

БАКАЛАВРСЬКА РОБОТА

ДРБ. ІІІ - 14.00.00.000 ІІІ

Група ІІІ-21-1

Перцович Владислав

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Перцович Владислав Іванович

(прізвище, ім'я, по батькові)

УДК 004
(індекс)

БАКАЛАВРСЬКА РОБОТА

Автоматизація та інтеграція засобів і інструментів віртуального

порталу типу "FoodMarket"

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач освітнього рівня Перцович В.І.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Юрчишин Володимир Миколайович, д.т.н., професор
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою ІІЗ

доц.

В.В. Бандура

“ ” 2025 р.

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Перцовичу Владиславу Івановичу

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) “ Автоматизація та інтеграція засобів і інструментів віртуального порталу типу "FoodMarket"”

керівник проекту (роботи) Юрчишин Володимир Миколайович, професор

затвержені наказом закладу вищої освіти від “ 28 ” квітня 2025 р. № 264/7

2. Строк подання студентом проекту (роботи) 10 червня 2025 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз предметної області розробки інформаційних систем замовлення їжі

2. Проектування архітектури та системні вимоги віртуального порталу типу "FoodMarket"

3. Представлення системи з використанням UML

4. Реалізація інструментів віртуального порталу типу "FoodMarket" замовлення їжі

5. Реалізація інтерфейсу системи замовлення їжі

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Архітектура системи онлайн замовлення їжі (рис. 1.1)

2. Основні екрани додатку автоматизації роботи закладу харчування (рис. 1.2)

3. Екрани відстеження замовлення (рис. 1.3)

4. Функціонал системи замовлення їжі (рис. 1.4)

5. Записи про витрати клієнта (рис. 1.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 28 квітня 2025 р.

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області розробки інформаційних систем замовлення їжі	07.05.2025	виконано
2	Проектування архітектури та системні вимоги віртуального порталу типу "FoodMarket"	17.05.2025	виконано
3	Представлення системи з використанням UML	27.05.2025	виконано
4	Реалізація інструментів віртуального порталу типу "FoodMarket" замовлення їжі	01.06.2025	виконано
5	Реалізація інтерфейсу системи замовлення їжі	06.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник _____

(підпис)

Керівник роботи _____

(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 76 сторінок, 40 рисунків, список використаних джерел із 43 найменуваннями, 1 додаток.

Метою дипломної роботи є проектування та реалізація віртуального порталу типу "FoodMarket", який забезпечує повний цикл обробки онлайн-замовлень їжі — від формування замовлення користувачем до його обробки адміністратором закладу.

Об'єкт дослідження - інформаційні системи замовлення їжі та автоматизації процесів у ресторанному бізнесі.

Предмет дослідження - методи та засоби проектування та реалізації інтерактивного віртуального порталу онлайн-замовлення їжі.

В першому розділі виконано аналіз сучасних платформ виявив ключові функціональні обмеження та дозволив сформулювати вимоги до покращеної інформаційної системи

В другому розділі на основі вимог розроблено структурну та функціональну архітектуру системи із застосуванням UML-моделей та обґрунтуванням вибору інструментів розробки

В третьому розділі реалізована Система з використанням Python, Django та MySQL, забезпечивши функціональність інтерфейсів замовлення, адміністрування та обробки бази даних

Висновок: здійснено створення інформаційної системи типу "FoodMarket" для автоматизації онлайн-замовлень їжі та адміністрування діяльності ресторанів. Результати дослідження можуть бути використані для впровадження в малих та середніх підприємствах ресторанного бізнесу.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, FOODMARKET, ЗАМОВЛЕННЯ ЇЖІ, АВТОМАТИЗАЦІЯ РЕСТОРАНУ, ОНЛАЙН-СЕРВІС, DJANGO, БАЗА ДАНИХ, PYTHON, ІНТЕРФЕЙС КОРИСТУВАЧА, UML-ДІАГРАМИ.

ANNOTATION

The bachelor's thesis contains 76 pages, 40 figures, a list of used sources with 43 names, 1 appendix.

The purpose of the thesis is to design and implement a virtual portal of the "FoodMarket" type, which provides a full cycle of processing online food orders - from the formation of an order by the user to its processing by the administrator of the establishment.

The object of research is information systems for ordering food and automating processes in the restaurant business.

The subject of research is methods and means of designing and implementing an interactive virtual portal for online food ordering.

In the first section, an analysis of modern platforms was performed, key functional limitations were identified and requirements for an improved information system were formulated.

In the second section, based on the requirements, a structural and functional architecture of the system was developed using UML models and a justification for the choice of development tools.

In the third section, the system was implemented using Python, Django and MySQL, providing functionality for ordering, administration and database processing interfaces.

Conclusion: an information system of the "FoodMarket" type was created to automate online food orders and administer restaurant operations. The results of the study can be used for implementation in small and medium-sized enterprises in the restaurant business..

KEYWORDS: INFORMATION SYSTEM, FOODMARKET, FOOD ORDERING, RESTAURANT AUTOMATION, ONLINE SERVICE, DJANGO, DATABASE, PYTHON, USER INTERFACE, UML DIAGRAMS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	8
ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ ЗАМОВЛЕННЯ ЇЖІ ТА АВТОМАТИЗАЦІЇ РЕСТОРАННОГО БІЗНЕСУ	12
1.1. Особливості розробки віртуального порталу типу "FoodMarket"	12
1.2. Мета і мотиваційні чинники розробки віртуального порталу.....	13
1.3. Аналіз існуючих досліджень і платформ онлайн замовлення їжі	16
1.4. Функціональні можливості та технологічні обмеження існуючих систем замовлення страв.....	29
РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА СИСТЕМНІ ВИМОГИ ВІРТУАЛЬНОГО ПОРТАЛУ ТИПУ "FOODMARKET"	31
2.1. Вибір та обґрунтування інструментів для розробки	31
2.1.1. Фреймворк Django	31
2.1.2. База даних MySQL	31
2.1.3 Інструмент візуалізації SQLyog	32
2.1.4 Мова Python	33
2.1.5. ХАМРР	34
2.1.6. Середовище Visual Studio Code	35
2.1.7. Програмні вимоги та апаратні вимоги	35
2.2. Проектування архітектури системи	36
2.3. Розробка структури системи.....	39
2.4. Представлення системи з використанням UML	42

					БР.ІІ – 14.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Перцович В.І.			Автоматизація та інтеграція засобів і інструментів віртуального порталу типу "FoodMarket" Пояснювальна записка	Літ.	Арк.	Акрушіє
Перевір.		Юрчишин В.М.					6	
Реценз.						ІФНТУНГ Ш-21-1		
Н. Контр.		Піх М.М.						
Затверд.		Бандура В.В.						

2.4.1. Представлення випадків використання.....	42
2.4.2. Представлення класів.....	44
2.4.3. Представлення послідовності	47
2.4.4. Представлення діяльності	50
2.4.5. Представлення ER-моделі	51

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНСТРУМЕНТІВ ВІРТУАЛЬНОГО ПОРТАЛУ

ТИПУ "FOODMARKET" ЗАМОВЛЕННЯ ЇЖІ.....	54
3.1. Налаштування та конфігурація проекту.....	54
3.2. Представлення роботи з базою даних	58
3.3. Реалізація інтерфейсу системи замовлення їжі	61
3.3.1. Інтерфейс адміністратора	61
3.3.2. Інтерфейс користувача (клієнта)	63

ВИСНОВКИ.....	70
---------------	----

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	72
---------------------------------------	----

ДОДАТКИ

БІБЛІОГРАФІЧНА ДОВІДКА

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

IS – Information System – інформаційна система

UI – User Interface – інтерфейс користувача

DB – Database – база даних

ERD – Entity-Relationship Diagram – діаграма сутність-зв’язок

UML – Unified Modeling Language – уніфікована мова моделювання

MVC – Model-View-Controller – архітектурний шаблон “модель-вид-контролер”

IDE – Integrated Development Environment – інтегроване середовище розробки

OSS – Open Source Software – програмне забезпечення з відкритим кодом

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасних умовах діджиталізації бізнес-процесів та зростання попиту на онлайн-сервіси, автоматизація замовлення їжі набула особливої актуальності. Ресторанний бізнес, зокрема малі та середні заклади громадського харчування, потребує впровадження інноваційних рішень, які дозволяють оптимізувати роботу з клієнтами, спростити управління замовленнями та підвищити ефективність обслуговування. Існуючі платформи онлайн-замовлень їжі не завжди задовольняють вимоги користувачів щодо функціональності, доступності та гнучкості, що зумовлює потребу у створенні спеціалізованих віртуальних порталів. Отже, розробка інформаційної системи типу "FoodMarket" є актуальним завданням, яке відповідає потребам сучасного цифрового середовища та сприяє розвитку ІТ-рішень у сфері онлайн ресторанного бізнесу.

Актуальність роботи

Сучасний ринок громадського харчування стрімко змінюється під впливом цифрових технологій, що створює потребу в ефективних системах онлайн-замовлень та автоматизації ресторанного бізнесу. Особливо актуально це стало в умовах зростання попиту на доставку їжі, потреби в оптимізації внутрішніх процесів та забезпеченні зручності для клієнтів. Віртуальні портали типу "FoodMarket" дозволяють інтегрувати функціональність онлайн-замовлення, адміністрування меню, обліку замовлень та комунікації з клієнтами. Розробка таких інформаційних систем є важливою складовою цифрової трансформації галузі, що зумовлює актуальність даної роботи.

Сучасний етап розвитку інформаційних технологій характеризується активною цифровізацією бізнес-процесів у різних галузях, зокрема у сфері громадського харчування. Поширення мобільних додатків, веб-платформ та електронної комерції спричинило трансформацію підходів до взаємодії між

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

клієнтами та сервісами. Однією з найдинамічніших тенденцій останніх років стало широке впровадження онлайн-сервісів замовлення їжі, які дозволяють користувачам зручно і швидко здійснювати замовлення з будь-якого місця, а закладам харчування — оптимізувати процес обслуговування та розширити свою клієнтську базу.

У той же час, наявні програмні рішення на ринку часто не відповідають повною мірою потребам невеликих підприємств ресторанного бізнесу. Зокрема, обмеження у функціональності, складність налаштування, відсутність адаптації до конкретних моделей бізнесу та висока вартість впровадження перешкоджають ефективному використанню таких систем. Це створює потребу в розробці спеціалізованих рішень, які будуть зручними, доступними та технологічно гнучкими для малого та середнього бізнесу.

Метою даної дипломної роботи є проектування та реалізація віртуального порталу типу "FoodMarket", який забезпечує повний цикл обробки онлайн-замовлень їжі — від формування замовлення користувачем до його обробки адміністратором закладу.

Завдання дослідження

- Провести аналіз предметної області та існуючих систем онлайн-замовлень їжі.
- Визначити вимоги до функціональності та архітектури системи.
- Обґрунтувати вибір інструментів та технологій розробки.
- Створити UML-моделі майбутньої системи.
- Реалізувати інтерфейси адміністратора та користувача.
- Розробити інтегровану базу даних для обробки замовлень.
- Провести тестування та перевірку працездатності системи.

Об'єкт дослідження - інформаційні системи замовлення їжі та автоматизації процесів у ресторанному бізнесі.

Предмет дослідження - методи та засоби проектування та реалізації інтерактивного віртуального порталу онлайн-замовлення їжі.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Методи дослідження

Аналіз літературних джерел, методи структурного та об'єктно-орієнтованого проектування (UML), програмна реалізація на базі фреймворку Django, моделювання баз даних, тестування функціональності системи.

Наукова новизна

Розроблено модель інтерактивного порталу "FoodMarket", що поєднує сучасні технології веб-розробки з можливістю повної автоматизації процесу замовлення та обробки страв у реальному часі, із врахуванням специфіки ресторанного бізнесу.

Практичне застосування

Результати роботи можуть бути використані для впровадження в ресторанах, кафе, службах доставки їжі, а також адаптовані до інших сфер онлайн-комерції, де потрібна автоматизація замовлень та управління сервісами.

Таким чином, дана робота є вагомим внеском у напрямі створення інноваційних програмних засобів для сфери ресторанного бізнесу, що мають прикладне значення та перспективи подальшого розвитку.

Бакалаврська робота містить 76 сторінок, 40 рисунків, 3 розділи список використаних джерел із 43 найменуваннями, 1 додаток.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ ЗАМОВЛЕННЯ ЇЖИ ТА АВТОМАТИЗАЦІЇ РЕСТОРАННОГО БІЗНЕСУ

1.1. Особливості розробки віртуального порталу типу "FoodMarket"

Імплементація віртуальних порталів типу "FoodMarket" спричинила суттєві трансформації у парадигмі взаємодії між споживачами та суб'єктами господарювання у сфері ресторанного бізнесу. Ці спеціалізовані цифрові платформи забезпечують можливість здійснення повного циклу замовлення та фінансових транзакцій без необхідності прямого фізичного контакту з персоналом.

Структура функціоналу таких порталів, як правило, передбачає інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам здійснювати інтерактивний перегляд асортименту страв (цифрове меню), виконувати деталізовану персоналізацію обраних позицій відповідно до індивідуальних вподобань та оперативно формувати замовлення.

Доступ до функціоналу порталу може бути ініційовано різними методами, зокрема, шляхом сканування унікального QR-коду, асоційованого з конкретним закладом або столиком. Це також може інтегрувати функцію резервування столика. В такому випадку, система автоматично верифікує наявність вільних місць на обраний час та, у разі позитивного результату перевірки, здійснює автоматичне бронювання. При відсутності доступних столів, алгоритм системи може запропонувати інтеграцію користувача у цифрову чергу очікування, сповіщаючи про звільнення місця.

Після остаточного формування переліку бажаних страв, користувач отримує можливість здійснити безпечну фінансову транзакцію безпосередньо через інтерфейс порталу, валідуючи суму відповідно до

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

сформованого замовлення. Інтегровані платіжні шлюзи забезпечують конфіденційність та захист даних.

Логістика обслуговування у рамках цієї моделі передбачає, що підготовлені кулінарні вироби доставляються безпосередньо до локації розміщення клієнта (столика) персоналом закладу, мінімізуючи потребу в додаткових контактах на етапі видачі замовлення.

Таким чином, застосування віртуальних порталів типу "FoodMarket" представляє собою значний крок в оптимізації операційних процесів у сегменті харчування і доставки їжі, підвищуючи ефективність обслуговування, знижуючи ймовірність помилок при прийомі замовлень та покращуючи загальний користувацький досвід за рахунок швидкості, зручності та контрольованості процесу для споживача. Це технологічне рішення демонструє потенціал для подальшого розвитку та інтеграції з іншими системами автоматизації ресторанного бізнесу.

1.2. Мета і мотиваційні чинники розробки віртуального порталу

Застосування спеціалізованих цифрових платформ, що активуються, зокрема, за допомогою QR-кодів, надає користувачам можливість здійснювати безперешкодний доступ до цифрового меню, виконувати параметризацію замовлень та реалізовувати безпечні фінансові транзакції за допомогою портативних електронних пристроїв, таких як смартфони. Ці цифрові інтерфейси забезпечують ергономічний користувацький досвід та спрощують процес ознайомлення з асортиментом та сервісами ресторанів.

Використання систем віртуального замовлення елімінує потребу у фізичних носіях меню. Процес взаємодії ініціюється, як правило, шляхом сканування QR-коду, що являє собою ключовий елемент інтеграції фізичного та цифрового просторів у ресторанній індустрії. Архітектура таких систем передбачає імплементацію надійних протоколів для забезпечення безпеки

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

платіжних операцій. Спеціалізовані, структуровані процеси оплати гарантують безпеку фінансових транзакцій для користувачів. Впровадження подібних рішень дозволяє закладам забезпечити підвищений рівень епідеміологічної безпеки для відвідувачів, встановлюючи новий стандарт функціонування індустрії.

Загалом, віртуальний портал типу "FoodMarket" є ефективним та безпечним інструментом для оптимізації процесу надання послуг у ресторанному секторі. Ця технологія суттєво покращує споживацький досвід та сприяє зростанню операційної ефективності закладів.

Цифрові технології сприяли значній автоматизації процесів у численних секторах економіки, включаючи галузь харчової промисловості та ресторанного господарства. Розвиток бездротових комунікаційних технологій та мобільних платформ не лише підвищив зручність для кінцевих користувачів, але й створив основу для оптимізації управлінських функцій та інтенсифікації обміну інформацією. Першочерговою метою даного проекту/дослідження є розробка та імплементація системи віртуального замовлення їжі, що функціонує на базі механізму QR-кодів, з акцентом на підвищення безпеки транзакцій та чутливості інтерфейсу користувача.

Система передбачає можливість сканування клієнтами QR-коду для отримання оперативного доступу до цифрового меню та формування замовлення, мінімізуючи фізичний контакт. Для ресторанів процес генерації QR-кодів є технічно простим та інтегрованим у загальну систему управління, що дозволяє здійснювати ефективний трекінг замовлень та моніторинг платіжних операцій. На відміну від традиційного ведення фізичної документації, цифрові записи характеризуються високою доступністю, легкістю модифікації та аналізу. Слід зазначити, що сегмент технологій замовлення їжі є висококонкурентним.

Запропонований інтерфейс користувача орієнтований на забезпечення інтуїтивної навігації, дозволяючи клієнтам досліджувати різноманітність

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

закладів, здійснювати замовлення та персоналізацію страв, проводити електронні платежі та отримувати безпечний сервіс. Це сприяє пріоритезації безпеки та ефективності як для споживачів, так і для операторів ринку.

Протягом останніх років віртуальні портали типу "FoodMarket" здобули значну популярність та широке розповсюдження, що обумовлено декількома ключовими мотиваційними факторами. Одним із головних рушіїв є прагнення до мінімізації фізичного контакту між клієнтами, персоналом закладів та контактними поверхнями, що є критично важливим для зниження ризику передачі інфекційних захворювань та забезпечення безпечного та гігієнічного середовища. Крім того, такі системи сприяють підвищенню загальної ефективності процесів як для споживачів, так і для персоналу ресторанів, оптимізуючи час очікування та виконання замовлень.

