

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Аліна САВЧЕНКО
«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

Тема: «Система автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії мовою Python»

Виконавець: Ростислав ПРИПУТНЕНКО

Керівник: старший викладач Наталка РИБАСОВА

Нормоконтролер: к. т. н., доцент Вікторія СИДОРЕНКО

КИЇВ 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра комп'ютерних інформаційних технологій

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ:

завідувач кафедри КІТ

_____ Аліна САВЧЕНКО

(підпис)

«_____» _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Припутненка Ростислава Володимировича

(ПІБ випускника)

1. Тема роботи: «Система автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії мовою Python» затверджена наказом ректора № 517/ст від 05.04.2024 р.
2. Термін виконання роботи: з 6 травня 2024 року по 16 червня 2024 року.
3. Вихідні дані до роботи: розробка додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії мовою Python.
4. Зміст пояснювальної записки: 1. Огляд та особливості предметної області. 2. Аналіз методів, технологій та інструментів розробки додатку з автоматизації обліку та фільтрації даних клієнтів. 3. Опис процесу створення системи мовою програмування Python.
5. Перелік обов'язкового ілюстративного матеріалу: 1. Контекстна діаграма декомпозиції першого рівня. 2. Архітектура web-додатку. 3. Структура створення облікового запису на проєктованому сервісі. 4. Діаграма класів.

6. Календарний план-графік

№з / п	Завдання	Термін виконання	Підпис керівника
1.	Огляд та аналіз предметної області. Написання 1 розділу, представлення керівнику.	06.05.2024- 13.05.2024	
2.	Вибір та опис використаних технологій. Написання 2 розділу, представлення керівнику.	14.05.2024- 17.05.2024	
3.	Загальна характеристика додатків з автоматизації для закладів харчової індустрії.	18.05.2024- 21.05.2024	
4.	Обґрунтування засобів реалізації проекту Написання 3 розділу, представлення керівнику. Практична реалізація проекту	22.05.2024- 27.05.2024	
5.	Загальне редагування та друк пояснювальної записки.	28.05.2024- 30.05.2024	
6.	Проходження нормоконтролю, перепліт пояснювальної записки.	31.05.2024- 04.06.2024	
7.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	05.06.2024- 07.06.2024	

7. Дата видачі завдання «06» травня 2024 р.

Керівник кваліфікаційної роботи _____ Наталка РИБАСОВА
(підпис керівника)

Завдання прийняв до виконання _____ Ростислав ПРИПУТНЕНКО
(підпис здобувача вищої освіти)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Система автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії мовою Python» містить: 79 сторінок, 32 рисунка, 15 інформаційних джерел, 4 додатки.

Об'єктом дослідження є додаток з автоматизації обліку клієнтів.

Предметом дослідження є сукупність необхідних умов, що забезпечують найкращий підхід до створення додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії мовою Python.

Мета дослідження. Основною метою додатку з обліку та фільтрації даних клієнтів для галузі громадського харчування є підвищення якості обслуговування клієнтів та оптимізація роботи самого закладу.

Наукова новизна даної роботи полягає в розробці оптимізованого програмного рішення для системи автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії мовою Python.

Результати кваліфікаційної роботи рекомендується використовувати для покращення якості обслуговування відвідувачів в закладах харчування. Створений додаток з обліку та фільтрації даних клієнтів у закладі харчової індустрії дозволить значно прискорити швидкість обслуговування, дозволить краще задовольняти запити клієнтів за рахунок створення страв спеціально під кожного відвідувача (додаток враховує попередні замовлення відвідувача і інформує офіціанта про його смаки). Для створення проєкту використано мову програмування Python. Проведено тестування додатку та підтверджено його ефективність в швидкості обслуговування клієнтів в закладі харчування.

Ключові слова: ХАРЧУВАННЯ, АВТОМАТИЗАЦІЯ ОБСЛУГОВУВАННЯ, ДАННІ КЛІЄНТІВ, АРХІТЕКТУРА ДОДАТКА, ПРОГРАМНИЙ КОД.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМ АВТОМАТИЗАЦІЇ ДЛЯ ЗАКЛАДІВ ІНДУСТРІЇ ХАРЧУВАННЯ	9
1.1. Оцінка перспективності автоматизації в закладах індустрії харчування	9
1.2. Основне призначення баз даних в системах автоматизації.....	12
1.3. Аналіз існуючих на ринку аналогів автоматизації обліку та фільтрації даних клієнтів.....	14
ВИСНОВКИ ДО РОЗДІЛУ 1	20
РОЗДІЛ 2 ОБҐРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЄКТУ	16
2.1. Вибір архітектури програмного забезпечення.....	16
2.2. Характеристика мови програмування	22
2.3. ER-діаграма та діаграма варіантів використання додатку	24
2.4. Аналіз основних алгоритмів роботи додатку	28
ВИСНОВКИ ДО РОЗДІЛУ 2	36
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ	37
3.1. Математична модель автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії	37
3.2. Функціональні та нефункціональні вимоги	48
3.3. Вибір архітектури серверного додатку та технологій для обладнання.....	50
3.4. Опис архітектури програмного забезпечення.....	53
ВИСНОВКИ ДО РОЗДІЛУ 3	67
ВИСНОВКИ	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69
ДОДАТКИ	71
Додаток А	71
Додаток Б.....	72
Додаток В.....	73
Додаток Д	74

ВСТУП

Наразі управління закладами харчової сфери стає все більш значущим і активно розвивається щодня. ІТ-технології є неодмінною складовою цього процесу, безпосередньо впливаючи на його прогрес і успіх. Виконання транзакцій, контроль та навчання персоналу, передбачення ризиків, аналіз роботи, розробка меню та виробництво власних продуктів - усе це вже потребує автоматизації, оскільки в цій галузі надзвичайно важлива звітність. З точки зору клієнтів, автоматизація обліку та фільтрації даних клієнтів, розрахунків, доступу до меню, акцій, написання скарг та пропозицій - неймовірно зручна і вигідна, оскільки значно економить час. [14].

Актуальність дослідження. Ресторан – це один із галузей закладів харчової індустрії, в якому люди не тільки смакують різними стравами, але й відпочивають, отримують приємні емоції, естетичне задоволення та спілкуються один з одним. Використання інформаційних технологій для обліку та фільтрації даних клієнтів є однією з причин, що мають вплив на успішність функціонування бізнесу, особливо на управління закладом громадського харчування.

Автоматизація обліку та фільтрації даних клієнтів надає міцний фундамент для досконалого, якісного та швидкого сервісу, що неодмінно призводить до успішності роботи будь-якого закладу харчової індустрії.

Основна задача закладу харчової індустрії полягає у швидкому та якісному виконанні замовлень відвідувачів. Це суттєво відрізняє управління закладами харчування від інших бізнес-сфер. Автоматизація може значно допомогти у вирішенні цієї задачі. Вона дозволяє контролювати всі операції протягом всього робочого процесу з високою точністю, скорочує час обслуговування клієнтів і підвищує прибутковість закладу. Автоматизація призводить до покращення якості обслуговування та репутації закладу, забезпечує швидку обробку замовлень та обслуговування відвідувачів, підсилює координацію і контроль роботи персоналу, а також допомагає у відстеженні змін матеріальних цінностей та товарів, і зменшує час на пошук та аналіз інформації. Це дослідження

присвячене розробці додатку для автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії з використанням мови програмування Python.

Отже, додаток з обслуговування в закладі харчової індустрії є незамінним інструментом у роботі будь-якого закладу харчування.

Мета дослідження. Основною метою додатку з обліку та фільтрації даних клієнтів для галузі громадського харчування є підвищення якості обслуговування клієнтів та оптимізація роботи самого закладу.

Для досягнення поставленої мети дослідження, необхідно вирішити **наступні завдання:**

- провести літературний пошук за темою дослідження, систематизувати інформацію та визначити напрямки подальшого дослідження;
- зробити обґрунтування засобів реалізації проекту із створення додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії мовою Python»;
- згідно дерева цілей, контекстних діаграм та схематичного зображення архітектури провести теоретичне формування функціоналу додатку;
- виконати програмну частину роботи та провести її тестування.

Об'єктом дослідження є додаток з автоматизації обліку клієнтів.

Предметом дослідження є сукупність необхідних умов, що забезпечують найкращий підхід до створення додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії.

Методи дослідження: теоретичний аналіз наукової літератури за напрямком дослідження; статистичні методи аналізу літературних даних. В основу дослідження, лягли методи порівняльного аналізу та класифікації.

Поставлені у роботі завдання вирішувалися за допомогою системного підходу до відбору матеріалу, методів індуктивного і логічного аналізу, спостереження та статистичних методів аналізу літературних даних.

Теоретична та практична цінність роботи полягає у наявності відібраного теоретичного матеріалу, отриманого в процесі пошуку інформації з теми

дослідження, а також у систематизації матеріалу відповідного напрямку. Проведене дослідження характеризується більш глибоким аналізом, базуючись на попередніх роботах вчених, дисертантів та інших дослідників у цій сфері.

Структура роботи. Відповідно до мети та завдань дослідження структура дипломної роботи включає вступ, три розділи, загальні висновки та список використаних джерел. За час роботи опрацьовано 15 літературних джерел. Зміст роботи викладено на 79 сторінках машинописного тексту.

Джерельна база дослідження. Робота ґрунтується на аналізі науково-методичної літератури, методичних посібників, наукових статей, періодичних видань.

РОЗДІЛ 1
ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМ АВТОМАТИЗАЦІЇ ДЛЯ
ЗАКЛАДІВ ІНДУСТРІЇ ХАРЧУВАННЯ

1.1. Оцінка перспективності автоматизації в закладах індустрії харчування

Перед вибором програмного забезпечення для автоматизації в закладі харчової індустрії необхідно визначити критерії, які будуть вирішальними. Один з ключових аспектів для невеликих закладів, особливо в сучасних умовах, - це вартість програмного забезпечення. На сьогоднішньому ринку існує безліч рішень для автоматизації громадського харчування, і вибір залежить від потреб конкретного закладу.

Основний функціонал автоматизації включає в себе весь цикл роботи від приймання інгредієнтів до реалізації страв. Цей процес є ключовим для будь-якого закладу харчової індустрії, тому важливо мати додаток, який відповідає цим потребам. Також важлива аналітика для керівництва та можливість впроваджувати додаток лояльності для клієнтів. Надійність обладнання є ще одним важливим критерієм, особливо для невеликих закладів, де кожна технічна проблема може призвести до втрати доходів. Також варто звернути увагу на масштабоване програмне забезпечення, яке може починати з основних функцій і поступово розширюватися за потребами бізнесу.

На фоні складного часу для закладів харчової індустрії важливою є і мотивація персоналу. Навіть найкращі ідеї по управлінню та залученню клієнтів не допоможуть, якщо персонал не відчуває зацікавленості в успіху бізнесу. Тому адміністраторам закладу індустрії харчування важливо підтримувати мотивацію та професіоналізм свого персоналу.

Сучасні додатки можуть допомагати закладам харчової індустрії у

Кафедра КІТ				НАУ 24 41 19 000 ПЗ			
	<i>ПІБ</i>			ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМ АВТОМАТИЗАЦІЇ ДЛЯ ЗАКЛАДІВ ІНДУСТРІЇ ХАРЧУВАННЯ	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Виконав</i>	Припутненко Р. В.					9	80
<i>Керівник</i>	Рибасова Н. О.				ТП-415Б - 122		
<i>Н-Контролер</i>	Сидоренко В.М.						

багатьох аспектах, включаючи гостинність у залі та безперебійну роботу на кухні. Вони автоматизують касові операції, управління складом, мотивацію персоналу, фінансове планування, програми лояльності та навіть відтворення музики в закладі харчової індустрії. Дослідження показують, що завдяки оптимізації бізнес-процесів та зменшенню витрат можна заощадити до 25% операційних витрат [2].

Впровадження додатку швидко окупається через підвищення ефективності бізнесу, зниження витрат від зловживань, оптимізацію управління запасами та закупівельними цінами, а також через зростання прибутку завдяки збільшенню кількості клієнтів. На ринку існують різні додатки з автоматизації, від безкоштовних open source до повнофункціональних, і важливо обирати рішення, яке відповідає потребам конкретного закладу.

Наприклад, для невеликих закладів харчової індустрії важлива гнучкість конфігурацій і можливість працювати на декількох комп'ютерах під однією ліцензією. Також потрібно враховувати якість обладнання та умови його гарантійного обслуговування, особливо для малих закладів без штатного системного адміністратора [6].

Вибір програмного забезпечення також важливий, оскільки кожна додаткова функція збільшує вартість і може вимагати додаткової технічної підтримки. Простота програмного рішення допомагає уникнути частих звернень до служби технічної підтримки. Навчання персоналу щодо роботи з додатком також грає важливу роль у мінімізації проблем. Автоматичні додатки з управління бізнесом в закладі харчової індустрії є ключовими для забезпечення ефективності та прибутковості закладу. Вони дозволяють контролювати грошові потоки, виробничі процеси, облік продуктів харчування, замовлення та потік клієнтів. Це сприяє оптимізації робочих процесів та мінімізації витрат.

Незважаючи на це, важливо розуміти, що автоматизація сама по собі не гарантує миттєвого прибутку. Це лише інструмент, який в руках досвідченого керівника може призвести до позитивних результатів. Використання

автоматизації спрямоване на зменшення ручної праці, оптимізацію витрат та підвищення продуктивності.

Автоматизація також дозволяє контролювати персонал, що є важливим аспектом у галузі громадського харчування. Це допомагає уникнути крадіжок та інших зловживань, що може призвести до збільшення прибутковості закладу. В умовах економічної нестабільності, впровадження автоматизації стає особливо актуальним для досягнення ефективного управління та економії.

Правильно, автоматизація може здатися витратною на перший погляд, але вона є необхідною інвестицією для закладу харчової індустрії, особливо при відкритті нового закладу. Її вплив на робочий стиль і психологію персоналу значно, і ці зміни варто врахувати заздалегідь.

Автоматизація це вирішення проблем і оптимізація роботи компанії. Для того щоб оцінити витрати на автоматизацію, треба задати собі запитання: скільки коштують помилки при управлінні складом і інвентаризації? Скільки вартує недобросовісний персонал або незадоволені клієнти через помилки в обслуговуванні?

В умовах сучасного господарювання, кожна копійка на рахунку важлива. Автоматизація допомагає боротися з крадіжками, ефективно контролює витрати та підвищує якість обслуговування. Важливо пам'ятати, що це не просто витрати, а ефективна стратегія для зниження витрат і збільшення продуктивності. Завдяки автоматизації ви зможете впроваджувати програми лояльності та знижки для клієнтів, обліковувати бонуси, та спрощувати розрахунки для персоналу. Такий додаток не тільки оптимізує роботу, а й створює можливості для розвитку та залучення нових клієнтів [5].

В середньому термін окупності додатку з автоматизації для невеликого закладу харчової індустрії складає 3–4 місяці за умови застосування основного набору модулів і компонентів. Автоматизація закладу харчової індустрії має безліч переваг, серед яких:

- Повний контроль над усіма процесами у закладі.
- Зменшення часу очікування замовлення гостями.

- Підвищення швидкості приготування та подачі страв.
- Можливість офіціантам контролювати замовлення на своїх столах.
- Мінімізація помилкових дій персоналу.
- Зростання доходів компанії.
- Зменшення крадіжок та зловживань ресурсами.
- Покращення якості та швидкості обслуговування відвідувачів.
- Автоматичне формування фінансових звітів та збір маркетингової інформації.
- Ефективне управління та аналіз діяльності підприємства.
- Оперативний облік товарів та проведення інвентаризації.
- Контроль за роботою персоналу та закладу, в тому числі й дистанційно.

Однак існують деякі недоліки автоматизації:

- Несвоєчасне внесення даних, неточності та помилки можуть спотворити інформацію.
- Деякі програми потребують постійного підключення до Інтернету і можуть перестати працювати при проблемах з мережею.
- Складні в управлінні програми з великою кількістю непотрібних функцій можуть ускладнити роботу [9].
- При неполадках додатку потрібний фахівець для налагодження.

Незважаючи на ці недоліки, автоматизація є невід'ємною частиною підвищення якості обслуговування та ефективності бізнесу закладів харчової індустрії, а також зниження витрат та недоречного використання ресурсів.

1.2. Основне призначення баз даних в системах автоматизації

База даних є надзвичайно важливим програмним засобом у сучасному світі, оскільки без комп'ютера її створення практично неможливе. Реальні приклади баз даних включають телефонні каталоги, реєстри записів на прийоми до лікарів, архіви бібліотек тощо, хоча вони часто не дозволяють змінювати інформацію [7].

База даних – це великий та організований набір інформації, який зберігається разом з програмним забезпеченням на комп'ютері чи його носіях. Це дозволяє швидко знаходити потрібну інформацію, оновлювати та друкувати її. Сьогодні майже кожне підприємство чи державна установа має свою власну базу даних. Протягом останніх десятиліть системи обробки даних та управління інформацією розвивалися швидко, від простих методів середини ХХ століття до складних систем з можливістю інтеграції. Починаючи з письмових методів, сучасні додатки можуть накопичувати, керувати та аналізувати дані. Для цього існують призначені програми – системи управління базами даних (СУБД), які дозволяють створювати, оновлювати, шукати та аналізувати дані в БД.

Бази даних використовують різні моделі, такі як ієрархічна, мережева, реляційна та об'єктно-орієнтовна. Кожна з них має свої переваги і недоліки, але загалом бази даних разом із СУБД є невід'ємними складовими сучасних додатків та сприяють ефективному управлінню та обробці великих обсягів інформації.

Модель даних включає опис основних характеристик набору даних, який повинен бути підтриманий відповідною системою управління базами даних (СУБД), що базується на цій моделі. Відповідно, тип СУБД визначається видом моделі даних. Згідно з методом доступу до даних, визначають 2 види СУБД, а саме на персональні (такі як Access, dBase, Clipper) та мережеві (такі як MySQL, Oracle, Informix). Мережеві СУБД підтримують багатокористувацьке середовище.

