Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Прикладних інформаційних технологій та електроінженерії
(повна назва факультету)
Кафедра комп'ютерно-інтегрованих технологій
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Розроблення автоматизованої системи для керування ресторацією

Виконав(ла): студент(ка) IV курсу, групи ІКТу-41
спеціальності *151 – Автоматизація та комп'ютерно-інтегровані технології*
(шифр і назва спеціальності)

Прихідко Л.І.

Керівник _____ Золотий Р.З.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ _____
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Микитишин А.Г.
(підпис) (прізвище та ініціали)

Рецензент _____ _____
(підпис) (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)

Кафедра комп'ютерно-інтегрованих технологій
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Микитишин А.Г.
(підпис) (прізвище та ініціали)
«____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва спеціальності)

Студенту Прихідку Любомиру Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення автоматизованої системи для керування ресторацією

Керівник роботи Золотий Роман Захарійович, кандидат технічних наук, доцент кафедри КТ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «31» березня 2025 року №_4/7-238_____

2. Термін подання студентом завершеної роботи 26 червня 2025р.

3. Вихідні дані до роботи Технічні вимоги щодо параметрів системи керування ресторацією

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ
Аналітична частина
Проєктна частина
Спеціальна частина
Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Титульний слайд
Актуальність роботи
Завдання роботи
Основна частина
Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності,			
основи охорони праці			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	20.05.2025	
2.	Підбір джерел по темі роботи	22.05.2025-24.05.2025	
3.	Опрацювання публікацій та збір даних по	24.05.2025-27.05.2025	
	темі роботи		
4.	Виконання роботи згідно мети	27.05.2025-29.05.2025	
5.	Оформлення першого та другого розділів	29.05.2025-31.05.2025	
6.	Оформлення третього розділу	31.05.2025-11.06.2025	
7.	Оформлення четвертого розділу	11.06.2025-13.06.2025	
8.	Підготовка презентації	14.06.2025-15.06.2025	
9.	Підготовка доповіді	16.06.2025-17.06.2025	
10.	Оформлення кваліфікаційної роботи	18.06.2025-19.06.2025	
11.	Нормоконтроль	19.06.2025-20.06.2025	
12.	Перевірка на плагіат	21.06.2025	
13.	Попередній захист кваліфікаційної роботи	21.06.2025	
14.	Захист кваліфікаційної роботи		

Студент _____ Прихідко Л.І.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Золотий Р.З.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота бакалавра складається з пояснювальної записки та графічної частини (ілюстративний матеріал – слайди).

Об’єм графічної частини дипломної роботи становить _____.

Об’єм пояснювальної записки складає ___ друкованих сторінок формату А4.

В роботі використано ___ літературних джерел.

Було розроблено систему замовлення та управління ресторацією. Ця система була розроблена з метою задоволення потреб сучасної ресторанної індустрії та вирішення проблем низької ефективності та схильності до помилок у традиційному способі замовлення їжі.

Система використовує Microsoft Visual Studio як середовище розробки, .NET Framework як фреймворк, а також комбінацію мови програмування C# та бази даних PostgreSQL для забезпечення ефективної обробки серверної логіки. Система складається з низки модулів, зокрема для управління меню та столами, обробки замовлень, управління інформацією про персонал та клієнтів, а також управління виставленням рахунків. Менеджери ресторанів можуть легко додавати, змінювати або видаляти інформацію про страви, офіціанти можуть своєчасно обробляти замовлення, а кухонний персонал може миттєво отримувати замовлення, забезпечуючи своєчасне приготування їжі. Крім того, система спрощує класифікацію та управління типами замовлень, а також обробку платежів за допомогою імітованих транзакцій.

Впровадження цієї системи не лише підвищує операційну ефективність ресторану, але й значно покращує враження клієнтів від замовлень.

Ключові слова: АВТОМАТИЗОВАНА СИСТЕМА, РЕСТОРАЦІЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, БАЗА ДАНИХ, ЗАПИТ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 6

ВСТУП 7

1. АНАЛІТИЧНА ЧАСТИНА 8

1.1. Аналіз вимог до створення системи керування ресторацією 8

1.2. Аналіз бізнес-процесів ресторану 12

2.3. Аналіз функціональних вимог до системи 13

2. ПРОЄКТНА ЧАСТИНА 15

2.1. Проектування системи, архітектурний дизайн 15

2.2. Загальний функціональний дизайн 16

2.3 .Функціональний дизайн модуля 19

2.4. Проектування бази даних 26

3 СПЕЦІАЛЬНА ЧАСТИНА 31

3.1 Впровадження системи 31

3.2. Реалізація входу 32

3.3. Впровадження категорії 34

3.4. Впровадження продуктів 37

3.5. Впровадження персоналу 38

3.6. Реалізація таблиць 39

3.7. Виконання замовлень 40

3.8. Реалізація кухні 45

3.9. Результати тестування функціональності системи управління 48

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ 51

4.1 Ергономічні проблеми безпеки життєдіяльності при роботі за комп'ютером..... 51

4.2 Організація безпечної роботи електроустаткування задіяного при роботі системи електронного навчання 54

ВИСНОВКИ..... 57

ПЕРЕЛІК ПОСИЛАНЬ 58

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AS/RS (англ. Automated Storage and Retrieval System) - Автоматизовані системи зберігання та пошуку.

CIMLAB (англ. Computer Integrated Manufacturing Laboratory) - Комп'ютерно-інтегрована виробнича лабораторія.

BD (англ. Data Base) – база даних.

SQL (англ. Structured Query Language) – структурована мова запитів.

ВСТУП

У роботі розглядається проектування та впровадження веб-системи самообслуговування та системи управління рестораном. У контексті сучасного суспільства, постійний розвиток технологій призвів до зростаючої схильності до інтелектуальних систем обслуговування, особливо яскравим прикладом чого є ресторанне обслуговування. Інтелектуальна система замовлення сприяє задоволенню клієнтів, оскільки 76% споживачів віддають перевагу кіоскам самообслуговування для виконання замовлень. Хоча деякі ресторани високої кухні продовжують дотримуватися традиційної практики ручного замовлення, модель ресторанів, яка дозволяє клієнтам робити замовлення розумно, стає домінуючою тенденцією в галузі. Сучасна ресторанна індустрія зазнає жорсткої ринкової конкуренції, і підвищення ефективності обслуговування та задоволеності клієнтів стало ключовим фактором довгострокової стійкості ресторанів. Традиційний підхід до замовлення, який часто передбачає тривалі періоди очікування та нестабільну якість обслуговування, часто пояснюється людськими помилками та неефективністю. Отже, розробка ефективної, точної та зручної системи замовлення не тільки оптимізує враження клієнтів, але й підвищує операційну ефективність та економічні вигоди ресторанів.

1. АНАЛІТИЧНА ЧАСТИНА

1.1. Аналіз вимог до створення системи керування ресторацією

Поширення смартфонів та інтернет-технологій призвело до зростання попиту клієнтів на зручність та швидкий час реагування в індустрії громадського харчування. Цифрові системи замовлень можуть забезпечувати більш персоналізовані та ефективні послуги завдяки оновленню інформації в меню в режимі реального часу, миттєвій обробці замовлень та інтелектуальному аналізу даних клієнтів [1]. Крім того, такі системи можуть допомогти ресторанам точно отримувати бізнес-дані та оптимізувати управління запасами та стратегії продажів. Впровадження планшета самообслуговування для замовлення їжі задовольняє потребу клієнтів у легкому замовленні їжі на інтелектуальному пристрої, що скорочує час очікування. клієнтів у години пік. Системи самообслуговування дозволяють операторам ресторанів ефективно боротися з нестачею робочої сили. Водночас це також підвищує економічну ефективність та дохід бізнесу, дозволяючи клієнтам отримати кращий досвід харчування [2].

Хоча інтелектуальні системи замовлення широко використовуються в ресторанній індустрії, якість цих систем у кожному ресторані різна. Деякі системи були розроблені без урахування звичок інтерфейсу клієнтів, безпеки облікових записів клієнтів та ємності бази даних для зберігання даних. Це призвело до ігнорування зручності керування та експлуатації інтерфейсу користувачами. Отже, ці системи потребують удосконалення, щоб задовольнити потреби як клієнтів, так і ресторанів. Метою цієї дипломної роботи є розробка та впровадження повнофункціональної інтелектуальної системи замовлення, яку можна використовувати для задоволення потреб як клієнтів, так і ресторанів. Це буде досягнуто шляхом аналізу вподобань клієнтів та потреб ресторанів.

Конкретні цілі дослідження такі:

1. Мета полягає в підвищенні ефективності обслуговування шляхом зменшення кількості помилок, пов'язаних з ручним керуванням, пришвидшення

обслуговування та скорочення часу очікування клієнтів завдяки автоматизації процесів замовлення та обробки замовлень.

2. Мета полягає в тому, щоб підвищити задоволеність та лояльність клієнтів шляхом надання зручного інтерактивного інтерфейсу, який дозволяє клієнтам легко переглядати меню, подавати замовлення та відстежувати статус замовлення в режимі реального часу.

3. Інтегровані функції аналізу даних системи допомагають менеджерам ресторанів розуміти дані про продажі, оптимізувати розподіл меню та управління запасами, а також скорочувати витрати.

Впровадження автоматизованих систем замовлення та управління ресторанами може ефективно розширити управлінські можливості ресторанів, дозволяючи їм приймати обґрунтовані рішення. Перевагою цієї моделі є зниження витрат на оплату праці та мінімізація потреби клієнтів в офіціантах [3]. Ефективність роботи ресторанів, які використовують інтелектуальні системи замовлення та управління, значно підвищиться. Цінність, яку це приносить, набагато перевищує початкові інвестиційні витрати, що робить її оптимальним вибором для ресторанних підприємств у сфері інформаційних технологій.

У цьому розділі викладено вимоги до системи, проаналізовано поточні бізнес-процеси та доцільність реалізації системи, а також описано основні характеристики та загальну функціональність системи.

Бізнес-процес ресторану традиційної моделі ручного замовлення.

Значна частина малих та середніх ресторанів використовують модель ручного бухгалтерського обліку, як показано на рисунку 1.1.

Після прибуття до ресторану офіціант супроводжує клієнта до його столика та розсаджує його. Офіціант пропонує клієнту меню та записує страви, замовлені клієнтом. Після того, як клієнт підтвердить замовлення, воно передається шеф-кухарю та іншим співробітникам. Шеф-кухар приготує страви відповідно до замовлення, а офіціант доставити страви до столу, вказаного клієнтом.

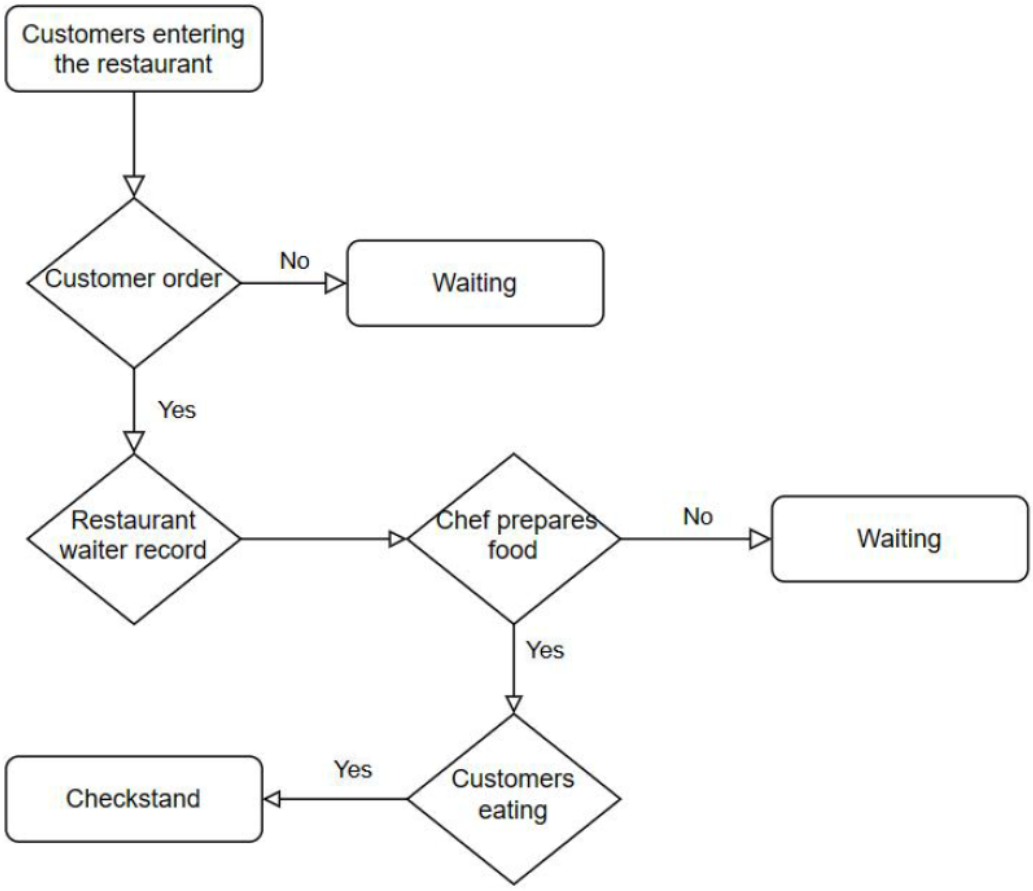

Рисунок 1.1 - Блок-схема бізнес-процесу ручного замовлення в ресторані.

Після того, як клієнт закінчить трапезу, він йде до каси для оплати рахунку, або ж клієнт проходить через офіціанта для оплати рахунку. Цей традиційний метод може призвести до деяких помилок під час пікових годин прийому їжі, що може бути пов'язано з низкою факторів, включаючи велику кількість замовлень та обмежену кількість персоналу. Наприклад, можуть виникнути помилки в подачі неправильної їжі, пропуску страв та інших невдалих кроків. Такі помилки можуть призвести до невдоволення клієнтів, що вказує на необхідність впровадження нової моделі бізнес-процесів.

Бізнес-процеси ресторану в режимі розумного замовлення.

В епоху інформації впровадження інтелектуальної системи замовлень може вирішити численні недоліки, пов'язані з традиційним харчуванням. Блок-схема бізнес-процесу інтелектуальної системи замовлень представлена на рисунку 1.2.

