

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Програмної інженерії _____

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти – другий (магістерський)

Дослідження методів проєктування програмних систем для автоматизованої
генерації цифрової звітності підприємства. Автоматизація документообігу
підприємства та цілісність цифрової документації

Виконав: студент 2 курсу, групи ІПЗм-18-4

Малько В.Ю.

спеціальності 121 – Інженерія програмного забезпечення

Освітньо-наукової програми
Інженерія програмного забезпечення

Керівник _____ проф. каф. ІІ Білоус Н.В. _____

Допускається до захисту

Зав. кафедри, проф. _____ З.В. Дудар

2020 р

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наукКафедра Програмної інженерії

Рівень вищої освіти - другий (магістерський)

Спеціальність 121-Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програмаОсвітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«___» _____ 20__ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУстудентові Мальку Владиславу Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів проєктування програмних систем для автоматизованої генерації цифрової звітності підприємства. Автоматизація документообігу підприємства та цілісність цифрової документації

затверджена наказом університету від "27" березня 2020 р № 473 Ст

2. Термін подання студентом роботи до екзаменаційної комісії

19 травня 2020 р.3. Вихідні дані до роботи методи проєктування програмних систем, цифрова звітність, пояснювальна записка, система електронного документообігу.4. Перелік питань, що потрібно опрацювати в роботі дослідити методи проєктування програмних систем для автоматизованої генерації цифрової звітності підприємства, проаналізувати підходи до збереження цілісності електронного документа, спроектувати та розробити систему електронного документообігу.

5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. каф. ПІ Білоус Н.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка*
1	Аналіз предметної галузі	10 квітня 2020 р.	виконано
2	Огляд систем електронного документообігу	15 квітня 2020 р.	виконано
3	Моделювання архітектури програмного забезпечення	20 квітня 2020 р.	виконано
4	Підготовка пояснювальної записки	23 квітня 2020 р.	виконано
5	Спецчастина	01 травня 2020 р.	виконано
6	Підготовка презентації та доповіді	04 травня 2020 р.	виконано
7	Попередній захист	07 травня 2020 р.	виконано
8	Нормоконтроль, рецензування	09 травня 2020 р.	виконано
9	Занесення диплома в електронний архів	12 травня 2020 р.	виконано
10	Допуск до захисту у зав. кафедри	18 травня 2019р.	

Дата видачі завдання _____ 2020 р.

Студент _____ Малько В.Ю.
(підпис)

Керівник роботи _____ проф. каф. ПІ Білоус Н.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Атестаційна робота магістра містить 82 с., 23 рис., 13 джер.

СИСТЕМА ДОКУМЕНТООБІГУ, ЦИФРОВА ЗВІТНІСТЬ, АНАЛІЗ СИСТЕМ, WEB-ПЛАТФОРМА, ГЕНЕРУВАННЯ ДОКУМЕНТІВ, КРОССПЛАТФОРМНІСТЬ.

Об'єкт дослідження – методи проєктування програмних систем для автоматизованої генерації цифрової звітності, підходи до збереження цілісності електронного документу.

Мета роботи - аналіз методів проєктування програмних систем для автоматизованої генерації цифрової звітності підприємства, аналіз підходів до збереження цілісності електронних документів розробка модуля автоматизованої генерації цифрової звітності та захисту електронних документів, а також проєктування, розробка і тестування системи електронного документообігу для малого та середнього бізнесу, яка дозволяє автоматизувати роботу організації чи підрозділу, в рамках якого виконуються бізнес-процеси, буде зручною та інтуїтивно зрозумілою для користувачів, які не є фахівцями роботи з подібними системами, та матиме можливість роботи з інформацією в режимі віддаленого доступу з будь-якої точки світу, а також буде мати невисоку вартість за необмежену кількість користувачів.

Метод рішення – Web-платформа, JetBrains PhpStorm 2020, мова PHP та JavaScript.

У результаті розробки створено систему документообігу, яка дозволяє автоматизувати роботу організації чи підрозділу, в рамках якого виконуються бізнес-процеси, пов'язані з цифровою звітністю.

Додаток розміщений на хмарній платформі, що дозволяє істотно знизити витрати на комп'ютеризацію підприємства.

ABSTRACT

DOCUMENT TRAFFIC SYSTEM, DIGITAL REPORTING, SYSTEM ANALYSIS, WEB-PLATFORM, DOCUMENT GENERATION, CROSS-PLATFORM.

Object of study - methods of designing software systems for automated generation of digital reporting, approaches to preserving the integrity of an electronic document.

Purpose - analysis of software design methods for automated digital reporting of the enterprise, analysis of approaches to maintaining the integrity of electronic documents, development of a module for automated digital reporting and protection of electronic documents, as well as design, development and testing of electronic document management for small and medium businesses. the organization or unit within which the business processes are performed will be convenient and intuitive for the user stuvachiv who are not experts working with similar systems and be able to work with information in remote access from anywhere in the world and will have a low cost for unlimited users.

Solution method - Web platform, JetBrains PhpStorm 2020, PHP language and JavaScript.

As a result of the development, a workflow system has been created to automate the work of an organization or unit within which business processes related to digital reporting are executed.

The application is hosted on a cloud platform, which can significantly reduce the cost of computerization, enterprise.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	7
ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1 Аналіз предметної галузі.....	10
1.2 Аналіз рішень для автоматизації документообігу.....	13
1.3 Постановка задачі	19
2 ДОСЛІДЖЕННЯ ТА АНАЛІЗ МЕТОДІВ ФОРМУВАННЯ ЦИФРОВОЇ ЗВІТНОСТІ ТА ЦІЛІСНОСТІ ІНФОРМАЦІЇ.....	21
2.1 Методи формування документів	21
2.2 Методи збереження цілісності	27
3 ОБҐРУНТУВАННЯ ВИБОРУ ПРОГРАМНИХ ЗАСОБІВ ТА ОПИС ПРИЙНЯТИХ ПРОЄКТНИХ РІШЕНЬ	32
3.1 Опис архітектури та вибір програмних засобів.....	32
3.2 Опис структури та модулів системи	36
4 ФОРМУВАННЯ ЦИФРОВОЇ ЗВІТНОСТІ ПІДПРИЄМСТВА	45
4.1 Формування документів.....	45
4.2 Цілісність цифрової звітності	47
ВИСНОВКИ.....	52
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТОК А ERD Бази даних.....	55
ДОДАТОК Б Фрагменти похідного коду програми.....	56
ДОДАТОК В Заява про реєстрацію авторського права на твір	70
ДОДАТОК Г Слайди презентації	74
ДОДАТОК Д Електронні матеріали	82

ПЕРЕЛІК СКОРОЧЕНЬ

- AJAX – asynchronous JavaScript and XML;
- COM – component object model;
- HTTP – hypertext transfer protocol;
- LAMP – зв’язка Linux + Apache + MySQL + PHP;
- PDM – product data management;
- UML – unified modeling language;
- URL – uniform resource locator;
- WAMP – зв’язка Windows + Apache + MySQL + PHP;
- ЕД – електронний документ;
- КІС – корпоративна інформаційна система;
- СЕД – система електронного документообігу;
- СУБД – система управління базами даних.

ВСТУП

Інформаційні технології відіграють важливу роль у сучасному світі. Їх застосування дає нові можливості для розвитку та оптимізації бізнесу, сприяє розширенню ринків збуту, скороченню витрат, підвищенню продуктивності праці, ефективному використанню ресурсів, підвищенню якості управління бізнесом та надання послуг.

В основі успішності будь-якої компанії лежить правильна стратегія роботи з клієнтами. Бізнес зростає в міру збільшення клієнтської бази та зміцнення відносин зі споживачами. У сучасних умовах роботи практично будь-якої компанії однією з найважливіших є інформація, пов'язана з клієнтом. Поставивши клієнта в центр свого бізнесу, компанія отримує конкурентні переваги у своїй діяльності, істотні перспективи розвитку і значне зниження ризиків, пов'язаних із взаємодією з клієнтом, його потребами і можливостями компанії.

Важливою передумовою завоювання і утримання позицій на ринку для кожної компанії є впровадження системи з автоматизації бізнес-процесів, зокрема системи автоматизації електронного документообігу.

Система автоматизації документообігу – це організаційно-технічна система, що забезпечує процес створення, управління доступом і поширення електронних документів у комп'ютерних мережах, а також забезпечує контроль над потоками документів в організації.

Доцільністю роботи є необхідність систем документообігу на українському ринку, які б мали систему захисту електронних документів, спрощували документообіг на підприємствах, та мали можливість розширення для специфічних задач компанії, незалежно від форми власності. Так як нинішні пропозиції на ринку є пропрієтарними та не дозволяють легко модифікувати програмне забезпечення для своїх специфічних цілей, а також системи є занадто вартісними.

Метою роботи є аналіз методів проєктування програмних систем для автоматизованої генерації цифрової звітності підприємства, аналіз підходів до збереження цілісності електронних документів розробка модуля автоматизованої генерації цифрової звітності та захисту електронних документів, а також проєктування, розробка і тестування системи електронного документообігу для малого та середнього бізнесу, яка дозволяє автоматизувати роботу організації чи підрозділу, в рамках якого виконуються бізнес-процеси, буде зручною та інтуїтивно зрозумілою для користувачів, які не є фахівцями роботи з подібними системами, та матиме можливість роботи з інформацією в режимі віддаленого доступу з будь-якої точки світу, а також буде мати невисоку вартість за необмежену кількість користувачів.

Об'єктом дослідження є методи проєктування та розробки систем документообігу для автоматизованої генерації цифрової звітності, а також методи збереження цілісності електронних документів.

Розроблена система дозволяє спростити роботу з електронною документацією організації завдяки простоті використання, можливості інтеграції та розширення, має модуль захисту документів від підробки та відкритий похідний код.

Розроблена система була апробована в II-му турі Всеукраїнського конкурсу студентських наукових робіт із напрямку «Інформатика і кібернетика». у 2019/2020 н.р у Вінницькому національному технічному університеті, а також подано до міністерства розвитку економіки, торгівлі та сільського господарства України на реєстрацію АП Заява № 93604 від 04 травня 2020р.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

В основу інформаційно-аналітичного забезпечення роботи підприємства передусім покладено опрацювання документів, які підлягають створенню, фіксуванню та обліку в певній формі.

Документи використовуються в різних галузях знань, сферах людської діяльності й суспільного життя. Вони є об'єктом дослідження різних наукових дисциплін, і тому поняття «документ» багатозначне і залежить від того, у якій галузі використовується.

Документ визначається як структурована одиниця інформації, призначена для сприйняття людиною і оформлена та зафіксована на матеріальному носіїві в установленому порядку з дотриманням певної форми її подання та формуванням обов'язкових ознак:

– функціональності – визначеності в межах існуючих суспільних відносин очікуваного впливу документа (в тому числі подальшого порядку дій) на його одержувачів після отримання та ознайомлення із вмістом;

– санкціонованості – визначеності фізичної і/або юридичної особи, яка відповідає у певний час за існування документа;

– реєстрованості – визначеності унікальної ідентифікації документа у множині документів.

Електронний документ (ЕД) – це документ, інформація в якому зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити документа, склад та порядок розміщення яких визначається законодавством країни обігу. При цьому ЕД може бути створений, переданий, збережений і перетворений електронними засобами у візуальну форму подання шляхом відображення даних, які він містить, електронними засобами або на папері у формі, придатній для сприймання його змісту людиною. Обов'язковими реквізитами ЕД є дані, без яких він не може бути підставою для його обліку і не матиме юридичної сили [1].

Документообіг – рух документів в організації з моменту їх створення або отримання до завершення виконання або відправлення; комплекс робіт з документами: прийом, реєстрація, розсилка, контроль виконання, формування справ, зберігання і повторне використання документації, довідкова робота [2].