Персональні СУБД – це набір програм та мовних засобів, що відповідають за створення та правильне функціонування баз даних на рівні окремих користувачів. Серверна складова та клієнтська складова є основними складовими мережевих СУБД, які можуть працювати на різних операційних системах. Стрімко набирає популярності новий спосіб доступу до даних, де основним аспектом є відсутність спеціалізованого клієнтського програмного забезпечення. Для роботи з віддаленими базами даних використовується стандартний веб-браузер. Код, вбудований у HTML-сторінки, що може бути написаний на різних мовах програмування, реєструє всі операції користувача та передає їх у вигляді SQL-запитів до бази даних. Цей метод ефективно

використовується як для віддалених користувачів, так і для користувачів локальної мережі [1].

1.3. Аналіз існуючих на ринку аналогів автоматизації обліку та фільтрації даних клієнтів

Значна частка додатків з автоматизації обліку та фільтрації даних клієнтів зберігають фотокартки паперового меню та несучасний інтерфейс. Через ці проблеми користувач більш за все надасть перевагу іншому додатку. Окрім цього сам інтерфейс додатку з обліку та фільтрації даних клієнтів має бути інтуїтивно зрозумілим і достатньо інформативним задля ефективної співпраці між клієнтом і закладом. Додаток з автоматизації обліку та фільтрації даних клієнтів повинен бути інтуїтивно доступним для клієнтів будь-якої соціальної категорії. Навігація має бути зрозумілою, щоб кожен користувач зміг ознайомитися з можливостями система та скористатись ними без усіяких проблем. Тож розглянемо приклади деяких додатків, що стосуються закладу харчової індустрії [8]:

1. «Vivaolive» – заклад індустрії харчування у м. Полтава представлений на рис. 1.1. Недоліки представленого додатку з автоматизації обліку та фільтрації даних клієнтів: – відсутня можливість здійснення замовлення з додатку; – немає представлених зображень страв, що пропонуються закладом. перефразувати українською мовою.



Рис. 1.1. Вигляд додатку з обліку клієнтів у закладі індустрії харчування

2. «Пеперончіно» популярний заклад харчування представлено на рис. 1.2.

Недоліки додатку з автоматизації обліку та фільтрації даних клієнтів:

- немає свого веб-сайту;
- можливість оформити замовлення тільки за допомогою інших послуг доставки;
- складна навігація;
- відсутність зображень більшості страв.

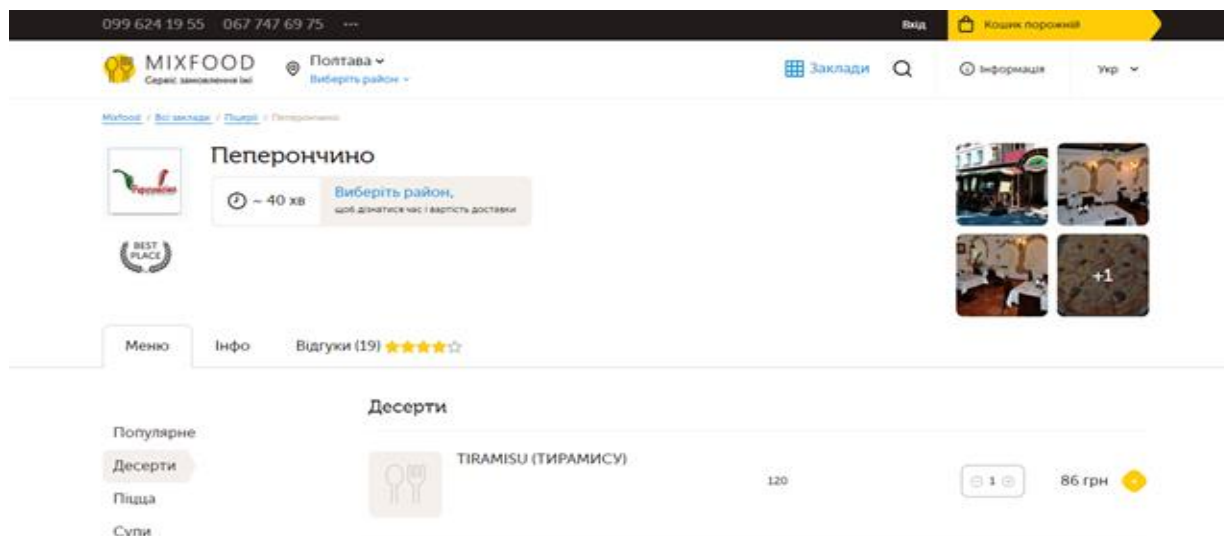


Рис. 1.2. Вигляд додатку з обліку клієнтів у закладі харчування

3. «Montanarizza» піцерія. Додаток представлено на рис. 1.3. Недоліки додатку

з автоматизації обліку та фільтрації даних клієнтів:

- немає свого веб-сайту;
- можливість оформити замовлення тільки за допомогою інших послуг доставки.

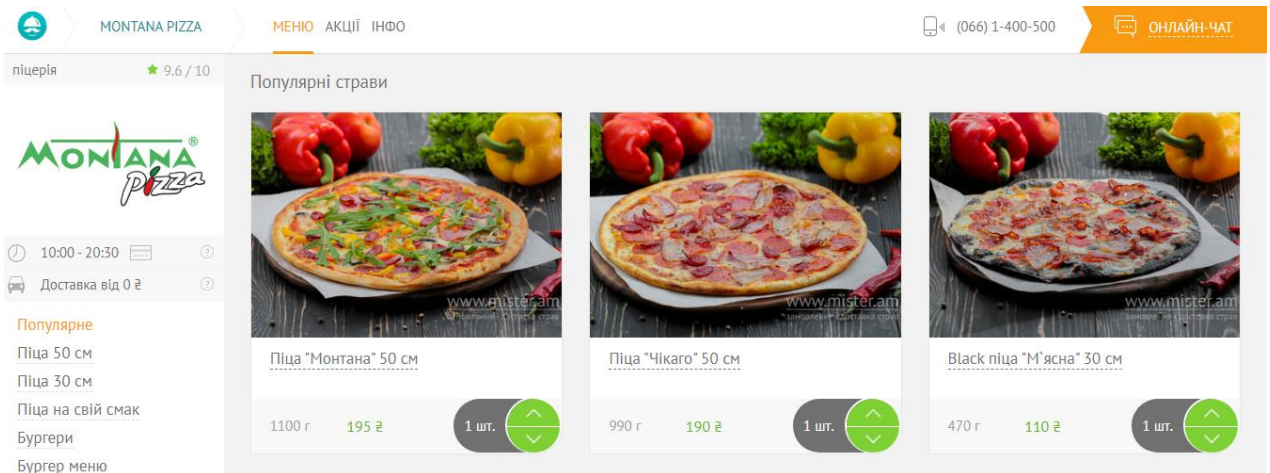


Рис. 1.3. Вигляд додатку з обліку клієнтів у закладі харчування

4. R-Keerer – Найстаріший гравець на ринку автоматизації закладу харчової індустрії. Першу ітерація системи R-Keerer була розроблена ще в 1992 році, в даний момент схожий додаток встановлено в 37 тисячах закладах харчової індустрії в різних країнах світу.

Незважаючи на те, що додаток містить різноманітні модулі, що включають фронт-офіс для безпосередньої роботи у закладі харчової індустрії, а також бек-офіс для регулювання процесів і витрат, деякі користувачі висловлюють незадоволення щодо його непростого та неінтуїтивного інтерфейсу.



Рис. 1.4. Вигляд додатку з обліку клієнтів у закладі харчування

Тип установки: Додаток встановлюється локально.

Вартість ліцензії: від 30000 до 70000 гривень.

кнопки, прив'язати відеозапис до кожного замовлення і багато іншого. [11]

Розробником є компанія Тензор, відома своїми системами для електронної звітності в державних установах і документообігу. У Presto реалізовано цей функціонал, що дозволяє відправляти звіти по алкоголю та підписувати бухгалтерські та кадрові документи без необхідності в додатковому програмному забезпеченні.



Рис. 1.6. Вигляд додатку з обліку клієнтів у закладі харчування Presto – а) та інтерфейс Jowi – б)

Підтримка працює цілодобово і надає допомогу через чат і віддалене підключення. Зібрана база знань містить докладні інструкції та кейси щодо роботи з додатком. Додаток доступний як хмарне рішення, а також існує офлайн-версія для випадків проблем з Інтернет-з'єднанням.

Вартість: від 300 до 760 гривень на місяць.

7. Jowi – це професійний гібридна програма для автоматизації, яка поєднує хмарний і локальний підходи – модуль Jowi встановлюється в закладі харчової індустрії, а потім дані синхронізуються на віддалені сервери (рис. 1.6б). Це дозволяє зберегти працездатність додатку навіть при «падінні» інтернету.

Система також має модульну структуру з додатками для менеджерів залу та офіціантів, які автоматично синхронізують дані між собою. Наприклад, офіціант приймає замовлення, кухар побачивши його, розпочинає готування, а менеджер отримує інформацію про час приготування та інші деталі. Цей додаток дуже гнучкий і дозволяє змінювати дані навіть у минулому часі - ця можливість важлива у реальному середовищі, де такі зміни виникають досить часто. Система

забезпечує зв'язок за допомогою різних способів, а відповіді надходять досить швидко.

Установлюється локально. Вартість: 6000 гривень в місяць.

8. Poster POS - це додаток з автоматизації, який використовує хмарне зберігання даних і працює як на техніці від Apple, так і на пристроях з операційною системою Android. Власне робоче місце касира або офіціанта реалізоване на планшеті, а чеки друкуються за допомогою термального принтера Epson TM-T20.

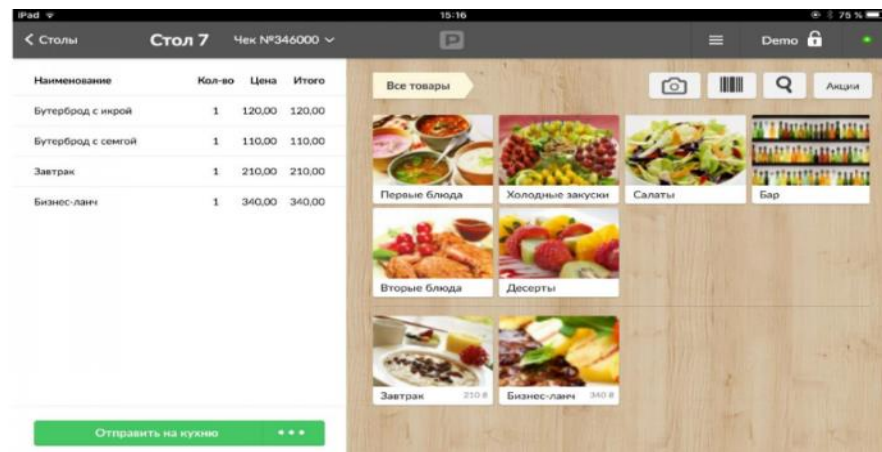


Рис. 1.7. Вигляд додатку з обліку клієнтів у закладі харчування Poster POS

Установлюється локально. Вартість: \$ 24- \$ 79 в місяць.

ВИСНОВКИ ДО РОЗДІЛУ 1

В даному розділі проведено оцінку перспективності автоматизації в закладах індустрії харчування, а також розглянуто основне призначення баз даних в цих системах. Проаналізовано існуючі на ринку аналоги автоматизації обліку та фільтрації даних клієнтів. На основі аналізу було виявлено, що автоматизація в закладах харчування є необхідним кроком для підвищення ефективності роботи, поліпшення обслуговування клієнтів та оптимізації внутрішніх процесів закладів. Використання програмного забезпечення дозволяє значно знизити витрати часу на обробку замовлень, підвищити точність виконання операцій і зменшити можливі помилки людського фактора.

РОЗДІЛ 2

ОБГРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЄКТУ

2.1. Вибір архітектури програмного забезпечення

В проєкті використано модель архітектури Клієнт-Сервер, яка є взаємовідносинами один до багатьох. Сервер надає / зберігає інформацію. У закладах індустрії харчування серверами є велика кількість різноманітних інтерфейсів для клієнтів та персоналу закладу індустрії харчування.

Користувачі – це клієнти, які повинні вводити інформацію в три окремі інтерфейси: доступ до системи винагород та проведення платежів, оформлення замовлення на їжу з меню, бронювання столиків.

Працівники – це клієнти, що застосовують сервер синхронізації для фіксації своїх робочих змін, за які їм нараховується оплата.

Менеджери – це клієнти, які використовують сервер для редагування інформації про робітників, планування робочих змін, внесення змін до меню та управління інформацією про столики.

Вибрана архітектура клієнт-сервер є широко розповсюдженою і слугує обчислювальною моделлю, де сервер відповідає за розміщення, розподіл і управління більшістю ресурсів та послуг, які використовуються клієнтами.

Такі системи складаються з одного або кількох клієнтських пристроїв, підключених до центральних або головних серверів через мережу, зазвичай через Інтернет. Усі подібні додатки обмінюються обчислювальними ресурсами. Архітектура клієнт-сервер, у деяких визначеннях, вважається структурою, яка виконує обчислення для мережі. Вона також відома через те, що запити та пов'язані з ними операції розподіляються по мережі. Далі наведено пояснення її роботи. [3].

Кафедра КІТ				НАУ 24 41 19 000 ПЗ			
	ПІБ			ОБГРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЄКТУ	Літ.	Аркуш	Аркушів
Виконав	Припутненко Р. В.					21	80
Керівник	Рибасова Н. О.				ТП-415Б - 122		
Н.Контролер	Сидоренко В.М.						

У системі клієнт-сервер, коли комп'ютер клієнта ініціює запит на передачу інформації до сервера через мережу, сервер отримує цей запит, обробляє потрібну інформацію та повертає запитовані дані клієнту.

Важливою характеристикою цієї моделі є здатність сервера одночасно керувати великою кількістю клієнтів. Крім того, один користувач може підключатись до декількох серверів за один раз, і кожен сервер надає клієнту різний набір послуг.

Іншим архітектурним підходом є модель "Peer-to-Peer" (P2P). Ця модель використовує розподілену архітектуру додатків, де завдання розподіляються між рівнозначними вузлами. У випадку оплати замовлення клієнтом у закладі харчування, модель є децентралізованою, тому кожен платіж обробляється окремо [4].

В додатку модель застосовується, коли клієнт здійснює оплату карткою, перетворюючи однорангову мережу на платіжний сервіс, який переказує кошти з грошового рахунку клієнта на рахунок закладу харчування.

2.2. Характеристика мови програмування

Python є поширеною мовою програмування, яка використовується в ІТ-сферах широкого спектру. Машинне навчання, розробка програмних додатків, написання скриптів автоматизації і багато іншого – мабуть, це один з найбільш універсальних інструментів у програмістів. Це мова високого рівня, яка домінує у багатьох сферах. За весь час свого існування мові програмування Python вдалося:

- зайняти лідерські позиції в сфері машинного навчання;
- щільно осісти на ринку десктопних додатків;
- охопити значну частину сегмента веб-програмування.

Python часто вважають першою мовою програмування через свою універсальність. Сучасні рейтинги та статистичні сервіси підтверджують цей статус. Наприклад, у рейтингах PYPL, TIOBE та на сайті [statista.com](https://www.statista.com) Python стабільно займає місце в трійці лідерів вже протягом кількох років поспіль.

Програмування стало більш динамічним завдяки таким особливостям Python:

- Динамічна типізація. Типи можуть бути статичними або динамічними. У випадку з динамічною типізацією розробнику не потрібно вказувати тип змінних – мова програмування визначає їх автоматично.

- Зручне повернення декількох значень функцією. Розробник може перерахувати значення через кому, і вони автоматично будуть перетворені в список [12].

- Автоматичне управління пам'яттю. Програмісту не потрібно вручну. Хоча це трохи знижує контроль над програмою, але значно прискорює процес розробки.

- Автоматичне збирання сміття. Ця функція оптимізує використання пам'яті та спрощує процес розробки.

- Динамічне прив'язування типів даних. У Python тип даних зв'язується зі значенням, а не зі змінною.

Мова програмування Python використовується для виконання наступних основних завдань:

1. для аналізу даних. Дані стали надзвичайно цінним ресурсом у сучасних галузях, і більшість компаній прагнуть збирати, обробляти та аналізувати релевантну інформацію, щоб отримати корисні бізнес-інсайти. У цьому контексті Python виступає беззаперечним лідером. Python цінується не лише за свою багату стандартну бібліотеку, але й за безліч додаткових модулів, створених спеціально для аналітичних задач. Серед найвідоміших бібліотек для аналізу даних – pandas і NumPy. Ці інструменти дозволяють виконувати широкий спектр операцій з даними: очищення, аналіз, статистичний аналіз і візуалізація прихованих тенденцій [13].

2. для візуалізації даних. Візуалізація даних є важливою складовою аналізу, яка дозволяє представити інформацію в зрозумілому та наочному вигляді. Python знову стає незамінним завдяки різноманіттю інструментів для візуалізації. Найпопулярнішими є бібліотеки matplotlib та seaborn.

Використовуючи ці інструменти, можна створювати всі види графіків і діаграм, від простих до складних.

3. для машинного навчання. Машинне навчання (ML) є ключовою складовою сучасної науки про дані. Це розділ нейронної мережі, який використовує алгоритми для того, щоб знайти необхідність у даних і створення моделей для прогнозування. За допомогою методів ML можна створювати моделі, що прогнозують відтік клієнтів, оцінюють ризики захворювань, оптимізують розміщення таксі тощо. Python дозволяє створювати моделі машинного навчання за допомогою кількох рядків коду [15].

