

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

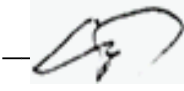
Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

—  — проф. Приходько С.Б.

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

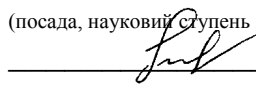
на здобуття ступеня вищої освіти «бакалавр»

на тему: Розробка вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Виконав: студент групи 4151

 Джуринський М.О.
(підпис, ПІБ)

Керівник роботи:

доцент кафедри ПЗАЗ, к.т.н., доцент
(посада, науковий ступень вчене звання)
 Макарова Л.М.
(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
 Гарант освітньої програми
 _____ доц. Макарова Л.М.
 (підпис)
 « 08 » _____ 04 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту Джуринському Максиму Олексійовичу _____
 (Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» _____

Керівник роботи Макарова Л.М. _____

Затверджені наказом ректора №153 від 08.04.2025 р.

2. Термін подання роботи: 02.06.2025 р. _____

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____
 Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами);
 Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз
 практичної задачі інженерії програмного забезпечення; Проект програмного забезпечення; Результати
 розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел;
 Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та
 методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів Титульний лист, мета кваліфікаційної роботи, аналіз вимог до вебзастосунку, архітектура вебзастосунку, діаграми діяльності, концептуальна модель, діаграма класів, діаграми взаємодії, логічна модель БД, робочий проект, результати розробки, висновки, заключний слайд. _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

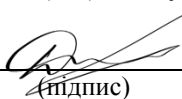
7. Дата видачі завдання 08.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	31.03.2025	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	07.04.2025	ПП*
3.	Аналіз вимог до ПЗ	14.04.2025	ПП*
4.	Розробка архітектури ПЗ	21.04.2025	
5.	Розробка проекту ПЗ	05.05.2025	
6.	Реалізація та тестування ПЗ	12.05.2025	
7.	Підготовка розділу Результати розробки ПЗ	16.05.2025	
8.	Підготовка розділу з охорони праці	19.05.2025	ПП*
9.	Підготовка розділу ВИСНОВКИ	23.05.2025	
10.	Оформлення списку використаних джерел та додатків	26.05.2025	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	02.06.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12.	Підготовка презентації та доповіді	06.06.2025	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	09.06.2025	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	16.06.2025	

* - за результатами переддипломної практики (ПП), яка була з 03.02.2025 до 23.03.2025 р.

Студент



(підпис)

Джуринський М.О.

(ПІБ)

Керівник роботи



(підпис)

Макарова Л.М.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена вирішенню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ». Проведено аналіз практичної задачі інженерії програмного забезпечення з розробки вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» та проаналізовано існуючі вебзастосунки аналоги. Було здійснено постановку задачі на розробку вебзастосунку. Виконано розробку вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ». Описано результати розробки. Проведено опис аспектів охорони праці на робочому місці.

Кваліфікаційна робота викладено на 116 сторінках машинописного тексту та містить: 32 рисунків, 52 таблиці, 5 додатків, список використаних джерел з 17 найменувань.

ABSTRACT

The qualification work is dedicated to solving a practical problem of software engineering, characterized by complexity and uncertainty of conditions, namely the development of a web application for financial analysis of debtors for the law firm "LEMBERG". An analysis of the practical problem of software engineering on the development of a web application for financial analysis of debtors for the law firm "LEMBERG" was carried out and existing analogue web applications were analyzed. The task for the development of a web application was formulated. The development of a web application for financial analysis of debtors for the law firm "LEMBERG" was carried out. The results of the development are described. A description of aspects of occupational safety in the workplace is provided.

The qualification work is presented on 116 pages of typewritten text and contains: 32 figures, 52 tables, 5 appendices, a list of sources used from 17 names.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ ФІНАНСОВОГО АНАЛІЗУ ДЕБІТОРІВ ДЛЯ ЮРИДИЧНОЇ КОМПАНІЇ «ЛЕМБЕРГ» ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1 Аналіз практичної задачі інженерії програмного забезпечення для розробки вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	10
1.2 Аналіз існуючих програмних продуктів	11
1.3 Постановка задачі на розробку вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	14
2 РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ФІНАНСОВОГО АНАЛІЗУ ДЕБІТОРІВ ДЛЯ ЮРИДИЧНОЇ КОМПАНІЇ «ЛЕМБЕРГ»	18
2.1 Аналіз вимог до вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	18
2.1.1 Розробка моделі варіантів використання вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	18
2.1.2 Розробка специфікацій прецедентів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	21
2.2 Розробка архітектури вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	32
2.3 Проєкт вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	38
2.3.1 Ескізний проєкт вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	38
2.3.2 Технічний проєкт вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	44

2.3.3 Робочий проєкт вебзастосування для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»	52
3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБЗАСТОСУВАННЯ ДЛЯ ФІНАНСОВОГО АНАЛІЗУ ДЕБІТОРІВ ДЛЯ ЮРИДИЧНОЇ КОМПАНІЇ «ЛЕМБЕРГ»	73
4 ОХОРОНА ПРАЦІ	75
4.1 Вступ	75
4.2 Аналіз небезпечних та шкідливих умов, які можливі відділі розробки програмного забезпечення	76
4.3 Розробка заходів по зменшенню дії небезпечних і шкідливих факторів у відділі розробки програмного забезпечення	77
ВИСНОВКИ	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
Додаток А – Технічне завдання	82
Додаток Б – Опис програмного забезпечення	89
Додаток Г – Керівництво користувача	97
Додаток Д – Код програмного забезпечення	108
Додаток Е – Акт впровадження	116

ВСТУП

У світі ефективне управління фінансами є ключовим фактором успішного життя компанії. Автоматизація фінансових процесів не лише скорочує час на обробку даних, що забезпечує точність, оперативність та прозорість фінансової аналітики.

В практиці фінансового менеджменту на підприємствах найбільш часто виникають проблеми, пов'язані з дебіторською заборгованістю. Це відсутність інформації про терміни погашення дебіторської заборгованості; неефективна оцінка кредитоспроможності покупців (іноді взагалі відсутність такої оцінки); відсутність відділу з управління дебіторською заборгованістю, що призводить до розподілу функцій аналізу дебіторської заборгованості, отримання грошових коштів, прийняття рішення про товарне кредитування між різними підрозділами підприємства; відсутність автоматизованого відстеження виникнення та погашення дебіторської заборгованості; відсутність профілактичних заходів щодо виникнення неконтрольованих обсягів заборгованості дебіторів; неефективна політика управління інкасацією (погашенням) дебіторської заборгованості у зв'язку з уникненням застосування різнопланових методів рефінансування [1].

Тоді виникає потреба створення вебзастосунку, який дозволить відстежувати стан рахунків компанії, аналізувати банківські виписки, виявляти боржників та формувати аналітичні звіти. Важливим аспектом є інтеграція рішення в існуючу фінансову систему компанії, забезпечення його зручності, гнучкості та масштабованості.

З збільшенням обсягів фінансових операцій та кількості клієнтів компанії стикаються з проблемами ведення обліку дебіторської заборгованості. Стандартні методи, що являють собою використання таблиць чи сторонніх програм, не завжди дозволяють швидко отримати актуальні дані та виявити проблемні рахунки.

Розвиток інформаційних технологій відкриває можливості для автоматизації процесів, що дозволяють підвищити точність розрахунків, зменшити втручання людини і своєчасно реагувати на фінансові ризики. Саме тому розробка вебзастосунку для фінансового аналізу дебіторів є надзвичайно важливим та актуальним завданням.

Метою даної кваліфікаційної роботи є створення вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ», який дозволить автоматизувати процеси фінансового аналізу та управління дебіторською заборгованістю.

Для реалізації вебзастосунку необхідно виконати такі завдання:

- Аналіз практичної задачі інженерії програмного забезпечення. Дослідити існуючі рішення для обліку дебіторської заборгованості та фінансового аналізу.
- Розробка вимог до вебзастосунку. Визначення функціональних та нефункціональних вимог до вебзастосунку.
- Проектування вебзастосунку. Розробити архітектуру вебзастосунку, що забезпечить зручність, ефективність роботи з фінансовими даними.
- Реалізація функціональних модулів. Створити інтерфейс користувача, розробити модулі для роботи з банківськими виписками, управління рахунками та генерації аналітичних звітів.
- Тестування. Перевірити коректність роботи застосунку, оцінити його продуктивність та масштабованість.

Однією з ключових особливостей даної роботи є комплексність та невизначеність умов. Зокрема, комплексність розробки полягає в необхідності:

- створення інтуїтивно зрозумілого інтерфейсу, який буде простим у використанні навіть для людей без технічного досвіду;
- інтеграції роботи з інформацією про компанії, рахунки, виписки та клієнтські дані;
- забезпечення високого рівня безпеки, оскільки робота з фінансами вимагає конфіденційності та захисту від несанкціонованого доступу.

Невизначеність умов пов'язана з тим, що вимоги до вебзастосунку можуть змінюватися в процесі розробки, оскільки вони залежать від технологічних рішень та очікувань користувачів. Це вимагає гнучкого підходу до проектування системи, щоб забезпечити її адаптивність та можливість масштабування відповідно до майбутніх потреб компанії.

У результаті виконання кваліфікаційної роботи буде створено вебзастосунок, який дозволить:

- автоматизувати аналіз банківських виписок та фінансових операцій компанії;
- ефективно відстежувати дебіторську заборгованість та виявляти боржників;
- формувати аналітичні звіти для фінансових директорів та керівництва компанії;
- покращити контроль над фінансовими потоками та зменшити ризики фінансових втрат.

Завдяки цьому компанія «ЛЕМБЕРГ» зможе підвищити ефективність управління фінансами, скоротити витрати часу на обробку фінансових операцій та зменшити час ухвалення рішень.

Об'єктом кваліфікаційної роботи є процес розробки вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ».

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ ФІНАНСОВОГО АНАЛІЗУ ДЕБІТОРІВ ДЛЯ ЮРИДИЧНОЇ КОМПАНІЇ «ЛЕМБЕРГ» ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз практичної задачі інженерії програмного забезпечення для розробки вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Дана кваліфікаційна робота присвячена вирішенню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ».

Практична задача інженерії програмного забезпечення охоплює розробку вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ», що пов'язаний з веденням дебіторської заборгованості та банківських виписок у сфері послуг, є важливим аспектом фінансового управління компаній. Організації, що надають послуги, зіштовхуються з різними завданнями, які вирішуються в цій галузі:

- Ведення обліку дебіторської заборгованості. Систематичний облік та моніторинг всіх фінансових зобов'язань, які клієнти мають перед компанією, є основним етапом в управлінні дебіторською заборгованістю. Це включає в себе відстеження термінів оплати та розробку стратегій для уникнення прострочених платежів.

- Аналіз дебіторської заборгованості. Глибокий аналіз фінансових звітів та дебіторської заборгованості допомагає виявити патерни, тенденції та можливі проблеми. Це дозволяє компаніям приймати обґрунтовані рішення щодо управління фінансами та мінімізації ризиків.

- Пошук боржників, які не внесли оплату за надані послуги. Важливо виявляти та вирішувати ситуації, де клієнти не виконали свої фінансові зобов'язання. Ефективна система виявлення боржників дозволяє вчасно реагувати та запобігати можливим фінансовим втратам.

- Створення аналітики на основі представлених даних. Використання сучасних технологій та аналітичних інструментів для створення комплексної аналітики. Це може включати в себе побудову звітів, візуалізацію даних та прогнозування розвитку дебіторської заборгованості.

Враховуючи ці аспекти, створення вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» стає ключовим елементом ефективного управління фінансами та забезпечення стійкості бізнесу в сфері надання послуг.

1.2 Аналіз існуючих програмних продуктів

На ринку існує ряд програмних продуктів, які дозволяють аналізувати дебіторську заборгованість та банківські виписки. До основних аналогів можна віднести наступні.

Програмний продукт «BAS Бухгалтерія»

Розробник: компанія "NetHelp"

Програмний продукт «BAS Бухгалтерія» включає технологічну платформу і конфігурацію (прикладне рішення) «BAS Бухгалтерія».

«BAS Бухгалтерія» призначена для автоматизації бухгалтерського і податкового обліку, включаючи підготовку обов'язкової (регламентованої) звітності, в організаціях, що здійснюють будь-які види комерційної діяльності: гуртову і роздрібну торгівлю, комісійну торгівлю (включаючи субкомісію), надання послуг, виробництво і т.д. Бухгалтерський і податковий облік ведеться відповідно до чинного законодавства України.

Правила ведення обліку настроєні саме в конфігурації.

«BAS Бухгалтерія» забезпечує вирішення всіх завдань, що стоять перед бухгалтерською службою підприємства, якщо бухгалтерська служба повністю відповідає за облік на підприємстві, включаючи, наприклад, виписку первинних документів, облік продажів тощо. Це прикладне рішення також можна використовувати виключно для ведення бухгалтерського та податкового обліку, а завдання автоматизації інших служб, наприклад, відділу продажів, вирішувати спеціалізованими конфігураціями або іншими системами [2]. Інтерфейс програми зображений на рисунку 1.1.

Дата	Надходження	Списання	Призначення платежу	Контрагент	Вид операції	Вх. номер	Вх. дата	Організація	Банківський рахунок	Коментар
04.04.2015	30 240,00			Будівельник	Оплата від покупця			Добро	Р/р Добро	
06.04.2015		10 196,24		Казначейство	Перерахування податків/...			Добро	Р/р Добро	
06.04.2015		27 043,13		Казначейство	Перерахування податків/...			Добро	Р/р Добро	
06.04.2015		2 606,71		Казначейство	Перерахування податків/...			Добро	Р/р Добро	
06.04.2015		1 086,10		Казначейство	Перерахування податків/...			Добро	Р/р Добро	
06.04.2015		3 256,33		Казначейство	Перерахування податків/...			Пілот ТД	Р/р Пілот	
06.04.2015		8 669,32		Казначейство	Перерахування податків/...			Пілот ТД	Р/р Пілот	
06.04.2015		833,45		Казначейство	Перерахування податків/...			Пілот ТД	Р/р Пілот	
06.04.2015		347,27		Казначейство	Перерахування податків/...			Пілот ТД	Р/р Пілот	
06.04.2015		3 976,47		Казначейство	Перерахування податків/...			ПП Немешев О...	АПБ "АВАЛЬ"...	
06.04.2015		10 767,35		Казначейство	Перерахування податків/...			ПП Немешев О...	АПБ "АВАЛЬ"...	
06.04.2015		1 035,51		Казначейство	Перерахування податків/...			ПП Немешев О...	АПБ "АВАЛЬ"...	
06.04.2015		431,43		Казначейство	Перерахування податків/...			ПП Немешев О...	АПБ "АВАЛЬ"...	
06.04.2015		482,21		Казначейство	Перерахування податків/...			ПП Петренко	Р/р Петренко	
06.04.2015		1 225,88		Казначейство	Перерахування податків/...			ПП Петренко	Р/р Петренко	
06.04.2015		120,06		Казначейство	Перерахування податків/...			ПП Петренко	Р/р Петренко	
06.04.2015		50,02		Казначейство	Перерахування податків/...			ПП Петренко	Р/р Петренко	
06.04.2015		58 519,56		АВАЛЬ	Перерахування зарплатної...			Добро	Р/р Добро	
06.04.2015		23 319,89		АВАЛЬ	Перерахування зарплатної...			ПП Немешев О...	АПБ "АВАЛЬ"...	
06.04.2015		2 682,00		АВАЛЬ	Перерахування зарплатної...			ПП Петренко	Р/р Петренко	
19.11.2018		500,00	Оплата за договором Основной...	Бахшиев Павел Иннокентьевич	Оплата поставщику			Добро	Р/р Добро	
27.02.2020		450 000,00	Оплата за договором Контракт на поставку бы...	Koenig Holding GmbH	Оплата поставщику			Добро	ПАТ "РАЙФФА...	
31.03.2021		500,00		Пенсійний	Перерахування підзвітній ...	6	09.05.2022	Добро	ПАТ "РАЙФФА...	
20.06.2022		600,00	Оплата за договором тест...	Тест	Оплата поставщику			Добро	Р/р Добро	
02.08.2022		1 800,00	Оплата за договором тест...	Тест пда	Оплата поставщику			Добро	Р/р Добро	

Рисунок 1.1 – Інтерфейс ПЗ «BAS Бухгалтерія»

Плюси: широкий функціонал, інтеграція з іншими продуктами BAS, можливість роботи в хмарі.

Мінуси: висока вартість, складність у використанні.

Програмний продукт «Finmar»

Розробник: Олександр Соловей, Іван Каунов та Дмитро Дубілет.

Finmar - це онлайн-інструмент для управління фінансами сучасного бізнесу.

З його допомогою можливо:

- Збирати всі дані про ваші рахунки, надходження і витрати в одному місці, завдяки інтеграціям з банками і сервісами.
- Аналізувати ваш грошовий потік, прибуток, проєкти, борги за допомогою наочних графіків і звітів, які створюються автоматично.
- Планувати ваше фінансове майбутнє, запобігаючи касовим розривам і створюючи прогнози на будь-який період.
- Виставляти красиві рахунки-фактури прямо в Finmar.
- Надавати доступ до вашої фінансової інформації вибірково, налаштовуючи рівні доступу для співробітників.
- Гарантувати безпеку і анонімність вашої фінансової інформації, використовуючи захищену хмару і шифрування.

Finmar підходить для малих і середніх бізнесів, яким потрібен простий і зручний інструмент для обліку грошей [3]. Інтерфейс програми зображений на рисунку 1.2.

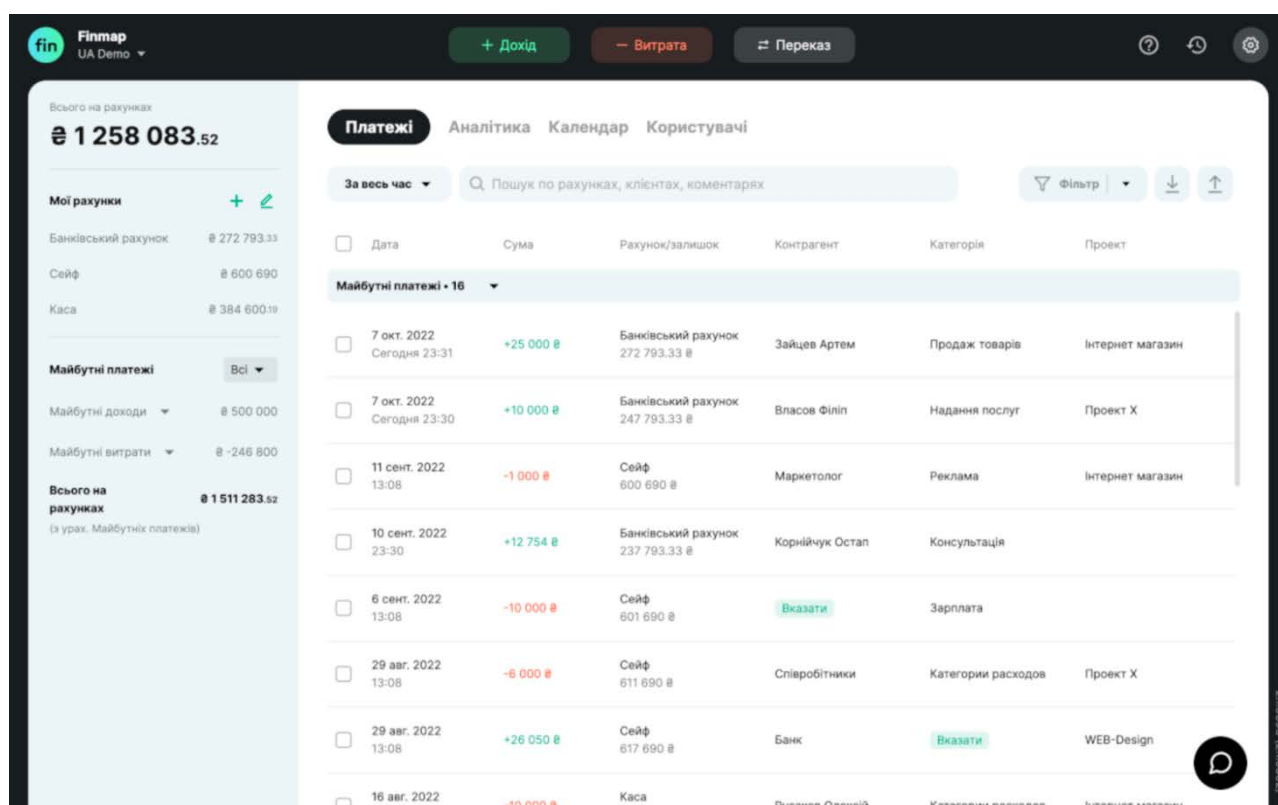


Рисунок 1.2 – Інтерфейс вебзастосунку «Finmar»

Плюси: простий у використанні, можливість роботи в хмарі, є мобільний застосунок.

Мінуси: не можна зробити аналітику по декільком рахункам компанії.

1.3 Постановка задачі на розробку вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Проаналізувавши існуючі програмні рішення, можна зробити висновок про необхідність створення вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ».

Вимоги до складу виконуваних функцій:

- Реєстрація та авторизація. Користувач повинен мати можливість створити новий обліковий запис, щоб мати можливість працювати з платформою. Якщо він вже має створений аккаунт він має можливість увійти до свого кабінету.
- Створювати компанію та додавати всі банківські рахунки, які має ця компанія. Користувач повинен мати можливість у своєму кабінеті додати власну компанію і при створенні цієї компанії у застосунку повинен мати можливість створювати банківські рахунки, які є у цієї компанії.
- Для кожної компанії можна додавати або видаляти працівників даної компанії, які мають можливість працювати з даними. Користувач повинен мати можливість після створення компанії додавати довірених осіб, які мають можливість переглядати, редагувати, видаляти та створювати нові записи. Користувач повинен мати можливість видаляти довірених осіб, що позбавляє їх доступу до даних.
- Можливість редагувати інформацію про компанію та додавати нові банківські рахунки. Користувач повинен мати можливість при необхідності змінювати будь яку інформацію про компанії, додавати нові банківські рахунки.
- Переглядати дані, редагувати, видаляти та створювати нові записи по банківських виписках по всіх банківських рахунках та компаніям. Користувач повинен переглядати дані, редагувати, видаляти та створювати нові записи по

банківських виписках по всіх банківських рахунках та компаніям, якщо виникає необхідність оновлення даних, або імпортування їх з інших платформ.

