Державний вищий навчальний заклад

«Чернівецький політехнічний коледж»

Циклова комісія інженерії програмного забезпечення

**Пояснювальна записка**

до комплексного курсового проекту

на тему: Електронне меню

Студента IV курсу 541 групи

спеціальності 5.05010301

«Розробка програмного забезпечення»

Гудан Ю. В.

(прізвище та ініціали)

Керівник Серьогіна І.В.

Оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Г. Смерека

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.М. Гуменна

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ І.В. Серьогіна

(підпис)

м. Чернівці – 2017 рік

**Зміст**

[Вступ 3](#_Toc497153783)

[розділ 1. Обґрунтування необхідності розробки 4](#_Toc497153784)

[1.1 Аналіз предметної області 4](#_Toc497153785)

[1.2 Аналіз сучасного стану та перспективи розвитку програмного забезпечення 5](#_Toc497153786)

[1.3 Аналіз існуючого програмного забезпечення 5](#_Toc497153787)

[1.4 Постановка задачі 5](#_Toc497153788)

[Розділ 2. Технічний проект 6](#_Toc497153789)

[2.1 Інформаційна модель предметної області 6](#_Toc497153790)

[2.2 Концептуальна модель предметної області 6](#_Toc497153791)

[2.3 Об’єктна модель 6](#_Toc497153792)

[2.4 Моделювання даних 6](#_Toc497153793)

[2.5 Представлення реалізації 6](#_Toc497153794)

[Розділ 3. Робочий проект 7](#_Toc497153795)

[3.1 Засоби розробки 7](#_Toc497153796)

[3.2 Інтерфейс програмного забезпечення 7](#_Toc497153797)

[3.3 Розробка документів на супроводження програмного забезпечення 7](#_Toc497153798)

[3.3.1 Інструкція програмісту 7](#_Toc497153799)

[3.4.2 Інструкція користувачеві 7](#_Toc497153800)

[Розділ 4. Тестування програмного забезпечення 8](#_Toc497153801)

[4.1 Тестування адміністраторської частини 8](#_Toc497153802)

[4.2 Тестування клієнтської частини 8](#_Toc497153803)

[4.3 Тестування користувацького інтерфейсу 8](#_Toc497153804)

[4.4 Тестування роботи бази даних 8](#_Toc497153805)

[Висновки 9](#_Toc497153806)

[Перелік джерел 10](#_Toc497153807)

[додатки 11](#_Toc497153808)

# Вступ

Актуальність теми розробки, в даний час дуже популярні заклади громадського харчування, і в таких закладах часто виникає проблеми з обслуговуванням. Дане тема було обрана для зручного, швидкого та ефективного обслуговування відвідувачів даних закладів.

Мета курсового проекту проведення передпроектного дослідження для розробки програмного продукту для автоматизації роботи.

Завдання курсового проекту є визначення функціональної структури, методології розробки програмного забезпечення для автоматизації заданої предметної області. (<визначити основні риси майбутньої розробки, оцінити існуючі методи та алгоритми реалізації її розробки>, «складено структурний алгоритм роботи майбутнього програми … та реалізовано його у вигляді програмного забезпечення …, що складається з … модулів, у тому числі стандартних …, з використанням об’єктно-орієнтованого програмування»)

Об’єктом дослідження є ресторанне господарство .

Предметом дослідження є методології та засоби реалізації функцій адміністратора або клієнта по обліку замовлення, перегляду інформації про страви, розрахунку продукції.

Практичне значення отриманих результатів (можливо, місце застосування програмного забезпечення)

Використане програмне забезпечення, буде розроблено за допомогою середовище RAD Studio Delphi XE7 в зв’язку з його відповідним функціоналом для створення даного програмного забезпечення. («При виконанні завдання курсового проекту було використано таке програмне забезпечення; стандартне середовище розробки …, операційна система …, web-браузер … для роботи з web-сайтами …, текстовий редактор … для підготовки та оформлення курсового проекту»)

У першому розділі пояснювальної записки представлені передпроектне дослідження та проведення моделювання предметної області;….

# розділ 1.Обґрунтування необхідності розробки

## 1.1 Аналіз предметної області

Заклад громадського харчування - підприємство, призначене для виробництва кулінарної продукції, борошняних, кондитерських і булочних виробів, їх реалізації та організації їх споживання.

