

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Факультет інформаційних технологій
(факультет)

Кафедра системного аналізу та управління
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістра

Студента Сазанської Ірини Олегівни
академічної групи 124м – 23 – 1
спеціальності 124 Системний аналіз
на тему: «Бізнес-аналіз вимог до розробки порталу постачальників»

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	Інституційною	
кваліфікаційної роботи	<i>д.т.н., проф. Молоканова В.М.</i>			
розділів:				
Інформаційно- аналітичний	<i>д.т.н., проф. Молоканова В.М.</i>			
Спеціальний розділ	<i>д.т.н., проф. Молоканова В.М.</i>			
Рецензент	<i>к.т.н., доц. Алексєєв О.М.</i>			
Нормоконтролер	<i>к.ф.-м.н., доц. Хом'як Т.В.</i>			

Дніпро
2024

ЗАТВЕРДЖЕНО:
завідувач кафедри
Системного аналізу та управління
(повна назва)

_____ к.т.н., доц. Желдак Т.А.
(підпис) (прізвище, ініціали)

« ____ » _____ 20__ року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістра

студенту Сазанській І.О. академічної групи 124м -23-1
спеціальності: 124 Системний аналіз

на тему «Бізнес-аналіз вимог до розробки порталу постачальників»
затверджену наказом ректора НТУ «Дніпровська політехніка»
від 16.10.2024 р. №1388 – С

Розділ	Зміст	Терміни виконання
1. Інформаційно-аналітичний розділ	<i>Проаналізувати структуру об'єкта дослідження. Визначити предметну область дослідження та проблему, що розв'язується. Обґрунтувати методи виконання поставлених завдань.</i>	10.09.2024 – 01.11.2024
2. Спеціальний розділ	<i>Розглянути методи розробки вимог до створення корпоративної інформаційної системи. Розробити вимоги до проектування корпоративної системи взаємодії з постачальниками.</i>	01.11.2024 – 30.12.2024

Завдання видано _____ проф. Молоканова В.М.
(підпис) (прізвище, ініціали)

Дата видачі: 06.09.2024 р.

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____ Сазанська І.О.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 71 с., 15 рис., 3 табл., 3 додатки, 20 джерел.

Об'єктом дослідження є процеси формування вимог до створення корпоративної платформи управління постачальниками, яка автоматизує взаємодію між сторонами бізнес-процесів компанії.

Предмет дослідження – процес дослідження та оптимізація вимог до корпоративної платформи управління постачальниками.

Метою кваліфікаційної роботи є комплексне обґрунтування вимог до розробки корпоративної платформи управління постачальниками для оптимізації взаємодії між сторонами бізнес-процесів компанії.

Методи дослідження: методи системного аналізу – для формування концептуальної моделі бази даних, методи проектно-орієнтованого аналізу – для визначення потреб зацікавлених сторін, семантичних зв'язків між джерелами даних.

В *інформаційно-аналітичному розділі* досліджено ключові етапи розробки програмного забезпечення та методології, що дозволяють ефективно адаптувати процеси до потреб замовника та швидко реагувати на зміни в умовах бізнес-середовища.

У *спеціальному розділі* виконано бізнес-аналіз порталу для постачальників, здійснено моделювання бізнес-процесів, а також наведено результати впровадження та оцінки ефективності веб-сайту.

Практична цінність результатів кваліфікаційної роботи полягає в розробці корпоративної платформи, яка оптимізує комунікацію та бізнес-процеси між ритейл-компанією та її постачальниками, підвищуючи прозорість, ефективність та задоволеність співпрацею.

Ключові слова: БІЗНЕС-ПРОЦЕС, ВИМОГИ, ЗАЦІКАВЛЕНІ СТОРОНИ, РИТЕЙЛЕР, КОМЕРЦІЙНА ПРОПОЗИЦІЯ, ВЛАСНА ТОРГОВА МАРКА, ПРОЄКТУВАННЯ, БАЗА ДАНИХ, ІНТЕГРАЦІЯ, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, ВЕБ-САЙТ.

ABSTRACT

Explanatory note: 71 pages, 15 figures, 3 tables, 3 annexes, 20 sources.

The object of the study is the processes of requirement formation for the development of a corporate supplier management platform that automates the interaction between the parties involved in the company's business processes.

The subject of the study is the process of investigating and optimizing the requirements for the corporate supplier management platform.

The purpose of the work is to provide a comprehensive justification of the requirements for the development of a corporate supplier management platform to optimize the interaction between the parties involved in the company's business processes.

Research methods: system analysis methods – for developing the conceptual database model, project-oriented analysis methods – to define the stakeholders' needs and semantic relationships between data sources.

The informational and analytical section examines the key stages of software development and methodologies that effectively adapt processes to client needs and allow for quick responses to changes in the business environment.

The special section includes the business analysis of the supplier portal, business process modeling, as well as the results of the implementation and effectiveness evaluation of the website.

The practical value of the results lies in the development of a corporate platform that optimizes communication and business processes between the retail company and its suppliers, improving transparency, efficiency, and satisfaction with the collaboration.

Keywords: BUSINESS PROCESS, REQUIREMENTS, STAKEHOLDERS, RETAILER, COMMERCIAL PROPOSAL, PRIVATE LABEL, DESIGNING, DATABASE, INTEGRATION, USER INTERFACE, WEBSITE.

ЗМІСТ

ВСТУП	7
1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ	10
1.1 Ключові аспекти розробки програмного забезпечення	10
1.1.1 Життєвий цикл програмного забезпечення	10
1.1.2 Значення вимог та тестування у загальному циклі розробки.....	12
1.2 Технології розробки вимог до програмного забезпечення	13
1.2.1 Рівні та етапи розробки вимог	13
1.2.2 Методи збору та обробки вимог	16
1.2.3 Показники якості вимог.....	18
1.3 Сучасні стратегії розробки вимог та контролю якості	19
1.3.1 Стратегія розробки ПЗ на основі каскадної методології	21
1.3.2 Стратегія розробки ПЗ на основі гнучкої методології.....	23
1.4 Висновки до розділу	27
2 СПЕЦІАЛЬНИЙ РОЗДІЛ	30
2.1 Визначення цілей та критеріїв успішності проєкту	30
2.1.1 Мета розробки порталу постачальників	31
2.1.2 Метрики успішності проєкту	31
2.2 Збір та обробка вимог до порталу постачальників	33
2.2.1 Методи дослідження зацікавлених сторін.....	35
2.2.2 Моделювання БП подачі КП постачальниками товарних категорій ...	40
2.3 Проєктування системи	54
2.3.1 Архітектура порталу постачальників.....	54
2.3.2 Проєктування бази даних	58
2.3.3 Інформаційна архітектура сайту.....	61

2.4 Результати впровадження та оцінка ефективності порталу	65
2.5 Висновки до розділу	66
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70
ДОДАТКИ.....	Ошибка! Закладка не определена.
Додаток А. Відомість матеріалів кваліфікаційної роботи.....	72
Додаток Б. Відгук на кваліфікаційну роботу магістра	73
Додаток В. Рецензія на кваліфікаційну роботу магістра.....	74

ВСТУП

В умовах сучасного бізнесу ефективна взаємодія між постачальниками та роздрібними мережами є ключовим чинником для забезпечення високої конкурентоспроможності та досягнення успіху. Проте, в існуючих умовах, багато підприємств стикаються з низкою проблем, що ускладнюють цей процес: затримки в розгляді пропозицій, неефективна комунікація між сторонами та відсутність прозорості в бізнес-процесах взаємодії компаній.

За результатами незалежного опитування серед постачальників, проведеного організацією Advantage, було виявлено низький рівень задоволеності постачальників умовами співпраці з компанією, що вказує на необхідність удосконалення комунікаційних і бізнес-процесів. Ці проблеми не лише впливають на ефективність роботи з постачальниками, але й можуть негативно позначитися на загальній ефективності компанії. Як відомо, в практиці розробки ПЗ іноді недооцінюються такі критично важливі стадії, як попередній збір та аналіз вимог, проєктування системи, а також забезпечення та контроль якості. Для вирішення зазначених проблем роботи з постачальниками компанія планує реалізувати два ключових напрями:

1. Підвищення зручності і прозорості бізнес-процесів при подачі комерційних пропозицій.
2. Забезпечення легкого доступу до необхідного персоналу компанії.

Створення корпоративної платформи постачальників дозволить консолідувати всю наявну в організації інформацію, в якій би формі вона не зберігалася і організувати її у вигляді системи, з якою зручно взаємодіяти, яка є гнучкою і змістовною.

Об'єктом дослідження є процеси формування вимог до створення корпоративної платформи управління постачальниками, яка автоматизує взаємодію між сторонами бізнес-процесів компанії.

Предмет дослідження – процес дослідження та оптимізація вимог до корпоративної платформи управління постачальниками.

Метою кваліфікаційної роботи є комплексне обґрунтування вимог до розробки корпоративної платформи управління постачальниками для оптимізації взаємодії між сторонами бізнес-процесів компанії.

Методи дослідження: методи системного аналізу – для формування концептуальної моделі бази даних, методи проектно-орієнтованого аналізу – для визначення потреб зацікавлених сторін, семантичних зв'язків між джерелами даних.

В рамках цього проєкту передбачається створення веб-платформи для постачальників, яка включатиме функціональність для подачі комерційних пропозицій, швидкого вирішення питань через прямий доступ до відповідальних осіб, а також спрощення процесів реєстрації та інтеграції нових постачальників. Ці зміни сприятимуть значному покращенню рівня задоволеності постачальників співпрацею з компанією.

Успішність реалізації проєкту буде оцінюватися за допомогою ключових показників ефективності (KPI), таких як рівень лояльності постачальників, а також результативність комунікацій, що вимірюватиметься через подальші опитування постачальників, що проводяться організацією Advantage.

Інформаційною базою дослідження стали законодавчі та нормативно-правові акти України, монографічна та періодична проблемно-орієнтована література, Інтернет-публікації, навчальна література з системного аналізу, маркетингу, менеджменту та економіки підприємств.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

API	Application Programming Interface
REST	Representational State Transfer
HTTP	Hypertext Transfer Protocol
UML	Unified Modeling Language
BPM	Business Process Management, управління бізнес-процесами
UI	User Interface, користувацький інтерфейс
UX	User Experience, користувацький досвід
KPI	Key Performance Indicators, ключові показники ефективності
SDLC	Software Development Life Cycle, життєвий цикл програмного забезпечення
MVP	Minimum Viable Product, мінімально життєздатний продукт
ПЗ	Програмне забезпечення
БД	База даних
БП	Бізнес-процес
ВТМ	Власна торгова марка
КП	Комерційна пропозиція
АК	Асортиментний комітет
КМ	Категорійний менеджер, менеджер із закупівлі

1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ

1.1 Ключові аспекти розробки програмного забезпечення

Розробка програмного забезпечення охоплює кілька важливих етапів, кожен з яких має своє значення для успішного створення якісного продукту. Цей розділ зосереджений на ключових аспектах, таких як життєвий цикл програмного забезпечення, розробка вимог та їх вплив на весь процес, а також роль тестування як важливого елемента для досягнення високої якості. Розуміння цих аспектів є основою для ефективної розробки програмних продуктів, що відповідають потребам користувачів.

1.1.1 Життєвий цикл програмного забезпечення

Стратегії формування вимог і процеси контролю якості програмного забезпечення дуже пов'язані з іншими етапами роботи над проектом через що не можуть існувати окремо. Виконання проекту охоплює тривалий проміжок часу та являє собою життєвий цикл розробки програмного забезпечення. SDLC – це послідовність етапів, які включають розробку, тестування та внесення змін, необхідних для успішного втілення проекту, починаючи від ідеї до виходу готового продукту на ринок і його подальшої підтримки [1].

На рисунку 1.1 зображено основні етапи життєвого циклу розробки програмного забезпечення:

1. Збір та аналіз вимог – збирання інформації про специфіку проекту, узагальнення отриманих даних і створення документації, яка слугує основою для визначення пріоритетів завдань.

2. планування – узгодження деталей, визначення мети, ідентифікація проблем і формування бачення та змісту майбутнього продукту;

3. проектування та дизайн – розробка архітектури системи, організація взаємодії клієнтської частини із сервером, створення користувацького інтерфейсу;

4. розробка – безпосереднє написання коду на основі документації, прототипів, дизайну та архітектури;

5. тестування – перевірка відповідності створеного продукту вимогам, визначеним на етапі збору й аналізу, а також виявлення помилок і дефектів;

6. впровадження – передача готового програмного забезпечення кінцевим користувачам;

7. підтримка та розвиток – регулярний моніторинг працездатності системи, вдосконалення її функціоналу та виправлення помилок.

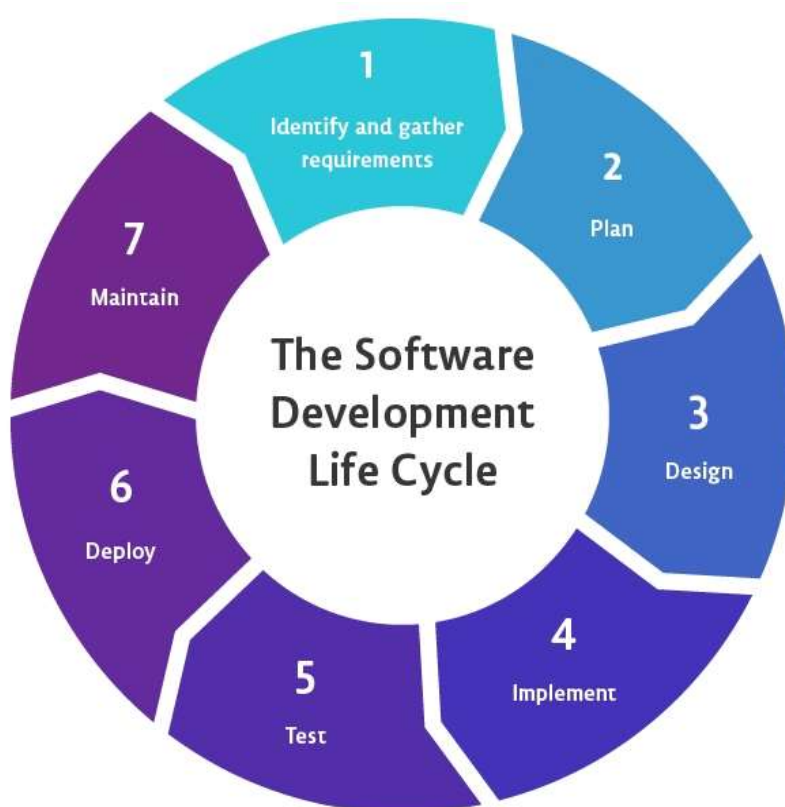


Рисунок 1.1 – Етапи життєвого циклу розробки ПЗ

Використання життєвого циклу розробки ПЗ дозволяє обрати оптимальні підходи для кожного етапу, що підвищує ефективність виконання проєкту.

1.1.2 Значення вимог та тестування у загальному циклі розробки

Тривалість кожного етапу проєкту залежить від обраної методології, а також від масштабу, складності й типу розробки [2]. Точна тривалість визначається під час детального планування.

Всупереч поширеній думці, центральне місце у процесі реалізації проєкту не завжди займає написання коду. Часто недооцінюються такі критично важливі стадії, як попередній збір та аналіз вимог, проєктування системи, а також забезпечення та контроль якості. Останні мають вирішальне значення не лише в процесі розробки, але й після виходу продукту на ринок.

Згідно зі статистичними даними, зібраними групою The Standish Group International у 1994–2000 роках у різних країнах [3], ключові фактори успіху і провалу проєктів мають наступні причинно-наслідкові зв'язки:

1. Причини провалів:

- недостатня інформація від клієнта;
- неповнота вимог і специфікацій;
- часті зміни вимог і специфікацій.

2. Причини успіхів:

- залучення користувача до процесу розробки;
- тестування на кожному етапі;
- чітка постановка вимог.

Отже, можна зробити висновок, що збір вимог є важливим етапом на початку життєвого циклу проєкта. Це підкреслює необхідність ретельного вивчення інженерії вимог.

