

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: «Розробка взаємодії модулів: “Кафедра”, “Документообіг”,
згідно вимог, з використанням “.NET Framework” для ТНТУ»

Виконав(я): студент(ка) VI курсу, групи СПм-61

спеціальності 121 — «Інженерія програмного

забезпечення»

(шифр і назва спеціальності)

Оліферчук О.Ю.

(підпис)

(прізвище та ініціали)

Керівник

Цуприк Г.Б.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Бойко І.В.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Петрик М.Р.

(підпис)

(прізвище та ініціали)

Рецензент

Лупенко С.А.

(підпис)

(прізвище та ініціали)

Тернопіль
2020

АНОТАЦІЯ

Кваліфікаційна робота на тему «Розробка взаємодії модулів: “Кафедра”, “Документообіг”, згідно вимог, з використанням “.NET Framework” для ТНТУ» Оліферчука Олександра Юрійовича. Дана робота розроблена за спеціальністю 121 – інженерія програмного забезпечення, факультет комп'ютерно-інформаційних систем і програмної інженерії, Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль, 2020.

Вона містить 87 сторінок, 28 рисунків, 2 таблиці, 4 додатки, 31 посилання, 16 слайдів.

Метою даної кваліфікаційної роботи магістра є аналіз сфери інженерії університетського програмного забезпечення та розвиток методів розробки програмних продуктів обраної галузі та створення підмодуля “Кафедра” в модулі електронного документообігу “Документообіг” для Тернопільського національного технічного університету імені Івана Пулюя.

Методи та програмні можливості, що використовуються під час створення програмного продукту: .NET Framework, ASP.NET Framework, мова програмування C#, мова перетворення XML-документів XSLT, документація “Latex”, ітераційний метод створення програмного забезпечення.

В результаті роботи було розроблено та відтестовано програмне забезпечення, що відповідає технічному завданню, а саме автоматизовує процес документообігу для кафедр університету, згідно задач, що містяться в технічному завданні, та має величезний потенціал до покращення.

Ключові слова: розробка програмного забезпечення, .NET Framework, ASP.NET MVC Framework, мова програмування C#, мова перетворення xml-документів XSLT, технологія “LaTeX”, javascript бібліотека “Knockout.js”, pdf-документ, документообіг, архітектура програмного забезпечення, UML-діаграми.

ABSTRACT

Qualification work on the topic "Implementation of the .NET Framework and interaction between modules "Academic department" "Document management" for TNTU" by student Oliferchuk Oleksandr Yurievich. This work was developed in the specialty 121 - Software Engineering, Faculty of Computer Information Systems and Software Engineering, Ternopil National Technical University named after Ivan Pulyuy, Ternopil, 2020.

It contains 87 pages, 28 figures, 2 tables, 4 appendices, 31 links, 16 slides.

The purpose of this master's thesis is to analyze the field of university software engineering and development of methods for developing software products in the chosen field and creating a submodule "Department" in the electronic document management module "Document Management" for Ternopil National Technical University named after Ivan Pulyuy.

Methods and software capabilities used when creating a software product: .NET Framework, ASP.NET Framework, C# programming language, XSLT XML document conversion language, Latex documentation, iterative software creation method.

As a result, software was developed and tested that meets the terms of reference, namely automates the document flow process for university departments, according to the tasks contained in the terms of reference, and has great potential for improvement.

Keywords: software development, .NET Framework, ASP.NET MVC Framework, C# programming language, XSLT xml document conversion language, "LaTeX" technology, javascript library "Knockout.js", pdf-document, document flow, software architecture, UML charts.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

Microsoft Visual Studio — лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів.

.NET Framework — програмна платформа, випущена компанією Microsoft в 2002 році. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), яка підходить для різних мов програмування.

ASP.NET MVC Framework — технологія для створення веб-додатків, який реалізує шаблон проектування Model-View-Controller.

C# — об'єктно-орієнтована мова програмування. Розроблено в 1998-2001 роках групою інженерів компанії Microsoft під керівництвом Андерса Хейлсберг і Скотта Вільтаумота як мову розробки додатків для платформи Microsoft .NET Framework.

LaTeX — мова розмітки даних та пакет макросів TeX для високоякісного оформлення документів.

XML (Extensible Markup Language) — розширювана мова розмітки. Рекомендована Консорціумом Всесвітньої павутини (W3C). Специфікація XML описує XML-документи і частково описує поведінку XML-процесорів (програм, які читають XML-документи і забезпечують доступ до їх вмісту).

XSLT (англ. Extensible Stylesheet Language Transformations) — функціональна повна за Тьюрінгом мова програмування, яка використовується для програмування переробки XML-документів.

PDF (англ. Portable Document Format) — формат файлу, створений і підтримуваний компанією Adobe Systems, для представлення двовимірних документів у незалежному від пристрою виведення та роздільної здатності вигляді.

Knockout.js — це бібліотека JavaScript, яка допомагає створювати

розширені, гнучкі користувацькі інтерфейси з чистою базовою моделлю даних.

UML — мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення, для моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

Діаграма послідовності — UML-діаграма, на якій для деякого набору об'єктів на єдиній тимчасовій осі показаний життєвий цикл об'єкта (створення-діяльність-знищення якоїсь сутності) і взаємодія акторів (дійових осіб) інформаційної системи в рамках прецеденту.

Діаграма пакетів — це один з підвидів UML-діаграми, що демонструє загальну структуру ієрархії пакетів системи та взаємозв'язків між ними.

JavaScript — динамічна, об'єктно-орієнтована прототипна мова програмування.

ЗМІСТ

ВСТУП.....	10
1 ІСТОРІЯ РОЗРОБКИ ПЗ ДЛЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ ...	13
1.1 Історія розвитку систем електронного документообігу.....	13
1.2 Особливості впровадження електронного діловодства в Україні.....	14
1.3 Система обробки документів LaTeX.....	16
1.3.1 Формат TeX	16
1.3.2 Поява LaTeX і CTAN	17
1.3.3 LaTeX, його призначення та функціонал.....	18
1.4 Редактор TeXworks	21
1.5 Нормативно-правова база електронного документообігу в Україні	22
2 АНАЛІЗ ПРОГРАМНОЇ СИСТЕМИ	27
2.1 Аналіз вимог до програмної системи.....	27
2.1.1 Огляд наявних систем та здійснення аналізу предметної області....	27
2.1.2 Огляд задач кваліфікаційної роботи магістра.....	31
2.1.3 Здійснення пошуку акторів та варіантів використання	33
2.2 Проектування програмної системи	39
2.2.1 Вибір процесу розробки.....	39
2.2.2 Вибір мови програмування та середовища розробки.....	42
2.2.3 Вибір інтерфейсу для роботи з графікою.....	45
3 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ	47
3.1 Конструювання програмної системи	47
3.1.1 Побудова UML-діаграм послідовності	47
3.1.2 Побудова UML-діаграм пакетів	49

3.2 Реалізація та тестування програмної системи	53
3.2.1 Реалізація та огляд програмної системи	53
3.2.2 Тестування програмної системи	61
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	62
4.1 Охорона праці.....	62
4.2 Освітлення приміщень під час роботи з комп'ютерною технікою	64
ВИСНОВКИ	69
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	71
ДОДАТКИ	74
ДОДАТОК А	75
ДОДАТОК Б.....	80
ДОДАТОК В	82
ДОДАТОК Г	87

ВСТУП

Організація та ведення діловодства є одним з основних напрямків діяльності по управлінню, що виконується в держорганах влади та органах самоврядування по місцях, процесами, і в тому числі на підприємствах, в установах і організаціях усіх типів підпорядкування та форм власності. Її нормативна та науково-методична база завжди зацікавлювала, не тільки одних документознавців та архівознавців, але й цілого суспільства загалом, так як від поставлених процесів діловодства залежить не лише представлення надійних джерел архівної історичної інформації в повній мірі, але й якість управління підприємством різних форм власності. Особливого значення в існуванні документа набуває його електронна версія. Актуальним стає питання функціонування електронного документа, адже в сучасному світі електронний документообіг займає провідну нішу в організації діловодства на підприємствах.

Початковою метою розроблюваної кваліфікаційної роботи є дослідження виникнення та вдосконалення програмного забезпечення для організації електронного документообігу, комунікації та видозмінення головних його парадигм. Поняття «електронний документ», як інформацію, що зафіксована певним чином із можливістю її ідентифікації свого часу визначав М. Ларін. Значної уваги функціонуванню електронного документа приділяли у своїх публікаціях В. Тихонов, А. Косовець, І. Корнєєв та В. Годін.

Розуміння структури організації сучасного діловодства в Україні, його звичайної форми вираження та можливості впровадити у ньому новітні технології має ґрунтуватися на науковому аналізі розвитку цієї галузі упродовж ранішого історичного періоду. Незвертаючи уваги на наявність величезної кількості українських публікацій по історії ведення діловодства, підсумовуюче дослідження питань, що пов'язані з ним, та розвиток нових його форм, особливо електронного документообігу, на сьогоднішній день є

відсутнім в Україні.

Метою даної кваліфікаційної роботи магістра є розробка субмодуля “Кафедра” в модулі “Документообіг”. В тому числі, супровідним завданням проєкту є деталізоване дослідження аспектів розробки програмного забезпечення для електронного документообігу, огляд існуючих програмних систем та дослідження методів їх створення.

Такий термін, як, «електронний документ» з’явився на початку 1990-х рр. у США, але в українському документознавстві він почав використовуватися тільки у 2003 році, коли, також, був прийнятий Закон України – «Про електронний цифровий підпис», і – «Про електронні документи та електронний документообіг» які врегулювали можливості експлуатування електронних документів з необхідними реквізитами – електронним цифровим підписом на загальнодержавному рівні, що дало початок встановлення офіційного комп’ютеризованого діловодства в організаціях (установах та підприємствах) України.

Електронна версія правочину є таким же оригіналом документа, як і паперова, з потрібними реквізитами, разом з електронним цифровим підписом персони, що створила документ, а також з підписом, що прирівнюється до власного підпису персони, що відповідає обмеженням описаним в Законі України «Про електронний цифровий підпис». ЕЦП є необхідним елементом, будь якої версії, електронного документа, який буде використовуватися для ідентифікації персони, що створила, електронну версію документа та інших фізичних або юридичних осіб, що підписали документ в електронному документообігу. Останнім етапом створення електронної версії документа є використання ЕЦП для вступу правочину в дію.

У випадку, коли відбувається відправлення електронного документа багатьом особам або тримання його на кількох носіях для зберігання цифрової інформації кожна версія в електронному примірнику є оригінальною версією документа. Коли персону створює такі ж за змістом та

атрибутами електронний та паперовий документ, обидва з цих документів є оригіналом і мають ідентичну юридичну силу. Оригінальна версія електронного документу повинна надавати можливість для доведення його цілісності та достовірності відповідно до українського законодавства; у передбачених законом випадках, електронний примірник документа має змогу бути представлений у явній формі представлений, навіть у паперовому варіанті.

На сайті Міністерства юстиції України сказано, що також законом були внесені обмеження на застосування електронної версії документа як оригіналу. Використовуючи електронну форму не можна створити свідоцтво на право спадщини; інший документ, який, відповідно українському законодавству, може мати тільки один екземпляр (коли буде створено єдине централізоване сховище оригінальних версій електронних документів, дані обмеження будуть зняті).

Електронний підпис надає можливість підтвердження цілісності та достовірності електронного документу, тобто ізолюваність його від забороненої зміни або знищення в процесі пересування між відправником до одержувачем, та визначити персону, що підписала його.

Договори, які були здійснені за допомогою використання ЕЦП і були передані персоні, якій вони адресувалися, по мережі є захищеними від незаконного доступу, змінення або знищення змісту документа, бо цей підпис створюється криптографічним видозміненням великого набору електронних даних.

За правовою силою, електронний підпис має таке ж значення, що й підпис або печатка конкретної персони, якщо: ЕЦП було підтверджено використовуючи підсилений сертифікат ключа, що в свою чергу використовує надійні засоби цифрового підпису; для перевірки було використано посилений сертифікат ключа, який на момент використання ЕЦП, мав ліцензію; приватний ключ персони, що підписує документ, відповідає відкритому ключу, визначеному у сертифікаті.

1 ІСТОРІЯ РОЗРОБКИ ПЗ ДЛЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

1.1 Історія розвитку систем електронного документообігу

Системи електронного документообігу беруть свій початок ще з кінця XIX ст., від моменту появи фотоапарату та плівки, тобто з того моменту, коли з'явилася можливість створювати набагато менші в розмірі копії документів.

Відкриття комп'ютерно - інформаційних систем із спільним доступом надало можливість підняти технологію електронного документообігу на набагато вищий рівень. Було розширено систему пошуку, надано можливість мати доступ до одного документа одночасно кільком користувачам, віддалена і розподілена робота при малій собівартості копіювання та зберігання інформації спричинили тенденцію стрімкої заміни мікрографічних документів на електронні.

Протягом часу обладнання та програмне забезпечення покращувалося. З'являлися все більш швидкі і якісні сканери, спеціальні плати, що надали можливість пришвидшити процес друкування, та збільшилися об'єми інформації, що зберігалася. З часом з'явилися і набули стрімкого розвитку системи, що можуть розпізнати рукописні і машинописні символи. Почали з'являтися перші системи work-flow, що використовувалися для організації процесів використання паперових документів для електронних архівів, які потім поширилися на бізнес-процеси різних типів організацій та підприємств.

Приблизно на початку 90-х років було сформовано ще один новий тип систем управління ЕД. Переважна більшість це були файли текстових процесорів. Запити на розробку таких рішення були від державних агенств і юридичних компаній.

Розвиток та покращення функціональності систем управління став

наслідком зменшення їхньої вартості близько у два рази. Окрім функції зберігання і пошуку, системи організації електронного документообігу були обладнані засобами, що дозволяли здійснювати контроль над життєвим циклом документа, тобто системи управління документообігом були орієнтовані на роботу з динамічними документами, на відміну від своїх попередників.

Відповідно статистики, 85% документообігу у США залишається паперовим, не беручи до уваги те, що 75% для адміністративних функцій і 73% для поштових операцій користувачі здійснюють через Інтернет. Також, не беручи до уваги екологічну загрозу зменшення площі лісів, передбачається, що виробництво і використання паперових ресурсів у найближчі п'ять років буде збільшено в світі у 1,5 рази. Відповідно цієї статистики, паперовий документообіг, найближчим часом своїх позицій здавати не буде [7].

1.2 Особливості впровадження електронного діловодства в Україні

Відповідно до Закону України "Про затвердження завдань Національної програми інформатизації" (1998 р.) одним із найбільш важливих напрямків є формування національної системи інформаційних ресурсів та використанням у діловодстві України електронного документообігу в усіх сферах діяльності. Автоматизація процесу пересування документів в установах, організаціях та на підприємствах від моменту їх формування або отримання до закінчення виконання або надсилання буде сприяти підвищенню ефективності керівницької діяльності, оптимізації процесів прийняття рішень по управлінні підприємством.

В умовах комп'ютизації суспільства ефективного забезпечення управління є неможливим без застосування електронного документообігу. При його відсутності виникають такі проблеми:

- менеджмент не має повної картини справ на підприємстві;

- структурні підрозділи, не маючи постійної, оперативної та стабільної інформації про діяльність один одного, не зможуть злагоджено діяти, через що відчутно погіршується якість обслуговування клієнтів і змога організацій підтримувати зовнішні ділові комунікації;
- низька продуктивність праці, також існує необхідність ресурсів: комунікаційних, людських і технічних.

Внаслідок цього стає необґрунтоване збільшення штату працівників, неоправдана зміна технології виробництва, інвестиції в облаштування нових робочих місць, комунікації, приміщення, перепідготовка працівників. В ситуаціях необґрунтованого зростання штатів та інвестицій у виробництво, з часом, з'явиться потреба збільшення обігового капіталу, що може призвести до нових кредитів і зменшити прибуток по створеному плані.

Тому, без покращення документаційного забезпечення менеджменту, розвиток підприємства здійснюється тільки екстенсивним шляхом за допомогою раніше накопичених прибутків або збільшенням дефіциту бюджету. Через це, першочерговим завданням впровадження інформатизації у діловодстві України є формування попиту на програми та послуги у даній галузі. Необхідність цього є зумовлена переліченими чинниками:

- з документами працюють усі типи підприємств (організації, установи, фірми) та велика кількість кадрів;
- кожна організація має багато проблем в діловодстві, навіть якщо ми не говоримо про інформатизацію: документи втрачаються, не контролюються;
- за допомогою покращення діловодства підприємств отримують реальну можливість покращити якість свого керівництва, що є одним з провідних завдань сьогоденної економіки;
- кваліфікація секретарів та осіб, що є відповідальними за ведення діловодства - недостатня, з'явилася потреба підвищення кваліфікації секретарів-референтів.

Збільшення запитів користувачів викликає необхідність покращення

програм інформатизації документообігу та їхньої інтеграції з великою кількістю робочих місць, прикладними та інформаційними програмними системами.

Вірну стратегію комп'ютеризації потрібно формувати з відповідно до національної специфіки. Відмінність між технологіями і продуктами інформатизації документообігу в західних країнах і в Україні є в тому, що рішення щодо електронного обігу документів на підприємствах України мають великою мірою враховувати наявність і широке розповсюдження по Україні ділових паперових документів.

Практично ніхто із розробників автоматизованих систем документообігу не застосовує всі технології в сумі: системи управління базами даних, системи аналізу текстів та пошуку документів, системи розпізнавання та сканування документів), клієнтсько-серверне середовище та ін. Внаслідок чого користувачі отримують електронні рішення від різних виробників, а це суттєво підвищує вартість купівлі, впровадження та експлуатації систем. Вибираючи систему автоматизації діловодства і ділових процедур, необхідно враховувати їх відповідність окремим завданням, які буде необхідно вирішувати, окрім того потрібно буде враховувати диференціацію версій продуктів, що були запропоновані.

Велика кількість національних особливостей ведення документообігу, мови, культури не дозволяють використовувати готові закордонні прикладні системи. День за днем, з'являється все більша кількість українських виробників, які пропонують недорогі і якісні рішення щодо автоматизації документообігу [7].

1.3 Система обробки документів LaTeX

1.3.1 Формат TeX

Перш за все, назва пакета TeX читається як "tex", а не "текс". Остання

буква X в назві TeX - зовсім не англійська літера "ікс" (x), а грецька "хі".

В основу TeX закладена досить проста ідея. TeX працює з боксами (box) і клеєм (glue). Елементарні бокси - це букви, які об'єднуються в бокси-слова, які в свою чергу зливаються в бокси-рядки, бокси-абзаци і т.д. Між боксами "розлитий" клей, який має деяку ширину за замовчуванням і ступеня збільшення / зменшення цієї ширини. Об'єднуючись в бокс більш високого порядку, елементарні бокси можуть ворухитися, але після того як знайдено оптимальне рішення, це стан заморожується, і отриманий бокс виступає як єдине ціле. Оптимальне рішення знаходиться за допомогою системи штрафів за те, що клею більше або менше ніж якесь оптимальне значення, певне Кнудом емпірично, а також за розриви абзацу в невідповідному місці. Чим менше штрафів було отримано, тим розміщення "красивіше". Залежно від обраної системи штрафів змінюється і форматування [8].

