

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

(повна назва)

Кафедра прикладної математики

(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Аналіз та впровадження системи  
забезпечення безперервності бізнесу

(тема)

Виконав:

здобувач 2 року навчання, групи САУМ-23-2

Кожин А.Ю.

(прізвище, ініціали)

Спеціальність 124 Системний аналіз

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системний аналіз і управління

(повна назва освітньої програми)

Керівник доц. Гибкіна Н.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ

(підпис)

Сидоров М.В.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 124 Системний аналіз

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системний аналіз і управління

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ \_\_\_\_\_

(підпис)

“ 25 ” листопада 2024 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Кожину Андрію Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз та впровадження системи забезпечення безперервності бізнесу

затверджена наказом по університету від 22 листопада 2024 р. № 1228 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 6 січня 2025 р.

3. Вихідні дані до роботи перелік вимог до інформаційно-аналітичної системи забезпечення безперервності бізнесу

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Системний аналіз предметної області

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Результати обчислювального експерименту

5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій \_\_\_\_\_

1. Актуальність теми роботи \_\_\_\_\_

2. Постановка задачі \_\_\_\_\_

3. Системний аналіз предметної області \_\_\_\_\_

4. Метод чисельного аналізу \_\_\_\_\_

5. Результати обчислювального експерименту \_\_\_\_\_

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	6 – 12 листопада 2024 р.	виконано
2	Вибір та обґрунтування методу	13 – 26 листопада 2024 р.	виконано
3	Розробка алгоритму і програми	27 листопада – 10 грудня 2024 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	11 грудня – 24 грудня 2024 р.	виконано
5	Робота над текстом пояснювальної записки	25 грудня 2024 р. – 9 січня 2025 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 січня 2025 р.	виконано

Дата видачі завдання 25 листопада 2024 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Гибкіна Н.В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 78 с., 9 табл., 19 рис., 1 дод., 16 джерел.

СИСТЕМНИЙ АНАЛІЗ, БЕЗПЕРЕРВНІСТЬ БІЗНЕСУ, ISO 22301, ІНФОРМАЦІЙНА СИСТЕМА, ВЛАСТИВОСТІ СИСТЕМ.

Об'єкт дослідження – інформаційна система забезпечення безперервності бізнесу.

Мета роботи – реалізація системи забезпечення безперервності бізнесу.

Методи дослідження – метод системного аналізу.

У кваліфікаційній роботі розглянуто проблему розробки системи забезпечення безперервності бізнесу та проаналізовано існуючі підходи до розв'язання цієї задачі. Виконано системний аналіз проблеми із врахуванням особливостей різних підходів та обрано найбільш підходящий метод розробки інформаційної системи для збору та зберігання необхідних даних.

Розглянуто необхідні властивості, такі як адаптивність та простота в використанні. На основі не функціональних вимог обрано оптимальний напрямок розробки системи у вигляді корпоративного веб застосунку. Розроблено програмний продукт, який дозволив розв'язати поставлену задачу. Надано рекомендації, щодо сучасних практик, які дозволяють спростити супровід та подальше розширення функціональності системи у реальних умовах.

## ABSTRACT

Introductory note: 78 pages, 9 tables, 19 figures, 1 appendix, 16 sources.

SYSTEM ANALYSIS, BUSINESS CONTINUITY, ISO 22301, INFORMATION SYSTEM, SYSTEM PROPERTIES.

Object of the study – the information system for business continuity.

Objective of the work – the implementation of a business continuity system.

Research methods – the method of system analysis.

The qualification work addresses the problem of developing a business continuity system and analyzes existing approaches to solving this task. A system analysis of the problem is performed, considering the specifics of various approaches, and the most suitable method for developing an information system for data collection and storage is selected.

The necessary properties, such as adaptability and ease of use, were examined. Based on non-functional requirements, the optimal direction for system development was chosen in the form of a corporate web application. A software product was developed that successfully addressed the task at hand. Recommendations were provided on modern practices that simplify maintenance and enable further expansion of the system's functionality in real-world conditions.

## ЗМІСТ

	С.
ВСТУП.....	8
1 Системний аналіз предметної області та постановка задач дослідження.....	10
1.1 Системний аналіз задачі реалізації та впровадження системи	
забезпечення безперервності бізнесу .....	10
1.1.1 Вербальна модель системи.....	10
1.1.2 Морфологічний опис системи .....	11
1.1.3 Функціональна модель системи .....	13
1.2 Аналіз сценаріїв вирішення задачі реалізації та впровадження системи	
забезпечення безперервності бізнесу .....	19
1.2.1 Модель аналізу проблеми .....	19
1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу	
ієрархій .....	23
1.2.3 Модель вирішення проблеми.....	27
1.3 Змістовна та формальна постановка задачі .....	28
1.3.1 Змістовна постановка задачі .....	28
1.3.2 Формальна постановка задачі .....	29
1.4 Постановка задач дослідження.....	30
2 Вибір та обґрунтування методу розв’язання.....	31
2.1 Нефункціональні вимоги.....	31
2.2 Розгортання.....	34
2.3 Концептуальна архітектура застосунку .....	38
2.4 Вимоги стандарту ISO 22301 до системи керування безперервністю	
бізнесу.....	39
Висновки за розділом 2.....	41
3 Програмна реалізація .....	43
3.1 Платформа .net як інструмент розробки корпоративних веб-додатків .....	43
3.2 Шаблони проєктування та архітектурні патерни.....	46

	7
Висновки за розділом 3.....	47
4 Розгортання пробної версії системи.....	49
4.1 Стандартне тестове середовище у Microsoft Azure .....	49
4.2 Інтерфейс системи.....	51
4.3 Візуалізація та аналіз даних про час відновлення .....	55
Висновки за розділом 4.....	56
Висновки .....	58
Перелік джерел посилання .....	59
Додаток А Лістинг програми .....	61

## ВСТУП

**Актуальність теми.** Актуальність розробки системи управління безперервністю бізнесу (BSM) в Україні зумовлена особливими соціально-економічними умовами, що склалися на тлі війни. У цих обставинах необхідність забезпечення безперервності бізнес-процесів та захисту критично важливих функцій підприємств стає надзвичайно важливою. Військовий конфлікт приніс численні загрози для бізнесу, такі як фізичне руйнування інфраструктури, втрати персоналу та ресурсів, а також загальну невизначеність щодо подальших подій. Система BSM здатна допомогти мінімізувати ці втрати, забезпечуючи роботу організації навіть за умов форс-мажору.

Ефективна система управління безперервністю дозволяє виконувати зобов'язання перед клієнтами, партнерами та співробітниками. У сучасних кризових умовах це сприяє захисту інтересів усіх зацікавлених сторін, що пов'язані з діяльністю підприємства. Крім того, все більше українських підприємств усвідомлюють необхідність відповідати міжнародним стандартам, таким як ISO 22301, що не лише демонструє відповідальність компанії, а й сприяє підвищенню її конкурентоспроможності на глобальному ринку.

**Мета і завдання кваліфікаційної роботи.** Метою кваліфікаційної роботи є розробка системи забезпечення безперервності бізнесу, яка б відповідала міжнародному стандарту ISO 22301 [1] і була адаптована до умов українських підприємств.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і системний аналіз сучасного стану задачі впровадження системи забезпечення безперервності бізнесу на підприємствах;
- дослідити методологію планування безперервності;
- розробити рекомендації щодо адаптації системи до умов конкретного підприємства;
- провести аналіз зібраних даних на наявність вразливостей у часі відновлення.

*Об'єктом дослідження є система забезпечення безперервності бізнесу на підприємстві.*

*Предметом дослідження є розробка та впровадження системи забезпечення безперервності бізнесу відповідно до стандарту ISO 22301.*

**Методи дослідження.** У роботі використовуються методи системного аналізу, проєктування та реалізації програмного забезпечення.

**Публікації.** Результати, отримані у роботі, було представлено на III Міжнародній молодіжній науково-практичній конференції НАВЧАННЯ І ВИКЛАДАННЯ: у світі після війни (м. Харків 08 листопада 2024 р.) [2].

# 1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

## 1.1 Системний аналіз задачі реалізації та впровадження системи забезпечення безперервності бізнесу

### 1.1.1 Вербальна модель системи

Розглянемо вербальну модель системи «Система забезпечення безперервності бізнесу».

Об'єкт аналізу – «Структура компанії щодо ризиків можливого припинення постачання послуг чи товарів».

Предмет аналізу – «Проблема формалізації та автоматизації системи забезпечення безперервності бізнесу».

Точка зору: дослідник.

Призначення системи: розв'язання проблеми збору необхідних даних про організацію для отримання сертифікації ISO 22301.

Мета системи – забезпечення безперервності бізнесу, що включає здатність організації підтримувати свою діяльність під час кризових ситуацій та швидко відновлюватися після них.

В роботі розглядаються вимоги та приклад реалізації системи забезпечення безперервності бізнесу, яка буде забезпечувати прийнятний для організації рівень порушення діяльності. Забезпечення безперервності бізнесу приділяє особливу увагу наступному:

- розуміння потреб організації та необхідності встановлення політик та цілей безперервності бізнесу;
- виконання та підтримка процесів і планів реагування для забезпечення виживання бізнесу під час порушень;
- перегляд ефективності;
- постійне вдосконалення на основі якісних і кількісних спостережень.

Із точки зору класифікації, систему можна розглядати як інформаційно-аналітичну, призначену для збору, зберігання та обробки інформації про структуру компанії, постачальників, використовувані інформаційні системи інших виробників, місця розташування власності та пріоритети внутрішніх процесів. Аналітична частина має використовуватись у процесі аналізу ризиків та побудові звітності, як доказів встановлених практик і процесів, що забезпечать змогу надання послуг у разі непередбачуваних перешкод.

Розроблена система не повинна залежати від специфіки аналізованої організації, а бути достатньо гнучкою, щоб мати можливість адаптуватися до різних сфер бізнесу, які хочуть її впровадити.

Дані в системі структуровані і зберігаються у системі управління реляційними базами даних. Кожна компанія має унікальну структуру і форму підпорядкування, що ускладнює побудову коректної схеми таблиць.

За рівнем конфіденційності система є системою з обмеженим відповідно до політик безпеки та конфіденційності доступом.

Класифікація системи допомагає зрозуміти її основні характеристики, що важливо для правильного настроювання, використання та супроводу системи бази даних.

### 1.1.2 Морфологічний опис системи

Найпростішою моделлю, яка описує структуру із точки зору зовнішнього світу є модель «чорний ящик» [3]. Стрілками позначають вхідні та вихідні дані системи. Вхідні величини – це те, що система використовує для своєї роботи, з чим вона взаємодіє і що перетворює. Вихідні величини – це результат діяльності системи, те, що вона створює та у що трансформує вхідні величини відповідно до своїх функцій.

Входом для системи керування безперервністю бізнесу (BCMS) будуть відомості про організаційну структуру, наявний персонал, необхідні ресурси,

постачальників, інформація щодо наслідків припинення процесу на певний період часу, внутрішні залежності між процесами та можливі тимчасові рішення, що можуть бути використані у випадку несподіваних обставин. Виходом системи будуть документи та звіти, що можуть бути використані як докази проведеного аналізу задля отримання міжнародної сертифікації ISO 22301, а також поширені серед співробітників інструкції для дій у надзвичайних ситуаціях (рисунок 1.1). Внутрішні компоненти в даній моделі не розглядаються, оскільки тут важливо показати лише взаємодію системи із зовнішнім середовищем.

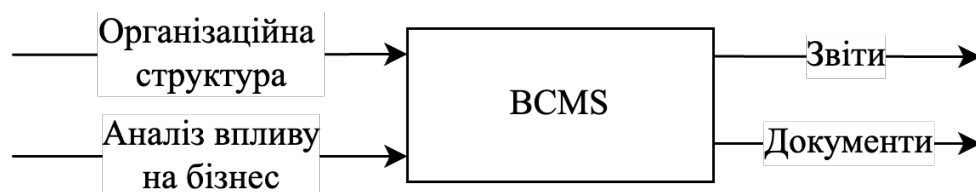


Рисунок 1.1 – Модель типу "чорний ящик"

Обмін інформацією між системою керування безперервністю бізнесу та елементами зовнішнього середовища виконується за допомогою WEB-інтерфейсу, що забезпечує простоту доступу та не потребує встановлення додаткового програмного забезпечення для організації.

Зовнішнє середовище системи включає всі фактори та компоненти, що взаємодіють з системою, але не входять до її складу. Це середовище впливає на роботу системи та визначає умови, в яких вона функціонує. Можна виділити такі основні елементи зовнішнього середовища:

- внутрішні користувачі: співробітники організації, що взаємодіють із системою для отримання інструкцій, подання звітів та участі в процесах, що забезпечують безперервність бізнесу;

- зовнішні користувачі: аудитори та клієнти, які можуть мати доступ до певних частин системи для отримання інформації стосовно стійкості організації згідно із стандартом;

- сервери та хмарні ресурси: система може бути розгорнута на фізичних

серверах або у хмарній інфраструктурі;

– мережеві ресурси: доступ забезпечується через інтернет або внутрішню корпоративну мережу; мережеві ресурси впливають на швидкість, безперервність зв'язку та безпеку даних;

– міжнародні стандарти: система повинна відповідати стандартам, таким як ISO 22301, що регулюють безперервність бізнесу; також можуть діяти вимоги стандартів інформаційної безпеки, таких як ISO 27001;

– локальне законодавство: організація повинна дотримуватись законів тієї країни або регіону, де вона працює; це може включати регулювання захисту персональних даних (наприклад, GDPR у ЄС);

– інтеграція з іншими системами: може бути інтегрована з іншими корпоративними системами, такими як ERP (системи управління ресурсами) або CRM (системи управління взаємовідносинами з клієнтами); це забезпечує обмін даними між системами та підвищує загальну ефективність.

