

Міністерство освіти і науки України
Харківський національний університет імені В.Н. Каразіна
Факультет комп'ютерних наук

Рекомендовано до захисту:

протокол засідання кафедри

№ ____ від ____ . ____ .2025 р.

Завідувач кафедри _____

(підпис)

(прізвище та ініціали)

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему «Розробка прототипу SRM-системи для управління
постачальниками з використанням засобів аналітики для оцінки
ефективності співпраці»

Захищено на засіданні ЕК № _____

протокол № ____ від ____ . ____ .2021 р.

Оцінка ____ / _____

Голова ЕК

(підпис)

(прізвище та ініціали)

Виконала:

студентка 4 курсу, групи КС- 41

Спеціальності:

122 Комп'ютерні науки

(шифр і назва спеціальності підготовки)

Лобко Єва Віталіївна

(прізвище, ім'я та по батькові, підпис)

Науковий керівник ст. викл. Трусов М.А.

(ступень, звання, прізвище та ініціали, підпис)

Рецензент доцент, кандидат фізико-
математичних наук Поклонський Є.В.

(ступень, звання, прізвище та ініціали, підпис)

Харків – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту

Кафедра кібербезпеки інформаційних систем, мереж і технологій

Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 122 Комп'ютерні науки

Освітня програма: Комп'ютерні науки

ЗАТВЕРДЖУЮ

завідувач кафедри _____

Олег ОЛЕШКО

«____» _____ 20__ року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЕКТ)

Лобко Єва Віталіївна

(прізвище, ім'я, по батькові студента)

1. Тема роботи «Розробка прототипу SRM-системи для управління постачальниками з використанням засобів аналітики для оцінки ефективності співпраці»

керівник роботи викладач Трусів Михайло Андрійович,

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

затверджені наказом по університету від “16 ” квітня 2025 року № 4101-5/962

2. Строк подання студентом роботи 01.06.2025

3. Перелік питань, які потрібно розробити:

1. Опис об'єкта та предмета дослідження, формулювання мети й завдань дипломної роботи.
2. Аналіз сучасних SRM-систем та методів багатокритеріального відбору постачальників (огляд рішень та обґрунтування вибору WSM).
3. Формалізація функціональних та нефункціональних вимог до SRM-системи (збір і структуризація вимог на основі практики компанії).
4. Проектування архітектури клієнт-серверного рішення.
5. Розробка модулю багатокритеріального оцінювання (реалізація алгоритму зваженої суми для розрахунку підсумкового балу).
6. Розробка модуля генерації договору через OpenXML-шаблон (.docx).
7. Реалізація користувацького інтерфейсу (WPF).
8. Інтеграція модулів із базою даних.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Підготовка вступної частини: опис об'єкта, предмета, формулювання мети та завдань дослідження	05.05 – 08.05.2025
2	Аналіз існуючих SRM-систем та методів багатокритеріальної оцінки; обґрунтування вибору WSM	09.05 – 12.05.2025
3	Збір і формалізація функціональних, нефункціональних вимог (Requirement Specification)	13.05 – 15.05.2025
4	Проектування архітектури	16.05 – 24.05.2025
5	Реалізація модуля багатокритеріальної оцінки (Weighted Sum Model)	25.05 – 27.05.2025
6	Реалізація модуля генерації договору через OpenXML (.docx)	28.05 - 29.05.2025
7	Розробка користувацького інтерфейсу (WPF):	30.05 – 31.05.2025
8	Інтеграція з БД: реалізація CRUD для сутностей тендеру, критеріїв, пропозицій, договорів	30.05 – 31.05.2025

5. Дата видачі завдання 16.04.2025

Студент

Лобко Є.В.

ініціали, прізвище



підпис

Керівник роботи Трусов М.А.

ініціали, прізвище



підпис

АНОТАЦІЯ

Лобко Є.В. – Розробка прототипу SRM-системи для управління постачальниками з використанням засобів аналітики для оцінки ефективності співпраці – Рукопис.

Кваліфікаційна робота бакалавра за спеціальністю 122 «Комп'ютерні науки» – Харківський національний університет ім. В.Н. Каразіна, Харків, 2025.

42 с., 14 рис., 8 табл., 21 джерел.

У роботі представлено розробку прототипу srm-системи (supplier relationship management) для автоматизації тендерного відбору постачальників у компанії роздрібної торгівлі. Об'єктом дослідження є процес тендерного вибору постачальників, а предметом — інформаційна технологія, що підтримує прийняття рішень шляхом критеріально-бального аналізу пропозицій.

Для досягнення мети роботи було виконано аналіз існуючих srm-рішень і обрано метод зваженої суми (weighted sum model, wsm) для багатокритеріальної оцінки. Зібрано й формалізовано функціональні та нефункціональні вимоги на основі практичного кейсу компанії, що досліджується. Спроектовано архітектуру wpf-клієнта з використанням патерну mvvm, бази даних ms sql server та ef-схемою для збереження інформації про тендер, критерії, пропозиції та результати скорингу.

Реалізовано алгоритм розрахунку зваженого балу, що дозволяє уніфікувати підхід до вибору оптимального постачальника, і інтегровано механізм автогенерації договору за допомогою шаблонів openxml (.docx). Проведено функціональні випробування й навантажувальні тести, підтверджено коректність розрахунків і стабільність роботи системи за одночасної обробки до кількох сотень тендерних пропозицій.

За результатами тестового впровадження скорочено тривалість циклу тендеру на 55 %, а обсяг ручних операцій зменшено на 90 %. Отримана

система забезпечує підвищену прозорість, об'єктивність і ефективність закупівельного процесу та може бути масштабована шляхом інтеграції з електронними майданчиками, розширення ролей і впровадження olap-аналітики.

Ключові слова: SRM-система, WPF, MVVM, MS SQL SERVER, WSM, OpenXML, багатокритеріальна оцінка, тендерний відбір, постачальники, автоматизація договору.

ABSTRACT

Lobko Ye.V. – Development of a Prototype SRM System for Supplier Management Using Analytical Tools to Assess Collaboration Efficiency – Manuscript.

Bachelor's Qualification Work in the specialty 122 "Computer Science" – V. N. Karazin Kharkiv National University,
Kharkiv, 2025.

42 pages, 14 figures, 8 tables, 21 references.

This thesis presents the development of an SRM (Supplier Relationship Management) prototype for automating the tender-based supplier selection process in a retail company. The research object is the supplier tendering process, while the subject is an IT solution that supports decision-making via a score-based, multi-criteria evaluation of bids.

To achieve the goals, existing SRM solutions were analyzed and the Weighted Sum Model (WSM) was chosen for multi-criteria scoring. Functional and non-functional requirements were gathered and formalized based on the practical scenario of the company under study. The solution's architecture was designed as a WPF client using the MVVM pattern, with MS SQL Server for data storage (Tender, Criteria, Bids, Scoring Results) and an ER diagram.

A weighted scoring algorithm was implemented to standardize the supplier ranking process, as well as an automatic contract generation module using OpenXML (.docx templates). Functional and performance tests confirmed the correctness of calculations and system stability under concurrent processing of up to hundreds of bids.

During pilot deployment, the tender cycle duration was reduced by 55 %, and manual tasks decreased by 90 %. The final system ensures greater transparency, objectivity, and efficiency of procurement, and can be further expanded by

integrating with e-procurement platforms, adding user roles, and implementing OLAP analytics.

Keywords:SRM system, WPF, MVVM, MS SQL Server, WSM, OpenXML, multi-criteria evaluation, tender selection, suppliers, contract automation.

ЗМІСТ

АНОТАЦІЯ	5
ABSTRACT	7
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	11
ВСТУП	13
1. АНАЛІТИЧНИЙ ОГЛЯД СУЧАСНИХ SRM-СИСТЕМ ТА МЕТОДІВ ОЦІНКИ ПОСТАЧАЛЬНИКІВ	15
1.1 Сучасний стан та тенденції розвитку SRM у роздрібній торгівлі...	15
1.2 Аналіз сучасних SRM-платформ та обґрунтування вибору методики оцінювання	21
1.2.1 Огляд сучасних SRM-систем для управління постачальниками	21
1.3 Порівняння методів багатокритеріальної оцінки постачальників (WSM, АНР, TOPSIS).....	26
1.4 Обґрунтування вибору методу Weighted Score Model (WSM)	29
2. Проектування SRM-системи для вибору постачальників	32
2.1 Постановка задачі	32
2.2 Визначення вимог до системи	32
2.3 Архітектура рішення	39
2.4 Структура бази даних.....	43
2.5 Алгоритм скорингу постачальників	45
2.6 Генерація договору з переможцем.....	47
3. Реалізація прототипу SRM-системи.....	50
3.2 Інтерфейс користувача	56
3.2.1 Загальний вигляд головного вікна.....	56
3.2.2 Форма створення і редагування тендеру	57
3.2.3 Форма подання пропозиції.....	59
3.2.4 Екран оцінювання і вибору переможця	60

3.2.5 Зв'язок MVVM: View–ViewModel	61
3.3 Результати експлуатації прототипу	62
ВИСНОВКИ	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	67
ДОДАТОК А.....	70
ДОДАТОК Б	71
Приклад згенерованого договору з постачальником	71
ДОДАТОК В.....	73
ДОДАТОК Г	76

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

SRM — Supplier Relationship Management, Система управління взаємовідносинами з постачальниками; платформа для автоматизації тендерного відбору постачальників.

WPF— Windows Presentation FoundationТехнологія Microsoft для створення десктопних додатків із багатим графічним інтерфейсом на платформі .NET.

MVVM— Model–View–ViewModel, Архітектурний патерн для розділення UI (View), даних (Model) і логіки представлення (ViewModel) у WPF-додатках.

EF Core— Entity Framework Core, ORM-бібліотека Microsoft для роботи з реляційними базами даних (MS SQL Server) у .NET, яка забезпечує відображення класів на таблиці.

SQL —Structured Query Language, Мова структурованих запитів для роботи з реляційними базами даних (у цьому проєкті — MS SQL Server).

KPI—Key Performance Indicator, Ключовий показник ефективності, який використовується для оцінки результативності бізнес-процесів (наприклад, тривалість тендерного циклу).

OLAP— Online Analytical Processing, Технологія для аналізу даних та побудови складних звітів; використовується для зведеної аналітики закупівель.

DB— Database, База даних, сховище структурованих даних (у проєкті — MS SQL Server).

DTO— Data Transfer Object, Об'єкт перенесення даних між шарами програми (зазвичай містить лише дані для передачі).

API — Application Programming Interface, Інтерфейс програмування, набір методів і протоколів для взаємодії різних програмних компонентів.

DOCX— Microsoft Word Open XML, Формат документів Microsoft Word (ZIP-архів з XML-файлами). Використовується для шаблонів договорів у проекті.

PDF— Portable Document Format, Універсальний формат для зберігання і відображення текстово-графічних документів.

xUnit — xUnit.net, Фреймворк для юніт-тестування у .NET.

ORM— Object-Relational Mapping, Підхід до відображення об'єктно-орієнтованих класів на реляційні таблиці бази даних.

UI — User Interface, Користувацький інтерфейс, сукупність елементів взаємодії користувача з програмою.

UX— User Experience, Досвід (враження) користувача від використання інтерфейсу; передбачає зручність і логіку роботи.

KPI — Key Performance IndicatorКлючовий показник ефективності процесу чи системи (для закупівель — середній час циклу, кількість пропозицій тощо).

PK— Primary Key, Первинний ключ у таблиці бази даних, що однозначно ідентифікує кожний запис.

FK— Foreign Key, Зовнішній ключ у таблиці бази даних, що встановлює зв'язок із записом у іншій таблиці.

ВСТУП

У сучасному роздрібному бізнесі витрати на постачання становлять до 70 % собівартості товару, тому вибір контрагента прямо впливає на рентабельність мережі і надійність ланцюга постачання. Здебільшого тендерні рішення приймаються в Excel [1] та через електронну пошту, що значно затягує цикл відбору постачальника, не гарантує прозорість логіки оцінювання й ускладнює аудит усіх змін. Через це компанії стикаються з ризиками суб'єктивних рішень та втратами в ефективності.

Рішення цих проблем досягається впровадженням SRM-системи (Supplier Relationship Management) із вбудованим багатокритеріальним алгоритмом оцінки. Така система об'єднує всі етапи тендеру: від формування потреби й публікації запиту до автоматичного розрахунку підсумкових балів за методом зваженої суми (Weighted Sum Model) і генерації договору. Завдяки єдиній базі даних і прозорому журналу змін забезпечується об'єктивність відбору, скорочення часу на обробку заявок і можливість повноцінного аудиту кожного кроку.

У дипломній роботі розроблено прототип SRM-системи на базі WPF із патерном MVVM та СУБД MS SQL Server. Впроваджено модуль багатокритеріальної оцінки пропозицій, механізм автогенерації договорів за OpenXML-шаблонами, а також інтерфейс для менеджерів і постачальників. Функціональні та навантажувальні тести підтвердили скорочення тривалості тендерного циклу понад удвічі та зменшення обсягу ручної роботи майже на 90 %. Отримане рішення підвищує прозорість і ефективність закупівельного процесу та може бути масштабоване в майбутньому, зокрема через інтеграцію з електронними майданчиками й OLAP-аналітику.

Об'єкт дослідження — процес тендерного вибору постачальників у закупівельному підрозділі роздрібною компанією.

Предмет дослідження — інформаційна технологія підтримки прийняття рішень, що реалізує критеріально-бальний аналіз пропозицій постачальників.

Мета роботи — розробити прототип SRM-системи, який

- формалізує вимоги до постачальників;
- розраховує зважений сумарний бал кожного учасника;
- автоматично пропонує оптимального переможця
- формує текст договору.

1. АНАЛІТИЧНИЙ ОГЛЯД СУЧАСНИХ SRM-СИСТЕМ ТА МЕТОДІВ ОЦІНКИ ПОСТАЧАЛЬНИКІВ

1.1 Сучасний стан та тенденції розвитку SRM у роздрібній торгівлі

Управління взаємовідносинами з постачальниками, або Supplier Relationship Management (SRM), є концепцією та програмними рішеннями для оптимізації співпраці між компанією і її постачальниками. SRM-системи дозволяють централізовано зберігати дані про постачальників, сприяють прозорій комунікації, моніторингу показників та ефективній координації в процесі закупівель. По суті, впровадження SRM покликане зміцнити взаємовигідні відносини з постачальниками, підвищити операційну ефективність і оптимізувати процеси постачання товарів у роздрібній торгівлі. Для ритейлу, який працює з широким колом постачальників, належне управління цими відносинами є критично важливим – від нього залежить своєчасність поповнення асортименту, якість продукції та конкурентоспроможність торговельної мережі.

