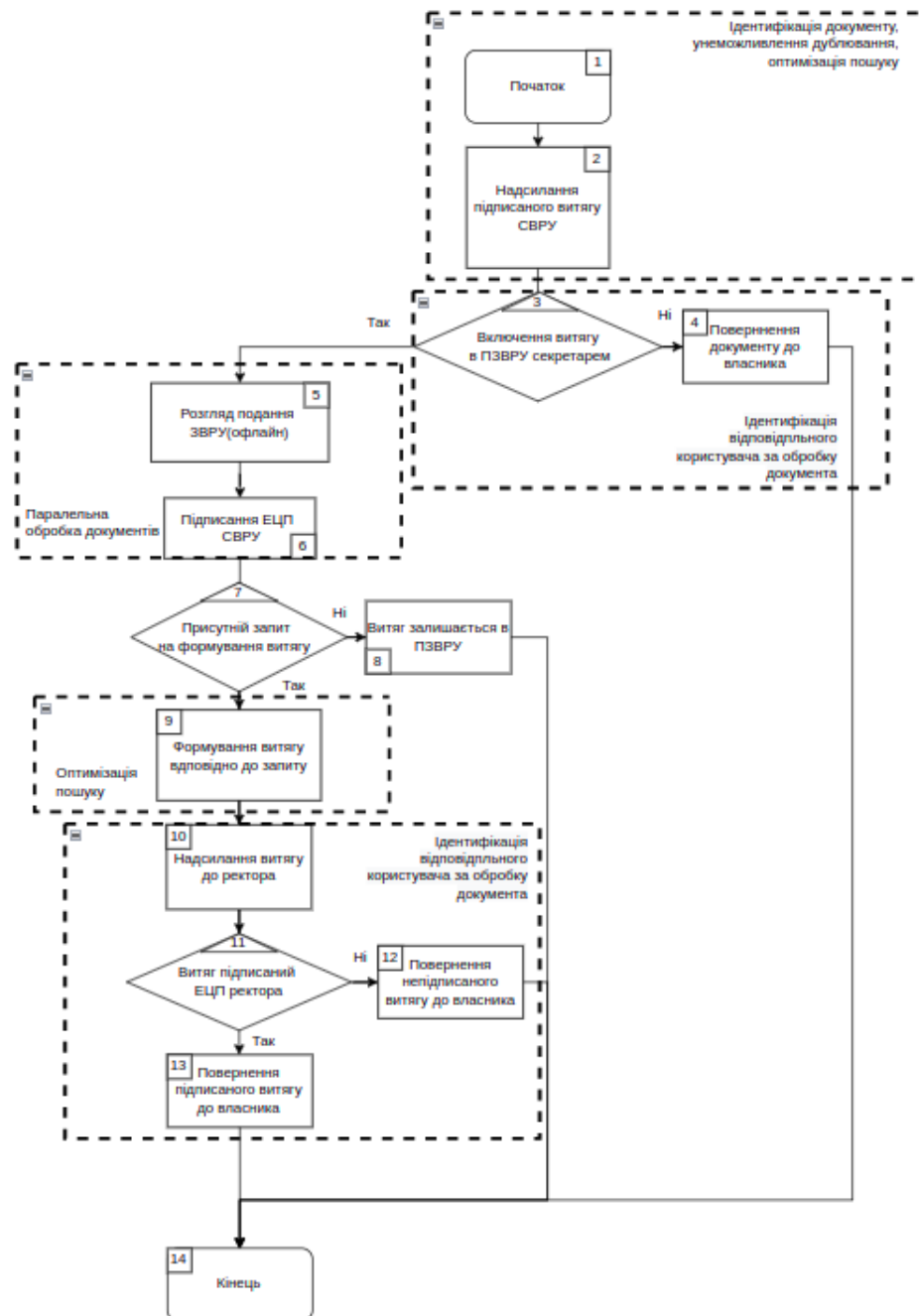


Рис. 4. Алгоритм обробки документів на рівні засідання вченої ради університету

Обробка документів на рівні вченої ради університету. Обробка документів на рівні кафедри передбачає врахування операцій асемблювання та реасемблювання, отримання підписів акторів структурного підрозділу третього рівня: секретаря та ректора (рис. 4).

