

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК

«ПОГОДЖЕНО»

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Декан факультету  
інформаційних технологій

Завідувач кафедри комп'ютерних наук

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

\_\_\_\_\_ 2024 р.

\_\_\_\_\_ 2024 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему \_\_ Інтелектуальна система вирішення оптимізаційних задач на основі  
методу Монте-Карло з детермінованим навчанням \_\_\_\_\_

Спеціальність \_\_ 121 «Інженерія програмного забезпечення» \_\_\_\_\_

(код і назва)

Освітня програма \_\_\_\_\_ «Програмне забезпечення інформаційних систем» \_\_\_\_\_

(назва)

Орієнтація освітньої програми \_\_\_\_\_

(освітньо-професійна або освітньо-наукова)

Керівник магістерської кваліфікаційної роботи

к.т.н., професор

(науковий ступінь та вчене звання)

(підпис)

Пархоменко І.І.

(ПІБ)

Виконав

(підпис)

Шепетило В.В.

(ПІБ студента)

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (НП) КОМП'ЮТЕРНИХ НАУК І ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри**  
**комп'ютерних наук**

К.Т.Н., доцент Голуб Б.Л.  
(науковий ступінь, вчене звання) (підпис) (ІПБ)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2024 року

**З А В Д А Н Н Я**

**ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ**

Шепетило Володимир Васильович

(прізвище, ім'я, по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

(код і назва)

Освітня програма «Програмне забезпечення інформаційних систем»

(назва)

Орієнтація освітньої програми \_\_\_\_\_

(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи «Інтелектуальна система вирішення оптимізаційних задач на основі методу Монте-Карло з детермінованим навчанням» \_\_\_\_\_

затверджена наказом ректора НУБіП України від “\_15\_”\_08\_2024р. №\_7\_

Термін подання завершеної роботи на кафедру \_\_\_\_\_

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи \_\_\_\_\_

Перелік питань, що підлягають дослідженню:

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

Перелік графічного матеріалу (за потреби) \_\_\_\_\_

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 2024 р.

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_

(підпис)

Пархоменко І.І.

(прізвище та ініціали)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

Шепетило В.В.

(прізвище та ініціали студента)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз проблемної області	03.09.2024 05.09.2024	Опис та аналіз актуальності та мети теми, об'єкту, предмету та методів дослідження, а також формування поставлених задач.
2		06.09.2024 21.09.2024	Збір даних по існуючим системам та огляд літературних джерел. Опис предметної області. Аналіз наукових джерел. Опис постановки задачі дослідження.
3	Моделювання системи	21.09.2024 04.10.2024	Розробка діаграм прецедентів використання, пакетів, класів та розгортання, які дозволяють отримати передумови для аналізу поведінки системи.
4	Інформаційне забезпечення системи	05.10.2024 10.10.2024	Побудова логічних моделей для відділу маркетингу та керівництва підприємства.
5		11.10.2024 14.10.2024	Розробка БД відділу маркетингу та керівництва підприємства.
6		15.10.2024 18.10.2024	Дослідження OLAP- технологій та методів Data Mining. Проектування СД.
7		19.10.2024 27.10.2024	Розробка куба інтерактивної аналітичної обробки даних (OLAP) і моделі інтелектуального аналізу даних.
8		28.10.2024 02.11.2024	Вибір та встановлення програмних засобів для розробки ПЗ. Розробка алгоритмів роботи системи.
9	Програмне забезпечення системи	03.11.2024 05.11.2024	Розробка ПЗ, що забезпечує моніторинг процесів
10		06.11.2024 08.11.2024	ПЗ, що забезпечує роботу із СД, а саме передачу інформації із реляційної БД в СД.
11		09.11.2024 13.11.2024	Оцінка ефективності роботи ІС
13	Написання пояснювальної записки	14.11.2024 19.11.2024	Оформлення пояснювальної записки магістерської роботи, а також інших супровідних документів (презентації, доповіді)

Студент \_\_\_\_\_ Шепетило В.В.  
(підпис)

Керівник роботи \_\_\_\_\_ Пархоменко І.І.  
(підпис)

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	10
ІС - інтелектуальна система .....	10
ВСТУП .....	11
<b>1. АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ .....</b>	<b>14</b>
<b>1.1. Актуальність оптимізаційних задач у сучасних бізнес-процесах.....</b>	<b>14</b>
<b>1.2. Постановка задач дослідження .....</b>	<b>23</b>
<b>1.2.1. Роль інтелектуальних систем у підвищенні ефективності управління .....</b>	<b>23</b>
<b>1.2.2. Головні аспекти методу Монте-Карло та можливості детермінованого навчання.....</b>	<b>28</b>
<b>1.3. Дослідження існуючих інтелектуальних систем вирішення оптимізаційних задач.....</b>	<b>34</b>
<b>2. МОДЕЛЮВАННЯ СИСТЕМИ.....</b>	<b>40</b>
2.1. Теоретико-методологічні аспекти моделювання системи.....	40
2.2. Діаграма прецедентів використання .....	46
2.3. Діаграма пакетів.....	53
Діаграма пакетів є важливим інструментом в об'єктно-орієнтованому моделюванні, що використовується для візуалізації структури програмної системи на рівні її компонентів (пакетів). Діаграма дозволяє організувати класи та інші елементи системи в логічні одиниці, що сприяє кращому управлінню та підтримці програмного коду. Для нашої інтелектуальної системи вирішення оптимізаційних задач на основі методу Монте-Карло з детермінованим навчанням, діаграма пакетів служить для визначення основних компонентів і їх взаємодій у межах системи.....	53
Діаграма пакетів для нашої системи має на меті поділ програмної системи на функціональні блоки, які відповідають за різні етапи обробки даних, моделювання, навчання та оптимізацію. Кожен з цих блоків є логічним пакетом, що містить певні класи або модулі. Нижче описані основні пакети, які можуть бути присутніми в діаграмі пакетів для нашої інтелектуальної системи.....	54
Пакет «Вхідні дані та інтеграція». Цей пакет відповідає за отримання та обробку вхідних даних для системи. Він включає модулі для завантаження даних з різних джерел (бази даних, файли), а також для попередньої обробки та валідації цих даних перед подальшим використанням у процесах моделювання.....	54
Класи: DataLoader, DataValidator, InputParser.....	54
Пакет «Моделювання та оптимізація». Включає основні класи, що реалізують метод Монте-Карло для розв'язання оптимізаційних задач. Цей пакет відповідає за генерування випадкових вибірок, виконання ітераційних обчислень та адаптацію результатів на основі зібраних даних. ....	54
Класи: MonteCarloModel, OptimizationAlgorithm, SimulationEngine. ....	54
Пакет «Детерміноване навчання». Цей пакет забезпечує реалізацію детермінованого навчання для вдосконалення оптимізації. Він містить класи, які виконують обчислення на основі попередніх результатів і забезпечують точніші прогнози та оптимальні стратегії. ....	54
Класи: DeterministicLearning, LearningAlgorithm, LearningModel.....	54
Пакет «Аналіз результатів». Включає класи для збору та обробки результатів після виконання оптимізаційних задач. Цей пакет відповідає за аналіз і візуалізацію результатів, що отримані після застосування моделі Монте-Карло та алгоритмів навчання. ....	54
Класи: ResultsAnalyzer, DataVisualizer, PerformanceMetrics.....	54

Пакет «Інтерфейс користувача». Відповідає за взаємодію користувача з системою. Цей пакет містить класи, що відповідають за графічний інтерфейс або інтерфейс командного рядка для введення даних, запуску обчислень та відображення результатів.....	54
Класи: <code>UserInterface</code> , <code>ControlPanel</code> , <code>InputForm</code> .....	54
Пакет «Управління процесами». Цей пакет відповідає за управління процесами в системі, такими як обробка помилок, моніторинг виконання, оптимізація часу виконання задач та інтеграція з іншими системами.....	55
Класи: <code>ProcessManager</code> , <code>ErrorHandler</code> , <code>TaskScheduler</code> .....	55
Пакет «Збереження даних». Включає класи для збереження та експорту результатів моделювання, зокрема в бази даних чи інші формати для подальшого аналізу або використання в інших системах..	55
Класи: <code>DatabaseManager</code> , <code>DataExporter</code> , <code>FileSaver</code> .....	55
Взаємодія між пакетами. На діаграмі пакетів ці компоненти можуть бути з'єднані лініями, які вказують на залежності між ними. Наприклад:.....	55
Пакет «Вхідні дані та інтеграція» взаємодіє з пакетами «Моделювання та оптимізація» і «Аналіз результатів», оскільки дані, що надходять, безпосередньо впливають на моделювання та подальший аналіз. ....	55
Пакет «Моделювання та оптимізація» тісно інтегрований з пакетом «Детерміноване навчання», оскільки результат моделювання є вхідними даними для алгоритмів навчання. ....	55
Пакет «Аналіз результатів» залежить від результатів, що генеруються в пакетах «Моделювання та оптимізація» і «Детерміноване навчання» для візуалізації та подальшого аналізу.....	55
Таким чином, діаграма пакетів для нашої інтелектуальної системи вирішення оптимізаційних задач відображає основні функціональні блоки системи та їх взаємозв'язки. ....	56
Вона дозволяє чітко структурувати код і забезпечує розподіл обов'язків між компонентами, що допомагає підтримувати систему на етапах її розробки та експлуатації.....	56
Використання діаграми пакетів є важливим кроком в організації логіки та забезпеченні ефективної взаємодії між різними елементами системи, що розробляється на основі ТЗ.....	56
<b>2.4. Діаграма класів</b> .....	57
<b>2.5. Діаграма розгортання</b> .....	59
Діаграма розгортання для інтелектуальної системи, яка підтримує прийняття управлінських рішень у сфері маркетингу, відображає основні обчислювальні вузли, програмні модулі та взаємодію компонентів між собою. Основне завдання діаграми UML - продемонструвати фізичну структуру системи, а також взаємозв'язок між програмними компонентами в процесі роботи. ....	59
Головними компоненти діаграми розгортання будуть (рис. 2.5): .....	60
<code>Server</code> . Основний сервер виступає головним обчислювальним вузлом, на якому розміщені: .....	60
- база даних (БД), де зберігаються дані про попередні маркетингові кампанії, показники ринку, цільову аудиторію та аналітичні звіти; .....	60
- модуль аналізу та прогнозування для обробки даних, аналізу ринкових тенденцій, оцінки конкурентних показників і прогнозування результатів маркетингових стратегій. Це ключовий компонент, який забезпечує роботу алгоритмів оптимізації та прогнозування на основі методів, таких як Монте-Карло. ....	60

- модуль звітності, який генерує індивідуальні звіти для маркетологів та керівників підприємства, дозволяючи оцінювати ефективність ухвалених рішень.....	60
Клієнтські пристрої маркетолога та менеджера. Клієнтські пристрої (комп'ютери, планшети чи смартфони), які використовують маркетологи й керівники для доступу до системи. Вони можуть переглядати прогнози, аналітичні звіти, проводити оцінку результатів маркетингових рішень та вносити свої дані або завдання. ....	60
Веб-інтерфейс для доступу до аналітичних інструментів системи, а також для перегляду візуалізованих даних, для спрощення користувачам доступ до важливої інформації в режимі реального часу. ....	61
Захищений зв'язок між клієнтськими пристроями та сервером. Сервер та клієнтські пристрої підключені через захищений канал зв'язку, забезпечуючи безпеку передачі даних, конфіденційність і захист від несанкціонованого доступу.....	61
<b>3. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ .....</b>	<b>62</b>
<b>3.1. Основні положення моделі інформаційного забезпечення.....</b>	<b>62</b>
Для розробки моделі інформаційного забезпечення інтелектуальної системи підтримки прийняття управлінських рішень для маркетологів і керівників підприємства пропонуємо розглянути її структуру, яка складається з кількох основних компонентів: бази даних, сховища даних для аналітики, інтерфейсу для взаємодії користувачів та зв'язків між цими компонентами (табл. 3.1).....	64
<b>3.2. Побудова логічної моделі.....</b>	<b>66</b>
Для інтелектуальної системи підтримки прийняття управлінських рішень було розроблено логічну модель, яка складається з основних сутностей та їхніх атрибутів, що підтримують процеси збору та обробки даних для аналітики і прийняття рішень. Ключовими елементами системи є клієнтська база, дані про продукцію та ринок, а також звітні та аналітичні показники. ....	66
У відповідності до попередніх розділів та відповідно до нашої теми наведемо основні сутності логічної моделі: .....	66
Сутність «Клієнт» - містить інформацію про клієнтів системи.....	66
ID_клієнта (int) - унікальний ідентифікатор клієнта. ....	66
Ім'я_клієнта (varchar, max 50) - ім'я клієнта. ....	66
Контакт_інформація (varchar, max 100) - контактні дані клієнта. ....	67
Історія_взаємодії (text) - дані про попередні взаємодії з клієнтом.....	67
Сутність «Продукт» - містить дані про продукцію, яку використовує компанія. ....	67
ID_продукту (int) - унікальний ідентифікатор продукту.....	67
Назва_продукту (varchar, max 50) - назва продукту. ....	67
Категорія (varchar, max 30) - категорія, до якої належить продукт. ....	67
Вартість (numeric) - вартість продукту.....	67
Сутність «Замовлення» - забезпечує інформацію про замовлення, які були зроблені.....	67
ID_замовлення (int) - ідентифікатор замовлення. ....	67
Дата_замовлення (datetime) - дата оформлення замовлення.....	67
Сума_замовлення (numeric) - загальна вартість замовлення. ....	67
Кількість_товарів (int) - кількість товарів у замовленні.....	67

Сутність «Маркетингові_показники» - зберігає дані про результати маркетингових кампаній та аналітику. ....	67
ID_показника (int) - унікальний ідентифікатор маркетингового показника. ....	67
Кампанія (varchar, max 50) - назва маркетингової кампанії. ....	67
Результат_кампанії (text) - опис результату кампанії. ....	67
ROI (numeric) - рентабельність інвестицій у кампанію. ....	67
Сутність «Ринковий_тренд» - містить інформацію про ринкові тенденції та динаміку. ....	67
ID_тренду (int) - ідентифікатор ринкового тренду. ....	67
Назва_тренду (varchar, max 50) - назва тренду. ....	67
Опис (text) - опис сутності тренду та його характеристик. ....	67
Дата_оновлення (datetime) - дата останнього оновлення інформації про тренд. ....	67
Сутність «Звіт» - зберігає дані про звіти, сформовані на основі аналітики. ....	67
ID_звіту (int) - унікальний ідентифікатор звіту. ....	67
Тип_звіту (varchar, max 50) - тип звіту (наприклад, щоквартальний, річний). ....	67
Зміст (text) - опис даних, які містяться у звіті. ....	68
Дата_створення (datetime) - дата створення звіту. ....	68
Сутність «Прогноз» - надає інформацію про прогнози, отримані з аналітики. ....	68
ID_прогнозу (int) - унікальний ідентифікатор прогнозу. ....	68
Період (varchar, max 20) - період, на який робиться прогноз. ....	68
Прогнозоване_значення (numeric) - прогнозовані значення продажів або попиту. ....	68
Точність_прогнозу (numeric) - показник точності прогнозу у відсотках. ....	68
Сутність «Аналітик» - містить інформацію про аналітиків, які працюють з системою. ....	68
ID_аналітика (int) - ідентифікатор аналітика. ....	68
Ім'я_аналітика (varchar, max 50) - ім'я аналітика. ....	68
Спеціалізація (varchar, max 50) - напрямок роботи аналітика. ....	68
Сутність «Адміністратор» - містить дані про адміністратора системи, який налаштовує та забезпечує її роботу. ....	68
ID_адміністратора (int) - ідентифікатор адміністратора. ....	68
Ім'я_адміністратора (varchar, max 50) - ім'я адміністратора. ....	68
Рівень_доступу (varchar, max 30) - рівень доступу до системи. ....	68
Сутність «Користувач» - містить інформацію про користувачів, які мають доступ до системи. ....	68
ID_користувача (int) - ідентифікатор користувача. ....	68
Ім'я_користувача (varchar, max 50) - ім'я користувача. ....	68
Роль (varchar, max 30) - роль користувача в системі (наприклад, маркетолог, керівник). ....	68
Визначимо основні зв'язки між сутностями:.....	68

Замовлення-Клієнт. Клієнт може зробити одне або більше замовлень, але кожне замовлення пов'язане тільки з одним клієнтом..... 68

Продукт-Замовлення. Замовлення може містити один, або більше продуктів. .... 68

Аналітик-Маркетингові\_показники. Аналітики мають доступ до маркетингових показників для їхньої обробки та аналізу..... 68

Звіт-Прогноз: Звіт може включати результати одного, або кількох прогнозів, що були зроблені аналітиками..... 69

Ринковий\_тренд-Маркетингові\_показники. Маркетингові показники можуть бути прив'язані до певного ринкового тренду. .... 69

Адміністратор-Користувач. Адміністратор має право керувати правами доступу користувачів до системи. .... 69

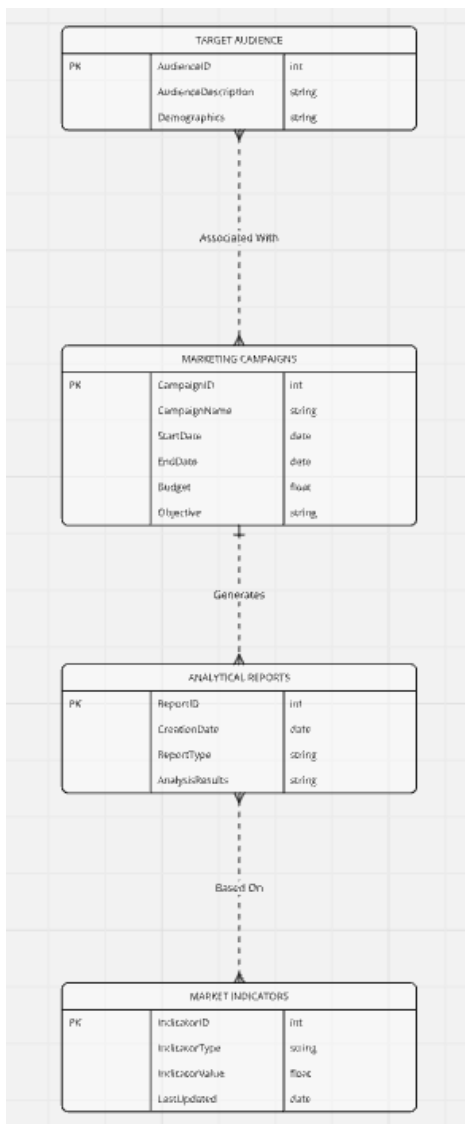


Рис. 3.1. ER-діаграма, яка відображає структуру бази даних ІСППР..... 69

Зв'язки між наведеними на рис. 3. Сутностями: ..... 69

Маркетингові кампанії → Генерують → Аналітичні звіти (I):..... 69

Кожна маркетингова кампанія може створювати кілька звітів, але кожен звіт належить до однієї кампанії. .... 69

Цільова аудиторія → Використовується у → Маркетингові кампанії (M): .....	70
Одна цільова аудиторія може брати участь у кількох кампаніях, і кожна кампанія може мати кілька аудиторій.....	70
Аналітичні звіти → Базуються на → Ринкові показники (N): .....	70
Один звіт може використовувати кілька ринкових показників, а один показник може бути застосований у кількох звітах. ....	70
Таким чином, розроблена логічна модель дозволяє створити структуру інформаційної бази для інтелектуальної системи підтримки управлінських рішень. Система зберігає ключові показники, звіти, прогнози і взаємодію між аналітиками, маркетологами та керівниками підприємства. Така модель допомагає ефективно зберігати та обробляти необхідні для прийняття рішень дані, забезпечуючи всім учасникам системи доступ до потрібної інформації в зручному форматі.....	70
<b>3.3. Основи фізичної моделі даних для оптимізаційної системи .....</b>	<b>70</b>
<b>3.3. Реалізація фізичної моделі.....</b>	<b>73</b>
Зазначимо основні особливості фізичної моделі бази даних:.....	73
<b>4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ .....</b>	<b>79</b>
<b>4.1. Вибір, обґрунтування інструментарію та спеціальні програмні засоби для розробки програмного забезпечення.....</b>	<b>79</b>
<b>ВИСНОВКИ .....</b>	<b>103</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API - інтерфейс програмування застосунків

DM - Data Mining

ERD - Essence Relation Diagramming

ID - ідентифікатор для бази даних

IDE - комплексне програмне рішення для розробки програмного забезпечення

Java - мова програмування

JS - JavaScript

MySQL - система управління SQL

OLAP - Online analytical processing

OLAP - технологія бази даних, оптимізована для запитів і звітування, а не для обробки транзакцій

SQL - Structured Query Language

UML - Unified Modeling Language

БД - база даних

ІС - інтелектуальна система

ІСППР - інтелектуальна система підтримки прийняття рішень

ІТ - інформаційні технології

СППР - системи підтримки прийняття рішень

СУБД - система управління базами даних

СУБД - система управління базою даних

ІІ - штучний інтелект

## ВСТУП

**Актуальність теми.** В наш час цифрової трансформації, турбулентності та високої конкуренції ті підприємства, що хочуть займати лідируючі позиції на ринку стикаються з потребою оптимізації своїх маркетингових та бізнес-процесів для підвищення ефективності та отримання прибутку. Через збільшення обсягів інформації, необхідної для прийняття управлінських рішень, та складність умов невизначеності зумовлюють попит на інтелектуальні системи, що здатні автоматизувати та покращити процеси планування, управління і контролю за діяльністю сучасного підприємства.

У відповідності до наукових математико-економічних джерел метод Монте-Карло, широко використовується для прийняття рішень в умовах ризику та невизначеності, демонструє ефективність в аналізі й прогнозуванні результатів на основі випадкових сценаріїв, що генеруються для імітування реальних подій. Але слід зазначити, що традиційні підходи до його застосування можуть бути ресурсозатратними, а це в свою чергу знижує ефективність при обробці масштабних даних (Big Data). Використання елементів детермінованого навчання дозволяє покращити продуктивність методу, адаптуючи його до сучасних умов та скорочуючи час розрахунків, що особливо цінно для прийняття швидких управлінських та маркетингових рішень у бізнес-середовищі в якому працює підприємство.

Саме тому розробка інтелектуальної системи, що використовує метод Монте-Карло з детермінованим навчанням, є важливим інструментом для підвищення ефективності управління бізнес-процесами підприємства. І сама система забезпечує автоматизацію та оптимізацію різних аспектів діяльності підприємства, що є критичним у сучасному високонкурентному середовищі.

**Мета роботи.** Метою даного дослідження є розробка та впровадження інтелектуальної системи, яка поєднує метод Монте-Карло з детермінованим навчанням для підвищення швидкості та точності розв'язання оптимізаційних

задач, що сприятиме ефективнішому управлінню ресурсами, адаптації до ринкових змін та підвищенню рентабельності бізнесу.

**Зміст поставлених завдань.** Для досягнення поставленої мети необхідно:

- провести системний аналіз процесів оптимізації на підприємстві;
- визначити вимоги до системи підтримки прийняття рішень;
- розробити структуру інформаційного забезпечення системи;
- створити алгоритми аналізу даних, провести тестування і оцінку ефективності системи.

**Об'єкт і предмет дослідження.** Об'єктом дослідження є процес оптимізації, що стосуються прийняття управлінських рішень у бізнес-середовищі, де наявні значні невизначеності та ризики. Предметом дослідження є інтелектуальна система, яка застосовує метод Монте-Карло з детермінованим навчанням для вирішення оптимізаційних задач.