Так само забезпечення та контроль якості ПЗ відіграють ключову роль протягом усього життєвого циклу розробки. Ефективна система якості дозволяє оперативно виявляти й усувати помилки, забезпечуючи створення робочого та якісного продукту, який є головним пріоритетом для замовника.

1.2 Технології розробки вимог до програмного забезпечення

Розробка вимог до програмного забезпечення є важливим етапом, який визначає основу для створення ефективного та надійного продукту. Використання сучасних технологій та методів дозволяє забезпечити точність, повноту та узгодженість вимог, що в свою чергу сприяє успішному виконанню проєкту. Важливим аспектом є правильне визначення потреб користувачів, що вимагає застосування різноманітних інструментів для збору та аналізу інформації, а також чіткого формулювання вимог для подальшої реалізації.

1.2.1 Рівні та етапи розробки вимог

У зв'язку з різноманітністю типів інформації про вимоги виникає необхідність у використанні єдиного набору визначень, що дозволяє уточнити і систематизувати термін "вимогу". У цьому розділі наводяться визначення ключових термінів, які будуть використовуватися для опису понять, що найчастіше зустрічаються у сфері розробки вимог, наведених у таблиці 1.1.

Таблиця 1.1

Ключові типи вимог

Концепція	Визначення
-1-	-2-
Бізнес вимога	Високорівнева бізнес-мета організації або замовників системи
Бізнес правило	Політика, розпорядження, стандарт чи правило, що визначає чи обмежує деякі сторони бізнес-процесів. За своєю суттю це не вимога до ПЗ, але воно є джерелом кількох типів вимог до ПЗ
Обмеження	Обмеження на вибір варіантів, доступних розробнику при проєктуванні та розробці продукту
Вимога до інтерфейсу	Опис взаємодії між ПЗ та користувачем, іншою програмною системою або пристроєм
Характеристика	Одна або кілька логічно пов'язаних можливостей системи, які представляють цінність для користувача та описані серією функціональних вимог
Функціональна вимога	Опис необхідної поведінки системи у певних умовах
Нефункціональна вимога	Опис властивості або особливості, яким повинна володіти система, або обмеження, якого повинна дотримуватися система
Атрибут якості	Вид нефункціональної вимоги, що описує характеристику сервісу чи продуктивності продукту

Продовження табл. 1.1

Системна вимога	Вимога верхнього рівня до продукту, що складається з багатьох підсистем, які можуть бути ПЗ або сукупність ПЗ та обладнання
Користувацька вимога	Завдання, яке певні класи користувачів повинні мати можливість виконувати в системі, або потрібний атрибут продукту

Вимоги до програмного забезпечення складаються з трьох рівнів – бізнес-вимоги, користувацькі та функціональні вимоги. До того ж у кожній системі є свої нефункціональні вимоги. Модель рисунку 1.2 ілюструє спосіб представлення цих типів вимог як узагальненої схематичної організації.

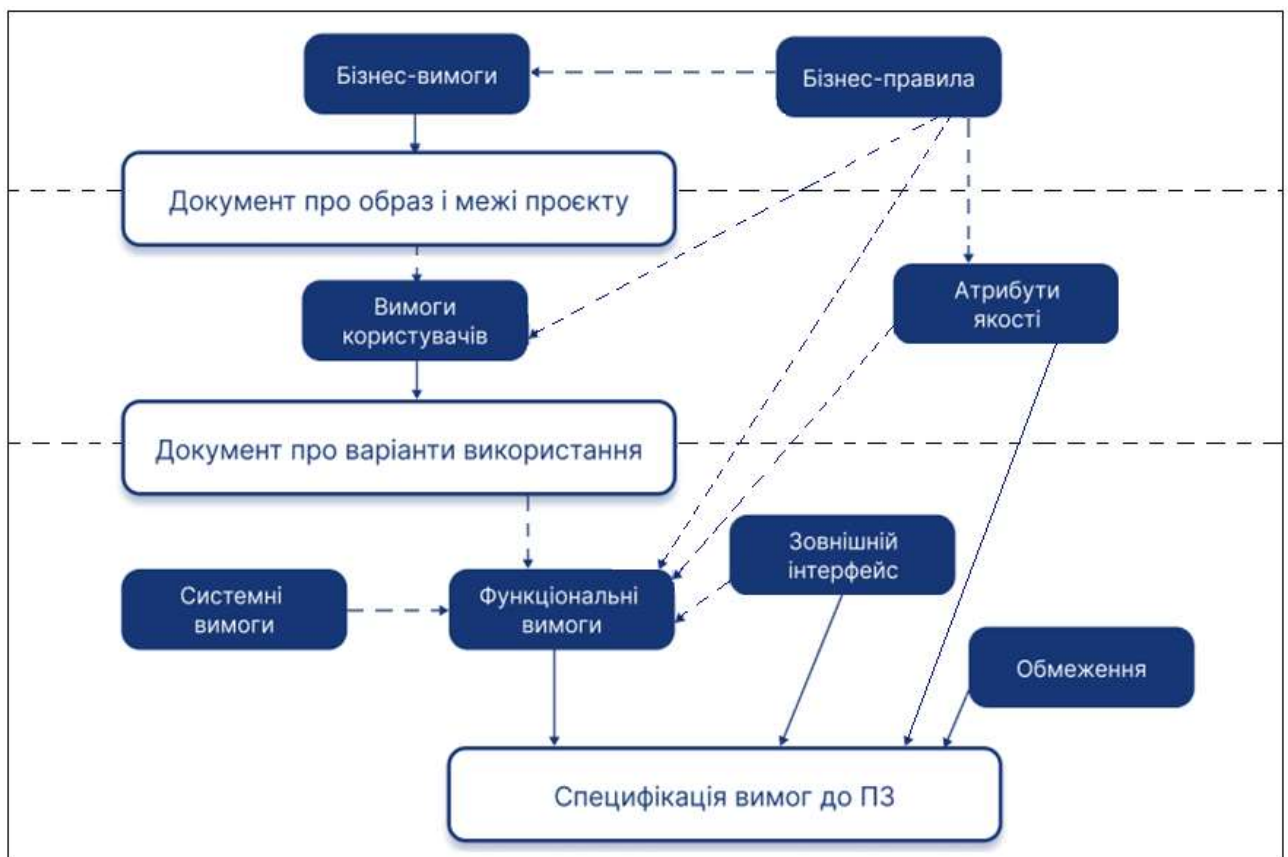


Рисунок 1.2 – Зв'язок типів інформації та способів їхнього зберігання для вимог

Зафарбовані прямокутники на схемі позначають типи інформації вимог, а прямокутники без фону – документи, у яких ця інформація зберігається. Суцільні лінії показують, що документ містить інформацію певного типу. Бізнес-правила та системні вимоги зберігаються окремо від вимог до програмного забезпечення, зазвичай у документах, таких як каталог бізнес-

правил або специфікація системних вимог. Пунктирні лінії вказують, що інформація одного типу впливає на вимогу або на інформацію іншого типу.

На рисунку 1.3 зображено робота над вимогами, що поділяється на етапи розробки та управління [4]. Їхня розробка є частиною процесу розробки технічних вимог, який складається з таких етапів:

1. Виявлення – визначення потреб користувачів та зацікавлених сторін.
2. Аналіз – оцінка отриманих даних для створення повної картини.
3. Документування – оформлення вимог у стандартизованому вигляді.
4. Перевірка – підтвердження правильності та повноти вимог.



Рисунок 1.3 – Складові інженерії вимог

Основні процеси розробки вимог приведені до наступних складових:

1. Визначення цільової аудиторії продукту – ідентифікація груп користувачів, для яких розробляється ПЗ.
2. З'ясування потреб, цілей та завдань кожного класу користувачів – аналіз очікувань і потреб різних категорій користувачів.
3. Встановлення бізнес-цілей та бізнес-правил – узгодження ключових завдань, які повинен вирішувати продукт, відповідно до стратегічних цілей організації.
4. Аналіз інформації та класифікація вимог – виділення функціональних вимог і відокремлення їх від нефункціональних.

5. З'ясування атрибутів якості – визначення параметрів, які впливають на якість системи (надійність, продуктивність, безпека тощо).

6. Документування вимог, побудова моделей і встановлення пріоритетів – створення формалізованих документів і моделей, які відображають вимоги, та визначення їх пріоритетності для подальшої розробки.

7. Опрацювання та деталізація специфікації вимог – завершення і уточнення всіх вимог у специфікації (SRS).

Основні процеси управління вимогами мають наступний цикл:

1. Визначення основної версії вимог – формування базової версії вимог, яка слугує основою для подальшої роботи.

2. Розгляд пропонованих змін – оцінка змін у вимогах, які виникають у процесі реалізації проєкту.

3. Включення схвалених змін у проєкт – внесення змін до вимог і їх інтеграція у плани розробки.

4. Узгодження плану проєкту з вимогами – забезпечення відповідності між вимогами та етапами життєвого циклу проєкту.

5. Спостереження за статусом вимог упродовж усього проєкту – моніторинг і контроль актуальності та виконання вимог на всіх етапах життєвого циклу.

Ці процеси забезпечують якісну розробку, управління і підтримку вимог, що дозволяє ефективно координувати реалізацію проєкту.

1.2.2 Методи збору та обробки вимог

Процес розробки вимог включає в себе дослідження, розуміння та документування тих вимог, які являється критичними для успішної реалізації проєкту. Якість вимог на етапі збору визначає подальшу ефективність розробки та впливає на успіх проєкту загалом.

Найпоширеніші методи збору вимог до програмного забезпечення:

1. Структуроване інтерв'ю зацікавлених осіб – спілкування з основними зацікавленими сторонами для виявлення їх потреб і вимог.

2. Пасивне та активне спостереження – спостереження за діяльністю користувачів або робочими процесами для виявлення вимог.

3. Перегляд та аналіз документів, наданих зацікавленими сторонами – вивчення наявних документів для збору інформації, яка допоможе визначити вимоги.

4. Фокус-групи та групові інтерв'ю з представниками класів користувачів, що представляє собою організацію групових обговорень для збору детальної інформації про потреби різних категорій користувачів.

5. Опитування та анкетування – використання структурованих інструментів для збору кількісних даних.

6. Прототипування – створення базових моделей продукту для збору зворотного зв'язку і уточнення вимог.

Після виявлення вимог, вони узгоджуються між собою і перевіряються на відповідність певним критеріям, таким як повнота, коректність, незмінність і відсутність суперечностей.

Процес узгодження та перевірки вимог:

1. Чорновий варіант документа – на цьому етапі створюється первинний документ, в якому містяться всі виявлені вимоги.

2. Обговорення і модифікація вимог – вимоги аналізуються, уточнюються, змінюються або видаляються, якщо вони не відповідають вимогам проекту.

3. Специфікація вимог – після обробки та уточнення вимог, вони описуються у специфікації, яка є основою для подальших етапів розробки.

Використання специфікації вимог:

1. Завдання для розробників – на основі вимог створюються завдання для команди розробників, зазвичай у вигляді історій користувачів (User Stories) і критеріїв приймання (Acceptance Criteria).

Історії користувачів описують функціональні вимоги до системи у вигляді коротких, чітких наративів про те, що повинен зробити користувач та чого він очікує від системи.

Критерії приймання визначають умови, які повинні бути виконані, щоб вважати історію користувача завершеною.

2. Документи для подальших етапів – опрацьовані вимоги стають основою для створення дизайну, прототипів, технічного завдання, а також документів, що стосуються тестування (план тестування, тест-кейси тощо).

Розроблені вимоги є основою для створення всіх інших артефактів, що необхідні для реалізації ПЗ, що робить цей етап критично важливим для успішної реалізації проєкту.

1.2.3 Показники якості вимог

Вимоги до програмного забезпечення відіграють критично важливу роль у забезпеченні його якості. Однак, враховуючи, що вимоги не завжди документуються, ключовим критерієм їхньої якості є сама їх наявність. Як окремі вимоги, так і їх сукупність повинні відповідати визначеним критеріям якості. Чітко сформульовані та належним чином описані вимоги мають спільні характеристики, які сприяють підвищенню ефективності та якості розробки проєкту [5]. Серед таких характеристик:

1. Коректність: вимоги є точними і правильними, виключають можливість неоднозначного трактування.
2. Зрозумілість: легко сприймаються і інтерпретуються розробниками та іншими учасниками проєкту.
3. Повнота: охоплюють усі необхідні функції та властивості програмного забезпечення.
4. Атомарність: кожна вимога є цілісною одиницею, яку не можна поділити без втрати суті.
5. Керованість: вимоги можуть змінюватися відповідно до потреб зацікавлених сторін.
6. Перевірюваність: вимоги піддаються перевірці на предмет виконання.

Недотримання цих характеристик часто призводить до низки проблем, зокрема:

1. Неповнота: відсутність опису всіх необхідних функцій та властивостей для задоволення потреб користувачів.
2. Неоднозначність: формулювання вимог допускає різне тлумачення, що спричиняє хибне розуміння функціоналу.
3. Суперечливість: вимоги конфліктують одна з одною або не узгоджуються;
4. Неактуальність: вимоги стають застарілими через зміни у версії програмного забезпечення чи платформі.
5. Низька якість документації: вимоги задокументовані фрагментарно або несистематизовано, що ускладнює їх розуміння.

Для забезпечення високої якості вимог важливо встановлювати стандарти та оцінювати прогрес їх дотримання. Метрики, що використовуються для оцінки, можуть змінюватися залежно від проекту, але до основних належать:

1. Рівень коректності, зрозумілості та повноти вимог у специфікації.
2. Кількість чітко описаних вимог до окремих компонентів системи.
3. Легкість визначення критеріїв прийняття на основі документації.
4. Кількість запитів на зміну вимог, поданих після завершення їх початкового опису.

1.3 Сучасні стратегії розробки вимог та контролю якості

Сучасні стратегії розробки вимог та контролю якості зосереджені на забезпеченні високої ефективності та точності на всіх етапах розробки програмного забезпечення. Вони включають інтеграцію гнучких підходів, які дозволяють швидко адаптуватися до змінюваних вимог та умов проєкту, а також акцент на постійному контролі якості, що є ключовим для досягнення високих стандартів продукту [6]. Ці стратегії допомагають знижувати ризики, підвищувати задоволеність кінцевих користувачів та забезпечують відповідність програмного забезпечення бізнес-цілям.

Стратегія — це довгостроковий план дій, спрямований на досягнення конкретної мети. У контексті розробки ПЗ вона є умовним планом, що

складається з набору принципів, методів, документів і артефактів, які визначають процес розробки вимог та забезпечення якості. Ця стратегія базується на методології життєвого циклу ПЗ. Її основна мета — створення продукту високої якості, який відповідає вимогам зацікавлених сторін і мінімізує кількість помилок та дефектів.

Основні складові стратегії:

1. Рівні вимог: визначення рівнів деталізації вимог залежно від специфіки продукту.
2. Методи розробки вимог: використання відповідних технік, таких як інтерв'ю, опитування, побудова моделей тощо.
3. Рівні та типи тестування: включення функціонального, нефункціонального, регресійного та інших типів тестування.
4. Тестові документи та артефакти: розробка тест-планів, чек-листів, тест-кейсів, матриць простежуваності тощо.
5. Порядок роботи з трекером завдань: встановлення процесу ведення і відстеження задач для ефективного управління дефектами.
6. Критерії початку та завершення тестування: чітке визначення умов, за яких тестування розпочинається і завершується.
7. Оцінка якості проекту: використання метрик для оцінки прогресу, виявлення проблем і визначення ефективності.

Кожен проєкт має унікальні особливості, тому вибір стратегії повинен бути обґрунтованим, спиратися на аналіз та враховувати:

- специфіку продукту;
- потреби бізнесу та зацікавлених осіб;
- очікування кінцевих користувачів.

Для досягнення найкращих результатів стратегія повинна поєднувати перевірені методи з адаптацією до конкретних умов проєкту. Хоча не існує універсальної стратегії, придатної для всіх організацій, багато популярних підходів є ефективними в різних контекстах. відповідність програмного забезпечення бізнес-цілям.

