

Пояснювальна записка
до дипломної роботи
бакалавр

(освітньо-кваліфікаційний рівень)

На тему «Інформаційна система для автоматизації процесів у ресторані»

Виконала: студентка 4 курсу, групи
ІСТ-19д напряму підготовки 126

„Інформаційні системи та технології”

_____ Могилкіна О.С.
(підпис)

Керівник, проф, д.т.н.

_____ Марченко Д.М.
(підпис)

Рецензент, професор, д.т.н.

_____ Захожай О.І.
(підпис)

м. Київ – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Програмування та математики
Освітній ступінь бакалавр
Спеціальність 126 Інформаційні системи та технології
(шифр і назва)

ЗАТВЕРДЖУЮ:
завідувач кафедри ІТІ
Лифар В.О.
« » 2023 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Могилкіної Олесі Сергіївни
(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційна система для автоматизації процесів у ресторані

Керівник проекту (роботи) Марченко Д.М., проф. д.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "26" 04 2023 р. № 240/15-
ОД

2. Термін подання роботи здобувача вищої освіти 16.06.2023

3. Вихідні дані до роботи матеріали переддипломної практики.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
а) аналіз предметної області;
б) вибір програмних засобів для розробки продукту;
в) розробка продукту;
г) висновки.

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

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 24.03.2023

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Ознайомлення з проблематикою досліджуваної галузі	03.04.2023-24.04.2023	
2	Аналіз існуючих інформаційних систем у ресторанному бізнесі	25.04.2023-07.05.2023	
3	Ознайомлення з необхідною літературою	03.05.2023-03.06.2023	
4	Розробка базової частини інтерфейсу	08.05.2023-20.05.2023	
5	Написання основних функцій системи	20.05.2023-28.05.2023	
6	Тестування фінальної версії продукту	28.05.2023-30.05.2023	
7	Оформлення пояснювальної записки	30.05.2023-04.06.2023	

Здобувач вищої освіти

_____ (підпис)

Могилкіна О.С.

_____ (ініціали, прізвище)

Керівник

_____ (підпис)

Марченко Д.М.

_____ (ініціали, прізвище)

РЕФЕРАТ

Дипломна робота складається з 4 розділів, 1 додаток, 27 рисунків, 2 формул.

Роботу присвячено дослідженню проблеми та перспективи застосування Інтернет технологій для автоматизації ресторанного бізнесу. Розробці вимог до функціональності інформаційних систем та схеми інформаційних потоків виробничого процесу ресторану. Зроблено аналітичний огляд літератури.

Метою даної роботи є: автоматизація виробничого процесу ресторану шляхом впровадження самостійних інформаційних потоків.

Актуальність роботи зумовлена тим, що вона дає змогу розв'язати такі завдання, як облік продажів, підвищення якості та швидкості обслуговування гостей, контроль дій персоналу, зменшення помилок під час роботи з клієнтами (людський чинник), централізоване розроблення та управління меню і преїскурантом, підвищення лояльності відвідувачів, отримання аналітичної звітності. Автоматизація кожного процесу скорочує час на обслуговування клієнтів, економить витрати, даючи змогу виконувати замовлення точно і мати повний контроль над ситуацією.

Результатом роботи є проект інформаційної системи ресторану.

Інформаційна система ресторану має виконувати такі функції:

- 1) представлення організованого онлайн-меню;
- 2) купівельний кошик замовлення з підрахунком вартості;
- 3) контроль руху замовлення - отримання замовлення, обробку замовлення на кухні або в барі, стан замовлення, доставку замовлення;
- 4) контроль руху товарних і продуктових запасів - замовлення продуктів зі складу до кухні, придатність продуктів на складі, розрахунок необхідної кількості запасів;

5) контроль руху грошових коштів закладу - можливість розрахунку за замовлення пластиковою карткою, наявність коштів у касі, на розрахунковому рахунку;

б) документальне оформлення замовлення - чеки, рахунки.

Для розробки проєкту було застосовано мову програмування С# і СУБД MySQL.

ІНФОРМАЦІЙНА СИСТЕМА, БАЗИ ДАНИХ, ІС, С#,
АВТОМАТИЗАЦІЯ

Зміст

РЕФЕРАТ	4
ВСТУП.....	7
РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА	9
1.1. Проблеми та перспективи розвитку інформаційних систем у ресторанному бізнесі	9
1.2. Принципи обробки онлайн-замовлень в ресторанному бізнесі ...	18
1.2.1. Огляд існуючих рішень.....	19
РОЗДІЛ 2. ДОСЛІДЖЕННЯ МЕТОДІВ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ.....	26
2.1. Огляд методів розробки інформаційних систем	26
2.2. Огляд методу рекурсії для розробки інформаційної системи	30
РОЗДІЛ 3. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	33
3.1. Постановка задачі	33
3.2. Вибір архітектури.....	34
3.3. Проектування системи.....	36
ВИСНОВОК.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТОК.....	55

ВСТУП

Інформаційні технології стали невід'ємною складовою сучасного суспільства і вплинули на всі сфери людської діяльності, включаючи галузь громадського харчування. Ресторанний бізнес, як важлива складова сфери послуг, постійно розвивається та залежить від ефективної організації процесів. У зв'язку з цим, розробка інформаційної системи для автоматизації процесів у ресторані є актуальною та перспективною задачею.

Цю дослідницьку роботу присвячено проблемі та перспективам інформаційної системи управління (ІСУ) на підприємстві (на прикладі ресторанного бізнесу). Ресторани зазнають швидких змін. Без будь-якого попереднього досвіду або допомоги управління рестораном може стати справжньою проблемою. Існують довгострокові та короткострокові проблеми, з якими можуть стикатися власники ресторанного бізнесу. Ресторани поступово відмовляються від звичних кілька років тому речей і замінюють їх більш ефективними та швидкими методами роботи. Інновації та технології зробили революцію в роботі ресторанів, і, правильно використовуючи програмне забезпечення для управління рестораном, ресторани можуть забезпечити свій успіх.

На додаток до управління кожним аспектом роботи ресторану, екосистема управління рестораном дає вам змогу зосередитися на інших процесах і обслуговуванні клієнтів. Прагнучи підвищити ефективність і продуктивність персоналу, більшість ресторанів уже розробили систему управління рестораном. У цій роботі розглядається рішення у вигляді інформаційної системи для управління ресторанами.

Метою даної дипломної є розробка інформаційної системи, спрямованої на автоматизацію роботи ресторану. Ця програма призначена для обробки онлайн-замовлень і дозволить підвищити ефективність роботи персоналу та забезпечити більш якісне обслуговування клієнтів.

Для досягнення поставленої мети в роботі мають бути вирішені такі завдання:

1. Вивчення існуючих інформаційних систем, що застосовуються в ресторанному бізнесі, та аналіз їх переваг та недоліків.

2. Визначення вимог до інформаційної системи, що розробляється, на основі потреб та особливостей роботи ресторану.

3. Проектування архітектури інформаційної системи та її компонентів, включаючи інтерфейси для онлайн-замовлень, систему керування замовленнями.

4. Розробка та реалізація інформаційної системи.

5. Тестування та налагодження розробленої системи для забезпечення її стабільної та коректної роботи.

Остаточне вирішення цих завдань дозволить створити інноваційну інформаційну систему, що сприятиме автоматизації роботи ресторану в обробці онлайн-замовлень, що, у свою чергу, підвищить якість обслуговування, покращить процеси всередині ресторану та створить міцну основу для успішної діяльності в конкурентному середовищі.

Результати даної дипломної роботи будуть корисними для власників ресторанів, керівників галузі громадського харчування, а також для фахівців у галузі інформаційних технологій, які зацікавлені в розробці подібних систем.

РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА

1.1. Проблеми та перспективи розвитку інформаційних систем у ресторанному бізнесі

Однією з перспективних галузей, що розвиваються, є ресторанний бізнес. Досягнення успіху у розвитку будь-якого бізнесу, зокрема ресторанного, залежить від багатьох факторів, одним з яких є використання інформаційних технологій. Процес управління ресторанним бізнесом є доволі складним та вимагає: контролю за обліковим процесом та поведінкою персоналу, аналізу транзакцій, обліку надходження продуктів, формування вартості страв і напівфабрикатів, процедур списання продуктів, дотримання санітарних і технологічних норм. Потреба автоматизації всіх цих процесів впливає з необхідності врахування великої кількості деталей.

Зручність автоматизації процесів у закладах громадського харчування очевидна не тільки з погляду власника, але й з позиції клієнтів, оскільки інформаційні системи дають змогу виконувати розрахунки з відвідувачами оперативніше, забезпечити систему знижок для зареєстрованих клієнтів, здійснювати обслуговування на основі черги, забезпечити меню усіма необхідними інгредієнтами та заощадити час. Аналіз останніх досліджень та публікацій Інформаційні технології сьогодні можна класифікувати за такими ознаками: способом реалізації в інформаційній системі, ступенем охоплення завдань управління, класами реалізованих технологічних операцій, типом користувацького інтерфейсу, варіантами використання мережі, що обслуговує предметну область. [1]

Управління – найважливіша функція, без якої неможлива цілеспрямована діяльність будь-якої соціально-економічної, організаційно-виробничої системи (підприємства, організації, території). Систему, що реалізує функції управління, називають системою управління. Найважливішими функціями, реалізованими цією системою, є прогнозування, планування, облік, аналіз, контроль і регулювання. Управління пов'язане з обміном інформацією між компонентами

системи, а також системи з довкіллям. У процесі управління одержують відомості про стан системи в кожен момент часу, про досягнення (або недосягнення) заданої мети з тим, щоб впливати на систему і забезпечити виконання управлінських рішень. Автоматизована інформаційна система – це сукупність інформації, економіко-математичних методів і моделей, технічних, програмних, технологічних засобів і фахівців, призначена для опрацювання інформації та прийняття управлінських рішень.

Отже, інформаційну систему можна визначити з технічного погляду як набір взаємопов'язаних компонентів, які збирають, опрацьовують, накопичують і розподіляють інформацію, щоб підтримати прийняття рішень і управління в організації. На додаток до підтримки прийняття рішень, координації й управління інформаційні системи можуть також допомагати менеджерам аналізувати проблеми, роблять видимими комплексні об'єкти й створюють нові вироби. Оскільки є різні інтереси, особливості та рівні в організації, існують різні види інформаційних систем.

Жодна єдина система не може повністю забезпечувати повні потреби організації. Організацію можна поділити на рівні: стратегічний, управлінський, знання та експлуатаційний; і на функціональні області типу продажу і маркетингу, виробництва, фінансів, бухгалтерського обліку та людських ресурсів.

Системи створюються, щоб обслуговувати ці різні організаційні інтереси. Різні організаційні рівні обслуговують чотири головні типи інформаційних систем: системи із експлуатаційним рівнем, системи рівня знання, системи рівня управління і системи зі стратегічним рівнем. Системи експлуатаційного рівня підтримують керуючих операціями, стежать за елементарними діями організації типу продажу, платежів, переводять у готівку депозити, платіжну відомість. Основна мета системи на цьому рівні полягає в тому, щоб відповісти на звичайні питання і проводити потоки транзакцій через організацію. [4]

Щоб відповідати на ці види питань, інформація взагалі повинна бути легкодоступною, оперативною і точною. Системи рівня знань підтримують

знання працівників і тих, хто опрацьовує дані в організації. Мета систем рівня знань полягає в тому, щоб допомогти інтегрувати нове знання в бізнес і допомагати організації управляти потоком документів. Системи рівня знань, особливо у формі робочих станцій і офісних систем, сьогодні є найбільш швидко зростаючими додатками в бізнесі.

Системи рівня управління розроблено, щоб обслуговувати контроль, управління, прийняття рішень та адміністративні дії середніх менеджерів. Вони визначають, чи добре працюють об'єкти і періодично сповіщають про це. Наприклад, система управління переміщеннями повідомляє про переміщення загальної кількості товару, рівномірності роботи торгового відділу та відділу, що фінансує витрати для службовців у всіх розділах компанії, зазначаючи, де фактичні витрати перевищують бюджети. [2]

Деякі системи рівня управління підтримують незвичайне прийняття рішень. Вони мають тенденцію зосереджуватися на менш структурних рішеннях, для яких інформаційні вимоги не завжди зрозумілі. Системи стратегічного рівня – це інструмент допомоги керівникам вищого рівня, які готують стратегічні дослідження і тривалі тренди у фірмі та в діловому оточенні. Їх основне призначення – приводити у відповідність зміни в умовах експлуатації з існуючою організаційною можливістю.

Інформаційні системи можуть також бути функціонально диференційовані.