Основними принципами електронного документообігу є:

- однократна реєстрація документа, що дозволяє однозначно ідентифікувати документ в будь-якій інсталяції даної системи;

- можливість паралельного виконання операцій, що дозволяє скоротити час руху документів і підвищення оперативності їх виконання;

- безперервність руху документа, що дозволяє ідентифікувати відповідального за виконання документа (завдання) в кожен момент часу життя документа (процесу);

- єдина (або погоджено розподілена) база документної інформації, що дозволяє унеможливити дублювання документів;

- ефективно організована система пошуку документа, що дозволяє знаходити документ, володіючи мінімальною інформацією про нього. Якщо автор і адресат у письмовій формі попередньо не домовилися про інше, датою і часом відправлення електронного документа вважаються дата і час, коли відправлення електронного документа не може бути скасовано особою, яка його відправила. У разі відправлення електронного документа шляхом пересилання його на електронному носії, на якому записано цей документ, датою і часом відправлення вважаються дата і час здавання його для пересилання. Вимоги підтвердження факту одержання документа, встановлені законодавством у випадках відправлення документів рекомендованим листом або передавання їх під розписку, не поширюються на електронні документи. У таких випадках підтвердження факту одержання електронних документів здійснюється згідно з вимогами цього закону;

- розвинена система звітності по різних статусах і атрибутах документів, що дозволяє контролювати рух документів по процесах документообігу і приймати управлінські рішення, ґрунтуючись на даних із звітів;

– одержання електронних документів. Електронний документ вважається одержаним адресатом з часу надходження авторіві повідомлення в електронній формі від адресата про одержання цього електронного документа автора, якщо інше не передбачено законодавством або попередньою домовленістю між суб'єктами електронного документообігу. Якщо попередньою домовленістю між суб'єктами електронного документообігу не визначено порядок підтвердження факту одержання електронного документа, таке підтвердження може бути здійснено в будь-якому порядку автоматизованим чи іншим способом в електронній формі або у формі документа на папері. Зазначене підтвердження повинно містити дані про факт і час одержання електронного документа та про відправника цього підтвердження. У разі ненадходження до автора підтвердження про факт одержання цього електронного документа вважається, що електронний документ не одержано адресатом. Якщо автор і адресат у письмовій формі попередньо не домовилися про інше, електронний документ вважається відправленим автором та одержаним адресатом за їх місцезнаходженням (для фізичних осіб – місцем проживання), у тому числі якщо інформаційна, телекомунікаційна, інформаційно-телекомунікаційна система, за допомогою якої одержано документ, знаходиться в іншому місці. Місцезнаходження (місце проживання) сторін визначається відповідно до законодавства;

– перевірка цілісності електронного документа проводиться шляхом перевірки електронного цифрового підпису.

Процес електронного документообігу ґрунтується на інтегрованій електронній обробці обліково-звітної інформації, яка включає формування первинних електронних документів, порядок обробки інформації, автоматизований банк даних тощо. Електронний документообіг покликаний забезпечувати процеси створення, управління доступом і розповсюдження великих обсягів документів у комп'ютерних мережах, а також контроль за рухом документів на підприємстві.

1.2 Аналіз рішень для автоматизації документообігу

Завдання управління документами можна вирішити за допомогою наступних основних видів програмних засобів [4]:

- засобів групової роботи;
- систем управління особливими видами документів (зокрема, PDM-систем);
- спеціальних модулів управління документообігом у складі корпоративних інформаційних систем для підприємств;
- універсальних систем електронного документообігу.

Програмне забезпечення групової роботи (Lotus Notes, MS Exchange, Novell GroupWise) орієнтоване на організацію ефективної взаємодії користувачів. За своєю суттю воно є розширенням програм електронної пошти і не завжди підходить для роботи з крупними архівами документів.

Системи управління особливими видами документів можна умовно розділити на дві категорії:

- у першу категорію входять програмні засоби, що орієнтовані на вузькопрофільне застосування і не мають засобів інтеграції з іншими інформаційними системами. Подібні системи ще є на підприємствах, і, в основному, вони представлені місцевими розробниками. Витрати на підтримку інфраструктури таких систем настільки великі, що вони поступово зживають себе;

- у другу категорію входять спеціалізовані PDM-системи, призначені для управління виробничою інформацією і що мають засоби інтеграції з іншими програмними системами. PDM-системи працюють вже і з електронними образами складних об'єктів (наприклад, кораблів і машин).

Спеціальні модулі управління документообігом містяться у складі переважної більшості корпоративних інформаційних систем (КІС). Проте можливості цих модулів достатньо обмежені, оскільки практично неможливо

створити універсальну і повнофункціональну КІС. До того ж ціна західних КІС настільки велика (сотні тисяч і мільйони доларів), що не кожне підприємство може дозволити собі їх впровадження.

Важливою властивістю універсальних систем електронного документообігу (СЕД) є те, що вони надають можливість комплексного вирішення великого числа завдань управління документами. Крім того, їх впровадження може виявитися дешевшим і ефективнішим.

У більшості випадків СЕД мають шлюзи для роботи з програмним забезпеченням групової роботи і при необхідності можуть використовувати наявні в ньому поштові програми. Слід також відзначити, що в більшості СЕД реалізована інтеграція з найбільш відомими КІС (зокрема, з SAP R/3, Baan, Oracle Applications і ін.). Саме можливість інтеграції з різними додатками є однією з характерних властивостей СЕД. Завдяки наявності цієї властивості, СЕД можуть виступати як сполучна ланка між різними системами, що функціонують на підприємстві, створюючи тим самим основу для організації всього діловодства.

Вибір системи електронного документообігу є важкою задачею, зважаючи на їх різноманітність на ринку. Під час вибору такої системи треба враховувати усі поточні потреби бізнесу та законодавство місцевості, на якій виконуються бізнес-процеси. Детально розглянувши український ринок, було виявлено, що найбільш популярними на даний час системами електронного документообігу є нижче проаналізовані додатки. Слід зазначити, що особливу увагу потрібно приділяти можливості генерування цифрової звітності підприємства та можливості використання системи за допомогою віддаленого доступу з будь-якої точки світу, тобто за допомогою WEB-інтерфейсу.

«М.Е.Дос» (My Electronic Document, також Medoc, Медок) [5] (рис. 1.1) — поширене українське програмне забезпечення для подання звітності до контролюючих органів та обміну юридично значущими первинними документами між контрагентами в електронному вигляді. У середині 2017 року програмою користувалася переважна більшість компаній в Україні.

Програма М.Е.Дос призначена для подачі звітності в усі контролюючі органи України (ДФСУ, ДССУ, ПФУ, ФСС з ТВП, ДКСУ, міністерства і відомства), для реєстрації податкових накладних та юридично значущого електронного документообігу. Крім того, програма надає модулі для нарахування зарплати, звітності великих компаній з розгалуженою структурою підрозділів і роботи банків.

М.Е.Дос містить усі актуальні форми звітності, які оперативно оновлюються відповідно до законодавства. Програма підтримує роботу з ЕЦП найбільш використовуваних центрів сертифікації, а також із захищеними носіями ключів SecureToken.

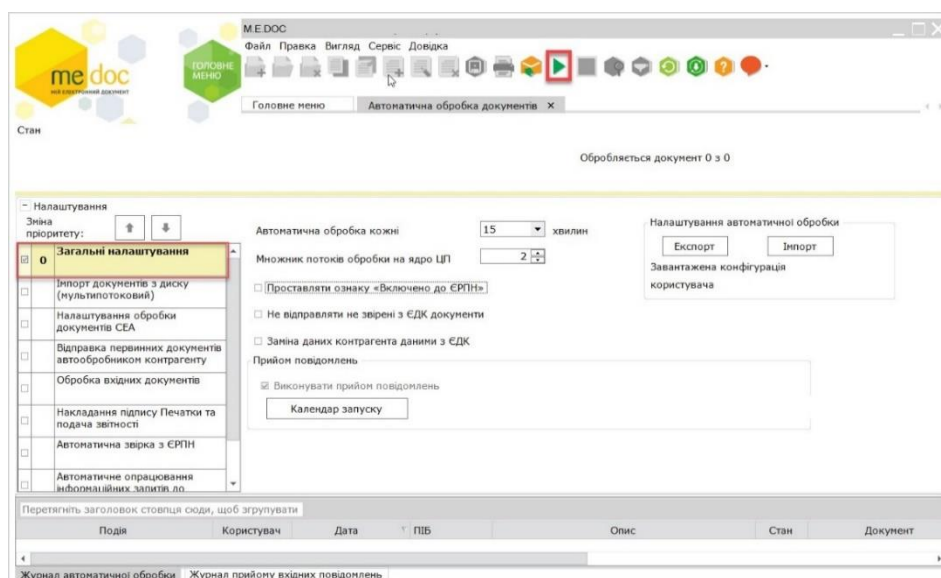


Рисунок 1.1 – Інтерфейс системи М.Е. Дос.

Над програмою працює численний штат розробників, аналітиків, тестувальників, фахівців техпідтримки та інших напрямків. М.Е.Дос отримав позитивний експертний висновок у сфері технічного захисту інформації рівня ГЗ від Держспецзв'язку, що свідчить про високий рівень безпеки і захищеності програми.

У програмі М.Е.Дос можна використовувати сертифікати ЕЦП, отримані в акредитованих центрах сертифікації ключів.

Програма підтримує роботу із захищеними носіями для ЕЦП. У програмі М.Е.Дос також реалізовано автоматичне продовження сертифікатів цифрових підписів.

Розробник програми надає код доступу кожному підприємству. Якщо в програму не введений код доступу, тоді вона буде працювати в демонстраційній версії, яка дозволяє ознайомитися з функціональними можливостями програми, але функції експорту, надсилання електронною поштою й друку документів будуть недоступні.

Система програм «1С: Підприємство 8» (рис. 1.2) [6] – включає в себе платформу і прикладні рішення, розроблені на її основі, для автоматизації діяльності організацій і приватних осіб. Сама платформа не є програмним продуктом для використання кінцевими користувачами, які зазвичай працюють з одним із багатьох прикладних рішень (конфігурацій), розроблених на даній платформі. Такий підхід дозволяє автоматизувати різні види діяльності, використовуючи єдину технологічну платформу.

Гнучкість платформи дозволяє застосовувати «1С: Підприємство 8» в найрізноманітніших галузях:

- автоматизація виробничих і торгових підприємств, бюджетних і фінансових організацій, підприємств сфери обслуговування і т.д.;
- підтримка оперативного управління підприємством;
- автоматизація організаційної і господарської діяльності;
- ведення бухгалтерського обліку з декількома планами рахунків і довільними вимірами обліку, регламентована звітність;
- широкі можливості для управлінського обліку і побудови аналітичної звітності, підтримка багатовалютного обліку;
- рішення задач планування, бюджетування і фінансового аналізу;
- розрахунок зарплати і управління персоналом;
- інші області застосування.

Дата	Номер	Контрагент	Сумма	Валюта	Склад	Вид операции	Организация
15.11.2012	4	Волшебная лань ...	2 076 800.00	руб.	Склад Интернет	Товары, услуги, ко...	Интертрейд ООО ...
29.11.2012	6	Плюснет	1 180.00	руб.	Склад Ромашка	Товары, услуги, ко...	Иванов И. И. ИП
11.01.2013	1	НТЦ "Нанотроника"	119 000.00	руб.	Основной склад	Товары, услуги, ко...	Сервислог
14.01.2013	2	Конфетром	15 000.00	руб.	Основной склад	Товары, услуги, ко...	Сервислог
15.01.2013	1	Торговый дом "Ко...	451 350.00	руб.	Основной склад	Товары, услуги, ко...	Конфетром
17.01.2013	1	ИноТрейд	82 600.00	руб.	Основной склад	Товары, услуги, ко...	Торговый дом "Ко...
22.01.2013	2	Свергумено М.Ф.	172 280.00	руб.	Склад НК2	Товары, услуги, ко...	Торговый дом "Ко...
22.01.2013	3	Штос С.А.	123 605.00	руб.	Основной склад	Товары, услуги, ко...	Торговый дом "Ко...
25.01.2013	4	Магазин 23	109 150.00	руб.	Основной склад	Товары, услуги, ко...	Торговый дом "Ко...
28.01.2013	5	Никитаева И.В.	135 000.00	руб.	Склад НК3	Товары, услуги, ко...	Торговый дом "Ко...
29.01.2013	3	Торговый дом "Ко...	15 000.00	руб.	Основной склад	Товары, услуги, ко...	Сервислог
31.01.2013	1	Свергумено М.Ф.	38 000.00	руб.	Основной склад	Товары, услуги, ко...	Свердцев П. А. ИП
02.02.2013	6	Магазин 23	196 000.00	руб.	Основной склад	Товары, услуги, ко...	Торговый дом "Ко...
11.02.2013	2	Торговый дом "Ко...	525 000.00	руб.	Основной склад	Товары, услуги, ко...	Конфетром
20.02.2013	1	Монолит	900 000.00	руб.	Основной склад	Товары, услуги, ко...	Комфорт-сервис
26.02.2013	2	Кабла Глобал	15 000.00	руб.	Основной склад	Товары, услуги, ко...	Свердцев П. А. ИП

Рисунок 1.2 – Интерфейс системи «1С:Підприємство 8».