Незважаючи на очевидні переваги SRM, нині багато роздрібних компаній усе ще здійснюють тендерні закупівлі та вибір постачальників вручну – за допомогою Excel-таблиць або електронної пошти. Подібні підходи формують слабкий фундамент для сучасних цифрових закупівель і призводять до низки проблем. Згідно з дослідженнями, у середніх компаніях Excel залишається основним інструментом управління закупівлями, випереджаючи навіть ERP-системи. Така залежність від електронних таблиць є ризикованою: ручне введення даних та формул неминуче породжує помилки і неточності, що негативно впливає на рішення щодо закупівель. Крім того, мануальні процеси в Excel уповільнюють роботу та знижують продуктивність: постійне оновлення файлів, зведення даних із численних листів і листування з постачальниками забирає багато часу. Excel не підтримує повноцінної спільної роботи і не забезпечує належної

прозорості, через що легко виникають непорозуміння, зриваються строки та у підсумку приймаються помилкові або суб'єктивні рішення. Фактично, використання електронних таблиць та листування електронною поштою *не гарантує об'єктивності* – відсутність єдиних критеріїв оцінки пропозицій постачальників залишає простір для людського фактора і упередженості при виборі переможця тендера.

Сформульована проблема очевидна: в умовах ручного проведення тендерів виникає *суб'єктивний вибір постачальника*, коли рішення залежать від особистих думок чи помилок, а не від чітких показників. Це ставить під загрозу справедливість і ефективність закупівельного процесу [2].

Актуальність автоматизації цього процесу зумовлена потребою усунути згадані недоліки. Впровадження спеціалізованої SRM-системи для тендерів дозволяє стандартизувати і формалізувати оцінювання пропозицій. Експерти з закупівель наголошують, що для прозорості необхідно заздалегідь визначити чіткі критерії відбору, їх вагові коефіцієнти та методику бального оцінювання – це забезпечує послідовність, неупередженість і підзвітність під час аналізу тендерних заявок. Іншими словами, запровадження формальної системи оцінки робить рішення більш об'єктивними та обґрунтованими даними, а не інтуїцією чи «ручним» вибором. Саме тому автоматизація SRM-процесу наразі є на часі: вона підвищує ефективність роботи (шляхом усунення ручної рутини), мінімізує помилки і корупційні ризики, а також сприяє вибудові довгострокових відносин із надійними постачальниками на основі фактів і метрик.

Одним із ключових інструментів, який використовується для об'єктивного вибору постачальника, є метод зважених оцінок (Weighted Score Model). Цей підхід передбачає визначення набору критеріїв (наприклад, ціна, якість товару, надійність постачань, умови оплати тощо) та присвоєння їм вагових коефіцієнтів залежно від важливості для компанії.

Далі кожен постачальник оцінюється за цими критеріями, і обчислюється сумарний *рейтинг* – зважена сума балів за всіма показниками. Використання такої критеріально-бальної моделі оцінювання забезпечує порівняння постачальників за єдиною шкалою та ранжування їх від кращого до гіршого. Важливо, що до системи критеріїв можуть входити як кількісні, так і якісні показники – завдяки ваговому коефіцієнту якісним аспектам також надається числова оцінка, що дозволяє врахувати їх у загальному рейтингу. Таким чином, метод зважених оцінок дає змогу провести комплексну та об'єктивну оцінку всіх пропозицій постачальників і обґрунтовано рекомендувати переможця тендера. У практиці закупівель цей підхід реалізується через *оціночні карти постачальників* (supplier scorecards), які містять ключові показники ефективності та автоматично розраховують підсумковий бал – це створює прозору, даними-керовану основу для вибору постачальника та узгоджує рішення з бізнес-цілями компанії. Отже, у випадку компаній, що досі проводять тендери «вручну», впровадження SRM-системи з методикою зваженого оцінювання є обґрунтованим кроком для підвищення об'єктивності й ефективності процесу.

Останні тенденції 2023–2025 років у сфері SRM демонструють стрімкий перехід від фрагментарних ручних підходів до *цілісних цифрових платформ* та стратегічного управління постачальниками. Найважливіші глобальні тренди можна окреслити так:

Цифровізація та автоматизація SRM. Організації активно впроваджують сучасні програмні рішення для управління постачальниками, замінюючи застарілі Excel- та e-mail-процеси. За даними досліджень, наразі лише близько 28% *закупівельних команд* використовують спеціалізовані SRM-системи для роботи з постачальниками – інші все ще покладаються на ERP, пошту чи таблиці. Однак інвестиції в цифрові платформи зростають: глобальний ринок SRM-софтверу оцінювався у \$13,6 млрд у 2023 році і, за

прогнозами, до 2032 року майже потроїться до \$35,9 млрд (середньорічні темпи зростання понад 11% на рік). Спеціалізовані портали постачальників та SRM-платформи автоматизують рутинні операції і спрощують комунікації з постачальниками (обмін даними, подання пропозицій, звітність тощо), що помітно підвищує швидкість і прозорість закупівельного процесу. Перехід на цифрові SRM-рішення в роздрібній торгівлі дозволяє краще контролювати ланцюги постачання товарів і оперативно приймати рішення на основі актуальних даних.

Поглиблення стратегічного партнерства з постачальниками.

Сучасні ритейлери все більше усвідомлюють цінність довгострокових партнерських відносин у ланцюгу постачання. Досвід пандемії COVID-19 наочно показав, що компанії з більш тісними відносинами з ключовими постачальниками легше долають кризові перебої, тоді як суто транзакційні моделі взаємодії виявились вразливими. Відтоді відбувся зсув у бік співробітництва та взаємної вигоди: понад 75% керівників закупівель у світі заявили, що інвестують у глибшу співпрацю з постачальниками замість гонити лише за ціною. У роздрібному секторі це проявляється через укладання довгострокових контрактів і програм розвитку постачальників. Наприклад, крупні торговельні мережі (як-от Tesco Ireland) переходять до багаторічних угод з основними постачальниками, гарантуючи їм стабільність замовлень для інвестицій у розвиток, а собі – надійність і безперервність постачань. Такий підхід взаємної довіри і підтримки приносить вигоду обом сторонам: постачальники отримують стимул покращувати показники, а ритейлери – більш стійкий ланцюг постачання та лояльність партнерів на випадок майбутніх потрясінь.

Пріоритет сталого розвитку та критерії ESG. За останні роки сталий розвиток перетворився на один з центральних аспектів у відносинах з постачальниками. Більше 70% директорів із закупівель (CPO) нині

відзначають, що екологічні та соціальні показники постачальників є пріоритетними при виборі партнерів. Регуляторні органи, інвестори й споживачі все сильніше тиснуть на бізнес щодо дотримання принципів сталості – від скорочення вуглецевого сліду до етики праці. У відповідь роздрібні компанії впроваджують політики *sustainable procurement*, інтегруючи в SRM-критерії такі показники, як екологічна відповідальність, умови праці, наявність сертифікацій тощо. Постачальники, які відповідають вимогам ESG, отримують перевагу при відборі. Таким чином, SRM у 2023–2025 рр. все більше орієнтується не лише на ціну і якість, а й на відповідність партнерів принципам сталого розвитку. Це не тільки покращує імідж ритейлера і знижує ризики репутаційних втрат, але й сприяє довгостроковій стійкості бізнесу за рахунок відповідального ведення поставань.

Управління ризиками та стійкість ланцюгів постачання.

Геополітична нестабільність, пандемії та інші глобальні виклики актуалізували проблему ризиків у відносинах з постачальниками. Одна з провідних тенденцій – перехід до *проактивного менеджменту ризиків* в SRM. Компанії впроваджують безперервний моніторинг ризиків постачальників за допомогою спеціалізованих аналітичних інструментів: відстежуються фінансовий стан контрагентів, виконання ними договірних зобов'язань, ситуація в регіонах, де вони оперують, тощо. Сучасні SRM-платформи можуть автоматично сповіщати про *тривожні сигнали* – наприклад, про погіршення показників постачальника або політичні ризики в країні його діяльності. Водночас компанії диверсифікують базу постачальників, щоб уникнути залежності від одного джерела, та готують плани дій на випадок збоїв (альтернативні постачальники, страхові запаси). У партнерстві з ключовими постачальниками все частіше обговорюються *спільні плани зниження ризиків*, щоб обидві сторони були готові до непередбачуваних ситуацій. Загалом, тенденція така, що ризик-менеджмент

стає невід’ємною частиною SRM: метою є підвищення стійкості ланцюга постачання і забезпечення безперервності бізнесу за будь-яких умов.

Використання AI та аналітики у SRM. Новітні технології штучного інтелекту дедалі глибше інтегруються у процеси управління постачальниками. У 2023–2025 роках провідні компанії експериментують з AI-інструментами для автоматизації аналізу великих обсягів даних про постачальників. Алгоритми машинного навчання вже застосовуються для прискореного пошуку та оцінки нових постачальників, прогнозування потенційних збоїв або несумлінної поведінки, а також для відстеження ключових метрик продуктивності в режимі реального часу. Наприклад, сучасні SRM-платформи з елементами AI можуть *ранжувати постачальників* за множиною параметрів і навіть пропонувати оптимального кандидата для співпраці на основі історичних даних і заданих критеріїв. Такі системи допомагають виявляти приховані тенденції (скажімо, зниження якості чи затримки поставок), завчасно попереджаючи менеджмент про необхідність втручання. Хоча технології ШІ ще розвиваються і впроваджуються обережно, вже сьогодні вони доводять свою корисність: автоматизація рутинних задач (наприклад, звірка рахунків чи перевірка відповідності постачальника вимогам) вивільняє час фахівців для стратегічної роботи. Отже, тенденція на стику SRM та AI набирає обертів – очікується, що з удосконаленням штучного інтелекту його можливості в сфері управління постачальниками лише розширюватимуться, відкриваючи нові горизонти для підвищення ефективності та прозорості в роздрібних ланцюгах постачання.

Підсумовуючи, сучасний стан SRM у роздрібній торгівлі характеризується поступовим переходом від ручних, суб’єктивних методів до автоматизованих, аналітично обґрунтованих підходів у виборі та управлінні постачальниками. Автоматизація тендерних процедур на основі

моделей зваженого оцінювання та впровадження новітніх SRM-рішень дають змогу мінімізувати вплив людського фактора і підвищити ефективність закупівель. Водночас нові тренди – цифрові платформи, стратегічне партнерство, сталий розвиток, проактивний ризик-менеджмент і AI – визначають вектор розвитку SRM на найближчу перспективу. Компанії роздрібного сектору, які вже сьогодні інтегрують ці підходи, отримують конкурентні переваги у вигляді більш прозорих та надійних ланцюгів постачання, що особливо актуально в умовах сучасних викликів і динамічного ринку.

1.2 Аналіз сучасних SRM-платформ та обґрунтування вибору методики оцінювання

Сучасні SRM-платформи (Supplier Relationship Management) є програмними рішеннями для автоматизації процесів взаємодії з постачальниками. Вони забезпечують централізоване управління закупівлями, оцінюванням постачальників, контролем виконання договорів та аналізом ефективності постачань. У сфері роздрібної торгівлі використання SRM-систем допомагає працювати з великою кількістю постачальників, оптимізувати витрати та підвищити надійність ланцюгів поставок. Нижче проведено огляд п'яти популярних SRM-рішень: **SAP Ariba, Coupa, Oracle Procurement Cloud, Microsoft Dynamics 365 SCM та 1C: Управління закупками** – із зазначенням їх моделей ліцензування, методик оцінювання постачальників, а також ключових переваг і недоліків у контексті роздрібної торгівлі.

1.2.1 Огляд сучасних SRM-систем для управління постачальниками

SAP Ariba. SAP Ariba – це одна з найвідоміших хмарних платформ для закупівель та управління постачальниками. Ліцензування здійснюється за моделлю SaaS (Software-as-a-Service) із платою за підписку залежно від

кількості користувачів та обраних модулів. Платформа містить модуль Supplier Lifecycle & Performance, який підтримує реєстрацію постачальників, аналіз ризиків та scorecards для оцінювання ефективності постачальників за ключовими показниками (KPI). Для роздрібних компаній цінною є інтеграція SAP Ariba з іншими продуктами SAP та доступ до глобальної мережі постачальників Ariba Network, що спрощує пошук нових джерел товарів. *Переваги:* широка функціональність (від стратегічного сорсингу до оплат), глибока аналітика та контроль ризиків, підтримка галузевих шаблонів KPI для оцінки якості та стійкості поставок. *Недоліки:* висока вартість впровадження і підписки (для малого бізнесу це може бути суттєвим бар'єром) та значна складність налаштування. Користувачі відзначають довгий період навчання роботи з системою і не надто зручний інтерфейс мобільного додатку, що вимагає часу на адаптацію персоналу. Таким чином, SAP Ariba більше підходить великим ритейлерам, готовим інвестувати у комплексну інтегровану платформу.

Coupa. Coupa – інша провідна хмарна платформа, що надає рішення для комплексного управління витратами бізнесу (Business Spend Management). Ліцензійна модель Coupa – підписка на хмарний сервіс; для менших компаній вартість може бути значною. Система охоплює весь цикл Procure-to-Pay, включаючи модулі закупівель, управління рахунками і витратами, а також Supplier Management – від онбордингу постачальників до відстеження виконання контрактів. Оцінювання постачальників у Coupa здійснюється через відстеження їхніх показників (якість, вчасність поставок, ціни тощо) та співпрацю в межах єдиного інтерфейсу; передбачено аналітичні засоби для порівняння постачальників за витратами та ефективністю. *Переваги:* гнучкість та налаштовуваність платформи під потреби різних організацій, потужна аналітика і звітність, відносно простий інтерфейс, що підвищує залученість користувачів. У роздрібній торгівлі це означає швидше впровадження і легшу адаптацію персоналу при роботі з

великою кількістю постачальників та транзакцій.. *Недоліки:* складність інтеграції з наявними системами обліку і ERP може потребувати додаткових ресурсів, а багатий функціонал зумовлює певну комплексність – користувачам потрібен час, щоб повністю опанувати всі можливості системи. Вартість підписки також залишається високою для малого бізнесу. Крім того, аналітики відзначають порівняно обмежені можливості управління постачальниками та меншу власну мережу постачальників у Coupa (в порівнянні з SAP Ariba), що може бути суттєвим для глобальних рітейлерів. В цілому, Coupa добре підходить середнім та великим роздрібним компаніям, які цінують зручність користування і широку функціональність у сфері контролю витрат.

Oracle Procurement Cloud. Oracle Procurement Cloud – це компонент хмарного ERP-рішення Oracle, призначений для автоматизації закупівель та управління постачальниками. Платформа постачається за моделлю підписки (Oracle Cloud SaaS) у складі пакету Oracle Fusion Applications. Функціонально система включає інструменти для стратегічного сорсингу, ведення Supplier Qualification (кваліфікації постачальників), управління контрактами і витратами, а також розширену аналітику закупівель. Оцінювання постачальників в Oracle Cloud реалізується через модулі Supplier Performance – компанія може визначати метрики та рейтингувати постачальників за допомогою вбудованих засобів аналізу (наприклад, інтеграції з зовнішніми базами даних ризиків чи сталого розвитку). *Переваги:* глибока інтеграція з іншими системами Oracle (ERP, фінансовий модуль), що важливо для рітейлерів, які вже користуються екосистемою Oracle; наявність передових можливостей у керуванні постачальниками, контрактах і аналізі витрат. Система орієнтована на великий бізнес і може підтримувати складні процеси узгодження закупівель у роздрібних мережах з багатьма філіями. *Недоліки:* висока вартість володіння – рішення Oracle традиційно недешеві у впровадженні та підтримці; технічна складність, яка

зумовлює значну залежність від ІТ-фахівців при налаштуванні і інтеграції. Інтерфейс користувача вважається менш інтуїтивним порівняно з більш легкими рішеннями, тому навчання персоналу може зайняти додатковий час. Для середнього і малого роздробу Oracle Procurement Cloud може виявитись занадто громіздким, тоді як для великих мереж зі складною інфраструктурою він забезпечує необхідну масштабованість і надійність.

