

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»**

Кафедра інформаційних систем в економіці

**ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«Інформаційні управляючі системи і технології»**

галузь знань 12 Інформаційні технології
спеціальність 122 Комп'ютерні науки

Форма навчання: заочна

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему: «Розробка управляючої інформаційної системи для торгової
компанії»

здобувача Суховієнка Олександра Віталійовича
(ПІБ, підпис)

Науковий керівник: Устенко Станіслав Веніамінович
(науковий ступінь, учене звання, ПІБ)

(підпис)

**Робота допущена до захисту перед екзаменаційною комісією з
атестації здобувачів вищої освіти (ЕК)**

Завідувач кафедри: к.е.н., доцент Тішков Б.О.
(науковий ступінь, учене звання, ПІБ)

(підпис)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці

**ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«Інформаційні управляючі системи і технології»**

галузь знань 12 Інформаційні
технології

спеціальність 122 Комп'ютерні
науки

ПОГОДЖЕНО:

Керівник проєктної групи (гарант)
освітньо-професійної програми

ЗАТВЕРДЖУЮ:

Завідувач кафедри
інформаційних систем в економіці

_____ Устенко С.В.
«___» _____ 202_ р.

Б.О.

_____ Тішков
«___» _____ 202_ р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Суховієнко Олександр Віталійович

(прізвище, ім'я, по батькові)

заочної форми навчання

очної (денної), заочної, дистанційної

на підготовку кваліфікаційної магістерської роботи

на тему: «Розроблення інформаційної управляючої системи торгової компанії»

Тему затверджено наказом ректора Університету від «___» _____ 202_ р.

№ _____

Кваліфікаційна магістерська робота виконується на матеріалах

План кваліфікаційної магістерської роботи

Розділ I «Дослідження та аналіз підходів до створення інформаційних управляючих систем предметної області»

(назва розділу)

Розділ II «Характеристика інформаційних управляючих систем та методи і моделі»

(назва розділу)

Розділ III «Розроблення проєктних рішень та їх реалізація»

(назва розділу)

Об'єкт дослідження кваліфікаційної магістерської роботи є процес розробки управляючої інформаційної системи торгової компанії.

Предмет дослідження: є сукупність теоретичних, методичних і практичних підходів до технології проєктування управляючої інформаційної системи

Мета кваліфікаційної магістерської роботи: розробка та впровадження управляючої інформаційної системи (IUC) для торгової компанії

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I

дослідити предметну область, де визначаються переваги та проблеми предметної області та їх причини та наслідки. Провести порівняльний аналіз зі схожими управляючими інформаційними системами, їх сильні та слабкі сторони та надати висновок щодо них. Обґрунтувати вибір підходів та методів до створення IUC, де потрібно описати кожний підхід, визначити та проаналізувати його переваги й проблеми та надати висновок під час аналізу.

У розділі II

дати характеристику інформаційної управляючої системи торгової компанії, а також описати методи та моделі, що будуть використовуватись. Визначити цілі підприємства та на основі них визначити переваги й недоліки, описати вимоги до системи. Описати структуру системи, а також, можливі та моделі в інформаційних управляючій системі і технологіях. Описати методи, що будуть закладені в систему, включаючи алгоритми, Соціально-економічні аспекти впровадження, бізнес вимоги. На основі вищезазначених методів надати опис та побудувати відповідну модель для використання в системі.

У розділі III

розробити проєктні рішення та реалізацію. Спроекувати БД до системи, в процесі чого визначити сутності та їх атрибути, розробити інфологічну модель БД, на основі інфологічної моделі розробити фізичну модель в обраному середовищі реалізації БД, де описати кожний атрибут надавши його характеристику, як от розмір поля, ключ, тощо. Провести тестування БД, визначивши вимоги до тестових даних та спосіб їх отримання. Обрати, описати та обґрунтувати методологію проєктування, архітектуру системи, середовище. Визначити склад

технічних та програмних засобів, а також оглянути можливості їх реалізації, описавши мінімально допустимий склад програмного та технічного забезпечення. Виконати тестування розробленої системи, скріншоти самої системи подати у додатках. Провести техніко-економічний аналіз, Реалізувати інформаційну систем, зробити якісний аналіз ризиків та визначити й Розробити Технічне забезпечення. Надати рекомендації на основі досліджень у цьому та попередніх розділах, щодо системи. Визначити результати дослідження та реалізації предметної області, описавши розроблені та запропоновані підходи та методики

Завдання підготував

науковий керівник _____ Устенко.

С.В.

(ініціали, прізвище)

(підпис)

«____» _____
2024 р.

Завдання одержав

здобувач

О.О.

(ініціали, прізвище)

_____ Суховієнко О.В

(підпис)

«____» _____ 2024 р

Реферат

Кваліфікаційна магістерська робота містить 90 сторінок, 2 таблиці, 61 рисунок, список використаних джерел з 78 найменувань, додатки.

«Розроблення інформаційної управляючої системи для торгівельної компанії»

Об'єктом дослідження кваліфікаційної магістерської роботи є процес розробки управляючої інформаційної системи торгової компанії.

Предметом дослідження є сукупність теоретичних, методичних і практичних підходів до технології проєктування управляючої інформаційної системи.

Мета і завдання дослідження. є розробка та впровадження управляючої інформаційної системи (ІУС) для торгової компанії. Відповідно до поставленої мети визначені такі *завдання*:

- дослідити предметну область, виконати аналіз існуючих систем;
- представити обґрунтування вибору підходів і технологій для створення системи;
- визначити структуру та представити характеристику системи;
- описати методи та моделі в управляючій інформаційній системі;
- спроектувати базу даних для ІУС;
- спроектувати базу знань для ІУС;
- спроектувати прототип ІУС;
- описати програмне забезпечення для системи інтелектуального аналізу дани
- представити результати реалізації системи.

Теоретична, методична та практична значущість отриманих результатів. Під час дослідження було досліджено предметну область, проаналізовано варіанти створення ІУС та можливих варіантів програмного забезпечення, розроблено прототип інформаційної управляючої системи, також було спроектовано реаліційну базу даних, та базу знань

Рік виконання кваліфікаційної магістерської роботи – 2024.

Рік захисту роботи – 2024.

Ключові слова: ІУС, бази даних, прототип, проєктування, база знань інформаційна система, інтелектуальний аналіз даних, СКБД, Use Case, інфологічна модель, Wix, UML, SQL.

В і д г у к
про кваліфікаційну магістерську роботу
здобувача навчально-наукового інституту «Інститут інформаційних
технологій в економіці»
освітньо-професійної програми «Інформаційні управляючі системи і
технології»
Суховієнка Олександра Віталійовича
(*прізвище, ім'я, по батькові*)
на тему «**Розроблення управляючої інформаційної системи торгової
компанії**»

1. Актуальність теми: Розроблення управляючої інформаційної системи торгової компанії є надзвичайно актуальною темою. У сучасній цифровій економіці ефективне управління даними та автоматизація бізнес-процесів стають ключовими факторами конкурентоспроможності. Інформаційна система дозволяє оптимізувати роботу компанії, покращити обслуговування клієнтів та приймати більш обґрунтовані управлінські рішення.
2. Позитивні риси кваліфікаційної магістерської роботи. В роботі досліджено предметну область, обґрунтовано вибір підходів, методів і технологій для створення системи, описати методи та моделі в ІУС малого підприємства, спроектована БД і ПЗ для системи,
3. Наявність самостійних розробок автора: Усі розділи КМР виконані самостійно
4. Цінність теоретичних висновків та практичних рекомендацій: Заслуговує на увагу наявність креативних підходів і технологій в процесі проєктування ІУС для торгової компанії
5. Наявність недоліків: недоліків: Бажано було би надати методику впровадження авторського проєкту
6. Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК: .
КМР допущена до захисту і заслуговує задовільну оцінку.

Науковий керівник д.е.н., професор С.В. Устенко

«17 » травня 2024 р.

Рецензія

на кваліфікаційну магістерську роботу
здобувача вищої освіти

Суховієнка Олександра Віталійовича

(прізвище, ім'я, по батькові)

Тема «Розроблення інформаційної управляючої системи для торгової компанії»

Актуальність теми кваліфікаційної магістерської роботи і доцільність її розроблення визначається тим, що в сучасних умовах, незважаючи на зацікавленість більшості торгових компаній у засобах підвищення ефективності та розвитку бізнесу, багато з них, особливо малі підприємства, стикаються з труднощами у впровадженні та використанні інформаційних управляючих систем (ІУС). Значна кількість таких підприємств зазнають банкрутства та занепаду через невдале та погано організоване прийняття рішень, оскільки бракує відповідних політик, методів та засобів на підприємствах. Магістерська робота присвячена розробці ІУС для торгової компанії, що сприятиме вирішенню зазначених проблем. З огляду на це, робота є актуальною, доцільною та становить практичний інтерес для сектору малого підприємництва.

Робота містить необхідні структурні елементи та складається з вступу, дослідження та аналізу підходів до створення інформаційних управляючих систем для обраної предметної області, характеристики інформаційних управляючих систем, методів й моделей, розроблення проектних рішень та їх реалізації, висновків та додатків. Загалом дослідження демонструє глибоке розуміння потреб торгових компаній, має комплексний характер та охоплює достатню кількість підходів до проектування та впровадження управляючої системи, включаючи аналіз області, обґрунтування вибору методик, технічних та програмних засобів та аналіз літератури й джерел на обрану тему.

Серед позитивних рис кваліфікаційної магістерської роботи можна виділити наступні аспекти. По-перше, дослідження предметної області чітко визначає та описує конкретні потреби і недоліки у торгових компаній, особливо малих підприємств, що дозволяє формувати характеристики системи у відповідності з їхніми вимогами. По-друге, автор демонструє увагу до вибору та опису методів і моделей, що використовуються у системі. По-третє, була розроблена просто і інтуїтивно зрозуміла база даних, що значно спрощує шлях навчання робітників

Основне зауваження до роботи полягає в тому, що при описі впровадження було б також доцільно надати рекомендації щодо майбутнього масштабування системи по мірі розвитку підприємства, покрокову стратегію впровадження

системи в цілому та структурований план навчання й розширення можливостей працівників.

У рамках роботи досить глибоко розкрита тема. Надана велика кількість матеріалу, що до можливих альтернативних рішень. Висновки обґрунтовані та логічно випливають із дослідження. Надані у роботі рекомендації, щодо впровадження та використання програми змістовні, логічно викладені та практично спрямовані на вирішення визначених у дослідженні проблем та ризиків.

Вищезазначене дозволяє стверджувати, що дослідження має практичну значущість, магістерська робота виконана з дотриманням вимог та стандартів, має теоретичну цінність і практичне значення, і заслуговує оцінки хорошої оцінки.

”””

Місце роботи та посада рецензента

Науковий ступінь, учене звання (за наявності) канд екон наук, доц Ващаєв С.С .

(підпис,

ПІБ)

ЗМІСТ

Реферат	3
ЗМІСТ	3
ВСТУП.....	4
РОЗДІЛ 1 «ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ПРЕДМЕТНОЇ ОБЛАСТІ»	7
1.1 Аналіз існуючих інформаційних систем.....	7
1.1.1 Дослідження предметної області.....	7
1.1.2 Дослідження інформаційних управляючих систем.....	10
1.2 Обґрунтування вибору підходів і технологій для створення інформаційної управляючої системи	21
РОЗДІЛ 2 «ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ТА МЕТОДИ І МОДЕЛІ».....	23
2.1 «Структура і характеристика системи»	23
2.2 «Методи та моделі в інформаційних управляючих системах і технологіях».....	37
РОЗДІЛ 3 «РОЗРОБЛЕННЯ ПРОЄКТНИХ РІШЕНЬ ТА ЇХ РЕАЛІЗАЦІЯ»	54
3.1 Проєктування бази даних та/або сховища даних для інформаційної управляючої системи	54
3.2 «Проєктування бази знань інформаційної управляючої системи та засоби інтелектуального аналізу даних»	65
3.2.1 Реалізація інтелектуального аналізу великих даних з метою наповнення бази знань	68
3.3 Програмне забезпечення.....	73
3.3.1 Проєктування програмного забезпечення	73
3.3.2 Вибір архітектури програмного забезпечення	74
3.3.3 Інструментальні засоби розробки для створення прикладного програмного забезпечення інформаційних управляючих систем.....	77
3.4.3 Опис розробленого технічного пристрою.....	84
3.4.4 «Реалізація інформаційної управляючої системи»	87
ВИСНОВКИ	89
ДЖЕРЕЛА.....	91
ДОДАТКИ.....	96

ВСТУП

Актуальність теми. У сучасних умовах розвитку ринку та посилення конкуренції, торгові компанії стикаються з необхідністю підвищення ефективності управління бізнес-процесами та покращення якості обслуговування клієнтів. Зростаючий обсяг інформації, яку потрібно обробляти, та необхідність оперативного прийняття управлінських рішень вимагають застосування сучасних інформаційних технологій. Управляюча інформаційна система (іуч) стає ключовим інструментом, що забезпечує підтримку прийняття рішень, оптимізацію ресурсів та підвищення загальної продуктивності компанії.

Впровадження та використання управляючих інформаційних систем в торгових компаніях є не лише актуальним, але й необхідним кроком для забезпечення їх конкурентоспроможності та успішного розвитку в умовах сучасного ринку. Цей дипломний проект спрямований на дослідження та розробку ефективної ІУС для торгової компанії, яка дозволить підвищити ефективність її діяльності, оптимізувати бізнес-процеси та покращити взаємодію з клієнтами.

В процесі підготовки дипломного проекту на тему "Управляюча інформаційна система торгової компанії" важливо ознайомитися з сучасними дослідженнями та публікаціями, які висвітлюють теоретичні та практичні аспекти розробки та впровадження таких систем. Розглянемо ключові наукові праці та статті, які впливають на розвиток даної галузі.

У наукових працях розглядаються основні концепції та теоретичні підходи до управлінських інформаційних систем. Автори, такі як Laudon і Laudon у своїй книзі "Management Information Systems: Managing the Digital Firm", детально описують архітектуру ІУС, їх функціональні можливості та переваги для бізнесу. Їх роботи підкреслюють важливість інтеграції ІУС з іншими бізнес-системами та підходів до забезпечення безпеки даних.

Важливим аспектом досліджень є аналіз впливу ІУС на ефективність бізнес-процесів. У статтях таких авторів, як Davenport і Harris ("Competing on Analytics: The New Science of Winning"), розглядається, як використання аналітичних можливостей ІУС може допомогти компаніям приймати обґрунтовані рішення на основі даних. Вони демонструють, що впровадження ІУС сприяє оптимізації бізнес-процесів, зниженню витрат та підвищенню продуктивності.

Аналіз останніх досліджень і публікацій показує, що впровадження ІУС у торгових компаніях має суттєвий позитивний вплив на їх діяльність. Сучасні дослідження підкреслюють необхідність інтеграції новітніх технологій та адаптації систем до конкретних потреб компаній. Подальші дослідження можуть зосередитися на вивченні конкретних методів оптимізації бізнес-процесів за допомогою ІУС, а також на аналізі довгострокового впливу таких систем на стійкий розвиток компаній.

Цей аналіз допомагає краще зрозуміти поточний стан і перспективи розвитку управлінських інформаційних систем, що є основою для розробки ефективних рішень у рамках дипломного проекту.

Наукова проблема, яка розглядається в даному дослідженні, полягає у відсутності ефективної інформаційної системи, яка б забезпечувала комплексне управління бізнес-процесами торгової компанії, включаючи облік запасів, управління продажами, аналітику та взаємодію з клієнтами. Існуючі системи часто не відповідають вимогам сучасного бізнесу, не інтегровані між собою, що призводить до дублювання даних, помилок та затримок у прийнятті рішень. Це знижує ефективність управління і може негативно впливати на конкурентоспроможність компанії.

Мета роботи проекту - є розробка та впровадження управляючої інформаційної системи (ІУС) для торгової компанії, яка забезпечить підвищення ефективності управління бізнес-процесами, оптимізацію ресурсів та покращення взаємодії з клієнтами.

Методи дослідження - Для досягнення поставленої мети і вирішення конкретних завдань даного дослідження були використані наступні методи: *аналіз літературних джерел, системного аналізу, проектування інформаційних систем, збору та обробки даних, моделювання та оптимізації, впровадження інформаційних систем, оцінки ефективності, методи порівняльного аналізу, емпіричні методи.*

Практичне значення одержаних результатів полягає в тому, що спроектована інформаційна система може використовуватися для кормстування персоналом невеликої фірми. Зокрема, така система може бути застосована на невеликих підприємствах надання послуг, тощо, де може потребуватися функціонал ІУС для ведення малого бізнесу

Використані інструментальні засоби: середовище проектування Microsoft Access, операційна система Windows 10, ninjamock , WizWhy, текстовий редактор Microsoft Word 2016 для підготовки та оформлення роботи.

Структура роботи зумовлена метою і завданнями та складається зі вступу, трьох розділів, висновків, додатків та переліку джерел посилання.

РОЗДІЛ 1 «ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ПРЕДМЕТНОЇ ОБЛАСТІ»

1.1 Аналіз існуючих інформаційних систем

1.1.1 Дослідження предметної області

Для початку, розглянемо, що з себе представляє інформаційна система. Інформаційна система (ІС) — сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів. Інформаційні системи включають в себе: технічні засоби обробки даних, програмне забезпечення і відповідний персонал. Чотири складові частини утворюють внутрішню інформаційну основу: засоби фіксації і збору інформації; засоби передачі відповідних даних та повідомлень; засоби збереження інформації; засоби аналізу, обробки і представлення інформації. Основна мета ІС - підтримка діяльності організації шляхом ефективного управління інформаційними потоками, використовується для автоматизації бізнес-процесів, забезпечення інформаційної підтримки прийняття рішень, зберігання та обробки даних.

Також існує інший, схожий тип, - це інформаційна управляюча система (ІУС) , вона схожа на попередню, але має свої відмінності і переваги, є різновидом інформаційної системи, яка не тільки збирає та обробляє інформацію, але й активно бере участь у процесі управління організацією. ІУС надає інструменти для планування, контролю та аналізу діяльності підприємства. В першу чергу - це комплексне програмне забезпечення та апаратні засоби, призначені для збору, обробки, аналізу, зберігання та використання інформації з метою управління діяльністю організації або підприємства.

Основні відмінності між ними в тому, що ІС зосереджується на зборі та обробці інформації, може мати обмежену або відсутню управлінську функціональність, в той час як ІУС також зосереджується на управлінні та прийнятті рішень, але включає функції планування, контролю та аналізу, що є ключовими для управління організацією, та зазвичай зазвичай є більш складною системою, оскільки вона інтегрує більше функцій та інструментів для підтримки управління.

Торгові компанії займаються продажем товарів різних категорій, таких як електроніка, побутова техніка, одяг, взуття тощо. Її діяльність включає в себе такі основні процеси: закупівля товарів у постачальників, зберігання товарів на складі, продаж товарів клієнтам через офлайн та онлайн канали, обслуговування клієнтів (підтримка, гарантійне обслуговування тощо).

Інформаційна управляюча система, в даному разі, торгової компанії, призначена для автоматизації та оптимізації ключових бізнес-процесів. Основні функції ІУС можуть включати управління товарними запасами та залишками на складі, допомога в закупівлі товару на відсезонення руку постачальників. Також вона може допомагати працівникам в управлінні продажами, а саме: обробка замовлень, управління торговими точками, інтеграція з онлайн платформами. Дуже велику користь може також дати система CRM, яка не рідко використовується у напрямках взаємодії з клієнтом, щоб мати певну інформацію про покупців, та зберігання їх даних, для подальшого покращення сервісу, та швидкості обробки їх запитів. Невід'ємною частиною систем у бізнесі, є ведення фінансового обліку, аналіз та прогнозування продажів. Ще один не менш важливий момент, це можливість ітерації з платіжними системами, адже світ йде до того, що звичайні гроші поступово переходять за задній план, а зараз досить популяризовано використовуються різні платіжні системи, для підвищення безпеки та зручності використання даного методу проведення фінансових операцій.

З інтенсивністю розвитку технологій, ведення будь-якого бізнесу, не являється можливим, без інформаційних систем, і у користувачів вже

сформувались певні вимоги, що до подальшого впровадження, використання, та розвидку програмного забезпечення. ІУС повинна забезпечувати комплексний підхід до управління всіма бізнес-процесами компанії, повина включати в себе всі необхідні модулі для управління закупівлями, продажами, складом, клієнтами та фінансами. Досить часто, йде комбінація різних систем, для покращення, та прискорення роботи бізнесу, тому система повинна мати можливість інтеграції з іншими програмними продуктами та сервісами, легко налаштовуватися під змінні вимоги бізнесу. Забезпечення високого рівня надійності роботи системи та захисту даних. Можливість роботи з системою через мобільні пристрої. Легкість адміністрування та підтримки системи. Так як технології не стоять на місці, можливість оновлення, буде дуже важливою функцією, щоб не проєктувати всю систему наново.

Для ефективного проєктування ІУС необхідно вести спостереження, та аналіз ринку, конкурентів. Адже з'являється все більше і більше нових методів розробок, типів оптимізації, не вийняток того, що нові вимоги користувачі, можуть стати стимулом до нового програму та подальшого розвинення та удосконалення системи.

Дослідження цієї області є критично важливим етапом для успішного проєктування ІУС торгової компанії. Воно дозволяє зрозуміти основні бізнес-процеси, визначити вимоги до системи та вибрати найбільш відповідні технології та рішення. Це дослідження забезпечує основу для подальшої розробки ефективного, надійного та зручного програмного забезпечення.

Фактично інформаційна система призначена для забезпечення поточною і відповідною інформацією менеджерів для підтримки прийняття стратегічних рішень.

У багатьох випадках основною проблемою, яка постає перед малими підприємствами, є дефіцит ресурсів, зокрема фінансових, які є засобом для отримання інших ресурсів, таких як працівники та обладнання. Впровадження потужних інформаційно-управляючих систем потребує значних фінансових вкладень у програмне забезпечення, кваліфікований персонал та фізичну

інфраструктуру. Такі обмеження неминуче призводять до дилеми між ефективним веденням підприємницької діяльності та повною або майже повною відмовою від впровадження цих рішень. Як наслідок, підприємство поступово втрачає конкурентоспроможність.

Другою важливою проблемою є набуття досвіду та навичок для використання, та проектування подібних програмних рішень. Ця проблема полягає в тому, що дуже важко залучити фахівців, та недостатній кваліфікації наявного персоналу, що призводить до нездатності підприємства повною мірою задовольнити свої потреби. Крім того, збільшення навантаження на працівників без відповідного підвищення заробітної плати ускладнює утримання кваліфікованих спеціалістів.

Також, одна із доволі розповсюджених проблем, це погана якість даних, і не бажання персоналу вивчати. Виникає ситуація, що в компанії може бути кілька сховищ даних, які можуть бути не сумісті між собою, що знатно ускладнює процес бізнесу, а персонал, який звик до рутинної праці, не проявляє будь-якої зацікавленості в цифровій обізнаності, за для полегшення робочого процесу, та оптимізації своєї праці. В результаті чого продуктивність може знизитися чи наявна технічна структура може не підтримувати нові навантаження.

1.1.2 Дослідження інформаційних управляючих систем.

На даній момен існує безліч інформаційних системи і залежно від сфери застосування виділяють такі види інформаційних систем:

Наукові ІС - призначені для автоматизації діяльності науковців, аналізу статистичної інформації, управління експериментом.

Системи автоматизованого проектування або САПР – системи автоматизованого проектування (CAD / CAM – Computer Aided Design / Computer Aided Manufacturing), призначені для автоматизації функцій інженерів-проектувальників, конструкторів, архітекторів, дизайнерів при створенні нової техніки або технології.

Такі ІС допомагають здійснювати:

- розробку нових виробів і технологій їх виробництва;

- різні інженерні розрахунки (визначення технічних параметрів виробів, витратних норм – трудових, матеріальних тощо);
- створення графічної документації (креслень, схем, змістувань);
- моделювання проєктованих об’єктів;
- створення програм, що управляють, для верстатів з числовим програмним управлінням.

Інформаційні системи організаційного управління призначені для автоматизації функцій управлінського персоналу. Враховуючи найбільш широке застосування і різноманітність цього класу систем, часто будь-які інформаційні системи розуміють саме в цьому тлумаченні. До цього класу належать інформаційні системи управління як промисловими підприємствами, так і непромисловими організаціями (банки, біржі, страхові компанії, готелі тощо) та окремими офісами (офісні системи).

ІС управління технологічними процесами або АСУТП – автоматизовані системи управління технологічними процесами (SCADA – Supervisory Control And Data Acquisition), призначені для автоматизації різних технологічних процесів (гнучкі виробничі процеси, металургія, енергетика тощо).

Інтегровані (корпоративні) ІС використовуються для автоматизації всіх функцій фірми і охоплюють весь цикл робіт від проєктування до збуту продукції. Класифікація інформаційних систем за вартістю та масштабами. Залежно від вартості та масштабів виділяють такі основні класи ІС: локальні системи, фінансово-управлінські системи, середні інтегровані системи, великі інтегровані системи. Основні характеристики виділених класів систем представлені в таблиці 1.

Таблиця 1, - Основні характеристики
виділених класів систем

Характеристика	Локальні системи	Середні інтегровані системи	Великі інтегровані системи
Особливості та період упровадження	Простої, коробковий варіант. Два тижні	Поетапне. Один і більше місяців	Поетапне, складне. Більше 9–12 місяців
Функціональна повнота	Облікові системи (за напрямками)	Комплексний облік і управління фінансами	Комплексне управління: облік, управління, виробництво
Співвідношення витрат: ліцензія /	1 / 0,5 / 2	1 / 2 / 1	1 / 1–5 / 1

упровадження устаткування			
Орієнтовна вартість, тисяч доларів	100- 5000\$	10 000 – 100 000 \$	100 000 – 500 000 \$ та більше

Джерело: сформовано автором на основі виконаного дослідження

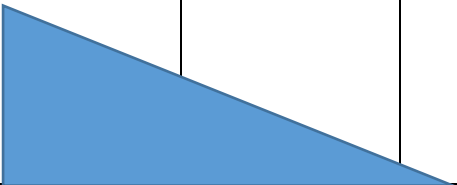
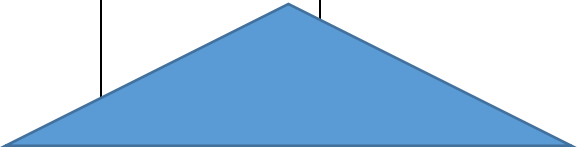
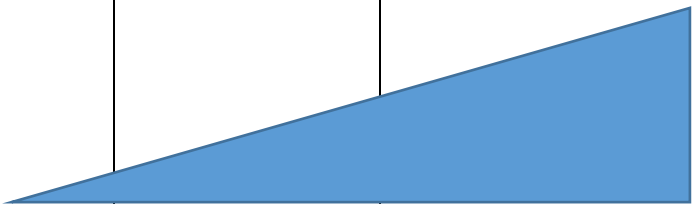
Локальні системи - призначені переважно для автоматизації обліку за одним або кількома напрямками (бухгалтерія, збут, склади, персонал тощо). Локальною системою може скористатися практично будь-яке підприємство, яке потребує управління фінансовими потоками й автоматизації облікових функцій. Локальні системи за багатьма критеріями універсальні, але низка розробників пропонує галузеві рішення, наприклад, особливі способи нарахування податків і т. ін. Цикл впровадження локальних систем невеликий, іноді можна скористатися “коробковим” варіантом, купивши програму і самостійно встановивши її на підприємстві. Вартість локальних систем коливається в діапазоні до \$50 000.

Фінансово-управлінські системи (малі інтегровані системи). Такі системи гнучко настроюються на потреби конкретного підприємства, добре інтегрують діяльність підприємства і призначені, насамперед, для обліку й управління ресурсами невиробничих компаній, хоча у багатьох системах цього класу присутні базові можливості управління виробництвом. Зазвичай вони універсальні, функціональні можливості таких систем ширші, ніж локальних. Вартість впровадження фінансовоуправлінських систем можна умовно визначити в діапазоні від \$50 000 до \$200 000, а іноді до \$300 000. До цього ж класу належить низка продуктів фірми 1С (“1С: Бухгалтерія” та ін.).

Середні інтегровані системи призначені для управління виробничим підприємством й інтегрованого змістування виробничого процесу. Облікові функції пропрацьовано глибоко, але вони виконують допоміжну роль. Ланцюжок змістування “збут–виробництво–закупівлі” є ядром цих систем. Підрозділи підприємства (фінанси, бухгалтерія, маркетинг та ін.) будують свою діяльність, спираючись на дані цього ланцюжка. Середні системи значно складніші в установці: цикл впровадження займає від 6 місяців до півтора року і більше. Причина в тому, що система покриває потреби кількох підрозділів і повністю інтегрує виробниче підприємство, що вимагає значних спільних зусиль співробітників підприємства, постачальника ІС або консалтингової компанії, яка здійснює впровадження. Середні системи передбачають, що виробниче підприємство має працювати як добре налагоджений годинник, де основними

механізмами управління є змістування й оптимальне управління запасами і виробничим процесом, а не облік кількості рахунків-фактур за період. Вартість упровадження середніх систем починається, як і в фінансово-управлінських системах, у межах \$50 000, але, залежно від обсягу проекту, може досягати \$500 000 і більше. Прикладами можуть бути, зокрема, системи “Галактика”, “Інфософт”, “NS2000” і “ABACUS Financial”, творці яких (корпорація “Галактика”, фірма 55 “Інфософт”, фірма “Нікос-Софт” і фірма “Омега”) мають сертифікати розроблювачів інтегрованих управлінських систем. Якісна характеристика ефективності застосування виділених видів систем залежно від особливостей підприємств, дивитись(табл 2)

Таблиця 2 – застосування видів систем залежно від особливостей підприємств

Локальні системи				
Середні інтегровані системи				
Великі інтегровані системи				
	Малі підприємства	Підприємства без виробництва (торгівля, послуги)	Виробництво	Великі фінансовопромислові структури

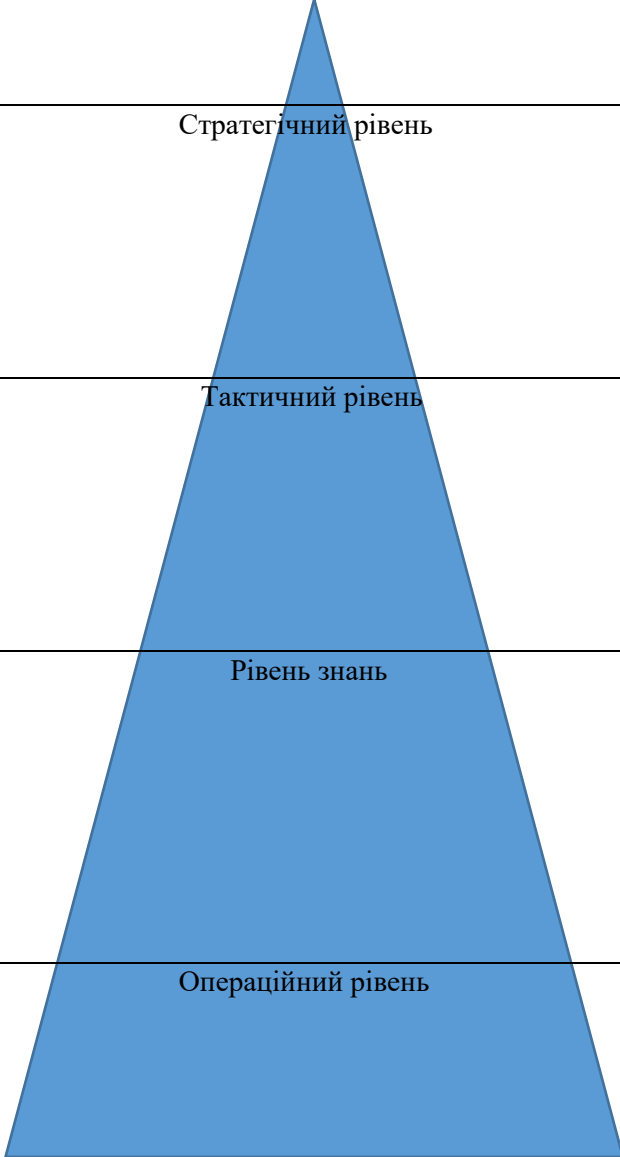
Джерело: сформовано автором на основі виконаного дослідження

На даній таблиці, ми бачимо, на скільки розмір системи, грає велику роль у впровадженні будь якої ІУС, і головне правильно обирати необхідний тип обсягу

системи, щоб не потрапити в категорію ризиків не можливості обслуговуванні інформаційної системи

Великі інтегровані системи відрізняються від середніх набором вертикальних ринків і глибиною підтримки процесів управління великими багатофункціональними групами підприємств (холдингами або фінансовопромисловими угрупованнями). Такі системи мають найбільшу функціональність, включно з управлінням виробництвом, управлінням складними фінансовими потоками, корпоративною консолідацією, глобальним змістуванням і бюджетуванням тощо. Схожі функції наявні і в багатьох фінансово-управлінських (за винятком виробництва) і середніх інтегрованих системах, однак із нижчим ступенем опрацювання. Терміни впровадження великих інтегрованих систем зазвичай займають понад рік, а вартість проєкту – понад \$500 000.