Типи закладів громадського харчування

* ресторани;
* бари;
* кафе, кафетерії;
* їдальні, в тому числі їдальні на підприємствах, у навчальних закладах тощо;
* закусочні;
* буфети, магазини кулінарних виробів;
* фабрики-кухні, фабрики-заготівельні.

Однією із основних вимог сучасної індустрії ресторанного бізнесу є швидке і якісне обслуговування споживачів. Такі можливості реалізовані в системі інтерактивного електронного меню. Цей новітній засіб комунікації є потужним інструментом, що дозволяє надавати клієнтам високий рівень обслуговування, а також більш ефективно керувати закладом. Електронне меню повністю замінює традиційне меню і пропонує цілий ряд додаткових функцій. Воно є не тільки каталогом продукції, яку пропонує заклад, але і прискорює процес обслуговування, виключає людський фактор (помилки) під час приймання замовлення. Меню представлене у вигляді візуально дуже схожому на паперовий прототип, що сприяє легкому сприйняттю інтерфейсу.

Для відвідувачів переваги використання електронного меню є наступними:

- зручність замовлення, швидка і легка навігація по меню, пошук страв;

- меню є максимально інформативним: подаються описання і фото страв, інгредієнти, способи і строки приготування;

- по стравах автоматично пропонуються доповнення і добавки;

- наявність можливості швидкого виклику офіціанта і запиту рахунку;

- зручна форма зворотного зв’язку про якість обслуговування.

На практиці експлуатації такого програмного забезпечення приведе до вирішення таких завдань як: фізичне перерозподіл обчислень і даних, забезпечення паралелізму обчислень, реплікація бази даних, забезпечення безпеки доступу до програмного забезпечення, оптимізація балансування навантаження інформаційної системи, стійкість до збоїв і т.п. З цим програмним забезпеченням робота в сфері ресторанного господарства прискориться в багато раз.

## 1.2 Аналіз сучасного стану та перспективи розвитку програмного забезпечення

Ресторанне господарство постійно розвивається та удосконалюється. Якість та рівень обслуговування закладів є об’єктом постійних досліджень та аналізу. Стратегічна мета розвитку даної індустрії в Україні полягає у створенні конкурентоспроможного, високоякісного, здатного максимально задовольнити потреби споживачів ресторанного господарство, забезпечити на цій основі комплексний розвиток галузі.

Сучасні електронні пристрої сьогодні з'являються в усіх сферах життя, не стала винятком і сфера ресторанного господарства.

Електронне меню - це ресторанне меню 21-го століття. Меню, яке дозволить відмовитися від звичних паперових меню і дасть гостям готелю відчути себе комфортно у виборі страв і при оформленні замовлення.

Статистика по закладах, що ввели дану інновацію показує, що результатом покращення якості обслуговування клієнтів, якого ми досягнемо з електронним меню, стане:

* Оборот закладів збільшується, аж до 40%;
* Збільшення пропускної здатності закладу в прайм-тайм на 20-25 %;
* Зменшення помилок , пов'язаних з прийомом замовлень і випискою рахунків;
* Зменшення витрат на обслуговуючий персонал;
* Збільшення середньої суми чека на 15 % (за рахунок стимулювання імпульсивних замовлень);
* Зменшення часу очікування на прийом і обробку замовлень на 21 %;
* Скорочення часу очікування на прийом повторного замовлення на 15 %;
* Істотне збільшення кількості клієнтів (за рахунок підвищення популярності та впізнаваності);
* Виняток витрат, пов'язаних з людським фактором;
* Можливість обслуговування іноземних клієнтів (багатомовна підтримка);
* Відсутність тимчасових і фінансових витрат на внесення змін в меню ;
* Швидка окупність системи електронного меню (3-6 місяців)

Було проведено порівняння доходу закладу при використанні старого та нового зразку меню за роки їх використання (рис. 1.).



Рисунок 1 – Рівень доходу закладу при використанні електронного меню та меню паперового

За результатами дослідження було виявлено, що надання послуг в ресторанах ще не досягло бажаного рівня. В Україні ресторани мають підвищувати свою конкурентоспроможність і виходити на світовий ринок підготовленими. Тому потрібно сконцентрувати свою увагу на розвитку чогось нового, що покращить рівень обслуговування, та вводити в дію нові технології.