FossDoc (рис. 1.3) [7] — рішення на платформі FossLook, призначене для створення електронного архіву документів, організації корпоративного документообігу (workflow) і автоматизації бізнес-процесів на підприємствах, в установах і організаціях будь-якого роду діяльності. Програма дозволяє вирішити велику кількість завдань, реалізація яких покладена на відповідні модулі. Система може бути легко перелаштована з урахуванням специфіки роботи кожного конкретного підприємства.

Система автоматизує всі аспекти сучасного діловодства: створення реєстраційно-контрольних карток, РКК (карток документа); відправку доручень (аналог "бігунка" в звичайному документообігу); облік паперових оригіналів за допомогою спеціальних журналів; здійснення контролю над виконанням документів; підготовку резолюцій; роботу з електронно-цифровим підписом, генерацію звітів.

Усе різноманіття документів, з якими працюють користувачі, розділяється на окремі категорії - типи документів. У системі існують зумовлені типи документів (поширені у вітчизняному діловодстві): вхідні та вихідні листи, звернення громадян, службові записки, накази і т.д. Документи можуть посилатися на інші документи або бути дочірніми по відношенню до головних. Поля документів можуть заповнюватися значеннями з довідників.

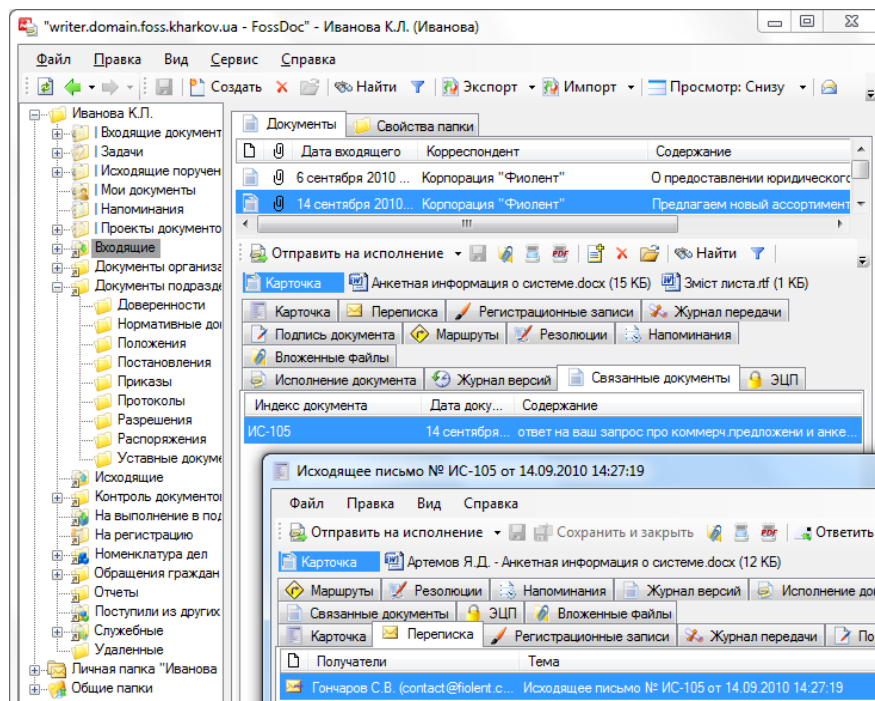


Рисунок 1.3 – Интерфейс системы FossDoc.

У проаналізованих системах є забагато зайвого функціоналу, який не є потрібним для 80% малого та середнього бізнесу в Україні. Також подібні системи є занадто дорогими для деяких сфер бізнесу та здебільшого не підтримують роботу в режимі віддаленого доступу та не є доволі зручними. Через високу вартість даних систем та обмеження кількості користувачів системи підприємства вимушені використовувати неліцензійні копії програмного забезпечення, порушуючи чинне законодавство. Тому було прийнято рішення розробки системи електронного документообігу, яка буде зручною та інтуїтивно зрозумілою для користувачів, які не є фахівцями роботи з подібними системами, та матиме можливість роботи з інформацією в режимі віддаленого доступу з будь-якої точки світу, а також буде мати невисоку вартість за необмежену кількість користувачів та не матиме надлишкового функціоналу.

Дана система буде актуальною на українському ринку програмного забезпечення, адже систем, які мають можливість доступу в режимі онлайн, на

українському ринку дуже мало та досить часто такі системи мають занадто велику вартість та часто розробляються під замовлення підприємства.

1.3 Постановка задачі

На основі аналізу предметної галузі та ринку можна чітко сформулювати постанову задачі та всі необхідні вимоги до програмного продукту, який буде створено.

Метою роботи є аналіз методів проектування програмних систем для автоматизованої генерації цифрової звітності підприємства, аналіз підходів до збереження цілісності електронних документів розробка модуля автоматизованої генерації цифрової звітності та захисту електронних документів, а також проектування, розробка і тестування системи електронного документообігу для малого та середнього бізнесу, яка дозволяє автоматизувати роботу організації чи підрозділу, в рамках якого виконуються бізнес-процеси, буде зручною та інтуїтивно зрозумілою для користувачів, які не є фахівцями роботи з подібними системами, та матиме можливість роботи з інформацією в режимі віддаленого доступу з будь-якої точки світу, а також буде мати невисоку вартість за необмежену кількість користувачів та повинна бути розроблена з використанням офіційних рекомендацій розробки під web-платформу.

Система, що розробляється, надаватиме користувачеві можливість генерації цифрової документації підприємства (договори, рахунки, накладні, специфікації тощо).

Технології, які будуть використанні при розробці системи електронного документообігу:

- мова програмування PHP 7.2;
- мова програмування JavaScript;
- мова розмітки гіпертексту HTML;

- каскадна таблиця стилів CSS;
- веб-сервер Apache;
- веб-сервер Nginx;
- планувальник задач Cron
- клієнтський фреймворк Twitter Bootstrap 4;
- середовище розробки PHP Storm 2020.

Основною перевагою використання саме цього стеку технологій є те, що вони не потребують потужного обладнання та система буде кроссплатформною [8]. Виходячи з того, що PHP є однією з найпопулярніших мов програмування багатоплатформових веб-застосунків, було прийнято рішення використовувати її для подальшої розробки системи документообігу для можливості використання її на різних платформах (комп'ютери, ноутбуки, нетбуки, планшети, смартфони, телевізори тощо).

2 ДОСЛІДЖЕННЯ ТА АНАЛІЗ МЕТОДІВ ФОРМУВАННЯ ЦИФРОВОЇ ЗВІТНОСТІ ТА ЦІЛІСНОСТІ ІНФОРМАЦІЇ

2.1 Методи формування документів

У сучасних умовах з розвитком інформаційного суспільства, введенням електронного документообігу все частіше виникають передумови до створення єдиного електронного архіву. Уже в даний час все більша частина документації створюється в електронній формі, і з плином часу обсяг електронних документів (ЕД) буде тільки зростати. Здатність їх прийняти, зберегти і використовувати стає життєво важливим завданням для ефективного державного управління. У зв'язку з цим архівам необхідно шукати шляхи організації роботи з ЕД. Однак функціонування електронних архівів по суті починається з моменту формування електронних справ, причому дана процедура не має єдиного науково-методичного апарату, що задовольняє вимогам керівних документів, що стало важливим науково-технічним завданням, для вирішення якого потрібна розробка методики формування цифрової звітності.

З метою забезпечення оперативного пошуку, полегшення роботи і фізичного збереження всі носії інформації, що були розроблені або поступили в організацію з числа вихідних і вхідних документів, після їх виконання підлягають розподілу в справі.

Справа - сукупність документів або окремих документів, що відносяться до одного питання або ділянки діяльності, поміщені в окрему обкладинку.

Виконані вихідні і вхідні документи розподіляються в справі. Справи заводяться за предметною (тематичною) ознакою з урахуванням важливості документів і термінів їх архівного зберігання.

Недотримання цього правила може призвести до великої і складної роботи при підготовці та здачі справ в архів, а також при відборі для знищення архівних справ з вичерпаними термінами зберігання, що підлягають знищенню, доведеться витягати документи, терміни зберігання яких ще не закінчилися.

Тенденції розвитку сучасних інформаційних технологій характеризуються переходом до хмарних обчислень та зв'язку з автоматизованою обробкою даних.

Якщо розглянути даний процес з точки зору роботи з електронними документами, то він передбачає перехід від роботи з офісними додатками на локальній машині користувача до роботи з хмарними веб-сервісами, в яких відбувається заповнення, обробка і генерація готових текстових документів, електронних таблиць та ін.

Документи, які раніше створювалися, редагувалися і зберігалися локально, зараз зберігаються на віддалених мережових дисках.

Розглянемо коло вирішуваних завдань при автоматизованій обробці даних електронних документів.

Перше завдання пов'язане з персоналізацією електронного документа, тобто адаптацією електронного документа до конкретного користувача, персоналізований електронний документ формується шляхом заповнення заданого макета індивідуальною користувальницькою інформацією, являє собою кінцевий інформаційний продукт і не передбачає внесення нової індивідуальної інформації в документ. При вирішенні даного завдання процедура взаємодії сервера з користувачем є односпрямованою. Прикладами таких завдань є формування наступних видів документів: рахунок, накладна, договір.

Друге завдання пов'язане з багаторазовим використанням проміжного документа, який модифікується користувачем і передається назад на сервер. Таке завдання діє у двох напрямках і призначене для отримання додаткової інформації від користувача і її подальшого збереження на сервері. Наприклад, при створенні анкет формується документ, який передається користувачеві. У передбачені в документі поля користувач вносить свої дані і передає документ назад на сервер. Отримані дані зберігаються на сервері разом з наявною інформацією даного користувача і можуть бути використані при генерації інших документів.

Перше і друге завдання автоматизованої обробки даних припускають роботу з шаблоном електронного документа.

Шаблон являє собою форматований текстовий документ, в якому ділянку

змінюваних даних замінено на послідовність спеціальних символів, що дозволяють однозначно визначити цю ділянку.

При вирішенні першого завдання виконується запис даних в шаблон, а при вирішенні другого завдання здійснюється і запис, і читання даних. Редактор текстових документів Microsoft Word є одним з найбільш популярних текстових процесорів на сьогоднішній день. Тому в роботі розглянуті методи автоматизованої обробки текстових документів Microsoft Word:

а) COM-об'єкти. COM (Component Object Model – об'єктна модель компонентів) – технологія від компанії Microsoft, яка полягає в створенні програмного забезпечення на основі різних взаємодіючих компонентів, кожен з яких доступний для використання багатьма програмами і процесами. У сучасних версіях Windows COM використовується дуже широко. На основі COM були створені технології: Microsoft OLE Automation, ActiveX і інші. Дана технологія дозволяє створювати і редагувати документи будь-якої складності, для роботи використовується мова програмування Visual Basic for Applications (VBA) і Microsoft Office API. Недоліками даної технології є:

- 1) необхідність установки пакета Microsoft Office і, відповідно, необхідність використання операційної системи із сімейства Windows;
- 2) низька швидкість створення або відкриття нового документа, тому що робота здійснюється через запуск Microsoft Word у фоновому режимі;
- 3) велика надмірність коду навіть при створенні простих документів.