1.3.2 Поява LaTeX і CTAN

Кнут почав писати TeX за часів, коли, щоб програма працювала, ще доводилося замислюватися про оптимізацію, так як комп'ютери були не швидкі, а пам'яті було небагато. Передбачалося, що друкарня, яка вирішила почати використовувати TeX, заведе у себе TeX експерта для створення і підтримки власної спеціалізованої версії TeX. Але все пішло іншим шляхом і під тиском спільноти TeX став повноцінною мовою програмування, на якому пізніше були написані тисячі і тисячі його розширень.

Початок LaTeX поклав Леслі Лампорт. На початку 80-х років ХХ століття Леслі представив набір макросів на мові TeX, що дозволяють вирішити ту чи іншу стандартну для друкаря завдання. Іншими словами він створив збірник рецептів.

Завдяки цьому, щоб користуватися системою LaTeX і створювати зручні для читання текстові твори, зовсім не треба бути техніком - досить вибрати готовий стиль і використовувати кілька простих команд в залежності від

того, що потрібно в даному випадку. Звичайно, доведеться прочитати документацію, але нестача документації точно не відноситься до складнощів, які можуть виникнути при освоєнні LaTeX.

Збірник рецептів LaTeX все поповнювався і поповнювався, тому в 1992 році був організований файловий архів CTAN. CTAN - це акронім "Comprehensive TeX Archive Network". У цьому архіві знаходиться все, що хоч якось відноситься до LaTeX і має вільну ліцензію. Пошук потрібного пакету LaTeX і його опису слід починати саме звідти.

Основний сервер CTAN розташований в Німеччині по URL-адресою dante.ctan.org. Там же зосереджені і основні сервіси з пошуку і завантаження нових пакетів. Також весь вміст основного сервісу копіюється на безліч дзеркал по всьому світу. Для автоматичного вибору найближчого дзеркала можна скористатися спеціальним сайтом <http://mirror.ctan.org/>.

Цікаво, що модель збірки рецептів LaTeX CTAN була успішно скопійована і активно використовується в багатьох відкритих проєктах. Досить згадати CPAN (The Comprehensive Perl Archive Network), CRAN (Comprehensive R Archive Network), PyPI (Python Package Index) і т.д [8].

1.3.3 LaTeX, його призначення та функціонал

LaTeX (вимовляється як «латéx») — це мова розмітки даних та пакет макросів TeX для відмінного форматування документів. По факту вважається стандартом для підготовки технічних і математичних документів для публікації в наукових виданнях. Використовується форматування LATEX в типографському наборі.

Відмінно від текстових процесорів, особлива увага в LaTeX приділяється розмежуванню оформлення статті від змісту. У LaTeX є засоби для створення структурованих текстів — документів, у авторів яких є можливість основну увагу сфокусувати на змісті, а оформлення і решту шаблонної роботи перекласти на систему. Так же, як і у TeX — вхідні файли LaTeX

можна прирівняти до програм.

Функції системи, загалом, не обмежені (через можливість програмування нових і нових макросів). Макроси по замовчуванню і макроси, які можна знайти на CTAN, надають авторам документів такі функції:

- якісні алгоритми розставлення пробілів між словами, переносів, форматування тексту в абзацах.
- автоматична програмна генерація таблиць ілюстрацій, таблиці змісту документа тощо.
- зручна для використання процедура роботи з посиланнями (на таблиці, формули, ілюстрації, їхній порядковий номер або сторінку).
- зручна для використання процедура роботи з бібліографічними картотеками, цитування бібліографічних джерел.
- створення ілюстрацій (таблиці, ілюстрації та підписи до них, на сторінці, автоматично розташовуються та нумеруються).
- форматування фізичних та математичних формул і рівнянь (високоякісне відображення, нумерація сторінок враховуючи односторонність або двосторонність документа). Також, форматування багаторядкових формул та є велика кількість математичних символів.
- форматування хімічних формул та схем молекул неорганічної та органічної хімії.
- форматування діаграм, схем, графів та синтаксичних графів.
- форматування алгоритмів, розмалювання вихідних текстів програмних систем (у вихідних текстів є можливість безпосередньо входити в документ).
- рознесення документу на окремі частини (функція Topic Maps).

Програмою BibTeX надаються розширені засоби для роботи з бібліографічними документами. Базові функції роботи з математичними формулами розширюються за допомогою пакету-розширення AMS-LaTeX.

Документ LaTeX є текстовим файлом, що включає особливі команди мови розмітки. Документ містить такі частини: преамбулу та тіло. Преамбула містить дані про клас документа, пакети макросів, які використовуються, автора та дату, коли був створений документ та інші допоміжні дані.

Тіло документа включає в себе сам текст документу та команди розмітки. Тіло має бути в оточенні `document`, яке повинно починатися командою `\begin{document}` та закінчується `end{document}`.

Слова розділяються пробілами, при чому, кілька пробілів перетворюється в один пробіл. Пробілом також є початок нового рядка. Пусті рядки розділяють абзаци.

Тому що, формат вхідних файлів LaTeX є простим текстом, їх можна редагувати будь-яким редактором для текстових файлів. Але, існують спеціальні програми для редагування документів LaTeX. Серед них є такі:

- AUCTeX — це гнучкий та потужний пакет макросів для редагування TeX/LaTeX файлів в GNU Emacs та XEmacs.
- GNU TeXmacs — це WYSIWYG редактор, що працює з структурованими текстами документів. Даний редактор розповсюджується за умовами GPL.
- Kile — це редактор для середовища KDE.
- LyX — це текстовий процесор, що працює за принципом WYSIWYM на основі LaTeX з графічним інтерфейсом.
- TeXnicCenter — це потужний редактор текстів LaTeX файлів, що працює тільки на ОС Windows.
- TeXworks — це редактор текстів LaTeX файлів, що працює на Windows, GNU/Linux, Mac OS X.
- TeXstudio — це редактор текстів LaTeX файлів, що працює на Windows, Linux, Mac OS X;
- LaTeX Editor — це вільно розповсюджуваний редактор для роботи з файлами документів TeX та LaTeX для операційної системи Windows.

LaTeX є одним з представників сучасних дистрибутивів TeX і доступний на більшості основних програмних та апаратних платформах. LaTeX, в більшості випадків, розповсюджується на безкоштовній основі, відповідно до відкритої ліцензії.

- MiKTeX — це дистрибутив для MS Windows.
- TeX Live — це дистрибутив для MS Windows та Unix.

На сьогоднішній день, крім стандартних застосувань, LaTeX, також, застосовується як ядро для відображення математичних звітів, формул [9].

1.4 Редактор TeXworks

TeXworks є вільним середовищем для роботи з TeX-документами, що містить в собі редактор, переглядача PDF та має простий в використанні інтерфейс. Воно було створене на основі програми для TeXShop з метою зниження вхідного рівня, людям, у світ TeX, що використовують не Mac OS X операційні системи.

Для роботи у середовищі TeXworks потрібно встановити TeX Live, MiKTeX чи proTeXt [10].

Основні характеристики:

- GUI для системи набору тексту TeX.
- Простий, знайомий інтерфейс на основі поширених додатків.
- Інтегрований переглядач PDF для попереднього перегляду результатів проєкту.
- Синхронізація вихідного / попереднього перегляду для швидкого пошуку місця розташування в вихідному коді з попереднього перегляду PDF.

Графічний інтерфейс користувача, що надається TeXworks, спрощує процес створення документів TeX. Його дизайн заснований на інших кроссплатформених інструментах з відкритим вихідним кодом, щоб забезпечити знайомий користувальницький досвід. TeXworks надає інтегро-

ваний переглядач PDF, щоб легко переглядати результати вашого проєкту з синхронізацією джерела / попереднього перегляду. З синхронізацією джерела / попереднього перегляду ви можете натиснути і утримувати контрольну точку в попередньому перегляді PDF і подивитися, де вона знаходиться в вашому джерелі.

TeXworks робить роботу з TeX більш доступною, ніж використання командного рядка. Якщо ви шукаєте безкоштовний і простий спосіб створення документів TeX, TeXworks - хороший вибір. Якщо ви хочете більше функцій, таких як управління проєктами, ви можете подивитися в іншому місці [11].

1.5 Нормативно-правова база електронного документообігу в Україні

Нормативно-правова база України побудована на принципах інформаційної відкритості та свободи, гарантуючи інформаційну безпеку індивідам, суспільству та державі відповідно до Конституції України.

Державні органи законодавчої гілки влади регулюють суперечки між потребами окремих індивідів, суспільства в цілому та держави збільшуючий вільний обмін інформацією і окремими обмеженнями, що стосуються її розповсюдження.

На сьогоднішній день, головною проблемою в галузі загальнодержавного контролю за електронним документообігом в Україні, яким потрібне невідкладне вирішення, є:

- нестача конкретних організаційно-методичних принципів створення та успішності функціонування «електронного документообігу» в державній галузі України;
- покращення нормативно-правової бази технології формування, обробки, отримання, обміну, використання та ліквідація електронних документів;
- надання законних прав і інтересів суб'єктам господарювання електронного документообігу;

- впровадження більшої кількості допоміжних систем безпеки і захисту інформації, що є конфіденційною, в тому числі правова доказовість і доказовість одержаної інформації, що потребує у виділенні бюджетних ресурсів (коштів) у величезних розмірах;
- залучення кваліфікованих кадрів у сфері сучасних інформаційних технологій та програмних рішень;
- створення успішних та результативних процесів, інструментів та методів держконтролю за дотриманням процедур внесення, створення, зберігання, модифікації, обміну, архівування, ліквідації та передачі інформації в електронно-документообігові системи та якісного виконання працівниками своїх функціональних та нефункціональних обов'язків;
- інші проблеми, що стосуються держрегулювання ЕД в державних органах влади України.

Закони «Про електронний цифровий підпис» та «Про електронні документи та електронний документообіг» безпосередньо регулюють загальні відносини у галузі електронного документообігу.

Закон України «Про електронні документи та електронний документообіг» вступив в силу у травні 2003 року. Відповідно до цього закону встановлюється основна маса організаційно-правових засад електронно-документообігових систем та описує використання електронних документів. З дня офіційного оприлюднення закону, описаного вище, в Україні було почато процес повної комп'ютиризації суспільних відносин та органів влади.

Документообіг в держорганах влади — це система, що втілює в життя процеси отримання, зберігання, модифікації інформації, а також процеси управління: формування та ухвалення бізнес рішень та здійснювати контроль за належним їх виконанням.

Проблеми введення електронного документообігу та ЕЦП в Україні стають все актуальнішими день за днем. Вони мають значний політичний та економічний вплив у сумі зі збільшенням у використанні інформаційно-

комунікаційних систем в суспільних взаємовідносинах, створення систем для електронних платежів або електронної торгівлі і т.д.

Одна з основних проблемних питань, що нам потрібно вирішити на правовому рівні у цій галузі – це робота підприємств по створенню і сертифікації ключів, які повинні надавати послуги, що пов'язані до ЕЦП. Фахівці переконані, що кількість людей, що бажають залучатися в такий діяльності буде досить малою. Через те, що фінансова перешкода виходу таких структур на ринок за сьогодейнішніх умов буде досить великим з огляду на своєрідність їх можливостей (надання функціональності для ЕЦП, формування, поширення, відмінювання, блокування та поновлення сертифікації електронних ключів, формування відкритих та особистих ключів і т.д.). Також, якщо врахувати те, що все має починатися фактично з нуля, так як майже таким самим на цьому етапі є ринок користувачів, тому «повернення» своїх вкладень буде достатньо довгим процесом.

Відповідно до Закону України «Про електронний цифровий підпис» встановлюється юридичний статус ЕЦП і регулюються взаємовідносини, виникачі під час користуванням електронним цифровим підписом. У даному Законі є визначеними такі терміни, як:

- електронний підпис — це електронні дані, що додаються до іншої цифрової інформації або логічно з нею взаємопов'язана та призначена для ідентифікації персони, що підписала дані на якому він був використаний;
- електронний цифровий підпис — це різновид електронного підпису, що отримуються в результаті криптографічного перебору великої кількості електронних даних, що додаються до даного масиву даних або мають логічний зв'язок з ним і дає можливість підтвердити його цілісність та ідентифікувати персони, що підписала документ.

Електронний цифровий підпис використовується для забезпечення діяльності суб'єктів господарування, що здійснюються за допомогою ЕД. ЕП не може бути визначеним, як недійсний, тільки через те, що у нього є елек-

тронну форму або не базується на посиленому сертифікаті ключа. Законом також описується:

- можливості для використання ЕЦП;
- фізичні та юридичні особи правовідносин у галузі послуг ЕЦП;
- права та обов'язки персон, що підписали документ;
- сертифікаційний центр електронних ключів;
- відповідальність за недотримання законів про електронний цифровий підпис;
- визнання сертифікатів, що були видані іноземними сертифікаційними центрами, ключів тощо.

На сьогоднішній день законодавством України методологія електронного документообігу використовуючи електронний цифровий підпис в повній мірі не сформульована. Але робота над формулюванням нормативно-правової бази для електронного документообігу проводиться в держорганах влади. Правда, норми, що є наявними на даний момент часто регулюють дуже вузькоспеціалізовані закриті системи і є переривистими.

Для подальших кроків розвитку української нормативної і правової бази необхідно визнати членство України в Міжнародній організації зі стандартизації ISO, узаконити міжнародні договори, створити та використати необхідні технічні та інші види стандартів, що вже працюють у Європі, пристосувати вже ухвалені закони до вимог міжнародних законів. Також, варто сказати, що навіть швидкі дії у даному напрямку не нададуть швидкого результату у впровадженні нових взаємовідносин у галузі електронного документообігу через відставання у розробці підзаконних нормативно-правових актів. Особливо, впровадження електронного документообігу у роботу державних підприємств та установ тісно взаємопов'язано з необхідністю впровадження реформ у роботу держслужбовців, тобто з пришвидшенням реформи в галузі адміністрування.

Розроблення інформаційних технологій та подальше збільшення кількості галузей в яких буде використовуватися ЕД є об'єктивним

процесом, якому потрібне правове підкріплення. Без відповідного правового підкріплення в цій сфері є неможливим величезний прорив у розвитку електронної комерції. Наявний законопроект в Україні «Про електронний документ та електронний документообіг» призначений вирішувати виникаючі проблемні питання. Але і даний закон ще потребує певного допрацювання.

Узагальнюючи, відмітимо, що на сьогоднішній день вступили в силу кілька нормативно-правових актів (Закони України, Постанови Верховної Ради України, Кабінету Міністрів України, Укази Президента України, та окремих міністерств), що створюють нормативно-правову і законодавчу базу для проведення електронного документообігу. Нормативна база, що регулює інформаційні загальнодержавні взаємовідносини та задовільняє реалізацію держполітики в питаннях, що стосуються, діловодства, документообігу, дає узагальнену теоретичну базу дослідження під час впровадження систем електронного документообігу. В області сьогоденних уявлень електронний документообіг є одним із найбільш важливих засобів менеджменту в установі та базою здійснюваних у ній процесів інформаційного забезпечення підприємства, зокрема реалізованих за допомогою сучасних комп'ютерних технологій. Також, слід зауважити, що законодавство не встановило норми щодо застосування мов для систем електронного документообігу [12].

2 АНАЛІЗ ПРОГРАМНОЇ СИСТЕМИ

2.1 Аналіз вимог до програмної системи

2.1.1 Огляд наявних систем та здійснення аналізу предметної області

1С: Документообіг є програмним продуктом російської фірми «1С» на функціональній платформі «1С: Підприємство 8», призначений першочергово для автоматизації процесу документообігу. (див. рис. 2.1). Дана система надає автоматизацію повного життєвого циклу по роботі з документами, також дозволяє покращити комунікацію між співробітниками і здійснювати контроль над використанням робочого часу. Облік електронних документів є реалізованим відповідно до положень чинної нормативної документації (ГОСТів, інструкцій, вимог і т. д.) та відповідно традицій ведення діловодства. Програмна система забезпечує розраховану на роботу багатьох користувачів як в локальній мережі, так і через інтернет (також через веб-браузери). Програмна система вирізняється великою гнучкістю, високим ступенем деталізації відомостей по збережених даних і широким спектром функціональності. Дозволяє покращити ефективність менеджменту робочого часу працівників, стандартизувати процеси, забезпечити цілковитий контроль і збереження документації і будь-якої іншої необхідної інформації. Можливості системи постійно розширюються.

На даний момент існує кілька варіантів поставок товарів - "1С: Документообіг ПРОФ" і "1С: Документообіг КОРП", що розрізняються функціональністю (версія "КОРП" має більшу функціональність), а також є версія для державних установ - "1С: Документообіг державної установи" .

1С: Документообіг наданий у версіях КОРП і ПРОФ. Версія КОРП в більшості випадків підходить для середніх і великих комерційних підприємств і бюджетних установ зі складною структурою організації або документообігом. Версія ПРОФ може забезпечити потреби організацій з

невеликою структурою організації і нескладним документообігом. [13].

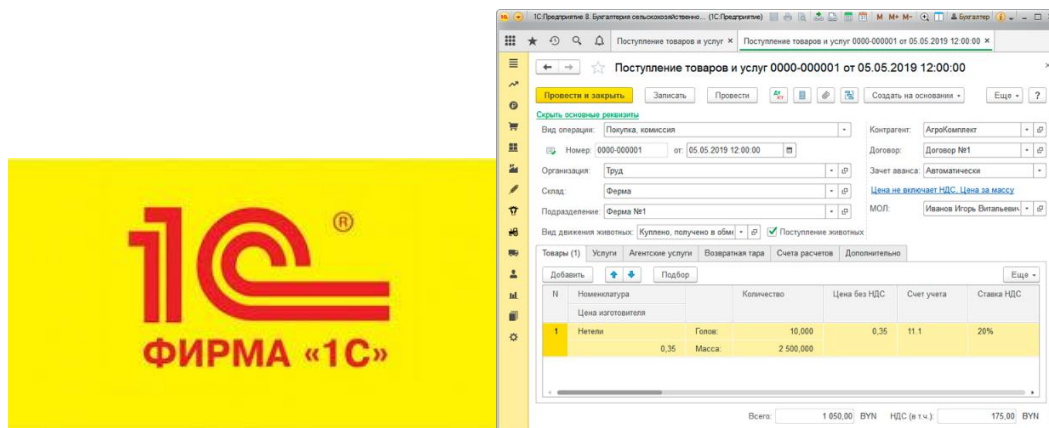


Рис. 2.1 — Логотип та інтерфейс системи поточної версії

1С: Документообіг

Поява ідеї електронного документообігу викликала спроби розробників ПЗ створити свій новітній аналог «1С: Документообіг» в Україні. Одним із таких є українська система «Парус».