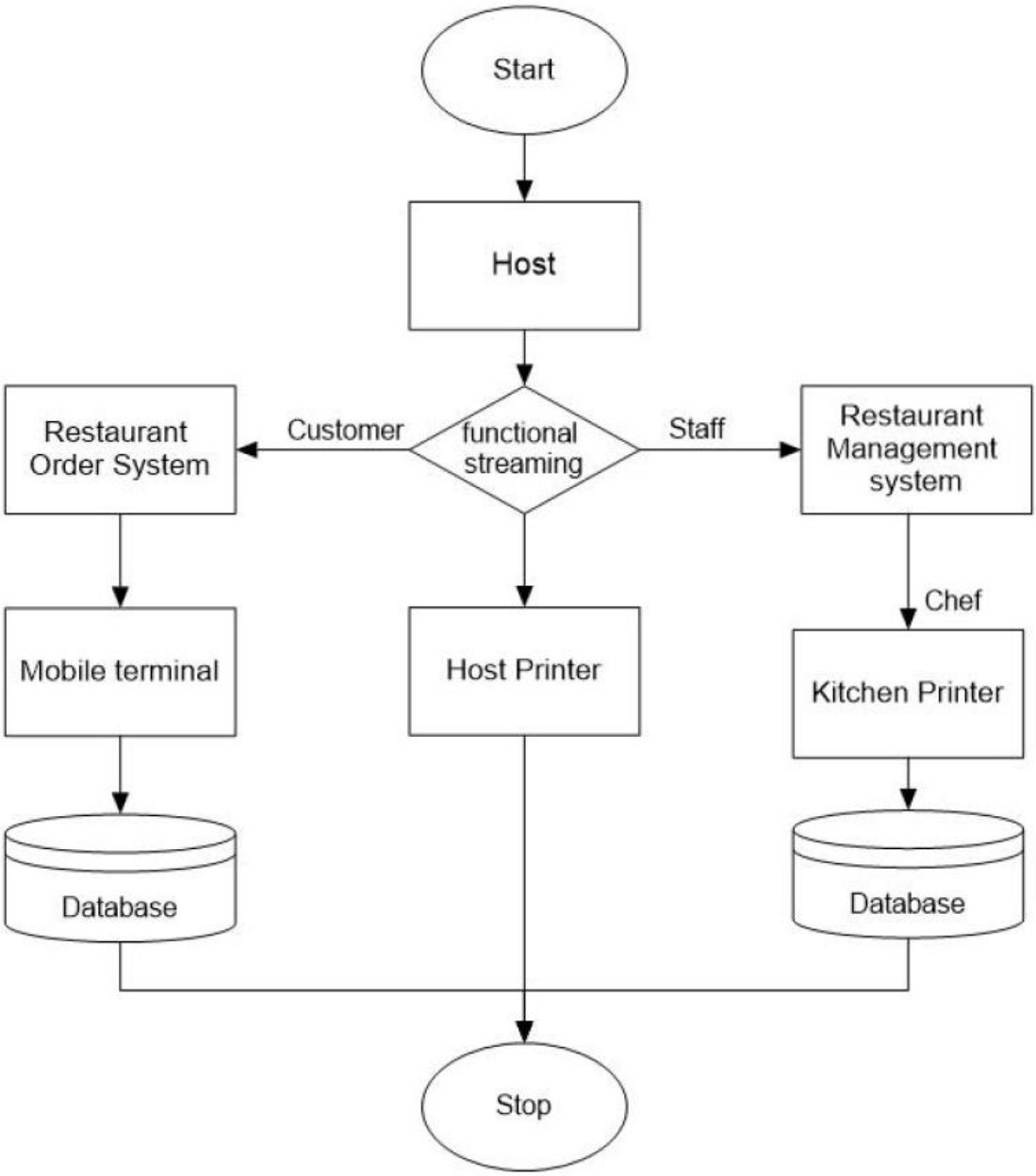

Рисунок 1.2 - Блок-схема інтелектуального замовлення в ресторані.

На рисунку 1.2 термін «Хост» використовується для позначення центрального вузла всієї системи, який відповідає за координацію та управління підсистемами та процесами. Система отримує дані від системи замовлень ресторану та передає інформацію про замовлення на кухонний принтер. Крім того, вона може відповідати за управління іншими адміністративними завданнями в ресторані, такими як управління запасами, звітність про продажі тощо. Система замовлень ресторану являє собою інтерфейс, через який клієнти розміщують замовлення на їжу. Цифрове меню, доступне через смарт-пристрій.

Система збирає інформацію про замовлення клієнта та передає її до смарт-системи управління рестораном. Термін «мобільний термінал» використовується для опису мобільного пристрою, що використовується клієнтом, такого як пристрій самообслуговування в ресторані. Мобільний термінал дозволяє клієнтам переглядати меню, подавати замовлення та здійснювати платежі. Зручність та гнучкість, що забезпечуються мобільними терміналами, мають особливе значення для покращення обслуговування клієнтів. Важливо визнати, що клієнти представляють собою початок і кінець процесу. Клієнт використовує систему замовлень ресторану, щоб розмістити своє замовлення, пройти весь процес замовлення та, зрештою, спожити замовлену їжу. Кухонний принтер отримує інформацію про замовлення від інтелектуальної системи управління рестораном, яка потім відображається шеф-кухарю. Шеф-кухар відповідає за приготування їжі відповідно до часової послідовності на основі роздрукованої інформації про замовлення. Це ключ до ефективності роботи кухні, оскільки безпосередньо впливає на швидкість і точність обробки замовлень на кухні. Нарешті, головний принтер можна використовувати для друку рахунків за замовлення або інших документів, що стосуються роботи ресторану.

1.2. Аналіз бізнес-процесів ресторану

Цінність аналізу бізнес-процесів ресторану полягає в можливості отримати всебічне розуміння різних аспектів роботи ресторану та визначити ключові моменти та потенційні проблеми в них. Зокрема, бізнес-процес такого ресторану можна описати наступним чином:

1. Доступ до основного інтерфейсу здійснюється клієнтами через обладнання для самообслуговування та замовлення їжі.

2. Виберіть потрібні страви з системи замовлень та відправте замовлення.

3. Інформація про замовлення передається на кухню в режимі реального часу, де вона підтверджується та обробляється. Після цього офіціант може перевірити статус замовлення.

4. Після того, як замовлення буде виконано, офіціант доставить його до столу клієнта.

5. Після того, як клієнт закінчить свою трапезу, офіціант оновлює статус замовлення, а потім перевіряє та підтверджує рахунок.

6. Після цього клієнт залишає ресторан.

2.3. Аналіз функціональних вимог до системи

Відповідно до вищезазначених бізнес-процесів ресторану, функціональні модулі системи зображені на рисунку 1.3.

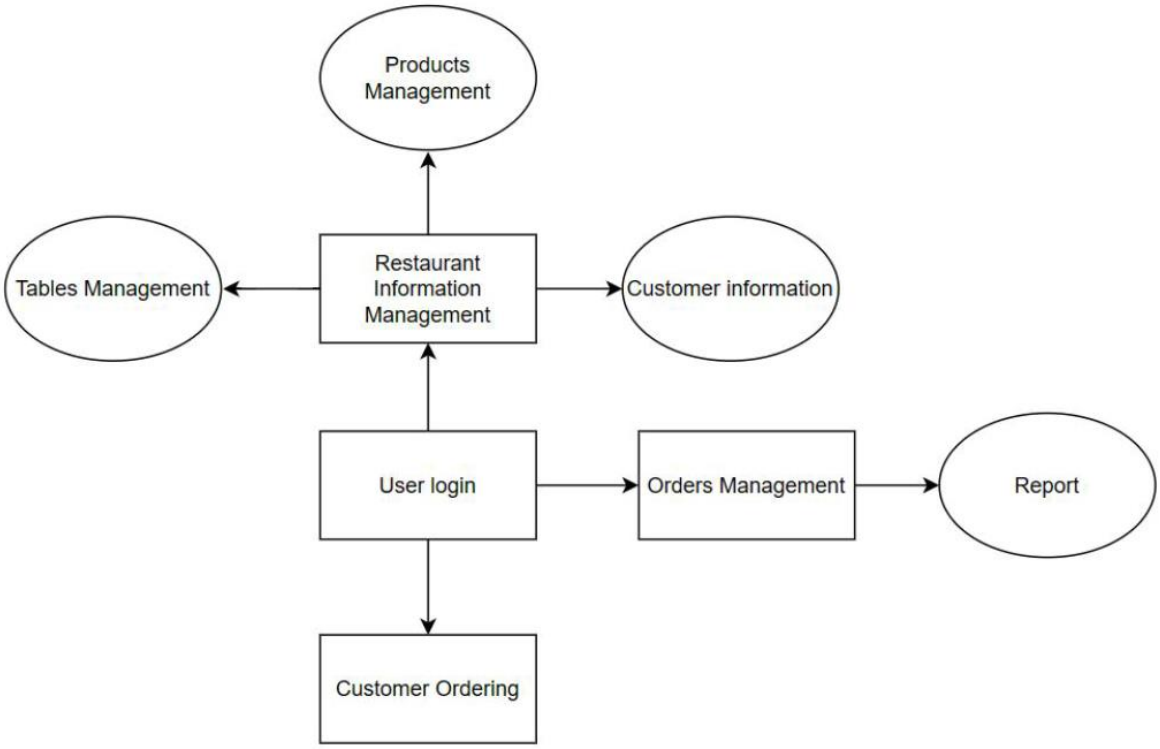

Рисунок 1.3 - Модуль системних функцій.

Після того, як клієнт сяде за стіл, він може отримати доступ до системи замовлень через свій мобільний пристрій та ввести відповідний номер столу. Система розрахує загальну вартість відповідно до різних страв, що зручно для клієнтів для контролю загальної вартості. Після оплати замовлення буде передано до системи управління рестораном. Потім шеф-кухар налаштує страви відповідно до останньої інформації про розподіл страв у хронологічному

порядку. Після оформлення замовлення, після завершення, шеф-кухар змінить статус замовлення в системі, щоб відобразити це. Система автоматично реєструє замовлення та генерує звіти відповідно до вказівок адміністратора. Крім того, менеджери можуть використовувати систему для редагування та додавання нових страв, редагування та керування інформацією про столи та клієнтами.

Аналіз вимог до продуктивності системи.

Під час процесу розробки системи важливо провести ретельний аналіз та визначити різні показники продуктивності, яким система повинна відповідати під час виконання. За допомогою процесу аналізу вимог до продуктивності системи можна визначити стандарти продуктивності, яким система повинна відповідати. Мета полягає в тому, щоб гарантувати, що система може працювати ефективно та стабільно в реальних умовах використання та забезпечувати задовільний користувацький досвід. Аналіз вимог до продуктивності цієї системи охоплює такі аспекти:

1. Час відгуку: Час відгуку системи на дії користувача повинен бути менше 2 секунд, щоб забезпечити безперебійну роботу користувача.

2. Пропускна здатність: Система повинна бути здатною обробляти велику кількість одночасних запитів у періоди пікового навантаження та мати можливість обробляти щонайменше 200 одночасних користувачів.

3. Надійність: Система повинна залишатися онлайн 99,9% часу протягом щонайменше одного року, щоб забезпечити безперервність бізнесу.

4. Узгодженість даних: Забезпечте своєчасне оновлення даних замовлення, щоб уникнути плутанини із замовленнями.

2. ПРОЄКТНА ЧАСТИНА

2.1. Проектування системи, архітектурний дизайн

Ця система використовує багаторівневу архітектурну модель для забезпечення високого ступеня модульності та простоти обслуговування. Система складається з чотирьох основних рівнів: рівня інтерфейсу користувача, рівня бізнес-логіки, рівня доступу до даних та рівня бази даних.

Рівень користувацького інтерфейсу визначається як інтерфейс, через який користувачі взаємодіють із системою. Програмне забезпечення надає інтерфейс для взаємодії користувачів, включаючи модулі для входу, управління продуктами, обробки замовлень, управління кухнею та управління виставленням рахунків. Конструкція цього програмного забезпечення базується на технології Windows Forms, що забезпечує кросплатформну сумісність та швидку адаптацію.

Рівень бізнес-логіки відповідає за реалізацію бізнес-логіки системи. Рівень бізнес-логіки відповідає за перевірку замовлень, розрахунок цін та оновлення статусу системи. Рівень бізнес-логіки служить провідником між рівнем інтерфейсу користувача та рівнем доступу до даних, забезпечуючи належну обробку даних та ефективне виконання операцій.

Рівень доступу до даних відповідає за безпосередню взаємодію з базою даних, включаючи операції додавання, видалення, перевірку та модифікацію даних. Рівень доступу до даних відповідає за безпосередню взаємодію з базою даних, включаючи операції додавання, видалення, перевірку та модифікацію даних. Для операцій з базою даних використовується фреймворк ADO.NET, що сприяє ефективній обробці даних та управлінню винятками.

Рівень бази даних відповідає за зберігання та пошук даних. Цей рівень відповідає за зберігання всіх постійних даних, що стосуються системи, включаючи інформацію про користувачів, каталоги продуктів, записи замовлень та інші подібні дані. База даних PostgreSQL використовується завдяки своєму

відкритому коду, економічній ефективності та надійним можливостям обробки транзакцій.

Система використовує багаторівневий механізм автентифікації для гарантування безпеки даних та легітимності користувачів. Швидкість відгуку та можливості обробки системи покращуються завдяки оптимізації SQL-запитів та використанню ефективних стратегій кешування.

2.2. Загальний функціональний дизайн

Управління користувачами та дизайн основних функцій інтерфейсу.

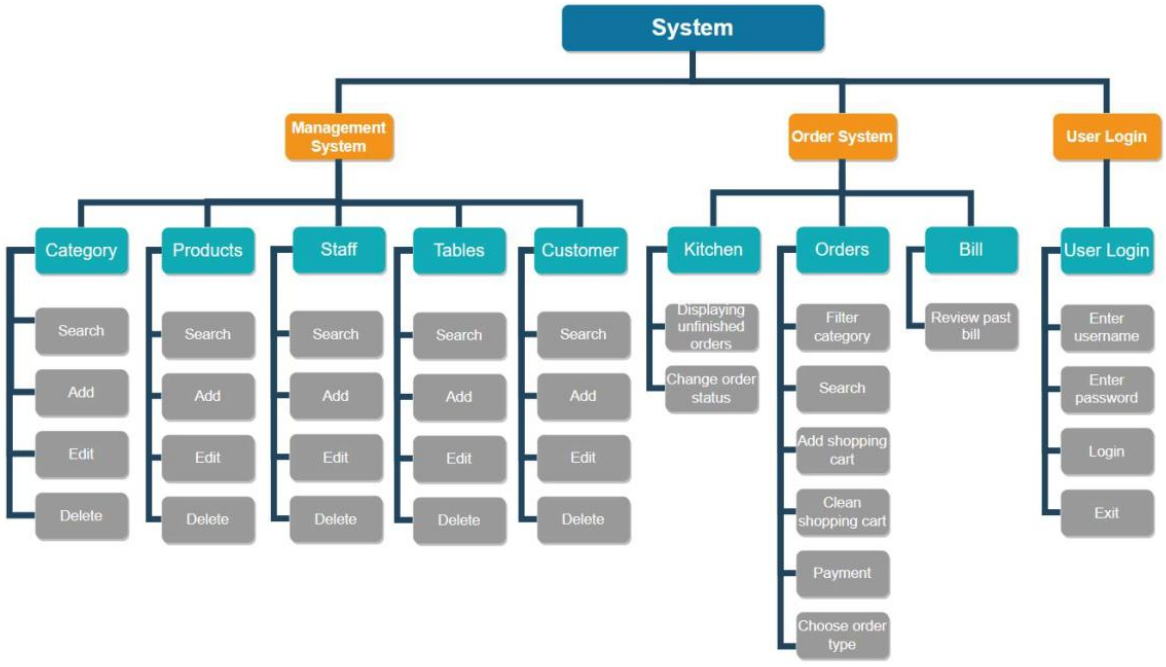

Рисунок 2.1 - Діаграма структури функцій системи

На рисунку 4 модуль входу призначений для спрощення автентифікації користувачів. Щоб отримати доступ до системи, користувачі повинні ввести ім'я користувача та пароль. Система здатна підтримувати вхід користувачів з різним рівнем привілеїв, включаючи тих, кому призначено права адміністратора та клієнта.