б) об'єкти .NET. Для роботи з цими об'єктами на мові програмування PHP потрібна установка програмної платформи .NET Framework. Клас DOTNET дозволяє створювати об'єкти зі збірок .NET, викликати їх методи і використовувати їх властивості. Дане рішення повторює недоліки першого методу, але значно підвищує читаність коду, що відповідає за обробку документа за рахунок використання об'єктного підходу .NET. Іншою перевагою даного методу є переносимість коду, що відповідає за генерацію

документа, між програмами, написаними на різних мовах програмування, але на єдиній платформі .NET;

в) бібліотека PHPWord. Бібліотека дозволяє створювати нові документи, редагувати шаблони документів і має наступні можливості:

- 1) вставка і форматування розділів документа;
- 2) вставка і форматування тексту;
- 3) вставка розривів тексту;
- 4) вставка розривів сторінок;
- 5) вставка і форматування зображень і бінарних OLE-об'єктів;
- 6) вставка і форматування водяних знаків;
- 7) вставка колонтитулів;
- 8) вставка і форматування таблиць;
- 9) вставка власних назв і таблиць змісту;
- 10) вставка і форматування списку елементів;
- 11) вставка і редагування гіперпосилань;
- 12) робота з шаблонами.

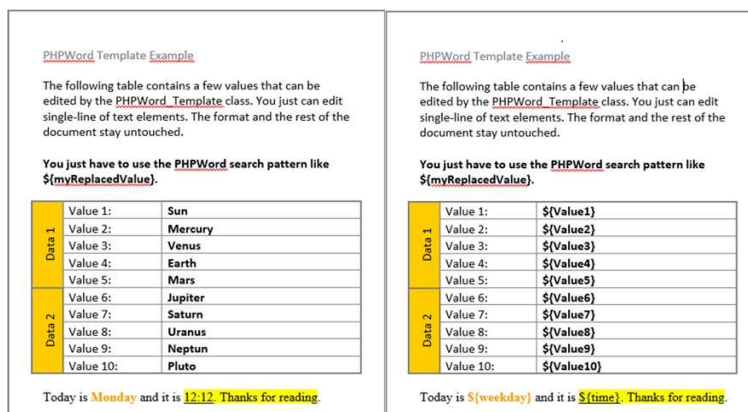


Рисунок 2.1 – Шаблон (зліва) та документ з даними (справа), сформований за допомогою PHPWord

Для роботи бібліотеки потрібно PHP версії 5.2 або новіше, розширення `xmlwriter` та `ziparchive`. Її недоліком є підтримка роботи тільки з файлами формату DOCX.

При завантаженні шаблону PHPWord дозволяє замінювати введені в форматі \$ {var1} змінні на реальні строкові дані.

На рисунку 2.1. показані шаблон документа і документ з даними, сформований з використанням бібліотеки PHPWord.

Недоліком при роботі з шаблонами в бібліотеці PHPWord є, наприклад, відсутність можливості заміни зображень. Тому запропонований варіант дозволяє як працювати зі змінними рядками в шаблоні за аналогією з бібліотекою PHPWord, так і здійснювати заміну зображень.

г) власна реалізація автоматизованої обробки текстових документів. Документи Microsoft Word з розширенням DOC є бінарними файлами, а документи з розширенням DOCX є файлами у відкритому форматі Office Open XML. Office Open XML – серія форматів файлів для зберігання електронних документів пакетів офісних додатків, зокрема Microsoft Office. Формат є zip-архівом, що містить текст у вигляді XML, графіка та інші дані.

Виходячи з вищесказаного, можна відзначити, що робота з документами у форматі DOC ускладнена через відсутність документації про структуру файлу. Робота з файлами DOCX, навпаки, досить проста за рахунок відкритості формату.

Єдиною вимогою для роботи з документами з розширенням DOCX, як і для роботи бібліотеки PHPWord, є наявність розширення ZipArchive. Розглянемо скрипт, який виконує заміну шести строкових змінних і одного зображення. Текст документа зберігається в XML-файлі word/document.xml, заміна здійснюється строковою функцією str_replace без використання спеціалізованих бібліотек для роботи з XML. Заміна зображення здійснюється видаленням вихідного і додаванням нового зображення в zip-архів під колишнім ім'ям. Зображення зберігаються в папці word/media. Файл "my_photo.png" замінює поточний файл "image1.png". Далі наведено вихідний код скрипту.

```
<?php
define('ROOTPATH', getcwd());
if( is_file(ROOTPATH . '/tmp/template_copy.docx') )
{
    unlink(ROOTPATH . '/tmp/template_copy.docx');
```

```

}
copy(ROOTPATH . '/template.docx', ROOTPATH . '/tmp/template_copy.docx');
$zip = new ZipArchive();
if( $zip->open(ROOTPATH . '/tmp/template_copy.docx') )
{
    $xmlstr = $zip->getFromName('word/document.xml');
    $xmlstr = str_replace('${FIO}', 'Малько Владислав Юрійович', $xmlstr);
    $xmlstr = str_replace('${AGE}', 23, $xmlstr);
    $xmlstr = str_replace('${EDUCATION}', 'вища', $xmlstr);
    $xmlstr = str_replace('${WORK_PLACE}', 'ХНУРЕ', $xmlstr);
    $xmlstr = str_replace('${POST}', 'магістр', $xmlstr);
    $xmlstr = str_replace('${SKILLS}', 'Автоматизація, PHP, MySQL,
JavaScript', $xmlstr);
    $zip->addFromString('word/document.xml', $xmlstr);
    $zip->addFile(ROOTPATH . '/my_photo.png', 'word/media/image1.png');
    $zip->close();
}
?>

```

Результатом роботи даного скрипту є створення нового файлу з даними, який знаходиться за адресою “/tmp/template_copy.docx”.

Представлений приклад не є універсальним засобом для роботи з шаблонами документів Microsoft Word. Однак він показує простоту створення і обробки документів у форматі Office Open XML. Цей формат, крім текстових документів, включає зберігання електронних таблиць і презентацій.

Варто відзначити, що Office Open XML є форматом за замовчуванням для додатків Microsoft Office 2007 і більш пізніх. Однак часто при документообігу зустрічаються документи з розширенням DOC, створені в пакеті Microsoft Office. Третій і четвертий методи рішення не підтримують роботу з такими документами. У цьому випадку можна використовувати конвертер, який спочатку перетворює файл з розширенням DOC в файл формату Office Open XML з розширенням DOCX. В якості конвертера може використовуватися пакет OpenOffice, звернення до якого здійснюється з командного рядка.

Для розробки системи генерування цифрової звітності підприємства було обрано бібліотеку PHPWord, так як бібліотека підтримує повністю мануальний

режим формування документів, що буде корисним при формуванні специфічних документів підприємства, які не підлягають шаблонізації.

2.2 Методи збереження цілісності

Алгоритм MD5 – 128-бітний алгоритм хешування даних, призначений для створення «відбитків» або «дайджестів» повідомлень довільної довжини та подальшої перевірки їх дійсності. Принцип роботи алгоритму наведено на рисунку 2.2.

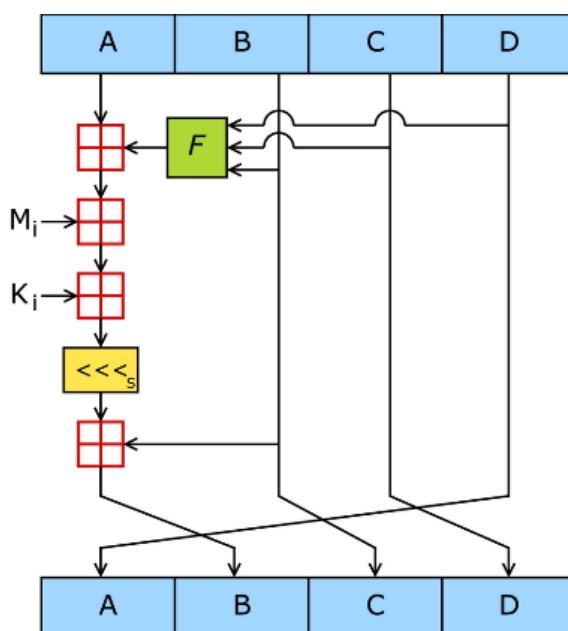


Рисунок 2.2– Схема криптографічного алгоритму хешування даних MD5

Спочатку на вхід алгоритму надходить вхідний потік даних, хеш якого необхідно знайти. Довжина повідомлення вимірюється в бітах і може бути будь-якою (в тому числі нульовою).

Запишемо довжину повідомлення в L . Це число ціле і невід'ємне. Кратність будь-яким числам необов'язкова. Після надходження даних йде процес підготовки потоку до обчислень. Далі проводиться хешування інформації в 5 етапів:

2.2.1 Вирівнювання потоку

Спочатку до кінця потоку дописують одиничний біт. Потім додають певну кількість нульових біт так, щоб нова довжина потоку L' стала порівнянна з 448 по модулю 512, ($L' = 512 \times N + 448$). Вирівнювання відбувається в будь-якому випадку, навіть якщо довжина вихідного потоку вже порівнянна з 448.

2.2.2 Додавання довжини повідомлення

В кінець повідомлення дописують 64-бітове представлення довжини даних (кількість біт в повідомленні) до вирівнювання. Спочатку записують молодші 4 байти, потім старші. Якщо довжина перевершує $2^{64} - 1$, то дописують тільки молодші біти (еквівалентно взяттю по модулю 2^{64}). Після цього довжина потоку стане кратною 512. Обчислення будуть ґрунтуватися на уявленні цього потоку даних у вигляді масиву слів по 512 біт.

2.2.3 Ініціалізація буфера

Для обчислень ініціалізуються 4 змінних розміром по 32 біта і задаються початкові значення шістнадцятирічними числами (порядок байтів little-endian, спочатку молодший байт):

а) $A = 01\ 23\ 45\ 67; //\ 67452301h;$

б) $B = 89\ AB\ CD\ EF; //\ EFC DAB89h;$

в) $C = FE\ DC\ BA\ 98; //\ 98BADCFEh;$

г) $D = 76\ 54\ 32\ 10; //\ 10325476h;$

У цих змінних будуть зберігатися результати проміжних обчислень. Початковий стан ABCD називається ініціалізувальним вектором.

2.2.4 Обчислення в циклі

Визначимо функції і константи, які знадобляться нам для обчислень:

а) Для кожного раунду буде потрібна своя функція. Введемо функції від трьох параметрів-слів, результатом також буде слово:

$$1) \text{ 1-й раунд: } FunF(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z);$$

$$2) \text{ 2-й раунд: } FunG(X, Y, Z) = (X \wedge Z) \vee (\neg Z \wedge Y);$$

$$3) \text{ 3-й раунд: } FunH(X, Y, Z) = (X \oplus Y \oplus Z);$$

$$4) \text{ 4-й раунд: } FunI(X, Y, Z) = Y \oplus (\neg Z \vee X);$$

де \oplus , \wedge , \vee , \neg побітові логічні операції XOR, AND, OR і NOT відповідно.

б) Визначимо таблицю констант $T[1 \dots 64]$ – 64-елементна таблиця даних, побудована наступним чином: $T[n] = \text{int}(2^{32} \cdot |\sin n|)$.

в) Кожен 512-бітний блок проходить 4 етапи обчислень по 16 раундів. Для цього блок представляється у вигляді масиву X з 16 слів по 32 біта. Всі раунди однотипні і мають вигляд: $[abcd \text{ ksi}]$, який визначається як $a = b + ((a + Fun(b, c, d,) + X[k] + T[i]) \lll s$, де k - номер 32-бітного слова з поточного 512-бітного блоку повідомлення і $\dots \lll s$ – циклічний зсув вліво на s біт отриманого 32-бітного аргументу. Число s задається окремо для кожного раунду.