- Переглядати дані, редагувати, видаляти та створювати нові записи щодо наданих послуг компанією. Користувач повинен мати можливість переглядати записи, редагувати записи, видаляти записи та створювати нові записи наданих послуг компанією.

- Видалення компаній та банківських рахунків: Користувач повинен мати можливість видаляти створені компанії та банківські рахунки, які вже не будуть оброблятися.

- Формування дебіторських заборгованостей по кожній компанії: Користувач повинен мати можливість формувати «Дебіторську заборгованість» на основі отриманих записів банківської виписки та записів наданих послуг.

- Формування аналітичних звітів (місячних, квартальних, річних) по банківських виписках по кожній компанії. Користувач повинен мати можливість формувати аналітичні звіти по існуючих у системі банківським випискам.

- Формування аналітичних звітів (місячних, квартальних, річних) по наданим послугам по кожній компанії. Користувач повинен мати можливість формувати аналітичні звіти по існуючих у системі даних наданих послуг компанією.

Вимоги до організації вхідних і вихідних даних

Вхідні дані:

- Реєстрація. Для створення власного облікового запису Користувач повинен ввести такі дані: ПІБ, email, телефон, пароль. Після введення цих даних користувач повинен підтвердити свій телефон та email, вказавши код у спеціальні поля надісланий на телефон у формі SMS, та електронний лист надісланий на email.

- Авторизація. Для входження до свого облікового запису Користувач повинен ввести email/ телефон та пароль.

- Дані про компанію. Користувач при створенні компанії повинен ввести назву своєї компанії, код ЄДРПОУ та створити банківські рахунки.

- Дані про банківські рахунки. При створенні банківського рахунку користувач повинен вказати до якого банку відноситься даний банківський рахунок, назву банківського рахунку, IBAN.

Вихідні дані:

- Звіт з дебіторської заборгованості надається у виді таблиці та надає таку інформацію: загальна заборгованість, дані про боржників(та скільки вони винні)

- Аналітика по банківській виписці Cash flow (CF) та The profit and loss (P&L) надається у виді таблиці та надає таку інформацію: загальні доходи/витрати та всі доходи/витрати, класифіковані по категоріям та підкатегоріям.

- Аналітика по послугам, наданим компанією надається у виді таблиці та надає таку інформацію: загальні доходи та доходи класифіковані по категоріям та підкатегоріям.

Вимоги до складу й характеристик технічних засобів

Сервер повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 4-х ядерний процесор з тактовою частотою 2,0 ГГц.
- Оперативна пам'ять: 32 Гб.
- Жорсткий диск: 1Тб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

Комп'ютер користувача повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 2-х ядерний процесор з тактовою частотою 1,8 ГГц.
- Оперативна пам'ять: 4 Гб.
- Жорсткий диск: 100 Гб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

Вимоги до інформаційної й програмної сумісності

На сервері повинна бути встановлена одна з операційних систем: Windows 11 не нижче версії 21H2, Linux Ubuntu не нижче версії 22.04.3 LTS, Linux Debian не нижче версії 12, Linux Fedora не нижче версії 38 або MacOS версії не нижче Monterey. На серверному пристрої повинен бути встановлений Docker Desktop не нижче версії 4.24.0.

На комп'ютері користувача повинна бути встановлена одна з операційних систем: Windows 11 не нижче версії 21H2. Linux Ubuntu не нижче версії 22.04.3 LTS, Linux Debian не нижче версії 12, Linux Fedora не нижче версії 38 або MacOS версії Monterey. На клієнтському пристрої повинен бути встановлений веб-браузер Google Chrome не нижче версії 117, або Mozilla Firefox не нижче версії 118.0.

Більш детально постановка задачі викладена у технічному завданні, що наведено у додатку А.

2 РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ФІНАНСОВОГО АНАЛІЗУ ДЕБІТОРІВ ДЛЯ ЮРИДИЧНОЇ КОМПАНІЇ «ЛЕМБЕРГ»

Після виконання аналізу практичної задачі програмної інженерії, можна переходити до розробки вебзастосунку.

У даному розділі кваліфікаційної роботи буде представлено: аналіз вимог до вебзастосунку, розробка архітектури вебзастосунку, проєкт вебзастосунку.

2.1 Аналіз вимог до вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

2.1.1 Розробка моделі варіантів використання вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Діаграма варіантів використання – це діаграма, що зображує різні сценарії чи шляхи використання системи або ПЗ з точки зору користувача. Ця діаграма допомагає зрозуміти, як повинна працювати система чи ПЗ при використанні нею користувачем.

Виявимо акторів. Після проведення аналізу практичної задачі програмної інженерії, актором даного застосунку буде користувач.

Користувач - це особа, яка має можливість додавати компанії, банківські рахунки, операції, банківські виписки, працівників, контрагентів та змінювати інформацію про них та видаляти дані. На основі доданих даних операцій та банківських виписок може отримувати аналітику та дані про кредиторів та дебіторів.

Виявлені варіанти використання вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» наведені в таблиці 2.1.

Таблиця 2.1 – Варіанти використання вебзастосунку

№	Назва	Короткий опис
1	Авторизація	Дозволяє авторизуватись у системі
2	Реєстрація	Дозволяє реєструватись у системі
3	Додавання нових компаній	Дозволяє додавати нові компанії до системи.
4	Редагування інформації про компанії	Дозволяє змінювати дані про існуючі компанії.
5	Видалення компаній	Дозволяє видаляти дані про компанії.
6	Додавання банківських рахунків	Дозволяє додавати нові банківські рахунки до системи.
7	Редагування рахунків	Дозволяє редагувати інформацію про банківські рахунки.
8	Видалення рахунків	Дозволяє видаляти банківські рахунки з системи.
9	Формування аналітичних звітів	Дозволяє створювати звіти на основі банківських виписок, операцій та даних компаній.
10	Додавання нових операцій	Дозволяє додавати нові фінансові операції.
11	Редагування інформації про операції	Дозволяє змінювати дані про фінансові операції.
12	Видалення операцій	Дозволяє видаляти фінансові операції з системи.
13	Додавання банківських виписок	Дозволяє додавати нові банківські виписки до системи.
14	Редагування банківських виписок	Дозволяє змінювати дані про існуючі банківські виписки.
15	Видалення банківських виписок	Дозволяє видаляти дані про банківські виписки з системи.
16	Додавання контрагентів	Дозволяє додавати нових контрагентів.
17	Редагування інформації про контрагентів	Дозволяє змінювати інформацію про контрагентів.
18	Видалення контрагентів	Дозволяє видаляти контрагентів з системи.
19	Додавання нових працівників	Дозволяє додавати нових працівників компанії.
20	Видалення працівників	Дозволяє видаляти працівників із системи.
21	Перегляд інформації	Дозволяє переглядати дані про компанії, рахунки, операції, працівників, контрагентів, банківські виписки та сформовані аналітичні звіти.

Ці варіанти використання допомагають розуміти, як користувач буде взаємодіяти з вебзастосунком для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» та які функції та можливості повинні бути реалізовані для задоволення його потреб. Аналіз і уточнення цих варіантів використання допомагає забезпечити ефективну та задовільну взаємодію користувачів з вебзастосунком. Рисунок 2.1 відображає діаграму варіантів використання для користувача.

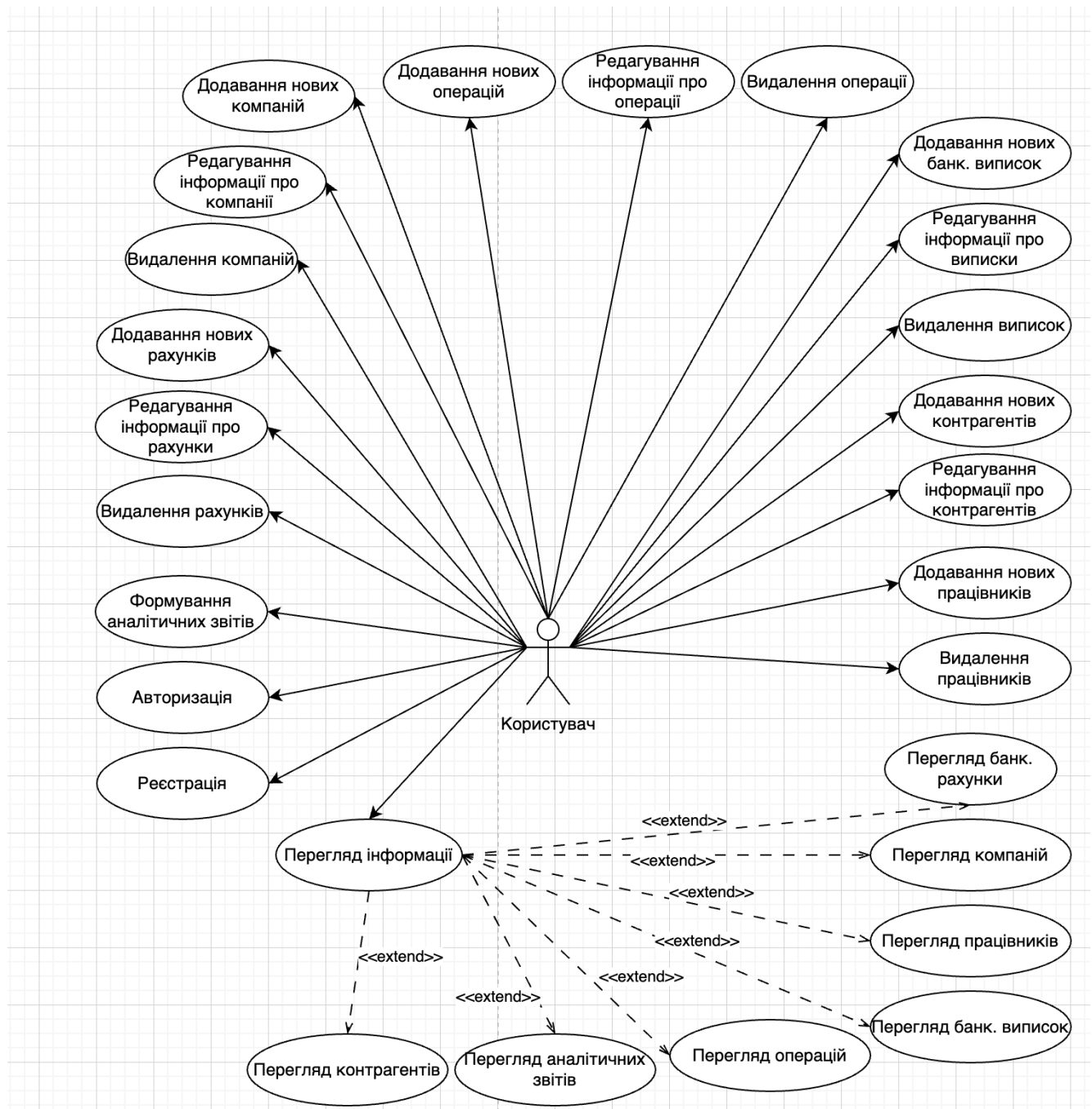


Рисунок 2.1 – Діаграма варіантів використання

Далі можна приступити до розробки специфікацій прецедентів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ».

2.1.2 Розробка специфікацій прецедентів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Специфікацій прецедентів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» представлено у таблицях 2.2 – 2.27.

Таблиця 2.2 – Специфікація варіанту використання "Додавання нових компаній"

Складова	Опис
Опис преценденту	Додавання нових компаній
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач ініціалізує відкриття меню компаній 2. Система відкриває меню компаній 3. Користувач надсилає запит на дію "Створити компанію" 4. Перед користувачем система показує сторінку з потрібними полями для заповнення 5. Після заповнення всіх полів Користувач відправляє запит на збереження нової компанії 6. Система зберігає нову компанію і підтверджує успішне створення.
Альтернативні потоки	Немає
Постумови	Система додала нову компанію у БД

Таблиця 2.3 – Специфікація варіанту використання "Редагування інформації про компанії"

Складова	Опис
Опис преценденту	Редагування інформації про компанії
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач ініціалізує відкриття меню компаній. 2. Система відкриває список компаній. 3. Користувач надсилає запит на редагування інформації обраної компанії, натискаючи кнопку "Шестерня" справа від компанії. 4. Система відкриває сторінку з потрібними полями для редагування. 5. Користувач заповнює потрібні поля та надсилає запит на збереження змін. 6. Система зберігає оновлену інформацію про компанію і підтверджує успішне редагування.
Альтернативні потоки	Немає
Постумови	Система відредагувала інформацію з обраної компанії в БД

Таблиця 2.4 – Специфікація варіанту використання "Видалення компаній"

Складова	Опис
Опис преценденту	Видалення компаній
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач ініціалізує відкриття налаштувань і надсилає запит на відкриття меню компаній. 2. Система відкриває меню компаній. 3. Користувач надсилає запит на перегляд дій для обраної компанії, скролячи донизу сторінки. 4. Система показує кнопку "Видалити компанію". 5. Користувач надсилає запит на видалення компанії, натиснувши на кнопку "Видалити компанію". 6. Система показує форму підтвердження видалення. 7. Користувач надсилає запит на підтвердження видалення компанії. 8. Система видаляє обрану компанію та підтверджує видалення.
Альтернативні потоки	Немає
Постумови	Обрана компанія була видалена з БД

Таблиця 2.5 – Специфікація варіанту використання "Додавання нових рахунків"

Складова	Опис
Опис преценденту	Додавання нових рахунків
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач надсилає запит на вибір компанії в меню компаній. 2. Система відображає інформацію про вибрану компанію. 3. Користувач надсилає запит на створення нового рахунку, натиснувши кнопку "Додати новий рахунок". 4. Система відкриває сторінку з полями для заповнення. 5. Користувач заповнює необхідні поля та надсилає запит на збереження рахунку, натиснувши кнопку "Зберегти". 6. Система зберігає новий рахунок і підтверджує успішне створення.
Альтернативні потоки	Немає
Постумови	Система додала новий рахунок у БД

Таблиця 2.6 – Специфікація варіанту використання "Редагування інформації про рахунки"

Складова	Опис
Опис преценденту	Редагування інформації про рахунки
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач надсилає запит на відкриття меню компаній. 2. Система відкриває меню компаній. 3. Користувач надсилає запит на редагування рахунку, натискаючи кнопку "Переглянути" справа від рахунку. 4. Система відкриває віконце з полями для редагування. 5. Користувач заповнює необхідні поля та надсилає запит на збереження змін, натискаючи кнопку "Зберегти". 6. Система зберігає зміни в рахунку і підтверджує успішне редагування.
Альтернативні потоки	Немає
Постумови	Система відредагувала інформацію з обраного рахунку в БД

Таблиця 2.7 – Специфікація варіанту використання "Видалення рахунків"

Складова	Опис
Опис преценденту	Видалення рахунків
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач надсилає запит на відкриття налаштувань і відкриття меню компаній. 2. Система відкриває меню компаній. 3. Користувач скролить до списку рахунків обраної компанії та надсилає запит на вибір рахунку. 4. Користувач надсилає запит на видалення рахунку, натискаючи кнопку "Видалити" справа від рахунку. 5. Система показує форму підтвердження видалення. 6. Користувач надсилає запит на підтвердження видалення рахунку. 7. Система видаляє обраний рахунок і підтверджує видалення.
Альтернативні потоки	Немає
Постумови	Обраний рахунок було видалено з БД

Таблиця 2.8 – Специфікація варіанту використання "Додавання нових операцій"

Складова	Опис
Опис преценденту	Додавання нових операцій
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач надсилає запит на відкриття меню компаній. 2. Система відкриває меню компаній. 3. Користувач надсилає запит на вибір компанії. 4. Система відображає інформацію про вибрану компанію. 5. Користувач надсилає запит на відкриття розділу "Операції". 6. Система відкриває розділ "Операції". 7. Користувач надсилає запит на додавання нового запису, натиснувши кнопку "Додати новий запис". 8. Система відкриває сторінку з полями для заповнення. 9. Користувач заповнює необхідні поля та надсилає запит на збереження запису, натиснувши кнопку "Зберегти". 10. Система зберігає нову операцію та підтверджує успішне додавання.
Альтернативні потоки	Немає
Постумови	Система додала нову операцію в БД

Таблиця 2.9 – Специфікація варіанту використання "Редагування інформації про операції"

Складова	Опис
1	2
Опис преценденту	Редагування інформації про операції
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач надсилає запит на відкриття меню компаній. 2. Система відкриває меню компаній. 3. Користувач надсилає запит на вибір компанії. 4. Система відображає інформацію про вибрану компанію. 5. Користувач надсилає запит на відкриття розділу "Операції". 6. Система відкриває розділ "Операції". 7. Користувач надсилає запит на вибір потрібного запису операцій. 8. Система відображає інформацію про вибраний запис, включаючи кнопку "Редагувати запис". 9. Користувач надсилає запит на редагування запису, натиснувши кнопку "Редагувати запис". 10. Система відкриває сторінку з полями для редагування. 11. Користувач заповнює необхідні поля та надсилає запит на збереження змін, натиснувши кнопку "Зберегти". 12. Система зберігає зміни до запису операцій та підтверджує успішне редагування.

Продовження таблиці 2.9

1	2
Альтернативні потоки	Немає
Постумови	Система відредагувала інформацію з обраним записом операцій у БД

Таблиця 2.10 – Специфікація варіанту використання "Видалення операцій"

Складова	Опис
Опис прецеденту	Видалення компаній
Дійові особи прецеденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Операції" 4. Користувач обирає потрібний запис і справа від нього буде кнопка "Видалити запис" 5. Користувач натискає кнопку "Видалити запис" 6. При натисненні на неї перед користувачем з'являється форма підтвердження видалення 7. Після підтвердження обраний запис операції буде видалений
Альтернативні потоки	Немає
Постумови	Обраний запис операції було видалено з БД

Таблиця 2.11 – Специфікація варіанту використання "Додавання нових банківських виписок"

Складова	Опис
Опис прецеденту	Додавання нових банківських виписок
Дійові особи прецеденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач обирає рахунок 4. Перед користувачем відкривається банківська виписка 5. Користувач натискає кнопку "Додати новий запис" 6. Перед Користувачем з'являється сторінка з потрібними полями для заповнення 7. Після заповнення всіх полів користувач повинен натиснути кнопку "Зберегти"
Альтернативні потоки	Немає
Постумови	Система додала новий запис банківської виписки у БД

Таблиця 2.12 – Специфікація варіанту використання "Редагування інформації про виписки"

Складова	Опис
Опис преценденту	Редагування інформації про виписки
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач обирає рахунок 4. Перед Користувачем відкривається банківська виписка 5. Користувач обирає потрібний запис 6. Користувач натискає кнопку "Редагувати запис" 7. Перед Користувачем з'являється сторінка з потрібними полями для заповнення 8. Після заповнення всіх полів Користувач повинен натиснути кнопку "Зберегти"
Альтернативні потоки	Немає
Постумови	Система відредагувала інформацію обраного запису банківської виписки у БД

Таблиця 2.13 – Специфікація варіанту використання "Видалення виписок"

Складова	Опис
Опис преценденту	Видалення виписок
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач обирає рахунок 4. Перед Користувачем відкривається банківська виписка 5. Користувач обирає потрібний запис і справа від нього буде кнопка "Видалити запис" 6. Користувач натискає кнопку "Видалити запис" 7. При натисненні на неї перед Користувачем з'являється форма підтвердження видалення 8. Після підтвердження обраний запис виписки буде видалений
Альтернативні потоки	Немає
Постумови	Обраний запис банківської виписки було видалено з БД

Таблиця 2.14 – Специфікація варіанту використання "Додавання нових контрагентів"

Складова	Опис
Опис преценденту	Додавання нових контрагентів
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Контрагенти" 4. Користувач натискає кнопку "Додати нового контрагента" 5. Перед Користувачем з'являється сторінка з потрібними полями для заповнення 6. Після заповнення всіх полів Користувач повинен натиснути кнопку "Зберегти"
Альтернативні потоки	Немає
Постумови	Система додала нового контрагента у БД

Таблиця 2.15 – Специфікація варіанту використання "Редагування інформації про контрагента"

Складова	Опис
Опис преценденту	Редагування інформації про контрагента
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Контрагенти" 4. Користувач обирає потрібний запис і справа від нього буде кнопка "Редагувати запис" 5. Користувач натискає кнопку "Редагувати запис" 6. Перед Користувачем з'являється сторінка з потрібними полями для заповнення 7. Після заповнення всіх полів Користувач повинен натиснути кнопку "Зберегти"
Альтернативні потоки	Немає
Постумови	Система відредагувала інформацію з обраним контрагентом у БД

Таблиця 2.16 – Специфікація варіанту використання "Додавання нових працівників"

Складова	Опис
Опис преценденту	Додавання нових працівників
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Працівники" 4. Користувач натискає кнопку "Додати новий запис" 5. Перед Користувачем з'являється сторінка з потрібними полями для заповнення 6. Після заповнення всіх полів користувач повинен натиснути кнопку "Зберегти"
Альтернативні потоки	Немає
Постумови	Система додала нового працівника у БД

Таблиця 2.17 – Специфікація варіанту використання "Видалення працівників"

Складова	Опис
Опис преценденту	Видалення працівників
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Працівники" 4. Користувач обирає потрібний запис і справа від нього буде кнопка "Видалити запис" 5. Користувач натискає кнопку "Видалити запис" 6. При натисненні на неї перед користувачем з'являється форма підтвердження видалення 7. Після підтвердження обраного працівника буде видалено і позбавлено прав
Альтернативні потоки	Немає
Постумови	Обраного працівника було видалено і позбавлено прав взаємодії з інформацією

Таблиця 2.18 – Специфікація варіанту використання "Формування аналітичних звітів"