В рамках функціоналу віртуального порталу типу "FoodMarket", користувачі можуть легко здійснювати перегляд цифрового меню, виконувати деталізовану персоналізацію замовлень та реалізовувати безпечні електронні платежі. Імплементация цієї системи забезпечує можливість ефективного трекінгу життєвого циклу замовлень та автоматизованого ведення записів, що суттєво знижує ймовірність помилок, пов'язаних з людським фактором. Ключовим елементом підвищення задоволеності клієнтів є надання опцій для кастомізації замовлень. Це сприяє покращенню загального користувацького досвіду та формуванню більш персоналізованої та інтерактивної взаємодії між ресторанами та їх відвідувачами. Заклади отримують можливість агрегувати значні обсяги даних щодо споживацької поведінки та преференцій, що є основою для проведення глибокої аналітики. Результати цієї аналітики можуть бути використані для коригування маркетингових стратегій, оптимізації асортименту та вдосконалення операційних процесів, що у підсумку підвищує загальну ефективність діяльності підприємства ресторанного бізнесу.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

1.3. Аналіз існуючих досліджень і платформ онлайн замовлення їжі

Робота [3] зосереджується на відгуках клієнтів, пропозиціях та варіантах налаштування для клієнтів. Брак вподобань, варіантів налаштування та відгуків у реальному часі описані та пояснені в цій бездротовій системі замовлення їжі. Для подолання цих викликів, з якими стикаються онлайн-системи замовлення їжі, впроваджуються кілька нових методів, завдяки яким клієнти можуть замовляти їжу дистанційно зі своїх пристроїв. Зручний інтерфейс користувача дозволяє клієнтам легко робити замовлення. За допомогою зручного інтерфейсу користувачі можуть взаємодіяти з системою інтерактивним способом для розміщення замовлення. Після розміщення замовлення система в ресторані обробляє його, і ресторан починає готувати їжу. Після завершення приготування їжі її відправляють до столу. Використання таких систем покращує точність замовлень, зменшує помилки, мінімізує час очікування та покращує якість обслуговування.

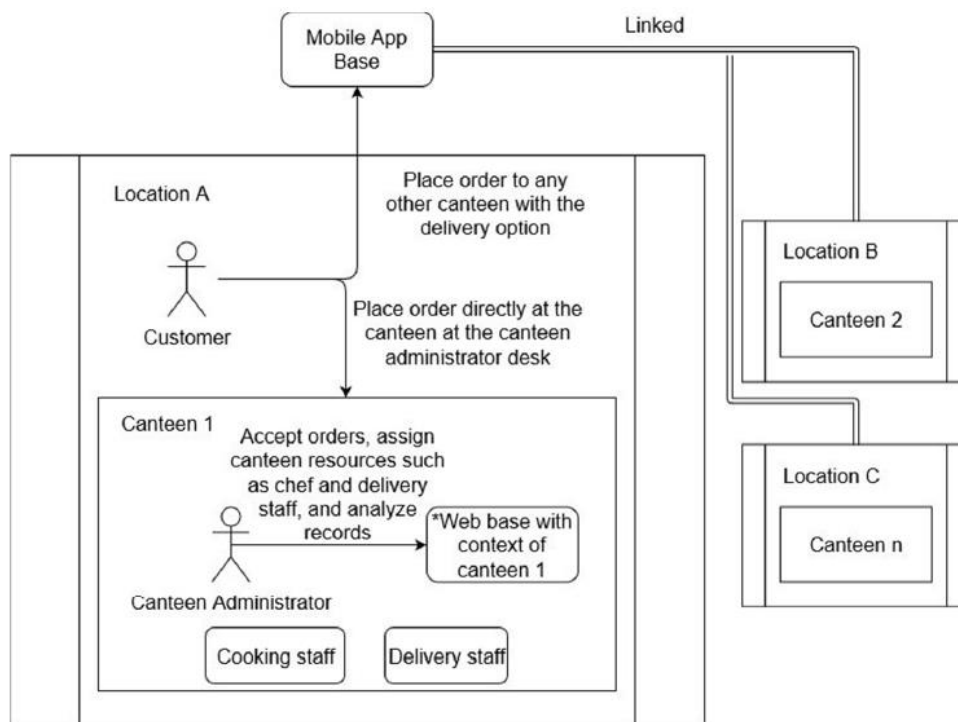


Рисунок 1.1 – Архітектура системи онлайн замовлення їжі

Залучені сутності (рис. 1.1):

Адміністратор - основна сутність, яка відповідатиме за підтвердження замовлень, призначення персоналу їдальні до різних замовлень, а також сутність, яка прийматиме бізнес-рішення на основі аналітичних даних, наданих системою автоматизації.

Користувач на місці (On-Spot User) - клас користувачів, які здійснюватимуть замовлення на стійці адміністратора, аналогічно до описаної вище методології самообслуговування. Це також включає клас користувачів, які замовляють через наданий мобільний інтерфейс та забирають своє замовлення на місці.

Користувач, який очікує доставку свого замовлення - клас користувачів, які замовлятимуть свої страви через мобільний інтерфейс, що буде наданий системою, і очікуватимуть доставку своїх страв персоналом їдальні до бажаних місць. Наприклад, на рисунку 1.1, клієнт, який замовляє їжу з їдальні 2, розташованої у Локації В, і очікує доставку у Локації А.

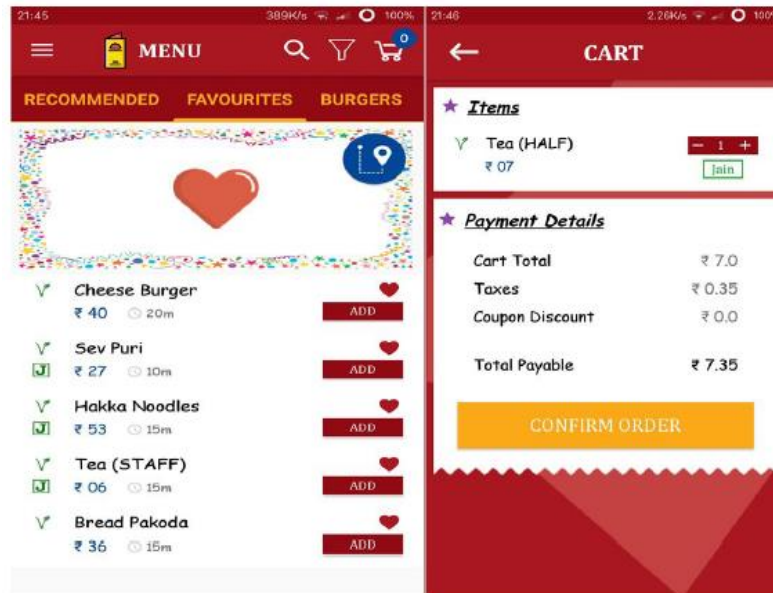
Кухонний персонал - цьому класу користувачів буде надано інтерфейс, де відобразатимуться незавершені замовлення, а також відповідні функції, що дозволяють змінювати статус готовності замовлення. Ці зміни потім відобразатимуться у всій системі.

Персонал доставки - цей клас користувачів відповідатиме за доставку готових страв до відповідних місць. Призначення співробітника для доставки здійснюватиме адміністратор їдальні. Зміна статусу також здійснюватиметься адміністратором їдальні після завершення доставки. Наразі інтерфейс для співробітника з доставки не передбачається, але може бути включений у майбутніх версіях для навігації особи до необхідної локації за допомогою карти.

На рисунку 1.2 зображено основні екрани, що представлені в додатку. Рисунок 1.2 а відображає загальне меню їдальні, обраної користувачем. Рисунок 1.2 б показує остаточну суму за товари, вибрані користувачем.

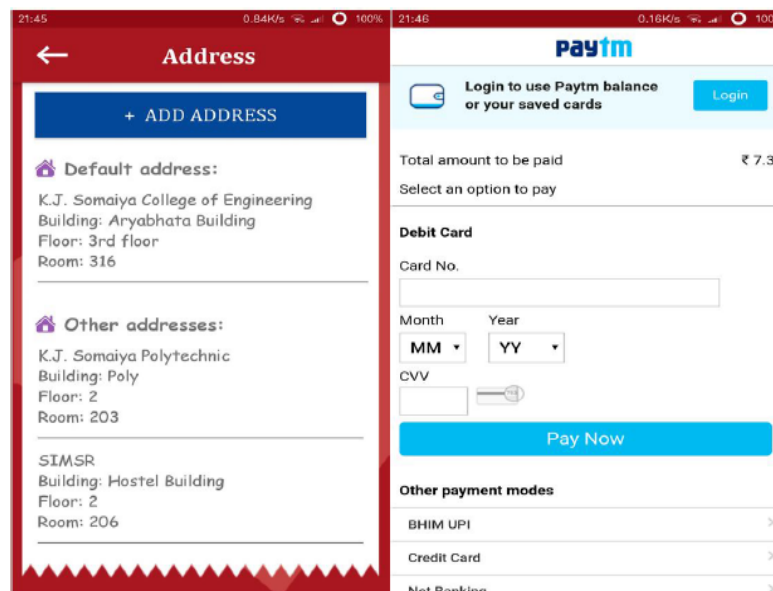
					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Рисунок 1.2 в відображається, коли замовлення є замовленням на доставку. У цьому випадку користувачеві потрібно вибрати одну з адрес або ввести нову. Рисунок 4г висвітлює аспект оплати в системі. Для цієї системи для процесу оплати інтегровано API Paytm.



а) Меню

б) Вартість



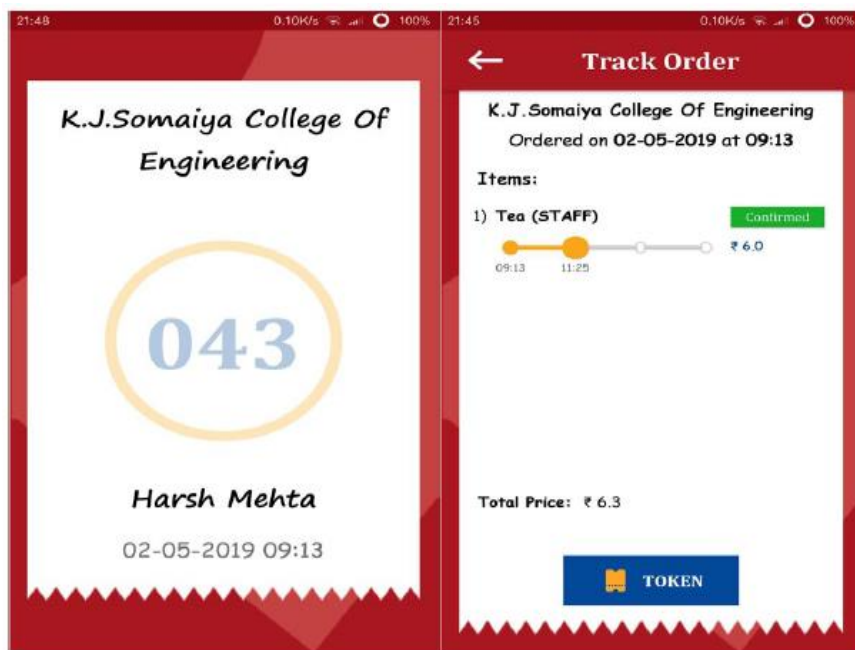
в) Адреса доставки

г) Оплата

Рисунок 1.2 – Основні екрани додатку автоматизації роботи закладу харчування

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Рисунок 1.3 зосереджений на процесі доставки продукту. Екран на рисунку 1.3а відповідає за перші два потоки (процеси). Анімація також додається до токена, таким чином, співробітник доставки може легко автентифікувати токен. Рисунок 1.3 відповідає за відображення статусу замовлення. Зміни статусу, здійснені сутностями (залученими сторонами), відображаються у цьому розділі додатку.



а) Токен

б) Статус замовлення

Рисунок 1.3 – Екрани відстеження замовлення

Ресторани можуть надати клієнтам дуже ефективні та ефективні послуги. За допомогою відгуків у реальному часі клієнти можуть ставити оцінки та писати відгуки. Відгуки та оцінки, надані клієнтами, ресторани можуть використовувати для покращення своїх маркетингових стратегій, що допомагає у зростанні їхнього бізнесу. В цілому, мета авторів — представити налаштовану бездротову систему замовлення їжі за допомогою бездротових технологій зв'язку.

Стаття [4] зосереджена на системі Smart Menu, щоб клієнти могли налаштувати свої замовлення за допомогою цифрових або особистих пристроїв для покращення досвіду харчування. Для розробки та створення інтелектуальної системи ресторану Smart з останніми технологіями. Клієнти приваблюються цими цифровими системами, пропонуючи їм інтерактивні та налаштовувані варіанти меню. За допомогою історії попередніх замовлень клієнтів можна робити рекомендації, щоб полегшити процес вибору бажаної їжі. Ці цифрові дисплеї надають інформацію про різні страви в ресторані, що допомагає клієнтам швидко приймати рішення при виборі бажаних страв. Використання розумного меню може пропонувати рекомендації клієнтам на основі їхніх попередніх замовлень і полегшити процес замовлення їжі. Мета автора — надати клієнтам кращий та простіший спосіб налаштування процесу вибору меню. Автор пояснює необхідність та важливість створення користувацького інтерфейсу з налаштовуваними вподобаннями в цих цифрових дисплеях. До страв у цих цифрових дисплеях додаються зображення страви, щоб приваблювати користувачів перед розміщенням замовлення. Процес приготування також можна додати до цих розумних цифрових меню, щоб клієнт міг прочитати процес приготування, перш ніж він зробить замовлення. Система також дозволяє клієнтам надавати оцінки та відгуки на веб-сайті ресторану. Ці відгуки, які надають клієнти, можуть бути використані бекенд-системою для покращення загального процесу вибору меню. В цілому, автор пояснює значення та важливість використання цифрових дисплеїв Smart Menu для інтерактивного та зручного інтерфейсу користувача, щоб приваблювати клієнтів, пропонуючи їм усі останні вбудовані функції в системі.

Дослідження [5] чітко пояснює розробку та впровадження електронної системи ресторану, яка використовує технологію бездротової локальної мережі та радіочастотну ідентифікацію. Автор запропонував вирішити традиційні методи в системі ресторану, такі як ручне приймання замовлень

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

на їжу, затримка часу та брак налаштувань у меню. Використання технології RFID допомагає відстежувати місцезнаходження столу, доступність столу, розподіл столів, надаючи унікальний ідентифікаційний номер для клієнтів під час бронювання столу. Використання цього процесу допомагає мінімізувати час очікування та зменшити ручну роботу навантаження.

В роботі інтегрують технології WLAN та RFID для імплементації системи замовлення страв, як показано на рисунку 1.4. Ця система надає активні послуги клієнтам і дозволяє використовувати членську картку на основі RFID як електронний гаманець.

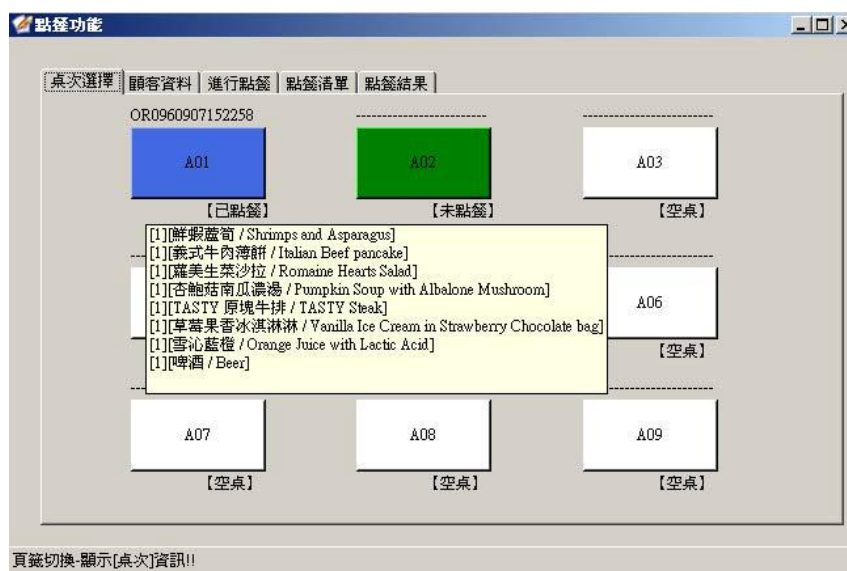


Рисунок 1.4 – Функціонал системи замовлення їжі

Коли клієнт входить до ресторану, співробітник на стійці може використовувати ПК для перевірки поточного стану кожного столика, щоб побачити, чи є вільні обідні столики. Якщо так, то буде призначено офіціанта, який проведе клієнта до столика, і статус столика зміниться з "вільний" на "очікує замовлення". Після того, як офіціант супроводить клієнта до столика, він запитає клієнта, чи має він/вона членську картку. Якщо клієнт є членом, його/її попросять пред'явити членську картку на

основі RFID, і офіціант зможе використати зчитувач RFID на КПК, щоб отримати доступ до RFID-мітки

Таким чином, дані клієнта, записи про витрати та особисті вподобання можуть бути отримані негайно, як показано на рисунку 1.5.



Рисунок 1.5 - Записи про витрати клієнта

Потім офіціант може надати клієнту пропозиції та актуальні акції. Після підтвердження замовлення інформація буде негайно передана на кухню. Статус столика на ПК на стійці зміниться з "очікує замовлення" на "замовлення прийнято", а номер та деталі замовлення також відобразяться на ПК. Після завершення прийому їжі клієнт може сплатити рахунок готівкою або за допомогою членської картки на основі RFID з поповнюваним балансом.