Заносимо в блок даних елемент n з масиву 512-бітних блоків. Зберігаються значення A , B , C і D , що залишилися після операцій над попередніми блоками (або їх початкові значення, якщо блок перший):

$$а) AA = A;$$

$$б) BB = B;$$

$$в) CC = C;$$

г) $DD = D$;

2.2.4.1 Этап 1

```
/* [abcd k s i] a = b + ((a + F(b, c, d) + X[k] + T[i]) <<< s).*/
[ABCD 0 7 1][DABC 1 12 2][CDAB 2 17 3][BCDA 3 22 4]
[ABCD 4 7 5][DABC 5 12 6][CDAB 6 17 7][BCDA 7 22 8]
[ABCD 8 7 9][DABC 9 12 10][CDAB 10 17 11][BCDA 11 22 12]
[ABCD 12 7 13][DABC 13 12 14][CDAB 14 17 15][BCDA 15 22 16]
```

2.2.4.2 Этап 2

```
/* [abcd k s i] a = b + ((a + G(b, c, d) + X[k] + T[i]) <<< s).*/
[ABCD 1 5 17][DABC 6 9 18][CDAB 11 14 19][BCDA 0 20 20]
[ABCD 5 5 21][DABC 10 9 22][CDAB 15 14 23][BCDA 4 20 24]
[ABCD 9 5 25][DABC 14 9 26][CDAB 3 14 27][BCDA 8 20 28]
[ABCD 13 5 29][DABC 2 9 30][CDAB 7 14 31][BCDA 12 20 32]
```

2.2.4.3 Этап 3

```
/* [abcd k s i] a = b + ((a + H(b, c, d) + X[k] + T[i]) <<< s).*/
[ABCD 5 4 33][DABC 8 11 34][CDAB 11 16 35][BCDA 14 23 36]
[ABCD 1 4 37][DABC 4 11 38][CDAB 7 16 39][BCDA 10 23 40]
[ABCD 13 4 41][DABC 0 11 42][CDAB 3 16 43][BCDA 6 23 44]
```

[ABCD 9 4 45][DABC 12 11 46][CDAB 15 16 47][BCDA 2 23 48]

2.2.4.4 Етап 4

/ [abcd k s i] a = b + ((a + I(b, c, d) + X[k] + T[i]) <<< s).*/*

[ABCD 0 6 49][DABC 7 10 50][CDAB 14 15 51][BCDA 5 21 52]

[ABCD 12 6 53][DABC 3 10 54][CDAB 10 15 55][BCDA 1 21 56]

[ABCD 8 6 57][DABC 15 10 58][CDAB 6 15 59][BCDA 13 21 60]

[ABCD 4 6 61][DABC 11 10 62][CDAB 2 15 63][BCDA 9 21 64]

Підсумовуємо з результатом попереднього циклу:

а) $A = AA + A$

б) $B = BB + B$

в) $C = CC + C$

г) $D = DD + D$

Після закінчення циклу необхідно перевірити, чи є ще блоки для обчислень. Якщо так, то переходимо до наступного елементу масиву ($n + 1$) і повторюємо цикл.

2.2.5 Результат обчислень

Результат обчислень знаходиться в буфері ABCD, це і є хеш. Якщо виводити побайтово, починаючи з молодшого байта A і закінчивши старшим байтом D, то ми отримаємо MD5-хеш. 1, 0, 15, 34, 17, 18 ...

3 ОБҐРУНТУВАННЯ ВИБОРУ ПРОГРАМНИХ ЗАСОБІВ ТА ОПИС ПРИЙНЯТИХ ПРОЄКТНИХ РІШЕНЬ

3.1 Опис архітектури та вибір програмних засобів

3.1.1 Системні та функціональні вимоги

Спочатку, для того щоб почати моделювання та проєктування системи, потрібно визначити обмеження предметної галузі цієї системи. Система, яку буде створено, є веб-додатком. Це є найпершим обмеженням нашої системи, адже її не можна буде використовувати без веб-сервера. Також програма буде безпосередньо працювати в інтернет-браузері користувача. Це також є обов'язковими обмеженнями, адже для того щоб система працювала на пристрої користувача, потрібно задовольняти наступні умови системи:

а) Рекомендовані системні вимоги для серверної машини:

- 1) операційна система Windows Server 2016 і вище, Debian 8 і вище, CentOS 7 і вище, Red Hat Enterprise Linux 7 і вище, Ubuntu 14.04 і вище;
- 2) веб-сервер Apache 2.4 і вище або Nginx 1.17 і вище;
- 3) PHP 7.2 і вище;
- 4) MySQL 5.8 і вище;
- 5) процесор Intel Core i3 або новіше;
- 6) RAM 4 GB і більше.

б) Мінімальні системні вимоги для клієнтської машини:

- 1) операційна система Windows або Linux;
- 2) процесор Intel Pentium Gold або новіше (для десктопних пристроїв);
- 3) RAM 2 GB і більше;
- 4) дозвіл на використання JavaScript в браузері;
- 5) браузер Chrome 64.0 і вище, Safari 11.0 і вище, Opera 51.0 і вище, Mozilla Firefox 58.0 і вище, Microsoft Edge 41 і вище.

в) Рекомендовані системні вимоги для клієнтської машини:

- 1) операційна система Windows або Linux;
- 2) процесор Intel Core i3 або новіше (для десктопних пристроїв);
- 3) RAM 4 GB і більше;
- 4) дозвіл на використання JavaScript в браузері;
- 5) браузер Chrome 64.0 і вище, Safari 11.0 і вище, Opera 51.0 і вище, Mozilla Firefox 58.0 і вище, Microsoft Edge 41 і вище.

г) Крім програмних і системних обмежень є обмеження зовнішніх факторів, пов'язаних з нею:

- 1) швидкість інтернет-з'єднання 512 Кбіт/с і вище.

Отже, в якості сховища даних була обрана СУБД MySQL, так як вона є найбільш популярною для роботи з PHP [9]. Систему було вирішено створити у вигляді веб-додатку (сайту), який має дві ролі користувачів: користувач та адміністратор. Після вводу свого логіну і паролю користувачам та адміністратору надається доступ до функціоналу системи відповідно до його дозволів та прав у системі.

Для того щоб розуміти, що в системі електронного документообігу повинно бути реалізовано, складемо функціональні вимоги до системи.

Основна функціональність даної системи полягає у наступному:

- зберігання інформації про контрагентів;
- генерація електронних договорів;
- генерація електронних актів;
- генерація електронних рахунків;
- генерація електронних видаткових накладних;
- генерація електронних прибуткових накладних;
- генерація електронних специфікацій;
- формування захисних ключів для згенерованих документів.

При створенні додатку повинно бути взято до уваги:

- зручність використання додатку – інтерфейс має бути зручним та інтуїтивно зрозумілим, не вимагати додаткової підготовки користувачів;

- простота доступу до необхідної інформації;
- зручність генерації документації в системі;
- можливість перевірки документів на дійсність.

Підводячи підсумок, можна зібрати до купи всі обмеження предметної області програмного продукту. Користувач повинен мати пристрій на базі операційної системи Windows Server 2016 і вище або Linux, з процесором Intel Core i3 або новіше, та має дозволити використовувати JavaScript в браузері. Також програмне забезпечення потрібно використовувати на WAMP або LAMP сервері, щоб продукт працював як треба. У подальшому, із розвитком інформаційних технологій, деякі із цих пунктів можна буде виключити із списку обмежень системи [10].

3.1.2 Прототипування інтерфейсу

Після постановки задачі та визначення системних і функціональних вимог до системи переходимо до її схематичного прототипування.

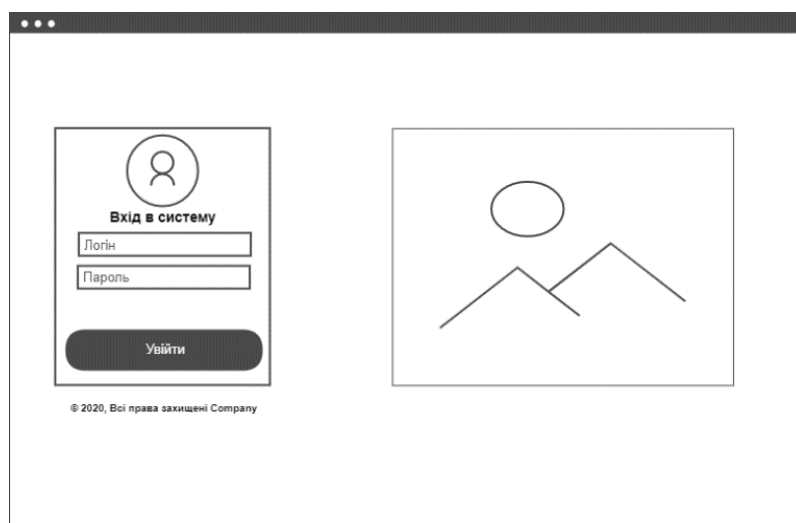


Рисунок 3.1 – Графічний прототип системи. Сторінка автентифікації.

Спочатку побудуємо графічний прототип сторінки автентифікації в системі (рис. 3.1). На рисунку ми бачимо, що сторінка має форму входу в систему за допомогою даних користувача, а саме логіну та паролю.

Також побудуємо графічний прототип основного інтерфейсу нашої системи документообігу (рис. 3.2).



Рисунок 3.2 – Графічний прототип системи. Основний інтерфейс.

На прототипі основного графічного інтерфейсу зображено основну концепцію дизайну системи та місцезнаходження блоків з контентом системи.

Система документообігу, що розробляється, – це web-орієнтований застосунок, який матиме адаптивний та інтуїтивно зрозумілий користувачу інтерфейс.

3.1.3 Опис прийнятих програмних рішень для реалізації

При створенні системи було вирішено використовувати технологію AJAX та клієнтський фреймворк Bootstrap 4 для комфортної роботи із системою.

AJAX (Asynchronous JavaScript And XML) – підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані. AJAX – один з компонентів концепції DHTML.

Bootstrap – фреймворк, набір HTML + CSS інструментів і шаблонів для верстки та більш ефективного і швидкого створення веб-додатків.

Bootstrap – сучасний помічник розробників інтерфейсів, дизайнерів і веб-майстрів, доступний для використання за відкритою ліцензією. Цей фреймворк дуже динамічний і регулярно оновлюваний, тому не всі його функції можуть коректно підтримуватися старими браузерами.

Bootstrap зараз на піку популярності і можна впевнено сказати, що на його основі можна зробити будь-який веб-інтерфейс.

3.2 Опис структури та модулів системи

3.2.1. UML проєктування системи

Існує безліч варіантів проєктування програмного забезпечення, але найвлучніший – проєктування системи з використанням мови UML.

Як зазначалося раніше, UML – мова графічного опису для об'єктного моделювання у галузі розробки програмного забезпечення, системного проєктування та відображення організаційних структур.

UML дозволяє розробникам та архітекторам програмного забезпечення досягти узгодження у графічному відображенні та позначенні загальних понять та сконцентруватися саме на проєктуванні та архітектурі програмного продукту або системи в цілому.

Для того щоб спроектувати систему, потрібно:

- проаналізувати вимоги майбутнього користувача програмного забезпечення;
- проаналізувати існуючі стандарти та методи у розробці програмного забезпечення;
- проаналізувати обмеження майбутньої системи, які залежать від існуючих недоліків та технологій;
- створити діаграму компонентів майбутньої системи чи програмного забезпечення;
- створити діаграму класів майбутньої системи чи програмного забезпечення;
- створити діаграму послідовності майбутньої системи чи програмного забезпечення;
- створити діаграму варіантів використання системи чи програмного забезпечення для остаточного користувача.

Надалі буде проаналізовано майбутню систему як програмний продукт та побудовано необхідні діаграми, які буде використано при створенні програмного забезпечення.