Головні організаційні функції типу продажу і маркетингу, виробництва, фінансів, бухгалтерського обліку та людських ресурсів обслуговуються власними інформаційними системами. У великих організаціях підфункції кожної з цих головних функцій також мають власні інформаційні системи. Наприклад, функція виробництва могла б мати системи для управління запасами, управління процесом, обслуговування організації, автоматизованого розроблення і матеріального планування вимог. Типова організація має системи різних рівнів: експлуатаційну, управлінську, знань і стратегічну для кожної функціональної області. Наприклад, комерційна функція має комерційну

систему на експлуатаційному рівні, щоб записувати щоденно комерційні дані і опрацьовувати замовлення.

Система рівня знання створює відповідні дисплеї для демонстрації виробів фірми. Системи рівня управління відстежують щомісячні комерційні дані всіх комерційних територій і доповідають про території, де продаж перевищує очікуваний рівень або падає нижче очікуваного рівня. Система прогнозу пророкує комерційні тренди протягом п'ятирічного періоду – обслуговує стратегічний рівень.

Нові інформаційні системи втілюють передовий досвід управлінських технологій. При цьому вони оптимізують бізнес-процеси відповідно до останніх досягнень теорії та практики менеджменту. У зв'язку з використанням таких технологій інформація стає важливою складовою виробничого процесу і витісняє з нього традиційні компоненти: природні ресурси, працю і капітал

Ефективна економічна діяльність сьогодні ґрунтується на перетворенні інформації, яке передбачає цілеспрямований обмін впорядкованими даними (отримання та передавання відомостей) з іншими структурами і людьми. Можна сказати, що інформація сприяє впорядкованості та стійкості економічної системи. Корисність інформаційного продукту завжди відносна, оскільки визначається його цінністю для конкретного користувача. Однак не завжди можна заздалегідь передбачити навіть область застосування деяких інновацій.

Інформаційна складова впливає не тільки на структуру капіталу, а й на виробничі, ринкові параметри продукції. Підприємства сфери послуг мають свою специфіку і відрізняються від промислових підприємств. Первинною і обов'язковою умовою виробництва послуг є безпосередній контакт із замовником. У зв'язку з цим виникає потреба у використанні систем управління, скерованих на клієнтів. Інформаційні технології управління підприємствами сфери послуг орієнтуються не стільки на технологічний, скільки на комунікаційний аспект виробництва. [3]

На відміну від підприємств, що займаються виробництвом продукції, інформаційні складові яких орієнтовані на їхні внутрішні потреби, інформаційні

системи підприємств сфери послуг повинні насамперед бути орієнтовані на поліпшення сервісу для різних клієнтів. Найкращою комбінацією інформаційних систем для цих підприємств є система як основа інформаційного середовища в комбінації з системою для взаємодії з клієнтами.

Для аналізу даних в інформаційну інфраструктуру підприємства інтегрують систему-аналітик. Використання саме цієї системи збільшує ефект і капіталізацію підприємства від застосування інформаційних технологій набагато більше, ніж встановлення перших двох систем, але водночас без них такого ефекту досягти було б неможливо. Існує ще одна особливість застосування нових інформаційних технологій на підприємствах сфери послуг. Використання таких технологій збільшує їх конкурентоспроможність.

Якщо на Заході інформаційні технології є обов'язковою умовою існування підприємств і всі їх використовують, то в Україні організації, що впровадили інформаційні технології, мають додаткові можливості для конкуренції за допомогою застосування передового західного досвіду. Інформаційні технології сьогодні сприяють трансформації самого менеджменту. Основним їх змістом є не набір технічних інновацій, а сукупність світового управлінського досвіду і рішень, втілених у відповідному інструментарії за допомогою сучасних способів опрацювання та зберігання інформації.

Найкращим способом оцінювання ефективності використання таких технологій є зіставлення витрат на їх купівлю, впровадження і приросту капіталізації підприємства після їх впровадження. Для виявлення приросту нематеріальних активів підприємства, які підвищуються із застосуванням сучасних інформаційних технологій, одним з найкращих способів є система збалансованих показників. [5]

Під час реалізації нових інформаційних технологій управління підприємством виникає необхідність перетворення даних на інформацію, а інформації – на знання. Підходи до реалізації цього ланцюжка дає концепція менеджменту знань. Сьогодні застосування інформаційних систем для автоматизації виробництва набирає все більшої популярності в ресторанному

бізнесі. Інформаційні технології роблять операції організації та управління підприємством ефективнішими і легшими.

Інформаційна технологія – це сукупність методів, виробничих процесів та програмнотехнічних засобів, які об'єднані у технологічний ланцюжок, що забезпечує виконання інформаційних процесів з метою підвищення їх надійності та оперативності і зниження трудомісткості під час використання інформаційного ресурсу. [2] Підвищення конкурентоспроможності українських підприємств можна тільки на основі прискорення інноваційних процесів, які повинні забезпечити рентабельне використання нововведень у вигляді нових інформаційних технологій, видів продукції і послуг, організаційнотехнічних і соціально-економічних рішень виробничого, фінансового, комерційного, адміністративного або іншого характеру.

Стратегічні цілі інформаційних технологій: забезпечити розвиток бізнесу, його керованість і якість, конкурентоспроможність, зниження вартості виконання бізнес-процесів. Більшість підприємств ресторанного бізнесу роблять ставку на автоматизацію своєї основної діяльності. Зазвичай автоматизують робочий процес насамперед ті підприємства, які мають об'ємні обсяги продажів. Таким ресторанам необхідна оперативність і керованість, а вони, навпаки, стикаються зі збільшенням різної шаблонної діяльності.

Необхідно позбавити працівників від неї і ефективніше використовувати їх робочий час. Тому для прийняття важливих рішень потрібно знати у точних числових значеннях стан справ на певний момент часу. А для великих і середніх підприємств найточніші дані можна отримати завдяки автоматизації та залученню інформаційних систем. Також сучасні програми автоматично визначають витрати інгредієнтів на кожну страву, списують потрібну кількість продуктів і розраховують собівартість страв.

І, нарешті, інформаційні системи значно полегшують і роблять суворішим облік продуктів і страв на декількох кухнях і точках реалізації, формують меню для залу й преїскурант барної продукції. Завдяки автоматизації спрощується процес руху продукції, наприклад, надходження продуктів безпосередньо на

кухню або місце реалізації. Автоматизація підприємства ресторанного господарства сьогодні потрібна, оскільки вона набагато спрощує процес роботи і допомагає звести до мінімуму збитки і зайві витрати.

В результаті автоматизації підвищуються якість сервісу і престиж закладу; швидше опрацьовуються замовлення та обслуговуються відвідувачі; поліпшуються узгодженість і контроль за роботою персоналу; ведеться облік руху матеріальних цінностей і руху товарів; скорочуються тимчасові витрати на пошук та аналіз інформації. Отже, автоматизація ресторанного господарства допоможе зробити підприємство конкурентоспроможним і затребуваним на ринку.

Необхідною умовою успішного розвитку ресторану стає впровадження системи автоматизації управління та обліку, яка дає змогу оптимізувати всі ключові бізнес-процеси, створити ефективну систему лояльності постійних гостей, підвищити якість виробництва, запобігти розкраданням та зловживанням персоналу. Кожен ресторатор знає, що його дохід безпосередньо залежить від того, наскільки заклад здатний витримувати конкуренцію.

Технології, які дають змогу автоматизувати ресторанний бізнес, швидко розвиваються, тому успішний ресторатор має бути в курсі останніх новинок і переваг і за необхідності оновлювати та закуповувати нове обладнання для ресторанів. Останні технологічні досягнення, які успішно застосовуються в галузі ресторанного господарства:

Приймання кредитних карток. За статистикою, клієнти з кредитками купують набагато більше. Сьогодні транзакції займають секунди, а клієнтів з картами стає все більше. Крім того, тепер у разі оплати за кредитною картою не треба підписувати чек, і власник картки може передати її будь-кому, доручивши сплатити рахунок.

Мобільні термінали. Для приймання платежів за банківськими картами з'явилися бездротові мобільні термінали. За всієї зручності обслуговування за допомогою мобільних пристроїв чек пробивають відразу на касі або в реєстраторі. Також з'явилася система радіовиклику офіціанта. За одного

натискання кнопки офіціант має негайно з'явитися у столика. Мобільні термінали допомагають оформляти замовлення безпосередньо біля столика, і замовлення миттєво надходить на кухню або в бар.

Принтери на кухні і в барі роздруковують замовлення, а бармен і кухар його виконують. Мобільні термінали дозволяють не тільки обслуговувати клієнтів швидко і грамотно, але й надають ресторану особливого іміджу. Мобільний термінал – це максимальна якість і швидкість обслуговування в ресторані. Для застосування мобільних рішень заклад має бути оснащено “Wi-Fi”-мережею. Програмне забезпечення.

Програмне рішення – це готова бізнес-схема. Новачкові вона допоможе мінімізувати ризики. Як правило, всі новинки техніки з обліку та обслуговування вже входять до цільної, розробленої системи програмного забезпечення, і можна не ламати голову над тим, що придбати і наскільки покупка впишеться в роботу ресторану. Питання лише відповідності специфіці конкретного підприємства. Вибираючи програмне забезпечення, слід враховувати конкретні особливості бізнесу, а також перспективи.

Автоматизація кафе та ресторану неможлива без впровадження допоміжного обладнання, такого як

- pos комп'ютер;
- pos система;
- касовий апарат;
- грошовий ящик;
- принтер чеків;
- програмована клавіатура;
- кнопки виклику офіціанта;
- сенсорний монітор;
- інші допоміжні складові.

Запропонована автоматизація системи обслуговування в закладах ресторанного бізнесу має позитивні аспекти, а саме:

- орієнтована на сучасні умови діяльності закладів;

— крім цього, не має аналогів у своїх принципах, існуючі напрями скеровані на розширення асортименту продукції, задоволення потреб різних контингентів споживачів та моделювання оптимальних варіантів просторової структури організаційно-обслуговувальної системи за різних методів обслуговування;

— під час реалізації передбачає одержання результатів оцінювання якості всіх елементів системи обслуговування, а також інтегральної оцінки

— якості;

— дає змогу забезпечувати однотипним закладам високий рівень сервісу, розвивати своєрідні відмінності в обслуговуванні, що є важливим чинником в умовах жорсткої конкуренції, яка здебільшого в групі однотипних закладів зводиться до регульованої цінової тактики.

Організація системи обслуговування повинна забезпечувати раціональне використання ресурсів з оптимальним застосуванням різних типів виконання процесів системи обслуговування із забезпеченням максимально високих можливих результатів, які необхідні для досягнення запланованого рівня сервісу. Доцільність розроблення інтелектуальної інформаційної системи надання послуг у ресторанному бізнесі полягає в такому:

— впровадження єдиного центру управління потоками даних (головний комп'ютер розмістити в бухгалтерії, куди надходитиме інформація з залу та кухні);

— система заміняє значні людські ресурси;

— з погляду економії часу та грошей система є вигіднішою, ніж інші подібні системи;

— можливість надання чітко сформованих даних про діяльність ресторану.

Сьогодні на ринку України є небагато програмних продуктів, які можуть підтримувати прийняття рішень у процесі обліку та аналізу діяльності підприємств ресторанного бізнесу. Крім того, більшість з них розроблені за

кордоном і тому мало застосовні до українських реалій (не враховують особливостей законодавчої бази та економічної системи України).

1.2. Принципи обробки онлайн-замовлень в ресторанному бізнесі

Система онлайн-замовлення їжі - одна з останніх новинок сервісу швидкого харчування. За допомогою цього методу їжа замовляється онлайн і доставляється замовнику. Це стало можливим завдяки використанню електронної платіжної системи. Клієнти розплачуються своїми кредитними картами або по факту отримання замовлення [9]. В цілому, обробка онлайн-замовлень у ресторанному бізнесі зазвичай включає низку кроків, щоб забезпечити плавний та ефективний робочий процес. Хоча конкретний процес може відрізнятись залежно від ресторану та його діяльності, ось деякі загальні принципи, які слід враховувати:

Платформа онлайн-замовлень. Необхідно обрати надійну та зручну платформу онлайн-замовлень, яка добре інтегрується з роботою ресторану. Це може бути служба третьої сторони або спеціальне рішення.

Керування меню. Важливо підтримувати актуальне та точне меню на платформі онлайн-замовлень. Треба переконатись, що всі товари, описи, ціни та наявність правильно відображені. Будь-які зміни або оновлення меню слід негайно вносити.