**Методи дослідження.** Теоретичною та методологічною основою дослідження є праці вітчизняних та зарубіжних вчених з проблем розробки інтелектуальних систем прийняття рішення. Дослідження ґрунтується на системному методі пізнання процесів і явищ та використанні статистичного, графічного методів, методу порівняння та узагальнення, гнучких методів розробки програмних продуктів. У дослідженні використовуються методи моделювання даних та обробки інформації, в тому числі з використанням інструментів Java, Android Studio та спеціалізованих бібліотек для реалізації методу Монте-Карло.

**Наукова новизна.** У процесі дослідження була розроблена інтелектуальна система підтримки прийняття управлінських рішень, що дозволяє підвищити ефективність маркетингової та бізнес-діяльності підприємства ТОВ «Яструб-Тур».

**Апробація результатів дослідження.** В процесі розробки магістерської роботи на тему «Інтелектуальна система вирішення оптимізаційних задач на основі методу Монте-Карло з детермінованим навчанням» були проведені виступи на XV міжнародній науково-практичній конференції молодих вчених

«Інформаційні технології: економіка, техніка, освіта» (м. Київ, 2024р.). А також ІС впроваджена в діяльність ТОВ «Яструб-Тур».

**Публікації.** За темою магістерської були опубліковані тези «Розробка інтелектуальної системи вирішення оптимізаційних задач на основі методу монте-карло з детермінованим навчанням», що ввійшли до «Збірник матеріалів Міжнародної науково-практичної конференції молодих вчених «Інформаційні технології: економіка, техніка, освіта» (07-08 листопада 2024 року, Київ, Україна).

**Практичне значення.** Розроблена система впроваджена в ТОВ «Яструб-Тур» та може бути використана підприємствами для підвищення ефективності управління, оптимізації ресурсів та прийняття рішень в умовах ризику, що є важливим для підвищення їх конкурентоспроможності сучасного підприємства.

**Структура роботи.** Структурні елементи роботи розміщені в такій послідовності.

**Перший розділ** присвячений огляду проблемної області, постановці задачі дослідження та аналізу актуальних підходів до оптимізації процесів у бізнесі.

**Другий розділ** охоплює моделювання системи, створення діаграм прецедентів, класів, компонентів та розгортання.

**Третій розділ** зосереджений на інформаційному забезпеченні, включаючи логічне та фізичне моделювання даних і створення сховища даних.

**Четвертий розділ** описує розробку програмного забезпечення, алгоритмів системи, а також інтерфейс користувача для моніторингу та аналізу результатів.

## 1. АНАЛІЗ ПРОБЛЕМНОЇ ОБЛАСТІ

### 1.1. Актуальність оптимізаційних задач у сучасних бізнес-процесах

У сучасному динамічному світі, з турбулентною економікою, здатність швидко адаптуватися до змінюваних умов та приймати обґрунтовані управлінські рішення в умовах невизначеності є критичною для успіху в різних сферах бізнес-діяльності. Саме це є обґрунтування в контексті розробки інтелектуальних систем для оптимізації бізнес, маркетингових та логістичних процесів, де ефективність та швидкість обробки інформації безпосередньо впливають на реальну конкурентоспроможність сучасного підприємства підприємства.

Посилення діджиталізації усіх сфер суспільного життя та бурхливий розвиток ІІІ, Big Data та машинного навчання, призвели до того, що розробка інтелектуальних систем, здатних вирішувати оптимізаційні задачі з використанням методу Монте-Карло у поєднанні з детермінованим навчанням, набуває особливої актуальності у зв'язку зі зростанням обсягів даних і складністю їх аналізу.

Глобальна економічна нестабільність, насичення ринку послуг та інтенсифікація конкуренції, а також зростаючі вимоги споживачів до якості продукції вимагають пріоритетного вирішення завдань оптимізації управління проектами в сучасних компаніях.

Різні підходи та аспекти комплексного вирішення проблем ефективності економіки, організації, планування та управління на всіх рівнях, зокрема в умовах сучасного бізнесу та маркетингової діяльності, широко досліджені та висвітлені в економічній і науково-практичній літературі.

Теми удосконалення управлінських систем підприємств та підвищення їхньої ефективності досліджуються у працях таких фахівців, як Ф. Котлер, М. Хаммер, Дж. Чампі, Т. Сааті, М. Портер та Р. Каплан. Ці дослідники заклали основу сучасних підходів до стратегічного управління, маркетингової оптимізації

та реінжинірингу бізнес-процесів, що сьогодні активно застосовуються в різних сферах, включно з маркетинговою діяльністю.

Зокрема, роботи Майкла Портера з конкурентної стратегії та аналізу конкурентних переваг є визначальними для розуміння стратегічного управління та позиціонування підприємств у умовах зростаючої конкуренції. Теоретичні та практичні підходи, запропоновані Робертом Капланом та Девідом Нортонем у концепції збалансованої системи показників, допомагають встановлювати ключові показники ефективності, що особливо актуально в умовах цифровізації бізнесу [9].

Сучасні підходи до оптимізації управління на підприємствах розвинулися також завдяки працям Томаса Сааті, Філіпа Котлера та Джеймса Чампі [18]. Зокрема, використання методу аналізу ієрархій (АНП) Сааті дозволяє керівникам приймати багатокритеріальні рішення, що особливо важливо в маркетинговій діяльності, де необхідно швидко адаптуватися до змін ринку.

Внесок Майкла Хаммера та Джеймса Чампі у розвиток концепції реінжинірингу бізнес-процесів допоміг підприємствам підвищувати гнучкість та швидкість реагування, що є важливим для досягнення успіху у швидкозмінному маркетинговому середовищі [7].

Не дивлячись на існуючі досягнення в галузі дослідження бізнес-процесів та методів управління, чимало питань залишаються дискусійними та звичайно ж потребують подальших досліджень. Існуючі підходи охоплюють різні методи, критерії та фактори розвитку сучасних компаній, проте часто не мають єдиної концептуальної основи для застосування процесного підходу до управління.

Саме тому використання системного аналізу у цьому контексті забезпечує глибше розуміння взаємодії між різними процесами та вдосконалення управління ними. Зі зростанням масштабів та складності сучасних проєктів виникає потреба у вдосконаленні процесів керування ресурсами підприємства шляхом застосування сучасних методів і технік управління.

Тимбільше, що завдяки розширенню функціональних можливостей та обсягу даних, сучасні проекти стають більш масштабними, що призводить до збільшення навантаження на ресурси і підвищення вимог до координації.

Для розкриття теми нашого дослідження наведемо таблицю з тлумаченнями поняття «оптимізаційна задача» (табл. 1.1).

Таблиця 1.1

## Тлумаченнями поняття «оптимізаційна задача»

Автор	Визначення терміну	Джерело
George Dantzig	Лінійна оптимізація для вирішення задач, де метою є максимізація або мінімізація цільової функції під визначеними обмеженнями	Dantzig, G. B. (2016). <i>Linear Programming and Extensions</i> . Princeton University Press.
Thomas L Saaty	Багатокритеріальна оптимізація, яка застосовується для прийняття рішень, що враховують кілька критеріїв одночасно	Saaty, T. L (2012). <i>Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World</i> . RWS Publications.
Thomas H. Davenport & Rajeev Ronanki	Використання штучного інтелекту для оптимізації рутинних процесів з метою підвищення продуктивності та економії часу і ресурсів	Davenport, T. H.. & Ronanki, R. (2018). <i>Artificial Intelligence for the Real World</i> . Harvard Business Review. 96(1). 108-116.
Jan M. Müller, Olga Buliga, Kai-Ingo Voigt	Оптимізація бізнес-моделей, зокрема для малого і середнього бізнесу, з фокусом на адаптацію до цифрової трансформації та Індустрії 4.0	Muller, J. M., Buliga, O., & Voigt, K. I. (2018). <i>Fortune Favors the Prepared: How SMEs Approach Business Model Innovations in Industry 4.0</i> . Technological Forecasting and Social Change. 132.2-17.
Dmitry Ivanov & Alexandre Dolgui	Оптимізація для підвищення стійкості ланцюгів поставок у кризових ситуаціях включаючи забезпечення	Ivanov, D., & Dolgui, A. (2020). <i>Viability of Intertwined Supply Networks: Extending the Supply Chain Resilience</i>
Рао Ч. та Сривастава С.	Оптимізаційні моделі, що використовують цифрові стратегії для підвищення ефективності комунікації з клієнтами та прогнозування.	Rao, C., & Srivastava, S. (2018). <i>Digital Transformation Strategies for Competitive Advantage: An Empirical Study of Indian Organizations</i> . Journal of Global Information Technology Management, 21(3), 196-208.

Оптимізаційні методи та моделі мають широке застосування в різних сферах, зокрема в економіці, техніці, науці та управлінні. Тому, завдяки їм можна знаходити найкращі рішення для функціонування складних систем, незалежно від їхнього рівня ієрархії, а також підвищувати якість, надійність і достовірність створюваних моделей. Оптимізація допомагає спрощувати математичні моделі,

зменшувати їх розмірність, роблячи їх більш доступними та зручними для використання.

Управління бізнес-процесами в складних та динамічних системах повинно базуватися на основних принципах математики, логіки, системного аналізу із застосуванням обчислювальної техніки. Основними інструментами наукового вивчення діяльності сучасної компанії є математичні моделі та методи їх оптимізації.

Функціонування сучасного підприємства як об'єкта математичного моделювання і оптимізації включає комплекс різноманітних процесів і заходів, що забезпечують стабільність і конкурентоспроможність. Для опису цих процесів у науковій літературі розроблені спеціальні математичні моделі, які допомагають вирішувати такі завдання:

- планування обсягів виробництва чи послуг;
- оптимізація трудових і матеріальних ресурсів;
- управління логістичними процесами;
- фінансовий менеджмент;
- прогнозування ринкових тенденцій;
- оптимізація маркетингових стратегій;
- управління ризиками;
- підтримка прийняття управлінських рішень.

Математичні моделі, призначені для вирішення таких завдань, дозволяють абстрактно відобразити взаємозв'язки та закономірності між різними компонентами системи і є ефективним інструментом для ухвалення управлінських рішень. Вони сприяють спрощенню складних бізнес-процесів, підвищуючи продуктивність і ефективність компанії.

Моделювання бізнес-процесів сучасних компаній, що працюють у сфері послуг, має свою специфіку. Зростаюча складність операцій, у поєднанні зі швидкими змінами у вимогах клієнтів та динаміці ринку, ставить перед компаніями завдання постійної оптимізації бізнес-процесів. Розробка інформаційних систем для оптимізації процесів включає не лише стандартні

підходи, але й застосування творчих методів, що дозволяє швидко адаптуватися до ринкових змін. Зокрема, маркетингові проекти часто вимагають гнучких методологій, таких як Scrum чи Kanban, щоб забезпечити максимальну адаптивність.

Наукові дослідження українських та зарубіжних учених у сфері моделювання бізнес-процесів у сучасних компаніях охоплюють численні аспекти. Такі як наприклад, підвищення прибутковості завдяки ефективній організації маркетингових процесів, покращення якості та продуктивності роботи, оптимізацію витрат на розробку проектів тощо.

Слід зауважити, що математичні моделі управління бізнесом повинні враховувати такі фактори, як гнучкість, динамічність та інноваційність, щоб успішно моделювати та оптимізувати діяльність сучасних підприємств. Через складність таких економічних систем рекомендується використовувати системний підхід, який включає поетапне створення моделі з урахуванням різних характеристик - інформаційних, ресурсних параметрів, а також встановлення коректних взаємозв'язків між рівнями моделювання.

Управління проектами є процесом координації роботи команди та розподілу матеріальних і фінансових ресурсів протягом усього життєвого циклу проекту. Цей процес потребує застосування сучасних методів для досягнення цілей з оптимальною продуктивністю та мінімальними витратами.

У більшості випадків задача побудови математичної моделі зводиться до вирішення оптимізаційної проблеми, що полягає у знаходженні екстремумів функцій, або значень, що найкраще відповідають критеріям ефективності. Ці методи можуть працювати з різними типами обмежень, або без них, що робить їх універсальними для широкого спектра задач.

У відповідності до джерел наукових праць є декілька основних методів розв'язання оптимізаційних задач, які використовуються в залежності від специфіки задачі, типу змінних, цільової функції та обмежень. Наведемо найпоширеніші методи.

Аналітичні методи, що використовуються для розв'язання задач оптимізації, які можна виразити в аналітичній формі та для яких доступні точні математичні рішення. До них відносяться:

- лінійне програмування, застосовується, коли цільова функція і обмеження є лінійними. Одним із найвідоміших методів для розв'язання таких задач є симплексний метод Джорджа Данціга, який ефективно шукає оптимальні рішення у лінійних задачах;

- квадратичне програмування, фахівці використовують, коли цільова функція є квадратичною, а обмеження - лінійними. Наприклад, у фінансових моделях для портфельного інвестування;

- метод множників Лагранжа, використовується для таких задач, де необхідно знайти екстремум функції при наявності рівнянних обмежень. Цей метод часто застосовується в економіці та інженерії для оптимізації ресурсів.

Чисельні методи. Експерти та науковці їх використовують для задач, які складно, або неможливо розв'язати аналітично. Вони використовують ітеративні підходи для знаходження наближених рішень:

- метод градієнтного спуску, який використовується для задач оптимізації з гладкими (диференційованими) цільовими функціями. Алгоритм поступово зменшує значення цільової функції, рухаючись у напрямку протилежному до градієнта;

- Ньютонівські методи більш швидкі і точні, ніж методи градієнтного спуску, але вимагають обчислення другої похідної (гесіана). Ефективні для задач з нелінійними цільовими функціями;

- метод скінчених різниць: використовується для обчислення похідних, якщо неможливо аналітично обчислити градієнти.

Методи евристичної оптимізації. Такі методи корисні для складних задач оптимізації, особливо коли цільова функція не є гладкою, або для задач із великою кількістю локальних мінімумів:

- генетичні алгоритми. Метод, схожий з процесом природного відбору. Він включає операції схрещування, мутації і відбору для пошуку оптимального рішення. Використовується в задачах оптимізації маршруту, розкладках і плануванні;

- метод табуйованого пошуку, що використовує структуру пам'яті для збереження інформації про вже відвідані рішення, щоб уникнути зациклення на них. Застосовується для дискретних і комбінаторних задач;

- метод імітаційного відпалу, споріднений з фізичним процесом відпалу металів. Метод поступово знижує "температуру" системи для знаходження глобального мінімуму цільової функції.

Стохастичні методи. Ці методи використовуються для задач, де деякі параметри мають випадкову природу. Вони є популярними в задачах оптимізації в умовах невизначеності:

- Метод Монте-Карло, що використовує випадкові вибірки для оцінки результатів і визначення оптимальних рішень. Застосовується у фінансових та інженерних розрахунках;

- метод стохастичного градієнтного спуску, використовує випадкові вибірки даних для кожного ітеративного кроку. Популярний у задачах машинного навчання.

Методи динамічного програмування. Використовуються для розв'язання задач оптимізації, які можна розділити на підзадачі. Основна ідея полягає в побудові рішення через послідовне розв'язання підзадач і збереження їхніх результатів:

- Алгоритм Беллмана, ефективний для задач маршрутизації, логістики, планування і керування запасами. Алгоритм зберігає рішення підзадач, що знижує обчислювальні витрати.

Агентні та імітаційні методи. Використовуються для моделювання та оптимізації складних систем з багатьма взаємодіючими компонентами або агентами:

- агентне моделювання дає змогу досліджувати і оптимізувати процеси, де беруть участь автономні агенти, які взаємодіють один з одним. Використовується для моделювання поведінки на ринку, соціальних системах тощо.

- дискретно-подієве моделювання, моделює системи, де зміни відбуваються дискретно в часі. Підходить для оптимізації процесів у виробництві та обслуговуванні.

Наведемо таблицю методів рішення оптимізаційних задач в табл. 1.2.

Таблиця 1.2

Категорія методу	Метод	Опис	Застосування
Аналітичні методи	Лінійне програмування	Пошук оптимуму для лінійної цільової функції з лінійними обмеженнями.	Логістика, виробниче планування
	Квадратичне програмування	Мінімізація/максимізація квадратичної цільової функції з лінійними обмеженнями	Фінансове моделювання. портфельне інвестування
	Метод множників Лагранжа	Оптимізація з рівняними обмеженнями для знаходження екстремуму функцій	Економіка, інженерія
Чисельні методи	Метод градієнтного спуску	Поступове зменшення цільової функції шляхом руху у напрямку протилежному до градієнта	Машинне навчання, обробка даних
	Ньютоновські методи	Використовують другу похідну для швидкого пошуку мінімуму/максимуму функції	Нелінійна оптимізація
	Метод скінчених різниць	Чисельний метод для обчислення похідних, коли аналітичні обчислення неможливі	Чисельні розрахунки в інженерії
Евристичні методи	Генетичні алгоритми	Алгоритм, натхненний природним відбором, що включає мутацію, схрещування і відбір для пошуку оптимуму	Оптимізація маршруту, планування
	Метод табуїзованого пошуку	Зберігає вже відвідані рішення, щоб уникнути повторного пошуку; застосовується для задач з великим простором рішень	Дискретні та комбінаторні задачі
	Метод імітаційного відпалу	Пошук глобального мінімуму шляхом поступового зниження «температури» системи	Оптимізація виробничих процесів

## Закінчення таблиці 1.2

Стохастичні методи	Метод Монте-Карло	Використовує випадкові вибірки для оцінки можливих результатів	Фінансові розрахунки, інженерія
	Стохастичний градієнтний спуск	Використовує випадкові вибірки даних для кожної ітерації	Машинне навчання
Методи динамічного програмування	Алгоритм Веллмана	Розв'язання задач маршрутизації планування через збереження проміжних рішень	Логістика, керування запасами
Агентні та імітаційні методи	Агентне моделювання	Моделювання поведінки автономних агентів, що взаємодіють між собою	Соціальні та економічні моделі
	Дискретно-подієве моделювання	Моделювання систем, де зміни відбуваються в певні дискретні моменти часу.	Виробництво, оптимізація обслуговування

Як видно з наведених методів, аналітичні методи корисні для задач, що допускають точне математичне рішення, тоді як чисельні та евристичні методи застосовуються для складних або нелінійних задач, де важливо знайти наближене рішення.

Для ефективного застосування оптимізаційних методів необхідні ґрунтовні знання в галузі математичного моделювання та економіко-математичних підходів. Також важливо розуміти методику інтерпретації результатів моделювання, що дозволяє коректно аналізувати процеси та явища, які лежать в основі досліджуваних задач. Застосування оптимізаційних моделей є важливим інструментом для прогнозування та стратегічного планування, оскільки вони дозволяють знаходити найбільш вигідні сценарії розвитку подій і приймати обґрунтовані управлінські рішення, які сприяють підвищенню ефективності та рентабельності.

У відповідності до чого, застосування таких методів у бізнесі та підприємницькій діяльності дозволяє отримувати більш точні прогнози, ефективно розподіляти ресурси і приймати управлінські рішення маркетологами, менеджерами та навіть стейкхолдерами підприємства в умовах ризику та

невизначеності. А потреба в адаптивних, самонавчальних системах особливо актуальна в умовах швидкозмінного ринку, де підприємства стикаються з постійними змінами в умовах постачання, попиту, а також змінних економічних умов, як на мікро так і на макрорівні.

Традиційні підходи часто не забезпечують достатньої гнучкості для ефективного управління у ситуаціях, коли інформація є неповною, а зовнішні обставини швидко змінюються. Саме тому розробка інтелектуальних систем, що базуються на методі Монте-Карло та включають адаптивні механізми навчання, є перспективним напрямом у сфері автоматизації прийняття рішень. Такі системи можуть враховувати численні сценарії розвитку подій та обирати найкращі стратегії для підтримки стабільності та ефективності діяльності.

Для підприємств, які працюють у висококонкурентних ринкових умовах, така система стає інструментом для ефективного управління, що дозволяє не лише швидко реагувати на зміни, а й проактивно оптимізувати свої бізнес-процеси, знижуючи витрати та підвищуючи рентабельність.

## **1.2. Постановка задач дослідження**

### **1.2.1. Роль інтелектуальних систем у підвищенні ефективності управління**

Одним з визначальних факторів досягнення ефективності діяльності підприємства є процеси управління, або менеджмент. Як зазначив «батько кібернетики» Норберт Вінер, «якщо XVII століття було віком годинників, а XIX століття - віком парових машин, то наш час є віком зв'язку та управління». У цих умовах ефективне досягнення сформульованих цілей управління потребує новітніх інструментів, здатних значно скоротити витрати, що обмежуються можливостями людини в опрацюванні даних [21].

Загалом, процес ухвалення рішення включає три основні етапи або фази (рис. 1.1). Фаза аналізу пов'язана з визначенням головних та допоміжних цілей ухвалення рішення. На цьому етапі проводиться діагностика проблеми,

ідентифікація та опис проблемної ситуації, а також збір релевантної інформації та даних. Це дозволяє визначити напрямок пошуку рішень і виключити варіанти, що не відповідають поставленим цілям.



Рис. 1.1. Основні етапи прийняття управлінських рішень

Тому інтелектуальні системи, зокрема такі як Decision Support Systems (DSS), створені для підтримки управлінських рішень у складних та динамічних середовищах, дозволяють використовувати, як традиційні алгоритми обробки даних, так і передові методи машинного навчання та штучного інтелекту. Від початку 70-х років XX століття інтенсивно розроблялися засоби автоматизованої підтримки прийняття рішень, у результаті чого з'явилися DSS, здатні обробляти велику кількість даних, забезпечувати швидке реагування на зміни, знижувати ризики та підвищувати продуктивність [16].

Саме тому поняття «управління знаннями» стало важливою складовою сучасного менеджменту, оскільки виникає потреба у «трансформації» інформації у знання для використання, як базису для прийняття правильних управлінських рішень. Сучасні інтелектуальні системи забезпечують цей процес завдяки збиранню, аналізу та перетворенню даних у знання. Роль DSS у підвищенні ефективності управління полягає у зниженні витрат, скороченні часу на аналіз даних та оптимізації ресурсів [21].

Системи підтримки прийняття рішень, що здатні автоматично збирати та обробляти інформацію, допомагають сучасним компаніям оперативно реагувати на ринкові зміни, підвищуючи тим самим їх конкурентоспроможність на ринку.

Дослідження таких мислителів як Аристотель, Декарт, Платон та Демокрит стали основою для сучасних концепцій системного аналізу. У XIX-XX століттях ідеї науковців та натуралістів, таких як Богданов, Берталанфі та сам Вінер,

акцентували роль модельного мислення у дослідженні процесів [24]. Ці ідеї знайшли своє продовження у розвитку штучних систем, здатних адаптуватися до середовища та ефективно підтримувати процеси управління

Таким чином, роль інтелектуальних систем у сучасному бізнесі полягає не лише в автоматизації управлінських процесів, але й у можливості створення адаптивних систем, що дозволяють приймати швидкі та ефективні рішення на основі аналізу великих обсягів даних. Що робить інтелектуальні системи надзвичайно важливим інструментом, який не лише підвищує якість управлінських рішень, але й сприяє стійкості та ефективності маркетингової, логістичної та й взагалі бізнес діяльності підприємства.

Отже, ухвалення рішення можливе на основі знань про об'єкт управління, процеси, які в ньому відбуваються або можуть відбутися з часом, а також за умови наявності набору показників, що визначають ефективність та якість прийнятого рішення.

Враховуючи особливості об'єкта управління, умови та сам процес ухвалення рішення, можна здійснити класифікацію рішень, яка представлена у таблиці 1.3.