Складова	Опис
Опис преценденту	Формування аналітичних звітів
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Аналітика" 4. Користувач натискає кнопку "Сформувати звіт"
Альтернативні потоки	Немає
Постумови	Система сформувала аналітичний звіт

Таблиця 2.19 – Специфікація варіанту використання "Перегляд компаній"

Складова	Опис
Опис преценденту	Перегляд компаній
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває налаштування 2. Після Користувач відкриває меню "Компанії"
Альтернативні потоки	Немає
Постумови	Система показала всі компанії та їх дані

Таблиця 2.20 – Специфікація варіанту використання "Перегляд банківських рахунків"

Складова	Опис
Опис преценденту	Перегляд банківських рахунків
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває налаштування 2. Після Користувач відкриває меню "Компанії" 3. Після вибору певної компанії обирає "Показати банківські рахунки"
Альтернативні потоки	Немає
Постумови	Система показала всі банківські рахунки та їх дані

Таблиця 2.21 – Специфікація варіанту використання "Перегляд контрагентів"

Складова	Опис
Опис преценденту	Перегляд контрагентів
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Контрагенти"
Альтернативні потоки	Немає
Постумови	Система показала всі контрагенти компанії

Таблиця 2.22 – Специфікація варіанту використання "Перегляд працівників"

Складова	Опис
Опис преценденту	Перегляд працівників
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Працівники"
Альтернативні потоки	Немає
Постумови	Система показала всіх працівників компанії

Таблиця 2.23 – Специфікація варіанту використання "Перегляд банківських виписок"

Складова	Опис
Опис преценденту	Перегляд банківських виписок
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач обирає рахунок
Альтернативні потоки	Немає
Постумови	Система показала записи банківської виписки рахунку

Таблиця 2.24 – Специфікація варіанту використання "Перегляд операцій"

Складова	Опис
Опис преценденту	Перегляд операцій
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Операції"
Альтернативні потоки	Немає
Постумови	Система показала всі записи операцій компанії

Таблиця 2.25 – Специфікація варіанту використання "Перегляд аналітичних звітів"

Складова	Опис
Опис преценденту	Перегляд аналітичних звітів
Дійові особи преценденту	Користувач
Предумови	Авторизація
Базовий потік	1. Користувач відкриває меню компаній 2. Користувач обирає компанію 3. Користувач натискає кнопку "Аналітика"
Альтернативні потоки	Немає
Постумови	Система показала аналітичні дані компанії

Таблиця 2.26 – Специфікація варіанту використання "Авторизація"

Складова	Опис
Опис преценденту	Авторизація
Дійові особи преценденту	Користувач
Предумови	Немає
Базовий потік	1. Прецедент починається коли користувач починає роботу з застосунком. 2. Користувач вводить логін та пароль від свого облікового запису. 3. Система перевіряє надані дані на збіг з даними у базі даних. 4. Система авторизує користувача та перенаправляє на домашню сторінку застосунку
Альтернативні потоки	Немає
Постумови	Користувача було авторизовано

Таблиця 2.27 – Специфікація варіанту використання "Реєстрація"

Складова	Опис
Опис преценденту	Реєстрація
Дійові особи преценденту	Користувач
Предумови	Немає
Базовий потік	1. Прецедент починається коли користувач починає роботу з застосунком. 2. Якщо користувач немає свого облікового запису. Користувач переходить на сторінку реєстрації вводить такі дані: ім'я, email, телефон, пароль. 3. Система перевіряє надані дані на валідацію. 4. Система реєструє користувача та перенаправляє на сторінку авторизації.
Альтернативні потоки	Немає
Постумови	Користувача було зареєстровано

Після розробки специфікацій варіантів використання вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» переходимо до розробки архітектури.

2.2 Розробка архітектури вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Відповідно до вимог вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» для проектування вебзастосунку було обрано шаблон проектування MVC (Model-View-Controller).

Модель надає дані та методи роботи з ними: запити в базу даних, перевірка на коректність введених даних. Модель не залежить від вигляду та контролера (немає точок взаємодії з користувачем), вона має доступ до даних та може керувати ними.

Модель має такі ознаки:

- це бізнес-логіка додатку;
- містить відомості про себе і не містить відомостей про контролер та подання;
- для деяких проектів – це просто шар даних (DAO, БД, XML-файл);

- для інших проектів – це менеджер бази даних, набір об'єктів або просто логіка додатку.

Подання відповідає за отримання необхідних даних з моделі і відправляє їх користувачу. Подання не обробляє введені дані користувача.

Подання має такі ознаки:

- реалізується відображення даних, що отримуються від моделі;
- в деяких випадках, подання може мати код, який реалізує деяку бізнес-логіку.

Прикладом подання є: HTML сторінка, WPF форма, Windows Forms тощо.

Контролер забезпечує зв'язок між системою та користувачем, контролює і направляє дані від користувача до системи і навпаки, використовує модель та подання для реалізації необхідної дії.

Контролер має такі ознаки:

- визначає, які подання повинні бути відображені на даний час;
- події подання можуть впливати тільки на контролер, контролер може вплинути на модель і визначити інше подання;
- є можливість створити декілька подань для одного контролера.[4]

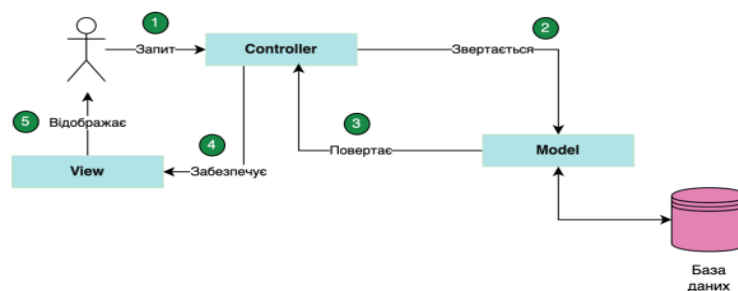


Рисунок 2.2 – Діаграма взаємодії використання патерну MVC

Розробка діаграми компонентів для вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Діаграма компонентів – це структурна діаграма мови уніфікованого моделювання, вона визначає особливості фізичного уявлення системи. Діаграма

компонентів дозволяє визначити архітектуру системи, що розробляється, встановивши залежності між програмними компонентами.[5]

Діаграму компонентів для вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» представлено на рисунку 2.3.

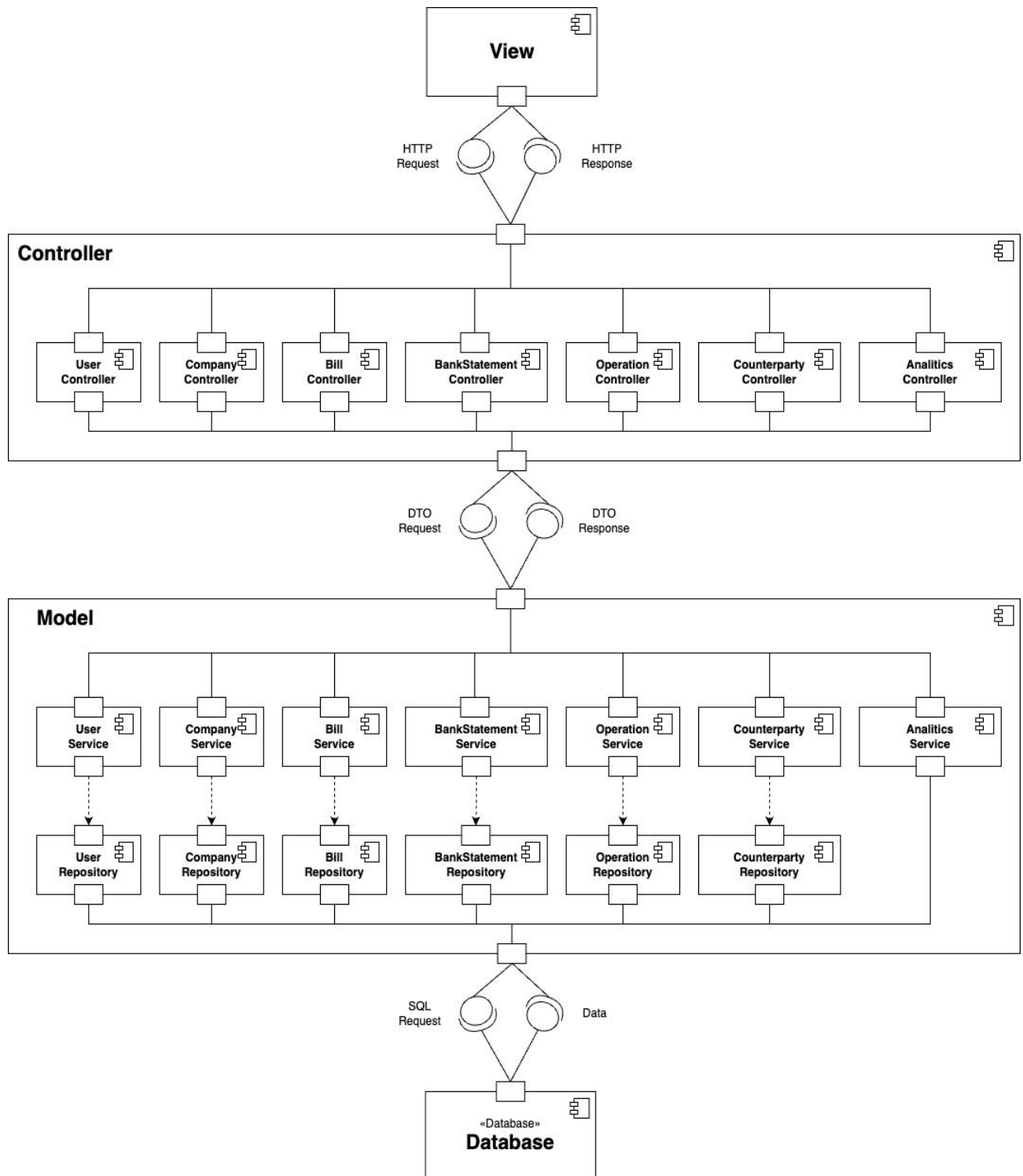


Рисунок 2.3 – Діаграма компонентів

Model (Модель)

Модель представляє логіку та дані ПЗ і відповідає за управління станом даних, маніпуляцію ними та бізнес-логіку. Основна роль моделі полягає у:

Зберіганні даних: Модель містить структури, що описують сутності системи, і забезпечує доступ до даних, які можуть зберігатися у базі даних.

Бізнес-логіка: Всі операції, які модифікують стан системи або виконують основні функції ПЗ, зосереджені в моделі. Наприклад, обчислення, валідація, бізнес-правила тощо.

Model містить наступні компоненти:

Service (Сервіси) та Repository (Репозитарії).

Сервісний рівень - це додатковий рівень, який виступає посередником між контролером і рівнем репозиторію. Сервісний рівень містить бізнес-логіку. Зокрема, він містить логіку валідації.

Репозиторій відповідає за відокремлення логіки отримання даних з бази даних і відображення їх на сутнісну модель від бізнес-логіки, яка діє на цій моделі. Тобто репозиторії призначені для створення рівня абстракції між рівнем доступу до даних і рівнем бізнес-логіки програми.

UserService: компонент сервісу, що призначений для виконання операцій пов'язаних з гористувачами.

CompanyService: компонент сервісу, що призначений для виконання операцій пов'язаних з інформацією про Компанії.

BillService: компонент сервісу, що призначений для виконання операцій пов'язаних з інформацією про Рахунки.

BankStatementService: компонент сервісу, що призначений для виконання операцій пов'язаних з інформацією записів Банківських виписок.

OperationService: компонент сервісу, що призначений для виконання операцій пов'язаних з інформацією про Операції Компанії.

CounterpartyService: компонент сервісу, що призначений для виконання операцій пов'язаних з інформацією про Контрагентів.

AnalyticsService: компонент сервісу, що призначений для виконання операцій пов'язаних з формуванням аналітичних даних.

Кожен з сервісів використовує відповідний до своєї області роботи репозитарій. Наприклад, **WorkerService** використовує **WorkerRepository** і так далі.

UserRepository: компонент репозиторію, що призначений для управління збереженням та отриманням даних користувачів з бази даних.

CompanyRepository: компонент репозиторію, що призначений для управління збереженням та отриманням даних про компанії з бази даних.

BillRepository: компонент репозиторію, що призначений для управління збереженням та отриманням даних про рахунки з бази даних.

BankStatementRepository: компонент репозиторію, що призначений для управління збереженням та отриманням даних про записи Банківських виписок з бази даних.

OperationRepository: компонент репозиторію, що призначений для управління збереженням та отриманням даних про операцій з бази даних.

CounterpartyRepository: компонент репозиторію, що призначений для управління збереженням та отриманням даних про контрагентів з бази даних.

View (Вид)

View відповідає за відображення інформації користувачеві та прийом його вводу:

Візуалізація даних: Компонент View відображає поточний стан моделі у вигляді, який розуміє користувач, наприклад, у формі графічного інтерфейсу, тексту чи інших елементів.

Обробка користувацького вводу: View відповідає за прийом вводу від користувача (наприклад, натискання кнопок чи введення тексту), обробку і подальшу реакцію на ці, відправляючи запити до Controller.

View містить компоненти у вигляді веб-сторінок, з якими працює користувач.

Controller (Контроллер)

Контролер обробляє користувацькі дії та координацію між Model та View:

Обробка запитів: Коли користувач взаємодіє з View (наприклад, вводить дані або натискає кнопку), та відправляє запити до Controller, то Controller обробляє подію/запит та обробляє її.

Взаємодія з моделлю: Контролер відповідає за звернення до моделі для виконання операцій, зміну стану даних або отримання необхідної інформації. Він є "посередником", який спрямовує запити та відповіді між View та Model.

Оновлення View: Після отримання результату від Моделі, контролер може ініціювати зміну відображення у View, наприклад, оновити відображувані дані чи відкрити нове представлення.

Controller містить наступні компоненти:

UserController: компонент-контролер, що призначений для обробки запитів щодо інформації про користувачів.

CompanyController: компонент-контролер, що призначений для обробки запитів щодо інформації про компанії.

BillController: компонент-контролер, що призначений для обробки запитів щодо інформації про рахунки.

BankStatementController: компонент-контролер, що призначений для обробки запитів щодо інформації про записи Банківських виписок.

OperationController: компонент-контролер, що призначений для обробки запитів щодо інформації про операції.

CounterpartyController: компонент-контролер, що призначений для обробки запитів щодо інформації про контрагентів.

AnaliticsController: компонент-контролер, що призначений для обробки запитів щодо інформації про формування аналітичних даних

Розробка діаграми розгортання до вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Діаграма розгортання – це діаграма UML, яка показує як розміщені апаратні вузли системи та їх взаємозв'язки.

Діаграма розгортання до вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» представлена на рисунку 2.4

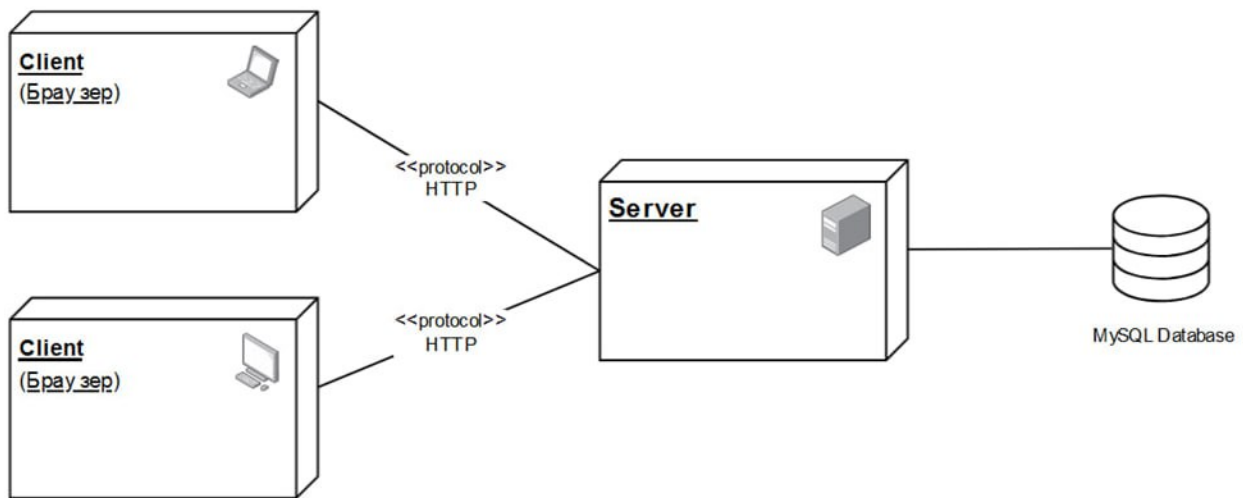


Рисунок 2.4 – Діаграма розгортання до вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

2.3 Проєкт вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Під час розробки вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» були виконані роботи з ескізного, технічного та робочого проектування.

2.3.1 Ескізний проєкт вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Додатково було розроблено діаграми діяльності для прецедентів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ», до них же було створено проєкт користувацького інтерфейсу, а також створено інформаційну модель в реалізації концептуальної моделі даних.

Розробка діаграм діяльності для прецедентів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Діаграма діяльності – візуалізований граф діяльностей, що використовується для опису потоку дій в системі.

Діаграми діяльності для варіантів використання «Додавання нових рахунків», «Перегляд певної компанії», «Перегляд всіх компаній» представлено на рисунках 2.5 – 2.7.

Діаграма діяльності для варіантів використання «Додавання нових рахунків» представлена на рисунку 2.5.

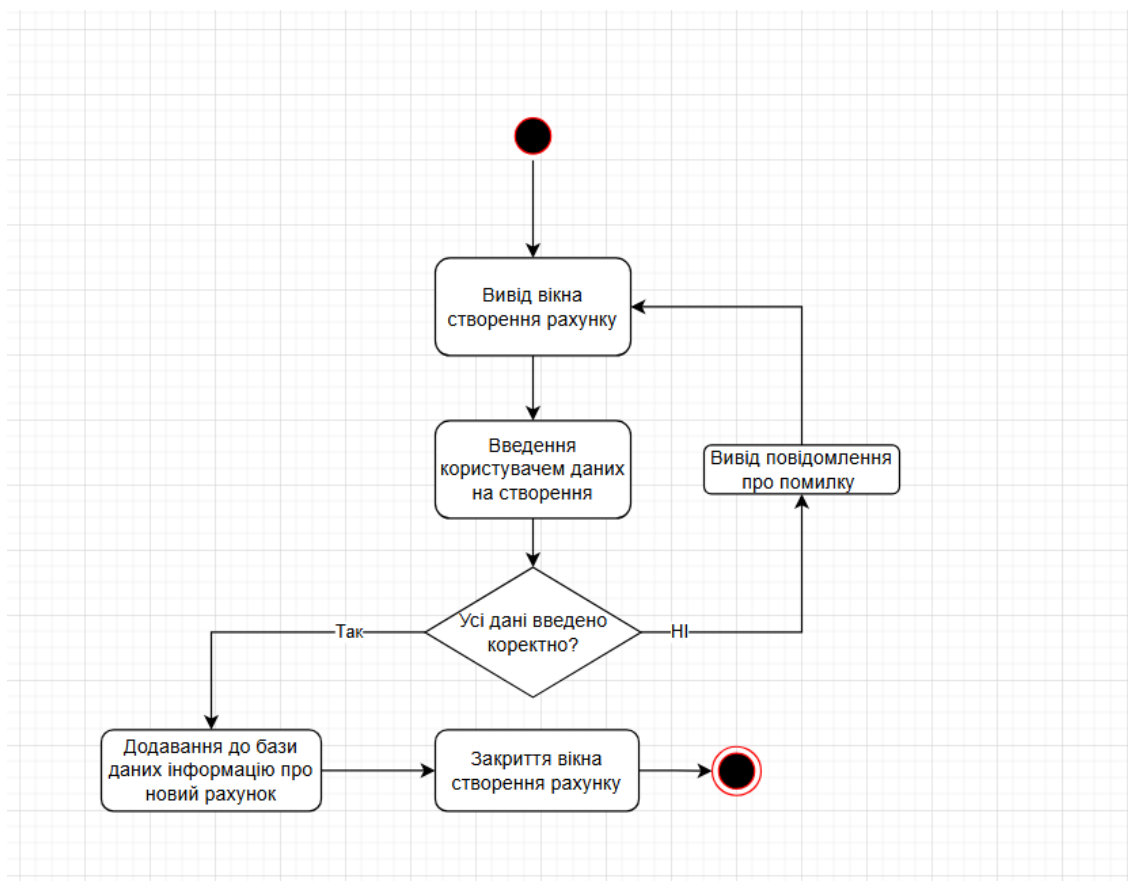


Рисунок 2.5 – Діаграма діяльності для варіантів використання «Додавання нових рахунків»

Діаграма діяльності для варіантів використання «Перегляд певної компанії» представлена на рисунку 2.6.



Рисунок 2.6 – Діаграма діяльності для варіантів використання «Перегляд певної компанії»

Діаграма діяльності для варіантів використання «Перегляд всіх компаній» представлена на рисунку 2.7.



Рисунок 2.7 – Діаграма діяльності для варіантів використання «Перегляд всіх компаній»

Розробка концептуальної моделі даних вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Список сутностей вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»: Компанія, Користувач, Рахунок, Банківська виписка, Операція, Контрагент.

Тепер визначимо атрибути для кожної сутності.

Атрибути сутності Компанія: Код компанії, Назва компанії, ЕДРПОУ

Атрибути сутності Користувач: Код користувача, Ім'я, Email, Номер телефону, Пароль, Доступ.

Атрибути сутності Рахунок: Код рахунку, Назва рахунку, IBAN, Банк.

Атрибути сутності Банківська виписка: Код банківської виписки, Дата операції, Сума, Валюта, Контрагент, Вид, Категорія, Примітка, Деталі.

Атрибути сутності Операція: Код операції, Дата операції, Сума, Валюта, Контрагент, Вид, Категорія, Примітка, Деталі.