### 1.1.3 Функціональна модель системи

Функціональна модель системи управління безперервністю бізнесу (BCMS) складається з кількох ключових модулів, які забезпечують структуроване управління та координацію процесів, необхідних для підтримки безперервності бізнесу в умовах кризових ситуацій. Ці модулі є необхідними, оскільки вони реалізують та зберігають інформацію про процеси, які повинні бути впроваджені організацією для отримання сертифікації відповідно до стандарту ISO 22301. Кожен із модулів виконує важливу роль у виконанні вимог цього стандарту, що забезпечує готовність організації до надзвичайних ситуацій. Розглянемо основні модулі системи [4]:

– облікові записи та безпека: адміністратор системи повинен мати змогу налаштовувати рівні доступу базуючись на ролях (RBAC);

– організаційна структура;

- комунікація із користувачами: задля збору аналітичних даних та повідомлень про необхідність виконати ті чи інші дії у системі повинна бути можливість зв'язку із користувачем;

- аналіз впливу на бізнес: відповідальні співробітники компанії надають дані про важливість і вплив зупинки процесів в залежності від часу;

- модуль політик: забезпечує підтримку актуальності політик безперервності бізнесу, регулює розклад, за яким мають проводитися тренування, та регулярність перегляду планів реагування;

- навчання та тестування: відповідає за навчання співробітників і тестування планів безперервності; проводить навчальні сесії для персоналу, тестування планів реагування через симуляції та оцінює готовність організації до можливих загроз;

- перегляд та підтвердження результатів аналізу: зацікавлені сторони повинні мати змогу переглянути зібрані дані та підтвердити їх коректність.

Модуль безпеки реалізує організацію та управління доступом за принципами рольової моделі контролю доступу (RBAC, role-based access control), що має кілька ключових переваг, які роблять цей підхід ефективним у великих організаціях:

- завдяки RBAC права доступу надаються не окремим користувачам, а цілим ролям. Кожен користувач отримує лише ті привілеї, які відповідають його ролі в організації. Це значно спрощує управління доступом, оскільки не потрібно вручну налаштовувати права для кожного співробітника;

- легко додавати нові ролі або змінювати існуючі в міру зростання організації або зміни її потреб. Це дозволяє організаціям масштабувати свої системи управління доступом відповідно до бізнес-процесів;

- управління доступом відбувається на рівні ролей, адміністратори можуть швидко і легко керувати доступом великої кількості користувачів. Це зменшує витрати на адміністрування системи доступу та покращує ефективність роботи;

- дозволяє організаціям легко дотримуватись вимог стандартів безпеки та

конфіденційності, таких як ISO 27001 або HIPAA.

Графічне подання функціонального опису системи можна здійснити за допомогою контекстної діаграми IDEF0 (рисунок 1.2).

Входом є інформація про внутрішні бізнес-процеси організації, яка необхідна для аналізу, оцінки ризиків та прийняття рішень щодо безперервності бізнесу. До механізмів відносяться співробітники організації, які беруть участь у впровадженні та підтримці системи BCMS та обчислювальна техніка, яка використовується для забезпечення роботи системи BCMS, зберігання даних і створення звітів. Справа знаходяться виходи системи, зокрема, звіти для аудиту та інструкції для персоналу, що допомагають співробітникам діяти в умовах надзвичайних ситуацій. Елементи управління включають вимоги стандарту та загальні вимоги до інформаційних веб-систем, такі як безпечність, доступність та продуктивність.



Рисунок 1.2 – Контекстна діаграма (рівень A-0)

Декомпозиція створення системи керуванням безперервністю бізнесу зображена на рисунку 1.3 із відображенням ключових етапів:

– визначити функціональні вимоги: початковий етап, де збираються і формуються основні вимоги до функціональності системи;

- визначити нефункціональні вимоги: встановлення критеріїв, таких як продуктивність, надійність, безпека, адаптивність, доступність та інші параметри, що не входять до прямої функціональності;
- проєктна документація: документування вимог і створення проєктної документації для чіткого плану реалізації;
- запрограмувати та протестувати систему: етап, на якому реалізується програмний продукт та проводиться його тестування;
- впровадження: проведення навчання та використання системи кінцевими користувачами.

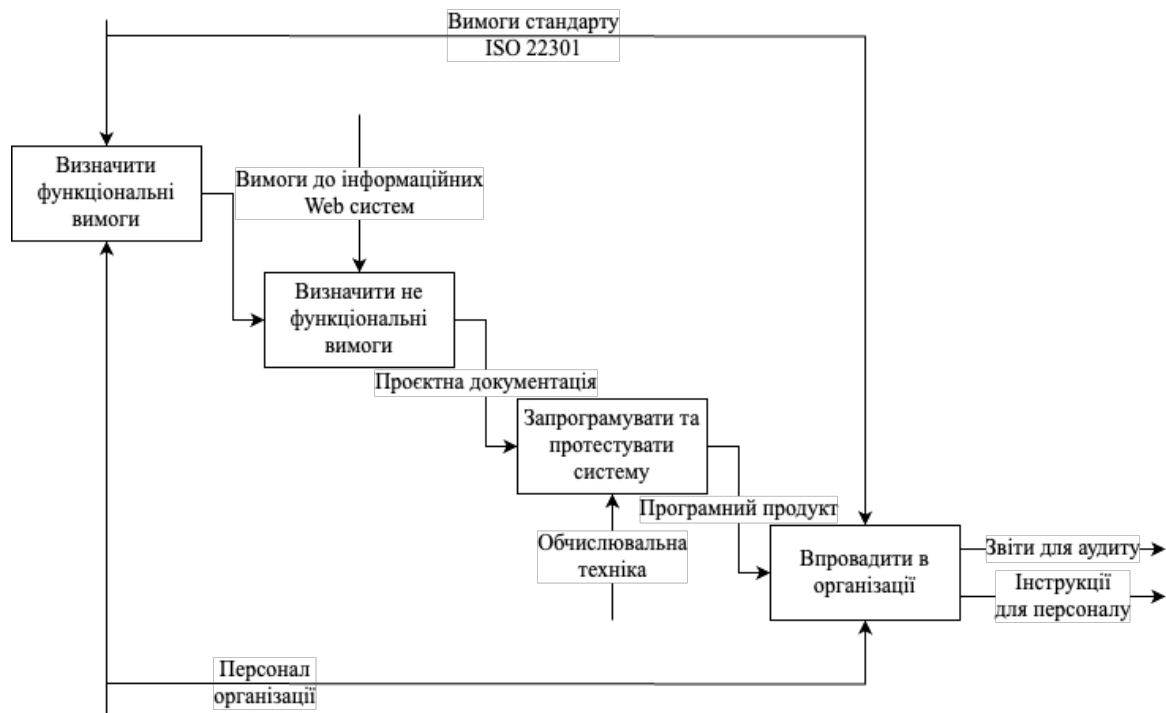


Рисунок 1.3 – Декомпозиція роботи «Створити систему BCMS»: рівень A0

Декомпозиція роботи «Запрограмувати та протестувати систему» (рисунок 1.4) складається із наступних складових:

- проєктування: створюється документ технічної специфікації, щоб окреслити технічні особливості, такі як мова програмування, апаратне забезпечення, джерела даних, архітектура та сервіси;
- кодування та впровадження: вихідний код розробляється на основі мо-

делей, логіки та специфікацій вимог, визначених на попередніх етапах;

– тестування: на цьому етапі здійснюються тести, що дозволяють виявити помилки, які необхідно усунути. Це може змусити повторити етап кодування для налагодження. Якщо система проходить інтеграцію та тестування, процес продовжується;

– експлуатація та розгортання: продукт або застосунок вважається повністю функціональним і впроваджується в робоче середовище.

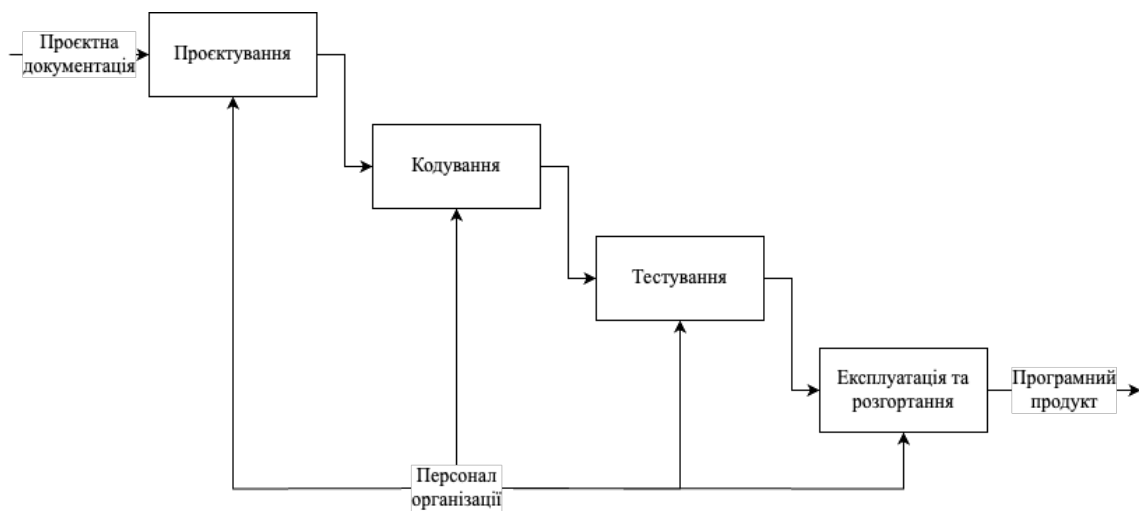


Рисунок 1.4 – Декомпозиція роботи «Запрограмувати та протестувати систему»: рівень A1

Етапи впровадження показані на рисунку 1.5. Визначаються відповідальні особи, і встановлюються їхні ролі в забезпеченні безперервності бізнесу. Наступним етапом є проведення аналізу бізнес-процесів організації для визначення критичних постачальників, ресурсів, інформаційних систем, компетенцій, тощо, які можуть впливати на безперервність аналізованого процесу.

Політики визначають розклад тренувань та розклад оновлення зібраних даних на основі критичності бізнес-процесу. Це забезпечує регулярне оновлення та вдосконалення внутрішніх правил організації в разі виникнення перешкод для звичайного функціонування.

Результатом є з одного боку звіти задля отримання сертифікації, а з іншо-

го – інструкції дій для персоналу, що слугують як для внутрішнього контролю, так і для навчання співробітників.

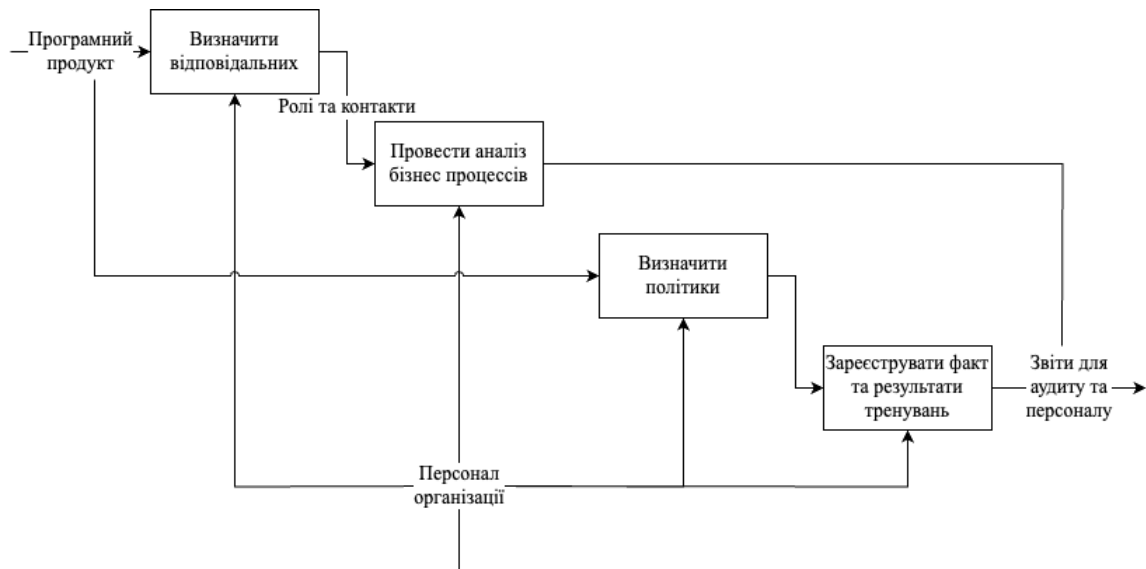


Рисунок 1.5 – Декомпозиція роботи «Впровадити в організації»: рівень А1

На рисунку 1.6 у нотації DFD-діаграми [5] зображені основні потоки даних між модулями:

- створити організаційну структуру: включає визначення ролей співробітників та їх обов'язків у контексті безперервності бізнесу;
- аналіз впливу на бізнес (BIA): використовуються дані про бізнес-процеси, постачальників, необхідні ресурси, а також інформаційні системи, необхідні для забезпечення роботи процесу. Метою цього етапу є експертна оцінка впливу інцидентів на бізнес і формування переліку критичних процесів та ресурсів. На основі цього формується план реагування на надзвичайні ситуації;
- політики: в результаті аналізу визначаються пріоритети процесів, на основі політик формуються розклади тренувань та переглядів актуальності створених планів;
- тренування: на основі політик створюється розклад тренувань та проводяться навчання персоналу. Під час тренувань збирається інформація про виконання планів, а також коригуючі дії, якщо виявлено недоліки;

– перевірка та підтвердження: відповідальними особами проводиться перевірка та підпис планів. На цьому етапі формується звітність про впровадження практик забезпечення безперервності бізнесу.