Електронне меню - це потужний інструмент, що дозволяє ефективно управляти закладом, і надати клієнтам найбільш високий рівень обслуговування.

## 1.3 Аналіз існуючого програмного забезпечення

Існують багато програмних забезпечень для закладів громадського харчування, які спрощують обслуговування, швидкість замовлення тощо.

Один із аналогів «Quick Resto» - це системи для залучення гостей, інструментів обліку та аналітики для збільшення прибутку, а також універсальна POS-система для обслуговування відвідувачів. За допомогою неї можна контролювати роботу і спостерігати за ростом бізнесу.

Меню програмного забезпечення зображено на рисунку 2.



Рисунок 2 – ПЗ «Quick Resto»

Дане програмне забезпечення має можливості: ведення обліку страв, стеження за прибутком, повний контроль над програмним забезпеченням, формування звітності за різними вимогами, налаштування повідомлень для постійного стеження за роботою в закладі, прибутком тощо.

## 1.4 Постановка задачі

Завданням комплексного курсового проекту є створити автоматизовану інформаційну систему «електронне меню» для обліку страв, інгредієнтів, напоїв, замовлень, цінових характеристик, розрахунку вартості замовлень, виведення меню в електронному вигляді, замовлення та страви і т.д. Звернути увагу, що кількість інгредієнта вимірюється в різних одиницях виміру (штуки, грами і т.п.).

Дана система повинна мати можливості:

* додавання інформації;
* редагування інформації;
* видалення інформації;
* пошук інформації про страву або напій;
* візуально в інтерактивному режимі ознайомитися з меню закладу та наочно й оперативно побачити високоякісні фотографії і детальний опис кожної страви;
* викликати офіціанта, наприклад: для того, щоб попросити рахунок;
* відправити повідомлення на кухню для приготування страв, які вказані в замовленні;
* розрахунок інгредієнтів для страв;
* розрахунок використаної та втраченої продукції;
* виведення страви в меню при наявності інгредієнтів.

Програмний продукт повинен мати простий та зручний, інтуїтивно-зрозумілий інтерфейс користувача.

Вхідні дані:

* інформація про страви: фотографія страви, назва страви, категорія, собівартість, ціна, націнка, відомості;
* інформація про напої: фотографія напою, назва напою, категорія, собівартість, ціна, націнка, відомості;
* інформація про склад страви: назва інгредієнту, собівартість, одиниця виміру, наявність, похибка;
* інформація про замовлення: дата замовлення, прізвище та ініціали офіціанта, номер стола, оплата замовлення.

Вихідні дані: інформація про збір страви, рахунок відвідувача (чек), звітність за різними критеріями.

Функціональна специфікація:

* ввід та вивід інформації в залежності від доступу до системи;
* пошук для перегляду інформації про страви, напої;
* формування вихідних даних, тобто звіти з необхідної інформації або чеки з інформацією про замовлення;
* розрахунок використаних продуктів;
* розрахунок вигоди та втрати фінансів;
* реплікація бази даних.

Розроблена специфікація вимог описує необхідні вхідні дані, які мають бути введені, та вихідні дані для виводу відповідної інформації.

# Розділ 2. Технічний проект

## 2.1 Інформаційна модель предметної області

Алгоритм роботи програмного забезпечення зображено на рисунку .

Вхідна інформація

Інтерфейс програмного забезпечення

Сервер бази даних

Вихідна інформація

Системи управління бази даних

Дані бази даних

Запит на отримання даних

Передача

нових даних

Отримання даних

Передача параметрів

Передача даних

Рисунок – Схема функціонування програмного забезпечення

Життєвий цикл автоматизованої інформаційної системи (ЖЦ) показує безперервний процес із моменту ухвалення рішення про необхідність створення до повного завершення її експлуатації. Життєвий цикл програмного забезпечення зображений на рисунку 12.