- 1) Початок алгоритму
- 2) На етапі надсилання підписаного витягу секретареві вченої ради університету (СВРУ), секретар отримує сповіщення про надісланий йому документ. Перш ніж надіслати такі витяги програма перевіряє кожен з них на предмет присутності необхідних підписів.
- 3) Секретар включає надісланий йому документ до протоколу засідання вченої ради університету (ПЗВРУ). У той же час файлу включеному в протокол надається помітка “ЗАТВЕРДЖЕНО”. В іншому разі відхилений файл повертається до автора документу з поміткою “ВІДХИЛЕНО”.
- 4) У разі якщо секретар не включив файл до протоколу засідання кафедри відхилений файл повертається до автора документу з поміткою “ВІДХИЛЕНО”
- 5) Далі відбувається оффлайн засідання вченої ради університету на якому розглядаються документи включені секретарем до протоколу.
- 6) На етапі підписання протоколу засідання вченої ради університету (ПЗВРУ) секретар створює його фінальну версію (з рішенням засідання стосовно кожного з документів) використовуючи інтерфейс програми. Для цього секретарю необхідно надати інформацію стосовно дати

проведеного засідання, присутніх членів засідання, порядку денного, матеріали розглянутих питань і рішень винесених по кожному з питань окремо. Після чого надані матеріали будуть автоматично сформовані у кінцеву версію протоколу. Після формування кінцевої версії протоколу, протокол підписується електронно-цифровим підписом (ЕЦП) секретаря, Далі протокол сканується на предмет визначення документів необхідних для витягу для кожного із надсилаючих користувачів.

- 7) Тут система визначає файли, що мають бути сформовані у витяг.
- 8) У разі якщо надсилаючий користувач не запросив витяг надісланого ним документу – документ залишається в ПЗВРУ з поміткою “ЗАКРИТО”.
- 9) У разі якщо документ необхідно оформити у витяг з протоколу вибираються всі матеріали, що стосуються даного витягу і формуються в окремий файл. Сформованому файлу надається помітка “ПІДПИСАНО”.
- 10) Далі всі витяги, що вимагають підпису ректора надсилаються ректорові, про що останній отримує відповідне сповіщення. Перш ніж надіслати такі витяги програма перевіряє кожен з них на предмет присутності підпису секретаря засідання вченої ради університету для кожного з них в БД.
- 11) Ректор переглядає документи надіслані йому задля отримання підпису і підписує потрібні. На цьому етапі програма викликає окремий сервіс для генерації підпису директора, прив'язки його до відповідного документу і створення запису в БД.
- 12) Після успішного підписання документи автоматично надсилаються їх авторам, а файлу надається помітка “ЗАКРИТО”.
- 13) У разі якщо документ не було підписано ректором сформований витяг надсилається до його автора, а мітка документу змінюється на мітку “ВІДХИЛЕНО”.
- 14) Кінець алгоритму.

Ключові механізми САСД та їх взаємодія на рівні клієнт-сервер

Перш ніж деталізувати ключові механізми слід розглянути базові інструменти, що програма використовує для ідентифікації та аутентифікації користувачів та інших об'єктів в системі. Будь яку сутність, що необхідно представити програмно зберігається в БД у вигляді об'єктів. Об'єкти в свою чергу містять набір даних у вигляді полів, які дозволяють дізнатись більше про власне об'єкт. Одним із таких полів є полу ID. Кожен об'єкт на спосіб аутентифікації користувачів в системі. Процес аутентифікації імплементований на базі використання JWT(JSON Web Token) токенів. JWT токен – це стрічка зашифрованих секретом даних, що містять відомості про ID. Зазвичай JWT токен (рис. 5, а) складається з заголовку (рис. 5, б), вмісту (рис. 5, г) та підпису (рис. 5, в). Заголовок вказує на алгоритм шифрування використаний для формування токенау.

```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJkaDM2Y1
A4WXh5anYiLCJhdXoiOiJVU0VSIiwiaWF0IjoiU
29JVERlCGFydG11bnQlCj1eNAiOjE2NTQxOTIz
NTMsIm1hdCI6MTY1MzU4ODE1M30.1QdPdE5iG1M
SptxTYtNtHsXJnhkgV2NXvedQRqzCgYg
```

а) Приклад JWT токенау

```
1 wR5j7qmC7g04yxJcv3mCbJi4ZnTp9UYNQicLZKbi7rA
```

в) підпис токенау

Рис. 5. JWT токен, представлення його заголовку (а), вмісту (б) та підпису (в) JWT токенау

HEADER: ALGORITHM & TOKEN TYPE
<pre>{ "alg": "HS256" }</pre>
PAYLOAD: DATA
<pre>{ "sub": "dh36bP8Yxyjv", "aud": "USER", "dpt": "SoITDepartment", "exp": 1654192353, "iat": 1653588153 }</pre>