Функціонал даної практично в повній мірі копіює головні функції «1С: Документообіг», призначена для автоматизації бізнес-задач підприємств малого та середнього бізнесу, великих корпорацій та холдингів, а також бюджетних установ і організацій тощо [14]. Логотип та інтерфейс системи «Парус» зображені на рис. 2.2.

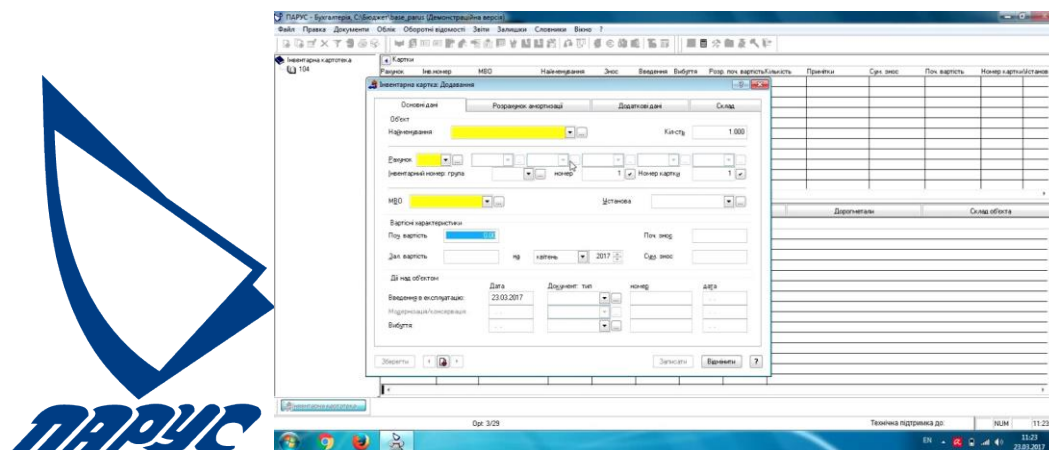


Рис. 2.2 — Логотип та користувацький інтерфейс програмного продукту «Парус»

Систему для електронного документообігу було створено і для приватних підприємців, через це головною задачею було простота використання і простота інтерфейсу системи. Але навіть не зважаючи на свою прос-

тоту система здатна перетворити інформацію в звіт за наступними критеріями:

- Звіт по виробництві.
- Звіт по торговим операціям.
- Зарплата і ЕСВ.
- Здача звітів [15].

Найуспішнішим послідовником «ІС: Документообіг» в сфері програмного забезпечення для організації електронного документообігу можна сміливо назвати українську кросплатформну автоматизовану систему управління підприємством «Дебет Плюс». Програмна система, написана поверх над Eclipse RCP. Вона містить багато можливостей для конфігурування. Як систему керування базою даних може бути використано СКБД по замовчуванню – Apache Derby, сервер бази даних MySQL та PostgreSQL чи, для корпоративної версії, підписку на яку потрібно купувати, комерційні СКБД – Microsoft SQL Server, DB2 чи Oracle.

Вже готові конфігурації орієнтовані на організацію бухгалтерського обліку відповідно законодавству України. Зараз на ринку України пропонуються такі конфігурації системи:

- система для бюджетних та державних установ України;
- система для комунальних некомерційних підприємств;
- система для сільського господарства;
- система для виробничих і переробних підприємств різних сфер промисловості України;
- система для хлібокомбінатів, хлібозаводів, пекарень;
- система «Елеватор, КХП, ХПП»;
- система для водоканалів та ЖКГ;
- система для підприємств електроенергетики, постачання та транспортування газу.

Безкоштовна версія програмного продукту не обмежена в часі роботи,

величині баз даних, можливостями конфігурування і дозволяє повноцінну роботу по мережі.

На завершальній стадії тестування програмного продукту знаходиться розробка повноцінного web-інтерфейсу системи (робота через веб-браузер).

Є підтримка для механізмів оновлення програми, аналогічні схожим механізмам для платформи 1С: Підприємство (оновити програму можна самому без допомоги інших осіб).

Наявні пакети для інтеграції з системами клієнт-банк найбільш популярних українських банків.

Хоча і вихідний код платформи «Дебет Плюс» є закритим і вона розповсюджується за пропрієтарною ліцензією (забороняється декомпілювати і дизасемблювати систему та змінювати вихідний код), в тому, що залежне до бізнес-логіки системи, нескладно побачити ряд ознак відкритої ліцензії. JavaScript і XML коди бізнес-логіки системи повністю відкриті, а Ліцензійна угода безкоштовної версії програмної системи «Дебет Плюс» дозволяє розповсюджувати новостворені програмні продукти на комерційній або безоплатній основі за умови збереження логотипу «Дебет Плюс» та внесення помітних повідомлень про зміни у конфігурації та збереження в складі похідного продукту тексту ліцензії [16].

Логотип та користувацький інтерфейс продукту «Дебет Плюс» зображено на рис. 2.4.

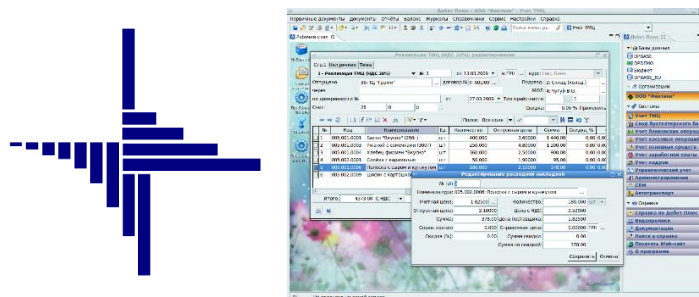


Рис. 2.4 — Логотип та користувацький інтерфейс «Дебет Плюс»

Переглянуті системи відмінно виконують поставлення завдання, але в їхніх ідей і зараз є потенціал до покращення. Істотним мінусом систем «Парус» та «Дебет Плюс» є їхні сильно урізані можливості, а система

«1С: Документообіг», хоч він і має доволі багату функціональність, має дуже велику вартість купівлі підписки та обслуговування в майбутньому.

Велика кількість користувачів кафедр університету бажає отримати розширення «1С: Документообіг» або повністю нову систему для електронного документообігу з переліком функцій представлених в технічному завданні.

Нова ідея також має істотні шанси бути розширеною для використання інтеграцій з іншими модулями університету або зовнішніми системами, чого система «1С: Документообіг» не має по замовчуванню.

2.1.2 Огляд задач кваліфікаційної роботи магістра

Основним завданням цієї кваліфікаційної роботи магістра є розробка програмного забезпечення, що буде автоматизувати організацію електронного документообігу для субмодуля “Кафедра” в модулі “Документообіг”. Можливе розширення системи: інтеграція субмодуля з іншими модулями система або іншими зовнішніми системами. Також, можлива розробка більшої кількості шаблонів звітів та автоматизація інших бізнес-процесів потрібних для кафедр університету.

Спочатку, потрібно створити головний перелік завдань, що потрібно реалізувати в межах даної кваліфікаційної роботи:

- агрегувавши підсумки оглядів наявних програмних продуктів та проаналізувавши обрану предметну область створити список вимог до програмного розробляемого продукту;
- виділити акторів та варіантів використання розроблюваного програмного продукту;
- вибрати мову програмування, модель розробки, інтеграційне середовище для розробки програмного забезпечення та користувацький інтерфейс;
- змоделювати архітектуру програмної системи: розробити UML-

діаграми пакетів і послідовності;

- вдало сконструювати програмну систему, втіливши в життя усі задачі, що були передбачені в рамках розроблюваного програмного проекту;
- розгорнути програмний продукт для користувацького доступу;
- здійснити тестування розробленого програмного продукту та перевірити працездатність та правильність роботи створеного програмного забезпечення.

Під час виконання етапу визначення вимог до ПЗ було створено список функцій, що мають бути розроблені у створюваній програмній системі:

- можливість створення наказів: “Направлення студентів для проходження практики”, “Призначення рецензентів на кваліфікаційні роботи”, “Затвердження тем кваліфікаційних робіт, керівників та консультантів” з субмодуля “Кафедра” в модулі “Документообіг”;
- можливість внесення студентів, рецензентів в накази;
- можливість видалення студентів з наказів;
- можливість генерування PDF-документа наказів відповідно шаблонів;
- можливість видалення та регенерації PDF-документів наказів;
- можливість редагування тексту PDF-документа наказів з використанням мови розмітки LaTeX [9];
- можливість додавання вкладень до наказу;
- можливість перегляду деталей наказів;
- можливість зміни статусу наказів.

Кінцевою та головною вимогою до розроблюваного ПЗ є стабільність і ефективність роботи даного програмного продукту, архітектурна цілісність та неможливість виникнення виключних ситуацій, функціональна придатність, покривання задач описаних в технічному завданні та можливість розширення в майбутньому.

2.1.3 Здійснення пошуку акторів та варіантів використання

Пошук акторів є одним з перших та основних кроків у розробці моделі програмного продукту. Кожен тип зовнішніх та внутрішніх сутностей, з якими повинна здійснювати взаємодію програмний продукт, має бути описаний потрібним актором. Середовище для роботи програмної системи містить в собі користувачів, пристосування та системи — акторів, з якими здійснює взаємодію програмна система. Вони не повинні обов'язково описувати конкретні сутності, а можуть описувати тільки деякі з окремих ролей, які відносяться до потрібних варіантів використання. Конкретно одна сутність може перетворитися в роль для певної кількості акторів, так як і один актор може інкапсулювати в собі ролі певної кількості різних за значенням фізичних сутностей.

Проаналізувавши предметну область та вимоги до розроблюваного програмного забезпечення, було вивлено, що для коректного функціонування системи достатньо таких акторів: ініціатор, погоджувач (див. рис. 2.5).



Рис. 2.5 — Актори програмного продукту

Стислий, підсумовуючий опис знайдених акторів представлено у таблиці 2.1.

Табл. 2.1 — Короткий опис акторів програмної системи

<i>Актор</i>	<i>Короткий опис</i>
Погоджувач	Має змогу переглянути список наказів на погодження, погодити, переглянути та прокоментувати наказ.

Продовження таблиці 2.1

Ініціатор	Має змогу створити наказ та відправити його на погодження, додати/видалити студентів в наказі, згенерувати PDF-документ наказу, видалити згенерований документ, змінити текст в документі наказу, додати вкладення до наказу, переглянути деталі, змінити статус наказу.
-----------	--

Результат зробленого аналізу вимог до розроблюваного програмного забезпечення та пошуку акторів для даної програмної системи допомогли визначити варіанти використання, що агреговані у таблиці 2.2.

Табл. 2.2 — Стислий опис варіантів використання програмного продукту

<i>Актори</i>	<i>Найменування</i>	<i>Формулювання</i>
Ініціатор, погоджувач	Початок роботи з субмодулем “Кафедра”	Користувач субмодуля “Кафедра” здійснює вхід в модуль “Документообіг”, після чого, вибирає пункт меню “Кафедра”, тоді йому відкривається користувацький інтерфейс системи.
Ініціатор	Створити наказ та відправити його на погодження	Користувач натискає на кнопку створити в субмодулі “Кафедра”, після чого йому потрібно вибрати тип наказу, ректора, ініціатора (за умови, коли їм є інша персона), погоджувачі.
Ініціатор	Додати студентів / рецензентів в наказ	Користувач натискає на кнопку “Внести” (підпункт “Студенти” / “Рецензенти”) в модальному вікні наказу, після чого йому відкривається модальне вікно, з відповідною формою. Потім натиснути на “Ок”.

Продовження таблиці 2.2

Ініціатор	Видалити студентів з наказу	Користувач натискає на кнопку: "Список" (підпункт "Студенти") в модальному вікні наказу, після чого йому відкривається модальне вікно, зі списком внесених студентів. Після чого потрібно натиснути кнопку: "Видалити", біля студента якого потрібно видалити.
Ініціатор	Згенерувати PDF-документ наказу	Користувач натискає на кнопку: "Згенерувати" в модальному вікні наказу, після чого відбувається створення PDF-документа наказу відповідно шаблону.
Ініціатор	Редагувати PDF-документ наказу	Користувач натискає на кнопку: "Редагувати" в модальному вікні наказу, після чого відкривається веб-редактор для можливості зміни тексту/структури PDF-документа наказу. Коли користувач вніс необхідні правки потрібно натиснути на кнопку зберегти.
Ініціатор	Переглянути деталі наказу	Користувач натискає на кнопку: "Докладно" в модальному вікні наказу, після чого йому відкривається модальне вікно з деталями наказу.

Продовження таблиці 2.2

Ініціатор	Переглянути PDF-файли наказу	Користувач натискає на кнопку: "Показати" (підпункт "Файли") в модальному вікні наказу, після чого йому відкривається модальне вікно з PDF-файлами наказу.
Ініціатор	Додати вкладення до наказу	Користувач натискає на кнопку: "Внести" (підпункт "Файли") в модальному вікні наказу, після чого йому відкривається форма з можливістю додавання вкладення до наказу.
Ініціатор	Змінити статус наказу	Користувач вибирає статус з підпункту "Встановити статус" в модальному вікні наказу. Після чого відповідний статус буде встановлений для наказу.
Ініціатор	Видалити наказ	Користувач натискає на кнопку: "Видалити" в модальному вікні наказу, після чого йому відкривається модальне вікно з попередженням про видалення наказу. Тоді потрібно натиснути на кнопку "Ок" і наказ буде видалений з системи.

Продовження таблиці 2.2

Погоджувач	Переглянути список наказів на погодження	Користувач переходить по шляху “Робочі документи” / ”Погодження наказів”. Після чого він отримує список наказів, що очікують погодження, або відхилення.
Погоджувач	Переглянути деталі наказу	На сторінці “Погодження документів”, користувач тисне на кнопку “Докладно”. Після чого відкривається модальне вікно з деталями наказу.
Погоджувач	Переглянути PDF-документ наказу	На сторінці “Погодження документів”, користувач тисне на кнопку “Переглянути”. Після чого відкривається модальне вікно з PDF-документом наказу.
Погоджувач	Переглянути вкладення наказу	На сторінці “Погодження документів”, користувач тисне на кнопку “Переглянути”. Потім тисне на кнопку “Файли”. Після чого відкривається модальне вікно з списком вкладень наказу.

Продовження таблиці 2.2

<p>Погоджувач</p>	<p>Коментувати наказ</p>	<p>На сторінці “Погодження документів”, користувач тисне на кнопку “Переглянути”. Потім тисне на кнопку “Коментувати”. Після чого відкривається модальне вікно з формою для коментування наказу. Тоді користувачу потрібно написати потрібний комент та натиснути на кнопку “Ок”.</p>
<p>Погоджувач</p>	<p>Погодити наказ</p>	<p>На сторінці “Погодження документів”, користувач тисне на кнопку “Переглянути”. Потім тисне на кнопку “Погодити”. Після чого відбувається погодження наказу.</p>

Після пошуку акторів та визначення варіантів використання програмного продукту, розробимо узагальнюючу діаграму варіантів використання розроблюваної системи (див. рис. 2.6).

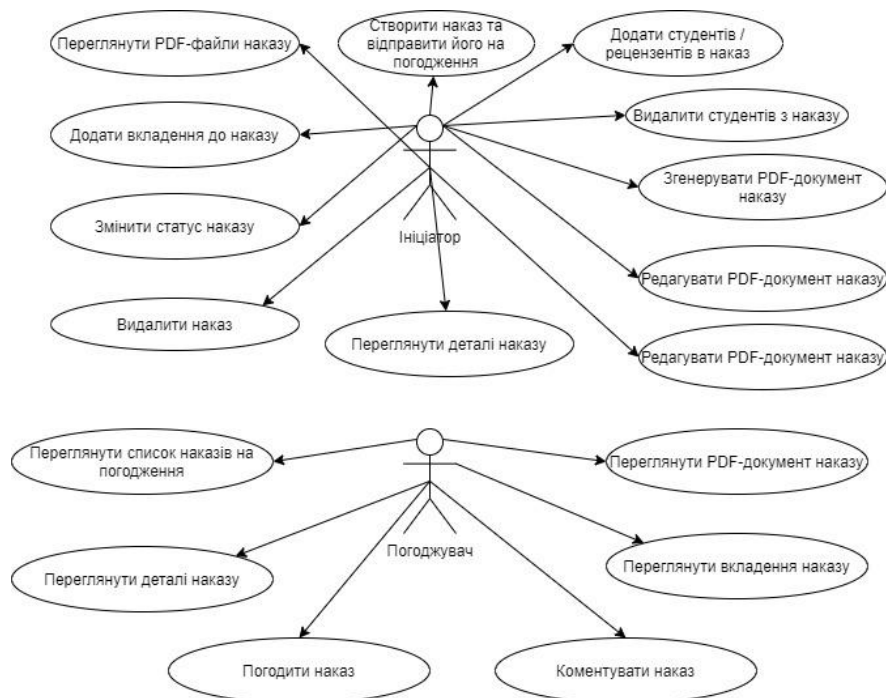


Рис. 2.6 — Діаграма варіантів використання програмного продукту

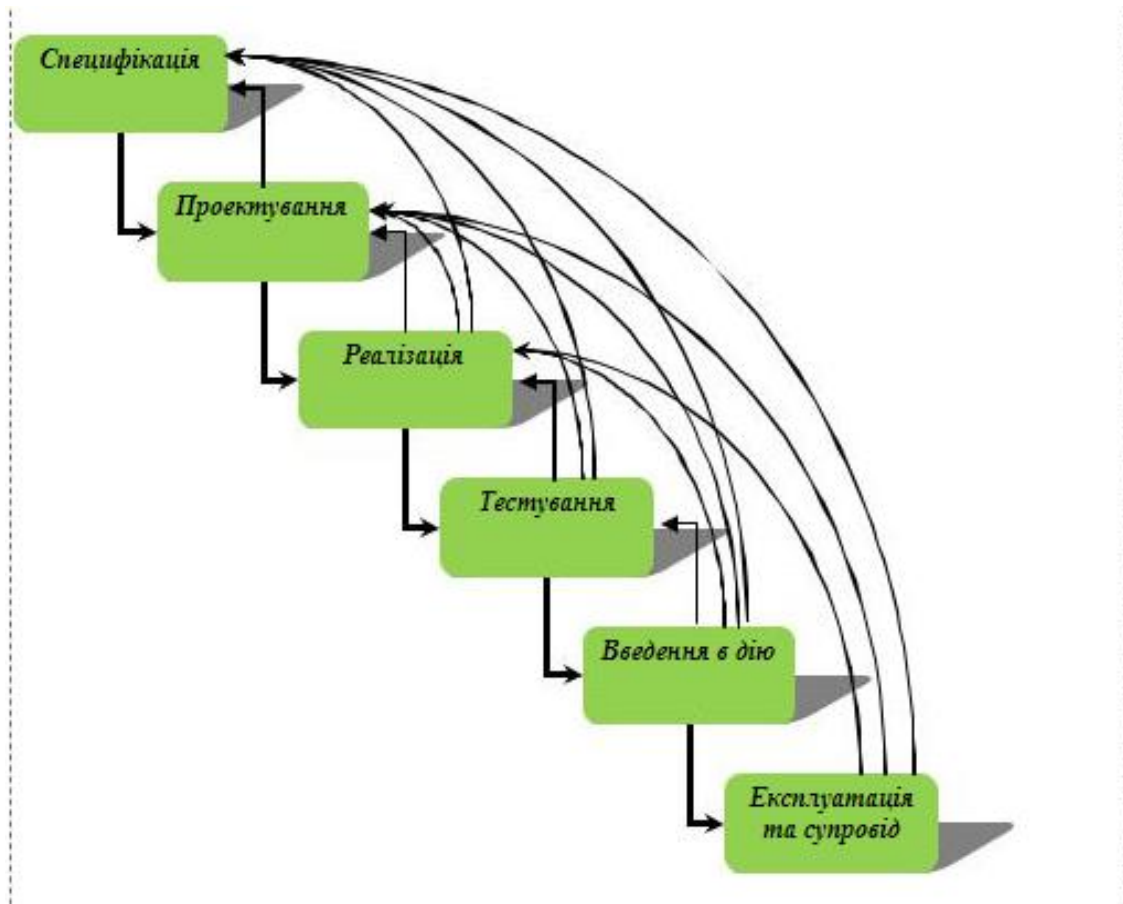


Рис. 2.7 — Класична ітеративна модель

2.2 Проектування програмної системи

2.2.1 Вибір процесу розробки

Зробивши аналіз по існуючих методологіях розробки програмного забезпечення та специфіці програмної системи, що розробляється, в якості моделі розробки було обрано ітеративну модель.