1.3.1 Стратегія розробки ПЗ на основі каскадної методології

Каскадна (водоспадна) методологія є класичним підходом до розробки програмного забезпечення, що базується на чітко визначених етапах, кожен з яких має бути завершений перед початком наступного [7]. Вона отримала свою назву через те, що процес розробки нагадує спад водоспаду, де кожен етап передбачає завершення попереднього, і немає можливості повернутися до попереднього етапу без значних зусиль.

Каскадна модель має кілька особливостей:

1. Чітка послідовність етапів – кожен етап має бути завершений перед тим, як перейти до наступного.
2. Важливість вимог – успіх всього процесу залежить від коректно розроблених вимог. Якщо на етапі збору вимог були допущені помилки, це може вплинути на весь цикл розробки.
3. Тестування на останньому етапі – водоспадна модель передбачає тестування після завершення етапів розробки та проектування. Це може призводити до виявлення багатьох помилок на пізніх етапах.



Рисунок 1.4 – Стратегія розробки ПЗ на основі каскадної моделі

Сучасні адаптації водоспадної моделі іноді дозволяють повернення до попередніх етапів, коли потрібно внести корективи або уточнення в вимоги чи дизайн. Однак, водоспадна модель все одно залишається більш статичною в

порівнянні з гнучкими методологіями, такими як Agile [8], де допускається більше ітерацій і змін протягом усього циклу розробки.

У водоспадній методології розробки програмного забезпечення можна виділити кілька ключових особливостей стратегії розробки вимог та контролю якості:

1. Збір та детальне описання вимог: всі вимоги до програмного забезпечення збираються на початковому етапі, оцінюються і описуються в найбільш детальному вигляді. Вони зазвичай оформлюються у вигляді специфікації, яка перевіряється на правильність і повноту, а потім затверджується замовником. Цей документ стає основою для подальших етапів розробки і тестування.
2. Мінімальне залучення замовника на наступних етапах: після збору вимог та їх затвердження, замовник або користувачі не залучаються до подальших етапів розробки до моменту завершення всіх технічних та тестових фаз.
3. Чітке визначення цілей, термінів та вартості: визначаються конкретні цілі та завдання проєкту, на початку також розраховуються строки та вартість. Ці параметри зазвичай не змінюються в процесі реалізації проєкту, що може створити певні труднощі, якщо з'являються нові вимоги або зміни.
4. Статичний процес тестування: тестування проводиться відповідно до вимог, зібраних і зафіксованих на початку, без значних змін у процесі розробки. Тест-кейси, тестові сценарії та чеклисти складаються до початку етапу тестування, що дозволяє забезпечити чітке й систематичне виконання тестів.
5. Мінімізація змін вимог після етапу аналізу: однією з основних особливостей є мінімізація змін вимог після того, як фаза аналізу та збору вимог завершена. Це зменшує ризик невизначеності та спрощує контроль за проєктом, хоча і може обмежити гнучкість, якщо нові вимоги з'являються під час розробки.

6. Описані правила та документація: для всіх етапів розробки та тестування існують чітко визначені правила і документація, включаючи специфікації, тестові стратегії, тест-плани та інші необхідні артефакти. Це дозволяє забезпечити стандартизацію процесу та належний контроль за його виконанням.
7. Тестування після завершення розробки: тестування в водоспадній моделі часто проводиться як окрема фаза після завершення розробки. Це може призвести до того, що помилки будуть виявлені на пізніх етапах, що може збільшити витрати на виправлення дефектів. Проте виявлені проблеми реєструються й оперативно передаються команді розробників для виправлення.
8. Пріоритезація дефектів: усі виявлені дефекти реєструються в баг-трекерах і пріоритезуються в процесі тестування, що дозволяє ефективно розподіляти ресурси для їх виправлення.

Таким чином, водоспадна методологія надає велику вагу документуванню та чіткому виконанню кожного етапу, що забезпечує контроль якості на кожній фазі розробки, хоча і може бути менш гнучкою при необхідності внесення змін до вимог або проекту в цілому.

1.3.2 Стратегія розробки ПЗ на основі гнучкої методології

Scrum є однією з найпоширеніших гнучких методологій для розробки програмного забезпечення, що орієнтується на швидкі, короткі ітерації та регулярні відгуки від замовника. Як зазначено в статті *Harvard Business Review* 1986 року, ключовою концепцією є підхід, подібний до гри в регбі, де команди активно взаємодіють і передають «м'яч» між собою, а не працюють за етапами поетапного планування, як це було характерно для традиційного підходу.

Основні принципи Scrum:

1. Пряма комунікація між командою та замовником: це дозволяє краще розуміти потреби замовника, миттєво реагувати на зміни і сприяє більш тісній співпраці.

2. Розробка від загального до конкретних функцій: в Scrum перевага віддається спочатку загальному розумінню потреб замовника, а потім деталям, що дозволяє зосередитись на пріоритетних завданнях і найважливіших функціональних можливостях.
3. Інкрементність та безперервна інтеграція: проєкт розбивається на маленькі частини, які постійно розвиваються і інтегруються в продукт, що дозволяє швидше виявляти та виправляти дефекти і проблеми.
4. Крос-функціональні команди: команди складаються з різних спеціалістів, що дозволяє кожному учаснику працювати над усіма аспектами розробки продукту, від кодування до тестування, і безперешкодно співпрацювати в межах однієї команди.
5. Розуміння бізнес-контексту: для успішної реалізації проєкту команда повинна чітко розуміти бізнес-цілі і контекст, в якому буде використовуватися продукт. Це дозволяє приймати більш ефективні рішення і забезпечує більш високу якість розробки.

Оскільки Scrum передбачає активну співпрацю команди з зацікавленими сторонами і постійну ітерацію розробки, тестування починається вже на перших етапах і триває на кожній ітерації. Важливим аспектом є також включення тестування на ранніх етапах і в процесі всього циклу розробки, що допомагає виявляти дефекти ще до того, як вони стануть критичними, та дозволяє оперативно коригувати їх.



Рисунок 1.5 – Стратегія на основі гнучкої методології
Переваги Scrum:

1. Швидка адаптація до змін: завдяки коротким ітераціям та регулярним зустрічам з замовниками, команда може швидко реагувати на зміни в вимогах або технічних аспектах проєкту.
2. Безперервна інтеграція та тестування: це дозволяє виявляти дефекти на ранніх етапах і швидко їх виправляти, забезпечуючи постійну якість продукту.
3. Активна співпраця між членами команди і замовником: дозволяє досягти кращого розуміння кінцевих потреб і спільно створювати цінний продукт для бізнесу.

Завдяки інкрементальній та гнучкій природі Scrum, він є потужним інструментом для розробки ПЗ, який дозволяє створювати високоякісні продукти в мінливому середовищі.

У рамках методології Scrum стратегії [9] розробки вимог і контролю якості значно відрізняються від традиційних підходів. Основні характеристики цієї стратегії полягають у гнучкості, постійному оновленні вимог та інтеграції тестування на всіх етапах розробки.

Характерні особливості стратегії розробки вимог та контролю якості за методологією Scrum:

1. Зміна та доповнення вимог: вимоги постійно уточнюються та змінюються протягом усього життєвого циклу проєкту. Це дозволяє команді адаптуватися до змін у бізнесі чи потребах замовника, що є важливим елементом гнучкості Scrum.
2. Залучення зацікавлених осіб: зацікавлені особи активно залучені на кожному етапі розробки: вони надають необхідну інформацію, приймають важливі рішення і можуть брати участь в процесі розробки. Це забезпечує високу орієнтованість на потреби користувачів та замовника.
3. Пріоритезація завдань: завдання в Scrum пріоритезуються за важливістю для бізнесу. Найперше опрацьовуються найбільш критичні або цінні для замовника функції, що забезпечує швидке отримання корисного продукту.
4. Ітеративний підхід до документації: документація, зокрема специфікація, дизайн та прототипи, розробляються поетапно. Вони постійно уточнюються і доповнюються, а дошка проєкту заповнюється завданнями, які описуються через історії користувачів та критерії прийняття.
5. Робота з тестовою стратегією та тест-планом: тестування в Scrum є постійним процесом. Тестова стратегія та тест-план розробляються й коригуються протягом всього циклу розробки, але вони не перевищують важливість робочого продукту та задоволення потреб замовника.
6. Модульне, інтеграційне та системне тестування: тестування починається з модульного і інтеграційного тестування, а системне тестування проводиться після завершення попередніх етапів. Це дозволяє мінімізувати помилки та дефекти на ранніх стадіях розробки.
7. Паралельна розробка тест-кейсів та чек-лістів: чек-лісти та тест-кейси створюються паралельно з розробкою нових функцій. Це забезпечує високий рівень тестування в реальному часі, одночасно з розробкою ПЗ.

8. Регресійне тестування: регресійне тестування є важливою частиною кожної ітерації і проводиться після кожної зміни в ПЗ. Це допомагає забезпечити стабільність і сумісність нового коду з попередніми версіями.
9. Безперервне виправлення дефектів: виявлені дефекти оперативно фіксуються в баг-репортах, і відповідні помилки виправляються за пріоритетністю. Це дозволяє швидко реагувати на проблеми та мінімізувати їхній вплив на кінцевий продукт.
10. Підсумковий звіт після кожної ітерації: після кожної ітерації складається підсумковий звіт про тестування, що дозволяє команді оцінити досягнутий прогрес і визначити завдання на наступний період.

Особливості документації та контролю якості в Scrum:

В Scrum документація не є основним пріоритетом, і її обсяг зменшується порівняно з класичними методами розробки. Натомість акцент ставиться на постійну взаємодію з користувачами, регулярні перевірки та тестування. Докладне документування вимог відбувається поступово: початково визначаються загальні бізнес-потреби, потім формуються історії користувачів та критерії їх прийняття. Цей процес повторюється з кожною новою ітерацією розробки.

Контроль якості також реалізується на кожному етапі розробки і забезпечує безперервний прогрес, адже продукт має бути готовий до випуску в будь-який момент. Це дозволяє оперативно реагувати на зміни і коригувати функціональність ПЗ відповідно до вимог замовника та кінцевих користувачів.

1.4 Висновки до розділу

В інформаційно-аналітичному розділі було детально розглянуто низку ключових аспектів, що впливають на процес розробки програмного забезпечення. Приділено особливу увагу життєвому циклу програмного забезпечення, який є основою для побудови всієї структури розробки. Життєвий цикл включає послідовність етапів, які повинні бути виконані для

створення продукту, починаючи з визначення вимог і закінчуючи тестуванням і впровадженням програмного забезпечення. Важливою частиною цього процесу є визначення вимог та тестування, адже правильно сформульовані вимоги є запорукою успішної розробки, а тестування дозволяє виявити можливі дефекти на ранніх етапах і забезпечити якість кінцевого продукту.

Розглянуті різні рівні та етапи розробки вимог, а також методи збору і обробки вимог, що дозволяють максимально точно відобразити потреби користувачів та вимоги замовників. Важливим аспектом також є оцінка якості вимог, оскільки на їхній основі будується вся подальша робота над проектом. Визначено ключові показники якості вимог, що дозволяють оцінити їх повноту, ясність і узгодженість, що, у свою чергу, сприяє зменшенню ризиків у процесі розробки.

Важливим елементом є розробки програмного забезпечення також огляд сучасних методологій розробки вимог та контролю якості програмного забезпечення, оскільки в умовах швидких змін в технологіях та підходах до розробки, необхідно постійно адаптувати процеси розробки ПЗ до нових реалій. У цьому контексті розглянуто дві основні методології: каскадну та гнучку. Каскадна методологія передбачає чітку послідовність етапів, де кожен етап є основою для наступного, що дозволяє досягти стабільності, але іноді зменшує гнучкість. Натомість гнучка методологія дозволяє здійснювати розробку в більш динамічному режимі, що особливо важливо при роботі з проектами, де вимоги можуть змінюватися на кожному етапі розробки.

Завершив аналіз ключових аспектів розробки програмного забезпечення, таких як життєвий цикл, розробка вимог та контролю якості, можна зазначити, що успішна реалізація цих принципів є основою для створення ефективних бізнес-інструментів. Зокрема, для ритейлерів, які прагнуть оптимізувати взаємодію з постачальниками, де впровадження спеціалізованих порталів стає важливим кроком у досягненні стратегічних цілей.

Портал постачальника, розроблений відповідно до специфічних вимог компанії, дозволяє автоматизувати ключові процеси, підвищити ефективність

комунікацій та забезпечити високу прозорість у взаємодії з партнерами. Таким чином, використання сучасних методологій та технологій розробки ПЗ, які розглянуті у цьому розділі, є основою для створення інструментів, що відповідають потребам ритейлерів і забезпечують успіх на ринку.

2 СПЕЦІАЛЬНИЙ РОЗДІЛ

2.1 Визначення цілей та критеріїв успішності проєкту

Попередній розділ кваліфікаційної роботи включав аналіз наукової літератури та технічної документації, що є необхідними для розробки функціональних вимог до системи. У цьому розділі буде розглянута проєктна частина реалізації.

Задля того, щоб портал постачальників відповідав усім вимогам замовника та забезпечив зручність й інтуїтивно зрозумілий інтерфейс, доцільним буде застосування методів бізнес-аналізу.

Бізнес-аналіз — діяльність, яка уможливорює проведення змін у організації, які приносять користь зацікавленим сторонам, шляхом виявлення потреб та обґрунтування рішень, що описують можливі шляхи реалізації змін. Основне завдання бізнес-аналізу — зробити можливим проведення змін в організації шляхом реалізації обраного рішення. Рішення розробляється з метою усунення виявлених у процесі бізнес-аналізу бізнес-проблем. Поняття «рішення» включає широкий діапазон можливих шляхів усунення виявлених бізнес-проблем: розробка нових або зміна існуючих бізнес-процесів або бізнес-правил, оптимізація організаційної структури організації, розробка нових стратегічних планів організації тощо.

Портал повинен стати інструментом для оптимізації бізнес-процесів, надання зручного доступу до необхідного персоналу компанії та полегшення комунікацій. Реалізація цього рішення дозволить не лише покращити поточні умови співпраці, але й забезпечить постачальникам зручні інструменти для вирішення питань, що виникають, та прискорить процеси прийняття рішень. В результаті очікується, що це сприятиме значному підвищенню рівня задоволеності та лояльності стратегічних партнерів, а також забезпечить сталий розвиток довгострокових і взаємовигідних відносин.

2.1.1 Мета розробки порталу постачальників

Метою проєкту є розробка веб-сайту, який забезпечить ефективну взаємодію між ритейлером та постачальниками в рамках програми лояльності. За результатами незалежного опитування, проведеного організацією Advantage, було виявлено низький рівень задоволеності постачальників поточними процесами взаємодії. Проєкт ініційований комерційним департаментом з метою оптимізації процесів та зміцнення довгострокових відносин із постачальниками через впровадження прозорих і ефективних комунікаційних каналів.

Основні бізнес-цілі проєкту:

1. Метою є підвищення лояльності постачальників через впровадження прозорості та ефективної системи комунікації, що дозволить побудувати міцніші партнерські відносини та забезпечити взаємну вигоду для обох сторін.

2. Постачальники зможуть подавати свої пропозиції ритейлеру через онлайн-форму, прискорюючи процес розгляду та вибору товарів.

3. Система включатиме функціональність для подання питань на певні теми та отримання відповідей від призначених співробітників ритейлера, що підвищить ефективність комунікації та прискорить вирішення проблем постачальників.

4. Платформа повинна включати функціональність для спрощення та прискорення процесу реєстрації та інтеграції нових постачальників, що забезпечить зручність і швидкість взаємодії.

2.1.2 Метрики успішності проєкту

Для оцінки успішності реалізації проєкту було визначено KPI, які допоможуть оцінити ефективність роботи веб-платформи як з погляду користувачів, так і бізнесу:

1. Метрики для оцінки функціональності

- час обробки комерційної пропозицій: середній час, що минає від подання комерційної пропозиції до представлення пропозиції менеджером закупівлі на асортиментному комітеті;

- кількість поданих пропозицій: кількість пропозицій, поданих постачальниками через платформу за певний період;

- відсоток вчасно вирішених питань через систему довідника комунікації: частка питань, які вирішуються при прямій взаємодії зі співробітниками ритейлера.