4. для розробки програмного забезпечення. Python широко застосовується на всіх етапах розробки ПЗ, включаючи контроль версій, автоматичну безперервну інтеграцію, прототипування, виявлення помилок, тестування та підтримку. Ця мова дозволяє створювати програми для роботи з аудіо та відео, застосовувати методи штучного інтелекту та машинного навчання, працювати з API та GUI, а також розробляти інші типи програмного забезпечення.

5. для автоматизації задач/скриптингу. Python є чудовим інструментом для написання скриптів, які автоматизують рутинні завдання та підвищують ефективність роботи.

2.3. ER-діаграма та діаграма варіантів використання додатку

ER-діаграма є графічною репрезентацією, що показує взаємозв'язки між сутностями в додатку (об'єктами). Цей тип діаграми використовується для моделювання баз даних, їх налаштування для подальшої експлуатації, розробки додатків та проведення аналізів. Основними елементами ER-діаграми, які ілюструють зв'язки між об'єктами різними геометричними фігурами.

На даній ER-діаграмі вказано такі взаємозв'язки між сутностями:

- Меню – Продукти: 1 – N;
- Меню – Замовлення: 1 – N;
- Меню – Обслуговуючий персонал: 1 – N;

Крім того, кожна сутність має первинні та вторинні ключі, за винятком "Продуктів":

- Меню: код меню, код продукту;
- Замовлення: код замовлення, код меню;
- Обслуговуючий персонал: код обслуговуючого персоналу, код меню;
- Продукти: код продукту (лише первинний ключ).

Крім ключів, сутності також містять інші власні поля:

- Меню: назва продукту, ціна;
- Замовлення: кількість, ціна;
- Обслуговуючий персонал: прізвище, ім'я, вік;
- Продукти: назва продукту, ціна.

Утворена діаграма зображена на рисунку 2.1.

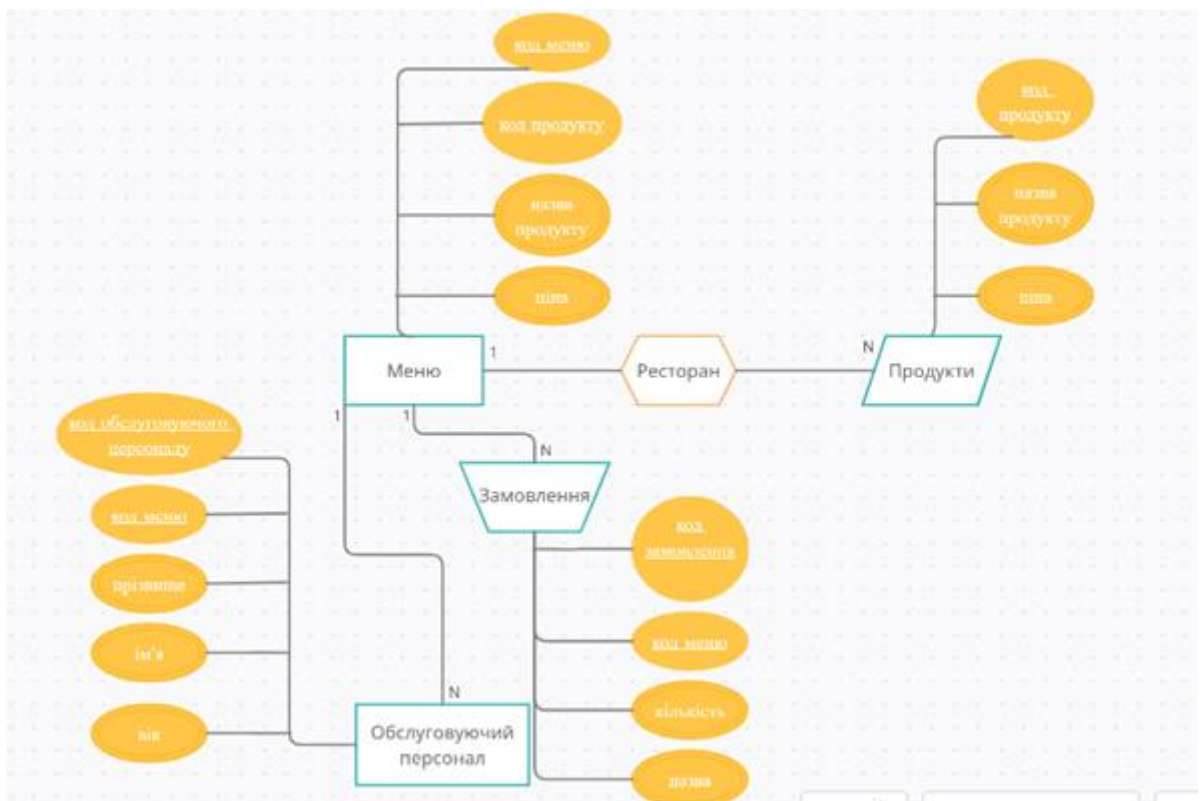


Рис. 2.1. ER-діаграма для додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії

Діаграма прецедентів у UML – це зображення, яке показує взаємозв'язки між акторами (користувачами додатку) та прецедентами (функціональними

можливостями додатку). Ця діаграма дозволяє представити додаток як набір можливих дій, які користувачі можуть виконувати взаємодіючи з додатком. Кожен прецедент визначає певну діяльність, яку додаток виконує для користувачів.

Інформація про користувачів буде зберігатися в таблиці з назвою "User".

Ця таблиця міститиме наступні дані про кожного користувача:

- Ім'я користувача (Name)
- Вік (Age)
- Стать (Gender)
- Адреса електронної пошти, яка буде використовуватися для авторизації (Email)
- Фотографія користувача перед змінами (Beforeimage)
- Фотографія користувача після змін (Afterimage)
- Ідентифікатор користувача, який є первинним ключем (UserID)
- Зовнішні ключі WaterBalance і BodyParameters, які вказують на інші таблиці з даними про водний баланс та параметри тіла відповідно.

У контексті додатку, діаграма прецедентів відображає взаємодію користувачів (акторів) з різними функціями додатку (прецедентами). Наприклад, прецедентами можуть бути реєстрація в додатку, пошук інформації, відправлення повідомлень або замовлення товарів. Кожен прецедент відображається у вигляді операції або послуги, яку додаток надає користувачам, та визначається у контексті акторів, які взаємодіють з цими функціями.



Рис. 2.2. Ключовими складовими цієї діаграми є варіант використання, актор і інтерфейс

Прецедент використовується для структурованого опису загальних особливостей роботи додатку, не затрагуючи при цьому організацію його компонентів (наприклад, замовлення товару, перевірка кредитної історії клієнта, відображення графіків).

Актори є зовнішніми сутностями додатку, які використовують його функціонал для того, щоб вирішити конкретні завдання. Вони виконують певні ролі під час взаємодії з додатком. Ім'я актора повинне бути інформативним та відображати його роль у системі (наприклад, відвідувач ресторану, менеджер магазину, пацієнт лікарні, клієнт туристичної фірми).

У цій роботі в додатку визначені такі наступні:

- Користувач, який може оформити замовлення та перевіряти потрібну інформацію в додатку;
- Адміністратор, який володіє більш широким спектром прав.

Для кращого управління завданнями виначимо пріоритет виконання для кожної задачі(де 5 – найвищий пріоритет, 1 – найнижчий).

Задачі проєкту із створення додатку з автоматизації обліку та фільтрації
даних клієнтів у закладі харчової індустрії

Пріоритет	Задачі
4	Після входу в додаток користувачам буде доступна можливість вибору вільного столу завдяки інтерактивній схемі посадочних місць. Обраний стіл буде автоматично позначений як "Зайнято".
5	Після того, як клієнт сість за стіл, він отримає доступ до інтерактивного
5	Після того, як клієнт сість за стіл, він отримає доступ до інтерактивного графічного меню.
5	Додаток автоматично відстежуватиме години праці співробітників для подальшого розрахунку заробітної плати на основі звітів про робочі зміни.
3	Після того, як клієнт оплатить рахунок і покине стіл, додаток сповістить персонал, щоб стіл можна було прибрати.
3	Менеджерам та офіціантам буде доступна функція швидкої зміни статусу столів для ефективної організації роботи. .
4	Клієнти зможуть розділити рахунок за допомогою додатку.
3	Додаток дозволить клієнтам здійснювати попереднє бронювання місць в ресторані.
4	Додаток автоматично передаватиме замовлення шеф-кухару для приготування та додасть їх в чергу відповідно до часу їх надходження.
3	Клієнти зможуть замовити їжу на винос або доставку.
4	Додаток дозволить клієнтам реєструватися для участі в програмі винагород та акціях.
4	Кожна покупка клієнта буде супроводжуватися наданням бонусів.
4	Бонуси будуть автоматично застосовуватися до загальної суми рахунку при їх використанні для оплати.
4	Додаток буде сповіщати офіціантів про готовність замовлення клієнта для подачі.
3	Меню в додатку буде доступне англійською мовою.
5	Клієнти зможуть фільтрувати позицію в меню згідно їхніх дієтичних обмежень або вподобань та сповіщати кухарів про будь-які видозміни у замовленні.
2	В додатку можна залишати відгук стосовно меню
3	Офіціанти зможуть відзначати столи як "Зайнято" в додатку.
4	Клієнти матимуть можливість викликати офіціанта за допомогою кнопки у додатку.
3	Додаток автоматично оновить статус столу на "Вільно", якщо його прибере ранер.

5	Клієнти матимуть можливість оплатити рахунок на місці за допомогою терміналу, якщо це необхідно
---	---

2.4. Аналіз основних алгоритмів роботи додатку

Для кращого розуміння роботи додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії і полегшення розробки була спроектована use-case діаграма з описом роботи додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії за участі акторів.

Опис діаграми:

UC – 1: Reservation – Дає можливість зарезервувати стіл онлайн.

UC – 2: Payment – Клієнти можуть отримувати рахунок за замовлення, вибрати формат чеку (електронний / паперовий) та обирати спосіб оплати.

UC – 3: View Menu – Дає можливість користувачам переглядати весь асортимент страв у додатку.

UC – 4: Meal Prep – Дозволяє кухарям отримувати оновлену інформацію про статус замовлень та спеціальні вимоги до страв.

UC – 5: Rate Food – Дозволяє клієнтам залишити відгук стосовно обслуговування та страв.

UC – 6: Food Filters – Дає можливість користувачам відфільтрувати меню за власними уподобаннями та змінювати склад страв.

UC – 7: Clocking In / Out – Дозволяє працівникам вказувати початок та кінець робочої зміни.

UC – 8: Serving – Дозволяє офіціантам відстежувати статус замовлень та клієнтів.

UC – 9: Placing an Order – Дає можливість користувачам робити замовлення на страву.

UC – 10: Table Marking – Дозволяє персоналу вказувати статус столів ("Зайнято", "Вільно").

UC – 11: Earning Rewards – Кожен раз, коли клієнт робить покупку, він отримує бонуси.

UC – 12: Redeeming Rewards – Клієнти можуть використовувати бонуси для оплати замовлень.

UC – 13: Take-Out – Клієнти можуть замовити страви для самовивозу.

UC – 14: Table Selection – Дозволяє користувачам вибирати не зарезервоване місце у закладі.

UC – 15: Floor Plan Status – Дозволяє менеджерам і персоналу змінювати статуси столів.

UC – 16: Login – Дозволяє користувачеві увійти до додатку, при цьому інтерфейс, який він побачить і функціональність, якою він зможе користуватися, залежатимуть від його ролі в системі.

UC – 17: Create Account – Дозволяє користувачам створювати нові облікові записи.

UC – 18: Translation – Дозволяє користувачам вибирати мову перегляду меню.

UC – 19: Menu Changes – Дозволяє менеджерам вносити зміни до асортименту страв.

На Use-Case діаграмі відображаються всі взаємозв'язки між Акторами, Акторами-учасниками, Випадками використання та базою даних.

Актори та актори-учасники – замовник/гість, шеф-кухар, ведучий, офіціант / офіціантка, менеджер, ранер. Іноді взаємодія ініціюється актором. Іноді взаємодія поширюється на інші, що є пов'язаними з ними у роботі. Описуються випадки взаємодії з базою даних.

Оплата

Актор: Клієнт.

Мета актора: Оплатити виконане замовлення.

Актори-учасники: База даних, Офіціант.

Передумови:

– Користувач встановив додаток.

- Користувач увійшов як Клієнт і відкрив меню.

Постумови:

- Користувачу потрібно вибрати спосіб оплати.
- Користувачеві потрібно вибрати спосіб отримання.

Перелік подій для основного сценарію успіху:

- Клієнт натискає кнопку «Оформити замовлення» на екрані меню.
- Додаток відображає меню оплати.
- Додаток відображає варіанти способу оплати.
- Клієнт обирає кнопку «Готівка» для оплати готівкою.
- Додаток пропонує клієнтові варіанти отримання чеку.
- Клієнт вибирає кнопку «Друк квитанції».
- Додаток повідомляє офіціанту / офіціантці надрукувати квитанцію та повернути її замовнику.

Перелік подій для альтернативного сценарію успіху:

- Клієнт натискає кнопку «Оформити замовлення» на екрані меню.
- Додаток відображає меню оплати.
- Додаток відображає варіанти способу оплати.
- Клієнт обирає кнопку «Кредит» для оплати кредитною картою.
- Додаток пропонує клієнту варіанти отримання чеку.
- Клієнт вибирає кнопку «Квитанція електронною поштою».
- Додаток пропонує користувачеві ввести свою електронну пошту.
- Клієнт вказує електронну пошту і отримує туди квитанцію.

Опис діаграми.

Клієнт натискає кнопку оплати та обирає метод, яким планує оплатити. База даних реєструє необхідну інформацію. У випадку оплати готівкою, офіціант отримує сповіщення та приймає готівку. У разі обрання карткового платежу, інформація автоматично надсилається до Stripe API для обробки, після чого Stripe повертає результат операції.

Перегляд меню

Актор: Клієнт.

Мета актора: Переглянути меню.

Актори-учасники: База даних.

Передумови:

- Користувач встановив додаток.

Постумови:

- Користувачеві показується доступне меню.

Перелік подій для основного сценарію успіху:

- Додаток надає параметри для «Вхід», «Продовжити як гість» та «Створити рахунок».
- Клієнт вирішує увійти до свого облікового запису.
- Додаток пропонує клієнтові увійти зі своїми даними користувача.
- Клієнт вводить інформацію для входу у додаток.
- Додаток надає опції «Dine-In» або «Take-Out».
- Клієнт обирає параметр «Dine-In».
- Додаток дозволяє замовнику вибрати вільний столик.
- Клієнт сідає за столик.
- Додаток надає можливості вибору дієтичних обмежень.
- Клієнт вирішує не фільтрувати меню.
- Додаток відображає меню.

Перелік подій для альтернативного сценарію успіху:

- Додаток надає параметри «Вхід», «Реєстрація» або «Продовжити як гість».
- Клієнт вибирає «Продовжити як гість».
- Додаток надає параметри «Dine-In» або «Take-Out».
- Клієнт вибирає опцію «Dine-In».
- Додаток дає можливість замовнику обрати вільне місце.
- Клієнт сідає за вільний столик.
- Додаток дає можливість вибрати будь-які дієтичні обмеження.
- Клієнт вирішує не фільтрувати меню.
- Додаток відображає доступне меню.

Опис діаграми.

Клієнт запитує додаток показати меню, після чого додаток звертається до бази даних для отримання відповідної інформації про меню, яка відповідає будь-якому заданому фільтру. Після цього пункти меню відображаються користувачеві у додатку.

Відкриття / закриття зміни

Актор: Співробітник.

Мета актора: Чітко фіксувати час, який працівники витрачають на роботу, і забезпечити перевірку присутності працівника у закладі харчування під час початку та завершення робочої зміни.

Актори-учасники: всі працівники закладу галузі харчування.

Передумови:

- Користувач закладу індустрії харчування встановив додаток.

Постумови:

- Користувач успішно здійснив вхід / вихід.
- База даних реєструє інформацію для подальшого використання. Перелік подій для основного сценарію успіху:
- Співробітник відкриває програму та обирає вхід як звичайний працівник.
- Додаток запитує працівника дані входу в додаток.
- Працівник автентифікується в додатку.
- Додаток відкриває співробітнику інтерфейс робітників.
- Працівник натискає на кнопку входу.
- Додаток зберігає інформацію про робочу зміну.
- Додаток перевіряє, чи працівник насправді знаходиться в закладі індустрії харчування, перевіряючи IP-адресу та точку на мапі.
- Коли працівник завершує свою робочі зміну протягом дня, він має обрати кнопку виходу.
- Додаток відстежує дані часу виходу співробітника.
- Додаток перевіряє, чи працівник насправді знаходиться в закладі індустрії харчування, перевіряючи IP-адресу та точку на мапі.

Перелік подій для альтернативного сценарію успіху:

- Співробітник відкриває програму і вирішує увійти в додаток як менеджер.
- Додаток дає можливість вказати дані для входу.
- Менеджер автентифікується в додатку.
- Додаток відображає менеджеру потрібний інтерфейс.
- Менеджер натискає на кнопку входу.
- Додаток відстежує інформацію стосовно робочих годин менеджера.
- Додаток перевіряє, що менеджер насправді знаходиться в закладі, перевіряючи IP- адресу та точку на мапі.
- Коли менеджер закінчить зміну за день, менеджер вибере кнопку виходу.
- Додаток відстежує інформацію стосовно часу виходу менеджера.
- Додаток перевіряє, що менеджер насправді знаходиться в закладі індустрії харчування, перевіряючи IP-адресу та точку на мапі.

Після запуску додатка клієнти спочатку побачать заставковий екран, а потім будуть перенаправлені на домашню сторінку. Звідти вони матимуть можливість створити новий обліковий запис безкоштовно, увійти вже існуючими даними або продовжити використання додатка як гість.