Ітеративна модель (англ. Iterative and incremental development) — це процес розбиття розробки проєкту на послідовність з ізольованих ітерацій, кожна з яких схожа на "міні-проєкт" з усіма існуючими етапами життєвого циклу системи.

Звичайна ітераційна модель дозволяє повернутися на попередні етапи (див. рис. 2.7, аркуш 39). Цей факт залишає серйозний фактор програмних

розробок: бажання попередньо знайти всі варіанти використання системи та у більшості випадків неможливість досягти цього. Загальні технології в програмуванні націлені тільки на те, щоб зменшити моливість повернення. Але загальна ідея від цього не міняється: при відкаті на попередню ітерацію завжди потрібно повторювати побудову того, що вже було готовим.

Зі сторони структури життєвого циклу проекту та процесу розробки дана модель — ітеративна (iterative) (див. рис. 2.8, аркуш 40). Зі сторони розвитку програмного продукту є інкрементальною (incremental). Досвід в галузі розробки програмного забезпечення показує, що розглядати кожен описану вище сторону ізольовано є неможливим. Через це дану програмну модель дуже часто називають моделлю ітеративної та інкрементної розробки (Iterative and incremental development, IID) [17].

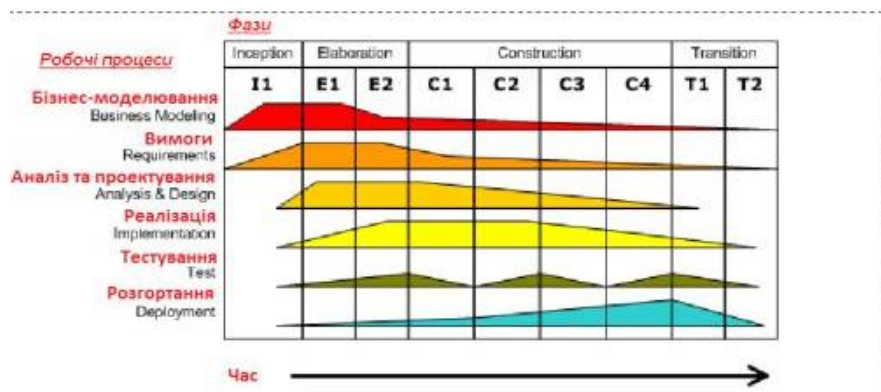


Рис. 2.8 — Ітеративне та інкрементне розроблення програм

Дуже часто, зробивши помилку, вважають, що «ітеративний» це те ж поняття, що й «інкрементний». Але, в розробці програмного забезпечення та програмних рішень вони дуже часто використовуються разом. Основний принцип міститься в тому, щоб циклічно розробити систему, що повторюються (ітеративний метод) та в менші часові рамки (інкрементний метод), дозволяючи розробнику використати переваги того, що було опановано під час розробки попередніх етапів розробки системи. Навчання приходить з процесів розробки та використання системи, де можуть бути ключові етапи процесу починаються з простої реалізації функціональних та нефункціональних вимог до програмної системи та ітеративного покращення версій, які розвиваються доти, поки ціла система не буде розроблена та впроваджена. На

кожному етапі (ітерації) змінюється дизайн системи та розробляються нові можливості.

Процес складається з таких кроків: формування, ітеративного процесу та контрольних вимог до проєкту. Етап формування створює базову (початкову) версію програмної системи. Метою початкової версії системи є формування продукту на який можуть залишити свій відгук користувачі. Розроблюваний продукт повинен запропонувати набір зразків ключових факторів проблемної задачі і допомогти у розв'язанні, яке буде доволі простим для його розуміння та реалізації. Для управління ітеративним процесом складається контрольний перелік задач проєкту, який містить опис усіх завдань, що необхідно виконати. Він містить такі аспекти: нові функціональні та нефункціональні можливості системи, які потрібно розробити та частини наявного рішення, які потрібно модифікувати. Контрольний перелік задач постійно зазнає змін, в результаті проходження фази аналізу.

Етап ітерації включає в себе реконструювання та імплементацію задач з контрольного переліку, а також аналіз наявної версії системи. Рівень висвітлення проєктного плану(дизайну) не описується ітеративним підходом. У малих ітеративних проєктах джерелом документації програмного продукту може бути програмний код. Але, у важливих для підприємства проєктах може використовуватися формальний документ, який є планом програмної системи. Аналіз та огляд ітерації ґрунтується на відгуках користувачів та на наявних засобах аналізу системи. Він містить в собі аналіз структури, надійності, модульності, зручності використання, продуктивності та досягнення поставлених цілей. Контрольний перелік задач проєкту змінюється відповідно до результатів аналізу.

Інкрементна розробка розбиває систему на окремі, незалежні інкременти(порції). На кожному етапі розробляється та впроваджується певна частина функціональності. Уніфікований процес розбиває порції/ітерації на такі етапи: початок, створення плану, побудова та перехід.

Початок розробки визначає величину проєкту, ризики і вимоги (функціональні та нефункціональні) на загальному рівні, але є достатнім для того, щоб оцінити складність розробки рішення.

В результаті створення плану виникає виробнича архітектура, яка зменшує найбільші ризики та заповнює нефункціональні вимоги.

Побудова ПЗ поступово (інкрементно) наповнює архітектуру готовим програмним кодом, який створено за допомогою аналізу, планування, реалізації та тестуванню функціональних вимог.

Кожен з етапів можна розділити на кілька ітерацій, які, зазвичай, є обмеженими у часових рамках, а не у функціональності. Аналітики та архітектори є завжди на одну ітерацію попереду програмістів та тестувальників, для того, щоб тримати їхній перелік задач повним [18].

2.2.2 Вибір мови програмування та середовища розробки

Наступною стадією проєктування є вибір мови програмування для розробки програмного продукту. Це визначення є фундаментальною особливо важливим для створюваного майбутнього проєкту. У будь-якій мові програмування є свій перелік ключових слів та особливості синтаксиса розробки програмних рішень, рівень входження та особливості використання, плюси та мінуси. Беручи до уваги, рейтинг використання мов програмування, особливості проєкту і технічну особливість наявної системи “Документообігу” було обрано C# .NET Framework.

C# (вимовляється як "сі Шарп") - сучасна об'єктно-орієнтована і типобезпечна мова програмування. C# відноситься до широковідомого сімейства мов C, і здається добре знайома кожному, хто працював з C, C++, Java або JavaScript.

C# - це об'єктно-і компонентно-орієнтована мова програмування. C# надає мовні конструкції для безпосередньої підтримки такої концепції для розробки. Через це мова програмування C# є чудовим вибором для створення і

впровадження програмних рішень. З моменту створення мова C# збагатилася функціями для підтримки нових робочих навантажень і сучасними рекомендаціями по розробці ПЗ.

Нижче перелічено тільки кілька можливостей мови C#, що надають надійність і цілісність систем, що на ній розроблені. Прибирач сміття, автоматичний очисник пам'яті, яка є зайнятою об'єктами, що є недосяжні або не використовуються. Обробка виключних ситуацій дозволе структуровано і розширювано виявити помилки і відновитися після них. Лямбда-вирази підтримують прийоми функціонального програмування. Синтаксис запитів створює загальний шаблон для роботи з даними з будь-якого джерела. Підтримка мов для асинхронних операцій надає синтаксис для створення розподілених систем. Зіставлення шаблонів надає синтаксис для простого поділу даних з алгоритмів в сучасних розподілених системах. У C# діє загальна система типізації. Всі типи даних в мові програмування C#, що включають, також, типи-примітиви, такі що й, наслідуються від одного базового типу. Кожний тип має загальний набір операцій, а значення, що містить будь-який тип підлягає зберіганню, передаванню і обробці схожим методом. Більш того, C# дозволе як визначити користувачам, як посилальні тип, так і тип значень. C# дозволяє динамічно створювати об'єкти та містити малі структури даних в стеці.

У C# особлива увага приділяється управлінню версіями для забезпечення сумісності програм і бібліотек при їх зміні. Проблема управління вірсійністю здійснило суттєвий вплив на такі нюанси розробки C#, як модифікатори розділення і, особливості вирішення проблеми перевантаження методів і надання можливості прмого оголошення членів інтерфейсу [19].

Платформа .NET Framework є технологією, що підтримує створення і виконання веб-служб та додатків ОС Windows.

Платформа .NET Framework містить загальномовне середовище виконання (середовища CLR) і бібліотеки класів .NET Framework. Ядром платформи .NET Framework є середовище CLR. Дане середовище виконання

можна назвати агентом, який керує кодом під час виконання і надає основні служби, такі як управління пам'яттю, управління потоками і віддалена взаємодія. При цьому середовищем застосовуються умови суворої типізації та інші види перевірки точності коду, що забезпечують безпеку і надійність. Загалом основним завданням середовища виконання є управління кодом. Код, який звертається до середовища виконання, називають керованим кодом, а код, який не обертається до середовища виконання, називають некерованим кодом. Бібліотека класів є загальною об'єктно-орієнтованою колекцією повторно використовуваних типів, які застосовуються для розробки додатків - починаючи зі звичайних додатків, що запускаються з командного рядка, і додатків з графічним інтерфейсом (GUI) і закінчуючи додатками, що використовують останні технологічні можливості ASP.NET, такі як веб-форми і веб-служби XML.

Платформа .NET Framework має можливість розміщуватися некерованими компонентами, які завантажують середу CLR у власні процеси і запускають виконання керованого коду та створюють таким чином програмне середовище, що дозволяє використовувати кошти як керованого, так і некерованого виконання. Платформа .NET Framework не тільки надає кілька базових середовищ виконання, але також дозволяє розробку базових середовищ виконання незалежними виробниками [20].

Завдяки великій різноманітності синтаксичних конструкцій та можливості роботи з платформою .NET Framework, дана мова дозволяє швидко впроваджувати програмні задачі, характеризується надійністю та елегантністю, в зв'язку з чим найкраще підходить для розробки субмодуля “Кафедра” в модулі “Документообіг” для автоматизації електронного документообігу.

В якості середовища розробки було обрано Visual Studio 2019 — інтегроване середовище розробки програмного забезпечення, що користується надзвичайно великими популярністю та довірою, і є найкращим вибором для реалізації проєкту магістерської кваліфікаційної роботи.

2.2.3 Вибір інтерфейсу для роботи з графікою

В даний момент найбільш популярними інтерфейсами для роботи з графікою є веб-клієнти. В платформі .NET Framework є дві технології для створення веб-інтерфейсів: ASP.NET Web Forms та ASP.NET MVC.

ASP.NET Web Forms - це платформа веб-додатків і одна з декількох моделей програмування, підтримуваних технологіями Microsoft ASP.NET. Програми веб-форм можуть бути написані на будь-якій мові програмування, що підтримує Common Language Runtime, наприклад C# або Visual Basic. Основними будівельними блоками веб-сторінок веб-форм є серверні елементи управління, які представляють собою повторно використовувані компоненти, що відповідають за відображення розміток HTML та реагування на події. Метод, що називається системою перегляду, використовується для збереження стану сервера серверних елементів управління між звичайними HTTP-запрошеннями без збереження стану [21].

ASP.NET MVC Framework - технологія для створення веб-додатків, який реалізовує шаблон проектування Model-View-Controller. У квітні 2009 року вихідний код ASP.NET MVC був опублікований під ліцензією Microsoft Public License (MS-PL). 27 березня 2012 року ліцензія була змінена на ліцензію Apache 2.0. Платформа ASP.NET MVC розроблена, відповідно патерна, на взаємодії трьох елементів програмного рішення: контролери, моделі та представлення. Контролер отримує запити, здійснює обробку користувачького вводу, обмінюється даними з бізнес-моделлю та користувачьким представленням та здійснює повернення користуваль-ницького результату обробки запитів. Моделлю описується бізнес-шар, що містить логіку організації та обміну даними у програмних рішеннях. З контролера дані відправляються представленню, коли воно отримує дані, то починається процес генерування елементів користувачького інтерфейсу для графічного зображення інформації.

Таким чином, ASP.NET MVC та ASP.NET Web Forms є двома рідними

технологіями в основі яких лежить одна платформа ASP.NET. Однак ASP.NET MVC має ряд переваг перед веб-формами ASP.NET:

- розділення відповідальності (окрема розробка різних компонентів - контролери, моделі, представлення);
- покращена можливість тестування;
- підвищена гнучкість і настраюваність під власні потреби.

В даний час розробляється ASP.NET MVC 6, як частина ASP.NET Core; 27 червня 2016 року відбувся вихід версії 1.0.0 [22].

Беручи до уваги, маленьку складність графічної складової програмного продукту, що розробляється, та простоту елементів користування програмним інтерфейсом ASP.NET MVC Framework при використанні Visual Studio, платформу ASP.NET MVC було вибрано для розробки усіх частин графічної складової цієї кваліфікаційної роботи магістра.

3 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

3.1 Конструювання програмної системи

3.1.1 Побудова UML-діаграм послідовності

Діаграма послідовності (англ. *sequence diagram*) — це різновид UML діаграми. Дана діаграма відображає взаємодії об'єктів, які є впорядковані в часових рамках. Такі діаграми відображають використані об'єкти та послідовність надісланих повідомлень.

Діаграми послідовностей показують у вигляді вертикальних ліній різноманітні процеси та об'єкти обраної предметної області, що існують в один момент часу. Відправленні повідомлення горизонтальними лініями зображуються, в порядку, у кому вони були надіслані [23]. Описану вище діаграму можна використовувати для графічної візуалізації простих дій, якщо є необхідність в дослідженні поведінки певної кількості об'єктів в межах одного варіанту використання програмного продукту. Також, діаграма послідовності може використовуватися представляючи взаємодію об'єктів до потрібного варіанту використання.

Діаграма послідовності зображується на рис. 3.1 для опису варіанту використання «Згенерувати / редагувати PDF-документ наказу». Користувач відправляє запит для генерування PDF-документа, після чого логічна частина системи розпочинає формування документа і відправляє запит на початок роботи процесу графічної візуалізації для дії по створенню файлу. Графічний інтерфейс програмної системи виконує потрібні дії, після чого надсилає повідомлення про успішне чи не успішне закінчення роботи запиту до логічної частини програмної системи, яка зі своєї сторони надсилає повідомлення про результати роботи запиту до кінцевого користувача. За аналогічним сценарієм в процесі виконання відбувається редагування.

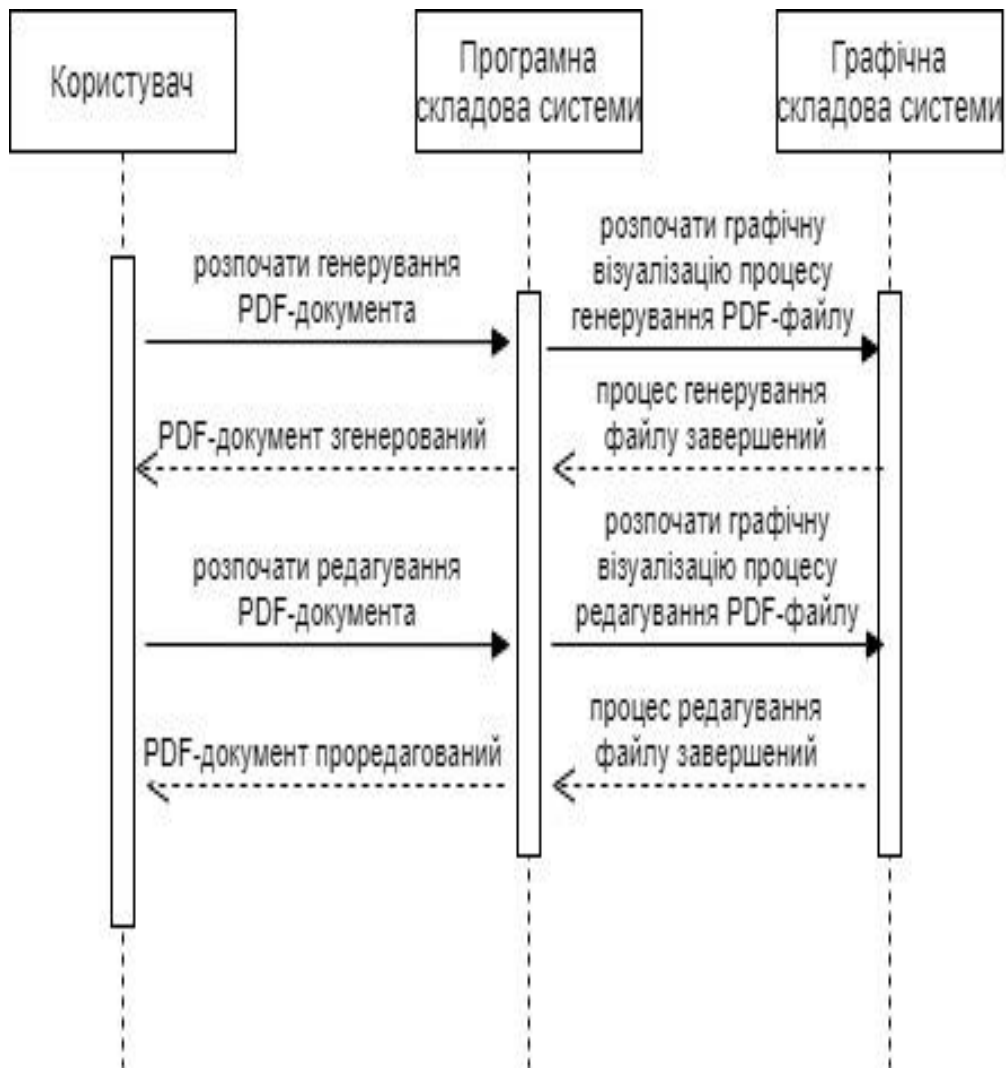


Рис. 3.1 — Діаграма послідовності варіанту використання
«Згенерувати / редагувати PDF-документ наказу»

Діаграма послідовності зображується на рис. 3.2 для опису варіанту використання «Внести студентів / рецензентів в наказ». Користувач відправляє запит на внесення студентів / рецензентів в наказ, логічна складова системи створює та повертає йому форму для внесення потрібної інформації, за допомогою якої буде відбуватися внесення персон в наказ. Після заповнення необхідних полів користувачу поотрібно надіслати повідомлення логічній складовій системи, після чого вона відправляє запит для внесення персони в наказ. Графічний інтерфейс програмної системи виконує запит та надсилає повідомлення про успішність виконання запиту програмній частині системи, яка у свою чергу надсилає повідомлення про успішність виконання запиту користувачеві. Аналогічно відбувається редагування.