Таблиця 1.3

## Класифікація видів рішень

Ознака	Вид рішення		
Ступінь структуризації проблеми	Що приймається в умовах гарно структурованог проблеми	Що приймається в умовах погано структурованог проблеми	Що приймається в умовах не структурованої проблеми
Рівень інформованості про стан проблеми	Що приймається в умовах визначеності	Що приймається в умовах ризику	Що приймається в умовах невизначеності
Кількість етапів процесу прийняття рішення	Статичне (один етап)		Динамічне (кілька етапів)
Кількість ОПР	Одноособове		Групове (багато осіб, що приймають рішення)
Зміст рішення	Тактичне		Стратегічне

Поставлену перед нами мету нашого дослідження досягнемо шляхом розробки інтелектуальної системи підтримки прийняття рішень маркетолога та директора підприємства.

Сучасні системи підтримки прийняття рішень (СППР) сформувалися як результат інтеграції управлінських інформаційних систем з технологіями управління базами даних.

СППР спрямовані на вирішення широкого спектра завдань щоденної управлінської діяльності та надають цінну підтримку маркетологам та управлінцям, що здійснюють дослідження та приймають управлінське рішення. Завдяки СППР можна ефективно обирати варіанти рішень для неструктурованих та слабоструктурованих завдань, зокрема тих, що мають численні критерії оцінки та аналізу, що дає змогу глибше зрозуміти складні проблеми та прогнозувати можливі результати за певних умов.

У відповідності до наукових джерел, головною метою СППР є надання аналітикам інструментарію для ефективного аналізу даних, що відрізняється за рівнем деталізації та функціональністю.

Так прийнято виділяти, у контексті аналізу даних СППР такі поставлені перед ними основні типи завдань:

- інформаційно-пошукові завдання. Коли СППР виконує пошук необхідних даних на основі попередньо визначених запитів, забезпечуючи доступ до необхідної інформації швидко і в зрозумілому форматі;
- оперативно-аналітичні завдання. Завдання коли СППР здійснює сортування, групування та консолідацію даних у форматі, придатному для аналізу, що дозволяє аналітикам краще розуміти структуру і тенденції в даних;
- інтелектуальні завдання. Коли СППР знаходить функціональні та логічні закономірності в накопичених даних, будує моделі та правила, які допомагають пояснити ці закономірності та прогнозують розвиток процесів із певною вірогідністю.

За роки розробок, було встановлено, що сучасна архітектура СППР включає компоненти, що дозволяють об'єднувати, аналізувати та обробляти дані у реальному часі.

А схематично класичну архітектуру ІС можна зобразити, показуючи ключові елементи, такі як сервер сховища даних, OLAP-інструментарій для

багатовимірному аналізу даних та інструменти Data Mining для виявлення прихованих закономірностей і трендів рис. 1.2.



Рис. 1.2. Загальна архітектура системи прийняття управлінських рішень

Важливим елементом сучасних систем підтримки прийняття рішень є підсистема введення даних OLTP (Online Transaction Processing), яка відповідає за швидку обробку транзакцій у режимі реального часу [12]. Що в свою чергу забезпечує оперативне опрацювання поточних даних за допомогою стандартних систем управління базами даних (СУБД) і відповідно дозволяє підприємствам значно підвищити швидкість та ефективність обробки щоденних операцій.

OLTP-системи, завдяки своїй здатності швидко обробляти великий обсяг транзакцій, забезпечують надійне підґрунтя для подальшого аналізу, на якому базуються управлінські рішення.

Підсистема зберігання даних у СППР побудована на основі сучасних СУБД та концепції сховищ даних, що дозволяє розділяти структури для оперативної обробки та для виконання аналітичних запитів маркетологів. Концепція сховища даних передбачає створення окремих шарів зберігання, що дозволяє ефективно організувати великий обсяг даних і оптимізувати доступ до них під час виконання як рутинних, так і аналітичних завдань.

Підсистема аналізу, як правило, складається з трьох основних компонентів, що забезпечують різні типи аналізу даних:

Підсистема інформаційно-пошукового аналізу - ця підсистема базується на реляційних СУБД і статичних SQL-запитах, що дозволяють проводити точний та швидкий пошук інформації за попередньо визначеними критеріями.

Підсистема оперативного аналізу (OLAP) - для реалізації оперативного аналізу застосовують технологію багатовимірної аналітичної обробки даних OLAP (Online Analytical Processing), дозволяє представляти дані у форматі, придатному для детального аналізу за кількома вимірами. А це дає можливість швидкого доступу до важливої інформації, і є фактором прийняття рішень.

Підсистема інтелектуального аналізу (Data Mining) - ця підсистема включає алгоритми та методи добування даних, що дозволяють виявляти приховані закономірності в даних та будувати прогностичні моделі, корисні для стратегічного планування та оптимізації процесів.

Наведені вище компоненти СППР розглядають важливі аспекти, такі як накопичення та моделювання даних на концептуальному рівні, інтеграцію даних з різних незалежних джерел, а також їхній аналіз для виявлення тенденцій і моделей.

Використання технологій OLAP надає можливість доступу до багатовимірних даних, хоча їх функціонал часто обмежується виконанням аналітичних запитів та підтримкою звітності, не охоплюючи всі аспекти глибокого інтелектуального аналізу. Таким чином, підсистеми OLTP, сховищ даних та аналітичні компоненти створюють основу для багатогранного аналізу, що дозволяє приймати обґрунтовані рішення для сучасних підприємств будь-якої сфери діяльності.

### **1.2.2. Головні аспекти методу Монте-Карло та можливості детермінованого навчання**

У відповідності до наукових праць, метод Монте-Карло (ММК) є одним із найвідоміших методів, що використовуються для моделювання складних систем і вирішення задач оптимізації.

Він базується на використанні випадкових величин для імітації процесів і передбачення результатів, що дозволяє отримувати наближені рішення у ситуаціях, де аналітичні, або детерміновані методи не можуть бути ефективно застосовані через надмірну складність або багатовимірність задачі. Метод отримав популярність у багатьох сферах, зокрема в управлінні фінансовими ризиками, наукових дослідженнях, розробці програмного забезпечення та оптимізації бізнес-процесів.

Метод Монте-Карло, названий на честь відомого казино в Монако, представляє собою статистичні алгоритми, які застосовуються для чисельного розв'язання задач та моделювання. Його основа - повторювана випадкова вибірка для отримання числових результатів. Сама концепція полягає у використанні випадкових значень для розв'язання задач, які в іншому випадку можна було б вирішити детермінованими методами.

Метод Монте-Карло базується на законі великих чисел, який стверджує, що середнє значення багатьох спостережень наближається до очікуваного значення, і точність підвищується з кожною наступною ітерацією.

При використанні методу відбуваються наступні етапи проведення типового моделювання методом Монте-Карло:

- визначення області вхідних даних. Етап де необхідно задати можливі вхідні значення, які можуть охоплювати різні параметри - від цін на акції до фізичних величин;

- генерація випадкових вхідних даних. На цьому етапі випадково генеруються вхідні значення, спираючись на розподіл ймовірностей, який відповідає реальній ситуації, яку необхідно змоделювати;

- виконання детермінованих обчислень. Такий етап, на якому для кожного згенерованого набору вхідних значень здійснюються детерміновані обчислення, які можуть включати певні математичні формули.

- агрегування результатів. Заключний етап, після проведення багатьох ітерацій, де результати окремих обчислень об'єднуються для формування кінцевого результату.

Метод дозволяє отримати надійні прогнози та рішення для складних задач, використовуючи елемент випадковості для моделювання реальних сценаріїв.

Цей науковий підхід є особливо ефективним при роботі з багатовимірними моделями, де кількість можливих рішень може бути дуже великою. Такі алгоритми використовують не тільки для випадкового моделювання, але й для оптимізації, коли відомо, що цільова функція має кілька локальних максимумів, або мінімумів.

Суттєвою перевагою ММК є його здатність працювати із оптимізаційними задачами, що включають широкий спектр невизначеностей, а також з обмеженнями, що робить його ефективним для застосування у бізнес-середовищі. Наприклад, у ситуаціях, коли потрібно оцінити ризики інвестиційних проектів, аналіз фінансових потоків, або оцінити ринкові перспективи при маркетингових дослідженнях, ММК дозволяє отримати надійні прогнози на основі симуляційних моделей.

Однак слід зауважити, що класичний метод Монте-Карло має свої обмеження, серед яких значна залежність від обсягу даних і часу виконання обчислень. Детерміноване навчання дозволяє значно підвищити ефективність ММК, використовуючи систематичний підхід до вибору сценаріїв і зменшуючи обсяг випадкових факторів, що знижує час обчислення і підвищує точність результатів.

А саме, включення детермінованого навчання у процес симуляції дозволяє алгоритму Монте-Карло розширити можливості моделювання завдяки оптимізації обробки великих обсягів даних. Тобто детерміноване навчання означає, що система здатна аналізувати попередні результати, робити висновки і пристосовувати модель до нових сгенерованих, або отриманих умов. Що може бути досягнуто шляхом додавання фільтрації даних, алгоритмів кластеризації, або іншого аналізу даних, який допомагає зосередитися на найбільш релевантних випадках. Такий підхід зменшує кількість випадкових вибірок, що значно прискорює процес оптимізації, особливо у задачах з великою кількістю вхідних параметрів або змінних.

Метод Монте-Карло широко використовується у задачах оптимізації, особливо коли ці задачі потребують врахування багатьох змінних та різних сценаріїв розвитку подій. Наприклад, у процесах управління запасами, логістиці, маркетинговій діяльності, плануванні ресурсів та фінансовому моделюванні ММК допомагає розробити оптимальні стратегії за умов невизначеності.

Для оптимізаційних задач, що містять велику кількість локальних оптимумів, детерміноване навчання дозволяє сфокусувати метод Монте-Карло до найбільш перспективних рішень, не виконуючи повний перебір всіх можливих варіантів. Це дозволяє скоротити час обчислення і підвищити продуктивність, зберігаючи точність і надійність розрахунків.

При розробці інтелектуальних систем метод Монте-Карло у поєднанні з детермінованим навчанням може використовуватися для створення адаптивних моделей, що постійно вдосконалюються на основі отриманих результатів. Такі системи здатні аналізувати попередні прогнози, виявляти закономірності і оновлювати свої параметри, що дозволяє отримувати більш точні результати у процесі оптимізації.

В інтелектуальних системах для автоматизації управлінських процесів ММК з детермінованим навчанням може забезпечувати ефективність та гнучкість прийняття рішень, що дозволяє компаніям підвищувати продуктивність та конкурентоспроможність у мінливих ринкових умовах.

У бізнес діяльності підприємства метод Монте-Карло з детермінованим навчанням використовується для розв'язання широкого спектру завдань, зокрема:

- управління ризиками. ММК допомагає прогнозувати і знижувати фінансові ризики, враховуючи численні фактори, які впливають на кінцевий результат;

- оптимізація виробництва та логістики. Компанії застосовують цей метод для аналізу процесів постачання, визначення оптимальних маршрутів та управління запасами.

- маркетингові дослідження та прогнозування. Використовують для прогнозування попиту, кількості лідів, ефективності рекламних компаній, ROMI, тощо.

- фінансовий аналіз і прогнозування. Метод використовується для прогнозування грошових потоків, аналізу рентабельності інвестицій та оптимізації капіталовкладень.

Розглянемо основні принципи методу Монте-Карло:

- Випадковість. У відповідності до вище розглянутого, головною в методі Монте-Карло є генерація випадкових величин для апроксимації шуканих величин, або функцій. Що дозволяє розглянути широкий спектр можливих значень, які могли б виникнути в реальності, забезпечуючи тим самим більш точне та об'єктивне відображення системи. Що корисно, коли система має багато невизначених параметрів, і прогнозування її поведінки потребує врахування численних варіацій.

- Моделювання системи. ММК широко застосовується для моделювання різних систем, зокрема фізичних, фінансових, економічних та інших. Такі системи можуть бути описані математичними моделями, які потім аналізуються шляхом генерації випадкових величин. Наприклад, у фінансовому моделюванні метод допомагає оцінити поведінку ринків за умов нестабільності, імітуючи різні сценарії для надійнішого прогнозування.

- Стохастичні обчислення. Тобто коли замість точних обчислень, які можуть бути складними, або навіть неможливими для виконання, метод Монте-Карло використовує стохастичні (випадкові) обчислення для апроксимації значень. В основі цього підходу лежить проведення великої кількості ітерацій, що дозволяє отримати точніші результати. Чим більше ітерацій виконується, тим вищою є надійність отриманих середніх значень, оскільки вони стають більш наближеними до справжніх величин системи.

- Збіжність: Один із ключових принципів методу полягає у збіжності - тобто можливості методу Монте-Карло наближатися до точного значення або правильного рішення зі збільшенням кількості випробувань. Це властивість

гарантує, що в разі достатньої кількості симуляцій, результати набудуть високої точності, що є важливим для задач із великою кількістю змінних і непередбачуваних параметрів.

- Варіації методу. Існує декілька варіацій методу Монте-Карло, таких як метод «великих частинок», метод Метрополіса-Гастінгса, та метод зразків із важкими хвостами, кожен з яких має свою специфіку та переваги залежно від поставленої задачі. Ці варіації дозволяють підвищити ефективність і точність обчислень, вибираючи підходи, які найбільше відповідають характеру конкретної задачі.

- Універсальність застосування. Метод Монте-Карло може бути застосований у розв'язанні широкого спектра задач, включаючи обчислення інтегралів, розв'язання диференціальних рівнянь, апроксимацію функцій, моделювання поведінки складних систем та оптимізацію параметрів. У задачах, де аналітичні методи складні або неможливі, метод Монте-Карло стає особливо ефективним завдяки своїй гнучкості та здатності охоплювати велику кількість випадкових варіантів.

Наведемо основні переваги та недоліки методу Монте-Карло, див. табл. 1.3.

У відповідності до таблиці представлено основні переваги та недоліки цього методу:

- Основною перевагою методу Монте-Карло є його гнучкість і здатність до адаптації у різних галузях. Його інтеграція з детермінованим навчанням дозволяє не тільки покращити точність і швидкість обчислень, а й зменшити залежність від випадковості, що є критично важливим для систем, що працюють у реальному часі.

- Однак, існують і певні виклики у використанні цього методу. Зокрема, метод потребує значних обчислювальних ресурсів, особливо у випадках, коли необхідно виконувати велику кількість симуляцій. Крім того, метод Монте-Карло є досить чутливим до якості вхідних даних, що означає, що помилки або неточності у даних можуть вплинути на результат.

Таблиця 1.3

## Переваги та недоліки методу Монте-Карло

Переваги	Недоліки
Гнучкість. метод Монте-Карло може бути застосований до широкого спектру проблем, включно з тими, для яких складно, або навіть неможливо отримати аналітичні розв'язки	Швидкість збіжності. для досягнення точних результатів може знадобитися велика кількість випробувань (семплів), що може призвести до високих обчислювальних втрат
Спрощення складних інтегралів. Завдяки випадковому семплюванню великої кількості точок в області метод Монте-Карло може наближено обчислювати значення складних інтегралів	Проблеми з вимірністю. Так у високорозмірних просторах вимірів метод Монте-Карло може потребувати дуже великої кількості семплів для досягнення достатньої точності
Придатність для вирішення задач великих розмірів. Метод Монте-Карло може бути застосований до великих обсягів даних або складних систем, де інші методи можуть бути непридатними через обчислювальну складність	Залежність від випадковості. Результати методу Монте-Карло можуть мати деякий рівень випадковості, що означає, що під час виконання аналізу може бути необхідно враховувати деяку невизначеність
Адаптивність до розподілу. Метод Монте-Карло працює незалежно від розподілу випадкових величин, тому він може бути застосований до проблем зі складними або невідомими розподілами	Необхідність у великій кількості обчислень. Коли залежно від проблеми може знадобитися значна обчислювальна потужність для виконання великої кількості семплів

Таким чином, метод Монте-Карло у поєднанні з детермінованим навчанням є потужним інструментом для вирішення складних оптимізаційних задач, особливо у сфері бізнесу.

Завдяки своїй гнучкості, адаптивності та здатності працювати з великим обсягом даних, цей метод надає підприємствам і організаціям можливість приймати обґрунтовані рішення, адаптуючись до змінних умов ринку.

А саме використання методу Монте-Карло у розробці інтелектуальних систем дозволяє створювати динамічні моделі, які здатні підвищувати ефективність управління та забезпечувати стратегічну стабільність у мінливому бізнес середовищі.

### 1.3. Дослідження існуючих інтелектуальних систем вирішення оптимізаційних задач

Зважаючи на згадані складнощі у прийнятті рішень, варто відзначити, що процес ухвалення рішення людиною має певні обмеження у здатності до аналізу, обробки інформації, отримання якісних і обґрунтованих рішень, а також у швидкості прийняття рішень. Обмеження в роботі особи, що приймає рішення

(ОПР), обумовлені як взаємовідносинами між людьми, так і внутрішніми психологічними та фізіологічними факторами. Крім того, під час аналізу та розв'язання багатокритеріальних задач ОПР нерідко виявляє нестабільність, невпевненість, нелогічність і схильність до спрощення задачі. Таким чином без застосування допоміжних засобів процес ухвалення рішення зазвичай має п'ять наведених нижче обмежень (рис. 1.3).



Рис. 1.3. Обмеження процесу прийняття рішення, що виконується індивідом без допоміжних засобів

Охарактеризуємо обмеження процесу прийняття рішень ОПР:

- Обмежена робоча пам'ять. Людина обробляє інформацію у проміжній робочій пам'яті (РП), яка вміщує від 3 до 8 елементів. Інформація в РП швидко зникає (через 7-13 секунд), якщо не оновлюється, і одночасно обробляти всі елементи без допомоги неможливо.

- Швидкість мислення. Кожна операція (порівняння, асоціація, висновок) потребує приблизно 0,1 секунди. Для складних процесів, що потребують реального часу, необхідний час зростає, що може перешкодити прийняттю швидких рішень.

- Джерела інформації. Людина отримує інформацію від органів чуття і з довгострокової пам'яті, де дані з часом можуть зникати. Люди схильні використовувати інформацію, яка є частіше застосовуваною, або пов'язана зі знаннями в РП.

- Обробка числових даних. Людина, навіть добре підготовлена в арифметиці, може допускати помилки при обчисленнях і забувати проміжні

результати. Кожне числове обчислення займає більше часу, ніж інші ментальні операції, і потребує повторного перерахунку, якщо проміжні дані забуті. Це обмеження змушує ОПР уникати складних обчислень і більше покладатися на якісне мислення.

- Залежність від візуального сприйняття. Люди часто покладаються на візуальне представлення результатів для ухвалення рішень. Однак цього може бути недостатньо для точного прогнозування, наприклад, траєкторій, або змін у часі. Як і в обробці числових даних, в ухваленні рішень у реальному часі потрібна висока швидкість обробки для точного аналізу просторових і часових параметрів.

Зазначені обмеження є загальними і характерні для випадків, коли рішення приймаються без допоміжних засобів (наприклад, комп'ютера), що створює специфічні труднощі при розробці систем підтримки ухвалення рішень.

Саме через це та виходячи з того, що було нами зазначено у попередніх підрозділах, а саме те, що інтелектуальні системи для розв'язання оптимізаційних задач широко застосовуються в різних галузях, забезпечуючи ефективність та точність прийняття рішень. Наведемо деякі з них:

1. Інтелектуальні системи підтримки прийняття рішень (ІСППР). ІСППР використовуються для аналізу великих обсягів даних та надання рекомендацій щодо оптимальних рішень. Вони базуються на онтологіях предметних галузей та задач, що дозволяє моделювати знання та процеси прийняття рішень.

2. Системи на основі штучного інтелекту та машинного навчання. Штучний інтелект (ШІ) та машинне навчання (МН) активно застосовуються для оптимізації управління проектами, генерації контенту, просування, дизайну та автоматизації. Наприклад, діджитал-агентство Mobios.ua впровадило інструменти ШІ та МН для автоматизації рутинних процесів, покращення SEO та маркетингових стратегій, що підвищило ефективність управління проектами та якість послуг [29].

3. Інтелектуальні системи управління логістичними процесами. У логістиці ШІ та МН використовуються для оптимізації маршрутів, передбачення подій та автоматизації процесів. Системи аналізують великі обсяги даних, що дозволяє

підприємствам ефективніше використовувати ресурси та підвищувати якість обслуговування клієнтів [27].

4. Експертні системи. Експертні системи моделюють процес прийняття рішень експертами в певній галузі. Вони використовують бази знань та правила виведення для надання рекомендацій. Наприклад, система MYCIN, розроблена для діагностики бактеріальних інфекцій та призначення відповідного лікування, демонструє ефективність експертних систем у медицині [29].

5. Нечіткі системи управління. Нечіткі системи використовують нечітку логіку для моделювання процесів, де існує невизначеність або неточність. Вони застосовуються в управлінні складними системами, де традиційні методи неефективні. Прикладом є нечіткі контролери в системах автоматичного керування, які забезпечують стабільність та адаптивність роботи системи [29].

6. Комбіновані інтелектуальні системи. Ці системи поєднують різні підходи, такі як нейронні мережі, нечітка логіка та генетичні алгоритми, для досягнення більшої ефективності в розв'язанні оптимізаційних задач. Вони дозволяють враховувати різні аспекти проблеми та знаходити оптимальні рішення в складних умовах [27].

7. Системи на основі генетичних алгоритмів. Генетичні алгоритми імітують процес природного відбору для пошуку оптимальних рішень. Вони ефективні в задачах, де існує велика кількість можливих рішень, і традиційні методи пошуку не дають результатів. Наприклад, генетичні алгоритми використовуються для оптимізації розкладів, маршрутизації та інших комбінаторних задач.

8. Системи на основі методів Монте-Карло. Методи Монте-Карло використовують випадковість для моделювання та аналізу складних систем. Вони застосовуються для оцінки ризиків, прогнозування та оптимізації в умовах невизначеності. Наприклад, у фінансовій сфері методи Монте-Карло використовуються для моделювання поведінки ринків та оцінки вартості опціонів.

9. Системи на основі нейронних мереж. Нейронні мережі здатні навчатися на основі даних та виявляти складні залежності. Вони застосовуються для

прогнозування, класифікації та оптимізації в різних галузях, включаючи розпізнавання образів, обробку природної мови та управління процесами.

10. Системи на основі багатокритеріальної оптимізації. Багатокритеріальні системи враховують кілька критеріїв при прийнятті рішень, що дозволяє знаходити компромісні рішення в складних ситуаціях. Вони застосовуються в управлінні проектами, плануванні ресурсів та інших сферах, де необхідно балансувати між різними вимогами та обмеженнями.

Отже, ІСПП насамперед спрямована на підвищення ефективності прийняття рішень. На відміну від адміністративних систем, які зосереджені на інформаційному та аналітичному процесах, головним результатом ІСПП є саме ефективність ухвалення рішень. Це забезпечується ключовими характеристиками сучасних ІСПП (рис. 1.4).



Рис. 1.4. Основні характеристики сучасних інтелектуальних систем підтримки прийняття рішень

Таким чином, можна стверджувати, що сучасні інтелектуальні системи для розв'язання оптимізаційних задач є потужним інструментом, що дозволяє підвищити ефективність та точність прийняття рішень у різних галузях. Використання сучасних технологій, таких як штучний інтелект, машинне навчання та методи Монте-Карло, відкриває нові можливості для оптимізації процесів та підвищення конкурентоспроможності підприємств.