Атрибути сутності Контрагент: Код контрагента, Ім'я, Email, Телефон.

Побудуємо концептуальну модель даних вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ».

Концептуальна модель даних вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» представлена на рисунку 2.8.

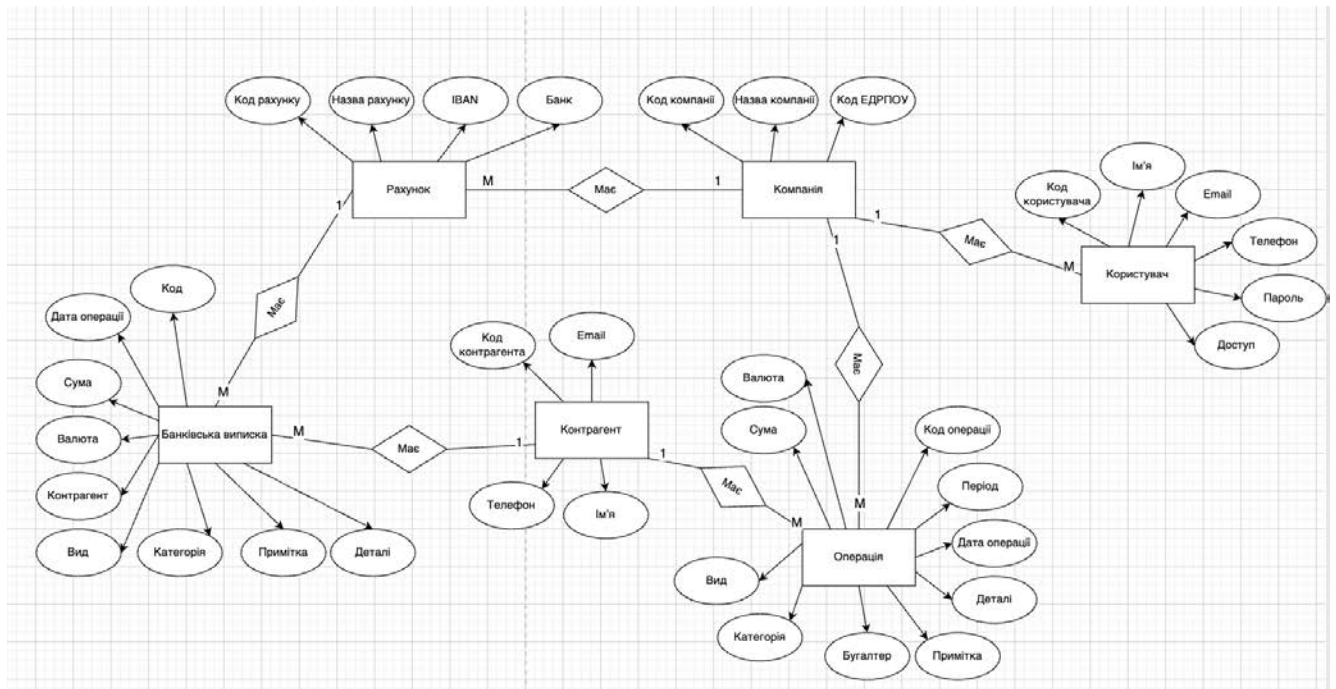


Рисунок 2.8 – Концептуальна модель даних вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Таблиця сутностей та зв'язків представлена у таблиці 2.28.

Таблиця 2.28 – Таблиця сутностей та зв'язків

Сутність 1	Сутність 2	Зв'язок
Користувач	Компанія	Багато-один
Компанія	Рахунок	Один-багато
Компанія	Операція	Один-багато
Рахунок	Банківська виписка	Один-багато
Банківська виписка	Контрагент	Багато-один
Операція	Контрагент	Багато-один

Розробка користувацького інтерфейсу вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

На цьому етапі проектування було розроблено інтерфейс користувача вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ». Основні ескізи екранних форм та сторінок приведено на рисунках 2.9 – 2.14.

Реєстрація

Ім'я
Максим

Email
test@gmail.com

Телефон
+380505004115

Пароль
.....

[Зареєструватися](#)

[Вже маєте акаунт? Увійти](#)

Рисунок 2.9 – Сторінка реєстрації

Вхід

Логін
test@gmail.com

Пароль
.....

[Увійти](#)

[Зареєструватися](#)

Рисунок 2.10 – Сторінка авторизації

Виберіть компанію

Перша компанія зв'язку
[Вибрати](#)

РекламаЮА
[Вибрати](#)

[Створити компанію](#)

[Вийти](#)

Рисунок 2.11 – Сторінка вибору та створення компаній

Панель керування - Перша компанія зв'язку

[Рахунки](#) [Аналітика](#) [Операції](#) [Працівники](#) [Logout](#)

Рисунок 2.12 – Сторінка вибору операцій над компанією

Рахунки

[Створити новий рахунок](#)

Назва	IBAN	Банк	Дії
Перший	UA947749534645634534523543525	PrivatBank	Перейти Переглянути
Другий	UA947749534443534534345523543525	MonoBank	Перейти Переглянути

Рисунок 2.13 – Сторінка роботи з рахунками компанії

2.3.2 Технічний проєкт вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Базуючись на створеному ескізному проєкті, виконаємо розробку технічного проєкту вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ». У ньому буде виконано та побудовано статичну модель у вигляді діаграми класів та динамічну модель у вигляді діаграми послідовності.

Побудова статичної моделі вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Статичну модель вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» представлено у вигляді класів. Діаграма класів – це діаграма, яка відображає статичну структуру ПЗ чи системи, показуючи класи їх атрибути, методи та зв'язки. Діаграма класів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ», була згенерована через вбудовані інструменти засобу розробки «IntelliJ IDEA» та імпортовано через PlantUML. Діаграму класів наведено на рисунку 2.14.

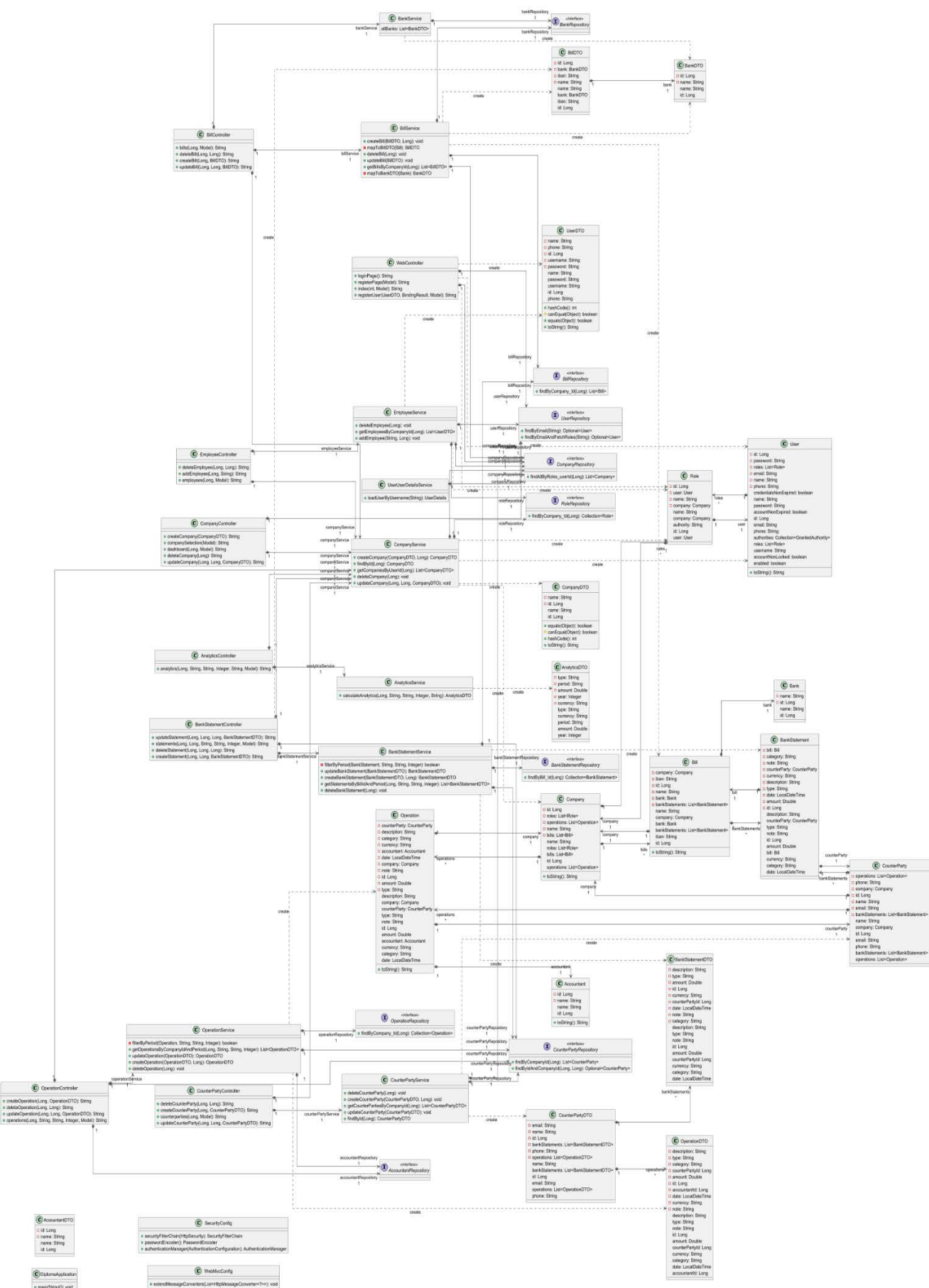


Рисунок 2.14 – Діаграма класів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Специфікація класів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Нижче приведено специфікації основних класів з діаграми класів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» у таблицях 2.29 – 2.39.

Таблиця 2.29 – Специфікація класу Company

Характеристика	Опис
Відповідальність	Відповідає за об'єкт сутності Company
Атрибути	id: Long name: String edrpou: String operations: List<Operation> bills: List<Bill>
Методи	toString(): String

Таблиця 2.30 – Специфікація класу Bill

Характеристика	Опис
Відповідальність	Відповідає за об'єкт сутності Bill
Атрибути	company: Company - iban: String - id: Long - name: String - bank: Bank - bankStatements: List<BankStatement>
Методи	toString(): String

Таблиця 2.31 – Специфікація класу Operation

Характеристика	Опис
Відповідальність	Відповідає за об'єкт сутності Operation
Атрибути	- counterParty: CounterParty - description: String - category: String - currency: String - accountant: Accountant - date: LocalDateTime - company: Company - note: String - id: Long - amount: Double - type: String
Методи	toString(): String

Таблиця 2.32– Специфікація класу BankStatement

Характеристика	Опис
Відповідальність	Відповідає за об'єкт сутності BankStatement
Атрибути	- bill: Bill - category: String - note: String - counterParty: CounterParty - currency: String - description: String - type: String - date: LocalDateTime - amount: Double - id: Long
Методи	-

Таблиця 2.33 – Специфікація класу CounterParty

Характеристика	Опис
Відповідальність	Відповідає за об'єкт сутності CounterParty
Атрибути	- operations: List<Operation> - phone: String - company: Company - id: Long - name: String - email: String - bankStatements: List<BankStatement>
Методи	-

Таблиця 2.34 – Специфікація класу CompanyController

Характеристика	Опис
Відповідальність	Відповідає за об'єкт контролер CompanyController
Атрибути	-
Методи	+ createCompany(CompanyDTO): String + companySelection(Model): String + dashboard(Long, Model): String + deleteCompany(Long): String + updateCompany(Long, Long, CompanyDTO): String

Таблиця 2.35 – Специфікація класу BillController

Характеристика	Опис
Відповідальність	Відповідає за об'єкт контролер BillController
Атрибути	-
Методи	+ bills(Long, Model): String + deleteBill(Long, Long): String + createBill(Long, BillDTO): String + updateBill(Long, Long, BillDTO): String

Таблиця 2.36 – Специфікація класу OperationController

Характеристика	Опис
Відповідальність	Відповідає за об'єкт контролер OperationController
Атрибути	-
Методи	+ createOperation(Long, OperationDTO): String + deleteOperation(Long, Long): String + updateOperation(Long, Long, OperationDTO): String + operations(Long, String, String, Integer, Model): String

Таблиця 2.37 – Специфікація класу BankStatementController

Характеристика	Опис
Відповідальність	Відповідає за об'єкт контролер BankStatementController
Атрибути	-
Методи	+ updateStatement(Long, Long, Long, BankStatementDTO): String + statements(Long, Long, String, String, Integer, Model): String + deleteStatement(Long, Long, Long): String + createStatement(Long, Long, BankStatementDTO): String

Таблиця 2.38 – Специфікація класу CounterPartyController

Характеристика	Опис
Відповідальність	Відповідає за об'єкт контролер CounterPartyController
Атрибути	-
Методи	+ deleteCounterParty(Long, Long): String + createCounterParty(Long, CounterPartyDTO): String + counterparties(Long, Model): String + updateCounterParty(Long, Long, CounterPartyDTO): String

Таблиця 2.39 – Специфікація класу WebController

Характеристика	Опис
Відповідальність	Відповідає за об'єкт контролер WebController
Атрибути	-
Методи	+ loginPage(): String + registerPage(Model): String + index(int, Model): String + registerUser(UserDTO, BindingResult, Model): String

Розробка динамічної моделі вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Динамічна модель може вебзастосунку представлена у вигляді діаграми послідовності. Діаграма послідовності – тип діаграми UML, для моделювання динамічної поведінки системи. Вона показує, як компоненти системи взаємодіють та у якій послідовності. Діаграми послідовності прецедентів «Додавання нових рахунків», «Перегляд певної компанії», «Перегляд всіх компаній» представлено на рисунках 2.15 – 2.17.

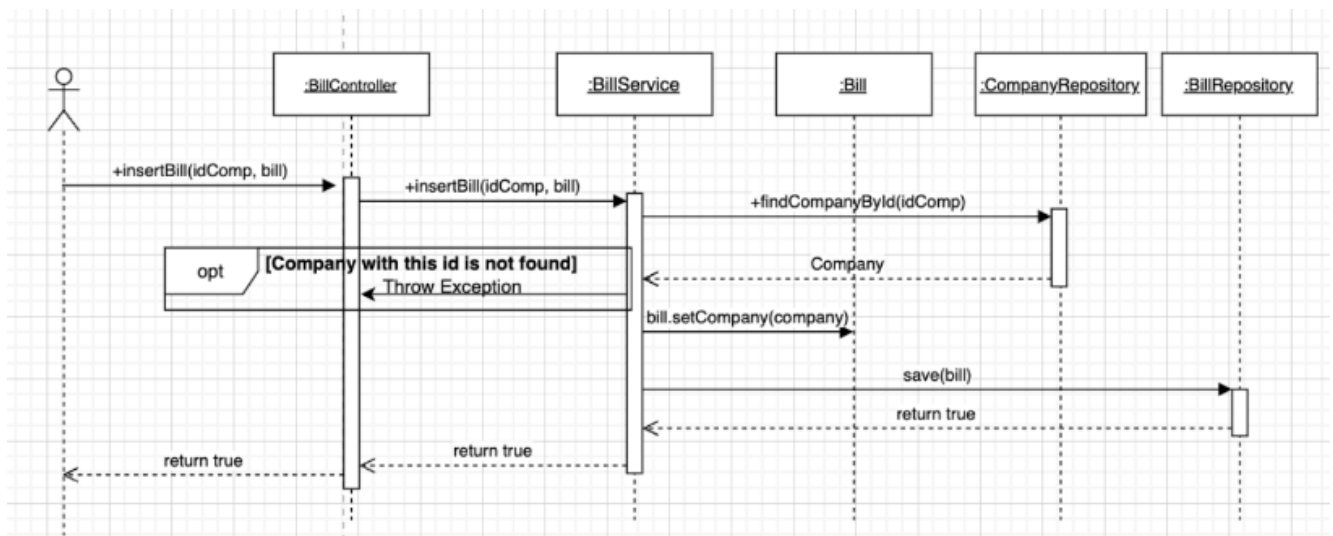


Рисунок 2.15 – Діаграма послідовності прецеденту «Додавання нових рахунків»

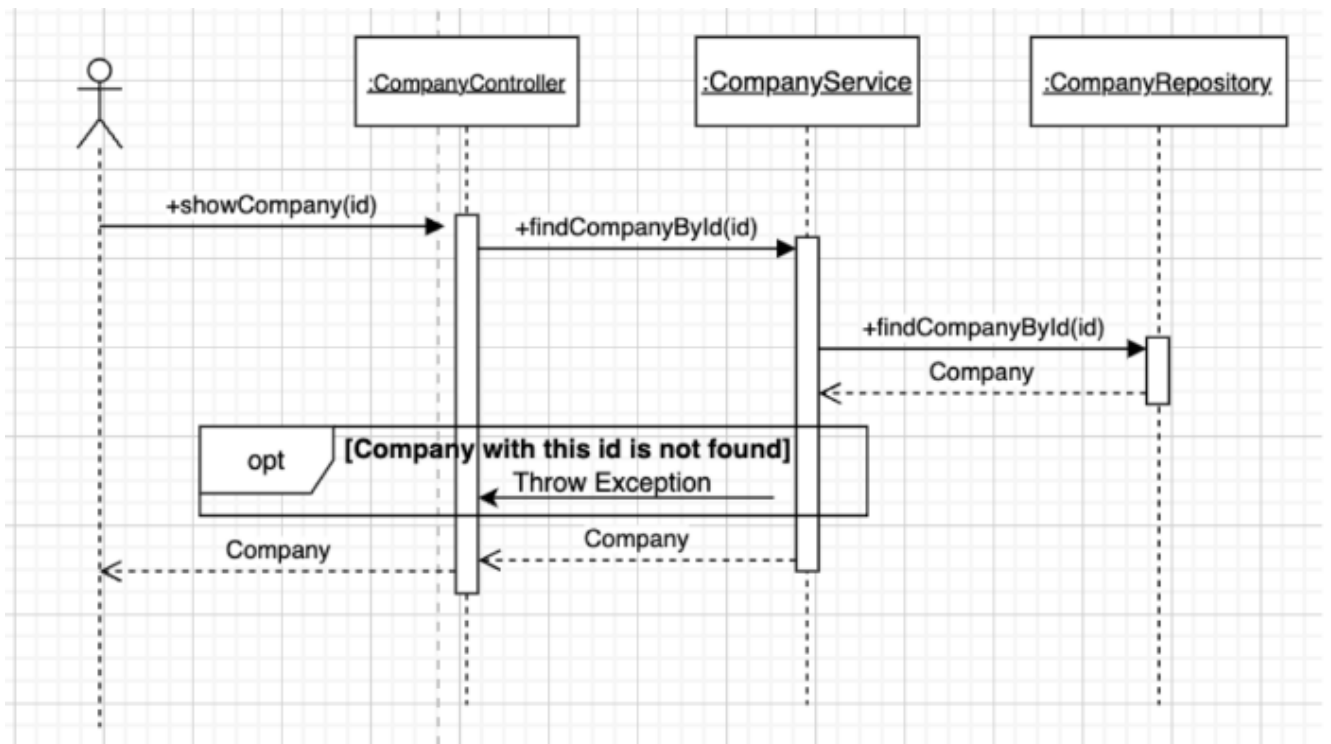


Рисунок 2.16 – Діаграма послідовності прецеденту «Перегляд певної компанії»

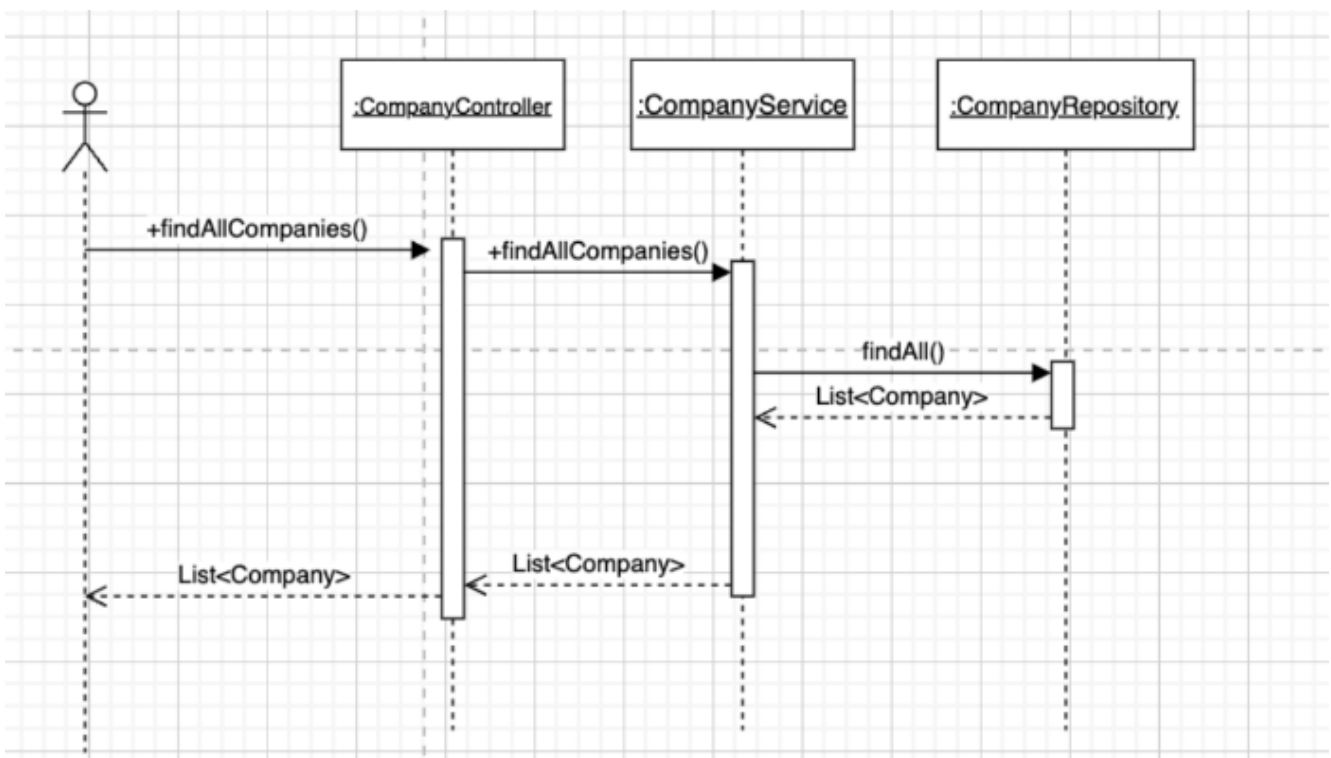


Рисунок 2.17 – Діаграма послідовності прецеденту «Перегляд всіх компаній»

Розробка логічної моделі даних вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

При побудові логічної моделі даних було додано ще три додаткові таблиці «Role», «Accountant» та «Bank». Логічна модель даних вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» представлена на рисунку 2.18.