Рис. 3.2 — Діаграма послідовності варіанту використання «Внести студентів / рецензентів в наказ»

3.1.2 Побудова UML-діаграм пакетів

Діаграми пакетів в уніфікованій мові моделювання програмних систем (UML) відображають взаємозв'язки між пакетами, які входять в розроблену програмну модель.

Пакет (package) — це один з елементів програмної моделі, що може бути використаний для групування та систематизації інших частин моделі. Частини моделі, які належать до складу певного пакету, є членами пакету. Пакету належать усі його члени. Кажуть, що члени програмного пакету містяться у пакеті, тобто пакет володіє ними. Коли якийсь пакет видаляється з програмної моделі, всі його члени, що належать цьому пакунку, також

видаляються з моделі.

Крім загальних залежностей в UML наявні спеціальні типи залежностей, що можуть існувати між пакетами:

- Імпорт пакету.
- Злиття(суміщення, об'єднання) пакету.

Імпортом пакетів є відношення між іменним простором (namespace), що здійснює імпорт та пакунком, який допомагає зрозуміти той факт, що даний простір імен сумує назви членів пакету з їхнім власним простором імен. За алгоритмом, невстановлене відношення між двома пакунками розуміється за залежністю імпорту пакетів.

Об'єднання пакетів є направлений зв'язок між кількома пакетами, що дозволяє зрозуміти той факт, що між вмістом двох пакунків повинно здійснитися злиття. Даний вид зв'язку має величезне зходження з «узагальненням» за тим значенням, що початковий елемент, можна сказати, додає характеристику початкового елемента до своїх елементів, після чого формується новий елемент, що включає в себе всі особливості двох попередніх елементів.

Діаграмами пакетів можуть використовувати елементи, що включають в себе прецедентів для огляду функціональності програмного продукту. Діаграмами, можуть відображатися пакети, що демонструють різні шари архітектури програмної системи, що містять певну кількість архітектурних рівнів. У відношення між даними пакетами можуть входити позначки або стереотипи, для демонстрації зв'язків між ними [24].

У програмній системі використовується спрощена багаторівнева архітектура. Тут відразу потрібно зауважити, що багаторівневою архітектурою частенько називають не дуже пов'язані визначення: n-layer та n-tier. Як layer, так і tier, найчастіше, називають поняттям «рівень», інколи до поняття «layer» ще можуть використати визначення «шар». Але в обох визначеннях рівні будуть мати різний порядок.

Tier є фізичним рівнем. Особливо, коли ми використовуємо класичну

трирівневу архітектуру, то n-tier рішення може бути розділено на такі елементи: сервер на якому буде розміщена база даних системи, веб-сервер на якому буде працювати веб-додаток та для доступу користувача до системи – браузер. Тобто кожний виділений вище рівень працював би, в даному випадку, в окремому фізичному процесі, навіть в тому випадку, якщо всі елементи архітектури користувача фізично розташувати на одному сервері. Коли б для клієнта використати мобільний додаток, то до нашої архітектури ми б додали ще один додатковий фізичний рівень.

Елемент даної архітектури – Layer, описує логічний рівень системи. Тобто ми можемо виділити рівень в якому буде відбуватися доступ до джерела даних, рівень для інкапсуляції бізнес-логіки системи, рівень користувацького інтерфейсу, рівень сервісів та нескінчену множину інших рівнів. При такому підході фізичні рівні не перетинаються з логічними. Звісно, додатки написані на ASP.NET Framework включають в себе і контролери, які обробляють ввід від користувачів так і представлення, які будуть генерувати користувацький інтерфейс, тобто здійснюється розбиття на фізичних два рівні [25].

Основними пакетами системи, що забезпечують її функціонал, є:

- Common – збірка з розширеннями, що використовуються в веб-клієнті.
- Database – SQL Server база даних, що містить в собі всі необхідні дані для коректної роботи системи.
- DataManager – збірка для абстрагування роботи з базою даних. Інкапсулює в собі всю роботу додатку з базою даних. В ній використовується Entity Framework 6 та .csx скрипти для синхронізації статичних перелічень в коді та в таблицях бази даних. Для комунікації зі збіркою ззовні використовується патерн “Репозиторій”.
- DocumentBuilder.Common – збірка для абстрагування створення PDF-документів наказів відповідно шаблонів. В ній використовуються шаблони, створені з використанням мови розмітки “LaTeX” та мова для

трансформацій xml-файлів “XSLT” для прив’язки даних з шаблоном.

- **DocumentBuilder.Tests** – збірка для тестування створення PDF-документів наказів. Для виконання тестів використовується фреймворк від компанії Майкрософт “MSTest.TestFramework”.
- **DepartmentService** – віддалений сервіс, написаний з використанням технології WCF, що використовується для інкапсуляції логіки по отриманню даних по студентах / рецензентах та інших допоміжних даних необхідних для формування наказів.
- **WebUI** – збірка, що містить в собі основну логіку система для зв’язку даних з LaTeX-шаблонами наказів та користувацький інтерфес. Для отримання даних використовує **DataManager** та **DepartmentService**, для створення користувацького інтерфейсу використовується фреймворк від компанії Майкрософт “ASP.NET MVC Framework” та javascript бібліотека “Knockout.js” для інтеративності системи.

Розроблену діаграму основних пакетів програмного продукту зображено на рис. 3.3.

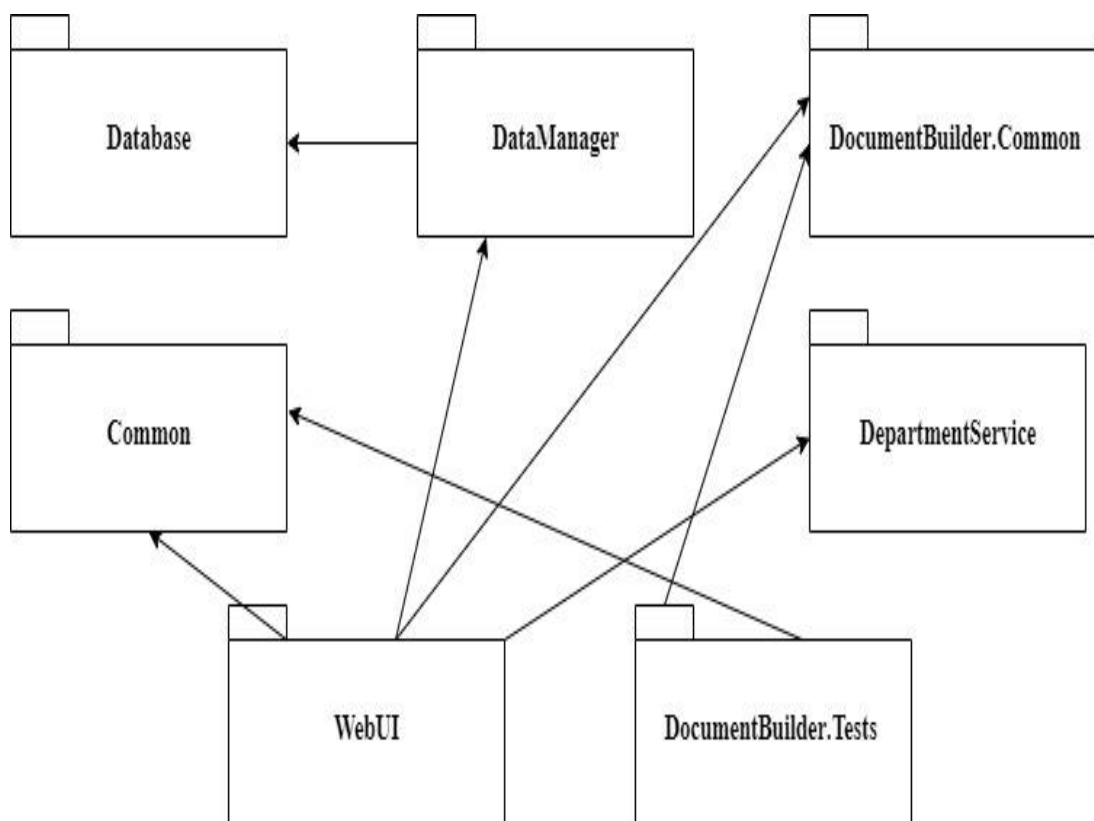


Рис. 3.3 — Діаграма основних пакетів програмної системи

3.2 Реалізація та тестування програмної системи

3.2.1 Реалізація та огляд програмної системи

Перейшовши по відповідному посиланні системи першою показується сторінка входу в систему, що дозволяє користувачу увійти в модуль “Документообіг” (див. рис. 3.4).

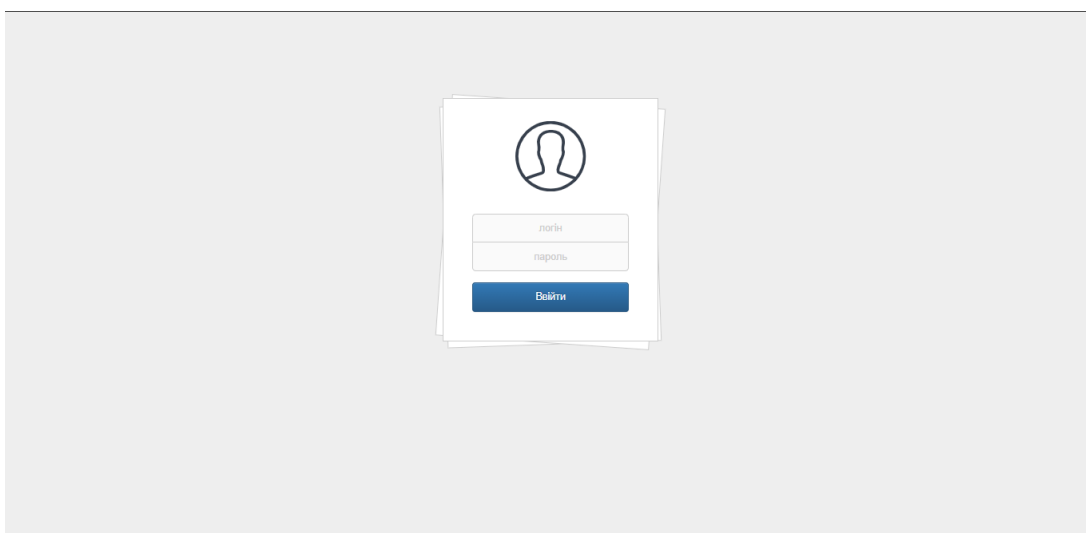


Рис. 3.4 — Сторінка входу в систему

Після введення логіна користувача та паролю буде здійснено вхід в систему та перенаправлено користувача на головну сторінку документообігу (див. рис. 3.5).

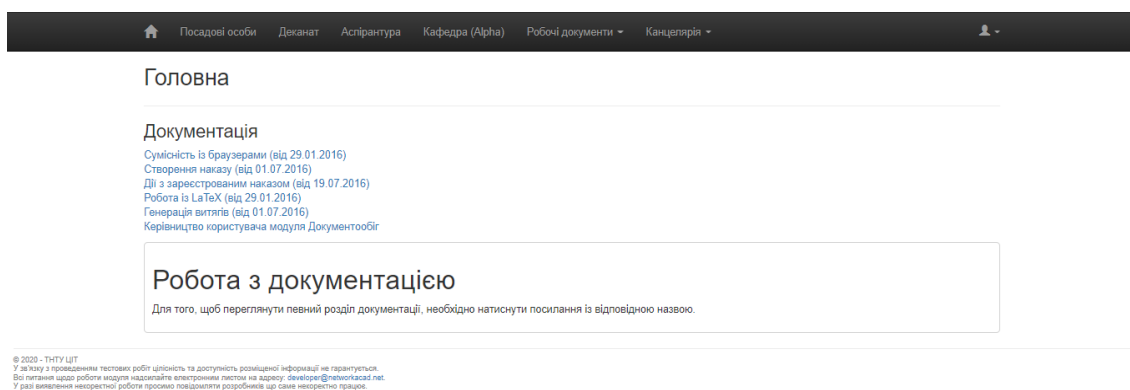


Рис. 3.5 — Головна сторінка документообігу

Для переходу на новорозроблений модуль потрібно натиснути на посилання “Кафедра (Alpha)”, після чого користувача буде перенаправлено на головну сторінку субмодуля “Кафедра” (див. рис. 3.6).

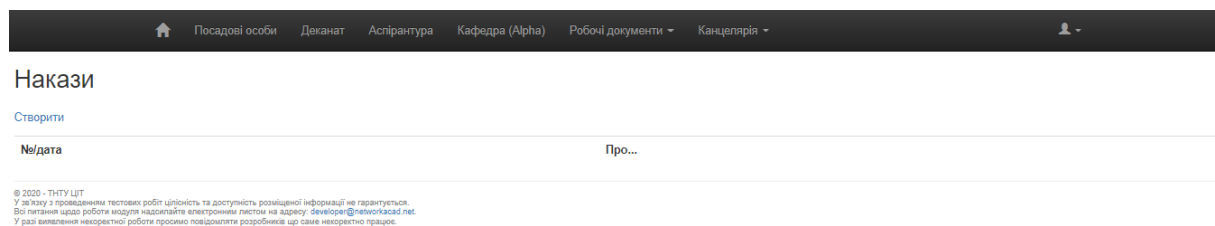


Рис. 3.6 — Головна сторінка субмодуля “Кафедра”

Для створення нового наказу, користувачу потрібно натиснути на кнопку “Створити”. Після чого користувача буде перенаправлено на сторінку створення нового наказу (див. рис. 3.7).

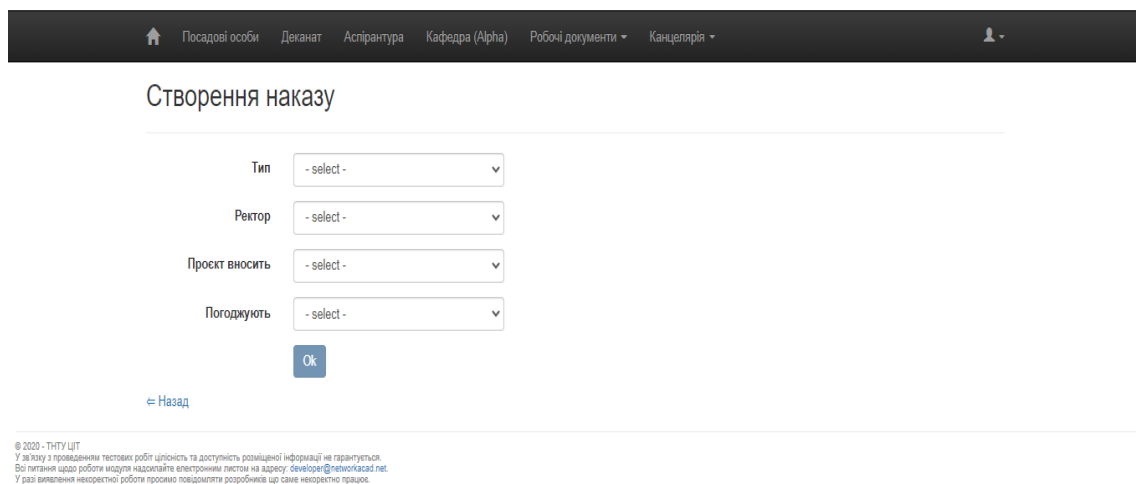


Рис. 3.7 — Сторінка створення нового наказу

Для створення нового наказу потрібно вибрати тип створюваного наказу, ректора (ІПІ персони, що буде записаний в наказі, як ректор університету), ініціатора наказу та погоджувачів (персон, що повинні погодити

наказ перед реєстрацією). Погоджувачів може бути довільна кількість, але не менше трьох. Після внесення всіх необхідних даних потрібно натиснути на кнопку “Ок”. Після чого наказ буде відправлений на погодження, а користувача перенаправлено на головну сторінку субмодуля “Кафедра”, і на ній з’явиться новостворений наказ (див. рис. 3.8).

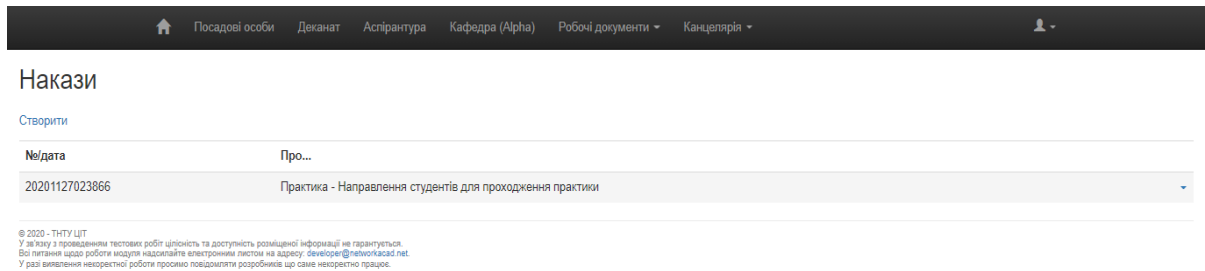


Рис. 3.8 — Головна сторінка субмодуля “Кафедра” з новоствореним наказом

На описаній вище сторінці користувач має можливість відкрити діалогове меню системи по роботі з наказом. Для цього потрібно натиснути на кнопку в правому куті (іконка з стрілочкою вниз) на обраному наказі. Після чого меню буде відкрите (див. рис. 3.9).