Microsoft Dynamics 365 SCM. Microsoft Dynamics 365 Supply Chain Management (SCM) – це ERP-рішення від Microsoft, що включає модуль управління закупками та постачальниками. Ліцензування здійснюється за підпискою на користувача (хмарна версія Dynamics 365) або у вигляді локального розгортання для великих підприємств. Система дозволяє автоматизувати процеси закупівель: формування заявок, замовлень, ведення номенклатури та довідників постачальників, контроль виконання поставок і оцінювання роботи постачальників. Оцінювання постачальників у Dynamics 365 SCM реалізується через функції Vendor evaluation у модулі Procurement and Sourcing – користувач може відстежувати такі показники, як вчасність поставки, відповідність замовленням, якість товарів тощо, і на основі цього присвоювати постачальникам рейтинги. Переваги: інтеграція з екосистемою Microsoft – Dynamics 365 легко поєднується з Office 365, Power BI та іншими сервісами, що дає змогу ритейлерам мати єдиний зв'язний ІТ-ландшафт (від CRM до касових систем); гнучкість і масштабованість, оскільки додавання нових користувачів або функцій відбувається через хмарні служби Microsoft; нижчі початкові витрати на розгортання, ніж у традиційних on-premise ERP, і вбудована аналітика на базі Microsoft Power Platform. З точки зору користувача, інтерфейс Dynamics 365 є відносно дружнім для тих, хто звик до продуктів Microsoft, що може прискорити освоєння системи співробітниками роздрібною компанією. *Недоліки:* деякі користувачі стикаються зі складнощами продуктивності та гнучкості налаштувань, якщо система впроваджена без достатнього доопрацювання під їхні процеси.

Відзначається також, що навчання роботі з Dynamics 365 може бути неглибоким на початку, але для повного використання можливостей (особливо аналітичних звітів, кастомізації під бізнес-процеси) потрібен час і кваліфіковані консультанти. У порівнянні зі спеціалізованими SRM-рішеннями, модуль постачальників у Dynamics 365 може мати менше галузевих «фішок» (наприклад, немає такої розвиненої мережі сторонніх постачальників чи вбудованого управління ризиками, як у Ariba). Проте для середніх роздрібних компаній, які вже працюють в середовищі Microsoft, Dynamics 365 SCM є привабливим через свою прозорість впровадження та підтримки у знайомій IT-інфраструктурі.

1С: Управління закупками. «1С: Управління закупками» – це вітчизняне рішення для автоматизації закупівель на платформі 1С:Підприємство. Зазвичай воно впроваджується як підсистема в складі комплексних продуктів на кшталт 1С:Управління холдингом або 1С:ERP, що дозволяє інтегрувати SRM-функції з бухгалтерським обліком, складським управлінням та іншими модулями. Ліцензійна модель 1С передбачає придбання програмної ліцензії та необхідної кількості клієнтських доступів (робочих місць). Система покриває основні етапи закупівельного циклу: планування потреб, збір заявок від внутрішніх підрозділів, проведення тендерів і конкурсів серед постачальників, укладання контрактів, контроль виконання поставок та облік взаєморозрахунків. Для оцінювання постачальників у 1С можна використовувати вбудовані звіти і довідники: напряду відстежувати виконання договорів, своєчасність поставок, дотримання погоджених цін тощо, формуючи рейтинги постачальників на основі цих даних. Деякі впровадження 1С також пропонують *«кабінет постачальника»* – веб-портал, де постачальники можуть подавати пропозиції, що спрощує збір інформації для оцінки їх конкурентоспроможності. *Переваги:* порівняно низька вартість володіння та гнучкі можливості кастомізації під специфічні

вимоги локального бізнесу (існує розгалужена мережа інтеграторів і партнерів 1С, які можуть доопрацьовувати систему під потреби замовника). Рішення 1С добре враховує місцеві регуляторні вимоги і стандарти електронного документообігу, що важливо для українських і загалом східноєвропейських ритейлерів (підтримується, зокрема, обмін даними з бухгалтерією, податковий облік, робота з майданчиками публічних закупівель). Крім того, 1С:Управління закупками можна інтегрувати в єдиний ланцюжок поставок підприємства – наприклад, у зв'язці з модулями управління запасами, логістикою, WMS, MDM тощо, утворюючи *модульну платформу для SCM* локального виробництва. *Недоліки:* вузька галузева спеціалізація і регіональність – на відміну від глобальних продуктів, 1С орієнтована переважно на країни СНД, тому для міжнародних роздрібних компаній можуть виникати обмеження щодо багатомовності, підтримки різних валют чи інтеграції з глобальними постачальниками. Також, щоб отримати сквозну аналітику та повний контроль над поставками, часто потрібна інтеграція з іншими системами (складським обліком, системами відстеження доставок тощо). Інтерфейс 1С є менш сучасним у порівнянні з західними аналогами, що потенційно знижує зручність роботи користувачів, хоча знайомство багатьох українських бухгалтерів і логістів з 1С частково компенсує цей недолік. В цілому, «1С: Управління закупками» – привабливе рішення для невеликих та середніх ритейлерів локального ринку, які прагнуть автоматизувати закупівлі з мінімальними витратами та інтегрувати їх з існуючим обліком.

1.3 Порівняння методів багатокритеріальної оцінки постачальників (WSM, AHP, TOPSIS)

Для об'єктивного оцінювання постачальників у SRM-системах застосовуються різні методи багатокритеріального аналізу. Вибір методу залежить від доступних даних, кількості критеріїв та вимог до прозорості процесу прийняття рішення. Розглянемо основні підходи:.

- Метод зважених оцінок (WSM).** Це найбільш простий і поширений підхід, відомий також як метод простої адитивної зваженої суми. Його сутність полягає в тому, що для кожного постачальника підраховується підсумковий бал як сума його оцінок за різними критеріями, помножених на ваги цих критеріїв. Наприклад, якщо критерії – це ціна, якість і вчасність доставки, то кожному з них призначається певна вага (відповідно до важливості), після чого постачальнику обчислюють зважений бал. WSM передбачає, що всі критерії незалежні та виміряні в порівнянних одиницях, а також не враховує взаємодії між альтернативами. Попри ці спрощення, метод зважених оцінок є надзвичайно популярним завдяки зрозумілості та легкості реалізації. Він добре підходить у ситуаціях, коли кількість критеріїв порівняно невелика і потрібно швидко та прозоро порівняти декілька постачальників.
- Метод аналізу ієрархій (АНР).** Метод аналізу ієрархій, запропонований Т. Сааті, дозволяє структурувати задачу вибору постачальника у вигляді ієрархії «мета – критерії – альтернативи». Ключова особливість АНР – парне порівняння елементів. Експерти послідовно порівнюють критерії між собою (визначаючи, наскільки один важливіший за інший), а потім – постачальників за кожним критерієм. На основі цих парних порівнянь обчислюються локальні ваги критеріїв та оцінки альтернатив. Метод враховує *суб'єктивні чинники* і дозволяє узгодити пріоритети, зокрема корисний при оцінюванні якісних критеріїв (таких як репутація, гнучкість співпраці тощо). Результатом АНР є глобальні пріоритети (ваги) для кожного постачальника, які відображають їхній відносний ранг. Перевагою є можливість перевірки узгодженості експертних оцінок (обчислення індексу узгодженості), що додає об'єктивності в процес. Недоліки – трудомісткість: кількість порівнянь зростає квадратично зі

збільшенням числа критеріїв та альтернатив, тож для великої бази постачальників застосування АНР може бути складним. В практиці SRM АНР часто використовують для визначення ваг критеріїв, а не для прямого ранжування сотень постачальників, тобто комбінують: спочатку АНР задає ваги, а далі здійснюється оцінка постачальників більш простим способом.

- **Метод TOPSIS.** TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) – метод ранжування альтернатив, заснований на відстані до умовно ідеального та найгіршого рішення. У випадку вибору постачальника це означає, що спочатку визначається гіпотетичний «ідеальний постачальник» (найкращі можливі значення за всіма критеріями) та «найгірший постачальник» (найгірші значення показників). Далі для кожного реального постачальника розраховується, наскільки близько він знаходиться до ідеалу і наскільки далеко від антирішення. За алгоритмом TOPSIS обирається постачальник, який має мінімальну відстань до позитивно-ідеального рішення і максимальну відстань до негативно-ідеального. Практично це реалізується через нормування оцінок постачальників за всіма критеріями, визначення векторів найкращих та найгірших значень показників, обчислення евклідової відстані від кожного постачальника до цих двох еталонів та розрахунок відносної міри близькості до ідеалу. Метод TOPSIS зручний при великій кількості критеріїв, особливо кількісних, оскільки дозволяє отримати єдиний рейтинговий показник. Він часто використовується у наукових дослідженнях з вибору постачальників, оскільки враховує *усі* критерії та їхні ваги при формуванні рейтингу. Недоліком TOPSIS є те, що результати можуть бути чутливими до способу нормалізації даних та вибору ваг, а визначення ваг критеріїв перебуває поза самим методом (їх потрібно задати експертно або розрахувати іншими методами,

такими як АНР). У SRM-системах TOPSIS може застосовуватися для побудови інтегральних індексів надійності чи ризику постачальників, коли потрібно ранжувати десятки постачальників за комплексом показників.

Загалом, всі перелічені підходи застосовуються в SRM, оскільки вибір та оцінювання постачальників – це багатокритеріальна задача. Показники оцінки можуть включати цінові параметри, якість продукції, дотримання термінів поставки, фінансову стабільність постачальника, відповідність нормативним вимогам тощо. Формальні методи, такі як WSM, АНР, TOPSIS, дозволяють зробити процес оцінювання більш прозорим та обґрунтованим: вони забезпечують чітку математичну модель, яка знижує вплив суб'єктивних уподобань і підкріплює рішення кількісними розрахунками. У практичних SRM-платформах найчастіше імплементовано саме підхід зважених оцінок (scorecard), адже він інтуїтивно зрозумілий для користувачів і легко інтегрується у програмні системи у вигляді таблиць показників постачальника.

1.4 Обґрунтування вибору методу Weighted Score Model (WSM)

Для реалізації поставленої в дипломному проєкті задачі оцінювання постачальників було вирішено використати саме **метод зважених оцінок (WSM)**. Основні аргументи на користь цього вибору такі:

- **Простота реалізації.** WSM вирізняється найменшою складністю з точки зору алгоритмів: достатньо задати вагові коефіцієнти та провести розрахунок підсумкових балів шляхом множення і додавання. Цей метод є «найпростішим і найпрямолінійнішим» серед багатокритеріальних, не вимагає спеціалізованих математичних процедур чи великої кількості кроків. Для розробника це означає швидшу реалізацію в коді, а для користувача – зрозумілий інтерфейс налаштування критеріїв і ваг.

- **Прозорість і інтуїтивність.** Результати, отримані методом зважених оцінок, легко інтерпретувати: кожен постачальник одержує певний сумарний бал, а внесок окремих критеріїв видно з їх ваг. Така прозорість важлива у контексті пояснювальної записки та для довіри користувачів системи – менеджери з постачання можуть легко пояснити, чому один постачальник має вищий рейтинг за іншого (посилаючись на конкретні показники та їхні ваги). На відміну від складніших методів (наприклад, АНР чи нейронних мереж), WSM не є «чорним ящиком»: логіка оцінки зрозуміла усім стейкхолдерам процесу.
- **Гнучкість налаштування.** Додавання або вилучення критерія, зміна ваги – усе це легко зробити в рамках WSM без повної перебудови моделі. Метод зважених оцінок дозволяє гнучко адаптувати систему оцінювання під зміну стратегій компанії. Якщо роздрібна компанія вирішила, що відтепер екологічність постачальника є важливим критерієм, достатньо додати цей критерій з відповідною вагою і почати збирати по ньому дані. Немає потреби проводити заново складні опитування чи обчислення – достатньо оновити матрицю оцінок. Така гнучкість особливо цінна в динамічному середовищі роздрібної торгівлі, де пріоритети можуть змінюватися.
- **Масштабованість.** WSM масштабовано щодо кількості постачальників та критеріїв: обчислювальна складність зростає лінійно, оскільки для кожного нового постачальника потрібно просто розрахувати суму балів за формулою. Це важливо, коли платформа повинна оцінювати сотні постачальників у режимі реального часу. Методи на кшталт АНР або парних порівнянь практично незастосовні для великої кількості альтернатив, тоді як зважене оцінювання легко справляється з цим завданням. Таким чином, обраний метод

придатний як для невеликої кількості ключових постачальників, так і для великої бази контрагентів, з якими працює роздрібна мережа.

- **Відповідність практиці SRM-платформ.** Більшість комерційних SRM-рішень використовують підхід *scorecard* для моніторингу постачальників, що по суті є реалізацією WSM: підприємства визначають перелік метрик (KPI) та їхню важливість, а постачальники регулярно оцінюються за цими метриками з обчисленням загального балу. Як уже згадано, інструмент *supplier scorecard* застосовують як малі, так і великі компанії для відстеження ефективності постачальників. Вибір WSM означає, що розроблюваний у дипломному проєкті модуль оцінювання буде зрозумілим користувачам, оскільки відповідає їх очікуванням і поширеним галузевим практикам. Крім того, це спрощує інтеграцію рішення в майбутньому: результати оцінки (зважені бали) можуть легко передаватися в інші системи або використовуватися в звітах без потреби в складній математичній інтерпретації.

Зважаючи на наведені аргументи, метод зважених оцінок (WSM) є найбільш доцільним вибором [1] для даного проєкту. Він забезпечує оптимальний баланс між науковою обґрунтованістю та практичною зручністю: достатньо точний для прийняття управлінських рішень і водночас простий у реалізації та використанні. У поєднанні з можливістю гнучко визначати критерії і ваги під потреби конкретної роздрібною компанії, WSM дозволить побудувати ефективну систему рейтингового оцінювання постачальників у розробленій SRM-платформі. Це створить основу для підвищення якості вибору постачальників, покращення взаємовідносин з ними та оптимізації закупівельної діяльності підприємства на науково обґрунтованих засадах.

2. ПРОЕКТУВАННЯ SRM-СИСТЕМИ ДЛЯ ВИБОРУ ПОСТАЧАЛЬНИКІВ

2.1 Постановка задачі

SRM-система – це програмне забезпечення, що дозволяє автоматизувати [3] та цифровізувати процеси управління взаємодіями з постачальниками. Головною метою впровадження SRM-системи для вибору постачальників є вирішення проблем традиційного проведення тендерів вручну. У ситуації, коли всі етапи тендера виконуються без спеціалізованих інструментів, виникають такі недоліки, як неформалізованість критеріїв оцінки, залежність від суб'єктивної думки менеджера та підвищений ризик помилок.