Визначення вимог

Розробка технічного завдання

Планування розробки поставлених задач

Проектування , вивчення завдання (аналіз предметної області)

Реалізація поставлених задач

Об’єднання системи

Тестування програмного забезпечення

Супровід програмного забезпечення

Супровід ПЗ

Рисунок 12 – Життєвий цикл програмного забезпечення

## 2.2 Концептуальна модель предметної області

Концептуальна модель показує структуру майбутньої автоматизованої інформаційної системи, взаємодію основних сутностей. Концептуальна модель зображена на рисунку 1.

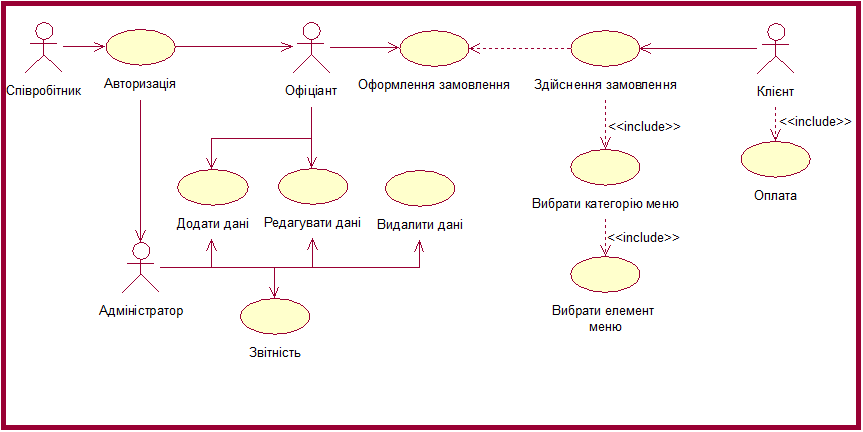


Рисунок 1 – Концептуальна модель

Опис концептуальної моделі:

1. Актор «Офіціант» – даний актор при взаємодії з програмним забезпеченням має доступ до: додавання інформації та її редагування. Також даний актор взаємодіє з актором «Клієнт» під час оформлення замовлення.
2. Актор «Адміністратор» – даний актор при взаємодії з програмним забезпеченням має повний доступ до оброблення інформації та створення звітності.
3. Актор «Клієнт» – взаємодії з актором «Офіціант» під час здійснення замовлення та оплати замовлення.

Розроблена концептуальна модель описує структуру взаємодії між співробітником, клієнтом та програмним забезпеченням.

## 2.3 Об’єктна модель

Базовий клас Zakaz містить атрибути: dates (дата замовлення), nomer\_stola (номер стола), price\_zakaz (ціна заказу), Mas\_Strav(об’єкт класу Strava), Mas\_napoi (об’єкт класу Napoi). Методи базового класу :

* сеттери (для введення даних): setDates(),setNomer\_stola(),setPrice();
* геттери (для отримання даних): getDates(),getNomer\_stola(),getPrice();
* методи для обробки даних в базі даних: Add(), Delete(), Update().

Дочірній клас Oficiant містить атрибут: PIB(Прізвище та ініціали). Також даний клас містить методи:

* сеттери (для введення даних): setPIB();
* геттери (для отримання даних): getPIB().

Клас Napoi містить такі атрибути: nazva(назва напою), foto (фотографія напою), cost\_price(собівартість), price(ціна напою), natsinka(націнка), data(додаткові відомості). Методи класу :

* сеттери (для введення даних): setNazva(), setFoto(), setCost\_price(), setPrice(), setNatsinka(), setData();
* геттери (для отримання даних): getNazva(),getFoto(),getCost\_price(), getPrice(), getNatsinka(), getData();
* методи для обробки даних в базі даних: Add(), Delete(), Update().

Клас Strava містить такі атрибути: nazva(назва страви), foto (фотографія страви), cost\_price(собівартість), price(ціна страви), natsinka(націнка), data(додаткові відомості). Методи класу :

* сеттери (для введення даних): setNazva(), setFoto(), setCost\_price(), setPrice(), setNatsinka(), setData();
* геттери (для отримання даних): getNazva(),getFoto(),getCost\_price(), getPrice(), getNatsinka(), getData();
* методи для обробки даних в базі даних: Add(), Delete(), Update().

Дочірній клас Kategorii класів Napoi та Strava, містить атрибут: nazva (назва категорії). Методи класу:

* сеттери (для введення даних): setNazva(), setFoto(), setCost\_price(), setPrice(), setNatsinka(), setData();
* геттери (для отримання даних): getNazva(),getFoto(),getCost\_price(), getPrice(), getNatsinka(), getData();
* методи для обробки даних в базі даних: Add(), Delete(), Update().