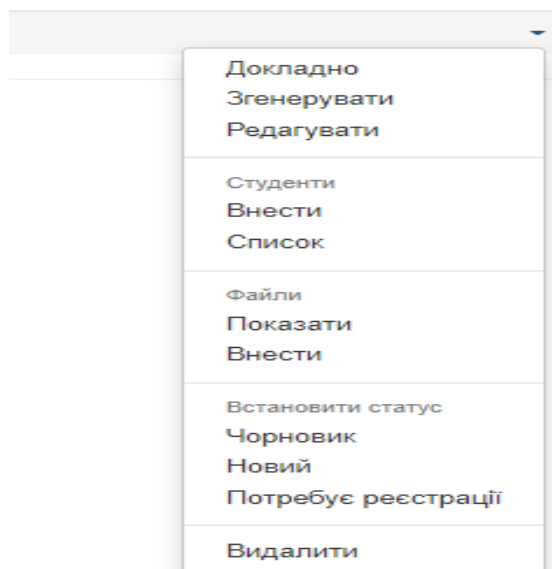


Рис. 3.9 — Діалогове меню системи по роботі з наказом

Діалогове меню системи по роботі з наказом в субмодулі “Кафедра”

містить наступні пункти меню:

- Загальні (докладно, згенерувати, редагувати, видалити).
- Студенти / рецензенти (внести, список).
- Файли (показати, внести).
- Встановити статус (чорновик, новий, потребує реєстрації).

Розглянемо тепер їх детальніше.

Загальні пункти модального меню відповідають за спільний функціонал по наказах.

При натисненні на кнопку “Докладного” загального пункту модального меню відкриється нове модальне вікно з докладною інформацією наказу (див. рис. 3.10).

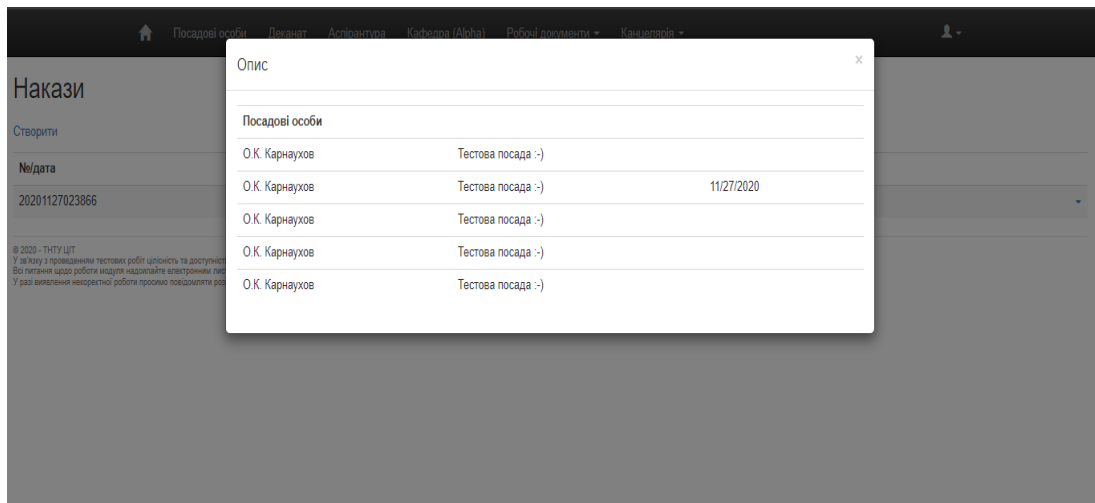


Рис. 3.10 — Модальне вікно з докладною інформацією наказу

При натисненні на кнопку “Згенерувати” відбудеться генерування PDF-документу наказу. Генерування буде відбуватися з використання технології LaTeX, після чого згенерований файл буде збережений в базі даних.

Натиснувши на кнопку “Редагувати”, користувачу буде відкрита сторінка для редагування PDF-документа наказу використовуючи мову форматування LaTeX (див. рис. 3.11). На даній сторінці користувач має такі кнопки для виконання обраних дій: “Назад”, “Зеркальні поля”, “Оновити”, “Зберегти”. Натиснувши на кнопку “Назад” користувача буде перенаправлено на головну сторінку субмодуля “Кафедра”. Кнопка “Оновити” відповідає за

перегенерування PDF-документу наказу, “Зберегти” – за збереження документа в базу даних. При натисненні на кнопку “Зеркальні поля” частини ініціатора наказу та погоджувачів буде перенесено на наступну сторінку PDF-документа відповідно вимог.

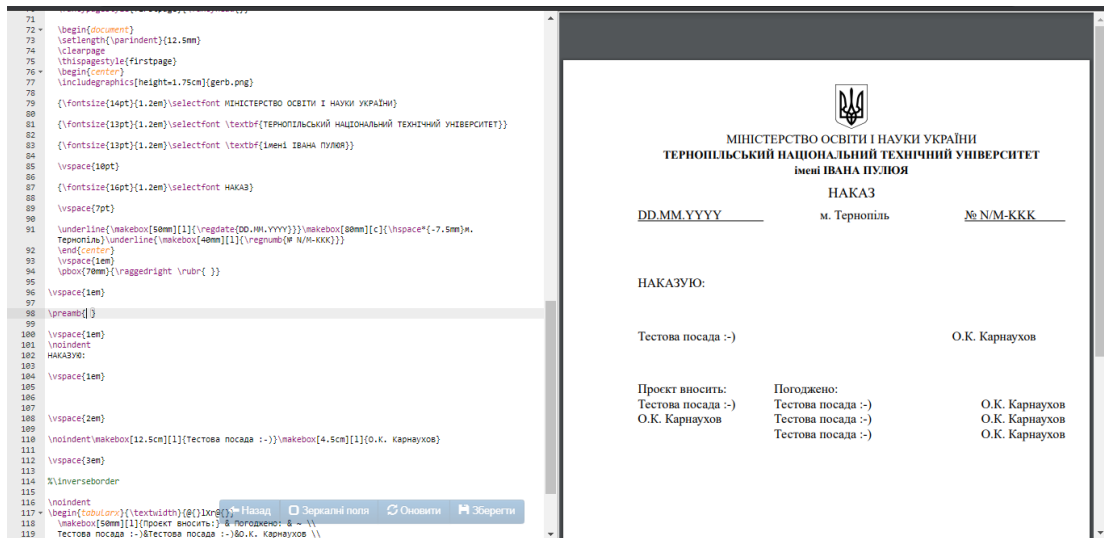


Рис. 3.11 — Сторінка для редагування PDF-документів

Також користувач має можливість видалити наказ. Для цього потрібно натиснути на кнопку “Видалити” після чого з’явиться модальне вікно з підтвердженням (див. рис. 3.12), і якщо обрана дія буде підтверджена, наказ буде видалений.

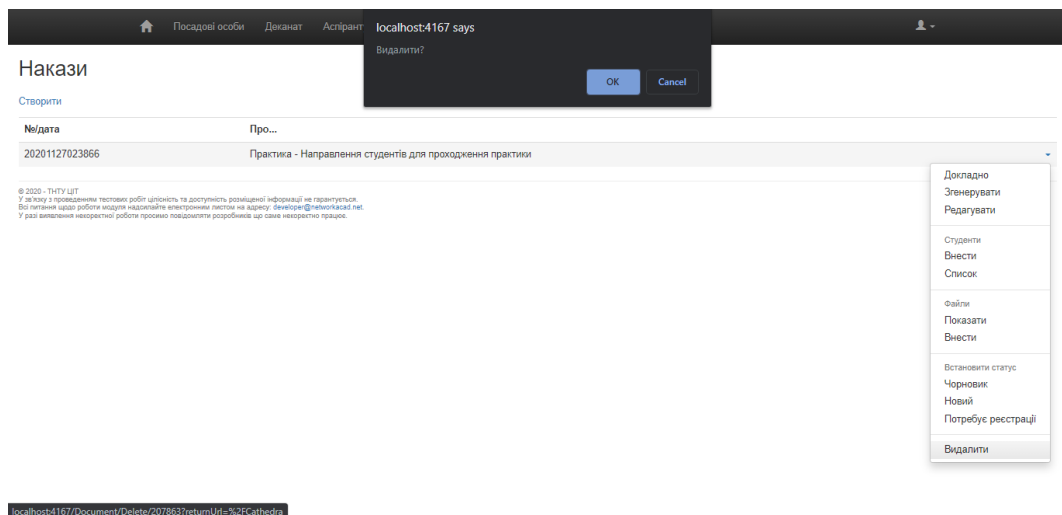
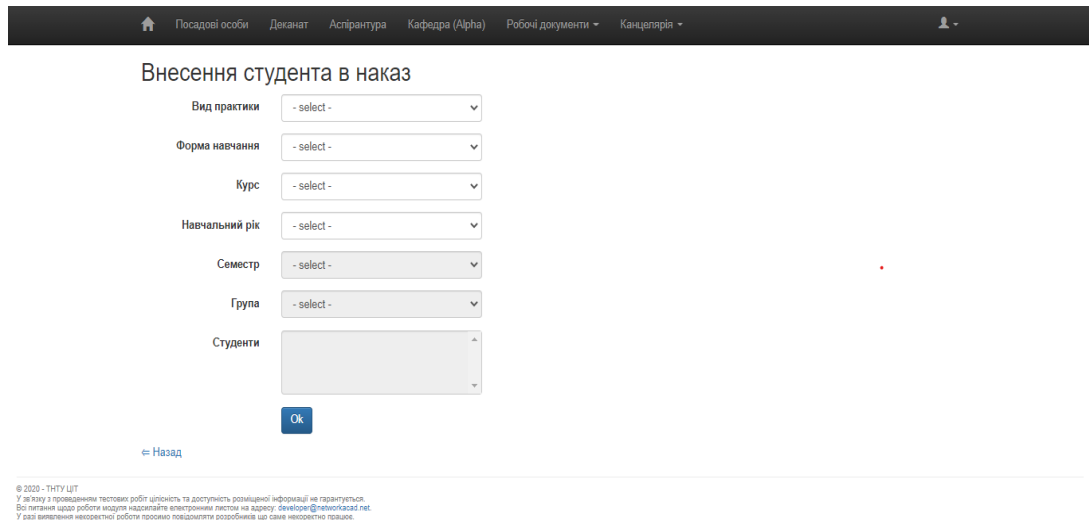


Рис. 3.11 — Попередження про видалення наказу

У модальному меню по роботі з наказом є пункт “Студенти / рецензенти”. У ньому міститься можливість додати персону до наказу, переглянути список персон та видалити персону з наказу. Натиснувши на кнопку

“Внести” користувача буде перенаправлено на сторінку додавання студентів / рецензентів (див. рис. 3.12). Після чого користувачу необхідно обрати: вид практики, форму навчання, курс, навчальний рік, семестр, група, студентів, та натиснути на кнопку “Ок”. У разі внесення всіх даних обраних студентів буде додано до наказу.



The screenshot shows a web interface for adding a student to a decree. The title is "Внесення студента в наказ". The form contains several dropdown menus: "Вид практики", "Форма навчання", "Курс", "Навчальний рік", "Семестр", and "Група", each with a "- select -" option. Below these is a text input field for "Студенти". A blue "Ok" button is positioned below the input field. A "← Назад" link is located at the bottom left of the form area. At the very bottom of the page, there is a small copyright notice: "© 2020 - ТНТУ-ЦІТ. У зв'язку з проведенням тестових робіт цілісність та доступність розкритої інформації не гарантується. Зосередити увагу роботи можуть надсилати електронні листи на адресу: dnevnoi@tntu.edu.ua. У разі виявлення некоректної роботи просимо повідомити розробників цю справу некоректно працює."

Рис. 3.12 — Сторінка внесення студента в наказ

Також, користувач має можливість переглянути список внесених студентів та в разі потреби видалити їх. Для цього в пункті “Студенти” потрібно натиснути на кнопку “Список”. Після чого буде відкрите модальне вікно з списком студентів (див. рис. 3.13).

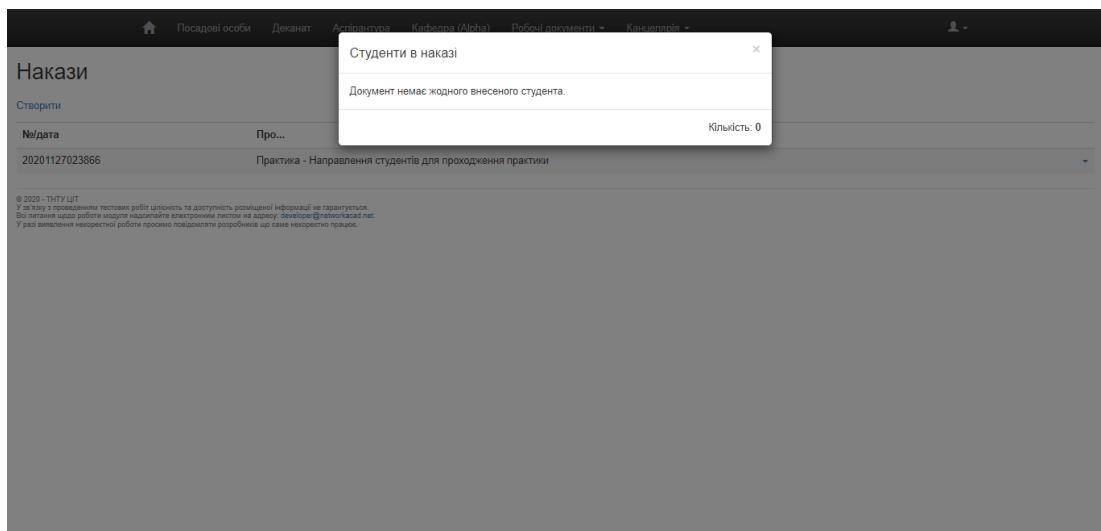


Рис. 3.13 — Модальне вікно зі списком студентів в наказі

Користувачі, також, мають можливість переглянути згенерований PDF-

документ наказу, PDF-документи вкладень та видалити їх. Для цього є пункт меню “Файли”. Для перегляду документу наказу та вкладень потрібно натиснути на кнопку “Показати” користувачу відкриється модальне вікно з переліком документів, що містить наказ (див. рис. 3.14). Також, користувач з використанням інтерфейсу відкритого вікна може завантажити обраний документ або видалити його.

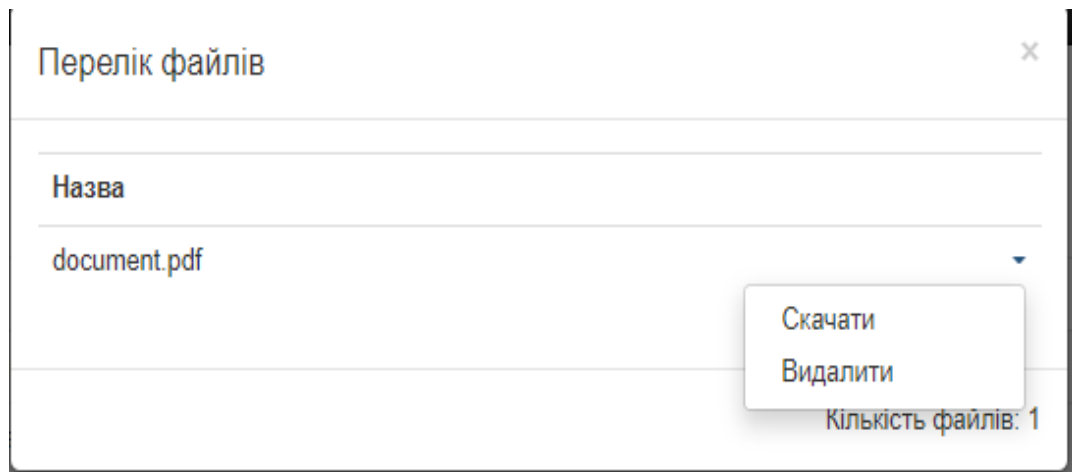


Рис. 3.14 — Модальне вікно з переліком документів, що містить наказ

Останнім пунктом меню даного субмодуля є змінення статусів наказу. Користувач має можливість змінити статус наказу, для цього потрібно в пункті меню “Встановити статус” просто обрати необхідний. Потім статус наказу зміниться, також, колір його на інтерфейсі також зміниться (див. рис. 3.15).

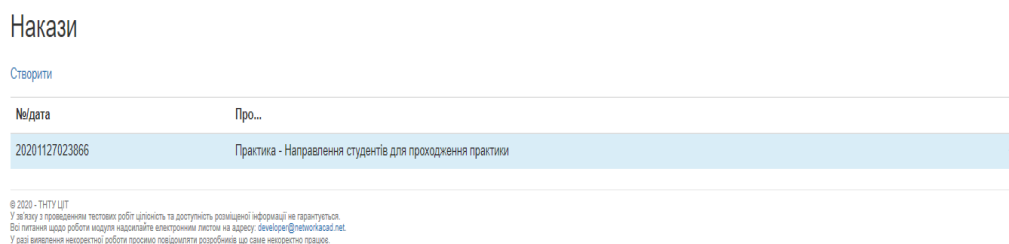


Рис. 3.15 — Результат зміни статусу наказу

Наступним етапом життєвого циклу наказу є відправлення його на погодження. Для цього погоджувачу необхідно перейти по шляху “Робочі документи / Погодження наказів” після чого буде відкрита сторінка зі списком

усіх наказів, що були відправленні на погодження (див. рис. 3.16).

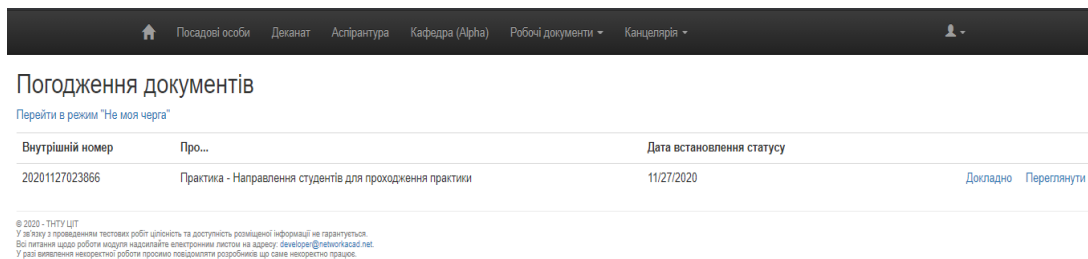


Рис. 3.16 — Сторінка погодження документів

Також, на цій сторінці користувач має можливість переглянути докладну інформацію по наказу та PDF-документ з вкладеннями наказу (див. рис. 3.17). Так як обраний наказ не містить студентів, тому деталі про наказ пусті.