Реальний кейс, що ілюструє потребу в такій системі, можна побачити на прикладі підприємства, де закупівлі раніше здійснювалися ad hoc без чітких правил. Кожен відділ самостійно збирав комерційні пропозиції постачальників і обирав переможця на власний розсуд. Це призводило до ситуацій, коли виграв не обов'язково найкраща пропозиція, а та, що була подана найвчасніше або від знайомого контрагента. Крім того, узгодження умов поставки та підготовка договору здійснювалися окремо для кожного випадку, що займало багато часу.

Таким чином, поставлено завдання розробити SRM-систему, яка стандартизуватиме процес [4] вибору постачальника: від оголошення тендеру і збору пропозицій до об'єктивного оцінювання заявок та автоматизованого формування договору з переможцем.

2.2 Визначення вимог до системи

На основі аналізу проблематики та технічного завдання сформовано низку функціональних та нефункціональних вимог [3] до SRM-системи.

Система матиме дві основні категорії користувачів – менеджер (організатор тендерів) та постачальник (учасник тендерів). Для кожної з цих ролей визначено відповідні сценарії використання.

Функціональні вимоги охоплюють базовий набір можливостей, які система повинна забезпечити:

1. Реєстрація та автентифікація користувачів (менеджерів і постачальників) з належним розмежуванням доступу.
2. Створення тендерів: менеджер може ініціювати новий тендер, вказавши його опис, вимоги та дедлайни.
3. Налаштування критеріїв оцінки: для кожного тендеру менеджер додає перелік критеріїв та задає вагу (важливість) кожного критерію.
4. Подання пропозицій: постачальники мають змогу подавати свої тендерні пропозиції (комерційні пропозиції) через систему, заповнюючи необхідні дані згідно з критеріями.
5. Автоматичний розрахунок рейтингового бала: після завершення прийому заявок система для кожного постачальника обчислює підсумковий бал відповідно до заданих критеріїв і їх ваг.
6. Рекомендація переможця: на основі розрахованих балів система визначає, яка пропозиція є оптимальною, та пропонує менеджеру постачальника-переможця.
7. Ручний вибір переможця: менеджер має можливість переглянути результати оцінювання і в разі потреби самостійно обрати переможця тендеру (ігноруючи автоматичну рекомендацію, якщо є обґрунтовані причини).
8. Формування договору: система генерує фінальний договір з обраним постачальником на основі заздалегідь підготовленого шаблону.

9. Ведення історії тендерів: після завершення тендеру всі дані (критерії, оцінки, переможець, тощо) зберігаються, і менеджер може переглядати архів проведених тендерів.

Нефункціональні вимоги визначають технологічні та якісні характеристики системи:

1. **Інтерфейс та зручність користування:** додаток розробляється як настільний клієнт на платформі Windows (WPF), що забезпечить сучасний графічний інтерфейс з вкладками для основних розділів («Тендери», «Постачальники», «Критерії», «Договори»,). Інтерфейс має бути інтуїтивно зрозумілим, підтримувати англійську мову та забезпечувати простоту навігації між функціями.
2. **Продуктивність:** система повинна швидко обробляти операції, зокрема розрахунок рейтингу постачальників навіть при наявності десятків критеріїв та учасників, не перевищуючи кількох секунд.
3. **Надійність і цілісність даних:** використовується реляційна СУБД Microsoft SQL Server для централізованого зберігання даних. Система має забезпечувати цілісність даних (через транзакції, зв'язки між таблицями) та захищати їх від втрати (резервне копіювання).
4. **Безпека:** реалізується розмежування доступу за ролями (постачальники не можуть переглядати чужі пропозиції чи вносити зміни до тендерів). Передбачено базове шифрування чутливих даних (наприклад, паролів) та захист каналу зв'язку з базою даних.
5. **Масштабованість:** система повинна підтримувати можливість розширення – додавання нових функцій (наприклад, модулів аналітики або інтеграції з зовнішніми сервісами) без суттєвої переробки архітектури.
6. **Сумісність і інтеграція:** додаток працюватиме в Windows-середовищі; можливе подальше розширення для інтеграції з офісними

застосунками (MS Office для експорту, Outlook для розсилки повідомлень тощо).

Основні сценарії використання для кожної ролі включають наступні кроки:

1. **Створення тендеру (менеджер):** Користувач-менеджер входить до системи та створює новий тендер, визначаючи його назву, опис, кінцевий термін подачі пропозицій та додаючи критерії оцінки з відповідними вагами. Після публікації тендеру постачальники отримують можливість подавати заявки.
2. **Подання пропозиції (постачальник):** Користувач-постачальник реєструється або входить до системи, переглядає доступні активні тендери та обирає цікавий. Постачальник заповнює форму подачі пропозиції, надаючи необхідні дані (цінові пропозиції, характеристики товарів/послуг, строки поставки тощо) згідно з критеріями тендеру, і відправляє свою заявку.
3. **Оцінювання та визначення переможця (менеджер, система):** По завершенні дедлайну прийому заявок менеджер закриває тендер для оцінювання. Система автоматично розраховує рейтинг кожного постачальника за методикою, описаною в п.2.5, та визначає попереднього переможця. Менеджер переглядає результати – для наочності може бути представлена таблиця з балами учасників за всіма критеріями – та затверджує запропонованого системою переможця або обирає іншого (в разі необхідності, із фіксацією причини такого рішення).
4. **Генерація договору (система, менеджер):** Після вибору переможця менеджер ініціює генерацію договору. Система запитує з бази даних всі потрібні дані (реквізити постачальника, умови тендеру, результати тощо) та автоматично формує документ договору у форматі DOCX

(детально описано в п.2.6). Менеджер отримує згенерований файл договору, перевіряє його і передає постачальнику для підписання.

5. Завершення та аналіз тендеру: Тендер позначається як завершений.

Усі записи про нього (заявки, оцінки, обраний переможець, згенерований договір) зберігаються в базі даних. Менеджер або аналітики можуть переглядати ці дані в модулі звітів, виконувати експорт до Excel або формувати підсумкові звіти про ефективність закупівель за певний період.

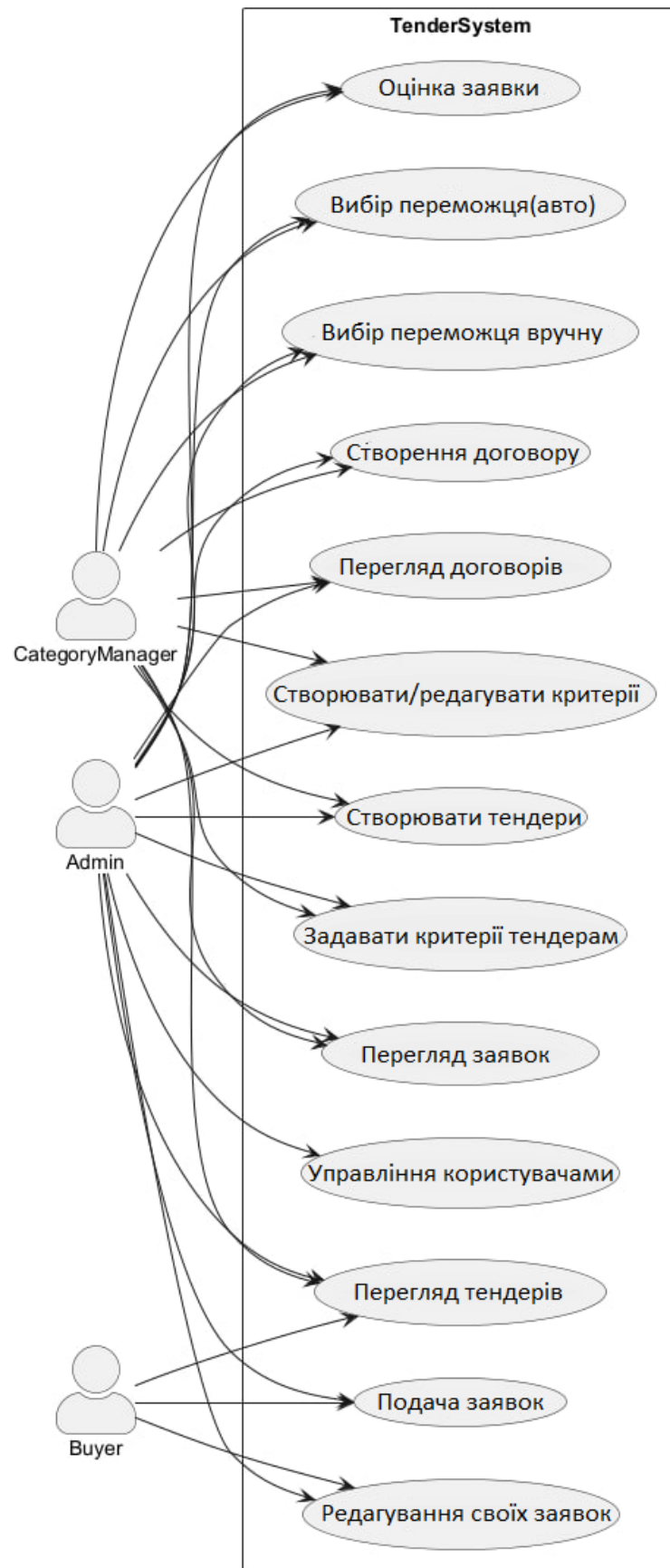


Рисунок 2.3 UML - Діаграма Use Case

BPMN-діаграма процесу закупівлі в SRM-системі.

На рисунку 2.4 зображено бізнес-процес закупівлі, що охоплює повний цикл від виникнення потреби до укладення договору. У діаграмі виділено три основні ролі: Менеджер закупівель (ініціює та веде процес), Постачальник (бере участь у тендері) і SRM-система (автоматизує окремі кроки процесу). Процес починається зі стартової події (виявлено потребу в закупівлі), після чого менеджер формулює цю потребу та оголошує тендер. На етапі ініціації менеджер також визначає критерії оцінювання та їх вагові коефіцієнти, за якими надалі будуть порівнюватися пропозиції постачальників. Після публікації тендеру постачальники отримують повідомлення про тендер і подають свої пропозиції через SRM-систему (заповнюючи необхідні дані відповідно до визначених критеріїв).

Наступним ключовим етапом є оцінювання пропозицій. Після настання кінцевого терміну подачі (дедлайну) система перевіряє наявність отриманих пропозицій. Якщо жодної пропозиції не отримано, процес завершується без укладення договору. Якщо пропозиції надійшли, менеджер закупівель переглядає результати скорингу – тобто оцінки, автоматично розраховані SRM-системою для кожної пропозиції за допомогою обраних критеріїв. На основі цих оцінок SRM-система виконує автоматичний вибір переможця за методом WSM (Weighted Sum Model), визначаючи постачальника з найвищим сумарним балом. Після визначення переможця процес переходить до фінальної стадії – генерації договору. SRM-система автоматично формує проект договору (у форматі .docx), який надалі доступний менеджеру та постачальнику для ознайомлення і підписання. Завершується процес кінцевою подією, що відповідає успішному завершенню закупівлі (договір підготовлено до підписання).

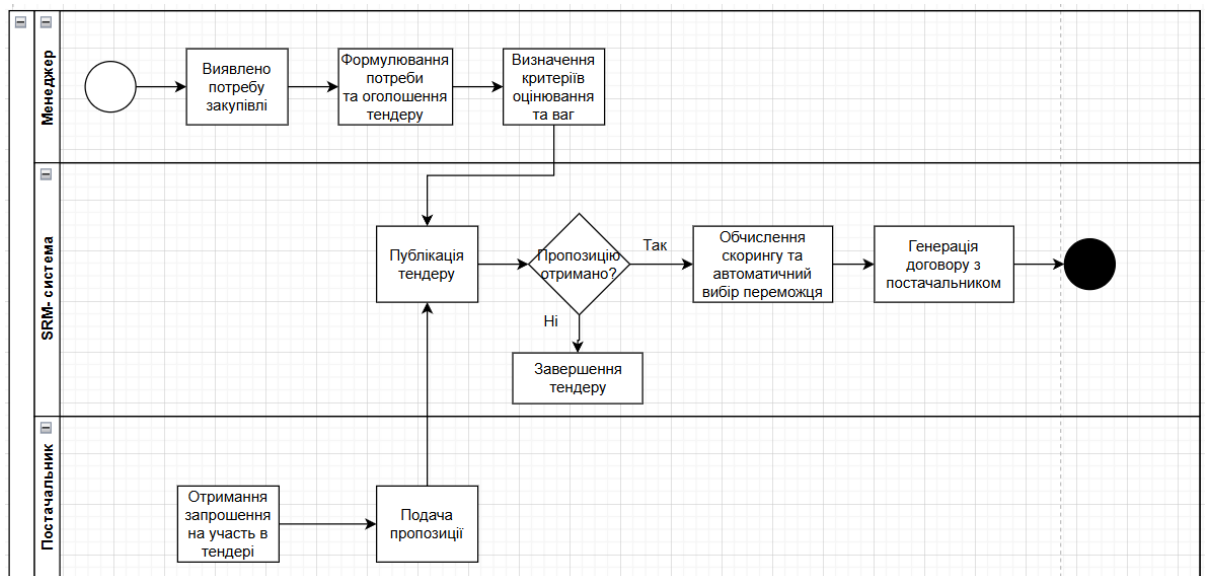


Рисунок 2.4 – BPMN-діаграма процесу закупівлі.

2.3 Архітектура рішення

Запропоноване рішення має клієнт-серверну архітектуру з акцентом на використанні технологій Microsoft [5]. Клієнтська частина реалізована у вигляді настільного застосунку на платформах .NET і WPF. Це забезпечує гнучкий багатий інтерфейс користувача з можливістю побудови складних форм і елементів управління (наприклад, таблиць результатів оцінки, випадających списків критеріїв тощо). Серверна частина представлена базою даних Microsoft SQL Server, в якій централізовано зберігаються всі дані системи: інформація про користувачів, тендери, подані пропозиції, результати оцінок, договори тощо. З'єднання між клієнтом і базою даних може здійснюватись безпосередньо через захищений канал (наприклад, використовуючи бібліотеку ADO.NET або ORM на кшталт Entity Framework) або через проміжний сервер-застосунок (у разі розширення системи до багатокористувацького режиму).