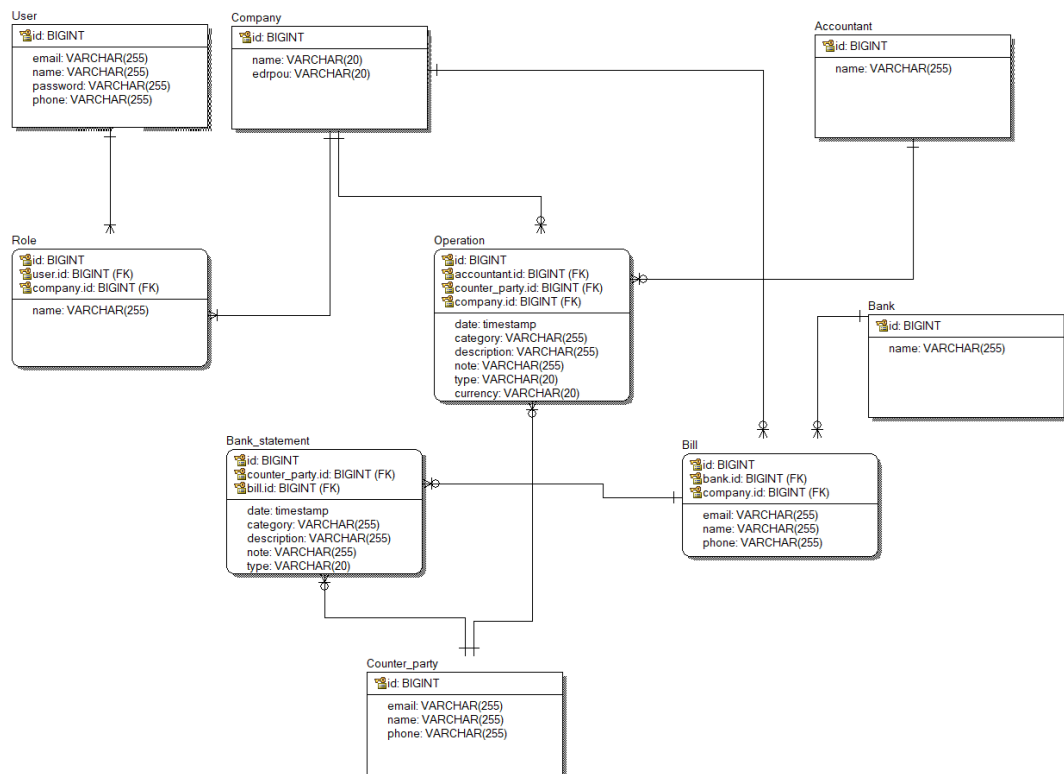


Рисунок 2.18 – Логічна модель даних вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Після розробки технічного проекту вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» можна переходити до створення робочого проекту.

2.3.3 Робочий проєкт вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Обґрунтування вибору засобів розробки вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Для розробки вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» були обрані такі засоби розробки:

- Мова програмування Java 17. Java – це мова програмування загального призначення, створена компанією Sun Microsystems 1995 року. Її використовують для розробки програмного забезпечення, вебсервісів, ігор і застосунків.

Java є однією з найпопулярніших мов програмування у світі завдяки нескладному синтаксису, гнучкості, безпеці, портативності та масштабованості [6].

- Мови HTML та CSS. HTML (HyperText Markup Language) – це мова, яка використовується для створення структури веб-сторінки. Вона складається з ряду “тегів”, які визначають, як буде виглядати вміст на сторінці. Знання HTML допомагає тестувальникам ПЗ розуміти, як веб-сторінка повинна виглядати та функціонувати [7]. CSS (Cascading Style Sheets) – це мова, яка використовується для опису вигляду веб-сторінки, створеної за допомогою HTML. Знання CSS дозволяє тестувальникам ПЗ краще розуміти, як веб-сторінка повинна виглядати, і допомагає виявляти проблеми з дизайном та візуальним представленням [7].

- Середовище розробки JetBrains IntelliJ IDEA. IntelliJ IDEA – комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains. Система поставляється у вигляді урізаної по функціональності безкоштовної версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition», для якої активні розробники відкритих проєктів мають можливість отримати безкоштовну

ліцензію. Програмний код Community-версії поширюються в рамках ліцензії Apache 2.0. Бінарні збірки підготовлені для Linux, Mac OS X і Windows [8].

- СУБД PostgreSQL. PostgreSQL – це потужна, відкрита об’єктно-реляційна система управління базами даних (СУБД), призначена для зберігання, обробки та керування великими обсягами даних. Розроблений з акцентом на розширюваність та сумісність зі стандартом SQL, PostgreSQL підтримує як реляційні, так і деякі NoSQL-функції, такі як зберігання JSON-даних та ключ-значення. Її основне призначення - надавати безпечне та надійне рішення для зберігання даних, яке дозволяє користувачам ефективно керувати складними транзакціями та виконувати аналіз даних у режимі реального часу [9].

- Фреймворк Bootstrap. Bootstrap – це фронтенд-фреймворк, який допомагає розробляти веб-інтерфейси з готовими компонентами CSS JavaScript для поширених елементів, таких як навігація, форми та кнопки. Він також має адаптивну систему макета сітки, яка налаштовує ваш сайт під пристрої з різними розмірами екрану, такі як телефони та планшети. Використовуючи систему макета Bootstrap, ви можете створити єдиний сайт, який добре виглядатиме на будь-якому пристрої, який можуть використовувати ваші клієнти [10].

- Фреймворк Spring. Java-spring – це фреймворк з модульною архітектурою, який надає великий набір інструментів для створення масштабованих Java-додатків. Він дає змогу керувати залежностями, реалізовувати інверсію управління (IoC), а також пропонує потужну систему для роботи з аспектно-орієнтованим програмуванням (AOP). Основним його завданням є спрощення процесу розробки за рахунок надання готових рішень для роботи з даними, безпеки та інтеграції з іншими бібліотеками. Spring Boot – це надбудова над Spring, яка спрощує конфігурацію та розгортання додатків. Основним його завданням є автоматизація процесу конфігурації, що дає змогу мінімізувати ручні налаштування та прискорити процес створення застосунку. Тобто він надає готові шаблони для створення мікросервісів і веб-додатків, що робить його чудовим вибором для старту нових проєктів [11].

- Фреймворк Hibernate. Hibernate – засіб відображення між об'єктами та реляційними структурами (object-relational mapping, ORM) для платформи Java. Hibernate є вільним програмним забезпеченням, яке поширюється на умовах GNU Lesser General Public License. Hibernate надає легкий для використання каркас (фреймворк) для відображення між об'єктно-орієнтованою моделлю даних і традиційною реляційною базою даних [12].

- Контейнеризатор Docker. Docker - це програмне забезпечення для автоматизації розгортання і управління додатками в середовищах з підтримкою контейнеризації, контейнеризатор додатків [13].

- Збирач проектів Maven. Maven – інструмент для управління та збирання проектів – справжній помічник Java-програміста. Він полегшує життя девелоперу всіх стадіях роботи: від створення структури проекту та підключення необхідних бібліотек до розгортання продукту на сервері. При роботі з будь-яким фреймворком доведеться використовувати Maven. Тому давай сьогодні розберемо його основні функції та подивимося, як їх потрібно використовувати [14].

Реалізація основних класів вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Нижче представлено програмний код сутності «Операція» та його сервісу реалізовано мовою програмування Java. Повний код застосунку буде представлено у додатку Д.

Operation.java

```
package org.dzhurynskyi.diploma.model;

import jakarta.persistence.*;
import lombok.*;

import java.time.LocalDateTime;

@Entity
@AllArgsConstructor
@NoArgsConstructor
@Getter
@Setter
@ToString
```

```

public class Operation {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private LocalDateTime date;
    private Double amount;
    private String currency;
    private String description;
    private String type;
    private String category;
    private String note;

    @ManyToOne
    private Accountant accountant;

    @ManyToOne
    private CounterParty counterParty;

    @ManyToOne
    private Company company;
}

```

OperationService.java

```

package org.dzhurynskiy.diploma.service;

import org.dzhurynskiy.diploma.dto.OperationDTO;
import org.dzhurynskiy.diploma.model.Operation;
import org.dzhurynskiy.diploma.repository
    .AccountantRepository;
import org.dzhurynskiy.diploma.repository.CounterPartyRepository;
import org.dzhurynskiy.diploma.repository.OperationRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.stream.Collectors;

@Service
public class OperationService {

    @Autowired
    private OperationRepository operationRepository;

    @Autowired
    private AccountantRepository accountantRepository;

    @Autowired
    private CounterPartyRepository counterPartyRepository;

    public OperationDTO createOperation(OperationDTO operationDTO, Long companyId)
    {
        Operation operation = new Operation();
        operation.setDate(operationDTO.getDate());
        operation.setAmount(operationDTO.getAmount());
        operation.setCurrency(operationDTO.getCurrency());
        operation.setDescription(operationDTO.getDescription());
        operation.setType(operationDTO.getType());
        operation.setCategory(operationDTO.getCategory());
        operation.setNote(operationDTO.getNote());
    }
}

```

```

operation.setAccountant(accountantRepository.findById(operationDTO.getAccountantId())
    .orElse(null));

operation.setCounterParty(counterPartyRepository.findById(operationDTO.getCounterPartyId())
    .orElse(null));
    operation = operationRepository.save(operation);
    operationDTO.setId(operation.getId());
    return operationDTO;
}

    public List<OperationDTO> getOperationsByCompanyIdAndPeriod(Long companyId,
String periodType, String period, Integer year) {
        return operationRepository.findByCompany_Id(companyId)
            .stream()
            .filter(operation -> filterByPeriod(operation, periodType, period,
year))
            .map(operation -> {
                OperationDTO dto = new OperationDTO();
                dto.setId(operation.getId());
                dto.setDate(operation.getDate());
                dto.setAmount(operation.getAmount());
                dto.setCurrency(operation.getCurrency());
                dto.setDescription(operation.getDescription());
                dto.setType(operation.getType());
                dto.setCategory(operation.getCategory());
                dto.setNote(operation.getNote());
                dto.setAccountantId(operation.getAccountant().getId());
                dto.setCounterPartyId(operation.getCounterParty().getId());
                return dto;
            })
            .collect(Collectors.toList());
    }

    public OperationDTO updateOperation(OperationDTO operationDTO) {
        Operation operation =
operationRepository.findById(operationDTO.getId()).orElse(null);
        if (operation == null) return null;
        operation.setDate(operationDTO.getDate());
        operation.setAmount(operationDTO.getAmount());
        operation.setCurrency(operationDTO.getCurrency());
        operation.setDescription(operationDTO.getDescription());
        operation.setType(operationDTO.getType());
        operation.setCategory(operationDTO.getCategory());
        operation.setNote(operationDTO.getNote());

operation.setAccountant(accountantRepository.findById(operationDTO.getAccountantId())
    .orElse(null));

operation.setCounterParty(counterPartyRepository.findById(operationDTO.getCounterPartyId())
    .orElse(null));
        operationRepository.save(operation);
        return operationDTO;
    }

    public void deleteOperation(Long id) {
        operationRepository.deleteById(id);
    }
}

```

Тестування та випробування вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Тестування проводиться в типових умовах використання, включаючи запуск застосунку, введення даних у форми, обробку подій і перевірку відображення повідомлень. Результати тестів фіксуються у вигляді тест-кейсів з урахуванням тестового оточення, залежностей та дій користувача.

Для оцінювання заявлених характеристик ПЗ застосовуються методи «чорної скриньки»:

Ідентифікатор: TC-REG-001

Предмети тестування: Форма реєстрації нового користувача

Специфікація вхідних даних:

- ПІБ: Іваненко І.І.
- Email: ivan.test@example.com
- Пароль: StrongPass1
- Телефон: 0671234567

Специфікація результатів:

Повідомлення про успішну реєстрацію, створення облікового запису

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: Відсутні.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-REG-001 представлено у таблиці 2.40.

Таблиця 2.40 – Класи еквівалентності TC-REG-001

Клас вхідних даних	Очікуваний результат
1	2
Email валідний	Проходить
Email не було зареєстровано	Проходить

Продовження таблиці 2.40

1	2
Пароль містить ≥ 8 символів, велику літеру і цифру	Проходить
Пароль без великої літери	Проходить
Пароль без цифри	Проходить
Телефон містить лише цифри	Проходить

Ідентифікатор: TC-REG-02 (невалідний пароль)

Предмети тестування: Форма реєстрації нового користувача

Специфікація вхідних даних:

- ПІБ: Іваненко І.І.
- Email: ivan.test@example.com
- Пароль: weew
- Телефон: 0671234567

Специфікація результатів:

Повідомлення про помилку валідації пароля. Обліковий запис не було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: відсутні.

Залежність з іншими тестами: TC-REG-01

Методи тестування:

Чорна скринька: Перевірка збереження даних без знання внутрішньої логіки.

Класи еквівалентності TC-REG-002 представлено у таблиці 2.41.

Таблиця 2.41 – Класи еквівалентності TC-REG-002

Клас вхідних даних	Очікуваний результат
Email валидний	Проходить
Email не було зареєстровано	Проходить
Пароль містить ≥ 8 символів, велику літеру і цифру	Не проходить
Пароль без великої літери	Не проходить
Пароль без цифри	Не проходить
Телефон містить лише цифри	Проходить

Ідентифікатор: TC-REG-03 (Email вже було зареєстровано)

Предмети тестування: Форма реєстрації нового користувача

Специфікація вхідних даних:

- ПІБ: Іваненко І.І.
- Email: ivan.test@example.com
- Пароль: StrongPass1
- Телефон: 0671234567

Специфікація результатів:

Повідомлення «Обліковий запис з таким Email вже було зареєстровано».

Обліковий запис не було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: відсутні.

Залежність з іншими тестами: TC-REG-02

Методи тестування:

Чорна скринька: Перевірка збереження даних без знання внутрішньої логіки.

Класи еквівалентності TC-REG-003 представлено у таблиці 2.42.

Таблиця 2.42 – Класи еквівалентності TC-REG-003

Клас вхідних даних	Очікуваний результат
Email валідний	Проходить
Email не було зареєстровано	Не проходить
Пароль містить ≥ 8 символів, велику літеру і цифру	Проходить
Пароль без великої літери	Проходить
Пароль без цифри	Проходить
Телефон містить лише цифри	Проходить

Ідентифікатор: TC-REG-04 (Номер телефону не правильного формату)

Предмети тестування: Форма реєстрації нового користувача

Специфікація вхідних даних:

- ПІБ: Іваненко І.І.
- Email: ivan.test1@example.com
- Пароль: StrongPass1
- Телефон: 06к7123456

Специфікація результатів:

Повідомлення «Неправильний номер телефону». Обліковий запис не було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: відсутні.

Залежність з іншими тестами: TC-REG-01

Методи тестування:

Чорна скринька: Перевірка збереження даних без знання внутрішньої логіки.

Класи еквівалентності TC-REG-004 представлено у таблиці 2.43.

Таблиця 2.43 – Класи еквівалентності TC-REG-004

Клас вхідних даних	Очікуваний результат
Email валідний	Проходить
Email не було зареєстровано	Проходить
Пароль містить ≥ 8 символів, велику літеру і цифру	Проходить
Пароль без великої літери	Проходить
Пароль без цифри	Проходить
Телефон містить лише цифри	Не проходить

Ідентифікатор: TC-AUTH-001

Предмети тестування: Форма авторизації користувача

Специфікація вхідних даних:

- Email: ivan.test@example.com
- Пароль: StrongPass1

Специфікація результатів:

Повідомлення про авторизацію, користувача було авторизовано

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: Відсутні.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-AUTH-001 представлено у таблиці 2.44.

Таблиця 2.44 – Класи еквівалентності TC-AUTH-001

Клас вхідних даних	Очікуваний результат
Email зареєстровано	Проходить
Правильний пароль	Проходить

Ідентифікатор: TC-AUTH-002 (неправильний пароль)

Предмети тестування: Форма авторизації користувача

Специфікація вхідних даних:

- Email: ivan.test@example.com
- Пароль: wrongPass1

Специфікація результатів:

Повідомлення «Пароль не вірний», користувача не було авторизовано

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: TC-AUTH-001.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-AUTH-002 представлено у таблиці 2.45.

Таблиця 2.45 – Класи еквівалентності TC-AUTH-002

Клас вхідних даних	Очікуваний результат
Email зареєстровано	Проходить
Правильний пароль	Не проходить

Ідентифікатор: TC-AUTH-003 (email не зареєстровано)

Предмети тестування: Форма авторизації користувача

Специфікація вхідних даних:

- Email: ivan.tes11t@example.com
- Пароль: wrongPass1

Специфікація результатів:

Повідомлення «Обліковий запис з таким email не зареєстровано», користувача не було авторизовано

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: TC-AUTH-001.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-AUTH-003 представлено у таблиці 2.46.

Таблиця 2.46 – Класи еквівалентності TC-AUTH-003

Клас вхідних даних	Очікуваний результат
Email зареєстровано	Не проходить
Правильний пароль	Не проходить

Ідентифікатор: TC-COMP-001

Предмети тестування: Форма створення компанії

Специфікація вхідних даних:

- Назва: ТОВ «Сервіс+»
- Код ЄДРПОУ: 12345678

Специфікація результатів:

Повідомлення компанію створено, компанію було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: Відсутні.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-COMP-001 представлено у таблиці 2.47.

Таблиця 2.47 – Класи еквівалентності TC-COMP-001

Клас вхідних даних	Очікуваний результат
Код ЄДРПОУ — 8 цифр	Проходить
Назва унікальна	Проходить

Ідентифікатор: TC-COMP-002 (Неправильний код)

Предмети тестування: Форма створення компанії

Специфікація вхідних даних:

- Назва: ТОВ «Сервіс+»
- Код ЄДРПОУ: 1234567

Специфікація результатів:

Повідомлення «Неправильний код ЄДРПОУ», компанію не було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: TC-COMP-001.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-COMP-002 представлено у таблиці 2.48.

Таблиця 2.48 – Класи еквівалентності TC-COMP-002

Клас вхідних даних	Очікуваний результат
Код ЄДРПОУ — 8 цифр	Не проходить
Код ЄДРПОУ унікальний	Проходить
Назва унікальна	Проходить

Ідентифікатор: TC-COMP-003 (Назва не унікальна)

Предмети тестування: Форма створення компанії

Специфікація вхідних даних:

- Назва: ТОВ «Сервіс+»
- Код ЄДРПОУ: 12345678

Специфікація результатів:

Повідомлення «Компанію з такою назвою вже створено», компанію не було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: TC-COMP-001.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-COMP-003 представлено у таблиці 2.49.

Таблиця 2.49 – Класи еквівалентності TC-COMP-003

Клас вхідних даних	Очікуваний результат
Код ЄДРПОУ — 8 цифр	Проходить
Код ЄДРПОУ унікальний	Проходить
Назва унікальна	Не проходить

Ідентифікатор: TC-COMP-004 (Компанію з таким кодом вже було створено)

Предмети тестування: Форма створення компанії

Специфікація вхідних даних:

- Назва: ТОВ «Сервіс+»
- Код ЄДРПОУ: 12345678

Специфікація результатів:

Повідомлення «Компанію з таким кодом вже було створено», компанію не було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: TC-COMP-001.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-COMP-004 представлено у таблиці 2.50.

Таблиця 2.50 – Класи еквівалентності TC-COMP-004

Клас вхідних даних	Очікуваний результат
Код ЄДРПОУ — 8 цифр	Проходить
Код ЄДРПОУ унікальний	Не проходить
Назва унікальна	Проходить

Ідентифікатор: TC-ACC-001

Предмети тестування: Форма створення компанії

Специфікація вхідних даних:

- Назва: Рахунок1
- IBAN: UA123456789012345678901234567
- Банк: ПриватБанк

Специфікація результатів:

Повідомлення «Рахунок було створено», Рахунок було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: Відсутні.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-ACC-001 представлено у таблиці 2.51.

Таблиця 2.51 – Класи еквівалентності TC-ACC-001

Клас вхідних даних	Очікуваний результат
IBAN = 29 символів (латиниця + цифри)	Проходить
IBAN унікальний	Проходить
Назва унікальна	Проходить

Ідентифікатор: TC-ACC-002 (Вказаний IBAN не коректний)

Предмети тестування: Форма створення компанії

Специфікація вхідних даних:

- Назва: Рахунок13
- IBAN: UA12345678901234567890123456
- Банк: ПриватБанк

Специфікація результатів:

Повідомлення «Вказаний IBAN не коректний», Рахунок не було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: TC-ACC-001.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-ACC-002 представлено у таблиці 2.52.

Таблиця 2.52 – Класи еквівалентності TC-ACC-002

Клас вхідних даних	Очікуваний результат
IBAN = 29 символів (латиниця + цифри)	Не проходить
IBAN унікальний	Проходить
Назва унікальна	Проходить

Ідентифікатор: ТС-АСС-003 (Вказаний IBAN вже є у системі)

Предмети тестування: Форма створення компанії

Специфікація вхідних даних:

- Назва: Рахунок1
- IBAN: UA123456789012345678901234567
- Банк: ПриватБанк

Специфікація результатів:

Повідомлення «Рахунок з вказаним IBAN вже є», Рахунок небуло створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: ТС-АСС-001.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності ТС-АСС-003 представлено у таблиці 2.53.