Дочірній клас Sklad\_stravu класу Strava містить атрибути: kilkisti (кількість інгредієнтів), cost\_price (собівартість). Методи класу:

* сеттери (для введення даних): setKilkisti(), setCost\_price();
* геттери (для отримання даних): getKilkisti(), getCost\_price().

Дочірній клас Ingredientu класу Sklad\_stravu містить атрибути: nazva (назва інгредієнту), cost\_price (собівартість), vumir (одиниця виміру), nayavnisti (наявність інгредієнту), pohubka (похибка). Методи класу:

* сеттери (для введення даних): setNazva(), setCost\_price(), setVumir(), setNayavnitsti(), SetPohubka();
* геттери (для отримання даних): getNazva(), getCost\_price(), getVumir(), getNayavnisti(), getPohubka();

На рисунку 16 зображено діаграма класів.

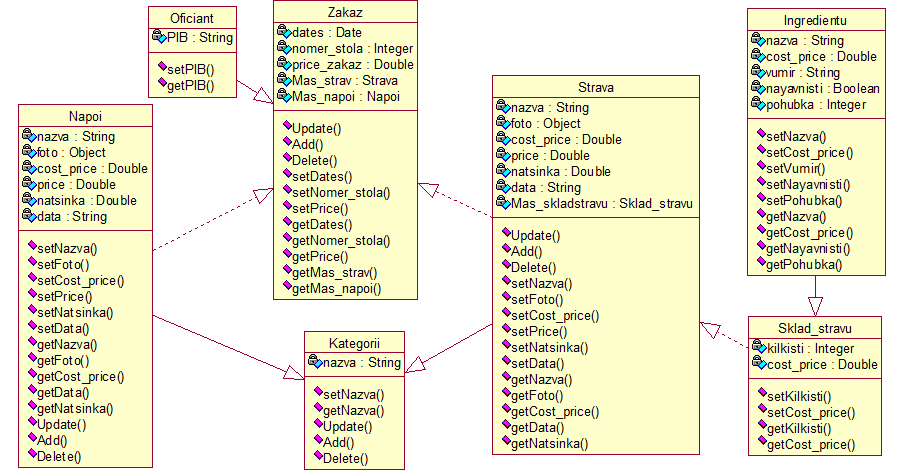


Рисунок 16 – Діаграма класів

На даній діаграмі класі описується методи та атрибути класів, відображається ієрархія між класами, залежність між класами. В класах використовується інкапсуляція для захисту даних, також в класах використовується поліморфізм.

## 2.4Моделювання даних

Концептуальна моделі даних

2.4.1 Фізична модель даних

Для опису предметної області розроблені сім таблиці, в яких зберігається необхідна інформація для майбутньої системи.

Таблиця «Strava» призначена для зберігання інформації про страви. Структура таблиці «Strava» зображена на рисунку 2.

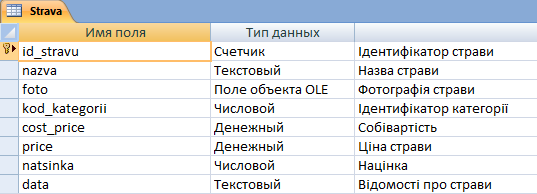


Рисунок 2 – Структура таблиць «Strava»

Таблиця «Sklad\_stravy» призначена для зберігання інформації яка зв’язує страву з інгредієнтами. Структура таблиці «Sklad\_stravy» зображена на рисунку 3.

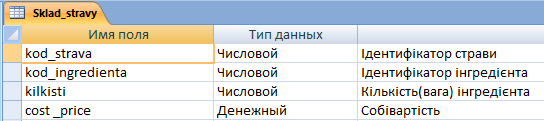


Рисунок 3 – Структура таблиць «Sklad\_stravy»

Таблиця «Ingredienty» призначена для зберігання інформації про інгредієнти. Структура таблиці «Ingredienty» зображена на рисунку 4.