б) Заголовок токенау

г) Вміст токенау

Вміст складається з даних, що використовуються для авторизації користувача. Кожна частина даних вмісту ідентифікується власним іменем, які можуть міститися в реєстрі тверджень JWT специфікації або бути надані системою, що згенерувала токен. В даному випадку ID користувача зберігається під іменем “SUB”, час генерації токена - “IAT”, а департамент якому належить користувач – “DPT”. Підпис служить верифікатором токена. Він будується шляхом конкатенації двох попередніх частин через крапку, шифрування алгоритмом Base64 та подальшої передачі побудованої строки до криптографічного алгоритму вказаного в заголовку. Всі три частини (заголовок, вміст, підпис) у зашифрованому вигляді згодом конкатинуються через крапку формуючи завершений JWT токен.

Висновки

Представлена робота використовує системний підхід до вирішення задачі асемблювання та реасемблювання документів у спеціалізованій системі документообігу. Основна увага приділена опису та реалізації алгоритмів автоматизованого генерування документів в середовищі спеціалізованої

автоматизованої системи документообігу на стороні сервера, а також опису обробки документів на рівні кожного з департаментів. Деякі ключові механізми САСД, такі як формування надійних підписів для документів та уникнення дублювання документів, були розглянуті, включаючи формування JWT токенів для ідентифікації користувачів в програмі та хешування документів за допомогою алгоритму SHA256.

Література

1. На шляху до Індустрії 4.0: інформаційні технології, моделювання, штучний інтелект, автоматизація : монографія / кол. авт. : В. Б. Артеменко, Л. В. Артеменко, О. В. Артеменко [та ін.] ; за заг. ред. С. В. Котлика. – Одеса : Астропринт, 2021. – 544 с. – ISBN 978-966-927-702-2.
2. Wissenschaft für den modernen Menschen 2021 / Science for modern man 2021. Innovative engineering and technology, informatics, security systems, transport development, architecture. Book 4. Part 4. monograph. ScientificWorld-NetAkhatAV Lußstr 13, Karlsruhe, Germany, 2021. 236 p. ISBN 978-3-949059-12-4 DOI: 10.30890/2709-2313.2021-04-04.
3. Big Data Processing: methods, models and information technologies: monograph. edited by Oleg I. Pursky. Shoida GmbH, Steyr, Austria, 2019. 234 p. ISBN 978-3-953794-29-8.
4. Коробейнікова Т. І. Удосконалений метод розробки API підвищеної швидкодії / Т. І. Коробейнікова, Л. А. Савицька // Інформаційні технології та комп'ютерна інженерія. – Вінниця, 2021. – № 1 (том 50). – С. 31-35.
5. Коробейнікова Т.І. Засіб створення каталогу електронних посилань на основі користувацької системи категоризації / Коробейнікова Т.І., Мичуда Л.З., Савицька Л.А. // Електронні інформаційні ресурси: створення, використання, доступ : збірник матеріалів Міжнародної науково-практичної Інтернет-конференції 9-10 листопада 2020 р. – Суми/Вінниця : НІКО/ВНТУ, 2020. – С. 144-151. – ISBN 978-617-7422-13-5.
6. Коробейнікова Т. І. Метод та програмний засіб застосування метаданих в процесах пошуку / Т. І. Коробейнікова, Л. А. Савицька, Д. В. Тягун // Інформаційні технології та комп'ютерна інженерія – 2020. – № 46. – С. 21-27.