Розміщення замовлення. Важливо надати чіткі інструкції на веб-сайті або в додатку, щоб клієнти могли легко орієнтуватися в процесі онлайн-замовлення. Переконайтесь, що платформа замовлення є інтуїтивно зрозумілою та зручною для користувача, дозволяючи клієнтам вибирати товари, налаштовувати свої замовлення та додавати будь-які спеціальні інструкції.

Підтвердження замовлення. Клієнти мають отримувати автоматичне повідомлення з підтвердженням або електронний лист після успішного розміщення замовлення. Це підтвердження має містити деталі замовлення, приблизний час доставки або отримання, а також будь-яку відповідну платіжну інформацію.

Обробка платежів. Клієнти мають отримати доступ до безпечних та зручних варіантів оплати, такі як кредитна/дебетова картка, мобільні гаманці або платіжні онлайн-платформи. Платіжний шлюз має бути надійним, а конфіденційна інформація клієнтів обробляється безпечно.

Виконання замовлення. Важливо встановити спрощений процес виконання замовлення. Для цього необхідно призначити спеціалізований персонал для керування вхідними замовленнями та забезпечення їх швидкої підготовки до отримання або доставки. Чітко повідомте клієнтам очікуваний час підготовки.

Канали зв'язку. Ресторатори мають підтримувати відкриті лінії зв'язку з клієнтами під час обробки замовлення. Важливо надати контактну інформацію, таку як номер телефону чи електронну адресу, щоб клієнти могли зв'язатися, якщо у них виникнуть запитання чи проблеми.

Відстеження замовлень. Система має забезпечувати функцію відстеження замовлень, щоб клієнти могли контролювати статус свого замовлення. Це можна зробити за допомогою автоматичних сповіщень або функції відстеження на вашому веб-сайті чи в додатку.

Аналіз даних. Дані, отримані з онлайн-замовлень можна використовувати, щоб отримати уявлення про вподобання клієнтів, моделі замовлень та інші відповідні показники. Ця інформація допоможе прийняти рішення на основі даних, щоб покращити пропозиції та оптимізувати роботу [9].

В цілому ресторанам не потрібна дорога система обробки для автоматизації робочого процесу. Є служби, які надають форми замовлень і їх можна підключити до автоматизованого робочого процесу затвердження замовлень на покупку. Використовуючи навіть найпростіші таблиці, в ресторанах можна автоматично заповнювати дані форми в електронні таблиці та відстежувати замовлення в міру їх надходження [9].

1.2.1. Огляд існуючих рішень

Сьогодні існує безліч рішень для бізнесу в сфері харчування, які допомагають в обробці замовлень. Серед них є рішення як для обробки онлайн-

замовлень, так і зроблених у самому ресторані або кафе. Нижче наведено декілька таких систем.

Poster POS. Це програмне забезпечення для управління точкою продажів (POS), яке призначене для автоматизації бізнес-процесів у роздрібній торгівлі та громадському харчуванні. Воно дозволяє власникам та керуючим підприємств ефективно керувати своїми операціями, відстежувати продажі, керувати складом та контролювати фінанси.

Принцип роботи Poster POS ґрунтується на використанні хмарних технологій. Коли ви встановлюєте Poster POS на своєму пристрої, всі дані про ваш бізнес і операції зберігаються у хмарі, а не на локальному сервері. Це дозволяє отримати доступ до системи з будь-якого місця, де є інтернет, і використовувати її на різних пристроях, таких як комп'ютери, планшети або смартфони.[6]

Користувачі можуть налаштувати меню, додавати продукти, встановлювати ціни та контролювати доступ до системи для працівників. Під час проведення продажу співробітники можуть використовувати касовий апарат або мобільний пристрій із встановленим Poster POS для сканування товарів, прийняття оплати та видачі чека покупцю. Вся інформація про продаж, інвентар та фінансові операції автоматично синхронізується з хмарою і доступна в режимі реального часу для аналізу та управління.

Poster POS також надає додаткові функції, такі як облік робочого часу співробітників, управління програмами лояльності та звітність про продаж та фінанси. Система також інтегрується з іншими платіжними системами та онлайн-сервісами, що дає змогу розширити можливості вашого бізнесу. Загалом Poster POS полегшує управління бізнесом у роздрібній торгівлі та громадському харчуванні, підвищує ефективність роботи та надає власникам повний контроль над своїми операціями.

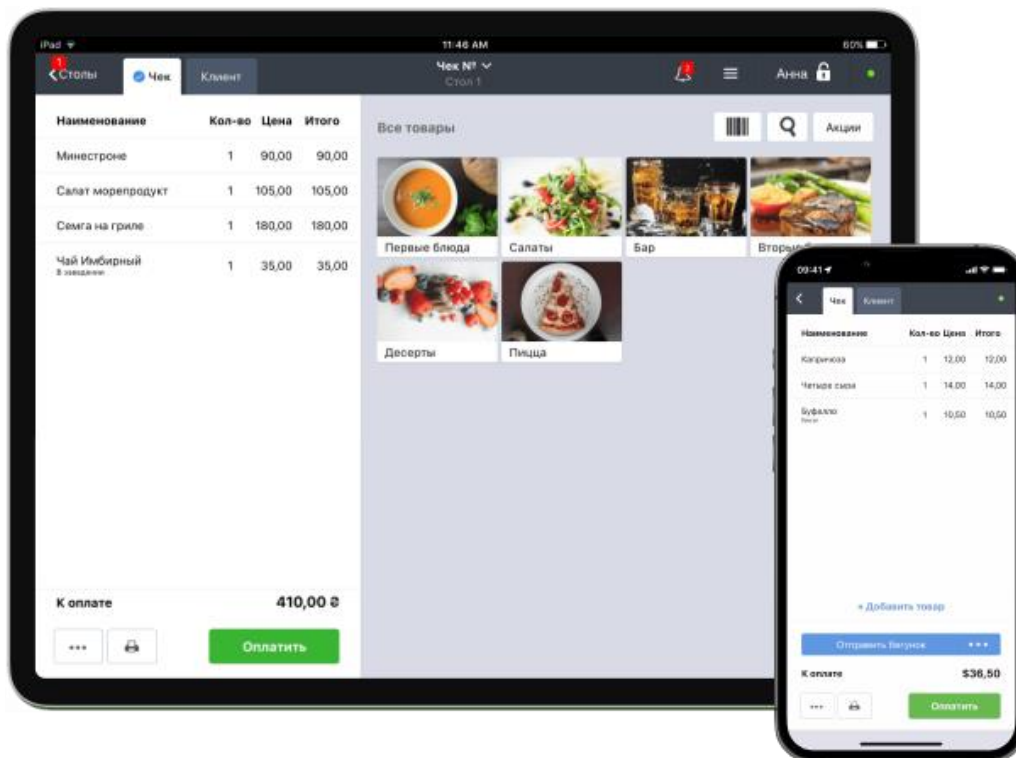


Рисунок 1.1. – Poster POS

R-Keeper. Це програмне забезпечення для управління рестораном, яке використовується в індустрії громадського харчування. Воно надає комплексні рішення для автоматизації операцій у ресторані, включаючи управління замовленнями, облік товарів, управління персоналом та звітність.

Принцип роботи R-Keeper ґрунтується на інтеграції комп'ютерів, периферійного обладнання та програмного забезпечення для створення єдиної системи управління рестораном. В основі системи лежить клієнт-серверна архітектура, де сервер обробляє всі дані та команди, а клієнтські пристрої (комп'ютери, термінали або мобільні пристрої) використовуються для доступу до функціональності системи.

Основні компоненти R-Keeper включають:

1. Термінали офіціанта. Офіціанти можуть приймати замовлення, редагувати їх, передавати на кухню та отримувати інформацію про статус замовлення.
2. Кухонні монітори. Дозволяють персоналу на кухні отримувати замовлення та відстежувати їх виконання.

3. База даних. Зберігає інформацію про меню, інгредієнти, ціни та інші ресторанны дані.

4. Облікові системи. Відстежують фінансову інформацію, таку як продажі, оплати, податки та звітність.

5. Мобільні програми. Можуть бути використані для прийняття замовлень та обробки платежів на мобільних пристроях.

R-Keerreg дозволяє автоматизувати та оптимізувати процеси в ресторані, спростити управління замовленнями, облік товарів та персоналом, а також покращити обслуговування клієнтів. Система має багатofункціональні можливості, які можуть бути налаштовані під конкретні потреби та вимоги кожного ресторану.



Рисунок 1.2. – r_keeper

POS Sector. POS Sector – це скорочення від "Point of Sale Sector", що означає сектор або галузь точок продажу. POS Sector – це сфера бізнесу, пов'язана з продажем товарів та послуг безпосередньо споживачам через точки продажу, такі як магазини, ресторани, супермаркети, бари тощо.

Принцип роботи POS Sector залежить від конкретної організації та системи POS (точка продажу). Зазвичай у POS Sector включено кілька компонентів:

1. POS-термінали: це фізичні пристрої, зазвичай касові апарати або комп'ютери із встановленим спеціалізованим програмним забезпеченням POS. Вони використовуються для реєстрації продажів, обробки платежів та управління інвентаризацією.

2. Програмне забезпечення POS: це спеціалізоване програмне забезпечення, яке встановлюється на POS-термінали. Воно надає функції обробки обробки, управління інвентаризацією, обліку платежів та інших операцій, що з точками продажу.

3. Пристрої для обробки платежів: це обладнання, що використовується прийому платежів від клієнтів. Це можуть бути карт-рідери, термінали для безконтактних платежів, сканери штрих-кодів та інші пристрої, що дозволяють проводити транзакції.

4. Інтеграція з іншими системами: POS Sector зазвичай інтегрується з іншими системами, такими як системи обліку, системи управління запасами та системи управління клієнтськими відносинами (CRM). Це дозволяє обміну даними між різними системами та спрощує управління бізнесом.

Загалом, принцип роботи POS Sector полягає в наступному: при проведенні продажу співробітник вводить інформацію про товари або послуги до POS-терміналу, який потім обчислює загальну вартість, обробляє платежі та оновлює інформацію в системі обліку та інвентаризації. Це дозволяє організаціям ефективно управляти продажами, контролювати запаси та полегшувати взаємодію з клієнтами.



Рисунок 1.3. - POS Sector

Найважливішим елементом виконання замовлення є робочий процес, що забезпечує належну підготовку замовлень клієнтів та їх доставку у потрібне місце. Залежно від масштабу бізнес-операцій, обробка замовлень може відбуватися в розподільчому центрі або на складі, де комплектувальники, сортувальники та пакувальники працюють синхронно для виконання замовлення, або це може виконуватися однією людиною або невеликою групою.

Хоча обробка замовлень може виконуватися вручну за допомогою ручки та паперу, зі зростанням компанії її складність зростає, а життєздатність цього методу знижується. Ось де програмне забезпечення для обробки замовлень демонструє свою цінність — воно може мінімізувати людський фактор і забезпечити задоволеність клієнтів, незалежно від масштабу операції.

Програмне забезпечення для обробки замовлень зберігає та обмінюється даними про замовлення, перевіряє наявність на складі та відстежує доставку замовлень, і все це може допомогти забезпечити точне та своєчасне виконання замовлень. Це важливо, тому що точність та надійність підвищують задоволеність клієнтів, а задоволеність клієнтів веде до збільшення продажів. Система управління замовленнями (OMS) може бути інтегрована з іншим програмним забезпеченням, таким як виставлення рахунків та бухгалтерський

облік, щоб об'єднати обробку замовлень та всі пов'язані з ними завдання на одній централізованій платформі.[7]

Висновок по першому розділу

Обробка замовлення - це процес, який відбувається після того, як клієнт розміщує замовлення. [1] Це починається з підтвердження того, що продукти є на складі, потім вибираються товари з інвентарю та вирушають у зону сортування. Потім кожне окреме замовлення складається, ретельно упаковується, маркується та відправляється на адресу клієнта або безпосередньо (для підприємств з меншими обсягами відвантаження), або шляхом консолідації замовлень (для підприємств з більшими обсягами відвантаження).

Обробка замовлень є ключовим компонентом для виконання замовлень, а ефективні робочі процеси обробки замовлень можуть допомогти клієнтам залишатися задоволеними. Цей робочий процес включає збір інвентарю, сортування товарів, упаковку замовлень і їх відправку. Програмне забезпечення для обробки замовлень може принести компанії значні переваги, оскільки воно допомагає автоматизувати складські процеси, підвищує точність та скорочує час, необхідний для виконання замовлень.