Рисунок 1.6 – Опис потоків даних між модулями системи BCMS

## 1.2 Аналіз сценаріїв вирішення задачі реалізації та впровадження системи забезпечення безперервності бізнесу

### 1.2.1 Модель аналізу проблеми

Для забезпечення безперервності бізнесу організація може обрати різні підходи до впровадження системи управління безперервністю бізнесу (BCMS). Основними альтернативами, які можуть бути розглянуті, є: ручна обробка у програмних пакетах на кшталт Excel, використання конструкторів форм таких як Google Forms, придбання спеціалізованого програмного забезпечення або розробка власної системи. Розглянемо детально кожен з цих альтернатив за такими критеріями:

- критерій 1 (K1): час виконання;
- критерій 2 (K2): ймовірність помилок;
- критерій 3 (K3): вартість впровадження;

– критерій 4 (К4): адаптивність.

Альтернативні варіанти вирішення наступні:

- альтернатива 1 (А1): ручна обробка;
- альтернатива 2 (А2): конструктор форм;
- альтернатива 3 (А3): придбання;
- альтернатива 4 (А4): власна розробка.

Ієрархічна модель вибору методу для вирішення проблеми впровадження системи забезпечення безперервності бізнесу наведена на рисунку 1.7. Фокусом ієрархії є проблема розв’язання поставленої задачі, першим рівнем ієрархії – критерії, за якими відбувається порівняння, другим рівнем – альтернативи, з яких обираємо метод розв’язання задачі.

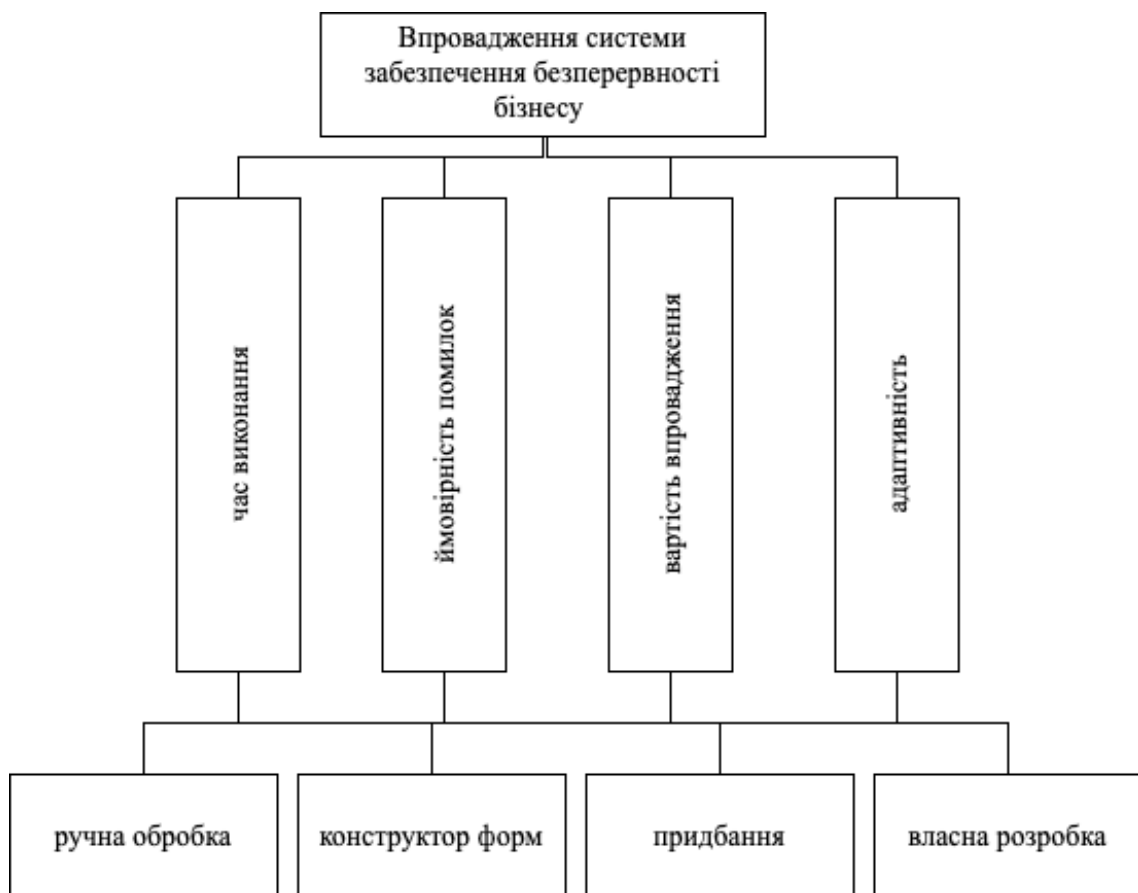


Рисунок 1.7 – Ієрархічна модель вибору методу впровадження системи забезпечення безперервності бізнесу

Розглянемо детально кожен з запропонованих альтернатив вирішення

проблеми.

Виконання вручну. Цей підхід передбачає використання паперових документів або електронних таблиць, таких як Excel, для збору та обробки даних. Основною перевагою цього варіанту є відсутність необхідності у великих початкових витратах. Однак, цей підхід має суттєві недоліки: він займає багато часу, оскільки всі операції виконуються вручну, і існує висока ймовірність людських помилок через відсутність автоматизації. Крім того, паперові документи та Excel не забезпечують автоматичної перевірки даних, що підвищує ризики втрати або некоректного введення інформації. Такий підхід може бути ефективним тільки для невеликих організацій або в тих випадках, коли немає можливості впровадити більш складні системи.

Використання конструктора форм. Цей підхід забезпечує часткову автоматизацію процесу, оскільки дані вводяться через онлайн-форми, що зменшує ймовірність помилок, властивих ручній роботі. Однак, для ефективного використання Google Forms потрібні певні технічні навички, такі як налаштування форм, вивантаження та обробка зібраних даних. Крім того, зібрані дані все одно потрібно буде завантажувати в інші системи для подальшої обробки, що додає додаткові кроки в процесі. Незважаючи на безкоштовність цього рішення, його функціональність може бути обмеженою для складних бізнес-процесів, особливо коли йдеться про масштабування або інтеграцію з іншими інструментами. Іншим важливим недоліком використання Google Forms для реалізації безперервності бізнесу є те, що дані, зібрані за допомогою цього інструменту, зберігаються на серверах сторонньої компанії. Це означає, що організація втрачає повний контроль над конфіденційною інформацією, яка може включати критично важливі бізнес-дані, персональні дані співробітників або інформацію про постачальників. У випадку роботи з конфіденційними або чутливими даними такий підхід може бути неприпустимим через ризики витоку інформації або її неналежного використання. Крім того, деякі організації, зокрема ті, які працюють у секторах із підвищеними вимогами до безпеки, можуть бути зобов'язані дотримуватися суворих нормативних вимог щодо зберігання та обробки

даних, що унеможлиблює використання сторонніх сервісів для цих цілей.

Придбання спеціалізованого програмного забезпечення. Існує можливість придбати готове спеціалізоване рішення для управління безперервністю бізнесу. Такі рішення часто містять функції для автоматизації всіх процесів, пов'язаних з аналізом ризиків, плануванням та управлінням безперервністю бізнесу, а також можуть бути налаштовані під конкретні потреби організації. Однак в Україні такі рішення є важкодоступними або взагалі відсутніми, що створює додаткові труднощі для їх впровадження. Окрім того, такі продукти можуть мати високу вартість, особливо якщо йдеться про імпортовані рішення, а також вимагатимуть додаткової підтримки та адаптації для локальних умов.

Розробка власної системи. Останньою альтернативою є розробка власної системи управління безперервністю бізнесу. Це рішення, хоча і є найбільш дорогим на початковому етапі, може мати суттєві переваги в довгостроковій перспективі. Власна система дозволяє створити програмне забезпечення, яке максимально відповідає специфічним вимогам і потребам організації. Це забезпечує високу гнучкість і якість отриманих даних, що дозволить краще управляти ризиками та підтримувати безперервність бізнесу. Інвестиції в розробку можуть окупитися завдяки зниженню операційних витрат, автоматизації процесів і поліпшенню управління кризовими ситуаціями, а також отриманий прикінцевий продукт може стати джерелом додаткових прибутків в разі продажу його іншим компаніям. Проте варто врахувати, що цей підхід потребує наявності компетентної команди розробників, тривалого часу для розробки та впровадження, а також подальшої підтримки та модернізації системи.

Якщо позначити час виконання як  $T$ , ймовірність помилок як  $P$ , вартість впровадження як  $C$ , а гнучкість та адаптивність як  $A$ , то підсумувати формальні властивості можна за наведеними нерівностями:

– за часом виконання:

$$T_{\text{вручну}} > T_{\text{конструктор}} > T_{\text{придбання}} > T_{\text{власна розробка}} ;$$

– за ймовірністю помилок:

$$P_{\text{вручну}} > P_{\text{конструктор}} > P_{\text{придбання}} \approx P_{\text{власна розробка}} ;$$

– за вартістю впровадження:

$$C_{\text{власна розробка}} \approx C_{\text{придбання}} > C_{\text{конструктор}} > C_{\text{вручну}} ;$$

– за адаптивністю

$$A_{\text{власна розробка}} > A_{\text{придбання}} > A_{\text{конструктор}} > A_{\text{вручну}} .$$

Отже розробка власної системи для управління безперервністю бізнесу є найкращим рішенням завдяки високій гнучкості та адаптації під специфічні потреби організації. Вона забезпечує контроль над даними, зменшує ризики помилок і підвищує якість отриманих даних. Незважаючи на початкові високі витрати, така система окупиться у довгостроковій перспективі через автоматизацію процесів та зниження операційних витрат. Це рішення забезпечує безпеку та надійність у порівнянні з іншими альтернативами.

### 1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій

Для аналізу ієрархії побудуємо матриці парних порівнянь моделі, а також критеріїв системи.

Матриця парних порівнянь критеріїв записана у таблиці 1.1. Останній стовпчик цієї таблиці містить результати розрахунків для вектора пріоритетів критеріїв.

Випадкова узгодженість для матриці  $n$ 'ятого порядку дорівнює  $0,9$ .

За даними таблиці 1.1:

$$- \text{індекс узгодженості } IY = \frac{4,07 - 4}{4 - 1} \approx 0,023;$$

$$- \text{відносна узгодженість } ВУ = \frac{0,023}{0,9} \approx 0,03 = 3\% .$$

Таблиця 1.1– Матриця парних порівнянь критеріїв

Критерії оцінювання	К1	К2	К3	К4	Оцінки компонентів	Вектор пріоритетів
К1	1	2	$\frac{1}{2}$	3	1,32	0,27
К2	$\frac{1}{2}$	1	$\frac{1}{5}$	2	0,67	0,14
К3	2	5	1	4	2,51	0,51
К4	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{4}$	1	0,45	0,09
Усього					4,95	

Оскільки відносна узгодженість менше ніж 0,1, то робимо висновок, що матриця парних порівнянь критеріїв побудована правильно.

Вектор локальних пріоритетів критеріїв відносно проблеми вибору дорівнює  $\vec{p}^K = (0,27; 0,14; 0,51; 0,09)$ .

Сформуємо матриці попарних порівнянь альтернатив за кожним критерієм (таблиці 1.2 – 1.5) та виконаємо розрахунки за ними.

За даними таблиці 1.2:

$$- \text{індекс узгодженості } IY = \frac{4,143 - 4}{4 - 1} \approx 0,048;$$

$$- \text{відносна узгодженість } ВУ = \frac{0,048}{0,9} \approx 0,053 = 5,3\% .$$

Таблиця 1.2 – Матриця попарних порівнянь за першим критерієм

К1	A1	A2	A3	A4	Оцінки компонентів	Вектор пріоритетів
A1	1	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{6}$	0,34	0,06
A2	3	1	$\frac{1}{3}$	$\frac{1}{4}$	0,71	0,13
A3	4	3	1	$\frac{1}{3}$	1,41	0,26
A4	6	4	3	1	2,91	0,54
Усього					5,38	

Таблиця 1.3– Матриця попарних порівнянь за другим критерієм

К2	A1	A2	A3	A4	Оцінки компонентів	Вектор пріоритетів
A1	1	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{7}$	0,30	0,05
A2	3	1	$\frac{1}{4}$	$\frac{1}{6}$	0,59	0,10
A3	6	4	1	$\frac{1}{3}$	1,68	0,28
A4	7	6	3	1	3,35	0,57
Усього					5,93	

Таблиця 1.4 – Матриця попарних порівнянь за третім критерієм

КЗ	A1	A2	A3	A4	Оцінки компонентів	Вектор пріоритетів
A1	1	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{4}$	0,34	0,06
A2	3	1	$\frac{1}{4}$	$\frac{1}{5}$	0,62	0,12
A3	6	4	1	$\frac{1}{2}$	1,86	0,35
A4	4	5	2	1	2,51	0,47
Усього					5,34	

Таблиця 1.5 – Порівняння за четвертим критерієм

КЗ	A1	A2	A3	A4	Оцінки компонентів	Вектор пріоритетів
A1	1	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{7}$	0,30	0,05
A2	3	1	$\frac{1}{3}$	$\frac{1}{4}$	0,71	0,13
A3	6	3	1	$\frac{1}{3}$	1,57	0,28
A4	7	4	3	1	3,03	0,54
Усього					5,06	

За даними таблиці 1.3:

$$- \text{індекс узгодженості } I_U = \frac{4,176 - 4}{4 - 1} \approx 0,059;$$

– відносна узгодженість  $ВУ = \frac{0,059}{0,9} \approx 0,065 = 6,5\%$ .

За даними таблиці 1.4:

– індекс узгодженості  $IУ = \frac{4,212 - 4}{4 - 1} \approx 0,071$ ;

– відносна узгодженість  $ВУ = \frac{0,071}{0,9} \approx 0,079 = 7,9\%$ .