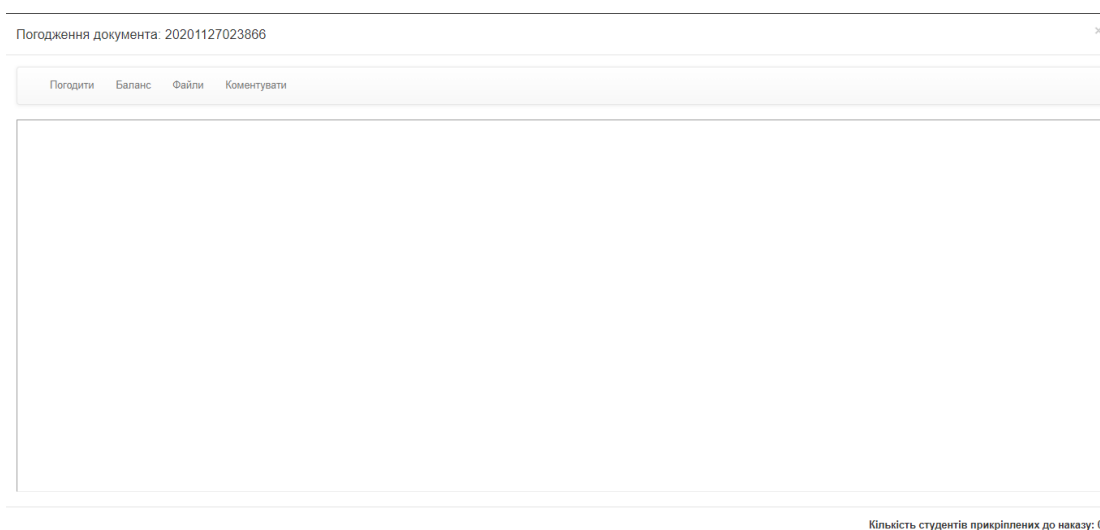


Рис. 3.16 — Сторінка погодження документів та перегляду інформації по наказу

На відкритій сторінці користувач може прокоментувати наказ, переглянути вкладення, та погодити наказ. Після погодження наказ буде відправлено на реєстрацію, що є останнім етапом в життєвому циклі документа.

3.2.2 Тестування програмної системи

Відповідно до специфічної галузі та призначенням розробленого програмного продукту було обрано інтеграційне та ручне тестування щоб перевірити його на відсутність проблем. Інтеграційне тестування буде автоматизовано перевіряти роботу основних компонентів системи. Проведення ручного тестування може робити як тестер, так і кінцевий користувач системи шляхом пошуку та створення можливих сценаріїв використання програмної системи. Головною метою проведення ручного тестування є перевірка коректності роботи розробленої програмної системи, перевірка на відсутність помилок та отримання потрібних результатів роботи програмного продукту.

Якщо врахувати специфіку галузі використання програмної системи є доцільним перевірити процес створення наказів, відповідно шаблонів з використанням автоматизованого виконання інтеграційних тестів (див. рис. 3.17).

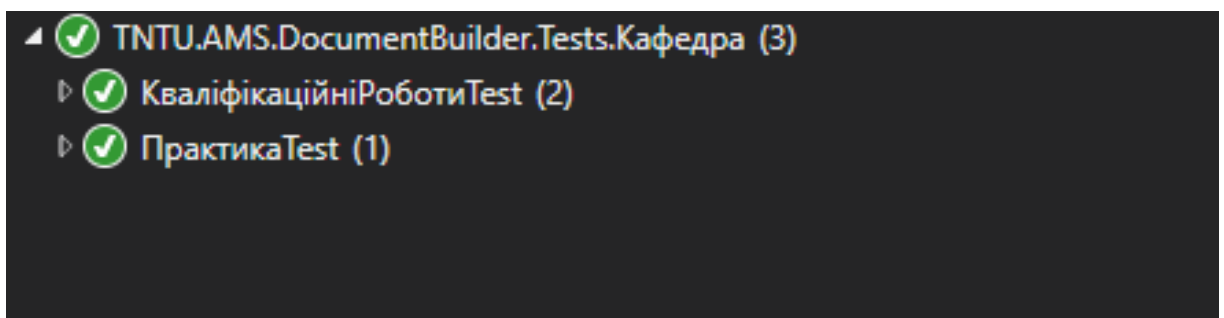


Рис. 3.17 — Результат виконання інтеграційних тестів

Результати перевірки показали, що виконання інтеграційних тестів на субмодулі “Кафедра” модуля “Документообіг”, цілком задовольняють вимогам, поставленим до розробленої програмної системи. Відповідно до результатів проведення ручного тестування було встановлено, що система функціонує коректно та відмінно виконує покладені на неї завдання.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Розробка програмного забезпечення повинна завжди відповідати державним стандартам і здійснювати мінімальну шкоду здоров'ю людині на робочому місці. Метою кваліфікаційної роботи магістра є розробка підмодуля “Кафедра” в модулі електронного документообігу “Документо-обіг” для Тернопільського національного технічного університету імені Івана Пулюя, що тягне за собою виконання великої кількості робіт за електронно-обчислювальною машиною протягом великої кількості часу, через це потрібно дотримуватися норм по охороні праці та техніки безпеки при роботі за електронно-обчислювальною технікою, що регламентуються НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», вимогами та нормами державно-санітарної служби при роботі з дисплеями комп'ютерної техніки ДСанПН 3.3.2.007-98, Санітарними вимогами шуму, інфразвуку та ультразвуку на виробництві ДСН 3.3.6.037-99 та Санітарними мікрокліматичними вимогами до приміщень на виробництві ДСН 3.3.6.042- 99 [25].

Відповідно до нормативного-правового акту описаного вище під час організації користувацького робочого місця електронно-обчислювальної машини, потрібно вибирати таке обладнання, що не буде формувати великої кількості шуму та виділяти багато тепла. Величина виділених шумів на робочому місці персони, що здійснює роботу на електронно-обчислювальній машині, повина бути в нормі відповідно вимог ДСН 3.3.6.037-99. Даний нормативно-правовий документ описує поділ акустичних коливань на виробництві, особливо виділених шумів, інфразвуку та ультразвуку, способи для їхнього вимірювання та устаткування, призначене для цих дій, та дозволені значення відповідно нормативів виділених шумів, інфразвуку та ультразвуку [26].

Облаштування робочого місця розробника підмодуля “Кафедра” в модулі електронного документообігу “Документообіг” має відповідати вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сідачів. Загальні ергономічні вимоги» та згаданими вище вимогам та нормам державно-санітарної служби при роботі з дисплеями комп’ютерної техніки ДСанПІН 3.3.2.007-98 [27].

Облаштування робочого місця персони, що займається розробкою програмного забезпечення повинно відповідати новітнім ергономічним вимогам, особливістю роботи, що буде виконуватися і спроектувати розташування робочого обладнання електронно – обчислювальної машини (монітора, клавіатури, системного блоку, гарнітури «миша» та різних пристроїв периферії враховуючи кількісні одиниці та особливості).

Монітор потрібно встановити так, щоб верхня частина екрана знаходилася паралельно до очей. Розміщення монітора повинне забезпечувати безпеку при роботі загалом, зручність та ефективність роботи очей з ним за кутом близько 30 градусів в вертикальній площині від зорової лінії, а екран має розташовуватися перпендикулярно до лінії зору розробника.

Робочі місця повинні бути сплановані і мати відповідні розміри, щоб у людей був вільний простір для пересування та, щоб вони, відчували себе розслаблено. Площа, що надається для одного робочого місця, має бути не меншою шести метрів квадратних, з об’ємом не меншим двадцяти метрів кубічних.

Мікроклімат в приміщенні розробника з електронно-обчислювальною машиною потрібно підтримувати на сталому рівні та відповідно нормам описаним в ДСН 3.3.6.042-99 [28]. Різні перепади температури в приміщеннях з комп’ютером здійснює негативний вплив на самопочуття його користувача. Недостатньо вологе повітря в приміщенні призводить до надмірної сухості в оболонках очей, горла, носа та до накопичення зарядів статичної електроенергії, що створюється в процесі роботи комп’ютерних систем. Також вологість повітря, яка є більшою за сімдесят п’ять відсотків є

недопустимою. Під час роботи за електронно – обчислювальною машиною оптимальними умовами в приміщенні є температура повітря від +21 градусів за Цельсієм до +22 градусів за Цельсієм та з відносною вологістю від 60 до 40 відсотків, швидкість руху повітря не повинна перевищувати 0,1 метра за секунду.

Отже, під час виконання даної кваліфікаційної роботи магістра в приміщенні було забезпечено необхідні умови праці, відповідно до нормативно-правових актів, описаних вище. Всі показники були в нормі, що дозволило мінімізувати вплив даної розробки на здоров'я розробника.

4.2 Освітлення приміщень під час роботи з комп'ютерною технікою

Всеохоплююча глобальна комп'ютиризація суспільства, загальний процес для всього суспільства та багатогранні модифікації в організації праці підштовхує велику кількість людей використовувати різні технології електронно-обчислювальних машин та вивчати потрібні інформаційні технології. Також, впровадження комп'ютерних систем в усі галузі життєдіяльності людини допомогли виявити як позитивні, так і негативні наслідки від їхнього використання, свідченням цього є скарги користувачів комп'ютера. Найчастішими, є скарги погіршення стану здоров'я та зменшення працездатності людини через погане освітлення.

Вірно спроектоване і розроблене освітлення на виробництві поліпшує умови роботи для зору, знижує стомлюваність організму, корисно впливає на виробниче середовище, допомагає в підвищенні продуктивності праці, надаючи хорошу психологічну дію на працівника, підвищує безпеку на виробництві і зменшує травматизм.

Недостатня освітленість призводить до напруження зору, послабляє увагу, призводить до настання швидкої стомленості. Занадто яскраве світло викликає напруження, подразнення і різкий біль в очах. Невірний напрямок світла на робочому місці працівника може формувати різкі тіні, відблиски

або дезорієнтувати його. Весь цей перелік причин може привести до нещасного випадку на виробництві, вигорання його на займаній посаді, або до професійних захворювань, тому надзвичайно важливим є правильне освітлення приміщення.

Наявні три типи освітлення - природне, штучне і комбіноване (це ситуація, коли природне та штучне використовуються сумарно).

Під час виконання робіт з високим зоровим напруженням (це ситуація коли розмір розпізнаваного об'єкту коливається між 0,3мм до 0,5мм) величина коефіцієнту природного освітлення має бути не меншою 1,5 відсотка, а коли зір працює над середньою точністю (щонайменший розмір розпізнаваного об'єкту коливається між 0,5мм до 1,0 мм) коефіцієнт природнього освітлення повинен бути не меншим 1,0 відсотка Як джерело штучного освітлення звісно можна використовувати ДРЛ, або ЛБ люмінесцентні лампи, які парами об'єднуються в світильники, які мають розташовуватися синхронно над робочою поверхньою працівника.

Штучне освітлення рекомендується розміщувати з обох сторін від екрану паралельно до напрямку зорової лінії. Аби уникнути світових блисків в напрямку очей потрібно використовувати антиблискові сітки, що є спеціальними фільтрами для екрану.

Повинна використовуватися система вимикачів, що дозволить регулювати потужність штучного освітлення в залежності від потужності природного, а також що дозволяє освітлювати тільки необхідні роботі області приміщення. Для надання значень відповідно норм освітлення потрібно очищати скло вікна та світильники не рідше ніж 2 рази рік, та вчасно проводити заміну перегорівших ламп та є непрацездатними.

Також відомо, що протяжна робота за комп'ютерною технікою та з документами при малому рівні світла може призвести до великого напруження очей, через це рівень освітленості є доволі важливими аспектом.

Вимоги до освітленості є встановленими в ДБН В.2.5-28-2006

«Природне і штучне освітлення» та були затверджені нормативно-правовим актом Мінрегіону від 15.05.2006 року № 168.

Відповідно до розташування джерел природнього світла, робоче місце потрібно облаштувати так, щоб джерела природнього світла були по лівій стороні (п. 4.3 ДСанПіН 3.3.2.007-98). Місце працівника потрібно розташовувати так, щоб уникнути попадання прямих сонячних променів в очі. Для підтримання захисту і отримання нормованих рівнів комп'ютерних випромінювань потрібно застосувати приєкранні фільтри, місцевих світлофільтрів (пристосувань для особистого захисту зору) та інших пристосувань для захисту, пройшовші відповідні випробування та є акредитованими в лабораторіях і в яких є відповідний гігієнічний сертифікат (п. 4.19 ДСанПіН 3.3.2.007-98).

Організацію штучного освітлення приміщень потрібно здійснювати системою, що дозволить створити загальне рівномірне освітлення (п. 3.2.2 ДСанПіН 3.3.2.007-98). У виробничих приміщеннях при, в більшій мірі, роботі з паперовими джерелами інформації дозволяють використовувати системи комбінованого освітлення, тобто дозволяють встановлювати світильники локального світла для допомоги загальному.

Джерелами штучного освітлення потрібно використовувати люмінесцентні лампи. Відповідно до норм, описаних в п. 3.2.5 ДСанПіН 3.3.2.007-98 система загального освітлення повинна бути організована за допомогою суцільних або пунктирних ліній світильників, що розташовані по боках від робочих поверхонь (в більшості випадків зліва) паралельно до лінії зору працівників.

Дозволяється використання розжарбвальних ламп у світильниках місцевого світла і, у разі облаштування відбитого світла у виробничих чи адміністративно-громадських будівлях, металогалогенних ламп з максимальною потужністю не більше 250 Вт.

Коефіцієнт пульсації не може більшим ніж 5 відсотків (п. 3.2.14 ДСанПіН 3.3.2.007-98). Величина рівня освітлення на робочій поверхності в

області розміщення документів повинна бути в рамках від 300 люксів до 500 люксів. Світильники локального освітлення потрібно розміщувати таким чином, щоб не були відблиски на поверхні екрана, а світло що падає на екран має не бути більшою ніж 300 люксів.

Для забезпечення значень відповідно норм освітленості у приміщеннях відповідно до п. 3.2.15 ДСанПіН 3.3.2.007-98 потрібно здійснювати миття вікон і світильників не менше ніж 2 рази в рік, а також вчасно замінювати лампи, що перегоріли чи непрацюють.

Колірна гама приміщень та меблів повинна допомагати створенню сприятливих умов для зору та підтримання хорошого настрою.

Джерела природнього та штучного освітлення, які відзеркалюються від поверхні монітора, сильно зменшують точність сприйняття зображення на екрані чи букв клавіатури і створюють внаслідок роботи фізіологічні перешкоди, що проявлятимуться в значній напрузі, особливо при тривалих навантаженнях. Віддзеркалення другорядних джерел освітлення, має бути мінімізоване. Захищаючись від дуже великої яскравості з джерел природнього світла можуть використовуватись штори або захисні екрани.

Описане вище середовище, здійснює суттєвий вплив на працездатність працівників загалом, тому дизайн виробничих будівель, інших елементів середовища є надзвичайно важливим.

Беручи до уваги те, що велика кількість систем пов'язані з людським виконанням розумової роботи, розглянемо дизайн приміщення як елемента навколишнього середовища. Офіс є робочим приміщенням, яке призначене для виконання працівниками певних функцій (трудової діяльності) з максимальною повнотою й у встановлені часові рамки. Офіс, як і будь яка промислова будівля, формується за допомогою технологів та фахівців, що будуть працювати у ньому, щоб як можна більше, наблизити дизайн-проект до конкретних технологічних задач.

Неберучи до уваги специфікацію, різноманітність функцій, виділяють дві основні групи функцій в офісній роботі: концентрація та спілкування.

Відповідно до основних особливостей офісу, співвідношення є змінними в цих категоріях, а пошук оптимального виду освітлення, щоб урівноважити спілкування та увагу, між працівниками, є головним завданням дизайн-проекту зовнішнього середовища [29].

ВИСНОВКИ

У результаті виконання даної кваліфікаційної роботи було розроблено та протестовано субмодуль “Кафедра” в модулі “Документообіг” для автоматизації процесу документообігу для кафедр університету. Для реалізації поставленого технічного завдання було зроблено аналіз розвитку інженерії програмного забезпечення для автоматизації електронного документообігу, моделей розробки даних програмних продуктів та здійснення аналізу перспективності їхнього використання в галузі електронних документів. У першому розділі кваліфікаційної роботи досліджено обставини при яких з’явилася концепція електронного документообігу та було розглянуто основні аспекти, що підтверджують її популярність.

Проаналізувавши предметну область було переглянуто існуючі програмні системи, а саме: «1С: Документообіг», «Парус» та «Дебет Плюс». Перелічені вище системи успішно реалізують автоматизацію процесу електронного документообігу, але й містять значні дефекти, через це описана вище концепція і на даний момент має великий потенціал до покращення.

Як модель розробки було обрано ітеративний метод розробки ПЗ, що практичним методом підтвердило – правильність вибору моделі. Представлена модель, що є зрозумілою та простою в реалізації, чудово підійшла для розробки поставлених технічних завдань. Ітеративний метод дозволив контролювати процес створення та впровадження програмного забезпечення, так як графік розробки має суворі терміни завершення кожної ітерації. В результаті виконання даної кваліфікаційної роботи було доведено, що описаний вище метод розробки програмного забезпечення може бути використаний для проєктів вибраної предметної області з наявними на то ресурсами.

Для розробки даного програмного продукту було використано мову програмування C# та .NET Framework. Як середовище для розробки програмного забезпечення було використано Visual Studio 2019. Мова C# та .NET

Framework дозволили швидко написати якісну програмну систему, що ефективно та надійно справляється зі своїми функціями. Дані технології, ще відомі як такі, що надають можливість швидкої розробки висоякісного, гнучного, тестованого коду та формування зручного користувацького інтерфейсу програмної системи, дали очікуваний результат.

Для графічного інтерфейсу було обрано ASP.NET MVC Framework та Javascript бібліотеку Knockout.js, що дозволили швидко, якісно та ефективно реалізувати субмодуль “Кафедра”.

UML-діаграми були використані для візуалізації, документування та глибшого розуміння розроблюваної програмної системи. Діаграми послідовності допомогли розділити проєкт на маленькі складові частини та побудувати коректні алгоритми виконання поставлених завдань, а діаграми компонентів допомогли коректно побудувати структуру програмного рішення, щоб створити програмне забезпечення, відповідно поставленого технічного завдання, невеликої складності та найвищої якості.

Програмна система вдало пройшла розгортання та тестування на внутрішніх серверах університету, також має величезний потенціал для розширення: проєкт можливо буде покращити в майбутньому, якщо додати до програмного продукту більшої кількості автоматизованих процесів документообігу кафедри, що може допомогти створеному програмному продукту представляти в майбутньому покоління систем новітнього електронного документообігу.