Дані клієнта та записи про витрати зберігаються у внутрішній базі даних для зручності резервного копіювання, запитів та статистичної звітності. Клієнти також можуть зайти на вебсайт ресторану та використати ID з членської картки на основі RFID для доступу до внутрішньої бази даних та отримання відповідної інформації. Крім того, як на стійці, так і біля столика офіціант може вручну переглянути інформацію про витрати клієнта,

рейтинг продажів страв та дохід. Після розрахунку статус столика негайно змінюється з "замовлення прийнято" на "вільний", очікуючи на наступного клієнта. Перелік форм, доступних на ПК на стійці, показано на рисунку 1.6.

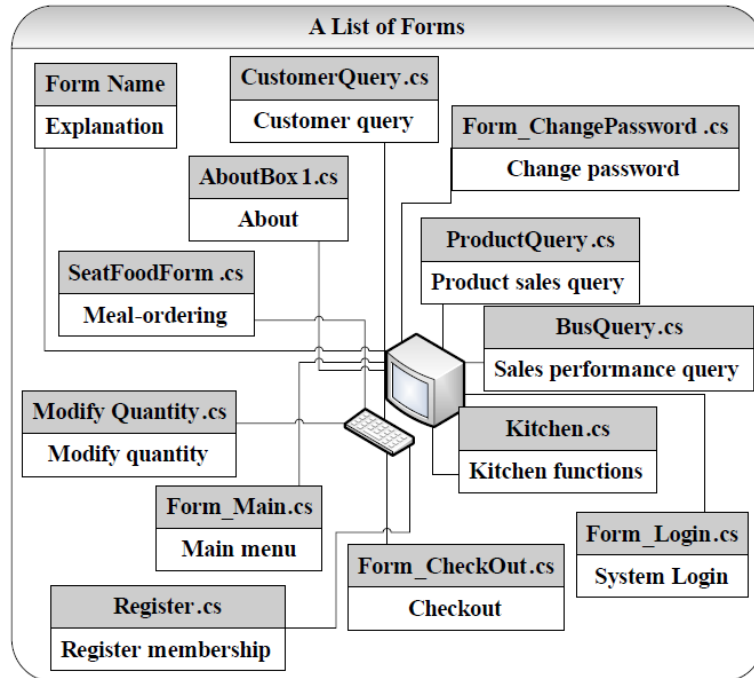


Рисунок 1.6 – Перелік форм, доступних на ПК за стійкою

Автор пояснює значну важливість використання цієї технології для покращення точності, зменшення помилок та підвищення ефективності. Ця електронна система ресторану отримує вподобання та варіанти персоналізації від клієнта. Відгуки та дані попередніх замовлень записуються для покращення та надання клієнтам кращого досвіду харчування. Онлайн-методи оплати та шлюзи допомагають з прозорістю та безпечними платежами. Клієнти також можуть здійснювати безпечні платежі за допомогою платежів, увімкнених за допомогою технології RFID, щоб усунути необхідність готівкових транзакцій. Автор підкреслює, що за допомогою цієї технології ресторани можуть автоматизувати процес бронювання столів дуже легко.

Дослідження [6] зосереджується на та пояснює систему електронного ресторану з багаторазовим дотиком, де використовується передова технологія сенсорного екрану для покращення ефективності та ефективності ресторанів. Ці сенсорні екрани розміщені в різних місцях. За допомогою цих сенсорних екранів, розміщених на столах, клієнти можуть легко отримати доступ до них та замовляти бажані страви. Ручний процес приймання замовлень персоналом ресторану та ручний процес здійснення платежів можна усунути. Автор розглянув ці проблеми та пояснив про електронну систему управління рестораном. Ця система побудована навколо концепції багаторазових сенсорних екранів, дозволяючи клієнтам взаємодіяти з цифровим інтерфейсом користувача. Клієнти можуть шукати свої конкретні елементи, зазначені в категоріях меню. Автор намагається пояснити використання багаторазової технології та її переваги для клієнтів і ресторанів. Клієнти можуть безпосередньо замовляти їжу з цих багаторазових екранів, що зменшує ручну роботу навантаження та час очікування. Автор запропонував цю статтю для вирішення проблем, які є у використанні традиційної системи управління рестораном.

Клієнти можуть переглядати різні категорії меню та налаштовувати свої замовлення перед розміщенням замовлень. Після розміщення замовлення персонал кухні ресторану починає готувати їжу відповідно до вподобань клієнта. Персонал ресторану може відстежувати замовлення та надавати ефективні послуги клієнтам. Власник ресторану або персонал може отримати доступ до цих дисплеїв і відстежувати записи та дані клієнтів, що допомагає покращити операційну ефективність.

Стаття [6] зосереджується на веб-застосунку для процесу замовлення їжі. Основна мета автора — надати користувацький інтерфейс платформи для клієнтів для взаємодії та замовлення їжі. Це веб-застосунок, де клієнти можуть замовляти їжу онлайн на основі своїх вподобань, таких як додавання

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

додаткових інгредієнтів, збільшення або зменшення рівня спецій, додавання додаткових топінгів тощо.




FOOD NAME	FOOD IMAGE	FOOD PRICE	FOOD CATEGORY	ACTIONS
Cheese Burger		RM 16.90	Burger	Delete
Bolognese Spaghetti		RM 10.90	Pasta	Delete
Apple Juice		RM 8.50	Beverage	Delete

Рисунок 1.7 – Деталі замовлення

На основі попередніх замовлень також пропонується персоналізоване меню, що також допомагає в задоволенні користувачів і надає клієнтам ефективний досвід харчування. Автор також пояснює задоволеність користувачів під час використання належного користувацького інтерфейсу. Значення та важливість додавання додаткових функцій, таких як безпечні платежі, оновлення статусу, відстеження замовлень та транзакцій. Процес приготування та доставки можна відстежувати клієнтом за допомогою застосунку на його пристрої.

Сповіщення про замовлення можуть бути надіслані клієнтам для оновлення їх щодо очікуваного часу доставки або будь-яких змін у замовленні. Система дозволяє клієнтам переглядати меню, замовляти їжу та здійснювати платежі безпечно та відстежувати прогрес своїх замовлень. Автор підкреслює важливість зручного інтерфейсу користувача, варіантів налаштування та механізмів відгуків у забезпеченні безпечного та задовільного онлайн-досвіду замовлення їжі.

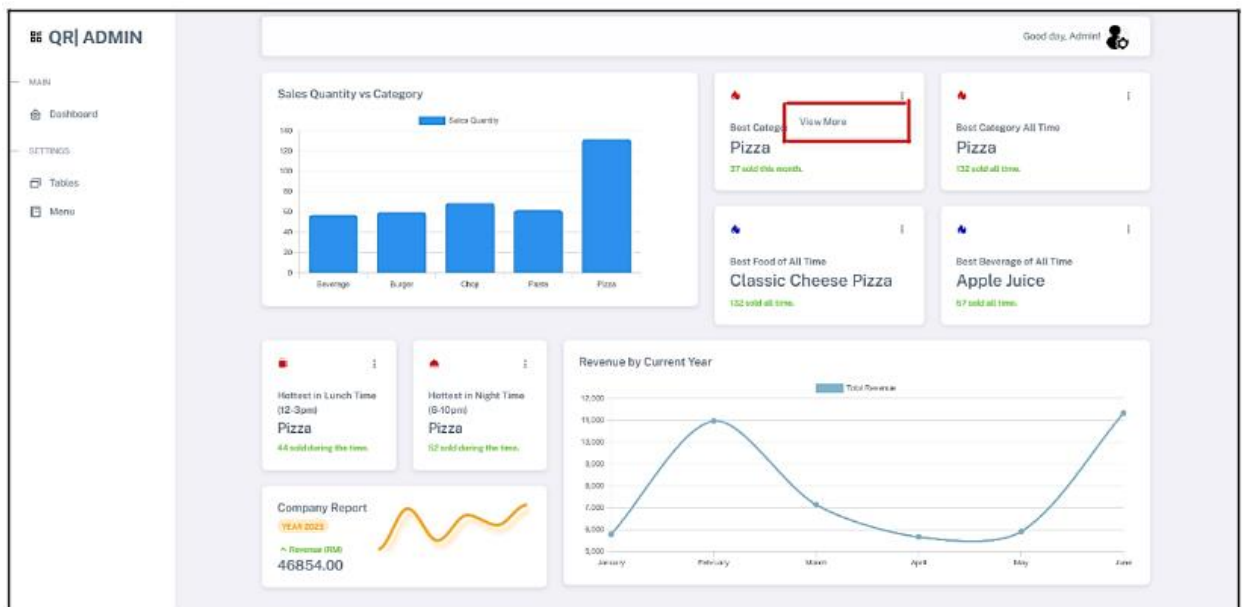


Рисунок 1.8 - Аналітична панель адміністратора щодо замовлень їжі

Система автоматизованої підтримки покупки їжі була розроблена зусиллями шляхом інтеграції популярної платформи разом із бездротовою технологією лише для того, щоб скоротити навантаження на традиційну роботу ресторанів.

З використанням QR-кодів [8] система підтримки покупки їжі, орієнтована на кафетерій та всі інші процеси, пов'язані з покупкою їжі. Система підтримки покупки їжі, орієнтована на кінокомплекс, була розроблена шляхом надання користувачам можливості замовляти через Інтернет та скануючи QR-коди. Кожну деталь закладу обслуговування їжі показала система, розроблена за допомогою QR-коду. Незважаючи на те, що вона могла зменшити навантаження на офіціантів у закладі обслуговування їжі, ця система була хорошою для всіх закладів.

Рисунок 1.9 представляє архітектуру системи, яка містить деталі щодо фронтенд (клієнтської) та бекенд (серверної) частин, що використовуються в системі. На фронтенді були доступні функції резервування столиків, продажів, замовлення, цифрове меню, а також POS-система ресторану. Повний процес інтелектуальної ресторанної системи на основі QR,

включаючи замовлення, оплату, систему меню та хмарний сервер, який використовується для зберігання цих даних, був реалізований у бекенді системи.

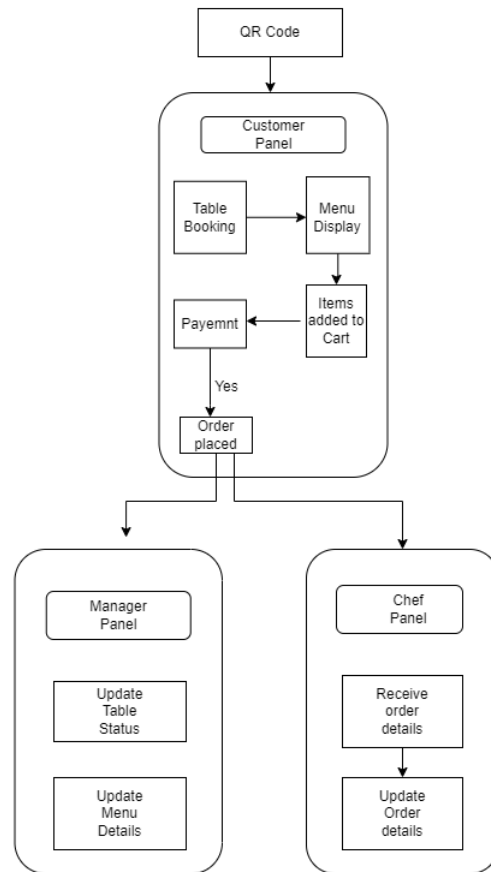


Рисунок 1.9 – Архітектура системи замовлення їжі з використанням QR кодів

На рисунку 1.10 показано головну сторінку системи, яку бачать клієнти після сканування наданого для них QR-коду. Ця сторінка відображає інформацію про ресторан, таку як назва ресторану, адреса ресторану, контактний номер телефону ресторану та години його роботи. На ній відображається кнопка "Забронювати столик", за допомогою якої він (клієнт) може забронювати столик заздалегідь. Також відображається кнопка "Наше меню", за допомогою якої користувач може безпосередньо переглянути меню, доступне в ресторані.

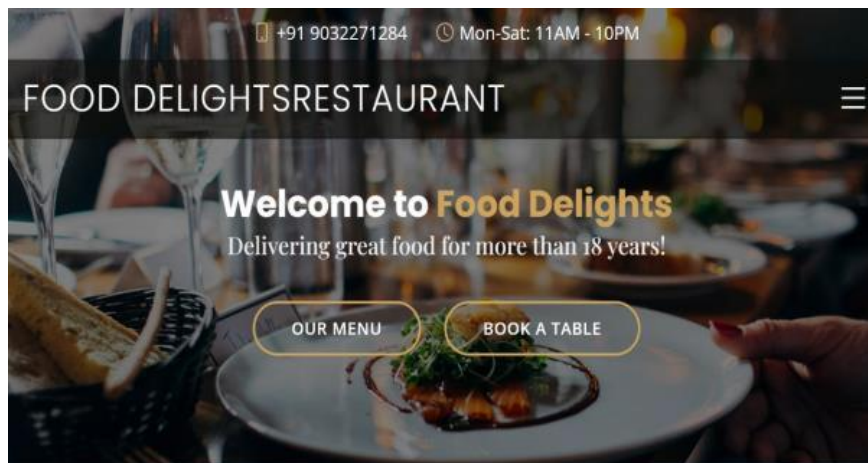


Рисунок 1.10 - Сторінка системи, яку бачать клієнти після сканування QR-коду

Мета проекту "Інтелектуальна ресторанна система з використанням QR-коду" полягає у покращенні досвіду відвідування ресторану для клієнтів та автоматизації процесів, які зазвичай відбуваються у ресторанах, таких як бронювання столиків, вибір страв з меню, замовлення, виставлення рахунку, отримання шеф-кухарем списків замовлень (через офіціантів), а також можливість менеджера контролювати весь процес через свою панель. Робоче навантаження в ресторанах буде зменшено, а також скоротиться кількість відходів паперу та потреба в офіціантах. Використовуючи QR-код, доступний у різних місцях, або використовуючи додаток, клієнт зможе перейти на сторінку ресторану. За допомогою цього він зможе забронювати столик, переглянути позиції меню, вибрати страви, які його цікавлять, і здійснити замовлення виключно через цю систему. Він також зможе оплатити рахунок онлайн або офлайн. Шеф-кухар безпосередньо отримає список замовлених клієнтом страв разом з номером столика на свою панель. Використовуючи ці дані, він може готувати їжу та оновлювати статус замовлення: як тільки його доставлено, або вказати, що це займе певний час. Менеджер може управляти всім рестораном за допомогою панелі менеджера. Ця система зменшить кількість ручних помилок у ресторані, таких як доставка неправильних

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

замовлень, запізнення з наданням меню, коли в ресторані багато відвідувачів і поспіх, скоротить час, витрачений клієнтами, шеф-кухарем та менеджером.

1.4. Функціональні можливості та технологічні обмеження існуючих систем замовлення страв

Сучасні цифрові платформи для замовлення страв надають користувачам функціонал для електронного перегляду асортименту (цифрового меню), параметризації замовлень та безготівкового здійснення фінансових транзакцій. У контексті ресторанного сервісу, імплементація QR-кодів є поширеним методом для ініціації безконтактної взаємодії. Незважаючи на очевидні переваги у сфері зручності та операційної ефективності, застосування таких систем пов'язане з певними технологічними та операційними викликами.

Ідентифіковані обмеження та виклики:

- Існуючі цифрові системи можуть створювати технологічні бар'єри для певних демографічних груп або осіб з обмеженим доступом до сучасних мобільних пристроїв чи стабільного інтернет-з'єднання. Це ускладнює або унеможлиблює повноцінне використання функціоналу системи для частини потенційних клієнтів.

- Відсутність можливості фізичного огляду страв перед замовленням обмежує сенсорне сприйняття. Надані на цифрових платформах зображення та текстові описи можуть не повною мірою відповідати фактичній презентації, розміру порції чи візуальним характеристикам готової страви. Такий дисонанс може потенційно знижувати рівень задоволеності клієнтів після отримання замовлення.

- Надійність функціонування цифрових систем замовлення є критично залежною від стабільності інтернет-з'єднання. Перебої або низька швидкість передачі даних, особливо в локаціях зі слабким покриттям мережі, можуть

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

спричиняти переривання процесу замовлення, затримки у передачі даних та системні збої, що негативно впливає на користувацький досвід.

- Відсутність прямої комунікації з персоналом обмежує можливості для формулювання спеціальних запитів або модифікацій, не передбачених стандартним інтерфейсом замовлення. Це може знизити гнучкість системи для внесення останніх змін або доповнень до замовлення порівняно з традиційними методами обслуговування, де можлива безпосередня взаємодія з офіціантом.

- Можливість виникнення технічних несправностей, програмних помилок або збоїв у процесі обробки замовлень та платежів вимагає розробки ефективних механізмів виявлення та усунення цих проблем. Необхідним є також забезпечення функціоналу для оперативного вирішення складних ситуацій через службу підтримки клієнтів.

- Збір та обробка персональних даних та платіжної інформації клієнтів створює ризики, пов'язані з можливими порушеннями безпеки даних, несанкціонованим доступом або нецільовим використанням інформації. Забезпечення високого рівня кібербезпеки, імплементація надійних протоколів захисту даних та дотримання політики конфіденційності є пріоритетними завданнями для розробників та операторів таких систем.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА СИСТЕМНІ ВИМОГИ ВІРТУАЛЬНОГО ПОРТАЛУ ТИПУ "FOODMARKET"

2.1. Вибір та обґрунтування інструментів для розробки

2.1.1. Фреймворк Django

Django — це веб-фреймворк, який використовується для створення веб-застосунків за допомогою мови Python. Django допомагає нам легко розробляти веб-застосунки, надаючи всі вбудовані інструменти та бібліотеки. У цьому проекті ми використовували стабільну версію Django. Цей застосунок побудовано за допомогою фреймворку з архітектурою моделі-представлення-шаблон, яка відповідає за генерування HTML-виходу. За допомогою цього фреймворку ми можемо виконувати різні дії та переглядати URL-адреси, відображаючи їх на правильних об'єктах. Тут ми використовуємо ORM, який є вбудованою базою даних у цьому фреймворку.