Адміністратор може класифікувати страви в меню ресторану. Крім того, адміністратор має право додавати, редагувати та видаляти інформацію, що стосується страв, включаючи назву страви, опис, ціну та зображення. Адміністратору надано право вводити, змінювати або видаляти персональні дані співробітників.

На столах відображається інформація про статус кожного столу в ресторані, зокрема, чи вільний стіл, заброньований чи використовується. Крім того, фіксується кількість столів та кількість людей, яких може розмістити кожен стіл.

Інтерфейс замовлень служить інтерфейсом замовлення клієнтів, дозволяючи клієнтам замовляти страви та надсилати замовлення на кухню для обробки.

Кухня відображає статус замовлення кожного столу в режимі реального часу та надає інформацію про поточне замовлення, яке потрібно обробити. Це включає деталі замовлення, час замовлення та статус обслуговування. Після завершення замовлення шеф-кухар може змінити його статус.

Модуль управління клієнтами відповідає за збереження особистої інформації клієнтів, включаючи їхнє ім'я, номер телефону, дані облікового запису члена та баланс.

Функція виставлення рахунків дозволяє ресторану переглядати інформацію про всі виконані замовлення за попередні періоди.

Діаграми варіантів використання системи для різних сценаріїв.

На рисунку 2.2 представлено діаграму варіантів використання, яка ілюструє основні функції ресторанної системи та взаємодію між різними ролями користувачів. Діаграма варіантів використання окреслює три основні варіанти використання системи та три основні ролі користувачів: клієнт, шеф-кухар та менеджер.

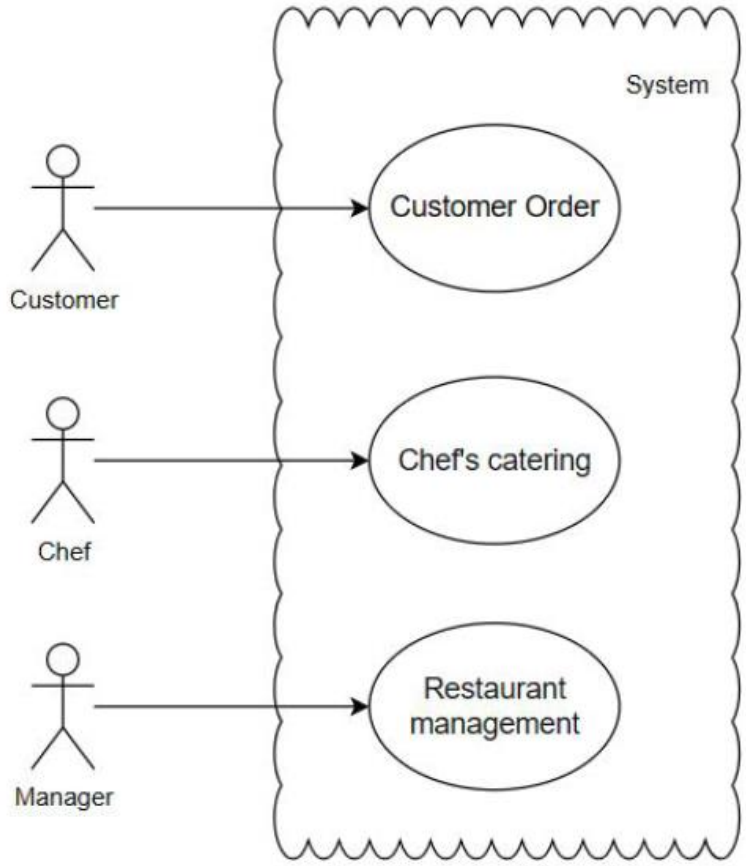

Рисунок 2.2 - Діаграма варіантів використання

Кінцевими користувачами системи є клієнти, які мають можливість замовляти їжу через систему. Цей варіант використання стосується вибору клієнтом пунктів з меню, введення інформації про замовлення та підтвердження замовлення.

Шеф-кухар – це користувач, відповідальний за підготовку та виконання замовлення клієнта. Шеф-кухар може переглядати замовлення, які ще не виконані, та готувати й створювати страви на основі інформації, наданої в замовленні. Цей варіант використання передбачає отримання шеф-кухарем сповіщення про замовлення, перегляд деталей замовлення та підтвердження його завершення.

Менеджер – це користувач, відповідальний за роботу та управління рестораном. Менеджер може здійснювати щоденне управління рестораном через

систему. Цей варіант використання стосується управління менеджером меню, обробки інформації про співробітників та перегляду бізнес-даних.

2.3 .Функціональний дизайн модуля

Дизайн входу

Модуль входу є початковою точкою доступу до системи ресторану та служить для автентифікації користувачів, тим самим гарантуючи, що доступ до системи можуть отримати лише авторизовані особи. Модуль спрощує введення імен користувачів та паролів з метою безпечного входу, а також пропонує можливість виходу з системи. Контент, який відображається користувачам з різними ролями під час входу, відрізняється. Наприклад, адміністратор ресторану бачитиме систему управління рестораном, тоді як клієнт бачитиме систему замовлень ресторану.

1. Функціональні вимоги до входу.

Вхід користувача. Після доступу до екрана входу користувачам пропонується ввести своє ім'я користувача та пароль. Системі необхідно перевірити введене ім'я користувача та пароль. Після успішної перевірки користувачеві надається доступ до основного інтерфейсу системи. У разі невдалої перевірки відображається повідомлення про помилку.

Вихід. Рекомендується передбачити кнопку з написом «Вихід», яка дозволить користувачам безпечно вийти з системи в будь-який час. Після натискання кнопки «Вихід» вікно програми закривається.

2. Дизайн інтерфейсу користувача для входу.

У верхній частині інтерфейсу відображається логотип або назва системи. У центрі знаходиться область введення, що включає два текстові поля: одне для введення імені користувача, а інше для введення пароля. Під текстовим полем розташовані дві кнопки: одна — кнопка «Вхід» для надсилання інформації, а інша — кнопка «Вийти» для закриття програми.

3. Керування даними.

Процес перевірки. Ім'я користувача та пароль, надані користувачем, будуть зіставлені з даними, що зберігаються в базі даних. Необхідно забезпечити шифрування паролів, що зберігаються в базі даних, для безпеки. Операції з базою даних використовують безпечні запити, щоб уникнути атак SQL-ін'єкцій. (Gonzalez and Jung, 2019)

Обробка помилок. Якщо введене ім'я користувача не існує або пароль не збігається, система повинна відобразити повідомлення про помилку, яке запитує користувача. Слід вжити заходів безпеки, щоб обмежити кількість спроб входу, щоб уникнути злому методом грубої сили.

Керування сеансами. Після успішного входу система повинна створити безпечний сеанс для користувача. Інформація про сеанс повинна керуватися на стороні сервера, а на стороні клієнта має зберігатися лише ідентифікатор сеансу.

Дизайн категорій

Модуль категорій пропонує можливість контролю за управлінням категоріями їжі та напоїв у контексті меню ресторану. Модуль дозволяє адміністраторам додавати, змінювати та видаляти категорії, щоб забезпечити актуальність меню.

1. Функціональні вимоги до категорій.

Додати категорію. Система надає інтерфейс, через який адміністратори можуть додавати нові категорії. Після натискання кнопки «Додати» користувачі будуть перенаправлені до нового вікна, де вони зможуть ввести назву нової категорії. Рекомендується, щоб система передбачила кнопки «Зберегти» та «Закрити». Кнопка «Зберегти» ініціює процес додавання нової категорії до бази даних, тоді як кнопка «Закрити» завершує його процес без збереження. Після успішного збереження система відобразить повідомлення про підтвердження, яке свідчить про успішне завершення операції.

Пошук за категорією. Рекомендується, щоб система включала функцію пошуку для швидкого пошуку категорій. Користувач може ввести назву в поле пошуку, що призведе до динамічної фільтрації таблиці, щоб відображати лише ті записи, які відповідають введеній назві.

Змінити категорію. Система повинна дозволяти адміністраторам змінювати існуючі категорії. Після вибору значка «Редагувати» в таблиці система відкриє той самий екран, що й для додавання категорії, але з наявною назвою категорії. Користувач може змінити назву та зберегти оновлення, натиснувши кнопку «Зберегти», що оновить категорію в базі даних.

Видалити категорію. Рекомендується, щоб кожен запис категорії містив значок видалення. Після вибору значка «Видалити» вибрану категорію буде видалено з бази даних. Щоб запобігти випадковій втраті даних, система повинна запитувати підтвердження перед видаленням категорії.

2. Дизайн інтерфейсу користувача категорії.

Інтерфейс категорії. Сторінка категорії містить текстове поле для пошуку та фільтрації списку категорій. Сітка відображає існуючі категорії зі стовпцями для «ідентифікатора» та «назви» та значками дій для редагування та видалення. У верхній частині сторінки розташована кнопка «Додати», після натискання якої розпочинається процес додавання нової категорії.

Інтерфейс додавання/редагування категорії. модальне діалогове вікно з простою формою, що включає підписи та текстові поля для введення назв категорій. Інтерфейс також містить дві кнопки з позначками «Зберегти» та «Закрити».

3. Управління даними категорії.

Коли додається нова категорія, система повинна створити новий запис у таблиці. Коли існуюча категорія змінюється, відповідний запис у таблиці має бути оновлений. Видалення категорії має призвести до видалення відповідного запису в таблиці.

Дизайн продукції.

Модуль «Продукти» є ключовим компонентом системи управління рестораном, який служить для контролю різноманітного асортименту продуктів, що пропонуються рестораном, включаючи їжу та напої. Цей модуль дозволяє адміністратору додавати, змінювати та видаляти інформацію про продукти. Більшість функціональних можливостей модуля «Продукти» аналогічні модулю

«Категорії». З огляду на модуль «Категорії», модуль «Продукти» потребує додавання кількох нових функцій. Після натискання кнопки «Додати» користувач потрапляє до нового вікна, де він може ввести назву та ціну продукту, вибрати зображення та вибрати категорію з випадаючого меню.

Дизайн персоналу.

Модуль «Персонал» служить для запису інформації про співробітників ресторану, включаючи їхнє ім’я, номер телефону та посаду. Цей модуль дозволяє адміністратору додавати, змінювати та видаляти інформацію про співробітників. Більшість функціональних можливостей модуля «Персонал» аналогічні модулю «Категорія». Також варто зазначити, що у вікні «Додати» користувач має можливість ввести ім’я співробітника, номер телефону та вибрати посаду з випадаючого меню.

Дизайн столів.

Модуль «Столи» служить для запису інформації, що стосується столів у ресторані. Це включає, але не обмежується, номери столів, назви столів та максимальну кількість людей, яку може вмістити кожен стіл. Цей модуль дозволяє адміністраторам додавати, змінювати та видаляти інформацію про столи. Більшість функціональних можливостей модуля «Столи» аналогічні модулю «Категорії».

Дизайн замовлень.

Модуль замовлень пропонує інтуїтивно зрозумілий та ефективний інтерфейс для клієнтів, щоб вони могли замовляти їжу, підтримуючи як харчування в ресторані, так і їжу на винос. Модуль дозволяє клієнтам переглядати детальну інформацію про різні страви, вибирати номери столів, керувати кошиками для покупок і, нарешті, відправляти замовлення.

1. Функціональні вимоги до замовлень

Відображення меню та фільтрація за категоріями. У лівій частині інтерфейсу відображаються всі категорії страв (наприклад, напої, десерти, основні продукти харчування тощо). Натискаючи на різні категорії, клієнти можуть фільтрувати страви відповідних категорій. Крім того, інтерфейс містить

опцію «Усі категорії», яка дозволяє клієнтам переглядати всі доступні страви. Крім того, клієнти можуть ввести назву страви в поле пошуку, щоб швидко знайти певний елемент.

Варіанти замовлення. Клієнт може вибрати один із двох варіантів замовлення: «Їсти в закладі» або «На винос». Після вибору опції «Їсти в закладі» система запропонує користувачеві вибрати номер столу. Цей номер буде взято з бази даних столів, що зберігається на серверній частині. Після того, як клієнт вибере потрібний номер столу, він відобразиться у призначеній області. У випадку, якщо користувач вибере опцію «Їжа на винос», тип замовлення автоматично встановлюється на «Їжа на винос».

Відображення та керування деталями замовлення. Страви, обрані клієнтом, відображатимуться у правій частині інтерфейсу, включаючи назву страви, кількість, ціну за одиницю та загальну ціну страви. Загальна ціна всіх обраних страв відображатиметься внизу інтерфейсу. Кнопка «Очистити» дозволяє клієнту видалити всі товари з кошика. Кнопка «Оплатити» використовується для передачі замовлення. Після натискання кнопки інформація про замовлення передається на кухню, де її бачить шеф-кухар, який потім може переглянути та приготувати страву. Вартість буде списана з балансу рахунку клієнта.

2. Дизайн інтерфейсу користувача для замовлень.

Основний інтерфейс. Інтерфейс містить панель навігації для категоризації страв, панель пошуку, область відображення страв та область детальної інформації про кошик для покупок. Область відображення страв відображає зображення та основні дані про страви, а також дозволяє користувачам вибирати товари для кошика, натискаючи на них. В області кошика для покупок перелічено вибрані страви та надано можливість змінити кількість, відображаючи ціну та загальну суму кожного товару.

Інтерфейс вибору номера столу. Коли клієнт вибирає опцію «Їсти в закладі», відображається спливаюче вікно з доступними номерами столів для вибору клієнтом. Номер столу відображається візуально відповідно до макета, і клієнт може вибрати його, натиснувши на нього.

Дизайн кухні

Модуль «Кухня» надає кухонному персоналу можливість контролювати та відстежувати замовлення клієнтів. Модуль отримує замовлення та відображає відповідну інформацію, що дозволяє кухонному персоналу ефективно створювати страви та оновлювати статус замовлення після його завершення.

1. Функціональні вимоги до кухонного модуля.

Отримання замовлень. Після завершення замовлення клієнту буде запропоновано перейти до етапу оплати. На цьому етапі інформація про замовлення буде автоматично передана до кухонного модуля. Вкрай важливо, щоб кухонний модуль оновлювався в режимі реального часу, щоб шеф-кухар міг переглядати найактуальнішу інформацію про замовлення.

Управління замовленнями. Кожен блок інформації про замовлення повинен містити номер столу, час та тип замовлення (наприклад, «Їсти в закладі» або «На винос»). Щоб кухня готувала в порядку замовлення клієнта, замовлення слід розташовувати в хронологічному порядку.

Завершення замовлення. Рекомендується розміщувати кнопку «Завершити» поруч із кожним замовленням. Після завершення процесу приготування кухар повинен натиснути кнопку «Завершити», що оновить статус замовлення в системі, вказуючи на його завершення. Після завершення процесу приготування інформацію про замовлення слід видалити з поточного інтерфейсу, тим самим свідчачи про завершення кухонної роботи.