Таблиця 3, - впровадження видів систем залежно від особливостей підприємств



СППРС – системи підтримки прийняття стратегічних рішень	Стратегічний рівень	Менеджери вищої ланки
ІУС – управлінські інформаційні системи СППР – системи підтримки прийняття рішень	Тактичний рівень	Менеджери середньої ланки
СРСЗ – системи роботи зі знаннями Офісні системи	Рівень знань	Робітники, що оперують даними, знаннями
СОТ – системи обробки транзакцій	Операційний рівень	Менеджери операціоністи

Джерело: сформовано автором на основі виконаного дослідження

Ця класифікація умовна. Низка представлених на ринку систем за своїми функціональними можливостями, технічними особливостями, термінами впровадження, вартості та іншими параметрами може належати до різних класів представленої класифікації

У сучасному світі інформаційні технології відіграють визначальну роль у функціонуванні підприємств та організацій усіх галузей. Зокрема, інформаційні управляючі системи (ІУС) стали невід'ємною складовою частиною сучасного управління, забезпечуючи швидку, ефективну та цілісну обробку інформації для прийняття управлінських рішень. Їх можна охарактеризувати кількома класифікаціями, наприклад:

За сферою застосування: фінансові системи, логістичні системи, HRM системи, CRM системи, ERP системи. Кожна з них приносить свій великий вклад в роботу системи.

За масштабом:

- Малий бізнес: ІУС для невеликих підприємств або підприємців.
- Середній бізнес: ІУС для середніх підприємств зі складнішими потребами управління.
- Великі підприємства: ІУС для корпорацій або міжнародних компаній з великим обсягом операцій та персоналу.

За типом платформи:

- Локальні системи: Встановлюються та обслуговуються безпосередньо на пристроях або серверах компанії.
- Хмарні системи: Зберігання та обробка даних відбувається на віддалених серверах, а доступ до системи здійснюється через Інтернет.
- Гібридні системи: Комбінація локальних та хмарних рішень, що дозволяє підприємствам використовувати переваги обох підходів.

За функціональністю:

- Адміністративні системи: Управління ресурсами та операціями компанії.
- Аналітичні системи: Аналіз та звітність за допомогою різних видів аналізу даних.
- Системи підтримки прийняття рішень: Допомога в ухваленні стратегічних рішень на основі аналізу даних та моделей.

Це лише кілька з можливих класифікацій Інформаційних Управляючих Систем, і вони можуть варіюватися в залежності від конкретної області застосування та потреб користувачів.

Вони мають велику кількість різних варіацій, так як:

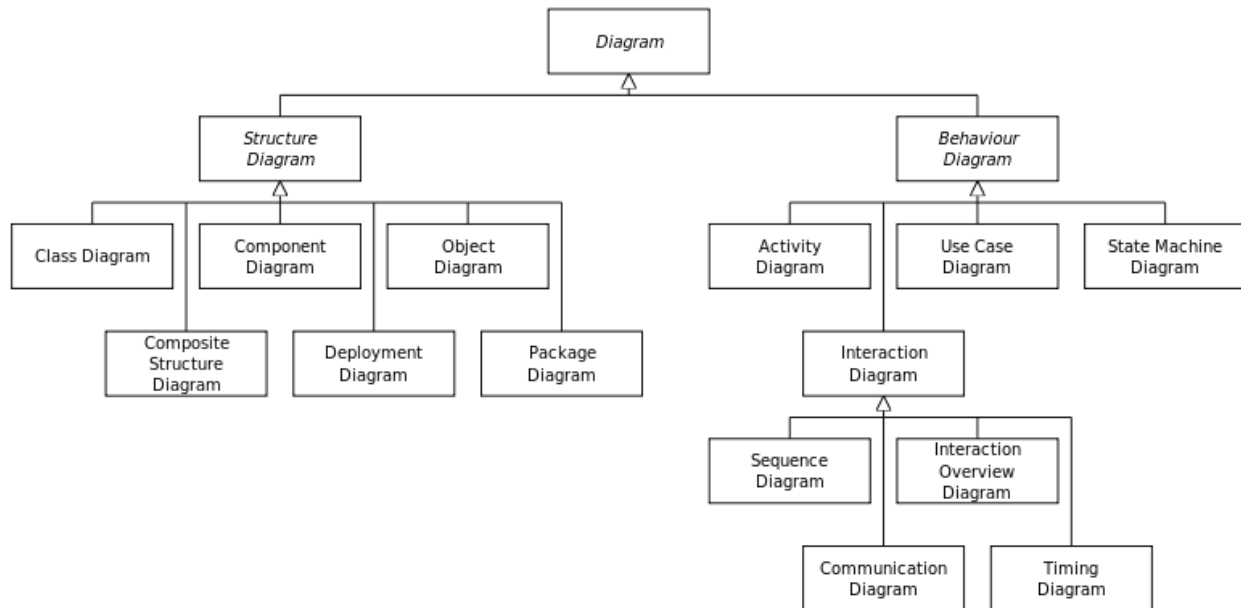


Рисунок 3 – Діаграми класів або сутностей-зв'язків

Джерело: https://uk.m.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Uml_diagram2.png

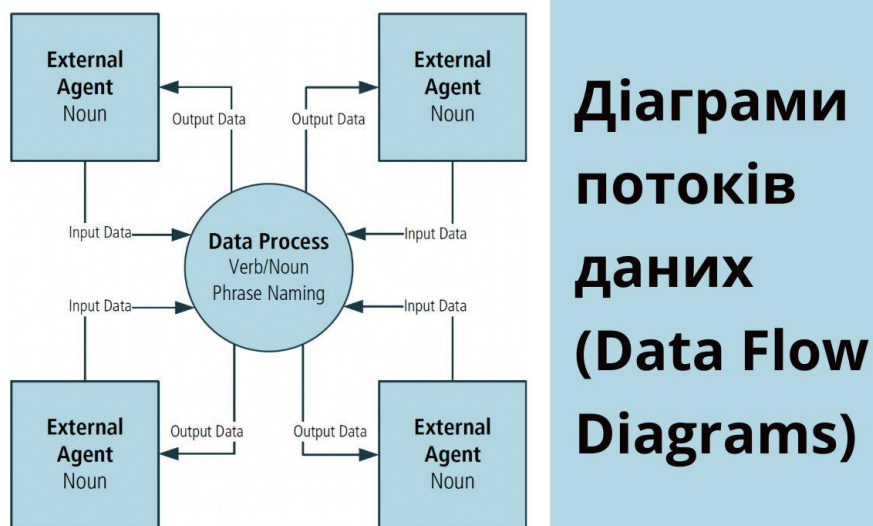


Рисунок 4– Діаграми потоку даних (DFD)

Джерело: <https://www.maxzosim.com/page/3/>

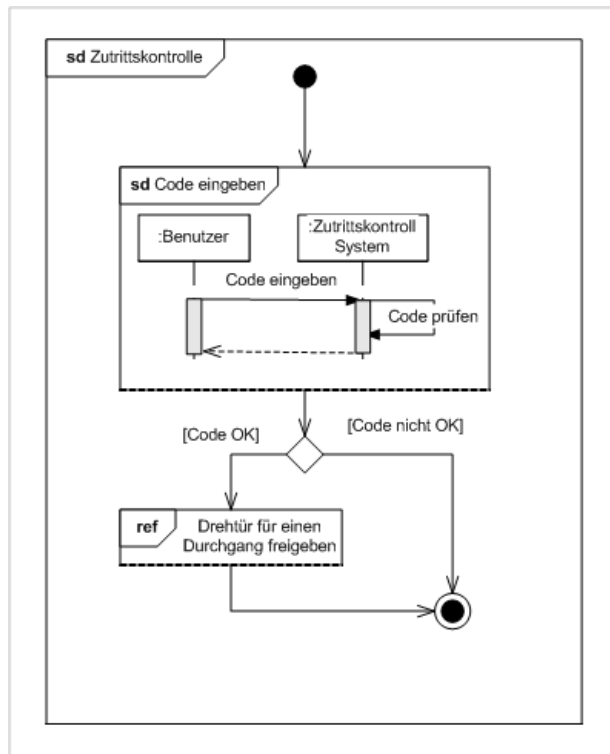


Рисунок 5 – Діаграми сеансів або взаємодії (UML)

Джерело: <https://uk.m.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Iau-diagramm-1.png>

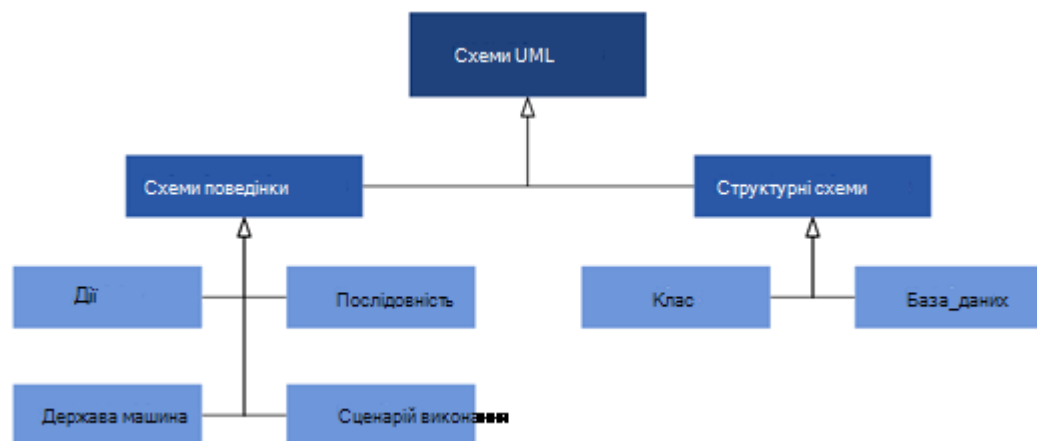


Рисунок 6 – Діаграми структури програми (UML)

Джерело: <https://support.microsoft.com/ukua/topic/%D1%81%D1%85%D0%B5%D0%BC%D0%B8-uml-%D1%83-visio-ca4e3ae9-d413-4c94-8a7a-38dac30cbcd6>

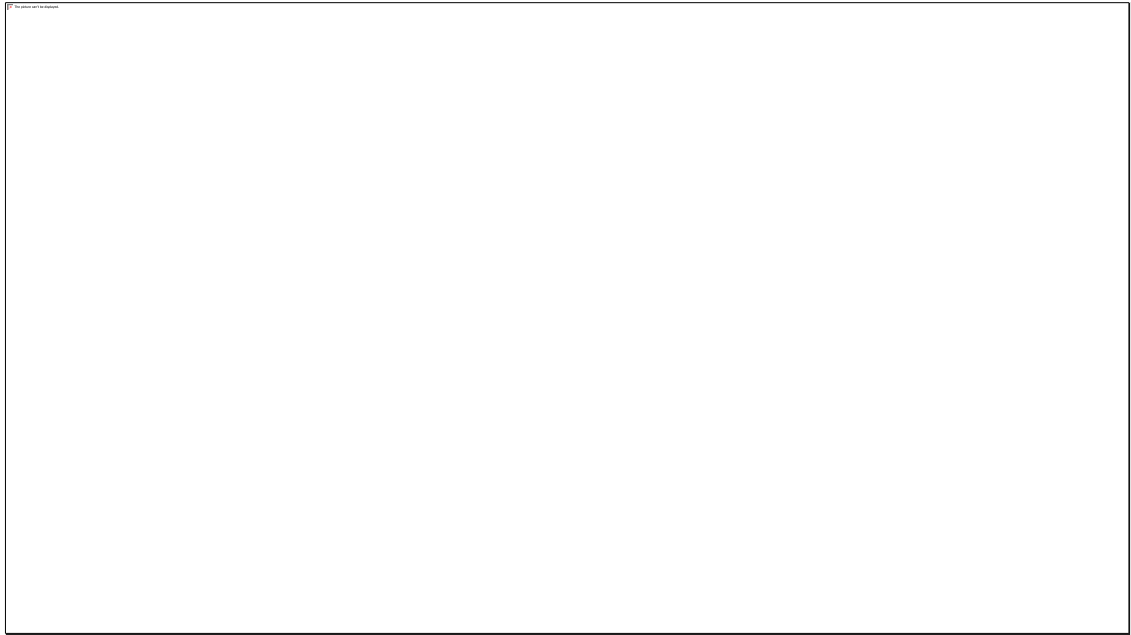


Рисунок 7 – Архітектурні схеми

Джерело: <https://devzone.org.ua/post/nayvazlyvishi-arkhitekturni-shablony-iaki-neobkhidno-znaty>

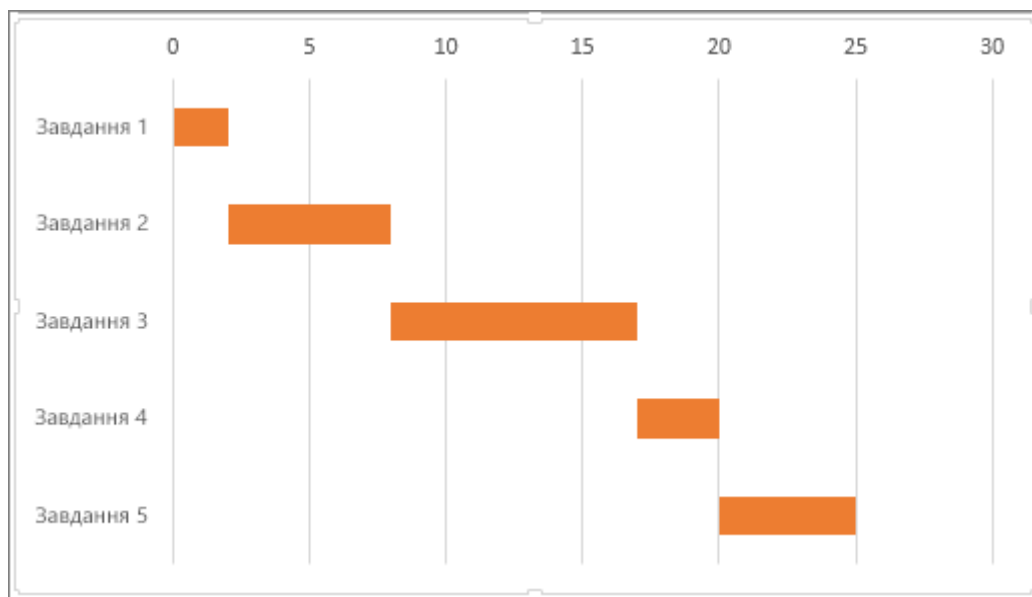


Рисунок 8 – Діаграми Ганта або діаграми Робінсона

Джерело: розроблено автором на основі



Рисунок 9 – Графіки та діаграми для аналітичних даних:

Джерело: <https://www.templatemonster.com/ru/infographic-elements/?page=13>

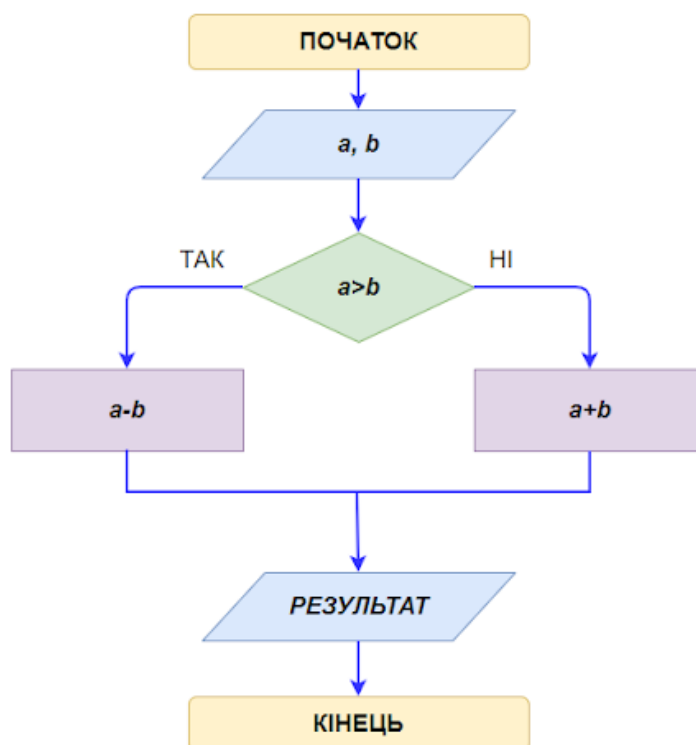


Рисунок 10 – Діаграми алгоритмів

Джерело: <http://nikolay.in.ua/distantsijne-navchannya/5-klas/818-algorithm-vikonavtsi-algoritmiv-ta-jikhni-sistemi-komand>

1.2 Обґрунтування вибору підходів і технологій для створення інформаційної управляючої системи

При виборі підходів та технологій для створення інформаційної управляючої системи (ІУС) важливо керуватися рядом критеріїв, які забезпечать ефективність, надійність та зручність у використанні системи. Нижче ми описуємо концептуальні рішення та обґрунтуємо їх вибір за допомогою аналізу та оцінки за визначеними критеріями.

agile-модель

Я вирішив реалізувати ІУС за допомогою гнучкої методології замість монолітної. Це дозволить нам розділити функціонал системи на окремі сервіси, що забезпечить більшу гнучкість у розробці та підтримці, а також полегшить масштабування окремих компонентів системи. тому було вирішено використати agile-модель.

Обрані підходи та технології відповідають вимогам до ІУС та дозволяють забезпечити високий рівень функціональності, ефективності та надійності. Вони також забезпечують гнучкість та легкість у масштабуванні та підтримці системи, що робить їх оптимальними вибором для нашого проекту.

Цей підхід дозволить нам створити сучасну та ефективну інформаційну управляючу систему, яка задовольнить потреби нашої організації та забезпечить її успішну діяльність у майбутньому.

Напрочут, у «гнучкій» методології розробки після кожної ітерації замовник може спостерігати результат і розуміти, чи задовольняє він його чи ні, зміни легко внести.

Даний тип методу гнучкої розробки, націлений на довготривалі проєкти, які можуть змінюватись на вчасно адаптуватись.

Гнучка розробка добре підходить під малі підприємства, бо вони можуть досить швидко змінитись і потребувати внесення відповідних змін. Така методологія дозволяє навіть за чітких вимог реалізувати проєкт, адже в процесі допустимі невеликі відхилення та зміни, на відміну від інших не «гнучких» методологій, де невідповідність вимогам говорить про те, що проєкт потрібно переробляти.

Agile-методології дуже ефективні для розробки інформаційно-управляючих систем (ІУС), а саме, торгової компанії. Такий підхід дозволяє гнучко адаптуватися до змінних вимог бізнесу, швидко впроваджувати нові функції та забезпечувати високу якість розроблених рішень. Ключові переваги Agile полягають у можливості регулярного зворотного зв'язку з клієнтом, ітераційному підході до розробки, швидкому реагуванні на зміни на ринку та постійному зосередженні на підвищенні ефективності та якості процесу

розробки. Такий підхід дозволяє торговим компаніям ефективно розвивати та підтримувати ІУС, забезпечуючи їм конкурентні переваги на ринку та сприяючи їхньому успіху.

Схема циклу розробки у «гнучкій» методології проектування та розробки представлена на рисунку 1.1.

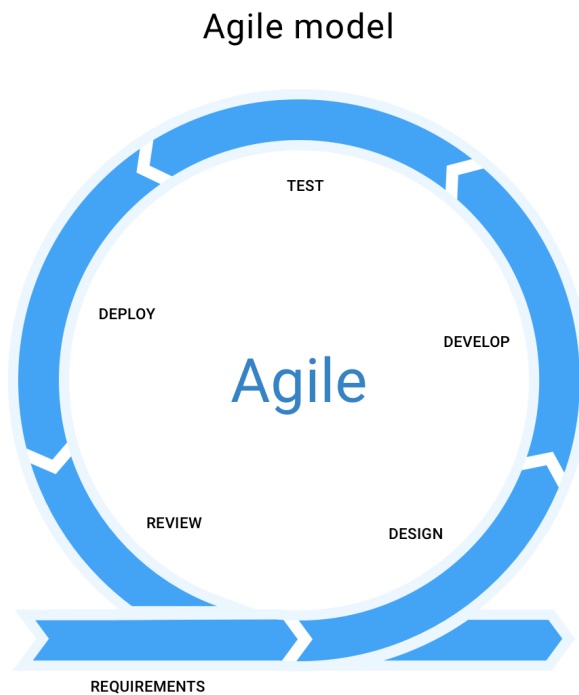


Рисунок 10 – Діаграми алгоритмів

Джерело: {24}

За допомогою Agile-моделі проектування різноманітні програмні продукти та проекти, які потребують гнучкого та ітеративного підходу до розробки. Ось деякі приклади проєктів, які часто реалізуються за допомогою Agile-методологій: розробка програмного забезпечення, веб-розробка, мобільна розробка, розробка ігор, корпоративні рішення та інноваційні проекти. Агільний підхід дозволяє ефективно реалізовувати ці проекти, забезпечуючи гнучкість, швидкість реакції на зміни та високу якість продукту.

РОЗДІЛ 2 «ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ТА МЕТОДИ І МОДЕЛЬ»

2.1 «Структура і характеристика системи»

В більшості випадків, малі та середні підприємства, не мають широків обсягів бюджету, та подальших масштабних розширень, це варто враховувати при розробці програмного забезпечення, адже ІУС, в більшості випадків, належить саме до таких видів підприємств.

Тому в першу чергу, треба враховувати ці фінансові можливості, і розробити систему, яка не буде надто складною і виконуватиме поставлені задачі.

По-друге, треба брати до уваги той факт, що підприємства не стоять на місці, а активно розвиваються і масштабування системи, це лише питання часу.

Одним із важливих факторів, є безпека системи, щоб вона справно могла фіксувати, запам'ятовувати данні, використовувати накопичений матеріал, та у випадку збоїв, можна було відновити втрачену інформацію. Це значно підвищить рейтинг довіри системи, що може стати фактором подальшого розвитку та більш вигідного фінансування на подальший розвиток.

При виборі підходів та технологій для створення інформаційної управляючої системи (ІУС) важливо керуватися рядом критеріїв, які забезпечать ефективність, надійність та зручність у використанні системи.

Слід зауважити, що ІУС це дуже широкий спектр областей, з якими працює система, у кожній з них, є свої умови використання, переваги та комплексний підхід, але все це зазвичай підлаштовується індивідуально під кожен компанію.

Один з прикладів компанії, яка зазнала успіху завдяки впровадженню інформаційної управляючої системи (ІУС), - це Amazon.

Amazon використовує свою ІУС для ефективного управління величезним обсягом даних, логістики, складського управління, взаємодії з клієнтами та інших аспектів їх бізнесу. Наприклад, їхній інтегрований підхід до управління логістикою та постачанням дозволяє їм забезпечувати швидку доставку замовлень своїм клієнтам. ІУС також допомагає Amazon аналізувати дані про покупців та їхні вподобання, що дозволяє створювати персоналізовані рекомендації та пропозиції для кожного користувача. Крім того, ІУС Amazon забезпечує безпеку та захист даних, що є критичним аспектом в сучасному електронному комерції.

Завдяки ефективному використанню ІУС, Amazon зміг стати однією з найбільших та найуспішніших торгових компаній у світі, надаючи високоякісні

послуги своїм клієнтам та швидко реагуючи на зміни на ринку. Це є яскравим прикладом того, як не велика компанія змогла досягти великого успіху, завдяки впровадженню ІУС.

Для бізнес компаній, які користуються системою, визначення цілей є критично важливим, оскільки вони визначають результати, яких вони прагнуть досягти, і визначають напрямок для досягнення вищих стратегічних цілей. Ця схема цілей-мотиваторів, відображає цілі, які підприємство ставить перед собою, та узагальнені вимоги, яким повинна відповідати проектована система. Візьмемо дані функціональні вимоги, які підійдуть для компанії з не великим бюджетом, але з чіткими поставленими задачами

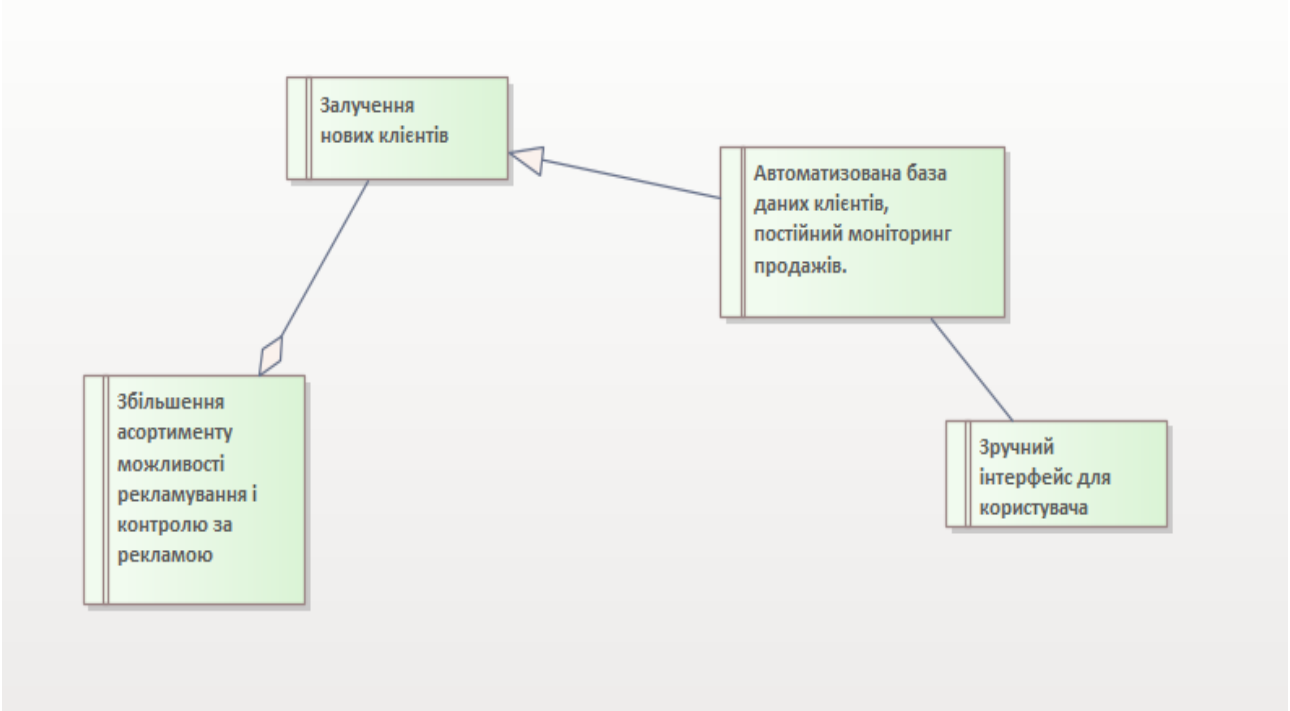
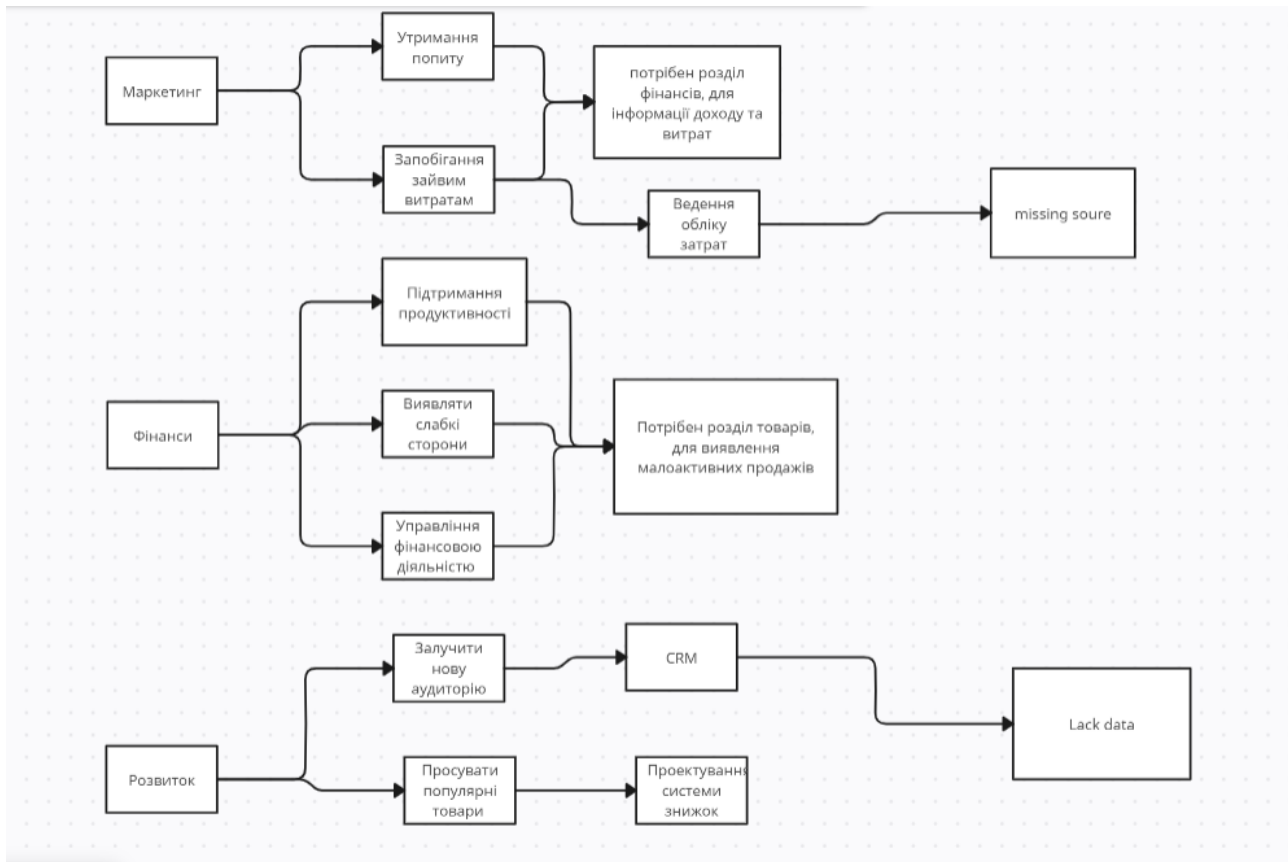


Рис діаграма вимог

Item	Stereotype	Status	Difficulty	Priority
☒ Авторизація в систему, коли вірно введені дані і видавати помилку, коли введені дані неправильно	FunctionalRequirement	Proposed	Medium	Medium
☒ Функція "Змінити мову" перемикання мови(мінімум три мови: Українська, Англійська, російська)	FunctionalRequirement	Proposed	Medium	Medium
☒ Функція "Поповнити особистий рахунок" - поповнення особистого рахунку за рахунок власних коштів	FunctionalRequirement	Proposed	Medium	Medium
☒ Функція "Пошук" по критеріям(за фільтром) або за ключовими словами	FunctionalRequirement	Proposed	Medium	Medium
☒ Функція "Технічна підтримка"	FunctionalRequirement	Proposed	Medium	Medium
☒ Функція(Можливість) сплатити за рекламу з особистого рахунку	FunctionalRequirement	Proposed	Medium	Medium
☒ Функція "Зворотній зв'язок" щоб клієнт(замовник) міг зв'язатися в разі проблеми з компанією	FunctionalRequirement	Proposed	Medium	Medium

Рис Специфікація вимог

Бізнес вимоги представлені спільно з бізнес-цілями, вимогами та ризиками



(рис) Діаграма бізнес-вимог до системи

Функціональні вимоги представлені на рисунку 2.4, разом з системою до якої належать і ризиками для них.