## 2. МОДЕЛЮВАННЯ СИСТЕМИ

### 2.1. Теоретико-методологічні аспекти моделювання системи

Розробка інтелектуальної системи вирішення оптимізаційних задач на основі методу Монте-Карло з детермінованим навчанням вимагає ретельного підходу до кожного етапу життєвого циклу розробки програмного забезпечення. Дотримуючись стандартних фаз життєвого циклу розробки програмного забезпечення, ми можемо забезпечити високу ефективність і надійність системи, що буде здатна приймати оптимальні рішення в умовах змінних і складних сценаріїв.

Відповідно до наукових джерел, існує кілька моделей життєвого циклу (ЖЦ) для розробки програмного забезпечення, які визначають послідовність і зміст етапів створення продукту. Розуміння життєвого циклу програмного продукту є ключовим для якісної розробки, оскільки ця концепція охоплює всі етапи, через які проходить програмне забезпечення з моменту його ідеї до впровадження та обслуговування.

Життєвий цикл програмного забезпечення, або цикл розробки ПЗ, є структурною схемою, яка включає різні етапи, що є стадіями процесу розробки. На кожному з цих етапів виконується спеціалізована діяльність, від аналізу потреб і проектування до кодування, тестування та впровадження. Основна мета цього циклу полягає у створенні методичного підходу до розробки, що сприяє забезпеченню високої якості, структурованості, стабільності та простоти підтримки програмного продукту.

ЖЦ розробки програмного забезпечення є підходом, який визначає, які саме процеси, методи та інструменти необхідні для успішного завершення проєкту. Важливо, що принципи розробки життєвого циклу можуть відрізнятися залежно від специфіки проєкту чи компанії, але на міжнародному рівні існують стандарти,

такі як ISO/IEC 12207, які визначають прийняті практики для розробки та підтримки ПЗ.

Модель ЖЦ спрямована на забезпечення ефективності, рентабельності та якості програмного продукту, який принесе користь усім зацікавленим сторонам. ЖЦ програмного продукту передбачає початковий етап - визначення вимог, і завершальний - впровадження програмного продукту (рис. 2.1).



Рис. 2.1. Водоспадна модель життєвого циклу розробки програмного продукту

Цикл розробки починається з аналізу, на якому команди розробників, аналітиків та інших учасників процесу обговорюють і деталізують вимоги до кінцевого продукту. Основна мета цього етапу - сформулювати чіткі, детальні вимоги до системи, щоб забезпечити спільне розуміння між усіма учасниками, які завдання стоять перед проектом і як кожна вимога буде реалізована на практиці.

Для кращого розуміння розглянемо, етапи ЖЦ, які ми пройдемо для розробки інтелектуальної системи.

Етап планування та аналіз вимог. На цьому етапі важливо зібрати і структурувати всі вимоги, щоб сформувати повне уявлення про кінцевий продукт. Це включає визначення специфіки задач, які буде вирішувати система, серед яких можуть бути задачі управління ресурсами, розподілу робочого часу, оптимізації логістики чи планування маркетингових процесів. Крім того, ми визначаємо вимоги до продуктивності, обчислювальної потужності, швидкості виконання і

точності рішень, оскільки метод Монте-Карло потребує великої кількості ітерацій, щоб досягти високої точності.

Крім цього, також оцінюємо можливість впровадження детермінованого навчання, яке має підвищити ефективність системи через адаптацію до отриманих даних. Результатом є документ вимог (ТЗ), в якому описано всі необхідні характеристики системи, її функції, обмеження і передбачувані умови експлуатації (табл. 2.1).

Таблиця 2.1

Технічне завдання на створення інтелектуальної системи прийняття рішення  
(скорочений варіант)

Розділ ТЗ	Опис
<b>Мета проєкту</b>	Створення інтелектуальної системи, що допомагає приймати оптимальні рішення для задач з невизначеністю, поєднуючи метод Монте-Карло з детермінованим навчанням для підвищення точності та адаптивності
<b>Опис системи</b>	Система включатиме модулі для генерації випадкових вибірок, детермінованого навчання, оптимізації, інтерфейсу користувача та зворотного зв'язку. На основі методу Монте-Карло система аналізуватиме сценарії, адаптуючись до змінних вхідних даних
<b>Функціональні вимоги</b>	<ul style="list-style-type: none"> <li>- Швидке виконання розрахунків</li> <li>- Обробка великих обсягів даних</li> <li>- Зручний інтерфейс для введення даних та візуалізації результатів</li> </ul>
<b>Нефункціональні вимоги</b>	<ul style="list-style-type: none"> <li>- Надійність роботи</li> <li>- Масштабованість для розширення функцій</li> <li>- Захист користувацьких даних і обмежений доступ до адміністрування</li> </ul>
<b>Етапи розробки</b>	<ol style="list-style-type: none"> <li>1. Аналіз вимог</li> <li>2. Проектування архітектури</li> <li>3. Кодування</li> <li>4. Тестування</li> <li>5. Впровадження та підтримка</li> </ol>
<b>Технологічні вимоги</b>	<ul style="list-style-type: none"> <li>- Мова: Java</li> <li>- Інструмент розробки: Android Studio</li> <li>- Платформа: кросплатформена для ОС Windows, macOS, Linux</li> </ul>
<b>Метрики якості</b>	<ul style="list-style-type: none"> <li>- Точність прогнозів</li> <li>- Швидкість обробки даних (не більше 5 секунд на один сеанс моделювання)</li> <li>- Оцінка задоволення користувачів інтерфейсом</li> </ul>
<b>Вимірювання якості</b>	Тестування системи для оцінки точності прогнозування, продуктивності та зручності інтерфейсу
<b>Документація</b>	Користувацький посібник та технічна документація для адміністраторів і розробників
<b>Підтримка та обслуговування</b>	Регулярне оновлення системи, оперативна технічна підтримка для вирішення критичних проблем

Етап проектування архітектури. Проектування архітектури системи є фундаментальним етапом, що дозволяє побудувати міцну структуру майбутньої інтелектуальної системи. На цьому етапі ми створюємо концептуальну модель системи, що включає такі основні компоненти, як:

- модуль генерації випадкових вибірок - цей модуль необхідний для імітації різних можливих сценаріїв та обчислень за допомогою методу Монте-Карло. Він генерує варіанти рішень, що допомагає знаходити оптимальні підходи до задач (алгоритм методу наведемо на рис. 2.2);

- модуль детермінованого навчання - цей компонент дозволяє системі адаптуватися на основі зібраних даних, що допомагає моделювати більш точні результати в умовах змінних даних;

- оптимізаційний модуль - відповідає за оцінку і вибір найкращих рішень серед згенерованих варіантів відповідно до встановлених критеріїв;

- система зворотного зв'язку - забезпечує моніторинг та корекцію рішень на основі аналізу отриманих результатів, дозволяючи системі підвищувати свою точність із кожним новим набором даних.

На цьому етапі також створюються діаграми прецедентів, класів та діаграми взаємодій, що забезпечують наочність та структурованість усіх компонентів.

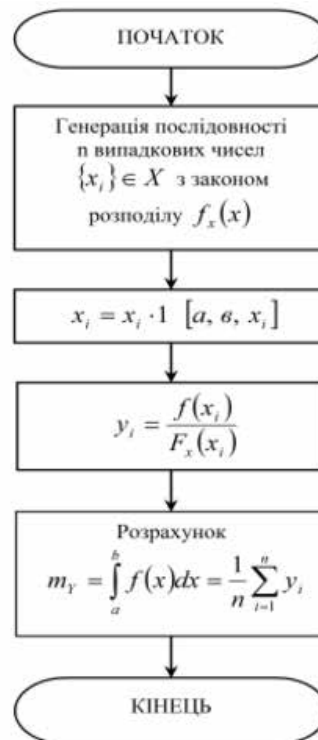


Рис. 2.2. Алгоритм методу Монте-Карло

Етап розробка програмного забезпечення. Етап розробки для нашої системи включає написання коду для кожного модуля системи на мові програмування Java. Java обрана за здатність забезпечити модульність та масштабованість системи, а також можливість інтеграції з Android Studio, що дозволяє створити мобільний інтерфейс для керування системою у разі потреби.

Основними завданнями цього етапу розробки ПЗ є:

- реалізація алгоритму генерації випадкових вибірок для оцінки різних варіантів рішень за методом Монте-Карло;
- інтеграція модуля детермінованого навчання, який використовує історичні дані для корекції й оптимізації рішень;
- розробка інтерфейсу користувача для моніторингу та управління процесом оптимізації, що дозволяє відстежувати результати, коригувати параметри і вводити дані.

Результатом буде робочий прототип, що містить функціональні компоненти системи, здатні виконувати базові завдання з оптимізації та моделювання.

Етап тестування та верифікації. Тестування є критичним етапом для перевірки надійності та продуктивності системи. Ми проводимо різні типи тестів:

- функціональне тестування для перевірки правильності роботи кожного модуля.

- продуктивне тестування для оцінки швидкості обчислень і ефективності алгоритмів.

- інтеграційне тестування для перевірки узгодженості роботи компонентів у складі всієї системи.

- тестування на стійкість для визначення надійності роботи системи при великих обчислювальних навантаженнях.

Завершення цього етапу означає, що система відповідає заданим вимогам і працює стабільно в різних робочих умовах.

Етап впровадження та налаштування. На етапі впровадження система інтегрується в робоче середовище. Вона налаштовується відповідно до потреб кінцевого користувача і специфіки конкретного підприємства, де буде використовуватися. Для цього проводиться адаптація параметрів детермінованого навчання під реальні дані, а також інтеграція з існуючими системами за необхідності.

Також у відповідності до менеджменту якості, забезпечується навчання користувачів, які будуть працювати із системою, що включає інструкції з використання і коригування параметрів системи в реальних умовах.

Експлуатація та обслуговування. Після впровадження система переходить в режим експлуатації. Протягом цього етапу здійснюється регулярний моніторинг її ефективності і коригування параметрів за потреби.

Система підтримується в актуальному стані, виконується технічне обслуговування, а також оновлення компонентів, щоб забезпечити високу точність і адаптивність.

Основні завдання обслуговування зазвичай включають:

- оцінку ефективності роботи та коректності прийнятих рішень;

- моніторинг адаптивності алгоритмів детермінованого навчання для постійного вдосконалення системи;
- планові оновлення компонентів для збереження оптимальної продуктивності.

Етап модернізації та удосконалення. З часом змінюються як потреби користувачів, так і технологічні вимоги. Тому необхідно проводити періодичну модернізацію системи, розширювати її функціональні можливості та адаптувати до нових умов.

На основі отриманих статистичних даних і відгуків користувачів розробляються нові алгоритми, додаються функції, або модифікуються існуючі, щоб відповідати зростаючим вимогам.

Таким чином, система, що розроблюється проходить повний цикл розробки, що включає всі необхідні етапи для створення, тестування та вдосконалення.

І відповідно весь процес дозволяє забезпечити високу продуктивність і надійність системи у вирішенні складних оптимізаційних задач, що робить її цінним інструментом для сучасного українського підприємства.

## **2.2. Діаграма прецедентів використання**

При розробці якісного програмного продукту діаграма прецедентів (use case diagram) є одним із важливих інструментів. Вона є засобом для візуалізації функціональних вимог системи, представляючи взаємодію зовнішніх користувачів (акторів) із системою через конкретні сценарії або прецеденти використання.

Діаграми прецедентів використовуються для визначення функціональних можливостей системи та відображення очікуваних взаємодій між її учасниками і системою в цілому визначення діаграми прецедентів. Головною метою діаграми прецедентів є відображення ключових дій, які система може виконувати у відповідь на запити користувача або зовнішньої системи. Це дозволяє аналізувати вимоги до функціональності ще на початкових етапах проектування. Водночас

діаграма допомагає визначити ролі та обов'язки кожного користувача, що сприяє кращому розумінню очікувань від системи, особливо в разі великих та складних проєктів, де важливо чітко уявляти всі можливі взаємодії.

У відповідності до наукових артефактів, діаграма прецедентів містить кілька основних елементів:

Актори (Actors) - зовнішні користувачі або системи, які взаємодіють із програмною системою. Актори не обов'язково є людьми, вони також можуть бути іншими системами або компонентами.

Прецеденти використання (Use Cases) - специфічні дії, які виконуються у системі як відповідь на запити акторів. Кожен прецедент описує, як система реагує на певний запит.

Зв'язки (Associations) - лінії, що вказують на взаємодію між акторами та прецедентами. Вони показують, які дії актор може виконувати у системі.

Використання діаграми прецедентів надає розробникам наступні переваги:

- полегшення комунікації. Діаграми прецедентів слугують інтуїтивно зрозумілим інструментом для обговорення функціональних вимог між розробниками, бізнес-аналітиками, замовниками та кінцевими користувачами;

- структуризація функціональних вимог. Вони дозволяють структурувати функціональні вимоги та візуально відобразити всі сценарії використання системи;

- аналіз ролей та обов'язків. Діаграми допомагають визначити ролі акторів і їхні обов'язки, що особливо важливо для розподілу функцій між різними типами користувачів;

- визначення обмежень системи. Оскільки діаграми прецедентів показують взаємодію між зовнішніми акторами та системою, вони допомагають чітко визначити, що саме входить до системи, а що ні [16].

Згідно з дослідженнями, діаграми прецедентів є критичним компонентом методології об'єктно-орієнтованого аналізу і проєктування (ООАП) та широко застосовуються в методології UML (Unified Modeling Language). Як зазначають автори, діаграма прецедентів є одним із перших етапів розробки, що дозволяє

формалізувати вимоги користувачів і надати чітке уявлення про функціональні межі системи. Робота з діаграмами прецедентів полегшує управління складними проектами, оскільки дозволяє ідентифікувати та ізолювати окремі функціональні частини системи на початкових етапах .

Таким чином, діаграми є важливим інструментом для визначення і візуалізації функціональних вимог системи. Вони не лише допомагають формалізувати вимоги користувачів, але й полегшують комунікацію між учасниками проекту та забезпечують структурований підхід до проектування програмного забезпечення.

У нашому випадку діаграма відображає ключові прецеденти використання для трьох ролей: маркетолога/аналітика, користувача/менеджера та адміністратора/розробника. Вона описує, як кожна роль взаємодіє із системою для введення даних, оптимізації, аналізу результатів та забезпечення підтримки й безпеки.

Основні ролі для нашої ІС:

Маркетолог/Аналітик - здійснює аналітичну роботу в системі, вводить дані, налаштовує параметри моделювання, аналізує результати та використовує оптимізаційні функції.

Користувач/Менеджер - відповідає за управління процесами, моніторинг виконання завдань, підтримує безперервну роботу, а також має доступ до результатів аналізу.

Адміністратор/Розробник - виконує технічне обслуговування та налаштування системи, займається розробкою, тестуванням і оновленням функціоналу, контролює безпеку та доступ до системи.

Детальний опис взаємодій для кожної ролі розпишемо детально нижче.

Роль Маркетолог/Аналітик. Взаємодіє із системою для введення необхідних даних, налаштування параметрів моделювання та запуску процесу моделювання.

Аналізує отримані результати через інтерфейс системи, що надає можливість візуалізації результатів у вигляді графіків і звітів. Експортує дані для створення звітів і подальшого оброблення.

Роль Користувач/Менеджер. Керує процесом та здійснює контроль за виконанням завдань, забезпечуючи безперервність роботи системи.

Аналізує показники ефективності системи та переглядає загальний стан продуктивності, вчасно реагуючи на будь-які відхилення для оптимізації процесів.

Роль Адміністратор/Розробник. Налаштовує облікові записи користувачів і забезпечує належний рівень доступу для кожної ролі.

Виконує резервне копіювання, оновлення функціоналу, тестування та підтримку системи, що забезпечує її стабільну роботу й безпеку.

Розробляє нові функції та забезпечує відповідність системи вимогам, зокрема новим алгоритмам для оптимізації на основі детермінованого навчання.

Таким чином, наведена діаграма прецедентів використання описує основні взаємодії між маркетологом, менеджером та адміністратором з інтелектуальною системою, показуючи основні процеси, від налаштування та запуску моделювання до зберігання даних, забезпечення безпеки і підтримки. Така система дозволяє ефективно вирішувати задачі оптимізації, що особливо корисно для аналітичної роботи в сучасних бізнес-процесах. Основні прецеденти наведено в таблиці 2.2.

Таблиця 2.2

### Прецеденти використання ІС з методом Монте-Карло

Прецедент	Опис	Роль
<b>Введення даних для моделювання</b>	Маркетолог вводить вихідні дані, вибирає тип задачі та задає початкові параметри для моделювання	Маркетолог/Аналітик
<b>Налаштування параметрів моделювання</b>	Маркетолог налаштовує такі параметри, як кількість ітерацій, ймовірнісні розподіли для методу Монте-Карло, а також критерії для оптимізації	Маркетолог/Аналітик
<b>Запуск моделювання з методом Монте-Карло</b>	Маркетолог запускає процес моделювання, що генерує сценарії для оптимізації на основі випадкових значень	Маркетолог/Аналітик
<b>Виконання детермінованого навчання</b>	Система проводить детерміноване навчання для підвищення точності прогнозів і результатів на основі попередніх сценаріїв	Маркетолог/Аналітик
<b>Перегляд результатів та аналіз тенденцій</b>	Маркетолог переглядає результати моделювання у вигляді діаграм, звітів і графіків для аналізу тенденцій та обрання	Маркетолог/Аналітик

<b>Прецедент</b>	<b>Опис</b>	<b>Роль</b>
	найкращих рішень	
<b>Збереження результатів та їх експорт</b>	Маркетолог зберігає результати обчислень, а також експортує дані в зовнішні формати (наприклад, CSV, PDF) для подальшого аналізу та звітності	Маркетолог/Аналітик
<b>Управління процесом і контролем виконання</b>	Менеджер контролює загальний процес виконання завдань, підтримує безперервну роботу та здійснює моніторинг показників ефективності процесів у системі	Користувач/Менеджер
<b>Моніторинг продуктивності системи</b>	Менеджер переглядає загальну ефективність роботи системи, аналізує показники продуктивності для подальшої оптимізації процесів.	Користувач/Менеджер
<b>Налаштування облікових записів користувачів</b>	Адміністратор створює та редагує облікові записи для маркетологів, менеджерів та інших користувачів, налаштовує права доступу	Адміністратор/Розробник
<b>Забезпечення безпеки та доступу</b>	Адміністратор контролює доступ до функціоналу системи, забезпечує захист даних та впроваджує політики безпеки	Адміністратор/Розробник
<b>Виконання резервного копіювання даних</b>	Адміністратор здійснює резервне копіювання системних даних для забезпечення надійності та безперервності роботи системи.	Адміністратор/Розробник

Закінчення таблиці 2.2

Прецедент	Опис	Роль
<b>Розробка та оновлення функціоналу системи</b>	Адміністратор займається розробкою нових функцій, вдосконалює наявний функціонал та проводить оновлення системи відповідно до потреб користувачів та нових вимог	Адміністратор/Розробник
<b>Тестування та перевірка системи</b>	Адміністратор проводить тестування, щоб перевірити коректність роботи системи та виявити можливі недоліки перед впровадженням змін	Адміністратор/Розробник
<b>Оновлення та підтримка системи</b>	Адміністратор забезпечує своєчасні оновлення, технічну підтримку, а також проводить виправлення можливих помилок і багів для стабільної роботи системи	Адміністратор/Розробник

Крім того, для розкриття досліджуваної теми нашої роботи опишемо для приклада декілька важливих прецедентів.

Назва прецеденту використання «Оптимізація рекламних кампаній».

Метою прецеденту використання є розробка ефективної стратегії для максимального залучення клієнтів і підвищення продажів при оптимальному бюджеті.

Оптимістичний сценарій:

Маркетолог запитує дані щодо поточних рекламних кампаній у базі даних.

Маркетолог отримує аналітичні показники з попередніх кампаній, зокрема інформацію про рентабельність, залучення аудиторії та ефективність окремих каналів реклами.

Система прогнозує очікуваний результат для запропонованого бюджету, включаючи кількість потенційних клієнтів, які будуть залучені.

Маркетолог коригує налаштування кампанії, збільшуючи бюджети на найбільш ефективні канали.

Система надає рекомендації щодо запуску кампанії з оптимальними параметрами.

Прагматичний сценарій:

Умова 1: Низька ефективність поточної кампанії.

A1: Перерозподілити бюджети на більш результативні рекламні канали.

A2: Застосувати додаткові інструменти таргетингу для охоплення ширшої аудиторії.

Умова 2: Обмежений бюджет на рекламу.

B1: Сконцентрувати ресурси на каналах з найвищою рентабельністю.

B2: Використовувати більш економічні методи реклами, як-от SEO та контент-маркетинг.

B3: Збільшити тривалість кампанії для досягнення поступового ефекту.

Назва прецеденту використання «Аналіз конкурентного середовища».

Метою прецеденту використання є визначення ключових конкурентів, аналіз їхніх стратегій та виявлення можливостей для підвищення конкурентоспроможності компанії.

Оптимістичний сценарій:

Маркетолог запитує дані щодо основних конкурентів у системі, включаючи інформацію про їх ринкові частки, цінову політику, стратегії та рекламні активності.

Система виконує аналіз і порівняння основних показників продукції та послуг компанії з показниками конкурентів.

Маркетолог отримує звіт з рекомендаціями щодо покращення позиціонування компанії, зокрема пропозиції щодо зміни ціни або введення нових акцій.

За наявності можливостей система пропонує варіанти співпраці з партнерами для посилення позицій на ринку.

Прагматичний сценарій:

Умова 1: Велика ринкова частка конкурентів.

A1: Розробити пропозиції для залучення нових клієнтів через поліпшення сервісу або знижки.

A2: Ввести акційні пропозиції, що акцентують унікальні особливості продукції компанії.

Умова 2: Низька популярність бренду порівняно з конкурентами.

V1: Підвищити впізнаваність через збільшення рекламних активностей на популярних платформах.

V2: Поліпшити візуальний стиль і комунікацію бренду для залучення молодшої аудиторії.

V3: Впровадити спільні проєкти з популярними особами, інфлюенсерами чи відомими брендами для підвищення авторитету.

Наведемо діаграму прецедентів у графічному виді для нашого проєкту інтелектуальної системи (рис. 2.2).

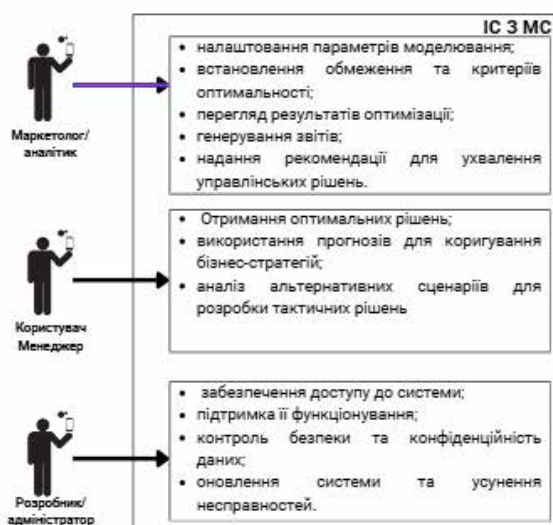


Рис. 2.2. Діаграма прецедентів ІСПР

### 2.3. Діаграма пакетів

Діаграма пакетів є важливим інструментом в об'єктно-орієнтованому моделюванні, що використовується для візуалізації структури програмної системи на рівні її компонентів (пакетів). Діаграма дозволяє організувати класи та інші елементи системи в логічні одиниці, що сприяє кращому управлінню та підтримці програмного коду. Для нашої інтелектуальної системи вирішення оптимізаційних задач на основі методу Монте-Карло з детермінованим навчанням, діаграма пакетів служить для визначення основних компонентів і їх взаємодій у межах системи.