2. Дизайн інтерфейсу користувача кухні.

Область відображення замовлень: має бути реалізований інтерфейс форми списку, що відображає номер столу, час замовлення та тип замовлення. Важливо, щоб була чітка візуальна ідентичність встановлено для розрізнення замовлень з різними статусами, такими як «виконується» або «завершено».

Кнопки дій. Кожен блок інформації про замовлення містить кнопку з написом «Готово», яка використовується для позначення завершення створення замовлення. Важливо, щоб кнопки були розроблені таким чином, щоб їх було легко розпізнати та використовувати, щоб забезпечити швидку відповідь.

3. Управління даними кухні.

Отримання замовлень. Система повинна мати можливість отримувати та зберігати дані замовлень з модуля «Замовлення».

Оновлення замовлення. Завершені замовлення необхідно оновлювати в базі даних, щоб забезпечити синхронізацію статусу замовлення в усіх модулях.

Дизайн клієнта.

Модуль «Клієнт» записує інформацію про членство клієнта, включаючи ідентифікатор клієнта, ім'я клієнта, номер членства клієнта, номер телефону клієнта та баланс. Цей модуль дозволяє адміністратору додавати, змінювати та видаляти інформацію про клієнта. Більшість функціональних можливостей модуля «Клієнт» аналогічні модулю «Категорія».

Дизайн рахунків.

Модуль виставлення рахунків відповідає за відображення всіх виконаних замовлень, забезпечення ресторанів точними фінансовими записами та відстеження обслуговування клієнтів.

1. Функціональні вимоги до рахунку.

Запис замовлення. Цей модуль отримує замовлення з модуля Кухня, статус яких змінюється на «Завершено». Після оновлення статусу замовлення в модулі Кухня на «Завершено» відповідна інформація про замовлення має відображатися в модулі Рахунок у режимі реального часу.

Відображення замовлення. Відображення має містити такі деталі замовлення: номер столу, тип замовлення, статус замовлення та загальну суму замовлення. Інтерфейс повинен представляти всю інформацію чітко та розбірливо, що полегшує читання та перевірку даних.

Організація даних. Для швидкого огляду нещодавно виконаних замовлень рекомендується сортувати замовлення в хронологічному порядку відповідно до часу їх виконання. Для швидкої ідентифікації конкретних замовлень рекомендується передбачити фільтри, що дозволять користувачам шукати замовлення за їх типом, номером столу або статусом.

2. Дизайн інтерфейсу користувача для рахунків

Для відображення всіх виконаних замовлень, включаючи серійний номер, номер столу, тип замовлення (наприклад, «Їсти в закладі» або «На винос»), статус (наприклад, «Завершити») та загальну суму, слід використовувати формат списку. Список слід розробити таким чином, щоб встановити чітку візуальну ієрархію, з використанням відповідних кольорів та розмірів шрифту для оптимізації взаємодії з користувачем.

3. Управління даними рахунків

Відстеження замовлень. Вкрай важливо, щоб усі виконані замовлення були точно зареєстровані та збережені.

Цілісність даних. вкрай важливо підтримувати узгодженість і точність даних, щоб запобігти будь-яким несанкціонованим операціям зміни або видалення.

Резервне копіювання та відновлення. Вкрай важливо впровадити систему регулярного резервного копіювання даних замовлень, щоб гарантувати швидке відновлення цих даних у разі збою системи.

2.4. Проектування бази даних

Концептуальний дизайн сутності бази даних

На рисунку 2.3, у таблиці User, userID – це атрибут сутності User, який служить первинним ключем для унікальної ідентифікації кожного користувача в базі даних. loginName – це ім'я користувача для входу, яке використовується для його ідентифікації під час входу в систему. password – це пароль, пов'язаний з обліковим записом користувача, який використовується для автентифікації логін користувача. Ім'я користувача – це ім'я користувача. Телефон користувача – це номер телефону користувача, який використовується для зв'язку з ним.

Сутність Category відповідає за запис категорії кожної страви в ресторані. Атрибут categoryID служить первинним ключем для кожної категорії в базі даних, що дозволяє унікально ідентифікувати кожну категорію. Атрибут categoryName – це назва категорії страви.

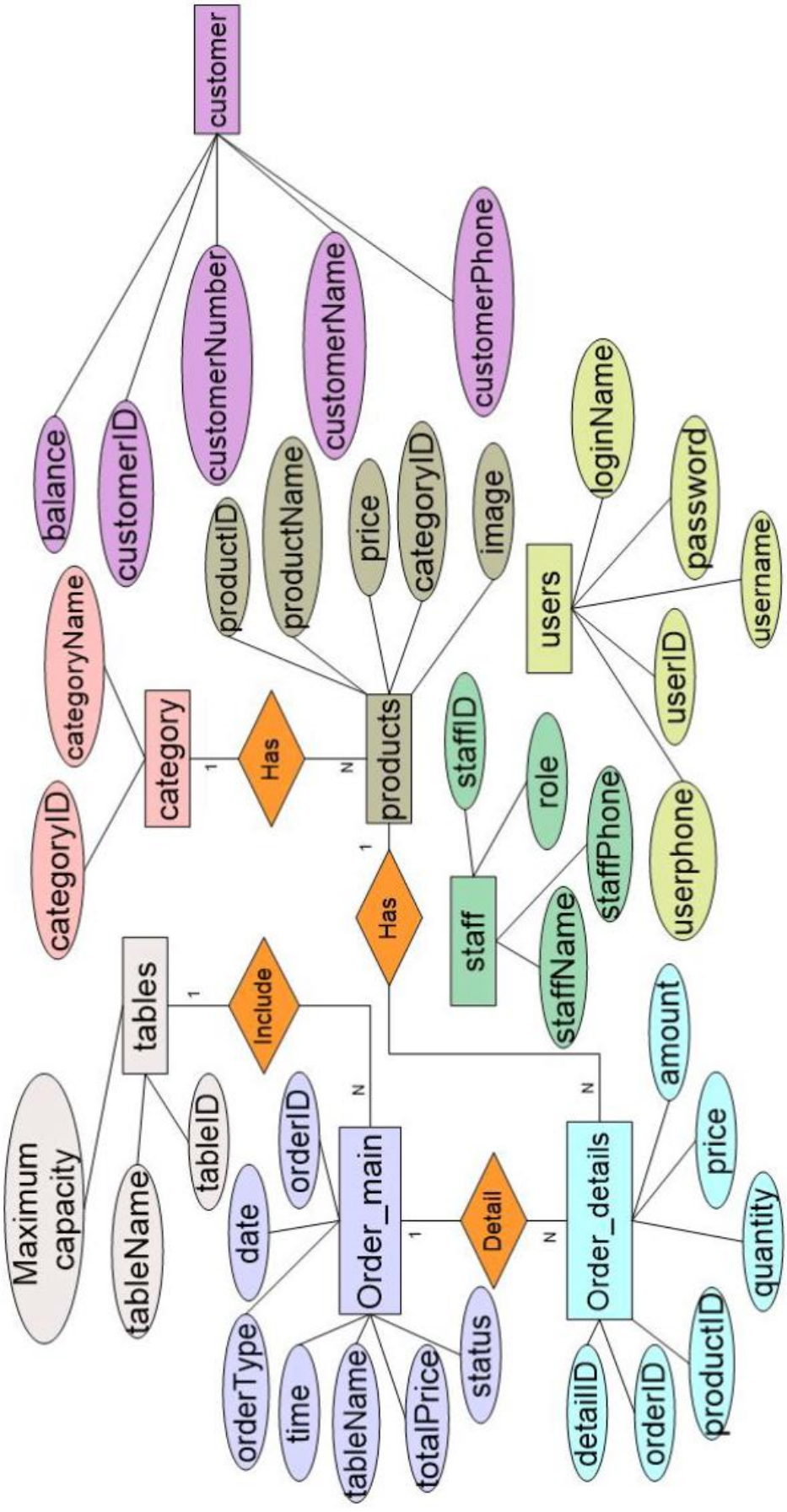

Рисунок 2.3 - Діаграма «сутність-зв'язок».

У сутності Products (Продукти) атрибут productID слугує первинним ключем, який використовується для унікальної ідентифікації кожного продукту з метою внутрішнього відстеження та посилання. Атрибут productName зберігає назву продукту, тоді як атрибут price записує ціну продажу продукту. categoryID – це зовнішній ключ, пов’язаний із сутністю Category та визначає категорію, до якої належить кожен продукт, що спрощує впорядкування та пошук продуктів за категоріями. Нарешті, атрибут image зберігає дані зображення про зовнішній вигляд продукту. Атрибут image зберігає дані зображення, що стосуються зовнішнього вигляду страви.

У сутності Table властивість tableID слугує первинним ключем, який використовується для унікальної ідентифікації кожного столу в ресторані. Ця властивість використовується для внутрішніх систем управління та бронювання. Властивість tableName надає назву столу. Властивість Maximum capacity записує максимальну кількість людей, яких можна розмістити за кожним столом. Ця властивість допомагає керівництву ресторану максимально ефективно використовувати простір, зберігаючи при цьому якість обслуговування.

У сутності Customer (Клієнт) customerID (Ідентифікатор клієнта) слугує первинним ключем, який унікально ідентифікує кожного клієнта. Це ключове поле, яке керує інформацією про клієнта в базі даних. customerNumber (Нумбер клієнта) – це унікальний номер, який клієнт отримує, коли стає учасником. Він використовується для ідентифікації та обробки транзакцій клієнтів. Ім'я клієнта вказано в полі customerName (Ім'я клієнта). Поле customerPhone записує номер телефону клієнта і часто використовується для зв'язку з ним. Поле balance (баланс) відображає передплачений баланс на рахунку клієнта, що важливо для обробки кредитних транзакцій та швидкого оформлення замовлення, особливо якщо клієнт вирішує оплатити балансом рахунку.

У сутності «Співробітник» (Staff) параметр staffID слугує первинним ключем, що однозначно ідентифікує кожного співробітника в базі даних. Поле staffName зберігає ім'я співробітника, яке використовується для щоденного спілкування та управління. Поле role описує посаду співробітника в ресторані.

Поле staffPhone записує контактний номер телефону співробітника, який в основному використовується для внутрішнього спілкування. Ця структура даних дозволяє менеджерам ресторанів ефективно керувати своїми людськими ресурсами.

У сутності Order_main, orderID служить первинним ключем, унікально ідентифікуючи кожне замовлення в базі даних. Поля дати та часу записують дату та час замовлення, надаючи інформацію, яка допомагає кухарям видавати страви в хронологічному порядку. Поле tableName вказує назву столу, з яким пов'язане замовлення. Поле orderType описує тип замовлення, наприклад, харчування в закладі або на винос. Поле status показує поточний статус замовлення (наприклад, «Очікує виконання» або «Завершено»). Наприклад, поле status вказує поточний статус замовлення. Поле totalPrice записує загальну суму замовлення, що необхідно для фінансового обліку та подальшого виставлення рахунків. Така структура гарантує, що ресторан може ефективно обробляти дані про замовлення та підвищувати операційну ефективність.

У сутності Order_details, detailID слугує первинним ключем, який використовується в базі даних для унікальної ідентифікації кожного запису деталей замовлення. orderID – це зовнішній ключ, який посилається на сутність Order_main та визначає конкретне замовлення, до якого належать ці деталі замовлення. productID2 позначає конкретний продукт, що входить до замовлення. Поле quantity записує кількість кожного продукту в замовленні. price зберігає ціну за одиницю кожного продукту сума – це обчислюване поле, зазвичай ціна, помножена на кількість, яке представляє загальну суму цього замовлення.

Фізична реалізація бази даних

На рисунку 2.4 представлено модель діаграми сутно-зв'язкових зв'язків (ERD) усієї бази даних. На ній показано ім'я, тип даних та первинний ключ кожного стовпця в таблиці бази даних, а також структуру таблиці та зв'язки між таблицями.

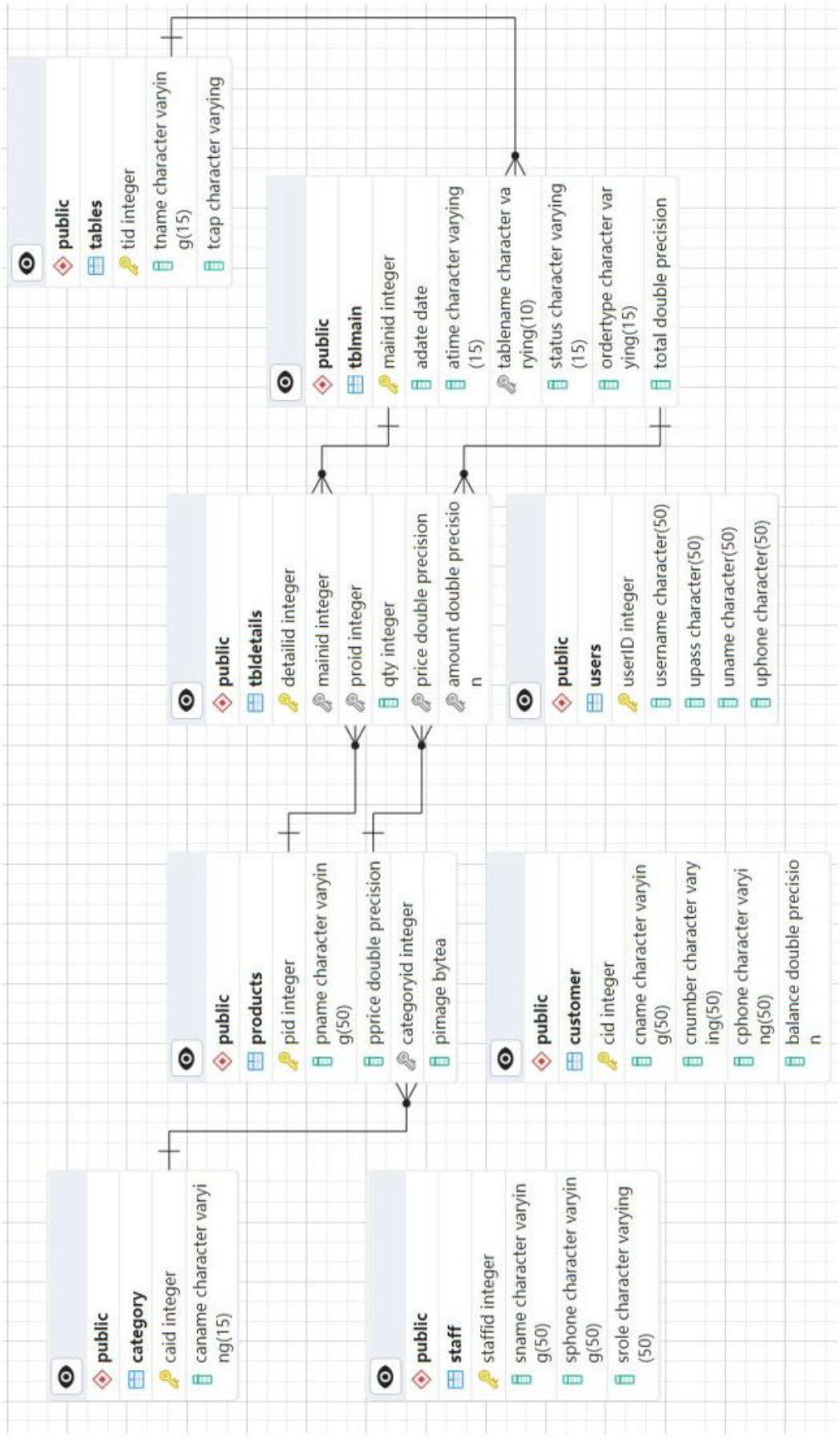

Рисунок 2.4 - Модель діаграми «сутність-зв'язок» (інструмент pgAdmin ERD).