За даними таблиці 1.5:

– індекс узгодженості  $IУ = \frac{4,151 - 4}{4 - 1} \approx 0,05$ ;

– відносна узгодженість  $ВУ = \frac{0,05}{0,9} \approx 0,056 = 5,6\%$ .

### 1.2.3 Модель вирішення проблеми

На основі результатів п. 1.2.2 виконаємо остаточні розрахунки та зробимо за ними висновок про вибір методу впровадження системи забезпечення безперервності бізнесу.

Результати розрахунків на цьому етапі наведені у таблиці 1.6.

Таблиця 1.6 – Остаточні розрахунки

Критерій Альтернатива	К1	К2	К3	К4	Узагальнені пріоритети
A1	0,06	0,05	0,06	0,05	0,06
A2	0,13	0,10	0,12	0,13	0,12
A3	0,26	0,28	0,35	0,28	0,31
A4	0,54	0,57	0,47	0,54	0,51

За даними цієї таблиці можна зробити висновок, що кращою для вирішення поставленої проблеми є четверта альтернатива – власна розробка.

### 1.3 Змістовна та формальна постановка задачі

#### 1.3.1 Змістовна постановка задачі

Сучасні організації стикаються з численними ризиками, що можуть призвести до порушення бізнес-процесів, таких як технічні збої, природні та техногенні катастрофи. Забезпечення безперервності бізнесу є критично важливим для збереження конкурентоспроможності, репутації та стабільності компанії в умовах кризових ситуацій. Багато компаній не знають, як формалізувати процес забезпечення безперервності бізнесу, оскільки стикаються зі складнощами в розумінні та управлінні внутрішніми залежностями між процесами. Зібрати всю необхідну інформацію про бізнес-процеси та ресурси, необхідні для їх підтримки, складно та займає багато часу.

Завдання полягає у створенні системи, яка дозволить автоматизувати процеси збору, аналізу та документування даних, а також надавати інструкції для співробітників під час надзвичайних ситуацій.

Основними вимогами до системи є висока гнучкість, здатність адаптуватися під специфічні потреби організації, можливість інтеграції з існуючими інформаційними системами та відповідність міжнародному стандарту ISO 22301. Розробка цієї системи сприятиме підвищенню стійкості організації до кризових ситуацій, оптимізації бізнес-процесів і забезпеченню безперервної роботи компанії навіть в умовах надзвичайних обставин.

### 1.3.2 Формальна постановка задачі

Розробка та впровадження системи забезпечення безперервності бізнесу має на меті забезпечення безперебійної роботи критично важливих бізнес-процесів організації в умовах збоїв, надзвичайних ситуацій чи катастрофічних подій, мінімізацію ризиків втрат та відновлення операційної діяльності у мінімально можливі строки.

Вихідними даними для вирішення поставленої задачі є:

- інформація про структуру організації, бізнес-процеси, що у ній протікають, та задіяні ресурси (людські, матеріальні, технічні, фінансові тощо);
- інформація про вразливості та потенційні ризики (технічні, операційні, природні тощо);
- інформація про ймовірності та наслідки виникнення збоїв;
- державні та міжнародні вимоги та стандарти (ISO 22301, ISO 31000, законодавча база тощо).

Формальна постановка задачі полягає у розробці та систематичному документуванні вимог до архітектури, інтерфейсів, структури даних, технологій, а також формалізованих процесів, необхідних для забезпечення безперервності бізнесу.

Реалізація та впровадження системи забезпечення безперервності бізнесу передбачає виконання наступних завдань:

- аналіз критично важливих процесів та їх залежності;
- оцінка можливих ризиків, що загрожують безперервності бізнесу, розробка сценаріїв впливу;
- розробка стратегії забезпечення безперервності;
- реалізація заходів щодо зниження ризиків;
- розробка системи політик періодичного перегляду планів;
- аналіз взаємозв'язків процесів з точки зору дотримання необхідного часу для відновлення.

#### 1.4 Постановка задач дослідження

Метою кваліфікаційної роботи є аналіз та побудова системи забезпечення безперервності бізнесу. Виходячи з цього можна навести наступний перелік задач, які необхідно виконати під час дослідження:

- провести огляд і системний аналіз сучасного стану задачі впровадження системи забезпечення безперервності бізнесу на підприємствах;
- дослідити методологію планування безперервності;
- розробити рекомендації щодо адаптації системи до умов конкретного підприємства;
- провести аналіз зібраних даних на наявність вразливостей у часі відновлення.

## 2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

### 2.1 Нефункціональні вимоги

Важливим аспектом, що впливає на вибір технологій при побудові інформаційних систем, є нефункціональні вимоги [6]. Вони визначають, як система повинна працювати, а не що вона робить, і охоплюють різні аспекти. В рамках даної роботи розглянемо такі вимоги як доступність, здатність до розгортання, адаптивність, продуктивність, безпека, можливість протестувати та зручність використання.

**Доступність.** Це властивість системи, яка гарантує, що вона буде функціонувати тоді, коли виникне потреба у її використанні. Веб-система підходить для цього завдяки тому, що доступ до неї можна забезпечити різними шляхами, зокрема через мобільний інтернет, корпоративну мережу та інші канали зв'язку. Крім того, є можливість використовувати додаткові дзеркала серверів у різних локаціях, що забезпечить доступність навіть у разі відмови одного з дата-центрів, гарантуючи безперервність роботи системи.

**Здатність до розгортання.** Система накладає вимоги лише до серверної компоненти. Для кінцевих споживачів необхідним є лише сучасний браузер і доступ до мережі Internet.

**Адаптивність.** Це властивість системи, яка відображає її здатність пристосовуватися до змін. Кожна організація має унікальну структуру та різні вимоги до забезпечення безперервності бізнесу. Стандарт ISO 22301 лише в загальних термінах визначає, які аспекти бізнесу повинні бути проаналізовані, проте сама організація визначає, які елементи слід включати до системи управління безперервністю бізнесу. Тому адаптивність є важливою властивістю, яку можна забезпечити за допомогою використання мови програмування високого рівня, такої як C#.

Іншим методом підвищення адаптивності системи є забезпечення максимальної незалежності модулів один від одного. Це означає, що зміни у

процедурі збору інформації про вплив на бізнес не повинні вимагати змін у таких модулях, як модуль керування політиками або модуль тренувань. Також пропонується відокремити деталі доступу до бази даних від бізнес-логіки додатку. Одним із засобів реалізації цього підходу є використання ORM (Object-Relational Mapping) [7], такого як Entity Framework. Завдяки цьому бізнес-логіка додатку буде незалежною від конкретних рішень щодо зберігання та структури даних, що дозволить легше змінювати або оновлювати систему без впливу на інші модулі.

Система також має підтримувати високий рівень варіативних налаштувань та мати точки інтеграції для адаптації до різних типів організацій. Ізоляція даних клієнтів є критичною вимогою. Це забезпечує відповідність вимогам безпеки клієнтів, які можуть вимагати додаткових заходів для захисту своїх даних. Ізоляція даних також гарантує, що дані одного клієнта залишаються конфіденційними і недоступними для інших клієнтів системи, що є важливою умовою для дотримання нормативних актів і стандартів безпеки, таких як GDPR або ISO 27001.

Підхід до розробки має ґрунтуватися на agile з короткими ітераціями та регулярним тестуванням з боку бізнесу для перевірки правильності обраного напрямку розробки. Для підвищення адаптивності деякі модулі системи можуть бути відокремлені в окремі мікросервіси з власними циклами розробки та розгортання.

Одним з таких модулів є модуль повідомлень та комунікації, оскільки він вимагає численних інтеграцій із зовнішніми системами, такими як Twilio або SendGrid. Цей модуль може мати власну чергу виконання і масштабуватися незалежно від основної системи. Під час інциденту може виникнути потреба швидко розіслати повідомлення великій кількості співробітників, тоді як навантаження на інші функції системи, такі як аналіз впливу на бізнес, залишається стабільним або знижується.

Іншим незалежним сервісом може бути модуль керування обліковими записами, для якого можна використовувати Microsoft Entra ID у разі розгортання

системи на Microsoft Azure. Це дозволяє масштабувати систему, забезпечуючи адаптивність до специфічних вимог і умов різних організацій.

Продуктивність характеризує чи система виконує поставлені задачі у прийнятний час. Цей час може залежати як від кількості одночасних користувачів, так і від конкретної функціональності, що використовується. Для веб-систем зазвичай встановлюється метрика, за якою 95% запитів повинні оброблятися швидше, ніж за 1 секунду. Система керування безперервністю бізнесу не є високонавантаженою і має обслуговувати орієнтовно до 100 конкурентних користувачів. Однак деякі запити, такі як запити на формування звітів, можуть тривати довше, і це вважається прийнятним, оскільки подібні операції є більш ресурсномісткими.

Безпека. Специфічних вимог щодо безпеки система управління безперервністю бізнесу не має. Тому для забезпечення базового рівня захисту використовується стандартний для індустрії підхід – «OWASP Top 10». Це список найбільш критичних ризиків безпеки для веб-додатків, рекомендований Open Web Application Security Project (OWASP), який включає заходи для запобігання найбільш поширеним вразливостям, таким як ін'єкції, проблеми з автентифікацією та управлінням сесіями, неправильно налаштовані безпечні конфігурації тощо.

Можливість протестувати. Система не використовує спеціалізованого обладнання та не потребує специфічних умов для тестування. Тому стандартна комбінація модульного, інтеграційного, регресійного тестування є достатньою.

Зручність використання – це ключова вимога для системи управління безперервністю бізнесу, оскільки її цільова аудиторія складається переважно з нетехнічного персоналу організації. Тому інтерфейс повинен бути максимально простим та інтуїтивно зрозумілим і таким, що направляє дії користувача в необхідному напрямку. Дуже часто такі системи використовуються для державних установ, де всі інтерфейси обов'язково повинні бути адаптовані для людей з обмеженими можливостями. Тому важливо враховувати рекомендації щодо доступності веб-вмісту Web Content Accessibility Guidelines (WCAG), які

забезпечують доступність системи для всіх категорій користувачів, незалежно від їхніх фізичних можливостей.

## 2.2 Розгортання

Для забезпечення стабільної роботи та доступності системи пропонується використовувати хмарний хостинг Microsoft Azure. Основні переваги Azure включають:

- широкий набір послуг: Azure пропонує понад 200 продуктів і хмарних сервісів, що дозволяє створювати, запускати та керувати додатками на різних платформах, включаючи Windows, Linux, iOS та Android;
- глобальна доступність: завдяки розгалуженій мережі дата-центрів у понад 60 регіонах світу, Azure забезпечує високу доступність та низьку затримку для користувачів у різних куточках планети;
- масштабованість та гнучкість: платформа дозволяє легко масштабувати ресурси відповідно до потреб бізнесу, забезпечуючи ефективне управління навантаженнями та оптимізацію витрат;
- інтеграція з продуктами Microsoft: Azure тісно інтегрується з іншими продуктами Microsoft, такими як Office 365, Dynamics 365 та Active Directory, що спрощує управління та підвищує продуктивність;
- безпека та відповідність стандартам: Azure забезпечує багаторівневий захист даних та інфраструктури, відповідаючи міжнародним стандартам безпеки, таким як ISO 27001 та GDPR;
- підтримка різних мов програмування та фреймворків: платформа підтримує широкий спектр мов програмування та фреймворків, що дозволяє розробникам використовувати знайомі інструменти та технології;
- економічна ефективність: модель оплати «pay-as-you-go» дозволяє сплачувати лише за використані ресурси, що допомагає оптимізувати витрати та уникнути зайвих інвестицій у інфраструктуру.

Пропонується використовувати наступну архітектуру веб-додатку, розгорнутого у Microsoft Azure з використанням різних сервісів для забезпечення доступності, безпеки та моніторингу. Ось опис основних компонентів:

- Azure DNS: сервіс для керування доменними іменами, який спрямовує запити користувачів до Application Gateway;
- Application Gateway: шлюз, що забезпечує балансування навантаження та маршрутизацію запитів до додатку;
- WAF (Web Application Firewall): веб-брандмауер, який фільтрує та захищає додаток від поширених вразливостей, таких як SQL-ін'єкції, XSS-атаки тощо;
- App Service: хостингова служба для розгортання та управління веб-додатками.
- PostgreSQL: база даних, яка зберігає дані, необхідні для роботи додатку;
- Azure AD (Azure Active Directory): сервіс авторизації користувачів, який забезпечує доступ до додатку та інші функції безпеки;
- App Insights: сервіс для моніторингу продуктивності та збору аналітики додатку, який дозволяє відслідковувати помилки, ефективність додатку та інші метрики.

Ця архітектура є стійкою та безпечною, оскільки включає рівень безпеки (WAF), балансування навантаження (Application Gateway), керування ідентичністю (Azure AD) та моніторинг. Перелік сервісів та їх діаграма представлена на рисунку 2.1.

Маючи перелік використаних сервісів та їх рівні доступності, можна визначити очікуваний рівень доступності системи в цілому, використовуючи формулу:

$$SLA_{total} = \prod_{i=1}^n SLA_i,$$

де  $SLA_i$  – рівень доступності  $i$ -го сервісу;

$n$  – кількість використаних сервісів.