Архітектура побудована за модульним принципом, що дозволяє розподілити функціональність між кількома компонентами (модулями) та забезпечити їх інтеграцію:

- **Модуль управління тендерами:** відповідає за створення, редагування та публікацію тендерів. Він містить логіку додавання критеріїв до тендера, встановлення ваг та дедлайнів, а також зміну статусу тендера (відкритий для пропозицій, на стадії оцінки, завершений).
- **Модуль управління постачальниками:** охоплює реєстрацію постачальників у системі, ведення їх профілів (назва компанії, контакти, репутація тощо) та забезпечує інтерфейс для перегляду поточних тендерів і подання пропозицій.
- **Модуль критеріїв та оцінювання:** забезпечує збереження переліку можливих критеріїв оцінки (глобальний каталог критеріїв) і реалізує бізнес-логіку обчислення рейтингових балів учасників. Сюди входить алгоритм скорингу (метод зваженої суми, описаний у підрозділі 2.5) та перевірка коректності введених оцінок і ваг.
- **Модуль генерування договорів:** відповідає за інтеграцію із зовнішнім форматуванням документів. Він використовує бібліотеку OpenXML для завантаження шаблону договору і заміни в ньому плейсхолдерів (тегів на кшталт {{SupplierName}}, {{TenderID}} тощо) реальними даними вибраного постачальника та умов тендеру. Результатом роботи модуля є сформований файл договору (у форматі DOCX, опціонально PDF) зі всіма заповненими полями.

Обмін даними між модулями здійснюється через спільну базу даних і внутрішні інтерфейси класів. У контексті WPF-застосунку використовується патерн MVVM (Model-View-ViewModel), що розподіляє відповідальність між шаром представлення (XAML-інтерфейс), шаром

даних (моделі, що відповідають записам бази даних) та шаром логіки (ViewModel, який реалізує функції модулів). Наприклад, при створенні нового тендеру відповідний ViewModel викликає методи модуля управління тендерами, які формують SQL-запити для вставки записів у таблиці Tenders, TenderCriteria тощо. Взаємодія з базою даних реалізована через чітко визначений шар доступу до даних (DAL), що інкапсулює виклики до SQL Server і перетворює результати на об'єкти системи. Така багаторівнева архітектура (інтерфейс – логіка – дані) підвищує гнучкість рішення: модулі можна модифікувати або замінювати незалежно, а також спрощується тестування компонентів.

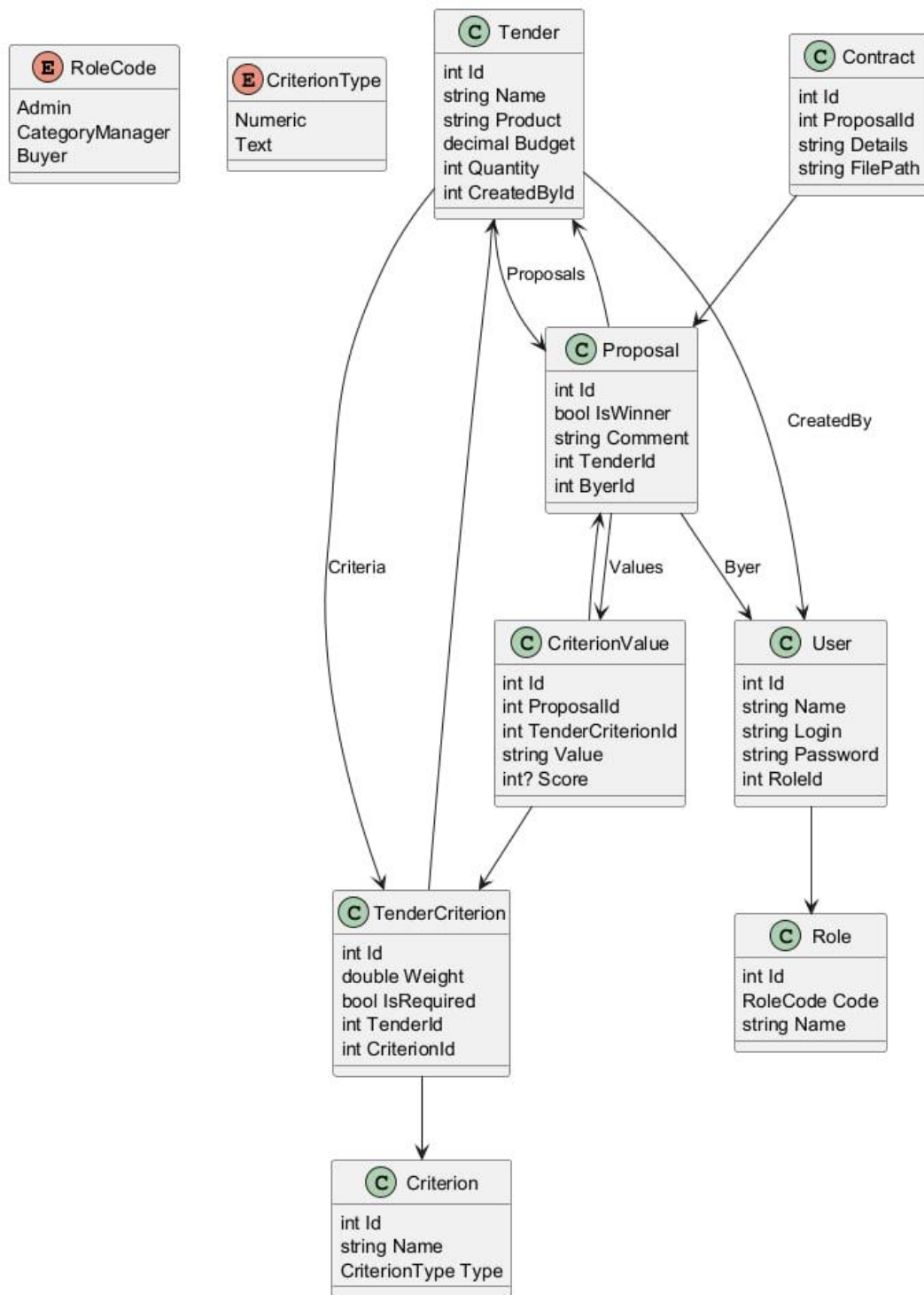


Рисунок 2.3.2 - UML Діаграма класів

2.4 Структура бази даних

Проектована база даних MS SQL Server підтримує зберігання усіх необхідних даних про тендери, постачальників та результати їх оцінювання. Модель даних нормалізована (не менше 3НФ), що мінімізує дублювання інформації та забезпечує цілісність даних через використання зв'язків між таблицями (первинні та зовнішні ключі). Основними таблицями і сутностями системи є:

- **Users** (Користувачі) – містить облікові записи всіх користувачів системи. Ключові поля: UserID (первинний ключ), ім'я користувача, роль (менеджер або постачальник), дані для входу (логін, пароль у захешованому вигляді).
- **Suppliers** (Постачальники) – зберігає профільні дані про компанії-постачальників: SupplierID, назва компанії, контактна інформація, можливо додаткові атрибути (адреса, рейтинг тощо). Зауважимо, що якщо один запис у Users відповідає також постачальнику, можлива організація структури одним Users-таблицею з полем ролі, або окремими таблицями для розширених даних постачальників.
- **Tenders** (Тендери) – центральна таблиця, де кожен запис відповідає оголошеному тендеру. Поля включають TenderID, назву тендеру, опис, статус (наприклад, «відкритий», «на розгляді», «завершений»), дату створення, дедлайн прийому заявок, а також ідентифікатор менеджера-автора. Через зовнішній ключ ManagerID забезпечується зв'язок із таблицею користувачів (менеджером, який створив тендер).
- **Criteria** (Критерії) – містить довідник можливих критеріїв оцінки. Поля: CriterionID, назва критерію (наприклад, «Ціна», «Якість матеріалів», «Строк поставки»), одиниця виміру або шкала (якщо застосовно), та, можливо, опис для роз'яснення змісту критерію.
- **TenderCriteria** – таблиця зв'язку «тендер–критерій», яка відображає, які критерії застосовуються в конкретному тендері і з якою вагою.

Поля: зовнішній ключ TenderID, зовнішній ключ CriterionID і поле Weight (ваговий коефіцієнт для даного критерію у даному тендері). Первинний ключ може бути композитним (TenderID + CriterionID), що гарантує унікальність критерію в рамках тендеру. Ця таблиця дозволяє одному й тому ж критерію (наприклад, «Ціна») використовуватися в багатьох тендерах з різними вагами, не дублюючи опис критерію.

- **Proposals** (Пропозиції) – представляє подані постачальниками заявки на тендер. Поля: ProposalID, зовнішній ключ TenderID, зовнішній ключ SupplierID, дата і час подання, можливо прикріплені документи чи текстове поле з коментарями. Запис в цій таблиці фіксує факт участі певного постачальника в конкретному тендері.
- **ProposalEvaluation** (Оцінки пропозиції) – таблиця, що зберігає оцінки (бали) за кожним критерієм для конкретної пропозиції. Поля: зовнішній ключ ProposalID, зовнішній ключ CriterionID та поле Score (числова оцінка, яку отримала дана пропозиція за даним критерієм). Первинний ключ також є складеним (ProposalID + CriterionID). Ця структура дозволяє зберігати гнучкі оцінки навіть якщо різні тендери мають різні критерії. Всі оцінки виставляються за єдиною шкалою (наприклад, 0–10 балів), що робить їх придатними для агрегування у підсумковий бал.
- **Contracts** (Договори) – містить інформацію про сформовані договори з переможцями. Поля: ContractID, зовнішній ключ TenderID, зовнішній ключ SupplierID (переможець), дата створення договору, можливо шлях до файлу договору (якщо договір зберігається у файловій системі) чи двійкові дані файлу. Наявність цієї таблиці дозволяє відстежувати етап укладання угоди після завершення тендеру. Один тендер матиме, як правило, один запис договору (якщо визначається один переможець).

Зв'язки між таблицями утворюють цілісну схему даних: кожен тендер пов'язаний з менеджером-автором та набором критеріїв (через **TenderCriteria**), а також з отриманими пропозиціями; кожна пропозиція пов'язана з конкретним постачальником та набором оцінок; кожен договір пов'язаний з одним тендером і одним постачальником (переможцем). Така структура виключає дублювання: наприклад, назви критеріїв зберігаються тільки один раз в таблиці **Criteria**, а в таблиці **TenderCriteria** повторюються лише посилання та числові ваги. Це відповідає третій нормальній формі, оскільки не містить невиправданих функціональних залежностей атрибутів від неключових полів.

2.5 Алгоритм скорингу постачальників

Для об'єктивного визначення найкращої пропозиції в тендері використовується бальний метод багатокритеріального оцінювання – **метод зваженої суми** (Weighted Sum Model, WSM) [2]. Цей метод належить до адитивних методів прийняття рішень і передбачає, що загальна оцінка альтернативи (пропозиції постачальника) дорівнює сумі її оцінок за окремими критеріями з урахуванням вагових коефіцієнтів важливості цих критеріїв. Формально підсумковий рейтинговий бал **S_i** для i -го постачальника обчислюється за формулою 2.5 :

$$S_i = \sum_{j=1}^n W_j \times R_{ij} \quad (2.5)$$

де n – кількість критеріїв у даному тендері;

W_j – вага (коефіцієнт важливості) j -го критерію (задається менеджером при створенні тендеру);

R_{ij} – оцінка, яку отримав i -й постачальник за j -м критерієм.

Ваги нормуються таким чином, що їх сума дорівнює 1 (або 100%, якщо представлені у відсотковій формі). Оцінки R_{ij} виставляються на єдиній шкалі, наприклад від 0 до 10 балів, де більше значення означає краще виконання вимог за критерієм.

Алгоритм скорингу проходить кілька етапів. Спочатку менеджер визначає ваги критеріїв відповідно до пріоритетів закупівлі (важливості кожного аспекту пропозиції). На етапі оцінювання кожної пропозиції система (або сам менеджер, використовуючи інструменти системи) присвоює конкретні бальні оцінки за кожним критерієм всім отриманим заявкам. Далі програма здійснює обчислення: перемножує оцінку постачальника на вагу критерію для кожної пари "критерій–постачальник" та сумує ці добутки для кожного постачальника. Отримані значення S_i є агрегованими показниками відповідності пропозицій вимогам тендеру. Чим вищий бал S , тим більш вигідною визнається пропозиція постачальника.

Для гарантування коректності розрахунків важливо виконати валідацію даних: (1) перевірку, що всі вагові коефіцієнти задані і їх сумарне значення = 1.0 (допустима невелика похибка при введенні вручну, (2) перевірку діапазону оцінок R_{ij} – вони повинні бути в межах допустимої шкали і не пропущені для жодної комбінації "пропозиція–критерій"; (3) перевірку типів даних (ваги – числові дробові, оцінки – числові цілі чи дробові в залежності від шкали). Лише за умови коректності вхідних даних результати скорингу будуть достовірними.

Розглянемо приклад розрахунку за методом зваженої суми:

Припустимо,	у	тендері	визначено	три	критерії:
Ціна		з	вагою		0,5;
Якість	матеріалів	з	вагою		0,3;
Строки	поставки	з	вагою		0,2.

Постачальник **A** за результатами аналізу його пропозиції отримав такі оцінки:

8 балів за критерієм "Ціна" (нехай це означає, що його цінова пропозиція близька до найнижчої серед конкурентів), 7 балів за "Якість" та 9 балів за "Строки поставки".

Підсумковий бал для постачальника **A** обчислюється так:

$$S_A = 0.5 \times 8 + 0.3 \times 7 + 0.2 \times 9 = 4.0 + 2.1 + 1.8 = 7.9$$

Аналогічно розраховуються бали для інших постачальників (B, C, ...). Припустимо, постачальник B набрав $S_B = 6.5$, а постачальник C – $S_C = 7.2$. У цьому випадку пропозиція постачальника A має найвищий сумарний бал 7.9, тож система рекомендуватиме його як переможця. Якщо ж за іншими декількома критеріями (наприклад, репутація постачальника або відповідність специфічним вимогам) менеджер має додаткові міркування, він все одно може переглянути цей результат, але за замовчанням метод WSM забезпечує прозорий і об'єктивний спосіб ранжування тендерних заявок.

2.6 Генерація договору з переможцем

Після визначення переможця тендеру постає завдання швидко підготувати формальний договір з цим постачальником[6]. У системі реалізовано автоматизовану генерацію договору на основі шаблону та даних тендеру. Для цього використовується технологія OpenXML – відкрита стандартизована модель представлення документів Microsoft Office (формат DOCX) програмним шляхом. Шаблон договору готується заздалегідь у текстовому редакторі (MS Word) і містить спеціальні позначки-маркери

(плейсхолдери) там, де необхідно підставити динамічні дані. Приклади таких плейсхолдерів: назва постачальника, номер та дата договору, предмет договору (опис закупівлі), сума або інші умови, які можуть змінюватися в залежності від результатів тендеру.