3 СПЕЦІАЛЬНА ЧАСТИНА

3.1 Впровадження системи

Система використовує C# як мову розробки. C# — це об'єктно-орієнтована мова програмування. Вона є компонентом .NET Framework, який добре підходить для розробки масштабованих, зручних у підтримці та потужних додатків. C# пропонує комплексну бібліотеку класів та низку сучасних парадигм програмування, включаючи асинхронне програмування, запити LINQ та інші, що спрощують маніпулювання даними та багатопотокове програмування ефективним та простим способом.

Ця система використовує PostgreSQL як систему баз даних. PostgreSQL — це об'єктно-реляційна система баз даних з відкритим кодом, яка має високу сумісність, підтримує більшість стандартів SQL та надає численні розширені функції, включаючи складні запити, зовнішні ключі, транзакції тощо. Крім того, вона здатна підтримувати великі програми та середовища з високим навантаженням, а також забезпечувати надійні функції безпеки, такі як примусове шифрування та багаторівневі політики безпеки.

Система використовує середовище розробки Visual Studio. Як інтегроване середовище розробки (IDE), Visual Studio забезпечує підтримку низки завдань розробки, включаючи редагування коду, налагодження, контроль версій та інші функції, що сприяють ефективнішим процесам розробки. Visual Studio особливо добре підходить для розробки програм на C#.

.NET Framework використовується як фреймворк для розробки програмного забезпечення, розроблений компанією Microsoft. Його можна використовувати для розробки, розгортання та запуску програм. Фреймворк містить значну бібліотеку класів, які пропонують безліч функцій, включаючи графічні інтерфейси, підключення до бази даних та мережевий зв'язок. .NET Framework дозволяє використовувати C# та Windows Forms, що надає

користувачам можливість розробляти потужні інтерфейси. Крім того, .NET Framework містить вбудовані механізми безпеки для підвищення безпеки.

Компонент ADO.NET .NET Framework використовується для доступу до даних. Це модель класів .NET, які забезпечують функціональність для обробки доступу до даних. ADO.NET Framework може ефективно взаємодіяти з джерелами даних PostgreSQL, надаючи інтерфейси прикладного програмування (API) для операцій з базами даних, таких як підключення до баз даних, виконання команд, обробка транзакцій тощо. Це спрощує програмування баз даних, роблячи його простішим та зрозумілішим для розуміння та підтримки. Крім того, ADO.NET пропонує можливість встановлювати безпечні з'єднання з базою даних та виконувати запити, тим самим запобігаючи виникненню поширених загроз безпеці, таких як SQL-ін'єкції.

3.2. Реалізація входу.

Рисунок 3.1 – Інтерфейс входу користувача

Інтерфейс входу, зображений на рисунку 8, являє собою початкову точку доступу до системи. Після введення імені користувача та пароля користувачі проходять автентифікацію. Дизайн інтерфейсу простий та орієнтований на

користувача, з основним акцентом на безпеку паролів для захисту даних користувачів. Такий дизайн дозволяє системі ефективно контролювати права доступу користувачів та забезпечувати безпеку та надійність системи.

Основний код інтерфейсу входу користувача:

```
private void button1_Click(object sender, EventArgs e) {

    if(MainClass.IsValidUser(txtUser.Text, txtPass.Text)==false) {

        MessageBox.Show("Wrong User name or Password");

        return;

    }

    else {

        this.Hide();

        frmMain frm = new frmMain();

        frm.Show();

    }

}
```

Цей код відповідає за обробку облікових даних для входу та визначення того, чи дозволено користувачеві перейти до наступного етапу системи – головного інтерфейсу. Це визначення базується на облікових даних, введених користувачем. У випадку, якщо облікові дані, надані користувачем, неправильні, відображається повідомлення про помилку. І навпаки, якщо облікові дані правильні, поточне вікно приховується, і відображається головний екран. «if(MainClass.IsValidUser(txtUser.Text, txtPass.Text)==false)», цей рядок коду ініціює виклик MainClass, який приймає введені користувачем ім'я користувача (txtUser.Text) та пароль (txtPass.Text) як параметри. Цей метод використовується для перевірки дійсності облікових даних користувача. У випадку негативного результату можна зробити висновок, що ім'я користувача або пароль неправильні. З'явиться вікно повідомлення з текстом «Неправильне ім'я користувача або пароль».

3.3. Впровадження категорії

Інтерфейс модуля категорій, зображений на рисунку 3.2, являє собою ключовий компонент системи, який призначений для допомоги менеджерам ресторанів в ефективному управлінні категоріями в їхніх меню. Надання можливості додавати, шукати, редагувати та видаляти категорії полегшує використання системи та забезпечує точність та актуальність інформації в меню.

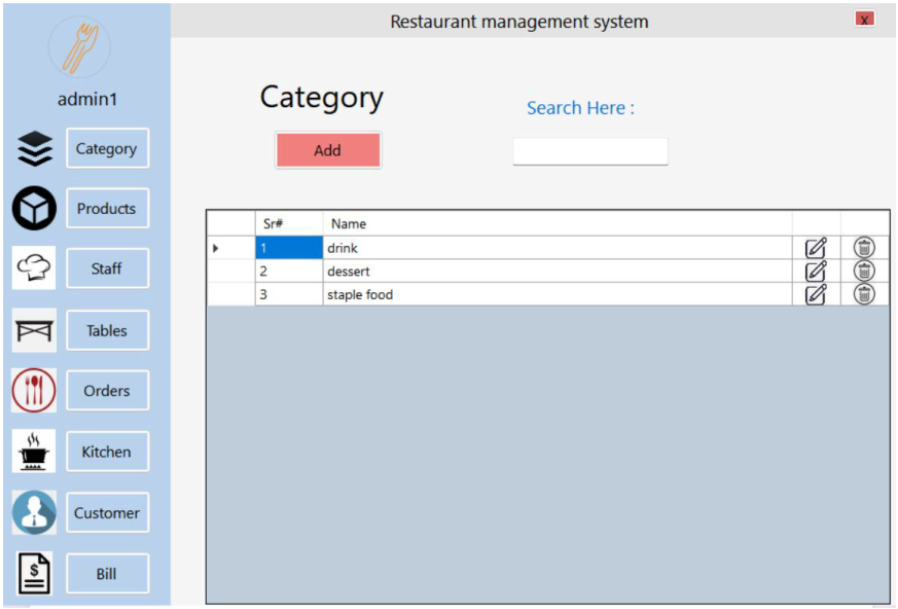

Рисунок 3.2 - Інтерфейс категорії.

На рисунку 3.3 зображено інтерфейс спливаючого вікна «Додати категорію», який є важливою особливістю системи. Цей інтерфейс спрощує процес додавання нових категорій, тим самим підвищуючи загальну ефективність системи. Зручний інтерфейс та просте керування системою дозволяють менеджерам ресторанів легко додавати нові страви чи напої до своїх меню, забезпечуючи різноманітний та актуальний вибір.

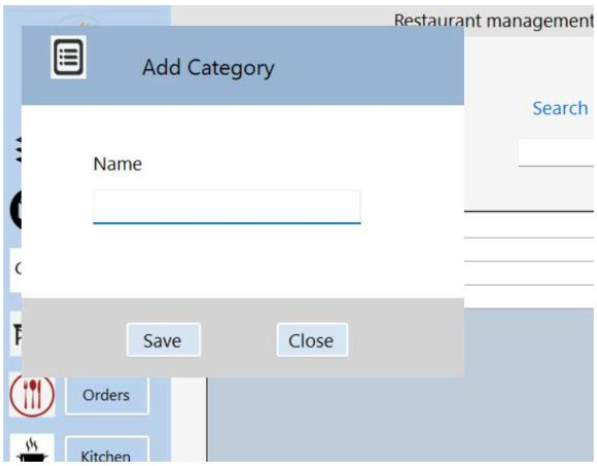

Рисунок 3.3 - Інтерфейс додавання категорії

1. Додати категорію основний код:

```
public override void button2_Click(object sender, EventArgs e) {
    string message = "";
    if (id == 0) { message = "INSERT INTO category (caname) VALUES (@Name)";}
    else { message = "Update category Set caname = @Name where caid = @id";}
    Hashtable hashtable = new Hashtable();
    hashtable.Add("@id", id);
    hashtable.Add("@Name",categoryName.Text);
    if (MainClass.SQl(message, hashtable) >0) {
        MessageBox.Show("Saved successfully");
        id = 0;
        categoryName.Text = "";
        categoryName.Focus();
    }
}
```

У разі нової операції, для вставки нової категорії має бути підготовлена команда SQL INSERT. Якщо це не нове додавання, для оновлення вже існуючої категорії підготовлена команда SQL UPDATE. У коді має бути створена хеш-

таблиця для зберігання параметрів команди SQL. Потім до хеш-таблиці слід додати ідентифікатор та назву категорії, введені користувачем на інтерфейсі.

2. Видалення та редагування категорії Код функції:

```
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e){
    if (dataGridView1.CurrentCell.OwningColumn.Name == "dgvdel") {
        int num = Convert.ToInt32(dataGridView1.CurrentRow.Cells["dgvid"].Value);
        string message = "DELETE FROM category WHERE caid = @id";
        Hashtable hashtable = new Hashtable();
        hashtable.Add("@id", num);
        if (MainClass.SQl(message, hashtable) > 0) { MessageBox.Show("Deleted Successfully"); }
        else { MessageBox.Show("Delete failed");}
        GetData();
    }
    if (dataGridView1.CurrentCell.OwningColumn.Name == "dgvedit") {
        frmCategoryAdd frm = new frmCategoryAdd();
        frm.id = Convert.ToInt32(dataGridView1.CurrentRow.Cells["dgvid"].Value);
        frm.categoryName.Text =
        Convert.ToString(dataGridView1.CurrentRow.Cells["dgvName"].Value);
        frm.ShowDialog();
        GetData();
    }
}
```

Операція видалення виглядає наступним чином:

Для зберігання параметрів команди SQL створюється хеш-таблиця. Необхідно додати ідентифікатор категорії як параметр до хеш-таблиці. Наступним кроком є виконання команди видалення та перевірка повернутого результату. Якщо видалення успішне, має відобразитися вікно повідомлення

«Успішне видалення». У разі невдачі користувачеві буде показано вікно повідомлення про те, що видалення не вдалося.

Операція редагування включає такі кроки:

З DataGridView отримати поточний рядок ідентифікатора класифікації. Встановити це значення для властивості id форми редагування. Система отримає назву категорії поточного рядка та призначить її відповідному текстовому полю форми редагування. Після завершення процесу редагування дані за категоріями необхідно перезавантажити, щоб оновити вміст, що відображається в DataGridView.

3.4. Впровадження продуктів

На рисунках 3.4 та 3.5 зображено інтерфейс керування продуктами, де адміністратор може керувати інформацією про продукти ресторану. Панель навігації розташована в лівій частині інтерфейсу та містить функціональні модулі системи. Вибраний модуль — «Продукти», який відображає список доданих продуктів. Після натискання кнопки «Додати» з’явиться нове вікно, що дозволить адміністратору ввести нову інформацію про продукт. Адміністратор може ввести відповідні деталі продукту, включаючи назву, ціну та категорію, а також вибрати зображення продукту.

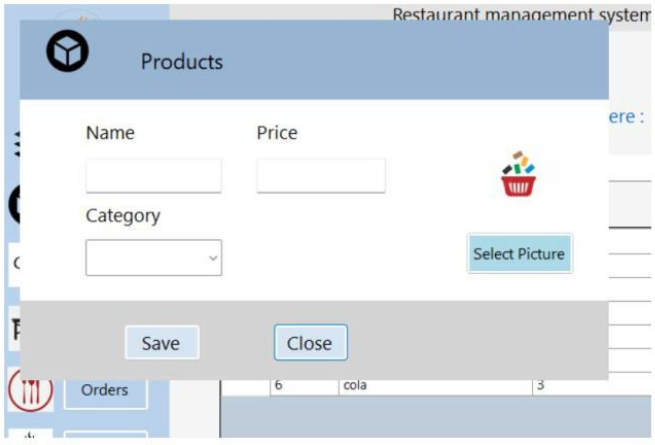

Рисунок 3.4 - Інтерфейс додавання продуктів.

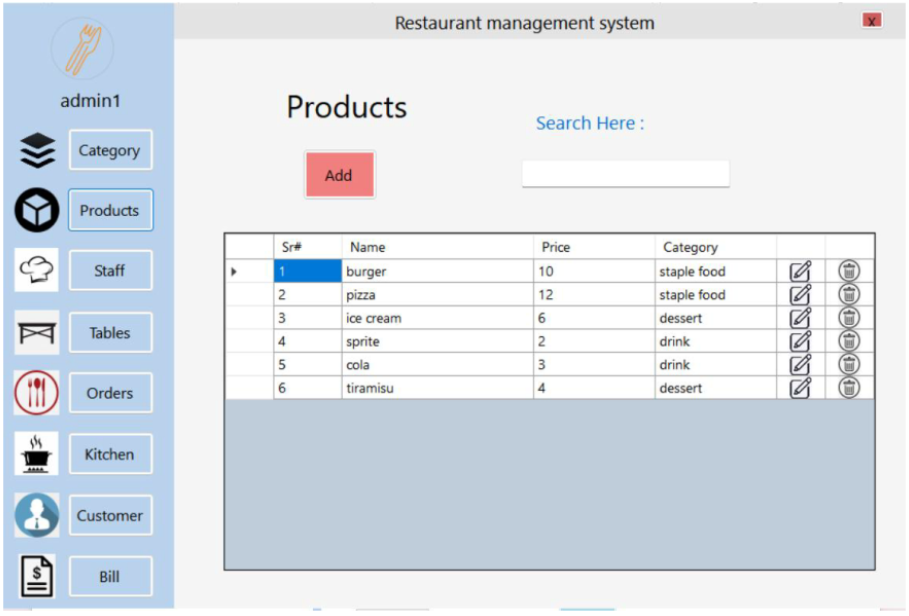

Рисунок 3.5 - Інтерфейс продуктів

3.5. Впровадження персоналу

На рисунках 3.6 та 3.7 зображено інтерфейс модуля «Персонал», який використовується адміністраторами для контролю даних про співробітників. Інтерфейс містить серійний номер співробітника (Sr#), ім’я, номер телефону та посаду. Поле пошуку дозволяє адміністраторам знаходити конкретних співробітників. Додавання нових співробітників можна здійснити, натиснувши кнопку «Додати». Кожен рядок співробітника містить піктограму «Редагувати» та «Видалити», яку можна вибрати для зміни або видалення інформації про співробітника. На екрані «Додати співробітника» адміністратор має опцію введення імені, номера телефону працівника та вибір ролі. Щоб зберегти інформацію про нового працівника, адміністратор повинен натиснути кнопку «Зберегти». Щоб закрити вікно, адміністратор повинен натиснути кнопку «Закрити».