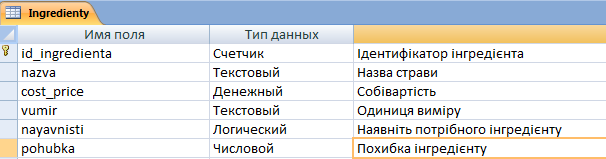


Рисунок 4 – Структура таблиць «Ingredienty»

Таблиця «Napoi» призначена для зберігання інформації про напої. Структура таблиці «Napoi» зображена на рисунку 5.

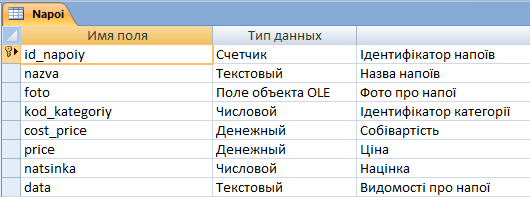


Рисунок 5 – Структура таблиць «Napoi»

Таблиця «Kategorii» призначена для зберігання інформації про категорії. Структура таблиці «Kategorii» зображена на рисунку 6.

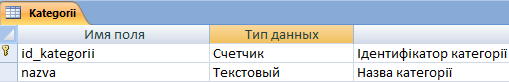


Рисунок 6 – Структура таблиць «Kategorii»

Таблиця «Oficiant» призначена для зберігання інформації про співробітника. Структура таблиці «Oficiant» зображена на рисунку 7.

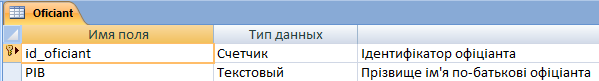


Рисунок 7 – Структура таблиць «Oficiant»

Таблиця «Strava» призначена для зберігання інформації про страви. Структура таблиці «Strava» зображена на рисунку 8.

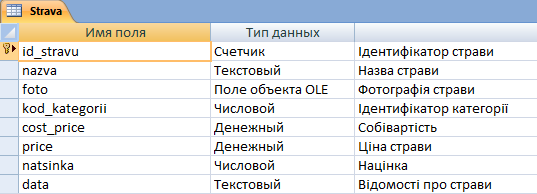


Рисунок 8 – Структура таблиць «Strava»

Таблиця «Zakaz\_menu» призначена для зберігання інформації про кількість страв або напоїв, які є в замовлені. Структура таблиці «Zakaz\_menu» зображена на рисунку 9.

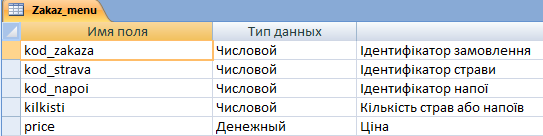


Рисунок 9 – Структура таблиць «Zakaz\_menu»

Таблиця «Zakaz» призначена для зберігання інформації про замовлення. Структура таблиці «Zakaz» зображена на рисунку 10.

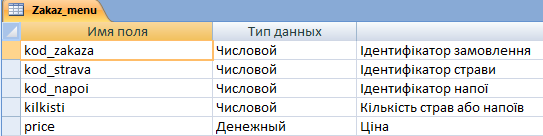


Рисунок 10 – Структура таблиць «Zakaz»

2.4.2 Логічна модель даних

На рисунку 11 зображено логічну схему даних, в який відображені зв`язких між таблицям.

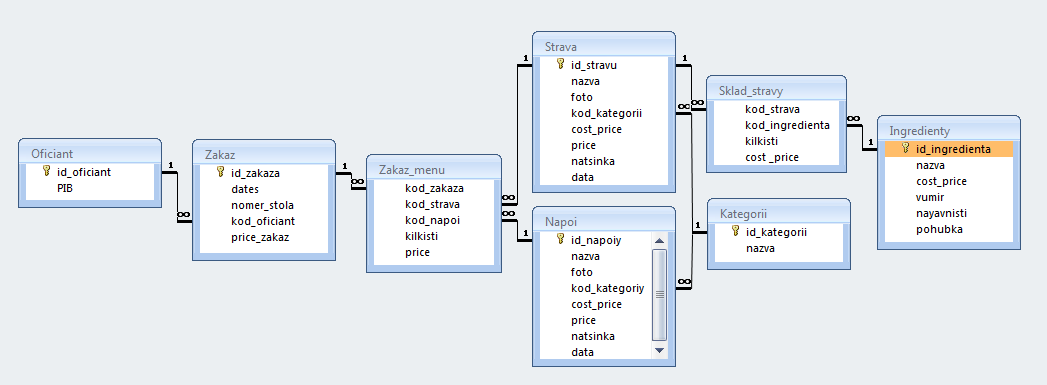


Рисунок 11 – Логічна модель даних

Розроблена фізична та логічна модель описує інформацію про базу даних.