Діаграма варіантів використання в UML – діаграма, що відображає відносини між акторами і прецедентами і є складовою частиною моделі прецедентів, що дозволяє описати систему на концептуальному рівні.

Система «FLUSSO» має двох акторів, які є користувачами цієї системи:

- користувач;
- адміністратор.

У кожного з акторів є декілька прецедентів використання системи.

Користувач в системі може використовувати систему для таких цілей:

- управління документами;
- управління клієнтами;
- управління постачальниками;
- верифікація документів;

– управління товарами.

Адміністратор може використовувати систему для:

– управління документами;

– управління клієнтами;

– управління постачальниками;

– верифікації документів;

– управління товарами;

– управління API системами;

– налаштування системи;

– управління користувачами.

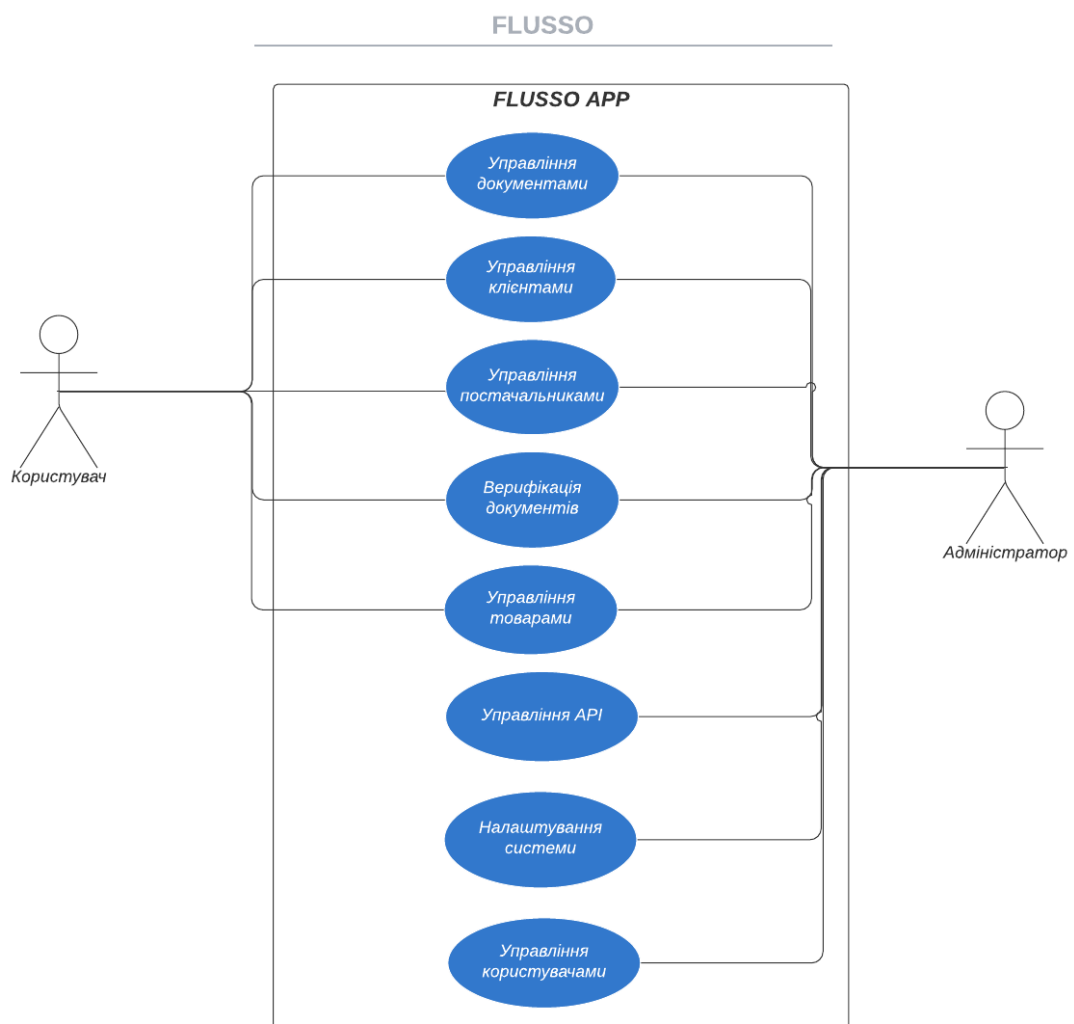


Рисунок 3.3 – UML проєктування ПЗ. Діаграма варіантів використання системи

Побудована діаграма варіантів використання системи (рис. 3.3) дозволяє мати картину системи в цілому та визначити, який саме функціонал та варіанти дій будуть можливі для користувача.

Ця діаграма в подальшому буде використана перш за все як документ, який визначає функціональні вимоги до системи, та може бути використана для керування всім процесом розробки програмного забезпечення. Усі основні види діяльності, такі як аналіз, проєктування та тестування, виконуються на підставі варіантів використання системи.

Під час аналізу і проєктування варіанти використання дозволяють зрозуміти як результати, які хоче отримати користувач, впливають на архітектуру системи і як повинні поводитися компоненти системи, для того щоб реалізувати потрібну для користувача функціональність.

У процесі тестування описані раніше варіанти використання дозволяють простіше оцінити точність реалізації вимог користувачів і дозволяють провести покрокову перевірку цих вимог.

Стратегія використання прецедентів при визначенні вимог визначає необхідність додатково до питання "що користувачі чекають від системи?" задавати питання "що система повинна зробити для користувачів?". Такий підхід дозволяє шукати функції, які потрібні багатьом користувачам, і виключати ті можливості, які не можуть допомогти користувачам виконувати свої повсякденні завдання.

3.2.2 Проєктування архітектури програмного забезпечення

Для того щоб мати уявлення про майбутню систему, потрібно проаналізувати середовище, у якому ця система буде створюватися та працювати.

Цю задачу вирішує діаграма компонентів системи.

Діаграма компонентів – статична структурна діаграма, яка зображує розбиття програмної системи на структурні компоненти та зв'язок між компонентами. У якості компонентів можуть виступати файли, бібліотеки, модулі, пакети та файли, що виконуються.

Компоненти діаграми зв'язуються через залежності, коли з'єднується необхідний інтерфейс одного компонента з наявним інтерфейсом іншого компонента. Таким чином ілюструються відносини «клієнт – джерело» між двома компонентами.

Залежність показує, що один компонент надає сервіс, необхідний іншому компоненту. Залежність зображується стрілкою від інтерфейсу або порту клієнта до імпортованого інтерфейсу.

Коли діаграма компонентів використовується, щоб показати внутрішню структуру компонентів, інтерфейси, що надаються та використовуються у складеному компоненті, можуть делегуватися до відповідних інтерфейсів внутрішніх компонентів.

Перед створенням діаграми компонентів слід дотримуватися наступних правил:

- перед розробкою діаграм компонентів необхідно вирішити, з яких фізичних частин буде складатися програмна система. При цьому повинно бути вирішено два завдання – розподіл класів по файлах початкових кодів і по підсистемах. В останньому випадку може допомогти розподіл класів по спеціалізованих (функціонально-орієнтованих на предметну область) пакетах. На цьому етапі слід звернути увагу на таку реалізацію системи, яка забезпечувала б можливість повторного використання коду за рахунок раціональної декомпозиції системи, тобто мінімізувати кількість зв'язків між компонентами;

- при специфікації загальної структури вихідного коду системи необхідно враховувати специфіку мови програмування, за допомогою якої реалізуються компоненти. Зокрема, в РНР рекомендується функції окремого сервісу описувати в окремому файлі, незважаючи на те, що мова дозволяє описувати кілька функцій у одному файлі і використовувати механізм внутрішніх функцій;

– для специфікації системи необхідно мати в наявності попередню топологію системи. Для кожного вузла в мережі може бути побудована діаграма компонентів, що визначає набір файлів, необхідних для роботи підсистеми (підсистем) на окремому робочому місці;

– на діаграмі можуть бути представлені відносини залежності між компонентами і включеними в них функціями. Ця інформація має важливе значення для забезпечення узгодженості між логічним і фізичним уявленнями системи;

– рекомендується мінімізувати кількість зв'язків між виконуваними компонентами, а необхідна взаємодія здійснюється через збережену в єдиній базі даних інформацію;

– для наочного відображення специфіки компонентів можна замість стандартного символу компонента зі строковим стереотипом всередині використовувати графічні стереотипи.

Виходячи з вищезазначеного, можна побудувати діаграму компонентів системи.

У додатку А зображено ER-діаграму архітектури бази даних, на якій ми бачимо, що таблиці приведені до третьої нормальної форми, та можемо спостерігати зв'язки між полями таблиць.

Взаємодія програмного забезпечення користувача з БД web-порталу має здійснюватися з використанням трирівневої (англ. Three-tier architecture) архітектури «клієнт – сервер». Трирівнева архітектура «клієнт – сервер» використовується в Інтранет- та Інтернет-мережах. Серверна частина web-порталу, що представляє собою базу даних, підключається до SQL-серверу. Клієнтська частина web-порталу, звана «тонким клієнтом» (англ. «Thin client»), являє собою набір активних серверних сторінок, що підключаються до web-серверу, а також до браузера клієнта. Уся програмна логіка винесена на web-сервер, який забезпечує видачу запитів до бази даних, переданих на виконання SQL-серверу.

Для того щоб цей програмний продукт, який використовує мову програмування PHP, працював у будь-якій операційній системі, потрібен віртуальний веб-сервер Apache або Nginx.

Віртуальний веб-сервер – це сервер, що приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, видає їм HTTP-відповіді, зазвичай разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними. Веб-сервером називають як програмне забезпечення, що виконує функції веб-серверу, так і комп'ютер, на якому це програмне забезпечення працює.

Клієнти дістаються веб-серверу за URL-адресою потрібної їм веб-сторінки або іншого ресурсу.

3.2.3 Опис інтерфейсу системи

Після завершення проектування та прототипування системи переходимо до реалізації програмного продукту. Створена система має інтуїтивно зрозумілий та простий інтерфейс, що є доволі зручним для нових користувачів системи та фахівців ведення документообороту, які не використовували подібних систем у своїй практиці.

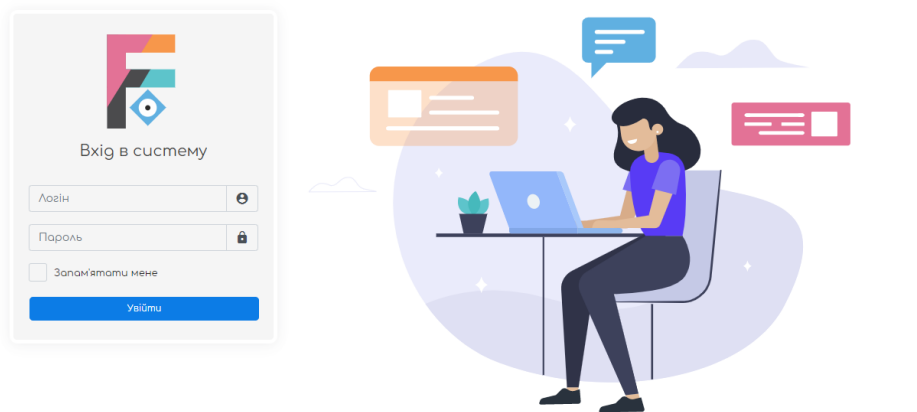


Рисунок 3.4 – Графічний інтерфейс системи. Форма входу

Коли користувач запускає систему, він потрапляє спочатку на форму авторизації в системі, де повинен ввести свої логін та пароль доступу до системи (рис. 3.4.).

Після введення логіну та паролю користувач має натиснути кнопку «Увійти». Якщо логін та пароль були введені вірно, користувач отримує доступ до системи FLUSSO. На головній сторінці (рис. 3.5) користувач може побачити назву додатку, його версію, версію PHP, яка встановлена на його сервері, де розміщена система, операційну систему серверу, об'єм кешу системи та об'єм локальної бази даних. Також на цій сторінці можна відразу очистити кеш системи, якщо він займає зайве місце на диску. В цілях безпеки очистка локального кешу (локальної бази даних) унеможливлена.