Увійшовши в додаток, клієнти мають можливість вибрати опцію "Пообідати у закладі індустрії харчування". Якщо вони обирають замовлення з собою, їх безпосередньо перенаправляють до меню, тоді як вибір обіду у закладі індустрії харчування дозволяє клієнту обрати доступний стіл за їхнім вибором. Після цього клієнт може оглядати меню та додавати до замовлення все, що вони бажають. Увійшовши в додаток, вони можуть взаємодіяти зі значка дзвоника вгорі праворуч, щоб надіслати сповіщення офіціанту для виконання будь-яких запитів. Після зробленого остаточного вибору вони можуть перевірити своє замовлення та оплатити його. Після успішної транзакції, за потреби можливо надіслати квитанцію на електронну пошту клієнта

Щодо інтерфейсу адміністратора (менеджера), після входу в додаток через відповідний екран він отримує доступ до кількох функцій. Перша кнопка дозволяє переглядати поточний стан всіх столиків у закладі. Друга кнопка

викликає електронний лист з інформацією про заробітну плату. Третя кнопка веде менеджера до інтерфейсу, де він може переглядати всіх поточних співробітників і отримувати інформацію про кожного окремо. Наступна кнопка дозволяє переглядати меню. Функція "Черга замовлень" дозволяє менеджеру переглядати статус усіх поточних замовлень. Редагування меню дозволяє додавати, редагувати та видаляти позиції страв з меню, а також здійснювати редагування всіх властивостей кожної позиції.

ВИСНОВКИ ДО РОЗДІЛУ 2

В цьому розділі було здійснено вибір архітектури програмного забезпечення для автоматизації обліку та фільтрації даних клієнтів, розглянуто характеристики мови програмування Python, побудовано ER-діаграму та діаграму варіантів використання додатку, а також проаналізовано основні алгоритми роботи додатку. На основі проведеного аналізу обґрунтовано, що обрані засоби та методи реалізації проекту є оптимальними для досягнення поставлених цілей. Використання Python як основної мови програмування забезпечує високу продуктивність і гнучкість у розробці додатку, а обрана архітектура сприяє зручному масштабуванню та підтримці системи в майбутньому.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ

3.1. Математична модель автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії

Опишемо математичну модель програмного ядра додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії. Застосування методів статистичного моделювання Вінерівських і узагальнених Вінерівських процесів стає дедалі ширшим. Зокрема, моделі вінерівських і узагальнених вінерівських випадкових процесів знаходять використання у фінансовій математиці, для розв'язання завдань обчислювальної математики, при визначенні розмірності Хаусдорфа, а також у задачах моделювання функціонування закладів харчової індустрії.

В даному дослідженні використано метод статистичного моделювання дробового броунівського руху для моделювання роботи закладу харчової індустрії з метою автоматизації обліку та фільтрації даних клієнтів. Метод базується на властивості дробового броунівського руху, що має стаціонарні прирости, і створює реалізацію руху з заданою проекцією.

Математичний опис програмного продукту для автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії є важливим завданням, яке потребує точного і ефективного моделювання складних систем і процесів. Використання методу дробового броунівського руху (fBm) для моделювання роботи закладу харчової індустрії з метою автоматизації обліку та фільтрації даних клієнтів є актуальним з кількох причин:

1. Стаціонарність Приростів: Дробовий броунівський рух має властивість стаціонарних приростів, що означає, що статистичні властивості процесу не змінюються з часом. Це є надзвичайно корисним для моделювання

Кафедра КІТ				НАУ 24 44 31 000 ПЗ			
	ПІБ			ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ	Літ.	Аркуш	Аркушів
Виконав	Припутненко Р. В.					37	80
Керівник	Рибасова Н. О.				ТП-415Б - 122		
Н.Контролер	Сидоренко В.М.						

поведінки клієнтів, яка може бути постійною протягом тривалих періодів часу.

2. Гнучкість і Точність: fVm дозволяє точно моделювати залежності між подіями та процесами, що є важливим для створення точних алгоритмів обліку і фільтрації даних. Це забезпечує високу точність прогнозування і аналізу поведінки клієнтів.

3. Обробка Випадкових Процесів: Використання методів, заснованих на fVm, дозволяє ефективно обробляти і аналізувати випадкові процеси, які часто зустрічаються в роботі закладів харчової індустрії, таких як варіації в попиті, зміни в поведінці клієнтів, та інші випадкові фактори.

4. Автоматизація та Оптимізація: Метод дозволяє автоматизувати процеси обліку та фільтрації даних, що знижує людський фактор і помилки, підвищуючи ефективність і надійність системи. Це також сприяє оптимізації ресурсів і процесів у закладі.

5. Універсальність і Застосовність: Методи, засновані на fVm, можуть бути застосовані в різних галузях, включаючи фінансову математику та обчислювальні задачі. Це свідчить про їхню універсальність і ефективність у вирішенні складних завдань, що також актуально для харчової індустрії.

6. Інноваційність і Конкуренентоспроможність: Використання передових математичних методів, таких як fVm, для моделювання і автоматизації процесів, забезпечує закладу інноваційну перевагу і підвищує його конкурентоспроможність на ринку.

Таким чином, застосування методу дробового броунівського руху в програмному ядрі додатку для автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії є актуальним і перспективним рішенням, що дозволяє підвищити ефективність, точність та надійність системи, сприяючи покращенню обслуговування клієнтів та оптимізації роботи закладу.

Використання методу дробового броунівського руху для моделювання роботи закладу харчової індустрії з метою автоматизації обліку та фільтрації даних клієнтів дозволяє створити точні і ефективні алгоритми, що підвищують

ефективність роботи закладу, забезпечуючи точність прогнозування і оптимізацію ресурсів.

Для прогнозування потоку клієнтів у закладі використано властивість стаціонарних приростів fVm. Це дозволить передбачати кількість клієнтів на певний період часу на основі попередніх даних.

Кроки реалізації:

1. Зібрати історичні дані про кількість клієнтів.
2. Використовувати fVm для моделювання приросту кількості клієнтів.
3. Реалізувати алгоритм прогнозування потоку клієнтів.

Для автоматизації обліку та фільтрації даних клієнтів використано методи fVm для виявлення аномалій та трендів у даних.

Кроки реалізації:

1. Зібрати дані про транзакції та взаємодії клієнтів.
2. Використовувати алгоритми fVm для аналізу даних.
3. Реалізувати фільтрацію та класифікацію даних на основі результатів аналізу.

Для оптимізації запасів і ресурсів використано моделі fVm для прогнозування попиту на продукти та матеріали.

Кроки реалізації:

1. Зібрати дані про попит на продукти.
2. Використовувати fVm для моделювання змін у попиті.
3. Реалізувати алгоритм оптимізації запасів на основі прогнозу.

Для реалізації математичних моделей у програмному продукті використано мову програмування, такі як Python, яка має потужні бібліотеки для роботи зі статистичними моделями.

Визначення 1. Узагальнений Вінерівський процес (дробовий броунівський рух) з індексом Херста $H \in (0, 1)$ називається гаусівський процес $W^H(t)$, $t \in [0, 1]$ такий, що $W^H(0) = 0$, $E W^H(t) = 0$ і кореляційною функцією. Наведено приклад у формулі (3.1).

$$R_H(t, s) = \frac{1}{2} (|t|^{2H} + |s|^{2H} - |t-s|^{2H}) \quad (3.1)$$

Якщо $H = 0,5$, то отримаємо стандартний Вінерівський процес.

Вінерівський процес $W(t)$ – це процес з незалежними приростами. Дробовий Броунівський рух $W_H(t)$ – це процес зі стаціонарними приростами.

Тоді випадковий процес $w(t) = W_H(t + \Delta) - W_H(t)$ з фіксованим Δ є стаціонарним гаусівським процесом з кореляційною функцією. Приклад наведено в формулі (3.2).

$$E w(t + \tau) w(t) = \frac{1}{2} (|\tau + \Delta|^{2H} + |\tau - \Delta|^{2H} - 2|\tau|^{2H}) \quad (3.2)$$

і спектральну щільність показано в формулі (3.3).

$$g(\lambda) = \frac{A^2}{\pi} \left(\frac{1 - \cos(\lambda\Delta)}{|\lambda|^{2H+1}} \right), \lambda \in (-\infty, +\infty), \quad (3.3)$$

де:

$$A^2 = \left(\frac{2}{\pi} \int_0^{\infty} \frac{1 - \cos(\lambda)}{\lambda^{2H+1}} d\lambda \right)^{-1} = \left(-\frac{2}{\pi} \Gamma(-2H) \cos(H\pi) \right)^{-1} \quad (3.4)$$

Оскільки $W_H(0) = 0$, то ми можемо зобразити будь-яку модель дробового броунівського руху наступним чином (3.5).

$$W_H(t + \Delta) = W_H(t) + w(t) \quad (3.5)$$

Моделювання дробового броунівського руху зводиться до моделювання гаусівського стаціонарного процесу.

Нехай $\xi(t)$ – це дійсний гаусівський стаціонарний випадковий процес з кореляційною функцією $R(\tau)$ і спектральною функцією, як у нас показано в формулі (3.6)

$$F(\lambda), \quad R(\tau) = \int_0^{\infty} \cos(\lambda t) dF(\lambda) \quad (3.6)$$

Гаусівський стаціонарний випадковий процес можна представити у вигляді наступної формули (3.7)

$$\xi(t) = \int_0^{\infty} \cos(\lambda t) d\xi_1(\lambda) + \int_0^{\infty} \sin(\lambda t) d\xi_2(\lambda) \quad (3.7)$$

де $\xi_1(t)$ та $\xi_2(t)$ – центровані і некорельовані випадкові процеси, такі що при $0 < \lambda_1 < \lambda_2$ має місце, показано в формулі (3.8)

$$\begin{aligned} E(\xi_1(\lambda_2) - \xi_1(\lambda_1))^2 &= F(\lambda_2) - F(\lambda_1), \\ E(\xi_2(\lambda_2) - \xi_2(\lambda_1))^2 &= F(\lambda_2) - F(\lambda_1). \end{aligned} \quad (3.8)$$

Нехай $D\Lambda$ – деяке розбиття інтервалу $[0, \Lambda]$. Модель випадкового процесу $\xi(t)$ може бути представлена у вигляді центральної некорельованої субгаусівської послідовності, показано в формулі (3.9).

$$E(\eta_{1i})^2 = E(\eta_{2i})^2 = F(\lambda_{i+1}) - F(\lambda_i). \quad (3.9)$$

Тобто випадковий процес $w(t)$ можна представити у вигляді формули (3.10).

$$w(t) = \int_0^{\infty} \cos(\lambda t) d\xi_1(\lambda) + \int_0^{\infty} \sin(\lambda t) d\xi_2(\lambda). \quad (3.10)$$

Для розбиття/модель випадкового процесу $w(t)$ маємо формулу (3.11).

$$w_n(t, \Lambda) = \sum_{k=0}^{n-1} (\sin(\lambda_k t) X_k + \cos(\lambda_k t) Y_k), \quad (3.11)$$

де $\{X_k, Y_k\}$ – некорельовані субгаусівські випадкові величини з $EX_k = EY_k = 0$.

$$E(X_k)^2 = E(Y_k)^2 = \int_{\lambda_k}^{\lambda_{k+1}} g(\lambda) d\lambda. \quad (3.12)$$

Нехай випадковий процес $X(t)$ і всі $X_n(t, \Lambda)$ належать деякому функціональному простору $A(T)$ з нормою $\|\cdot\|$. Нехай дано два числа: $\delta > 0$ і $0 < \varepsilon < 1$.

Визначення 2. Модель $X_n(t, \Lambda)$ апроксимує процес $X(t)$ з надійністю $1 - \varepsilon$ і точністю $\delta > 0$ в нормі простору $A(T)$, якщо існує нерівність, наведена в формулі (3.13):

$$P\{\|X(t) - X_n(t, \Lambda)\| > \delta\} \leq \varepsilon. \quad (3.13)$$

Нехай для $D\Lambda$: / та $\frac{T\Lambda}{n} \leq 1$.

Теорема 1. Модель $W_n(t, \Lambda)$ апроксимує процес $w(t)$ з надійністю $1-\varepsilon$ і точністю δ у просторі $L_2(T)$, якщо для чисел Λ і n виконуються нерівності : / та

$$\exp\left\{\frac{1}{2}\right\} \frac{\delta}{\sqrt{G_{2n,\Lambda}}} \exp\left\{-\frac{\delta^2}{2G_{2n,\Lambda}}\right\} \leq \varepsilon, \quad (3.14)$$

Де
$$G_{2n,\Lambda} = \frac{T^3 \Lambda^2}{3n^2} \int_0^\Lambda g(\lambda) d\lambda + T \left(\int_\Lambda^\infty g(\lambda) d\lambda \right).$$

Для чисел Λ і n ставимо:

$$\Lambda = \left(\frac{3n^2}{T^3} \right)^{\frac{1}{2H+2}}, \quad G_{2n} = T^{\frac{3H}{H+1}} \left(1 + \frac{T}{H} \right) \left[\left(3n^2 \right)^{\frac{H}{H+1}} \right]^{-1}. \quad (3.15)$$

Як ми бачимо, у цій математичній моделі мережа представлена у вигляді графа $G = (V, E)$, де $V \square \{1, 2, \dots, N\}$ та $E = \{1, 2, \dots, M\}$ - множини маршрутизаторів і M ліній зв'язку між ними. Пряма лінія зв'язку m має місткість u_m (в одиницях/сек). Уся множина вузлів в мережі MPLS розбивається на дві підмножини: $V^+ = \{V_i^+, i=1, n_{LER}\}$ - множина граничних маршрутизаторів LER і $V^- = \{V_i^-, i=1, n_{LSR}\}$ – множина комутаторів міток LSR.

Тоді кожен елемент множини V^\square може бути як джерелом трафіку, так і одержувачем. Якщо маршрутизатор є джерелом трафіку, то розуміється, що на даний маршрутизатор надходить трафік з суміжної мережі (IP, MPLS, ATM або ін.). Трафік буде доставлений на вузол отримувача, який також буде точкою дотику із суміжними мережами. Будемо розглядати випадок, коли кожен граничний маршрутизатор LER є і джерелом, і одержувачем, а LSR не може бути одержувачем трафіку, що надійшов на нього із суміжної мережі.

Припустимо, що в кожен момент $t = T$ на один із маршрутизаторів надходить трафік інтенсивністю $\lambda(t)$, що належить до одного з класів обслуговування $q \in Q$ з вимогами QoS, якому відповідають значення максимально допустимої затримки Q_q і максимально допустимого відсотку втрат l_q . Весь вхідний трафік розбивається на потоки класів обслуговування / таким

способом, щоб забезпечити передачу вимог усіх класів $Q(t)$ у повному обсязі. Тоді множина всіх каналів QoS трафіку Y (в одній лінії зв'язку m може існувати множина каналів $y=Y$) має вигляд формули (3.16).

$$Y = Y(d_y, P_y, L_y), \quad (3.16)$$

де d_y – смуга пропускання каналу; $P_y = \{p_y^1, \dots, p_y^{L_y}\}$ – допустима множина шляхів для шляху L_y певна для кожного каналу трафіку.

В даній ситуації інтенсивність трафіку при автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії можна вказати: в момент t на граничний маршрутизатор надходить трафік інтенсивністю λ , який належить до q -го класу обслуговування, який необхідно доставити на вихідний маршрутизатор по будь-яких шляхах із множини P_y , щоб не перевищити заданих максимально допустимих значень затримки Q_q і максимально допустимого відсотку втрат.

Кожен вузол мережі V в t -й момент характеризується продуктивністю, коефіцієнтом відносних втрат / і середнім часом очікування пакета в черзі. На величину втрат для всіх вузлів мережі накладаються обмеження, показано в формулі (3.17).

$$0 \leq X_V^q(t), \sum_V^{P_y} X_V^q(t) \leq l_q, \quad (3.17)$$

де $V = \overline{1, P_y}$, l_q – заданий максимально допустимий відсоток втрат. Втрати визначаються як співвідношення між кількістю відкинутих даних та кількістю даних, що надійшли для обробки. Обмеження на час затримки для всіх вузлів мережі визначаються таким самим чином, показано в формулі (3.18).

$$0 \leq T_V^q(t), \sum_V^{P_y} T_V^q(t) \leq \tau_q, \quad (3.18)$$

де $V = \overline{1, P_y}$, τ_q – встановлено максимально допустимий час затримки затримки. Вартість маршрутизації c_m призначається лінії зв'язку m і може бути

залежним від багатьох параметрів, а саме: швидкості, довжини, надійності та ін. Вартість шляху дорівнює сумі вартості ліній зв'язку, показано в формулі (3.20).

$$C_y^l = \sum_{m \in p_y^l} c_m. \quad (3.19)$$

Якщо це рівняння являє собою пропускну здатність закладу харчової індустрії, яка направляє на дозволений маршрут каналу передачі у трафіку, тоді діє наступне (3.20):

$$\sum_{t \in T; l=1}^{L_y} x_y^l(t) = d_y, \quad \forall y \in Y, \quad \forall l \in \{1, \dots, L_y\}. \quad (3.20)$$

Сукупне навантаження на зв'язок m визначається за наступною формулою (3.21):

$$f_m = \sum_{\lambda_{p_y^q}(t) \in Y} \sum_{l=1}^{L_y} a_{y,l}^m x_y^l(t), \quad \forall m \in E, \quad (3.21)$$

якщо шлях P у використовує лінію зв'язку m , в іншому випадку;

$x_y^l(t)$ – пропускну здатність закладу харчової індустрії при автоматизації обліку та фільтрації даних клієнтів згідно вище приведеного рівняння.

Представлена модель дозволяє розробити комплекс методів для оптимізації автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії і структур мереж із технологією MPLS для віддаленої роботи.

Створений додаток з графічним інтерфейсом дозволяє запускати SQL-запити через браузер. Дизайн програми розроблено з урахуванням запитів різних користувачів: а саме адміністраторів баз даних, розробників, системних адміністраторів та підприємств, що надають послуги клієнтам.