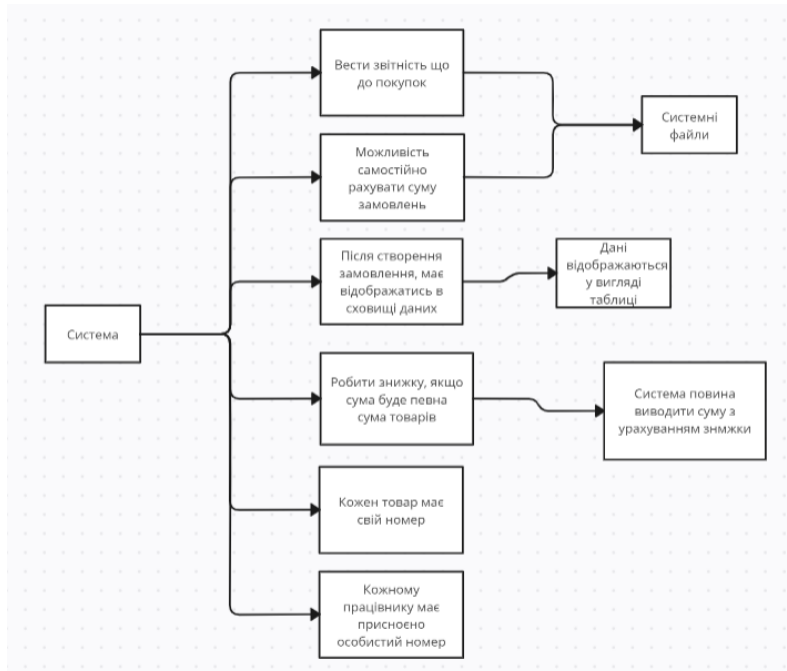


Рисунок 2.4 – Діаграма функціональних вимог до системи

Джерело: розроблено автором

На діаграмі зліва представлена система до якої належать вимоги, а справа – функціональні вимоги та ризики для них.

Нефункціональні вимоги грають важливу роль у системі, та загалом описують певні моменти, такі як зовнішній вигляд, безпеку, доступність та відновлюваність системи, обмеження викликані тим чи іншим блоком чи модулем, тому їх децю більше.

Нефункціональні вимоги представлені на рисунку 2.5, тут кожна з вимог представляє собою квадрат с описом

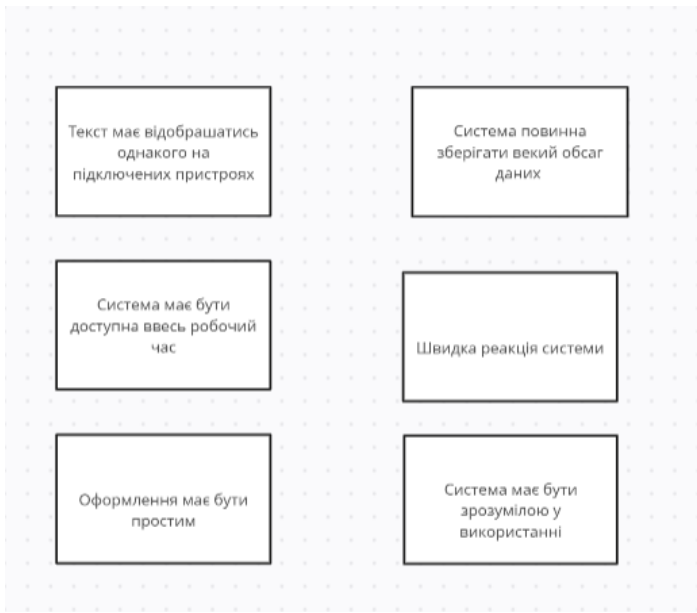


Рисунок 2.5 – Діаграма нефункціональних вимог до системи

Джерело: розроблено автором

На діаграмі представлена система до якої належать нефункціональні вимоги.

Структура системи складається з безлічі підзадач, які потрібно виконувати і вимог, які потрібно досягти.

Діаграми розгортання(Deployment Diagrams) відображають фізичні взаємозв'язки між програмними і апаратними компонентами розроблюваної системи, а також дозволяють відобразити маршрути переміщення об'єктів у розподіленій системі.

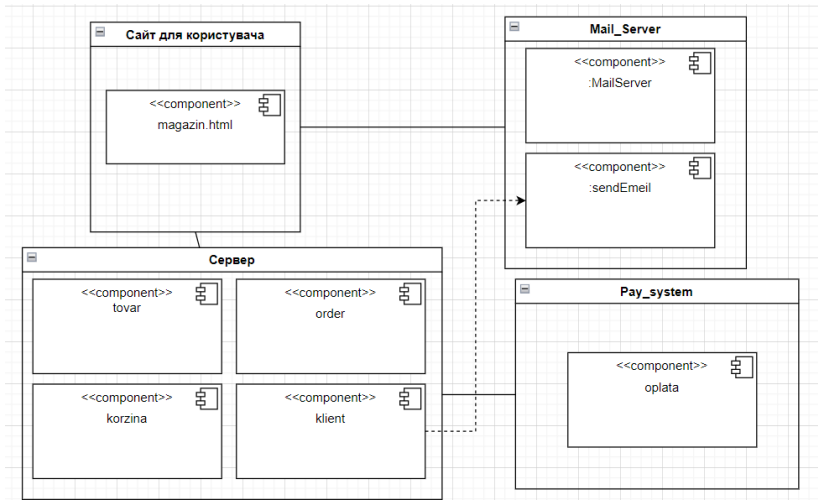


Рисунок 2.5 – Діаграма розгортання

Джерело: розроблено автором

При створенні програмної системи недостатньо відповісти на питання що робить система? і з чого вона складається? – потрібно відповісти на питання як працює система?.

Відповідь на це питання дає модель поведінки. Поведінка реальної програми цілком і повністю визначається її кодом – як програма складена, так вона і виконується. Але програма – це просто запис алгоритму. Таким чином, модель поведінки (behavior model) – це опис алгоритму роботи системи. Поведінка системи визначається множиною об'єктів, що обмінюються повідомленнями, і задається діаграмами діяльності, діаграмами взаємодії, діаграмами кінцевого автомата. Для опису потоку управління, тобто послідовності виконуваних елементарних кроків при виконанні окремої операції або реалізації складного варіанта використання, зручно використовувати діаграму діяльності. Взаємодія декількох програмних об'єктів між собою описується діаграмами взаємодії в одній з двох еквівалентних форм (діаграми послідовності або діаграми комунікації).

У UML передбачено декілька різних засобів для опису поведінки. Вибір того чи іншого засобу диктується типом поведінки, яку потрібно описати. Для опису життєвого циклу конкретного об'єкта, поведінка якого залежить від історії цього об'єкта, або обробки асинхронних стимулів, використовується кінцевий автомат у формі діаграми станів. При цьому стани кінцевого автомата відповідають станам об'єкта, тобто різним наборам значень атрибутів, а переходи відповідають виконанню операцій. Як і діаграми видів діяльності, діаграми кінцевих автоматів (станів) UML відображають динамічну модель системи. За їх допомогою ілюструються події і стани об'єктів – транзакцій, прецедентів, людей тощо.

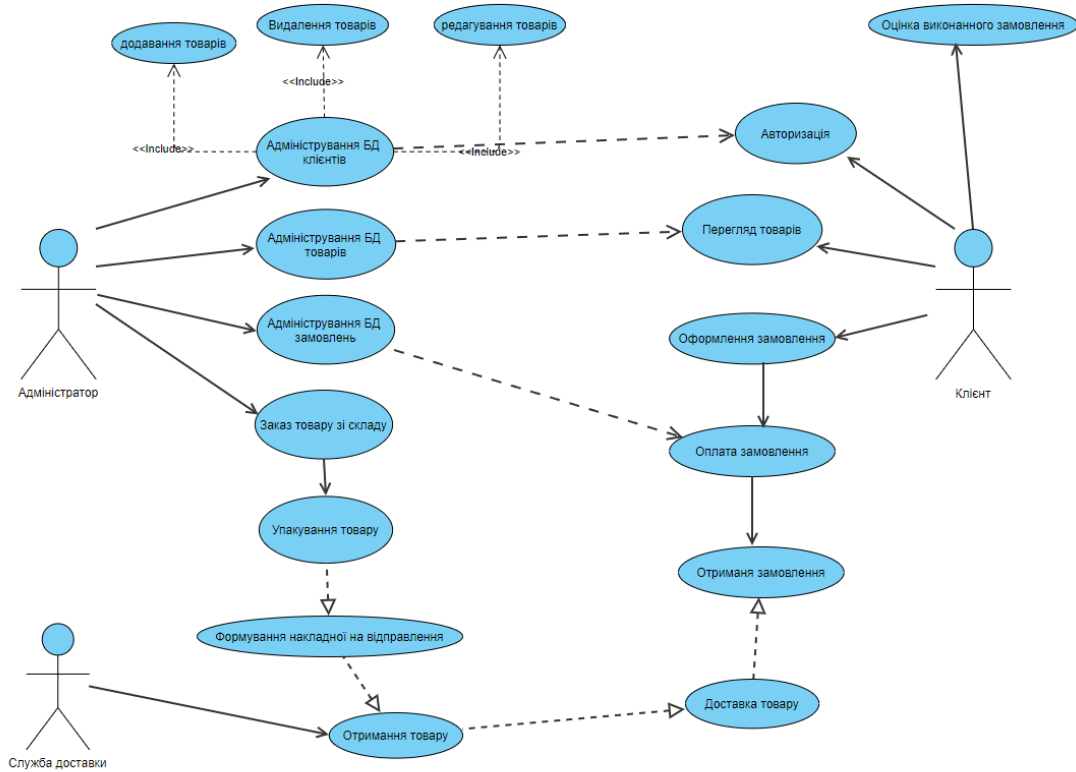


Рисунок 2.5 – Діаграма прецедентів

Джерело: розроблено автором

Торгова компанія може працювати як у офлайн режимі, так і онлайн , що значно розширює можливості продажу та підвищує зручність для клієнтів. В цьому випадку, інформаційна система компанії повинна підтримувати як офлайн, так і онлайн процеси. Нижче розглянемо детально процеси, які забезпечують функціонування торгової компанії в обох режимах.

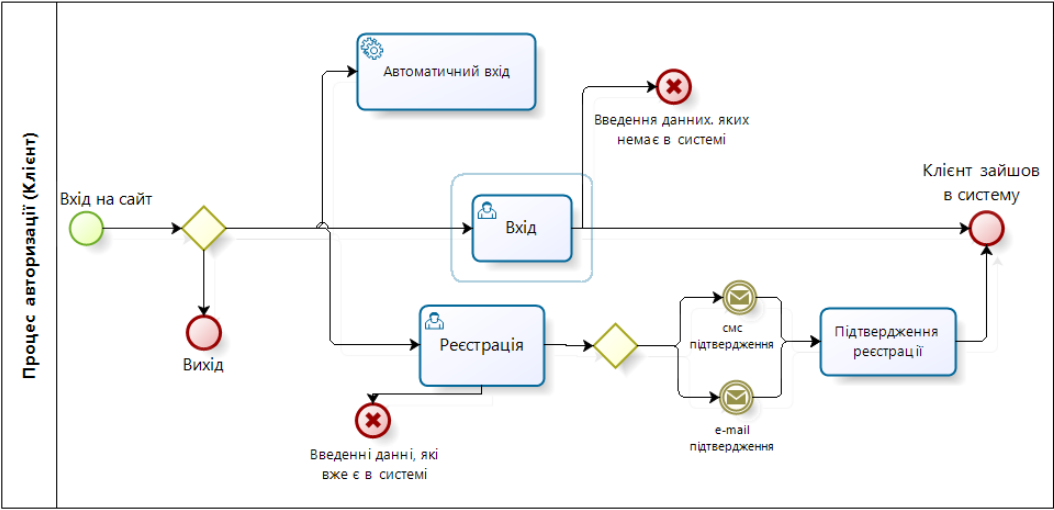


Рисунок 11 – процес авторизації

Джерело: розроблено автором

Онлайн:

Інтернет-магазин має цифровий каталог товарів, який відображається на веб-сайті або в мобільному додатку. Клієнти можуть переглядати товарні картки з описами, цінами, фотографіями та відгуками.

Управління каталогом здійснюється через централізовану базу даних, що забезпечує актуальність інформації як в офлайн, так і в онлайн режимах.

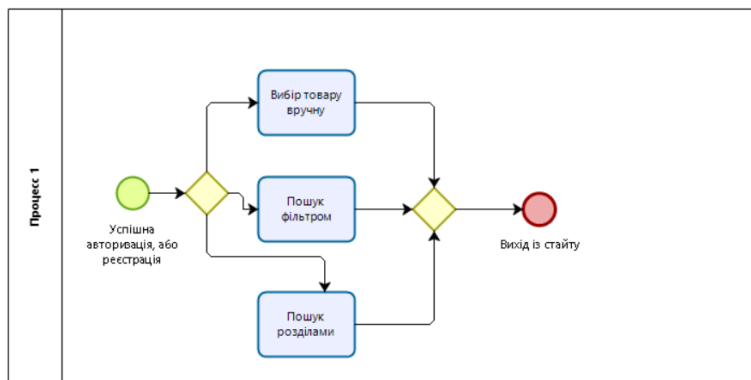


Рисунок 12 – вхід в додаток

Джерело: розроблено автором

Система інтегрується з інтернет-магазином, відстежуючи доступність товарів для онлайн-замовлень. Запаси оновлюються в режимі реального часу, запобігаючи продажу товарів, яких немає в наявності.

Замовлення та оплата

Компанія приймає замовлення через веб-сайт або мобільний додаток. Оплата здійснюється онлайн за допомогою банківських карт, електронних гаманців або інших платіжних систем.

Компанія може пропонувати доставку товарів до пункту видачі клієнта, або можливість самовивозу з найближчого магазину.

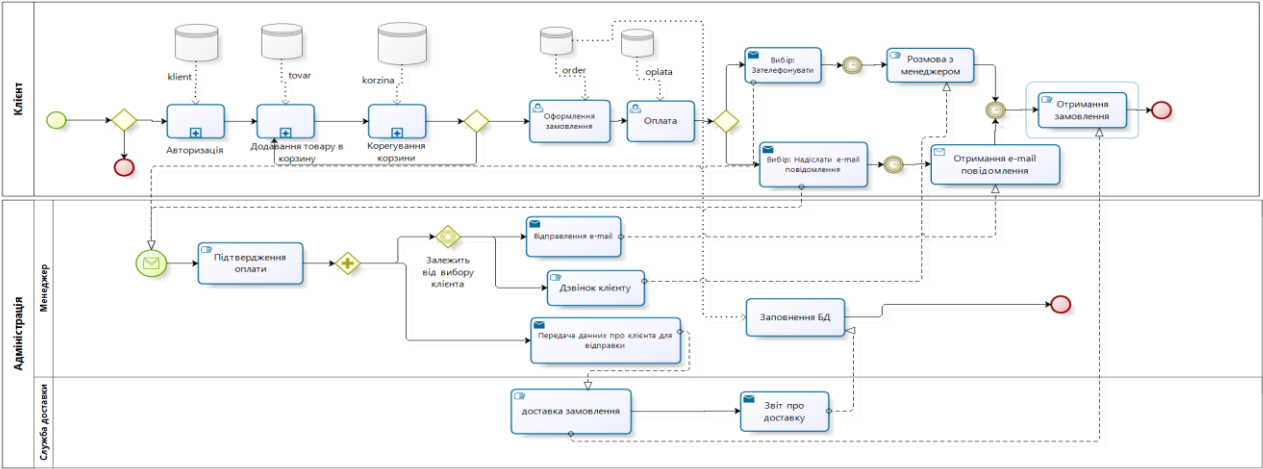


Рисунок 13 – Діаграма діяльності **BPMN**

Джерело: розроблено автором

Наступни рисунок показує там, як продить процес отримання замовлення та просес звязку з службою доставки

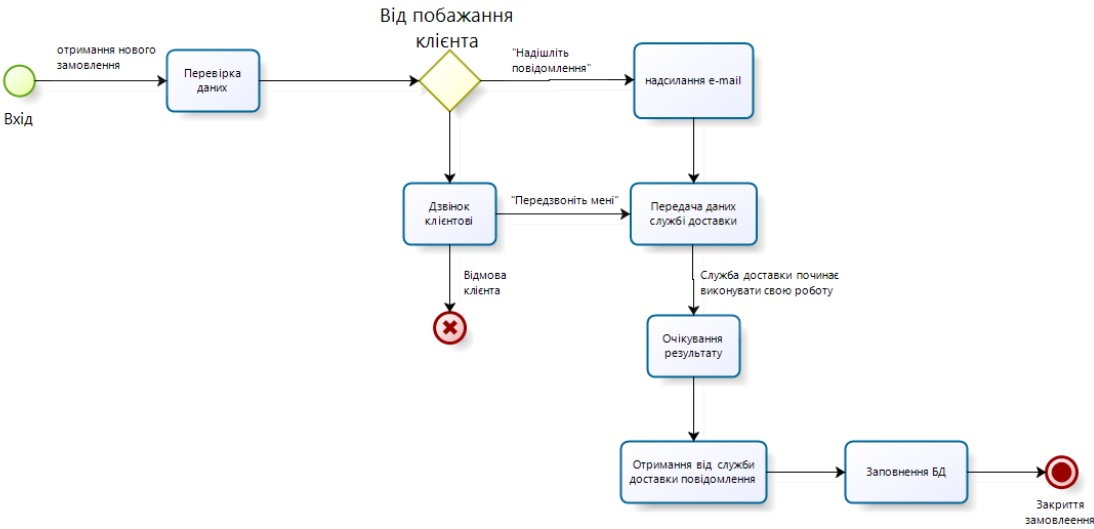


Рисунок 14 – підтвердження замовлення

Джерело: розроблено автором

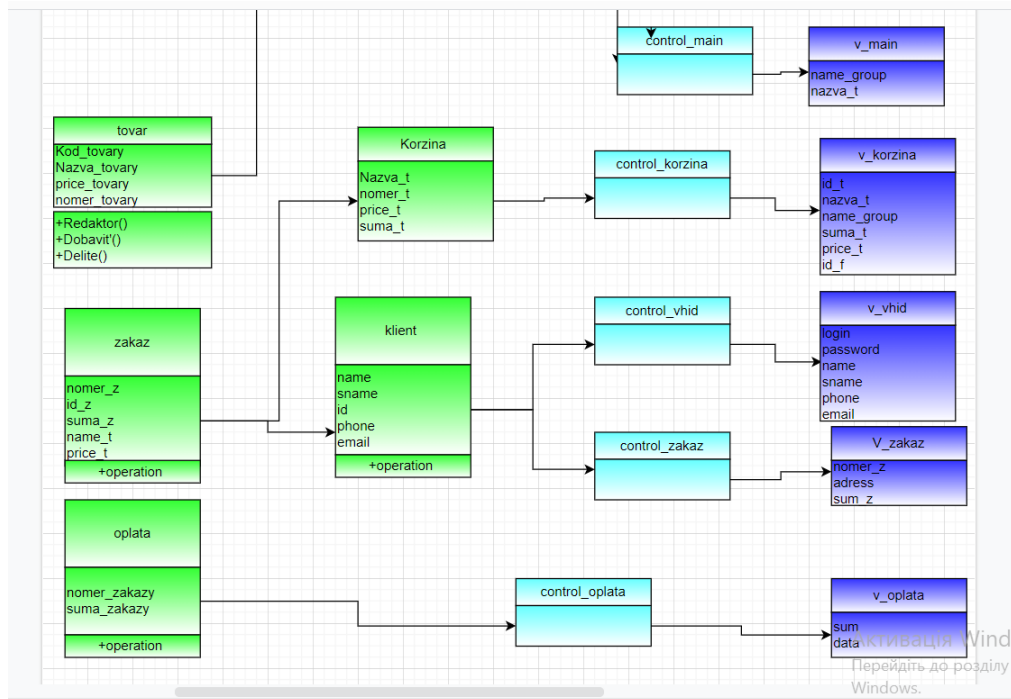


Рисунок 15– Діаграма “Послідовності”: Діаграма “Класів”

Джерело: розроблено автором

Переваги Офлайн Продажів

- Незважаючи на можливість роботи як інтернет-магазин, торгова компанія може надавати перевагу офлайн продажам з наступних причин:
- Безпосередній контакт з клієнтами: Можливість особистого спілкування з клієнтами дозволяє краще розуміти їхні потреби та будувати довірчі відносини.
- Огляд та тестування товарів: Клієнти можуть особисто оглянути товари, спробувати їх в дії перед покупкою, що особливо важливо для складних або дорогих товарів.
- Негайне отримання товарів: Клієнти можуть одразу забрати товари з магазину, без необхідності чекати доставки.
- Регіональні особливості: Фізичні магазини можуть враховувати місцеві уподобання та специфіку ринку, пропонуючи відповідні товари та послуги.

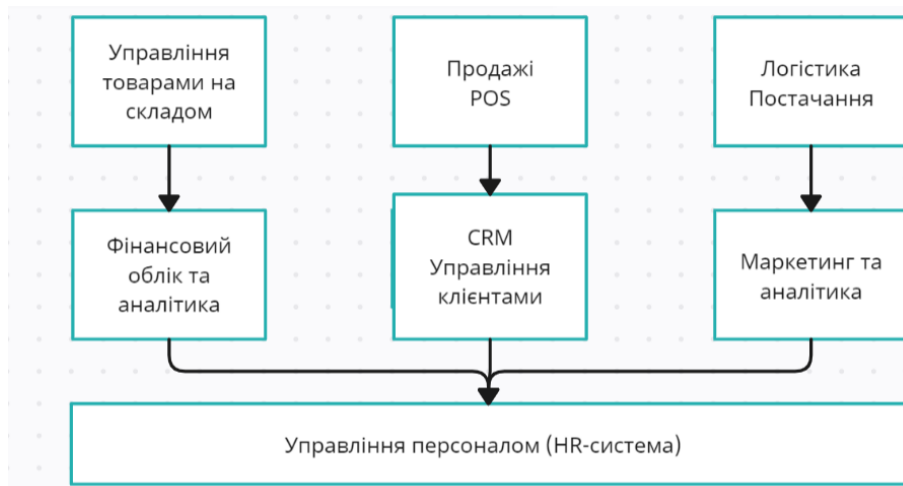


Рисунок 15 Схема структурної організації ІС торгової компанії
Джерело: розроблено автором

Схема бізнес-процесів ІС:

Ця діаграма показує основні бізнес-процеси, автоматизовані в ІС торгової компанії, включаючи закупівлі, управління запасами, продажі, обробку замовлень, доставку, фінансовий облік, та управління взаємовідносинами з клієнтами. Діаграма ілюструє, як інформаційна система допомагає оптимізувати та синхронізувати ці процеси, забезпечуючи безперебійну роботу компанії.

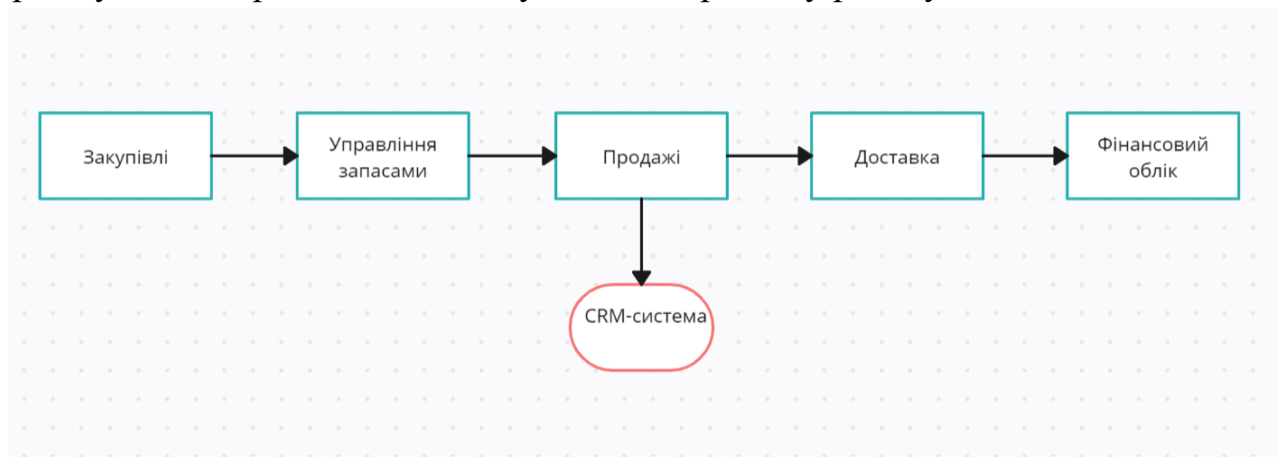


Рисунок 16 Схема бізнес-процесів ІС
Джерело: розроблено автором

Схема управління запасами

На цій схемі зображено процес управління запасами в ІС торгової компанії. Вона показує, як система обробляє інформацію про наявні запаси, визначає необхідність поповнення та створює замовлення на закупівлю товарів. Також відображається інтеграція з іншими модулями системи, такими як фінансовий облік та постачання.

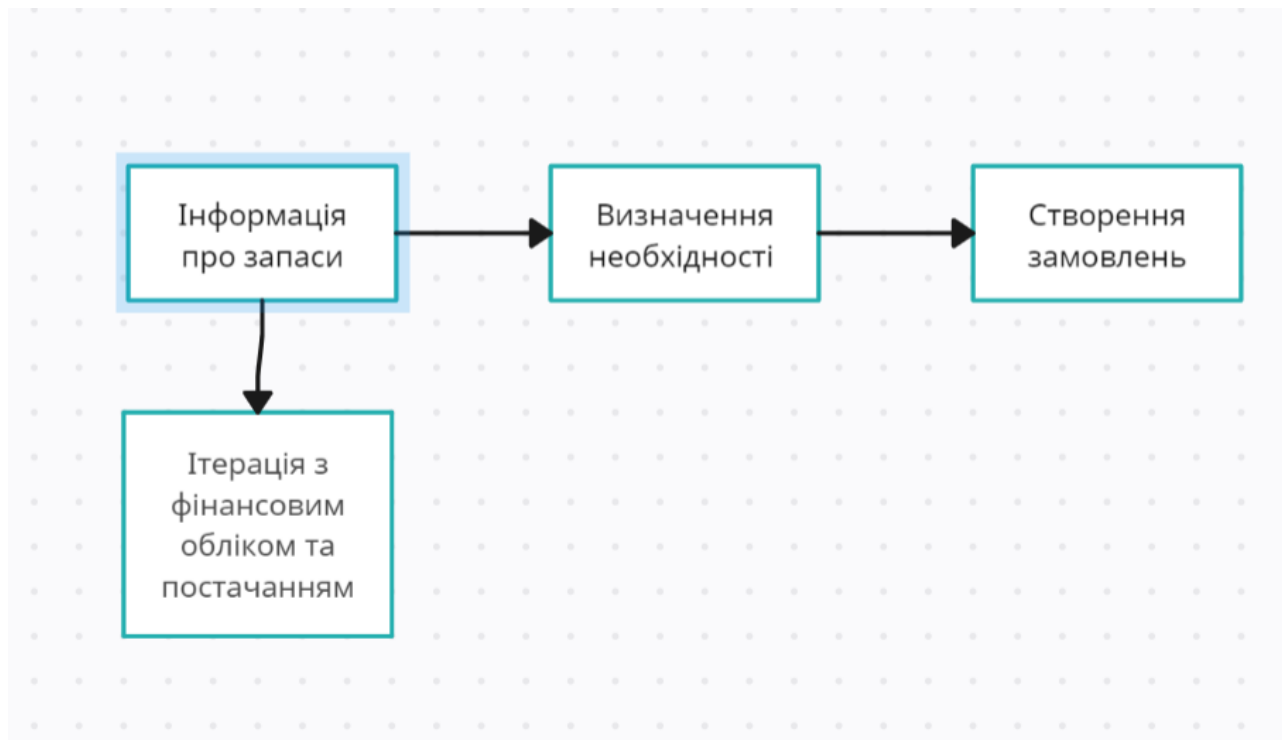


Рисунок 17 Схема управління запасами

Джерело: розроблено автором

Схема роботи CRM-системи

Цей малюнок демонструє функціональні можливості CRM-модуля в ІС торгової компанії. Схема показує, як CRM-система обробляє дані про клієнтів, підтримує програми лояльності, управляє маркетинговими кампаніями та надає інструменти для аналізу клієнтських даних. Це дозволяє компанії ефективно взаємодіяти з клієнтами та підвищувати рівень їх задоволеності.

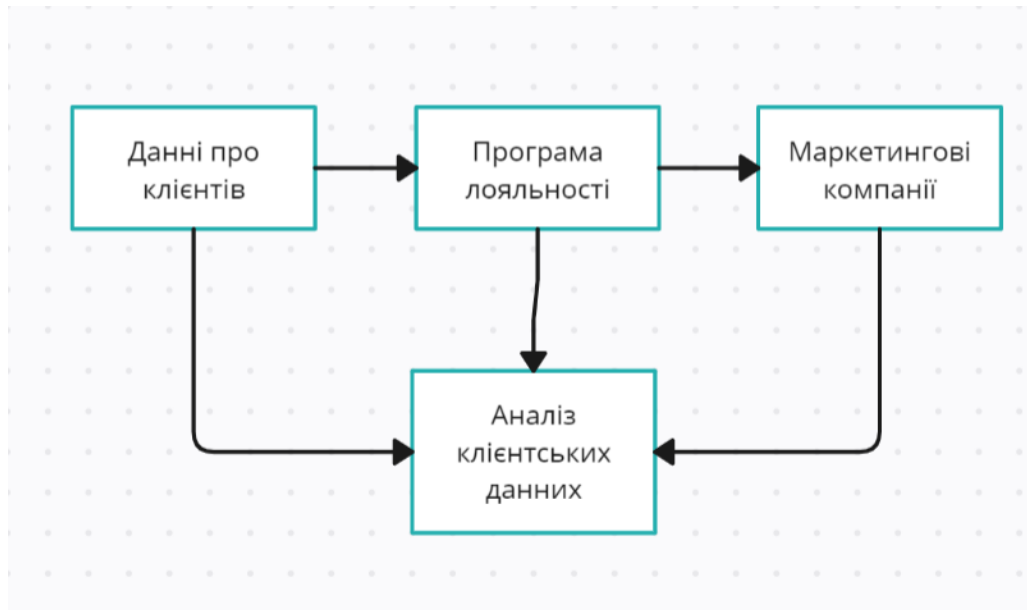


Рисунок 18 Схема роботи CRM-системи

Джерело: розроблено автором

Схема фінансового обліку

Цей малюнок ілюструє процес фінансового обліку в ІС торгової компанії. Схема відображає основні компоненти фінансового модуля, включаючи управління рахунками, облік доходів та витрат, звітність та фінансову аналітику. Взаємодія з іншими модулями системи, такими як управління запасами та продажі, дозволяє забезпечити точність та актуальність фінансової інформації.



Рисунок 19 Схема фінансового обліку

Джерело: розроблено автором

Схема логістики та постачання

На цій схемі представлено процес управління логістикою та постачанням в ІС торгової компанії. Схема показує, як система допомагає планувати та координувати поставки товарів, оптимізувати маршрути доставки та управляти автопарком. Взаємодія з модулями управління запасами та фінансового обліку забезпечує ефективне управління логістичними процесами.