2. Метрики для оцінки досвіду користувача

- задоволеність користувачів: оцінка досвіду використання платформи користувачами через опитування та відгуки;

- час, витрачений користувачами виконання ключових операцій: середній час, який користувач витрачає виконання завдань, як-от подання пропозиції, заповнення профілю чи пошуку відповідної теми для створення питання в довіднику комунікації.

3. Метрики для оцінки продуктивності системи

- час відгуку сторінки: час, який сайт витрачає на завантаження сторінок, а також на виконання запитів користувача;

- продуктивність системи при високих навантаженнях: здатність сайту витримувати велику кількість одночасних користувачів без значних втрат у швидкості роботи.

4. Метрики для оцінки бізнес-процесів

- відсоток успішних онбордінгів: частка нових постачальників, які успішно пройшли реєстрацію та успішно інтегровані в систему;

- час введення товару до асортименту: середній час, що минає від подання заявки на комерційну пропозицію до внесення товару до асортименту мережі ритейлера.

Ці метрики є показниками того, наскільки вправно веб-сайт вирішує завдання, для яких був створений. Вони дозволяють не лише відслідковувати

успіх у реалізації функціоналу, а й оцінювати, наскільки проєкт задовольняє бізнес-потреби та вимоги користувачів.

2.2 Збір та обробка вимог до порталу постачальників

Аналіз вимог полягає в визначенні потреб та умов, які висуваються щодо нового, чи зміненого продукту, враховуючи можливо конфліктні вимоги різних замовників, таких як користувачі чи бенефіціари. Вимоги мають бути задокументованими, вимірними, тестовними, пов'язаними з бізнес-потребами, і описаними з рівнем деталізації достатнім для конструювання системи. Вимоги можуть бути архітектурними, структурними, поведінковими, функціональними, та нефункціональними. Це дає змогу зрозуміти, які функції та характеристики повинна мати система, щоб задовольнити різноманітні вимоги користувачів і стейкхолдерів.

У цьому контексті важливим інструментом є Use Case діаграма [10], яка є частиною сімейства UML (Unified Modeling Language), є потужним інструментом для опису взаємодії між користувачами (акторами) та системою. Вона дозволяє наочно представити основні функції системи з точки зору кінцевих користувачів, що робить її важливим інструментом на етапах аналізу та проєктування [11]. Основне призначення цієї діаграми — визначити, які саме функції система повинна реалізовувати, щоб задовольнити потреби користувачів.

У контексті корпоративного порталу для постачальників, Use Case діаграма, представлена на рисунку 2.1, допомагає чітко визначити, як постачальники взаємодіють із системою при подачі комерційних пропозицій, реєстрації товарів або комунікації з представниками компанії. За допомогою цієї діаграми можна ідентифікувати основні сценарії використання порталу, зокрема, для різних ролей користувачів — таких як постачальники, адміністратори або менеджери з закупівель.

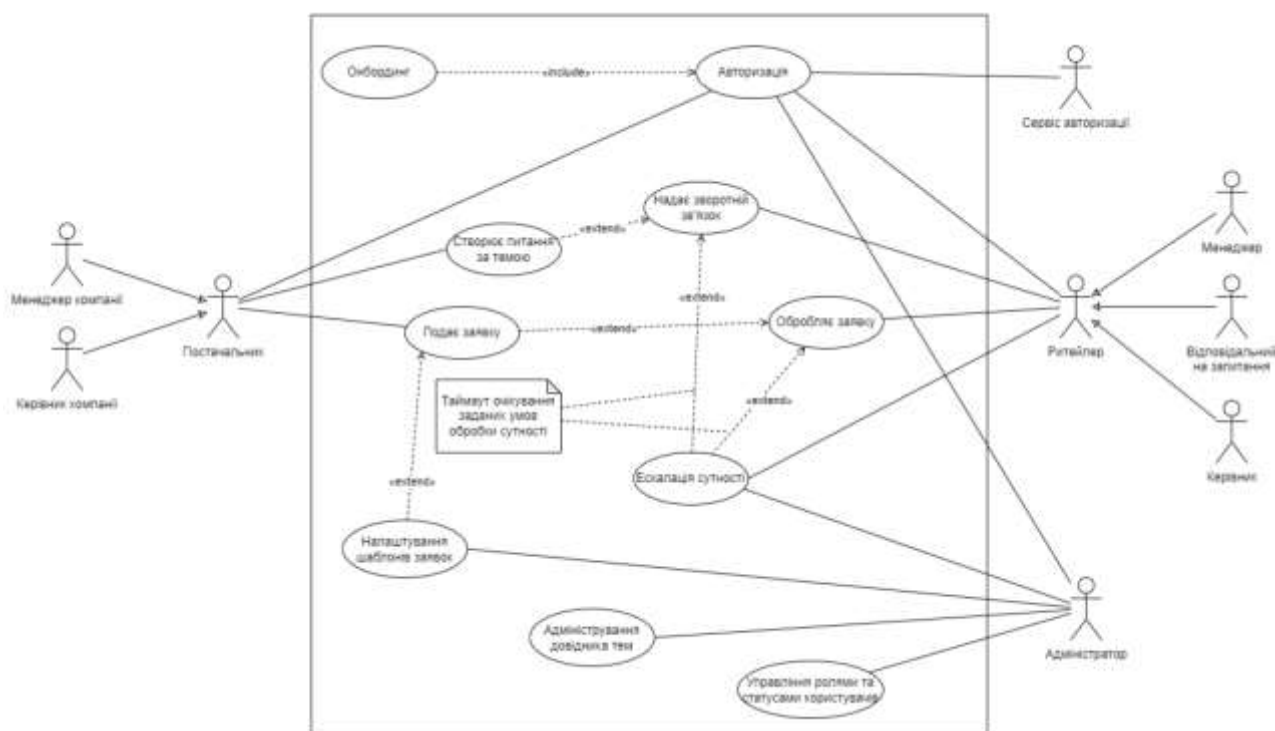


Рисунок 2.1 – Use case діаграма порталу постачальників

Таким чином, Use Case діаграма дозволяє створити зрозумілу картину процесів і функцій, що мають бути реалізовані в рамках системи, і забезпечує чітке бачення того, як кожен актор взаємодіє з порталами і яким чином ці взаємодії допомагають оптимізувати роботу між постачальниками та компанією. Це дозволяє не лише визначити функціональні вимоги, а й зрозуміти ключові сценарії використання системи, що є основою для подальшої розробки та реалізації проекту.

Вимоги до програмного забезпечення — набір вимог щодо властивостей, якості та функцій програмного забезпечення, що буде розроблено, або знаходиться у розробці. Вимоги визначаються в процесі аналізу вимог та фіксуються в специфікації вимог, діаграмах прецедентів та інших артефактах процесу аналізу та розробки вимог. Для забезпечення успішної реалізації проекту необхідно не лише визначити вимоги, а й розуміти, хто і як буде взаємодіяти з системою. Це потребує детального аналізу стейкхолдерів та побудови чіткої моделі бізнес-процесів, які будуть підтримувати подальший розвиток програмного забезпечення.

2.2.1 Методи дослідження зацікавлених сторін

Карта зацікавлених сторін — це візуальний процес розміщення всіх зацікавлених сторін продукту, проекту чи ідеї на одній карті. Основна перевага карти зацікавлених сторін полягає в тому, щоб отримати візуальне представлення всіх людей, які можуть вплинути на ваш проект, і те, як вони пов'язані [12]. Іноді люди плутають зацікавлених сторін з акціонерами. Хоча акціонери володіють часткою публічної компанії (через акції) і зацікавлені в ефективності компанії, це не означає, що вони повинні бути зацікавленими сторонами кожного проекту чи продукту, запущеного компанією. Зацікавлені сторони можуть працювати на більш детальному рівні, і вони також часто зацікавлені в ефективності проекту чи продукту, не лише тому, що це впливає на ефективність акцій компанії.

Чотири кроки до створення карти зацікавлених сторін:

1. мозковий штурм: визначення всіх потенційних стейкхолдерів;
2. категоризація: групування визначених зацікавлених сторін;
3. розстановка пріоритетів, ієрархія;
4. комунікації із зацікавленими сторонами.

Нижче було розроблено наступну карту зацікавлених осіб, що показане на рисунку 2.2 у вигляді ієрархії прямих клієнтів, користувачів та зацікавлених осіб.

Карта стейкхолдерів проекту:

1. Ключові стейкхолдери (High Power, High Interest) — це стейкхолдери, які мають високий інтерес та можуть сильно впливати на проект. Вони визначають стратегію та приймають ключові рішення:

- комерційний директор (Product Owner) — основний стейкхолдер, який керує проектом, приймає рішення про комерційну частину, відповідає за стратегію та бізнес-результати;

- бренд-менеджери відповідають за узгодження дизайну та брендингу, що важливо для створення правильного іміджу продукту;

- керівники підрозділів (логістика, бухгалтерія, операційний департамент)
— ці люди зазвичай приймають рішення, які можуть суттєво вплинути на виконання проекту. Їхні завдання включають забезпечення ефективної роботи в рамках їх сфер відповідальності.



Рисунок 2.2 – Карта зацікавлених сторін

2. Гарні союзники (High Power, Low Interest) — це ті учасники, які можуть впливати на проект, але їхній щоденний інтерес до нього невеликий. Вони важливі для прийняття стратегічних рішень, але не залучені до повсякденного управління:

- CEO (Chief Executive Officer) — генеральний директор компанії може мати високий рівень впливу, оскільки він визначає загальну стратегію бізнесу та рішення з ключових питань. Однак його інтерес до проекту може бути більш стратегічним та обмеженим, якщо проект не стосується безпосередньо всієї компанії. Його підтримка або схвалення є критичними для успішного

виконання проекту, особливо якщо проект пов'язаний з великими комерційними або бізнес-ініціативами;

- CIO (Chief Information Officer) — директор з інформаційних технологій, який контролює технологічну інфраструктуру компанії. Якщо проект вимагає впровадження нових технологій або змін в IT-інфраструктурі, CIO може вплинути. Але його інтерес може бути обмежений лише з погляду технологічної реалізації, якщо проект не торкається головних IT-ініціатив компанії. Може мати важливий вплив для прийняття рішень щодо інфраструктурних та технічних питань, але може не бути залученим до щоденних операцій;

- CFO (Chief Financial Officer) — фінансовий директор стежить за бюджетами та фінансовими результатами проекту. Він може брати участь у проекті на етапі затвердження бюджету і у випадках, коли потрібно забезпечити фінансове забезпечення, але у повсякденній роботі не буде сильно залучено. Має високий вплив на фінанси проекту, схвалення бюджету та контроль за фінансовими ризиками.

3. Ті, хто потребує інформування (Low Power, High Interest) — ці стейкхолдери зацікавлені у проекті, але мають вирішального впливу прийняття рішень. Вони активно взаємодіють у межах своєї ролі, і їхня думка важлива для успіху проекту:

- прями користувачі: постачальники та менеджери закупівель - вони відіграють ключову роль у процесі реалізації проекту, оскільки їхня робота безпосередньо пов'язана із КП, його введенням та висновком. Постачальники, у свою чергу, зацікавлені в оптимальному процесі постачання, а менеджери закупівель — у тому, щоб товар надходив вчасно та з необхідною якістю. Їхній інтерес полягає у стабільності та ефективності бізнес-процесів, пов'язаних з продуктом, і вони беруть активну участь у процесі;

- співробітники з інших функціональних підрозділів (наприклад, логістика, бухгалтерія, операційний департамент) — хоча вони й не можуть

впливати на проект у стратегічному плані, вони зацікавлені у його успіху, оскільки їхні щоденні завдання залежать від успішної реалізації операцій.

4. Мінімальний вплив (Low Power, Low Interest) — ці стейкхолдери мають низький вплив та обмежений інтерес до проекту, але можуть бути залучені до його реалізації у непрямій чи допоміжній якості:

- Support клієнтів/користувачів - служба підтримки клієнтів та користувачів забезпечує допомогу користувачам продукту, але їх вплив на проект обмежений. Ці співробітники можуть бути залучені, якщо продукт або проект викликають питання чи проблеми кінцевих користувачів;

- співробітники маркетингу можуть бути залучені до проекту у разі, якщо проект потребує промо-матеріалів, реклами чи інших маркетингових активностей. Однак їхня роль обмежена в рамках проекту, та їх інтерес буде скоріше непрямим, якщо проект не пов'язаний безпосередньо з маркетингом;

- юридичний відділ може бути залучений до проекту для забезпечення юридичної підтримки, перевірки контрактів, дотримання норм та стандартів, захисту інтелектуальної власності та ін.

Були визначені ролі користувачів у рамках двох основних категорій: постачальники та співробітники ритейлу, які безпосередньо та регулярно залучені до забезпечення життєдіяльності та правильного функціонування програмного забезпечення та системи.

Для аналізу стейкхолдерів була застосована методологія, що передбачає виділення основних учасників проекту з визначенням їхніх вигід, очікувань, інтересів та обмежень. В таблиці 2.1 були зафіксовані ключові аспекти, що допомогли наочно представити взаємодії та ролі зацікавлених сторін у проекті.

З цієї таблиці бачимо, що опитування та інтерв'ю користувачів та зацікавлених сторін допомогли зібрати усю необхідну вхідну інформацію для побудови ієрархії клієнтів, користувачів та зацікавлених осіб, а також визначити їх вигоди, очікування інтереси та обмеження. Це, у свою чергу, дозволить на етапі проектування чітко врахувати ці фактори для розробки функцій веб-сайту, що найбільш ефективно задовольняють потреби усіх

учасників системи. У результаті, такі дослідження сприяють створенню системи, яка максимально відповідає вимогам користувачів та бізнес-цілям компанії.

Таблиця 2.1

Аналіз стейкхолдерів: вигоди, інтереси та обмеження

Зацікавлена сторона	Вигода	Очікування	Інтереси	Обмеження
-1-	-2-	-3-	-4-	-5-
Постачальник	Зрозумілі та чіткі умови співпраці, можливість моніторингу статусу своїх пропозицій	Підтримка своєчасних поставок, чіткість вимог до виконання контракту, можливість отримати зворотний зв'язок	Збільшення обсягів замовлень, дотримання термінів виконання замовлень, прозорість взаємодії ритейлером, забезпечення ефективної комунікації	Технічні обмеження системи, необхідність відповідати вимогам платформи
Категорійний менеджер	Оптимізація оцінки та відбору постачальників, ефективне управління строками та виконанням КП	Швидкий доступ до даних про постачальників та комерційних пропозицій, товарів конкурентів	Здатність своєчасно оцінювати пропозиції постачальників, можливість порівняння постачальників за ціною та умовами	Потрібно дотримуватись специфікацій, обмеження в бюджеті, недостатня прозорість від постачальників
Керівник категорійного менеджера	Контроль за своєчасністю виконання КП, своєчасне реагування на ескалації	Вчасне отримання інформації про можливі проблеми у виконанні КП	Забезпечення виконання закупок вчасно, зменшення ризиків затримок у постачаннях, своєчасне реагування на ескалації	Обмеження у можливості керувати великою кількістю ескалацій, необхідність гнучко реагувати на зміни ситуації
Відповідальний за надання відповідей на питання	Забезпечення зручності комунікації, покращення	Швидкість обробки запитів постачальників,	Мінімізація затримок у відповіді на питання,	Високе навантаження на обробку запитів, необхідність

внутрішні відділи сторони ритейлу	зі зворотного зв'язку з постачальниками	доступність інформації	надання чіткої інформації	своєчасного оновлення довідника
-----------------------------------	---	------------------------	---------------------------	---------------------------------

Продовження табл. 2.1

Адміністратор графіка асортиментних комітетів	Покращення контролю за асортиментом, гнучкість у коригуванні графіків та узгодженості із постачальниками	Легкість у плануванні та коригуванні графіків комітетів, ефективно управління термінами	Своєчасне отримання даних від постачальників, узгодження термінів поставок	Необхідність погодження з іншими відділами, зміни в планах можуть бути обмежені
Адміністратор порталу	Керування статусами користувачів, ролями, гнучкість в налаштуванні полів форм та забезпечення прозорості в управлінні платформою	Налаштування та управління доступом для різних користувачів, гнучкість у формуванні форм, ефективна організація роботи порталу	Підтримка безперебійної роботи порталу, налаштування користувацьких ролей, контроль за інтерфейсом	Технічні обмеження системи, необхідність постійних оновлень для адаптації до нових вимог

2.2.2 Моделювання БП подачі КП постачальниками товарних категорій

Моделювання бізнес-процесів є ключовим інструментом для візуалізації, аналізу та оптимізації різних етапів роботи з постачальниками та внутрішніми департаментами компанії. У контексті порталу постачальника бізнес-процес подання комерційної пропозиції описує послідовність дій, які мають бути виконані постачальником та внутрішніми співробітниками для того, щоб заявку на постачання товару чи послуги було коректно оформлено, погоджено та прийнято. Це важливий процес для ефективного функціонування компанії і має бути чітко визначено його структуру, роль кожного учасника та етапи взаємодії.