ORM означає відображення об'єктно-реляційних даних. Це вбудована база даних, яка надається в фреймворку Django. За допомогою цього в нашому застосунку ми можемо легко спілкуватися з базою даних безпосередньо. ORM головним чином допомагає у відображенні моделей даних на таблиці бази даних, а операції з базою даних можуть бути керовані та виконуватися легко. Немає необхідності використовувати окремі запити SQL, ORM взаємодіє між мовою Python та базою даних. Це дозволяє розробникам взаємодіяти з базою даних за допомогою Python замість запитів SQL. ORM допомагає зменшити кількість коду, а веб-застосунки можуть бути побудовані дуже ефективно.

2.1.2. База даних MySQL

У цьому проекті ми використовували MySQL як базу даних. Це мова структурованих запитів, де ми можемо виконувати дії під час збереження

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

даних. Дані в базі даних можуть бути маніпульовані та керовані. Дані в цьому проекті будуть збережені та автоматизовані у форматі таблиці.

MySQL є однією з найбільш поширених у світі систем керування реляційними базами даних (СКРБД) з відкритим вихідним кодом (у версії Community Edition) та комерційною підтримкою (Enterprise Edition). Вона функціонує на основі мови структурованих запитів (SQL) і відповідає основним принципам реляційної моделі даних. Приклад таблиці, який показаний нижче, представляє користувача з усіма характеристиками, необхідними для цього користувача. Він представлений з сутностями, такими як ID, Ім'я, Вік і Статус.

Таблиця 2.1 - Приклад таблиці бази даних

ID	Ім'я	Вік	Статус
1	Джон	10	Заброньовано
2	Сара	12	Доступно
3	Марія	14	Список очікування

2.1.3 Інструмент візуалізації SQLyog

У цьому застосунку ми використовували SQLyog для візуалізації бази даних у цій системі. Це програма з відкритим вихідним кодом, яку користувачі можуть завантажити безкоштовно.

SQLyog є спеціалізованим програмним інструментом з графічним інтерфейсом користувача (GUI), призначеним для адміністрування, розробки та ефективного керування базами даних MySQL та MariaDB. Він забезпечує набір функцій, що спрощують взаємодію зі СКРБД та візуалізацію її структури і даних.

Дані можуть бути візуалізовані з користувацьким інтерфейсом, наданим у ній. За допомогою цього інструменту ми можемо виконувати різні дії під час візуалізації даних у базі даних. Дані можуть бути легко керовані

шляхом створення, оновлення та маніпулювання даними в базі даних. Це легко використовувати та може працювати за допомогою простих запитів. Через різні формати файлів дані можуть бути візуалізовані та керовані. Приклад таблиці даних для SQLyog показаний на рисунку нижче для деталей адміністратора ресторану. Таблиця представлена з ID, Ім'ям, Прізвищем, Паролем та Електронною поштою.

Таблиця 2.2 - Представлення даних у SQLyog

ID	Ім'я	Прізвище	Пароль	Електронна пошта
1	Джон	Марк	*****	John@gmail.com
2	Майк	Мет'ю	*****	Mike@gmail.com
3	Тоня	Іваненко	*****	Tonya@gmail.com

2.1.4 Мова Python

Python — це популярна мова програмування високого рівня, яка цінується за свою зрозумілість. Він має легко читаний синтаксис, який зосереджений на зрозумілості. Python доступний для різних середовищ і відомий своєю кросплатформенною сумісністю. Він досяг значного прогресу в таких галузях, як веб-розробка, наука про дані та машинне навчання.

Python є інтерпретованою, високорівневою мовою програмування загального призначення, розробленою Гвідо ван Россумом наприкінці 1980-х - початку 1990-х років. Ключовим принципом її дизайну є підвищення читабельності коду та простота синтаксису, що робить її доступною як для початківців, так і потужним інструментом для досвідчених розробників.

Основні характеристики та особливості:

- Абстрагується від деталей апаратної реалізації та керування пам'яттю, дозволяючи розробникам зосередитися на логіці застосунку.
- Код виконується рядок за рядком інтерпретатором, що спрощує процес розробки та налагодження.

- Тип змінної визначається під час виконання програми, а мова не виконує неявне приведення типів між несумісними типами даних, забезпечуючи безпеку операцій.

- Підтримує об'єктно-орієнтований, процедурний та елементи функціонального стилів програмування.

- Використовує відступи (ідентифікацію) замість фігурних дужок чи ключових слів для визначення блоків коду, що сприяє візуальній чистоті та читабельності.

- Дотримується філософії "batteries included", надаючи широкий набір модулів та пакетів для вирішення різноманітних завдань "з коробки" (робота з мережею, файлами, операційними системами тощо).

- Код, написаний на Python, може виконуватися на різних операційних системах (Windows, Linux, macOS) без необхідності його модифікації.

- Легко інтегрується з кодом, написаним іншими мовами, зокрема C/C++, що дозволяє використовувати Python як "клеюву" мову або для оптимізації критичних за продуктивністю ділянок коду.

2.1.5. XAMPP

XAMPP — це рішення для веб-сервера з відкритим вихідним кодом. Розробники можуть легко налаштувати локальне веб-серверне середовище на своїх комп'ютерах для розробки та тестування. Це програмне забезпечення, яке надає та дозволяє вам запускати Apache, MySQL, PHP і Perl разом без необхідності окремих встановок та конфігурацій. Це спрощує процес використання локального веб-середовища для розробників і доступне для різних операційних систем.

XAMPP інтегрує ключові компоненти серверного стеку, необхідні для розгортання повноцінного локального вебсерверного середовища. Він призначений переважно для розробницьких цілей, дозволяючи фахівцям тестувати вебсайти та вебдодатки на своїх персональних комп'ютерах.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

ХАМРР є де-факто стандартом для локальних середовищ розробки (Development Environments). Він ідеально підходить для вивчення вебтехнологій, розробки та налагодження вебсайтів та застосунків на локальній машині. Однак, через деякі налаштування безпеки за замовчуванням, які оптимізовані для зручності розробки, ХАМРР зазвичай не рекомендується для використання на продуктивних серверах в Інтернеті без додаткового посилення безпеки.

2.1.6. Середовище Visual Studio Code

Visual Studio Code — це інтегроване середовище розробки з відкритим вихідним кодом. Він використовується для розробки коду, такого як виконання коду, і всі типи розширень вбудовані та можуть бути використані та доступні. Він набув великої популярності серед розробників завдяки своїй продуктивності, розширюваності та зручному інтерфейсу користувача. Він також підтримує широкий діапазон мов програмування, фреймворків і інструментів.

Основні характеристики:

- Кросплатформність.
- На відміну від повноцінних IDE, він є менш вимогливим до ресурсів.
 - Забезпечує підсвічування синтаксису, автодоповнення (IntelliSense), форматування коду для величезної кількості мов програмування та розмітки.
- Має інтеграцію з системою контролю версій Git, вбудований термінал, налагоджувач коду.

Головна перевага — величезний маркетплейс розширень, які дозволяють додавати підтримку нових мов, інструментів, тем оформлення та іншої функціональності

2.1.7. Програмні вимоги та апаратні вимоги

В таблиці 2.3 наведені програмні вимоги до проектованої системи.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

Таблиця 2.3 – Програмні вимоги

Операційна система:	Windows 8 або 10
Скрипт на стороні сервера	HTML, MySQL, CSS і Bootstrap
Мова програмування	Python
IDE	PyCharm, Visual Studio Code.
Фреймворк	Django
Інструменти	XAMPP, ORM, MySQL

Апаратні вимоги:

- Оперативна пам'ять - 4 ГБ;
- Жорсткий диск - 160 ГБ;
- Процесор - конфігурація Intel i3.

2.2. Проектування архітектури системи

Архітектура в цій запропонованій системі замовлення їжі підтримує як клієнта, так і адміністратора ресторану. Ця архітектура побудована з багатьма функціями та можливостями, які допомагають адміністратору ресторану та персоналу швидко отримувати доступ до замовлень. Ця система розроблена таким чином, щоб клієнти могли отримати кращий досвід використання цього інтерфейсу.

Клієнти можуть забронювати столики онлайн або по телефону, щоб забронювати стіл заздалегідь. Адміністратор повинен зареєструватися, а потім увійти до свого облікового запису, щоб переглянути замовлення клієнтів та платежі. Клієнти можуть зареєструватися та увійти до свого облікового запису, вони можуть переглядати ресторани, скануючи QR-код, вводячи кількість учасників і замовляючи їжу, здійснюючи платежі та виходячи з системи. Якщо ресторан повний, він буде доданий до списку очікування на певний проміжок часу, поки він не отримає бронювання.

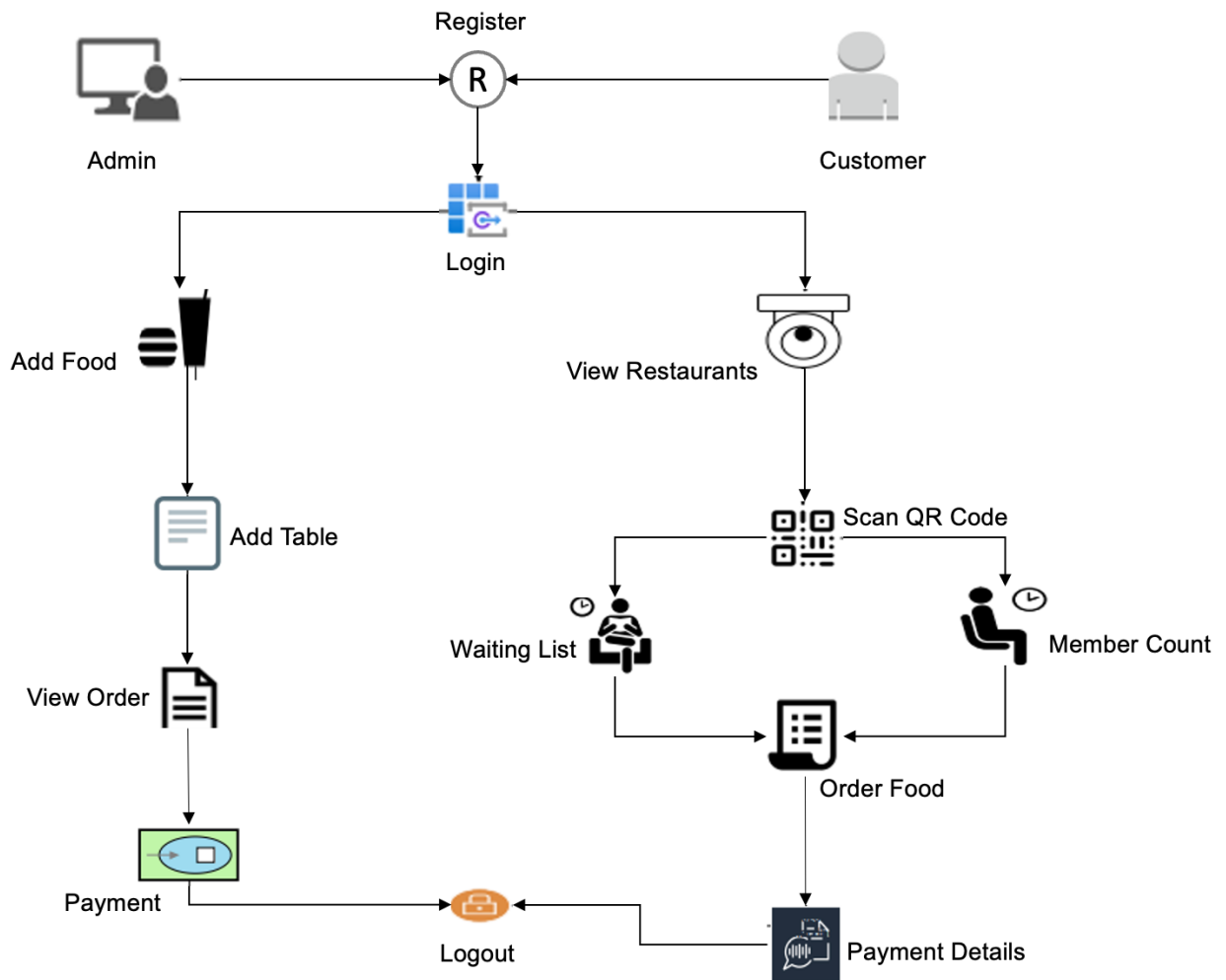


Рисунок 2.1 - Архітектура пропонованої системи

Архітектура системи, на основі якої побудована наша система замовлення їжі за допомогою QR-коду, підтримує як адміністраторів, так і клієнтів. Рисунок 2.1 представляє архітектуру системи.

Наведемо опис архітектури системи замовлення їжі як багаторівневу структуру, що забезпечує взаємодію між користувачами (Адміністратор, Клієнт) та основними функціональними модулями.

Рівень користувацького інтерфейсу (Presentation Layer) - представлений взаємодією Адміністратора та Клієнта із системою. Цей рівень відповідає за відображення інформації (меню, статус замовлення, дані про ресторани) та прийом вхідних даних від користувача (вибір страви, введення платіжних даних, сканування QR-коду).

Рівень бізнес-логіки (Application/Business Logic Layer) обробляє запити користувачів та реалізує основні функції системи, що зображені на схемі як процеси:

Керування доступом: Реєстрація (Register), Вхід (Login), Вихід (Logout). Забезпечує автентифікацію та авторизацію користувачів.

Керування контентом (для Адміністратора) – це процеси Додавання Страв (Add Food), Додавання Столиків (Add Table). Дозволяє адміністратору поповнювати базу даних системи.

Керування замовленнями - Перегляд Ресторанів (View Restaurants), Сканування QR-коду (Scan QR Code), Замовлення Їжі (Order Food), Перегляд Замовлень (View Order - для Адміністратора). Включає логіку вибору, формування та обробки замовлень.

Керування списком очікування - Список Очікування (Waiting List), Кількість Учасників (Member Count). Цей модуль, ймовірно, керує чергою клієнтів, якщо столики зайняті.

Обробка платежів - Платіжні Деталі (Payment Details), Платіж (Payment). Включає логіку обробки транзакцій та взаємодії з платіжними системами.

Рівень даних (Data Access Layer) не відображений безпосередньо на схемі у вигляді сховища, він імпліцитно присутній через операції з даними (додавання страв, столиків, перегляд замовлень, платіжні деталі). Цей рівень відповідає за зберігання, отримання та управління даними системи (інформація про користувачів, ресторани, меню, столики, замовлення, платежі) за допомогою системи керування базами даних (MySQL).

Користувачі взаємодіють із системою через Рівень користувачького інтерфейсу. Інтерфейс користувача передає запити на Рівень бізнес-логіки. Рівень бізнес-логіки виконує відповідні операції, звертаючись до Рівня даних для читання або запису інформації.

Для функціоналу платежів Рівень бізнес-логіки може взаємодіяти із зовнішніми платіжними шлюзами. Адміністратор має доступ до спеціалізованих функцій управління та моніторингу. Клієнт проходить шлях від вибору ресторану/столика до оформлення замовлення та його оплати.

Загалом, представлена архітектура реалізує функціональність онлайн замовлення їжі з розподілом ролей та чітким визначенням основних операційних потоків як для кінцевого клієнта, так і для адміністратора системи.

2.3. Розробка структури системи

Це зручний додаток, де клієнти сканують QR-код і отримують доступ до системи. Вони можуть переглядати меню, налаштовувати замовлення та здійснювати платежі безпечно. Адміністратор і клієнт — це дві сутності в нашій системі замовлення їжі за допомогою QR-коду. Рисунок 2.2 пояснює проектування структури системи замовлення їжі типу "FoodMarket".

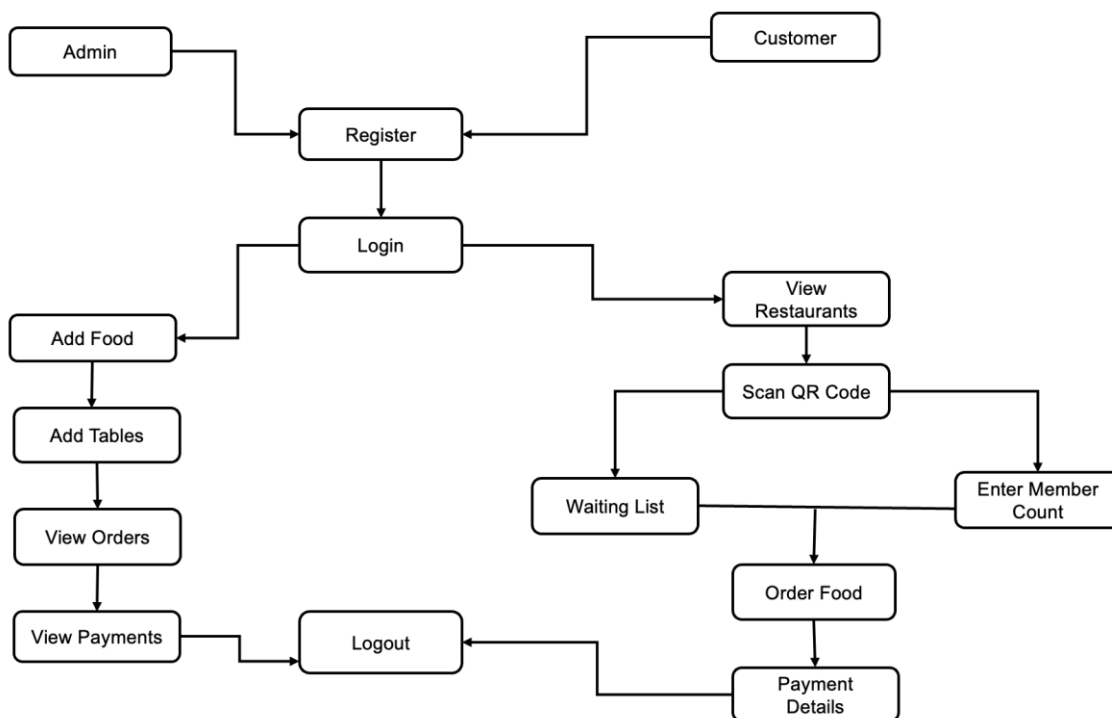


Рисунок 2.2 – Структура проектованої системи "FoodMarket"

Адміністратор може отримати доступ та керувати меню, оновлюючи страви, переглядаючи замовлення, керуючи цінами, налаштовуючи інші системні параметри та керуючи доступністю бронювання столиків. Він також обробляє оновлення списку очікування в реальному часі. Безпечні платіжні шлюзи інтегровані для обробки платежів клієнтів безперебійно.