Існують системи обробки замовлень, які допомагають зробити виконання замовлень ефективнішим та найточнішим. Для цього вони збирають і зберігають усі дані, пов'язані із замовленням – від доступних запасів до відстеження відвантаження – у центральній базі даних. Це дозволяє відповідним бізнес-відділам легко отримати доступ до інформації про замовлення в режимі реального часу в будь-який момент. Наприклад, відділ доставки може використовувати систему, щоб визначити, які замовлення необхідно надіслати і куди.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ МЕТОДІВ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ

2.1. Огляд методів розробки інформаційних систем

Процес розробки хорошого програмного забезпечення зазвичай не такий простий, як сісти та написати код. Щоправда, іноді програміст може швидко написати коротку програму на вирішення будь-якої завдання. Але здебільшого створення програмного забезпечення — це ресурсомісткий процес, у якому беруть участь кілька різних груп людей організації. Розглянемо кілька різних методологій розробки програмного забезпечення.

Життєвий цикл розробки систем

Перша методологія розробки - це життєвий цикл розробки систем (SDLC). Ця методологія була вперше розроблена у 1960-х роках для управління великими програмними проектами, пов'язаними з корпоративними системами, що працюють на мейнфреймах. Це дуже структурована методологія, що не допускає ризику, призначена для управління великими проектами, що включають кілька програмістів і систем, які можуть вплинути на організацію.

Існують різні визначення методології SDLC, але більшість із них містять такі етапи.

1. Попередній аналіз. На цьому етапі виконується перевірка запиту. Чи можливе створення рішення? Які альтернативи? Що зараз робиться для цього? Ключовою частиною цього кроку є техніко-економічний аналіз, який включає аналіз технічної здійсненності (чи можна це створити?), економічної здійсненності (чи можемо ми дозволити собі це зробити?) і юридичної здійсненності (чи можна нам це зробити? зробити це?). Цей крок важливий для визначення того, чи слід розпочинати проект.

2. Системний аналіз. На цьому етапі один або кілька системних аналітиків працюють із різними групами зацікавлених сторін, щоб визначити конкретні вимоги до нової системи. На цьому етапі не виконується програмування. Натомість процедури документуються, ключові гравці опитуються та розробляються вимоги

до даних, щоб отримати загальну картину того, що саме має робити система. Результатом цього етапу є документ системних вимог.

3. Системний дизайн. На цьому етапі проєктувальник бере документ із системними вимогами, створений на попередньому етапі, та розробляє конкретні технічні деталі, необхідні для системи. Саме на цьому етапі бізнес-вимоги перетворюються на конкретні технічні вимоги. Тут розробляється дизайн інтерфейсу користувача, бази даних, введення і виведення даних і звітності. Результатом цього етапу є проєктний документ системи. У цьому документі буде все, що потрібне програмісту для фактичного створення системи.

4. Програмування. Код пишеться на етапі програмування. Використовуючи документ із проєктування системи як керівництво, програміст (чи група програмістів) розробляє програму. Результатом цього етапу є початкова робоча програма, що відповідає вимогам, викладеним на етапі системного аналізу, та проєкту, розробленому на етапі системного проєктування.

5. Тестування. На етапі тестування програма, розроблена попередньому етапі, проходить серію структурованих тестів. Перший це модульний тест, який перевіряє окремі частини коду на наявність помилок або багів. Далі слідує системний тест, при якому різні компоненти системи тестуються, щоб переконатися, що вони працюють разом належним чином. Нарешті, приймальний тест дозволяє тим, хто використовує програму, протестувати систему, щоб переконатися, що вона відповідає їхнім стандартам. Будь-які помилки, помилки або проблеми, виявлені під час тестування, усуваються, а потім знову тестуються.

6. Виконання. Після того, як нова система розроблена та протестована, її необхідно впровадити в організації. Цей етап включає навчання користувачів, надання документації і перехід з будь-якої попередньої системи на нову систему. Впровадження може набувати різних форм залежно від типу системи, кількості та типу користувачів, а також від того, наскільки терміново потрібно ввести систему в експлуатацію. Ці різні форми реалізації розглядаються далі у цьому розділі.

7. Обслуговування. Цей завершальний етап відбувається після завершення етапу реалізації. На цьому етапі система має структурований процес підтримки:

виправляються повідомлення про помилки, а запити на нові функції оцінюються та реалізуються; оновлення системи та резервне копіювання виконуються на регулярній основі.

Методологію SDLC іноді називають методологією водоспаду, щоб уявити, як кожен крок є окремою частиною процесу; тільки коли один крок завершено, може розпочатись інший крок. Після кожного кроку організація має вирішити, чи переходити до наступного кроку чи ні. Ця методологія піддавалася критиці через те, що вона досить жорстка. Наприклад, зміна вимог не допускається після початку процесу. Програмне забезпечення недоступне до закінчення етапу програмування.

Методологія RAD

Швидка розробка програм (RAD) – це методологія розробки програмного забезпечення (або системної розробки), яка спрямована на швидке створення робочої моделі програмного забезпечення, отримання відгуків від користувачів та подальше використання цих відгуків для оновлення робочої моделі. Після кількох ітерацій розробки розробляється та впроваджується остаточний варіант.[8]

Методологія RAD складається з чотирьох етапів:

1. Планування вимог. Цей етап аналогічний етапам попереднього аналізу, системного аналізу та проектування SDLC. На цьому етапі визначаються загальні вимоги до системи, визначається команда та визначається здійсненність.

2. Користувальницький дизайн. На цьому етапі представники користувачів працюють із системними аналітиками, дизайнерами та програмістами для інтерактивного створення дизайну системи. Одним із методів роботи з усіма цими різними зацікавленими сторонами є так звана JAD-сесія. JAD – це абревіатура від спільної розробки програм. Сесія JAD збирає всіх зацікавлених осіб для структурованого обговорення системи дизайну. Розробники додатків теж сидять на цій зустрічі та спостерігають, намагаючись зрозуміти суть вимог.

3. Будівництво. На етапі побудови розробники програми, працюючи з користувачами, створюють таку версію системи. Це інтерактивний процес і зміни можуть бути внесені в міру того, як розробники працюють над програмою. Цей

крок виконується паралельно з етапом дизайну користувача ітеративно, поки не буде розроблена прийнятна версія продукту.

4. Перемикання. На цьому етапі, аналогічному етапу реалізації SDLC система запускається. Тут завершуються всі кроки, необхідні переходу від попереднього стану до використання нової системи.

Методологія RAD набагато стисліша, ніж SDLC. Багато кроків SDLC об'єднані, і основна увага приділяється участі користувачів та ітераціям. Ця методологія краще підходить для невеликих проектів, ніж SDLC, і має додаткову перевагу, що користувачі можуть залишати відгуки протягом усього процесу. SDLC вимагає більше документації та уваги до деталей та добре підходить для великих ресурсомістких проектів. RAD більше підходить для невеликих проектів, які менш вимогливі до ресурсів та потребують швидкої розробки.

Гнучкі методології

Гнучкі методології - це група методологій, які використовують поступові зміни з упором на якість та увагу до деталей. Кожен інкремент випускається у певний період (званий тимчасовим інтервалом), створюючи регулярний графік випуску з дуже конкретними цілями. Хоча вони вважаються окремою від RAD методологією, вони поділяють деякі з тих самих принципів: ітеративна розробка, взаємодія з користувачем, можливість змін. Гнучкі методології засновані на Маніфесті Agile, вперше опублікованому в 2001 році.

До характеристик гнучких методів належать:

- невеликі міжфункціональні групи, до складу яких входять члени команди розробників та користувачі;
- щоденні статусні зустрічі для обговорення стану проекту;
- короткі часові інтервали (від днів до одного-двох тижнів) для виконання кожної зміни; і
- наприкінці кожної ітерації завершується працюючий проект для демонстрації заінтересованим сторонам.

Метою гнучких методологій є забезпечення гнучкості ітеративного підходу у забезпеченні якості продукту.

Насправді зараз дуже багато різних методологій для розробки ІС. У цьому випадку було обрано метод рекурсії.

2.2. Огляд методу рекурсії для розробки інформаційної системи

Метод рекурсії є одним з підходів до розробки інформаційних систем. Рекурсія в програмуванні означає виклик функції зсередини самої себе. У випадку розробки інформаційної системи, рекурсія може бути застосована для створення структури системи та обробки даних.

Основна ідея рекурсивного підходу полягає в тому, що складна задача розбивається на менші підзадачі, що мають подібну структуру. Кожна підзадача вирішується за допомогою виклику тієї ж функції (або методу) для менших підзадач, поки не буде досягнуто базового випадку або критерію зупинки.

Для розробки інформаційної системи з використанням рекурсії можуть бути виконані наступні кроки:

1. Визначення основного завдання системи: Визначте, що саме має виконувати система та які є її основні функції.
2. Розбиття завдання на менші підзадачі: Розбийте основне завдання на менші підзадачі, які можна вирішити окремо.
3. Визначення базового випадку: Визначте базовий випадок або критерій зупинки, коли рекурсивний виклик перестає відбуватись. Це може бути досягнення певного стану або обробка останньої підзадачі.
4. Виконання рекурсивних викликів: Використовуйте рекурсивні виклики для вирішення кожної підзадачі. Кожен виклик має виконуватись зі зменшенням розміру задачі або зі зміною параметрів, щоб наблизитись до базового випадку.
5. Об'єднання результатів: Поєднайте результати виконання підзадач, отримані з рекурсивних викликів, для отримання кінцевого результату.

Важливо правильно проектувати рекурсивну функцію, щоб уникнути зациклення та забезпечити правильну роботу системи. Також варто враховувати, що деякі задачі можуть бути більш ефективно розв'язані іншими методами, тому слід обдумано застосовувати рекурсію в розробці інформаційних систем.

Щоб побудувати рекурсивний алгоритм, ви розіб'єте цю постановку задачі на дві частини. Перший – базовий, а другий – рекурсивний крок.

Базовий випадок: це не що інше, як найпростіший приклад проблеми, що складається з умови, яка завершує рекурсивну функцію. Цей базовий варіант оцінює результат під час виконання заданої умови.

Рекурсивний крок: він обчислює результат, виконуючи рекурсивні виклики тієї ж функції, але зі зменшенням розміру чи складності вхідних даних.

Наприклад, розглянемо таку постановку завдання: вивести суму n натуральних чисел, використовуючи рекурсію. Це твердження пояснює, що нам потрібно сформулювати функцію, яка обчислюватиме суму всіх натуральних чисел у діапазоні від 1 до n . Отже, математично ви можете уявити функцію як:

$$F(n) = 1 + 2 + 3 + 4 + \dots + (n-2) + (n-1) + n$$

Далі його можна спростити так:

$$F(n) = \sum_{k=1}^{k=n} (k)$$

Цю функцію можна розділити на 2 частини наступним чином (рис. 2.1.)

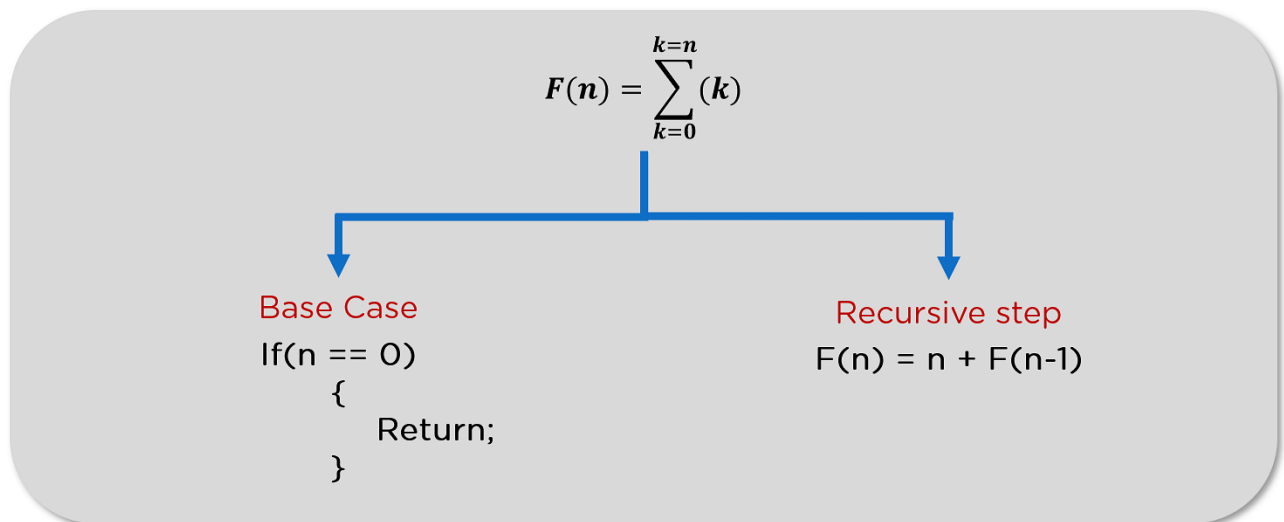


Рисунок 2.1. – Розділення рекурсивної функції

Існує чотири різні типи рекурсивних алгоритмів, ви будете розглядати їх один за одним.