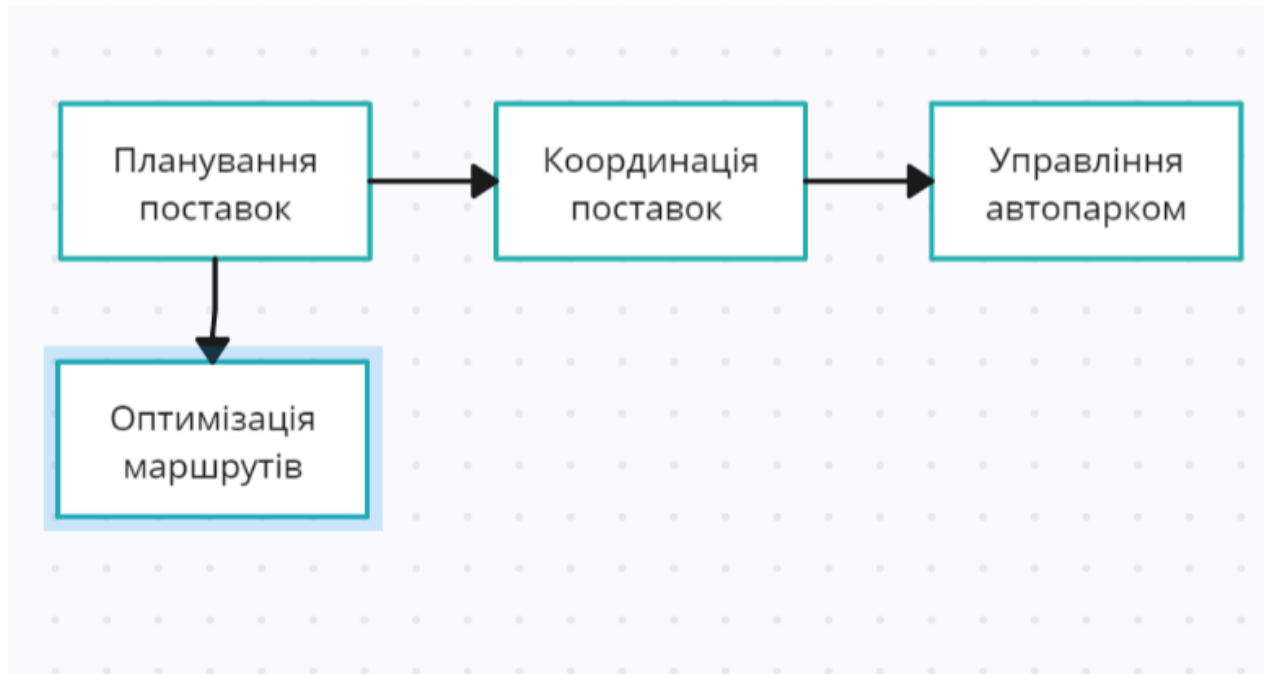


Рисунок 20 Схема логістики та постачання

Джерело: розроблено автором

Схема управління персоналом (HR)

Цей малюнок демонструє функціональні можливості HR-модуля в ІС торгової компанії. Схема відображає процеси обліку працівників, управління робочим часом, планування змін, навчання та розвитку персоналу. Інтеграція з іншими модулями системи, такими як фінансовий облік та CRM, дозволяє забезпечити ефективне управління людськими ресурсами.

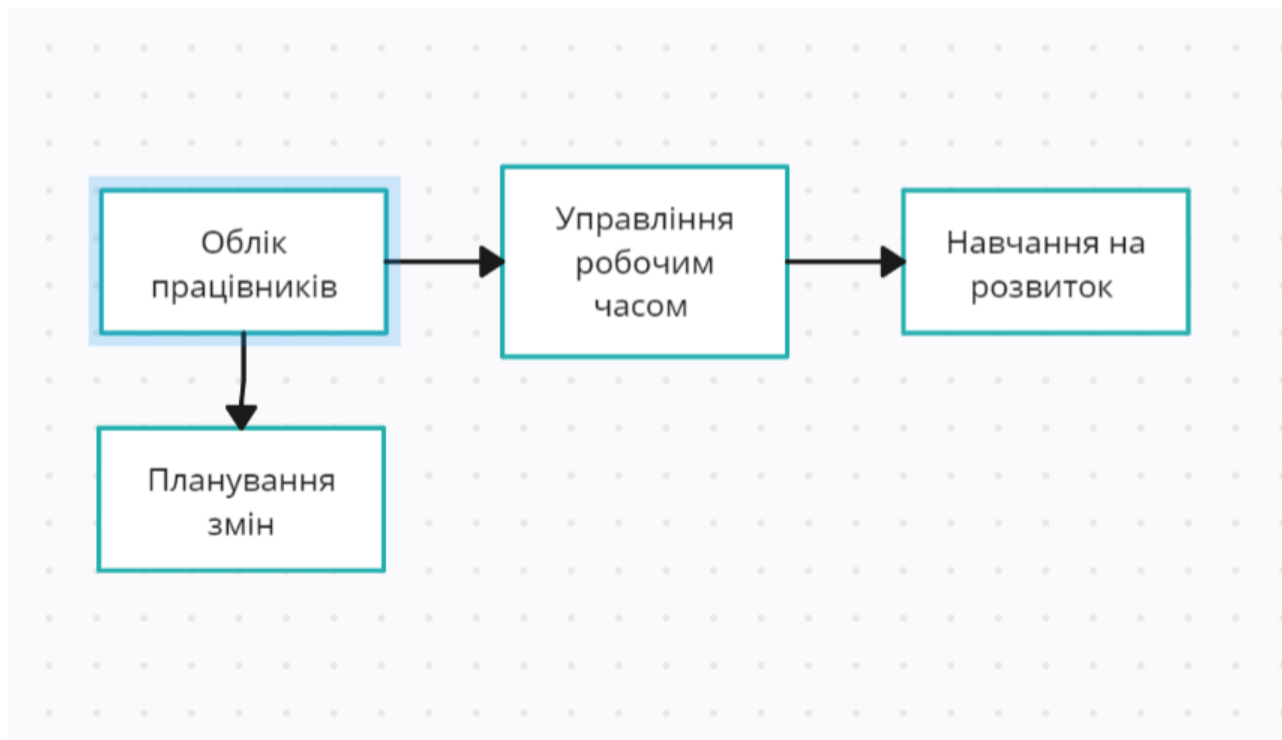


Рисунок 20 Схема управління персоналом (HR)

Джерело: розроблено автором

Розробка інформаційної системи для торгової компанії вимагає ретельного планування та інтеграції різноманітних модулів для забезпечення ефективного управління всіма аспектами діяльності компанії. Використання візуальних діаграм та схем допомагає краще зрозуміти структуру та взаємодію компонентів системи, що сприяє успішному впровадженню ІС у практику компанії. В результаті компанія отримує інструмент, який покращує управління бізнес-процесами, підвищує продуктивність та конкурентоспроможність на ринку.

Інформаційна система компанії повинна забезпечувати ефективне управління як офлайн, так і онлайн процесами. Важливо інтегрувати всі аспекти діяльності компанії в єдину систему для забезпечення безперебійної роботи та високого рівня обслуговування клієнтів. Хоча можливості інтернет-магазину розширюють ринки збуту, багато торгових компаній надають перевагу офлайн продажам, оскільки вони дозволяють підтримувати безпосередній контакт з клієнтами та пропонувати унікальний сервіс, який неможливо забезпечити онлайн.

2.2 «Методи та моделі в інформаційних управляючих системах і технологіях»

Проектування інформаційних управляючих систем (ІУС) для торгової компанії є критично важливим завданням, яке забезпечує ефективне управління

всіма бізнес-процесами, від закупівлі товарів до їх реалізації клієнтам. Сучасні ІУС повинні відповідати вимогам системності, комплексності, відкритості, адаптивності, надійності, безпеки, мобільності та простоти в обслуговуванні. Вибір відповідних методів і моделей для розробки таких систем забезпечує їх ефективність і здатність до інтеграції з іншими програмними продуктами. У цьому рефераті розглядаються основні етапи проектування ІУС, включаючи дослідження предметної області, обґрунтування вибору архітектури, методів і інструментів розробки, а також надаються рекомендації щодо створення документації для різних груп користувачів системи.

Модель сутність-зв'язок (Entity-Relationship Model, ER-модель)

Модель сутність-зв'язок (ER-модель) є однією з найпоширеніших методологій для проектування баз даних. Вона використовується для візуалізації та опису структур даних, що зберігаються в системі, а також для визначення взаємозв'язків між цими даними. ER-модель була вперше запропонована Пітером Ченом у 1976 році і з того часу стала основним інструментом для аналізу та проектування баз даних.

Основні компоненти ER-моделі

Сутності (Entities):

- Сутність представляє об'єкт реального світу або концепцію, про яку зберігаються дані.
- Сутності можуть бути фізичними об'єктами (наприклад, товар) або абстрактними поняттями (наприклад, замовлення).

Атрибути (Attributes):

- Атрибути описують властивості або характеристики сутностей.
- Наприклад, для сутності "Товар" атрибутами можуть бути назва, ціна, опис, виробник тощо.

Зв'язки (Relationships):

- Зв'язки визначають, як сутності взаємодіють між собою.
- Наприклад, зв'язок між сутностями "Клієнт" і "Замовлення" може вказувати на те, які клієнти зробили які замовлення.

Типи зв'язків (Cardinalities):

- Типи зв'язків визначають кількісні відношення між сутностями (наприклад, один до одного, один до багатьох, багато до багатьох).
- Вони показують, скільки екземплярів однієї сутності можуть бути пов'язані з екземплярами іншої сутності.

Розглянемо приклад ER-моделі для торгової компанії, що продає товари:

Сутності:

- Клієнт: Атрибути - клієнтський номер, ім'я, прізвище, адреса, телефон.
- Товар: Атрибути - товарний код, назва, ціна, категорія, виробник.
- Замовлення: Атрибути - номер замовлення, дата замовлення, статус.

Зв'язки:

- Робить замовлення (Клієнт → Замовлення): Один клієнт може зробити багато замовлень, але кожне замовлення належить одному клієнту (один до багатьох).
- Містить товари (Замовлення → Товар): Одне замовлення може містити багато товарів, і один товар може бути в багатьох замовленнях (багато до багатьох).

Додаткові сутності для зв'язків багато до багатьох:

- Замовлення-Товар: Атрибути - номер замовлення, товарний код, кількість товарів, ціна за одиницю.

Переваги ER-моделі

- Візуалізація структури даних: ER-модель надає наочне уявлення про структуру бази даних, що полегшує розуміння та спілкування між розробниками і замовниками.
- Чітке визначення зв'язків: Модель дозволяє точно визначити взаємозв'язки між сутностями, що знижує ризик помилок при реалізації бази даних.
- Гнучкість та масштабованість: ER-модель дозволяє легко додавати нові сутності та зв'язки, адаптуючи базу даних до змінних вимог бізнесу.
- Модель сутність-зв'язок є ефективним інструментом для проєктування баз даних, який забезпечує чітке та структуроване представлення даних і їх взаємозв'язків. Використання ER-моделі сприяє створенню надійних і зрозумілих баз даних, що відповідають потребам користувачів і бізнесу в цілому.

Реляційна модель є однією з найпоширеніших моделей для організації даних в інформаційних управляючих системах. Вона була розроблена Едгаром Коддом у 1970 році та стала основою для сучасних систем управління базами даних (СУБД). Реляційна модель організовує дані у вигляді таблиць (відношень), що дозволяє ефективно зберігати, запитувати та маніпулювати даними.

Основні компоненти реляційної моделі

Таблиці (Relations):

- Таблиці є основними структурними елементами реляційної бази даних.
- Кожна таблиця складається з рядків (записів) і стовпців (полів).

- Наприклад, таблиця "Клієнти" може містити поля "Клієнтський номер", "Ім'я", "Прізвище", "Адреса", "Телефон".

Рядки (Записи, Tuples):

- Кожен рядок у таблиці представляє окремий запис і містить конкретні дані.
- Наприклад, один рядок у таблиці "Клієнти" може містити інформацію про одного клієнта: {1, "Іван", "Петров", "Київ, вул. Шевченка, 10", "123-45-67"}.

Стовпці (Поля, Attributes):

- Кожен стовпець у таблиці представляє певний тип даних або властивість сутності.
- Наприклад, у таблиці "Клієнти" стовпець "Ім'я" містить імена всіх клієнтів.

Ключі (Keys):

- Первинний ключ (Primary Key):
- Унікальний ідентифікатор для кожного запису в таблиці.
- Наприклад, "Клієнтський номер" у таблиці "Клієнти".
- Зовнішній ключ (Foreign Key):
- Поле в одній таблиці, яке посилається на первинний ключ в іншій таблиці, створюючи зв'язок між таблицями.
- Наприклад, "Клієнтський номер" у таблиці "Замовлення", який посилається на таблицю "Клієнти".

Відношення (Relationships):

- Відношення визначають, як таблиці взаємодіють між собою.
- Наприклад, відношення між таблицями "Клієнти" і "Замовлення" може бути визначене через зовнішній ключ "Клієнтський номер" у таблиці "Замовлення".

Розглянемо приклад реляційної моделі для торгової компанії, що продає товари:

- Таблиця "Клієнти": клієнтський номер (РК), Ім'я, Прізвище, Адреса, Телефон.
- Таблиця "Товари": товарний код (РК), Назва, Ціна, Категорія, Виробник.
- Таблиця "Замовлення": номер замовлення (РК), Дата замовлення, Клієнтський номер (FK), Сума.
- Таблиця "Замовлення-Товари": номер замовлення (FK), Товарний код (FK), Кількість, Ціна за одиницю.

У реляційної моделі є дуже важливі переваги, такі як:

- Простота використання: Таблична структура є інтуїтивно зрозумілою та легкою у використанні.

- Гнучкість запитів: Мова запитів SQL (Structured Query Language) дозволяє виконувати складні операції з даними.
- Цілісність даних: Ключі та обмеження забезпечують цілісність і консистентність даних.
- Логічна та фізична незалежність даних спрощує їх адміністрування та зміну структури без впливу на додатки.

Реляційна модель є основою для більшості сучасних баз даних завдяки своїй простоті, гнучкості та потужності. Використання реляційної моделі в інформаційних управляючих системах дозволяє ефективно організовувати, зберігати та обробляти великі обсяги даних, що є критично важливим для забезпечення безперебійної роботи торгової компанії та досягнення її бізнес-цілей.

Модель "клієнт-сервер" є однією з основних архітектурних моделей, що використовується в інформаційних управляючих системах (ІУС). Вона передбачає розподіл функцій і завдань між двома типами компонентів: клієнтами та серверами. Клієнти відповідають за інтерфейс з користувачем і виконання запитів, тоді як сервери забезпечують централізоване зберігання даних, виконання обчислень і надання послуг.

Основні компоненти моделі "клієнт-сервер"

Клієнт (Client):

- Клієнт - це програма або пристрій, який ініціює запити до сервера.
- Клієнтський додаток забезпечує інтерфейс користувача, через який здійснюється взаємодія з системою.
- Клієнти можуть бути тонкими (Thin Clients), які виконують мінімальні функції і покладаються на сервер для обробки даних, або товстими (Thick/Fat Clients), які виконують значну частину обробки даних локально.

Сервер (Server):

- Сервер - це програма або пристрій, що обробляє запити від клієнтів і надає їм необхідні ресурси та послуги.
- Сервери можуть спеціалізуватися на різних функціях, таких як бази даних (Database Servers), веб-сервіси (Web Servers), файлові сервіси (File Servers) тощо.
- Сервери забезпечують централізоване управління даними, безпеку та надійність.

Мережа (Network):

- Мережа з'єднує клієнтів і сервери, забезпечуючи передачу даних між ними.
- Мережеві протоколи, такі як TCP/IP, забезпечують надійну і ефективну комунікацію.

Принципи роботи моделі "клієнт-сервер"

Запити та відповіді:

- Клієнт ініціює запит до сервера, запитуючи певні ресурси або послуги.
- Сервер отримує запит, обробляє його і надсилає відповідь клієнту.
- Цей цикл запит-відповідь є основою взаємодії в моделі "клієнт-сервер".

Розподіл завдань:

- Завдання розподіляються між клієнтом і сервером, щоб оптимізувати продуктивність і ефективність системи.
- Клієнти виконують завдання, пов'язані з інтерфейсом користувача і деякими обчисленнями, тоді як сервери обробляють дані, забезпечують доступ до баз даних та інші централізовані послуги.

Переваги моделі "клієнт-сервер"

- Масштабованість: Система може легко масштабуватися за рахунок додавання нових клієнтів або серверів.
- -Централізоване управління даними: Сервери забезпечують централізоване зберігання і управління даними, що спрощує їх обслуговування і захист.
- -Розподіл навантаження: Завдання розподіляються між клієнтами і серверами, що підвищує ефективність системи.
- Безпека: Централізоване управління дозволяє впроваджувати ефективні механізми безпеки для захисту даних.

Приклад моделі "клієнт-сервер" для торгової компанії

Клієнти:

- Програмні додатки на робочих станціях співробітників компанії, що використовуються для введення та отримання даних.
- Мобільні додатки для доступу до системи з планшетів або смартфонів.

Сервери:

- База даних (Database Server): Центральний сервер бази даних, що зберігає інформацію про клієнтів, товари, замовлення тощо.
- Веб-сервер (Web Server): Сервер, що обробляє веб-запити клієнтів і надає веб-інтерфейс для взаємодії з системою.
- Файловий сервер (File Server): Сервер для зберігання та управління файлами, такими як звіти та документи.

Модель "клієнт-сервер" є основою для побудови багатьох сучасних інформаційних управляючих систем, забезпечуючи ефективний розподіл завдань і ресурсів між клієнтами і серверами. Використання цієї моделі дозволяє

торговій компанії забезпечити надійне і безпечне управління даними, а також забезпечити гнучкість і масштабованість системи відповідно до потреб бізнесу.

Модель мікросервісів є сучасною архітектурною підходом до розробки програмного забезпечення, яка полягає в розподілі монолітної програми на набір невеликих, незалежних сервісів, кожен з яких виконує окрему бізнес-функцію. Кожен мікросервіс є автономним і може бути розроблений, розгорнутий і масштабований незалежно від інших мікросервісів. Цей підхід дозволяє значно підвищити гнучкість, масштабованість і стійкість системи.

Основні компоненти моделі мікросервісів:

Мікросервіси (Microservices):

- Невеликі, автономні служби, кожна з яких відповідає за виконання конкретної бізнес-функції.
- Кожен мікросервіс має власну базу даних і модель даних, що дозволяє уникати тісної залежності між сервісами.

API Gateway:

- Сервіс, що виступає єдиною точкою входу для всіх запитів до системи.
- Забезпечує маршрутизацію запитів до відповідних мікросервісів, авторизацію, аутентифікацію, обробку помилок та інші функції.

Система обміну повідомленнями (Message Broker):

- Забезпечує асинхронну комунікацію між мікросервісами.
- Використовується для передачі повідомлень і подій, що дозволяє мікросервісам взаємодіяти без прямого виклику один одного.

Контейнеризація та оркестрація:

- Контейнери (наприклад, Docker) використовуються для ізоляції мікросервісів і забезпечення їх незалежного розгортання.
- Оркестратори контейнерів (наприклад, Kubernetes) керують розгортанням, масштабуванням і управлінням контейнерами.

Принципи роботи моделі мікросервісів:

Незалежність розгортання: кожен мікросервіс може бути розгорнутий і оновлений незалежно від інших, що дозволяє швидко впроваджувати нові функції і виправляти помилки.

Чітке розділення обов'язків: кожен мікросервіс відповідає за конкретну бізнес-функцію, що забезпечує чітке розділення обов'язків і спрощує розробку та підтримку.

Стійкість до збоїв: збої одного мікросервісу не впливають на роботу інших, що підвищує загальну стійкість системи.

Масштабованість: зожен мікросервіс може бути масштабований незалежно, що дозволяє оптимально використовувати ресурси і ефективно обробляти змінні навантаження

Приклад моделі мікросервісів для торгової компанії

Мікросервіси:

- Сервіс управління клієнтами: Відповідає за зберігання та обробку даних про клієнтів.
- Сервіс управління товарами: Відповідає за зберігання та обробку даних про товари.
- Сервіс управління замовленнями: Відповідає за обробку замовлень клієнтів.
- Сервіс платежів: Відповідає за обробку платежів і взаємодію з платіжними системами.

API Gateway:

- Приймає запити від клієнтів (веб-додатків, мобільних додатків) і перенаправляє їх до відповідних мікросервісів.

Система обміну повідомленнями:

- Використовується для передачі подій між сервісами, наприклад, повідомлення про нові замовлення, зміни в статусі замовлень тощо.

Переваги моделі мікросервісів:

- Гнучкість розробки: Незалежний розвиток мікросервісів дозволяє командам працювати паралельно і швидко впроваджувати нові функції.
- Масштабованість: Можливість масштабування окремих мікросервісів дозволяє ефективно управляти ресурсами.
- Стійкість до збоїв: Локалізація збоїв в одному мікросервісі не впливає на роботу всієї системи.
- Технологічна різноманітність: Кожен мікросервіс може бути розроблений з використанням різних технологій та мов програмування, що забезпечує гнучкість вибору оптимальних інструментів для вирішення конкретних завдань.

Модель мікросервісів забезпечує високу гнучкість, масштабованість і стійкість інформаційних управляючих систем. Використання цієї моделі дозволяє торговій компанії швидко адаптуватися до змін ринкових умов, ефективно управляти ресурсами і забезпечувати надійну роботу системи. Модель мікросервісів є оптимальним вибором для сучасних торгових компаній, що прагнуть досягти високої продуктивності та ефективності своїх інформаційних управляючих систем.

Методи аналізу та проектування систем є ключовими етапами в розробці інформаційних управляючих систем (ІУС). Ці методи дозволяють систематично вивчати потреби користувачів і бізнес-процеси, а також розробляти ефективні рішення для їх автоматизації. На основі результатів аналізу створюється проект системи, що включає специфікації вимог, архітектуру, дизайн і плани реалізації. Існують основні методи аналізу, такі як:

Аналіз вимог (Requirements Analysis):

- Збір вимог: Процес збору інформації про те, які функції та властивості повинна мати система. Використовуються інтерв'ю, опитування, аналіз документів, спостереження тощо.
- Документування вимог: Формалізація вимог у вигляді специфікацій, які включають функціональні та нефункціональні вимоги.
- Валідація вимог: Перевірка коректності, повноти та узгодженості вимог шляхом їх огляду, тестування та погодження з користувачами.

Бізнес-аналіз (Business Analysis):

- Аналіз бізнес-процесів: Вивчення та моделювання поточних бізнес-процесів з метою їх оптимізації та автоматизації. Використовуються методи BPMN (Business Process Model and Notation).
- Аналіз зацікавлених сторін (Stakeholder Analysis): Визначення та аналіз потреб і впливу різних груп зацікавлених сторін на проект.

Аналіз даних (Data Analysis):

- Моделювання даних: Створення моделей даних, що відображають логічну та фізичну структуру даних у системі. Використовуються ER-моделі (Entity-Relationship Model) та реляційні моделі.
- Аналіз потоків даних: Визначення, як дані рухаються через систему, і розробка моделей потоків даних (DFD - Data Flow Diagrams).

Розглянемо основні методи проектування, наприклад функціональне проектування - це розробка функціональних моделей: Визначення та опис основних функцій системи за допомогою діаграм функцій та моделей Use Case, специфікація функціональних вимог: Документування функцій системи, включаючи опис входів, виходів, алгоритмів обробки та взаємодії між компонентами.

Розробка архітектурної моделі: Визначення основних компонентів системи, їх взаємодії та розподілу обов'язків. Використовуються моделі архітектури, такі як багатошарова архітектура, мікросервіси, клієнт-сервер тощо.

Вибір технологій і платформ: Обґрунтування вибору технологій, мов програмування, фреймворків та платформ для реалізації системи.

Проектування інтерфейсу користувача (User Interface Design):

Розробка прототипів інтерфейсу: Створення ескізів та прототипів інтерфейсів користувача для тестування і узгодження з користувачами.

Юзабіліті-тестування: Перевірка зручності використання інтерфейсів шляхом проведення тестів з реальними користувачами.

Проектування баз даних (Database Design):

Логічне проектування баз даних: Створення логічної моделі бази даних на основі вимог до даних та їх взаємозв'язків.

Фізичне проектування баз даних: Визначення фізичної структури бази даних, включаючи таблиці, індекси, ключі та інші об'єкти бази даних.

Функціональне проектування має кілька важливих переваг.

По-перше, систематичність: використання методів забезпечує систематичний підхід до розробки систем, що знижує ризики і помилки.

По-друге, зрозумілість: документування вимог і проектних рішень робить процес розробки прозорим і зрозумілим для всіх учасників проекту.

Третя перевага – оптимізація ресурсів: ефективне планування і проектування дозволяє оптимально використовувати ресурси, що скорочує час і витрати на розробку. Нарешті, гнучкість: використання сучасних методів аналізу і проектування забезпечує гнучкість у адаптації системи до змінних вимог і умов. Методи аналізу та проектування систем є фундаментальними етапами в розробці інформаційних управляючих систем. Вони забезпечують глибоке розуміння вимог, оптимізацію бізнес-процесів, а також розробку ефективних і надійних рішень. Використання цих методів дозволяє створювати якісні системи, що відповідають потребам користувачів і сприяють досягненню бізнес-цілей.

Методи розробки програмного забезпечення є наборами практик, методологій та процесів, які використовуються для організації роботи над проектами з метою створення якісного програмного забезпечення. Існує кілька основних методів, кожен з яких має свої переваги і недоліки, а також підходить для різних типів проектів і умов роботи.

Основні методи розробки програмного забезпечення
Каскадна модель (Waterfall Model):

- Послідовність етапів: Каскадна модель передбачає послідовне проходження етапів розробки: збір вимог, аналіз, проектування, кодування, тестування та обслуговування.
- Документування: Кожен етап закінчується створенням детальної документації, що слугує основою для наступного етапу.
- Переваги: Чітка структура та зрозумілість процесу, добре підходить для проектів з чітко визначеними вимогами.
- Недоліки: Важкість внесення змін після завершення етапу, ризик виявлення проблем на пізніх етапах.

Гнучка розробка (Agile Development):

- Ітеративний підхід: Agile підхід передбачає ітеративну розробку, де проект розбивається на невеликі цикли (ітерації), кожен з яких завершується створенням працюючої версії ПЗ.
- Інкрементальні покращення: Кожна ітерація додає нові функціональні можливості до системи.
- Переваги: Гнучкість, швидка адаптація до змін вимог, активна взаємодія з користувачами.
- Недоліки: Може бути важко керувати великими проектами без чіткого плану.

Метод Scrum:

- Ролі та процеси: Scrum визначає чіткі ролі (Scrum Master, Product Owner, Команда розробників) та процеси (спринти, щоденні стендапи, огляди спринтів).
- Спринти: Розробка проходить у фіксованих періодах (спринтах), які зазвичай тривають від двох до чотирьох тижнів.
- Переваги: Прозорість процесу, швидкий зворотний зв'язок, можливість швидкого виявлення та усунення проблем.
- Недоліки: Вимоги до дисципліни команди, можливість конфліктів через інтенсивність процесу.

Метод Kanban:

- Візуалізація робочого процесу: Kanban використовує візуальні дошки для відображення завдань і їх статусу (наприклад, To Do, In Progress, Done).
- Поточкова робота: Основний акцент робиться на управлінні потоком роботи, зменшенні часу виконання завдань і підвищенні продуктивності.
- Переваги: Гнучкість, можливість швидкої адаптації до змін, зниження перевантаження команди.

- Недоліки: Менше структурованості порівняно зі Scrum, можливість неефективного управління ресурсами.

Екстремальне програмування (Extreme Programming, XP):

- Ітеративний підхід: XP акцентує увагу на коротких ітераціях і частих випусках програмного забезпечення.
- Практики XP: Використання практик парного програмування, безперервної інтеграції, тестування перш за все (Test-Driven Development, TDD), колективне володіння кодом.
- Переваги: Висока якість коду, швидке виявлення і виправлення помилок, покращена співпраця в команді.
- Недоліки: Високі вимоги до дисципліни і залученості команди, можливість вигорання через інтенсивність процесу.

Модель V-Model (Verification and Validation Model):

- Розширена каскадна модель: V-Model є варіантом каскадної моделі, де кожен етап розробки пов'язаний з відповідним етапом тестування.
- Верифікація і валідація: Кожен етап вимагає чіткої верифікації і валідації, що забезпечує високу якість продукту.
- Переваги: Підвищена якість через чітку структуру тестування, зменшення ризиків виявлення помилок на пізніх етапах.
- Недоліки: Відсутність гнучкості, важкість внесення змін після завершення етапу.

Вибір методу розробки програмного забезпечення залежить від багатьох факторів, включаючи вимоги проекту, розмір і складність системи, наявність ресурсів і досвіду команди, а також бажаний рівень гнучкості і контролю. Наприклад:

- Каскадна модель добре підходить для проектів з чітко визначеними вимогами і обмеженим бюджетом.
- Agile і Scrum підходять для динамічних проектів, де вимоги можуть змінюватися, і потрібна висока гнучкість.
- Kanban ефективний для команд, що вже мають налагоджений робочий процес і прагнуть до підвищення продуктивності.
- XP підходить для проектів, де важлива висока якість коду і часті релізи.
- V-Model підходить для проектів, де потрібна висока якість і чіткий контроль тестування.

Методи розробки програмного забезпечення відіграють ключову роль у забезпеченні успішної реалізації проектів. Вибір правильного методу дозволяє оптимізувати процес розробки, зменшити ризики, підвищити якість програмного забезпечення і забезпечити задоволення потреб користувачів. Кожен метод має свої переваги і недоліки, тому важливо ретельно аналізувати вимоги і умови проекту перед вибором конкретного підходу.

Методи інтелектуального аналізу даних (Data Mining)

Інтелектуальний аналіз даних, або Data Mining, є процесом виявлення корисних закономірностей, моделей та знань з великих обсягів даних. Цей процес використовує різні методи з галузей статистики, машинного навчання, баз даних та штучного інтелекту для аналізу та інтерпретації даних. Data Mining знаходить застосування в різних сферах, таких як маркетинг, фінанси, медицина, роздрібна торгівля та інші.

Основні методи інтелектуального аналізу даних

Класифікація (Classification):

- Призначення: Прогнозування категорії або класу об'єкта на основі його характеристик.
- Приклади алгоритмів: Дерева рішень (Decision Trees), метод опорних векторів (Support Vector Machines), наївний байєсівський класифікатор (Naive Bayes), нейронні мережі (Neural Networks).
- Застосування: Спам-фільтри, діагностика хвороб, кредитний скоринг.

Кластеризація (Clustering):

- Призначення: Групування об'єктів у кластери так, щоб об'єкти в одному кластері були більш схожими між собою, ніж з об'єктами з інших кластерів.
- Приклади алгоритмів: k-середніх (k-means), ієрархічна кластеризація (Hierarchical Clustering), DBSCAN (Density-Based Spatial Clustering of Applications with Noise).
- Застосування: Сегментація ринку, аналіз поведінки клієнтів, виявлення аномалій.

Асоціативний аналіз (Association Analysis):

- Призначення: Виявлення цікавих зв'язків і залежностей між змінними в великих наборах даних.
- Приклади алгоритмів: Apriori, FP-Growth.
- Застосування: Аналіз кошика покупок (market basket analysis), рекомендаційні системи.

Регресія (Regression):

- Призначення: Прогнозування числових значень на основі вхідних змінних.
- Приклади алгоритмів: Лінійна регресія (Linear Regression), логістична регресія (Logistic Regression), регресійні дерева (Regression Trees).
- Застосування: Прогнозування продажів, оцінка ризиків, аналіз ринку.

Аналіз часових рядів (Time Series Analysis):

- Призначення: Аналіз даних, які змінюються з часом, для виявлення тенденцій, циклів і сезонних коливань.
- Приклади алгоритмів: ARIMA (AutoRegressive Integrated Moving Average), LSTM (Long Short-Term Memory) нейронні мережі.
- Застосування: Прогнозування попиту, фінансовий аналіз, моніторинг виробничих процесів.

Аналіз аномалій (Anomaly Detection):

- Призначення: Виявлення рідкісних об'єктів або подій, які відрізняються від більшості даних.
- Приклади алгоритмів: Ізоляційний ліс (Isolation Forest), локальна факторна аномалія (Local Outlier Factor, LOF).
- Застосування: Виявлення шахрайства, моніторинг безпеки, контроль якості.