Рисунок 2.3 ілюструє нотацію BPM [13], яка дозволяє наочно уявити взаємодію між учасниками процесу та логіку виконання КП постачальників товарних категорій. Впровадження моделі бізнес-процесу дозволяє не тільки підвищити прозорість та прискорити взаємодію, а й оптимізувати етапи узгодження, мінімізуючи ризики та покращуючи комунікацію.

Бізнес-процес подання КП постачальниками товарних категорій від подання заявки до введення товару в асортимент мережі складається з декількох етапів:

1. Подання заявки постачальником

Постачальник подає заявку на включення свого товару в асортимент мережі. У заявці зазначаються основні параметри товару: його характеристики, ціна, обсяги поставок, умови доставки та інше.

2. Розгляд заявки ритейлером

Відповідальні фахівці ритейлера (наприклад, менеджери із закупівель або категорій товарів) починають розглядати заявку. На цьому етапі може оцінюватись відповідність товару асортиментної стратегії мережі, його затребуваність, відповідність критеріям якості та ціновим сегментам.

3. Комерційні та юридичні переговори

Якщо товар цікавий ритейлеру, то починається переговорний процес. Сюди входить обговорення комерційних умов, таких як ціни, знижки, участь у промо-акціях, умови оплати, логістика, а також юридичні аспекти (договірні відносини, відповідальність сторін, строки постачання, штрафні санкції тощо).

4. Відділ якості за ритейлера проводить перевірку товару, запропонованого постачальником. Ці перевірки є важливим і невід'ємним елементом взаємодії з постачальниками, особливо коли йдеться про товари, які повинні відповідати суворим стандартам безпеки, якості та нормативам мережі.

5. Узгодження упаковки та дизайну (для продукції ВТМ напрямку)

Після того, як товар «пройшов» за основними комерційними параметрами, починається робота з упаковки та дизайну. У цей момент

постачальник та ритейлер спільно працюють над упаковкою, щоб вона відповідала стандартам ритейлера (та маркетингової концепції BTM).

Важливим моментом тут є те, що упаковка може бути змінена залежно від того, як товар виглядатиме на полиці, як сприйматиметься покупцем, і як вона відповідатиме спільній «лінії» бренду мережі.

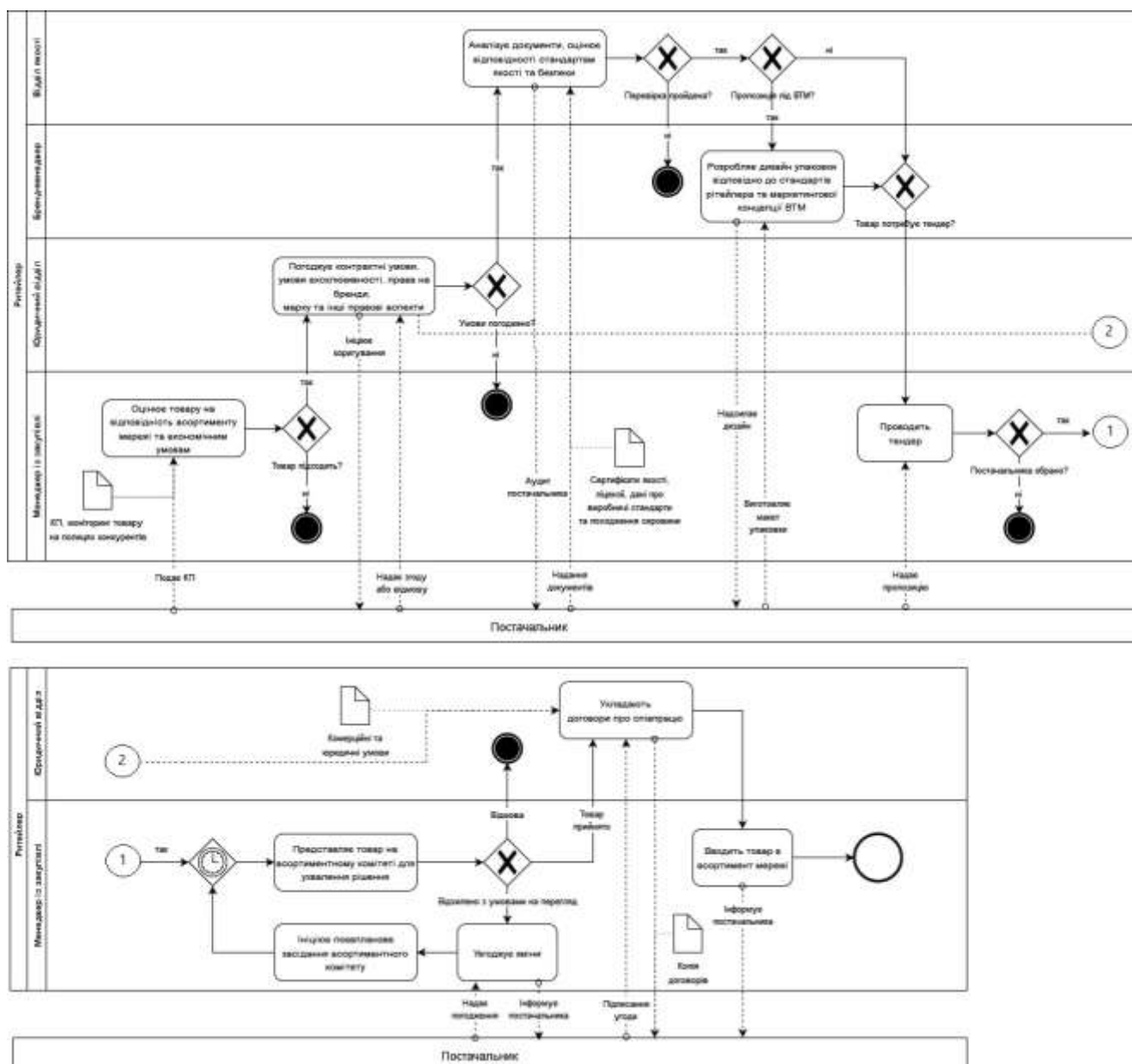


Рисунок 2.3 – BPMN бізнес-процесу подачі КП постачальником товарних категорій

6. Проведення тендеру (необов'язковий етап)

Якщо товар вимагає тендера (наприклад, для великих закупівель або специфічних категорій товарів), ритейлер може організувати тендер. Постачальники подають свої пропозиції, і на основі запропонованих умов вибирається найбільш вдалий варіант.

7. Розгляд на асортиментному комітеті

Після того, як комерційні та юридичні умови були погоджені, пропозиція постачальника може бути представлена на асортиментний комітет (або його аналог), де обговорюється, наскільки товар відповідає потребам цільової аудиторії та стратегії мережі.

Асортиментний комітет включає ключових осіб із різних відділів (категорійних менеджерів, фахівців з логістики, маркетологів, генерального директора, директора комерційної дирекції та електронної комерції та інших), та рішення щодо включення товару до асортименту приймається колегіально.

8. Укладання договору

Якщо товар був схвалений, укладається остаточний договір між постачальником та ритейлером. Договір включає всі раніше узгоджені умови (ціну, обсяги постачання, терміни, умови повернення товару тощо).

9. Введення в асортимент та постачання

Після підписання договору товар вноситься до асортименту мережі, і починається процес його постачання. Також відбувається розміщення товару на полицях (або електронної версії каталогу для інтернет-магазинів).

В рамках розробки системи порталу постачальника основним завданням є чітке визначення та формалізація функціональних вимог до системи, які забезпечать успішну роботу процесу подання та обробки комерційних пропозицій. Ці вимоги були зібрані на основі аналізу бізнес-процесів, представлених у діаграмі BPMN, а також у тісній взаємодії із замовником для уточнення всіх деталей та потреб.

Процес подання комерційної пропозиції починається з подання заявки користувачем та завершується затвердженням або відхиленням комерційної пропозиції. Усі етапи, що включають завантаження документів, перевірку

даних, зміну статусу заявки та повідомлення користувача про хід процесу, є критичними для успішної роботи системи. Для їх реалізації потрібна продумана структура функціональних вимог, що охоплює всі стадії взаємодії із системою.

У зв'язку з цим для кожної стадії бізнес-процесу були сформульовані конкретні функціональні вимоги, що включають як дії користувачів, так і механізми, які має підтримувати система. Частина цих вимог, зібраних від замовників, була обрана як приклад і систематизована в таблиці 2.2, яка буде використовуватися в процесі розробки та реалізації системи.

Кожна вимога документації має кілька ключових атрибутів [14], які допомагають детально описати його характеристики та впорядкувати процес розробки. Ці атрибути можна запам'ятати за допомогою абрєвіатури CARA'S SOUPS, де кожна літера означає певний аспект вимоги:

C — Complexity (Складність): Оцінка того, наскільки легко чи складно реалізувати вимогу, що допомагає визначити ресурси та зусилля, необхідні для її виконання.

A — Absolute Reference (Абсолютне посилання): Це посилання на джерело, звідки виникла вимога, наприклад, інший документ чи законодавство.

R — Risks (Ризики): Можливі непередбачувані події, які можуть вплинути на вимогу, зробити її більш складною, терміновою або змінити її пріоритет.

A — Author (Автор): Особа або група, які сформулювали вимогу та відповідають за її уточнення. Це той, до кого слід звертатися з запитаннями чи роз'ясненнями.

S — Source (Джерело): Походження вимоги, наприклад, внутрішні бізнес-потреби або нормативні документи.

S — Status (Статус): Поточний стан вимоги в процесі його життєвого циклу (наприклад, Proposed - запропоновано, Implemented - реалізовано).

O — Ownership (Власник): Особа або група, яка використовуватиме результат реалізації вимоги і отримуватиме від цього вигоду.

U — Urgency (Терміновість): Тимчасові рамки для реалізації вимог, які можуть впливати на його пріоритет у проекті.

P — Priority (Пріоритет): Відносна важливість вимоги в контексті проекту. Вимоги з вищим пріоритетом мають бути виконані раніше.

S — Stability (Стабільність): Рівень мінливості вимоги. Вимоги, які часто змінюються, можуть знизити пріоритет, щоб уникнути зайвих витрат на зміни.

Таблиця 2.2

Таблиця вимог до заявки на подання КП постачальника товарних категорій

Вимога	Опис	Автор	Джерело	Власник	Складність	Пріоритет	Терміновість	Стабільність	Ризики	Статус
-1-	-2-	-3-	-4-	-5-	-6-	-7-	-8-	-9-	-10-	-11-
Заявка може містити товари лише за однаковою категорією та групою	В одній заявці можуть бути зазначені товари лише однієї категорії та групи.	Бондар О.М.	Бізнес-правило	Відділ закупівель	Середня	Високий	Висока	Стабільне	-	Accepted
Редагувати дані заявки та товарів може лише постачальник	Постачальник має право редагувати тільки свої дані та товари в заявці.	Бондар О.М.	Функціональні вимоги	Відділ закупівель	Середня	Середній	Висока	Стабільне	-	Accepted
Постачальник може додавати товари через шаблон	Постачальник може додавати товари через шаблон, якщо товарів ще немає в заявці. Для повторного внесення товарів потрібно попередньо видалити всі товари в заявці.	Штиба Д.В.	Функціональні вимоги	Постачальники	Висока	Високий	Середня	Мінливе	Можливість помилок при видаленні товарів	Proposed

Продовження табл. 2.2

-1-	-2-	-3-	-4-	-5-	-6-	-7-	-8-	-9-	-10-	-11-
Не обмежуємо кількість товарів в одній заявці	В одній заявці не існує обмежень щодо кількості товарів.	Бондар О.М.	Бізнес-потреби	Відділ закупівель	Низька	Середній	Середня	Стабільне	-	Accepted
Перевірка дублікатів товарів	Перевірка на дублікати товарів за штрих-кодом.	Штиба Д.В.	Функціональні вимоги	Постачальники, КМ	Середня	Середній	Низька	Стабільне	Можливість невідповідності штрих-кодів	Proposed
Функціонал для завантаження шаблону товарів	Користувачі можуть завантажити заповнений шаблон, який містить один або більше товарів, і система автоматично перетворює його в перелік товарів у заявці.	Штиба Д.В.	Функціональні вимоги	Постачальники	Висока	Високий	Середня	Мінливе	Можливі проблеми із форматом шаблону	Proposed
Додавання товарів по одному	Користувач може додавати товари по одному через заповнення полів у відповідній формі на сайті.	Бондар О.М.	Функціональні вимоги	Відділ закупівель	Середня	Середній	Висока	Стабільне	-	Accepted

Продовження табл. 2.2

-1-	-2-	-3-	-4-	-5-	-6-	-7-	-8-	-9-	-10-	-11-
КМ може прийняти або відхилити товар	КМ може прийняти або відхилити товар незалежно від поточного статусу, коли заявка перебуває у нього в роботі.	Бондар О.М.	Бізнес-потреби	Відділ закупівель	Середня	Середній	Висока	Стабільне	-	Accepted
Редагування товару постачальником	Постачальник може редагувати товар при створенні або доопрацюванні заявки. Після доопрацювання товар повертається зі статусом "відредаговано" з історією змін.	Бондар О.М.	Функціональні вимоги	Відділ закупівель, КМ	Середня	Середній	Висока	Мінливе	Можливість зміни даних товару	Accepted
Комунікація через чат по заявці	Всі учасники процесу можуть комунікувати через чат, який працює по всій заявці. Вкладення можуть бути додані як до заявки, так і до товару в заявці.	Лисенко М.С.	Функціональні вимоги	Постачальники, КМ	Висока	Високий	Висока	Стабільне	-	Verified
Автоматичне заповнення специфікації	Система повинна автоматично заповнювати специфікацію для прийнятих товарів у заявці. Дія доступна автору та виконавцю заявки.	Бондар О.М.	Функціональні вимоги	Відділ закупівель, КМ	Висока	Середній	Низька	Мінливе	Можливість невідповідності даних	Verified

Продовження табл. 2.2

-1-	-2-	-3-	-4-	-5-	-6-	-7-	-8-	-9-	-10-	-11-
Делегування заявки при розподілі	При автоматичному розподілі заявки має бути опція «Не моя група або категорія, відхилити». Заявка через неї направляється на керівника того КМ, який відхилив. Керівники можуть делегувати на відповідального КМ: підлеглого співробітника або іншого керівника.	Лисенко М.С.	Бізнес-потреби	КМ, керівники, директор	Середня	Високий	Висока	Мінливе	Можливі помилки при делегуванні	Accepted
Підтримка завантаження вкладень у популярних форматах з обмеженням розміру файлів.	Можливість завантаження файлів у форматах pptx pdf, xls, rar/zip, doc/docx, png/jpeg. При цьому розмір кожного файлу, що завантажується, не повинен перевищувати 16 Мб. Сумарно вкладення за заявкою не повинні перевищувати 50 Мб.	Бондар О.М.	Системні вимоги	Відділ закупівель	Висока	Середній	Середня	Стабільне	-	Accepted

процесів, яка забезпечить прозорість і зручність для всіх учасників. На основі запропонованих етапів створення та обробки заявки можна виділити два основні процеси.