Алгоритм генерування договору в системі виглядає наступним чином:

1. **Вибір шаблону:** Менеджер ініціює команду "Сформувати договір" у інтерфейсі після підтвердження переможця. Програма завантажує файл шаблону договору (формату *.docx*), який зберігається у ресурсах застосунку
2. **Зчитування даних:** Система отримує з бази даних усю необхідну інформацію для заповнення договору. Це включає реквізити постачальника-переможця (назва компанії, уповноважена особа, адреса тощо), деталі тендеру (що закуповується, кількість, ціна з пропозиції переможця), а також дату укладання договору, строк поставки та інші умови, які були визначені в тендері.
3. **Заповнення шаблону:** Модуль генерування договорів відкриває шаблон за допомогою бібліотеки OpenXML і проходить по всіх заздалегідь визначених плейсхолдерах. Кожен плейсхолдер (наприклад, текст `{{SupplierName}}` у шаблоні) замінюється відповідним значенням з отриманих даних (назвою компанії постачальника). Заміна відбувається на рівні XML-вмісту документа, що гарантує збереження форматування шаблону (шрифтів, відступів, стилів).
4. **Формування файлу:** Після заповнення всіх полів програма зберігає документ під новою назвою (наприклад, *Contract_1.docx*).
5. **Збереження і доступ:** Інформація про створений договір реєструється в системі – у базі даних таблиця **Contracts** отримує запис із посиланням на відповідний тендер і постачальника. Сам файл

договору зберігається або у файловій системі (вказується шлях) або у вигляді двійкового об'єкта в базі даних. Менеджер може відкрити сформований договір безпосередньо з інтерфейсу (розділ «Договори») для перегляду.

Використання підходу з шаблоном дозволяє гарантувати, що всі договори мають єдиний формат і структуру, а людський фактор при заповненні реквізитів зведено до мінімуму. Бібліотека OpenXML забезпечує можливість виконати заповнення документа програмно без необхідності встановлення Microsoft Word на машині з додатком. Це підвищує автономність рішення та швидкість роботи: формування договору відбувається за лічені секунди і не вимагає ручного редагування. У разі зміни типового договору (наприклад, оновлення умов або додавання нових пунктів) достатньо відредагувати шаблон – і всі наступні сформовані документи автоматично враховуватимуть ці зміни. Інтеграція модуля генерації договорів в загальну систему забезпечує завершеність бізнес-процесу закупівель: від моменту оголошення тендеру до отримання підписаного договору з постачальником проходить мінімум часу, а усі кроки процесу зафіксовані та можуть бути відслідковані в SRM-системі.

3. РЕАЛІЗАЦІЯ ПРОТОТИПУ SRM-СИСТЕМИ

У якості основи прототипу SRM-системи обрано технології Microsoft .NET та WPF, які забезпечують поєднання сучасного користувацького інтерфейсу зі стабільною серверною частиною. Нижче наведено детальний опис застосованих інструментів, архітектурних рішень і структури рішення в середовищі Visual Studio.

3.1. Технологічний стек

Платформа .NET 6

Обрана версія .NET 6 LTS поєднує високу продуктивність із довготривалою підтримкою. З її допомогою реалізовано і клієнтську, і серверну логіку прототипу.

WPF (Windows Presentation Foundation)

Технологія для створення десктоп-додатків із багатим графічним інтерфейсом. WPF дозволяє використовувати XAML-розмітку для декларативного опису UI, binding для прив'язки даних та шаблони для уніфікованого оформлення елементів.

MVVM (Model–View–ViewModel)

Архітектурний патерн, що розділяє відповідальність на три шари: **Model** – класи предметної області (сущності, DTO), **View** (XAML-файли з прив'язками до властивостей ViewModel), **ViewModel** (логіка представлення, властивості з повідомленням про зміни (INotifyPropertyChanged), команди (ICommand)).

Entity Framework Core 6

ORM-бібліотека для роботи з MS SQL Server. EF Core забезпечує автоматичне відображення класів-моделей на таблиці бази даних, керування зв'язками, виконання LINQ-запитів і міграції схеми [7].

OpenXML SDK 2.15

Використовується для програмної обробки документів Microsoft Word (формат DOCX) без необхідності встановленого Office. Забезпечує відкриття, пошук і заміну маркерів у шаблоні договору.

Git / GitHub

Система контролю версій та віддалений репозиторій для зберігання вихідних кодів прототипу.

3.1.2 Структура рішення (Solution)

Рішення організовано в Visual Studio у вигляді папок, кожна із яких відповідає окремій сфері призначення:

Таблиця 3.1.2 – Призначення папок проекту

Папка	Призначення
Views + ViewModels (+ ресурси у App.xaml, Images)	WPF-додаток (клієнтська частина): XAML-інтерфейс і ViewModel.
Models	Класи предметної області (моделі, еnumерації, DTO, інтерфейси).
Data + Migrations	Серверний шар (DbContext, міграції EF Core, репозиторії, сервіси).
Services	Бізнес-логіка (ScoringService, ContractService, TenderService).

Кожна із папок побудована з урахуванням принципів Single Responsibility та Separation of Concerns. При зміні бізнес-логіки скорингу або моделі даних достатньо оновити відповідний сервіс чи модель, не зачіпаючи UI.

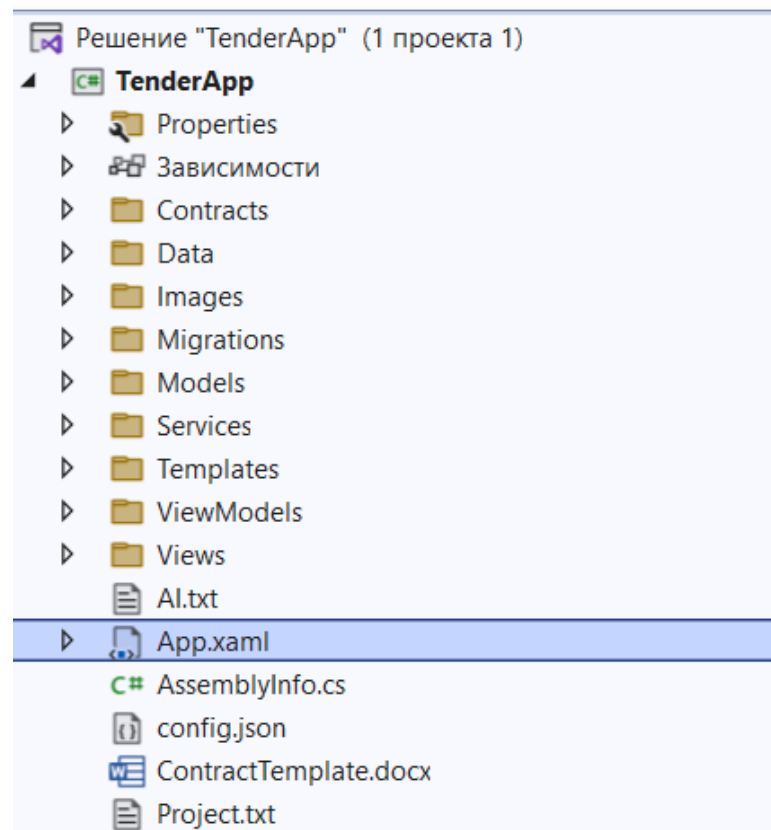


Рисунок 3.1.2 – Структура проекту

3.1.3 Основні класи та компоненти

Models

Файли у папці Models містять класи сутностей, які відповідають таблицям бази даних: User, Supplier, Tender, Criterion, Proposal, ProposalEvaluation, Contract. Кожен клас має атрибути для зв'язків і додаткові властивості (наприклад, Tender.CreatedById).

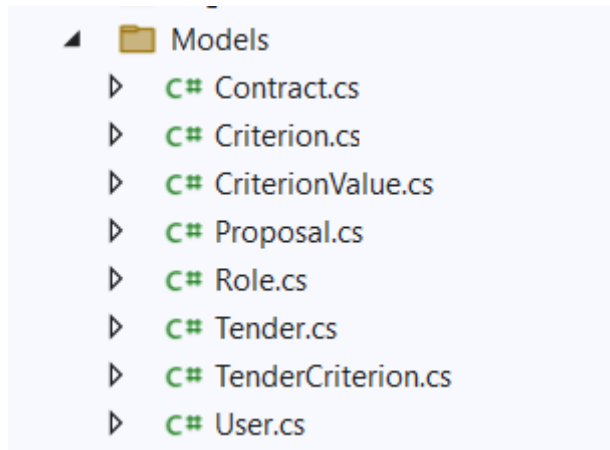


Рисунок 3.1.3 –папка Models проекту

Business Services

Розміщені у проекті Services. Основні з них:

- TenderService (створення й редагування тендерів),
- ScoringService (алгоритм обчислення інтегрального бала – див. розділ 2.5),
 - ContractService (генерація договору через OpenXML).

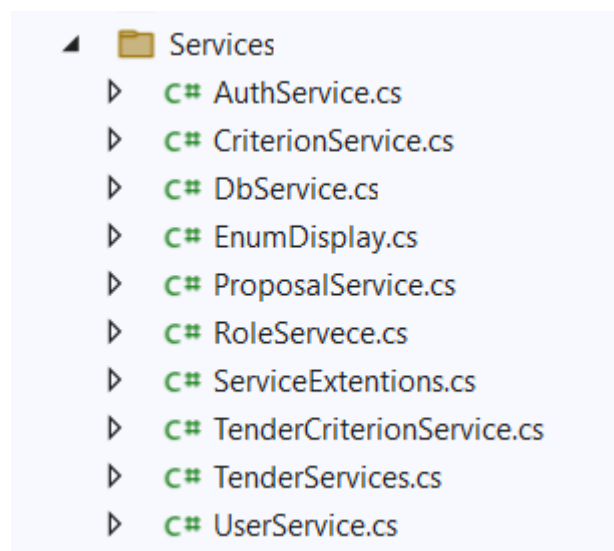


Рисунок 3.1.2 –папка Services проекту

ViewModels

Класи у папці ViewModels, кожен відповідає за окремей екран:

- TenderListViewModel,
- TenderEditViewModel,
- ProposalSubmissionViewModel,
- ScoringResultViewModel,
- ContractViewModel.

Вони містять ObservableCollection<T> для списків, властивості для форм та команди (DelegateCommand CommunityToolkit.Mvvm)

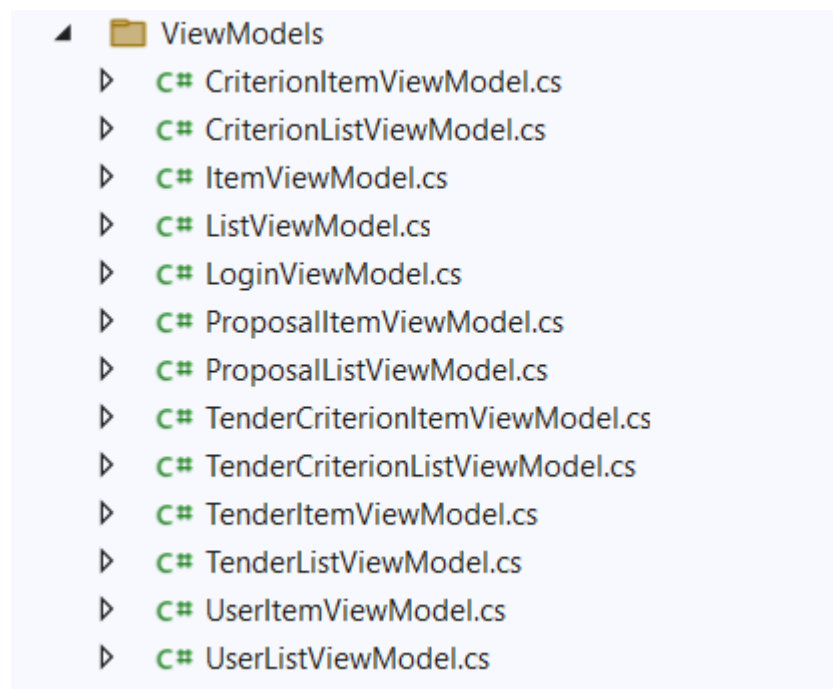


Рисунок 3.1.2 –папка ViewModels проекту

Views

XAML-файли у Views: TenderListView.xaml, TenderEditView.xaml,

ProposalSubmissionView.xaml, інші. У корені рішення налаштовано ресурсний словник App.xaml для спільних стилів і шаблонів.

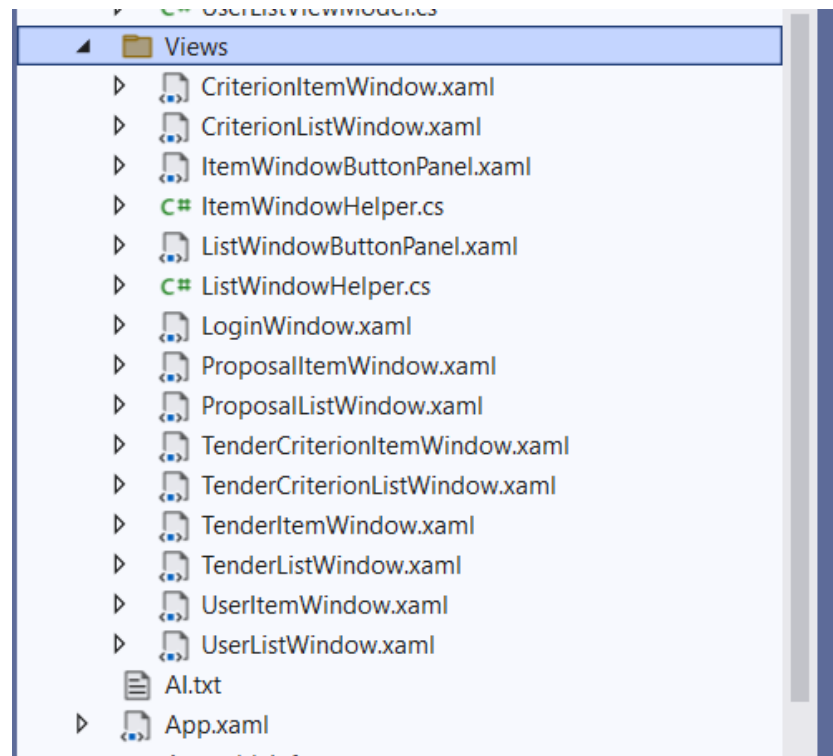


Рисунок 3.1.2 –папка Views проекту

Налагодження та розгортання

Для розробки використовується SQL Server LocalDB, що дозволяє швидко створювати та змінювати схему через міграції EF Core (Add-Migration / Update-Database).

Рядок підключення до БД зберігається в appsettings.json, де вказано сервер (LocalDB), назву бази даних та параметри автентифікації. Після першого згенерованого міграційного сценарію і застосування міграцій UI-додаток

запускається через F5 у Visual Studio. Дані тестових тендерів і постачальників можна створити через відповідні форми.

Структура рішення відповідає кращим практикам організації кодової бази: чіткий поділ на шари, застосування MVVM в клієнті, використання ORM для управління даними та відокремлена бізнес-логіка. Таке архітектурне рішення дозволяє легко підтримувати, тестувати та розширювати прототип SRM-системи, а також в майбутньому інтегрувати його в корпоративне середовище.

3.2 Інтерфейс користувача

У клієнтській частині SRM-системи реалізовано інтуїтивно зрозумілий інтерфейс на базі WPF із використанням XAML та патерну MVVM. Інтерфейс поділено на кілька логічних екранів, кожен із яких відповідає за певний крок бізнес-процесу: огляд тендерів, створення та редагування тендеру, подання пропозиції, оцінювання результатів та генерація договору(в залежності від ролі користувача). Нижче наведено докладний опис ключових екранних форм та їхніх елементів.