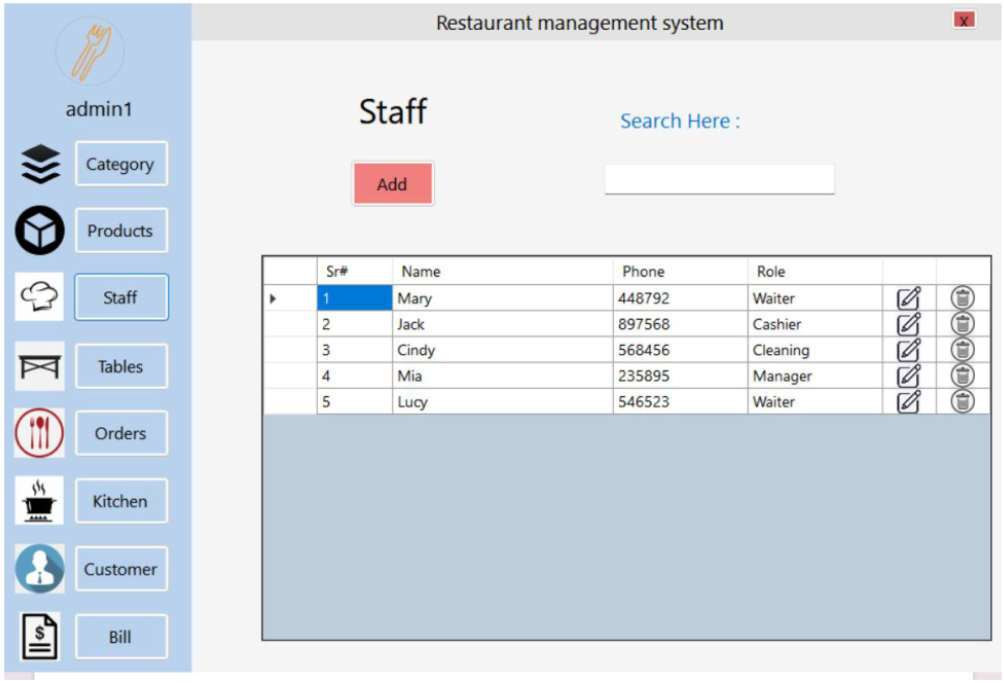

Рисунок 3.6 - Інтерфейс персоналу.

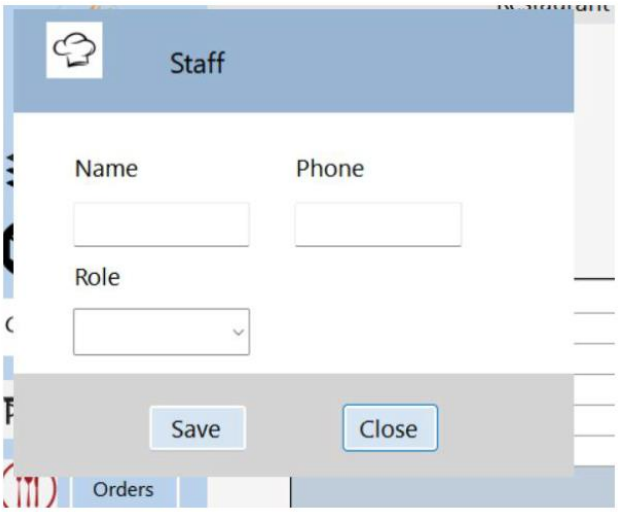

Рисунок 3.7 - Інтерфейс додавання персоналу

3.6. Реалізація таблиць

На рисунках 3.8 та 3.9 представлено вичерпний огляд інформації про столики ресторану, включаючи назву столу та максимальну місткість. Користувачі можуть вводити ключові слова в поле пошуку, щоб знайти певний стіл. Адміністраторам надається можливість змінення або видалення інформації про стіл, натиснувши на відповідний значок. Екран «Інформація про стіл»

дозволяє користувачам ввести назву та максимальну місткість відповідного столу.

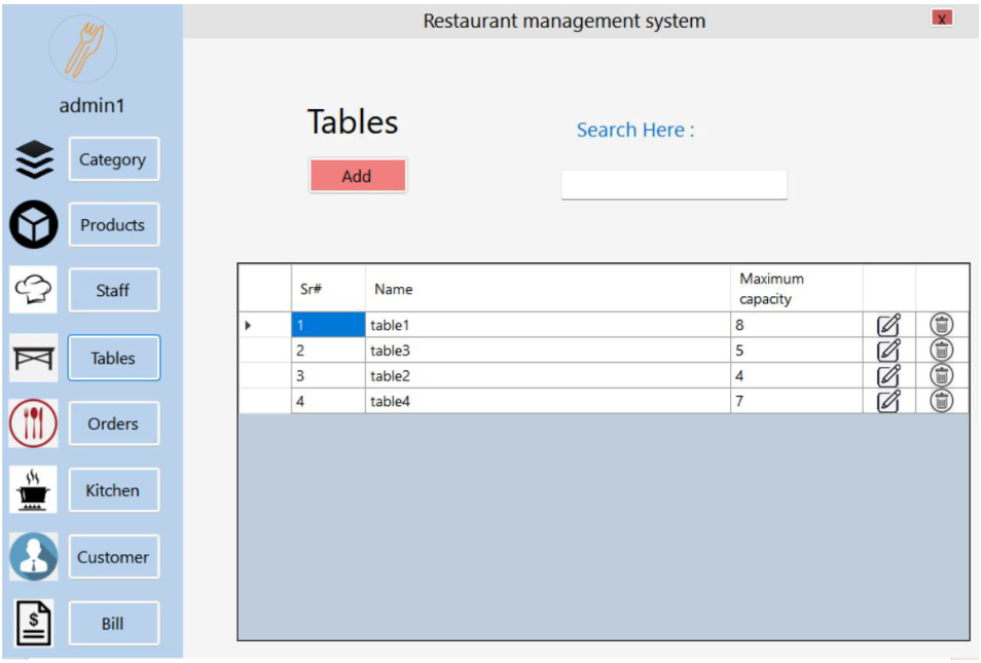

Рисунок 3.8 - Інтерфейс таблиць

Рисунок 3.9 - Інтерфейс додавання таблиць

3.7. Виконання замовлень

На рисунках 3.10-3.12 приведено форми, які використовуються для полегшення замовлення клієнтами. Клієнт може вибрати та додати до свого

замовлення різні продукти, включаючи такі товари, як бургери, піца та морозиво, як показано на супровідному зображенні.

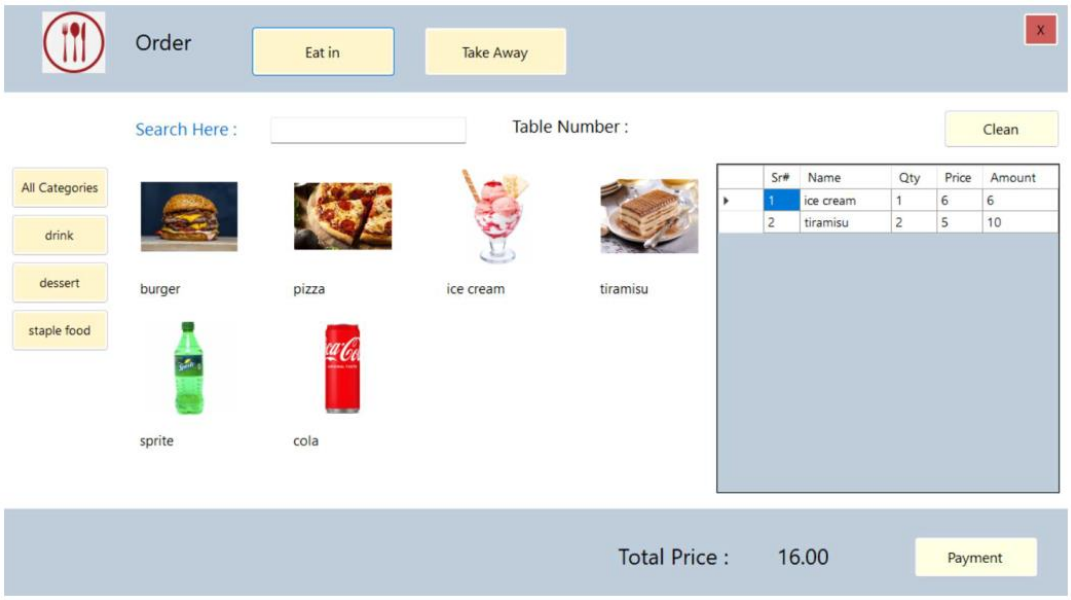

Рисунок 3.10 - Інтерфейс замовлень

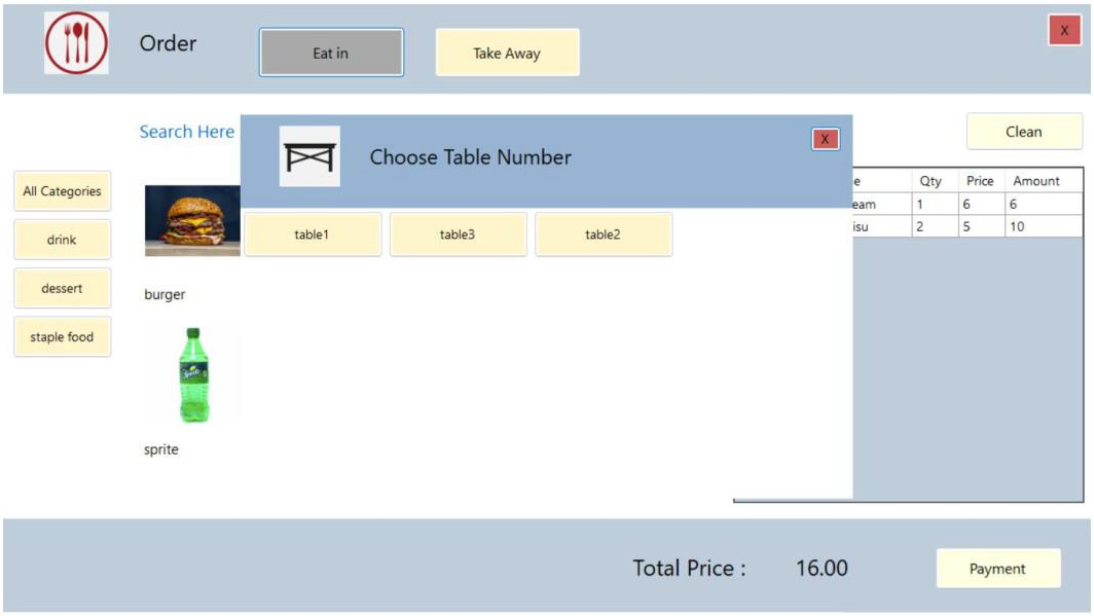

Рисунок 3.11 - Інтерфейс вибору таблиці

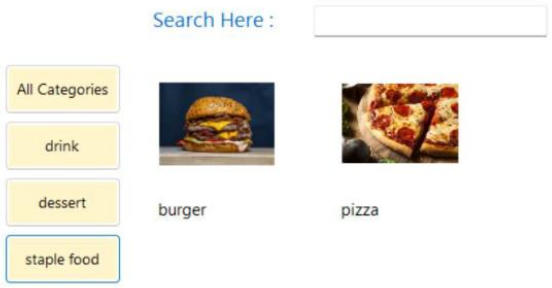

Рисунок 3.12 - Скринінг категорії харчових продуктів

Ліворуч користувачі можуть фільтрувати продукти за категоріями (наприклад, напої, десерти, основні страви тощо) або використовувати рядок пошуку для швидкого пошуку певних страв. Після натискання кнопки категорії «основні продукти харчування» інтерфейс відображатимуться лише страви, що належать до категорії «основні продукти» (наприклад, бургери та піца). Відвідувачі можуть обрати обід у закладі або замовити їжу на винос, і під час вибору опції «в ресторані» їм потрібно ввести номер столу. У столі праворуч відображається назва, кількість, ціна за одиницю та загальна ціна вибраної страви. Загальна вартість замовлення відображається внизу. Після того, як клієнт вибере страву в закладі, з’явиться спливаюче вікно з пропозицією вибрати номер столу. На цьому екрані відображаються всі доступні номери столів для вибору клієнтом. Після вибору номера столу відповідний номер столу відображається в полі «Номер столу».

На рисунках 3.13-3.14 показано мобільний інтерфейс та інтерфейс кошика для покупок системи замовлення ресторану. Після завершення покупки вибраних страв користувачеві відображається інтерфейс кошика для покупок, натиснувши кнопку «Показати кошик». Ці два інтерфейси дозволяють користувачам зручно переглядати, шукати, вибирати страви та переглядати вміст своїх кошиків для покупок. Нарешті, користувачі можуть здійснювати платежі. Процес замовлення простий та інтуїтивно зрозумілий, що покращує взаємодію з користувачем.

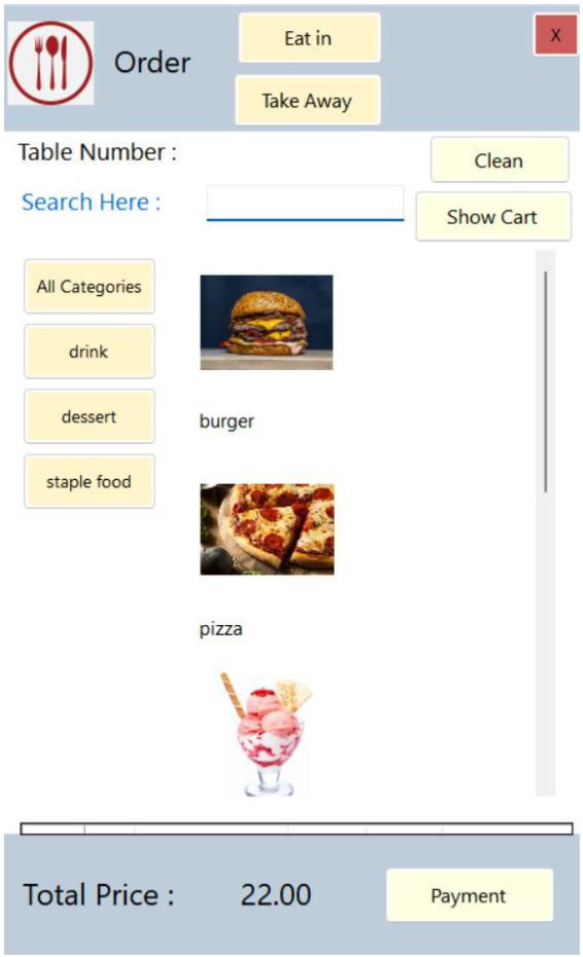

Рисунок 3.13 - Інтерфейс мобільного замовлення.