7. Коробейнікова Т. І. Метод та кросплатформений засіб архівації однотипних файлів / Т. І. Коробейнікова, Савицька Л.А., П. О. Чирва // Інформаційні технології та комп'ютерна інженерія – 2020. – № 47. – С. 14-21.
8. Рівень розвитку техніки і технологій в XXI столітті. Частина 1 : Серія монографій / [авт.кол. : М.В. Князева, В.М. Крамар, І.Я. Львович, А.П. Преображенський, О.Н. Романюк і ін.]. – Одеса : КУПІРІЄНКО СВ, 2019 - 227 с. : іл., табл. – (Серія «Рівень розвитку техніки і технологій в XXI столітті», Частина 1). – ISBN 978-617-7414-75-8.
9. Савицька Л. А. Методи та засоби захисту ресурсів в комп'ютерній SDN-мережі / Л.А. Савицька, Т.І. Коробейнікова, І.В. Леонт'єв, С.В. Богомолів // Інформаційні технології та комп'ютерна інженерія – 2023. – № 3. – С. 41–51. – ISSN 1999–9941.
10. Захарченко С. М. Застосування односторінкових веб-орієнтованих інтерфейсів в соціально значущих проєктах. / С. М. Захарченко, Т. І. Трояновська, О. В. Бойко В. С. Рибаченко // Вісник Хмельницького національного університету. – 2016. – № 3. – С. 33-39.
11. Коробейнікова Т.І. Аналіз сучасних систем документообігу / Т. І. Коробейнікова, І. К. Пустовіт // Modern systems of science and education in the USA, EU and other countries (25-26 січня, 2022). Seattle, Washington, USA. 2022. – С. 19-22. – ISBN 979-8-4435989-0-1. DOI: 10.30888/2709-2267.2022-9.
12. Гороховський О. І. Моделювання, створення та практика автоматизованих систем дистанційного навчання / О. І. Гороховський, Т. І. Трояновська // Інформаційні технології та комп'ютерна інженерія – 2007. – № 1(8). – С. 235–239. – ISSN 1999–9941.
13. Трояновська Т. І. Інформаційна технологія доставки контенту у системах комп'ютеризованої підготовки спеціалістів // Гороховський О. І., Трояновська Т. І., Азаров О. Д. – Вінниця : ВНТУ, 2016. – 160 с.
14. T. Korobeinikova, R. Chekhmestruk, P. Mykhaylov, O. Romanyuk, O. Romanyuk and N. Achanyar, "The Fault-Resistant Web Application Infrastructure Using Autoscaling," 2023 13th International Conference on Advanced Computer Information Technologies (ACIT), Wroclaw, Poland, 2023, pp. 479-482, doi: 10.1109/ACIT58437.2023.10275448.
15. T. Korobeinikova, V. Maidaniuk, O. Romanyuk, R. Chekhmestruk, O. Romanyuk and S. Romanyuk, "Web-applications Fault Tolerance and Autoscaling Provided by the Combined Method of Databases Scaling," 2022 12th International Conference on Advanced Computer Information Technologies (ACIT), 2022, pp. 27-32, doi: 10.1109/ACIT54803.2022.9913098.
16. Коробейнікова Т.І. Відмовостійкість та автомасштабування веб-ресурсу / Коробейнікова Т.І., Захарченко С. М. // International scientific journal «Grail of Science» – 2022. – № 14-15 (May, 2022). – С. 312–319. – ISSN: 2710–3056. ISBN 979-8-88526-799-1.
17. Cherniak I., Pustovit I., Korobeinikova T., Luzhetska N. Algorithm of work of specialized document management system. Information protection and information systems security: Materials of VIII-th International Scientific and Technical Conference, November 11 – 12, 2021. Lviv: NULP, 2021. P. 27-29.