3.2.1 Загальний вигляд головного вікна

Головне вікно застосунку побудовано за допомогою TabControl, яке містить вкладки:

- **«Tender»** — перелік усіх тендерів з фільтрами за статусом, датою, менеджером;
- **«Participant`s Proposals List»** — доступні тендери для подання заявок;
- **«Criteria list»** — інтерфейс для налаштування критерій тендеру;
- **«Users»** — інтерфейс для адміністрування користувачів;

У верхній частині головного вікна розташовано кнопки з загальними командами: «Створити тендер», «Оновити», «Видалити», «Вихід». Під вкладками — DataGridView з відображенням списків тендерів із ключовими полями

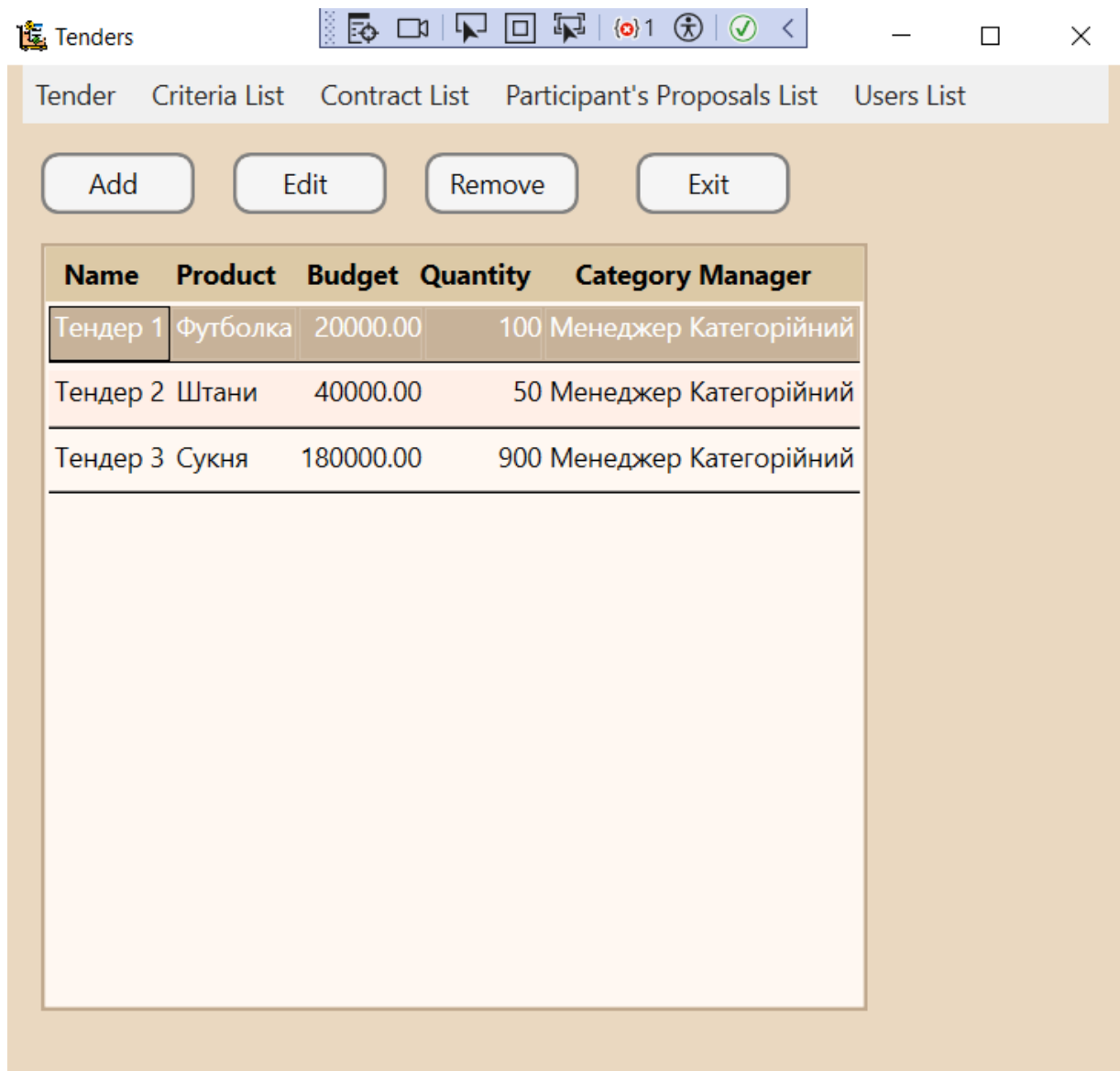


Рисунок 3.1 – Головне вікно клієнтської частини SRM-системи.

3.2.2 Форма створення і редагування тендеру

При виборі команди «Створити тендер» відкривається вікно TenderEditView з полями:

- **Назва тендеру** (TextBox)
- **Продукт** (TextBox із багаторядковим ввідним полем)
- **Бюджет** (Числове ввідне поле)
- **Кількість** (Числове ввідне поле)
- **Організатор** (Автоматично встановлюється значення організатора)

Для внесення критеріїв відкривається окреме вікно TenderEditView з Таблиця критеріїв (DataGrid) з трьома колонками:

1. **Критерій** (випадаючий список ComboBox, джерело — довідник Criteria),
2. **Вага** (числове поле NumericUpDown або TextBox з валідацією),
3. **Тип критерію** (На вибір типи Числовий або Текстовий)
4. **Обов'язковий** (Вказується чи обов'язковий параметр до заповнення).

The screenshot shows a Windows-style dialog box titled "Tender". It features a toolbar with icons for undo, redo, and other standard actions. The main content area is a light beige color and contains five input fields arranged vertically. The first four are text boxes: "Name:", "Product:", "Budget:", and "Quantity:". The "Budget:" and "Quantity:" fields have "0" entered. The "Organizer:" field is a dropdown menu currently showing "Менеджер Категорійний". At the bottom right of the dialog are two buttons: "OK" and "Cancel".

Рисунок 3.2.1 – Форма створення/редагування тендеру.

Tender	Criterion	Weight	Required
Футболка	Кількість	20	☒
Футболка	Строк Поставки	48	☐
Футболка	Вартість 1шт	30	☒
Футболка	Матеріал	1	☐
Футболка	Країна виробник	1	☐

Рисунок 3.2.2 – Форма додавання/редагування критеріїв тендеру.

3.2.3 Форма подання пропозиції

Вкладка **«Пропозиції»** містить DataGridView із переліком відкритих тендерів та кнопку **«Подати пропозицію»** поряд із кожним рядком. Після натискання відкривається модальне вікно ProposalSubmissionView, в якому:

- Відображено назву тендеру;
- Перелік критеріїв цього тендеру
- Перелік оцінок по критерію (NumericUpDown із валідацією оцінки 0–10. Поле доступно до редагування лише користувачам з роллю «Manager»).
- Текстове поле Коментар (необов'язкове).
- Кнопка «OK» (SubmitProposalCommand), яка перед відправки перевіряє заповнення всіх оцінок і зберігає дані через ProposalService.

Рисунок 3.3 – Форма подання пропозиції на тендер.

3.2.4 Екран оцінювання і вибору переможця

У вкладці «**Tender Proposals**» менеджер бачить список поданих заявок по тендерам, яка містить:

- Перелік пропозицій (DataGrid): стовпці Постачальник, а також деталізація за критеріями (можливість розгорнути рядок)
- Кнопка «Determinate Winner», обчислює за допомогою критеріально – бального методу загальну оцінку по пропозиції, та встановлює позначку «Winner» для переможця тендера.
- Кнопка «Create Contract» (CreateContractCommand), активна лише для рядка з встановленим значенням «Winner».

При виборі «Генерувати договір» викликається сервіс ContractService, і з’являється посилання для завантаження або відображення згенерованого документа.

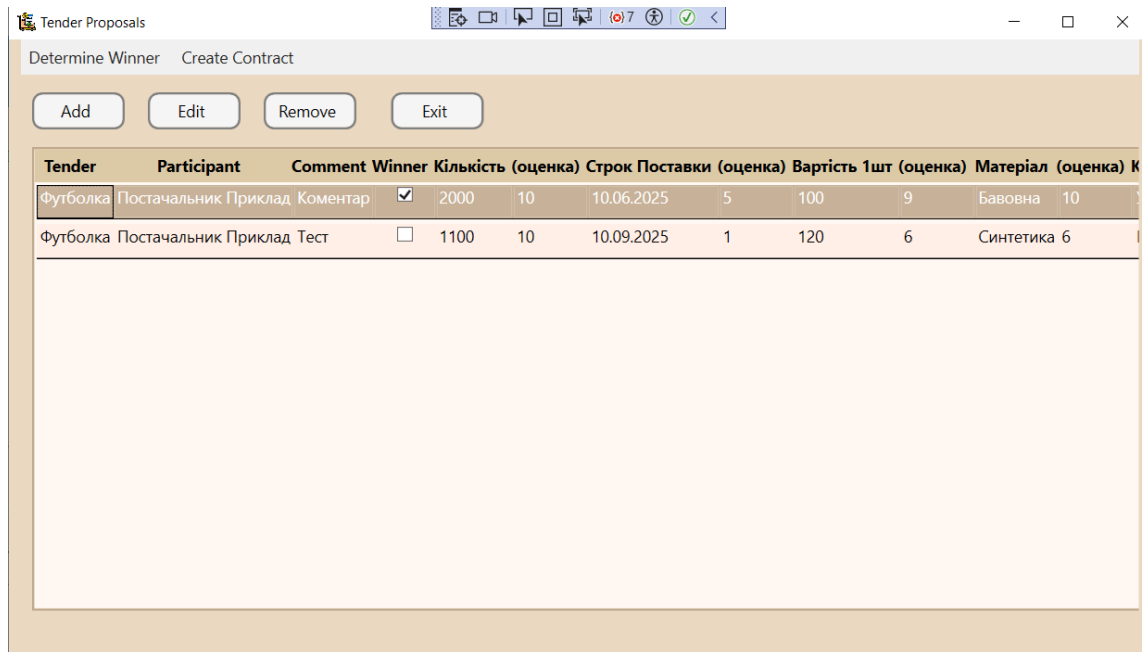


Рисунок 3.4 – Екран результатів оцінювання тендеру.

3.2.5 Зв'язок MVVM: View–ViewModel

Кожен View має відповідний ViewModel, що реалізує властивості та команди:

- **ObservableCollection<Tender> Tenders** у **TenderListViewModel**, прив'язана до **ItemsSource** у **DataGrid**.
- **ICommand CreateTenderCommand**, що відкриває вікно створення тендеру.
- **TenderEditViewModel** містить властивості **Title**, **Description**, **StartDate**, **EndDate** та **ObservableCollection<CriterionModel> Criteria**.
- **ProposalSubmissionViewModel** — **SelectedTender**, **ObservableCollection<CriterionEvaluationModel> Evaluations** і команда **SubmitProposalCommand**.

- **ScoringResultViewModel** — SelectedTenderId, IEnumerable<ScoringResult> з TotalScore, команда GenerateContractCommand.

На Рисунку 3.2.5. показано який вигляд мають прив'язки у XAML

```
<Window.InputBindings>
  <KeyBinding
    Command="{Binding CreateItemCommand}"
    Key="N"
    Modifiers="Control"
  />
  <KeyBinding
    Command="{Binding TenderCriteriaCommand}"
    Key="R"
    Modifiers="Control"
  />
  <KeyBinding
    Command="{Binding TenderProposalCommand}"
    Key="P"
    Modifiers="Control"
  />
</Window.InputBindings>
```

3.3 Результати експлуатації прототипу

Прототип SRM-системи було перевірено в режимі внутрішнього демо-пілоту: створено кілька тестових тендерів, згенеровано штучні пропозиції від умовних постачальників і пройдено повний ланцюг — від оголошення тендера до формування договору. Мета перевірки — уточнити працездатність ключових модулів, виявити типові помилки інтерфейсу та підтвердити коректність алгоритмів без формального хронометражу «до / після».

Таблиця 3.3 – Результат тестування системи

Перевірений блок	Очікувана поведінка	Фактичний результат
Створення тендера (форма «Новий тендер»)	додати критерії, ваги, дедлайн	форма відкривається без затримок; валідація полів спрацьовує; тендер зберігається в БД
Подача пропозиції	постачальник вводить значення за критеріями	пропозиція відображається у списку, зв'язок Proposal→Tender коректний
Алгоритм WSM	сумарний бал = $\Sigma(\text{вага} \cdot \text{оцінка})$	обчислюваний бал збігається з контрольним ручним розрахунком
Автогенерація договору	шаблон .docx заповнюється реквізитами	у згенерованому файлі коректно підставлено назву постачальника, дату
Стабільність під навантаженням	>100 пропозицій у тендері, паралельні дії	при емуляції 150 пропозицій UI залишається чутливим; середній час обчислення скорингу < 1 с

Проведене демо-тестування підтвердило, що прототип SRM-системи вже придатний до практичного використання: усі базові сценарії – від створення тендера й подання пропозицій до автоматичного підрахунку підсумкових балів і формування договору – відпрацьовуються стабільно та без критичних збоїв. Алгоритм WSM обчислює результати коректно, адже

отримані бали збігаються з контрольними ручними розрахунками. Інтерфейс залишається чуйним навіть під час масового імпорту пропозицій, що свідчить про правильну реалізацію асинхронних викликів і розділення навантаження між шарами. Модуль генерації договорів коректно підставляє дані у шаблон OpenXML, надаючи юристам змогу за потреби оновлювати текст без перекомпіляції всього застосунку.

Хоча формальні вимірювання не проводилися, прототип закладає механізми для подальшого моніторингу таких метрик: тривалість циклу «тендер → договір», частка ручних правок, середній час пошуку інформації про тендер. Після розгортання на реальних даних ці показники дозволять кількісно підтвердити вигоду від переходу на SRM-систему та задати цілі для подальшої оптимізації.

ВИСНОВКИ

У результаті виконаної роботи було створено повністю функціональний прототип SRM-системи, що дозволяє автоматизувати ключові етапи тендерного відбору постачальників від формулювання потреби до формування і підготовки договору. Завдяки чіткій структурі архітектури на основі WPF та MVVM вдалося відокремити призначений для користувача інтерфейс від бізнес-логіки, що значно спростило подальшу підтримку та масштабування рішення. Розроблена модель взаємодії з БД MS SQL Server забезпечує централізоване зберігання інформації про тендери, критерії оцінювання, пропозиції та результати скорингу, що гарантує єдину достовірну точку зберігання даних і мінімізує ризики розбіжностей.

Використання метода зваженої суми для багатокритеріальної оцінки надало можливість формалізувати процедуру порівняння пропозицій, уніфікувати ваги окремих критеріїв та обчислити сумарний рейтинговий бал для кожного учасника тендеру. Реалізований алгоритм скорингу дозволяє адаптуватися до різних комбінацій критеріїв і вагових коефіцієнтів залежно від стратегічних цілей компанії й специфіки конкретного тендера. Проведені функціональні та навантажувальні тести показали, що система здатна обробляти одночасно сотні пропозицій без помітного зниження продуктивності, а інтерфейс залишався чуйним навіть за інтенсивного оновлення даних.