Схема на рисунку 2.2 ілюструє функціональну структуру системи онлайн замовлення їжі, представляючи її як сукупність взаємопов'язаних модулів та потоків виконання для різних користувацьких ролей.

Користувацькі Ролі. В основі структури лежить розмежування доступу та функціональності для двох основних користувацьких ролей: Адміністратор (Admin) та Клієнт (Customer).

Модулі Керування Доступом:

- Реєстрація (Register): Модуль для створення нових облікових записів користувачів обох ролей.

- Вхід (Login): Модуль для автентифікації користувачів та надання доступу до відповідної функціональності.

- Вихід (Logout): Модуль для завершення сесії користувача.

Модулі Клієнтського Інтерфейсу та Процесів Замовлення:

- Перегляд Ресторанів (View Restaurants) - дозволяє клієнтам переглядати список доступних ресторанів.

- Сканування QR Коду (Scan QR Code) - модуль, який ініціює процес взаємодії за допомогою сканування QR-коду (для ідентифікації столика і доступу до меню).

- Список Очікування (Waiting List) - модуль, пов'язаний з керуванням чергою клієнтів.

- Введення Кількості Учасників (Enter Member Count) - модуль для введення кількості осіб, що використовується для списку очікування або розподілу столиків.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

- **Замовлення Їжі (Order Food)** - основний модуль для вибору страв з меню та формування замовлення.

- **Платіжні Деталі (Payment Details)** - модуль для введення та обробки інформації, необхідної для здійснення платежу.

Модулі Адміністрування:

- **Додавання Страв (Add Food)** - дозволяє адміністратору додавати нові позиції до меню.

- **Додавання Столиків (Add Tables)** - дозволяє адміністратору керувати інформацією про столики у ресторанах.

- **Перегляд Замовлень (View Orders)** - надає адміністратору можливість переглядати деталі всіх замовлень, що надійшли в систему.

- **Перегляд Платежів (View Payments)** - надає адміністратору доступ до інформації про здійснені платежі.

Структура системи визначається зв'язками (стрілками) між цими модулями, що представляють логічні потоки виконання:

- Обидві ролі розпочинають взаємодію з модулів Реєстрації/Входу.

- Після Входу потік виконання розгалужується залежно від ролі.

- Потік Клієнта проходить через етапи Перегляду Ресторанів, потенційне Сканування QR Коду та взаємодію зі Списком Очікування, переходячи до Замовлення Їжі та Платіжних Деталей.

- Потік Адміністратора спрямований на керування контентом (Додавання Страв, Додавання Столиків) та моніторинг операцій (Перегляд Замовлень, Перегляд Платежів).

- Модуль Виходу доступний з основних функціональних гілок обох ролей.

Таким чином, структура системи є модульною, де кожен блок представляє певний функціональний компонент або етап процесу, а зв'язки між блоками визначають послідовність дій та взаємодію між цими компонентами у рамках виконання завдань користувачами різних ролей.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

2.4. Представлення системи з використанням UML

UML (Unified Modeling Language) надає набір нотацій та вказівок для створення візуальних представлень програмних систем. візуалізувати різні аспекти системи, сприяючи аналізу, проектуванню та реалізації.

2.4.1. Представлення випадків використання

Це форма представлення поведінки та показує піктографічну схему можливостей, наданих системою. Рисунок 2.3 показує представлення випадків використання між рестораном і клієнтом.



Рисунок 2.3 - Представлення випадків використання

Ця діаграма (рис. 2.3) ілюструє взаємодію зовнішніх суб'єктів, або акторів, із системою, визначаючи її функціональність з точки зору

користувача. Надана діаграма відображає варіанти використання системи онлайн замовлення їжі.

Розглянемо ключові компоненти діаграми:

1. Системний Контур (System Boundary) - представляє межі самої системи онлайн замовлення їжі. Всі варіанти використання, розташовані всередині контуру, є функціями, які надає система.

2. Актори (Actors) - зовнішні сутності, які взаємодіють із системою. На діаграмі визначено два основних актори:

- Restaurant (Ресторан) - цей актор представляє роль адміністратора або менеджера ресторану, відповідального за управління контентом (меню, столики) та моніторинг операцій (замовлення, платежі) у системі.

- Customer (Клієнт) - цей актор представляє кінцевого користувача системи, який здійснює замовлення їжі.

3. Варіанти Використання (Use Cases) представляють собою окремі функції або послуги, які система надає акторам:

- Register (Зареєструватися)
- Login (Увійти)
- Add Food Info (Додати інформацію про їжу)
- Add Tables (Додати столики)
- View Orders (Переглянути замовлення)
- View Payment (Переглянути оплату)
- View Restaurants (Переглянути ресторани)
- Scan QR (Сканувати QR-код)
- Enter Member Count (Ввести кількість учасників)
- Waiting List (Список очікування)
- Order Food (Замовити їжу)
- Payment Details (Платіжні деталі)
- Logout (Вийти)

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

4. Асоціації (Associations) - лінії, що з'єднують акторів з варіантами використання, вказують на те, що даний актор взаємодіє з відповідною функцією системи.

Проведемо аналіз взаємодії акторів та варіантів використання.

Актор "Restaurant" взаємодіє з варіантами використання, пов'язаними з керуванням даними (Додати інформацію про їжу, Додати столики) та моніторингом діяльності системи (Переглянути замовлення, Переглянути оплату), а також з базовими функціями доступу (Зареєструватися, Увійти, Вийти).

Актор "Customer" взаємодіє з варіантами використання, що охоплюють типовий процес замовлення: пошук та перегляд (Переглянути ресторани), ідентифікація (Сканувати QR-код, Ввести кількість учасників, Список очікування), формування замовлення (Замовити їжу), здійснення платежу (Платіжні деталі), а також базові функції доступу (Зареєструватися, Увійти, Вийти).

Таким чином, діаграма варіантів використання порталу онлайн замовлення їжі чітко моделює функціональні вимоги системи з точки зору її основних користувачів, визначаючи, які дії можуть виконувати актори "Restaurant" та "Customer" у межах наданої функціональності. Вона слугує основою для подальшої деталізації системних процесів та розробки інтерфейсів.

2.4.2. Представлення класів

Представлення класів — це статичне представлення, яке повідомляє та описує структуру системи, представляючи класи, атрибути, процеси та асоціації між класами в системі.

Діаграми класів (рис. 2.4) використовуються для візуалізації статичної структури системи, описуючи класи, їхні атрибути (властивості), методи

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

(операції) та взаємозв'язки між ними. Ця діаграма моделює дві ключові сутності в системі онлайн замовлення їжі: "Restaurants" та "Customers".

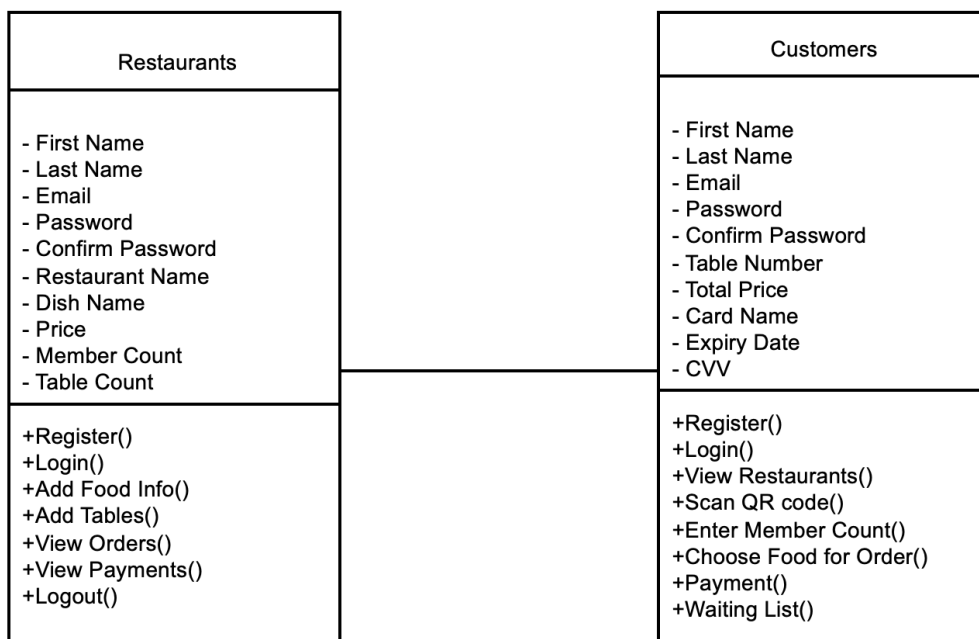


Рисунок 2.4 - Представлення класів

Кожен клас на діаграмі складається з трьох секцій: назва класу, атрибути та методи. Видимість атрибутів позначена символом - (private), що вказує на те, що доступ до них обмежений в межах класу. Видимість методів позначена символом + (public), що означає їхню доступність ззовні для взаємодії з об'єктами класу.

Наведемо опис класів:

1. Клас "Restaurants"

- Призначення - моделює сутність ресторану, а також, імовірно, обліковий запис адміністратора або представника ресторану в системі.

- Атрибути (Приватні):

- First Name, Last Name, Email, Password, Confirm Password - атрибути, що стосуються облікових даних користувача (представника ресторану).

- Restaurant Name - назва ресторану.

- Dish Name, Price - атрибути, що описують елементи меню та їх вартість. Ці атрибути, можливо, вказують на спрощену модель або є частиною даних, якими керує ресторан.

- Member Count, Table Count - атрибути, що можуть стосуватися управління столиками або контролю кількості відвідувачів (наприклад, для списку очікування або бронювання).

- Методи (Публічні):

- Register(), Login(), Logout() - методи для керування сесією користувача-ресторану.

- Add Food Info() - метод для додавання або оновлення інформації про страви в меню.

- Add Tables() - метод для додавання або керування інформацією про столики.

- View Orders() - метод для перегляду замовлень, що надійшли до ресторану.

- View Payments() - метод для перегляду інформації про платежі, пов'язані з замовленнями ресторану.

2. Клас "Customers"

Призначення - моделює сутність клієнта системи.

- Атрибути (Приватні):

- First Name, Last Name, Email, Password, Confirm Password - атрибути, що стосуються облікових даних клієнта.

- Table Number - атрибут, що вказує на номер столика, з якого здійснюється замовлення (у випадку замовлення в закладі).

- Total Price - атрибут для зберігання загальної вартості поточного замовлення.

- Card Name, Expiry Date, CVV - атрибути для зберігання платіжних реквізитів клієнта.

- Методи (Публічні):

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

- Register(), Login(), Logout() - методи для керування сесією клієнта.
- View Restaurants() - метод для перегляду доступних ресторанів.
- Scan QR code() - метод для ініціації дії сканування QR-коду (пов'язаного зі столиком або меню).
- Enter Member Count() - метод для введення кількості осіб (можливо, для функції очікування або бронювання).
- Choose Food for Order() - метод для вибору страв з меню та додавання їх до замовлення.
- Payment() - метод для ініціації та виконання процесу оплати замовлення.
- Waiting List() - метод, пов'язаний з функціоналом списку очікування.

Пряма лінія, що з'єднує класи "Restaurants" та "Customers", вказує на наявність асоціації (зв'язку) між цими сутностями в системі. Хоча на діаграмі не вказано тип асоціації (наприклад, агрегація, композиція) чи її кратність (співвідношення об'єктів, наприклад, "один ресторан має багато замовлень від багатьох клієнтів"), ця асоціація логічно відображає взаємодію клієнтів з ресторанами у процесі замовлення їжі.

Отже, діаграма класів надає статичну, об'єктно-орієнтовану модель основних сутностей системи, їхніх характеристик (атрибутів) та поведінки (методів), слугуючи фундаментальним елементом для розуміння структури даних та функціональних можливостей системи на рівні програмної реалізації.

2.4.3. Представлення послідовності

Представлення послідовності — це тип представлення комунікації, який показує, як операції обробляються та взаємодіють між собою. Він складається з діаграми послідовності повідомлень. Цей тип представлення також має різні назви, такі як діаграми подій, діаграми часу та сценарії подій.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

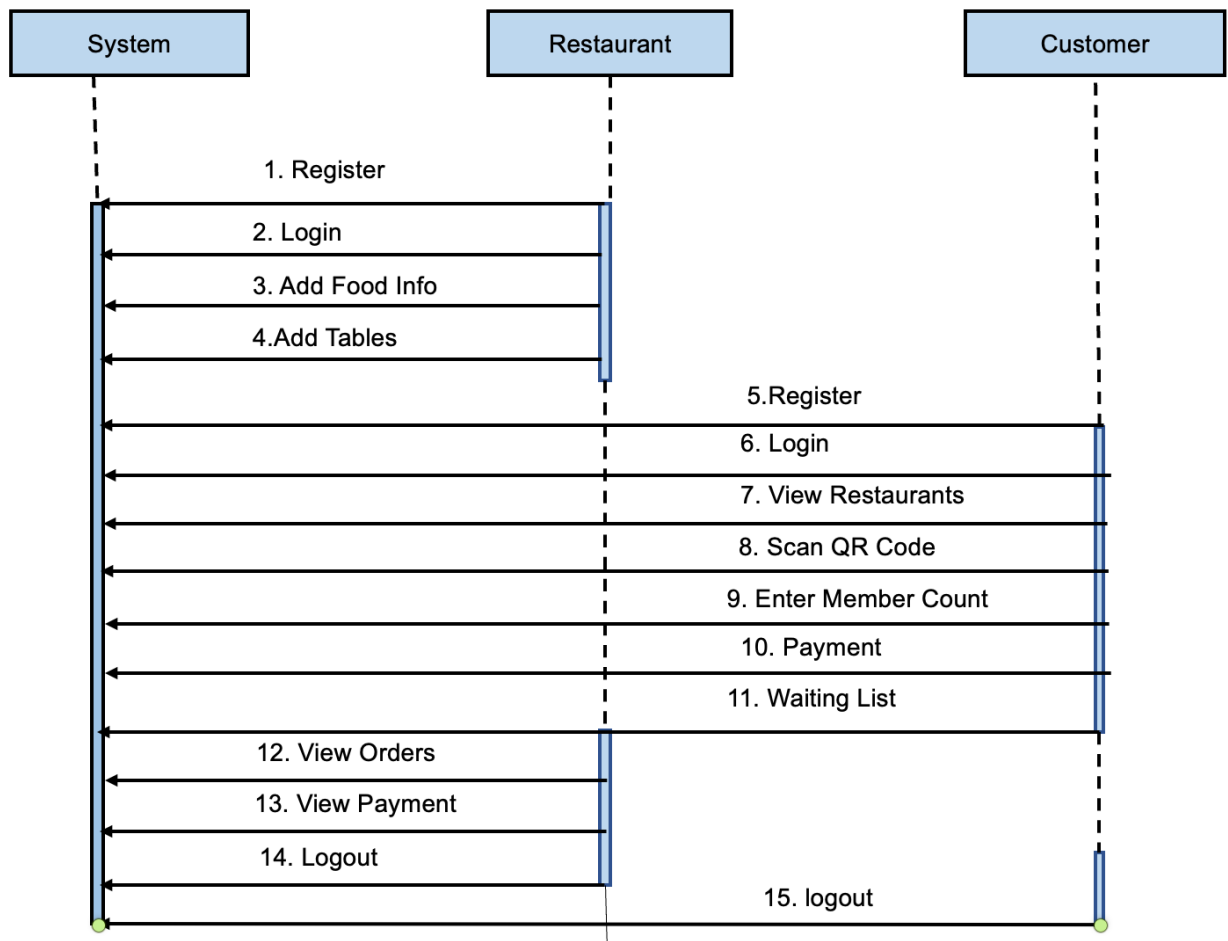


Рисунок 2.5 - Представлення послідовності

Діаграми послідовності використовуються для моделювання динамічної поведінки системи, ілюструючи, як об'єкти або ролі взаємодіють між собою у певному сценарії використання з плином часу. Вони показують послідовність повідомлень, що надсилаються між учасниками для виконання конкретної дії.

Компоненти діаграми послідовності (рис. 2.5):

1. Лінії життя (Lifelines) - вертикальні пунктирні лінії, що йдуть вниз від учасників взаємодії. Вони представляють існування учасника протягом певного періоду часу під час виконання сценарію. На цій діаграмі представлені лінії життя для:

- System (Система): Представляє саму систему онлайн замовлення їжі.

- Restaurant (Ресторан): Представляє актора або компонент, пов'язаний з управлінням рестораном.

- Customer (Клієнт): Представляє актора або компонент, що взаємодіє з системою з боку кінцевого користувача.

2. Повідомлення (Messages) - горизонтальні стрілки між лініями життя, що позначають виклики методів або обмін інформацією між учасниками. Повідомлення розташовані вертикально у хронологічному порядку, відображаючи послідовність взаємодії. Кожне повідомлення має номер, що вказує на його місце в послідовності.

3. Смуги активації (Activation Bars) - вузькі прямокутники на лініях життя, що показують період часу, протягом якого учасник активно виконує операцію або перебуває в стані очікування відповіді.