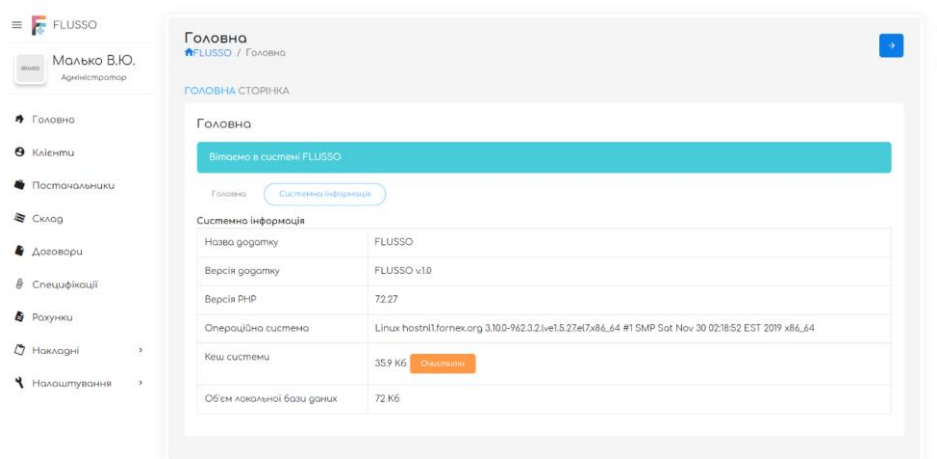


Рисунок 3.5 – Графічний інтерфейс системи. Головна сторінка

З головної сторінки системи FLUSSO користувач може потрапити в будь-яку частину системи за допомогою бокового меню зліва. Для зручності користувача системи на всіх сторінках витримано правило «трьох кліків», тобто користувач може потрапити в будь-яку частину системи з будь-якої сторінки за допомогою не більше ніж 3-х натискань кнопки маніпулятора. Така концепція навігації по системі також є загальноприйнятим світовим стандартом розробки додатків.

На рисунку 3.6 зображено сторінку переліку товарів та послуг в системі, в даному інтерфейсі користувач може додати, видалити чи відредагувати товари, ціни, залишок та тип оподаткування на них.

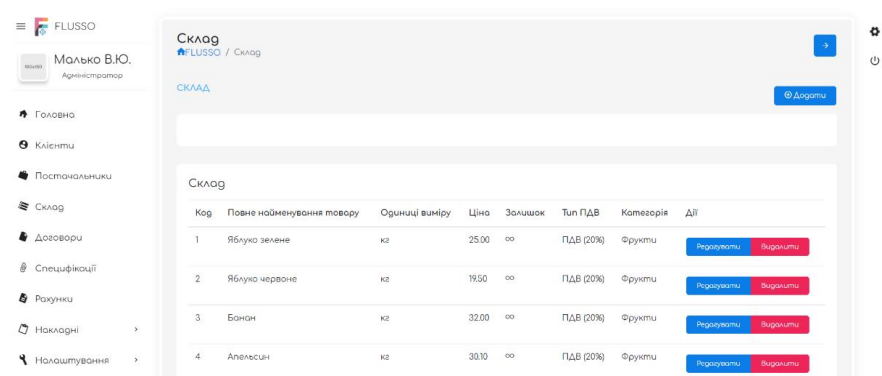


Рисунок 3.6 – Графічний інтерфейс системи. Перелік товарів

Для управління клієнтами, компаніями та постачальниками користувач може обрати відповідний пункт меню зліва та потрапити до інтерфейсу управління інформацією цієї категорії (рис. 3.7).

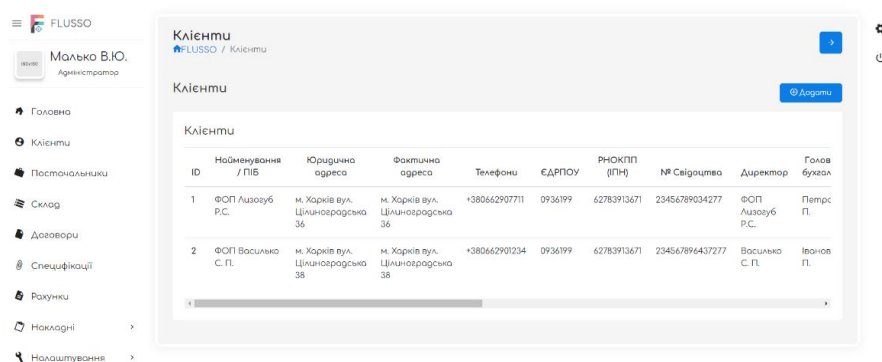


Рисунок 3.7 – Графічний інтерфейс системи. Список клієнтів

На сторінці можна відредагувати, видалити та переглянути інформацію про компанії, що були внесені до системи. Управління постачальниками та компаніями системи аналогічне управлінню клієнтами.

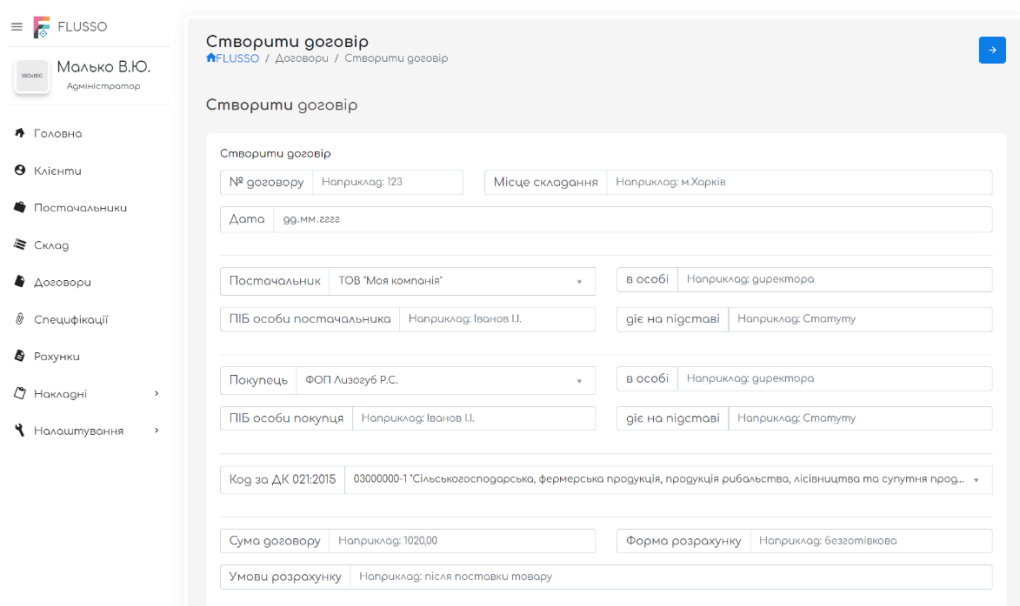
4 ФОРМУВАННЯ ЦИФРОВОЇ ЗВІТНОСТІ ПІДПРИЄМСТВА

4.1 Формування документів

Основним завданням системи є формування цифрової звітності підприємства, а саме таких документів:

- договори;
- специфікації до договорів;
- рахунки-фактури;
- видаткові накладні;
- прибуткові накладні.

Для створення договору користувач вибирає пункт меню «Договори» та натискає кнопку «Створити». Після цих дій користувачу система виведе форму генерації договору (рис. 4.1).



The screenshot displays the 'Створити договір' (Create Contract) form in the FLUSSO system. The interface includes a sidebar with navigation options and a main form area. The form contains the following fields:

- № договору** (Contract No.): Наприклад: 123
- Місце складання** (Place of signing): Наприклад: м.Харків
- Дата** (Date): 99.мм.2222
- Постачальник** (Supplier): ТОВ 'Моя компанія' (dropdown), **в особі** (Person): Наприклад: директора
- ПІБ особи постачальника** (Supplier's name): Наприклад: Іванов ІІ., **гіє на підставі** (Basis): Наприклад: Статуту
- Покупець** (Buyer): ФОП Лизозуб Р.С. (dropdown), **в особі** (Person): Наприклад: директора
- ПІБ особи покупця** (Buyer's name): Наприклад: Іванов ІІ., **гіє на підставі** (Basis): Наприклад: Статуту
- Код за ДК 021:2015** (Code): 03000000-1 'Сільськогосподарська, фермерська продукція, продукція рибальства, лісівництва та супутня прод...'
- Сума договору** (Contract sum): Наприклад: 1020,00
- Форма розрахунку** (Payment form): Наприклад: безготівкова
- Умови розрахунку** (Payment terms): Наприклад: після поставки товару

Рисунок 4.1 – Графічний інтерфейс системи. Створення договору

При генерації рахунка-фактури (рис.4.3) у користувача є можливість згенерувати відразу комплект (рахунок + накладна), натиснувши відповідну кнопку при створенні документа. Незалежно від того, був фіналізований рахунок чи ні, накладна буде сформована нефіналізованою в цілях безпеки та можливості її редагування.

Рисунок 4.3 – Графічний інтерфейс системи. Формування рахунку-фактури

4.2 Цілісність цифрової звітності

Цілісність електронного документа або його захист від підробки в системі FLUSSO досягається за допомогою безповоротного хешування інформації та файлу документа за допомогою криптографічного алгоритму Message Digest 5 (MD5), що дозволяє захистити документ від будь-яких несанкціонованих змін після його фіналізації, адже заміна хоча б одного байта файлу призведе до зміни його криптографічного відбитку, що миттєво зробить документ недійсним при

валідації в системі. Такий метод захисту документів є цілком безпечним та є більш зручним, ніж електронний цифровий підпис, який вимагає від користувачів навички його застосування.

По завершенні проходження хешування даних та файлу документа створюються дві хеш-суми:

- Security Hash – побітовий ключ захисту даних документа (змінних полей);
- File Hash – побітовий ключ захисту файлу.

Для перевірки достовірності інформації та валідності файлу розроблено систему валідації документів (рис. 4.5).



Рисунок 4.5 – Графічний інтерфейс системи. Головна сторінка сервісу валідації документів

Документ можна перевірити за файлом. Для цього потрібно обрати пункт меню «Перевірка документа за файлом» та за допомогою форми завантажити файл документа до системи валідації для перевірки (рис. 4.6).

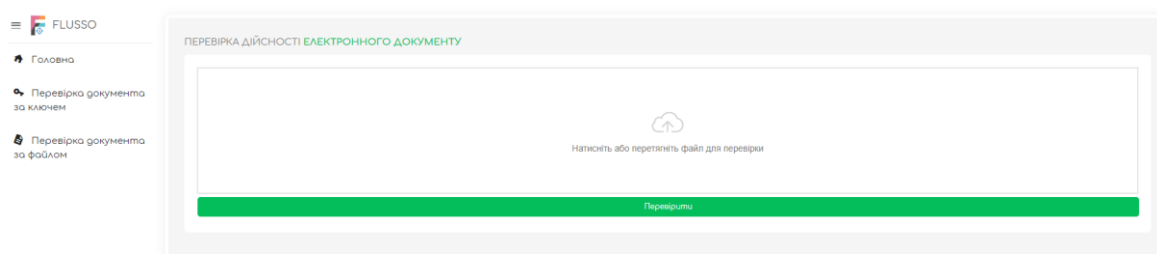


Рисунок 4.6 – Графічний інтерфейс системи. Форма завантаження документа на перевірку

Після завантаження файлу до системи користувач отримає повну інформацію про завантажений документ. На рисунку 4.7 зображено відповідь системи на завантажений коректний файл договору.

Система валідації надає користувачеві таку інформацію про документ:

- тип документа;
 - сторони, які мають відношення до цього документа, або номер документа, якого файл стосується (у випадку перевірки специфікації до договору);
 - дата створення документа;
 - вартість (сума) документа;
 - хеш захисту інформації;
- та інші специфічні дані до конкретного документа.