Таблиця 2.53 – Класи еквівалентності ТС-АСС-003

Клас вхідних даних	Очікуваний результат
IBAN = 29 символів (латиниця + цифри)	Проходить
IBAN унікальний	Не проходить
Назва унікальна	Проходить

Ідентифікатор: ТС-АСС-004 (Вказаний рахунок вже є у системі)

Предмети тестування: Форма створення компанії

Специфікація вхідних даних:

- Назва: Рахунок1
- IBAN: UA123456789012345678901234567
- Банк: ПриватБанк

Специфікація результатів:

Повідомлення «Рахунок з вказаною назвою вже є», рахунок не було створено

Тестове оточення: Веб браузер (Chrome, версія 120 або вище), локальний сервер Java, підключення до PostgreSQL.

Спеціальні процедурні вимоги: Відсутні.

Залежність з іншими тестами: TC-ACC-001.

Методи тестування:

Чорна скринька: Перевірка функціональності форми без знання її внутрішньої структури.

Класи еквівалентності TC-ACC-004 представлено у таблиці 2.54.

Таблиця 2.54 – Класи еквівалентності TC-ACC-004

Клас вхідних даних	Очікуваний результат
IBAN = 29 символів (латиниця + цифри)	Проходить
IBAN унікальний	Проходить
Назва унікальна	Не проходить

Випробування вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ»

Випробування вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» буде проведено відповідно до методики випробувань описаної у додатку В.

Під час проведення випробувань було опрацьовано такий функціонал: авторизація у системі з коректними даними, додавання нової компанії, додавання нового рахунку, видалення рахунку, реєстрація нового користувача з коректними даними, видалення працівника компанії.

1) Випробування авторизації у системі з коректними даними

При завантаженні вебзастосунку користувача переносить на сторінку авторизації. При вводі коректних даних, користувача переносить на сторінку вибору компаній. Випробування пройдено успішно. Результат випробування наведено на рисунках 2.19, 2.20.

Рисунок 2.19 – Авторизація

Рисунок 2.20 – Успішна авторизація

2) Випробування додавання нової компанії

Створення відбувається після натискання кнопки «Створити компанію», з'являється модальне вікно з формою для додавання нової компанії, після внесення валідних даних компанія створюється і відображається на сторінці вибору компаній. Випробування пройдено успішно. Результат випробування наведено на рисунках 2.21, 2.22.

Рисунок 2.21 – Створення нової компанії

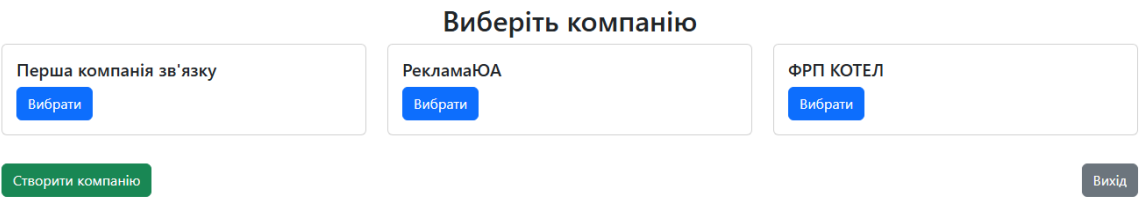


Рисунок 2.22 – Успішне створення нової компанії

3) Випробування додавання нового рахунку

При виборі відповідної компанії користувача переносить на сторінку рахунків, створення відбувається після натискання кнопки «Створити новий рахунок», з'являється модальне вікно з формою для додавання нового рахунку, після внесення валідних даних рахунок створюється і відображається на сторінці вибору рахунків. Випробування пройдено успішно. Результат випробування наведено на рисунках 2.23, 2.24.

4) Випробування видалення рахунку

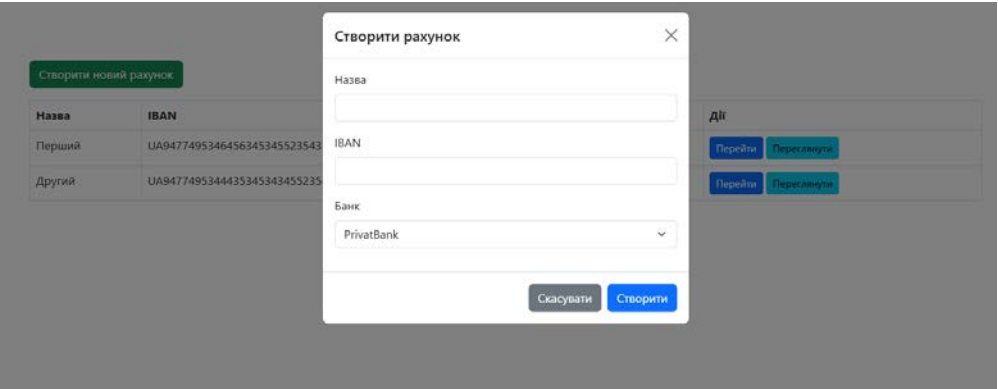


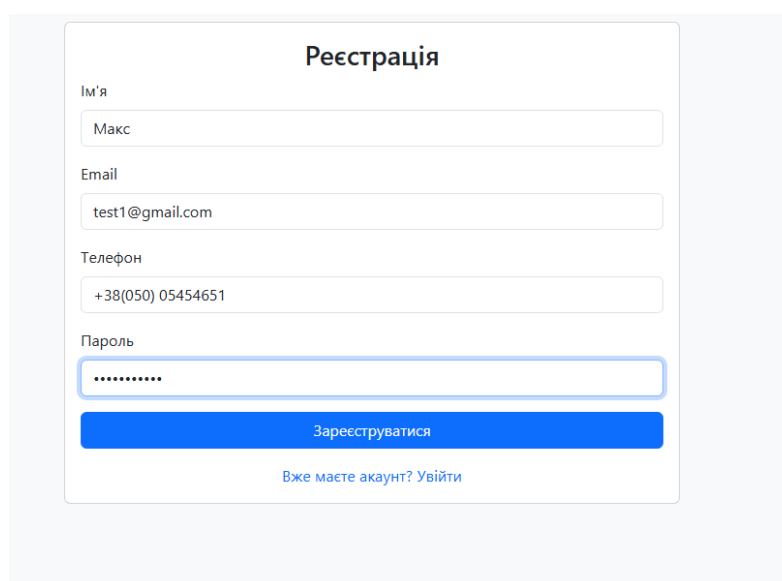
Рисунок 2.23 – Створення нового рахунку

Рахунки			
Створити новий рахунок			
Назва	IBAN	Банк	Дії
Перший	UA9477495346456345345523543	PrivatBank	Перейти Переглянути
Другий	UA9477495344435345345523543525	MonoBank	Перейти Переглянути
Рахунок1	UA123456789012345678901234567	PrivatBank	Перейти Переглянути

Рисунок 2.24 – Успішне створення нового рахунку

5) Випробування реєстрації нового користувача з коректними даними

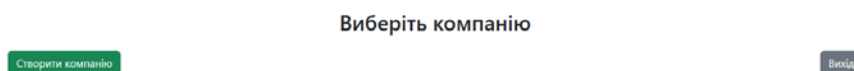
При завантаженні вебзастосунку користувача переносить на сторінку авторизації, якщо у користувача немає облікового запису, то при натисканні на посиланні під формою «Зареєструватись», користувача буде перенесено на сторінку реєстрації. При вводі коректних даних, користувача переносить на сторінку авторизації і при введенні реєстраційних даних, його буде пренесено на сторінку вибору компаній. Випробування пройдено успішно. Результат випробування наведено на рисунках 2.25, 2.26.



The screenshot shows a registration form with the following fields and content:

- Ім'я** (Name): Input field containing 'Макс'.
- Email**: Input field containing 'test1@gmail.com'.
- Телефон** (Phone): Input field containing '+38(050) 05454651'.
- Пароль** (Password): Input field with masked characters '.....'.
- Зареєструватися** (Register): A prominent blue button.
- Вже маєте акаунт? Увійти** (Already have an account? Log in): A link below the registration button.

Рисунок 2.25 – Реєстрація користувача



The screenshot shows a screen with the title 'Виберіть компанію' (Select a company). At the bottom, there are two buttons: 'Створити компанію' (Create company) in green and 'Вийді' (Exit) in grey.

Рисунок 2.26 – Успішна реєстрація користувача

6) Видалення працівника компанії

При виборі користувачем компанії і переході на сторінку «Працівники», користувач може бачити всіх працівників компанії та себе. При натисканні на

кнопку «Видалити», він видаляє працівника. Випробування пройдено успішно. Результат випробування наведено на рисунках 2.27, 2.28.

Працівники			
Додати			
Ім'я	Email	Телефон	Дії
Максим	test@gmail.com	+380505004115	Переглянути Видалити
Макс	test1@gmail.com	+38(050) 05454651	Переглянути Видалити

Рисунок 2.27 – Сторінка «Працівники»

Працівники			
Додати			
Ім'я	Email	Телефон	Дії
Макс	test1@gmail.com	+38(050) 05454651	Переглянути Видалити

Рисунок 2.28 – Сторінка «Працівники» після видалення працівника

3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ ФІНАНСОВОГО АНАЛІЗУ ДЕБІТОРІВ ДЛЯ ЮРИДИЧНОЇ КОМПАНІЇ «ЛЕМБЕРГ»

У результаті виконання кваліфікаційної роботи було розроблено вебзастосунок для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ». З самого початку було проведено аналіз практичної задачі інженерії програмного забезпечення. Далі були опрацьовані існуючі програмні продукти, що мало мету обґрунтувати доцільність створення вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ».

За результатом аналізу було сформовано постановку задачі на розробку вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ», було визначено функціонал та користувачів застосунку. А також було вибрано архітектуру вебзастосунку.

На етапі ескізного проектування було створено діаграму діяльності, концептуальну модель, а також було виконано моделювання користувацького інтерфейсу.

На етапі технічного проектування було розроблено статичну та динамічну моделі у вигляді діаграми класів та діаграм послідовностей відповідно. А також побудовано логічну модель даних вебзастосунку.

Під час робочого проектування було вибрано інструменти та засоби розробки, виконано кодування вебзастосунку. Наприкінці виконано тестування та випробування вебзастосунку, згідно додатку В.

Зовнішній вигляд сторінки операцій приведено на рисунку 3.1.

Операції

Місяць ▼

Січень ▼

2023 ▼

Показати

Додати

Дата	Сума	Валюта	Опис	Тип	Категорія	Бухгалтер	Контрагент	Дії
02.06.2025 18:00	500.0	UAH	Плата за зв'язок	Дебет	Зв'язок	Ірина	Петро	Переглянути Видалити

Рисунок 3.1 – Сторінка "Операції"

Зовнішній вигляд сторінки операцій приведено на рисунку 3.2.

Банківські виписки

Місяць

Червень

2025

Показати

Додати

Дата	Сума	Валюта	Опис	Тип	Категорія	Контрагент	Дії
02-06-2025T18:57	500.0	UAH	Плата за зв'язок	Дебет	зв'язок	Петро	Переглянути Видалити

Рисунок 3.2 – Сторінка "Банківські виписки"

Під час написання даної роботи було розроблено такі додатки: технічне завдання представлено у додатку А, опис програмного забезпечення у додатку Б, програму та методику випробувань у додатку В, інструкцію користувача у додатку Г, текст програми у додатку Д.

4 ОХОРОНА ПРАЦІ

4.1 Вступ

Законодавство України про охорону праці – це система взаємозв'язаних законів та інших нормативно-правових актів, що регулюють відносини у сфері реалізації державної політики щодо соціального захисту її громадян в процесі трудової діяльності [15].

Законодавство про охорону праці складається з (відповідно до ст. 3 Закону України «Про охорону праці»): Закону України «Про охорону праці»; Кодексу законів про працю України; Закону України «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності»; прийнятих відповідно до них нормативно-правових актів.

Складові законодавства України про охорону праці:

Конституція України (ст. 43, 45, 46, 49, 57)

Ст. 43. Кожен має право на працю, що включає можливість заробляти собі на життя працею, яку він вільно обирає або на яку вільно погоджується. ... Використання примусової праці забороняється. ... Кожен має право на належні, безпечні і здорові умови праці, на заробітну плату, не нижчу від визначеної законом. Використання праці жінок і неповнолітніх на небезпечних для їхнього здоров'я роботах забороняється. ...

Ст. 45. Кожен, хто працює, має право на відпочинок. Це право забезпечується наданням днів щотижневого відпочинку, а також оплачуваної щорічної відпустки, встановленням скороченого робочого дня щодо окремих професій і виробництв, скороченої тривалості роботи у нічний час. Максимальна тривалість робочого часу, мінімальна тривалість відпочинку та оплачуваної щорічної відпустки, вихідні та святкові дні, а також інші умови здійснення цього права визначаються законом.

Ст. 46. Громадяни мають право на соціальний захист, що включає право на забезпечення їх у разі повної, часткової або тимчасової втрати працездатності, втрати годувальника, безробіття з незалежних від них обставин, а також у старості та в інших випадках, передбачених законом. Це право гарантується загальнообов'язковим державним соціальним страхуванням за рахунок страхових внесків громадян, підприємств, установ і організацій...

Ст. 49. Кожен має право на охорону здоров'я, медичну допомогу та медичне страхування [16].

Охорона праці охоплює головні складові: правові норми трудового законодавства, техніку безпеки, виробничу санітарію, протипожежний захист та електробезпеку.

Метою охорони праці є забезпечення безпечних та придатних умов для праці.

4.2 Аналіз небезпечних та шкідливих умов, які можливі відділі розробки програмного забезпечення

Сучасна професійна діяльність фахівців невиробничої сфери характеризується збільшенням обсягів інформації, високою швидкістю її опрацювання і значною кількістю рішень, що приймаються з використанням різноманітних технічних засобів. Наразі в офісних приміщеннях дедалі ширше персональні комп'ютери (ПК), факси, ксерокопіювальні апарати, системи електронних платежів, системи комплексної автоматизації внутрішньофірмової діяльності тощо [17]. Проаналізуємо чинники детальніше.

Небезпечно використовувати монітори з частотою оновлення менше 75 Гц та з низькою роздільною здатністю, згідно стандартами.

Найбільш прийнятними кліматичними умовами є температури 23° у теплу пору року та 18° у холодну при 55% відносної вологості.

Погане освітлення погано впливає на робочий процес, а також на зір працівника. Тому прийнятна освітленість робочої зони повинна бути у межах 400-750 лк.

Також можуть виникнути й інші загрози у процесі трудової діяльності. Загроза ураження електричним струмом. Прийнятним струмом можна вважати такі, що не становлять загрози організму людини, а саме змінний струм промислової частоти 0,6-1,5 мА та постійний струм 5-7 мА.

Небезпека виникнення пожежі. Категорія пожежної небезпеки "В" включає споруди, де розміщені персональні комп'ютери. У таких приміщеннях як правило чимало джерел спалаху: кабелі, техніка, устаткування, меблі, папір тощо.

Підвищення технічного рівня виробництва нерозривно зв'язано з широким використанням різних видів електроніки, телебачення, оптичних квантових генераторів, електронно-обчислювальних машин. Усі ці пристрої є джерелами електромагнітних випромінювань, що зараз прийнято називати єдиним терміном неіонізуючі випромінювання. Коло осіб, що піддаються впливові таких випромінювань, увесь час розширюється.

4.3 Розробка заходів по зменшенню дії небезпечних і шкідливих факторів у відділі розробки програмного забезпечення

Монітори, повинні відповідати стандартам з вертикальною частотою розгортки не менше 75 Гц. Мікрокліматичні показники повинні бути оптимальними (23° С та 18° С у теплу та холодну пору року відповідно), рекомендовано, щоб в офісі був кондиціонер. Вологе прибирання у приміщенні з ПК повинно проводитись щодня. Потрібно встановити пристрої заземлення для безпеки персоналу.

Щодо освітлення у темні години дня слід користуватись люмінесцентними лампами, бо вони є довговічними (10000+год.) та мають високу світлову віддачу (75+лм/Вт).

Система пожежної безпеки задовольняє норми, але для покращення умов можна ще встановити підлогу з негорючої сировини, димові датчики тощо.

Шум може негативно впливати на працездатність людини на роботі. Враховуються такі параметри звуку, тиск, інтенсивність, частота коливань. відповідно до норм ДСН 3.3.6.037-99, рівень шуму не повинен перевищувати 50 дБ.

Нижче наведено перелік елементів, які можуть створювати шум у відділі:

- система охолодження персонального комп'ютера;
- працюючий принтер;
- транспорт із зовні;
- кондиціонер
- система охолодження серверу.

Згідно з НАПБ Б.03.002-2007 відділ належить до категорії В - приміщення, у якому наявні горючі тверді, волокнисті матеріали.

Відповідно до ПБУ, у кімнаті, що аналізується, необхідно мати:

1. Один вуглекислотний вогнегасник (3+кг/20м²). Із розрахунку на площу даного приміщення: 4 переносні вогнегасники марки ВВ-5 (по 3,5 кг);
2. План для евакуації з офісу;
3. Система протипожежних датчиків, пожежна сигналізація.

В офісній кімнаті 1 вихід та менше 25 працівників. Між робочими місцями у кімнаті проходи шириною більше 1 м. Сходові клітини комбінують природне та штучне освітлення. Сходи та офіс мають систему евакуаційного освітлення. Робітники ознайомлені з порядком і планом евакуування з офісу.

Отже, шляхи евакуації з приміщення повністю відповідають нормам.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було успішно вирішено практичну задачу інженерії програмного забезпечення з розробки вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ», що характеризується комплексністю та невизначеністю умов, а саме розроблено вказаний вебзастосунок.

Для досягнення цієї цілі було виконано такі завдання: аналіз вимог до вебзастосунку, розроблено архітектуру вебзастосунку, діаграми діяльності, концептуальну модель, діаграму класів, діаграми взаємодії, логічну модель БД, робочий проєкт, представлено результати розробки.

Під час виконання кваліфікаційної роботи було розроблено усю необхідну програмну документацію, яка представлена у додатках.

У результаті було створено вебзастосунок для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Акімова Н. С., Говоруха О. О., Євлаш Т. О., Топоркова О. В. Облік та аналіз дебіторської заборгованості в системі управління підприємств оптової торгівлі: монографія. Харків: ХДУХТ, 2016. 285 с.
2. BAS Бухгалтерія. URL: <https://inteltech.com.ua/uk/bas-buhgalteriya-prof> (дата звернення: 20.03.2025).
3. Finmap. URL: <https://www.finmap.online/> (дата звернення: 20.03.2025).
4. Шаблон проектування MVC. URL: https://informatika.udpu.edu.ua/?page_id=5450 (дата звернення: 28.04.2025).
5. UML: Огляд основних типів діаграм, діаграма компонентів. Частина 2. URL: <https://habr.com/articles/756552/> (дата звернення: 29.04.2025).
6. Що таке Java. <https://dou.ua/lenta/articles/how-to-learn-java/> (дата звернення: 27.05.2025).
7. HTML та CSS для тестувальників. <https://qalight.ua/baza-znaniy/html-ta-css-dlya-testuvalnykiv/> (дата звернення: 27.05.2025).
8. IntelliJ IDEA. https://uk.wikipedia.org/wiki/IntelliJ_IDEA#cite_note-:0-4 (дата звернення: 27.05.2025).
9. PostgreSQL: введення та встановлення. <https://thehost.ua/ua/wiki/administration/database/postgresql-install> (дата звернення: 27.05.2025).
10. Огляд початкового завантаження. <https://learn.microsoft.com/uk-ua/power-pages/configure/bootstrap-overview> (дата звернення: 27.05.2025).
11. Spring і Spring Boot: основні відмінності та особливості використання. <https://foxminded.ua/vidminnosti-mizh-spring-i-spring-boot/> (дата звернення: 27.05.2025).
12. Hibernate. <https://uk.wikipedia.org/wiki/Hibernate> (дата звернення: 27.05.2025).
13. Що таке Docker? https://westelecom.ua/blog/287_cto-takoe-docker.html (дата звернення: 27.05.2025).

14. Основи Maven. <https://javarush.com/ua/groups/posts/uk.2523.chastina-4-osnovi-maven> (дата звернення: 27.05.2025).

15. Про охорону праці: Закон України. URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> (дата звернення: 21.03.2025).

16. Конституція України. URL: <https://zakon.rada.gov.ua/laws/show/254%D0%BA/96-%D0%B2%D1%80#Text> (дата звернення: 21.03.2025).

17. Шкідливі та небезпечні чинники, що діють на працівника сучасного офісу. URL: <https://dspace.nuft.edu.ua/server/api/core/bitstreams/0a78dc47-3c8f-4466-80da-c100ecb4769d/content> (дата звернення: 21.03.2025).

Додаток А – Технічне завдання

1 Вступ

Вебзастосунок для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» є застосунком, спрямованим на ведення дебіторської заборгованості та банківських виписок у сфері послуг для виявлення своєчасного виявлення боржників та формування фінансових звітів.. Вебзастосунок надає ряд функцій для зручного та точного обліку фінансових операцій.

2 Підстави для розробки

Підставою для розробки даного програмного забезпечення є наказ н затвердження тем кваліфікаційних робіт бакалаврів зі спеціальності 121 Інженерія програмного забезпечення в Національному університет кораблебудування ім. адмірала Макарова.

3 Призначення розробки

3.1 Експлуатаційне призначення

Експлуатаційним призначенням розробки є ведення обліку дебіторської заборгованості та банківських виписок у сфері послуг для виявлення своєчасного виявлення боржників та формування фінансових звітів. Цей вебзастосунок вирішує завдання спрощення ведення фінансових операцій та надає зручні інструменти для точного обліку/обігу грошей.

Завдяки його функціоналу, підприємства можуть автоматизувати основні фінансові процеси, що дозволяє ефективно використовувати ресурси та зосереджуватися на стратегічному розвитку.

Крім цього, вебзастосунок дозволяє генерувати звіти та аналітику, що сприяє оперативному прийняттю управлінських рішень.