Пряма рекурсія. Функція називається прямою рекурсією, якщо вона часто викликає себе в тілі своєї функції. Щоб краще зрозуміти це визначення, погляньте на структуру прямої рекурсивної програми.

Непряма рекурсія. Рекурсія, коли функція викликає сама себе через іншу функцію, називається непрямою рекурсією. Тепер подивимося на структуру непрямої рекурсії програми.

Хвостова рекурсія. Рекурсивна функція називається хвостовою рекурсією, якщо рекурсивний виклик останнім виконанням функції.

Нехвостова рекурсія. Рекурсивна функція називається нехвостовою рекурсією, якщо виклик рекурсії перестав бути останнім дією, виконуваним функцією. Повернувшись назад, лишилося що оцінити.

Висновок по другому розділу

З написаного вище, можна дійти невтішного висновку у тому, що сьогодні пропонуються різні методології розробки інформаційних систем, але цінність багатьох їх піддається сумніву. І водночас продовжують впроваджуватись нові методології у спробі підтримати та покращити практику роботи з розробки інформаційних систем. У ході огляду запропонованих методів було прийнято рішення використовувати для вирішення поставленого завдання рекурсивний метод розробки.

Згідно з обраним методом, кожен рекурсивний виклик створює нову копію функції пам'яті стека. Як тільки процедура повертає деякі дані, копія видаляється зі сховища. Кожен рекурсивний виклик підтримує окремий стек, тому що всі параметри та інші змінні, визначені всередині функцій, зберігаються у стеку. Стек видаляється після повернення значення відповідної функції.

Рекурсія досить складна з погляду дозволу та моніторингу значень при кожному рекурсивному виклику. В результаті доводиться підтримувати стек і відстежувати значення вказаних у ньому змінних.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Постановка задачі

Інформаційна система управління - це організаційна система, за допомогою якої керівництво збирає і використовує інформацію, отриману з різних джерел (електронних та інших), щоб дати їм змогу ухвалювати правильні рішення в управлінні бізнесом. У 21 столітті багато інформаційних систем управління засновані на комп'ютерах, що вимагає подачі електроенергії для підтримання системи в робочому стані.

Багато інформаційних систем стикаються з деякими проблемами, ці проблеми стають ще більш серйозними для управління, коли в центрі уваги знаходиться проблема інформаційної системи в масштабі підприємства. Також завжди виникає проблема реалізації після концептуалізації організаційної інформаційної системи.

Тому інформаційна система управління організацією пов'язана з такими проблемами:

1. Впровадження інформаційної системи управління.
2. Втрата управлінського контролю над системою.
3. Необхідність ретельного управління організаційними змінами.
4. Необхідність забезпечення надійності мережі та управління.

Тому метою дослідження є вивчення того, якою мірою перелічена проблема вплинула на інформаційну систему управління в бізнес-секторі.

Основна мета дослідження - вивчити роль інформаційної системи управління в діяльності організацій. Конкретні цілі включають:

1. З'ясувати роль інформаційної системи управління в ухваленні рішень.
2. Визначити процеси інформаційної системи управління.
3. Визначити взаємозв'язок між інформаційною системою управління та засобами управління діяльністю.

3.2. Вибір архітектури

Архітектура клієнт-сервер є одною з найпопулярніших концепцій при створенні інформаційних систем.

В цій архітектурі передбачені наступні компоненти:

- серверна частина (збереження і обробка інформації)
- клієнтська частина (робочий інструмент користувача)
- мережа, яка забезпечує взаємодію (обмін інформацією) між клієнтом і сервером

Саме на цій архітектурі і базуються більшість веб-орієнтованих систем. Як типовий приклад архітектури клієнт-сервер розглянемо наступну схему:

Розглянемо як в цій системі буде виглядати облік руху товарної продукції в ресторані від моменту обліку його на складі і до моменту його використання на кухні:

- працівник вводить свої персональні авторизаційні дані в спеціальну форму авторизації, що сформована сервером;
- бізнес-логіка сервера обробляє ці дані звіряючи їх з збереженими в СУБД;
- якщо авторизаційні дані введені коректно, сервер формує інтерфейс користувача і відправляє його в додаток;
- працівник викликає інтерфейс обліку прихідної накладної і вносить товар;
- сервер отримавши дані по приходу товару обробляє їх, автоматично формує ціни реалізації згідно заздалегідь запрограмованих умов, зберігає в СУБД кінцевий результат обробки;
- автоматизований програмний модуль публікує дані про прийнятий на склад товар за допомогою синхронізації на сервер, що обслуговує електронний магазин, і одночасно здійснює прийом інформації про проданий товар;

- для покупців і партнерів, що підписані на отримання оперативної інформації про товар система формує лист і відсилає за допомогою електронної пошти;
- звичайний покупець завантаживши веб-сторінку компанії і побачивши новину про прихід товару в магазині, що також автоматично сформована системою, чи дізнавшись про прихід товару за допомогою електронної пошти завантажує сторінку електронного магазину компанії;
- авторизувавшись в системі покупець переглядає новий товар і здійснює покупку
- сервер електронного магазину обробляє отримані від покупця дані і зберігає їх в СУБД, відсилає необхідні повідомлення через електронну пошту
 - при наступній синхронізації дані про покупку отримуються на внутрішній сервер компанії і представляються до обробки працівникам
 - працівники оформляють на основі отриманих даних накладну (якщо звичайно покупець розраховувався через інтернет при формування замовлення в електронному магазині), дають вказівку відвантажити товар покупцю

Це лише загальний опис функціонування системи на базі “клієнт-сервер”. Такі системи можуть бути надзвичайно різноманітними і розгалуженими.

Переваги веб-орієнтованих систем базованих на клієнт-серверній архітектурі:

- мінімум затрат на обслуговування бізнес-процесів;
- максимальна оперативність при оперуванні даними;
- зручність в обслуговуванні, більшість операцій може виконуватись автоматично;
- один працівник може легко обслуговувати кілька процесів одночасно без особливих зусиль;
- мінімум затрат на комунікації між підрозділами компанії;
- оперативне і гнучке отримання звітів про діяльність компанії;
- веб-сервер, СУБД і програмні модулі що забезпечують функціонування бізнес-логіки як правило розміщуються на одному комп’ютері;

- працівникові для роботи з системою потрібний лише звичайний веббраузер;
- роботу з системою можна здійснювати з будь якого комп'ютера що приєднаний до інтернету.

3.3. Проектування системи

Під час проектування системи була розроблена контекстна діаграма, яка є графічним уявленням системи або процесу, який допомагає встановити його середовище та взаємодія із зовнішніми сутностями. Вона дозволяє виділяти головні гравці (людей, організації чи інші системи), які взаємодіють із системою, і показує основні потоки інформації чи даних між ними.

Контекстна діаграма зазвичай являє собою систему у вигляді центрального блоку, оточеного зовнішніми акторами або системами. Ці актори представляють інтереси, вимоги чи взаємодію з системою. Зв'язки між системою і гравцями позначаються стрілками, що вказують напрямок потоку інформації або даних. Окрім цього, контекстна діаграма є першим кроком в аналізі системи та допомагає встановити кордони системи, ідентифікувати акторів та основні взаємодії між ними. Вона також може служити основою для більш детального моделювання та аналізу системи, наприклад, через створення діаграми взаємодії або діаграми потоків даних.

Контекстні діаграми широко використовуються в системній інженерії, бізнес-аналізі та розробці програмного забезпечення для забезпечення розуміння середовища та вимог систем. Вони допомагають команді проекту обнародувати ключові гравці і встановити початкові зв'язки, необхідні для розробки ефективного рішення.



Рисунок 3.1. - Контекстна діаграма

Щоб зрозуміти принцип взаємодії з системою, розроблено діаграму використання. Діаграма використання (Use Case Diagram) є графічним інструментом моделювання, що демонструє функціональність системи або процесу з точки зору їх використання акторами (користувачами, зовнішніми системами або іншими сутностями).

Основна мета діаграми використання полягає в ідентифікації і описі основних функцій, які система надає своїм акторам. Вона допомагає визначити взаємодію акторів з системою та показати, які задачі виконуються системою відповідно до потреб акторів.

Діаграма використання є потужним засобом в аналізі та проектуванні системи, оскільки вона дозволяє визначити основні потреби та вимоги акторів до системи, встановити обсяг функціональності та визначити основні сценарії взаємодії. Вона також може служити основою для подальшого деталізованого моделювання системи та розробки випадків використання більш високого рівня деталей.



Рисунок 3.2. Діаграма використання

Для розуміння фізичної конфігурації системи було розроблено діаграму розгортання (Deployment Diagram). Ця діаграма є графічним засобом моделювання, який показує фізичну конфігурацію системи та розташування її компонентів на апаратному забезпеченні або віртуальних середовищах. Основна мета діаграми розгортання полягає в візуалізації фізичної архітектури системи та її відображення на різних апаратних або віртуальних ресурсах. Вона допомагає ілюструвати, як компоненти програмного забезпечення розподіляються та взаємодіють з апаратними компонентами, такими як сервери, комп'ютери, мережеві пристрої тощо.

На діаграмі розгортання компоненти системи зображаються у вигляді прямокутників або кола, які можуть представляти програмні модулі, сервери, комп'ютери, пристрої IoT, віртуальні машини або інші фізичні або віртуальні ресурси. Зв'язки між компонентами показуються за допомогою стрілок або ліній, які вказують на залежності та взаємодію між ними. Діаграма розгортання також може включати в себе вказівку комунікаційних каналів, протоколів зв'язку та фізичних з'єднань між компонентами системи.

Цей тип діаграми допомагає команді розробників, адміністраторам мережі та іншим зацікавленим сторонам зрозуміти, як система буде фізично розгортатися і працювати в реальному середовищі. Вона також дозволяє виявити можливі проблеми залежностей або обмежень апаратного забезпечення та забезпечити ефективне розподілення компонентів системи для забезпечення надійності та шкалування.

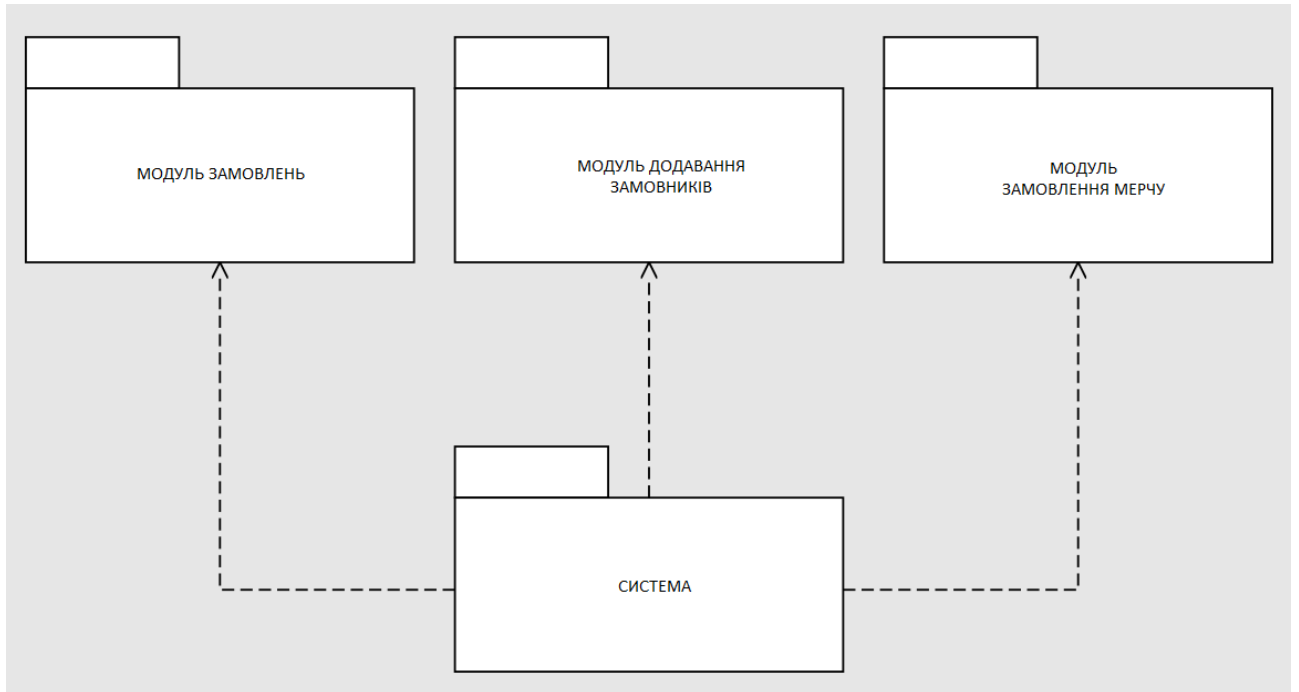


Рисунок 3.3. - Діаграма розгортання

Спроектвана інформаційна система (ІС) допомагає в процесі обробки замовлень, забезпечуючи виконання наступних завдань:

1. Збір інформації. ІС автоматизує збір інформації про замовників і замовлення. Це дозволяє зібрати повну і достовірну інформацію для аналізу.
2. Вироблення альтернатив. ІС може допомагати створювати і оцінювати альтернативні варіанти замовлень.
3. Моніторинг та контроль. ІС надає засоби для моніторингу та контролю реалізації замовлень.