References

1. Na shliakhu do Industriï 4.0: informatsiïni tekhnologii, modeliuvannia, shtuchnyï intelekt, avtomatyzatsiia : monohrafiia / kol. avt. : V. B. Artemenko, L. V. Artemenko, O. V. Artemenko [ta in.] ; za zah. red. S. V. Kotlyka. – Odesa : Astroprint, 2021. – 544 s. – ISBN 978-966-927-702-2.
2. Wissenschaft für den modernen Menschen 2021 / Science for modern man 2021. Innovative engineering and technology, informatics, security systems, transport development, architecture. Book 4. Part 4. monograph. ScientificWorld-NetAkhatAV Lußstr 13, Karlsruhe, Germany, 2021. 236 p. ISBN 978-3-949059-12-4 DOI: 10.30890/2709-2313.2021-04-04.
3. Big Data Processing: methods, models and information technologies: monograph. edited by Oleg I. Pursky. Shoida GmBH, Steyr, Austria, 2019. 234 p. ISBN 978-3-953794-29-8.
4. Korobeinikova T. I. Udoskonalenyi metod rozrobky ARI pidvyshchenoi shvydkodii / T. I. Korobeinikova, L. A. Savytska // Informatsiïni tekhnologii ta kompiuterna inzheneriia. – Vinnytsia, 2021. – № 1 (tom 50). – С. 31-35.
5. Korobeinikova T.I. Zasib stvorennia katalogu elektronnykh posylan na osnovi korystuvatskoi systemy katehoryzatsii / Korobeinikova T.I., Mychuda L.Z., Savytska L.A. // Elektronni informatsiïni resursy: stvorennia, vykorystannia, dostup : zbirnyk materialiv Mizhnarodnoi naukovo-praktychnoi Internet-konferentsii 9-10 lystopada 2020 r. – Sumy/Vinnytsia : NIKO/VNTU, 2020. – S. 144-151. – ISBN 978-617-7422-13-5.
6. Korobeinikova T. I. Metod ta prohramnyi zasib zastosuvannia metadanykh v protsesakh poshuku / T. I. Korobeinikova, L. A. Savytska, D. V. Tiahun // Informatsiïni tekhnologii ta kompiuterna inzheneriia – 2020. – № 46. – S. 21-27.
7. Korobeinikova T. I. Metod ta krosplatformenyi zasib arkhivatsii odnotypnykh failiv / T. I. Korobeinikova, Savytska L.A, P. O. Chyrva // Informatsiïni tekhnologii ta kompiuterna inzheneriia – 2020. – № 47. – S. 14-21.
8. Riven rozvytku tekhniky i tekhnologii v XXI stolitti. Chastyna 1 : Seriia monohrafii / [avt.kol. : M.V. Kniazieva, V.M. Kramar, I.Ia. Lvovych, A.P. Preobrazhenskyi, O.N. Romaniuk i in.]. – Odesa : KUPRIeNKO SV, 2019 - 227 s. : il., tabl. – (Seriia «Riven rozvytku tekhniky i tekhnologii v XXI stolitti», Chastyna 1). – ISBN 978-617-7414-75-8.
9. Savytska L. A. Metody ta zasoby zakhystu resursiv v kompiuterniï SDN-merezhi / L.A. Savytska, T.I. Korobeinikova, I.V. Leontiev, S.V. Bohomolov // Informatsiïni tekhnologii ta kompiuterna inzheneriia – 2023. – № 3. – S. 41–51. – ISSN 1999–9941.
10. Zakharchenko S. M. Zastosuvannia odnostorinkovykh veb-oriiantovanykh interfeisiv v sotsialno znachushchykh proektakh. / S. M. Zakharchenko, T. I. Troianovska, O. V. Boiko V. S. Rybachenko // Visnyk Khmelnytskoho natsionalnoho universytetu. – 2016. – № 3. – S. 33-39.
11. Korobeinikova T.I. Analiz suchasnykh system dokumentoobihu / T. I. Korobeinikova, I. K. Pustovit // Modern systems of science and education in the USA, EU and other countries (25-26 sichnia, 2022). Seattle, Washington, USA. 2022. – S. 19-22. – ISBN 979-8-4435989-0-1. DOI: 10.30888/2709-2267.2022-9.
12. Horokhovskiy O. I. Modeliuvannia, stvorennia ta praktyka avtomatyzovanykh system dystantsiinoho na-vchannia / O. I. Horokhovskiy, T. I. Troianovska // Informatsiïni tekhnologii ta kompiuterna inzheneriia – 2007. – № 1(8). – S. 235–239. – ISSN 1999–9941.
13. Troianovska T. I. Informatsiïna tekhnologiiia dostavky kontentu u systemakh kompiuteryzovanoi pidhotovky spetsialistiv // Horokhovskiy O. I., Troianovska T. I., Azarov O. D. – Vinnytsia : VNTU, 2016. – 160 s.
14. T. Korobeinikova, R. Chekhmestruk, P. Mykhaylov, O. Romanyuk, O. Romanyuk and H. Achanyar, "The Fault-Resistant Web Application Infrastructure Using Autoscaling," 2023 13th International Conference on Advanced Computer Information Technologies (ACIT), Wrocław, Poland, 2023, pp. 479-482, doi: 10.1109/ACIT58437.2023.10275448.
15. T. Korobeinikova, V. Maidaniuk, O. Romanyuk, R. Chekhmestruk, O. Romanyuk and S. Romanyuk, "Web-applications Fault Tolerance and Autoscaling Provided by the Combined Method of Databases Scaling," 2022 12th International Conference on Advanced Computer Information Technologies (ACIT), 2022, pp. 27-32, doi: 10.1109/ACIT54803.2022.9913098.
16. Korobeinikova T.I. Vidmovostiikist ta avtomasshtabuvannia veb-resursu / Korobeinikova T.I., Zakharchenko S. M. // International scientific journal «Grail of Science» – 2022. – № 14-15 (May, 2022). – S. 312–319. – ISSN: 2710–3056. ISBN 979-8-88526-799-1.
17. Cherniak I., Pustovit I., Korobeinikova T., Luzhetska N. Algorithm of work of specialized document management system. Information protection and information systems security: Materials of VIII-th International Scientific and Technical Conference, November 11 – 12, 2021. Lviv: NULP, 2021. R. 27-29.