Діаграма пакетів для нашої системи має на меті поділ програмної системи на функціональні блоки, які відповідають за різні етапи обробки даних, моделювання, навчання та оптимізацію. Кожен з цих блоків є логічним пакетом, що містить певні класи або модулі. Нижче описані основні пакети, які можуть бути присутніми в діаграмі пакетів для нашої інтелектуальної системи.

Пакет «Вхідні дані та інтеграція». Цей пакет відповідає за отримання та обробку вхідних даних для системи. Він включає модулі для завантаження даних з різних джерел (бази даних, файли), а також для попередньої обробки та валідації цих даних перед подальшим використанням у процесах моделювання.

Класи: `DataLoader`, `DataValidator`, `InputParser`.

Пакет «Моделювання та оптимізація». Включає основні класи, що реалізують метод Монте-Карло для розв'язання оптимізаційних задач. Цей пакет відповідає за генерування випадкових вибірок, виконання ітераційних обчислень та адаптацію результатів на основі зібраних даних.

Класи: `MonteCarloModel`, `OptimizationAlgorithm`, `SimulationEngine`.

Пакет «Детерміноване навчання». Цей пакет забезпечує реалізацію детермінованого навчання для вдосконалення оптимізації. Він містить класи, які виконують обчислення на основі попередніх результатів і забезпечують точніші прогнози та оптимальні стратегії.

Класи: `DeterministicLearning`, `LearningAlgorithm`, `LearningModel`.

Пакет «Аналіз результатів». Включає класи для збору та обробки результатів після виконання оптимізаційних задач. Цей пакет відповідає за аналіз і візуалізацію результатів, що отримані після застосування моделі Монте-Карло та алгоритмів навчання.

Класи: `ResultsAnalyzer`, `DataVisualizer`, `PerformanceMetrics`.

Пакет «Інтерфейс користувача». Відповідає за взаємодію користувача з системою. Цей пакет містить класи, що відповідають за графічний інтерфейс або інтерфейс командного рядка для введення даних, запуску обчислень та відображення результатів.

Класи: `UserInterface`, `ControlPanel`, `InputForm`.

Пакет «Управління процесами». Цей пакет відповідає за управління процесами в системі, такими як обробка помилок, моніторинг виконання, оптимізація часу виконання задач та інтеграція з іншими системами.

Класи: ProcessManager, ErrorHandler, TaskScheduler.

Пакет «Збереження даних». Включає класи для збереження та експорту результатів моделювання, зокрема в бази даних чи інші формати для подальшого аналізу або використання в інших системах.

Класи: DatabaseManager, DataExporter, FileSaver.

Взаємодія між пакетами. На діаграмі пакетів ці компоненти можуть бути з'єднані лініями, які вказують на залежності між ними. Наприклад:

Пакет «Вхідні дані та інтеграція» взаємодіє з пакетами «Моделювання та оптимізація» і «Аналіз результатів», оскільки дані, що надходять, безпосередньо впливають на моделювання та подальший аналіз.

Пакет «Моделювання та оптимізація» тісно інтегрований з пакетом «Детерміноване навчання», оскільки результат моделювання є вхідними даними для алгоритмів навчання.

Пакет «Аналіз результатів» залежить від результатів, що генеруються в пакетах «Моделювання та оптимізація» і «Детерміноване навчання» для візуалізації та подальшого аналізу.

У скороченому варіанті для реалізації інформаційної системи, нами було запропоновано топологію системи, що складається з таких основних компонентів (рис. 2.3.):

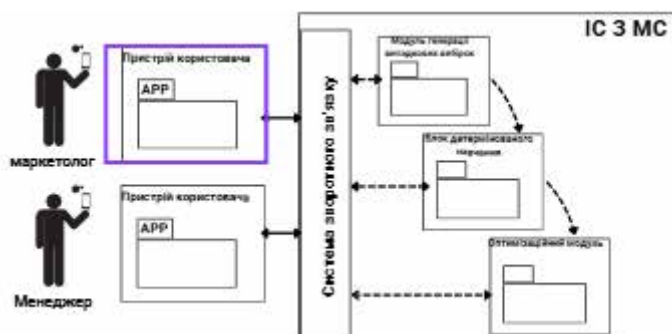


Рис. 2.3. Топологія системи

1. Модуль генерації випадкових вибірок - забезпечує отримання даних для моделювання на основі методу Монте-Карло. Він створює набір варіантів можливих рішень на основі випадкових вибірок, що важливо для дослідження всіх можливих сценаріїв.

2. Блок детермінованого навчання - на основі попереднього аналізу вибірок адаптує параметри моделювання. Це дозволяє покращити точність прогнозів та оптимізувати ресурси, скорочуючи кількість обчислень, необхідних для досягнення прийняттого результату.

3. Оптимізаційний модуль - визначає найкращі можливі рішення, враховуючи обмеження та критерії оптимальності. Цей модуль є основою для прийняття управлінських рішень та вибору оптимальної стратегії діяльності підприємства.

4. Система зворотного зв'язку - здійснює обмін інформацією між усіма модулями системи, дозволяючи інтерактивно коригувати результати та налаштовувати параметри моделювання відповідно до змінних умов.

Запропонована топологія системи забезпечує адаптивність і здатність ефективно функціонувати у реальному часі, а самі компоненти системи взаємодіють між собою, забезпечуючи циклічний процес удосконалення прогнозів та розв'язання оптимізаційних задач.

Таким чином, діаграма пакетів для нашої інтелектуальної системи вирішення оптимізаційних задач відображає основні функціональні блоки системи та їх взаємозв'язки.

Вона дозволяє чітко структурувати код і забезпечує розподіл обов'язків між компонентами, що допомагає підтримувати систему на етапах її розробки та експлуатації.

Використання діаграми пакетів є важливим кроком в організації логіки та забезпеченні ефективної взаємодії між різними елементами системи, що розробляється на основі ТЗ.

## 2.4. Діаграма класів

Діаграма класів є невід'ємним інструментом для моделювання структури програмного забезпечення, оскільки вона дозволяє відобразити класи, методи, атрибути і взаємозв'язки між ними. У нашій системі діаграма класів допомагає організувати та чітко уявити компоненти системи, спрямованої на вирішення оптимізаційних задач на основі методу Монте-Карло з детермінованим навчанням.

Для вирішення питань нашої інтелектуальної системи, розробленої для вирішення оптимізаційних задач, було виділено наступні класи, що представляють основні компоненти системи:

Клас OptimizationTask

Атрибути: taskId, taskDescription, parameters.

Методи: defineTask(), setParameters(), executeTask().

Опис класу. Цей клас є основою для роботи з оптимізаційними задачами, визначаючи опис задачі, вхідні параметри і метод для її запуску. Клас дозволяє створити задачу, налаштувати її параметри і запустити процес оптимізації.

Клас MonteCarloModel

Атрибути: sampleSize, iterations, probabilityDistribution.

Методи: generateSamples(), runSimulation(), calculateResults().

Опис класу. Цей клас реалізує метод Монте-Карло, генеруючи випадкові вибірки і виконуючи симуляції для задачі. Він забезпечує основну функціональність моделювання і працює з випадковими вибірками для оцінки ймовірнісних результатів.

Клас DeterministicLearning

Атрибути: learningRate, threshold, modelType.

Методи: trainModel(), updateModel(), evaluatePerformance().

Опис класу. Цей клас забезпечує алгоритми детермінованого навчання для покращення результатів, отриманих методом Монте-Карло. Він дозволяє

тренувати модель, оновлювати її параметри на основі нових даних і оцінювати ефективність навчання.

Клас `DataManager`

Атрибути: `databaseConnection`, `dataSet`, `query`.

Методи: `loadData()`, `saveResults()`, `queryData()`.

Опис класу. Відповідає за завантаження, зберігання і вибірку даних із бази. Цей клас зберігає результати обчислень і надає доступ до необхідних даних для процесу навчання та моделювання.

Клас `UserInterface`

Атрибути: `userId`, `userRole`, `userPreferences`.

Методи: `displayResults()`, `inputData()`, `configureSettings()`.

Опис класу. Клас забезпечує взаємодію з користувачем (маркетологом чи аналітиком) і надає інтерфейс для введення даних, налаштувань і відображення результатів моделювання.

Клас `AnalysisModule`

Атрибути: `reportFormat`, `analysisCriteria`, `timeFrame`.

Методи: `generateReport()`, `visualizeResults()`, `compareScenarios()`.

Опис класу. Клас аналізу даних використовується для створення звітів і візуалізації результатів моделювання. Він дозволяє маркетологу оцінювати різні сценарії і робити порівняння на основі заданих критеріїв.

Розпишемо взаємозв'язки між названими класами інтелектуальної системи.

Асоціація між `OptimizationTask` і `MonteCarloModel`. Клас `OptimizationTask` використовує клас `MonteCarloModel` для виконання симуляцій на основі заданих параметрів задачі. Цей зв'язок відображає, що метод Монте-Карло є основним підходом до вирішення задачі оптимізації.

Асоціація між `MonteCarloModel` та `DeterministicLearning`. Результати, отримані з `MonteCarloModel`, використовуються в `DeterministicLearning` для покращення моделей і підвищення ефективності процесу. Ця асоціація демонструє взаємозалежність між моделлю Монте-Карло і алгоритмами навчання.

Асоціація між DataManager і DeterministicLearning. DataManager зберігає дані, що використовуються і оновлюються в процесі навчання моделі. Це забезпечує постійний доступ до історичних даних для покращення моделі.

Асоціація між UserInterface і AnalysisModule. Інтерфейс користувача забезпечує взаємодію з маркетологом, або аналітиком для налаштування параметрів аналізу і перегляду звітів, які генеруються в AnalysisModule.

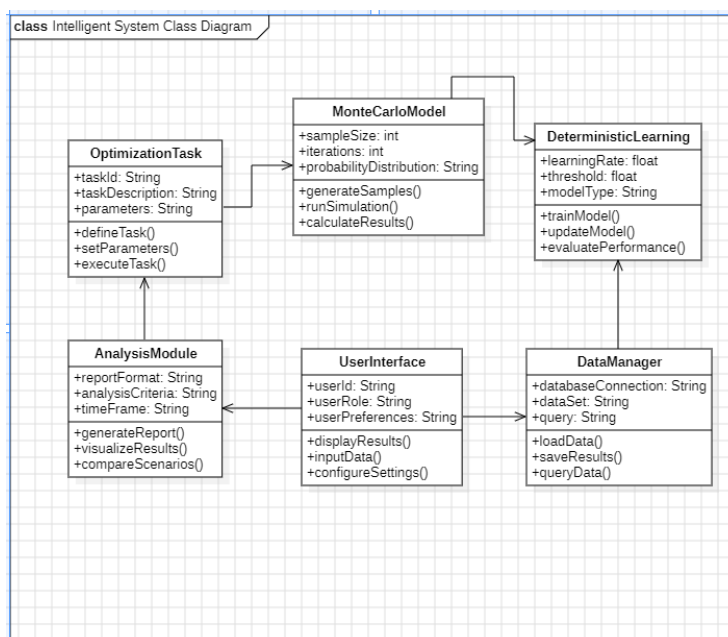


Рис.2.4. Діаграма класів

Підсумуємо вище написане, діаграма класів для нашої інтелектуальної системи, яка вирішує оптимізаційні задачі за допомогою методу Монте-Карло і детермінованого навчання, забезпечує візуалізацію структури системи та зв'язків між її компонентами. Класи та їхні взаємозв'язки відображають функціональність і відповідальність кожного елемента системи, що сприяє полегшенню процесу розробки, тестування та підтримки цієї системи.

## 2.5. Діаграма розгортання

Діаграма розгортання для інтелектуальної системи, яка підтримує прийняття управлінських рішень у сфері маркетингу, відображає основні

обчислювальні вузли, програмні модулі та взаємодію компонентів між собою. Основне завдання діаграми UML - продемонструвати фізичну структуру системи, а також взаємозв'язок між програмними компонентами в процесі роботи.

Головними компоненти діаграми розгортання будуть (рис. 2.5):

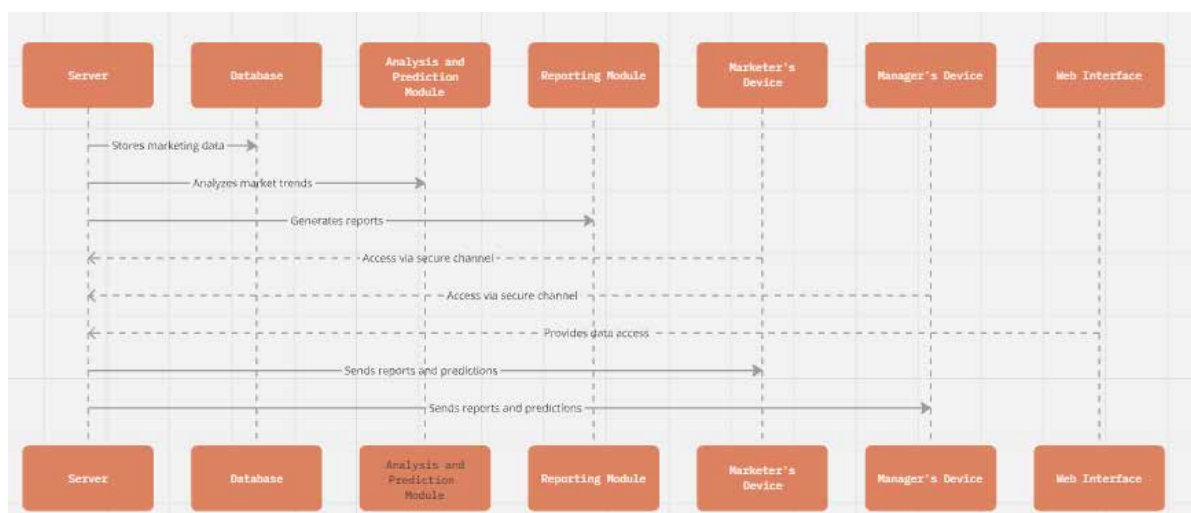


Рис. 2.5. Діаграма розгортання

Server. Основний сервер виступає головним обчислювальним вузлом, на якому розміщені:

- база даних (БД), де зберігаються дані про попередні маркетингові кампанії, показники ринку, цільову аудиторію та аналітичні звіти;

- модуль аналізу та прогнозування для обробки даних, аналізу ринкових тенденцій, оцінки конкурентних показників і прогнозування результатів маркетингових стратегій. Це ключовий компонент, який забезпечує роботу алгоритмів оптимізації та прогнозування на основі методів, таких як Монте-Карло.

- модуль звітності, який генерує індивідуальні звіти для маркетологів та керівників підприємства, дозволяючи оцінювати ефективність ухвалених рішень.

Клієнтські пристрої маркетолога та менеджера. Клієнтські пристрої (комп'ютери, планшети чи смартфони), які використовують маркетологи й керівники для доступу до системи. Вони можуть переглядати прогнози,

аналітичні звіти, проводити оцінку результатів маркетингових рішень та вносити свої дані або завдання.

Веб-інтерфейс для доступу до аналітичних інструментів системи, а також для перегляду візуалізованих даних, для спрощення користувачам доступ до важливої інформації в режимі реального часу.

Захищений зв'язок між клієнтськими пристроями та сервером. Сервер та клієнтські пристрої підключені через захищений канал зв'язку, забезпечуючи безпеку передачі даних, конфіденційність і захист від несанкціонованого доступу.

Така діаграма розгортання дозволяє побачити, як різні елементи системи пов'язані між собою та яким чином дані циркулюють у системі. Це важливо для забезпечення ефективності обміну інформацією між аналітичними та управлінськими модулями й зручного доступу до результатів аналізу для прийняття обґрунтованих маркетингових та управлінських рішень.

### 3. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

#### 3.1. Основні положення моделі інформаційного забезпечення

У розробці інтелектуальної системи підтримки управлінських рішень для маркетологів та керівників підприємств ключовим є інформаційне забезпечення, яке включає необхідні дані для прийняття стратегічних та оперативних рішень.

У цій системі інформаційна база (ІБ) є критичним компонентом. ІБ складається з організованого масиву даних, структурованого таким чином, щоб задовольнити інформаційні потреби користувачів у процесах планування, аналізу ринку та управління бізнес-процесами. Інформаційна база зберігається у вигляді бази даних (БД), яка обробляється системою для підтримки прийняття рішень.

Моделювання даних для ІБ здійснюється з метою створення чіткої концептуальної схеми, яка відобразить потреби управлінських рішень і буде легко адаптована до будь-якої системи управління базами даних (СУБД).

Для побудови такої структури широко використовуються діаграми «сутність-зв'язок» (ERD), що дозволяють представляти ключові елементи предметної області, зокрема маркетингові дані, бізнес-показники та інші сутності. Використовуючи ERD, можна деталізувати всі сутності, необхідні для аналізу ринкових даних, формування планів та прогнозів, а також ефективного управління комерційною діяльністю підприємства.

Саме тому у залежності від завдань і складності системи, вибираються відповідні методи моделювання. В науковій літературі виділяють наступні:

Метод Баркера - цей підхід, що використовується в програмному забезпеченні Oracle Designer, дозволяє створювати детальні ERD-діаграми з використанням зрозумілої нотації. Він особливо корисний у процесах, де важливо забезпечити візуальне уявлення сутностей, пов'язаних із ринковим аналізом та управлінням збутом.

Метод IDEFI - базується на підході Чена і дозволяє створювати реляційні моделі в третій нормальній формі, що зменшує надлишковість даних і підвищує їх

цілісність. Цей метод ефективний у проектах, де важливо забезпечити відповідність структури даних вимогам бізнес-процесів та інтеграції в систему підтримки прийняття рішень.

Метод IDEFIX - це модернізована версія IDEF1, яка враховує простоту у вивченні та автоматизації процесу. Використовується в багатьох сучасних програмних продуктах, таких як ERwin, і дозволяє швидко адаптувати модель даних для систем аналітичної обробки та прогнозування в маркетингу.

Створення логічної моделі даних починається з аналізу всіх сутностей і зв'язків, які потрібні для підтримки маркетингових рішень. Залежно від глибини деталізації, виділяють наступні три рівні логічних моделей:

ERD (Entity Relationship Diagram) - відображає основні сутності, що мають значення для аналізу маркетингових та управлінських даних, та ключові зв'язки між ними. Така діаграма дозволяє зрозуміти загальну структуру системи і сприяє комунікації між розробниками та експертами з маркетингу та управління.

Модель на основі ключів (KB) - деталізованіша модель, яка включає всі сутності з первинними ключами, необхідними для представлення структури маркетингових даних. Вона визначає основні показники, що є важливими для аналізу ринку та формування звітів про ефективність.

Повна атрибутивна модель (FA) - найбільш детальна модель, що охоплює всі сутності, атрибути та зв'язки між ними. Ця модель дозволяє забезпечити дані для всебічного аналізу ринкової ситуації, прогнозування продажів і вироблення рекомендацій для управління підприємством.

Таким чином, можна свтреджувати, що модель інформаційного забезпечення виконує кілька важливих функцій. Вона дозволяє впорядковувати інформаційні потоки, забезпечуючи наочність та цілісність даних, що використовуються для прийняття рішень. Завдяки моделі можливо швидко аналізувати ринкові тенденції, визначати ключові показники, формувати прогнози і забезпечувати точність та надійність інформації для керівників підприємства.

Ця модель підтримує всі етапи управлінських рішень - від визначення вимог до аналізу результатів. Це робить її важливим елементом у процесі оптимізації бізнес-процесів, пов'язаних з управлінням маркетингом та аналітикою.

Для розробки моделі інформаційного забезпечення інтелектуальної системи підтримки прийняття управлінських рішень для маркетологів і керівників підприємства пропонуємо розглянути її структуру, яка складається з кількох основних компонентів: бази даних, сховища даних для аналітики, інтерфейсу для взаємодії користувачів та зв'язків між цими компонентами (табл. 3.1).

Таблиця 3.1

## Логічна модель ІС

Компонент	Опис
<b>1. Структура інформаційної бази</b>	Інформаційна база є центральним елементом моделі інформаційного забезпечення, що містить дані для прийняття рішень. Включає маркетингову інформацію, бізнес-статистику, аналітичні показники, прогнози, звіти тощо
<b>Основні таблиці інформаційної бази:</b>	
Дані про клієнтів	- Ім'я клієнта - Контактна інформація - Історія взаємодії - Дані про лояльність
Дані про продукти/послуги	- Назва продукту/послуги - Категорія - Вартість - Опис - Статистика продажів
Дані про ринок	- Маркетингові показники - Тенденції ринку - Дані про конкурентів - Сегментація ринку
Аналітичні показники	- Прогнозовані обсяги продажів - Дані про успішність маркетингових кампаній - Коефіцієнти ефективності
Звіти та прогнози	- Історичні звіти - Прогнози - Рекомендації для оптимізації бізнес-стратегії
<b>2. Сховище даних для аналітики</b>	Сховище даних дозволяє швидкий доступ до великих обсягів історичних і прогнозних даних, інтегрує інформацію з різних джерел для аналізу та побудови звітів.
<b>Основні характеристики сховища:</b>	- Інтеграція даних з різних джерел - Багатовимірні запити для аналізу ринку та трендів - Інструменти для звітності та багаторівневого аналізу
<b>3. Інтерфейс користувача для маркетологів і</b>	Інтерфейс забезпечує доступ до аналітичних даних, звітів та прогнозів, дозволяє формувати стратегії на основі зібраної

<b>Компонент</b>	<b>Опис</b>
<b>керівників</b>	інформації

Закінчення таблиці 3.1

Компонент	Опис
<b>Основні функції інтерфейсу:</b>	
Маркетолог/аналітик	- Доступ до звітів - Створення та редагування прогнозів - Аналіз маркетингових кампаній
Користувач/менеджер	- Створення звітів - Отримання рекомендацій - Пошук клієнтських та ринкових даних
Адміністратор/розробник	- Управління даними - Контроль прав доступу - Підтримка та оптимізація сховища
<b>4. Зв'язки між компонентами</b>	Забезпечують інтеграцію та доступність даних для користувачів:
Зв'язок між ІБ і сховищем даних	Забезпечує передачу необхідних для аналізу даних зі сховища до інформаційної бази і навпаки
Зв'язок між інтерфейсом та ІБ	Дозволяє користувачам взаємодіяти з базою даних, проводити пошук інформації та виконувати запити
Зв'язок між інтерфейсом та сховищем даних	Підтримує аналітичні функції та прогнозування, забезпечує доступ до історичних і прогнозних даних

Як видно з таблиці, розроблена модель забезпечує маркетологів та керівників підприємства надійними даними для аналітики та прогнозування, сприяє оптимізації прийняття рішень.

### 3.2. Побудова логічної моделі

Для інтелектуальної системи підтримки прийняття управлінських рішень було розроблено логічну модель, яка складається з основних сутностей та їхніх атрибутів, що підтримують процеси збору та обробки даних для аналітики і прийняття рішень. Ключовими елементами системи є клієнтська база, дані про продукцію та ринок, а також звітні та аналітичні показники.

У відповідності до попередніх розділів та відповідно до нашої теми наведемо основні сутності логічної моделі:

Сутність «Клієнт» - містить інформацію про клієнтів системи.

ID\_клієнта (int) - унікальний ідентифікатор клієнта.

Ім'я\_клієнта (varchar, max 50) - ім'я клієнта.

Контакт\_інформація (varchar, max 100) - контактні дані клієнта.

Історія\_взаємодії (text) - дані про попередні взаємодії з клієнтом.

Сутність «Продукт» - містить дані про продукцію, яку використовує компанія.

ID\_продукту (int) - унікальний ідентифікатор продукту.

Назва\_продукту (varchar, max 50) - назва продукту.