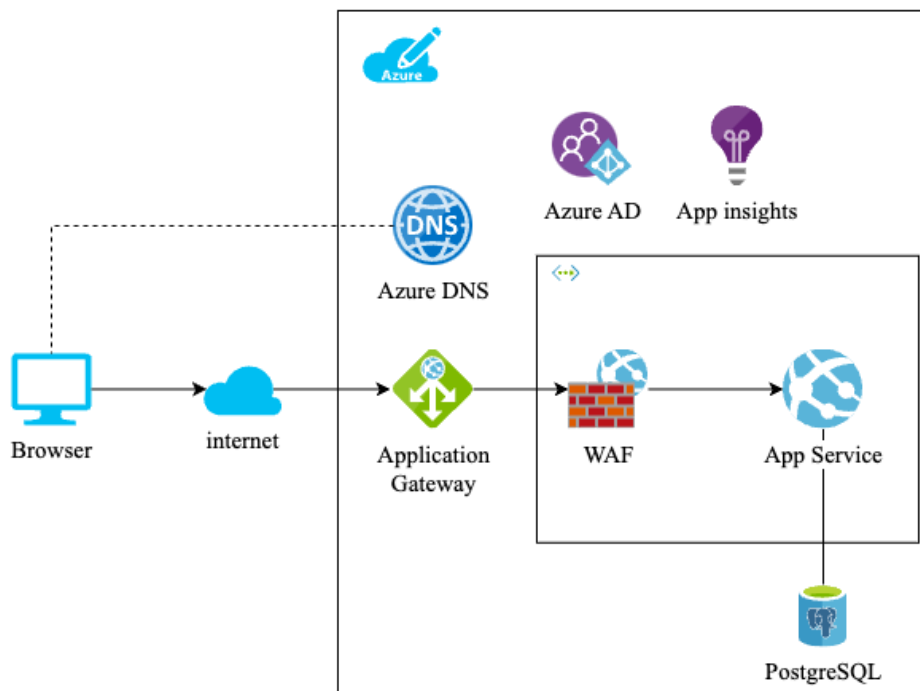


Рисунок 2.1– Розгорання системи

Перелік значень для задіяних сервісів наведено у таблиці 2.1 [8]. До переліку включено лише критично необхідні сервіси, оскільки перебої в роботі AppInsights не впливають на працездатність основної системи. Сервіс Azure DNS має широку мережу кешуючих серверів, що дозволяє знехтувати короткочасними зупинками в його роботі.

Таблиця 2.1 – Рівні умов використання сервісів

Сервіс	Заявлений рівень	Час недоступності в тиждень
Application Gateway	99,95%	5 хвилин
WAF	99,99%	1,01 хвилин
App Service	99,99%	1,01 хвилин
Database for PostgreSQL	99,99%	1,01 хвилин
Azure AD	99,99%	1,01 хвилин

Таким чином, розрахунковий рівень доступності складає:

$$SLA_{total} = \prod_{i=1}^5 SLA_i = 0,9995 \cdot 0,9999^4 = 0,9991,$$

тобто 99,91%. Це означає, що очікується приблизно 9 хвилин на тиждень коли система може бути недоступною.

Оцінка вартості володіння запропонованою архітектурою наведена у таблиці 2.2. Вона виходить із того, що система буде працювати безперервно в режимі нормального навантаження. Також варто взяти до уваги те, що не враховується вартість надсилання повідомлень користувачам, оскільки вона може змінюватись залежно від обраного провайдера та країни перебування.

Таблиця 2.2 – Оцінка вартості володіння

Категорія	Тип сервісу	Регіон	Опис	Оцінка вартості в місяць
Compute	App Service	West Europe	Standard Tier; 1 S2 (2 Core(s), 3.5 GB RAM, 50 GB Storage)	\$138,70
Databases	Azure Database for PostgreSQL	West Europe	Single Server Deployment, General Purpose Tier, 1 Gen 5 (2 vCore) x 730 Hours, 5 GiB Storage, 100 GiB Additional Backup storage - LRS redundancy	\$155,44
Networking	Application Gateway	West Europe	Web Application Firewall	\$103,02
<b>Усього</b>				<b>\$397,16</b>

Вартість може бути знижена на період тестування за рахунок зменшення вартості бази даних та обчислювальних ресурсів. Також є можливість скористатися знижками за рахунок резервування [9].

### 2.3 Концептуальна архітектура застосунку

Беручи до уваги те, що система має високі вимоги до зручності та простоти розгортання, розробка Web-застосунку виглядає як очевидний вибір.

Враховуючи необхідність забезпечення високої адаптивності та ізоляції між компонентами системи пропонується використовувати «шестикутну архітектуру» (Hexagonal Architecture) [10]. Концептуальна модель такої архітектури зображена на рисунку 2.2:

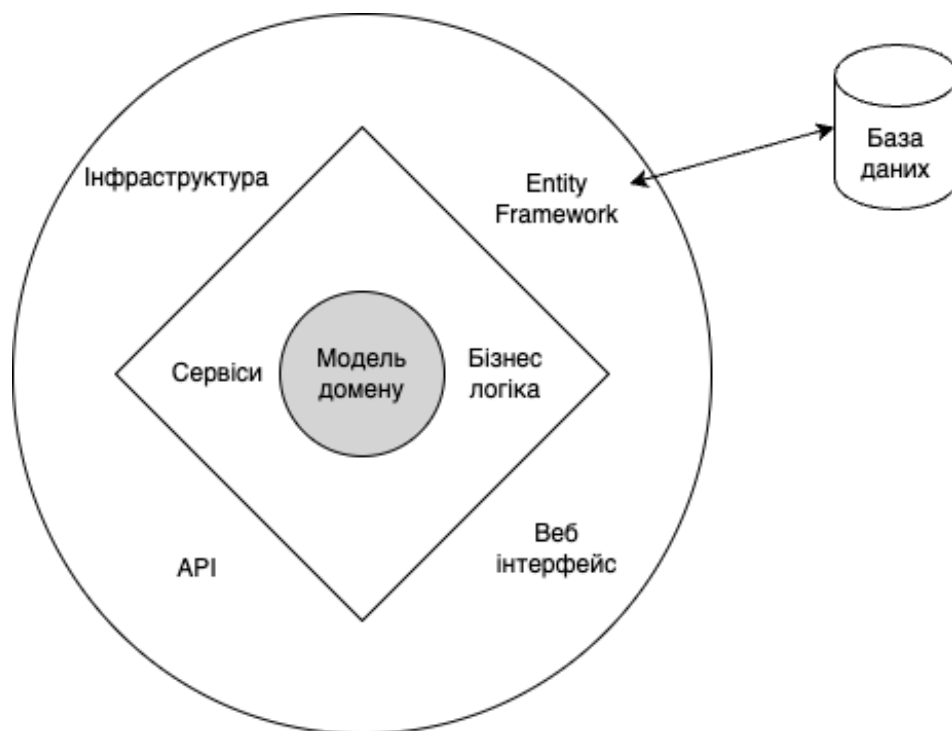


Рисунок 2.2 – Шестикутна архітектура

Ця архітектура також відома як «архітектура портів і адаптерів». Цей стиль дозволяє чітко розмежовувати бізнес-логіку від зовнішніх систем та інфраструктурних деталей (бази даних, зовнішні сервіси, інтерфейси користувача). У контексті ВСMS це дає можливість легко змінювати інтеграції з різними зовнішніми системами, такими як служби повідомлень, зовнішні каталоги ресурсів або контактів, не змінюючи основну бізнес-логіку.

## 2.4 Вимоги стандарту ISO 22301 до системи керування безперервністю бізнесу

Міжнародний стандарт використовує модель безперервного поліпшення (плануй, роби, перевіряй, дій) задля планування, впровадження, виконання, нагляду, перегляду, підтримки та сталого покращення ефективності системи забезпечення безперервності бізнесу (рисунок 2.3).

Вільям Едвардс Демінг описує цикл PDSA (Plan-Do-Study-Act), який є інструментом безперервного вдосконалення процесів. Цей цикл, відомий також як «Цикл Демінга», включає чотири етапи: планування змін (Plan), їх впровадження (Do), аналіз результатів (Study) і прийняття дій на основі отриманих висновків (Act). PDSA є основним елементом системного підходу до управління якістю, який допомагає організаціям досягати стабільності, впроваджуючи методи контролю і корекції на кожному етапі процесу [11].

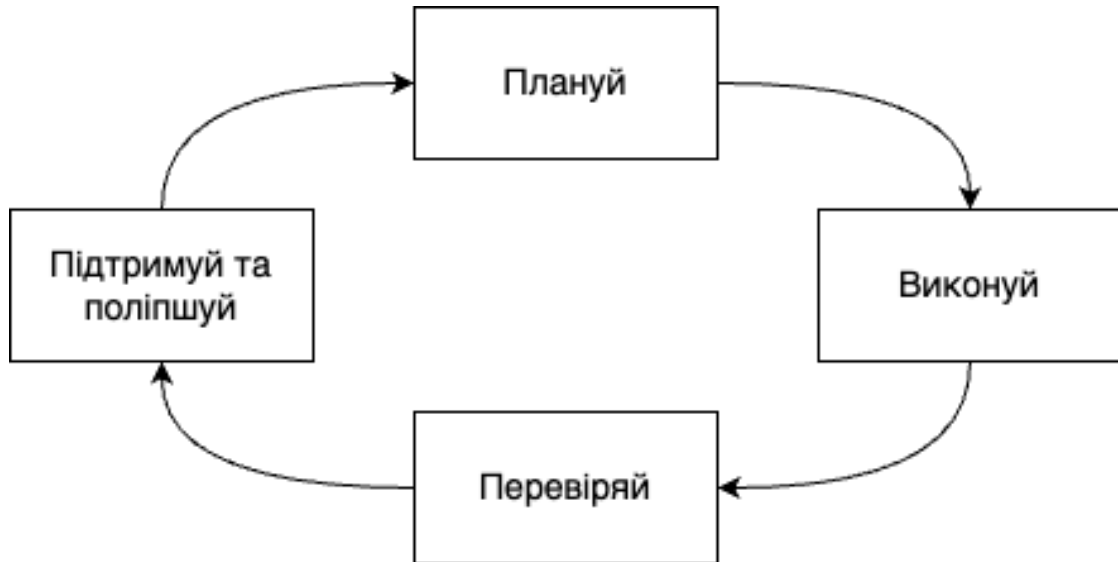


Рисунок 2.3 – Модель «Цикл Демінга»

Це забезпечує певну узгодженість з іншими стандартами систем управління, такими як ISO 9001 (Системи управління якістю) та ISO/IEC 27001 (Системи управління інформаційною безпекою). Пояснення щодо кожного із етапів цього циклу наведено у таблиці 2.3.

Таблиця 2.3 – Пояснення моделі «Цикл Демінга»

Плануй	Встановити політику, цілі, завдання, засоби контролю, процеси та процедури щодо покращення безперервності бізнесу, щоб забезпечити результати, які відповідають загальним політикам і цілям організації.
Виконуй	Впроваджувати та здійснювати політику безперервності бізнесу, засоби контролю, процеси та процедури.
Перевірй	Здійснювати моніторинг та перегляд результатів діяльності у відповідності до політики та цілей безперервності бізнесу, звітувати про результати керівництву для перегляду, а також визначати та затверджувати заходи для усунення недоліків та покращення.
Підтримуй та поліпшуй	Підтримувати та вдосконалювати систему управління безперервністю бізнесу (BCMS) шляхом виконання коригувальних заходів на основі результатів перегляду керівництвом.

Інформаційна система повинна допомагати та спрямовувати дії фахівця з безперервності бізнесу в організації. Для підвищення ефективності розробки пропонується використовувати підхід орієнтованого на домен проектування (Domain Driven Design) [12], який передбачає використання однакових назв для об'єктів як з точки зору програмування, так і з точки зору бізнесу, що називається універсальною мовою. Такий підхід дозволяє всім учасникам процесу – як технічним спеціалістам, так і представникам бізнесу, краще розуміти та узгоджувати свої дії, що забезпечує більш ефективну розробку та підтримку системи управління безперервністю бізнесу. Наступні сутності є важливими для розробника:

– діяльність: це процес або набір процесів, що здійснюються організацією (або від її імені), які виробляють або підтримують один або кілька продуктів і послуг;

- план забезпечення безперервності: це задокументовані процедури, які направляють організацію в разі реагування, відновлення та повернення до заздалегідь визначеного рівня операцій після порушення діяльності;
- коригувальна дія: це дія, спрямована на усунення причини невідповідності та запобігання її повторенню;
- максимальний допустимий період переривання (МТПД): це час, протягом якого негативні наслідки, що можуть виникнути внаслідок ненадання продукту/послуги або невиконання діяльності, стають неприпустимими;
- мінімальна мета безперервності бізнесу (МВСО): це мінімальний рівень послуг та/або продуктів, який є прийнятним для організації для досягнення її бізнес-цілей під час порушення діяльності;
- цільовий час відновлення (RTO): це період часу після інциденту, протягом якого послуга має бути відновлена;
- ресурси: всі активи, люди, навички, інформація, технології (включаючи обладнання та технічні засоби), приміщення, постачання та інформація (як в електронному вигляді, так і ні), які організація повинна мати у своєму розпорядженні для використання, коли це необхідно, з метою функціонування та досягнення своїх цілей;
- компетенції: організація має визначати необхідну компетентність персоналу, забезпечувати її на основі освіти, навчання та досвіду, а також вживати заходів для досягнення необхідного рівня компетентності. Вона також повинна зберігати документальні докази компетентності та вжитих заходів.

## Висновки за розділом 2

У даному розділі було детально розглянуто нефункціональні вимоги до інформаційних систем, зокрема таких, що застосовуються для забезпечення безперервності бізнесу. Описано ключові аспекти, такі як доступність, здатність до розгортання, адаптивність, продуктивність, безпека, можливість тестування

та зручність використання. Проаналізовано, як сучасні веб-технології та хмарні рішення, такі як Microsoft Azure, сприяють спрощенню процесу розгортання системи за рахунок того, що фізична інфраструктура серверів повністю є відповідальністю власника хмарного сервісу, а не розробника системи.