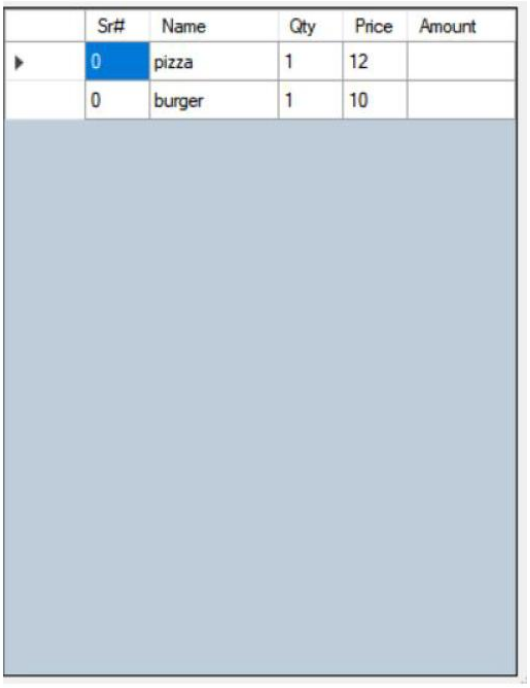

Рисунок 3.14 - Інтерфейс мобільного кошика для покупок

1. Код функції продуктів:

```
private void AddProduct(string id, String proID, string name, string cat, string price, Image pimage)
{
    var product1 = new ucProduct() {
        PName = name,
        PPrice = price,
        PCategory = cat,
        PImage = pimage,
        id = Convert.ToInt32(proID) };
    ProductPanel.Controls.Add(product1);
    product1.onSelect += (ss, ee) => {
        var product2 = (ucProduct)ss;
        bool found = false;
        foreach (DataGridViewRow item in dataGridView1.Rows) {
            if (Convert.ToInt32(item.Cells["dgvproID"].Value) == product2.id) {
                item.Cells["dgvAmount"].Value = int.Parse(item.Cells["dgvQty"].Value.ToString()) *
double.Parse(item.Cells["dgvPrice"].Value.ToString());
                item.Cells["dgvQty"].Value = int.Parse(item.Cells["dgvQty"].Value.ToString()) + 1;
                found = true;
                break; } }
        if (!found) { dataGridView1.Rows.Add(new object[] { 0, 0, product2.id, product2.PName, 1,
product2.PPrice, product2.PPrice });}
        GetTotal();
    };
}
```

Ця функція використовується для додавання товарів (елемент керування ucProduct) до ProductPanel. Для спрощення обробки операцій вибору товарів,

таких як оновлення даних кошика для покупок, необхідно реалізувати обробник подій для події onSelect цього елемента керування.

2. Загальний код функції:

```
private void GetTotal() {
    double total = 0;
    totalPrice.Text = "";
    foreach (DataGridViewRow item in dataGridView1.Rows) {
        total += double.Parse(item.Cells["dgvAmount"].Value.ToString());
    }
    totalPrice.Text = total.ToString("N2");
}
```

Ця функція обчислює та відображає загальну суму всіх рядків у dataGridView1. Щоб обчислити загальну суму, необхідно пройтися по кожному рядку dataGridView1 та накопичити значення комірок dgvAmount. Остаточна сума відображається в мітці з ідентифікатором "totalPrice".

3.8. Реалізація кухні

На рисунку 3.15 зображено інтерфейс кухонного модуля системи. Основна функція цього інтерфейсу полягає в допомозі шеф-кухарю в управлінні та обробці замовлень. Кожна Картка замовлення містить основні деталі замовлення, і шеф-кухар може обробити замовлення в хронологічній послідовності на основі інформації, представленої в картці замовлення. Після завершення замовлення шеф-кухарю пропонується натиснути кнопку «Завершити», яка оновлює статус замовлення на «Завершено». Це полегшує відстеження та управління замовленнями, тим самим гарантуючи своєчасну обробку кожного замовлення.

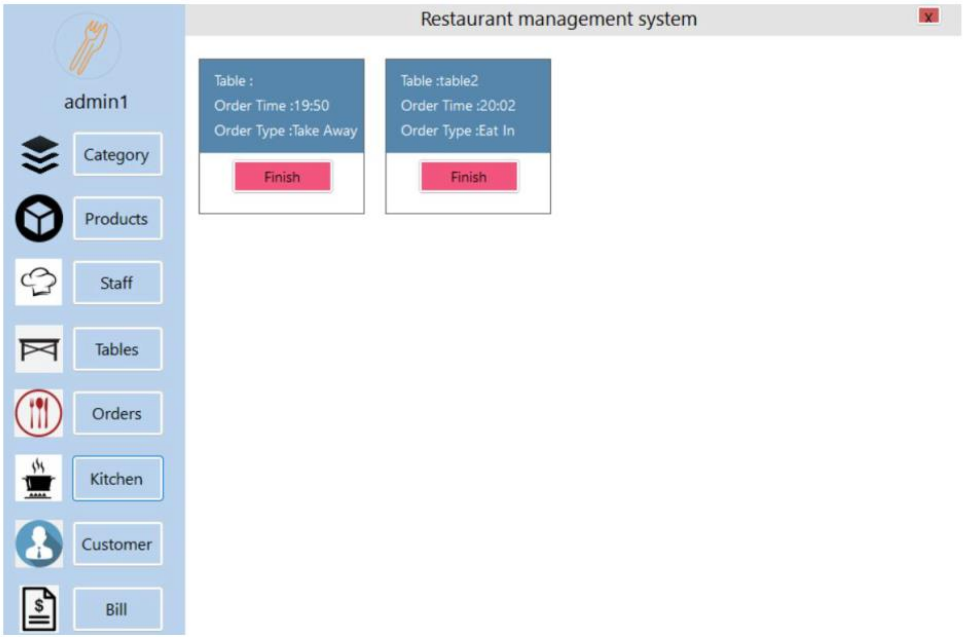

Рисунок 3.15 - Інтерфейс кухні.

На рисунках 3.16-3.17 показано модуль «Клієнт» у системі ресторану. Адміністратор може вводити, змінювати та видаляти інформацію про клієнта в цьому модулі. У таблиці відображається вся поточна інформація про клієнта, включаючи серійний номер, ім'я, номер клієнта, номер телефону та баланс.

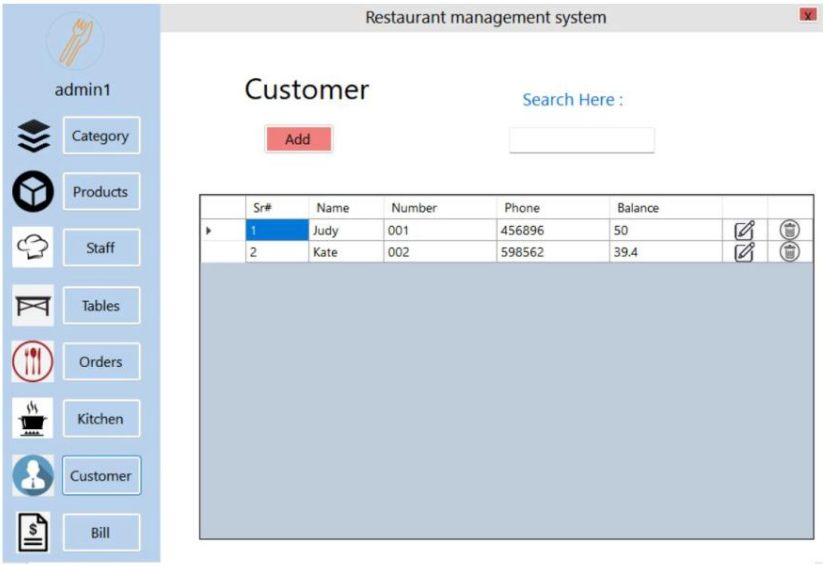

Рисунок 3.16 - Інтерфейс клієнта

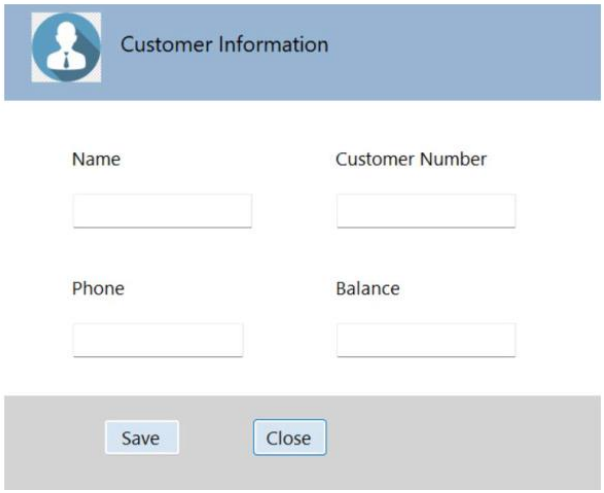

Рисунок 3.17 - Інтерфейс додавання клієнта.

Після натискання кнопки «Додати» з'явиться нове діалогове вікно, яке дозволить адміністратору ввести інформацію про нового клієнта. Крім того, адміністратор може змінювати або видаляти інформацію про клієнта за допомогою значків «Редагувати» та «Видалити» відповідно, розташованих у правій частині форми. Поле пошуку використовується для швидкої ідентифікації конкретного клієнта.

На рисунку 3.18 показано модуль «Рахунки» в системі управління рестораном.

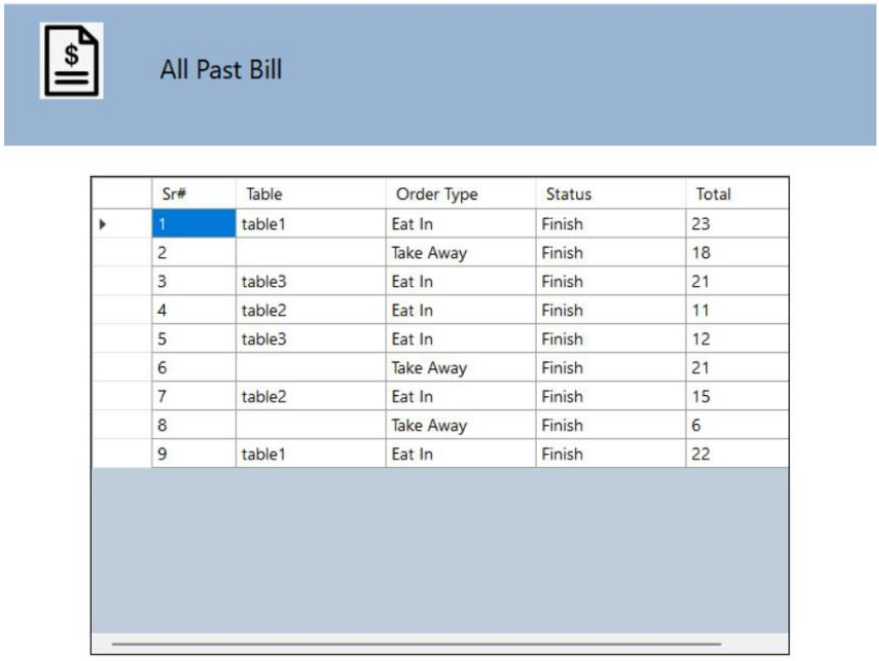

Рисунок 3.18 - Інтерфейс попередніх рахунків.

У цьому модулі адміністратор може переглядати записи всіх виконаних замовлень. Таблиця відображає деталі всіх історичних рахунків, включаючи серійний номер, номер столу, тип замовлення, статус замовлення та загальну суму. Цей інтерфейс забезпечує ефективний та точний перегляд записів про виставлення рахунків адміністраторами ресторану.

Основний код інтерфейсу попереднього рахунку:

```
private void GetData()
{
   string qry = @"select MainID, TableName,orderType,status,total from tblMain where status <> 'Pending'";
   ListBox lb = new ListBox();
   lb.Items.Add(dgvid);
   lb.Items.Add(dgvtable);
   lb.Items.Add(dgvType);
   lb.Items.Add(dgvStatus);
   lb.Items.Add(dgvTotal);
   MainClass.LoadData(qry, dataGridView1, lb);
}
```

Основне завдання цієї функції — отримати виконані замовлення з бази даних та представити їх у dataGridView1. Функція відповідає за виконання SQL-запитів та динамічне зв'язування результатів зі стовпцями dataGridView1.

3.9. Результати тестування функціональності системи управління

На рисунках 3.19-3.21 показано результати тестів чорної скриньки, проведених для кожного функціонального модуля системи. Тести поділено на такі категорії:

Результат тесту функції додавання: Користувач успішно додав новий елемент до системи. Користувачі змогли ввести правильні назви елементів, ціни та категорії, а також успішно додати елементи до списку.

Результат тестування функції пошуку: Поле пошуку дозволяє користувачам швидко знаходити елементи. Користувачі вводять часткові або повні назви елементів у поле пошуку, і список правильно відображає відповідні елементи.

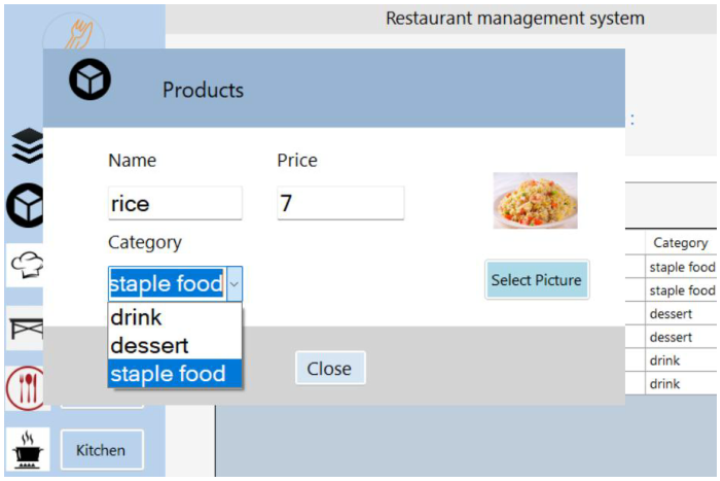

Рисунок 3.19 - Додати функціональний тест.

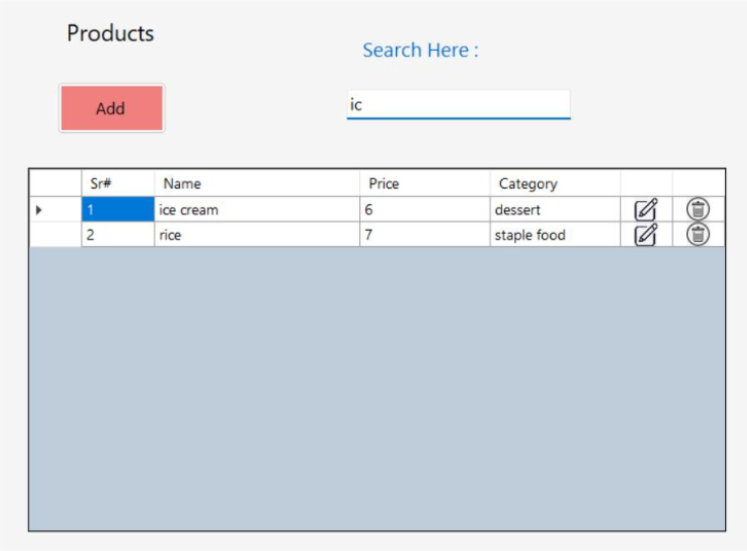

Рисунок 3.20 - Тест функції пошуку.