FLUSSO

- Головна
- Перевірка документа за ключем
- Перевірка документа за файлом

ПЕРЕВІРКА ДІЙСНОСТІ ЕЛЕКТРОННОГО ДОКУМЕНТУ

Дієсний документ (Договір № 1)

Тип документу	Договір
Місце складання	м.Харків
Постачальник/Виконавець	ТОВ "Моя компанія"
Особа постачальника	Пертров П. П.
Покупець/Клієнт	ФОП Василько С. П.
Особа покупця	Василько С. П.
ДК 0212015	03220000-9 "Овочі, фрукти та горіхи"
Сумма	440.00
Дата складання	2020-05-06
Дієсний до	2020-05-25
Хеш захисту	0010Pccde6c2950bb5679521e7a0b8e

Завантажити зображення документа

Рисунок 4.7 – Графічний інтерфейс системи. Відповідь системи на коректний файл договору

Також можна перевірити документ на коректність за допомогою ключа захисту. Для цього потрібно перейти за посиланням в файлі або скопіювати ключ перевірки та вставити до системи валідації. На рисунку 4.8 ключ виділений червоним квадратом

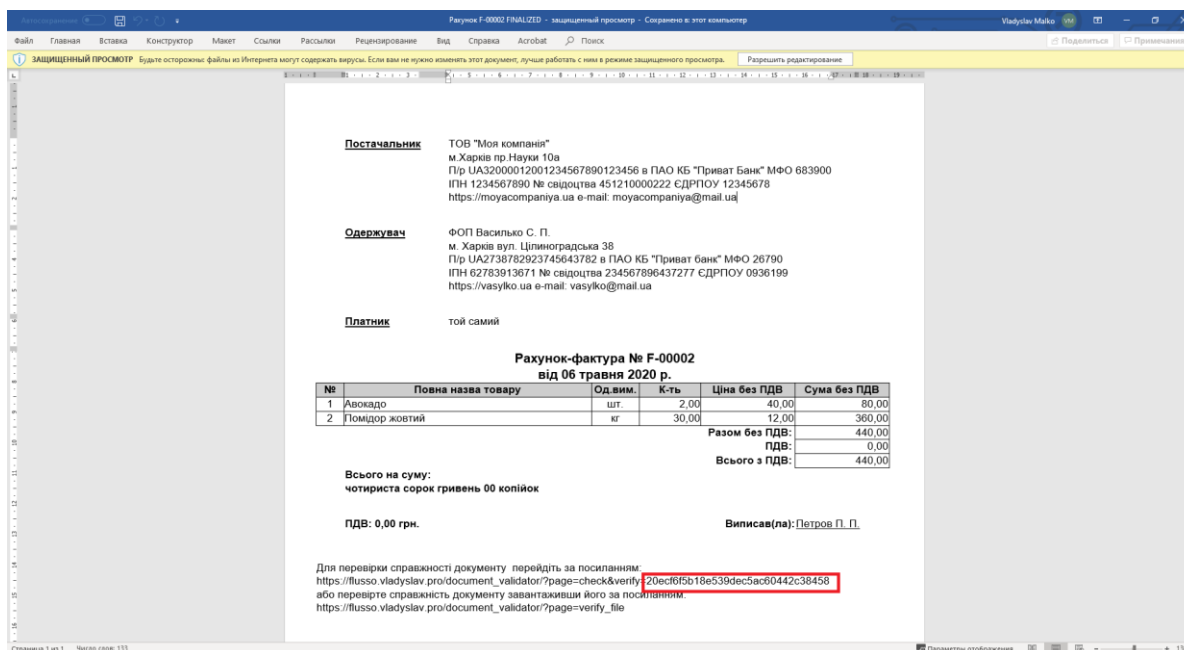


Рисунок 4.8 – Згенерований документ. Рахунок-фактура

Для перевірки документа в ручному режимі за ключем захисту в системі валідації потрібно вибрати пункт «Перевірка документа за ключем». Після вибору методу перевірки користувач побачить на екрані форму вводу ключа захисту (рис. 4.9).

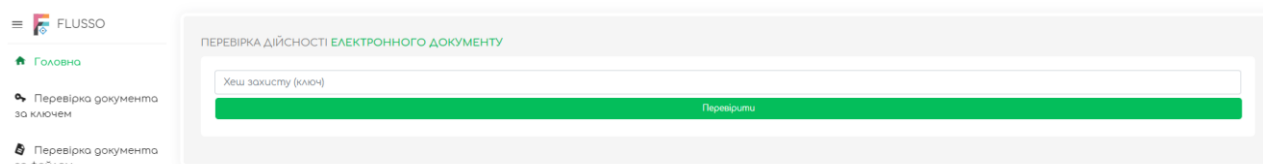


Рисунок 4.9 – Графічний інтерфейс системи. Перевірка документа за захисним ключем

Після введення ключа захисту користувач отримає повну інформацію про документ, який перевіряється на валідність (рис. 4.10), та зможе в разі потреби завантажити дублікат валідного файлу, якщо після перевірки документа за файлом він виявився не валідним.



Рисунок 4.10 – Графічний інтерфейс системи. Перевірка дійсності рахунка

Якщо файл або ключ документа, завантаженого на валідатор, не є коректним, користувач отримує повідомлення «Документ не дійсний або не існує» (рис. 4.11).

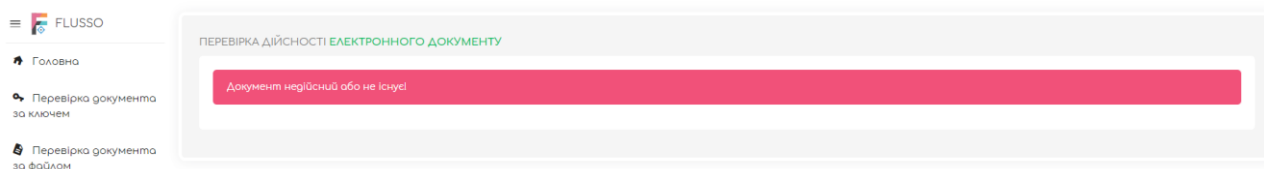


Рисунок 4.11 – Графічний інтерфейс системи. Відповідь системи про некоректний або неіснуючий документ

Отже як ми можемо побачити система є доволі прогресивною та захищеною в плані цифрової звітності, тому використання даної системи в малому та середньому бізнесі буде зручним та доцільним. Також особливістю даної системи є розширюваність, тому її швидко можна адаптувати під специфічні задачі використовуючи її як платформу.

ВИСНОВКИ

У результаті освітньо-наукової атестаційної роботи магістра було проведено дослідження методів проектування програмних систем для автоматизованої генерації цифрової звітності підприємства, аналіз підходів до збереження цілісності електронних документів та розроблено систему документообігу, що є інтуїтивно зрозумілою для користувачів, які не є фахівцями роботи з подібними системами, та має можливість роботи з інформацією в режимі віддаленого доступу з будь-якої точки світу, а також буде мати невисоку вартість за необмежену кількість користувачів та не має надлишкового функціоналу. Система може інтегруватися зі сторонніми сервісами за допомогою відкритого API.

Для розробки системи використано мову програмування PHP 7.2 та реляційну базу даних MySQL, що дозволяє системі бути кроссплатформною і використовуватися на великій кількості платформ, таких як персональні комп'ютери, ноутбуки, телевізори, смартфони, планшети, сервери тощо.

Дана система буде актуальною на українському ринку програмного забезпечення, адже систем, які мають можливість доступу в режимі онлайн на українському ринку, дуже мало та найчастіше такі системи мають занадто велику вартість та часто розробляються під замовлення підприємства.

Якщо оцінювати виконану роботу глобально, то можна зазначити такі позитивні моменти:

- кроссплатформність, можливість використання системи на смартфонах, планшетах, комп'ютерах, телевізорах тощо;
- недостатність аналогів системи для заданих цілей, створених за допомогою об'єднання використаних технологій, що робить систему конкурентоспроможною;
- фактична необхідність таких систем у малому та середньому бізнесі;
- можливість розширення системи для різних задач;

- можливість інтеграції системи зі сторонніми сервісами;
- можливість портування серверної частини системи на пристрої з невисокою продуктивністю;
- можливість використання системи як псевдопрогресивного веб-застосунку;
- можливість використання системи онлайн та оффлайн.

До значних негативних моментів проекту можливо віднести:

- використання веб-серверу, що може викликати труднощі з віддаленим доступом (у разі відсутності стабільного з'єднання з мережею Інтернет).

Наявність таких негативних моментів не стає перешкодою у повноцінному застосуванні системи, а також для подальшої розробки на його основі.

Унікальністю даної системи відносно існуючих рішень на ринку є те що система має відкритий похідний код, можливість швидкого її розширення, та функцію захисту згенерованих електронних документів від підробки. Також система має можливість одночасної роботи в мультирежимі, що дозволяє працювати із системою необмеженій кількості користувачів, а також її можна використовувати в режимі онлайн з будь-якої точки світу та оффлайн (локальному), що також є великою перевагою.

Надалі розроблене ПЗ може бути модифікованим та розширеним до специфічного функціоналу для бізнесу будь-якого масштабу, та в систему можуть бути інтегровані сервіси для державної та банківської звітності.

Отже в ході атестаційної роботи магістра було виконано всі вимоги відповідно до постановки задачі, функціональних та системних вимог, а також розроблено програмну систему таким чином, що її можна швидко модифікувати або адаптувати під різну специфіку підприємницької діяльності.

Розроблена система була апробована в II-му турі Всеукраїнського конкурсу студентських наукових робіт із напрямку «Інформатика і кібернетика». у 2019/2020 н.р у Вінницькому національному технічному університеті, а також подано до міністерства розвитку економіки, торгівлі та сільського господарства України на реєстрацію АП Заява № 93604 від 04 травня 2020р.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Електронний документообіг та захист інформації: навч. посіб. / О.Б. Кукарін; за заг. ред. Н.В. Грицяк. – Київ: НАДУ, 2015. – 84 с.
2. Тарнавський Ю.А. Системи електронного документообігу: опорний конспект лекцій. – Київ: ІПК ДСЗУ, 2007. – 37 с.
3. Белоус Н. В., Стужко С.М., Слипченко Н.И., Борисенко В.П. Интегрированная информационная система учебного научно-производственного центра аутсорсинга // 15-я международная научно-практическая конференция "Современные информационные и электронные технологии" ("СИЭТ-2014") 26-30 мая 2014, Украина, Одесса / Сборник научных трудов конференции, с. 96-97.
4. Матвієнко О.В., Цивін М.Н. Основи організації електронного документообігу: навч. посіб. – Київ, 2008. – 112 с.
5. М.Е.Дос – своєчасна звітність та простий обмін документами // Сайт сервісу М.Е.Дос. URL: <https://medoc.ua/> (дата звернення: 05.03.2020).
6. 1С:Підприємство // Сайт сервісу 1С. URL: <http://1c.ua/> (дата звернення: 04.03.2020).
7. Система електронного документообігу FossDoc // Сайт сервісу FossDoc. URL: <https://fossdoc.com/> (дата звернення: 09.03.2020).
8. Маккоу А. Веб-приложения на JavaScript: пособие. – Санкт-Петербург: Питер, 2012. – 288 с.
9. Веллинг Л., Томсон Л. Разработка веб-приложений на PHP и MySQL: учебник. – 4-е изд. – Москва: Вильямс, 2013. – 848 с.
10. Емельянова Н.З., Патрыка Т.Л., Попов И.И. Проектирование информационных систем: учеб. пособие. – Москва: ФОРУМ, 2009. – 432 с.
11. Степанова Е.Е., Хмелевская Н.В. Информационное обеспечение управленческой деятельности: учеб. пособие. – Москва: ФОРУМ, 2010. – 192 с.
12. Jake Spurlock. [Bootstrap. Responsive Web-Development](#): посібник. – [O'Reilly](#), 2013. – 128 с.
13. Васильков А.В., Васильков И.А., Безопасность и управление доступом в информационных системах: учеб. пособие. – Москва: ФОРУМ, 2010. – 368 с.