Діаграма ілюструє кілька паралельних або послідовних потоків взаємодії:

Потік для Ресторану:

1. Register: Ресторан ініціює процес реєстрації в Системі.
2. Login: Ресторан ініціює процес входу до Системи.
3. Add Food Info: Ресторан надсилає Системі повідомлення про додавання інформації про страви.
4. Add Tables: Ресторан надсилає Системі повідомлення про додавання столиків.
5. View Orders: Система надсилає повідомлення до компонента "Restaurant" (імовірно, для відображення або обробки замовлень).
6. View Payment: Система надсилає повідомлення до компонента "Restaurant" (імовірно, для відображення або обробки інформації про платежі).
7. Logout: Ресторан ініціює процес виходу з Системи.

Потік для Клієнта:

1. Register: Клієнт ініціює процес реєстрації в Системі.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Login: Клієнт ініціює процес входу до Системи.
3. View Restaurants: Клієнт надсилає Системі запит на перегляд ресторанів.
4. Scan QR Code: Клієнт ініціює дію сканування QR-коду.
5. Enter Member Count: Клієнт надсилає Системі інформацію про кількість учасників.
6. Payment: Клієнт ініціює процес здійснення платежу через Систему.
7. Waiting List: Клієнт взаємодіє із Системою щодо функціоналу списку очікування.
8. Logout: Клієнт ініціює процес виходу з Системи.

Діаграма послідовності наочно демонструє динамічну поведінку системи, деталізуючи порядок обміну повідомленнями між Системою та її основними користувачами (Рестораном та Клієнтом) для виконання ключових операцій, таких як реєстрація, вхід, керування даними (для ресторану), перегляд, замовлення та оплата (для клієнта).

2.4.4. Представлення діяльності

Діаграма діяльності надає візуальне представлення функцій та потоку керування всередині системи. Це полегшує розуміння та спілкування логіки обробки. Ці діаграми використовуються для моделювання систем робочого процесу та бізнес-процесів, які включають кілька кроків і точок прийняття рішень. Ці діаграми ефективно представляють комерційні та обробні функції елементів, що містяться в системі.

Діаграма діяльності є одним із типів діаграм поведінки в мові моделювання UML (Unified Modeling Language). Вона використовується для графічного моделювання потоку управління або потоку об'єктів у системі, бізнес-процесі, робочому потоці або алгоритмі. На відміну від діаграм взаємодії, які зосереджені на обміні повідомленнями між об'єктами, діаграма

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

діяльності акцентує увагу на самій послідовності виконуваних дій (активностей) та переходах між ними.

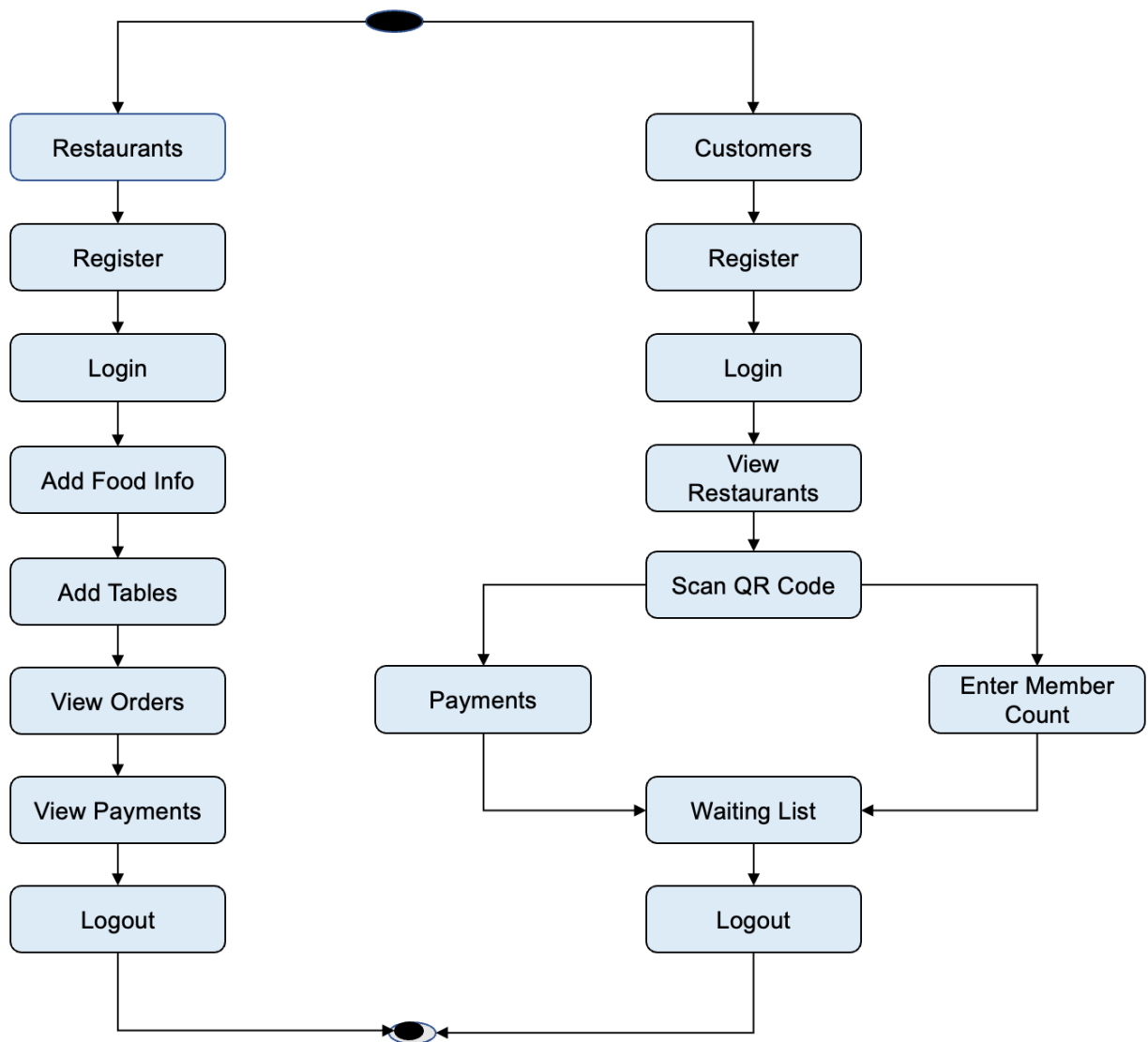


Рисунок 2.6 - Представлення діяльності

2.4.5. Представлення ER-моделі

Це модель «сутність-відношення», яка повідомляє про структуру бази даних. Основним елементом цього представлення є набір відношень і набір сутностей. Цей тип представлення представляє всю логічну структуру бази даних шляхом вказання асоціації між атрибутами.

Змн.	Арк.	№ докум.	Підпис	Дата

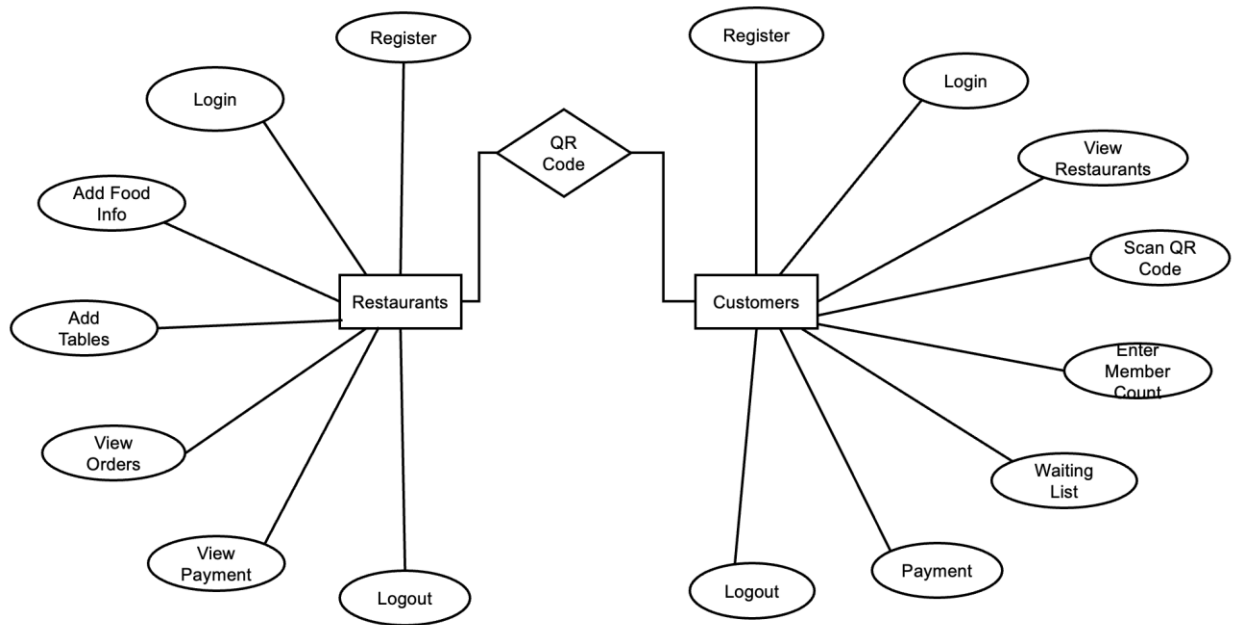


Рисунок 2.7 - Представлення ER моделі

Модель на рисунку 2.7 є візуальним представленням, яке ілюструє взаємозв'язок між ключовими сутностями (або ролями) системи онлайн замовлення їжі та основними функціональними процесами або можливостями, доступними в системі.

Опишемо ключові елементи схеми та їх інтерпретацію.

1. Сутності / Ролі:

- Restaurants (Ресторани): Представляє сторону постачальника послуг або адміністративну роль, що керує контентом і замовленнями.

- Customers (Клієнти): Представляє сторону кінцевого користувача, який здійснює замовлення.

2. Функціональні Процеси / Можливості. Кожен овал позначає конкретну дію або функцію, яку можна виконати в системі. Ці процеси згруповані навколо відповідних сутностей, вказуючи на їхню причетність:

- Спільні процеси: Register (Реєстрація), Login (Вхід), Logout (Вихід) – базові функції керування доступом для обох ролей.

- Процеси, пов'язані з Ресторанами: Add Food Info (Додати інформацію про їжу), Add Tables (Додати столики), View Orders (Переглянути

замовлення), View Payment (Переглянути оплату) – функції для управління контентом та моніторингу.

- Процеси, пов'язані з Клієнтами: View Restaurants (Переглянути ресторани), Scan QR Code (Сканувати QR-код), Enter Member Count (Ввести кількість учасників), Waiting List (Список очікування), Order Food (Замовити їжу), Payment (Оплата) – функції для вибору, замовлення та оплати.

Зв'язок "QR Code": Ромб з позначкою "QR Code", що з'єднує "Restaurants" і "Customers", виділяє специфічний тип взаємодії між цими двома сутностями. Це означає, що взаємодія клієнта з рестораном (наприклад, ініціація замовлення за столиком) здійснюється саме через механізм сканування QR-коду, наданого рестораном. Ромб, хоча і є символом зв'язку в ER-діаграмах, тут використовується для позначення конкретного способу взаємодії, а не чисто реляційного зв'язку даних.

Отже, ця схема слугує для візуалізації ключових учасників системи (Ресторани, Клієнти) та переліку основних функцій або дій, які виконуються у системі онлайн замовлення їжі. Вона надає високорівневе розуміння функціонального обсягу системи та зв'язку між основними дійовими особами та доступними можливостями, включаючи виділення специфічного каналу взаємодії через QR-код.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНСТРУМЕНТІВ ВІРТУАЛЬНОГО ПОРТАЛУ ТИПУ "FOODMARKET" ЗАМОВЛЕННЯ ЇЖИ

3.1. Налаштування та конфігурація проекту

В цьому підрозділі пояснюється процес реалізації та конфігурації цього проекту та надається загальний огляд високого рівня.

Спочатку було завантажено та встановлено панель керування XAMPP, яка надає веб-сервер Apache та базу даних MySQL. Потім запущено службу MySQL на панелі керування.

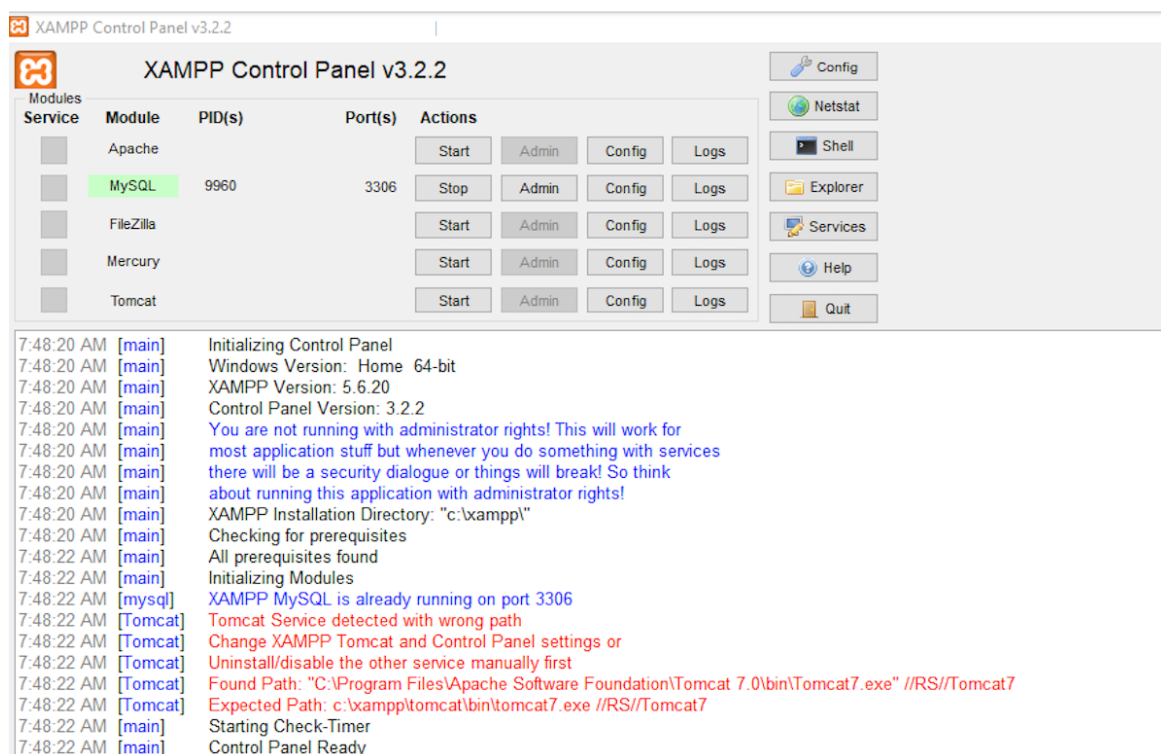


Рисунок 3.1 - Панель керування XAMPP

Потім завантажено та встановлено SQLyog або можна будь-який інший клієнт MySQL і налаштовано параметри підключення до бази даних, використовуючи той самий номер порту що в проекті Django. SQLyog — це потужний графічний інтерфейс користувача, де ми можемо створювати,

видаляти та змінювати таблиці бази даних, індекси та інші об'єкти бази даних.

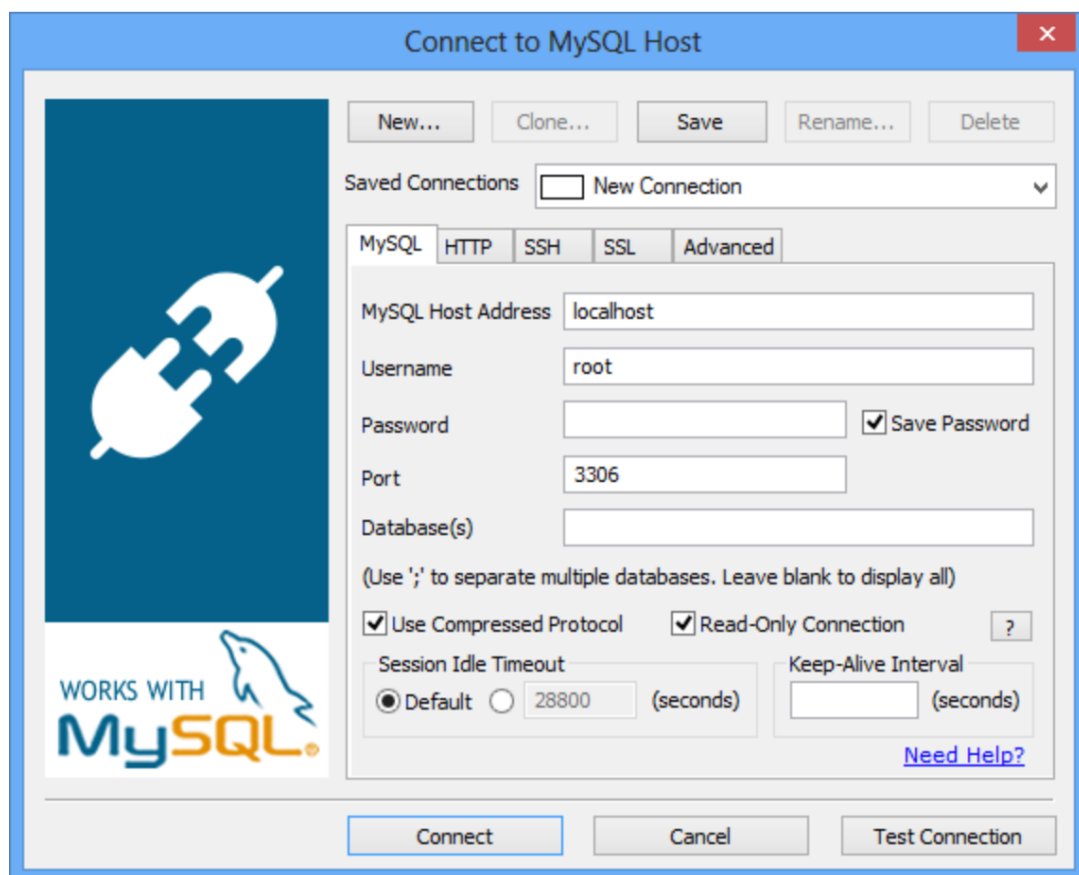


Рисунок 3.2 - Конфігурація SQLyog

Реалізація функціоналу системи електронного замовлення страв передбачає використання фреймворку Django, встановлення якого здійснюється засобами пакетного менеджера Python (pip). Архітектура даного проекту побудована відповідно до патерну Model-View-Template (MVT), характерного для фреймворку Django.