В цілому, інформаційна система управління включає різні процеси, які спрямовані на забезпечення ефективного функціонування системи. Одним з основних процесів є саме збір та обробка даних. Цей процес передбачає збір різних

видів даних про замовників, використовуючи форму вводу. Після збору дані обробляються, перевіряються на достовірність та готуються для подальшого використання.

В цілому ІС надає засоби для збору, обробки та аналізу інформації, необхідної для управління. Вона забезпечує швидкий та точний доступ до даних.

Проект інформаційної системи ресторану, розроблений на основі мови програмування C#, являє собою комплексне рішення для обробки онлайн-заказів. Система призначена для управління замовленнями і касовими операціями в ресторанному бізнесі. В основі лежить ресторан сніданків, який пропонує асортимент з круасанів і кави.

Основні функції та можливості системи:

Управління замовленнями: Система дозволяє створювати і редагувати замовлення ресторану, включаючи страви та їх заповнення, напої.

Замовлення і обробка. Ресторанна система дозволяє приймати замовлення, створювати і керувати даними про замовленнях.

Проект інформаційної системи ресторану на базі C# є потужним інструментом для управління ресторанним бізнесом, підвищення ефективності операцій і забезпечення якісного обслуговування відвідувачів.

Під час проектування програмного рішення була розроблена база даних (далі – БД), яка включає в себе 4 таблиці.

Для створення БД використовувався наступний запит: CREATE database RestDB

Після того, як було створено БД, були створені наступні таблиці:

1. Замовники.
2. Круасани.
3. Напої.

4. Статус замовлення.

Потім ці таблиці було поєднано в одну загальну таблицю – Замовлення.

Приклади запитів для створення таблиць:

1. Таблиця «Замовники»

```
CREATE TABLE Customer  
(  
CustID int PRIMARY KEY NOT NULL,  
Cust_Name nvarchar(50) NOT NULL,  
Cust_Surname nvarchar(50) NOT NULL,  
Mobile nvarchar(20) NOT NULL  
)
```

Після виконання запиту отримуємо наступну таблицю:

	Имя	Тип данных	Допустимы значения NULL	По умолчанию	
№	CustID	int	<input type="checkbox"/>		
	Cust_Name	nvarchar(50)	<input type="checkbox"/>		
	Cust_Surname	nvarchar(50)	<input type="checkbox"/>		
	Mobile	nvarchar(20)	<input type="checkbox"/>		
			<input type="checkbox"/>		

Рисунок 4.1. – Таблица «Замовники»

2. Таблиця «Круасани»

```
CREATE TABLE Croissants  
(  
EatID int PRIMARY KEY NOT NULL,  
Taste nvarchar(50) NOT NULL  
)
```

	Имя	Тип данных	Допустимы значения NULL	По умолчанию	
№	EatID	int	<input type="checkbox"/>		
	Taste	nvarchar(50)	<input type="checkbox"/>		
			<input type="checkbox"/>		

Рисунок 4.2. – Таблица круасани

3. Таблиця «Напої»

```
CREATE TABLE Drinks  
(  
DrinkID int PRIMARY KEY NOT NULL,
```

TypeOfDrink nvarchar(50) NOT NULL

)

Имя	Тип данных	Допустимы значения NULL	По умолчанию
DrinkID	int	<input type="checkbox"/>	
TypeOfDrink	nvarchar(50)	<input type="checkbox"/>	
		<input type="checkbox"/>	

Рисунок 4.3. – Таблица «Напої»

4. Таблица «Статус замовлення»

```
CREATE TABLE OrderStatus
```

```
(
```

```
OrderStat nvarchar(50) NOT NULL
```

```
)
```

Имя	Тип данных	Допустимы значения NULL	По умолчанию
OrderStat	nvarchar(50)	<input type="checkbox"/>	
		<input type="checkbox"/>	

Рисунок 4.4. – Таблица «Статус замовлення»

Далі необхідно заповнити кожену таблицю, виконуючі наступні запити:

```
INSERT INTO Customer VALUES  
(111, N'Олександр', N'Петренко',N'+380505005050')  
INSERT INTO Customer VALUES  
(112, N'Лілія', N'Процюк',N'+380665666666')  
INSERT INTO Customer VALUES  
(113, N'Владислав', N'Глушко',N'+380635632313')  
INSERT INTO Customer VALUES  
(114, N'Лілія', N'Процюк',N'+3809967544')  
INSERT INTO Customer VALUES  
(115, N'Даріна', N'Білоус',N'+380937765454')  
INSERT INTO Customer VALUES  
(116, N'Олександра', N'Осіпова',N'+38066142654')  
INSERT INTO Customer VALUES  
(117, N'Владислава', N'Сергієнко',N'+380672313275')  
INSERT INTO Customer VALUES  
(118, N'Ліза', N'Вовк',N'+380612321213')  
INSERT INTO Customer VALUES  
(119, N'Григорій', N'Ткач',N'+380657776655')  
INSERT INTO Customer VALUES  
(110, N'Анна', N'Борщ',N'+380579956575')
```

	CustID	Cust_Name	Cust_Surname	Mobile
▷	110	Анна	Борщ	+380579956575
	111	Олександр	Петренко	+380505005050
	112	Лілія	Процюк	+380665666666
	113	Владислав	Глушко	+380635632313
	114	Лілія	Процюк	+3809967544
	115	Даріна	Білоус	+380937765454
	116	Олександра	Осіпова	+38066142654
	117	Владислава	Сергієнко	+380672313275
	118	Ліза	Вовк	+380612321213
	119	Григорій	Ткач	+380657776655
◊	NULL	NULL	NULL	NULL

Рисунок 4.5. – Заповнення таблиця «Замовники»

INSERT INTO Croissants VALUES

- (1, N'Цезар'),
- (2, N'Банан та шоколад'),
- (3, N'Кремчиз з лососем'),
- (4, N'Мигдальний крем'),
- (5, N'Італія'),
- (6, N'Сирний бум'),
- (7, N'Кебаб')

	EatID	Taste
▷	1	Цезар
	2	Банан та шоко...
	3	Кремчиз з лос...
	4	Мигдальний к...
	5	Італія
	6	Сирний бум
	7	Кебаб
◊	NULL	NULL

Рисунок 4.6. – Заповнення таблиця «Круасани»

INSERT INTO Drinks VALUES

- (1,N'Капучіно'),
- (2,N'Латте'),
- (3,N'Какао'),
- (4,N'Раф'),
- (5,N'Флет уайт'),
- (6,N'Чай вітамінний'),

(7,N'Чай класичний')

	DrinkID	TypeOfDrink
▶	1	Капучіно
	2	Латте
	3	Какао
	4	Раф
	5	Флет уайт
	6	Чай вітамінний
	7	Чай класичний
⊕	NULL	NULL

Рисунок 4.7. – Заповнення таблиця «Напої»

INSERT INTO OrderStatus VALUES

(N'В обробці'),

(N'Готується'),

(N'Готово')

	OrderStat
▶	В обробці
	Готується
	Готово
⊕	NULL

Рисунок 4.7. – Заповнення таблиця «Статус замовлення»

Далі на основі цих таблиць створюємо таблицю «Замовлення»

CREATE TABLE Orders

```
(  
CustID int PRIMARY KEY NOT NULL,  
Cust_Name nvarchar(50) NOT NULL,  
Cust_Surname nvarchar(50) NOT NULL,  
Mobile nvarchar(20) NOT NULL,  
Taste nvarchar(50) NOT NULL,  
TypeOfDrink nvarchar(50) NOT NULL,  
OrderStatus nvarchar(50) NOT NULL  
)
```

І заповнюємо її:

INSERT INTO Orders VALUES

```
(111, N'Олександр', N'Петренко',N'+380505005050', N'Цезар',N'Какао', N'Готується'),  
(112, N'Лілія', N'Процюк',N'+380665666666', N'Кебаб',N'Чай вітамінний',N'В обробці'),  
(113, N'Владислав', N'Глушко',N'+380635632313',N'Італія', N'Капучіно',N'Готово'),  
(114, N'Лілія', N'Процюк',N'+3809967544',N'Мигдальний крем',N'Раф', N'Готово'),
```

(115, N'Даріна', N'Білоус',N'+380937765454', N'Банан та шоколад', N'Какао', N'Готується'),

(116, N'Олександра', N'Осіпова',N'+38066142654',N'Цезар',N'Чай вітамінний',N'Готується'),

(117, N'Владислава', N'Сергієнко',N'+380672313275',N'Кремчиз з лососем',N'Флет уайт', N'В обробці'),

(118, N'Ліза', N'Вовк',N'+380612321213',N'Кебаб', N'Латте', N'В обробці'),

(119, N'Григорій', N'Ткач',N'+380657776655', N'Сирний бум', N'Чай класичний',N'Готово'),

(110, N'Анна', N'Борщ',N'+380579956575',N'Мигдальний крем', N'Какао', N'Готується')

	CustID	Cust_Name	Cust_Surname	Mobile	Taste	TypeOfDrink	OrderStatus
▶	110	Анна	Борщ	+380579956575	??????????	Какао	Готується
	111	Олександр	Петренко	+380505005050	Цезар	Какао	Готується
	112	Лілія	Процюк	+380665666666	Кебаб	Чай вітамінний	В обробці
	113	Владислав	Глушко	+380635632313	Італія	Капучіно	Готово
	114	Лілія	Процюк	+3809967544	Мигдальний к...	Раф	Готово
	115	Даріна	Білоус	+380937765454	Банан та шоко...	Какао	Готується
	116	Олександра	Осіпова	+38066142654	Цезар	Чай вітамінний	Готується
	117	Владислава	Сергієнко	+380672313275	Кремчиз з лос...	Флет уайт	В обробці
	118	Ліза	Вовк	+380612321213	Кебаб	Латте	В обробці
	119	Григорій	Ткач	+380657776655	Сирний бум	Чай класичний	Готово
◇	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 4.9. – Таблиця «Замовлення»

Саме з цією таблицею будемо працювати надалі. Переходимо до огляду готової програми. На рисунку 4.10 зображено головну форму програми, через яку вводяться дані про замовлення.

The screenshot shows a web application window with a teal header. In the top left corner is the logo for 'КАВА ТА КРУАСАНИ' (Coffee and Pastries) featuring a coffee cup, a fork, and a knife. The main form area is white and contains several input fields and dropdown menus. The fields are labeled: 'ID', 'Ім'я' (Name), 'Прізвище' (Surname), 'Номер телефону' (Phone number), 'Круасан' (Pastry), 'Напій' (Drink), and 'Статус замовлення' (Order status). Below the form are three buttons: 'Додати' (Add), 'Змінити' (Change), and 'Переглянути всі замовлення' (View all orders). At the bottom of the window, there are three more buttons: 'Показати готові замовлення' (Show ready orders), 'Показати замовлення в очікуванні' (Show orders in progress), and 'Показати замовлення, що готуються' (Show orders being prepared).

Рисунок 4.10. - Головна форма

Для введення даних використовуються текстові поля та combobox, данні для комбобокс беруться їх таблиць в БД – Круасани, Напої та статус замовлення. Потім ці данні з форми записуються в загальну таблицю замовлень. Якщо в даних є помилка,

то нова інформація не додається, як показано на рисунку 4.11.