3.2 Функціональне призначення

Функціональним призначенням даного застосунку є вирішення користувачем таких завдань:

- Реєстрація та авторизація.
- Створення компаній та додавання всіх банківських рахунків, які має ця компанія.
- Для кожної компанії можна додавати або видаляти працівників даної компанії, які можливість працювати з даними.
- Можливість редагувати інформацію про компанію та додавати нові банківські рахунки.
- Переглядати дані, редагувати, видаляти та створювати нові записи по банківських виписках по всіх банківських рахунках та компаніям окремо.
- Переглядати дані, редагувати, видаляти та створювати нові записи щодо наданих послуг компанією.
- Видаляти компанії та банківські рахунки.
- Формування звіту дебіторських заборгованостей по кожній компанії окремо.
- Формування аналітичних звітів (місячних, квартальних, річних) по банківських виписках по кожній компанії окремо.
- Формування аналітичних звітів (місячних, квартальних, річних) по наданим послугам по кожній компанії окремо.

4 Вимоги до програми та програмної документації

4.1 Вимоги до функціональних характеристик

4.1.1 Вимоги до складу виконуваних функцій

- Реєстрація та авторизація. Користувач повинен мати можливість створити новий обліковий запис, щоб мати можливість працювати з платформою. Якщо він вже має створений аккаунт він має можливість увійти до свого кабінету.
- Створювати компанію та додавати всі банківські рахунки, які має ця компанія. Користувач повинен мати можливість у своєму кабінеті додати власну

компанію і при створенні цієї компанії у застосунку повинен мати можливість створювати банківські рахунки, які є у цієї компанії.

- Для кожної компанії можна додавати або видаляти працівників даної компанії, які мають можливість працювати з даними. Користувач повинен мати можливість після створення компанії додавати довірених осіб, які мають можливість переглядати, редагувати, видаляти та створювати нові записи. Користувач повинен мати можливість видаляти довірених осіб, що позбавляє їх доступу до даних.

- Можливість редагувати інформацію про компанію та додавати нові банківські рахунки. Користувач повинен мати можливість при необхідності змінювати будь яку інформацію про компанії, додавати нові банківські рахунки.

- Переглядати дані, редагувати, видаляти та створювати нові записи по банківських виписках по всіх банківських рахунках та компаніям. Користувач повинен переглядати дані, редагувати, видаляти та створювати нові записи по банківських виписках по всіх банківських рахунках та компаніям, якщо виникає необхідність оновлення даних, або імпортування їх з інших платформ.

- Переглядати дані, редагувати, видаляти та створювати нові записи щодо наданих послуг компанією. Користувач повинен мати можливість переглядати записи, редагувати записи, видаляти записи та створювати нові записи наданих послуг компанією.

- Видалення компаній та банківських рахунків. Користувач повинен мати можливість видаляти створені компанії та банківські рахунки, які вже не будуть оброблятися.

- Формування дебіторських заборгованостей по кожній компанії. Користувач повинен мати можливість формувати «Дебіторську заборгованість» на основі отриманих записів банківської виписки та записів наданих послуг.

- Формування аналітичних звітів (місячних, загальних) по банківських виписках по кожній компанії. Користувач повинен мати можливість формувати аналітичні звіти по існуючих у системі банківським випискам.

- Формування аналітичних звітів (місячних, квартальних, річних) по наданим послугам по кожній компанії. Користувач повинен мати можливість формувати аналітичні звіти по існуючих у системі даних наданих послуг компаніїю.

4.1.2 Вимоги до організації вхідних і вихідних даних

Вхідні дані:

- Реєстрація. для створення власного облікового запису Користувач повинен ввести такі дані. ПІБ, email, телефон, пароль. Після введення цих даних користувач повинен підтвердити свій телефон та email, вказавши код у спеціальні поля надісланий на телефон у формі SMS, та електронний лист надісланий на email.
- Авторизація: для входження до свого облікового запису Користувач повинен ввести email/ телефон та пароль.
- Дані про компанію. Користувач при створенні компанії повинен ввести назву своєї компанії, код ЄДРПОУ та створити банківські рахунки.
- Дані про банківські рахунки. при створенні банківського рахунку користувач повинен вказати до якого банку відноситься даний банківський рахунок, назву банківського рахунку, IBAN.

Вихідні дані:

- Звіт дебіторської заборгованості надається у виді таблиці та надає таку інформацію. загальна заборгованість, дані про боржників (та скільки вони винні)
- Аналітика по банківській виписці Cash Flow (CF) та The profit and loss (P&L) надається у виді таблиці та надає таку інформацію. загальні доходи/витрати та всі доходи/витрати класифіковані по категоріям та підкатегоріям.
- Аналітика по послугам наданих компаніїю надається у виді таблиці та надає таку інформацію. загальні доходи та доходи класифіковані по категоріям та підкатегоріям.

4.2 Вимоги до надійності

Надійне функціонування застосунку має бути забезпечене виконанням сукупності організаційно-технічних заходів, перелік яких подано нижче:

- система повинна мати можливість автоматичного відновлення після відмови або помилки протягом 15 хвилин. Це може включати резервне копіювання бази даних, автоматичну перезавантаження серверів та інші заходи для забезпечення продовження роботи системи:
- захист даних від несанкціонованого доступу.
- валідація даних перед додаванням або оновленням будь-якої інформації, важливо перевіряти вхідні дані на відповідність формату та обмежень

4.3 Вимоги до складу й характеристик технічних засобів

Сервер повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 4-х ядерний процесор з тактовою частотою 2,0 ГГц.
- Оперативна пам'ять: 32 Гб.
- Жорсткий диск: 1Тб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

Комп'ютер користувача повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 2-х ядерний процесор з тактовою частотою 1,8 ГГц.
- Оперативна пам'ять: 4 Гб.
- Жорсткий диск: 100 Гб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

4.4 Вимоги до інформаційної й програмної сумісності

На сервері повинна бути встановлена одна з операційних систем: Windows 11 не нижче версії 21H2, Linux Ubuntu не нижче версії 22.04.3 LTS, Linux Debian не нижче версії 12, Linux Fedora не нижче версії 38 або MacOS версії не нижче Monterey. На серверному пристрої повинен бути встановлений Docker Desktop не нижче версії 4.24.0.

На комп'ютері користувача повинна бути встановлена одна з операційних систем: Windows 11 не нижче версії 21H2. Linux Ubuntu не нижче версії 22.04.3 LTS, Linux Debian не нижче версії 12, Linux Fedora не нижче версії 38 або MacOS версії Monterey. На клієнтському пристрої повинен бути встановлений веб-браузер Google Chrome не нижче версії 117, або Mozilla Firefox не нижче версії 118.0.

4.4.1 Вимоги до захисту інформації

Вимоги до захисту інформації:

- забезпечити шифрування та безпеку передачі даних між клієнтом і сервером за допомогою протоколу HTTPS;
- усі паролі Користувачів повинні бути збережені у зашифрованому вигляді.

4.5 Вимоги до програмної документації

У програмну документацію повинні входити наступні документи:

- технічне завдання;
- програма та методика випробувань;
- керівництво Користувача;
- опис програми;
- код програми.

5 Техніко-економічні показники

Техніко-економічні показники не розраховуються.

6 Стадії й етапи розробки

Стадії та етапи розробки представлені у таблиці А.1.

Таблиця А.1 – Стадії й етапи розробки

Номер	Назва етапів розробки	Термін виконання
1	Аналіз практичної задачі інженерії ПЗ	09.03.2025
2	Аналіз вимог до ПЗ	15.03.2025
3	Розробка архітектури ПЗ	30.03.2025
4	Розробка проекту ПЗ	15.04.2025
5	Реалізація та тестування ПЗ	30.04.2025

7 Порядок контролю та приймання

Контроль за аналізом та проектуванням кожної окремої частини програмного забезпечення на кожному етапі треба робити з урахуванням вимог, визначених у технічному завданні. Кожна стадія розробки повинна бути представлена в зазначені строки.

Приймання проводиться відповідно до програми і методики випробувань і документують за допомогою протоколу проведення випробувань. У разі знаходження помилок під час приймання програмного виробу складається акт про знайдені помилки. Розробник повинен на протязі не більше ніж 2 тижнів виправити зазначені помилки.

Додаток Б – Опис програмного забезпечення

1 Загальні відомості

Розроблений вебзастосунок для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» представляє собою застосунок, який облегшує розрахунки різних операцій по компанії, аналіз банківських виписок та виявлення боржників компанії та формування аналітичних звітів по рахунках компанії, що облегшить роботу фінансових директорів.

Для розробки вебзастосунку були обрані наступні записи:

Мови: Java, HTML, CSS, JavaScript

СУБД: PostgreSQL

Середовище розробки: IntelliJ Idea

Фреймворки: Spring Framework, Hibernate, Lombok, Bootstrap

Для роботи з даним застосунком потрібен пристрій з будь-яким браузером та з підключенням до інтернету.

2 Функціональне призначення

Вебзастосунок виконує наступні функції:

- Реєстрація та авторизація
- Створення, перегляд, редагування, видалення компаній
- Створення, перегляд, редагування, видалення рахунків
- Створення, перегляд, редагування, видалення записів по банківських виписках
- Створення, перегляд, редагування, видалення записів наданих послуг компанією
- Додавання, видалення працівників
- Формування дебіторських заборгованостей
- Формування аналітичних звітів (місячних, квартальних, річних) по банківських виписках

- Формування аналітичних звітів (місячних, квартальних, річних) по наданим послугам

3 Опис логічної структури

Застосунок було розроблено на модульній основі, у якій кожен компонент відповідає дії. Архітектура побудована за принципами MVC, що забезпечує зручність у супроводі, масштабуванні.

Додатково застосунок побудований за клієнт-серверною архітектурою, де клієнтська частина виконує взаємодію з серверною частиною, представленою базою даних. Це дозволяє зберігати дані користувача у хмарі, синхронізувати інформацію між пристроями та забезпечувати безперервну роботу додатку.

4 Технічні засоби

Сервер повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 4-х ядерний процесор з тактовою частотою 2,0 ГГц.
- Оперативна пам'ять: 32 Гб.
- Жорсткий диск: 1Тб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

Комп'ютер користувача повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 2-х ядерний процесор з тактовою частотою 1,8 ГГц.
- Оперативна пам'ять: 4 Гб.
- Жорсткий диск: 100 Гб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

5 Виклик і завантаження

Вебзастосунок для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» запускається при переході за певним посиланням у браузері. Після відкриття завантажується вікно авторизації(якщо у користувача немає аккаунту, то він може створити новий через сторінку реєстрації). У разі правильного введення даних відкривається головна сторінка застосунку, яка надає доступ до основних функцій вебзастосунку.

6 Вхідні дані

Введення даних здійснюється за допомогою клавіатури. Дані можуть вводитися вручну або обиратися із представлених списків.

Вхідними даними будуть:

- Інформація про користувача(Ім'я, імейл, телефон)
- Інформація про компанію(Назва, код ЄДРПОУ)
- Інформація про рахунок(Назва, IBAN, банк)
- Інформація про запис банківської виписки(дата, вид, деталі, контрагент, сума, валюта, категорія, примітка, бухгалтер)
- Інформація про запис про надану послугу компанією(дата, вид, деталі, контрагент, сума, валюта, категорія, примітка, бухгалтер)
- Інформація про контрагента(Ім'я, телефон, імейл)

7 Вихідні дані

Виведення даних здійснюється через екран монітору у вигляді текстової інформації.

Вихідними даними будуть:

- Дані про користувача(Ім'я, імейл, телефон)
- Дані про компанію(Назва, код ЄДРПОУ)
- Дані про рахунок(Назва, IBAN, банк)
- Дані про запис банківської виписки(дата, вид, деталі, контрагент, сума, валюта, категорія, примітка, бухгалтер)

- Дані про запис про надану послугу компанією(дата, вид, деталі, контрагент, сума, валюта, категорія, примітка, бухгалтер)
- Дані про контрагента(Ім'я, телефон, імейл)
- Дебіторська заборгованість надається у виді таблиці та надає таку інформацію: загальна заборгованість, дані про боржників(та скільки вони винні)
- Аналітика по банківській виписці CF та P&L надається у виді таблиці та надає таку інформацію: загальні доходи/витрати та всі доходи/витрати класифіковані по категоріям та підкатегоріям.
- Аналітика по послугам наданих компанією надається у виді таблиці та надає таку інформацію: загальні доходи та доходи класифіковані по категоріям та підкатегоріям.

Додаток В – Програма та методика випробувань

1 Об'єкт випробувань

1.1 Найменування випробуваної програми

Повна назва об'єкта випробувань: вебзастосунок для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ».

1.2 Область застосування випробовуємої програми

Призначення вебзастосунку для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» полягає в тому, щоб забезпечити користувачам зручний і простий у використанні інструмент для розрахунків різних операцій по компанії, аналіз банківських виписок та виявлення боржників компанії та формування аналітичних звітів по рахунках компанії.

Програма забезпечує:

- Реєстрація та авторизація
- Створення, перегляд, редагування, видалення компаній
- Створення, перегляд, редагування, видалення рахунків
- Створення, перегляд, редагування, видалення записів по банківських виписках
- Створення, перегляд, редагування, видалення записів наданих послуг компанією
- Додавання, видалення працівників
- Формування дебіторських заборгованостей
- Формування аналітичних звітів (місячних, квартальних, річних) по банківських виписках
- Формування аналітичних звітів (місячних, квартальних, річних) по наданим послугам

1.3 Позначення випробуваної програми

Вебзастосунок для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ».

2 Мета випробувань

Мета проведення випробувань є необхідність перевірки відповідності програмного забезпечення вимогам, які викладені у технічному завданні, що наводиться у додатку А.

3 Вимоги до програми

Програма має реалізовувати функції, описані в технічному завданні, яке наведено в додатку А.

4 Вимоги до програмної документації

Для проведення випробувань повинна бути надана наступна документація:

- технічне завдання на розробку програмного забезпечення;
- текст програми;
- опис програми;
- інструкція користувача;
- програма і методика випробувань програмного забезпечення.

5 Засоби і порядок випробувань

5.1 Програмні засоби випробувань

До програмних засобів, необхідних для випробувань, належать:

- На сервері повинна бути встановлена одна з операційних систем: Windows 11 не нижче версії 21H2, Linux Ubuntu не нижче версії 22.04.3 LTS, Linux Debian не нижче версії 12, Linux Fedora не нижче версії 38 або MacOS версії не нижче Monterey. На серверному пристрої повинен бути встановлений Docker Desktop не нижче версії 4.24.0.

- На комп'ютері користувача повинна бути встановлена одна з операційних систем: Windows 11 не нижче версії 21H2. Linux Ubuntu не нижче версії 22.04.3 LTS, Linux Debian не нижче версії 12, Linux Fedora не нижче версії 38 або Monterey. На клієнтському пристрої повинен бути встановлений веб-браузер Google Chrome не нижче версії 117, або Mozilla Firefox не нижче версії 118.0.

5.2 Апаратні засоби випробувань

Сервер повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 4-х ядерний процесор з тактовою частотою 2,0 ГГц.
- Оперативна пам'ять: 32 Гб.
- Жорсткий диск: 1Тб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

Комп'ютер користувача повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 2-х ядерний процесор з тактовою частотою 1,8 ГГц.
- Оперативна пам'ять: 4 Гб.
- Жорсткий диск: 100 Гб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

5.3 Порядок проведення випробувань

Проведення випробування програмного забезпечення повинно відбуватися в наступному порядку:

- виконується інструкції до кожної функції;
- робиться аналіз отриманих результатів і встановлюється відповідність програмного продукту вимогам і системним документам.

При виявленні помилок у роботі системи складається їх перелік і обговорюється термін їх виправлення розробником. Після цього замовник проводить повторне тестування в повному обсязі (можливе використання нових додатків чи тестів).

6 Методи випробувань

Випробування буде проводитись за методом «чорної скриньки». Через велику кількість функцій які треба випробувати, буде представлено тільки результати декількох функцій застосунку. Програма випробувань наведено у таблиці В.1.

Таблиця В.1 – Програма випробувань

№	Дія	Результат
1	Авторизація у системі з коректними даними	Успішна авторизація у системі, перехід на сторінку компаній
2	Додавання нової компанії	Успішне створення нової компанії відображення його у списку компаній
3	Додавання нового рахунку	Успішне створення нового рахунку відображення його у списку рахунків
4	Видалення рахунку	Успішне видалення рахунку з системи
5	Реєстрація нового користувача з коректними даними	Успішна реєстрація у системі, перехід на авторизації, успішна авторизація
6	Видалення працівника компанії	Успішне видалення працівника

Додаток Г – Керівництво користувача

1 Призначення програми

1.1 Функціональне призначення програми

Функціональним призначенням даного застосунку є вирішення користувачем таких завдань:

- Реєстрація та авторизація.
- Створення компаній та додавання всіх банківських рахунків, які має ця компанія.
- Для кожної компанії можна додавати або видаляти працівників даної компанії, які можливість працювати з даними.
- Можливість редагувати інформацію про компанію та банківські рахунки.
- Переглядати дані, редагувати, видаляти та створювати нові записи по банківських виписках по всіх банківських рахунках по компаніям.
- Переглядати дані, редагувати, видаляти та створювати нові записи щодо наданих послуг компанією.
- Видаляти компанії та банківські рахунки.
- Формування дебіторських заборгованостей по кожній компанії окремо.
- Формування аналітичних звітів (місячних, квартальних, річних) по банківських виписках по кожній компанії окремо.
- Формування аналітичних звітів (місячних, квартальних, річних) по наданим послугам по кожній компанії окремо.

1.4 Експлуатаційне призначення програми

Експлуатаційним призначенням вебзастосунку є ефективне ведення обліку дебіторської заборгованості та банківських виписок у сфері послуг. Цей вебзастосунок вирішує завдання оптимізації фінансових операцій та надає зручні інструменти для точного обліку/обігу грошей. Завдяки його функціоналу, підприємства можуть автоматизувати основні фінансові процеси, що дозволяє ефективно використовувати ресурси та зосереджуватися на стратегічному розвитку. Крім цього, вебзастосунок дозволяє генерувати звіти та аналітику, що

сприяє оперативному прийняттю управлінських рішень.

1.5 Склад функцій

Вебзастосунок для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» забезпечує користувачів такими функціями:

Програма забезпечує:

- Реєстрація та авторизація
- Створення, перегляд, редагування, видалення компаній
- Створення, перегляд, редагування, видалення рахунків
- Створення, перегляд, редагування, видалення записів по банківських виписках
- Створення, перегляд, редагування, видалення записів наданих послуг компанією
- Додавання, видалення працівників
- Формування дебіторських заборгованостей
- Формування аналітичних звітів (місячних, квартальних, річних) по банківських виписках
- Формування аналітичних звітів (місячних, квартальних, річних) по наданим послугам

2 Умови виконання програми

2.1 Мінімальний склад апаратних засобів

Сервер повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 4-х ядерний процесор з тактовою частотою 2,0 ГГц.
- Оперативна пам'ять: 32 Гб.
- Жорсткий диск: 1Тб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

Комп'ютер користувача повинен задовольняти вказаним вимогам наступної мінімальної комплектації:

- Процесор: 2-х ядерний процесор з тактовою частотою 1,8 ГГц.
- Оперативна пам'ять: 4 Гб.
- Жорсткий диск: 100 Гб вільного дискового простору.
- Мережеві можливості: мережевий адаптер з підтримкою гігабітної Ethernet.

2.2 Мінімальний склад програмних засобів

До програмних засобів, належать:

- На сервері повинна бути встановлена одна з операційних систем: Windows 11 не нижче версії 21H2, Linux Ubuntu не нижче версії 22.04.3 LTS, Linux Debian не нижче версії 12, Linux Fedora не нижче версії 38 або MacOS версії не нижче Monterey. На серверному пристрої повинен бути встановлений Docker Desktop не нижче версії 4.24.0.

- На комп'ютері користувача повинна бути встановлена одна з операційних систем: Windows 11 не нижче версії 21H2. Linux Ubuntu не нижче версії 22.04.3 LTS, Linux Debian не нижче версії 12, Linux Fedora не нижче версії 38 або Monterey. На клієнтському пристрої повинен бути встановлений веб-браузер Google Chrome не нижче версії 117, або Mozilla Firefox не нижче версії 118.0.

2.3 Вимоги до користувачів

Вебзастосунок розрахований для користувачів, які володіють навичками роботи з комп'ютерами.

Вимоги до користувача:

- Вміння працювати у браузері;
- Здатність зареєструватися в системі та виконувати вхід до облікового запису;
- Навички роботи з таблицями.

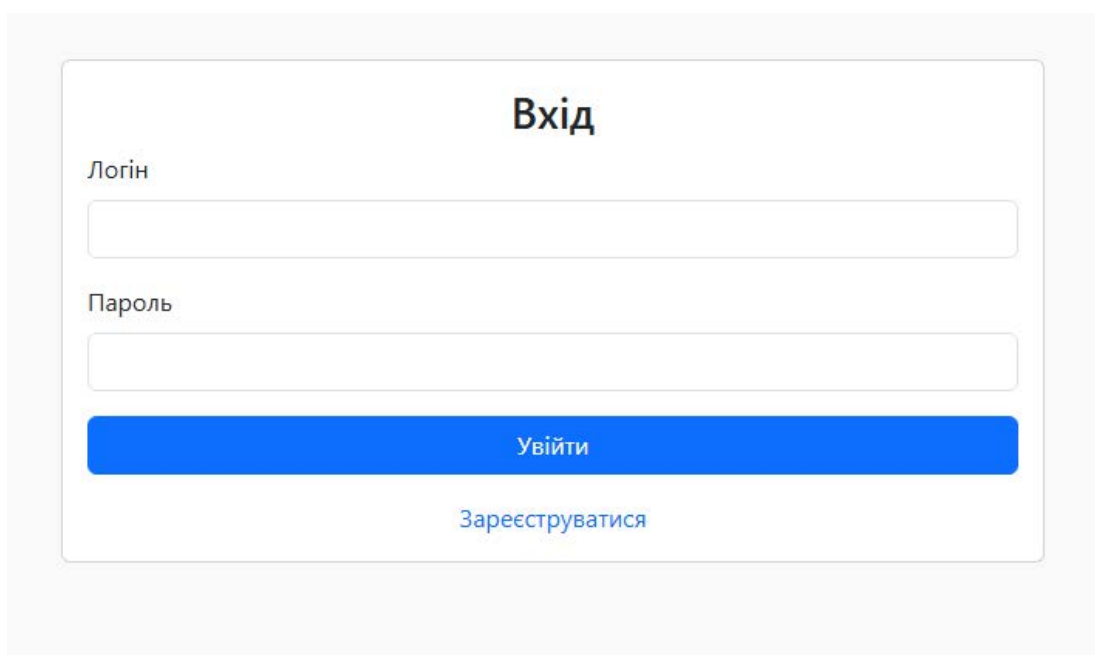
3 Виконання програми

3.1 Завантаження та запуск програми

Вебзастосунок для фінансового аналізу дебіторів для юридичної компанії «ЛЕМБЕРГ» запускається при переході за певним посиланням у браузері. Після відкриття завантажується вікно авторизації(якщо у користувача немає аккаунту, то він може створити новий через сторінку реєстрації). У разі правильного введення даних відкривається головна сторінка застосунку, яка надає доступ до основних функцій вебзастосунку.