У додатку відсутні функції реєстрації та авторизації клієнтів, тому що вони не є необхідними. Користувачі вказуватимуть всю потрібну інформацію під час оформлення замовлення. Для доведення замовлення до оформлення через додаток потрібно буде заповнити поля, вказавши при цьому особисту інформацію, а також можливі побажання.

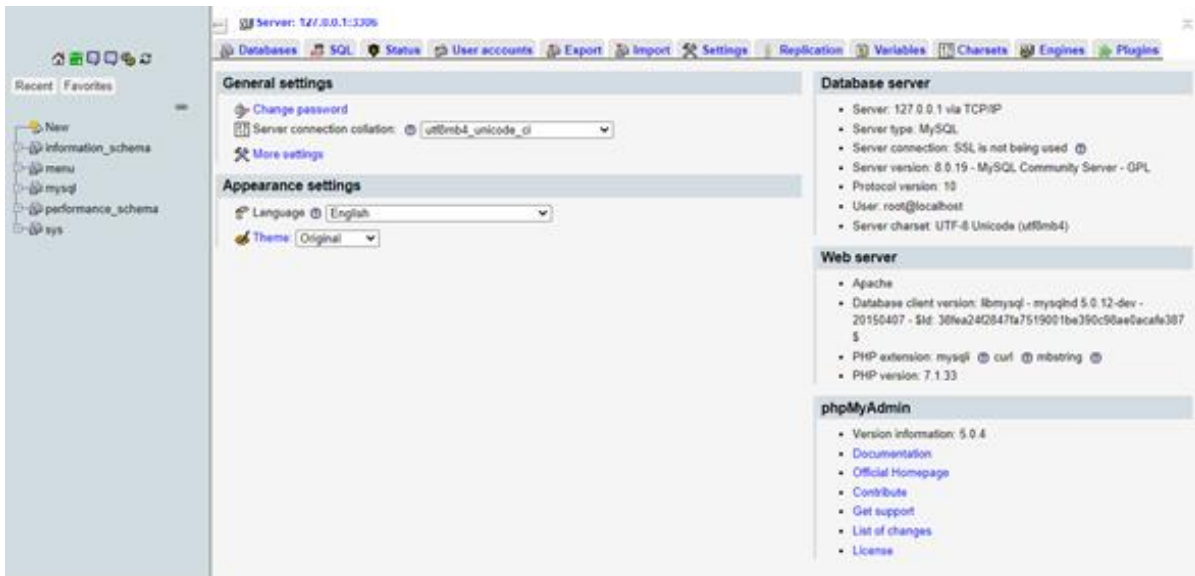


Рис. 3.1. Інтерфейс створеного додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії

Головна сторінка додатку містить стислу інформацію про заклад, без зайвих деталей. Додаток має сучасний мінімалістичний дизайн, що забезпечує приємне користування та сприяє залученню нових клієнтів. Основою успішного додатку є простота та зрозумілість навігації, що робить його ще більш зручним для користувачів (Рис. 3.2–3.4).

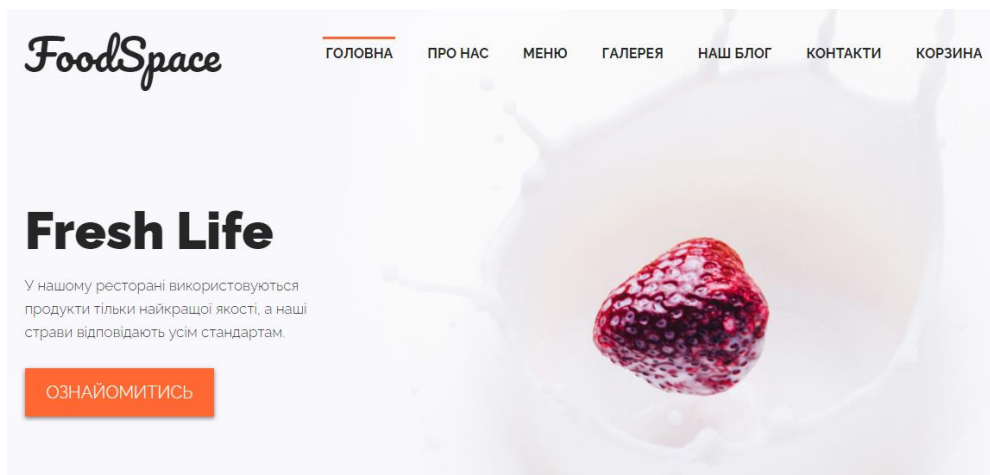


Рис. 3.2. Головна сторінка додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії

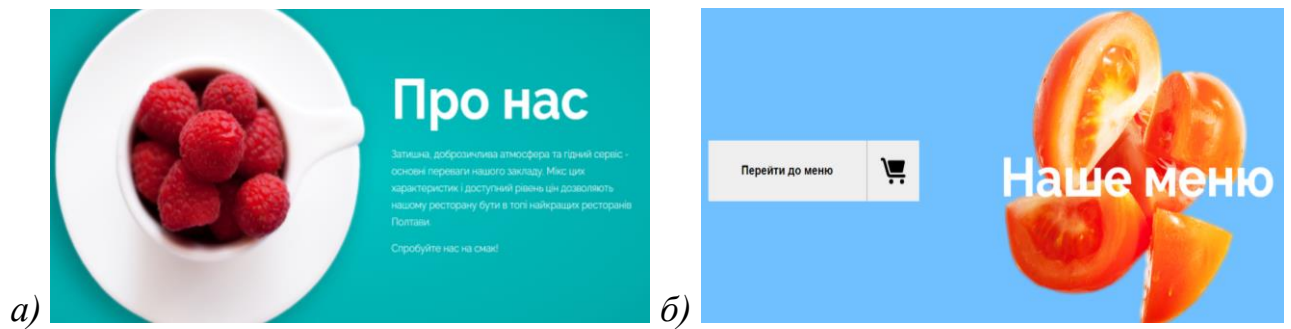


Рис. 3.3. Головна сторінка додатку з автоматизації відвідувачів закладу харчової індустрії – а) та приклад оформлення меню – б)

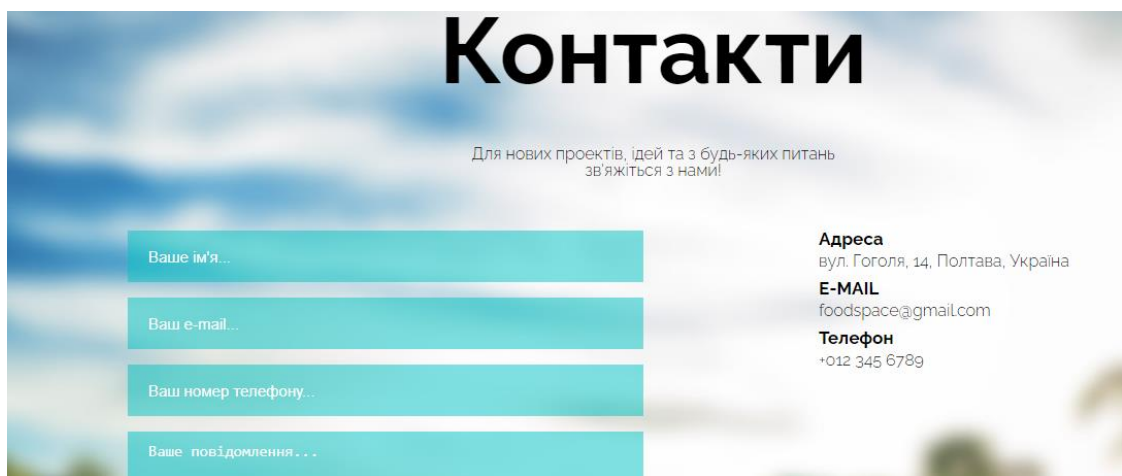


Рис. 3.4. Головна сторінка (продовження) додатку

Головна сторінка додатку містить наступні вкладки: Про нас; Меню; Галерея; Наш блог; Контакти.

Щоб отримати доступ до панелі адміністратора, виберіть «Додатково» в меню (Рис. 3.5). Потім увійдіть за допомогою електронної адреси та пароля адміністратора (Рис. 3.6).

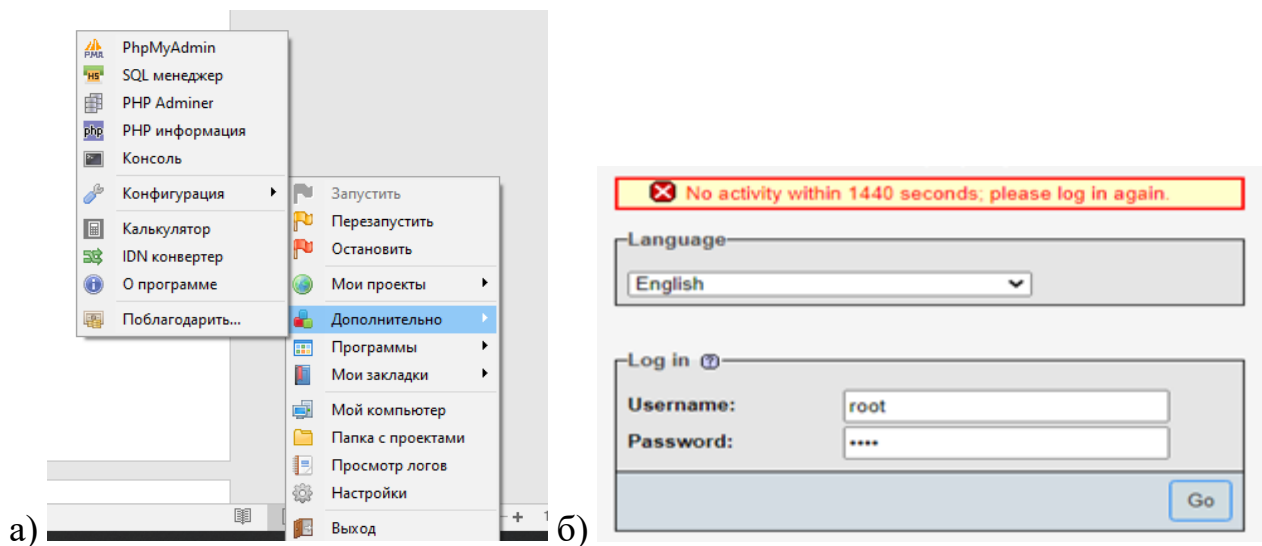


Рис. 3.5. Вхід до панелі управління адміністратора – а) та вхід до панелі управління адміністратора закладу харчової індустрії при автоматизації обліку та фільтрації даних клієнтів – б)

Щоб внести будь-які зміни до меню, адміністратору потрібно взаємодіяти з файлом Excel, а саме листом під назвою Menu (Рис. 3.6).

Назва страви	Вага	Ціна	Опис	Клас
Холодні закуски				
<i>Час приготування 15-20 хвилин</i>				
<i>Сирно-м'ясна тарілка</i>	250	165	саламі, чорізо, прошуто, Брі, Дорблю, Пармезан	В
<i>Брускетта з прошуто</i>	170	60		В
<i>Тар тар з лосося</i>	140	135		В
Основні страви				
<i>Час приготування 30-40 хвилин</i>				
<i>Шашлик зі свинини</i>	250	135		Г
<i>Стейк зі свинини</i>	300	149	подається з овочами гриль	Г
<i>Скумбрія гриль</i>	250	90		Г
<i>Картопля по-домашньому</i>	250	60		Г
<i>Картопляне пюре</i>	280	50		Г
<i>Галушки з м'ясом</i>	270	80		Г
Десерти				
<i>Панна – кота</i>	160	55		Р
<i>Брауні</i>	150	75		Р
<i>Наполеон</i>	150	80		Р

Рис. 3.6. Лист «Menu» в створеному додатку для автоматизації обліку та фільтрації даних клієнтів закладу харчової індустрії

Після того, як я створив таблицю з даними, потрібно натиснути кнопку кнопку «MakeSQL» (рис. 3.7), який вноситься безпосередньо адміністратором до вкладки

Make Table		INSERT INTO `menu`(`id`,`name`,`weight`,`price`,`description`,`class`,`image`) VALUES
		(1, 'Сирно-м'ясна тарілка', '250', '165', 'салामी, чорізо, прошуто, Брі, Дор блю, Пармезан', 'B', 'caesar.png'),
		(2, 'Брускетта з прошуто', '170', '60', '', 'B', 'caesar.jpg'),
		(3, 'Тар тар з лосося', '140', '135', '', 'B', 'caesar.jpg'),
		(4, 'Шашлик зі свинини', '250', '135', '', 'Г', 'kotleta.jpg'),
Make SQL		(5, 'Стейк зі свинини', '300', '149', 'подається з овочами гриль', 'Г', 'kotleta.jpg'),
		(6, 'Скумбрія гриль', '250', '90', '', 'Г', 'kotleta.jpg'),
		(7, 'Картопля по-домашньому', '250', '60', '', 'Г', 'kotleta.jpg'),
		(8, 'Картопляне пюре', '280', '50', '', 'Г', 'kotleta.jpg'),
		(9, 'Галушки з м'ясом', '270', '80', '', 'Г', 'kotleta.jpg'),
		(10, 'Панна – кота', '160', '55', '', 'P', 'kotleta.jpg'),
		(11, 'Брауні', '150', '75', '', 'P', 'kotleta.jpg'),
		(12, 'Наполеон', '150', '80', '', 'P', 'kotleta.jpg'),
		(13, 'Вино «Алазанська долина»', '50', '210', 'біле н/сол', 'Г', 'kotleta.jpg'),
		(14, 'Вино «Алазанська долина»', '50', '210', 'червоне н/сол', 'Г', 'kotleta.jpg'),
		(15, 'Ігристе вино «Артемівське»', '750', '210', 'біле сухе', 'Г', 'kotleta.jpg'),
		(16, 'Пиво «Kronenbourg Blanc»', '0,5', '45', 'нефільтроване', 'Г', 'kotleta.jpg'),
		(17, 'Чай чорний цейлонський', '450 мл', '35', '', 'K', 'kotleta.jpg'),
		(18, 'Еспресо', '100 мл', '25', '', 'K', 'kotleta.jpg'),
		(19, 'Американо', '150 мл', '27', '', 'K', 'kotleta.jpg'),
		(20, 'Капучино', '250 мл', '35', '', 'K', 'kotleta.jpg'),
		(21, 'Фреш апельсиновий', '0,2 л', '70', '', 'K', 'kotleta.jpg'),
		(22, 'Вода «Bonaqua»', '0,5 л', '20', 'раз/нераз', 'K', 'kotleta.jpg');

Рис. 3.7. Запит у створеному додатку для закладу харчової індустрії при автоматизації обліку та фільтрації даних клієнтів

Кнопки «MakeTable» та «MakeSQL» створились макросами. У панелі адміністратора формується вже змінена база даних (Рис. 3.8), звідки відображається інформація в самому додатку.

id	name	weight	price	description	class	image
1	Сирно-м'ясна тарілка	250	165	салामी, чорізо, прошуто, Брі, Дор блю, Пармезан	B	caesar.png
2	Брускетта з прошуто	170	60		B	caesar.jpg
3	Тар тар з лосося	140	135		B	caesar.jpg
4	Шашлик зі свинини	250	135		Г	kotleta.jpg
5	Стейк зі свинини	300	149	подається з овочами гриль	Г	kotleta.jpg
6	Скумбрія гриль	250	90		Г	kotleta.jpg
7	Картопля по-домашньому	250	60		Г	kotleta.jpg
8	Картопляне пюре	280	50		Г	kotleta.jpg
9	Галушки з м'ясом	270	80		Г	kotleta.jpg
10	Панна – кота	160	55		P	kotleta.jpg
11	Брауні	150	75		P	kotleta.jpg
12	Наполеон	150	80		P	kotleta.jpg
13	Вино «Алазанська долина»	50	210	біле н/сол	Г	kotleta.jpg
14	Вино «Алазанська долина»	50	210	червоне н/сол	Г	kotleta.jpg
15	Ігристе вино «Артемівське»	750	210	біле сухе	Г	kotleta.jpg

Рис. 3.8. База даних у створеному додатку для закладу харчової індустрії при автоматизації обліку та фільтрації даних клієнтів

3.2. Функціональні та нефункціональні вимоги

Функціональні вимоги відображають весь набір функцій розроблюваного додатку для закладу харчової індустрії при автоматизації обліку та фільтрації даних клієнтів, якими вона повинна бути наділена. Функціональні вимоги до

додатку дозволяє виявити модель прецедентів. Прецедент надає інформацію стосовно послідовності дій, які виконує додаток, щоб надати конкретні результати користувачеві. Модель прецедентів включає діаграму варіантів використання (use case diagram), яка містить представлення всіх можливих дійових осіб та їх взаємодію з різними варіантами використання. Діаграма випадків використання має опис функціональності додатку на певному рівні абстракції.

Всі параметри якості додатку, тобто атрибути якості, визначаються нефункціональними вимогами. Ці атрибути відображають користувацькі характеристики програми, такі як доступність, легкість установки, продуктивність, надійність, умови експлуатації тощо. До найбільш важливих атрибутів якості відносяться: надійність, продуктивність та зручність використання додатку.

Нефункціональні вимоги є критично важливими при розробці додатку, оскільки вони забезпечують комфортну роботу користувача з програмою.

Вимоги до надійності. Одним з ключових атрибутів якості програми для закладу харчової індустрії при автоматизації обліку та фільтрації даних клієнтів є її надійність – здатність працювати без збоїв і зберігати свої характеристики за певних умов використання.

До вимог щодо надійності додатку для автоматизації обліку та фільтрації даних клієнтів у закладах харчової індустрії включаються: Серед вимог до надійності додатку з автоматизації обліку та фільтрації даних клієнтів для закладу харчової індустрії виділяють:

1. Відновлення після збоїв. У разі апаратних збоїв (наприклад, відключення електроживлення), додаток повинен автоматично відновлюватися після усунення проблем. Винятком є ситуації, пов'язані з фізичним пошкодженням носіїв інформації або доступом до інформації на сторонньому ресурсі.