Також було приділено увагу модульній архітектурі, яка дозволяє легко адаптувати систему до змін без впливу на інші компоненти. Розглянуто методи підвищення гнучкості, такі як використання ORM (Object-Relational Mapping) та впровадження незалежних мікросервісів, що спрощують інтеграцію зовнішніх рішень. Описано підходи до забезпечення безпеки на основі OWASP Top 10, які допомагають запобігти основним вразливостям і забезпечити захист даних.

Розглянуто продуктивність системи, що досягається оптимізацією ресурсів та встановленням чітких метрик для обробки запитів. Описано важливість тестування системи з використанням стандартних підходів, а також врахування стандартів доступності WCAG для забезпечення зручності користування системою.

Таким чином, цей розділ надав теоретичну основу для подальшої практичної реалізації системи управління безперервністю бізнесу, враховуючи сучасні вимоги та технологічні можливості.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Платформа .net як інструмент розробки корпоративних веб-додатків

Платформа .NET – це безкоштовна, кросплатформна, з відкритим кодом платформа для розробників, призначена для створення різноманітних додатків. Вона може запускати програми, написані декількома мовами, серед яких найбільш популярною є C#. Платформа .NET базується на високопродуктивному середовищі виконання, яке використовується багатьма додатками високого масштабу [13].

Платформа .NET була розроблена для забезпечення продуктивності, швидкодії, безпеки та надійності. Вона надає автоматичне керування пам'яттю за допомогою збирача сміття (GC), пропонує підтримку паралельності через `async/await` та примітиви `Task`. Платформа включає великий набір бібліотек, що мають широкий функціонал і оптимізовані для високої продуктивності на різних операційних системах та архітектурах процесорів.

Платформа .NET має такі ключові принципи проектування:

- продуктивність охоплює повний стек з використанням середовища виконання, бібліотек, мов та інструментів, що разом покращують досвід розробника;
- підтримуються як статичний, так і динамічний код, що забезпечує широкий набір сценаріїв;
- код є портативним між платформами (ОС та архітектура процесора), при цьому націлювання на конкретну платформу дозволяє спеціалізацію та оптимізацію;
- адаптивність до різних програмних доменів (хмара, клієнтські додатки, ігри) забезпечується спеціалізованими реалізаціями загальної програмної моделі;
- віддається перевага галузевим стандартам, таким як OpenTelemetry та gRPC, перед унікальними рішеннями.

Платформа .NET підтримується Microsoft та спільнотою. Її регулярно оновлюють, щоб користувачі могли розробляти безпечні та надійні додатки.

Платформа .NET забезпечує повний набір інструментів для розробки корпоративних веб-додатків, що дозволяє створювати ефективні, масштабовані та надійні хмарні сервіси. Її компоненти підтримують ключові аспекти розробки сучасних веб-додатків, слідуючи принципам 12-факторного додатку [14], що забезпечує стандартизацію, продуктивність та гнучкість:

- єдина база коду для усіх аспектів життєвого циклу;
- декларативне керування необхідними бібліотеками за допомогою Nuget;
- можливість відокремлення конфігурації від коду і надання її через змінні середовища;
- .NET Aspire як середовище для налаштування доступу між сервісами;
- AzureDevops або Github Actions можуть бути використані як середовище для компіляції, конфігурації та розгортання;
- усі дані зберігаються у реляційній базі (stateless);
- є повністю самостійним і не залежить від динамічного підключення веб-сервера до середовища виконання для створення веб-сервісу;
- здатність до горизонтального масштабування;
- швидкій запуск та зупинка процесу: в хмарному середовищі сервіс має можливість бути швидко переміщеним із одного сервера на інший;
- надання логів назовні через потік stdout;
- адміністративні процедури: поєднання разових операцій (таких як зміна схеми бази даних) в єдиному пакеті із версією.

Серед усього різноманіття доступних бібліотек пропонується використовувати Blazor та EntityFrameworkCore.

Blazor – це фреймворк для створення веб-інтерфейсів на C#, що дозволяє створювати інтерактивні SPA-додатки (Single Page Applications), подібно до підходів у JavaScript-фреймворках (таких як React або Angular). З його допомогою можна створювати швидкі та зручні інтерфейси користувача, які працюють безпосередньо в браузері, що підходить для корпоративних веб-додатків і за-

безпечує високу інтерактивність. Використання Blazor також дозволяє значно спростити процес розробки інтерфейсу, оскільки розробники можуть використовувати єдину мову для бекенду і фронтенду, що скорочує час на навчання розробників і зменшує кількість залежностей які потрібно буде підтримувати протягом життєвого циклу сервісу.

EntityFrameworkCore (EFCore) є ORM (Object-Relational Mapping) фреймворком, який забезпечує легкий і ефективний доступ до баз даних. EF Core дозволяє працювати з базами даних, такими як SQLServer, PostgreSQL, MySQL та іншими, використовуючи об'єктно-орієнтований підхід. Завдяки EF Core розробники можуть маніпулювати даними як з об'єктами, без необхідності писати SQL-запити вручну, що дозволяє пришвидшити процес розробки та забезпечує гнучкість в управлінні даними. Крім того, EF Core підтримує міграції бази даних, що полегшує процес оновлення структури бази даних у відповідності до змін у додатку.

У контексті використання .NET Aspire, який застосовує контейнеризовану архітектуру, бази даних є ефемерними та можуть бути відтворені в будь-який момент. EntityFrameworkCore використовує функцію міграцій для створення та оновлення схем баз даних. Оскільки бази даних відтворюються під час запуску додатка, необхідно застосовувати міграції для ініціалізації схеми бази даних кожного разу при старті додатка. Це досягається шляхом реєстрації сервісу міграції в додатку, який виконується під час запуску [15].

.NET Aspire дозволяє спростити процес розгортання середовища розробки, полегшуючи підключення та налаштування для членів команди. Це забезпечує узгодженість середовищ розробки, спрощує інтеграцію та знижує витрати часу на підготовку інфраструктури. У поєднанні з ASP.NET, який підтримує розгортання на хмарних платформах, таких як Microsoft Azure, .NET Aspire дозволяє командам швидко масштабувати і підтримувати додатки в ізольованих контейнерних середовищах, таких як Docker.

### 3.2 Шаблони проєктування та архітектурні патерни

З метою спрощення довгострокового супроводу продукту спільнота розробників використовує підходи, що є спільними для багатьох корпоративних застосунків [7]. Доцільними будуть такі: Unit of work, Repository, Identity Map, Entity, Command Query Responsibility Segregation (CQS) [12] та Dependency Injection.

Базовий тип DbContext призначений для виконання однієї одиниці роботи (unit-of-work) [7], що включає створення екземпляра, відстеження змін, збереження змін у базі даних та видалення екземпляра. В ASP.NET Core додатках DbContext реєструється в контейнері залежностей за допомогою методу AddDbContext, що дозволяє налаштувати підключення до бази даних та інші параметри. Важливо правильно утилізувати екземпляри DbContext після використання для звільнення ресурсів та уникнення витоків пам'яті.

Платформа .NET пропонує вбудовану бібліотеку для Dependency Injection (DI), що значно спрощує управління залежностями між компонентами додатку. DI дозволяє створювати розширювану та здатну до юніт-тестування архітектуру, де окремі компоненти легко взаємодіють без жорстких зв'язків на конкретні реалізації типів. Це сприяє модульності додатку та спрощує його підтримку. Вбудовані бібліотеки логування, у свою чергу, надають зручні інструменти для відстеження дій користувачів, обробки помилок та аналізу продуктивності додатку.

Принцип Command Query Separation (CQS) передбачає чіткий поділ операцій на два типи: команди (commands), які змінюють стан системи, і запити (queries), які просто отримують дані без внесення змін. Такий підхід забезпечує кращу структурованість коду та полегшує його розуміння, тестування й підтримку. Для реалізації цього принципу можна використовувати бібліотеку MediatR, яка спрощує розробку завдяки інтеграції з механізмом впровадження залежностей (Dependency Injection). MediatR надає базові типи для команд і запитів, що дозволяє розробникам швидко почати роботу з цими концепціями.

Крім того, MediatR підтримує використання аспектно-орієнтованого програмування, що особливо корисно для додаткових функцій, таких як логування змін у системі. Наприклад, це дає можливість автоматично фіксувати всі дії, що призводять до змін у стані системи, протягом її роботи. Таким чином, бібліотека забезпечує не лише чітке дотримання принципу CQS, але й полегшує підтримку, розширення та моніторинг системи.

### Висновки за розділом 3

У даному розділі було розглянуто ключові аспекти використання платформи .NET для розробки сучасних веб-додатків. Платформа .NET демонструє високий рівень продуктивності, адаптивності та безпеки завдяки своїм принципам проєктування, які включають підтримку кросплатформності, використання галузевих стандартів та забезпечення масштабованості додатків. Особливу увагу приділено можливостям платформи щодо створення корпоративних веб-додатків, які відповідають сучасним вимогам до гнучкості, швидкодії та інтеграції.

Також було детально описано інструменти платформи, такі як Blazor та EntityFrameworkCore, які дозволяють спросити розробку інтерфейсу та взаємодію з базами даних відповідно. Використання Blazor дозволяє створювати інтерактивні SPA-додатки, які забезпечують єдину мову розробки для фронтенду і бекенду. EntityFrameworkCore, у свою чергу, забезпечує об'єктно-орієнтований підхід до управління базами даних, що полегшує процес розробки та підтримки додатків.

Увага була приділена також контейнеризованій архітектурі .NET Aspire, яка забезпечує узгодженість середовищ розробки та ефективність у розгортанні додатків. Інтеграція з хмарними сервісами, такими як Microsoft Azure, дозволяє швидко масштабувати додатки та підтримувати їх у різних середовищах, що відповідає сучасним вимогам до хмарної інфраструктури.

Таким чином, платформа .NET демонструє значні переваги для створення сучасних, надійних і масштабованих веб-додатків. Вона надає розробникам потужний інструментарій для вирішення широкого спектра завдань, що робить її однією з найкращих платформ для розробки корпоративних додатків.

## 4 РОЗГОРТАННЯ ПРОБНОЇ ВЕРСІЇ СИСТЕМИ

### 4.1 Стандартне тестове середовище у Microsoft Azure

Проекти .NET Aspire створені на основі принципів незалежності від хмарних платформ, що забезпечує гнучкість розгортання на різних платформах, які підтримують .NET і контейнери. Користувачі можуть адаптувати надані рекомендації для розгортання в інших хмарних середовищах або для локального хостингу. Процес ручного розгортання, хоча й можливий, включає велику кількість кроків і є схильним до помилок. Тому користувачі надають перевагу використанню CI/CD-пайплайнів та інструментів, орієнтованих на певну хмарну інфраструктуру, для більш зручного і ефективного розгортання відповідно до обраної інфраструктури [16]. Використавши команду `azd up` є можливість створити усі необхідні для роботи системи у тестовому режимі сервіси (рисунок 4.1).

```

SUCCESS: Your application was removed from Azure in 10 minutes 40 seconds.
OpsContinuity.AppHost git:(master) * azd up

Packaging services (azd package)

Provisioning Azure resources (azd provision)
Provisioning Azure resources can take some time.

Subscription: Azure for Students (f597afcf-2bc4-4260-8b1c-07921a520fc8)
Location: West Europe

You can view detailed progress in the Azure Portal:
https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2Ff597afcf%2Fmicrosoft.Resources%2Fdeployments%2F0psContinuity-1731935654

(✓) Done: Resource group: rg-OpsContinuity (741ms)
(✓) Done: Log Analytics workspace: law-omboeso2lfafk (932ms)
(✓) Done: Container Registry: acromboeso2lfafk (9.146s)
(✓) Done: Storage account: volomboeso2lfafk (22.976s)
(✓) Done: Container Apps Environment: cae-omboeso2lfafk (1m51.518s)

Deploying services (azd deploy)

(✓) Done: Deploying service migrations
- No endpoints were found

(✓) Done: Deploying service postgres
- Endpoint: https://postgres.internal.gentlemeadow-c51c42e3.westeurope.azurecontainerapps.io/

(✓) Done: Deploying service webfrontend
- Endpoint: https://webfrontend.gentlemeadow-c51c42e3.westeurope.azurecontainerapps.io/

Aspire Dashboard: https://aspire-dashboard.ext.gentlemeadow-c51c42e3.westeurope.azurecontainerapps.io

SUCCESS: Your up workflow to provision and deploy to Azure completed in 4 minutes 53 seconds.
OpsContinuity.AppHost git:(master) * █

```

Рисунок 4.1 – Розгортання тестової версії у Microsoft Azure

Це створює ресурси, зображені на рисунку 4.2.