Переваги методів інтелектуального аналізу даних

- Автоматизація: Здатність автоматично виявляти закономірності та аномалії без необхідності ручного втручання.
- Масштабованість: Можливість обробки великих обсягів даних завдяки використанню ефективних алгоритмів і обчислювальних ресурсів.
- Прогнозування: Здатність робити точні прогнози на основі існуючих даних, що допомагає у прийнятті обґрунтованих рішень.
- Оптимізація: Підвищення ефективності бізнес-процесів через оптимізацію ресурсів і виявлення найбільш значущих факторів.

Вибір методу інтелектуального аналізу даних залежить від специфіки завдання, типу даних і вимог до результатів. Наприклад:

- Для класифікації підходять завдання, де потрібно віднести об'єкти до конкретних категорій (наприклад, класифікація електронної пошти як спаму або ні).
- Кластеризація корисна для виявлення природних груп у даних (наприклад, сегментація клієнтів).
- Асоціативний аналіз ефективний для виявлення зв'язків між елементами (наприклад, які товари часто купуються разом).
- Регресія використовується для прогнозування числових значень (наприклад, прогнозування обсягів продажів).
- Аналіз часових рядів підходить для даних, змінних у часі (наприклад, прогнозування попиту на продукцію).
- Аналіз аномалій корисний для виявлення рідкісних подій або відхилень (наприклад, виявлення шахрайства в транзакціях).

Методи інтелектуального аналізу даних є потужними інструментами для виявлення прихованих знань та закономірностей у великих обсягах даних. Використання цих методів дозволяє організаціям підвищувати ефективність своєї діяльності, приймати обґрунтовані рішення та отримувати конкурентні переваги. Правильний вибір методу та його адаптація до конкретних завдань є ключем до успіху в сучасному інформаційному середовищі.

Машинне навчання (Machine Learning) - це галузь штучного інтелекту, яка досліджує алгоритми, які надають комп'ютерам здатність навчатися на основі даних, розпізнавати патерни та робити прогнози без явного програмування. Основна ідея полягає у створенні моделей, які можуть адаптуватися до нових даних і вдосконалювати свою продуктивність з часом.

Основні методи машинного навчання

Навчання з учителем (Supervised Learning):

- Призначення: Моделі навчаються на основі позначених даних, де кожен приклад має відомі відповіді.
- Приклади алгоритмів: Лінійна регресія, метод опорних векторів, нейронні мережі, дерева рішень.
- Застосування: Прогнозування цін на акції, класифікація зображень, розпізнавання мови.

Навчання без учителя (Unsupervised Learning):

- Призначення: Моделі навчаються на не позначених даних, спробуючи виявити природні структури або групи в наборі даних.
- Приклади алгоритмів: Кластеризація (k-середніх, ієрархічна кластеризація), асоціативний аналіз.

- Застосування: Сегментація ринку, виявлення аномалій, зменшення розмірності даних.

Підгонка моделей (Model Fitting):

- Призначення: Моделі навчаються на підгонці даних, де головна мета полягає в узгодженні моделі зі зваженими даними.
- Приклади алгоритмів: Методи оптимізації, такі як градієнтний спуск, стохастичний градієнтний спуск.
- Застосування: Побудова нейронних мереж, тренування моделей глибокого навчання.

Переваги машинного навчання

- Автоматизація: Можливість автоматичної адаптації до нових даних без необхідності ручного програмування.
- Здатність до узагальнення: Можливість робити прогнози на основі вивчених зразків і виявлення закономірностей, навіть на нових даних.
- Широкий спектр застосувань: Від класифікації до кластеризації, машинне навчання може бути використане у багатьох сферах.

Вибір конкретного методу машинного навчання залежить від типу даних, завдання і обсягу даних. Наприклад:

- Для навчання з учителем корисні у випадках, коли доступні позначені дані і потрібно здійснити класифікацію або прогнозування.
- Навчання без учителя використовується там, де відсутні позначені дані або потрібно виявити складні шаблони в даних.
- Підгонка моделей застосовується для побудови складних моделей, які враховують взаємозв'язки між даними.

Машинне навчання відіграє важливу роль у сучасній аналітиці даних, дозволяючи автоматизувати процеси аналізу та прийняття рішень. З правильним вибором методу та адаптацією до конкретних потреб, машинне навчання може стати потужним інструментом для вирішення різноманітних завдань у багатьох сферах життя.

Інструменти обробки великих даних (Big Data)

Обробка великих даних стала ключовим елементом сучасного інформаційного ландшафту. Це обумовлено зростанням обсягів даних та необхідністю їх аналізу для отримання цінних знань та прийняття стратегічних рішень. Інструменти обробки великих даних дозволяють збирати, зберігати, обробляти та аналізувати великі обсяги даних з різних джерел.

Основні інструменти обробки великих даних

Apache Hadoop:

- Опис: Фреймворк для розподіленої обробки великих даних на кластерах серверів.

- Функціональність: Забезпечує масштабованість, відмовостійкість та паралельну обробку даних за допомогою MapReduce.
- Застосування: Обробка структурованих і неструктурованих даних, аналіз логів, обробка великих обсягів даних.

Apache Spark:

- Опис: Відкритий фреймворк для обробки великих даних з підтримкою паралельної обробки та потокового аналізу даних.
- Функціональність: Забезпечує високу швидкодію через використання пам'яті для інтерактивних запитів та складних аналітичних операцій.
- Застосування: Машинне навчання, потоковий аналіз, обробка графових даних.

Apache Kafka:

- Опис: Розподілена система потокової обробки та розподілу повідомлень.
- Функціональність: Забезпечує можливість розподіленого зберігання та реплікації поточкових даних між вузлами.
- Застосування: Збирання, передача та обробка поточкових даних у реальному часі.

Apache Flink:

- Опис: Фреймворк для потокової та пакетної обробки даних в реальному часі.
- Функціональність: Забезпечує низьку затримку та високу надійність для потокової обробки даних.
- Застосування: Аналітика в реальному часі, обробка поточкових подій.

Apache Cassandra:

- Опис: Розподілена система керування базами даних, спроектована для обробки великих обсягів даних з високою доступністю та масштабованістю.
- Функціональність: Забезпечує розподілене зберігання даних з можливістю горизонтального масштабування.
- Застосування: Зберігання та обробка великих обсягів структурованих даних.

Інструменти обробки великих даних відіграють критичну роль у сучасних інформаційних технологіях, дозволяючи організаціям ефективно збирати, зберігати та аналізувати великі обсяги даних з різних джерел. Вибір конкретного інструменту залежить від конкретних потреб організації та характеру даних, але правильне використання цих інструментів може призвести до значного покращення ефективності та конкурентоспроможності.

РОЗДІЛ 3 «РОЗРОБЛЕННЯ ПРОЄКТНИХ РІШЕНЬ ТА ЇХ РЕАЛІЗАЦІЯ»

3.1 Проєктування бази даних та/або сховища даних для інформаційної управляючої системи

Проєктування баз даних є важливою складовою створення інформаційних систем, оскільки воно визначає структуру та організацію даних, що використовуються для підтримки бізнес-процесів. Особливо це актуально для торгових компаній, де обсяг даних може бути значним, а вимоги до їх обробки та аналізу - високими. Від правильного проєктування бази даних залежить ефективність її роботи, швидкість доступу до даних, надійність зберігання та масштабованість системи. Ми розглянемо основні етапи проєктування баз даних, включаючи аналіз вимог, концептуальне моделювання, логічне та фізичне проєктування, а також оптимізацію та тестування бази даних. Розуміння та дотримання цих етапів допоможе створити ефективну та надійну інформаційну систему, що відповідає потребам торгової компанії.

Основні поняття та визначення:

- База даних (БД): Організована сукупність структурованої інформації або даних, які зберігаються та управляються електронним способом. Бази даних використовуються для зберігання інформації, яка може бути легко доступна, керована та оновлювана.
- Сховище даних (DWH): Спеціалізована база даних, оптимізована для запитів та звітності, а не для обробки транзакцій. DWH збирає дані з різних джерел, очищує та перетворює їх для подальшого аналізу.
- Інформаційна управляюча система (ІУС): Система, що забезпечує збирання, зберігання, обробку та передачу інформації, необхідної для управління організацією. Включає програмне забезпечення, апаратні засоби та бази даних.

Існує декілька типів баз даних та їх класифікація, в даному розділі розглянемо декаліка із них.

Реляційні бази даних (RDBMS): Бази даних, засновані на реляційній моделі, де дані організовані у вигляді таблиць (відношень). Приклади: MySQL, PostgreSQL, Oracle, Microsoft SQL Server. Має певну характерну ознаку: Сувора схема, підтримка ACID (атомарність, узгодженість, ізоляція, стійкість).

Нереляційні бази даних (NoSQL):

- Бази даних, що не дотримуються реляційної моделі і можуть використовувати інші способи зберігання даних, такі як документи, графи, ключ-значення.
- Приклади: MongoDB, Cassandra, Redis, Neo4j.
- Характеристики: Гнучкість схеми, висока масштабованість, часто використовуються для роботи з великими обсягами даних.

Об'єктно-орієнтовані бази даних:

- Визначення: Бази даних, що зберігають дані у вигляді об'єктів, як у об'єктно-орієнтованому програмуванні.
- Приклади: ObjectDB, db4o.
- Характеристики: Підтримка об'єктно-орієнтованих принципів, таких як наслідування та поліморфізм.

Принципи та методи проектування баз даних включають кілька ключових аспектів. Нормалізація є процесом розділення бази даних на менші таблиці для зменшення надмірності даних та уникнення аномалій при оновленні. Вона визначається нормальними формами (від 1NF до 5NF), які вказують на різні рівні нормалізації. Протилежним до нормалізації є процес денаормалізації, при якому таблиці об'єднуються для підвищення продуктивності запитів за рахунок деякої надмірності даних.

Також важливим методом є ER-моделювання (ERD), що використовується для концептуального моделювання даних. Цей метод застосовує діаграми "сутність-зв'язок" (ER-діаграми) для візуалізації структури бази даних, що допомагає зрозуміти зв'язки між різними сутностями в системі. Сховища даних: призначення та особливості

Відмінності між БД та сховищем даних:

- База даних: Оптимізована для обробки транзакцій, забезпечує швидкі операції читання/запису.
- Сховище даних: Оптимізоване для аналітичних запитів, зберігає історичні дані для аналізу.

Архітектура сховищ даних:

- ETL-процес (Extract, Transform, Load): Витяг даних з різних джерел, трансформація для забезпечення єдиної структури та завантаження у сховище.
- Схеми даних: "Зоряна" схема, "Сніжинка", які організовують дані для швидкого доступу та аналітичного запиту.

- Теоретичні основи проєктування баз даних та сховищ даних є фундаментальними для створення ефективних інформаційних систем.
- Розуміння різних типів баз даних та принципів їх проєктування допомагає обрати оптимальне рішення для конкретних потреб торгової компанії.
- Використання сховищ даних дозволяє ефективно аналізувати великі обсяги даних та отримувати цінні інсайти для прийняття управлінських рішень.

В даному проєкті, буде використана сама реляційна база даних.

Реляційні бази даних досить часто використовуються в проєктуванні баз даних з моделюванням сутностей та зв'язків (ER-моделюванням) з кількох причин.

По-перше, РБД добре підходять для зберігання структурованих даних, що відповідають сутностям та атрибутам, які визначені в ER-моделі. Кожна сутність може бути представлена як окрема таблиця, а атрибути - як стовпці цієї таблиці. Далі, РБД дозволяють легко визначати взаємозв'язки між сутностями за допомогою зовнішніх ключів та посилань між таблицями. Це спрощує обробку даних та запити, що включають зв'язки між різними сутностями.

Також, РБД надають механізми для нормалізації даних, що допомагає уникнути аномалій в базі даних та забезпечує цілісність даних. Нормалізація дозволяє ефективно організувати дані, що відповідають структурі ER-моделі. І нарешті, реляційні бази даних часто є добре оптимізованими для швидкого доступу до даних, що робить їх ефективними для використання в сучасних системах. Результати розробки дивитись додаток А

Проектування бази даних здійснюється у три етапи:

- концептуальне проектування;
- логічне проектування;
- фізичне проектування.

Мета етапу концептуального проектування — створення концептуальної моделі даних виходячи з уявлень користувачів про предметну область. Для її досягнення виконується низка послідовних процедур. Приклад сутнісної (концептуальної) схеми:

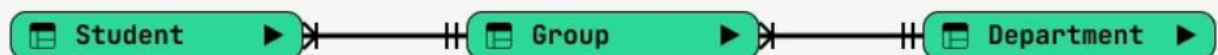


Рисунок 21 – Приклад сутнісної (концептуальної) схеми

Джерело: <https://javarush.com/ua/quests/lectures/ua.questhibernate.level17.lecture01>

Визначення сутностей та їхнє документування. Для ідентифікації сутностей визначаються об'єкти, які існують незалежно від інших. Такі об'єкти є суттю. Кожній сутності надається осмислене ім'я, зрозуміле користувачам. Імена та описи сутностей заносяться до словника даних. Якщо можливо, то встановлюється очікувана кількість примірників кожної сутності.

Визначення зв'язків між сутностями та їх документування. Визначаються ті зв'язки між сутностями, які необхідні для задоволення вимог до проекту бази даних. Встановлюється тип кожної їх. Виявляється клас власності сутностей. Зв'язкам надаються осмислені імена, виражені дієсловами. Розгорнутий опис кожного зв'язку із зазначенням її типу та класу належності сутностей, що беруть участь у зв'язку, заноситься до словника даних.

Створення ER-моделі предметної області. Для представлення сутностей та зв'язків між ними використовуються ER-діаграми. На їх основі створюється єдиний наочний образ предметної області, що моделюється — ER-модель предметної області.

Визначення атрибутів та їх документування. Виявляються всі атрибути, що описують сутність створеної моделі ER. Кожному атрибуту надається осмислене ім'я, зрозуміле користувачам. Про кожен атрибут у словник даних містяться такі відомості:

- ім'я атрибуту та його опис;
- тип та розмірність значень;
- значення, прийняте для стандартного атрибута (якщо таке є);
- може атрибут мати NULL-значення;
- чи є атрибут зіставним, і якщо це так, то з яких простих атрибутів він складається.

Наприклад, атрибут "П.І.Б. клієнта" може складатися з простих атрибутів "Прізвище", "Ім'я", "По батькові", а може бути простим, що містить єдині значення, як-то "Суховієнко Олександр Віталійович". Якщо користувач не потребує доступу до окремих елементів "П.І.Б.", то атрибут представляється як простий; чи є атрибут розрахунковим, і якщо це так, то як обчислюються його значення. (рис)

Табельний	Прізвище	Ім'я	По-батьков	Адреса	Телефон	Класифікація, щоб додати
101	Труханів	Олександр	Олександров	вул.Воскресе	(380) 95 742 9	
102	Ус	Дмитро	Миколайович	вул.Гоголівсь	(380) 67 143 8	
103	Свистун	Ярослав	Романович	вул. Жулянсь	(380) 96 715 4	
104	Бурбига	Яна	Сергіївна	вул.Нова, Ше	(380) 97 134 5	
105	Волошин	Максим	Олександроч	вул.Навізна		
0						

Рисунок 21 – модель ER

Джерело: розроблено автором

Також спроектована можливість додавання нових співробітників

Список Продавців

Табельний номер	101
Прізвище	Труханів
Ім'я	Олександр
По-батькові	Олександрович
Адреса	вул.Воскресенська
Телефон	(380) 95 742 9171

Додати запис Поперед запис Наступ запис Закрити форму

Рисунок 22 – можливість додавання персоналу

Джерело: розроблено автором

Визначення первинних ключів для сутностей та їх документування. На цьому кроці керуються визначенням первинного ключа – як атрибута чи набору атрибутів сутності, що дозволяє унікальним чином ідентифікувати її екземпляри. Відомості про первинні ключі містяться у словнику даних. (рис)

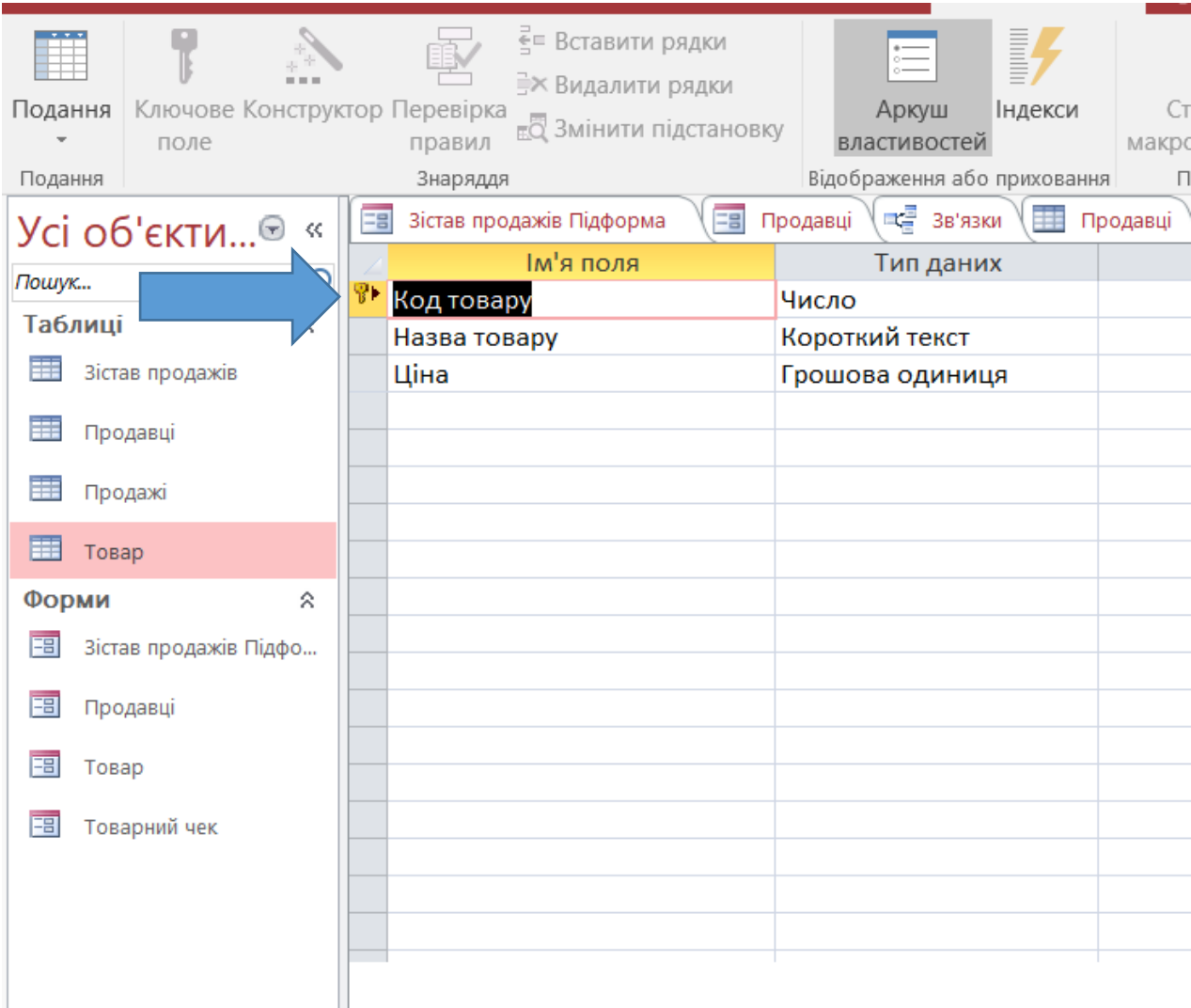


Рисунок 23 – первинний ключ

Джерело: розроблено автором

Обговорення концептуальної моделі даних із кінцевими користувачами. Концептуальна модель даних представляється ER-моделлю з супровідною документацією, що містить опис розробленої моделі даних. Якщо будуть виявлені невідповідності предметної області, то в модель вносяться зміни доти, доки користувачі не підтвердять, що запропонована ним модель адекватно відображає їхні особисті уявлення.

Приклад логічної схеми БД.

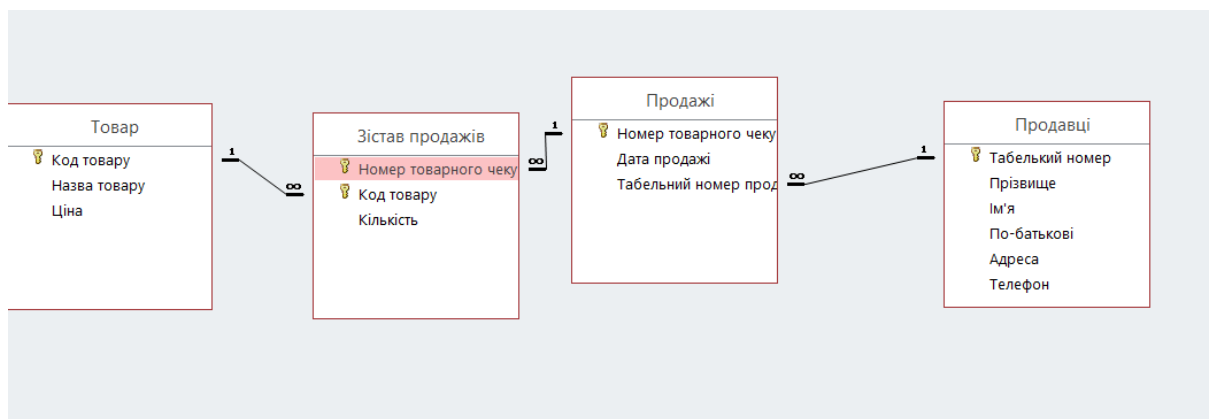


Рисунок 24 – логічна схема бд

Джерело: розроблено автором

Найчастіше вибирається реляційна модель даних у зв'язку з наочністю табличного подання даних та зручності роботи з ними.

The screenshot shows a database application interface. At the top, there are tabs for 'Подання' (Presentation), 'Буфер обміну' (Exchange Buffer), 'Сортування й фільтр' (Sorting and Filter), 'Записи' (Records), and 'Пошук' (Search). Below these are buttons for 'Усі об'єкти...' (All objects...), 'Пошук...' (Search...), and 'Таблиці' (Tables). The 'Таблиці' section lists 'Зістав продажів' (Sales List), 'Продавці' (Sellers), 'Продажі' (Sales), and 'Товар' (Goods). The 'Форми' (Forms) section lists 'Зістав продажів Підфо...' (Sales List Subform), 'Продавці' (Sellers), 'Продажі' (Sales), and 'Товар' (Goods). The 'Товар' form is currently active, showing fields for 'Код товару' (Goods code) with value '120', 'Назва товару' (Goods name) with value 'Телефон Samsung', and 'Ціна' (Price) with value '5 000,00 ₴'. Below the fields are buttons for 'Додати запис' (Add record), 'Поперед запис' (Previous record), 'Наступ запис' (Next record), and 'Закрити форму' (Close form).

Рисунок 25 – подання даних в БД

Джерело: розроблено автором

Програма здатна, через деякий проміжок часу, після накопичення даних продажу товарів, зробити звіт з кількістю проданих товарів, а також в яких чеках вони знаходяться, що підтверджує виконану вимогу, для створення звітів

Подання	Буфер обміну	Сортуння й фільтр	Записи
Усі об'єкти...	Продавці	Зв'язки	Продавці
Пошук...	Продажі	Зістав продажів	Товар
Таблиці	Товар	Товар	Товарний чек
Зістав продажів			
Продавці			
Продажі			
Форми			
Зістав продажів Підфо...			
Продавці			
Товар			
Товарний чек			

Код товару	Назва товару	Ціна	Клацніть, щоб додати
120	Телефон Samsung	5 000,00 ₴	
Номер товарного чеку	Кількість	Клацніть, і	
3	1		
*	0	0	
121	Телефон Iphone	10 000,00 ₴	
Номер товарного чеку	Кількість	Клацніть, і	
1	5		
*	0	0	
122	Телефон xiaomi	3 000,00 ₴	
Номер товарного чеку	Кількість	Клацніть, і	
1	1		
3	3		
*	0	0	
201	Ноутбук Lenovo	20 000,00 ₴	
Номер товарного чеку	Кількість	Клацніть, і	
*	0	0	
202	Ноутбук Pixus	8 000,00 ₴	
203	Ноутбук Acer	13 000,00 ₴	
301	Телевізор LG	35 000,00 ₴	
302	Телевізор Sony	40 000,00 ₴	
303	Телевізор KIVI	20 000,00 ₴	
*	0	0,00 ₴	

Рисунок 26 – звіт продажів

Джерело: розроблено автором

Згідно вимог, було спроектовано виконання умов, за яких клієнт, який хоче придбати товару на суму більше ніж 50 000 грн, отримує знижку від компанії, у розмірі 0,05 % від контрольної суми, обов’язково більше ніж 50 000 грн.

Код товару	Кількість	Ціна	Сума
Телефон Iphone	5	10 000,00 ₴	50 000,00 ₴
Телефон xiaomi	1	3 000,00 ₴	3 000,00 ₴

Всього: 53 000,00 ₴
Знижка: 2 650,00 ₴
Зі знижкою: 50 350,00 ₴

Табельний номер: 103 Свистун Романович Ярослав

Додати запис | Поперед запис | Наступ запис | Закрити форму

Рисунок 25 – результат згідно поставленої задачі
Джерело: розроблено автором

Більш детальна схема і демонстрація побудови виразів

Товарний чек № Номер

Дата продажі

ООО "Comfy ІНН 18384616480"

Номер товарного чеку

Код товару

Кількість

Ціна

Сума

Всього

Знижка

Зі знижкою

Табельний номер

Табельний номер продавця

ПІБ

Додати запис | Поперед запис | Наступ запис | Закрити форму

Побудовник виразів

Введіть вираз для створення обчислюваного елемента керування:

(Приклади виразів включають [поле1] + [поле2] і [поле1] < 5)

[тогоСума]-[Розмір знижки]

Елементи виразів

Категорії виразів

Значення виразів

Рисунок 26 – побудова виразів
Джерело: розроблено автором

Визначення набору таблиць, виходячи з ER-моделі та їх документування. Для кожної сутності ER-моделі створюється таблиця. Ім'я сутності — ім'я таблиці. Встановлюються зв'язки між таблицями за допомогою механізму первинних та зовнішніх ключів. Структури таблиць та встановлені зв'язки між ними документуються.

Перевірка логічної моделі даних щодо можливості виконання всіх транзакцій, передбачених користувачами. Транзакція — це набір дій, що виконуються окремим користувачем або прикладною програмою з метою зміни вмісту бази даних.

Визначення вимог підтримки цілісності даних та їх документування. Ці вимоги є обмеженнями, які запроваджуються, щоб запобігти розміщенню в базі даних суперечливих даних. На цьому етапі питання цілісності даних висвітлюються безвідносно до конкретних аспектів її реалізації. Повинні бути розглянуті такі типи обмежень:

обов'язкові дані. З'ясовується, чи є атрибути, які не можуть мати значення NULL; обмеження для значень атрибутів. Визначаються допустимі значення для атрибутів; цілісність сутностей. Вона досягається, якщо первинний ключ сутності не містить значення NULL; посиальна цілісність. Вона розуміється так, що значення зовнішнього ключа має обов'язково бути присутнім у первинному ключі одного з рядків таблиці для батьківської сутності; обмеження, що накладаються бізнес-правилами. Наприклад, у випадку з проектом БАНК може бути правило, яке забороняє клієнту розпоряджатися, скажімо, більш ніж трьома рахунками. Відомості про всі встановлені обмеження цілісності даних містяться у словнику даних. Створення остаточного варіанта логічної моделі даних та обговорення його з користувачами. На цьому кроці готується остаточний варіант ER-моделі, що представляє логічну модель даних. Сама модель та оновлена документація, включно зі словником даних та реляційною схемою зв'язку таблиць, надаються для перегляду та аналізу користувачам, які повинні переконатися, що вона точно відображає предметну область.

Мета етапу фізичного проектування — опис конкретної реалізації бази даних, що розміщується у зовнішній пам'яті комп'ютера. Це опис структури зберігання даних та ефективних методів доступу до даних бази. У логічному проектуванні відповідають на питання — що треба зробити, а у фізичному — обирається спосіб, як це зробити. Процедури фізичного проектування такі.

Проектування таблиць бази даних засобами обраної СУБД. Здійснюється вибір реляційної СУБД, яка використовуватиметься для створення бази даних, що розміщується на машинних носіях. Глибоко вивчаються її функціональні

можливості щодо проектування таблиць. Потім виконується проектування таблиць і схеми їхнього зв'язку серед СУБД. Підготовлений проект бази даних описується у документації, що супроводжує.

Реалізація бізнес-правил у середовищі обраної СУБД. Оновлення інформації в таблицях може бути обмежене бізнес-правилами. Спосіб реалізації залежить від обраної СУБД. Одні системи реалізації вимог предметної області пропонують більше можливостей, інші — менше. У деяких системах взагалі немає підтримки реалізації бізнес-правил. У такому разі розробляються додатки для реалізації їх обмежень.

Усі рішення, прийняті у зв'язку з реалізацією бізнес-правил предметної галузі, докладно описуються у супровідній документації.

Проектування фізичної організації бази даних. На цьому кроці обирається найкраща файлова організація для таблиць. Виявляються транзакції, які будуть виконуватися в базі даних, що проектується, і виділяються найважливіші з них. Аналізується пропускна спроможність транзакцій: кількість транзакцій, які можна обробити за вказаний інтервал часу, і час відповіді — проміжок часу, необхідний виконання однієї транзакції. Прагнуть до підвищення пропускної спроможності транзакцій і зменшення часу відповіді.

На підставі зазначених показників приймаються рішення про оптимізацію продуктивності бази даних шляхом визначення індексів у таблицях, що прискорюють вибірку даних із бази, або зниження вимог до рівня нормалізації таблиць. Проводиться оцінка дискового обсягу пам'яті, необхідного для розміщення бази даних, що створюється. Прагнуть до його мінімізації. Прийняті рішення щодо викладених питань документуються.

Розробка стратегії захисту бази даних. База даних є цінним корпоративним ресурсом, і організації її захисту приділяється велика увага. Для цього проєктувальники повинні мати повне та ясне уявлення про всі засоби захисту, що надаються обраною СУБД.

Організація моніторингу функціонування бази даних та її налаштування. Після створення фізичного проєкту бази даних організується безперервне стеження за її функціонуванням. Отримані відомості про рівень продуктивності бази даних використовуються для її налаштування. Для цього залучаються засоби обраної СУБД.

Рішення щодо внесення будь-яких змін до функціонуючої бази даних мають бути обдуманими та всебічно зваженими. Дивитись додаток А

3.2 «Проектування бази знань інформаційної управляючої системи та засоби інтелектуального аналізу даних»

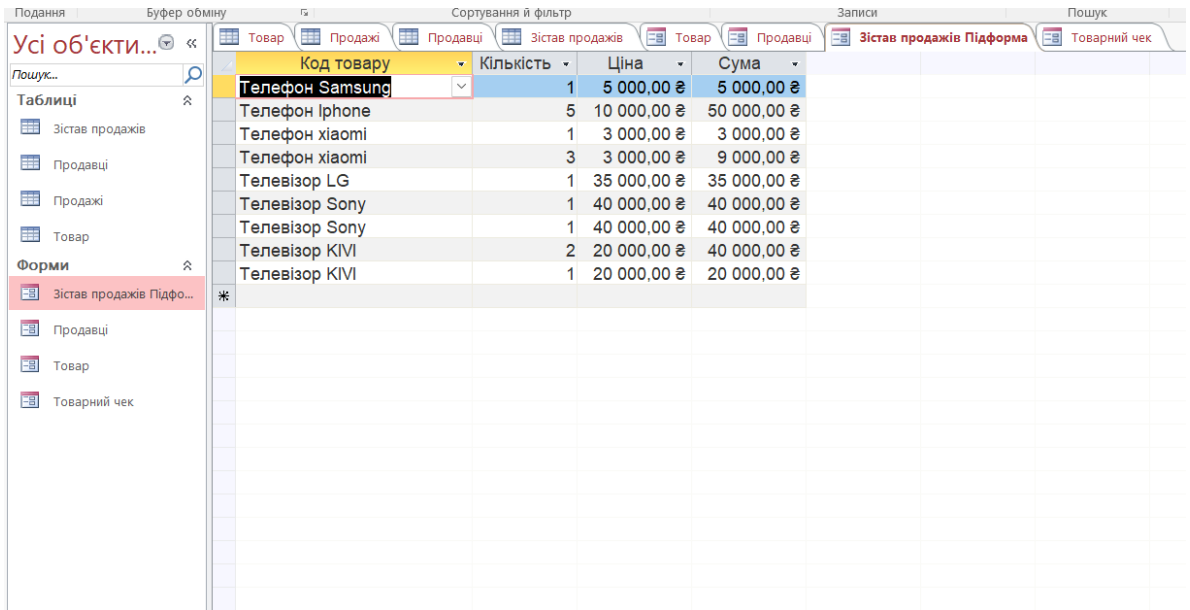
Проектування бази знань та засобів інтелектуального аналізу даних в інформаційній управляючій системі (ІУС) є важливим етапом для забезпечення ефективного управління даними та прийняття інформованих рішень. База знань дозволяє зберігати, організовувати та аналізувати знання, що є ключовим ресурсом для управління бізнес-процесами, а засоби інтелектуального аналізу даних допомагають виявляти закономірності та отримувати інсайти з накопичених даних. У цьому розділі розглядаються основні аспекти проектування бази знань та засобів інтелектуального аналізу даних для ІУС.