The screenshot shows a web application window titled 'Замовлення' (Order). The application has a teal header with a logo for 'КАВА ТА КРУАСАНИ' (COFFEE AND PASTRIES) on the left. The main form area is white and contains several input fields and dropdown menus. The fields are: ID (235), Ім'я (Олексій), Прізвище (Панчішко), Номер телефону (+38066534645), Круасан (Цезар), Напій (Какао), and Статус замовлення (Готово). Below the form are three buttons: 'Додати', 'Змінити', and 'Переглянути всі замовлення'. A modal dialog box is open in the center, with a close button (X) in the top right corner. The dialog contains the text 'Додати не вийшло!' and an 'ОК' button.

Рисунок 4.11. - Формування замовлення

В цьому випадку була проблема в ID, оскільки до цього вже було додано користувача з таким же айді. Тому, перш ніж додати нового замовника, необхідно переглянути вже існуючі та перевірити їхні айді. (рис. 4.12)

CustID	Cust_Name	Cust_Surname	Mobile	Taste	TypeOfDrink	OrderStatus
110	Анна	Борщ	+380579956575	?????????? ???? ?	Какао	Готується
111	Олександр	Петренко	+380505005050	Цезар	Какао	Готується
112	Лілія	Процюк	+380665666666	Кебаб	Чай вітамінний	В обробці
113	Владислав	Глушко	+380635632313	Італія	Капучіно	Готово
114	Лілія	Процюк	+3809967544	Мигдальний крем	Раф	Готово
115	Даріна	Білоус	+380937765454	Банан та шоколад	Какао	Готується
116	Олександра	Осіпова	+38066142654	Цезар	Чай вітамінний	Готується
117	Владислава	Сергієнко	+380672313275	Кремчиз з лосо...	Флет уайт	В обробці
118	Ліза	Вовк	+380612321213	Кебаб	Латте	В обробці
119	Григорій	Ткач	+380657776655	Сирний бум	Чай класичний	Готово

Рисунок 4.12 – Всі замовлення


Після цього змінюємо айді замовника і додаємо. Новий замовник додається в кінець списку, оскільки в нього порядковий номер збільшився на 1.

CustID	Cust_Name	Cust_Surname	Mobile	Taste	TypeOfDrink	OrderStatus
110	Анна	Борщ	+380579956575	?????????? ???? ?	Какао	Готується
111	Олександр	Петренко	+380505005050	Цезар	Какао	Готується
112	Лілія	Процюк	+380665666666	Кебаб	Чай вітамінний	В обробці
113	Владислав	Глушко	+380635632313	Італія	Капучіно	Готово
114	Лілія	Процюк	+3809967544	Мигдальний крем	Раф	Готово
115	Даріна	Білоус	+380937765454	Банан та шоколад	Какао	Готується
116	Олександра	Осіпова	+38066142654	Цезар	Чай вітамінний	Готується
117	Владислава	Сергієнко	+380672313275	Кремчиз з лосо...	Флет уайт	В обробці
118	Ліза	Вовк	+380612321213	Кебаб	Латте	В обробці
119	Григорій	Ткач	+380657776655	Сирний бум	Чай класичний	Готово
120	Олексій	Панчішко	+38066534645	Цезар	Какао	Готово

Рисунок 4.13 – Додавання нового замовлення

Також через форму можна подивитись замовлення з конкретним статусом.

Замовлення



ID: Ім'я: Прізвище:

Номер телефону: Круасан: Напій: Статус замовлення:

Додати Змінити Переглянути всі замовлення


CustID	Cust_Name	Cust_Surname	Mobile	Taste	TypeOfDrink	OrderStatus
113	Владислав	Глушко	+380635632313	Італія	Капучіно	Готово
114	Лілія	Процюк	+3809967544	Мигдальний крем	Раф	Готово
119	Григорій	Ткач	+380657776655	Сирний бум	Чай класичний	Готово

Показати готові замовлення Показати замовлення в очікуванні Показати замовлення, що готуються

4.14. – Готові замовлення

Замовлення

Возврат:
 Объект, представляющий выделенный элемент, или значение null, если выделенный элемент



ID: Ім'я: Прізвище:

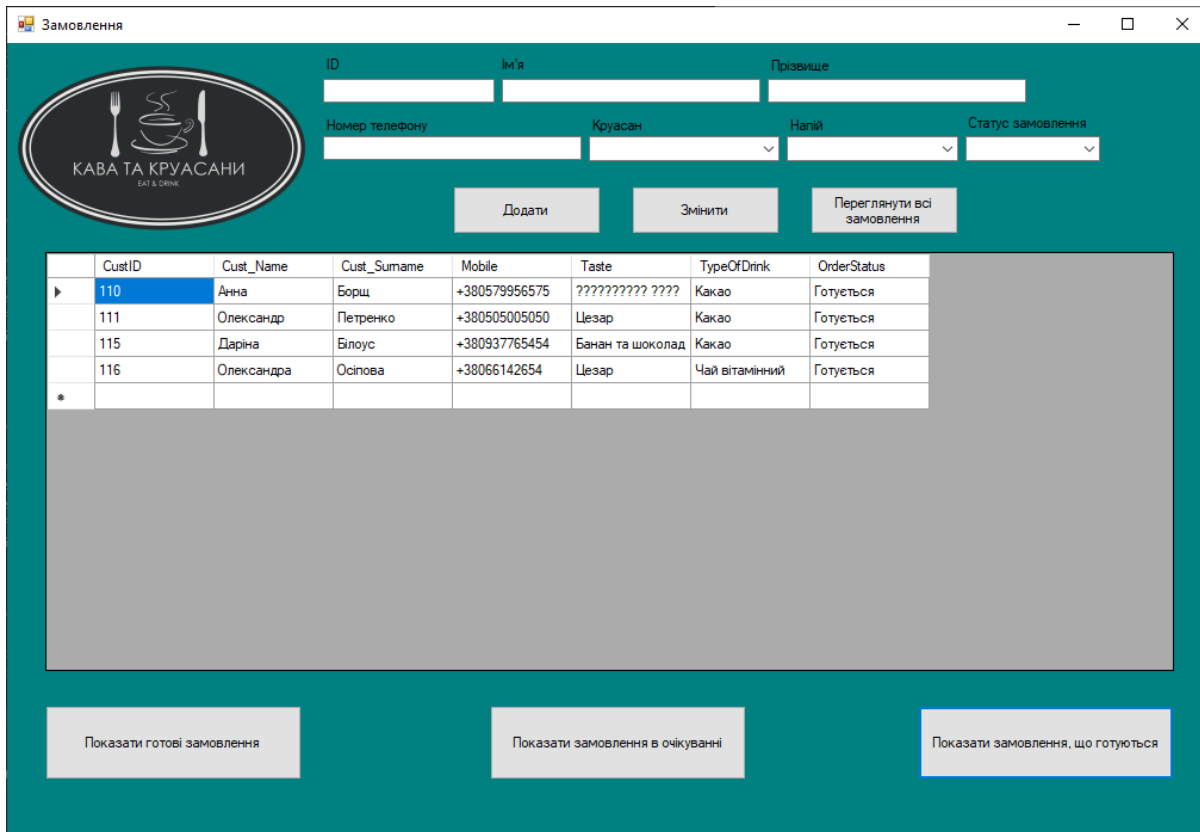
Номер телефону: Круасан: Напій: Статус замовлення:

Додати Змінити Переглянути всі замовлення

CustID	Cust_Name	Cust_Surname	Mobile	Taste	TypeOfDrink	OrderStatus
112	Лілія	Процюк	+380665666666	Кебаб	Чай вітамінний	В обробці
117	Владислава	Сергієнко	+380672313275	Кремчиз з лосо...	Флет уайт	В обробці
118	Ліза	Вовк	+380612321213	Кебаб	Латте	В обробці

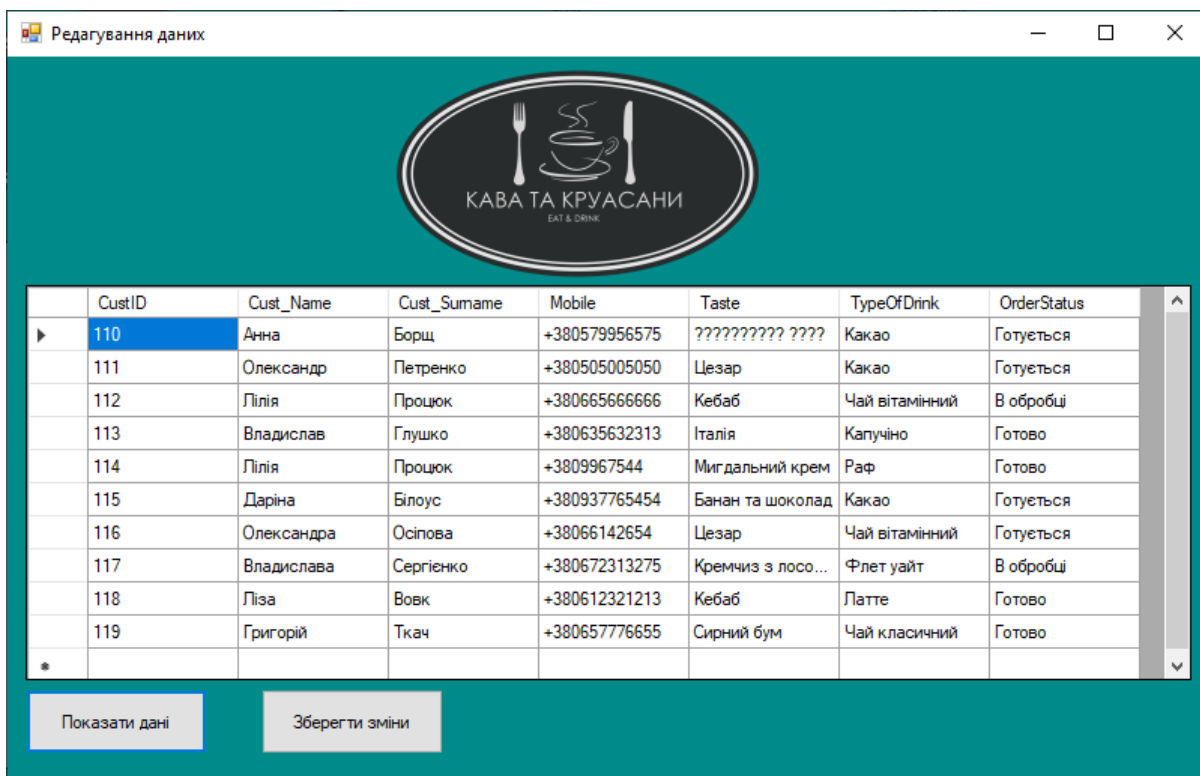
Показати готові замовлення Показати замовлення в очікуванні Показати замовлення, що готуються

4.15 – Замовлення в очікуванні




4.15. – Замовлення, що готуються

Через форму також можна редагувати замовлення, а саме змінювати їхній статус або додавати нові продукти:



4.16– Перегляд таблиці

Редагування даних



	CustID	Cust_Name	Cust_Surname	Mobile	Taste	TypeOfDrink	OrderStatus
▶	110	Анна	Борщ	+380579956575	?????????? ???? ?	Какао	Готується
	111	Олександр	Петренко	+380505005050	Цезар	Какао	Готується
	112	Лілія	Процюк	+380665666666	Кебаб	Чай вітамінний	В обробці
	113	Владислав	Глушко	+380635632313	Італія	Капучіно	Готово
	114	Лілія	Процюк	+3809967544	Мигдальний крем	Раф	Готово
	115	Даріна	Білоус	+380937765454	Банан та шоколад	Какао	В обробці
	116	Олександра	Осіпова	+38066142654	Цезар	Чай вітамінний	Готується
	117	Владислава	Сергієнко	+380672313275	Кремчиз з лосо...	Флет уайт	В обробці
	118	Ліза	Вовк	+380612321213	Кебаб	Латте	Готово
	119	Григорій	Ткач	+380657776655	Сирний бум	Чай класичний	Готово
*							

Показати дані Зберегти зміни

ВИСНОВОК

У ході роботи було розроблено інформаційну систему, призначену для автоматизації та оптимізації роботи ресторану. Система була розроблена мовою програмування C# з використанням різних технологій та інструментів, таких як .NET Framework, ADO.NET, SQL Server та інші.

Основною метою розробки системи було забезпечити ефективне управління бізнес-процесами ресторану, покращити обслуговування онлайн-клієнтів, підвищити ефективність роботи персоналу та забезпечити надійне зберігання та обробку даних про онлайн-замовлення.

У межах роботи було реалізовано ефективне управління замовленнями: система дозволяє приймати замовлення від клієнтів та автоматично передавати їх на кухню чи бар. Замовлення можуть бути легко відстежуваними.