Перший процес – це створення заявки, який починається з вибору типу та направлення заявки, продовжуючи додаванням товарів та завершуючи відправкою заявки на розгляд. Цей процес дозволяє постачальнику точно вказати, який товар хоче запропонувати мережі, і у якій категорії він буде представлений.

Другий процес – це робота із заявкою після її прийняття до системи. Цей процес включає кілька ключових етапів: від первинної обробки заявки, через затвердження комерційних та договірних умов, тендер, розгляд на аудиті якості, до остаточного погодження умов та підписання договору. Завершальним етапом є введення товарів в асортимент мережі, що сигналізує про завершення бізнес-процесу та введення товарів у асортимент мережі.

Така покрокова карта, приклад якої наведено на рисунку 2.5, дозволяє поставити чіткі орієнтири для всіх учасників процесу та забезпечує максимальну прозорість на кожному етапі, що сприяє успішній реалізації пропозицій постачальників товарних категорій.

Карта етапів створення заявки:

1. Вибір типу заявки: на першому етапі постачальник вибирає тип заявки, що відповідає його бізнес-інтересам. Наприклад, це може бути заявка для включення товарів до асортименту мережі, для продажу під брендом мережі, або для інтернет-магазину.

2. Вибір напряму співпраці: після вибору типу заявки, постачальник повинен вибрати один із напрямків, у якому діятиме його заявка:

- ВТМ (товари під маркою мережі);
- напрям брендової продукції – товари під індивідуальним брендом;
- інтернет-магазин - для додавання товарів до онлайн-асортименту.

3. Додавання товарів: на цьому етапі постачальник додає товари, які він хоче запропонувати для включення до асортименту мережі. Він повинен

вказати всі необхідні параметри, такі як опис, характеристики, ціни та інші дані, які будуть необхідні розгляду заявки.

4. Відправка заявки на розгляд: після того, як всі товари додані, заявка відправляється на розгляд. Система підтверджує успішне відправлення та надає інформацію про те, що заявка надійшла у чергу на обробку.

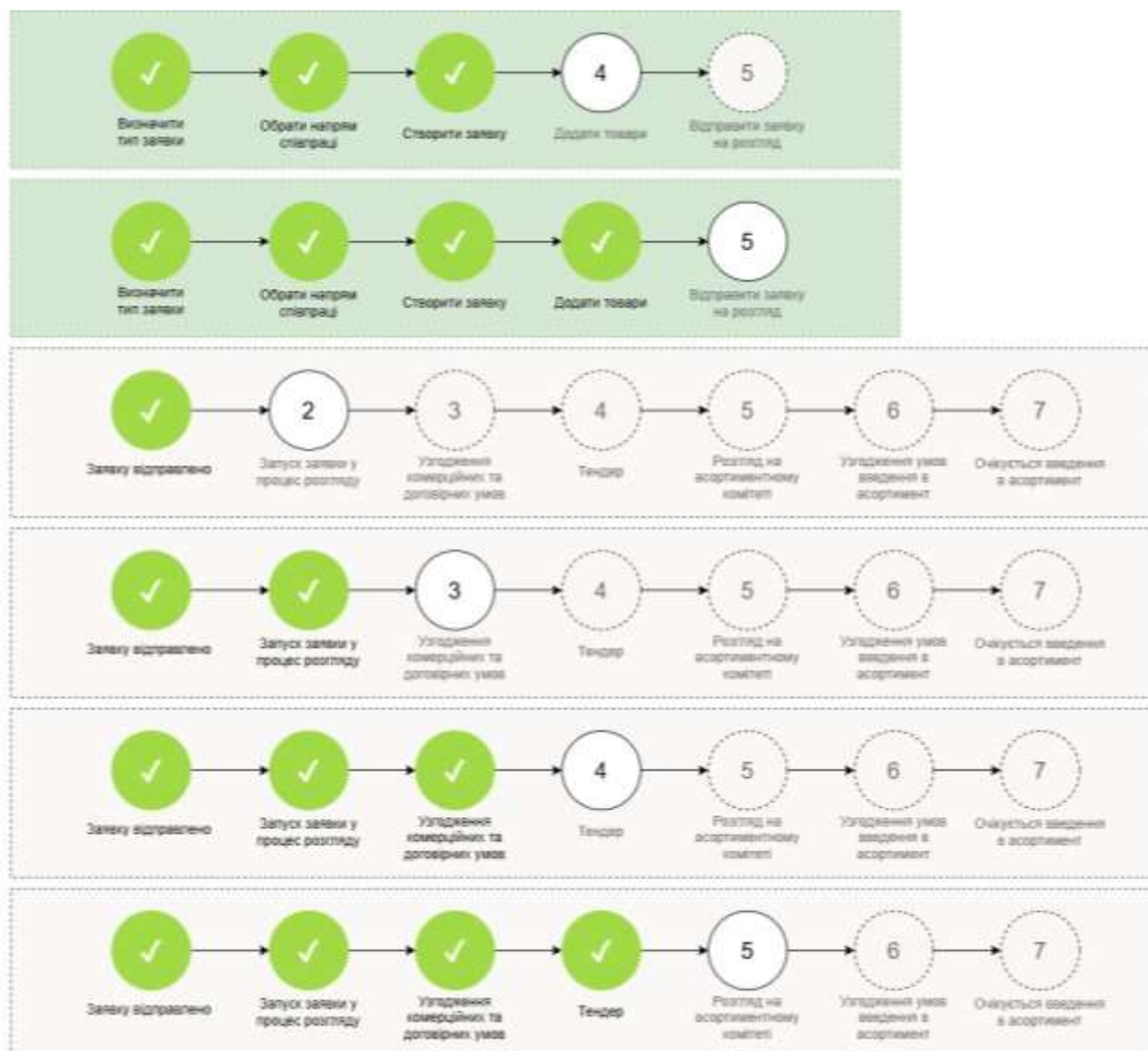


Рисунок 2.5 – Карта етапів створення та бізнес-процесу обробки КП

Карта етапів роботи із заявкою:

1. Заявка прийнята в роботу: після надсилання заявки вона надходить в обробку, і заявка отримує статус "Прийнята в роботу". На цьому етапі розпочинається її активна обробка в рамках процесу.

2. Запуск заявки до процесу розгляду: заявка надходить на наступний етап — процес розгляду. У рамках цього етапу виконується перевірка наданих даних, їх відповідність вимогам та аналіз бізнес-доцільності.

3. Затвердження комерційних та договірних умов: на цьому етапі відбувається погодження умов співробітництва між постачальником та мережею, включаючи комерційні умови та договірні положення.

4. Тендер: якщо необхідно, може бути організований тендерний процес, де оцінюватиметься вартість та якість товарів, пропонуваніх постачальником, та обирається найбільш вигідна пропозиція для мережі.

5. Розгляд на АК: після успішного проходження тендеру заявка передається на розгляд до комісії з асортиментного комітету.

6. Узгодження умов ведення (підписання договору): після успішного розгляду всіх умов відбувається остаточне узгодження всіх аспектів угоди та підписання договору.

7. Очікується введення товарів до асортименту мережі: на останньому етапі товарів очікують включення до асортименту мережі. Після завершення всіх процедур та підписання договору, товар надходить у продаж та стає доступним для клієнтів мережі.

У процесі розробки системи подачі комерційних пропозицій для постачальників було зібрано та систематизовано вимоги від замовників та кінцевих користувачів майбутнього порталу. На основі бізнес-процесу була побудована BPMN-модель, що відображає ключові етапи подання та обробки заявок. Ці вимоги потім стали основою дизайну рішення, що включає опис статусів заявок і можливих переходів між ними.

Приклад, наведений на основі заявки на подання комерційної пропозиції є лише частиною загальної картини. Подібні процеси слід моделювати для кожного функціонального блоку веб-сайту, щоб створити зручну та ефективну систему, що відповідає всім вимогам користувачів та забезпечує прозорість роботи на всіх етапах.

2.3 Проектування системи

Проектування порталу постачальника включає створення архітектури системи та проектування бази даних, що є основою для ефективної роботи всієї платформи. Архітектурні рішення повинні забезпечувати не лише високу продуктивність і масштабованість, але й надійний захист даних та підтримку безперервної роботи при зростанні навантажень. У цьому контексті важливо забезпечити можливість гнучкої інтеграції з іншими зовнішніми системами, а також адаптувати архітектуру до вимог бізнес-процесів, що постійно змінюються.

Проектування архітектури [15] має включати визначення необхідних компонентів системи, їх взаємодію та забезпечення ефективного обміну даними між різними модулями. Крім того, слід врахувати вимоги безпеки, такі як захист персональних даних, а також можливість масштабування системи в майбутньому, щоб врахувати потенційне зростання кількості постачальників та обсягів оброблюваних даних. Усі ці аспекти є важливими для забезпечення стабільної та безпечної роботи порталу в довгостроковій перспективі.

2.3.1 Архітектура порталу постачальників

Архітектура порталу постачальника, що зображена на рисунку 2.6, побудована з використанням мікросервісної архітектури для бекенда, що дозволяє досягти високої гнучкості, масштабованості та відмовостійкості системи. В основі архітектури лежить поділ на клієнтську та сервісну частину із їхньою взаємодією через API Gateway.

Система використовує балансувальник навантаження для ефективного розподілу запитів між сервісами. Це дозволяє забезпечувати високий рівень доступності та масштабованості при зростанні навантаження.

Платформа також використовує кілька баз даних для різних потреб:

- реляційну базу даних PostgreSQL для зберігання структурованих даних;
- MongoDB для зберігання файлових даних;

- Redis для кешування та забезпечення швидкої роботи з даними;
- Elasticsearch для логування, що допомагає адміністраторам та розробникам відстежувати стан системи та оперативно реагувати на можливі проблеми.

Інтеграція різних компонентів та сервісів порталу постачальника відіграє ключову роль у забезпеченні гнучкості, продуктивності та масштабованості системи [16]. Для цього в системі використовуються як синхронні, так і асинхронні підходи до взаємодії між компонентами, що дозволяє оптимізувати обробку запитів та підвищити стійкість до відмови системи.

1. Синхронні взаємодії

Синхронні взаємодії відбуваються, коли один компонент системи очікує на відповідь від іншого перед продовженням виконання. Ці взаємодії найчастіше застосовуються, коли необхідно отримати дані або виконати дії, які потребують негайної відповіді.

Технології для синхронних взаємодій:

- REST API (через HTTP/HTTPS) використовується для прямих запитів від фронтенду до бекенду через API Gateway;
- GraphQL може використовуватися для оптимізації запитів, дозволяючи фронтенду вказувати, які дані йому необхідні;
- OAuth 2.0 / JWT для забезпечення безпеки та авторизації у синхронних запитах.

Переваги синхронної інтеграції:

- простота та пряmlinійність: один запит, одна відповідь;
- легкість у налагодженні, тому що відповідь приходить відразу.

Недоліки:

- блокування, якщо один сервіс повільно опрацьовує запит (наприклад, при тривалих операціях або великих даних);
- менше гнучкості при навантаженні на систему - всі сервіси мають бути доступні, щоб опрацювати запити.

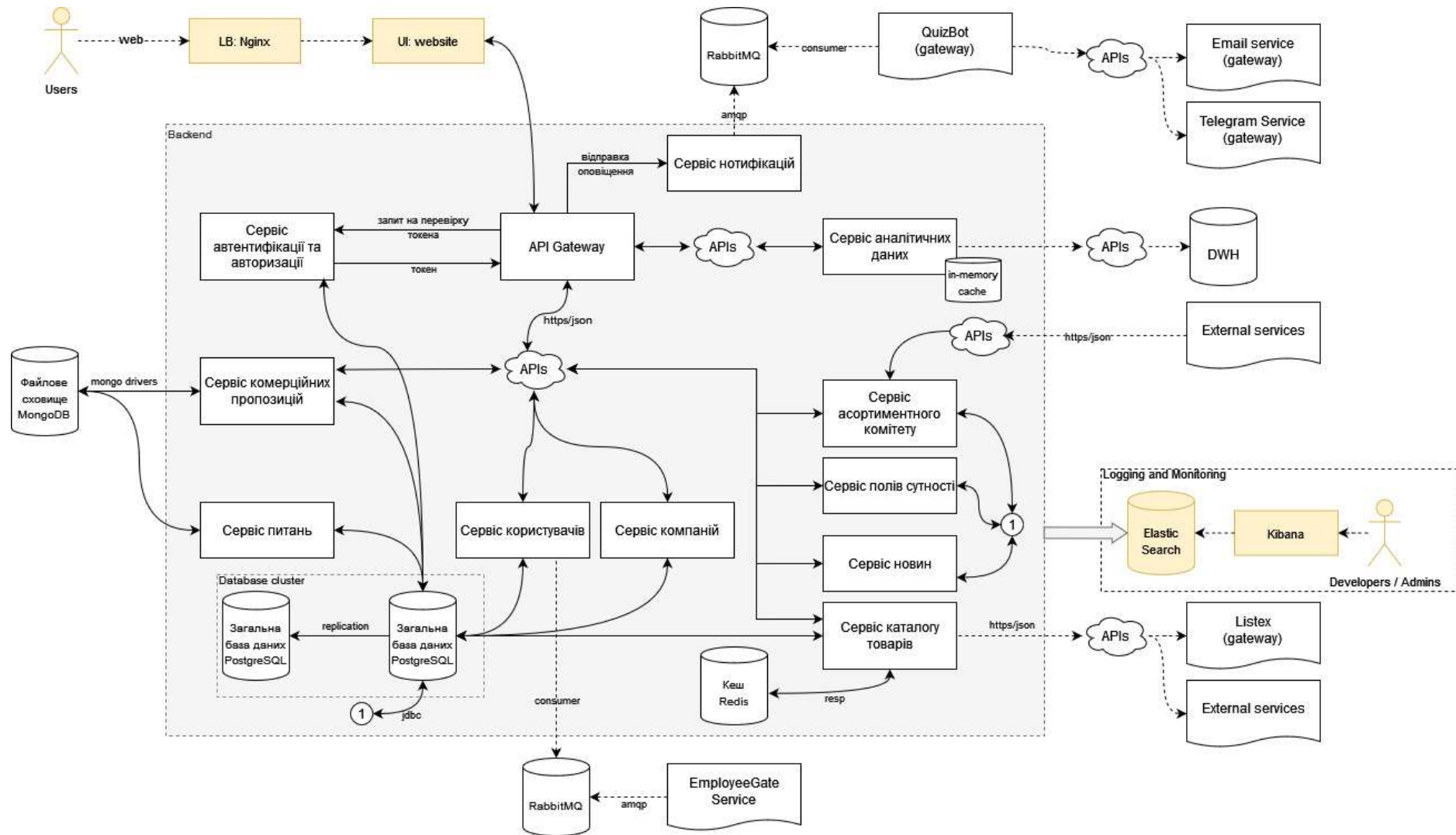


Рисунок 2.6 – Компонентна модель порталу постачальників

2. Асинхронні взаємодії

Асинхронні взаємодії припускають, що запит не вимагає негайної відповіді і може бути опрацьовано пізніше. Такий підхід дозволяє системі продовжувати роботу, не блокуючи виконання інших завдань, та ефективно справлятися з більшими обсягами даних.

Чати в реальному часі: під час надсилання повідомлення в чат використовується WebSocket, який підтримує двосторонній зв'язок у реальному часі. Чат-повідомлення можуть надходити та оброблятися асинхронно, не блокуючи інші операції в системі.

Технології для асинхронних взаємодій:

- Message Broker (наприклад, RabbitMQ) для черг повідомлень та асинхронної передачі між мікросервісами;
- WebSockets (наприклад, Socket.IO) для чатів в реальному часі;
- Event-Driven Architecture: мікросервіси можуть генерувати події (наприклад, створення нової заявки або оновлення статусу), які інші сервіси обробляють асинхронно.

Переваги асинхронної інтеграції:

- висока продуктивність: запити та завдання обробляються у фоновому режимі без блокування основних процесів;
- найкраща стійкість до відмов: навіть якщо один сервіс не може відповісти відразу, система продовжує працювати, а запити виконуються пізніше;
- менше залежно від синхронності сервісів, що дозволяє системі легко масштабуватись та обробляти пік навантаження.