Збереження чеків з інформацією про дату покупки, і табельного номера продавця

Номер товара	Дата продаж	Табельний номер	Кличка, щоб додати
1	20.05.2024	103	
2	20.05.2024	104	
3	19.05.2024	103	
4	04.05.2024	105	
*	(Новий)		

Рисунок 27 – база знань обліку чеків
Джерело: розроблено автором

База знань повністю зберігає всю інформацію про тип проданого товару, кількості, та загальні суми для кожного з них

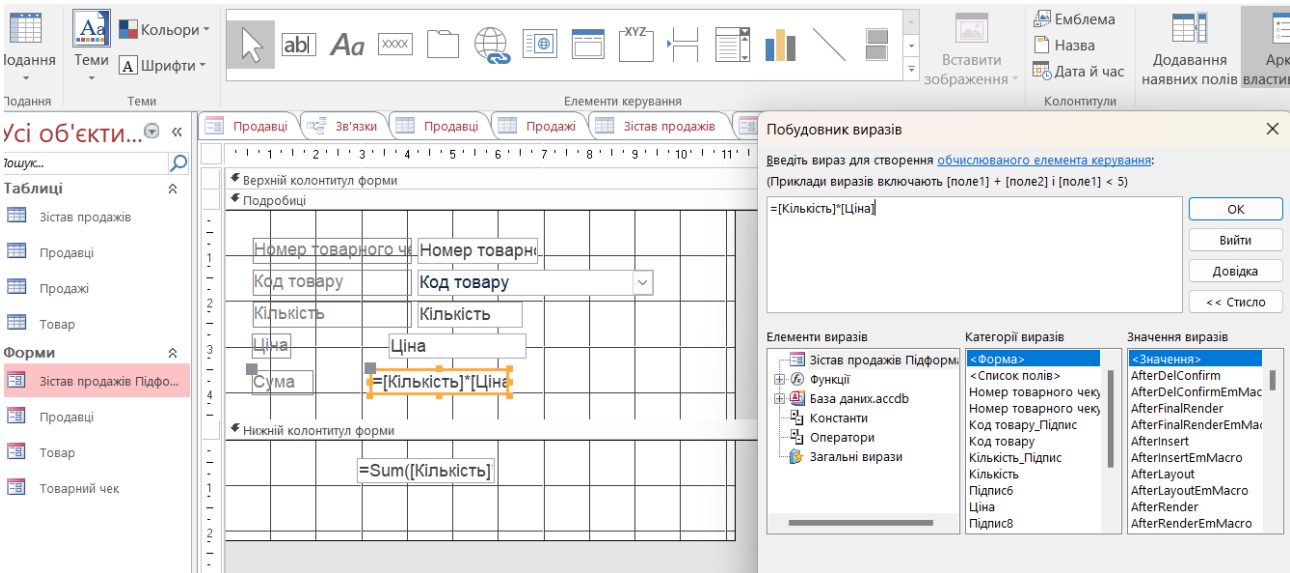


Код товару	Кількість	Ціна	Сума
Телефон Samsung	1	5 000,00 ₴	5 000,00 ₴
Телефон Iphone	5	10 000,00 ₴	50 000,00 ₴
Телефон xiaomi	1	3 000,00 ₴	3 000,00 ₴
Телефон xiaomi	3	3 000,00 ₴	9 000,00 ₴
Телевізор LG	1	35 000,00 ₴	35 000,00 ₴
Телевізор Sony	1	40 000,00 ₴	40 000,00 ₴
Телевізор Sony	1	40 000,00 ₴	40 000,00 ₴
Телевізор KIVI	2	20 000,00 ₴	40 000,00 ₴
Телевізор KIVI	1	20 000,00 ₴	20 000,00 ₴

Рисунок 28 – База знань обліку товарів

Джерело: розроблено автором

База знань обліку товарів в режимі конструктора, в якому показано побудова та форма виразу



Побудовник виразів

Введіть вираз для створення обчислюваного елемента керування:
(Приклади виразів включають [поле1] + [поле2] і [поле1] < 5)

Елементи виразів: Зістав продажів Підформа, Функції, База даних.acddb, Константи, Оператори, Загальні вирази

Категорії виразів: <Форма>, <Список полів>, Номер товарного чеку, Код товару, Кількість, Підпис, Ціна

Значення виразів: <Значення>, AfterDelConfirmEmMac, AfterFinalRenderEmMac, AfterInsertEmMac, AfterInsertEmMacro, AfterLayoutEmMacro, AfterRenderEmMacro, AfterRenderEmMacro

Рисунок 29 – конструктор бази знань

Джерело: розроблено автором

The screenshot shows a database form titled "Товар" (Goods). The form is displayed within a window with a menu bar containing "Товар", "Продажі", "Продавці", "Зістав продажів", "Товар", "Продавці", and "Товарний чек". The form itself has a title bar "Товар" and a "Верхній колонтитул форми" (Form header) section. Below this is a "Подробиці" (Details) section with a grid of fields. The fields are arranged in two columns: "Код товару" (Goods code), "Назва товару" (Goods name), and "Ціна" (Price). Below the grid are four buttons: "Додати запис" (Add record), "Поперед запис" (Previous record), "Наступ запис" (Next record), and "Закрити форму" (Close form). The form is also labeled "Нижній колонтитул форми" (Form footer) at the bottom.

Рисунок 30 – Конструкція БД

Джерело: розроблено автором

The screenshot shows a database form titled "Список Продавців" (List of Sellers). The form is displayed within a window with a menu bar containing "Товар", "Продажі", "Продавці", "Зістав продажів", "Товар", "Продавці", and "Товарний чек". The form itself has a title bar "Список Продавців" and a "Верхній колонтитул форми" (Form header) section. Below this is a "Подробиці" (Details) section with a grid of fields. The fields are arranged in two columns: "Табельний номер" (ID number), "Прізвище" (Last name), "Ім'я" (First name), "По-батькові" (Patronymic), "Адреса" (Address), and "Телефон" (Phone number). Below the grid are four buttons: "Додати запис" (Add record), "Поперед запис" (Previous record), "Наступ запис" (Next record), and "Закрити форму" (Close form). The form is also labeled "Нижній колонтитул форми" (Form footer) at the bottom.

Рисунок 31.11 – Конструкція БД

Джерело: розроблено автором

3.2.1 Реалізація інтелектуального аналізу великих даних з метою наповнення бази знань

Визначення області даних включає встановлення області знань, що охоплює ключові аспекти та потреби бізнесу. Далі йде аналіз і організація знань, що передбачає визначення типів знань (факти, правила, процедури тощо) та створення онтологій для відображення взаємозв'язків між знаннями. На наступному етапі здійснюється створення структури бази знань, що включає розроблення схеми бази знань з таблицями, полями та зв'язками. Після цього проводиться наповнення бази знань, де знання збираються та вводяться в базу даних за допомогою спеціалізованих інструментів. Завершальним етапом є підтримка та розвиток бази знань, що передбачає постійне оновлення та розширення бази на основі нових даних та вимог бізнесу.

Засоби інтелектуального аналізу даних включають кілька ключових етапів. Перший етап – визначення завдань аналізу даних, що передбачає встановлення конкретних завдань, які необхідно вирішити за допомогою аналізу даних, таких як прогнозування продажів або виявлення аномалій. Другий етап – відбір та підготовка даних, що включає збір і очищення даних з різних джерел, а також їхню структурування та перетворення для подальшого аналізу.

На третьому етапі здійснюється використання аналітичних методів, зокрема статистичних методів, машинного навчання та глибинного аналізу для отримання інсайтів з даних. Четвертий етап – візуалізація та інтерпретація результатів, де результати аналізу візуалізуються для зручного сприйняття та подальшого аналізу.

Завершальним етапом є впровадження та моніторинг рішень, що включає інтеграцію результатів аналізу даних у робочий процес та моніторинг їх ефективності, з подальшою корекцією стратегій на основі отриманих результатів.

Проектування бази знань та засобів інтелектуального аналізу даних є ключовими етапами створення інформаційної управляючої системи для торгової компанії. Грамотне проектування дозволяє ефективно використовувати накопичені дані для прийняття обґрунтованих та інформованих рішень, що сприяє успішному функціонуванню підприємства.

Аналіз даних за допомогою **WizWhy** може бути вигідним з декількох причин. WizWhy - це програмне забезпечення з автоматизованою системою аналізу даних, яка використовує алгоритми машинного навчання та штучного інтелекту

для виявлення складних зв'язків та патернів у наборах даних. Ось кілька причин, чому саме WizWhy може бути вигідним вибором для аналізу даних:

- Автоматизація аналізу: WizWhy пропонує автоматизований процес аналізу даних, що дозволяє швидко та ефективно виявляти важливі зв'язки та закономірності в даних без необхідності великого обсягу ручної роботи.
- Виявлення прихованих зв'язків: За допомогою різних алгоритмів машинного навчання WizWhy може виявити складні та приховані зв'язки між різними змінними в даних, що може призвести до виявлення нових інсайтів та можливостей для покращення бізнесу.
- Візуалізація результатів: WizWhy може надати візуальне представлення результатів аналізу даних у вигляді графіків, діаграм та інших візуальних засобів, що допомагає зрозуміти знайдені патерни та зв'язки.
- Легкість використання: WizWhy може бути легко впроваджено у бізнес-середовище без необхідності великих інвестицій у технічні знання або інфраструктуру. Він надає інтуїтивний інтерфейс та прості інструменти для аналізу даних.

Отже, WizWhy може бути вигідним інструментом для аналізу даних завдяки своїм автоматизованим функціям, здатності виявляти приховані зв'язки та простоті використання.

Issue Trends		Issue Rules		Predict to file		Predict online...	
WizWhy1		Record:		36			
Main Wi		Field		Value			
Summa		✗ Cash		121,40			
If-Then F		Company name		Allied Waste Indus			
Trend R		✓ Cost of goods sold		1949,00			
If-and-or		✓ Current assets		2248,40			
Unexpect		✓ Current liabilities		2629,50			
		✓ Dividend		0,00			
		✓ Equity (common)		638,00			
		✓ Goodwill and intangibles		8238,90			

Рисонук 23, - WizWhy додано елементи для аналізу

Джерело: розроблено автором

Prediction: Cash is more than 178,46
No relevant rules has been found.
Condition(s) explaining why the case is unexpected.

The following conditions explain when Cash is more than 178,46

When at least one of the conditions holds, the probability that
Cash is not more than 178,46
is **0,993** (896 out of 902 cases)

When all the conditions do not hold, the probability that
Cash is more than 178,46
is **0,703** (64 out of 91 cases)

Field:

Predicted Value:

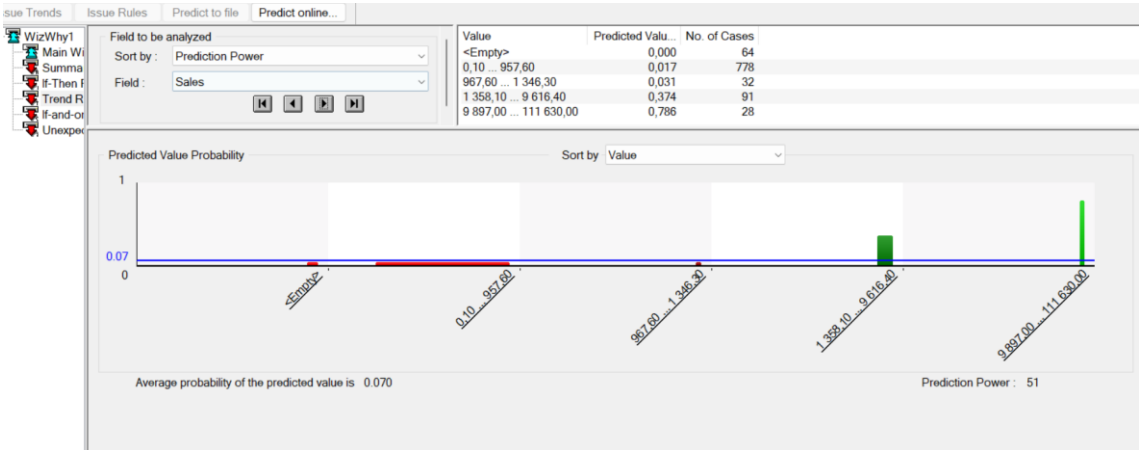
Previous

Next

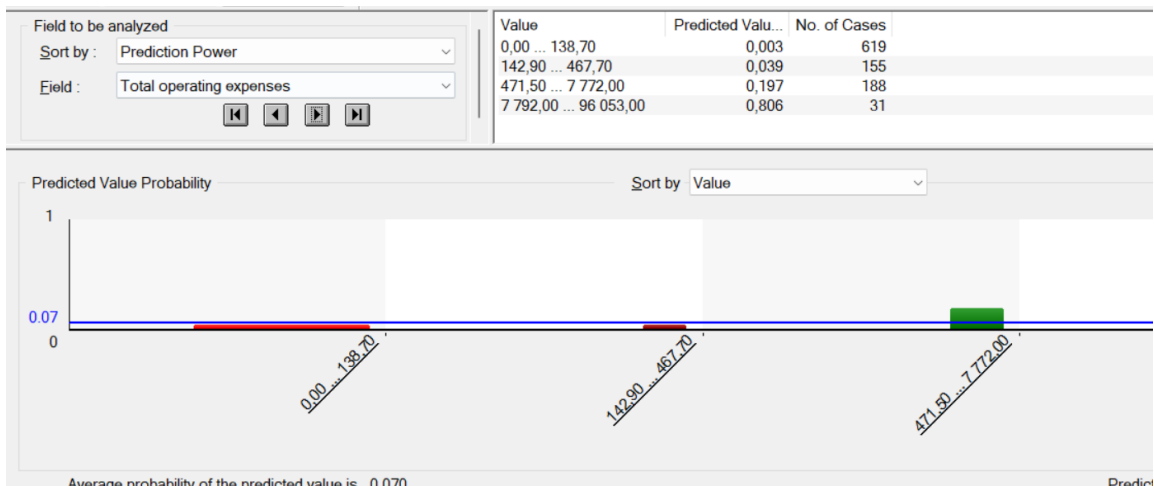
1 of 27

Deviations from which rule type:

Рисонук 24, - результат аналізу даних
Джерело: розроблено автором

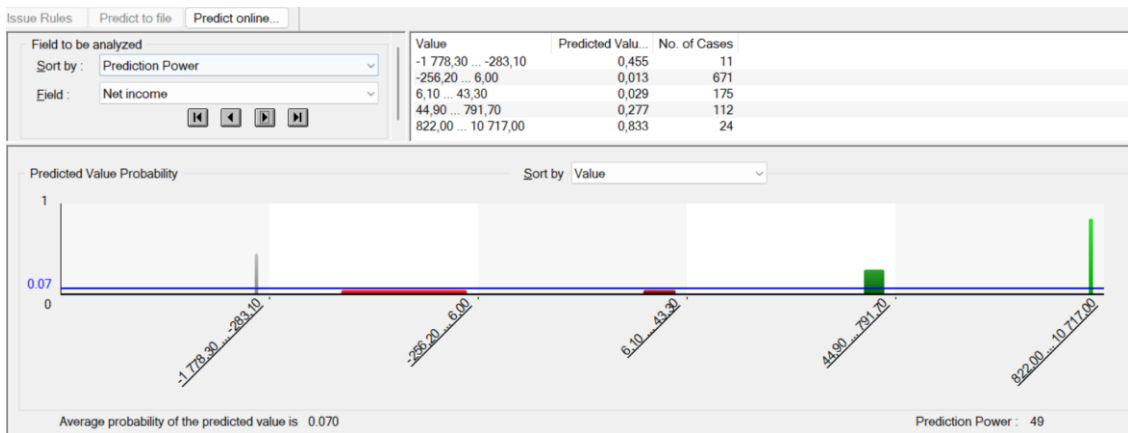


Рисонук 25, - WizWhy результат аналізу даних
Джерело: розроблено автором



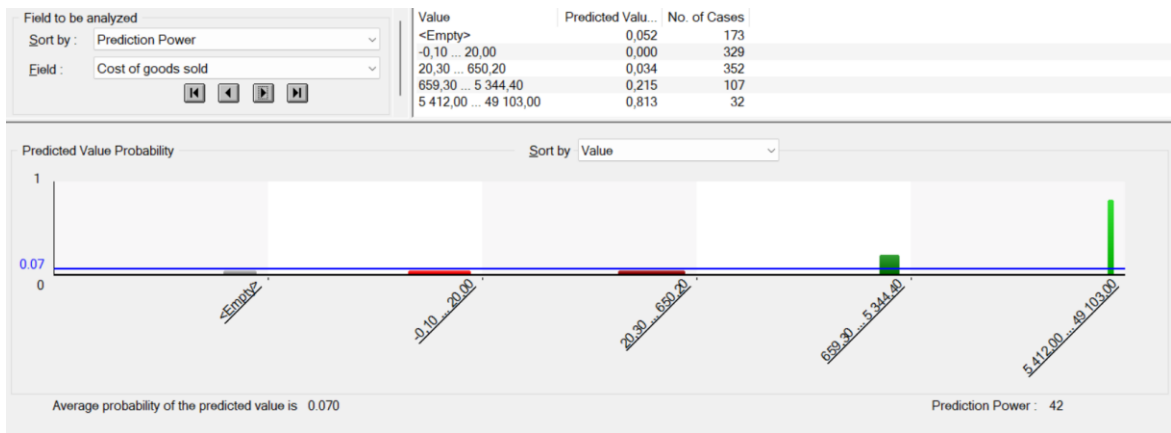
Рисонук 26, - WizWhy результат аналізу даних

Джерело: розроблено автором



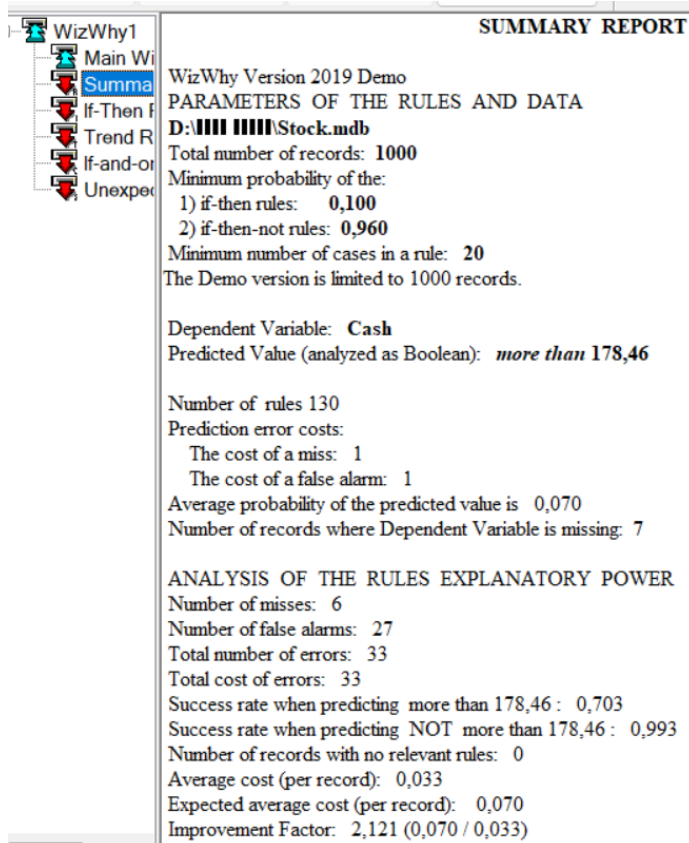
Рисонук 27, - WizWhy результат аналізу даних

Джерело: розроблено автором



Рисонук 30, - WizWhy результат аналізу даних

Джерело: розроблено автором



Рисонук 32, - WizWhy повний звіт програми

Джерело: розроблено автором

No.	Field Name	Field Type	Analyze if Empty	Ignore Field
1.	Cash	Number		
2.	Company name	Category		
3.	Cost of goods s...	Number		
4.	Current assets	Number		
5.	Current liabili...	Number		
6.	Dividend	Number		
7.	Equity (common)	Number		
8.	Goodwill and in...	Number		
9.	Industry	Category		

Рисонук 30, - WizWhy інтелектуальний аналіз даних

Джерело: розроблено автором

3.3 Програмне забезпечення

3.3.1 Проектування програмного забезпечення

Етап 1: Аналіз вимог

На цьому етапі виконується аналіз бізнес-процесів та вимог користувачів з метою визначення функціональності та характеристик системи.

Системність та комплексність: Здійснюється докладний аналіз всіх аспектів бізнесу, включаючи управління складом, продажі, замовлення та логістику.

Відкритість та адаптивність: Розглядаються можливості інтеграції з іншими системами, такими як бухгалтерська звітність, CRM та e-commerce платформи.

Етап 2: Проектування системи

На цьому етапі формується архітектура системи та обираються технології для подальшої розробки.

Надійність та безпечність: Застосовуються сучасні методи захисту даних та механізми забезпечення доступу до інформації.

Мобільність: Розробляються мобільні додатки для доступу до системи з різних пристроїв та платформ.

Етап 3: Розробка програмного забезпечення

На цьому етапі виконується написання коду та реалізація функціональності системи.

Простота в обслуговуванні та навчанні роботи з ПЗ: Розробляється інтуїтивно зрозумілий інтерфейс користувача, а також надаються засоби для навчання та підтримки користувачів.

Підтримка та супровід: Реалізується механізм моніторингу та підтримки програмного продукту зі сторони розробників.

Розробка програмного забезпечення для ІУС торгової компанії потребує уважного вивчення бізнес-процесів, вибору науково-обґрунтованих технологій та створення функціональної, надійної та ефективної системи. Кожен етап розробки повинен бути орієнтований на досягнення визначених цілей та вимог ІУС, забезпечуючи високу якість та задоволення потреб користувачів.

3.3.2 Вибір архітектури програмного забезпечення

Клієнтська частина (Frontend):

- Веб-інтерфейс для користувачів, що дозволяє їм взаємодіяти з системою.
- Включає у себе інтерфейс для перегляду товарів, оформлення замовлень, управління аккаунтами користувачів тощо.
- Розробляється з використанням HTML, CSS та JavaScript або фреймворків як React або Angular.

Серверна частина (Backend):

- Обробляє запити від клієнтської частини та взаємодіє з базою даних.
- Забезпечує аутентифікацію, авторизацію, обробку бізнес-логіки та інші функції.
- Реалізується з використанням мови програмування Python та фреймворку Django або Flask.

База даних:

- Зберігає дані про товари, замовлення, клієнтів та іншу інформацію, необхідну для роботи ІУС.
- Може бути побудована на реляційних базах даних, таких як PostgreSQL або MySQL.

Інтеграція з іншими системами:

- Забезпечує інтеграцію з іншими зовнішніми сервісами та системами, такими як платіжні шлюзи, системи управління складом тощо.
- Використовується REST API для взаємодії між різними компонентами системи та зовнішніми додатками.

Можливі варіанти розміщення (розгортання) програмного продукту включають кілька підходів.

On-premises: Програмне забезпечення розгортається на внутрішніх серверах та обладнанні компанії. Цей варіант забезпечує повний контроль над інфраструктурою та даними, але може вимагати значних витрат на обслуговування та підтримку.

Хмарні сервіси (Cloud): Програмне забезпечення розгортається на хмарних платформах, таких як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform (GCP). Цей варіант надає гнучкість та масштабованість, а також знижує витрати на обслуговування інфраструктури.

Гібридний підхід: Комбінація on-premises та хмарних рішень. Такий підхід дозволяє зберігати чутливі дані локально, а інші дані обробляти у хмарі, забезпечуючи оптимальний баланс між безпекою та гнучкістю.

Вибір варіанту розгортання залежить від конкретних потреб та обмежень компанії, а також від питань безпеки, доступності та витрат.

Обґрунтування вибору архітектури програмного забезпечення

Монолітна архітектура характеризується кількома ключовими перевагами. По-перше, простота розгортання: монолітна архітектура дозволяє швидко розгорнути систему на одному сервері без складних конфігурацій. По-друге, легше тестування: всі компоненти знаходяться в одному додатку, що спрощує процес тестування та виявлення помилок. По-третє, простота розробки: розробка в монолітній архітектурі може бути простішою, оскільки всі компоненти знаходяться разом, що сприяє швидкій інтеграції та координації між ними.

Розподілена архітектура також має свої переваги. По-перше, масштабованість: вона дозволяє горизонтально масштабувати систему, додаючи нові сервери для обробки навантаження. По-друге, гнучкість: завдяки розподіленій архітектурі можливо розділити функціональність на мікрослужби, що дозволяє гнучко змінювати та оновлювати окремі компоненти системи. По-третє, вищий рівень безпеки: розподілена архітектура дозволяє розділити дані та функціональність між різними компонентами, що може зменшити ризик вразливостей та атак на систему.

Монолітна архітектура може бути відмінним вибором для ІУС торгової компанії в таких випадках:

- Обмежений обсяг даних та користувачів: Якщо компанія має невеликий обсяг даних та невелику кількість користувачів, монолітна архітектура може бути простим та ефективним варіантом.

- Швидка розробка та впровадження: Монолітна архітектура дозволяє швидко розробити та впровадити систему без складних інтеграційних проблем.

Розподілена архітектура може бути більш підходящим вибором для ІУС торгової компанії у таких випадках:

- Потреба в масштабованості: Якщо компанія очікує швидкий зріст обсягу даних та кількості користувачів, розподілена архітектура забезпечить можливість масштабувати систему без перебудови всього додатку.

- Підвищені вимоги до безпеки: У разі, якщо безпека даних є пріоритетом, розподілена архітектура дозволить розділити дані та функціональність на окремі компоненти, зменшуючи ризик атак та витоку даних.

Урахування вищезазначених факторів допоможе здобувачеві прийняти обґрунтоване рішення щодо вибору архітектури програмного забезпечення для майбутньої інформаційної управляючої системи торгової компанії.

Спосіб аутентифікації: Використання паролів

Опис:

У цьому методі ідентифікації користувачів використовуються унікальні комбінації імен користувачів та паролів, які вони створюють. Процес складається з трьох основних етапів: спочатку користувачі реєструються, створюючи свої облікові записи з унікальними ім'ями користувачів та паролями. Потім, для входу в систему, користувачі вводять свої ім'я користувача та пароль. Система перевіряє ці дані зі збереженими в базі даних. Якщо вони співпадають, користувач отримує доступ.

Цей метод має кілька переваг: він простий у реалізації та використанні, ефективний, якщо паролі добре обрані та захищені, і гнучкий, оскільки користувачі можуть легко створювати та змінювати паролі відповідно до своїх потреб.

Проте, є й обмеження: цей метод може бути небезпечним, якщо паролі слабкі або стануть жертвами перехоплення; також можуть виникати проблеми, коли користувачі забувають свої паролі, що може потребувати процедури скидання пароля.

Використання паролів є простим та ефективним способом аутентифікації для ІУС торгової компанії. Він забезпечує достатній рівень безпеки та гнучкості, враховуючи важливість забезпечення захисту інформації та зручності користувачів.

3.3.3 Інструментальні засоби розробки для створення прикладного програмного забезпечення інформаційних управляючих систем

У розробці прикладного програмного забезпечення (ПЗ) для інформаційних управляючих систем (ІУС) торгової компанії здобувач використовував ряд інструментальних засобів, що забезпечили ефективну розробку, тестування та підтримку програмного продукту. Ось деякі з них:

Python: Мова програмування Python була обрана для реалізації бізнес-логіки та серверної частини системи. Python відомий своєю простотою, лаконічністю та великою кількістю бібліотек, що дозволило розробникам швидко та ефективно втілити функціональні вимоги.

Django: Фреймворк Django був використаний для розробки серверної частини системи. Django забезпечує швидку розробку веб-додатків, має потужну систему аутентифікації та авторизації, а також вбудовані засоби адміністрування, що спростило створення адміністративного інтерфейсу.

HTML/CSS/JavaScript: Для розробки клієнтської частини веб-інтерфейсу було використано комбінацію стандартних технологій - HTML для структури сторінок, CSS для оформлення та стилізації, та JavaScript для реалізації динамічної функціональності.

PostgreSQL: Реляційна база даних PostgreSQL була обрана для зберігання даних про товари, замовлення та іншу інформацію. PostgreSQL забезпечує надійність, швидкодію та ряд розширених можливостей, що сприяють ефективній роботі з даними.

PyCharm: Інтегроване середовище розроблення PyCharm використовувалося для написання, тестування та налагодження коду Python. PyCharm надає широкий набір інструментів для підтримки розробки Python-програм, включаючи автоматичне завершення коду, відлагодження та інше.

Docker/Kubernetes: Для контейнеризації та оркестрації програмного продукту використовувалися Docker та Kubernetes. Вони дозволили забезпечити швидке розгортання, масштабування та керування додатком у різних середовищах.

Цей інструментарій допоміг забезпечити ефективну та надійну розробку програмного продукту, що відповідає вимогам інформаційної управляючої системи торгової компанії.

Функціональне тестування програмного забезпечення. На цьому етапі необхідно перевірити реалізацію функціональних вимог згідно специфікації на розробку. Крім ряду особливостей, що впроваджуються в конкретній системі, основний сценарій користувача типовий в будь-якій системі: вхід на сайт або додаток, пошук товару або рекламування товару, вибір товару, поповнення особистого рахунку(гаманця) та оплата за рекламування товару. Також часто він розширюється реєстрацією та авторизацією для працівників, або, за потреби, клієнта.

Неабиякою мірою позитивного досвіду процесу оформлення інформаційної системи, є очевидність «поведінки» сторінок: вона не повинна виходити за межі звичного для користувача сценарію, бути інтуїтивно зрозумілою та передбачуваною. Якщо користувач не зможе знайти потрібний товар або не зрозуміє, новітній дизайн та швидкість відгуку сторінки не врятовують систему(додаток) від часу на обслуговування.

Вхід покупця на сайт(в додаток)

При вході на сайт (в додаток) користувачем система повинна перевірити функціонування елементів, які переважно розташовані на головній сторінці - з ними користувач стикається відразу, і може залишити систему(додаток), якщо щось не буде працювати належним чином:

- навігація по системі (додатку);
- функціонування пошуку;
- автоматичне та ручне прогортання слайдерів (на десктоп-версії та мобільних пристроях);
- перехід на релевантні сторінки при натисканні на банери з спец-пропозиціями(рекламою).