Розроблене програмне забезпечення є успішним проєктом, оскільки виконує усі поставлені технічні завдання, працює коректно, швидко та надійно. У даної програмної системи навний величезний потенціал до покращення та великі шанси зі сторони економічної вигоди.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

Багаточастинний документ

1. Ларин М. В. Управление документацией в организациях / М. В. Ларин М: Научная книга. – 2002. – 286 с.
2. Косовец А. А. Правовое регулирование электронного документооборота / А. А. Косовец // вестник Московского университета. – 1997. – № 4. – С. 46-60.
3. Годин В. В. Управление информационными ресурсами / В. В. Годин, И. Корнеев. – М: Инфра-М. – 1999. – 432 с.
4. Лаба О. В. Основні етапи розвитку електронного діловодства / О. В. Лаба // Бібліотекознавство. Документознавство. Інформологія. – 2011. – № 3. – С. 16-19.

Електронний ресурс

5. Системи електронного документообігу [Електронний ресурс] – Режим доступу: <https://studfile.net/preview/5123927/page:69>.
6. Електронний документ [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Електронний_документ.
7. Історія розвитку систем електронного документообігу. Особливості впровадження систем електронного документообігу в Україні [Електронний ресурс] – Режим доступу: https://studopedia.com.ua/1_53714_z-istorii-rozvitku-sistem-elektronnogo-dokumentooobigu-osoblivosti-vprovadzheniya-elektronnogo-dilovodstva-v-ukraini.html.
8. Особливості роботи з технологією LaTeX. Історія появи і внутрішня філософія LaTeX – [Електронний ресурс] – Режим доступу: https://www.ibm.com/developerworks/ru/library/latex_tutorial_02/index.html.
9. Мова розмітки даних LaTeX [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/LaTeX>.
10. Середовище для роботи TeXworks [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/TeXworks>.

11. Огляд середовища TeXworks [Електронний ресурс] – Режим доступу: <https://ru.scriptcult.com/6849.html>.
12. Сучасні тенденції та нормативно-правова база електронного документообігу в Україні [Електронний ресурс] – Режим доступу: <https://sites.google.com/site/elektrdokumentoobig/normativno-pravova-baza>.
13. 1С: Документообіг [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/1С:Документооборот>.
14. Програмний продукт Парус [Електронний ресурс] – Режим доступу: [https://uk.wikipedia.org/wiki/Парус_\(програмний_продукт\)](https://uk.wikipedia.org/wiki/Парус_(програмний_продукт)).
15. Аналоги програми 1С для обліку в Україні [Електронний ресурс] – Режим доступу: <http://erp-project.com.ua/index.php/ru/poleznye-materialy/publikatsii/razrabotat/739-analogi-programmy-1s-dlya-ucheta-v-ukraine>.
16. Програмний продукт Дебет Плюс [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Дебет_Плюс.
17. Ітеративна модель розробки ПЗ [Електронний ресурс] – Режим доступу: <https://studfile.net/preview/5203612/page:6>.
18. Ітеративна та інкрементна модель розробки і впровадження програмних систем [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/ітеративна_та_інкрементна_розробка.
19. Офіційна документація С# [Електронний ресурс] – Режим доступу: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp>.
20. Загальні відомості та документація платформи .NET [Електронний ресурс] – Режим доступу: <https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/overview>.
21. Платформа ASP.NET Web Forms [Електронний ресурс] – Режим доступу: https://ru.qaz.wiki/wiki/ASP.NET_Web_Forms.
22. Платформа ASP.NET MVC Framework [Електронний ресурс] – Режим доступу: https://ru.wikipedia.org/wiki/ASP.NET_MVC_Framework.
23. Діаграма послідовності [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Діаграма_послідовності.

24. Діаграма пакетів [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Діаграма_пакетів.

25. Огляд багаторівневої архітектури [Електронний ресурс] – Режим доступу: <https://metanit.com/sharp/mvc5/23.5.php>.

26. Наказ 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18#n14>.

27. Санітарні норми виробничого шуму, ультразвуку та інфразвуку [Електронний ресурс] : ДСН 3.3.6.037–99. – [Чинний від 1999–01–12]. – Режим доступу : <http://mozdocs.kiev.ua/view.php?id=1789>.

28. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сідачів. Загальні ергономічні вимоги [Електронний ресурс] – Режим доступу: http://online.budstandart.com/ua/catalog/doc-page?id_doc=71028.

29. Санітарні норми мікроклімату виробничих приміщень [Електронний ресурс] – Режим доступу: URL: <https://zakon.rada.gov.ua/rada/show/va042282-99>.

30. Вимоги до освітлення та облаштування робочого місця користувача ЕОМ [Електронний ресурс] – Режим доступу: <https://conferences.vntu.edu.ua/index.php/itpf/2016/paper/download/1472/1166>.

31. Методичні вказівки до виконання атестаційної роботи магістра студентами усіх форм навчання за спеціальністю 121 – «Інженерія програмного забезпечення» / Упорядники: М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк – Тернопіль: Вид-во ТНТУ імені Івана Пулюя. – 2020. – 51с.

ДОДАТКИ

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра програмної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедру
програмної інженерії
Петрик Михайло Романович
“ ____ “ _____ 2020 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання кваліфікаційної роботи на здобуття освітнього рівня «магістр»

на тему: «Розробка взаємодії модулів : “Кафедра”, “Документообіг”,
згідно вимог, з використанням “.NET Framework” для ТНТУ»

Керівник роботи:
к.т.н., доцент
Цуприк Галина Богданівна
“ ____ “ _____ 2020 р.

Виконавець: студент
групи СПм-61
Оліферчук Олександр Юрійович
“ ____ “ _____ 2020 р.

ЗМІСТ

Вступ

1. Підстави до розробки
2. Призначення до розробки
3. Вимоги до програмного продукту
 - 3.1 Функціональні характеристики
 - 3.2 Склад та параметри технічних засобів
 - 3.3 Інформаційна та програмна сполучність
4. Стадії розробки
5. Програмна документація
6. Порядок контролю та приймання

1 ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального плану на 2020 рік та згідно наказу на виконання кваліфікаційної роботи студента-магістра.

Тема проєкту: «Розробка взаємодії модулів : “Кафедра”, “Документообіг”, згідно вимог, з використанням “.NET Framework” для ТНТУ».

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Кваліфікаційна робота присвячена створенню програмного забезпечення для автоматизації процесу електронного документообігу працівників кафедри з використанням з використанням мови програмування C#, платформи .NET Framework та мови розмітки LaTeX.

Предметом дослідження є інженерія програмного забезпечення для автоматизації електронного документообігу, розвиток методів створення програмних систем даної галузі та розробка системи для автоматизації процесу електронного документообігу працівників кафедри.

Метою даної кваліфікаційної роботи магістра є розробка субмодуля “Кафедра” в модулі “Документообіг”.

За результатами виконаної роботи необхідно отримати програмне забезпечення, що відмінно виконує поставлене на нього завдання, а саме автоматизацію процесу електронного документообігу працівників кафедри.

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1 Функціональні характеристики

Програмне забезпечення має реалізувати наступні можливості:

- можливість створення наказів: “Направлення студентів для проходження практики”, “Призначення рецензентів на кваліфікаційні

роботи”, “Затвердження тем кваліфікаційних робіт, керівників та консультантів” з субмодуля “Кафедра” в модулі “Документообіг”;

- можливість внесення студентів, рецензентів в накази;
- можливість видалення студентів з наказів;
- можливість генерування PDF-документа наказів відповідно шаблонів;
- можливість видалення та регенерування PDF-документів наказів;
- можливість редагування тексту PDF-документа наказів з використанням мови розмітки LaTeX;
- можливість додавання вкладень до наказу;
- можливість перегляду деталей наказів;
- можливість зміни статусу наказів.

3.2 Склад та параметри технічних засобів

Функціонування програмної системи забезпечується: ПК x64, з 2 Гб оперативної пам'яті, встановленою системою Microsoft Windows Server 2008/2008 R2//2012/2012 R2/2016/2019, та з не менш, ніж 5 Гб вільного місця на жорсткому диску.

3.3 Інформаційна та програмна сполучність

Програмний продукт повинен коректно функціонувати під керуванням ОС Microsoft Windows Server 2008/2008 R2//2012/2012 R2/2016/2019. Програмний продукт повинен бути написаний з використанням мови C# та платформи .NET Framework.

4. СТАДІЇ РОЗРОБКИ

В ходів реалізації роботи проєкт повинен пройти крізь наступні стадії розробки:

- аналіз предметної області;
- проєктування архітектури;
- реалізація класів і модулів;

- тестування результатів розробки;
- оформлення супровідної документації;
- здача роботи.

5. ПРОГРАМНА ДОКУМЕНТАЦІЯ

Для програмного продукту повинні бути розроблені наступні документи:

- Пояснювальна записка;
- Технічне завдання;
- Презентаційний матеріал;
- Додатки.

6. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Розроблений програмний продукт має виконувати всі вимоги, що складаються з перерахованих у п. 3.1 характеристик.

Приймання проводиться спеціально створеною екзаменаційною комісією в термін до 23 грудня 2020 р.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

VIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



9–10 грудня 2020 року

**ТЕРНОПІЛЬ
2020**

УДК 004.41

О.Ю. Оліферчук, Г.Б. Цуприк, канд. техн. наук

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

РОЗРОБКА СИСТЕМИ ВЗАЄМОДІЇ МОДУЛІВ: “КАФЕДРА” ТА “ДОКУМЕНТООБІГ” З ВИКОРИСТАННЯМ “.NET FRAMEWORK”

UDC 004.41

O.Y. Oliferchuk, H.B. Tsupryk, Ph.D.

IMPLEMENTATION OF THE .NET FRAMEWORK AND INTERACTION BETWEEN MODULES “ACADEMIC DEPARTMENT” “DOCUMENT MANAGEMENT”

Сучасний розвиток суспільства потребує вдосконалення управління процесом виробництва через використання інформаційних технологій. Сьогодні саме впровадження електронного документообігу є одним з основних засобів безпечного підвищення ефективності, економічності, якості та розширення складу послуг, що надаються. Актуальність теми зумовлена тим, що паперовий документообіг вже не є ефективним при опрацюванні інформації, в зв'язку з чим особлива увага приділяється питанням впровадження електронного документообігу, розробка якого здійснюється при використанні новітніх інформаційних технологій.

В результаті аналізу ринку зроблено висновок про те що доступні аналоги, які будуть покривати всі вимоги та враховувати специфіку програмної системи під конкретного замовника, відсутні. Тому було розроблено систему, яка буде корисною не лише для працівників кафедр університету при формуванні наказів та подальшого їх затвердження, а й для всіх підрозділів, діяльність яких залежить в першу чергу від безпечного, якісного, коректно працюючого електронного документообігу. Оскільки розробка проводилась з врахуванням гнучкості розширення, в перспективі це дасть можливість автоматизації більшої кількості процесів електронного документообігу.

Заплановано, що розробка буде працювати, як окремий субмодуль, модуля “Документообіг” ТНТУ та дозволить покрити всі поставлені в технічному завданні вимоги. Систему розроблено з врахуванням сучасних інформаційних технологій, а саме: написано мовою програмування C# .NET Framework та технології “Latex”. Заявлені інструменти є оптимальними для вирішення поставленого завдання, зокрема через простоту використання та наявність необхідних API методів.

Передбачається, що завершена та протестована система буде розміщена на локальному сервері університету (модуль “Документообіг”, вкладка “Кафедра”) та доступна для використання через веб-клієнт всім хто задіяний при формуванні електронних наказів та подальшому їх супроводі.

Література.

1. Документація .NET Framework [Електронний ресурс] – Режим доступу: URL: <https://docs.microsoft.com/en-us/dotnet/framework>.
2. Нормативна база ТНТУ [Електронний ресурс] – Режим доступу: URL: <https://docs.tntu.edu.ua>.
3. Закон України “Про електронні документи та електронний документообіг” [Електронний ресурс] – Режим доступу: URL: <https://zakon.rada.gov.ua/laws/show/851-15#Text>.

Програмний код

Лістинг коду шаблону “Направлення студентів для проходженняпрактики”:Лістинг коду частини “Part”:

```

<?xml version="1.0"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="Part">
    <xsl:value-of select="PartNumber" />. Направити для проходження
    <xsl:choose>
      <xsl:when test="contains(PracticeType, '&quot;')">
        на практику <xsl:value-of select="PracticeType" />
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="PracticeType" /> практики
      </xsl:otherwise>
    </xsl:choose> студентів
    <xsl:value-of select="Course" /> курсу, групи <xsl:value-of se-
lect="Group" /> строком на <xsl:value-of select="PracticeDuration" />
    <xsl:choose>
      <xsl:when test="PracticeDuration = 1">
        тиждень
      </xsl:when>
      <xsl:when test="PracticeDuration > 1 and
not(PracticeDuration > 4)">
        тижні
      </xsl:when>
      <xsl:when test="PracticeDuration >= 5">
        тижнів
      </xsl:when>
    </xsl:choose> з <xsl:value-of select="PracticeStartDate" /> до
    <xsl:value-of select="PracticeEndDate" /> на перелічені нижче бази
    практики згідно з
    укладеними договорами:\par
    <xsl:for-each select="Students/Student">
      <xsl:value-of select="PartNumber" />.<xsl:value-of se-
lect="Number" />. <xsl:value-of select="FullName" /> -
      <xsl:value-of select="PracticeBase" />.\
      Керівник практики - <xsl:value-of select="PracticeManager"
/>.\par
    </xsl:for-each>
    2. Оплату витрат, пов'язаних з відрядженням студентів та
керівників до

```



```

місце практики, не проводити.\par
3. Загальне керівництво та контроль за проходженням практики
покласти на декана <xsl:value-of select="Faculty" />\
<xsl:value-of select="Dean" />\par
</xsl:template>
</xsl:stylesheet>

```

Лістинг коду частини “Preamble”:

```

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-
result-prefixes="msxsl">
  <xsl:template match="Preamble">
    <xsl:text>Відповідно до навчального плану підготовки фахівців
спеціальності</xsl:text>\
    <xsl:value-of select="SpecialityCode" /> ``<xsl:value-of se-
lect="SpecialityName" />'', освітнього рівня <xsl:value-of se-
lect="Qualification" />,
    <xsl:value-of select="FormOfTraining" /> форми навчання, <xsl:value-
of select="Faculty" />,-
  </xsl:template>
</xsl:stylesheet>

```

Лістинг коду частини “Topic”:

```

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-
result-prefixes="msxsl">
  <xsl:template match="Topic">
    <xsl:text>Про направлення студентів на практику</xsl:text>
  </xsl:template>
</xsl:stylesheet>

```

Лістинг коду шаблону “Призначення рецензентів на кваліфікаційні роботи”:

Лістинг коду частини “Part”:

```

<?xml version="1.0" encoding="utf-8"?>

```

```

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-
prefixes="msxsl">
  <xsl:template match="Part">
    <xsl:value-of select="PartNumber" />. Перелічених нижче осіб
призначити рецензентами кваліфікаційних робіт
    здобувачів освітнього ступеня ``<xsl:value-of se-
lect="Qualification" />' ' спеціальності <xsl:value-of se-
lect="SpecialityCode" />
    ``<xsl:value-of select="SpecialityName" />'':\par
    <xsl:for-each select="Reviewers/Reviewer">
      <xsl:value-of select="PartNumber" />.<xsl:value-of se-
lect="Number" />. <xsl:value-of select="FullName" /> - <xsl:value-of se-
lect="Position" />;\par
    </xsl:for-each>
    2. Контроль за виконанням наказу покласти на декана
    <xsl:value-of select="Faculty" />\ <xsl:value-of select="Dean"
/>.
  </xsl:template>
</xsl:stylesheet>

```

Лістинг коду частини “Preamble”:

```

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-
prefixes="msxsl">
  <xsl:template match="Preamble">
    <xsl:text>Відповідно до Положення про кваліфікаційні роботи
студентів
          Тернопільського національного технічного
університету імені Івана Пулюя, -
    </xsl:text>
  </xsl:template>
</xsl:stylesheet>

```

Лістинг коду частини “Topic”:

```

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-
prefixes="msxsl">
  <xsl:template match="Topic">
    <xsl:text>Про призначення рецензентів кваліфікаційних
робіт</xsl:text>

```

```
</xsl:template>
</xsl:stylesheet>
```

Лістинг коду шаблону “Затвердження тем кваліфікаційних робіт, керівників та рецензентів”:

Лістинг коду частини “Part”:

```
<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-
  prefixes="msxsl">
  <xsl:template match="Part">
    <xsl:value-of select="PartNumber" />. Затвердити теми
кваліфікаційних робіт та призначити керівників і
консультантів студентам групи <xsl:value-of select="Group"
/>:\par
    <xsl:for-each select="Students/Student">
      <xsl:choose>
        <xsl:when test="QualificationWorkConsultant != ''">
          <xsl:value-of select="PartNumber"
/>.<xsl:value-of select="Number" />. <xsl:value-of select="FullName" />
-
          <xsl:value-of select="QualificationWorkTheme"
/>.\
          Керівник: <xsl:value-of se-
lect="QualificationWorkManager" />.\
          Консультант: <xsl:value-of se-
lect="QualificationWorkConsultant" />.\par
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of select="PartNumber"
/>.<xsl:value-of select="Number" />. <xsl:value-of select="FullName" /> -
            <xsl:value-of select="QualificationWorkTheme"
/>.\
            Керівник: <xsl:value-of se-
lect="QualificationWorkManager" />.\par
            </xsl:otherwise>
          </xsl:choose>
        </xsl:for-each>
      <xsl:for-each select="Consultants/Consultant">
        <xsl:value-of select="PartNumber + position()" />.
Консультантом з розділу
        ``<xsl:value-of select="CounselingSection" />'' призначити
```

```

        <xsl:value-of select="Position" />\ <xsl:value-of se-
lect="ShortName" />.\par
    </xsl:for-each>
    <xsl:value-of select="EndPartNumber" />. Проведення нормоконтролю
покласти на
    <xsl:value-of select="NormControlConsultant" />.\par
    <xsl:value-of select="EndPartNumber + 1" />. Контроль за
виконанням наказу покласти на декана
    <xsl:value-of select="Faculty" />\ <xsl:value-of select="Dean" />
</xsl:template>
</xsl:stylesheet>

```

Лістинг коду частини “Preamble”:

```

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-
prefixes="msxsl">
    <xsl:template match="Preamble">
        <xsl:text>Відповідно до навчального плану підготовки фахівців
спеціальності</xsl:text> \
        <xsl:value-of select="SpecialityCode" /> ``<xsl:value-of se-
lect="SpecialityName" />'', освітнього рівня <xsl:value-of se-
lect="Qualification" />,
        <xsl:value-of select="FormOfTraining" /> форми навчання,
<xsl:value-of select="Faculty" />,
        які закінчили повний теоретичний курс навчання за освітньо-
професійною програмою,-
    </xsl:template>
</xsl:stylesheet>

```

Лістинг коду частини “Topic”:

```

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-
prefixes="msxsl">
    <xsl:template match="Topic">
        <xsl:text>Про затвердження тем кваліфікаційних робіт,
керівників та консультантів</xsl:text>
    </xsl:template>
</xsl:stylesheet>

```

Диск