Недоліки:

- більш складна реалізація: вимагає використання черг повідомлень, обробки помилок та управління станом завдань;
- затримка в обробці: відповідь може не надійти негайно, що може бути незручно для користувача у випадку, якщо система має надати зворотний зв'язок у реальному часі.

2.3.2 Проєктування бази даних

Нижче, на рисунку 2.7, представлена логічна схема основної бази даних порталу постачальника. Дана модель є типовою для розробки та налаштування реляційних баз даних, які використовуються в бізнес-інформаційних системах та дослідженнях. Реляційна база даних сприймається користувачем як набір нормалізованих зв'язків між таблицями, що забезпечує ефективне зберігання та обробку даних з урахуванням їх взаємозв'язків [17].

Якість розробленої бази даних цілком залежить від якості виконання окремих етапів її проєктування. Процес проєктування бази даних включає кілька важливих етапів, таких як визначення вимог до даних, моделювання даних, вибір відповідних технологій та інструментів для реалізації, а також оптимізація її роботи. На кожному з цих етапів важливо ретельно продумати всі аспекти, щоб створити ефективну та масштабовану базу даних, яка буде відповідати потребам системи і користувачів. При цьому кожен етап має свою специфіку і важливість, що впливає на результат. Наприклад, на етапі визначення вимог необхідно чітко сформулювати потреби користувачів і зацікавлених сторін, що допоможе зібрати та структурувати дані, які повинна зберігати база даних.

Величезне значення має якісна розробка логічної моделі даних, так як вона, з одного боку, забезпечує адекватність бази даних предметної області, а з іншого боку, визначає структуру фізичної БД і, отже, її експлуатаційні характеристики. Логічна модель є основою для побудови бази даних, оскільки вона формує структуру даних, взаємозв'язки між ними, а також правила обробки інформації. У разі неякісної розробки логічної моделі можуть виникнути проблеми з масштабованістю, швидкодією та підтримкою бази даних в майбутньому. Тому логічна модель повинна враховувати всі особливості предметної області та вимоги до продуктивності та безпеки даних [18].

Одні й ті ж дані можуть групуватися в таблиці-відносини різними способами, тобто можлива організація різних наборів відносин взаємопов'язаних інформаційних об'єктів предметної області.

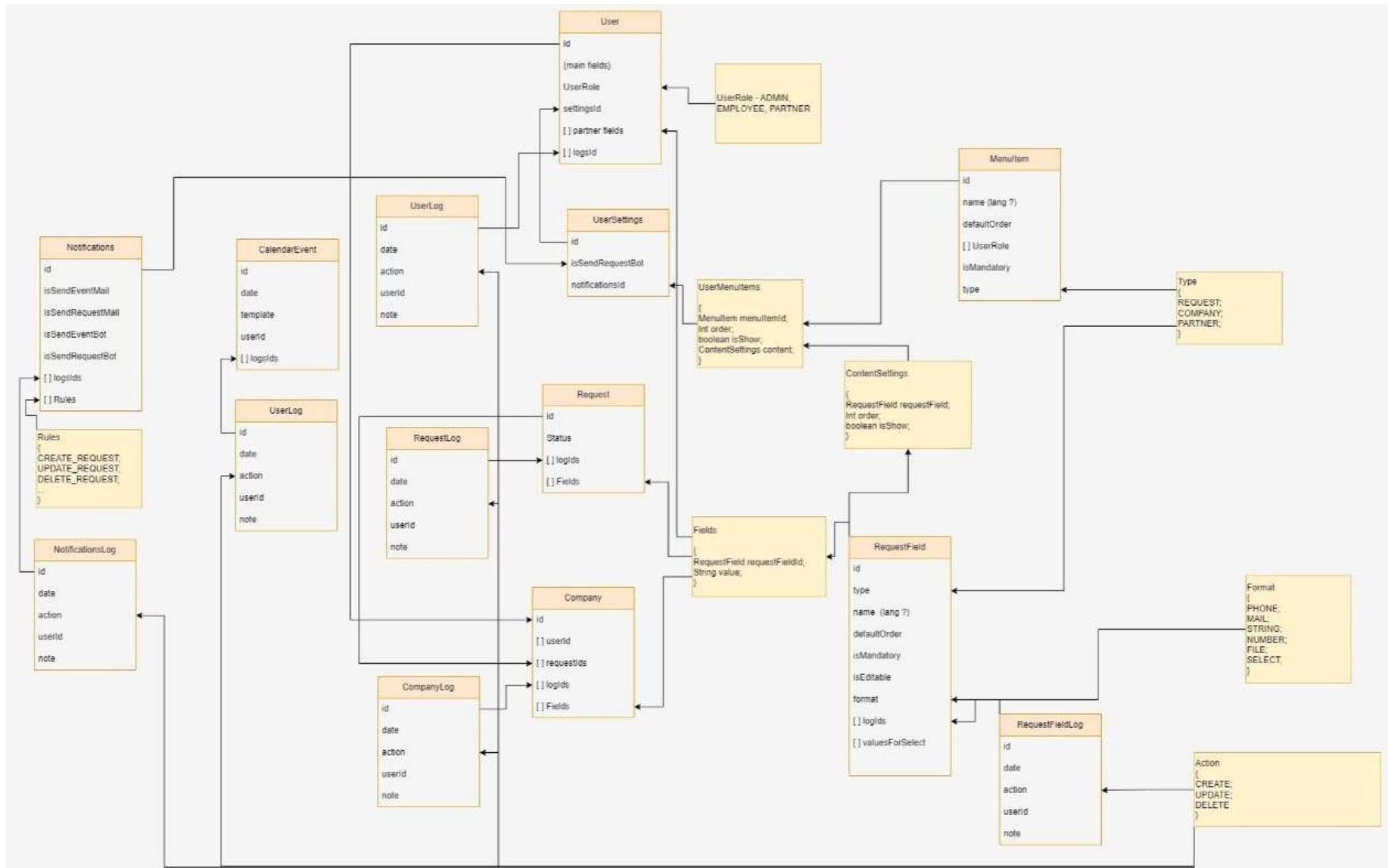


Рисунок 2.7 – Логічна схема основної бази даних систем

Це дозволяє вибрати найефективніший підхід до зберігання даних і забезпечення зручності для користувачів. Вибір структури таблиць і взаємозв'язків між ними безпосередньо впливає на швидкість виконання запитів, обробку та оновлення даних. Таким чином, при проектуванні бази даних потрібно враховувати як специфіку предметної області, так і вимоги до продуктивності, доступності та безпеки даних, щоб забезпечити ефективну роботу системи в майбутньому.

Угруповання атрибутів у відносинах повинна бути раціональною, що гранично скорочує дублювання даних і спрощує процедури їх обробки та оновлення. Певний набір відносин володіє кращими властивостями при включенні, модифікації і видалення даних, якщо він відповідає конкретним вимогам нормалізації відносин.

Нормалізація відносин — формальний апарат обмежень на їх формування, який дозволяє усунути дублювання даних, забезпечити їхню несуперечливість і зменшити витрати на підтримку БД. На практиці найбільш часто використовуються поняття першої, другої і третьої нормальних форм реляційних БД.

2.3.3 Інформаційна архітектура сайту

Інформаційна архітектура — це перш за все ефективне управління вмістом, чітко визначені процедури та інформаційна політика сайту, додатку, тощо. Інформаційна архітектура є складним процесом, який включає в себе багато інших галузей знання [19].

Існує багато визначень інформаційної архітектури, наприклад:

1. Поєднання схем організації, предметизації і навігації, реалізованих в інформаційній системі.
2. Структурне проектування інформаційного простору, що сприяє виконанню завдань і інтуїтивному доступу до вмісту.

Проведення кордонів інформаційної архітектури виявляється ще більш нетривіальним завданням.

Наприклад, деякі речі визначені але не відносяться до інформаційної архітектури:

- графічне оформлення не є інформаційною архітектурою;
- розробка програмного забезпечення не є інформаційної архітектурою;
- проектування юзабіліті не є інформаційною архітектурою.

Тобто інформаційна архітектура – це потужний інструмент для знайомства зі складною багатовимірної природою інформаційних просторів. Як і будівлі, веб-сайти/додатки володіють архітектурою, що визначає нашу роботу з ними. Деяким додаткам властива логічна структура, що дозволяє нам знайти відповідь і виконати завдання, в інших же відсутня яка-небудь логічна організація, і спроби навігації по ним виявляються марними.

Можливість знайти інформацію та необхідний функціонал – вирішальний фактор для зручності додатку в цілому.

Основні елементи інформаційної архітектури включають структуру навігації, взаємозв'язки між різними блоками даних, а також зручність доступу до основних функцій платформи. Важливо, щоб усі розділи порталу були логічно пов'язані між собою, і користувач міг легко переходити з однієї функції до іншої без труднощів. Наприклад, постачальники повинні мати можливість подати заявку на комерційну пропозицію, відслідковувати статуси своїх заявок, додавати та редагувати товари, а також отримувати аналітичні дані про свою діяльність на порталі.

Для того, щоб краще уявити, як працює інформаційна архітектура, було розроблено ментальну карту веб-сайту, що відображає взаємозв'язки між основними компонентами порталу для постачальників, яка представлена на рисунку 2.8. Ця карта допомагає зрозуміти, як взаємодіють різні елементи системи, як вони пов'язані між собою, і як постачальники можуть переміщатися по сайту, виконуючи свої завдання. Важливо, що кожен етап взаємодії з платформою чітко позначений, що дозволяє користувачам з комфортом і без зайвих зусиль досягати своїх цілей.

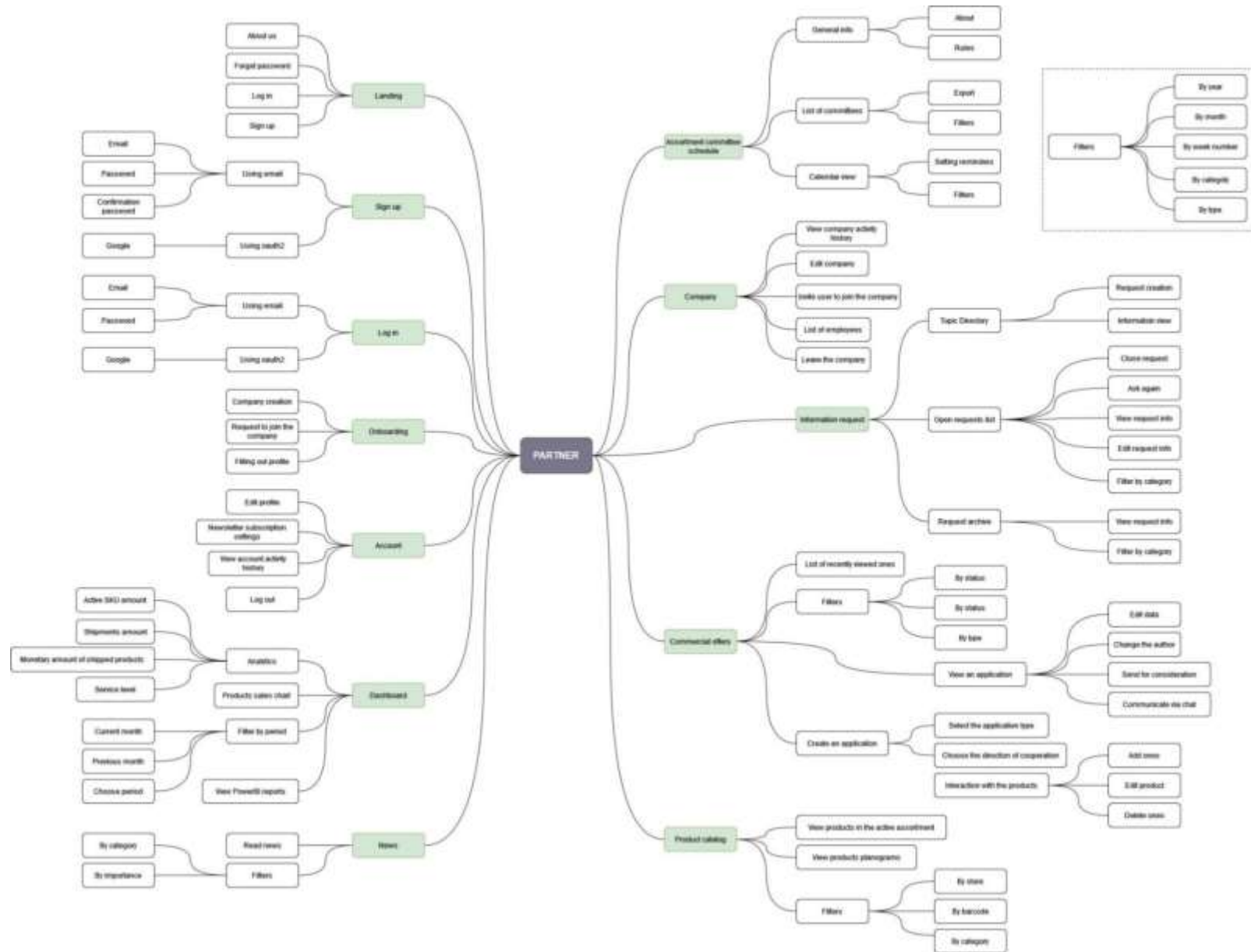


Рисунок 2.8 – Ментальна карта веб-сайту для постачальників

В рамках проєктування порталу також були створені макети двох ключових сторінок, які найбільш яскраво демонструють функціональні можливості платформи:

1. Дашборд з аналітичними даними — ця сторінка надає постачальнику вичерпну інформацію щодо його діяльності на платформі.

Тут постачальники можуть побачити важливі показники, що зображені на рисунку 2.9. Дашборд створений для того, щоб постачальник міг швидко оцінити поточний стан своїх справ і приймати рішення на основі актуальних даних.

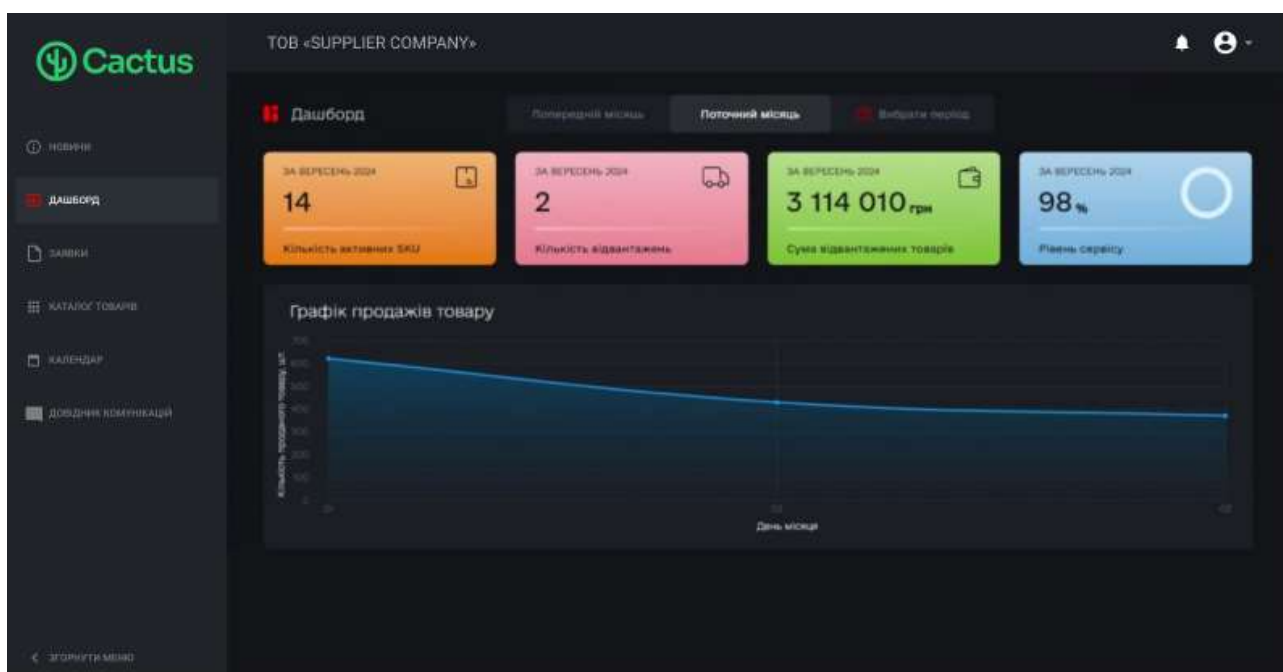


Рисунок 2.9 – Дизайн дашборду аналітичних даних

2. Календар асортиментних комітетів — сторінка, що дозволяє постачальникам переглядати заплановані дати засідань асортиментних комітетів. Макет дизайну сторінки календарних подій проілюстровано на рисунку 2.10. Це важливий інструмент, оскільки постачальники можуть заздалегідь спланувати свою діяльність, підготувати заявки на товари, які вони хочуть запропонувати для включення до асортименту, і врахувати терміни, у які потрібно подати ці заявки для розгляду комітетом.