Результат тесту функціональності редагування: Система здатна впровадити необхідні зміни. Користувач може вибрати існуючий товар та змінити його інформацію, таку як ціна. Натискаючи кнопку «Зберегти», користувач може перевірити, чи були зміни успішно застосовані та відображені у списку.

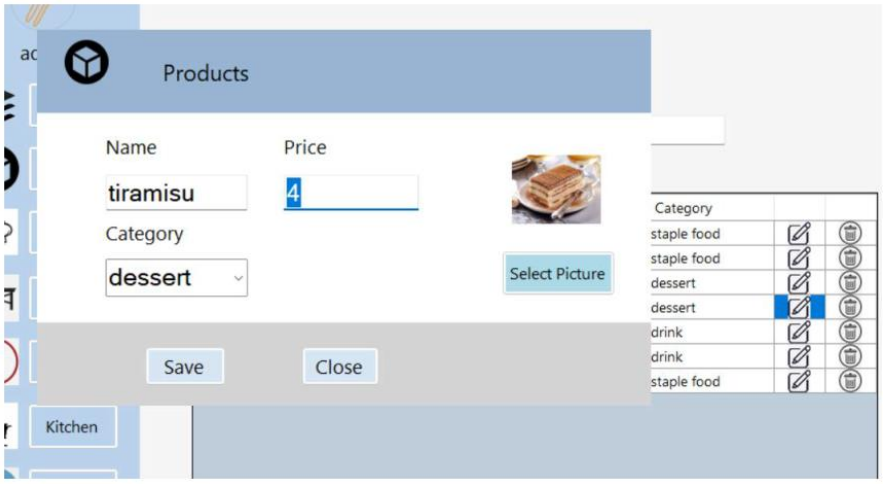

Рисунок 3.21 - Тест функції редагування

Результати тестування показують, що функціональність працює належним чином, інтерфейс користувача є інтуїтивно зрозумілим та відповідає основним бізнес-вимогам. Основною виявленою проблемою був повільний час відгуку основного інтерфейсу користувача. Кілька функцій демонстрували неочікувану поведінку після введення певних даних, і необхідні виправлення були внесені. Рекомендується оптимізувати швидкість реагування інтерфейсу та стабільність кількох функцій під час подальшої розробки. Крім того, рекомендується підтримувати високий стандарт охоплення тестами в майбутній розробці, щоб зберегти якість програмного забезпечення. У наступних випусках система повинна постійно вдосконалюватися та оптимізуватися, щоб забезпечити її надійне обслуговування користувачів.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Ергономічні проблеми безпеки життєдіяльності при роботі за комп'ютером

У сучасному світі все більше людей проводять значну частину свого робочого часу за комп'ютером, що призводить до ряду ергономічних проблем, які негативно впливають на безпеку та здоров'я людей. Неправильна позиція тіла, незручне розташування робочого місця, тривале сидіння та напружена робота з клавіатурою та мишею спричиняють м'язове напруження, біль у спині та напругу очей, що викликає дискомфорт та незручності.

Розташування робочого місця відіграє важливу роль у забезпеченні безпеки та здоров'я під час роботи за комп'ютером. Оптимальна висота столу та стільця є ключовим фактором для забезпечення комфорту і підтримки правильної позиції тіла. Стіл належної висоту, щоб лікті могли спокійно розташовуватися на клавіатурі, а стопи - на підлозі. Стілець обладнаний підтримкою для спини та належними регульованими підлокітниками. Клавіатура розташована на рівні ліктів, монітор - належним чином вирівняний перед очима, а миша - в зоні доступу для зап'ястя.

Правильне розташування робочого місця сприяє уникненню незручностей та проблем зі здоров'ям. Ергономічні пристрої підтримки є корисними для забезпечення комфорту та запобігання напрузі м'язів. Використання регульованих підлокітників та підставок для зап'ястя допомагає знизити напругу на м'язах і запобігти незручностям. Оптимальне розташування робочого місця повинне враховувати індивідуальні особливості користувача та забезпечувати комфортні умови для роботи.

Правильна позиція тіла та рухи є ключовими факторами для забезпечення комфорту та запобігання напрузі м'язів та незручностям.

Важливо пам'ятати про правильну позицію спини та уникати підгорблення або надмірного нахилу голови під час тривалої роботи за комп'ютером. Регулярні перерви для розтяжки та руху є важливими для підтримання здоров'я. Прості фізичні вправи, такі як розтягування шиї, плечей, рук і ніг, допомагають зняти напругу з м'язів та покращити кровообіг. Регулярні перерви дозволяють розслабитися і зберегти енергію для продуктивної роботи. Очі є одними з найбільш вразливих органів під час роботи за комп'ютером. Постійне спрямування погляду на екран призводить до напруження та втому очей.

Для зменшення негативного впливу необхідно використовувати екрани з антиблисковим покриттям, яке знижує відблиск та рефлексію світла. Також важливо налаштовувати яскравість та контрастність екрану для комфортного сприйняття. Щоб зменшити напругу на очі, необхідно робити перерви для відпочинку, фокусуючись на далеких предметах або виконуючи вправи для розслаблення очей.

Використання неправильної клавіатури та миші призводить до м'язового напруження і тунельного синдрому. Важливо використовувати ергономічні клавіатури та миші з додатковою підтримкою для зап'ястя та комфортною формою. Правильна позиція рук та зап'ястя під час роботи з ними має велике значення. Регулярні перерви для розтяжки рук та масажу зап'ястя допомагають уникнути негативних наслідків від тривалого використання клавіатури та миші. Розглядаючи ергономічні проблеми безпеки життєдіяльності при роботі за комп'ютером, варто зазначити, що їх вирішення є ключовим для забезпечення безпеки та здоров'я під час роботи.

Дотримання принципів ергономіки, налагодження робочого місця, правильна позиція тіла та виконання фізичних вправ сприяють покращенню безпеки та створенню здорового робочого середовища. Застосування ергономічних пристроїв підтримки, таких як регульовані підлокітники та підставки для зап'ястя, також сприяє комфорту та попередженню

незручностей. Загальною метою є створення безпечного та здорового робочого середовища для людей, які працюють за комп'ютером та виконують дослідження захищеності веб сервісу електронного навчання Atutor, що забезпечує збереження здоров'я та підвищення продуктивності працівників, сприяє запобіганню травмам та ергономічним проблемам, а також сприяє загальному комфорту та задоволенню від роботи.

Додатково, важливо зазначити, що належна освітленість приміщення також є важливим фактором, який впливає на комфорт та здоров'я під час роботи за комп'ютером. Потрібно забезпечити достатнє природне або штучне освітлення, яке не перевантажує очі. Розміщення робочого місця біля вікна або використання належної освітлювальної техніки допомагає забезпечити оптимальні умови освітлення.

Важливо також уникати відблисків на екрані, розташовуючи монітор під правильним кутом до джерел світла. Безпека та конфіденційність інформації також є важливим аспектом при роботі за комп'ютером. Важливо зберігати конфіденційні дані та захищати їх від несанкціонованого доступу. Використання паролів, шифрування даних та регулярне оновлення програмного забезпечення допомагає забезпечити безпеку інформації.

Крім того, необхідно усвідомлювати потенційні загрози з боку шкідливих програм та фішингових атак і приймати заходи для їх запобігання, такі як використання антивірусного програмного забезпечення та обережне відкривання електронних листів та посилань. Регулярне навчання та свідомість про важливість ергономіки та безпеки при роботі за комп'ютером є важливими.

Працівники повинні мати бути проінформовані про правильні методи роботи, виконання пауз і фізичних вправ, а також процедур безпеки. Організації можуть проводити навчання та інформаційні тренінги, щоб підвищити свідомість та забезпечити правильну поведінку під час роботи за комп'ютером.

Враховуючи всі ці аспекти, створення комфортного, безпечного та здорового робочого середовища є важливим завданням як для працівників, так і для роботодавців. Захист здоров'я та добробуту працівників при роботі за комп'ютером не тільки покращує їхню якість життя, але й сприяє збільшенню продуктивності та задоволення від роботи, що має позитивний вплив на всю організацію.

4.2 Організація безпечної роботи електроустаткування задіяного при роботі системи електронного навчання

Безпечна робота електроустаткування є ключовим елементом для забезпечення стабільної та безперебійної роботи системи електронного навчання. Це включає в себе правильне проектування, встановлення, експлуатацію та обслуговування електроустаткування. Дотримання стандартів і правил охорони праці допомагає мінімізувати ризики електричних ударів, пожеж та інших небезпек.

Перед встановленням електроустаткування необхідно ретельно оцінити його відповідність вимогам системи електронного навчання. Вибір обладнання повинен базуватися на таких критеріях: надійність та безпека: обладнання має відповідати стандартам якості та безпеки, мати сертифікати відповідності та бути розрахованим на довготривалу експлуатацію; відповідність технічним вимогам: обладнання повинно підтримувати необхідні технічні параметри, такі як напруга, потужність, тип з'єднання тощо;сумісність з іншими компонентами: всі компоненти системи повинні бути сумісні між собою, щоб уникнути збоїв та небезпек при експлуатації/

Під час встановлення необхідно дотримуватися таких заходів безпеки:

правильне заземлення: всі пристрої повинні бути заземлені відповідно до норм, щоб уникнути накопичення статичної електрики та можливості удару струмом;використання захисних пристроїв: встановлення автоматичних

вимикачів, пристроїв захисного вимкнення (ПЗВ) та інших захисних засобів є обов’язковим для забезпечення безпеки; професійний монтаж: монтаж повинен виконуватися кваліфікованими спеціалістами з дотриманням всіх норм і правил/

Для забезпечення безпечної експлуатації електроустаткування слід дотримуватися таких рекомендацій:регулярний контроль та технічне обслуговування: обладнання повинно регулярно перевірятися на наявність зношення, перегріву, пошкоджень проводів та інших дефектів. Технічне обслуговування повинно проводитися відповідно до інструкцій виробника; контроль температурного режиму: устаткування не повинно перегріватися. Необхідно забезпечити достатню вентиляцію та уникати розташування пристроїв поблизу джерел тепла; правильне використання: обладнання має використовуватися тільки за призначенням, з дотриманням інструкцій та рекомендацій виробника.

При обслуговуванні та ремонті електроустаткування необхідно дотримуватися таких заходів безпеки: відключення живлення: перед проведенням будь-яких робіт обладнання має бути відключене від джерела живлення; використання засобів індивідуального захисту (ЗІЗ): спеціалісти повинні використовувати відповідні ЗІЗ, такі як діелектричні рукавички, килимки, інструменти з ізоляційними покриттями; дотримання процедур: всі ремонтні роботи повинні проводитися відповідно до технічних інструкцій та нормативних документів;

Управління ризиками та навчання персоналу є невід'ємною частиною забезпечення безпеки електроустаткування: ідентифікація ризиків: постійний аналіз можливих ризиків та їх мінімізація є ключовим аспектом безпеки. Для цього проводяться регулярні оцінки стану обладнання та аналіз умов експлуатації; навчання та інструктажі: персонал повинен проходити регулярні навчання та інструктажі з питань безпечної роботи з електроустаткуванням. Це включає як базові знання, так і спеціальні навички для роботи з

конкретними видами обладнання; розробка та впровадження процедур безпеки: необхідно розробити детальні інструкції та процедури з безпеки, які повинні бути доступними для всіх працівників та регулярно оновлюватися. Таким чином, організація безпечної роботи електроустаткування задіяного при роботі системи електронного навчання вимагає комплексного підходу, що включає вибір надійного обладнання, правильне його встановлення, регулярне технічне обслуговування, навчання персоналу та управління ризиками. Дотримання цих заходів дозволяє забезпечити стабільну та безпечну роботу системи, що є критично важливим для ефективного функціонування освітнього процесу.

ВИСНОВКИ

У роботі проведено проектування та впровадження сучасної системи замовлення в ресторані, з особливим акцентом на функціональність модулів системи, включаючи вхід, замовлення, обробку замовлень на кухні та управління виставленням рахунків. Також розглянуто вплив модулів на операційну ефективність ресторану. Метою роботи є забезпечення ефективного, зручного та безпечного рішення, яке допоможе власникам ресторанів покращити обслуговування клієнтів та оптимізувати внутрішні процеси, тим самим підвищуючи ефективність бізнесу.

Модульна конструкція системи дозволяє незалежну розробку та тестування кожного модуля, тим самим покращуючи ремонтопридатність та масштабованість системи. Поділ системи на кілька незалежних модулів призвів до спрощення процесу розробки, завдяки якому кожну частину можна розробляти та тестувати незалежно. Це підвищило ремонтопридатність та масштабованість системи.

Зручний інтерфейс розроблений для полегшення використання системи користувачами без технічних знань. Система забезпечує інтуїтивно зрозумілий інтерфейс користувача, який гарантує, що користувачі можуть легко керувати системою, не заглиблюючись у технічні деталі, що робить її дуже зручною для персоналу ресторану без технічного досвіду.

ПЕРЕЛІК ПОСИЛАНЬ

1. Bhargave, A., Jadhav, N., Joshi, A., Oke, P. та Lahane. Digital Ordering System for Restaurant Using Android [Електронний ресурс] // International Journal of Scientific and Research Publications. – 2013. – Т. 3, № 4. – Режим доступу: http://www.ijsrp.org/research-paper-0413/ijsrp-p1647.pdf.
2. Chong, K.L. Factors affecting the consumers' embracement of manual self-ordering system (order chit) in restaurants // Journal of Foodservice Business Research. – 2022. – Т. 25, № 1. – С. 33–56. – DOI: 10.1080/15378020.2021.1911565.
3. Fujita, T., Hirata, M., Abe, K. та ін. Self-ordering system of restaurants for considering allergy information // Proceedings of the 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC). – Las Vegas, 2014. – С. 179–184. – DOI: 10.1109/CCNC.2014.6866553.
4. ghogen. C# development with Visual Studio - Visual Studio (Windows) [Електронний ресурс]. – Без дати. – Режим доступу: https://learn.microsoft.com/en-us/visualstudio/get-started/csharp/?view=vs-2022.
5. Hospitality Technology. STUDY: Business Expansion, Automation to Address Staffing Challenges are Top of Mind for Restaurants [Електронний ресурс]. – 2024. – Режим доступу: https://hospitalitytech.com/news-briefs/2024-01-30?article=study-business-expansion-automation-address-staffing-challenges-are-top-mind-restaurants.
6. Gonzalez, V., Jung, G. Database SQL Injection Security Problem Handling with Examples // 2019 International Conference on Computational Science and Computational Intelligence (CSCI). – Las Vegas, 2019. – С. 145–149. – DOI: 10.1109/CSCI49370.2019.00031.

7. pgAdmin. ERD Tool — pgAdmin 4 6.20 documentation [Електронний ресурс]. – Без дати. – Режим доступу: https://www.pgadmin.org/docs/pgadmin4/development/erd_tool.html.
8. Методичні рекомендації з виконання, оформлення та захисту кваліфікаційних робіт бакалаврів спеціальності 151 – «Автоматизація та комп'ютерно-інтегровані технології» / ТНТУ ім. І. Пулюя; уклад. А.Г. Микитишин, В.В. Левицький, Р.І. Королюк – Тернопіль: ТНТУ, 2023. – 81с. http://elartu.tntu.edu.ua/handle/lib/41603