2. Обробка помилкових ситуацій. Програма повинна мати засоби інформування користувача про неправильні дії. У разі виникнення винятків

додаток повинен аналізувати і спробувати ідентифікувати помилку, після чого надати користувачеві відповідне повідомлення у разі успішної ідентифікації.

Вимоги до продуктивності. Найважливішими аспектами розробки додатку для автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії є продуктивність, тобто здатність програми швидко та якісно виконувати передбачені функції.

До вимог щодо продуктивності додатку для закладів харчової індустрії відносяться:

1. Відгуку програми повинен бути не більше 3 секунд, а час відповіді зовнішнього ресурсу на запит даних – не більше 5 секунд.
2. Програма повинна забезпечувати виведення імпортованих даних у прийнятний термін, за винятком випадків помилки імпорту.

Вимоги до ергономіки і зручності експлуатації. Користувальницький інтерфейс забезпечує взаємодію між користувачем та програмним забезпеченням, тому від його характеристик залежить ефективність роботи та задоволення користувача.

До вимог щодо зручності використання та ергономіки додатку для автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії відносяться:

- інтерфейс додатку повинен бути інтуїтивно зрозумілим для користувача;
- повідомлення про помилки повинні бути короткими, але інформативними.
- поведінка однотипних елементів інтерфейсу при взаємодії з користувачем повинна бути однаково реалізована.

3.3. Вибір архітектури серверного додатку та технологій для обладнання

Для створюваного додатку з автоматизації клієнтів в закладі харчової індустрії зазвичай застосовується тришарова організація загальних комп'ютерних програм. Три шари – це рівні презентації, програми та даних.

- 1) Презентаційний рівень – це те, що бачать користувачі на фронт-енді,

включаючи як клієнтів, так і персонал закладу харчової індустрії.

2) Шар програм та інформації, який знаходиться на бек-енді. Він моніторить введену користувачем інформацію та реагує на команди з екрану. Така організація підходить для нашого застосування, оскільки шари функціонують незалежно один від одного, забезпечуючи роботу в одночасному режимі.

Підсистема, зображена на рисунку 3.9, демонструє нашу тришарову архітектуру, яка включає рівень презентації, рівень додатків та рівень даних. Кожен шар відповідає за певну частину системи.

3) Шар презентації відповідає за інтерфейс користувача. Він містить інформацію про шеф-кухаря, менеджера, ранера/офіціанта/офіціантку та клієнтські інтерфейси.

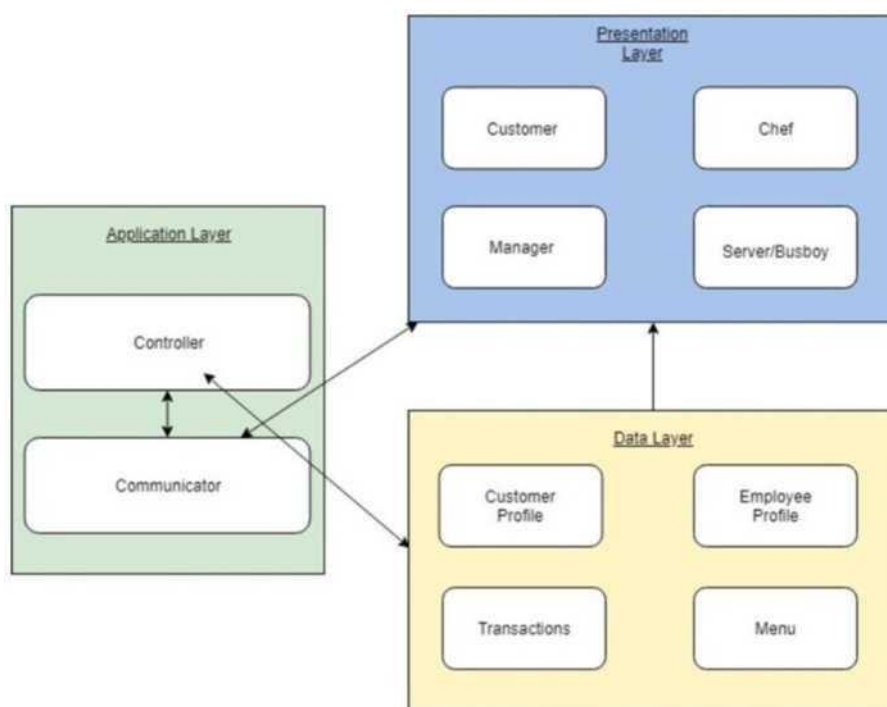


Рис. 3.9. Три шари архітектури додатку

Прикладний рівень є середнім шаром цієї архітектури, який забезпечує взаємодію між верхнім і нижнім шарами, фактично керуючи загальною роботою. Цей рівень відповідає за бізнес-логіку, тобто набір правил, необхідних для функціонування програми відповідно до настанов організації. На цьому рівні розташовані Комунікатор і Контролер.

Контролер спрощує виконання завдань між шарами, тоді як Комуникатор передає інформацію між ними. Найнижчий рівень – це рівень даних, який відповідає за збереження та вилучення даних. На цьому рівні зберігається інформація про профілі працівників і клієнтів, а також доступні транзакції меню і дані закладу харчової індустрії.

Кожен шар працює незалежно, що дозволяє ізолювати проблеми. Це означає, що зміни в одному шарі зазвичай не впливають на інші. Завдяки цьому архітектура спрощує тестування і діагностику проблем. У нашому програмному забезпеченні для закладу харчової індустрії буде клієнтська частина, яка працюватиме на планшетах та мобільних телефонах, а також бекенд-сервер, який міститиме REST API та базу даних. Додаток буде сумісний з багатьма різними пристроями.

Вимоги до технічного обладнання. Наше програмне забезпечення буде використовувати RESTful API для взаємодії з серверами. Клієнт ініціює запити POST і GET і отримує відповіді від сервера. Запити POST відправляють дані від клієнта на сервер, тоді як запити GET отримують корисну інформацію для клієнта.

Враховуємо, що буде фізичний сервер, яким ми керуватимемо використовуючи Node.js. Для сервера Node.js, який містить пакети Express і Sequelize, що сприяють взаємодії бази даних та сервера.

Express – це мінімалістична і гнучка структура для веб-додатків на Node.js, яка надає потужний набір можливостей для програмування веб- і мобільних систем. Express слугує для можливої маршрутизації та обробки запитів від користувача.

Sequelize – це ORM для Node.js, який дозволяє додатку працювати з об'єктами, що відображаються у реляційній базі даних. Він підтримує MySQL і має надійну підтримку транзакцій, реплікацію читання тощо..

3.4. Опис архітектури програмного забезпечення

Архітектура системи для автоматизації обліку та фільтрації даних клієнтів у ресторані надає загальну картину структури та функціональної логіки, визначає ключові компоненти і зв'язки між ними, а також забезпечує базу для реалізації програми. Опис архітектури виконується за допомогою уніфікованої мови моделювання (UML) згідно з методологією Rational Unified Process. Запропонована архітектура включає автоматизацію процесів прийому, зберігання, обробки та аналітичного аналізу даних. Прототип складається з двох основних робочих потоків для ефективної обробки різних типів даних: 1) Конвеєр обробки великих даних; 2) Конвеєр даних.

Конвеєр великих даних призначений для автоматизації переміщення значних обсягів різних типів даних із зовнішніх джерел до озера даних з метою подальшого аналізу. Крім перенесення даних, використовуються інформація з традиційних реляційних баз даних, баз даних NoSQL та статичних сховищ даних, таких як файли (jpg, mp3, mp4, json), для більш ефективного їх аналізу на наступних етапах.

Основні компоненти UML-специфікації додатку:

- Концептуальна модель: містить опис основних сценаріїв роботи додатку, варіантів використання та учасників.
- Логічна модель: включає опис класів і функціональних модулів додатку.
- Модель розміщення: містить вимоги до апаратного забезпечення користувача і опис розміщення компонентів додатку.
- Модель реалізації: включає опис фізичних артефактів додатку, таких як файли, виконувані файли, використовувані бібліотеки та зв'язки між ними.

Одне з найважливіших завдань перед розробником під час створення додатку для автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії – побудова правильної архітектури проєкту. Це фундамент, на якому базується весь процес розробки. Правильна архітектура допомагає уникнути проблем з повторенням коду, ускладненням підтримки, труднощами з впровадженням нових функцій, нечитабельністю коду та масштабованістю.

Основні критерії для правильної архітектури – це читабельність і чистота коду, можливість повторного використання, уникнення дублювання та інтеграція нових функцій без порушення існуючого коду.

Логічна модель. У рамках логічної моделі додатку для автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії описуються ключові функціональні модулі і класи. Модулі проєктованого додатку представлені в діаграмі класів (див. рис. 3.10).



Рис. 3.10. Діаграма класів додатку з автоматизації обліку та фільтрації даних клієнтів для закладу харчової індустрії

Опис модулів додатку з автоматизації обліку та фільтрації даних клієнтів для закладу харчової індустрії наведено в таб. 3.1.

Таблиця 3.1

Опис модулів додатку з автоматизації обліку та фільтрації даних клієнтів для закладу харчової індустрії

Найменування	Опис
Користувацький інтерфейс	Містить атрибути взаємодії користувача з програмою. Дозволяє вибрати параметри даних, реалізувати пошук і сортування.
Імпорт даних	Містить атрибут, що виконує імпорт даних з додатку finam. ua
Транслятор даних	Містить атрибут, що формує імпортовані дані в таблицю.
Графічний модуль	Містить атрибут, що виробляє побудову графіка і індикатора на основі імпортованих даних.

Модель розміщення. Модель розміщення додатку для автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії створюється діаграма розгортання, яка описує фізичне розташування компонентів. Діаграма розгортання представлена на рис. 3.11.

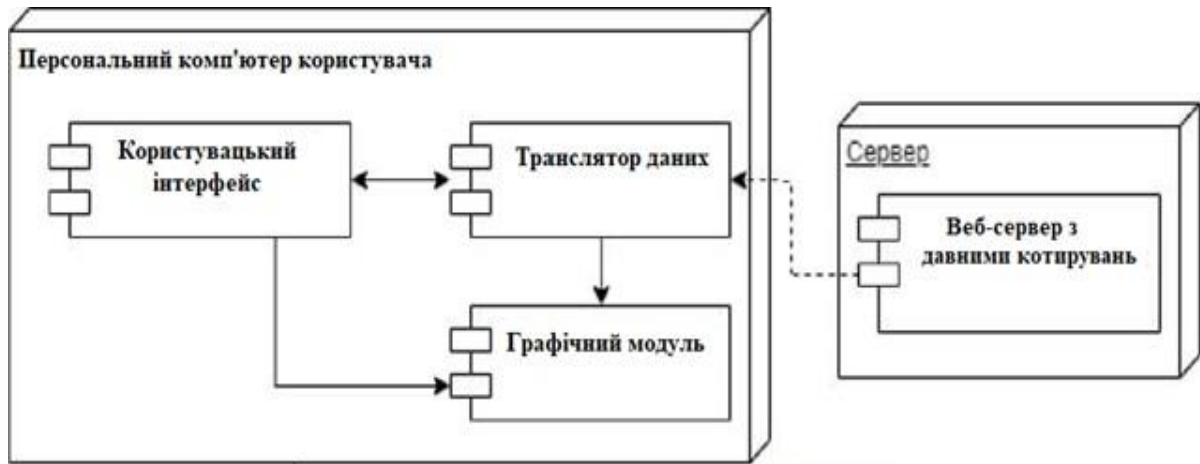


Рис. 3.11. Діаграма розгортання додатку з автоматизації обліку та фільтрації даних клієнтів для закладу харчової індустрії

Модель реалізації. Для безпосередньої реалізації додатку з автоматизації обліку та фільтрації даних клієнтів для закладу харчової індустрії потрібно визначати головні логічні і функціональні структури. Задля опису елементів додатку з автоматизації обліку та фільтрації даних клієнтів використовується діаграма компонентів.

Контекстна діаграма або діаграма нульового рівня для створюваного проекту зображена на рис. 3.12 є основою та базовою діаграмою для потоків даних.

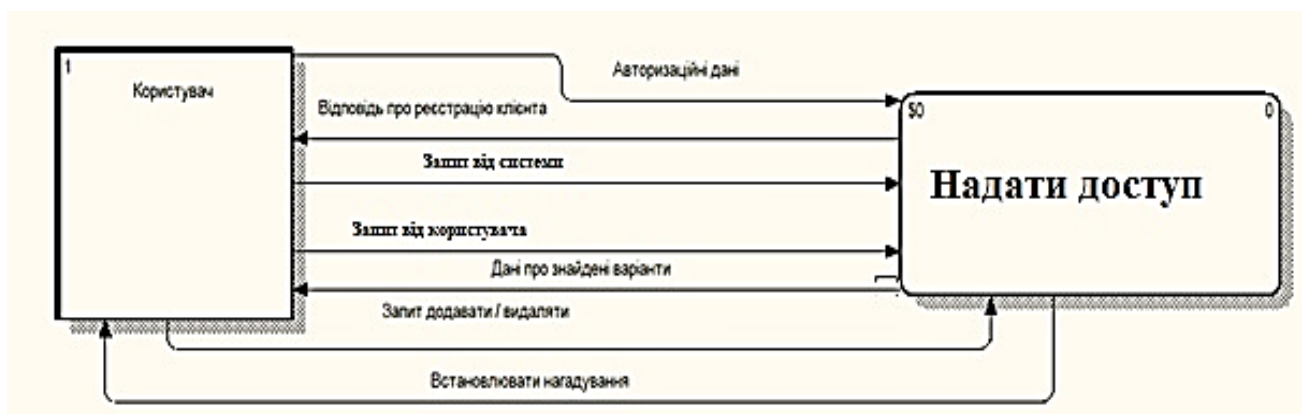


Рис. 3.12. Діаграма потоків даних нульового рівня

Під час розробки контекстної діаграми було визначено два елементи DFD для нульового рівня:

- Користувач. Це зовнішня сутність, що представляє користувача, який взаємодіє з додатком для автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії.
- Надати можливість доступу. Це основний процес, який виконується додатком і відповідає за обробку та потік даних. Процеси, потоки та бази даних, що сформовані, зображені на рисунку 3.13, включають:
 - створення облікового запису;
 - формування списку;
 - відправлення сформованого результату обробки;
 - бази даних: "Користувачі" та "База даних" та їх вимоги.

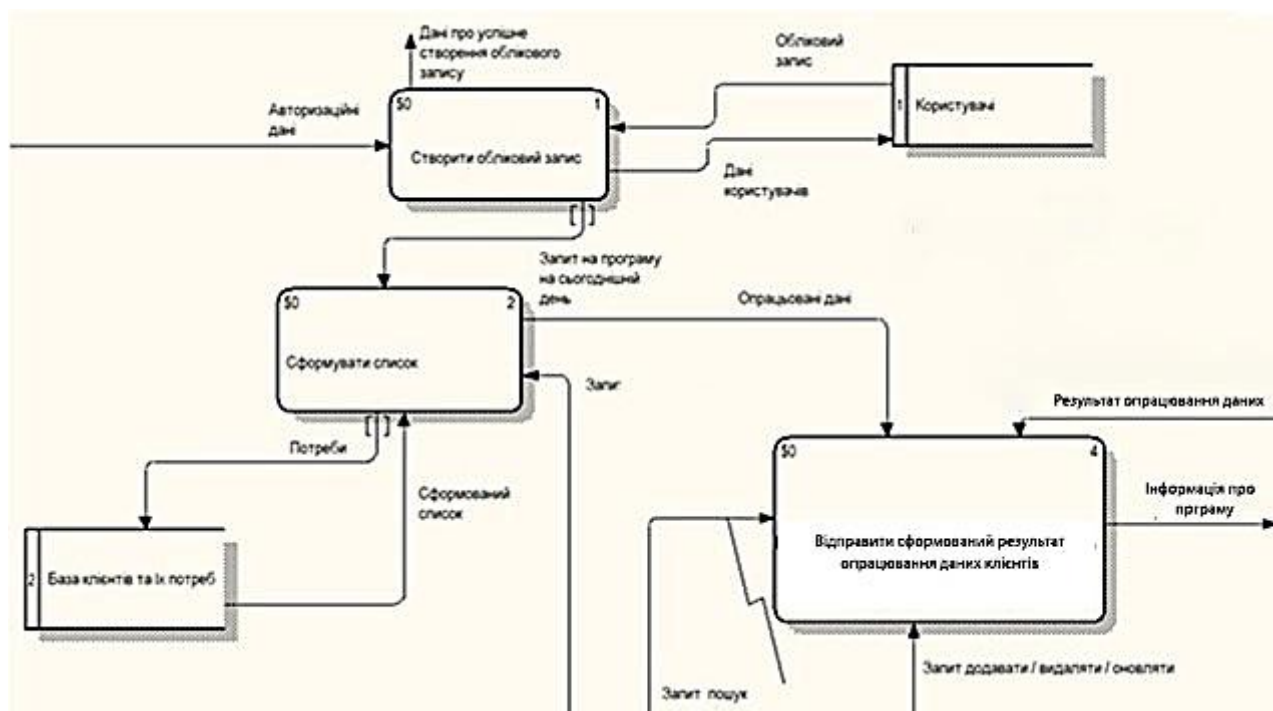


Рис. 3.13. Декомпозиція першого рівня додатку з автоматизації обліку та фільтрації даних клієнтів для закладу харчової індустрії