3.2 Виконання програми

При відкритому вебзастосунку користувача вітає сторінка авторизації, представлено на рисунку Г.1.

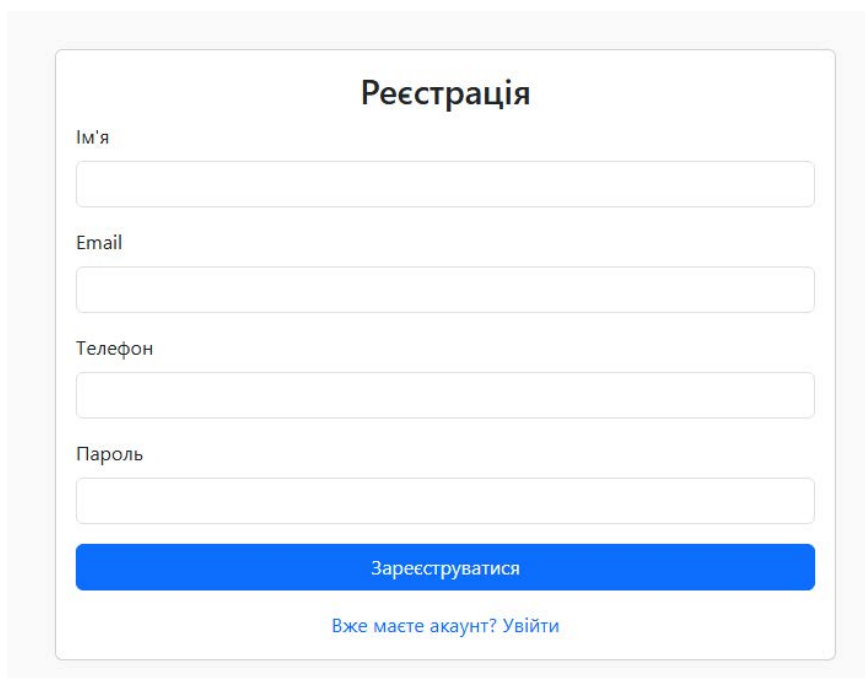


The image shows a login form with the title "Вхід" (Login) in bold. It contains two input fields: "Логін" (Login) and "Пароль" (Password). Below the password field is a blue button labeled "Увійти" (Login). At the bottom of the form is a blue link labeled "Зареєструватися" (Register).

Рисунок Г.1 – Сторінка авторизації

Якщо користувач був авторизований у системі він повинен ввести email та пароль, та натиснути кнопку «Увійти», при валідних даних користувача спрямовує на головну сторінку. Якщо користувач не був авторизований у системі, то він повинен зареєструватись, щоб зареєструвати користувач повинен на сторінці авторизації перейти за посиланням «Зареєструватися» під кнопкою «Увійти», після користувача спрямує на сторінку реєстрації, де користувач

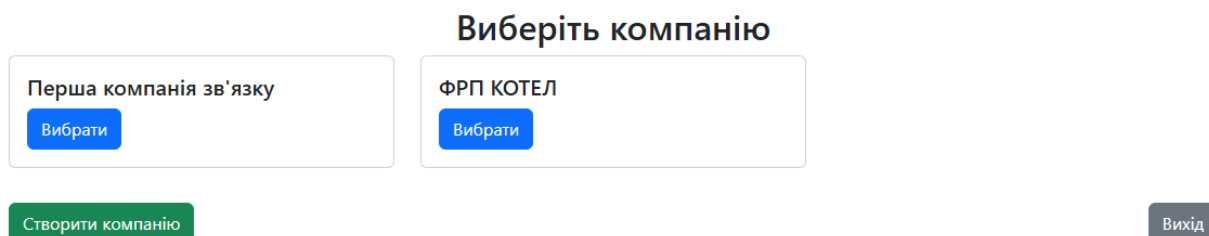
повинен ввести дані для реєстрації нового аккаунту та натиснути кнопку «Зареєструватися», представлено на рисунку Г.2.



The image shows a registration form titled "Реєстрація". It contains four input fields: "Ім'я" (Name), "Email", "Телефон" (Phone), and "Пароль" (Password). Below the fields is a blue button labeled "Зареєструватися" (Register). At the bottom, there is a link that says "Вже маєте акаунт? Увійти" (Already have an account? Log in).

Рисунок Г.2 – Сторінка реєстрації

На головній сторінці вебзастосунок запропонує вибрати компанію, якщо користувач її створював та натиснути на неї. Якщо це тільки зареєстрований користувач, то вебзастосунок запропонує створити компанію після чого з'явиться спливаюче вікно з полями вводу для створення компанії, як користувач введе всі дані він повинен натиснути кнопку «Створити», представлено на рисунку Г.3.



The image shows a company selection interface titled "Виберіть компанію". It features two selectable options: "Перша компанія зв'язку" (First company to contact) and "ФРП КОТЕЛ" (FRP KOTEL). Each option has a blue "Вибрати" (Select) button. At the bottom, there is a green "Створити компанію" (Create company) button and a grey "Вихід" (Exit) button.

Рисунок Г.3 – Сторінка вибору компанії

Після користувача спрямує на сторінку дій. Ця сторінка включає такі кнопки навігації: «Рахунки», «Аналітика», «Операції» та «Працівники», представлено на рисунку Г.4.

Панель керування - Перша компанія зв'язку

Рахунки Аналітика Операції Працівники Logout

Рисунок Г.4 – Сторінка дій

На сторінці «Рахунки» перед користувачем будуть відображені кнопка «Створити новий рахунок» та всі рахунки по вибраній компанії, у табличному вигляді, де справа від кожного рахунку буде по дві кнопки: «Перейти», «Переглянути». У користувача є можливість перейти до записів з банківських виписок, натиснувши «Перейти», представлено на рисунку Г.5.

Рахунки

Створити новий рахунок

Назва	IBAN	Банк	Дії	
Перший	UA9477495346456345345523543525	PrivatBank	Перейти	Переглянути
Другий	UA947749534443534534345523543525	MonoBank	Перейти	Переглянути
Рахунок1	UA123456789012345678901234567	PrivatBank	Перейти	Переглянути

Рисунок Г.5 – Сторінка рахунків

При натисканні на кнопку «Переглянути», перед користувачем відкривається спливаюче вікно з інформацією у полях, яку можна редагувати, а знизу спливаючого вікна будуть кнопки: «Видалити», «Скасувати», «Зберегти», представлено на рисунку Г.6.

Редагувати рахунок

Назва

Перший

IBAN

UA9477495346456345345523543525

Банк

PrivatBank

Видалити

Скасувати

Зберегти

Рисунок Г.6 – Спливаюче вікно з інформацією про рахунок

При натисканні на кнопку «Створити новий рахунок» на екрані з’явиться спливаюче вікно з полями вводу для створення рахунку, як користувач введе всі дані він повинен натиснути кнопку «Створити», щоб створити рахунок, або «Скасувати», для скасування створення рахунку.

На сторінці записів з банківських виписок перед користувачем з’являться зліва зверху селектори з вибором місяць/квартал, який місяць/квартал, рік та кнопка «Показати» та справа зверху кнопка «Додати», представлено на рисунку Г.7.

Банківські виписки

Місяць

Червень

2025

Показати

Додати

Дата	Сума	Валюта	Опис	Тип	Категорія	Контрагент	Дії
02-06-2025T18:57	500.0	UAH	Плата за зв'язок	Дебет	зв'язок	Петро	Переглянути Видалити

Рисунок Г.7 – Сторінка записів з банківських виписок

При виборі всіх селекторів та натисканні «Показати» нижче у табличній формі буде відображено всі записи за вказаними селекторами, де справа від кожного запису буде по дві кнопки: «Переглянути», «Видалити». При натисканні «Додати» на екрані з'явиться спливаюче вікно з полями вводу для створення запису, як користувач введе всі дані він повинен натиснути кнопку «Створити», щоб створити запис, або «Скасувати», для скасування створення запису. При натисканні на кнопку «Переглянути», перед користувачем відкривається спливаюче вікно з інформацією у полях, яку можна редагувати, а знизу спливаючого вікна будуть кнопки: «Скасувати», «Зберегти». При натисканні на кнопку «Видалити», відповідний запис буде видалено.

На сторінці «Аналітика» перед користувачем будуть відображені селектори для вибору періоду за який буде виконано підрахунок, та її тип, після користувач повинен натиснути на кнопку «Розрахувати» і буде відображено дані за вказаним типом та періодом, представлено на рисунку Г.8.

Аналітика

Місяць ▾

Січень ▾

Дохід ▾

Розрахувати

Рисунок Г.8 – Сторінка аналітики

На сторінці «Операції» перед користувачем з'являться зліва зверху селектори з вибором місяць/квартал, який місяць/квартал, рік та кнопка «Показати» та справа зверху кнопка «Додати», представлено на рисунку Г.9.

Операції

Місяць ▾

Січень ▾

2023 ▾

Показати

Додати

Дата	Сума	Валюта	Опис	Тип	Категорія	Бухгалтер	Контрагент	Дії
02.06.2025 18:00	500.0	UAH	Плата за зв'язок	Дебет	Зв'язок	Ірина	Петро	Переглянути Видалити

Рисунок Г.9 – Сторінка операцій

При виборі всіх селекторів та натисканні «Показати» нижче у табличній формі буде відображено всі записи за вказаними селекторами, де справа від кожного запису буде по дві кнопки: «Переглянути», «Видалити». При натисканні «Додати» на екрані з'явиться спливаюче вікно з полями вводу для створення запису, як користувач введе всі дані він повинен натиснути кнопку «Створити», щоб створити запис, або «Скасувати», для скасування створення запису. При натисканні на кнопку «Переглянути», перед користувачем відкривається спливаюче вікно з інформацією у полях, яку можна редагувати, а знизу спливаючого вікна будуть кнопки: «Скасувати», «Зберегти», представлено на рисунку Г.10. При натисканні на кнопку «Видалити», відповідний запис буде видалено.

Редагувати операцію

Дата
02.06.2025 18:00

Сума
500,0

Валюта
UAH

Опис
Плата за зв'язок

Тип
Дебет

Категорія
Зв'язок

Бухгалтер
Ірина

Контрагент
Петро

Скасувати Зберегти

Рисунок Г.10 – Спливаюче вікно з інформацією про операцію.

На сторінці «Працівники» користувач може додавати та видаляти працівників. На цій сторінці вгорі зліва буде кнопка «Додати» нижче у табличній формі будуть відображатися всі працівники компанії, де справа від кожного запису буде по дві кнопки: «Переглянути», «Видалити», представлено на рисунку Г.11.

Працівники			
Додати			
Ім'я	Email	Телефон	Дії
Максим	test@gmail.com	+380505004115	Переглянути Видалити

Рисунок Г.11 – Сторінка працівників

При натисканні «Додати» на екрані з'явиться спливаюче вікно з полем email, для пошуку користувача у системі, як користувач email він повинен натиснути кнопку «Додати», щоб додати працівника, або «Скасувати», для скасування додавання працівника. При натисканні на кнопку «Переглянути», перед користувачем відкривається спливаюче вікно з інформацією про працівника, а знизу спливаючого вікна будуть кнопки: «Скасувати», «Видалити», представлено на рисунку Г.12.

Інформація про працівника

Ім'я: Максим

Email: test@gmail.com

Телефон: +380505004115

Видалити

Скасувати

Рисунок Г.12 – Спливаюче вікно з інформацією про працівника.

При натисканні на кнопку «Видалити», у відповідного працівника буде вилучено доступ роботи з компанією та її записами.

3.3 Завершення роботи програми

Для завершення роботи з вебзастосунком користувач повинен натиснути «Вихід» у верхньому правому кінці екрану, користувача спрямує на сторінку авторизації.

Додаток Д – Код програмного забезпечення

Лістинг 1 – User.java

```
package org.dzhurynskyi.diploma.model;

import jakarta.persistence.*;
import lombok.*;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;

import java.util.Collection;
import java.util.List;

@Entity
@AllArgsConstructor
@NoArgsConstructor
@Getter
@Setter
@ToString
@Table(name = "users")
public class User implements UserDetails {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String email;
    private String phone;
    private String password;

    @OneToMany(mappedBy = "user", cascade = CascadeType.ALL, orphanRemoval = true)
    private List<Role> roles;

    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        return roles;
    }

    @Override
    public String getUsername() {
```

```

        return email;
    }

    @Override
    public boolean isAccountNonExpired() {
        return UserDetails.super.isAccountNonExpired();
    }

    @Override
    public boolean isAccountNonLocked() {
        return UserDetails.super.isAccountNonLocked();
    }

    @Override
    public boolean isCredentialsNonExpired() {
        return UserDetails.super.isCredentialsNonExpired();
    }

    @Override
    public boolean isEnabled() {
        return UserDetails.super.isEnabled();
    }
}

```

ЛІСТИНГ 2 – Role.java

```

package org.dzhurynskyi.diploma.model;

import jakarta.persistence.*;
import lombok.*;
import org.springframework.security.core.GrantedAuthority;

@Entity
@AllArgsConstructor
@NoArgsConstructor
@Getter
@Setter
public class Role implements GrantedAuthority {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
}

```

```

@ManyToOne(fetch = FetchType.LAZY)
private User user;

@ManyToOne(fetch = FetchType.LAZY) // Прив'язка ролі до компанії
private Company company;

@Override
public String getAuthority() {
    return (company != null && company.getId() != null) ? name + "_" +
company.getId() : name;
}
}

```

Лістинг 3 – CounterPartyController.java

```

package org.dzhurynskyi.diploma.controller;

import org.dzhurynskyi.diploma.dto.CounterPartyDTO;
import org.dzhurynskyi.diploma.service.CompanyService;
import org.dzhurynskyi.diploma.service.CounterPartyService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
public class CounterPartyController {

    @Autowired
    private CounterPartyService counterPartyService;

    @Autowired
    private CompanyService companyService;

    @GetMapping("/company/{id}/counterparties")
    public String counterparties(@PathVariable Long id, Model model) {
        model.addAttribute("company", companyService.findById(id));
        model.addAttribute("counterParties",
counterPartyService.getCounterPartiesByCompanyId(id));
    }
}

```

```

        return "counterparties";
    }

    @PostMapping("/company/{id}/counterparty/create")
    public String createCounterParty(@PathVariable Long id, CounterPartyDTO
counterPartyDTO) {
        counterPartyService.createCounterParty(counterPartyDTO, id);
        return "redirect:/company/" + id + "/counterparties";
    }

    @PostMapping("/company/{cid}/counterparty/{id}/update")
    public String updateCounterParty(@PathVariable Long cid, @PathVariable
Long id, CounterPartyDTO counterPartyDTO) {
        counterPartyDTO.setId(id);
        counterPartyService.updateCounterParty(counterPartyDTO);
        return "redirect:/company/" + cid + "/counterparties";
    }

    @GetMapping("/company/{cid}/counterparty/{id}/delete")
    public String deleteCounterParty(@PathVariable Long cid, @PathVariable
Long id) {
        counterPartyService.deleteCounterParty(id);
        return "redirect:/company/" + cid + "/counterparties";
    }
}

```

Лістинг 4 – UserDetailsService.java

```

package org.dzhurynskyi.diploma.service;

import lombok.extern.slf4j.Slf4j;
import org.dzhurynskyi.diploma.model.User;
import org.dzhurynskyi.diploma.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;

@Slf4j
@Service
public class UserDetailsService implements UserDetailsService {

    @Autowired
    private UserRepository userRepository;
}

```

```

@Override
public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException {
    User user =
userRepository.findByEmailAndFetchRoles(username).orElseThrow(() -> new
UsernameNotFoundException("User not found"));
    log.info("User {} {} {} logged in", user.getEmail(), user.getPassword(),
user.getAuthorities().toString());
    return new
org.springframework.security.core.userdetails.User(user.getEmail(),
user.getPassword(), user.getAuthorities());
}
}

```

Лістинг 5 – index.html

```

<!DOCTYPE html>
<html lang="uk" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Рахунки</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <h2 class="text-center">Рахунки</h2>
    <button type="button" class="btn btn-success mb-3" data-bs-toggle="modal"
data-bs-target="#createBillModal">Створити новий рахунок</button>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Назва</th>
                <th>IBAN</th>
                <th>Банк</th>
                <th>Дії</th>
            </tr>
        </thead>
        <tbody>
            <tr th:each="bill : ${bills}">

```

```

        <td th:text="${bill.name}"></td>
        <td th:text="${bill.iban}"></td>
        <td th:text="${bill.bank.name}"></td>
        <td>
            <a
th:href="@{/company/{cid}/bill/{id}/statements(cid=${company.id},id=${bill.id})}"
class="btn btn-primary btn-sm">Перейти</a>
            <button type="button" class="btn btn-info btn-sm" data-bs-
toggle="modal" th:data-bs-target="'#viewBillModal' +
${bill.id}">Переглянути</button>
        </td>
    </tr>
</tbody>
</table>
</div>

<div class="modal fade" id="createBillModal" tabindex="-1" aria-
labelledby="createBillModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="createBillModalLabel">Створити
рахунок</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
            </div>
            <form th:action="@{/company/{id}/bill/create(id=${company.id})}"
method="post">
                <div class="modal-body">
                    <div class="mb-3">
                        <label for="billName" class="form-label">Назва</label>
                        <input type="text" class="form-control" id="billName"
name="name" required>
                    </div>
                    <div class="mb-3">
                        <label for="iban" class="form-label">IBAN</label>
                        <input type="text" class="form-control" id="iban"
name="iban" required>
                    </div>
                    <div class="mb-3">
                        <label for="bank" class="form-label">Банк</label>
                        <select class="form-select" id="bank" name="bank.id"
required>

```

```

        <option th:each="bank : ${banks}"
th:value="${bank.id}" th:text="${bank.name}"></option>
    </select>
</div>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Скасувати</button>
    <button type="submit" class="btn btn-
primary">Створити</button>
</div>
</form>
</div>
</div>
</div>

<div th:each="bill : ${bills}" class="modal fade" th:id="'viewBillModal' +
${bill.id}" tabindex="-1" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Редагувати рахунок</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
            </div>
            <form
th:action="@{/company/{cid}/bill/{id}/update(cid=${company.id},id=${bill.id})}"
method="post">
                <div class="modal-body">
                    <div class="mb-3">
                        <label for="billName" class="form-label">Назва</label>
                        <input type="text" class="form-control" id="billName"
name="name" th:value="${bill.name}" required>
                    </div>
                    <div class="mb-3">
                        <label for="iban" class="form-label">IBAN</label>
                        <input type="text" class="form-control" id="iban"
name="iban" th:value="${bill.iban}" required>
                    </div>
                    <div class="mb-3">
                        <label for="bank" class="form-label">Банк</label>
                        <select class="form-select" id="bank" name="bank.id"
required>

```

```

        <option th:each="bank : ${banks}"
th:value="${bank.id}" th:text="${bank.name}" th:selected="${bank.id ==
bill.bank.id}"></option>
    </select>
</div>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-danger"
th:onclick="'location.href=\'/company/' + ${company.id} + '/bill/' + ${bill.id} +
'/delete\''">Видалити</button>
    <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Скасувати</button>
    <button type="submit" class="btn btn-
primary">Зберегти</button>
</div>
</form>
</div>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
></script>
</body>
</html>

```

Додаток Е – Акт впровадження

ТЗОВ «ЮРИДИЧНА КОМПАНІЯ «ЛЕМБЕРГ»
консультування з питань оподаткування, комерційної діяльності й керування
Код ЄДРПОУ 39777293, поштова адреса: 79026, м. Львів-26, а/с 2157

А К Т

впровадження результатів кваліфікаційної роботи
на здобуття ступеня вищої освіти «Бакалавр»
на тему: Розробка вебзастосунку для фінансового аналізу дебіторів для
юридичної компанії «ЛЕМБЕРГ»
Джуринського Максима Олексійовича

Я, що підписалася нижче, директор ТОВ "Юридична компанія "ЛЕМБЕРГ" Гавронська Наталя Олексіївна, склала цей акт про те, що результати кваліфікаційної роботи Джуринського Максима Олексійовича, представленої на здобуття освітнього ступеня бакалавр зі спеціальності 121 "Інженерія програмного забезпечення", впроваджені в поточну діяльність ТОВ "Юридична компанія «ЛЕМБЕРГ»", а саме вебзастосунок для фінансового аналізу дебіторів.

Розроблений вебзастосунок дозволив автоматизувати аналіз банківських виписок та фінансових операцій компанії, ефективно відстежувати дебіторську заборгованість та виявляти боржників, формувати аналітичні звіти для керівництва компанії, покращити контроль над фінансовими потоками та зменшити ризики фінансових втрат.

директор

ТОВ "Юридична компанія "ЛЕМБЕРГ"



Наталя ГАВРОНСЬКА