Розгортання фреймворку Django здійснюється командою `pip install Django`. Ініціація нового проекту виконується шляхом виконання команди `django-admin startproject <назва_проекту>`. Генерація окремих модулів-застосунків, що інкапсують специфічну логіку, здійснюється командою `python manage.py startapp <назва_застосунку>`. У межах даного проекту для ідентифікації основного проекту було використано назву

"система_електронного_замовлення_страв", а для функціональних модулів-застосунків – "ресторан" та "клієнт". В результаті створення проекту формується коренева директорія, що містить базові конфігураційні файли, включаючи settings.py, який відповідає за загальні налаштування системи та підключення.

Структуру даних системи визначають моделі Django, що декларативно описують сутності предметної області та їхні взаємозв'язки. Ці моделі імплементуються у файлі models.py відповідних застосунків. У контексті даної системи у models.py було визначено моделі, що відображають такі сутності, як облікові записи користувачів, елементи меню, замовлення, інформація про бронювання столиків, управління чергою очікування та деталі фінансових транзакцій.

Синхронізація визначених моделей зі схемою реляційної бази даних здійснюється за допомогою механізму міграцій Django. Команда python manage.py make migrations аналізує поточний стан моделей проекту, порівнює його з попередньою схемою бази даних та генерує відповідні файли міграції. Кожен файл міграції містить програмний код на Python, який декларативно описує необхідні зміни у структурі бази даних (наприклад, створення таблиць, додавання полів тощо). Виконання команди python manage.py migrate призводить до послідовного застосування цих файлів міграції, що транслюється у виконання відповідних SQL-команд, спрямованих на модифікацію схеми бази даних. Система міграцій забезпечує версійність бази даних та спрощує процеси розгортання та оновлення застосунку.

В результаті виконання міграцій у структурі бази даних імплементованої системи електронного замовлення страв були створені таблиці, що відповідають за зберігання даних щодо облікових записів клієнтів, інформації про фінансові транзакції клієнтів, параметрів резервування столиків, деталей замовлень користувачів, інформації про

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

ресторани, асоціацій QR-кодів з ресторанами та даних про столики ресторанів.

На рисунку 3.3 наочно продемонстровано факт успішного створення таблиць бази даних у рамках проекту системи електронного замовлення страв.

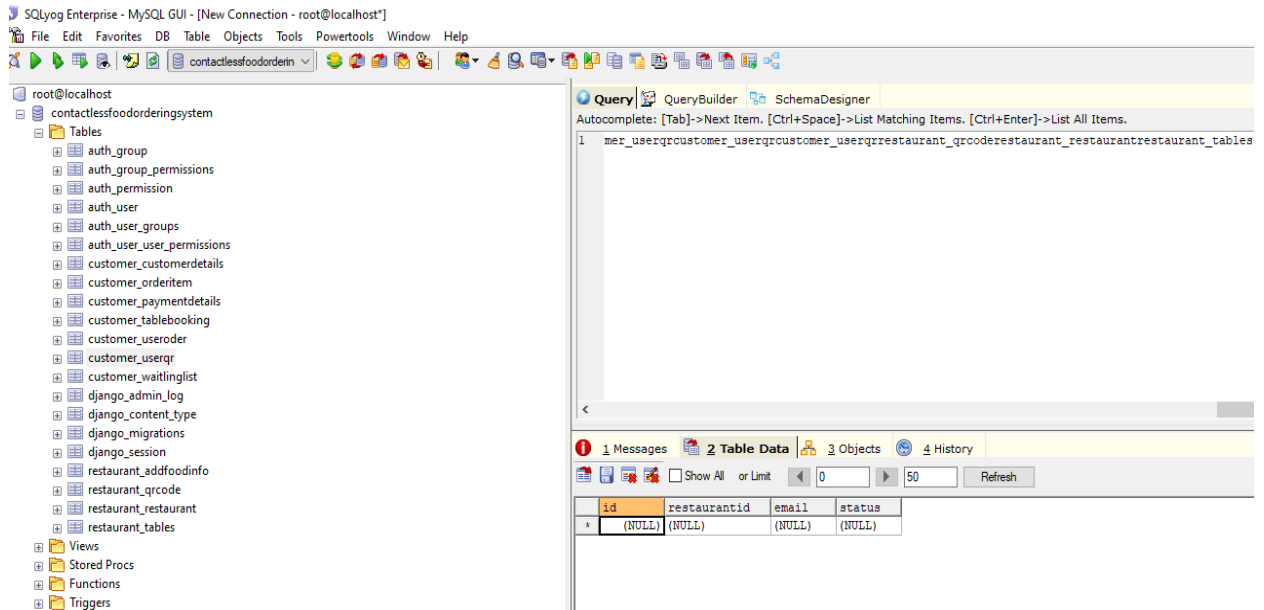


Рисунок 3.3 - Схема створення таблиць бази даних

Обробка вхідних HTTP-запитів та реалізація бізнес-логіки системи здійснюється у файлі views.py відповідних застосунків (Customer та Restaurant). У цьому файлі імплементується функціонал для аутентифікації користувачів, відображення меню, обробки замовлень тощо. Для рендерингу користувацького інтерфейсу у структурі застосунків створюються директорії для HTML-шаблонів. Маршрутизація запитів до відповідних функцій представлень (views) конфігурується у файлі urls.py кореневої директорії проекту, де визначаються URL-шаблони для модулів клієнта та ресторану. Для програмної генерації QR-кодів, асоційованих з цифровими меню ресторанів або окремими столиками, здійснено інтеграцію відповідної бібліотеки, наприклад, qrcode.

Запуск локального сервера розробки для тестування функціоналу здійснюється командою `python manage.py run server`. При початковому доступі до веб-інтерфейсу проекту, відбувається перенаправлення на головну сторінку імплементованої системи електронного замовлення страв, доступної за адресою локального хоста.

3.2. Представлення роботи з базою даних

Фреймворк Django має вбудовану базу даних, відому як ORM (Object Relational Mapping). Виберіть базу даних MySQL, яка налаштована в проекті в файлі `settings.py`. Таблиці можуть бути створені на основі моделей, визначених у проекті. `SQLyog` налаштований з базою даних, він надає інтерфейс для перегляду, зміни та візуалізації структури таблиці. За допомогою `SQLyog` ми можемо дослідити структуру кожної таблиці, яка включає назву таблиці, назву стовпця та типи даних.

Після виконання міграцій за допомогою команд (лістинг 3.1.) створюються блюпринти таблиць бази даних, визначених у `models.py`, для адміністратора, клієнтів, ресторанів, додавання столиків, інформації про їжу, елементів замовлення, списку очікування та платежів.

Лістинг 3.1 Команди виконання міграцій

```
python manage.py make migrations
python manage.py migrate
```

Дані, що зберігаються в базі даних у таблицях у формі рядків, представляють конкретні записи або записи, а стовпці представляють конкретну інформацію. Визначте всі моделі в файлі `models.py` для деталей клієнта, бронювання столиків, замовлень користувачів, списку очікування та деталей платежів. Нижче наведено код, який показує, як визначити моделі.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг 3.2. Код визначення моделі

```
class Customerdetails(models.Model):
    first = models.CharField(max_length=50)
    last = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    password = models.CharField(max_length=50)

    def __str__(self):
        return self.username
```

Таблиця 3.1 - Представлення даних таблиці для деталей клієнта

id	First Name	Last Name	Email	Password
1	mark	mark	mark@gmail.com	12345678
2	mike	mike	mike@gmail.com	34592678
3	tony	tony	tony@gmail.com	67543214

Приклад таблиці даних для списку очікування також показаний у таблиці нижче. Вона представлена з id, номером столика, кількістю учасників та часом для доступності.

Таблиця 3.2 - Представлення даних таблиці для списку очікування

id	Table Number	Member Count	Email	Time
1	T2	2	mark@gmail.com	0 хв
2	T6	3	mike@gmail.com	20 хв
3	T16	2	tony@gmail.com	60 хв

Таким чином, всі моделі бази даних повинні бути визначені в файлі models.py для модулів restaurant і customer, щоб створити таблиці бази даних. Файл settings.py кореневої папки містить різні налаштування та конфігурації для проекту Django: підключення до бази даних та URL-адреси. Усі конфігурації бази даних та встановлені модулі для цього проекту статичні

URL-файли (CSS, JavaScript файли) і конфігурації шаблонів визначені в цій кореневій папці.

```
# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'Contactlessfoodorderingsystem',
        'HOST': 'localhost',
        'USER': 'root',
        'PASSWORD': '',
        'PORT': 3306,
    }
}
```

Рисунок 3.4 - Підключення до бази даних

Підключення до бази даних налаштовано в файлі settings.py у кореневій папці. У терміналі необхідно запустити команду:

```
python manage.py runserver
```

Головна сторінка системи безконтактного замовлення їжі відображається на локальному хості <http://127.0.0.1:8000>.

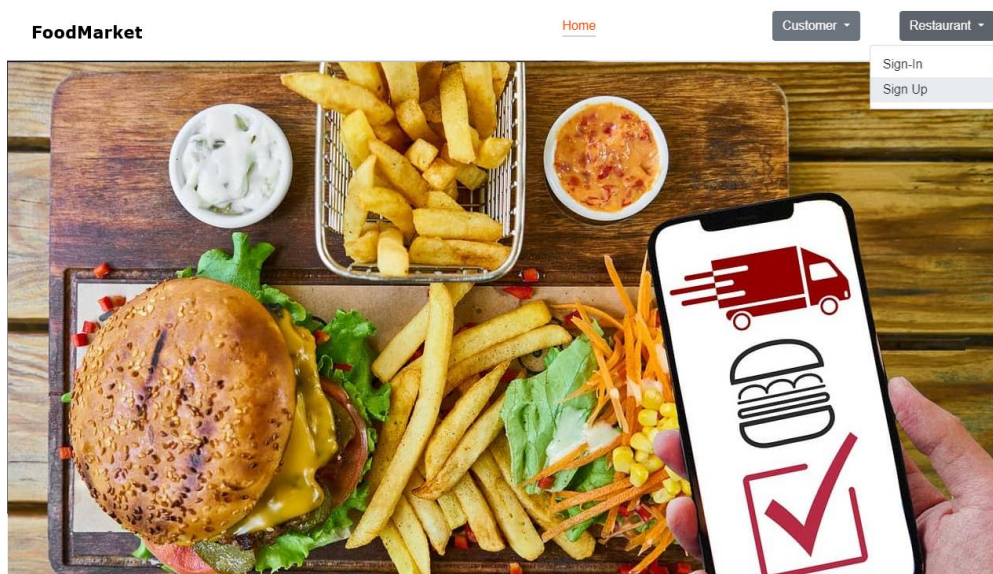


Рисунок 3.5 - Головна сторінка порталу замовлення їжі “FoodMarket”

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

3.3. Реалізація інтерфейсу системи замовлення їжі

Головна сторінка системи замовлення їжі за допомогою QR-коду спроектована з двома модулями: Ресторан (Адміністратор) та Клієнт. Власник ресторану або адміністратор повинен зареєструватися, щоб створити обліковий запис. Ця інформація буде використана для створення облікового запису.

3.3.1. Інтерфейс адміністратора

Sign Up

The screenshot shows a 'Sign Up' form with the following fields: First Name, Last Name, Email address, Password, Password Confirmation, and Restaurant Name. Below the fields is a black button labeled 'CREATE ACCOUNT' and a link that says 'Already have an account? Please Sign In'.

Рисунок 3.6 - Вікно реєстрації адміністратора ресторану

Після створення облікового запису адміністратор може увійти за своїми даними.

Sign In

The screenshot shows a 'Sign In' form with the following fields: User Email (with the example 'sam@gmail.com') and Password (with masked characters '*****'). Below the fields is a black button labeled 'SIGN IN' and a link that says 'Don't have an account? Create One'.

Рисунок 3.7 – Вікно входу адміністратора ресторану

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

Після входу в свій обліковий запис адміністратор може отримати доступ до меню.

Please Add Your Menu Here

<input type="text" value="Dish name"/>	
<input type="text" value="Price"/>	
<input type="button" value="ADD"/>	
Pizza	10 \$ X
burger	10 \$ X
brownie	5 \$ X
coffee	5 \$ X
sundae	10 \$ X

Рисунок 3.8 – Вікно додавання меню адміністратором

Адміністратор ресторану може додавати, редагувати або видаляти страви з меню. Адміністратор може додавати або створювати столики відповідно до кількості учасників, введених клієнтом. Адміністратор може ввести назву страви та деталі ціни. Адміністратор також має доступ до додавання та управління столиками, вказуючи номер столика та максимальну місткість, яку він може вмістити.

Please Add Tables

<input type="text" value="Members"/> 2	
<input type="text" value="Count"/> 2	
<input type="button" value="ADD"/>	
T16	2 X
T17	2 X
T18	3 X
T19	3 X
T20	3 X

Рисунок 3.9 – Вікно додавання столиків адміністратором

Адміністратор ресторану може вийти після введення всіх даних ресторану з облікового запису.

3.3.2. Інтерфейс користувача (клієнта)

Клієнт повинен зареєструватися або увійти за своїми даними та створити обліковий запис. Він може зареєструватися, ввівши свої дані, такі як Ім'я, Прізвище, Електронна пошта та Пароль, і створити обліковий запис.

Sign Up

First Name tony
Last Name tony
Email address tony@gmail.com
Password *****
Password Confirmation *****
CREATE ACCOUNT

[Already have an account? Please Sign In](#)

Рисунок 3.10 - Реєстрація клієнта

Якщо у клієнта вже є обліковий запис, він може безпосередньо увійти, використовуючи свої дані, і отримати доступ до свого облікового запису. Клієнти можуть переглядати всі ресторани, перелічені на головній сторінці, та отримати доступ до кожного ресторану. Клієнти можуть вибирати бажані страви з меню.

Клієнти можуть увійти, ввівши дані, надані під час реєстрації. Після входу з даними його перенаправляють на головну сторінку.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

Sign In

User Email
tony@gmail.com

Password

SIGN IN

Don't have an account? [Create One](#)

Рисунок 3.11 - Вікно входу клієнта

Тут клієнти можуть переглядати ресторани після входу.

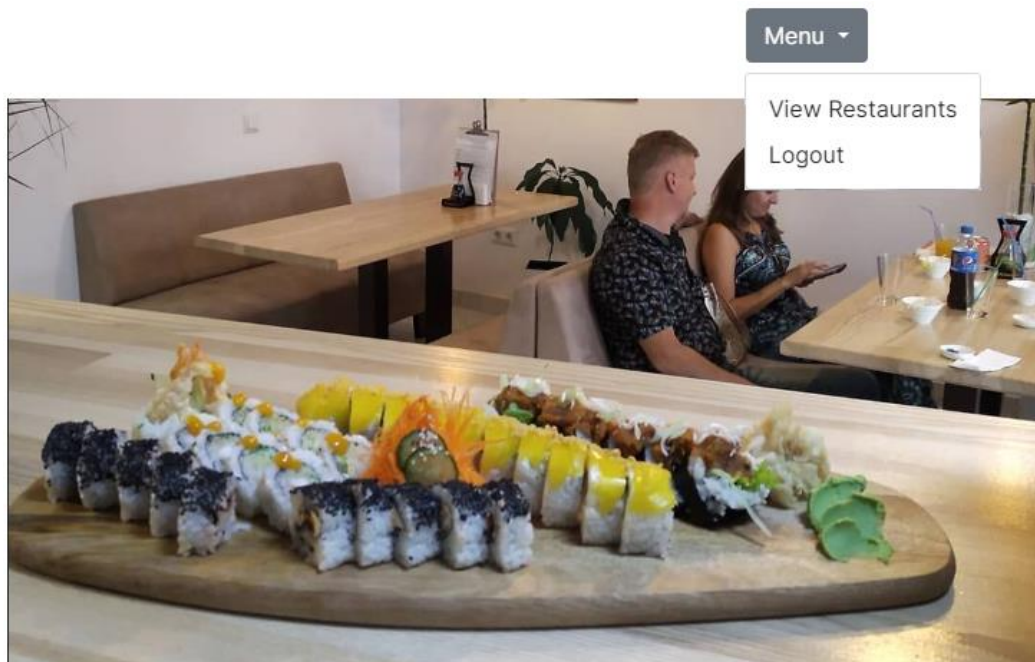


Рисунок 3.12 – Меню для перегляду ресторанів

Після натискання на "Перегляд ресторанів" усі ресторани, які оновлені в системі, будуть відображені. Генерується QR-код, коли клієнт вибирає та натискає на ресторан.

Змн.	Арк.	№ докум.	Підпис	Дата

БР.ІП – 14.00.00.000 ПЗ

Арк.

64



Рисунок 3.13 - QR-код

Тепер клієнту потрібно сканувати QR-код і ввести кількість осіб, щоб система забронювала стіл для клієнта в залежності від кількості.

Please Enter **Members count**

Рисунок 3.14 – Вікно введення кількості учасників

Після введення кількості учасників клієнт може переглядати та переглядати меню. Клієнти можуть додавати або вибирати бажані страви з меню та продовжувати процес замовлення для розміщення замовлення.

Menu

Pizza	10 \$ X	<input type="text" value="1"/>
burger	10 \$ X	<input type="text" value="1"/>
brownie	5 \$ X	<input type="text" value="2"/>
coffee	5 \$ X	<input type="text" value="1"/>
sundae	10 \$ X	<input type="text" value="2"/>

Рисунок 3.15 – Процес замовлення їжі

Користувачі можуть переглядати та переглядати свої замовлення після вибору бажаної їжі перед остаточним розміщенням та здійсненням платіжних транзакцій.