Категорія (varchar, max 30) - категорія, до якої належить продукт.

Вартість (numeric) - вартість продукту.

Сутність «Замовлення» - забезпечує інформацію про замовлення, які були зроблені.

ID\_замовлення (int) - ідентифікатор замовлення.

Дата\_замовлення (datetime) - дата оформлення замовлення.

Сума\_замовлення (numeric) - загальна вартість замовлення.

Кількість\_товарів (int) - кількість товарів у замовленні.

Сутність «Маркетингові\_показники» - зберігає дані про результати маркетингових кампаній та аналітику.

ID\_показника (int) - унікальний ідентифікатор маркетингового показника.

Кампанія (varchar, max 50) - назва маркетингової кампанії.

Результат\_кампанії (text) - опис результату кампанії.

ROI (numeric) - рентабельність інвестицій у кампанію.

Сутність «Ринковий\_тренд» - містить інформацію про ринкові тенденції та динаміку.

ID\_тренду (int) - ідентифікатор ринкового тренду.

Назва\_тренду (varchar, max 50) - назва тренду.

Опис (text) - опис сутності тренду та його характеристик.

Дата\_оновлення (datetime) - дата останнього оновлення інформації про тренд.

Сутність «Звіт» - зберігає дані про звіти, сформовані на основі аналітики.

ID\_звіту (int) - унікальний ідентифікатор звіту.

Тип\_звіту (varchar, max 50) - тип звіту (наприклад, щоквартальний, річний).

Зміст (text) - опис даних, які містяться у звіті.

Дата\_створення (datetime) - дата створення звіту.

Сутність «Прогноз» - надає інформацію про прогнози, отримані з аналітики.

ID\_прогнозу (int) - унікальний ідентифікатор прогнозу.

Період (varchar, max 20) - період, на який робиться прогноз.

Прогнозоване\_значення (numeric) - прогнозовані значення продажів або попиту.

Точність\_прогнозу (numeric) - показник точності прогнозу у відсотках.

Сутність «Аналітик» - містить інформацію про аналітиків, які працюють з системою.

ID\_аналітика (int) - ідентифікатор аналітика.

Ім'я\_аналітика (varchar, max 50) - ім'я аналітика.

Спеціалізація (varchar, max 50) - напрямок роботи аналітика.

Сутність «Адміністратор» - містить дані про адміністратора системи, який налаштовує та забезпечує її роботу.

ID\_адміністратора (int) - ідентифікатор адміністратора.

Ім'я\_адміністратора (varchar, max 50) - ім'я адміністратора.

Рівень\_доступу (varchar, max 30) - рівень доступу до системи.

Сутність «Користувач» - містить інформацію про користувачів, які мають доступ до системи.

ID\_користувача (int) - ідентифікатор користувача.

Ім'я\_користувача (varchar, max 50) - ім'я користувача.

Роль (varchar, max 30) - роль користувача в системі (наприклад, маркетолог, керівник).

Визначимо основні зв'язки між сутностями:

Замовлення-Клієнт. Клієнт може зробити одне або більше замовлень, але кожне замовлення пов'язане тільки з одним клієнтом.

Продукт-Замовлення. Замовлення може містити один, або більше продуктів.

Аналітик-Маркетингові\_показники. Аналітики мають доступ до маркетингових показників для їхньої обробки та аналізу.

Звіт-Прогноз: Звіт може включати результати одного, або кількох прогнозів, що були зроблені аналітиками.

Ринковий\_тренд-Маркетингові\_показники. Маркетингові показники можуть бути прив'язані до певного ринкового тренду.

Адміністратор-Користувач. Адміністратор має право керувати правами доступу користувачів до системи.

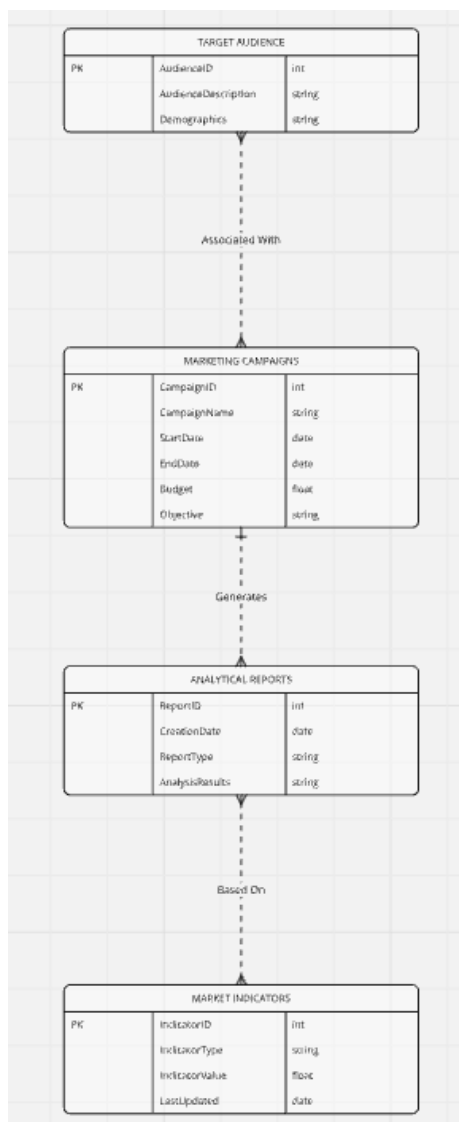


Рис. 3.1. ER-діаграма, яка відображає структуру бази даних ІСППР

Зв'язки між наведеними на рис. 3. Сутностями:

Маркетингові кампанії → Генерують → Аналітичні звіти (І):

Кожна маркетингова кампанія може створювати кілька звітів, але кожен звіт належить до однієї кампанії.

Цільова аудиторія → Використовується у → Маркетингові кампанії (M):

Одна цільова аудиторія може брати участь у кількох кампаніях, і кожна кампанія може мати кілька аудиторій.

Аналітичні звіти → Базуються на → Ринкові показники (N):

Один звіт може використовувати кілька ринкових показників, а один показник може бути застосований у кількох звітах.

Таким чином, розроблена логічна модель дозволяє створити структуру інформаційної бази для інтелектуальної системи підтримки управлінських рішень. Система зберігає ключові показники, звіти, прогнози і взаємодію між аналітиками, маркетологами та керівниками підприємства. Така модель допомагає ефективно зберігати та обробляти необхідні для прийняття рішень дані, забезпечуючи всім учасникам системи доступ до потрібної інформації в зручному форматі.

### **3.3. Основи фізичної моделі даних для оптимізаційної системи**

На сьогоднішній день важко уявити собі комп'ютерну техніку, локальні та глобальні мережі без основної частини прикладного програмного забезпечення - бази даних та систем управління базами даних. З базами даних ми стикаємось всюди.

Останнім часом найбільше розповсюдження отримали реляційні бази даних. В реляційних базах даних інформація зберігається в одній або декількох таблицях. Зв'язок між таблицями виконується за допомогою значень одного або декількох співпадаючих полів. Кожний рядок таблиці в реляційних базах даних унікальна. Для забезпечення унікальності використовуються ключі, які зберігають одне або декілька полів таблиці. Ключі зберігаються в упорядкованому вигляді, що забезпечує прямий доступ до записів таблиці під час пошуку.

Реляційні системи управління базами даних дозволяють зберігати інформацію без дублювань, надлишковостей та неузгодженостей, забезпечують оперативний доступ до даних, створення та експлуатацію засобів для їх швидкого

пошуку, фільтрування та сортування, незалежність структур даних від прикладних програм, коректне використання інформації в мережі. Отже, поширення баз даних та систем управління базами даних - це об'єктивний процес сучасності викликаний науково-технічним прогресом [15].

Для супроводу бази даних у процесі розробки системи було обрано MySQL - сучасну систему управління базами даних (СУБД), яка є однією з найбільш популярних і продуктивних у світі. MySQL підтримується компанією Oracle Corporation і активно використовується в багатьох великих та середніх підприємствах (рис. 3.2).

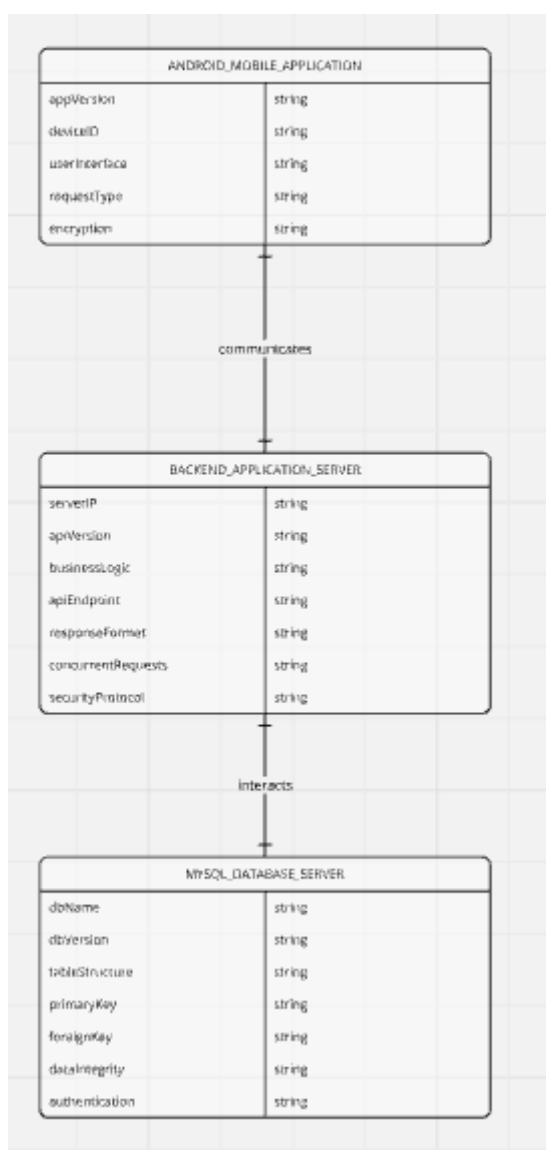


Рис. 3.2. UML-діаграма взаємодії Backend Server із MySQL Database

Наведена діаграма взаємодії Backend Server із MySQL Database допомагає зрозуміти, як і де зберігаються та обробляються дані.

MySQL є оптимальним вибором для систем, які потребують надійної клієнт-серверної архітектури. Вона повністю відповідає стандартам управління реляційними базами даних. СУБД забезпечує цілісність даних за рахунок підтримки первинних і зовнішніх ключів, запобігаючи некоректним операціям з видалення або оновлення записів. Крім того, MySQL дозволяє гнучко керувати даними завдяки потужному інструментарію перевірки допустимості значень та визначення типів даних.

Бази даних у MySQL підтримують широкий спектр типів полів, включаючи текстові, числові, логічні, грошові, дату/час та інші. Це дозволяє розробникам адаптувати структуру бази даних під будь-які потреби системи. У разі відсутності значень у певних полях система підтримує концепцію NULL, що сприяє збереженню цілісності та консистентності даних.

MySQL також підтримує реляційну обробку даних, що дозволяє ефективно вирішувати завдання будь-якої складності. Завдяки своїй гнучкості MySQL може використовуватися як самостійна СУБД або як частина більш складної системи у взаємодії з іншими інструментами, наприклад, з PostgreSQL чи Oracle Database. MySQL відома своєю стабільністю, масштабованістю та продуктивністю, що робить її популярним вибором для різних типів проєктів.

На відміну від застарілих архітектур, орієнтованих на файл-сервери, MySQL реалізує повноцінну клієнт-серверну модель. Це дозволяє не лише зберігати дані на виділених серверах, але й забезпечує інтелектуальне управління запитами, розподіл навантаження та високий рівень безпеки. Використання MySQL у складі сучасних інформаційних систем забезпечує не лише надійне зберігання даних, але й ефективну роботу в локальних і хмарних інфраструктурах.

Переваги використання MySQL для побудови інформаційних систем включають:

- високу продуктивність завдяки оптимізованим механізмам обробки запитів;

- надійність та стабільність у роботі з великими обсягами даних;
- гнучкість налаштування під конкретні потреби підприємства;
- безкоштовну ліцензію на використання з відкритим вихідним кодом, що робить її доступною для широкого спектра проєктів;
- потужну спільноту розробників, яка забезпечує підтримку та розвиток інструментів для роботи з MySQL.

Саме тому, вибір MySQL як основної СУБД для нашої системи є обґрунтованим рішенням, що відповідає вимогам надійності, ефективності та безпеки сучасних інформаційних систем.

### 3.3. Реалізація фізичної моделі

Для розробки фізичної моделі бази даних, що стане основою для мобільного додатку нашої інтелектуальної системи підтримки управлінських рішень, обираємо використання бази даних MySQL через її оптимальну продуктивність, підтримку багатокористувацького доступу, легку інтеграцію з мобільними платформами та високу швидкодію.

На фізичному рівні база даних буде забезпечувати зберігання та управління інформацією, необхідною для ефективного функціонування мобільного додатку. Модель даних на цьому рівні описує фізичне розміщення даних, таблиці, типи даних, індексацію та зв'язки, що забезпечують оптимальну структуру для виконання запитів у базі (табл. 3.2).

Зазначимо основні особливості фізичної моделі бази даних:

1. Індексція ключових полів. Використання первинних ключів для прискорення пошуку даних та забезпечення унікальності записів. Зовнішні ключі в таблицях допомагають зберігати цілісність даних, забезпечуючи правильність зв'язків.

2. Типи даних. Для кожного атрибута визначено відповідний тип даних - текстові дані, числові поля для фінансових показників, типи для збереження дат і часу, що допомагає оптимізувати роботу бази даних і економити пам'ять.

Таблиця 3.2

## Елементи моделі даних фізичного рівня ІС управлінських рішень

Таблиця	Опис	Атрибути
<b>Users</b>	Зберігає інформацію про користувачів (маркетологів, менеджерів, розробників)	ID (INT, PK), Name (VARCHAR), Role (VARCHAR), Email (VARCHAR), Password (VARCHAR), Created_at (DATETIME)
<b>ClientData</b>	Містить дані про клієнтів для аналітичного використання	Client_ID (INT, PK), Client_Name (VARCHAR), Contact_Info (VARCHAR), Loyalty_Score (FLOAT), Last_Interaction (DATETIME)
<b>Products</b>	Дані про продукти, необхідні для аналізу продуктивності і планування стратегій	Product_ID (INT, PK), Product_Name (VARCHAR), Category (VARCHAR), Price (DECIMAL), Sales_Stats (TEXT)
<b>MarketTrends</b>	Інформація про поточні ринкові тенденції та аналітичні показники	Trend_ID (INT, PK), Trend_Description (TEXT), Market_Segment (VARCHAR), Competitor_Analysis (TEXT)
<b>AnalyticsData</b>	Зберігає аналітичні результати для прогнозування і оцінки кампаній	Analysis_ID (INT, PK), Forecasted_Sales (DECIMAL), Campaign_Performance (DECIMAL), ROI (DECIMAL)
<b>Reports</b>	Історія звітів та прогнозів, наданих користувачам для ухвалення рішень	Report_ID (INT, PK), Created_By (INT, FK), Created_At (DATETIME), Report_Content (TEXT)
<b>DataStorage</b>	Сховище, що інтегрує дані з різних джерел для багатовимірного аналізу ринку та трендів	Data_ID (INT, PK), Source (VARCHAR), Data_Timestamp (DATETIME), Data_Value (TEXT)
<b>UserActivityLog</b>	Лог дій користувачів для відстеження взаємодії з системою	Log_ID (INT, PK), User_ID (INT, FK), Action_Timestamp (DATETIME), Action_Description (TEXT)

3. Інтеграція даних з різних джерел. Таблиця **DataStorage** інтегрує зовнішні джерела даних, дозволяючи легко проводити багатовимірний аналіз.

4. Безпека даних. База даних містить зашифровані поля для чутливої інформації, наприклад, для паролів користувачів.

5. Резервне копіювання. MySQL забезпечує можливість налаштування резервного копіювання для захисту даних від втрати.

Наведена фізична модель дозволяє ефективно реалізувати наш додаток, забезпечуючи збереження, швидкий доступ та обробку великих обсягів даних для маркетологів, менеджерів та аналітиків.

### 3.4. Проектування сховища даних

Для розкриття завдань нашої роботи, розглянемо проектування сховища даних - це процес створення спеціалізованої бази для зберігання, обробки та аналізу великих обсягів даних, зібраних з різних джерел, що підтримує аналітичні запити, звітність та прогнозування.

У нашій системі інтелектуальної підтримки прийняття управлінських рішень сховище даних стане ключовим елементом для надання глибокої аналітики маркетологам, менеджерам та керівникам.

Основні принципи проектування сховища даних ІС.

1. Централізація даних. Сховище об'єднує дані з різних джерел, таких як бази даних клієнтів, статистики продажів, дані ринкових досліджень, результати маркетингових кампаній тощо. Що надає можливість здійснювати єдиний, централізований доступ до всіх важливих показників та аналітичних звітів.

2. Багатовимірність даних. Сховище підтримує багатовимірну модель даних (модель «зірка», або «сніжинка»), що дозволяє зберігати дані за різними вимірами, наприклад, час, місце, продукт, клієнт. Що забезпечує швидкий доступ до будь-яких зрізів даних для глибокого аналізу.

3. Історичність даних. Сховище дозволяє зберігати історичні дані, що дає змогу порівнювати показники за різні періоди, відслідковувати тренди, оцінювати вплив маркетингових кампаній і прогнозувати майбутні результати.

4. Інтеграція даних. Для забезпечення консистентності дані обробляються перед внесенням у сховище (етапи ETL - Extract, Transform, Load). На цьому етапі проводиться очищення даних, їх трансформація у потрібний формат і завантаження у відповідні таблиці сховища.

Сховище складається з фактів і вимірів, що дозволяють аналізувати дані на різних рівнях деталізації (табл. 3.3).

Таблиця 3.3

## Основні елементи сховища даних ІС

Компонент	Опис
Факт продажів	Зберігає інформацію про продажі, включаючи дату продажу, суму, кількість одиниць проданої продукції, клієнтську категорію, рекламну кампанію
Факт відвідувань сайту	Інформація про відвідування сайту компанії, включаючи дату, кількість переглядів сторінок, джерела трафіку
Факт маркетингових кампаній	Зберігає дані про маркетингові кампанії, а саме: тривалість кампанії, охоплення, інвестиції, результативність
Вимір часу	Включає дані про день, місяць, квартал, рік. Використовується для відслідковування динаміки показників у часі
Вимір продукту	Інформація про продукт, включаючи категорію, назву, ціну
Вимір клієнтів	Містить деталі про клієнтів, такі як: демографія, категорії лояльності, географічне розташування
Вимір ринку	Включає сегментацію ринку, конкурентні позиції, тренди

Також слід виділити основні процеси сховища даних:

1. Завантаження даних. Здійснюється періодичне завантаження (ETL-процес) нових даних, зокрема, з продажів, аналітичних звітів, досліджень ринку тощо.
2. Оновлення даних. Інтеграція оновлених показників для підтримки релевантності аналітики та коректних прогнозів.
3. Звіти та аналітика. Формуються регулярні звіти для маркетологів та менеджерів, включаючи порівняльні аналізи, оцінки ефективності маркетингових кампаній, прогнозування обсягів продажів та аналіз ринкових тенденцій.

Для фізичної реалізації сховища даних розгортається на сервері бази даних MySQL для забезпечення швидкої обробки аналітичних запитів і можливості інтеграції з мобільним додатком та іншими інструментами для бізнес-аналітики.

Код для створення необхідної бази даних для нашої ІС представлено нижче.

```
-- Створення бази даних
CREATE DATABASE ManagementDecisionSupportSystem;
USE ManagementDecisionSupportSystem;
-- Таблиця для зберігання даних про клієнтів
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    CustomerName VARCHAR(100),
```

```

    ContactInfo VARCHAR(150),
    LoyaltyData VARCHAR(100),
    InteractionHistory TEXT
);
-- Таблиця для зберігання даних про продукти/послуги
CREATE TABLE Products (
    ProductID INT PRIMARY KEY AUTO_INCREMENT,
    ProductName VARCHAR(100),
    Category VARCHAR(50),
    Price DECIMAL(10, 2),
    Description TEXT,
    SalesStatistics JSON
);
-- Таблиця для зберігання даних про ринок
CREATE TABLE MarketData (
    MarketID INT PRIMARY KEY AUTO_INCREMENT,
    MarketingIndicator DECIMAL(10, 2),
    MarketTrends TEXT,
    CompetitorData TEXT,
    MarketSegmentation TEXT
);
-- Таблиця для зберігання аналітичних показників
CREATE TABLE Analytics (
    AnalyticsID INT PRIMARY KEY AUTO_INCREMENT,
    PredictedSalesVolume DECIMAL(10, 2),
    CampaignEffectiveness DECIMAL(5, 2),
    EfficiencyRatio DECIMAL(5, 2),
    CampaignID INT,
    FOREIGN KEY (CampaignID) REFERENCES MarketingCampaigns(CampaignID)
);
-- Таблиця для зберігання маркетингових кампаній
CREATE TABLE MarketingCampaigns (
    CampaignID INT PRIMARY KEY AUTO_INCREMENT,
    CampaignName VARCHAR(100),
    StartDate DATE,
    EndDate DATE,
    TargetAudience TEXT,
    Budget DECIMAL(15, 2),
    CampaignResults TEXT
);
-- Таблиця для зберігання продажів
CREATE TABLE Sales (
    SaleID INT PRIMARY KEY AUTO_INCREMENT,
    ProductID INT,
    CustomerID INT,
    SaleDate DATE,
    Quantity INT,
    SaleAmount DECIMAL(10, 2),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