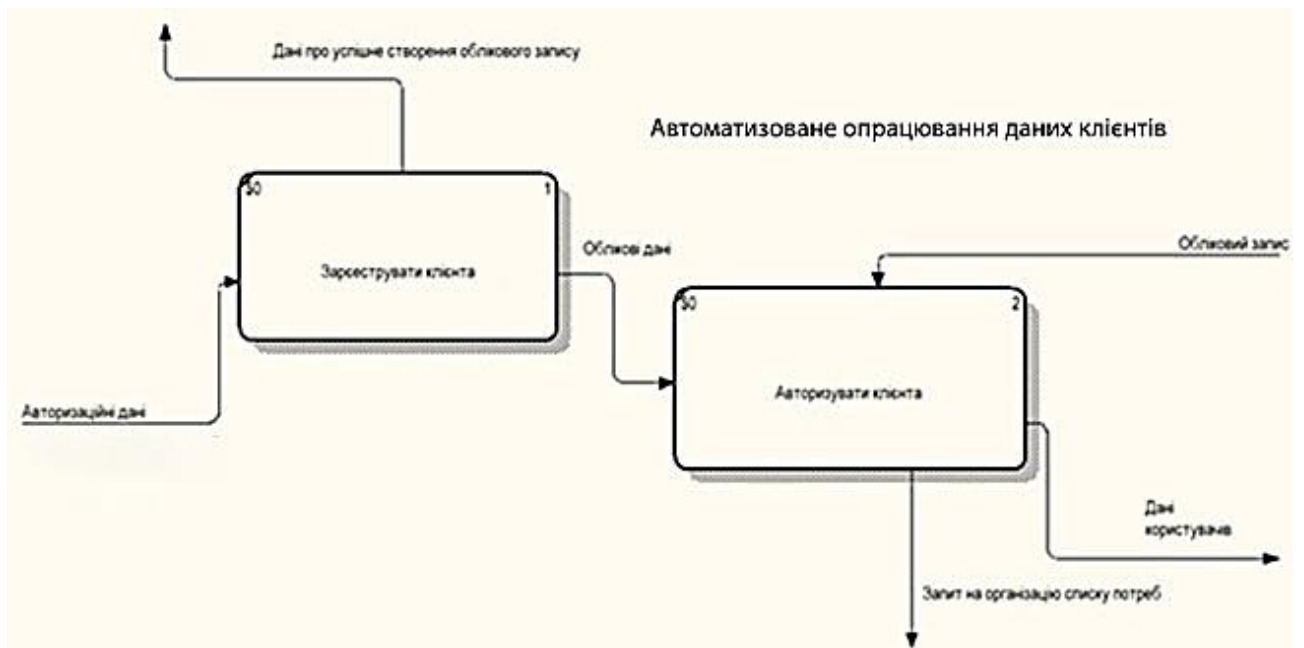


Рис. 3.14. Декомпозиція процесу «Створити обліковий запис» для додатку з автоматизації обліку та фільтрації даних

Наступним етапом є розбиття другого рівня процесу "Створення облікового запису", який включає наступні етапи:

- реєстрація клієнта;
- автентифікація клієнта.

Ці етапи пов'язані обліковими даними, які передаються при автентифікації користувача для обробки даних та надання доступу до програми.



Рис. 3.15. Декомпозиція процесу «Сформувати список автоматичної обробки даних клієнтів» для закладу харчової індустрії

Наступний процесом описано дії користувача в межі пошуку результатів прогнозу, зображено на рис. 3.15. Сформований запит на перегляд результатів даних клієнтів закладу індустрії харчування відправляється користувачем додатку та опрацьовується адміністратором. Додаток реагує на запит та формує список доступних даних для адміністратора закладу харчування.

Наступним складеним процесом є «Сформувати список даних клієнтів для закладу харчової індустрії прогнозу». Він складається із процесів, а саме: створити запит на завантаження даних; затвердити вибрані дані.

Ці процеси взаємодіють між собою, обробляючи облікові дані, які передаються під час авторизації користувача для доступу до даних та застосунків. На виході отримуємо опрацьовані дані та запит на перегляд даних..



Рис. 3.16. Декомпозиція процесу «Відправити сформований результат опрацювань»

Дана декомпозиція зображена на рис. 3.16 складається з двох процесів:

- вибрати шлях відображення даних для адміністратора закладу харчування;
- вибрати показ програми.

Ці процеси пов'язані між собою за допомогою програм, які вибирає користувач та оброблюються запитами. На вході отримується запит на пошук даних клієнтів закладу харчової індустрії за їхніми ініціалами. На виході ми отримуємо систематизовану інформацію.

Дерево цілей для додатку з автоматизації обліку та фільтрації даних клієнтів для закладу харчової індустрії будується шляхом переформулювання

проблем у цілі, створення взаємовідносин "засоби – результати" та обрання однієї чи кількох конкретних цілей. Алгоритм дерева рішень для цього додатку належить до сімейства алгоритмів навчання з наглядом. На відміну від інших алгоритмів навчання з наглядом, алгоритм дерева цілей можна використовувати для вирішення проблем регресії та класифікації.

Основна мета цього дослідження - створення додатку з автоматизації обліку та фільтрації даних клієнтів для закладу харчової індустрії. Для досягнення цієї мети необхідно виконати наступні цілі: аналіз предметної області, проектування додатку, реалізація програмної частини, вдосконалення додатку, підтримання системи. Запропоноване дерево цілей можна переглянути на рис. 3.17.



Рис. 3.17. Зовнішній вигляд дерева цілей згідно досліджуваної проблематики в роботі

В наступному етапі системного аналізу об'єкту дослідження проводиться складання матриці порівняння критеріїв з типами систем (таблиця 3.2). Ця

матриця дозволяє порівняти критерії з різними типами систем, щоб визначити їхню важливість у контексті конкретного об'єкту дослідження.

Таблиця 3.2

Ранжування критеріїв в дереві цілей для додатку з автоматизації обліку та фільтрації даних клієнтів в закладі харчової індустрії

№	Назва критерію	Порівняння критеріїв						Вектор пріоритету
		1	2	3	4	5	6	
1	Актуальність	1	0,25	0,33	5	0,25	2	0,15
2	Точність	1	3	7	7	1	1	0,26
3	Зручність	0,14	4	1	5	0,5	0,14	0,114
4	Коректність	2	1	4	0,17	3	1	0,17
5	Доступність	0,33	0,14	0,2	0,2	0,12	1	0,04
6	Якість	0,5	7	1	6	1	8	0,33

Проведене попарне порівняння критеріїв з типами систем відображає переваги та недоліки кожного типу системи для вирішення конкретної проблеми. Ці дані допоможуть обґрунтовано визначити, який тип системи краще відповідає поставленим вимогам та обмеженням проекту (таб. 3.3–3.9).

Таблиця 3.3

Матриця попарного порівняння за критерієм «Актуальність» в дереві цілей для додатку з автоматизації обліку та фільтрації даних клієнтів в закладі харчової індустрії

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-консультаційні	0,5	0,14	6	0,16	0,14	0,4
2	Інформаційно-керівні	6	0,33	5	0,5	0,5	0,2
3	Інформаційно-пошукові	0,2	0,16	0,5	0,5	0,16	0,3
4	Інтелектуальні інформаційні	4	0,33	5	1	0,5	0,5
5	СППР	5	3	5	0,5	1	0,45

Таблиця 3.4

Матриця попарного порівняння за критерієм «Точність» в дереві цілей для додатку з автоматизації обліку та фільтрації даних клієнтів в закладі харчової індустрії

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	

Продовження таблиці 3.4

1	Інформаційно-консультаційні	0,5	6	0,5	0,2	7	0,74
2	Інформаційно-керівні	0,5	3	0,5	0,5	6	0,28
3	Інформаційно-пошукові	0,2	0,14	0,2	0,33	5	0,22
4	Інтелектуальні інформаційні	0,5	3	3	0,12	0,33	0,27
5	СППР	0,16	0,5	0,5	0,2	0,5	0,21

Таблиця 3.5

Матриця порівняння для критерію «Зручність» в дереві цілей для додатку з автоматизації обліку та фільтрації даних клієнтів в закладі харчової індустрії

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-консультаційні	0,5	0,5	2	0,2	5	0,2
2	Інформаційно-керівні	4	0,5	5	0,5	3	0,5
3	Інформаційно-пошукові	0,2	0,16	0,2	0,5	7	0,4
4	Інтелектуальні інформаційні	0,14	3	5	2	4	0,38
5	СППР	3	0,33	0,33	0,5	4	0,17

Таблиця 3.6

Матриця порівняння для критерію «Коректність» дереві цілей для додатку з автоматизації обліку та фільтрації даних клієнтів в закладі харчової індустрії

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно- консультаційні	0,33	2	3	3	7	0,6
2	Інформаційно-керівні	0,33	4	0,2	4	5	0,24
3	Інформаційно-пошукові	0,5	3	0,33	3	0,5	0,36
4	Інтелектуальні інформаційні	0,5	1	0,5	0,5	3	0,22
5	СППР	0,5	3	4	0,2	0,5	0,28

Таблиця 3.7

Матриця порівняння для критерію «Доступність» в дереві цілей для додатку з автоматизації обліку та фільтрації даних клієнтів в закладі харчової індустрії

№	Назва системи	Порівняння систем					Вектор пріоритету
		0,5	3	4	3	5	
1	Інформаційно- консультаційні	3	0,5	½	5	0,33	0,24
2	Інформаційно-керівні	0,14	0,5	0,33	2	3	0,22
3	Інформаційно-пошукові	3	6	4	3	4	0,6
4	Інтелектуальні інформаційні	4	3	¼	3	0,33	0,6
5	СППР	3	2	0,33	0,5	1	0,49

Таблиця 3.8

Матриця порівняння для критерію «Якість» в дереві цілей для додатку з автоматизації обліку та фільтрації даних клієнтів в закладі харчової індустрії

№	Назва системи	Порівняння систем					Вектор пріоритету
		2	3	3	6	3	
1	Інформаційно-консультаційні	3	7	9	0,2	2	0,47

2	Інформаційно-керівні	5	3	0,12	3	4	0,33
3	Інформаційно-пошукові	0,5	0,14	0,33	0,5	3	0,24
4	Інтелектуальні інформаційні	0,11	4	0,33	0,14	0,5	0,26
5	СППР	0,14	6	4	0,5	6	0,44

Після аналізу дерева цілей, була складена таблиця, в якій зібрано коефіцієнти критеріїв для кожного типу системи. На основі цих даних було визначено кінцевий пріоритет додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії. Цей процес допоміг обґрунтувати вибір найбільш підходящого типу системи для вирішення поставленої задачі. у таб. 3.9.

Таблиця 3.9

Результати методу аналітичної ієрархії в дереві цілей для додатку з автоматизації обліку та фільтрації даних клієнтів в закладі харчової індустрії

Критерій	Актуальність	Точність	Зручність	Коректність	Доступність	Якість	Пріоритет
Тип систем							
Інформаційно-консультаційні	0,015	0,034	0,165	0,018	0,086	0,2566	0,49
Інформаційно-керівні	0,15	0,064	0,044	0,029	0,018	0,085	0,28
Інформаційно-пошукові	0,17	0,029	0,023	0,024	0,023	0,036	0,12
Інтелектуальні інформаційні	0,032	0,035	0,042	0,026	0,022	0,033	0,19
СППР	0,033	0,019	0,011	0,028	0,019	0,045	0,29

Результатом проведеного порівняння є вибір найбільше значення у інформаційно-консультаційному типу додатку. Тобто основним пріоритетним призначенням додатку з автоматизації обліку та фільтрації даних клієнтів в

закладі харчової індустрії є інформаційно-консультаційна спрямованість розробки.

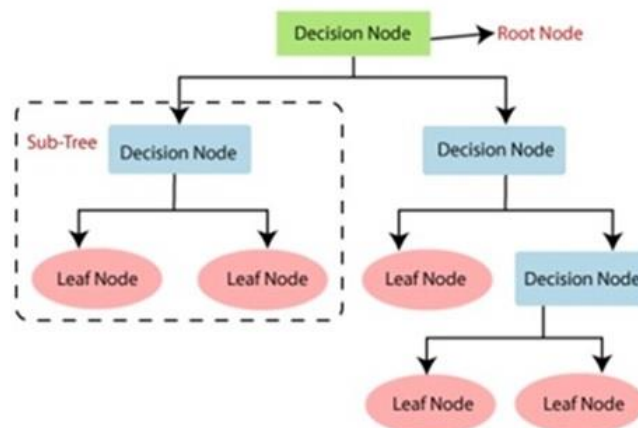


Рис. 3.18. Приклад роботи алгоритму дерева цілей для додатку з автоматизації обліку та фільтрації даних клієнтів в закладі харчової індустрії

Використання дерев рішень має на меті створення навчальної моделі, що дозволяє прогнозувати клас або значення цільової змінної. Вона досягається за допомогою вивчення простих правил прийняття рішень, які впливають з аналізу попередніх даних про фізичні можливості користувача при плануванні тренувань. У деревах рішень для передбачення мітки класу для запису процес розпочинається з кореня дерева, де значення кореневого атрибута порівнюється з атрибутом запису. На основі цього порівняння визначається гілка, яка відповідає цьому значенню, і переходить до наступного вузла. Типи дерев рішень базуються на типі цільової змінної, що ми хочемо прогнозувати.

Розроблено діаграму варіантів використання (Use Case) для мобільного додатку супроводження фітнес-тренувань з метою кращого розуміння його функціоналу. На цій діаграмі зображені актори, які взаємодіють з додатком, а також самі варіанти використання - сценарії, які показують, як актори користуються функціоналом додатку.



Рис. 3.19. Приклад роботи додатку з автоматизації даних клієнтів по офіціанту

«Фронт офіс» підтримує всі фінансові обліки, пристрої для друку рахунків, пристрої для зчитування магнітних карток та сканери штрих кодів.

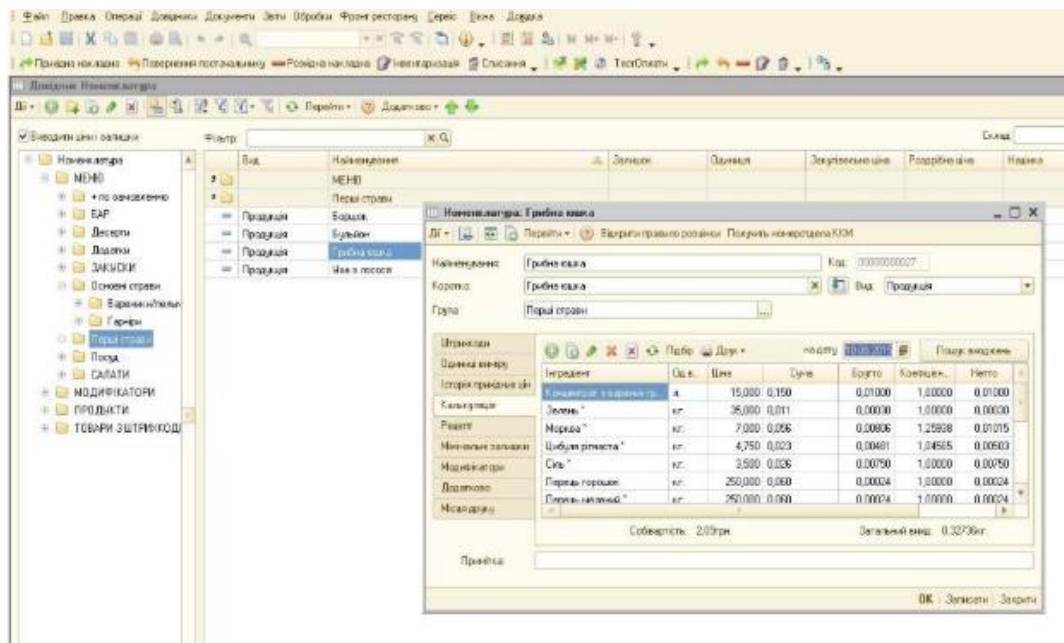


Рис. 3.20. Приклад роботи додатку з автоматизації даних клієнтів по типу замовлень в закладі харчування

Варіанти використання: – Огляд сторінок; – Змінення інформації учасника; – Переписка; – Тренування; – Пошук; – Участь у дискусіях; – Використання калькулятора індексу маси тіла; – Моніторинг прогресу учасника; – Перегляд статусу тренувань; – Реєстрація; – Редагування тренувань та меню; – Модерація.

Структура таблиці «Клієнти» в додатку з автоматизації даних клієнтів в закладі харчування представлена на рис. 3.21.

Имя столбца	Тип данных	Разрешить ...
ID_Client	int	<input type="checkbox"/>
Name	nchar(10)	<input checked="" type="checkbox"/>
Surname	nchar(10)	<input checked="" type="checkbox"/>
Gender	nchar(10)	<input checked="" type="checkbox"/>
Birthday	date	<input checked="" type="checkbox"/>
[E-mail]	nchar(10)	<input checked="" type="checkbox"/>
Adress	nchar(10)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Имя столбца	Тип данных	Разрешить ...
ID_Order	int	<input type="checkbox"/>
Name	nchar(10)	<input checked="" type="checkbox"/>
ID_Restaurant	int	<input type="checkbox"/>
Address	nchar(10)	<input checked="" type="checkbox"/>
Status_Order	nchar(10)	<input checked="" type="checkbox"/>
Date	date	<input checked="" type="checkbox"/>
Create_at	datetime	<input checked="" type="checkbox"/>
Count_people	int	<input checked="" type="checkbox"/>
ID_Client	int	<input type="checkbox"/>
		<input type="checkbox"/>

Рис. 3.21. Структура таблиці «Clients» та таблиці «Orders» в додатку з автоматизації даних клієнту

ВИСНОВКИ ДО РОЗДІЛУ 3

У третьому розділі представлено практичну реалізацію проекту автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії. Розроблено математичну модель автоматизації, визначено функціональні та нефункціональні вимоги, вибрано архітектуру серверного додатку та технології для обладнання, описано архітектуру програмного забезпечення та здійснено програмну реалізацію додатку. Проведені тестування підтвердили ефективність розробленого додатку в покращенні швидкості та якості обслуговування клієнтів, що дозволяє зробити висновок про успішність реалізації проекту та доцільність впровадження подібних систем в інших закладах харчової індустрії

ВИСНОВКИ

1. Проведено літературний пошук за темою дослідження, систематизовано інформацію та визначено напрямки подальшого дослідження. Наголошено на важливості використання інформаційних додатків для покращення обслуговування в закладах харчування. Розглянуто основне призначення баз даних в додатках автоматизації.