User Order

Id	Dish Name	Dish Price	Dish Count
15	Pizza	10	1
16	burger	10	1
17	brownie	5	2
18	coffee	5	1
19	sundae	10	2

Payment

Рисунок 3.16 – Замовлення користувача

Якщо ресторан зайнятий або немає доступних столиків для бронювання, клієнтів буде додано до списку очікування. Після введення кількості учасників його буде додано до списку очікування з вказаним часовим інтервалом. Він покаже очікуваний час клієнту, перш ніж він отримає доступність.

Table Number	Member Count	Time For Availability
T2	2	0 Min
T6	2	0 Min
T16	2	0 Min
T17	2	0 Min
T16	2	60 Min

Click on Menu

Рисунок 3.17 – Список очікування

Клієнт доданий до списку очікування, і тепер клієнт може натиснути на меню та вибрати бажані страви для розміщення замовлення, перш ніж він отримає стіл.

Menu

Pizza	10 \$ X	1
burger	10 \$ X	
brownie	5 \$ X	1
coffee	5 \$ X	2
sundae	10 \$ X	

ORDER

Рисунок 3.18 – Попереднє замовлення їжі

Після розміщення замовлення клієнт може переглядати та переглядати замовлення та продовжувати оплату в замовленні користувача. Клієнти можуть переглядати назву, ціну та кількість замовлених товарів.

User Order

Id	Dish Name	Dish Price	Dish Count
28	Pizza	10	1
29	brownie	5	1
30	coffee	5	2

Payment

Рисунок 3.19 – Перегляд замовлення

Натиснувши на "Оплата", його перенаправить на сторінку оплати, де клієнт може ввести всі дані своєї картки.

Payment page

Table Number T16
Total Amount 25
Name on Card tony
Card Number 1234567865473344
Expiry date 12/43
CVC ...
SUBMIT

Рисунок 3.20 – Сторінка оплати

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

Клієнт повинен ввести дані своєї кредитної картки та здійснити платіж. Після надсилання оплати буде показано статус оплати "успішно".

Адміністратор ресторану має доступ для перегляду та відстеження всіх замовлень і платежів клієнтів. Адміністратор може переглядати всі дані про платежі з номером столика, ідентифікатором, електронною поштою та загальною сумою, яку повинен сплатити клієнт.

User payments

Id	User Email	Amount	Table Number
7	tony@gmail.com	55	T17
8	tony@gmail.com	60	T15
9	tony@gmail.com	20	T16
10	tony@gmail.com	25	T16

Рисунок 3.20 - Перегляд платежів користувачів

Отже, функціонал віртуального порталу "FoodMarket" був удосконалений шляхом інтеграції додаткових можливостей, спрямованих на оптимізацію досвіду клієнтів у процесі замовлення їжі. Портал забезпечує низку переваг як для кінцевих користувачів, так і для суб'єктів ресторанного бізнесу. До ключових характеристик порталу відносяться інтуїтивно зрозумілий інтерфейс, високий рівень безпеки транзакцій та даних, ергономічність використання, а також оптимізація часових витрат користувачів, що є важливими перевагами його застосування.

Використання віртуального порталу "FoodMarket" надає клієнтам можливість оперативно переглядати асортимент страв (меню), формувати та кастомізувати свої замовлення. Портал інтегрує безпечний та зручний механізм здійснення електронних платежів. Процедура оплати є прозорою та характеризується високим ступенем захисту даних, що дозволяє користувачам ефективно відстежувати статус виконання замовлення без потреби фізичної присутності у закладі харчування.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

Крім цього, функціонал віртуального порталу "FoodMarket" має потенціал для подальшого розширення шляхом інтеграції додаткових сервісів. Критично важливими аспектами функціонування порталу є забезпечення відповідності гігієнічним стандартам під час доставки замовлень, оптимізація клієнтського досвіду (user experience) та підвищення операційної ефективності процесів.

Ергономічний дизайн інтерфейсу користувача оптимізує навігацію, спрощуючи процес пошуку страв, кастомізації замовлень та безперешкодного переходу до процедури оформлення покупки. Структурований та інтуїтивно зрозумілий інтерфейс позитивно корелює із загальним рівнем задоволеності клієнтів.

Інтеграція механізму сканування QR-коду в мобільному застосунку значно спрощує процедури доступу до актуального меню та ініціації процесу оплати. Цей метод взаємодії з інтерфейсом характеризується високою зручністю та ефективністю.

Системи відстеження замовлень у реальному часі та функціонал push-сповіщень підвищують рівень зручності взаємодії клієнтів із сервісом. Користувачі отримують можливість моніторингу статусу виконання своїх замовлень, від етапу підготовки до доставки або самовивозу. Ці сповіщення забезпечують актуальною та цінною інформацією, дозволяючи підтримувати інформованість про зміни у статусі замовлення та здійснювати необхідну взаємодію протягом усього життєвого циклу замовлення.

Імплементація зазначених функціональних покращень, зокрема оптимізація інтерфейсу користувача, системи відстеження замовлень у реальному часі та механізму сповіщень, сприяє забезпеченню вищого рівня клієнтського досвіду. Постійна актуалізація функціоналу на основі аналізу зворотного зв'язку від користувачів та інтеграція новітніх технологічних рішень відіграють ключову роль у підтримці конкурентоспроможності галузі.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

ВИСНОВКИ

В дипломній роботі було виконано повний цикл розробки віртуального порталу типу "FoodMarket", призначеного для автоматизації процесів замовлення їжі та підвищення ефективності функціонування ресторанного бізнесу. Робота охопила як теоретичні аспекти аналізу предметної області, так і практичну реалізацію інструментів системи.

На першому етапі здійснено комплексний аналіз сучасного стану галузі онлайн-замовлень їжі та існуючих платформ, виявлено їхні функціональні переваги та технологічні обмеження. Було встановлено, що більшість наявних рішень не забезпечують гнучкого адміністрування, мають обмежену адаптивність до потреб малого бізнесу та не інтегрують локальні сервіси доставки.

На підставі виявлених проблем визначено мету та мотиваційні чинники створення нового віртуального порталу, зокрема необхідність у зручному, масштабованому та економічно доцільному рішенні для автоматизації процесів прийому та обробки замовлень у сфері громадського харчування.

На етапі проектування було обґрунтовано вибір інструментів розробки, таких як фреймворк Django, система керування базами даних MySQL, мова програмування Python, а також засоби візуалізації та інтегровані середовища розробки. Створено архітектуру системи, побудовано структуру програмного забезпечення та представлено її моделі за допомогою нотації UML та ER-моделей, що дозволило формалізувати логіку взаємодії основних компонентів.

На завершальному етапі реалізовано ключові функціональні модулі системи, зокрема інтерфейси адміністратора та користувача, модулі управління меню, замовленнями та базою даних. Портал забезпечує інтуїтивно зрозумілу навігацію, підтримує операції CRUD та демонструє стабільність у роботі з великими обсягами транзакційних даних.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

Таким чином, розроблений віртуальний портал "FoodMarket" відповідає сучасним вимогам до інформаційних систем у ресторанному бізнесі, має потенціал до подальшого масштабування та адаптації під специфічні потреби різних закладів громадського харчування.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Y. C.Tan, et al., “Automated Food Ordering System with Interactive User Interface Approach“, Faculty of Engineering and Science, University Tunku Abdul Rahman, Malaysia, 2010.
2. N. A.Samsudin, et al., “A Customizable Wireless Food Ordering System with Real time Customer Feedback“, IEEE Symposium on Wireless Technology and Application (ISWTA), 2011
3. Goli, Rishivar Kumar. Contactless food ordering system. International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429 2023.
4. S. N. Cheong, et al., “Enriching Dining Experience with the Multi Touchable Entertainment Applications” Faculty of Engineering Multimedia University, Malaysia, 2010.
5. C. S. Chang, et al., “Development and Implementation of a Restaurant for Customer-Centric Service Using WLAN and RFID Technologies” Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, 2008.
6. X. Yang, et al., “Security and Privacy in RFID and Applications in Telemedicine,” IEEE Communications Magazine, pp. 64-72, 2006.
7. B. Ersin, “An RFID Application for the Disabled: Path Finder,” in Proc. of IEEE International Conference on RFID Eurasia, pp. 1-5, 2007.
8. G. Shang, “RFID Applications Toward Mobile Business,” in Proc. of IEEE International Conference on Management of Mobile Business, pp. 65-65, 2007.
9. Kirti Bhandge, Tejas Shinde, Dheeraj Ingale, Neeraj Solanki, Reshma Totare - “A Proposed System for Touchpad Based Food Ordering System

					БР.ІІІ – 14.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

- Using Android Application”-International Journal of Advanced Research in Computer Science Technology (IJARCST 2015)
10. K. J. Patel et al., "PDA-based Wireless Food Ordering System for Hospitality Industry - A Case Study of Box Hill Institute", Wireless Telecommunications Symposium 2007, pp. 1-8, 2007.
 11. Jun Zeng, Feng Li, Haiyang Liu, Junhao Wen, Sachio Hirokawa, "A Restaurant Recommender System Based on User Preference and Location in Mobile Environment", 5th IIAI International Congress on Advanced Applied Informatics, 2016.
 12. D. Sharma, "A Review of QR code Structure for Encryption and Decryption Process," International Journal of Innovative Science and Research Technology, vol. 2, no. 2, pp. 13-18, 2017.
 13. J. Qianyu, "Exploring the Concept of QR Code and the Benefits of Using QR Code for Companies," Bachelor's Thesis, School of Business and Culture, Lapland University of Applied Sciences, Rovaniemi, Finland, 2014
<https://www.theseus.fi/bitstream/handle/10024/85796/JI%20QIANYU-FINAL%20THESIS.pdf?sequence=1>
 14. M. Sadiku, T.J.Ashaolu, A.Ajayi-Majebi, S.Musa, "Big Data in Food Industry," International Journal of Scientific Advances, vol. 3, no.3, pp. 148 –152, 2020. doi: 10.51542/ijscia.v1i3.5
 15. Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A Software Architect's Perspective. Addison-Wesley.
 16. Fowler, M. (2012). Continuous Integration. Martin Fowler. <https://martinfowler.com/articles/continuousIntegration.html>
 17. Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley.
 18. Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.

					БР.ІІІ – 14.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

19. Richardson, C. (2018). *Microservices Patterns: With Examples in Java*. Manning Publications.
20. Taibi, D., Lenarduzzi, V., & Pahl, C. (2018). "Processes, motivations, and issues for migrating to microservices architectures: An empirical investigation". *IEEE Cloud Computing*, 5(5), 22–32.
21. Namiot, D., & Sneps-Sneppe, M. (2014). "On micro-services architecture". *International Journal of Open Information Technologies*, 2(9), 24–27.
22. Macaulay, L., Keeling, D., & Korpela, M. (2018). "User-centered agile development of a food-ordering application". *Behaviour & Information Technology*, 37(3), 214–231.
23. Pahl, C. (2015). "Containerization and the PaaS cloud". *IEEE Cloud Computing*, 2(3), 24–31.
24. Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
25. Tilley, S., & Huang, S. (2003). "A qualitative assessment of the efficacy of UML diagrams as a form of graphical documentation in aiding program understanding". *Proceedings of the 21st Annual International Conference on Documentation*, 184–191.
26. Chen, L. (2015). "Continuous delivery: Huge benefits, but challenges too". *IEEE Software*, 32(2), 50–54.
27. Kruchten, P. (2004). *The Rational Unified Process: An Introduction (3rd ed.)*. Addison-Wesley.
28. Runeson, P., & Höst, M. (2009). "Guidelines for conducting and reporting case study research in software engineering". *Empirical Software Engineering*, 14(2), 131–164.
29. Kitchenham, B. (2004). *Procedures for performing systematic reviews*. Keele University Technical Report TR/SE-0401.
30. Atkinson, C., & Kuhne, T. (2003). "Model-driven development: A metamodeling foundation". *IEEE Software*, 20(5), 36–41.

31. Hassan, A. E. (2008). "The road ahead for mining software repositories". Proceedings of the 2008 Frontiers of Software Maintenance, 48–57.
32. Wiederhold, G. (1992). "Mediators in the architecture of future information systems". IEEE Computer, 25(3), 38–49.
33. Tilkov, S., & Vinoski, S. (2010). "Node.js: Using JavaScript to build high-performance network programs". IEEE Internet Computing, 14(6), 80–83.
34. Jamshidi, P., Pahl, C., & Mendonça, N. C. (2017). "Microservices migration patterns". Software: Practice and Experience, 50(12), 2224–2251.
35. Fehling, C., Leymann, F., Retter, R., Schupeck, W., & Arbitter, P. (2014). Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications. Springer.
36. Kruchten, P., Lago, P., & Vliet, H. (2006). "Building up and reasoning about architectural knowledge". Quality of Software Architectures, 43–58.
37. Merson, P., Bass, L., & Kazman, R. (2005). "Rapid development of software architecture documentation via architecture decision templates". WICSA 2005.
38. Rittinghouse, J. W., & Ransome, J. F. (2016). Cloud Computing: Implementation, Management, and Security. CRC Press.
39. Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures (Doctoral dissertation, University of California, Irvine).
40. Pressman, R. S., & Maxim, B. R. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.
41. X. U. Hong Zhen, T. Bin, et al., "Wireless Food Ordering System Based on Web Services" Department of Computer Science and Technology, East China Institute of Technology Fuzhou, China, 2009.
42. Rahul Kata rya, Om Prakash Verma, "Restaurant Recommender System Based on Psychographic and Demographic Factors in Mobile Environment",

					БР.ІІІ – 14.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

International Conference on Green Computing and Internet of Things (ICGCIoT),2015.

43. Ashwini Bankar 1, Sheeja S. Suresh, "Intelligent Restaurant - Menu Ordering System "IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 5, 2015.

					БР.ІП – 14.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

ДОДАТКИ

Додаток А

Фрагменти програмних кодів

Код для визначення даних моделей

Клієнт

```
from django.db import models
from datetime import datetime
from django.conf.global_settings import TIME_ZONE # імпорт часового поясу
from django.utils.timezone import utc
current_time = datetime.now()
# Створіть свої моделі тут.
class Customerdetails(models.Model):
    first = models.CharField(max_length=50)
    last = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    password = models.CharField(max_length=50)
    def __str__(self):
        return self.username
class Userqr(models.Model):
    restaurantid = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    status = models.CharField(max_length=50, default="pending")
    def __str__(self):
        return self.restaurantid
class Tablebooking(models.Model):
    restaurantid = models.CharField(max_length=50)
    email = models.CharField(max_length=50)
    tableno = models.CharField(max_length=50)
    memberscount = models.CharField(max_length=50)
    status = models.CharField(max_length=50, default="pending")
    def __str__(self):
        return self.restaurantid
class OrderItem(models.Model):
    """Модель елемента замовлення"""
    price = models.IntegerField(null=True)
    restaurant_id = models.CharField(max_length=100)
    dish_id = models.CharField(max_length=100)
    dish_count = models.CharField(max_length=100)
    table_number = models.CharField(max_length=100)
    dish_name = models.CharField(max_length=100)
    order = models.ForeignKey("Customer.Userorder", on_delete=models.CASCADE)
    def __str__(self):
        return str(self.price)
class Userorder(models.Model):
    """Модель замовлення, що містить кілька елементів замовлення"""
    user_email = models.EmailField(null=True)
    total_cost = models.IntegerField(default=0)
    created_at = models.DateTimeField(auto_now_add=True, null=True)
    def __str__(self):
        return self.user_email
```

```

class PaymentDetails(models.Model):
    tablenumber = models.CharField(max_length=100, null=True)
    resid = models.CharField(max_length=100, null=True)
    Email = models.EmailField()
    cardname = models.CharField(max_length=100, null=True)
    cardno = models.CharField(max_length=100)
    amount = models.CharField(max_length=100)
    expirydate = models.CharField(max_length=100)
    cvv = models.CharField(max_length=100)
class WaitlingList(models.Model):
    """Модель списку очікування"""
    time = models.DateTimeField(auto_now_add=True)
    count = models.IntegerField(default=1)
    table_number = models.CharField(max_length=50)
    user_email = models.EmailField(null=True)
    @property
    def get_time(self):
        now = current_time.utcnow().replace(tzinfo=utc)
        time_diff = now - self.time
        print(time_diff)
        diff = round((60) - (time_diff.total_seconds() / 60))
        return diff if diff > 0 else 0
    def __str__(self):
        return str(self.time)

```

Ресторан

```

from django.db import models
# import uuid
# Створіть свої моделі тут.
class Restaurant(models.Model):
    # id = models.UUIDField(primary_key=True, default=uuid.uuid4, editable=False)
    firstname = models.CharField(max_length=100)
    lastname = models.CharField(max_length=100)
    email = models.CharField(max_length=100)
    password = models.CharField(max_length=100)
    restaurantname = models.CharField(max_length=100)
    qrcode = models.CharField(max_length=100, null=True)
    qrcodepath = models.CharField(max_length=100, null=True)
    def __str__(self):
        return self.restaurantname
class qrcode(models.Model):
    restaurantid = models.CharField(max_length=100)
    restaurantname = models.CharField(max_length=100, null=True)
    qrcode = models.CharField(max_length=100)
    qrcodepath = models.CharField(max_length=100)
    customeremail = models.CharField(max_length=100, null=True)
    status = models.CharField(max_length=100, null=True)
    def __str__(self):
        return self.restaurantid
class addfoodinfo(models.Model):
    restaurantid = models.CharField(max_length=100)
    foodname = models.CharField(max_length=100)
    foodprice = models.CharField(max_length=100)
    def __str__(self):
        return self.restaurantid

```

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: “ Автоматизація та інтеграція засобів і інструментів віртуального порталу типу "FoodMarket" ”

Обсяг пояснювальної записки: 76 аркушів.

Дата закінчення роботи: 10 червня 2025 р.

Підпис студента _____