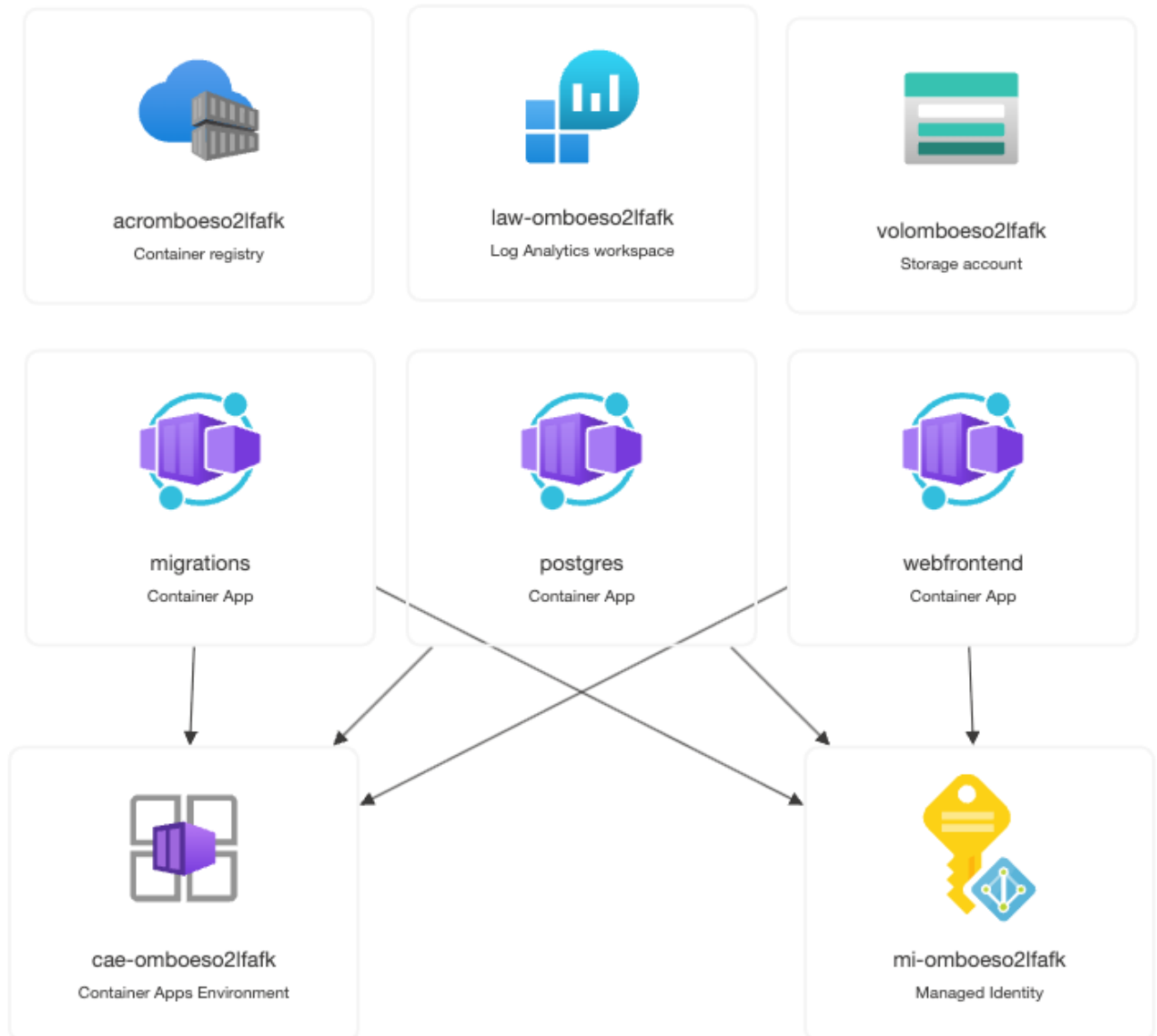


Рисунок 4.2 – Тестове розгортання інфраструктури в хмарному середовищі

Хоча такий підхід може бути застосований для швидкої перевірки працездатності прототипу, він не задовольняє всіх вимог системи. У цьому варіанті використовується контейнер для бази даних замість Azure-сервісу, що позбавляє систему вбудованих механізмів резервного копіювання та автоматичного оновлення версій серверного коду. Крім того, відсутність Application Gateway унеможливорює використання Web Application Firewall.

## 4.2 Інтерфейс системи

Відповідальний за забезпечення безперервності бізнесу розпочинає роботу із системою, формуючи організаційну структуру компанії. Зазвичай ця структура має вигляд однобічної ієрархії, яку можна візуалізувати у форматі дерева. Система не накладає обмежень на кількість елементів чи глибину такої структури, а на будь-якому рівні доступна можливість розпочати аналіз впливу на бізнес. Стандарт ISO 22301 дійсно наголошує на тому, що організація має самостійно визначати, які активності включати до аналізу забезпечення безперервності бізнесу. Приклад такої структури наведений на рисунку 4.1.

Далі формується перелік доступних ресурсів, систем та інших елементів, необхідних для організації. Цей список можна адаптувати відповідно до специфічних потреб конкретної компанії, та розширювати впродовж аналізу впливу на бізнес. Інтерфейс наведено на рисунку 4.2.

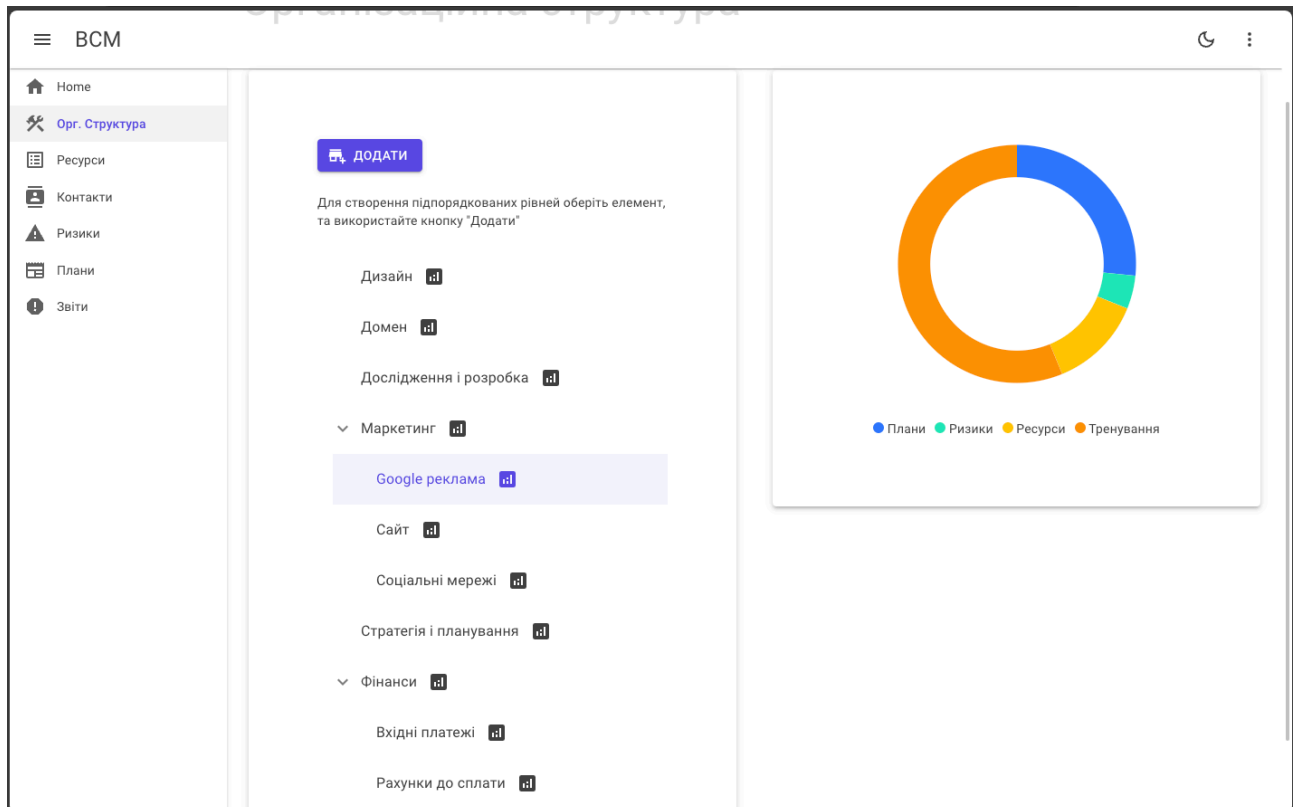


Рисунок 4.1 – Організаційна структура

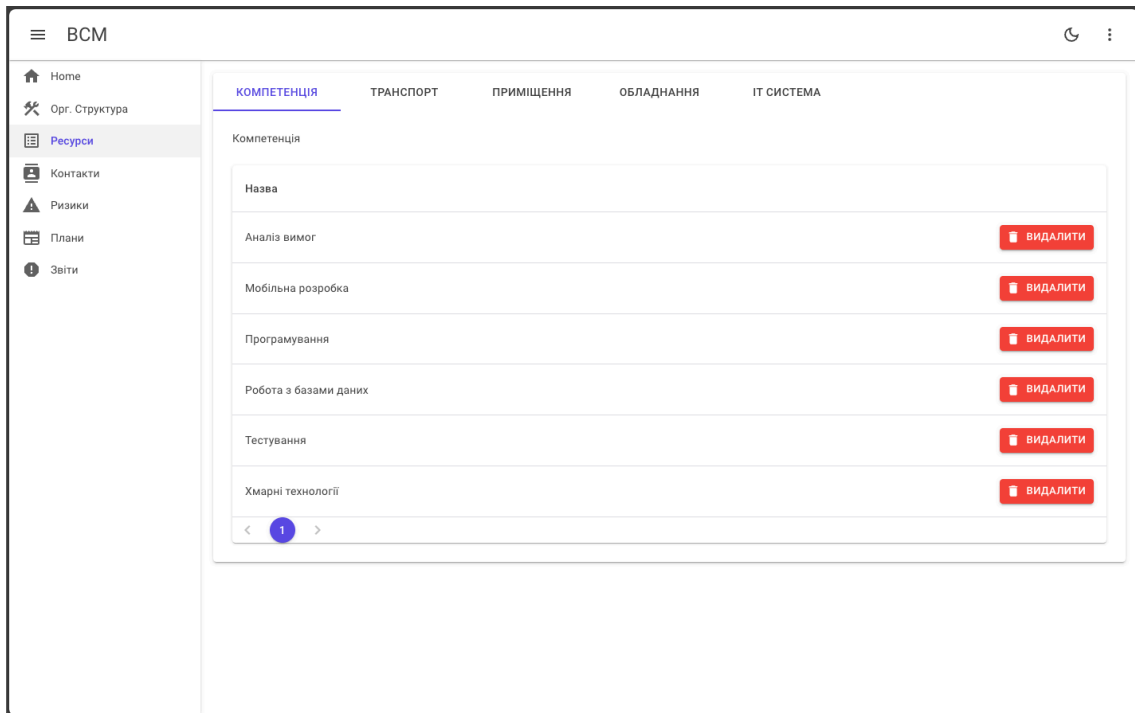


Рисунок 4.2 – Перелік ресурсів

Обрані компоненти системи забезпечення безперервності бізнесу були реалізовані з урахуванням принципів доступності, що включають підтримку управління як за допомогою миші, так і клавіатури. Це є важливим аспектом для забезпечення відповідності стандартам доступності, які рекомендують підтримку альтернативних способів навігації та взаємодії з системою.

Вибравши елемент зі сторінки організаційної структури, користувач розпочинає аналіз впливу на бізнес. Інтерфейс побудований у вигляді покрокової форми, яка починається із загального опису діяльності (рисунок 4.3).

На наступному етапі експерт визначає, як зупинка аналізованої діяльності вплине на бізнес у кожній із категорій залежно від тривалості перерви. Особливістю заповнення такої матриці є той факт, що рівень впливу не може зменшуватися в разі збільшення тривалості часу відновлення функціонування процесу. Таким чином, якщо експерт обрав «Високий» рівень впливу на значенні 8 годин, це означає, що рівні 5 днів, та 10 днів також мають високий рівень впливу у даній категорії. На основі цієї інформації можна визначити цільовий час відновлення, обираючи найменше значення часу, що спричиняє

високий або критичний вплив на бізнес (рисунок 4.4).

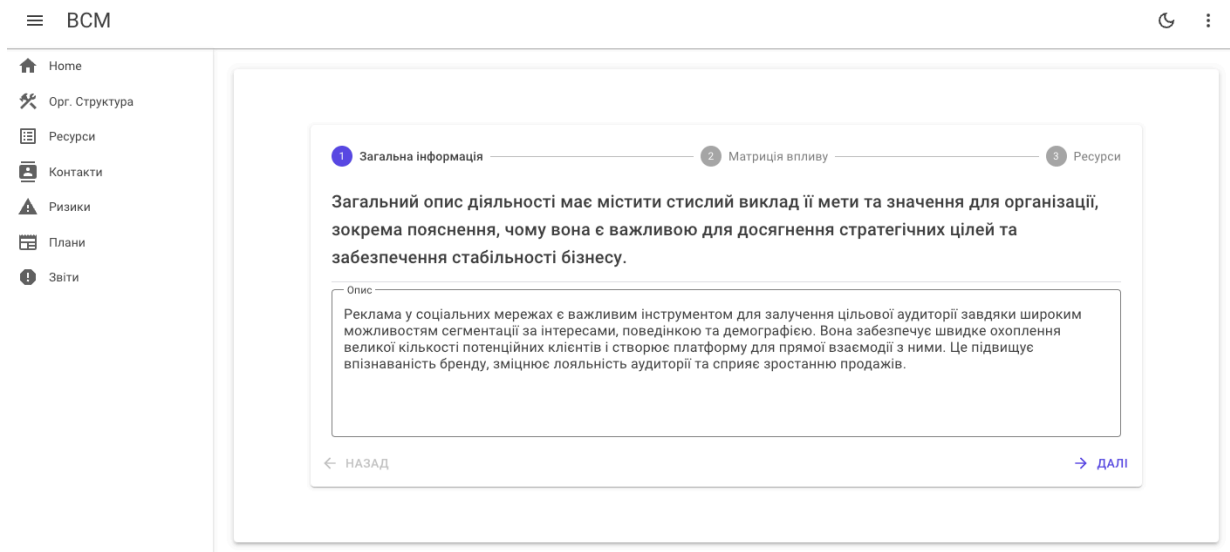


Рисунок 4.3 – Аналіз впливу на бізнес, загальна інформація

Перелік категорій впливу може змінюватися в залежності від сфери діяльності організації. Для прикладу використані такі категорії: постачання послуг клієнтам, здоров'я та безпека, репутаційний вплив, регуляторний і правовий.