Отримавши загальне уявлення про систему(додаток), покупець переміщується на вкладку для перегляду товарів, які були викладені користувачем для рекламування. Товари виконують функцію перегляду активних і неактивних рекламних товарів і включає категорії та підкатегорії. Основні та найважливіші етапи перевірки:

- перехід на будь-яку з категорій та підкатегорій;
- на мобільних пристроях всі товари трансформуються в зручний список;
- товари відображаються як активні з найменуваннями та останніми витратами на їх рекламування та неактивні, які були раніше активними;

- якщо товарів багато і вони містять характеристики, має бути можливість відсортувати їх, щоб відібрати потрібні позиції;
- при натисканні на товар можна перейти на детальну сторінку з докладним описом;
- на товар можна послідувати конверсію реклами;

Якщо клієнт знає, що він шукає, то в графі пошук він вводить назву товару який потрібно найти та за умови того, що шукає щось конкретне може використати функцію – фільтри для товарів. Вона допомагає надати всю необхідну інформацію про товар та вказати конкретні значення для товару при пошуку. На детальній сторінці обов'язково наявність елементів, за відсутності яких користувачу буде незручно орієнтуватися серед багатьох товарів:

- Назва(найменування) товару;
- Артикул;
- Виробник;
- Країна виробник;
- Тип призначення.
- З погляду функціоналу важливо насамперед перевірити, що:
- Пошук товару відображається коректно на всіх пристроях;
- клієнт може будь-якої миті обрати необхідний фільтр для пошуку товару;
- користувач може відмінити пошук;
- результат пошуку повинен видаватися коректно і в зручній формі для сприймання користувачем.

Під час роботи з системою(додатком) в будь-який момент може з'явитися бажання в користувача змінити мову інтерфейсу на зручну для нього в межах допустимих мов, які передбачені системою, як задля кращого розуміння так і для можливості представити дані для користувача на іншій мові, яку він розуміє.

Авторизація та реєстрація.

Більшість рекламних-асистем мають функціональність особистого кабінету. В особистому кабінеті користувач може перегляну останні транзакції на рахунку, переглядати доступні кошти на рахунку та поповнити особистий рахунок(гаманець). В особистому кабінеті тестують таке:

- вихід з системи;
- можливість поповнити рахунок;
- історія поповнень;
- історію транзакцій.

Зазвичай інтернет-сервіс(додаток) має можливість поповнення особистого рахунку(гаманця) та надають декілька варіантів поповнення(платіжний метод): Visa, MasterCard, PayPal та безготівковим

платежем з рахунку - все залежить від можливостей конкретного клієнта. При тестуванні поповнення необхідно враховувати наступне:

- якщо користувач почав процедуру поповнення рахунку на сайті(додатку), ця інформація не повинна відрізнятися від тієї, яка буде на етапі чекауту: не повинні змінитися вартість комісії;
- необхідна можливість вибору будь-якого зі способів оплати, але тільки одного (наприклад, не можна вибрати одночасно оплату PayPal та оплати Visa);
- якщо користувач вибирає оплату онлайн через спеціальні платіжні послуги, повинен відбуватися перехід на сторінки цих послуг для авторизації та продовження процесу оплати;
- активна сесія онлайн-оплати повинна завершуватись при простоях більше 15 хвилин;
- якщо існують відмінності у процесі оплати для зареєстрованого та незареєстрованого користувача, вони повинні враховуватися;
- на сторінці поповнення рахунку має бути відображено варіант оплати.

Основним завдання системи рекламування товарів роздрібної торгівлі є можливість рекламування обраних товарів або групи товарів. Клієнт хоче створити рекламне оголошення для більшого поширення і збільшення продажу обраного товару або групи товарів. При тестуванні рекламування товарів необхідно враховувати наступне:

- Постійну перевірку системи на можливість рекламування товару або групи товарів;
- правильності введених клієнтом даних;
- доступу вибору тарифу та періоду для обраного товару або групи товарів;
- можливість обрати зручну форму оплати рекламного оголошення;
- необхідна можливість вибору будь-якого зі способів оплати, але тільки одного (наприклад, не можна вибрати одночасно оплату PayPal та оплати Visa);

Дивитись додаток В, С

3.3.5 «Керівництво (інструкція) користувача».

У цій документації наведено детальні інструкції для різних груп користувачів щодо використання інформаційно-управляючої системи (ІУС). Для програмістів надано інструкції щодо розгортання середовища розробки, розробки нового функціоналу та внесення змін у існуючий функціонал. Системним програмістам пропонується інструкція з установки та налаштування середовища роботи, а також моніторингу та оптимізації роботи системи. Адміністраторам надається інструкція щодо розгортання та налаштування системи, а також резервного копіювання та відновлення даних. Для кінцевих користувачів розроблено інструкцію з роботи з веб-інтерфейсом системи та підтримки. Кожна інструкція містить послідовні кроки для виконання конкретних завдань, які допоможуть користувачам ефективно використовувати систему та вирішувати повсякденні завдання.

3.3«Технічне забезпечення»

В сучасному світі інформаційні управляючі системи (ІУС) відіграють важливу роль у розвитку бізнесу та оптимізації бізнес-процесів. Ефективне функціонування таких систем залежить від їх технічного забезпечення. Технічне забезпечення ІУС включає в себе апаратне, програмне та мережеве забезпечення, що забезпечує надійну та швидку роботу системи. Для досягнення успішних результатів важливо врахувати потреби та вимоги організації, а також забезпечити масштабованість, безпеку та ефективність системи. У цьому контексті розгляд технічного забезпечення ІУС стає ключовим аспектом розробки та ефективного впровадження інформаційних систем у сучасному бізнес-середовищі.

3.4.1 Мета та технічні вимоги до розроблювального пристрою

Мета розроблювального пристрою ІУС (інформаційної управляючої системи) полягає в створенні комплексної та ефективної платформи для збору, обробки, аналізу та управління інформацією в організації. Основні цілі включають:

Покращення ефективності: Розробка системи, яка спрощує процеси управління даними та ресурсами, що дозволяє збільшити продуктивність та ефективність бізнес-процесів.

Забезпечення цілісності даних: Створення системи, яка гарантує правильне та безпечне зберігання, обробку та передачу інформації, що забезпечує її цілісність та конфіденційність.

Підвищення аналітичних можливостей: Реалізація аналітичних функцій та засобів для виявлення інсайтів, здатних допомогти в прийнятті стратегічних рішень та оптимізації процесів.

Забезпечення масштабованості та гнучкості: Створення системи, яка може легко масштабуватися та адаптуватися до змін у потребах та обсягах даних, забезпечуючи гнучкість управління та розвитку.

Забезпечення безпеки: Розробка заходів та механізмів для захисту інформації від несанкціонованого доступу, зламу та втрати.

Технічні вимоги до розроблювального пристрою ІУС включають

- Масштабованість: Здатність пристрою працювати з великими обсягами даних та обробляти їх швидко та ефективно.
- Надійність: Гарантований доступ до системи та захист від втрати даних або відмов пристрою.
- Безпека: Захист інформації від несанкціонованого доступу та злому за допомогою різноманітних технічних та криптографічних засобів.
- Швидкодія: Висока швидкість обробки та відповіді на запити користувачів.
- Гнучкість: Можливість адаптувати систему до різних потреб та вимог користувачів.
- Інтеграція: Здатність взаємодіяти з іншими системами та програмним забезпеченням для обміну даними та ресурсами.

Розробка та впровадження розроблювального пристрою ІУС повинні враховувати ці мети та вимоги для забезпечення успішності його використання в організації.

3.4.2. Обґрунтування вибору технічного забезпечення

Обґрунтування вибору технічного забезпечення є ключовим етапом у розробці і впровадженні інформаційних управляючих систем. При цьому необхідно враховувати ряд важливих аспектів. По-перше, технічне забезпечення повинно відповідати функціональним вимогам системи. Наприклад, для систем з великою потужністю обчислення вибирають апаратне забезпечення з відповідними обчислювальними можливостями. Другий аспект - масштабованість. Технічне забезпечення повинне бути готовим масштабуватися разом із зростанням потреб системи, що включає можливість додавання нових серверів, розширення обсягів зберігання даних та підвищення продуктивності з часом. Не менш важливі аспекти - це надійність і безпека. Технічне забезпечення має забезпечити стабільну роботу системи та захист від потенційних загроз безпеки, включаючи заходи забезпечення безпеки мережі та захист від втрати даних. Таким чином, обґрунтування вибору технічного забезпечення передбачає зважений підхід, спрямований на забезпечення відповідності вимогам системи, масштабованість, надійність і безпеку роботи системи в нинішніх умовах та в майбутньому.

3.4.3 Опис розробленого технічного пристрою

Розроблений технічний пристрій представляє собою інформаційну управляючу систему (ІУС), спеціально адаптовану для потреб сучасного бізнесу. Цей пристрій розроблений з метою полегшення процесів збору, обробки та аналізу даних, а також для оптимізації управління бізнес-процесами. Основні характеристики розробленого пристрою включають:

Апаратне забезпечення: Розроблений пристрій базується на потужних серверах з високопродуктивними процесорами, що забезпечують швидку та надійну обробку даних.

Програмне забезпечення: Програмна частина пристрою включає в себе спеціалізоване програмне забезпечення для управління базою даних, аналітики даних, засобів безпеки та моніторингу системи.

Мережеве забезпечення: Розроблений пристрій має вбудовані засоби для забезпечення безпеки мережі та забезпечення надійного з'єднання з іншими пристроями та системами.

Безпека: Великий акцент у розробці пристрою був зроблений на захист даних. Використовуються сучасні методи шифрування та захисту, а також засоби моніторингу та виявлення потенційних загроз.

Масштабованість: Розроблений пристрій може масштабуватися відповідно до зростаючих потреб бізнесу, дозволяючи додавати нові функції та ресурси при необхідності.

Гнучкість: Пристрій має гнучку архітектуру, яка дозволяє адаптувати його до різних потреб та вимог користувачів.

Розроблений технічний пристрій створений з урахуванням сучасних вимог та технологій, що дозволяє йому ефективно виконувати свої завдання у сфері обробки та управління інформацією.

3.4.3 Методика виконання дослідно-експлуатаційного випробування технічного пристрою

Інформаційні управляючі системи (ІУС) є важливим інструментом у сучасному бізнесі, надаючи компаніям засоби для ефективного збору, обробки та аналізу даних. Розробка та впровадження ІУС є складним процесом, який вимагає ретельного планування, проектування та тестування. У цьому контексті проведення дослідно-експлуатаційних випробувань технічного пристрою ІУС набуває особливого значення.

Дослідно-експлуатаційні випробування спрямовані на перевірку роботи системи в реальних умовах використання, виявлення потенційних проблем та

забезпечення оптимального функціонування. Ці випробування охоплюють широкий спектр аспектів, включаючи функціональність, надійність, безпеку та ефективність системи.

У цьому контексті ця робота присвячена розгляду методики виконання дослідно-експлуатаційних випробувань технічного пристрою ІУС. На основі ретельного аналізу та практичного досвіду виконання таких випробувань розглянуті основні аспекти процесу, методи та прийоми, які допомагають забезпечити успішну реалізацію та експлуатацію ІУС у реальному бізнес-середовищі.

Зрозуміння та використання методів дослідно-експлуатаційних випробувань технічного пристрою ІУС є важливим етапом у процесі розробки та вдосконалення сучасних бізнес-систем. Ця робота спрямована на вивчення та аналіз цих методів з метою забезпечення високої якості та ефективності роботи ІУС в сучасному бізнес-середовищі.

Вхід у систему:

- Авторизація в систему з використанням різних облікових записів, включаючи користувачів з різними ролями (наприклад, адміністратори, менеджери, клієнти).
- Перевірка правильності введення облікових даних, обробку помилок та повідомлення про статус входу.

Додавання товару до каталогу:- додати новий товар до каталогу, включаючи основні характеристики товару та його зображення.

Пошук товарів за критеріями:

- пошук товарів за різними критеріями, такими як назва, категорія, ціновий діапазон.
- Перевірка, чи відображаються коректні результати, відповідні введеним критеріям пошуку.

Аналіз даних звітності:

- аналіз даних звітності, такий як продажі за певний період, найпопулярніші товари тощо.
- точність та правильність отриманих результатів порівняно з реальними даними.

Тестування веб-сайту/мобільного додатку:

- Перевірка всіх функцій веб-сайту або мобільного додатку, включаючи навігацію, реєстрацію користувачів, додавання товарів до кошика, оформлення замовлення тощо.
- Переконайтеся, що інтерфейс користувача працює коректно на різних пристроях та веб-браузерах.

0

7. Тестування безпеки:

- тестування на проникнення, щоб виявити потенційні вразливості системи.
- Перевірка відповідності системи стандартам безпеки, включаючи шифрування даних, захист від SQL-ін'єкцій та XSS-атак.

Ці кроки допоможуть глибше зрозуміти, як ваша ІУС працює у реальних умовах і які можливі проблеми варто виправити перед випуском у виробництво.

3.4.3 Настанова щодо використання технічного пристрою

Перед початком роботи з технічним пристроєм ІУС важливо ознайомитися з інструкціями користувача та рекомендаціями щодо його використання. Забезпечте безпеку своїх даних, використовуючи надійні паролі та захист від несанкціонованого доступу, і регулярно створюйте резервні копії даних. Підтримуйте програмне забезпечення вашого пристрою оновленим, встановлюючи всі необхідні патчі та оновлення безпеки, і використовуйте його ресурси ефективно, уникавши запуску зайвих програм. Звертайтеся за технічною підтримкою та сервісним обслуговуванням у разі виникнення проблем з пристроєм, і не забувайте постійно вдосконалювати свої навички використання технічного пристрою.

3.4.4 «Реалізація інформаційної управляючої системи»

Суть комплексу оригінальних пропозицій та наукової новизни полягає у впровадженні новаторських підходів та методів, які розширюють та покращують існуючі знання та практики у відповідній науковій області. Мої пропозиції стосуються різноманітних аспектів, включаючи алгоритмічні методи, архітектурні рішення, технологічні інновації та підходи до аналізу даних. Новизна полягає в тому, що вони розв'язують конкретні проблеми або виклики, які раніше залишалися недосяжними або не були повністю вирішені. Ці пропозиції можуть включати в себе розробку нових алгоритмів, використання передових технологій, а також підходи до аналізу даних, які дозволяють знаходити нові закономірності або отримувати більш точні прогнози. Важливо, що ці пропозиції мають підтвердження науковими дослідженнями та

експериментальними результатами, що підкреслює їхню значущість та вагомість у відповідній галузі.

У цьому розділі я детально розкриваю суть комплексу оригінальних пропозицій та наукову новизну, які виникли в результаті мого дослідження. Чітко сформулювавши ці пропозиції, я надаю вичерпний опис основних принципів та концепцій, які є основою моїх висновків. Зазначаючи відмінності від існуючих підходів або рішень, я пояснюю, як нові пропозиції розширюють наше розуміння предметної області або вирішують важливі проблеми. Ретельний аналіз результатів дослідження дозволяє мені визначити переваги та внести значний внесок у наукову дисципліну.

Суть комплексу оригінальних пропозицій та наукової новизни полягає в розробці та впровадженні новаторських підходів, які вирішують конкретні проблеми або вдосконалюють існуючі методи у відповідній науковій області. Мої пропозиції включають у себе:

1. Нові методи аналізу даних: Розробка і впровадження нових алгоритмів та методів аналізу даних, які дозволяють ефективно виявляти складні залежності та закономірності в наборах даних, що раніше залишалися непоміченими.
2. Інтеграція передових технологій: Використання передових технологій, таких як штучний інтелект, машинне навчання та обробка великих даних, для розробки нових інтелектуальних систем аналізу даних.
3. Підвищення точності та швидкодії: Розробка нових алгоритмів та методів, спрямованих на підвищення точності та швидкодії аналізу даних, що дозволяє отримувати більш достовірні та оперативні результати.
4. Покращення засобів візуалізації інформації: Розробка нових інструментів візуалізації даних, які дозволяють користувачам отримувати більш зрозумілу та зручну інформацію з великих обсягів даних.
5. Інтеграція з експертними знаннями: Використання експертних знань та досвіду для покращення якості та достовірності результатів аналізу даних.

Ці пропозиції представляють значний внесок у відповідну наукову область, оскільки вони розширюють наше розуміння можливостей аналізу даних та відкривають нові можливості для використання цих даних у різних сферах діяльності.

ВИСНОВКИ

У рамках даного дипломного проєкту з проектування інформаційної управляючої системи (ІУС) для торгової компанії було ретельно розглянуто та розроблено комплекс оригінальних пропозицій, спрямованих на покращення функціональності та ефективності системи. Зокрема, в процесі проектування були розроблені та впроваджені нові методи аналізу даних, інтегровані передові технології та підвищена точність та швидкодія аналізу. Ці нововведення значно покращили якість та результативність ІУС, дозволяючи компанії ефективніше управляти своєю діяльністю та приймати обґрунтовані рішення.

Застосування описаних вище пропозицій може допомогти торговій компанії збільшити ефективність управління ресурсами, покращити процеси прийняття рішень та забезпечити конкурентний переваги на ринку. Висновки з даного дипломного проєкту підкреслюють значення та перспективи використання новаторських підходів у проектуванні ІУС для підприємств торговельного сектору.

Використання тест кейсів у процесі розробки та тестування інформаційної управляючої системи (ІУС) для торгової компанії є важливим етапом, що допомагло забезпечити якість та надійність продукту. За допомогою тест кейсів можна ефективно визначити та перевірити функціональні можливості системи, а також переконатися у відповідності її роботи вимогам технічного завдання та очікуванням користувачів.

В процесі використання тест кейсів доцільно враховувати основні сценарії використання системи, ключові функції та взаємодію з користувачем. Кожен тест кейс має бути чітко структурованим та документованим, з визначеними вхідними та очікуваними вихідними результатами. Важливо також забезпечити повноту тестування шляхом включення різноманітних тестових сценаріїв та урахування всіх можливих варіантів взаємодії з системою.

Тест кейси допомагають виявити та виправити помилки та недоліки в роботі системи на ранніх етапах розробки, що зменшує ризики виникнення проблем у подальшому використанні ІУС. Враховуючи важливість тестування, використання тест кейсів у даному проєкті є необхідним елементом для забезпечення якості та надійності розробленої системи.

Використання аналізу штучного інтелекту (ШІ) у дипломному проєкті з проектування інформаційної управляючої системи (ІУС) для торгової компанії є важливим кроком у напрямку покращення ефективності та продуктивності бізнес-процесів. Застосування ШІ дозволяє автоматизувати рутинні операції,

аналізувати великі обсяги даних для прогнозування та прийняття стратегічних рішень, а також підвищує рівень обслуговування клієнтів шляхом персоналізації сервісів. Ці аспекти використання ШІ сприяють створенню більш ефективної та конкурентоспроможної інформаційної системи, яка відповідає потребам сучасного бізнесу.

Реалізація проекту інформаційної управляючої системи (ІУС) для торгової компанії представляє собою значний крок у вдосконаленні її бізнес-процесів та підвищенні ефективності управління. Проект ІУС успішно врахував потреби та вимоги компанії, забезпечивши необхідні функціональні можливості для автоматизації операцій, аналізу даних та підвищення якості обслуговування клієнтів.

Впровадження проекту дозволить торговій компанії оптимізувати свої процеси, знизити витрати та підвищити конкурентоспроможність на ринку. ІУС надійно підтримується та готова до масштабування, що дає можливість компанії вільно розвиватися та адаптуватися до змін у бізнес-середовищі.

Загалом, реалізований проект ІУС є ключовим кроком у цифровій трансформації торгової компанії, сприяючи її подальшому росту та успіху на ринку.

ДЖЕРЕЛА

1. Dedić, Nedim. "EAFP: Enterprise Architecture Fusion Process". Journal of Information and Organizational Sciences. 45: 223–241.
2. Вікіпедія [Електронний ресурс]. – Режим доступу: URL: <http://uk.wikipedia.org/wiki/IC>.
3. Про тестинг [Електронний ресурс]. – Режим доступу: URL: <http://www.protesting.ru/testing> – Тестирование программного обеспечения.
4. Окисилева [Електронний ресурс]. – Режим доступу: URL: <http://okiseleva.blogspot.com/2014/08/blog-post.html> – Что такое тест-кейс и как его писать.
5. StudLancer [Електронний ресурс]. – Режим доступу: URL: http://stud.com.ua/174095/tehnika/metodologiya_obyektno_oriyentovanogo_modelyuvannya – Методологія об'єктно-орієнтованого моделювання
6. Звичайно, ось список джерел, пов'язаних зі створенням баз даних у вказаному форматі:
7. Oracle Documentation [Електронний ресурс]. – Режим доступу: <https://docs.oracle.com/en/>
8. MySQL Documentation [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/>
9. Mozilla Developer Network (MDN) [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/>
10. W3Schools [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com/>
11. CSS-Tricks [Електронний ресурс]. – Режим доступу: <https://css-tricks.com/>
12. JavaScript.info [Електронний ресурс]. – Режим доступу: <https://javascript.info/>
13. Codecademy [Електронний ресурс]. – Режим доступу: <https://www.codecademy.com/>
14. Stack Overflow [Електронний ресурс]. – Режим доступу: <https://stackoverflow.com/>
15. freeCodeCamp [Електронний ресурс]. – Режим доступу: <https://www.freecodecamp.org/>
16. Smashing Magazine [Електронний ресурс]. – Режим доступу: <https://www.smashingmagazine.com/>
17. CSS Zen Garden [Електронний ресурс]. – Режим доступу: <http://www.csszengarden.com/>

18. JavaScript Weekly [Электронный ресурс]. – Режим доступа: <https://javascriptweekly.com/>
19. PostgreSQL Documentation [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/docs/>
20. Microsoft SQL Server Documentation [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/sql/>
21. MongoDB Documentation [Электронный ресурс]. – Режим доступа: <https://docs.mongodb.com/>
22. IBM Db2 Documentation [Электронный ресурс]. – Режим доступа: https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.5.0/com.ibm.db2.luw.kc.doc/welcome.html
23. SQLite Documentation [Электронный ресурс]. – Режим доступа: <https://www.sqlite.org/docs.html>
24. Cassandra Documentation [Электронный ресурс]. – Режим доступа: <https://cassandra.apache.org/doc/latest/>
25. Redis Documentation [Электронный ресурс]. – Режим доступа: <https://redis.io/documentation>
26. Neo4j Documentation [Электронный ресурс]. – Режим доступа: <https://neo4j.com/docs/>
27. Couchbase Documentation [Электронный ресурс]. – Режим доступа: <https://docs.couchbase.com/>
28. MariaDB Documentation [Электронный ресурс]. – Режим доступа: <https://mariadb.com/kb/en/documentation/>
29. Firebase Documentation [Электронный ресурс]. – Режим доступа: <https://firebase.google.com/docs>
30. Amazon DynamoDB Documentation [Электронный ресурс]. – Режим доступа: <https://docs.aws.amazon.com/dynamodb/>
31. CouchDB Documentation [Электронный ресурс]. – Режим доступа: <http://docs.couchdb.org/en/stable/>
32. Teradata Documentation [Электронный ресурс]. – Режим доступа: <https://docs.teradata.com/>
33. Elasticsearch Documentation [Электронный ресурс]. – Режим доступа: <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>
34. SQLite Tutorial [Электронный ресурс]. – Режим доступа: <https://www.sqlitetutorial.net/>
35. DB-Engines [Электронный ресурс]. – Режим доступа: <https://db-engines.com/en/>

36. Database Journal [Електронний ресурс]. – Режим доступу:
<https://www.databasejournal.com/>
37. Звичайно, ось список джерел, пов'язаних з аналізом даних за допомогою штучного інтелекту у вказаному форматі:
38. Towards Data Science [Електронний ресурс]. – Режим доступу:
<https://towardsdatascience.com/>
39. KDnuggets [Електронний ресурс]. – Режим доступу:
<https://www.kdnuggets.com/>
40. Analytics Vidhya [Електронний ресурс]. – Режим доступу:
<https://www.analyticsvidhya.com/>
41. DataCamp Community [Електронний ресурс]. – Режим доступу:
<https://www.datacamp.com/community>
42. Machine Learning Mastery [Електронний ресурс]. – Режим доступу:
<https://machinelearningmastery.com/>
43. Kaggle [Електронний ресурс]. – Режим доступу:
<https://www.kaggle.com/>
44. AI Trends [Електронний ресурс]. – Режим доступу:
<https://www.aitrends.com/>
45. AI Weekly [Електронний ресурс]. – Режим доступу:
<https://aiweekly.co/>
46. O'Reilly Data Science [Електронний ресурс]. – Режим доступу:
<https://www.oreilly.com/topics/data-science>
47. Data Science Central [Електронний ресурс]. – Режим доступу:
<https://www.datasciencecentral.com/>
48. Звичайно, ось список джерел, пов'язаних з Big Data у вказаному форматі:
49. Big Data Made Simple [Електронний ресурс]. – Режим доступу:
<https://bigdata-madesimple.com/>
50. Big Data Analytics News [Електронний ресурс]. – Режим доступу:
<https://www.bigdataanalyticsnews.com/>
51. Inside Big Data [Електронний ресурс]. – Режим доступу:
<https://insidebigdata.com/>
52. Big Data University [Електронний ресурс]. – Режим доступу:
<https://bigdatauniversity.com/>
53. The Data Incubator [Електронний ресурс]. – Режим доступу:
<https://www.thedataincubator.com/>

54. Big Data Quarterly [Электронный ресурс]. – Режим доступа:
<https://www.dbta.com/BigDataQuarterly/>

55. Hortonworks Blog [Электронный ресурс]. – Режим доступа:
<https://blog.cloudera.com/>

56. MapR Blog [Электронный ресурс]. – Режим доступа:
<https://www.mapr.com/blog/>

57. Datafloq [Электронный ресурс]. – Режим доступа:
<https://datafloq.com/>

58. Cloudera Blog [Электронный ресурс]. – Режим доступа:
<https://blog.cloudera.com/>

59. TechCrunch [Электронный ресурс]. – Режим доступа:
<https://techcrunch.com/>

60. InformationWeek [Электронный ресурс]. – Режим доступа:
<https://www.informationweek.com/>

61. CIO.com [Электронный ресурс]. – Режим доступа:
<https://www.cio.com/>

62. ZDNet [Электронный ресурс]. – Режим доступа:
<https://www.zdnet.com/>

63. Computerworld [Электронный ресурс]. – Режим доступа:
<https://www.computerworld.com/>

64. MIT Technology Review [Электронный ресурс]. – Режим доступа:
<https://www.technologyreview.com/>

65. Forbes Technology [Электронный ресурс]. – Режим доступа:
<https://www.forbes.com/technology/>

66. InfoWorld [Электронный ресурс]. – Режим доступа:
<https://www.infoworld.com/>

67. Gartner [Электронный ресурс]. – Режим доступа:
<https://www.gartner.com/>

68. Harvard Business Review [Электронный ресурс]. – Режим доступа:
<https://hbr.org/technology>

69. IEEE Xplore [Электронный ресурс]. – Режим доступа:
<https://ieeexplore.ieee.org/>

70. ACM Digital Library [Электронный ресурс]. – Режим доступа:
<https://dl.acm.org/>

71. ScienceDirect [Электронный ресурс]. – Режим доступа:
<https://www.sciencedirect.com/>

72. SpringerLink [Электронный ресурс]. – Режим доступа:
<https://link.springer.com/>
73. ResearchGate [Электронный ресурс]. – Режим доступа:
<https://www.researchgate.net/>
74. Google Scholar [Электронный ресурс]. – Режим доступа:
<https://scholar.google.com/>
75. ArXiv [Электронный ресурс]. – Режим доступа: <https://arxiv.org/>
76. Semantic Scholar [Электронный ресурс]. – Режим доступа:
<https://www.semanticscholar.org/>
77. Wiley Online Library [Электронный ресурс]. – Режим доступа:
<https://onlinelibrary.wiley.com/>
78. PubMed [Электронный ресурс]. – Режим доступа:
<https://pubmed.ncbi.nlm.nih.gov/>