Розроблена інформаційна система ресторану на C# дозволяє автоматизувати та спростити безліч бізнес-процесів, зв'язаних з управлінням рестораном. Вона володіє зручним інтерфейсом користувача, надійною архітектурою і добре структурованим кодом, що забезпечує легкість використання, масштабованість і підтримку системи в довгостроковій перспективі.

Подальший розвиток системи може включати додаткові функціональні можливості, такі як інтеграція з касовим апаратом, мобільний додаток для клієнтів або впровадження системи управління складом.

У цілому нині, розробка інформаційної системи ресторану на C# є значний внесок у автоматизацію та оптимізацію роботи ресторанного бізнесу, і може бути основою подальшого вдосконалення та розвитку системи з урахуванням специфічних потреб ресторану та змін у промисловості.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кісь Я. Інтелектуальна система моніторингу та обслуговування клієнтів у ресторанному бізнесі. [Електронний ресурс]: Електронний каталог 2.0 науковотехнічної бібліотеки ТНТУ імені Івана Пулюя. — Режим доступу: <http://koha.tntu.edu.ua/bib/71082>.
2. Архітектура клієнт-сервер [Електронний ресурс]: Intelligent System Design. — Режим доступу: <http://www.intelsd.com/?tc=175&sc=197&lvl=2>.
3. Transact-SQL [Електронний ресурс]: Microsoft SQL Server. — Режим доступу: [http://technet.microsoft.com/ru-ru/library/ms189826\(v=sql.90\).aspx](http://technet.microsoft.com/ru-ru/library/ms189826(v=sql.90).aspx)
4. C# and .NET Programming [Електронний ресурс]: Microsoft Developer Network.
5. — Режим доступу: <http://msdn.microsoft.com/en-us/library/orm-9780596521066-0101.aspx>
6. .NET Framework and .NET SDKs [Електронний ресурс]: .NET Framework
7. Developer Center. — Режим доступу: <http://msdn.microsoft.com/en-US/vstudio/aa496123>
8. ADO.NET [Електронний ресурс]: Microsoft Developer Network. — Режим доступу: [http://msdn.microsoft.com/library/e80y5yhx\(v=vs.110\).aspx](http://msdn.microsoft.com/library/e80y5yhx(v=vs.110).aspx)
9. https://www.academia.edu/22293120/DESIGN_AND_IMPLEMENTATION_OF_ONLINE_FOOD_ORDERING_SYSTEM

ДОДАТОК

SQL QUERY

```
CREATE database RestDB
```

```
CREATE TABLE Customer
```

```
(  
CustID int PRIMARY KEY NOT NULL,  
Cust_Name nvarchar(50) NOT NULL,  
Cust_Surname nvarchar(50) NOT NULL,  
Mobile nvarchar(20) NOT NULL
```

```
)
```

```
INSERT INTO Customer VALUES
```

```
(111, N'Олександр', N'Петренко',N'+380505005050')
```

```
INSERT INTO Customer VALUES
```

```
(112, N'Лілія', N'Процюк',N'+380665666666')
```

```
INSERT INTO Customer VALUES
```

```
(113, N'Владислав', N'Глушко',N'+380635632313')
```

```
INSERT INTO Customer VALUES
```

```
(114, N'Лілія', N'Процюк',N'+3809967544')
```

```
INSERT INTO Customer VALUES
```

```
(115, N'Даріна', N'Білоус',N'+380937765454')
```

```
INSERT INTO Customer VALUES
```

```
(116, N'Олександра', N'Осіпова',N'+38066142654')
```

```
INSERT INTO Customer VALUES
```

```
(117, N'Владислава', N'Сергієнко',N'+380672313275')
```

```
INSERT INTO Customer VALUES
```

```
(118, N'Ліза', N'Вовк',N'+380612321213')
```

```
INSERT INTO Customer VALUES
```

```
(119, N'Григорій', N'Ткач',N'+380657776655')
```

```
INSERT INTO Customer VALUES
```

```
(110, N'Анна', N'Борщ',N'+380579956575')
```

```
CREATE TABLE Croissants
```

```
(
```

```
EatID int PRIMARY KEY NOT NULL,
```

```
Taste nvarchar(50) NOT NULL
```

```
)
```

```
INSERT INTO Croissants VALUES
```

```
(1, N'Цезар'),
```

```
(2, N'Банан та шоколад'),
```

```
(3, N'Кремчиз з лососем'),
```

```
(4, N'Мигдальний крем'),
```

```
(5, N'Італія'),
```

```
(6, N'Сирний бум'),
```

(7, N'Кебаб')

```
CREATE TABLE Drinks
(
DrinkID int PRIMARY KEY NOT NULL,
TypeOfDrink nvarchar(50) NOT NULL
)
```

```
INSERT INTO Drinks VALUES
```

```
(1,N'Капучіно'),
(2,N'Латте'),
(3,N'Какао'),
(4,N'Раф'),
(5,N'Флет уайт'),
(6,N'Чай вітамінний'),
(7,N'Чай класичний')
```

```
CREATE TABLE OrderStatus
(
OrderStat nvarchar(50) NOT NULL
)
```

```
INSERT INTO OrderStatus VALUES
```

```
(N'В обробці'),
(N'Готується'),
(N'Готово')
```

```
CREATE TABLE Orders
(
CustID int PRIMARY KEY NOT NULL,
Cust_Name nvarchar(50) NOT NULL,
Cust_Surname nvarchar(50) NOT NULL,
Mobile nvarchar(20) NOT NULL,
Taste nvarchar(50) NOT NULL,
TypeOfDrink nvarchar(50) NOT NULL,
OrderStatus nvarchar(50) NOT NULL
)
```

```
INSERT INTO Orders VALUES
```

```
(111, N'Олександр', N'Петренко',N'+380505005050', N'Цезар',N'Какао', N'Готується'),
(112, N'Лілія', N'Процюк',N'+380665666666', N'Кебаб',N'Чай вітамінний',N'В обробці'),
(113, N'Владислав', N'Глушко',N'+380635632313',N'Італія', N'Капучіно',N'Готово' ),
(114, N'Лілія', N'Процюк',N'+3809967544',N'Мигдальний крем',N'Раф', N'Готово'),
```


(115, N'Даріна', N'Білоус',N'+380937765454', N'Банан та шоколад', N'Какао', N'Готується'),
 (116, N'Олександра', N'Осіпова',N'+38066142654',N'Цезар',N'Чай вітамінний',N'Готується'),
 (117, N'Владислава', N'Сергієнко',N'+380672313275',N'Кремчиз з лососем',N'Флет уайт', N'В обробці'),
 (118, N'Ліза', N'Вовк',N'+380612321213',N'Кебаб', N'Латте', N'В обробці'),
 (119, N'Григорій', N'Ткач',N'+380657776655', N'Сирний бум', N'Чай класичний',N'Готово'),
 (110, N'Анна', N'Борщ',N'+380579956575',Mигдальний крем', N'Какао', N'Готується')

```
SELECT * FROM Orders WHERE OrderStatus=N'Готово'
```

Замовлення.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace restaurant
{
    public partial class Замовлення : Form
    {
        private SqlConnection sqlConnection = null;
        private SqlDataAdapter sda = null;
        private DataTable table = null;
        DataSet ds;
        SqlDataAdapter adapter;
        SqlCommandBuilder commandBuilder;
        string connectionString = @"Data Source=.\SQLEXPRESS;Initial
Catalog=usersdb;Integrated Security=True";
        string sql = "SELECT * FROM Orders";
        public Замовлення()
        {
            InitializeComponent();
        }
    }
}
```

```

private void label5_Click(object sender, EventArgs e)
{

}

private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    comboBox2.Items.AddRange(new string[] { "Капучіно", "Латте", "Какао", "Раф",
"Флет уайт", "Чай вітамінний", "Чай класичний" });
}

private void Замовлення_Load(object sender, EventArgs e)
{
    SqlConnection = new SqlConnection(@"Data
Source=(localdb)\MSSQLLocalDB;Initial Catalog=RestDB;Integrated Security=True");
    SqlConnection.Open();
}

private void id_txt_TextChanged(object sender, EventArgs e)
{

}

private void nametxt_TextChanged(object sender, EventArgs e)
{

}

private void surnametxt_TextChanged(object sender, EventArgs e)
{

}

private void phonetxt_TextChanged(object sender, EventArgs e)
{

}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    sda = new SqlDataAdapter("SELECT Taste FROM Croissants", SqlConnection);
    table = new DataTable();
    table.Clear();
}

```

```

sda.Fill(table);
dataGridView1.DataSource = table;

// добавляем набор элементов
comboBox1.Items.AddRange(new string[] { "Цезар", "Банан та шоколад",
"Кремчиз з лососем", "Мигдальний крем", "Італія", "Сирний бум", "Кебаб" });
}

private void orderstatus_SelectedIndexChanged(object sender, EventArgs e)
{
    orderstatus.Items.AddRange(new string[] { "В обробці", "Готується", "Готово" });
}

private void addbtn_Click(object sender, EventArgs e)
{
    using (SqlConnection con = new SqlConnection(@"Data
Source=(localdb)\MSSQLLocalDB;Initial Catalog=RestDB;Integrated Security=True"))
    {
        string commText = "INSERT INTO Order (ID, Ім'я, Призвіще, Номер телефону,
Круасан, Напій, Статус замовлення) VALUES (@CustID, @Cust_Name,
@Cust_Surname, @Mobile, @Taste, @TypeOfDrink, @OrderStatus)";
        SqlCommand comm = new SqlCommand(commText, con);
        comm.Parameters.AddWithValue("@CustID", id_txt.Text += "");
        comm.Parameters.AddWithValue("@Cust_Name", nametxt.Text += "");
        comm.Parameters.AddWithValue("@Cust_Surname", surnametxt.Text += "");
        comm.Parameters.AddWithValue("@Mobile", phonetxt.Text += "");
        comm.Parameters.AddWithValue("@Taste", comboBox1.SelectedItem);
        comm.Parameters.AddWithValue("@TypeOfDrink", comboBox2.SelectedItem);
        comm.Parameters.AddWithValue("@OrderStatus", orderstatus.SelectedItem);
        con.Open();
        try
        {
            comm.ExecuteNonQuery();
        }
        catch
        {
            MessageBox.Show("Додати не вийшло!");
        }
    }
}

private void cngbtn_Click(object sender, EventArgs e)

```

```

{
    Form1 f1 = new Form1();
    f1.Show();
}

private void viewbtn_Click(object sender, EventArgs e)
{
    sda = new SqlDataAdapter("SELECT * FROM Orders", sqlConnection);
    table = new DataTable();
    table.Clear();
    sda.Fill(table);
    dataGridView1.DataSource = table;
}

private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
    dataGridView1.AllowUserToAddRows = false;

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        adapter = new SqlDataAdapter(sql, connection);

        ds = new DataSet();
        adapter.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];
        // делаем недоступным столбец id для изменения
        dataGridView1.Columns["Id"].ReadOnly = true;
    }
}

private void button1_Click(object sender, EventArgs e)
{
    sda = new SqlDataAdapter("SELECT * FROM Orders WHERE
OrderStatus=N'Готово' ", sqlConnection);
    table = new DataTable();
    table.Clear();
    sda.Fill(table);
    dataGridView1.DataSource = table;
}

private void button2_Click(object sender, EventArgs e)

```

```

    {
        sda = new SqlDataAdapter("SELECT * FROM Orders WHERE OrderStatus=N'В
обробці' ", sqlConnection);
        table = new DataTable();
        table.Clear();
        sda.Fill(table);
        dataGridView1.DataSource = table;
    }

private void button3_Click(object sender, EventArgs e)
{
    sda = new SqlDataAdapter("SELECT * FROM Orders WHERE
OrderStatus=N'Готується' ", sqlConnection);
    table = new DataTable();
    table.Clear();
    sda.Fill(table);
    dataGridView1.DataSource = table;
}
}
}

```

```

Form1.cs
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace restaurant
{
    public partial class Form1 : Form
    {
        private SqlConnection sqlConnection = null;
        private SqlDataAdapter sda = null;
        private DataTable table = null;
        private SqlCommandBuilder scb = null;
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            sqlConnection = new SqlConnection(@"Data
Source=(localdb)\MSSQLLocalDB;Initial Catalog=RestDB;Integrated Security=True");
            sda = new SqlDataAdapter("SELECT * FROM Orders", sqlConnection);
            table = new DataTable();
            table.Clear();
            sda.Fill(table);
            dataGridView1.DataSource = table;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            scb = new SqlCommandBuilder(sda);
            sda.Update(table);
        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

```

}
}
}