```

```
-- Таблиця для зв'язку ринкових даних та клієнтів
CREATE TABLE MarketCustomerSegment (
  SegmentID INT PRIMARY KEY AUTO_INCREMENT,
  MarketID INT,
  CustomerID INT,
  FOREIGN KEY (MarketID) REFERENCES MarketData(MarketID),
  FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
-- Таблиця для прогнозів та звітів
CREATE TABLE Reports (
  ReportID INT PRIMARY KEY AUTO_INCREMENT,
  ReportType VARCHAR(50),
  Content TEXT,
  CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Проектування такого сховища дозволить забезпечити маркетологів та керівників усіма необхідними даними, що допоможе приймати обґрунтовані управлінські рішення, відслідковувати успішність стратегій та коригувати їх для досягнення поставлених бізнес-цілей.

## 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

### 4.1. Вибір, обґрунтування інструментарію та спеціальні програмні засоби для розробки програмного забезпечення

#### 4.1.1. Мова програмування Java

У контексті нашої дипломної роботи важливо визначитися з операційною системою, під яку будемо розробляти мобільний додаток для підтримки прийняття управлінських рішень маркетологами та керівниками підприємств.

На сьогоднішній день мобільні додатки можна розробляти, використовуючи нативні мови програмування. Для платформи iOS застосовується Swift (раніше використовувались Objective-C та C++), а для Android — Java або Kotlin. Якщо ж потрібен додаток, який буде працювати одночасно на iOS та Android, виникає дилема, чи створювати два окремих додатки, чи обрати кросплатформений підхід.

Розробка окремих додатків для кожної платформи зазвичай вимагає більше часу та ресурсів. У зв'язку з цим, для компаній, які прагнуть залучити широкую аудиторію, кросплатформена розробка може стати оптимальним рішенням. Цей підхід дозволяє створювати додатки для кількох операційних систем, використовуючи одну базу коду, що значно зменшує витрати та скорочує час на розробку. Згідно зі статистикою інформаційного порталу StatCounter за 2023-2024 роки, приблизно 44,69% пристроїв у всьому світі працюють на операційній системі Android, тоді як 18,47% - на iOS (рис. 4.1).

Ці дані свідчать про те, що пріоритетом має бути розробка мобільного додатку саме для Android, оскільки ця платформа має більшу частку ринку. Однак остаточне рішення залежить від специфіки проекту, навичок розробників та вимог замовника. Якщо замовник наполягає на наявності додатку, або розглянути кросплатформені рішення.

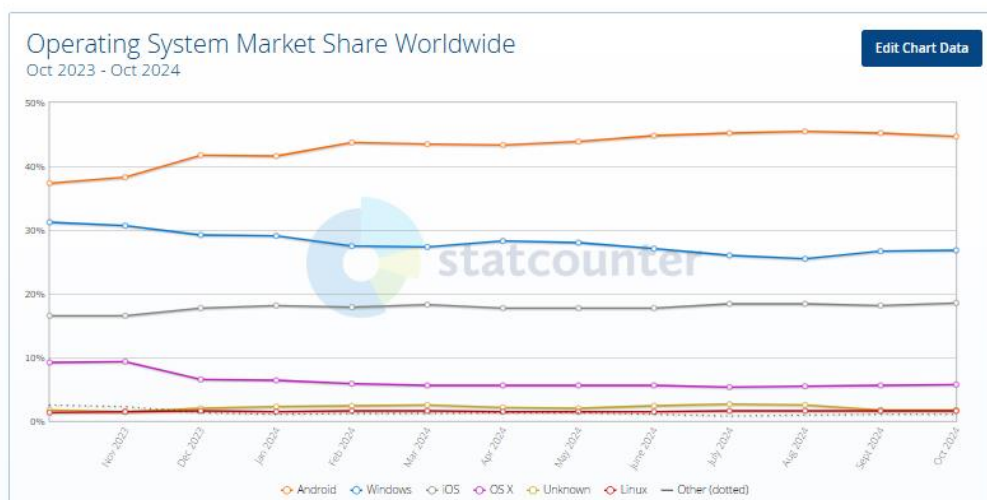


Рис. 1.10. Використання операційних систем у світі у 2023-2024 рр.

Тенденції України мало чим відрізняється від глобального світу в плані використання мобільних операційних систем. Так за даними 2023-2024 роки, Android залишається найпоширенішою мобільною операційною системою в країні (рис. 4.2).



Рис. 4.2. Популярність операційних систем в Україні за 2022-2023 рр.

Отже, орієнтація на Android є виправданою з точки зору охоплення максимальної аудиторії користувачів у нашій країні.

При виборі версії Android для розробки додатку варто врахувати актуальні тенденції. За світовою статистикою, найбільша частка пристроїв працює на Android 14 (рис. 4.3). Щодо вибору мови програмування, для Android доступні «Java» та «Kotlin». На сьогодні Java залишається однією з найпопулярніших мов програмування. Вона була вперше представлена в 1996 році компанією Sun Microsystems, яка згодом стала частиною Oracle.



Рис. 4.3. Світова статистика використання версій ОС Android

Java - це універсальна мова, що використовується для розробки різноманітних програмних продуктів: від десктопних застосунків до масштабних вебсервісів. Її ключовими перевагами є:

- портативність, а саме можливість запуску на різних пристроях і платформах;

- масштабованість, тобто здатність обробляти великі обсяги даних та підтримувати велику кількість користувачів;
- широка спільнота та підтримка, велика кількість бібліотек і фреймворків, що спрощують розробку.

Оскільки більшість додатків для Android написані саме на Java, її використання є логічним вибором для нашого проекту. Це забезпечить сумісність з платформою та дозволить скористатися напрацюваннями спільноти розробників.

Таким чином, вибір операційної системи «Android» та мови програмування «Java» для розробки нашого мобільного додатку є оптимальним рішенням, яке відповідає як ринковим тенденціям, так і технічним вимогам проекту. Це дозволить створити ефективний інструмент для підтримки управлінських рішень, який буде доступним для широкого кола користувачів.

Виходячи з потреб нашого проекту з розробки ІС, було обрано такі інструменти та технології: мова програмування Java, система управління базами даних MySQL, інтегроване середовище розробки (IDE) Android Studio (рис. 4.4.), або Android Studio, а також інші спеціалізовані бібліотеки та фреймворки.

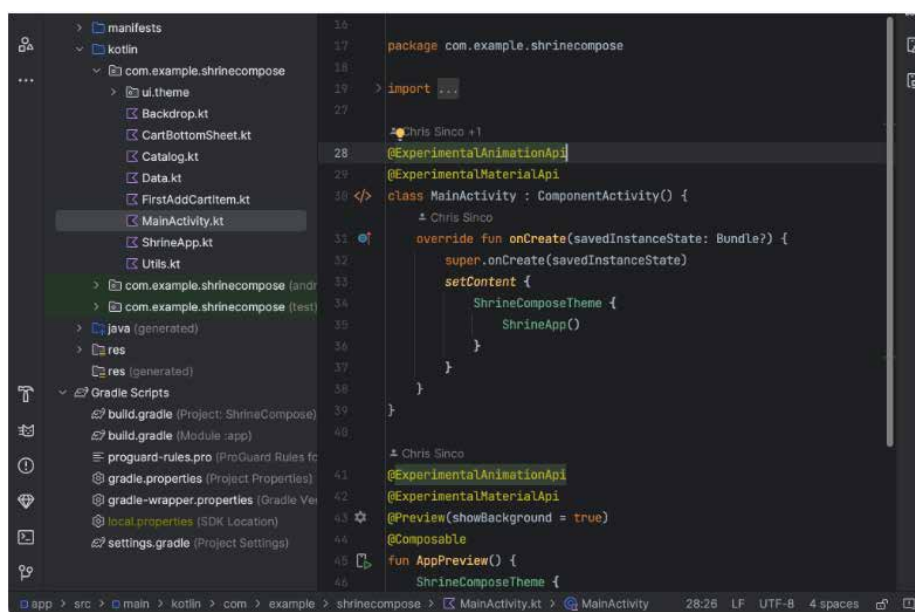


Рис. 4.4 Інтерфейс Android Studio

Мова програмування Java - це об'єктно-орієнтована мова програмування, що є ідеальною для розробки складних програмних рішень завдяки своїй надійності, портативності та багатопотоковості.

Для обґрунтування вибору Java як основної мови програмування для нашої системи наведемо її переваги:

**Портативність і кросплатформеність.** Java забезпечує розробку застосунків, що можуть виконуватись на різних платформах без додаткової адаптації, завдяки принципу «Write Once, Run Anywhere» (Напиши один раз, запускай будь-де). Що дозволяє нашій системі працювати як на комп'ютерах, так і на мобільних пристроях.

**Висока продуктивність.** Java має ефективний механізм управління пам'яттю, що особливо важливо для роботи з великими обсягами даних і аналітичних обчислень, які є основою системи підтримки прийняття рішень.

**Багатий набір бібліотек і фреймворків.** Java має потужну екосистему бібліотек, включаючи спеціалізовані інструменти для роботи з базами даних, обробки великих даних (Big Data), машинного навчання та інтеграції з іншими сервісами, що значно прискорює розробку.

**Підтримка багато поточності.** У Java легко реалізувати багатозадачність і паралельні обчислення, що є важливим для нашої системи, яка повинна обробляти велику кількість запитів та обчислювальних операцій одночасно.

**Безпека.** Java володіє вбудованими засобами безпеки, що забезпечують захист даних і конфіденційності інформації - критично важливий аспект для корпоративних і управлінських застосувань.

**Масштабованість.** Java дозволяє створювати системи, які легко розширюються та інтегруються з іншими модулями або сервісами, що є важливим для довгострокового розвитку та модернізації.

Java буде використовуватись як основна мова для створення ядра системи, що забезпечить її швидкість, надійність і здатність працювати з великими масивами даних, підтримуючи функціональні вимоги проекту.

### 4.1.2. Середовище розробки Android Studio

Для створення мобільного додатку, який підтримує систему управлінських рішень, було обрано Android Studio як основну платформу для розробки. Це середовище забезпечує широкий набір інструментів та можливостей, які оптимально відповідають нашим потребам, зокрема:

Інтеграція з SDK Android та сервісами Google. Android Studio надає всі необхідні інструменти для розробки додатків для Android, включаючи SDK, емулятори, засоби налагодження та тестування. Така інтеграція дозволяє зручно налаштовувати додаток, адаптуючи його до специфіки Android-платформи.

Можливості налагодження і тестування. Android Studio пропонує емулятори для різних моделей пристроїв та варіантів роздільної здатності екрана, що дає можливість виявити потенційні проблеми в додатку до його виходу на ринок. Також є інструменти для моніторингу продуктивності та оптимізації споживання ресурсів.

Підтримка декількох мов програмування. Android Studio працює з Java та Kotlin, що дає гнучкість у виборі та дозволяє використовувати обидві мови для розвитку функціональних можливостей. Це важливо для інтеграції з існуючими бібліотеками або при необхідності застосування сучасних підходів.

Інтеграція з Firebase та іншими сервісами. Завдяки Firebase від Google, Android Studio спрощує інтеграцію аналітики, зберігання даних, автентифікації та інших функцій. Це дозволяє швидко додавати до додатку функціональність, яка розширює можливості збору даних та управління ними.

Інструменти для роботи з базами даних. Android Studio має вбудовані засоби для роботи з локальними базами даних (як-от SQLite), а також для інтеграції з віддаленими базами через API, що необхідно для управління даними в нашій системі.

Регулярна підтримка від Google. Як офіційне середовище для Android, Android Studio постійно оновлюється, підтримуючи останні версії Android і

додаючи нові функції. Це робить середовище стабільним вибором для розробників, які планують довготривалий розвиток проєкту.

Таким чином, ми вважаємо, що Android Studio повністю відповідає вимогам нашої системи, дозволяючи створювати додаток, що об'єднує зручність користування, ефективність роботи з даними та можливості для гнучкої інтеграції з аналітичними сервісами.

#### **4.1.3. Спеціальні бібліотеки для реалізації методу Монте-Карло**

Для реалізації методу Монте-Карло на платформі Android оптимальним є використання спеціалізованих бібліотек, адаптованих до мобільного середовища. Нижче наведено основні бібліотеки, які можна ефективно інтегрувати в Android Studio для виконання обчислень Монте-Карло.

Бібліотека Kotlin Random підходить для генерації випадкових чисел і підтримує численні випадкові розподіли. Вона є частиною стандартної бібліотеки Kotlin, що робить її надійною та ефективною для базових обчислень Монте-Карло в Android-додатках. Це особливо корисно для симуляцій і моделювання, де випадковість є ключовим елементом.

RxJava забезпечує ефективну обробку асинхронних обчислень, що дозволяє запускати інтенсивні обчислення в фоновому режимі без перевантаження основного потоку. Це важливо для методу Монте-Карло, який вимагає великої кількості обчислень. RxJava також дозволяє легко управляти паралельними потоками, підвищуючи швидкість і оптимізуючи обробку даних у мобільному додатку.

Для візуалізації результатів обчислень Монте-Карло MPAndroidChart надає широкий вибір діаграм і графіків. Ця бібліотека дозволяє створювати інформативні візуальні представлення, що допомагає користувачам краще зрозуміти результати аналітики. Вона легко інтегрується з Android Studio та надає зручний API для налаштування.

Бібліотека Kotlin Statistics забезпечує базові статистичні методи, які можуть стати в пригоді при реалізації алгоритму Монте-Карло, наприклад, для розрахунку середнього значення або дисперсії. Це легка бібліотека, яка добре працює з Kotlin і є зручною для розробки статистичних моделей і обробки результатів симуляцій.

Якщо є потреба в більш просунутих математичних обчисленнях, зокрема з використанням NumPy, можна використовувати Chaquopy для інтеграції Python в Android-додаток. Це дозволяє запускати код Python і використовувати всі можливості бібліотеки NumPy для чисельних обчислень, що може бути корисним для більш складних моделей Монте-Карло.

Таким чином, ці бібліотеки забезпечують необхідну функціональність для реалізації методу Монте-Карло, дозволяючи обробляти великі обсяги даних, працювати з паралельними потоками, а також забезпечувати зручну візуалізацію.

#### **4.1.4. Розробка інтерфейсу користувача**

У сучасних умовах ефективний програмний продукт для підтримки управлінських рішень маркетологів і керівників має відповідати високим стандартам зручності використання. Навіть найфункціональніша програма втратить свою актуальність, якщо її інтерфейс складний або незручний для взаємодії. Тому ключове значення має дизайн інтерфейсу, який повинен бути зрозумілим, естетичним і зручним.

Інтерфейс користувача - це міст між користувачем і системою. Для нашої системи він відіграє вирішальну роль у забезпеченні ефективної взаємодії з аналітичними даними та підтримці процесу прийняття рішень. Важливо, щоб маркетологи, менеджери та керівники могли швидко орієнтуватися в системі, легко знаходити потрібну інформацію та отримувати релевантні дані.

Для розкриття цього питання необхідно зазначити основні принципи розробки інтерфейсу:

Розуміння цільової аудиторії. Для створення ефективного інтерфейсу важливо знати, хто є кінцевим користувачем. Наприклад, маркетолог може потребувати доступу до прогнозів і звітів, тоді як керівник — до стратегічних рекомендацій. Розробка «персонажів» із визначенням їхніх потреб і звичок допоможе моделювати поведінку користувача.

Ітеративний підхід. Початковий функціонал системи має бути мінімальним і містити лише найважливіші функції. Подальше розширення можливостей дозволить адаптувати програму до реальних потреб користувачів без перевантаження інтерфейсу.

Дотримання принципів UI/UX-дизайну. Використання інтуїтивно зрозумілих елементів: іконок, кнопок, меню. Оптимальне розташування елементів для швидкого доступу. Зворотний зв'язок. Користувач повинен знати, що його дії виконано. Підтримка адаптивного дизайну для різних пристроїв.

Інтерфейс нашої системи буде орієнтований на три основні ролі: маркетолог/аналітик, менеджер та адміністратор/розробник. Для кожної ролі передбачено відповідні набори функцій та екранів.

Основні екрани ІС:

Головний екран. Забезпечує доступ до основних функцій системи: перегляд аналітики, створення прогнозів, управління користувачами.

Аналітичні звіти. Інтерактивна панель, яка дозволяє маркетологам аналізувати дані про ефективність маркетингових кампаній, ринкові тенденції та динаміку продажів.

Модуль прогнозування. Інтерфейс для роботи з алгоритмами прогнозування, включаючи використання методу Монте-Карло.

Сегментація клієнтів. Візуалізація та аналіз клієнтських груп для створення персоналізованих маркетингових кампаній.

Налаштування користувачів. Дозволяє адміністратору керувати правами доступу та ролями користувачів.

Додаткові модулі. Налаштування звітів і графіків. Фільтри для швидкого пошуку необхідної інформації. Управління інтеграцією з іншими системами.

Основні вимоги до інтерфейсу:

**Інтуїтивність.** Користувачі повинні легко розуміти, як виконувати основні операції.

**Швидкодія.** Усі дії мають виконуватися швидко, без тривалих завантажень.

**Адаптивність.** Інтерфейс повинен коректно працювати як на мобільних пристроях, так і на десктопах.

**Візуальна привабливість.** Дизайн має бути сучасним, із використанням корпоративних кольорів та стилю.

Для проектування інтерфейсу рекомендовано використовувати Figma, оскільки вона дозволяє створювати інтерактивні прототипи та зручно співпрацювати в команді. На етапі розробки дизайн інтегрується в Android Studio, що забезпечує реалізацію адаптивного інтерфейсу з використанням бібліотек Material Design (рис. 4.5).

Переваги розробленого інтерфейсу полягають у:

- забезпечення швидкого доступу до всіх функцій системи;
- можливість персоналізації інтерфейсу для кожного користувача;
- естетичність і зручність використання навіть для нових користувачів;
- підтримка адаптивного дизайну для різних пристроїв.

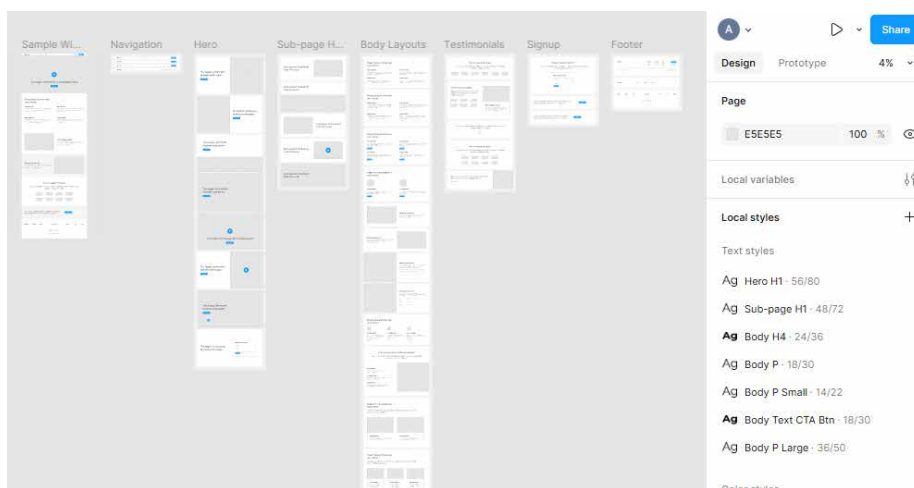


Рис. 4.5. Розробка інтерфейсу користувача для ІС за допомогою Figma

Завдяки правильно спроектованому інтерфейсу, наша система буде зрозумілою, продуктивною та корисною для всіх її користувачів.

## 4.2. ОПИС РОБОТИ ІС

### 4.2.1. Інсталяція програми на пристрій

Етапи встановлення мобільного додатку нашої інтелектуальної системи на пристрій залежать від типу пристрою та способу завантаження: через офіційний магазин додатків (Google Play) або шляхом локальної інсталяції адміністратором підприємства. Нижче наведено загальну інструкцію, яка підходить для більшості пристроїв на операційній системі Android.

1. Пошук додатку в Google Play. Користувач повинен відкрити магазин додатків на своєму мобільному пристрої. Для цього потрібно знайти піктограму Google Play (рис. 4.6).



Рис. 4.6. Вигляд піктограми Google Play для входу в магазин додатків

Після цього слід натиснути на піктограму, щоб відкрити магазин.

2. Використання пошуку. У магазині додатків скористайтеся рядком пошуку, щоб знайти наш мобільний додаток. Для цього введіть назву програми, наприклад, «Інтелектуальна система для маркетологів». Також можна скористатися категоріями.

3. Огляд та вибір додатку. Знайшовши додаток у списку, потрібно натиснути на його назву. Це відкриє сторінку програми, де буде доступна інформація про функціонал, огляди, знімки екрана та додаткові деталі.

4. Встановлення додатку. На сторінці програми натисніть кнопку «Встановити». Програма почне завантажуватися на пристрій.

5. Очікування завершення встановлення. Процес завантаження та встановлення може тривати від кількох секунд до кількох хвилин, залежно від розміру програми та швидкості інтернету. Після завершення інсталяції з'явиться повідомлення про успішне встановлення.

6. Запуск програми. Після інсталяції піктограма додатку з'явиться на екрані пристрою. Для запуску програми потрібно натиснути на її піктограму.



Рис. 4.2 Піктограма мобільного додатку ІС

Після запуску програми користувач зможе увійти в систему, використовуючи свої облікові дані, і почати роботу з аналітичними даними, прогнозами та звітами.

Також наведемо альтернативний спосіб встановлення, а саме локальна інсталяція додатку на мобільний пристрій маркетолога.

Для підприємств, де встановлення виконує системний адміністратор, додаток може бути розповсюджений як APK-файл. У цьому випадку інсталяція виконується наступним чином:

Отримайте APK-файл програми від адміністратора.

Увімкніть можливість встановлення додатків із невідомих джерел у налаштуваннях пристрою.

Відкрийте APK-файл через файловий менеджер.

Натисніть "Встановити".

Після завершення встановлення піктограма програми з'явиться на екрані.

#### Універсальність інструкції

Інструкція підходить для різних пристроїв та користувачів. Проте варто пам'ятати, що особливості процесу можуть змінюватися залежно від версії операційної системи або конфігурації пристрою. У таких випадках рекомендується звернутися до посібника користувача пристрою або онлайн-довідників для уточнення деталей.

Дотримуючись цих рекомендацій, користувачі зможуть легко встановити мобільний додаток та розпочати ефективну роботу з нашою інтелектуальною системою.

### 4.2.2. Робота з додатком

Після встановлення мобільного додатку користувач може приступити до його використання. Додаток призначений для маркетологів і керівників підприємств, щоб забезпечити зручний доступ до корпоративної інформації та аналітичних інструментів.

Крок 1. Запуск мобільного додатку. Для запуску програми потрібно натиснути на піктограму додатку на екрані свого пристрою. Після цього відкриється головний екран програми.

Крок 2. Авторизація. На початковому екрані з'явиться форма авторизації. Для забезпечення безпеки доступу користувачеві необхідно ввести свої облікові дані: логін і пароль. Поле для введення пароля приховує символи (у вигляді крапок або зірочок).

У разі успішного введення даних користувач отримує доступ до головного меню програми. Якщо логін або пароль введено неправильно, програма виведе відповідне повідомлення та запропонує повторити спробу. Алгоритм входу наведено на рис. 4.7.

Крок 3. Додавання корпоративного облікового запису. Після входу користувач має додати корпоративний обліковий запис:

Натиснути кнопку «Додати обліковий запис».

Ввести логін і пароль корпоративної пошти.

Підтвердити додавання облікового запису.

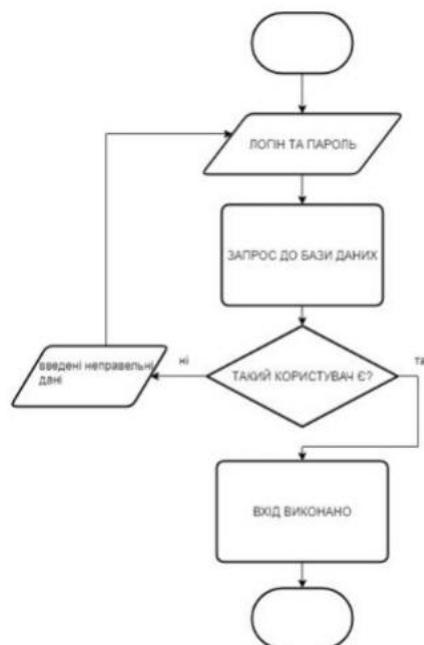


Рис. 4.7. Алгоритм авторизації в мобільному додатку

#### Крок 4. Налаштування сповіщень

Щоб оперативно отримувати інформацію, користувач може увімкнути сповіщення.

Перейти до Налаштувань.

Увімкнути опцію «Сповіщення».

Налаштувати типи сповіщень: звук, вібрація або світлодіод.

#### Крок 5. Робота з інформаційними та аналітичними модулями

Створення та відправка запиту або звіту

У головному меню натиснути «Створити».

Ввести дані, які необхідно зберегти або відправити, включаючи вибір типу звіту чи прогнозу.

При необхідності додати вкладення (графіки, таблиці, документи).

Натиснути «Зберегти» або «Відправити».

Перегляд аналітичних даних

Перейти до розділу «Аналітика», або «Звіти».

Обрати потрібний звіт або прогноз.

Відкрити документ для детального ознайомлення.

Фільтри та сортування. У розділі «Звіти» користувач може скористатися фільтрами для зручного пошуку потрібних документів, наприклад:

- Фільтрувати за періодом;
- Вибирати звіти за типом (продажі, ринковий аналіз тощо).
- Крок 6. Додаткові функції

Мобільний додаток надає широкий функціонал для маркетологів та керівників:

Керування даними клієнтів: доступ до інформації про клієнтів, їх активність та історію взаємодії.

Прогнозування: створення прогнозів продажів або аналізу ринкових тенденцій.

Підтримка багатокористувацького доступу: робота з командними завданнями.

Індивідуальні налаштування: налаштування віджетів та інтерфейсу для спрощення доступу до важливої інформації.

Наведена інструкція охоплює базові кроки для налаштування та початку роботи з додатком. Деталі можуть змінюватися залежно від версії додатку та оновлень. Для вирішення технічних питань слід звертатися до системного адміністратора або технічної підтримки.

Застосовуючи мобільний додаток, користувачі отримають ефективний інструмент для аналізу та прийняття управлінських рішень, що значно оптимізує бізнес-процеси підприємства.

### **4.2.3. Тестування розробленої системи**

У процесі розробки мобільного додатку для нашої інтелектуальної системи важливим етапом є тестування, що відповідає стандартам QM. Відповідно до

сучасних гнучких методологій, таких як Agile, процес тестування охоплює весь життєвий цикл розробки програмного забезпечення, починаючи з аналізу вимог і закінчуючи впровадженням та підтримкою.

Тестування дозволяє гарантувати якість роботи додатку, його надійність та зручність використання. Оскільки наша система спрямована на підтримку прийняття управлінських рішень, особливу увагу приділено перевірці функціональності, продуктивності та безпеки.

#### Основні етапи тестування

##### Етап 1: Планування тестування

На цьому етапі визначаються мета тестування та його ключові аспекти:

**Функціональність:** перевірка основних функцій додатку (вхід, авторизація, доступ до даних, звіти тощо).

**Сумісність:** тестування роботи на різних версіях Android (починаючи з 6.0).

**Продуктивність:** перевірка швидкості обробки запитів, використання ресурсів пристрою.

**Юзабіліті:** аналіз зручності інтерфейсу користувача.

Розробляються тест-плани, які включають:

Взаємодію з базою даних;

Перевірку стабільності роботи під час використання різних інтернет-з'єднань (Wi-Fi, мобільний інтернет);

Перевірку відображення інтерфейсу на різних екранах.

##### Етап 2: Визначення типів тестування

Для нашого додатку передбачено наступні типи тестування:

**Модульне тестування (Unit-тести):** перевірка роботи окремих модулів системи.

**Інтеграційне тестування:** перевірка взаємодії між різними компонентами (зокрема, між інтерфейсом користувача та базою даних).

**Функціональне тестування:** перевірка відповідності роботи системи її функціональним вимогам.

Тестування продуктивності: оцінка швидкості роботи додатку, обробки великих обсягів даних.

Етап 3: Розробка тестової документації

На основі вимог до додатку створюються:

Тестові сценарії: описують основні кейси використання додатку.

Тестові випадки: включають перевірку функціональних можливостей, зручності використання, безпеки.

Окремо додаються сценарії для перевірки:

Витрат заряду акумулятора;

Використання оперативної пам'яті;

Сумісності з різними розмірами екранів.

Етап 4: Автоматизоване та ручне тестування

На етапі розробки виконуються модульні тести у середовищі Android Studio.

Після створення готового білда проводиться ручне тестування:

Модульне тестування: перевірка функцій обробки даних, роботи з API.

Функціональне тестування: перевірка роботи інтерфейсу, взаємодії з базою даних.

Код модульного тесту для запуску додатку:

```
java
Копировать код
import org.junit.Test;
import static org.junit.Assert.assertNotNull;
public class ISDecisionSupportTest {
    @Test
    public void testAppLaunch() {
        ISApp app = new ISApp();
        app.launch();
        assertNotNull(app.getMainScreen());
    }
}
```

Етап 5: Тестування юзабіліті

Проводиться перевірка зручності роботи користувачів із додатком. Важливі аспекти:

Навігація між екранами;

Інтуїтивність розташування елементів інтерфейсу;  
 Реакція на дії користувача (наприклад, повернення на попередній екран, перехід між модулями).

Етап 6: Перевірка продуктивності

Для забезпечення стабільної роботи програми перевіряється:

Швидкість виконання запитів до бази даних;

Реакція системи на зміну інтернет-з'єднання;

Споживання ресурсів пристрою.

Етап 7: Тестування безпеки

Оскільки додаток зберігає конфіденційні дані, тестуються:

Захист даних у базі;

Перевірка SQL-ін'єкцій;

Використання захищених протоколів для передачі даних.

Етап 8: Бета-тестування. Проводиться внутрішнє тестування серед обмеженої кількості користувачів. Мета - виявлення недоліків перед публічним релізом.

Після завершення тестування мобільний додаток готовий до розгортання. Ретельне тестування забезпечує високу якість програмного продукту, задоволеність користувачів та відповідність функціональним вимогам.

Створимо чек-лист для перевірки нашої ІС.

Головний екран

1. Кнопка "Авторизація"

- Відкриває форму введення логіна та пароля.
- Логін та пароль вводяться коректно (текст пароля прихований).
- Повідомлення про помилку при введенні некоректних даних.
- Повертає користувача до головного екрана після успішної авторизації.

2. Кнопка "Реєстрація"

- Відкриває форму для створення нового облікового запису.

- Валідація полів (логін, пароль, email).
- Відображає повідомлення про успішну реєстрацію.

#### Екран аналітики

3. Кнопка "Створити звіт"
  - Відкриває форму вибору параметрів для створення звіту.
  - Генерує звіт за обраними параметрами.
  - Відображає помилку при відсутності даних для звіту.
4. Кнопка "Експорт звіту"
  - Відображає вибір формату експорту (PDF, Excel).
  - Зберігає файл у вибраному форматі.
  - Показує повідомлення про успішний експорт.

#### Екран користувачів

5. Кнопка "Додати користувача"
  - Відкриває форму для введення даних нового користувача.
  - Перевіряє, чи всі обов'язкові поля заповнені.
  - Відображає повідомлення про успішне створення користувача.
6. Кнопка "Редагувати користувача"
  - Відкриває форму редагування для вибраного користувача.
  - Зберігає внесені зміни в базу даних.
  - Відображає повідомлення про успішне редагування.
7. Кнопка "Видалити користувача"
  - Запитує підтвердження перед видаленням.
  - Видаляє вибраного користувача після підтвердження.
  - Відображає повідомлення про успішне видалення.

#### Екран даних

8. Кнопка "Оновити дані"

- Завантажує актуальні дані з бази.
  - Відображає повідомлення про успішне оновлення.
  - Показує помилку при недоступності сервера.
9. Кнопка "Фільтрувати"
- Відкриває форму для вибору параметрів фільтрації.
  - Відображає дані, що відповідають вибраним критеріям.
  - Показує повідомлення про відсутність даних, якщо фільтр не дав результатів.

#### Екран налаштувань

10. Кнопка "Змінити пароль"
- Відкриває форму для введення старого та нового паролів.
  - Перевіряє збіг нового пароля з підтвердженням.
  - Відображає повідомлення про успішну зміну пароля.
11. Кнопка "Скинути налаштування"
- Запитує підтвердження перед скиданням.
  - Відновлює налаштування за замовчуванням після підтвердження.
  - Відображає повідомлення про успішне скидання.

#### Глобальні функції

12. Кнопка "Назад"
- Повертає користувача до попереднього екрана.
  - Зберігає поточні зміни перед поверненням (якщо це потрібно).
13. Кнопка "Вихід"
- Запитує підтвердження перед виходом.
  - Виходить із системи після підтвердження.
  - Закриває програму при виборі опції "Вийти".

Наведений вище чек-лист охоплює всі основні функції кнопок мобільного додатку нашої інтелектуальної системи. Його виконання дозволить забезпечити коректну роботу всіх функціональних елементів програми та гарантувати якість кінцевого продукту.

Також наведемо приклад тестового звіту тестування ІС.

Інформація про тест

- Назва тесту: Тестування функціональності кнопок мобільного додатку
- Об'єкт тестування: Мобільний додаток для інтелектуальної системи підтримки управлінських рішень
- Метод тестування: Юніт-тест
- Інструмент: Android Studio (JUnit)
- Дата проведення: 10 листопада 2024
- Відповідальний: [В. Шепетило]

Мета тесту. Перевірити, чи всі кнопки мобільного додатку працюють відповідно до заявленої функціональності.

Таблиця 4.2

#### Сценарії тестування

Кнопка	Дія	Очікуваний результат	Фактичний результат	Статус
Авторизація	Введення логіна та пароля	Відкриття головного екрану програми	Відкривається головний екран	Успішно
Реєстрація	Введення нових даних користувача	Повідомлення про успішну реєстрацію	Відображено повідомлення	Успішно
Створити звіт	Натискання кнопки	Відкривається форма створення звіту	Форма створення відкривається	Успішно
Експорт звіту	Вибір формату експорту	Файл зберігається у вибраному форматі	Файл збережено	Успішно
Додати користувача	Введення даних нового користувача	Повідомлення про успішне додавання	Повідомлення відображено	Успішно
Оновити дані	Натискання кнопки	Завантажуються актуальні дані	Дані успішно оновлені	Успішно
Змінити пароль	Введення старого та нового пароля	Повідомлення про успішну зміну пароля	Повідомлення відображено	Успішно
Назад	Повернення до попереднього екрана	Повернення до попереднього екрана	Повернення здійснено	Успішно
Вихід	Підтвердження виходу	Вихід із системи	Вихід здійснено	Успішно

Проблеми, виявлені під час тестування

1. Всі перевірені кнопки працюють належним чином.
2. Не виявлено критичних помилок у функціональності.

Висновок тестування: Юніт-тестування функціональності кнопок мобільного додатку для інтелектуальної системи підтримки управлінських рішень завершено успішно. Усі кнопки працюють згідно з технічними вимогами.

Наступним кроком є проведення інтеграційного тестування для перевірки взаємодії між компонентами системи.

Інтеграційне тестування є важливим етапом життєвого циклу розробки програмного забезпечення, що дозволяє перевірити, як різні компоненти системи взаємодіють один з одним. Для нашої інтелектуальної системи підтримки управлінських рішень мета інтеграційного тестування полягає у перевірці взаємодії між такими компонентами:

Інтерфейс користувача (UI) та бекенд.

База даних та логіка обробки даних.

Зовнішні сервіси (наприклад, API для аналітики).

Модулі обробки даних (аналітика, прогнозування) та сховище даних.

Мета тестування - переконатися, що всі компоненти системи працюють узгоджено, дані передаються коректно, а взаємодія між компонентами не викликає помилок.

План інтеграційного тестування.

Підготовка тестового середовища:

Розгорнути тестову версію бекенд-сервера.

Налаштувати підключення до тестової бази даних.

Забезпечити доступ до API для аналітики.

Підключити мобільний додаток до тестового середовища.

Тестування сценаріїв взаємодії:

Передача даних між інтерфейсом користувача та сервером.

Запити до бази даних та отримання відповідей.

Використання API для аналізу даних та отримання результатів.

Перевірка коректності передачі даних:

Тестувати запити на створення, оновлення та видалення записів у базі даних.

Перевіряти відповідність даних, отриманих від сервера, очікуваним результатам.

Тестування збоїв:

Відключення мережі під час виконання запиту.

Неправильні або неповні дані, надіслані на сервер.

Збої в роботі бази даних або API.

Приклад тестового сценарію

Сценарій: Передача даних про маркетингову кампанію від інтерфейсу користувача до сервера та запис у базу даних.

Попередні умови:

У користувача є доступ до мобільного додатку.

Тестова база даних налаштована.

Кроки:

Ввести дані про нову маркетингову кампанію (назва, бюджет, терміни).

Натиснути кнопку "Зберегти".

Сервер отримує запит на створення запису.

Дані записуються в базу даних.

Очікуваний результат:

На інтерфейсі з'являється повідомлення про успішне збереження.

У базі даних з'являється новий запис із введеними даними.

Сервер повертає статус 200 ОК.

Фактичний результат:

Дані збережені успішно. Всі модулі працюють коректно.

Ключові метрики для інтеграційного тестування

Час відгуку: Час, необхідний для обробки запиту між компонентами.

Процент успішних запитів: Відсоток успішно виконаних сценаріїв.

Виявлені дефекти: Кількість помилок, знайдених під час тестування.

Стабільність: Частота збоїв при виконанні тестів.

Інтеграційне тестування дозволяє перевірити, чи всі компоненти системи узгоджено взаємодіють між собою, і виявити можливі дефекти на рівні інтеграції. Після завершення інтеграційного тестування можна перейти до системного тестування, щоб перевірити всю систему в цілому.

Таким чином проводили тестування розроблюваного програмного продукту. Усі тестові артефакти можна завантажити в таск менеджер, наприклад Jira, для подальшої командної роботи.

## ВИСНОВКИ

В магістерській роботі на тему «Інтелектуальна система вирішення оптимізаційних задач на основі методу Монте-Карло з детермінованим навчанням» були вирішені поставлені в дослідженні задачі.

Розробка мобільного додатку для управління інформаційною системою є критично важливим кроком у забезпеченні конкурентоспроможності сучасного підприємства. Цифрова трансформація змінює спосіб функціонування бізнесу, і створення інтелектуальних систем для підтримки управлінських рішень стає невід'ємною частиною цього процесу.

У межах дослідження було проведено аналіз теоретичних основ розробки мобільних додатків, включаючи принципи життєвого циклу програмного забезпечення, сучасні підходи до розробки інтерфейсів користувача, а також інтеграцію мобільних додатків з корпоративними системами. Було акцентовано увагу на важливості адаптації програмних продуктів до специфічних потреб підприємств.

Аналіз існуючих рішень на ринку показав, що жоден з доступних мобільних додатків не здатен повністю відповідати вимогам, пов'язаним із забезпеченням конфіденційності, ефективності та адаптивності. Багато готових рішень зберігають дані на сторонніх серверах, що створює ризик витоку інформації, а також містять функції, які не завжди відповідають бізнес-процесам підприємств. Це обумовило необхідність розробки індивідуального програмного забезпечення.

Проектування мобільного додатку було розпочато з визначення архітектури, яка включає серверну частину, базу даних, мобільний клієнт і функціональні модулі. Такий підхід дозволив створити модульну архітектуру, яка забезпечує простоту оновлення та розширення системи.

Розроблений інтерфейс користувача враховує сучасні принципи UI/UX-дизайну. Усі елементи інтерфейсу були адаптовані до мобільних пристроїв, що гарантує простоту використання та інтуїтивність. Рішення було орієнтоване на

забезпечення зручності взаємодії користувача із системою, відображення необхідної інформації та виконання управлінських завдань.

Крім цього, розроблено користувацькі сценарії, які включають дії, що відповідають ролям у системі, зокрема для маркетологів, аналітиків та керівників підприємства. Ці сценарії допомагають структурувати процеси використання додатку та оптимізувати взаємодію з інформаційною системою.

Тестування стало ключовим етапом забезпечення якості розробленого мобільного додатку. На стадії розробки проводилося модульне тестування з використанням середовища Android Studio, що дозволило виявити помилки в окремих компонентах. Інтеграційне тестування дало змогу перевірити взаємодію між модулями системи, а також між додатком і базою даних.

Було приділено значну увагу тестуванню безпеки, що включало перевірку шифрування даних, управління доступом і захист від зовнішніх загроз. Важливим етапом стало функціональне тестування, яке проводилося методом Ad-hoc для перевірки ключових сценаріїв використання.

Розроблений мобільний додаток має низку суттєвих переваг. По-перше, забезпечується високий рівень конфіденційності, оскільки всі дані зберігаються на внутрішніх серверах підприємства. Це дозволяє уникнути ризиків, пов'язаних із використанням сторонніх сховищ даних.

По-друге, мобільний додаток адаптований до потреб конкретного підприємства. Його функціонал обмежується лише необхідними інструментами, що підвищує ефективність роботи та спрощує користування системою. Відсутність зайвих функцій та реклами позитивно впливає на продуктивність.

Крім того, гнучка модульна архітектура відкриває можливості для подальшого вдосконалення та розвитку системи. У разі зміни бізнес-процесів або появи нових потреб додаток легко адаптується до цих змін.

Розроблений мобільний додаток сприяє підвищенню ефективності управлінських рішень за рахунок інтеграції всіх необхідних даних в єдину систему. Завдяки цьому керівники підприємства отримують можливість швидко

реагувати на зміни в ринковій ситуації, а маркетологи та аналітики - більш точно оцінювати результати своєї роботи.

Інтелектуальна система підтримки управлінських рішень сприяє автоматизації багатьох рутинних операцій, що дозволяє зосередитися на стратегічних завданнях. Крім того, мобільний доступ до системи забезпечує гнучкість у роботі, дозволяючи приймати рішення у будь-якому місці та в будь-який час.

Таким чином, розробка мобільного додатку для інтеграції з інформаційною системою підприємства є важливим кроком у підвищенні ефективності бізнес-процесів. Використання сучасних технологій дозволяє створювати програмні продукти, які відповідають індивідуальним потребам підприємств і забезпечують високий рівень безпеки даних.

Розроблений додаток підсилює інформаційну взаємодію всередині підприємства, оптимізує процеси прийняття рішень і сприяє впровадженню інноваційних підходів у роботу. Його впровадження дозволяє підприємству залишатися конкурентоспроможним і адаптуватися до викликів цифрової епохи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Best Hybrid Mobile App UI Frameworks: HTML5, CSS and JS - Режим доступу : <http://noeticforce.com/best-hybrid-mobile-app-ui-frameworks-html5-jscss/>. - Дата доступу : 08.11.2024.
2. Building Cross-Platform iOS/Android Apps with Xamarin, Visual Studio, and C# - Part 1 by Jim Wilson - Режим доступу : <https://app.pluralsight.com/library/courses/cross-platform-ios-android-visualstudiocsharp/table-of-contents/>. - Дата доступу : 08.11.2024.
3. Charles Petzold. Creating Mobile Apps with Xamarin.Forms First Edition / Documentation for app developers [Електронний ресурс]. - Режим доступу: <https://developer.android.com/docs>
4. H. Hoehle, V. Venkatesh, "Mobile Application Usability: Conceptualization and Instrument Development", MisQuarterly, №39 (2), p. 435-472, 2015.
5. H. Hoehle. V. Venkatesh. "Mobile Application Usability: Conceptualization and Instrument Development". MisQuarterly. №39 (2). p. 435-472. 2015.
6. habr.com [Електронний ресурс]. - Режим доступу: <https://habr.com/>
7. Hammer, M. What is Business Process Management?. In J. vom Brocke & M. Rosemann (Eds.), Handbook on Business Process Management 1 (pp. 3-16). Berlin: Springer, 2015.
- Including jQuery, AJAX, Web Design, Scripting and Mobile Application Development. - Independently published, 2019. - 546 с.
8. Jonathan Peppers. Xamarin Cross-platform Application Development Second Edition / Peppers J. - Birmingham : Packt, 2015. - 459 с.
9. Kaplan, R. S., & Norton, D. P. The Balanced Scorecard: Translating Strategy into Action. Boston: Harvard Business Review Press, 2020.
10. Laura Thomson, Luke Welling. PHP and MySQL Web Development (Developer's Library). - Addison-Wesley Professional, 2016. - 688 с
11. Meier R. Professional Android, 4-th ed. Wrox, 2018. 928 p.
12. Michael B. White. Mastering JavaScript: A Complete Programming Guide

Including jQuery, AJAX, Web Design, Scripting and Mobile Application Development. - Independently published, 2019. - 546 с.

13. Michael B. White. Mastering JavaScript: A Complete Programming Guide Mobile Operating System Market Share Ukraine [Електронний ресурс] - Режим доступу: [bit.ly/2HyPdKr](http://bit.ly/2HyPdKr) -

14. Nilanchala Panigrahy. Xamarin Mobile Application Development for Android Second Edition / Panigrahy N. - Birmingham : Packt, 2015. - 296 с.

15. Olena P. Kopishynska, Yuriy V. Utkin, Olga G. Kartashova. Monte carlo method application to support decision-making on investments allocation // Actual problems of economics #5(191), 2017

16. Phillips B., Stewart C., Marsicano K. Android Programming: The Big Nerd Ranch Guide : 3-rd ed. February 9, 2017. 624 p. Publishing, 2019. - 256 с

17. RoboVM documentation - Режим доступу: <http://docs.robovm.com/>. - Дата доступу : 03.11.2024.

18. Saaty, T. L., & Vargas, L. G. Decision Making with the Analytic Hierarchy Process. New York: Springer. 2018.

19. SQLite documentation [Електронний ресурс]. - Режим доступу: <https://www.sqlite.org/docs.html>

20. В.В. Білоус, "Мобільні додатки для навчання математики як засіб підвищення мотивації учнів молодшої школи", Відкрите освітнє е-середовище сучасного університету, №3, с. 303-309, 2017. [Електронний ресурс] - Режим доступу до ресурсу: <https://openedu.kubg.edu.ua/journal/index.php/openedu/article/view/96>

21. Зуєва , Н.Ю. 2024. Створення моделі за допомогою методу Монте-Карло. Інформаційні технології і системи в документознавчій сфері. (2024), 43-45.

22. Історія виникнення інтернету, 2016, [Електронний ресурс] - Режим доступу до ресурсу: <https://webbuilding.com.ua/ukr/articles/istoriya-interneta/>

23. Копішинська О.П., Калініченко А.В., Шестаченко К.В. Дослідження ефективності застосування методу Монте-Карло при оптимізаційному

модельованні управлінських задач про призначення // Науковий вісник Херсонського державного університету.– Серія: Економічні науки.– 2014.– Вип. 5, Ч. 4. – С. 228–231.

24. М.В. Колесник, О.Я. Ярмолюк, І.В. Созинова. Ефективність управління бізнес-процесами в ринковому середовищі цифрової інфраструктури // Цифрова економіка та економічна безпека, К.: - 2023. - с. 97-103.

25. Математичні методи дослідження операцій : підручник / Є. А. Лавров, Л. П. Перхун, В. В. Шендрік та ін. – Суми : Сумський державний університет, 2017. – 212 с.

26. Михайленко О. Навіщо бізнесу мобільний додаток? Режим доступу: <https://msb.aval.ua/news/?id=24743>.

27. Мобільний додаток: кому він дійсно потрібний і скільки коштує розробка. [Електронний ресурс] - Режим доступу до ресурсу: <https://forbes.ua/business/mobilniy-dodatok-komu-vin-diysno-potribniy-i-skilki-koshtue-rozrobka-17052021-1604>

28. Нестеренко О.В., Ковтун ець О.В., Фаловськнй О.О. Інтелектуальні системи і технології. Ввідний ку рс: Навч. посібник. К.: Національна академія управління, 2017. - 90 с.

29. Портер, М. Конкурентна стратегія: Техніки аналізу галузей і конкурентів. Київ: Наш Формат, 2020.

30. Фісун Ю.В., Борисенко О.С., Ярмолюк О.Я. Інноваційні підходи проведення маркетингових досліджень сучасних підприємств. Держава та регіони. Серія: Економіка та підприємництво. науково-виробничий журнал. Запоріжжя.: КПУ, 2022. Вип. 2(125). URL: [http://www.econom.stateandregions.zp.ua/journal/2022/2\\_2022/18.pdf](http://www.econom.stateandregions.zp.ua/journal/2022/2_2022/18.pdf)

31. Шепетило В.В. Розробка інтелектуальної системи вирішення оптимізаційних задач на основі методу монте-карло з детермінованим навчанням // Збірник матеріалів Міжнародної науково-практичної конференції молодих вчених «Інформаційні технології: економіка, техніка, освіта» (07-08 листопада 2024 року, Київ, Україна)