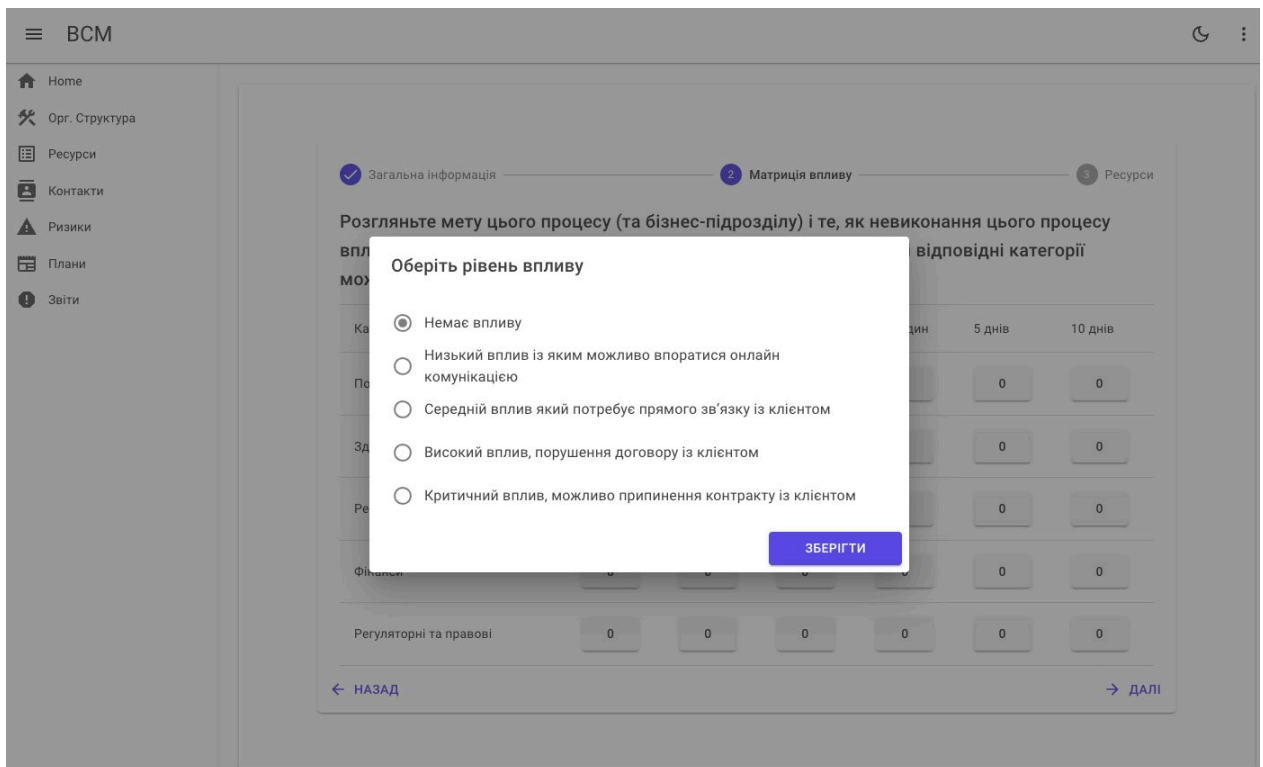


Рисунок 4.4 – Визначення рівня впливу на бізнес

На завершальному етапі формується перелік необхідних компетенцій та ресурсів, які забезпечують нормальне функціонування діяльності (рисунки 3.5). Інтерфейс дозволяє обирати серед вже існуючих ресурсів, так і додавати нові, які в свою чергу також можуть бути використані у аналізі інших процесів в компанії. Такий підхід дозволяє уникнути створення дублюючих записів у реєстрі ресурсів, постачальників або ІТ-систем, що використовуються різними підрозділами.

Якщо відоме значення часу відновлення (RTO) постачальника і воно менше за необхідне, система сповіщає про наявність вразливості. Якщо існує тимчасове рішення, експерт повинен його описати, вказавши також час дії цього рішення. Об'єднуючи RTO постачальника та час дії тимчасового рішення, можна усунути таку вразливість.

Визначте залежності та необхідні ресурси для виконання аналізованої активності, включаючи постачальників, партнерів з аутсорсингу та інших зацікавлених сторін.

Ресурс

Назва	Тип	Час відновлення	Деталі	Дата
Microsoft office	ІТ система	< 4 годин	Використовується PowerPoint для створення презентацій	11/18/2024 11:36 AM
Хмарні технології	Компетенція		Знання Microsoft Azure необхідні для забезпечення роботи продукту для користувачів.	11/18/2024 11:37 AM

Rows per page: 10 1-2 of 2

Рисунок 4.5 – Перелік необхідних ресурсів

Виконавши аналіз користувач створює план, обираючи організаційну структуру (рисунки 4.6).

BSM

Плани

Створити план

- Дизайн
- Домен
- >  Дослідження і розробка
- ▼  Маркетинг
  - Google реклама
  - Сайт
  - Соціальні мережі
  - Стратегія і планування
- ▼  Фінанси
  - Вхідні платежі
  - Рахунки до сплати

Назва\*  
Реклама та маркетинг

Дата наступного перегляду  
11/21/2024

СТВОРИТИ

Рисунок 4.6 – Форма створення плану

### 4.3 Візуалізація та аналіз даних про час відновлення

Граф на рисунку 4.7 ілюструє бізнес-процеси, представлені у вигляді вершин, та зв'язки між ними, які відображені у вигляді ребер. Кожна вершина представляє окремий бізнес-процес, для якого вказано назву та час відновлення. Наприклад, процес "Маркетинг" має час відновлення 4 години, а "Вхідні платежі" – 2 дні.

Ребра графу показують залежності між процесами. Зелені ребра означають, що час відновлення правого процесу не перевищує часу відновлення лівого, і залежність є безпечною. Червоні ребра сигналізують про проблему, коли час відновлення лівого процесу є меншим за правий, що унеможливорює вчасне відновлення роботи залежного процесу.

Вершини, що мають проблеми із доступністю ресурсів, виділені червоною рамкою. Це означає, що один із ресурсів, які використовує процес, має довший час відновлення, ніж необхідно для виконання цього процесу. Напри-

клад, процес "Вхідні платежі" залежить від сторонньої системи прийому платежів із часом відновлення 5 днів, тоді як для самого процесу потрібен час лише 2 дні.

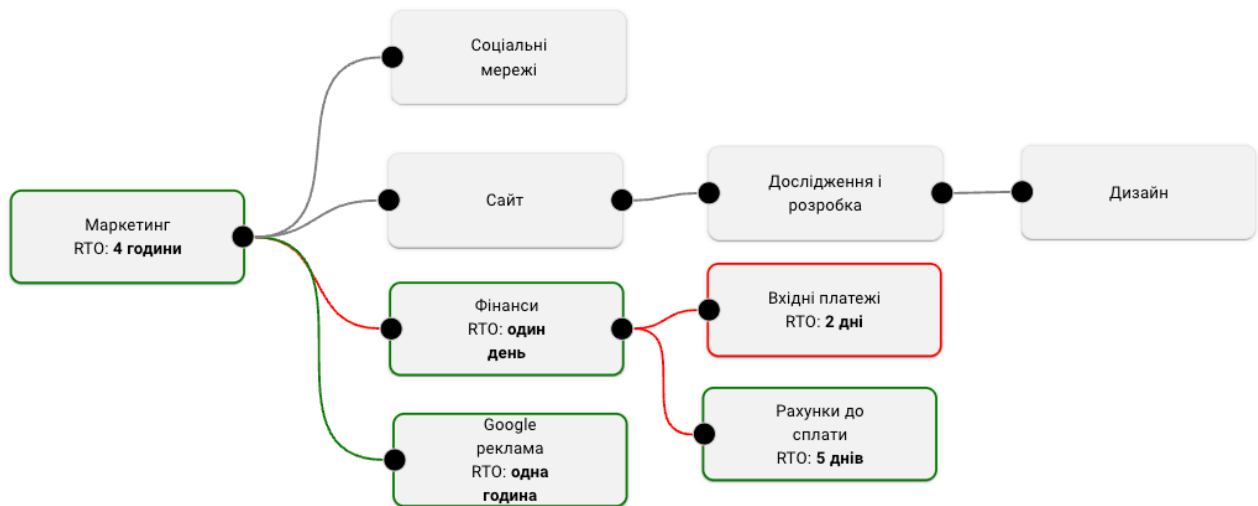


Рисунок 4.7 – Візуалізація зібраних даних

Такий підхід до візуалізації дозволяє чітко побачити залежності між бізнес-процесами та ідентифікувати проблемні точки, які можуть вплинути на загальний час відновлення роботи. Особливу увагу варто звертати на процеси з червоними рамками та зв'язки з червоними ребрами, оскільки вони вказують на критичні моменти у забезпеченні безперервності бізнесу.

#### Висновки за розділом 4

У цьому розділі було розглянуто стандартне тестове середовище у Microsoft Azure та особливості роботи інтерфейсу системи забезпечення безперервності бізнесу. Аналіз показав, що проекти на основі .NET Aspire демонструють високий рівень гнучкості завдяки незалежності від конкретних хмарних платформ. Це дозволяє адаптувати систему для розгортання у різних середовищах, включаючи локальні хости чи інші хмарні платформи.

Хоча тестове середовище Microsoft Azure дозволяє швидко перевірити працездатність системи, воно має певні обмеження, зокрема, відсутність вбудо-

ваних механізмів резервного копіювання бази даних і захисту додатків через Application Gateway та Web Application Firewall.

Інтерфейс системи побудований відповідно до принципів доступності, що робить його зручним для користувачів із різним рівнем технічної підготовки. Покроковий процес аналізу впливу на бізнес дозволяє фахівцям точно оцінювати наслідки перерв у діяльності та формувати плани відновлення з урахуванням пріоритетів. Гнучкість інтерфейсу також забезпечується можливістю розширення переліку ресурсів та налаштування матриці впливу відповідно до потреб конкретної організації.

Таким чином, представлена система поєднує ефективність розгортання у тестовому середовищі, інтеграцію сучасних хмарних сервісів та зручність інтерфейсу для аналізу безперервності бізнесу, що робить її потужним інструментом для організацій.

## ВИСНОВКИ

У даній кваліфікаційній роботі було вирішено низку актуальних питань, пов'язаних із реалізацією та впровадженням системи забезпечення безперервності бізнесу. Виконане дослідження підтвердило важливість застосування системного аналізу для виявлення ключових проблем, ризиків та потреб підприємств в умовах кризових ситуацій, зокрема в сучасних українських реаліях.

Робота охоплює аналіз існуючих підходів до впровадження систем управління безперервністю бізнесу, зокрема стандарти ISO 22301, що забезпечують ефективність і міжнародну відповідність систем. Використані методи дозволили розробити інформаційну систему, яка відповідає потребам різних типів організацій, зокрема завдяки модульній архітектурі та високому рівню адаптивності.

Програмна реалізація системи продемонструвала можливості платформи .NET для створення корпоративних веб-додатків. Інструменти, такі як Blazor і EntityFrameworkCore, забезпечили гнучкість розробки, інтуїтивно зрозумілий інтерфейс для користувачів, а також високу продуктивність і безпеку. Врахування вимог доступності та захищеності інформації дозволило створити систему, придатну для роботи як у приватному, так і у державному секторах.

Розроблена система охоплює ключові аспекти розробки програмного забезпечення на прикладі системи безперервності бізнесу: збір та аналіз даних, планування дій у кризових ситуаціях, навчання персоналу та регулярне вдосконалення процесів. Реалізовано підтримку відповідності міжнародним стандартам, що є ключовим для впровадження системи.

Отримані результати можуть бути використані як основа для подальших розробок, а також як практичний інструмент для впровадження у реальну діяльність підприємств. Це дозволяє підвищити їхню стійкість, конкурентоспроможність і здатність ефективно реагувати на кризові виклики.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ISO 22301:2019 URL: <https://www.iso.org/standard/75106.html> (дата звернення 18.09.2024).
2. Кожин А. Ю. Analysis and implementation of a business continuity management system. *III Міжнародна молодіжна науково-практична конференція НАВЧАННЯ І ВИКЛАДАННЯ: у світі після війни*, м. Харків, 2024. Р. 223.
3. Сорока К. О. Основи теорії систем і системного аналізу. Харків : ХНАМГ, 2004. 115 с.
4. Clements P., Bachmann F., Len B., Garlan D. Documenting Software Architectures. Boston : Addison-Wesley, 2022. 537 p.
5. Катренко А. В. Системний аналіз. Львів : Новий світ 2000, 2011, 396 с.
6. Bass L., Clements P., Kazman R. Software Architecture in Practice. Boston : Addison-Wesley Professional, 2021. 464 p.
7. Fowler M. Patterns of Enterprise Application Architecture. Boston : Addison-Wesley Professional, 2002. 560 p.
8. Azure SLA Board URL: <https://azurecharts.com/sla> (дата звернення 11.11.2024).
9. How reservation discounts apply to Azure App Service: URL: <https://learn.microsoft.com/uk-ua/azure/cost-management-billing/reservations/reservation-discount-app-service> (дата звернення: 17.11.2024)
10. Vernon V. Implementing Domain-Driven Design, Addison-Wesley Educational Publishers Inc. 2013. 656 p.
11. Deming W. E. Out of the Crisis, 26. print. Cambridge, Mass : Massachusetts Institute of Technology, 1998. 507 p.
12. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Boston : Addison-Wesley Educational Publishers Inc, 2003. 560 p.
13. Introduction to .NET. URL: <https://learn.microsoft.com/uk-ua/dotnet/core/introduction> (дата звернення 14.11.2024).
14. The Twelve-Factor App. URL: <https://12factor.net> (дата звернення:

15.11.2024).

15. Apply Entity Framework Core migrations in .NET. URL: <https://learn.microsoft.com/en-us/dotnet/aspire/database/ef-core-migrations> (дата звернення: 17.11.2024).

16. NET Aspire deployments. URL: <https://learn.microsoft.com/en-us/dotnet/aspire/deployment/overview> (дата звернення: 18.11.2024).