2. Зроблено обґрунтування засобів реалізації проєкту із створення додатку з автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії мовою Python». Проаналізовано існуючі аналоги та вказано їх недоліки. Для нашого проєкту передбачено способи покращення функціональності з швидшого обслуговування клієнтів. Проведено теоретичне планування черговості реалізації проєкту, вказано можливі взаємодії користувачів при вході в додаток та вказано способи використання автоматизації даних клієнтів для збільшення прибутковості закладу харчування.

3. Проведено досконалий процес проектування системи автоматизації обліку та фільтрації даних клієнтів

4. Мною було розроблено додаток, який допомагає співробітникам і відвідувачам закладу харчової індустрії взаємодіяти. Клієнтам допомагає скористатись меню, менеджерам закладу допомагає швидше отримати інформацію стососовно замовлення.

5. Проведено тестування та перевірено працездатність розробленого додатку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бавико О. Є. Синхронізація розвитку ринку інформаційно-комунікаційних технологій в Україні з глобальними трендами. *Маркетинг і менеджмент інновацій*. 2019. № 1. С. 272–282.
2. Галасюк К. А. Специфіка маркетингової діяльності підприємств ресторанного господарства. *Проблеми обліку, аудиту, аналізу та оподаткування в умовах глобалізації економіки*. Кривий Ріг, ДонНУЕТ. 2019. 512 с.
3. Гордієнко Д. О. Особливості впровадження CRM-систем у діяльність підприємств. *Облік, аналіз, аудит та оподаткування: сучасні концепції розвитку*. Зб. матеріалів VII Всеукраїнської науково-практичної студентської конференції; 11 травня 2021 р. К.: КНЕУ, 2021. С. 446–448.
4. Карпов В. В. Антропологічне осмислення архітектурної форми сучасності // *Архітектура та екологія: Матеріали XI Міжнародної науково-практичної конференції* (м. Київ, 16–18 листопада 2020 року). К.: НАУ, 2020. С. 5–8.
5. Марцінковська О., Легкий О. Організаційні аспекти впровадження CRM-систем у діяльність підприємства. *Регіональні аспекти розвитку продуктивних сил України*. 2019. Вип. 23. С. 81–85.
6. Міронова В. Л., Пирог М. В. Мікросервісна архітектура інтелектуальних транспортних систем інформаційно-аналітичного забезпечення. *Моделювання та інформаційні технології*. 2019. № 87. С. 36–40.
7. Назаренко І. А., Боднарук О. А. Організація обслуговування в закладах ресторанного господарства. Кривий Ріг: ДонНУЕТ, 2020. С. 128.
8. Обозна А. О., Шабельник Н. М., Федотов І. А. Аналіз сучасного ринку ресторанних послуг та ефективність їх застосування на підприємстві. *Інвестиції: практика та досвід*. 2021. № 13. С. 17–23.
9. Фісун М., Дворецький М., Дворецька С. Побудова моделей для оптимізації структури бази даних вузла у корпоративних інформаційних системах. *ІТКІ*, 2020. vol 48, № 2, С. 52–60.

10. J. Zhang, «Python Based Data Visualization and Configurable Teaching System Design and Implementation," *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, Dalian, China, 2021, pp. 1136–1140, doi: 10.1109 / IPEC51340.2021.9421127. Дата звернення: 30.05.24.
11. M. Liu, Y. Du and X. Xu, «Customer Value Analysis Based on Python Crawler," *2019 Chinese Control And Decision Conference (CCDC)*, Nanchang, China, 2019, pp. 4345–4350, doi: 10.1109 / CCDC. 2019.8832805. Дата звернення: 30.05.24.
12. Trofimova Natalya, Ermolaeva, Evgenia, Trofimov, Ivan. Development of a Software Product for the Automation of Hazard Analysis and Critical Control Points in Food Production. *Food Processing: Techniques and Technology*. 2020. 50. 167–175. 10.21603 / 2074–9414–2020–1-167–175. Дата звернення: 30.05.24.
13. Yi, N. Y., & Choi, J. H. (2019). A study of the consumer perception of meal kit using big data analysis. *Food Service Industry Journal*, 15 (4), 211–222. Дата звернення: 30.05.24.
14. Utilizing software design patterns in product-driven manufacturing system: A case study / D. Drozdov, U. D. Atmojo, C. Pang [et al.] // *Studies in Computational Intelligence*. 2020. Vol. 853. P. 301–312. DOI: https://doi.org/10.1007/978-3-030-27477-1_23. Дата звернення: 30.05.24.
15. Zeller, A. Functional verification of distributed automation systems: Assisting production line operators by an automated model composition / A. Zeller, N. Jazdi, M. Weyrich // *International Journal of Advanced Manufacturing Technology*. 2019. Vol. 105, № 9. P. 3991–4004. DOI: <https://doi.org/10.1007/s00170-019-03791-2>. Дата звернення: 30.05.24.

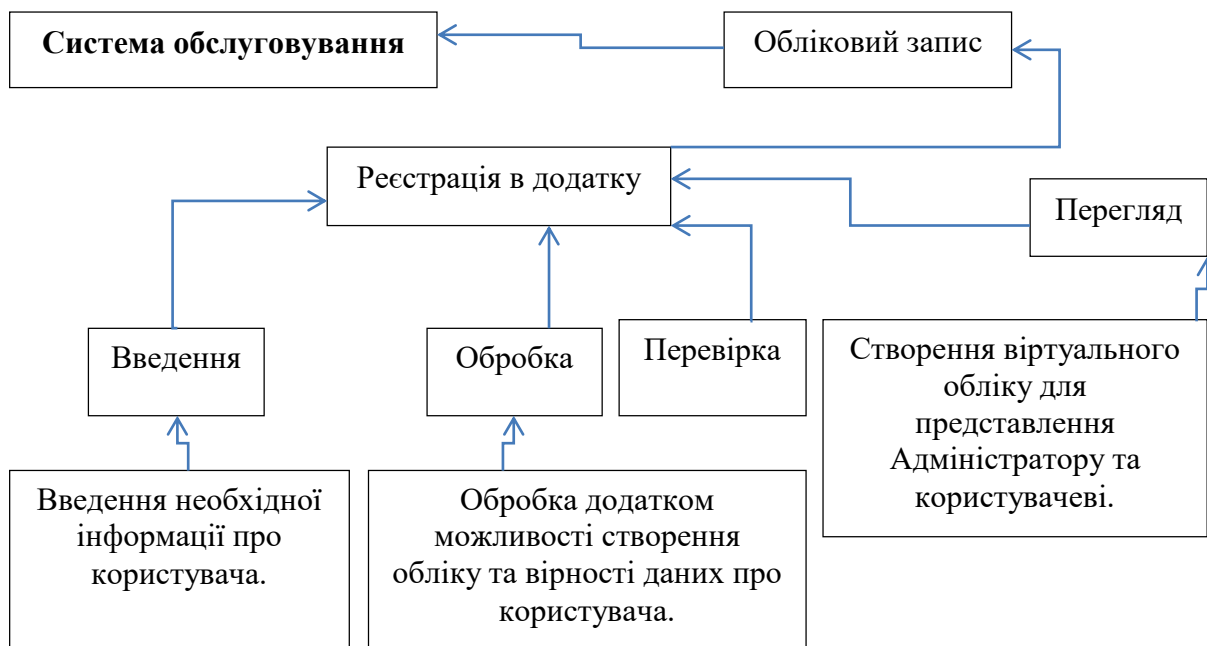
ДОДАТКИ

Додаток А

Архітектура web-додатку автоматизації обліку та фільтрації даних клієнтів у закладі харчової індустрії мовою Python



Структура створення облікового запису на проектованому сервісі



Уточнена USE CASE діаграма ІС надання послуг у закладі харчування



```

my_flask_app/
|
├── static/
|   ├── css/
|   |   ├── reset. css
|   |   ├── bootstrap-grid. css
|   |   └── style. css
|   ├── img/
|   |   ├── favicon. png
|   |   └── icons-sprite. svg
|   └── js/
|       └── script. js
└── templates/
    └── index. html

```

```

python
from flask import Flask, render_template
app = Flask(__name__)
@app.route('/')
def home():
    return render_template('index. html')
if __name__ == '__main__':
    app.run (debug=True)

```

Створюємо templates / index. html з наступним вмістом:

```

<!DOCTYPE html>
<html lang=«en»>

<head>
<meta charset=«UTF – 8»>
<meta name=«viewport» content=«width=device-width, initial-scale=1.0»>
<title>FoodSpace</title>
<link rel=«stylesheet» href="{{ url_for('static', filename='css / reset. css') }}">
<link rel=«stylesheet» href="{{ url_for('static', filename='css / bootstrap-grid. css') }}">
<link rel=«stylesheet» href="{{ url_for('static', filename='css / style. css') }}">
<link rel=«icon» type=«image / png» href="{{ url_for('static', filename='img / favicon. png')
}}">
</head>
<body>
<section class=«main»>

```

```

<header class=«main__header» id=«main__header»>
<div class=«container»>
<nav class=«main__navigation»>
<a href="#" class=«main__logo»>FoodSpace</a>
<ul class=«main__menu»>
<li class=«menu__item»><a href="#" class=«menu__link menu__link-
active»>Головна</a></li>
<li class=«menu__item»><a href="#" class=«menu__link»>Пронас</a></li>
<li class=«menu__item»><a href="#" class=«menu__link»>Меню</a></li>
<li class=«menu__item»><a href="#" class=«menu__link»>Галерея</a></li>
<li class=«menu__item»><a href="#" class=«menu__link»>Наш блог</a></li>
<li class=«menu__item»><a href="#" class=«menu__link»>Контакти</a></li>
<li class=«menu__item»><a href="#" class=«menu__link»>Корзина</a></li>
</ul>
</nav>
<div class=«row»>
<div class=«col – 4 info-box»>
<h1 class=«info-box__title»>Fresh Life</h1>
<p class=«info-box__desc»>У нашому закладі харчової індустрії використовуються
продукти тільки найкращої якості, а наші страви відповідають усім стандартам.</p>
<a href="#" class=«info-box__button»>Ознайомитись</a>
</div>
<a href="#" class=«scroll-down»>
<svg class=«icon»>
<use xlink:href="{ { url_for('static', filename='img / icons-sprite. svg#scroll') } }"></use>
</svg>
</a>
</div>
</div>
</header>
</section>
<section class=«about» id=«about»>
<div class=«container»>
<div class=«row about__row»>
<div class=«offset – 7 col – 5 about__block»>
<h1 class=«about__title»>Пронас</h1>
<p class=«about__text»>Затишна, доброзичлива атмосфера та гідний сервіс – основні
переваги нашого закладу харчування. Мікс цих характеристик і доступний рівень цін
дозволяють нашому закладі харчової індустрії бути в топ і найкращих закладів.</p>
<p class=«about__text»>Спробуйте нас на смак!</p>
</div>
</div>
</div>
</section>

<section class=«services» id=«services»>
<div class=«container»>
<div class=«row services__row»>
<div class=«col – 6 services__block»>
<button href=«menu. html» class=«services__button»>

```

```

<h2 class=«services__button-title»><a href=«menu» class=«next__pageservices__button-
title»>Перейти до меню</a></h2>
<div class=«services__icons»>
<svg>
<use xlink:href="{ { url_for('static', filename='img / icons-sprite. svg#basket') } }"></use>
</svg>
</div>
</button>
</div>
<h1 class=«services__title»>Наше меню</h1>
</div>
</div>
</section>
</body>

</html>
<!DOCTYPE html>
<html lang=«en»>

<head>
<meta charset=«UTF - 8»>
<meta name=«viewport» content=«width=device-width, initial-scale=1.0»>
<title>FoodSpace</title>
<link rel=«stylesheet» href="{ { url_for('static', filename='css / reset. css') } }">
<link rel=«stylesheet» href="{ { url_for('static', filename='css / bootstrap-grid. css') } }">
<link rel=«stylesheet» href="{ { url_for('static', filename='css / style. css') } }">
<link rel=«icon» type=«image / png» href="{ { url_for('static', filename='img / favicon. png')
}}">
</head>

<body>
<!-- Main Section -->
<section class=«main»>
<header class=«main__header» id=«main__header»>
<div class=«container»>
<nav class=«main__navigation»>
<a href="#" class=«main__logo»>FoodSpace</a>
<ul class=«main__menu»>
<li class=«menu__item»><a href="#header» class=«menu__link menu__link-
active»>Головна</a></li>
<li class=«menu__item»><a href="#about» class=«menu__link»>Про нас</a></li>
<li class=«menu__item»><a href="#services» class=«menu__link»>Меню</a></li>
<li class=«menu__item»><a href="#gallery» class=«menu__link»>Галерея</a></li>
<li class=«menu__item»><a href="#blog» class=«menu__link»>Наш блог</a></li>
<li class=«menu__item»><a href="#contact» class=«menu__link»>Контакти</a></li>
<li class=«menu__item»><a href=«cart» class=«menu__link»>Корзина</a></li>
</ul>
</nav>
<div class=«row»>
<div class=«col - 4 info-box»>
<h1 class=«info-box__title»>Fresh Life</h1>

```

```

<p class=«info-box__desc»>У нашому закладі харчової індустрії використовуються
продукти тільки найкращої якості, а наші страви відповідають усім стандартам.</p>
<a href="#" class=«info-box__button»>Ознайомитись</a>
</div>
<a href="#about» class=«scroll-down»>
<svg class=«icon»>
<use xlink: href="{{ url_for('static', filename='img / icons-sprite. svg#scroll') }}"></use>
</svg>
</a>
</div>
</div>
</header>
</section>

<! – About Section –>
<section class=«about» id=«about»>
<div class=«container»>
<div class=«row about__row»>
<div class=«offset – 7 col – 5 about__block»>
<h1 class=«about__title»>Про нас</h1>
<p class=«about__text»>Затишна, доброзичлива атмосфера та гідний сервіс – основні
переваги нашого закладу. Мікс цих характеристик і доступний рівень цін дозволяють нашому
закладу харчування бути в топі найкращих закладів Полтави.</p>
<p class=«about__text»>Спробуйте нас на смак!</p>
</div>
</div>
</div>
</section>

<! – Services Section –>
<section class=«services» id=«services»>
<div class=«container»>
<div class=«row services__row»>
<div class=«col – 6 services__block»>
<button href=«menu. html» class=«services__button»>
<h2 class=«services__button-title»><a href=«menu» class=«next__page services__button-
title»>Перейти до меню</a></h2>
<div class=«services__icons»>
<svg>
<use xlink: href="{{ url_for('static', filename='img / icons-sprite. svg#basket') }}"></use>
</svg>
</div>
</button>
</div>
</div>
<h1 class=«services__title»>Наше меню</h1>
</div>
</div>
</section>

<! – Gallery Section –>
<section class=«gallery» id=«gallery»>
<div class=«container»>

```

```

<div class=«gallery__text»>
  <h1 class=«gallery__title»>Галерея</h1>
  <p class=«gallery__desc»>Ми зібрали галерею найестетичніших фотографій від наших
співробітників.</p>
</div>
<div class=«row gallery__row»>
  <div class=«col – 8»>
    
  </div>
  <div class=«col – 4»>
    
    
  </div>
</div>
<div class=«row gallery__row»>
  <div class=«col – 6»>
    
  </div>
  <div class=«col – 6»>
    
  </div>
</div>
<div class=«gallery__more»>
  <button class=«gallery__button»>Подивитися більше</button>
</div>
</div>
</section>
<!-- Subscribe Section -->
<section class=«subscribe» id=«subscribe»>
  <div class=«container»>
    <div class=«row subscribe__row»>
      <div class=«col – 7 subscribe__block»>
        <h1 class=«subscribe__title»>Підписуйся!</h1>
        <p class=«subscribe__desc»>Підпишись на нас, аби отримувати повідомлення про акції
та події. Просто залиш свій e-mail.</p>
        <form action="#" class=«subscribe__form»>
          <input type=«email» class=«subscribe__form-input» placeholder=«Your email address...»>
          <input type=«submit» class=«subscribe__form-button» value=«Subscribe»>
        </form>
      </div>
    </div>
  </div>
</section>
<!-- Blog Section -->
<section class=«blog» id=«blog»>
<section class=«blog» id=«blog»>

```

```

<header class=«blog__header»>
<h1>Наш блог</h1>
<p>Отримай свіжу порцію новин про світ страв у нашому блозі!</p>
</header>
<div class=«blog__content»>
<div class=«blog__articles»>
<div class=«blog__post»>
<article class=«post__article»>
<a class=«post__link» href="#">
<div>

</div>
<div class=«post__header»>
<div class=«post__date»>
<span>20</span>
<span>травня</span>
</div>
<h2 class=«post__header-title»>Зелені смузи: надлишок корисних елементів?</h2>
<div class=«post__header-author»>
<p>Тетяна Самойлова, шеф-кухар</p>
</div>
</div>
</a>
</article>
</div>
<div class=«blog__post»>
<article class=«post__article»>
<a class=«post__link» href="#">
<div>

</div>
<div class=«post__header»>
<div class=«post__date»>
<span>14</span>
<span>травня</span>
</div>
<h2 class=«post__header-title»>Вирощування їжі та «спрага» до інновацій</h2>
<div class=«post__header-author»>
<p>Георгій Прохоров, директор</p>
</div>
</div>
</a>
</article>
</div>
<div class=«blog__post»>
<article class=«post__article»>
<a class=«post__link» href="#">
<div>

```