ДОДАТКИ

Додаток А

Додаток В

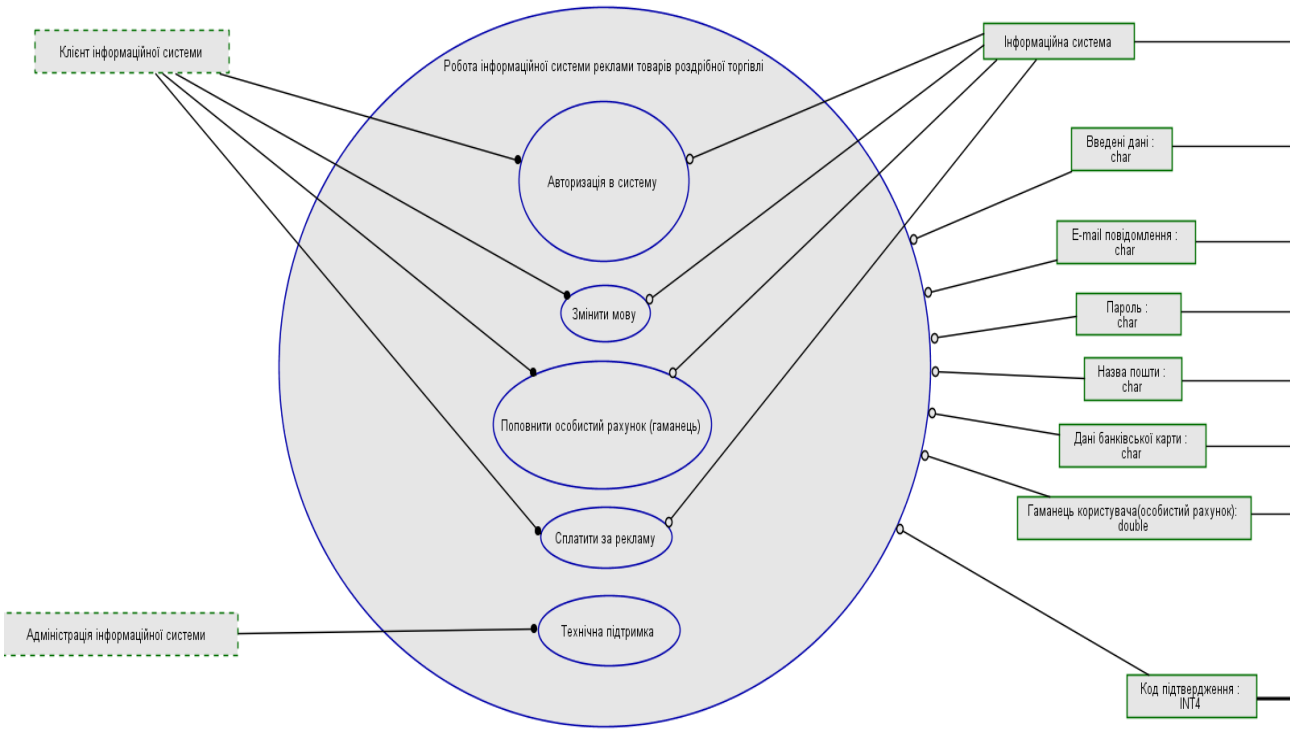


Рисунок 32 – Діаграмі верхнього рівня
Джерело: розроблено автором

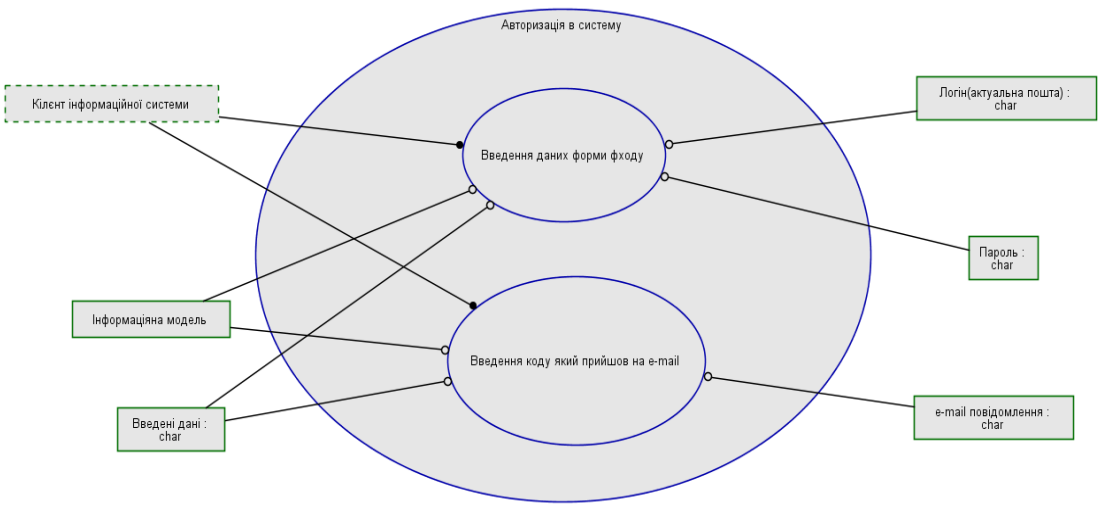


Рисунок 32.1 – Діаграмі нижчого рівня

Джерело: розроблено автором

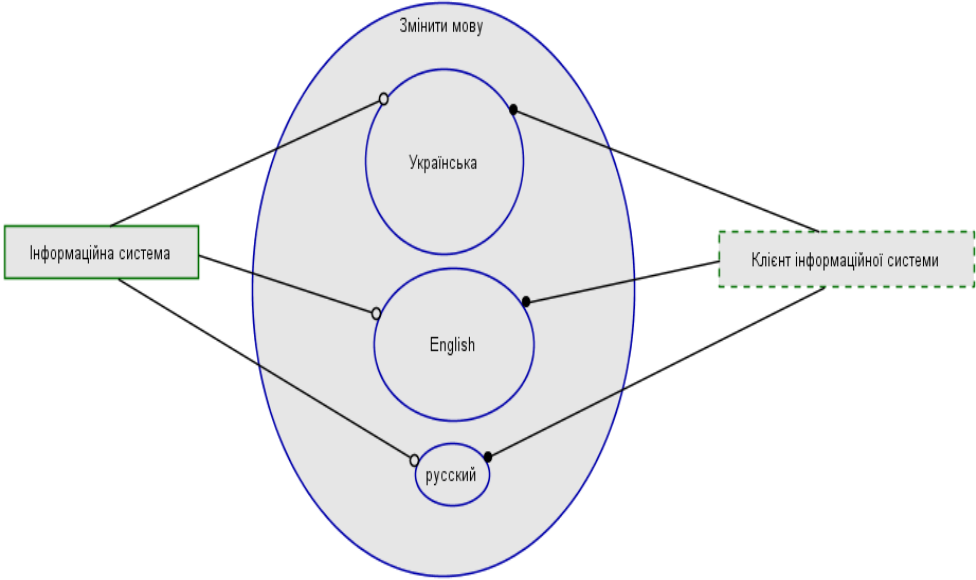


Рисунок 32.2 – Діаграми нижнього рівня

Джерело: розроблено автором

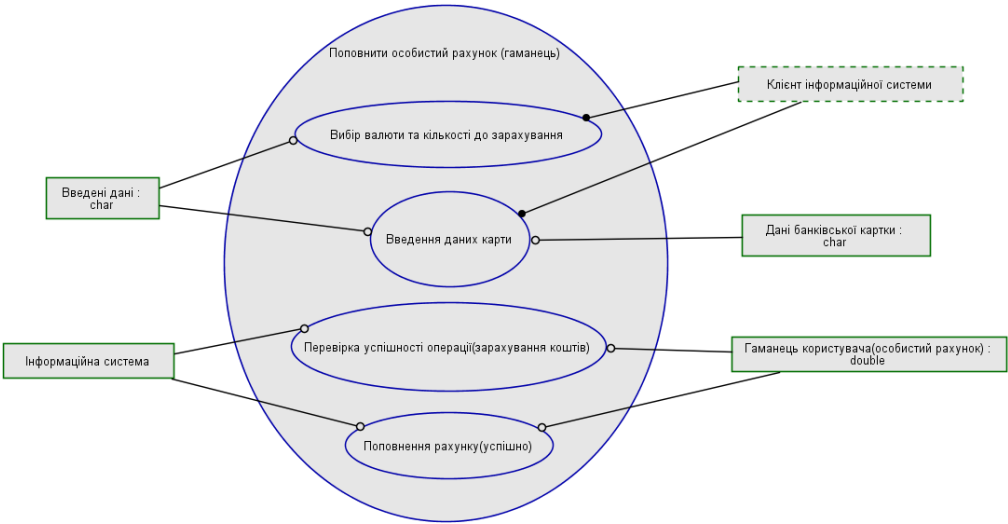


Рисунок 32.3 – Діаграми нижнього рівня

Джерело: розроблено автором

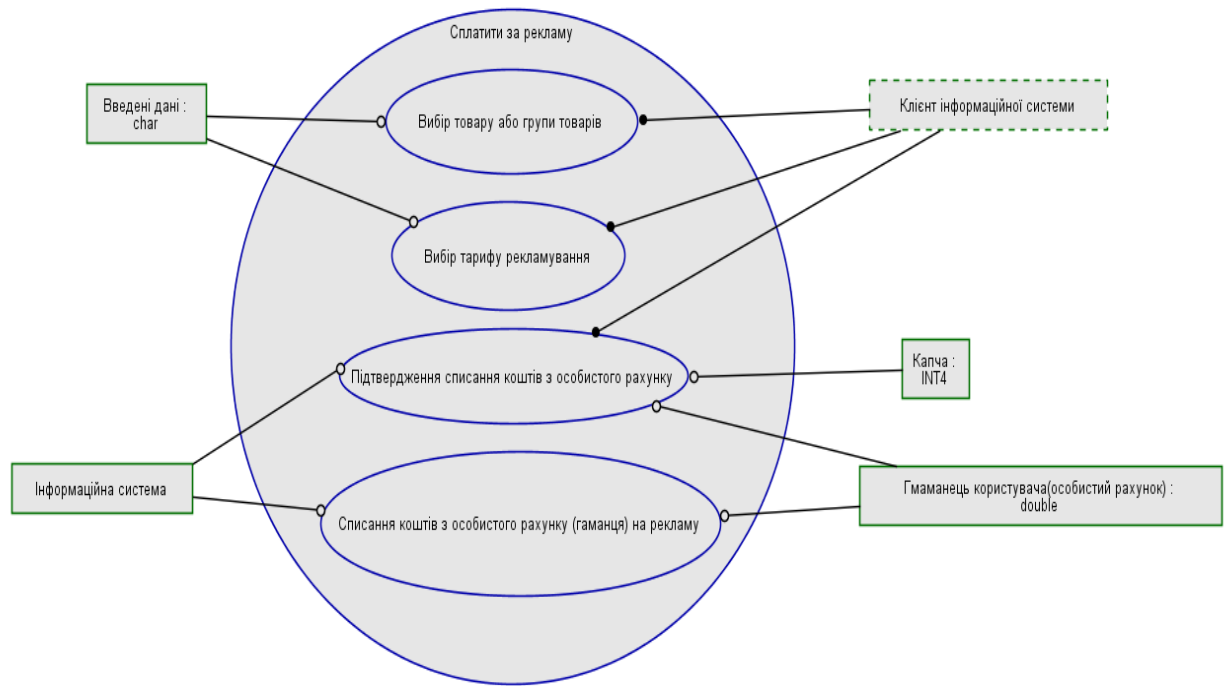


Рисунок 32.4– Діаграмі нижнього рівня

Джерело: розроблено автором

Додаток С

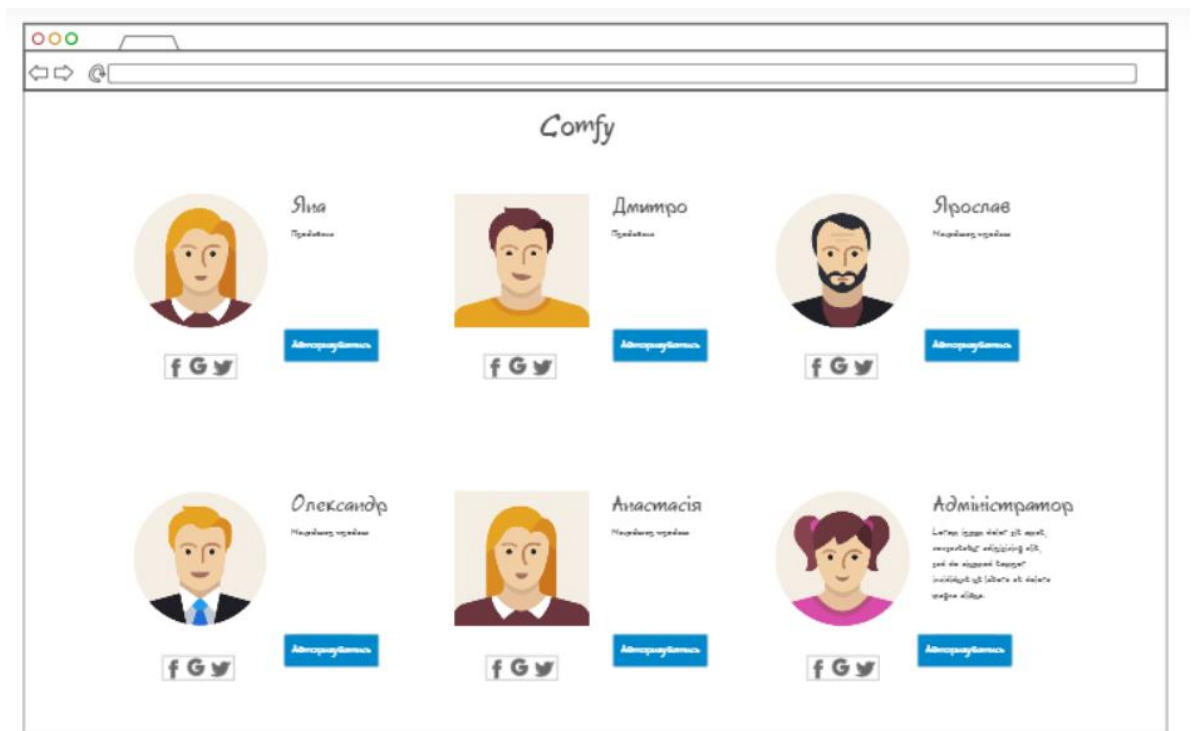
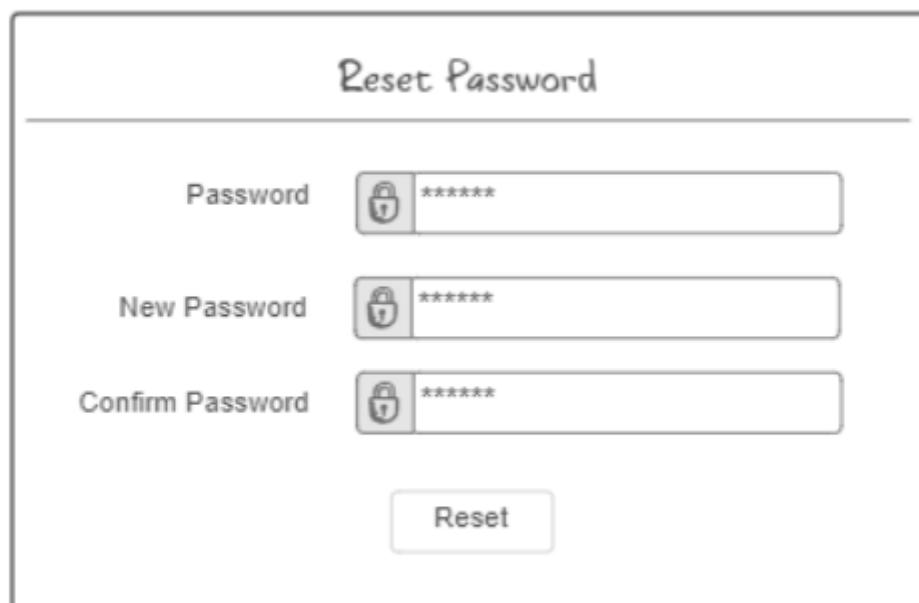


Рисунок 34 – прототип дизайну ІС
Джерело: розроблено автором



Reset Password

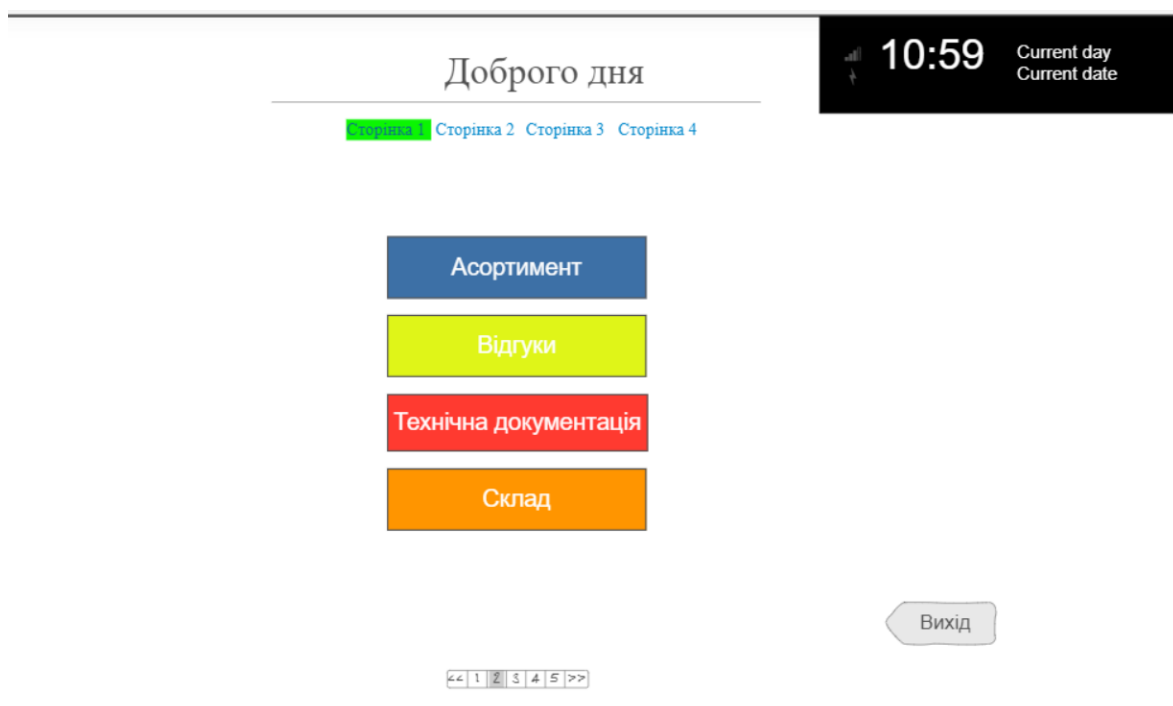
Password

New Password

Confirm Password

Reset

Рисунок 34.1 – прототип дизайну ІС
Джерело: розроблено автором



Доброго дня

10:59 Current day
Current date

Сторінка 1 Сторінка 2 Сторінка 3 Сторінка 4

Асортимент

Відгуки

Технічна документація

Склад

Вихід

<< 1 2 3 4 5 >>

Рисунок 34.2 – прототип дизайну ІС
Джерело: розроблено автором

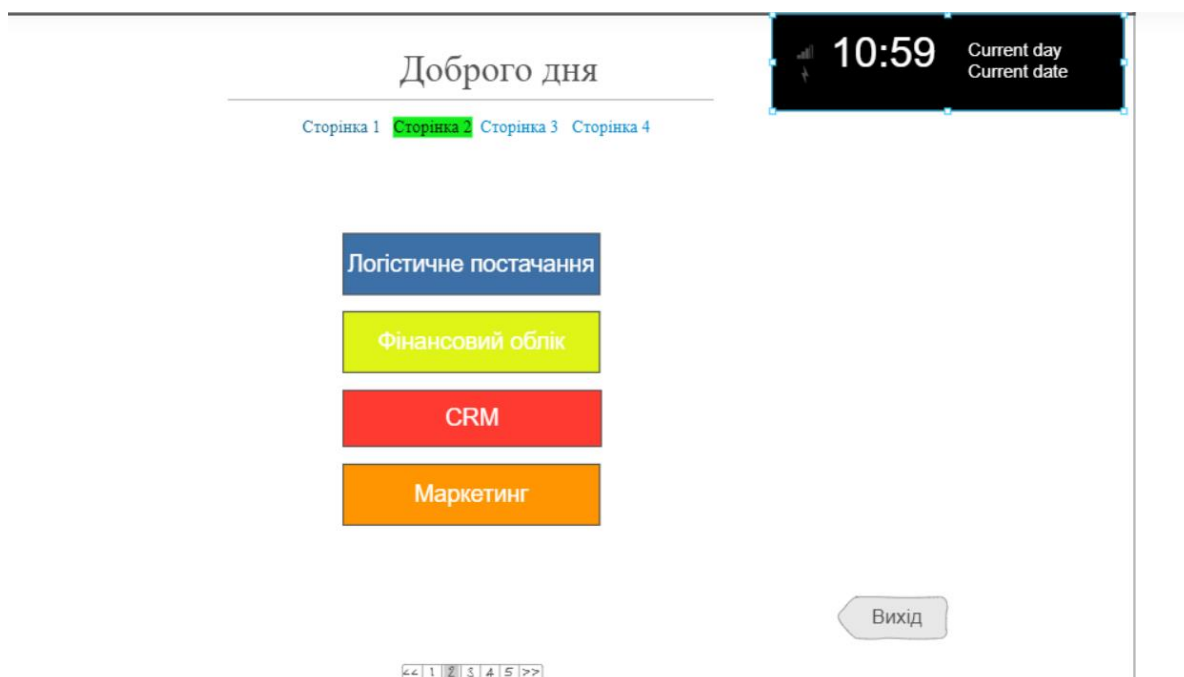


Рисунок 34.3 – прототип дизайну ІС
Джерело: розроблено автором

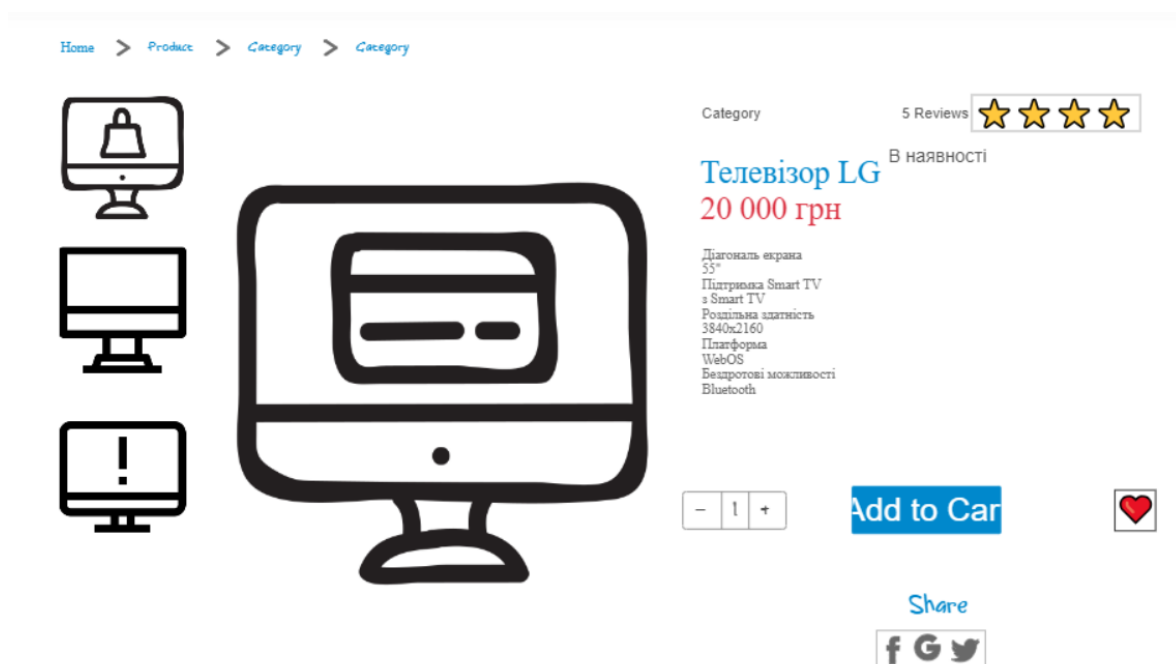


Рисунок 34.4 – прототип дизайну ІС
Джерело: розроблено автором

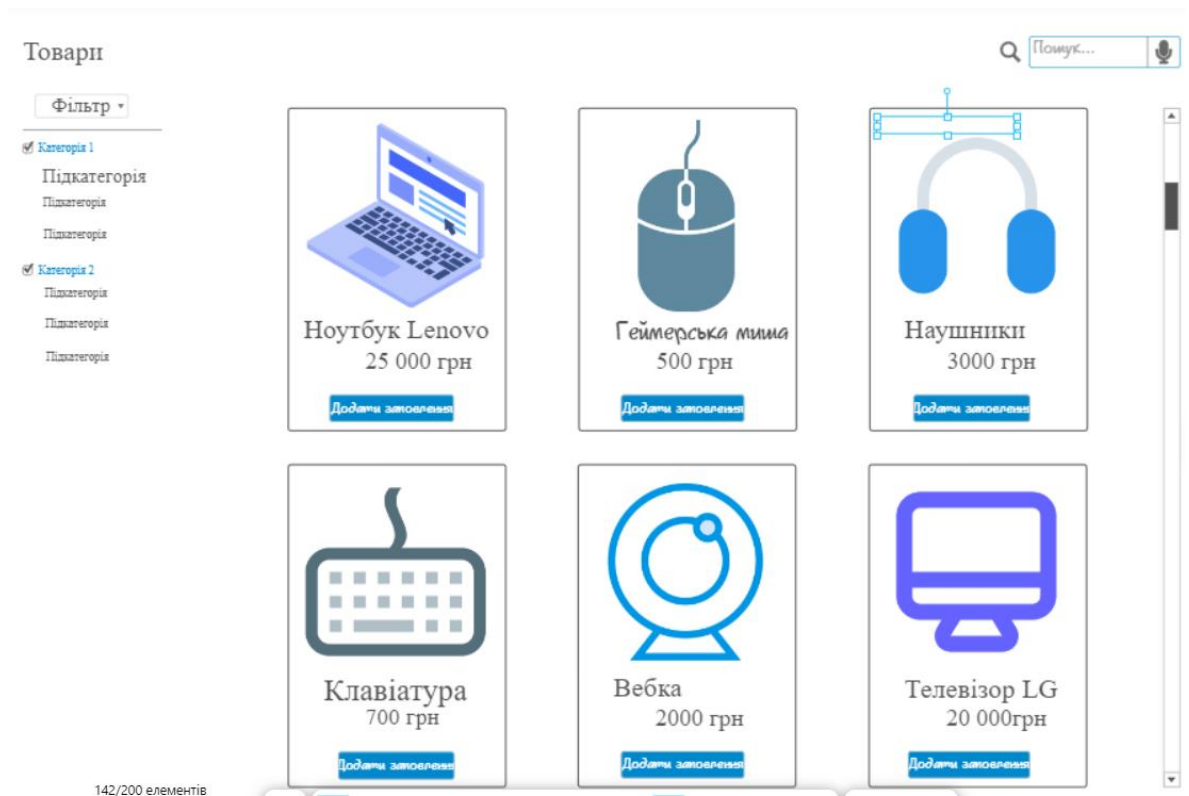


Рисунок 34.5 – прототип дизайну ІС

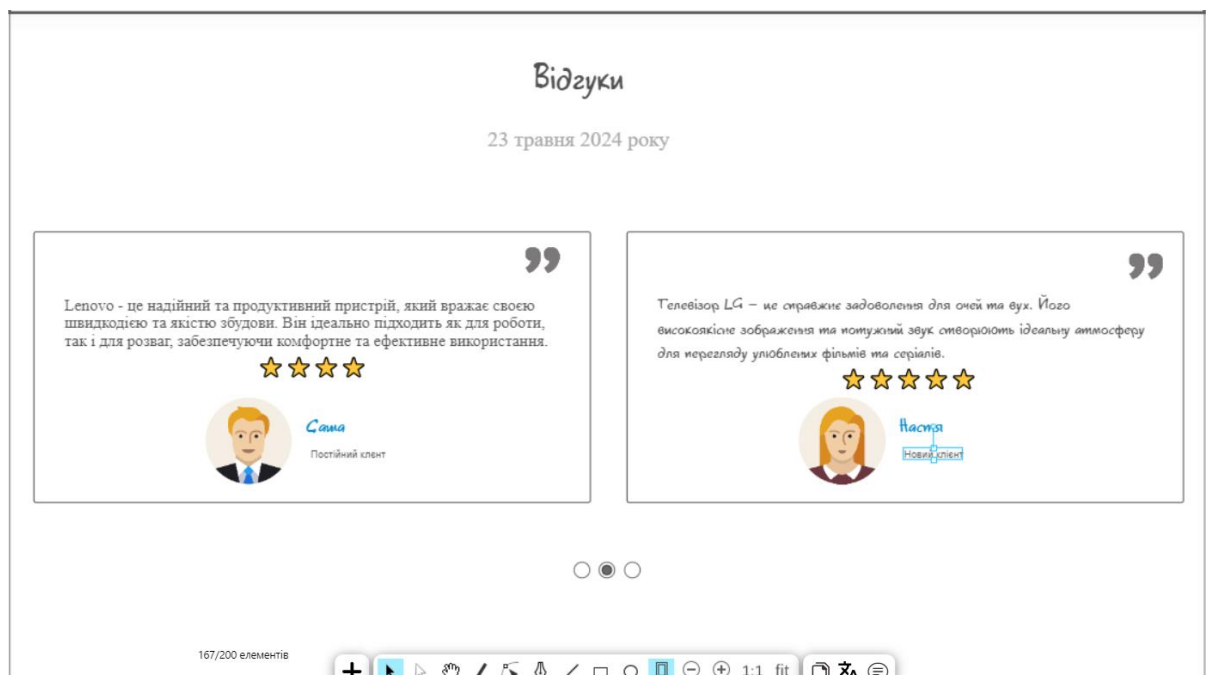
Джерело: розроблено автором

Рисунок 34.6 – прототип дизайну ІС

Джерело: розроблено автором

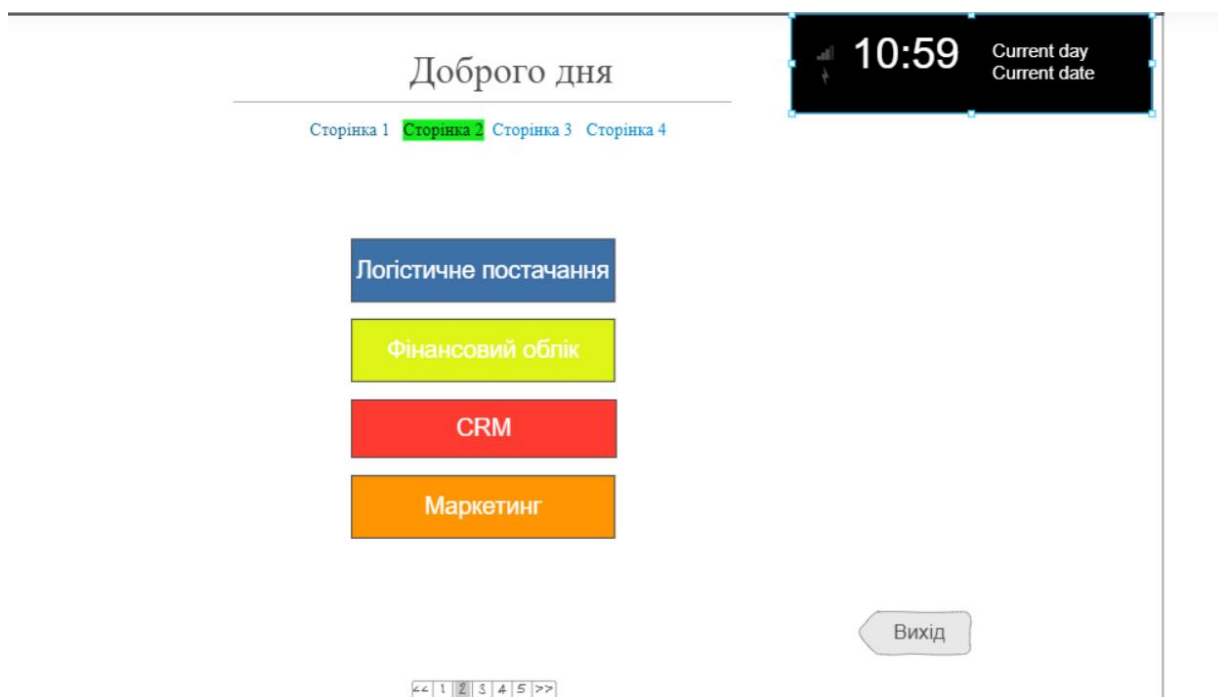


Рисунок 34.6 – прототип дизайну ІС
Джерело: розроблено автором

Ім'я користувача: Інформаційних систем в економіці Шкуратовська Те...	ID перевірки: 1016282759
Дата перевірки: 25.05.2024 21:49:46 EEST	Тип перевірки: Doc vs Internet + Library
Дата звіту: 25.05.2024 21:56:11 EEST	ID користувача: 100005745

Назва документа: Суховієнко О.ІУС-601з

Кількість сторінок: 92 Кількість слів: 15334 Кількість символів: 126338 Розмір файлу: 3.86 MB ID файлу: 10160759!

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

20.8%
СхожістьНайбільша схожість: 6.6% з Інтернет-джерелом (<https://javarush.com/ua/quests/lectures/ua.questhibernate.level17.lect>).

15.4% Джерела з Інтернету 257 Сторінка 94

13.7% Джерела з Бібліотеки 485 Сторінка 97

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 18
сторінок

РОЗДІЛ 1 «ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ПРЕДМЕТНОЇ ОБЛАСТІ»

1.1 Аналіз існуючих інформаційних систем

1.1.1 Дослідження предметної області

Для початку, розглянемо, що з себе представляє інформаційна система. Інформаційна система (ІС) — сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів. Інформаційні системи включають в себе: технічні засоби обробки даних, програмне забезпечення і відповідний персонал. Чотири складові частини утворюють внутрішню інформаційну основу: засоби фіксації і збору інформації; засоби передачі відповідних даних та повідомлень; засоби збереження інформації; засоби аналізу, обробки і представлення інформації. Основна мета ІС - підтримка діяльності організації шляхом ефективного управління інформаційними потоками, використовується для автоматизації бізнес-процесів, забезпечення інформаційної підтримки прийняття рішень, зберігання та обробки даних.

Також існує інший, схожий тип, - це інформаційна управляюча система (ІУС) , вона схожа на попередню, але має свої відмінності і переваги, є різновидом інформаційної системи, яка не тільки збирає та обробляє інформацію, але й активно бере участь у процесі управління організацією. ІУС надає інструменти для планування, контролю та аналізу діяльності підприємства. В першу чергу - це комплексне програмне забезпечення та апаратні засоби, призначені для збору, обробки, аналізу, зберігання та використання інформації з метою управління діяльністю організації або підприємства.

Основні відмінності між ними в тому, що ІС зосереджується на зборі та обробці інформації, може мати обмежену або відсутню управлінську функціональність, в той час як ІУС також зосереджується на управлінні та прийнятті рішень, але включає функції планування, контролю та аналізу, що є ключовими для управління організацією, та зазвичай зазвичай є більш складною системою, оскільки вона інтегрує більше функцій та інструментів для підтримки управління.

Торгові компанії займаються продажем товарів різних категорій, таких як електроніка, побутова техніка, одяг, взуття тощо. Її діяльність включає в себе такі основні процеси: закупівля товарів у постачальників, зберігання товарів на складі, продаж товарів клієнтам через офлайн та онлайн канали, обслуговування клієнтів (підтримка, гарантійне обслуговування тощо).

Інформаційна управляюча система, в даному разі, торгової компанії, призначена для автоматизації та оптимізації ключових бізнес-процесів. Основні функції ІУС можуть включати управління товарними запасами та залишками на складі, допомога в закупівлі товару на відстеження руку постачальників. Також вона може допомагати працівникам в управлінні продажами, а саме: обробка замовлень, управління торговими точками, інтеграція з онлайн платформами. Дуже велику користь може також дати система CRM, яка не рідко використовується у напрямках взаємодії з клієнтом, щоб мати певну інформацію про покупців, та зберігання їх даних, для подальшого покращення сервісу, та швидкості обробки їх запитів. Невід'ємною частиною систем у бізнесі, є ведення фінансового обліку, аналіз та прогнозування продажів. Ще один не менш важливий момент, це можливість ітерації з платіжними системами, адже світ йде до того, що звичайні гроші поступово переходять на задній план, а зараз досить популяризовано використовуються різні платіжні системи, для підвищення безпеки та зручності використання даного методу проведення фінансових операцій.

З інтенсивністю розвитку технологій, ведення будь-якого бізнесу, не являється можливим, без інформаційних систем, і у користувачів вже