Інформаційна архітектура сайту повинна бути спроектована так, щоб користувачі могли максимально швидко і зручно знаходити необхідні функції, а також мати доступ до важливої інформації про свою діяльність на платформі. Вона не лише забезпечує зручність у використанні, але й дозволяє організувати всі дані і процеси таким чином, щоб система могла ефективно працювати навіть при великих обсягах інформації та значній кількості користувачів. Саме тому, ретельно розроблена інформаційна архітектура є критично важливою для успішного функціонування порталу для постачальників.

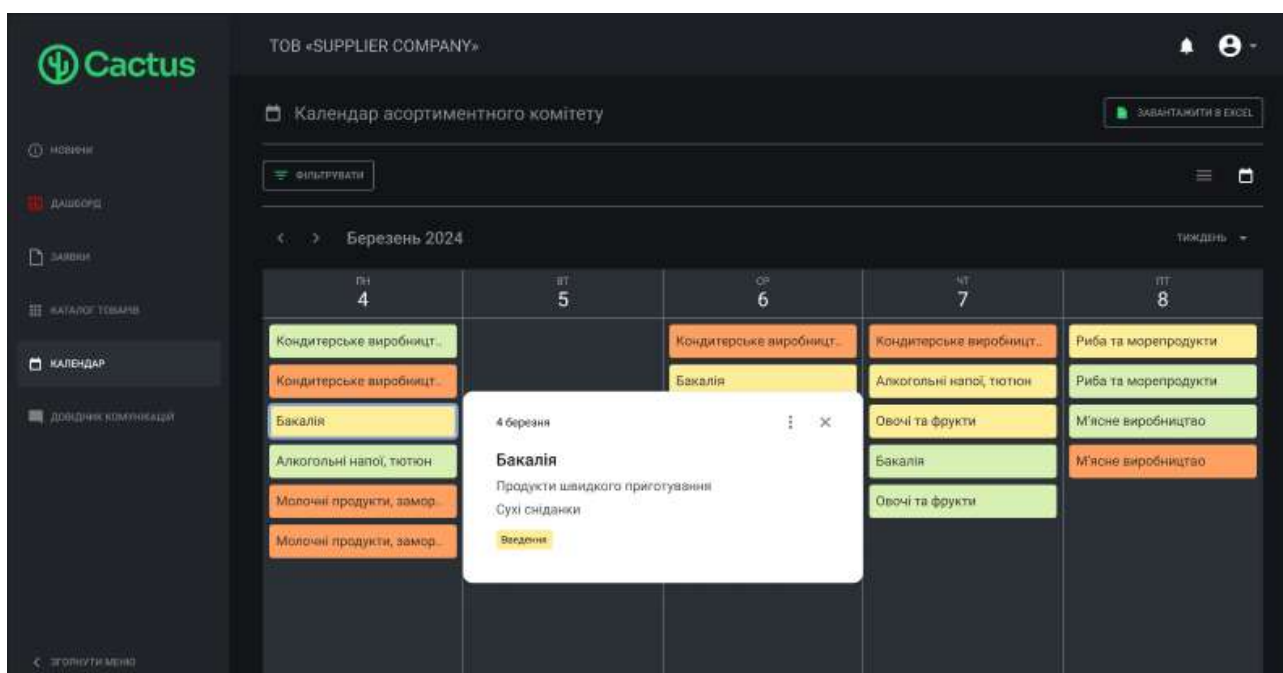


Рисунок 2.10 – Дизайн календаря асортиментних комітетів

Правильна організація інформації на порталі забезпечує користувачам швидкий доступ до ключових функцій і даних, необхідних для прийняття рішень. Чітко спроектована архітектура покращує взаємодію користувачів із системою, знижує ймовірність помилок та спрощує процеси, пов'язані з подачею та обробкою заявок.

2.4 Результати впровадження та оцінка ефективності порталу

Результати впровадження системи та оцінка її ефективності є важливими етапами для визначення успіху проекту та виявлення можливих напрямків для

подальшого вдосконалення платформи. У цьому контексті, впровадження порталу для постачальників має кілька основних результатів та метрик, за якими можна оцінити його ефективність.

Результати впровадженнь системи:

1. Підвищення прозорості процесу: користувачі порталу можуть легко відслідковувати статуси своїх заявок і бачити історію змін. Це значно підвищує прозорість процесу і знижує рівень непорозумінь між постачальниками та співробітниками мережі.

2. Зростання продуктивності: впровадження системи призвело до зростання кількості поданих заявок та товарних пропозицій.

3. Протягом перших двох тижнів після релізу MVP-версії порталу реєстрацію на сайті пройшло близько 26% постачальників (208 підприємств), з якими співпрацює ритейлер.

Подальші кроки для покращення системи:

1. У подальшому розвитку системи передбачена реалізація нового функціоналу:

- планограма магазину для того, щоб постачальники мали доступ до актуальної викладки їхніх товарів в мережі ритейлера;

- реалізація системи сповіщень, щоб постачальник завжди був обізнаний про найближчі терміни та події, пов'язані з подачею КП з індивідуальними налаштуванням каналів відправки нагадувань для мінімізації ризику пропуску важливих етапів.

2. Впровадження мобільної версії програми. З огляду на зростання використання мобільних пристроїв для роботи та взаємодії з різними сервісами створення мобільної версії є важливою частиною стратегії для забезпечення зручності та доступності системи на будь-якому пристрої.

2.5 Висновки до розділу

У розділі було проведено всебічне дослідження та проектування ключових аспектів порталу для постачальників, що включає як теоретичне

обґрунтування, так і практичне впровадження подання комерційних пропозицій постачальниками товарних категорій.

Методологія дослідження заінтересованих сторін дозволила чітко визначити потреби користувачів, що стало основою для подальшого моделювання бізнес-процесів замовниками.

Проектування інформаційної системи, включаючи архітектуру сайту та бази даних, забезпечило створення стійкої та масштабованої платформи, здатної підтримувати високу продуктивність та безпеку при зростаючій кількості користувачів.

Результати впровадження платформи показали позитивний вплив на процес подання заявок, збільшення залученості постачальників та покращення ефективності роботи.

Таким чином, виконана робота з проектування та впровадження порталу для постачальників дозволила досягти поставлених цілей, і надалі передбачається розширення функціоналу системи для покращення взаємодії з постачальниками та подальшого зростання бізнесу.

ВИСНОВКИ

Дана магістерська робота була присвячена обґрунтуванню і розробці вимог до веб-платформи для постачальників, яка включатиме функціональність для подачі комерційних пропозицій, швидкого вирішення питань через прямий доступ до відповідальних осіб, а також спрощення процесів реєстрації та інтеграції нових постачальників. Ці зміни сприятимуть значному покращенню рівня задоволеності постачальників співпрацею з компанією.

В результаті проведеного аналізу зацікавлених сторін були визначені ключові чинники, що можуть мати вплив на процес реалізації проекту. Це важливо, оскільки великий масштаб бізнесу може бути причиною значних фінансових результатів проекту. У процесі дослідження проаналізовано внутрішню структуру компанії. Виявлено, сильні і слабкі сторони функціонування, які можуть мати вплив на процес реалізації проекту.

Результатом даної магістерської роботи стала формування та обґрунтування архітектури порталу постачальників, яка буде спрямована на розв'язання проблеми спрощення процесів реєстрації та інтеграції нових постачальників серед великої кількості інформаційних систем та баз даних. Проведений аналіз дизайн рішення включає опис статусу запиту і можливих переходів між ними. В результаті було визначено два основних процеси: створення заявки та обробка заявки, кожен з яких розділений на кілька чітких етапів, що забезпечує прозорість та ефективність взаємодії.

В рамках розробки системи подачі комерційних пропозицій був проведений комплексний аналіз і моделювання бізнес-процесу. Спочатку були створені BPMN-моделі для візуалізації ключових етапів подачі заявки та взаємодії з постачальниками. На основі цього процесу були сформульовані функціональні вимоги, які потім систематизовані і представлені в таблиці.

Ці вимоги охоплюють найважливіші аспекти, такі як дії користувача, бізнес-правила та механізми, які має підтримувати система.

Проведений аналіз дозволив зробити позитивний висновок, щодо створення системи, адже вона представляє комерційний інтерес і економічно вигідна. Основні ризики в реалізації проекту незначні і знаходяться в межах компанії, що дозволяє ними якісно керувати. Дані висновки дозволяють вирішити про доцільність проекту і прийняти його до виконання.

Таким чином, моделювання процесів, збір вимог і проектування рішень забезпечили основу для подальшого розвитку системи, задоволення потреб користувачів і забезпечення безперебійної роботи платформи.

Результати магістерської роботи можуть бути використані у подібних проектах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технології програмування та створення програмних продуктів: конспект лекцій / укладач О. В. Алексенко. – Суми : Сумський державний університет, 2013. – 133 с.
2. Despa M. Comparative study on software development methodologies. Database Systems Journal; 2014.
3. Extreme Chaos. The Standish Group International Inc. – The Report of the Standish Group International, 2005. – 12 p.
4. Молоканова, В. М., & Шевченко, Ю. О. (2024). Управління проектною командою. <https://ir.nmu.org.ua/handle/123456789/167646>
5. K. Wiegers, J. Beatty. Software Requirements, 3rd edition, : книга /, 2014. — 736 с.
6. International Institute of Business Analysis (ІІВА). A Guide to the Business Analysis Body of Knowledge (BABOK Guide), 3rd edition, 2015. — 560 p.
7. Козак О.Л. Опорний конспект лекцій з курсу – Аналіз вимог до програмного забезпечення для студентів напрямку підготовки – Програмна інженерія / О.Л. Козак. – Тернопіль, 2011. – 56 с.
8. Stioca M., Ghlic-Micu B., Mircea M., Uscatu C. Analyzing Agile Development – from Waterfall Style to Scrumban. Informatica Economică. 2016. No4. С. 5-14
9. Project Management Institute (PMI). A Guide to the Project Management Body of Knowledge (PMBOK Guide), 6th edition, 2017. — 756 p.
10. Visual Paradigm. Use Case Diagram: A Notation Guide. 2020. [Електронний ресурс] — Режим доступу: <https://blog.visual-paradigm.com/ru/use-case-diagram-a-notation-guide/> (дата звернення: 08.11.2024).
11. DOU. Система управління проектами: як вибрати інструмент для своєї команди. 2020. [Електронний ресурс] — Режим доступу: <https://dou.ua/forums/topic/40575/> (дата звернення: 09.11.2024).

12. Ковалевська А. В., Зеленський С. В. Аналіз моделей оцінки стейкхолдерів: прикладні аспекти формування стратегії сталого розвитку міст України. НАУКОВІНФОРМ, № 9, 2019, с. 299–307. [Електронний ресурс] — Режим доступу: http://mail.business-inform.net/export_pdf/business-inform-2019-9_0-pages-299_307.pdf (дата звернення: 04.11.2024).

13. Object Management Group (OMG). BPMN Specification. [Електронний ресурс] — Режим доступу: www.omg.org/spec/BPMN/2.0.2/PDF, с. 367-424 (дата звернення: 12.11.2024).

14. Cherkasova O. Атрибути вимог. [Електронний ресурс] — Режим доступу: <https://medium.com/@olga.cherkasova.posts/> (дата звернення: 12.11.2024).

15. Richardson, C. Microservices Patterns: With Examples in Java. 1st edition, 2018. — 472 p.

16. Simplilearn. Kafka vs RabbitMQ: A Comparison. [Електронний ресурс] — Режим доступу: <https://www.simplilearn.com/kafka-vs-rabbitmq-article> (дата звернення: 18.11.2024).

17. Хом'як Т. В. Бази даних у професійних задачах аналітики [Електронний ресурс] : навч. наочн. посіб. / Т. В. Хом'як, К. С. Хабарлак, Д.М. Гаранжа ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2024. – 192 с.

18. DOU. Types of Databases. [Електронний ресурс] — Режим доступу: <https://dou.ua/lenta/articles/types-of-databases/> (дата звернення: 27.11.2024).

19. V. Nikitinsky. Посібник з інформаційної архітектури в UX для новачків. [Електронний ресурс] — Режим доступу: <https://vnikitinsky.medium.com/> (дата звернення: 03.11.2024).

20. Кваліфікаційна робота магістра [Електронний ресурс] : методичні рекомендації для здобувачів ступеня магістра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз / уклад. : Т.А. Желдак, Т.В. Хом'як, А.В. Малієнко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2024. – 33 с.

Додаток А. Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення				Найменування	Кількість аркушів	Примітки		
1									
2					Документація				
3									
4	САУ.КР.24.17.ПЗ				Пояснювальна записка	N1	Формат А4		
5									
6					Демонстраційний матеріал	N2	Презентація на CD-R		
7									
8					Копія роботи	1	Диск CD-R		
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
					САУ.КР.24.17.ДА.ПЗ.				
Змін.	Аркуш	№ докум.	Підпис	Дата	Матеріали кваліфікаційної роботи	Літ.	Аркуш	Аркушів	
Розроб.		Сазанська							
К. розд.		Молоканова							
Керівн.		Молоканова							
Н.контр.		Хом'як							
Зав. каф.		Желдак							
						НТУ «ДП», 12; 124м-23-1			

Відгук
на кваліфікаційну роботу магістра
 студента(ки) групи 124м – 23 – 1
 спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи: _____

Обсяг кваліфікаційної роботи _____ стор.

Мета кваліфікаційної роботи: _____

Актуальність теми _____

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності магістра спеціальності 124 Системний аналіз, оскільки _____

Виконані в кваліфікаційній роботі завдання відповідають вимогам другого (магістерського) рівня вищої освіти. Оригінальність наукових рішень полягає в _____

Практичне значення результатів кваліфікаційної роботи полягає в _____

Висновки підтверджують можливість використання результатів роботи в _____

Оформлення пояснювальної записки та демонстраційного матеріалу до неї виконано згідно з вимогами. Роботу виконано самостійно, відповідно до завдання та у повному обсязі (*в разі невідповідності – вказати*)

У роботі відзначено такі недоліки: _____

Кваліфікаційна робота в цілому заслуговує оцінки: _____

З урахуванням висловлених зауважень автор (не) заслуговує присвоєння кваліфікації «магістр з системного аналізу».

Керівник кваліфікаційної роботи магістра,
 науковий ступінь, вчене звання, посада _____

д.т.н, проф.

_____ / Молоканова В.М.

Рецензія
на кваліфікаційну роботу магістра
 студента(ки) групи 124м – 23 – 1
 спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи:

Обсяг кваліфікаційної роботи: _____

Висновок про відповідність кваліфікаційної роботи завданню та освітньо-професійній програмі спеціальності _____

Загальна характеристика кваліфікаційної роботи, ступінь використання нормативно-методичної літератури та передового досвіду

Позитивні сторони кваліфікаційної роботи:

Основні недоліки кваліфікаційної роботи:

Кваліфікаційна робота в цілому заслуговує оцінки: _____

З урахуванням висловлених зауважень автор (не) заслуговує присвоєння кваліфікації «магістр з системного аналізу».

Рецензент,
 науковий ступінь, вчене звання, посада

??., ??.

_____ / Прізвище І.Б.