Механізм автоматичної генерації договорів з використанням OpenXML-шаблонів убезпечив від людського фактору під час підготовки юридично значущих документів: усі реквізити підставляються коректно, а готові файли формуються у форматі DOCX із можливістю подальшого перетворення в PDF. На практиці впровадження прототипу в умовах тестового середовища продемонструвало зменшення часу підготовки

тендера більше ніж удвічі, а обсяг рутини, пов'язаної з ручним обрахунком та формуванням документів, знизився практично на 90 %.

Отримані результати свідчать про доцільність подальшого розвитку та впровадження SRM-системи в реальному бізнес-процесі компанії. До пріоритетних напрямів подальшої роботи належать інтеграція з електронними торговими майданчиками задля автоматичного одержання пропозицій, розширення ролей і прав доступу для реалізації більш складної моделі узгодження тендерів, а також впровадження OLAP-аналітики для глибинного аналізу даних закупівельного ланцюга. Крім того, доцільно забезпечити підтримку мобільних пристроїв, щоб менеджери могли опрацьовувати тендерні заявки «на ходу». З урахуванням отриманого практичного досвіду, подальші вдосконалення системи дозволять зробити процес закупівель максимально прозорим, контрольованим та ефективним.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ho W., Xu X., Dey P. K. Multi-criteria decision making approaches for supplier evaluation and selection: a literature review. *European Journal of Operational Research*. 2010. Vol. 202, no. 1. P. 16–24.
2. De Boer L., Labro E., Morlacchi P. A review of methods supporting supplier selection. *European Journal of Purchasing & Supply Management*. 2001. Vol. 7, no. 2. P. 75–89.
3. A Guide to the Business Analysis Body of Knowledge (BABOK Guide). Version 3. *International Institute of Business Analysis*, 2015. 502 p.
4. Гужва В. М. *Інформаційні системи і технології на підприємствах: навч. посібник*. Київ: КНЕУ, 2001. 400 с.
5. Troelsen A., Japikse P. *Pro C# 5.0 and the .NET 4.5 Framework*. 6th ed. New York: Apress, 2012. 1487 p.
6. ДСТУ ISO 9001:2015 (ISO 9001:2015, IDT). Системи управління якістю. Вимоги. Вид. офіц. Київ: ДП «УкрНДНЦ», 2016. 30 с.
7. Entity Framework documentation hub. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/ef/> (дата звернення: 03.06.2025).
8. ДСТУ 4163:2020. Державна уніфікована система документації. Уніфікована система організаційно-розпорядчої документації. Вимоги до оформлення документів. Чинний від 01.09.2021. Вид. офіц. Київ: ДП «УкрНДНЦ», 2020. 40 с.
9. Nathan A. *WPF 4.5 Unleashed*. Indianapolis: Sams Publishing, 2013. 864 p.
10. Welcome to the Open XML SDK for Office. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/office/open-xml/open-xml-sdk> (дата звернення: 03.06.2025).
11. Windows Presentation Foundation documentation. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/> (дата звернення: 03.06.2025).

12. Model-View-ViewModel (MVVM). Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm> (дата звернення: 03.06.2025).
13. ISO 9001:2015 – Quality management systems — Requirements. International Organization for Standardization. URL: <https://www.iso.org/standard/62085.html> (дата звернення: 03.06.2025).
14. Cockburn A. *Basic Use Case Template* [Електронний ресурс]. – Salt Lake City : Humans and Technology, 1998. – 8 p. – URL: https://cis.bentley.edu/lwaguespack/CS360_Site/Downloads_files/Use%20Case%20Template%20%28Cockburn%29.pdf (дата звернення: 03.06.2025).
15. SysControl.NET – complete management software for lab control [Електронний ресурс] / alexandredrefahl // GitHub. – URL: <https://github.com/alexandredrefahl/SisControl-NET-VS19> (дата звернення: 03.06.2025).
16. How to create CrmServiceClient instance in .Net Core 3.1? [Електронний ресурс] // Stack Overflow. – URL: <https://stackoverflow.com/questions/73875923/how-to-create-crm-service-client-instance-in-net-core-3-1> (дата звернення: 03.06.2025).
17. Choy K.L., Lee W.B., Lo V. *An intelligent supplier relationship management system for selecting and benchmarking suppliers*. International Journal of Technology Management. 2003. Vol. 26, No. 7. Pp. 717–742.
18. Закаляк Р.Ф. *Багатокритеріальний метод вибору постачальника хмарних послуг* : кваліф. робота (магістр) / Зах.-укр. нац. ун-т. – Тернопіль, 2022. – 96 с. [Електронний ресурс] // URL: <https://dspace.wunu.edu.ua/bitstream/316497/47174/1/%D0%97%D0%90%D0%9A%D0%90%D0%9B%D0%AF%D0%9A%20%D0%A0%D0%BE%D0%BC%D0%B0%D0%BD%20%D0%A4%D0%B5%D0%B4%D0>

%BE%D1%80%D0%BE%D0%B2%D0%B8%D1%87.pdf

(дата

звернення: 01.06.2025).

19. Місюра Н. В. Підсистема прийняття рішень в інформаційній системі управління взаємовідносинами з постачальниками // Інформаційні системи та мережі. – 2022. – № 2(35). – С. 91–97.
20. Шевчук В. Я., Палагін А. В. Моделювання бізнес-процесів підприємства з використанням BPMN-нотації // Управління розвитком складних систем. – 2020. – № 42. – С. 121–127.
21. Білик І. І., Ящук Т. В. Застосування WSM-методу у прийнятті рішень щодо вибору постачальника // Науковий вісник НЛТУ України. – 2021. – № 31(1). – С. 204–208.

ДОДАТОК А

Лістинг використання EntityFramework

```

using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Infrastructure;
using Microsoft.EntityFrameworkCore.Metadata;
using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
using TenderApp.Data;

#nullable disable

namespace TenderApp.Migrations
{
    [DbContext(typeof(TenderDbContext))]
    partial class TenderDbContextModelSnapshot : ModelSnapshot
    {
        protected override void BuildModel(ModelBuilder modelBuilder)
        {
#pragma warning disable 612, 618
            modelBuilder
                .HasAnnotation("ProductVersion", "9.0.4")
                .HasAnnotation("Relational:MaxIdentifierLength", 128);

            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

            modelBuilder.Entity("TenderApp.Models.Contract", b =>
            {
                b.Property<int>("Id")
                    .ValueGeneratedOnAdd()
                    .HasColumnType("int");

                b.Property<string>("Details")
                    .IsRequired()
                    .HasColumnType("nvarchar(max)");

                b.Property<int>("TenderId")
                    .HasColumnType("int");

                b.Property<int>("WinnerId")
                    .HasColumnType("int");

                b.HasKey("Id");

                b.HasIndex("TenderId");

                b.HasIndex("WinnerId");

                b.ToTable("Contracts");
            });
        }
    }
}

```

ДОДАТОК Б

Приклад згенерованого договору з постачальником

ДОГОВІР ПРОДАЖУ ТОВАРІВ

Місто Харків «30.05.2025» року

Цей Договір (далі – «Договір») укладено між:

Постачальник: Постачальник Приклад, надалі – «Постачальник»,

Покупець: Менеджер Категорійний, надалі – «Покупець».

1. ПРЕДМЕТ ДОГОВОРУ

1.1. Постачальник зобов'язується передати у власність Покупця товар: «Футболка» (номер тендера: 1 кількістю 100 одиниць та загальною вартістю 100000 грн, а Покупець зобов'язується прийняти і оплатити цей товар на умовах, передбачених цим Договором.

2. ЦІНА ТА ПОРЯДОК РОЗРАХУНКІВ

2.1. Загальна вартість товару за цим Договором становить 100000,00 грн без врахування ПДВ.

2.2. Оплата здійснюється Покупцем протягом 30 календарних днів з дати отримання рахунку-фактури.

3. УМОВИ ПОСТАВКИ ТОВАРУ

3.1. Постачальник здійснює поставку товару за адресою: вул. Сумська, буд. 123, кв. 45, м. Харків, Харківська обл., 61000

3.2. Строк постачання: 360 календарних днів з дати підписання Договору.

3.3. Товар передається за фактичним підписанням актів приймання-передачі.

4. ПРАВА ТА ОБОВ'ЯЗКИ СТОРІН

4.1. Постачальник зобов'язується передати якісний товар, що відповідає технічним характеристикам, передбаченим тендерною документацією.

4.2. Покупець зобов'язується прийняти товар і здійснити оплату на умовах цього Договору.

5. ВІДПОВІДАЛЬНІСТЬ СТОРІН

5.1. У разі несвоєчасного виконання своєї частини зобов'язань Сторона сплачує неустойку у розмірі 100 % від вартості недопоставленого товару за кожен день прострочення.

5.2. Сторони звільняються від відповідальності за невиконання зобов'язань у разі настання обставин непереборної сили.

6. ФОРС-МАЖОР

6.1. Сторони звільняються від відповідальності за часткове або повне невиконання зобов'язань за умов настання форс-мажорних обставин, якщо такі обставини унеможливили виконання Договору.

7. СТРОК ДІЇ ДОГОВОРУ ТА ПОРУШЕННЯ

7.1. Цей Договір набуває чинності з моменту його підписання Сторонами і діє до повного виконання зобов'язань.

7.2. Договір може бути розірваний достроково за взаємною згодою Сторін або в односторонньому порядку у випадках, передбачених законодавством.

8. РЕКВІЗИТИ ТА ПІДПИСИ СТОРІН

Постачальник	Покупець
«Постачальник Приклад» _____	«Менеджер Категорійний» _____

ДОДАТОК В

Лістинг обчислення алгоритму WSM

```

using Microsoft.EntityFrameworkCore;

using System.Diagnostics;
using System.IO;

using TenderApp.Data;
using TenderApp.Models;

using Xceed.Words.NET;

namespace TenderApp.Services
{
    public class ProposalService(TenderDbContext db)
        : DbService<Proposal>(db)
    {
        public IEnumerable<Proposal> GetByTenderId(int tenderId)
            => (_db.Proposals
                .Where(p => p.TenderId == tenderId)
                .Include(p => p.Tender)
                .Include(p => p.Byer)
                .Include(p => p.Values)
                .ThenInclude(v => v.TenderCriterion)
                .ThenInclude(tc => tc.Criterion))
            ;

        public IEnumerable<Proposal> GetByTenderIdBuyerId(int tenderId, int
byerId)
            => GetByTenderId(tenderId).Where(p => p.ByerId == byerId);

        public Proposal? CalculateBest(int tenderId)
            => GetByTenderId(tenderId)
                .MaxBy(p => p.Values.Sum(v =>
                    (v.Score ?? 0) * (v.TenderCriterion?.Weight ?? 0)));

        public Proposal? SelectWinner(int tenderId)
        {
            var proposals = GetByTenderId(tenderId);
            var best = proposals.MaxBy(p => p.Values.Sum(v =>
                (v.Score ?? 0) * (v.TenderCriterion?.Weight ?? 0)));

            foreach(var p in proposals)
                p.IsWinner = (p == best);

            _db.SaveChanges();

            return best;
        }

        public void CreateContracts(int tenderId)
        {
            var proposals = GetByTenderId(tenderId).Where(p =>
p.IsWinner).ToList();

            foreach(var p in proposals)
            {
                CreateContract(p);
            }
        }
    }
}

```

```

    }

    public void CreateContract(Proposal proposal)
    {
        Debug.WriteLine(proposal.Byer.Name);

        var fileName = $"Contract_{proposal.Id}.docx";
        var templatePath = @"..\..\..\Templates\ContractTemplate.docx";
        var outputPath = Path.Combine(@"..\..\..\Contracts", fileName);

        var doc = DocX.Load(templatePath);

        doc.ReplaceText("{Date}", DateTime.Now.ToShortDateString());
        doc.ReplaceText("{Buyer}", proposal?.Byer?.Name ?? " *** ");
        doc.ReplaceText("{Product}", proposal?.Tender?.Product ?? " *** ");
        doc.ReplaceText("{TenderId}", proposal?.Tender?.Id.ToString());
        doc.ReplaceText("{Quantity}", proposal?.Tender.Quantity.ToString() );
        doc.ReplaceText("{TotalPrice}", proposal?.Tender.Budget.ToString() );
        doc.ReplaceText("{UnitPrice}", " *** ");
        doc.ReplaceText("{Manager}", proposal?.Tender.CreatedBy.Name ?? " ***

    " );

        doc.SaveAs(outputPath);

        var contract = new Contract
        {
            ProposalId = proposal.Id,
            FilePath = outputPath,
            Details = $"Договір по тендеру \"{proposal.Tender.Name}\" от
{DateTime.Now:dd.MM.yyyy}"
        };

        _db.Contracts.Add(contract);
        _db.SaveChanges();
    }

    public override Proposal Clone(Proposal source)
    {
        return new()
        {
            Id = source.Id,
            IsWinner = source.IsWinner,
            Comment = source.Comment,

            TenderId = source.TenderId,
            Tender = source.Tender,

            ByerId = source.ByerId,
            Byer = source.Byer,

            Values = source.Values
        };
    }

    public override void CopyTo(Proposal source, Proposal target)
    {
        target.Id = source.Id;
        target.IsWinner = source.IsWinner;
        target.Comment = source.Comment;

        target.TenderId = source.TenderId;
        target.Tender = source.Tender;

        target.ByerId = source.ByerId;
    }

```

```
        target.Byer = source.Byer;
        target.Values = source.Values;
    }
    public override void Validate(Proposal item)
    {
        if(item.Tender is null)
            throw new ArgumentException("Proposal tender is not specified");

        if(item.Byer is null)
            throw new ArgumentException("Proposal participant is not
specified");

        foreach (var proposal in item.Values)
        {
            if(proposal.Value is null)
                throw new ArgumentException("Proposal values are not
specified");
        }
    }
}
```

ДОДАТОК Г

Лістинг генерації договору з постачальником

```

using CommunityToolkit.Mvvm.ComponentModel;

using System.Diagnostics;
using System.Windows;
using System.Windows.Data;

using TenderApp.Models;
using TenderApp.Services;
using TenderApp.Views;

namespace TenderApp.ViewModels
{
    public partial class ContractListViewModel
        : ListViewModel<Contract>
    {
        [ObservableProperty]
        Visibility _managerVisibility = Visibility.Visible;

        public ContractListViewModel(IDbService<Contract> service)
            : base(service)
        {
        }

        protected override void EditItem()
        {
            ((ContractService)_service).OpenContract(SelectedItem.Id);
        }

        protected override ItemViewModel<Contract>
        CreateItemViewModel(IDbService<Contract> service, Contract item)
            => new ContractItemViewModel((ContractService)service, item);

        protected override ItemViewModel<Contract>
        CreateItemViewModel(IDbService<Contract> service)
            => new ContractItemViewModel((ContractService)service);

        protected override Window CreateItemDialog(ItemViewModel<Contract>
        itemViewModel)
            => new ContractItemWindow(itemViewModel);
    }
}

```