## 2.5 Представлення реалізації

На рисунку 12 зображено діаграму послідовності

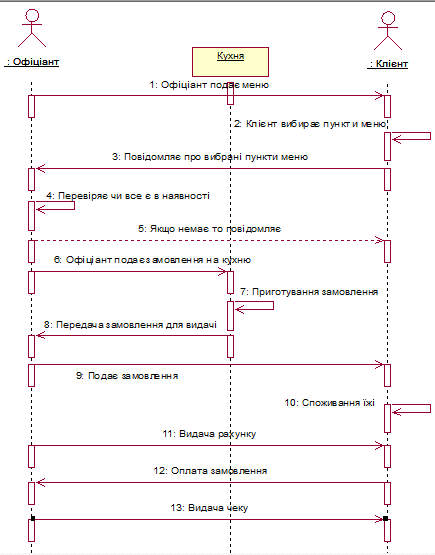


Рисунок 12 – Діаграма послідовності

На даній діаграмі описується, як актор «Офіціант» взаємодіє з актором «Клієнт» під час замовлення, описано покрокове виконання дії.

На рисунку 14-16 зображено діаграму взаємодій

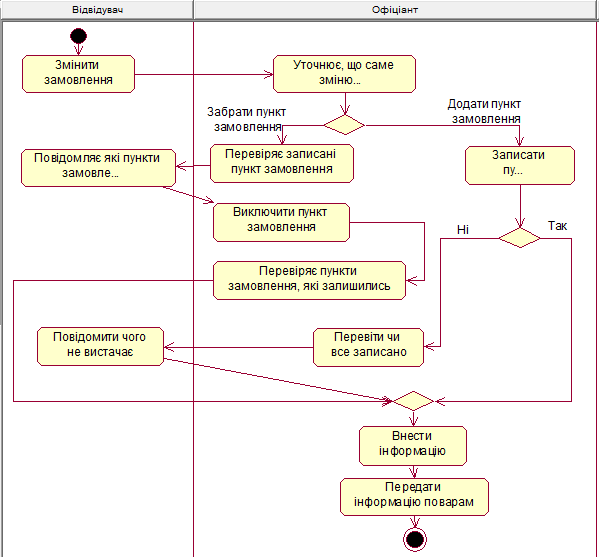


Рисунок 14 – Діаграма взаємодій

На даній діаграмі описується взаємодії між двома акторами «Офіціант» та «Відвідувач» у випадку, коли «Відвідувач» хоче змінити своє замовлення.



Рисунок 15 – Діаграма взаємодії

На даній діаграмі описується дії під час вибори страви з меню, та оплати замовлення.

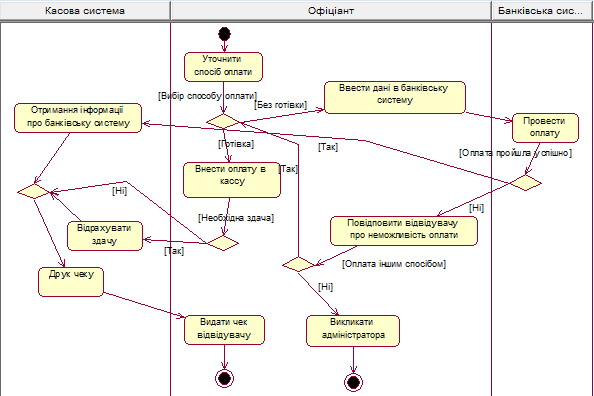


Рисунок 16 – Діаграма взаємодії

На даній діаграмі описується дії, які буде виконувати актор «Офіціант», коли відвідувач заплатить за замовлення. Описано кожну дію у випадку, коли відвідувач оплачуватиме замовлення готівкою або через банківську систему.

На рисунку 15 зображено діаграму стану.

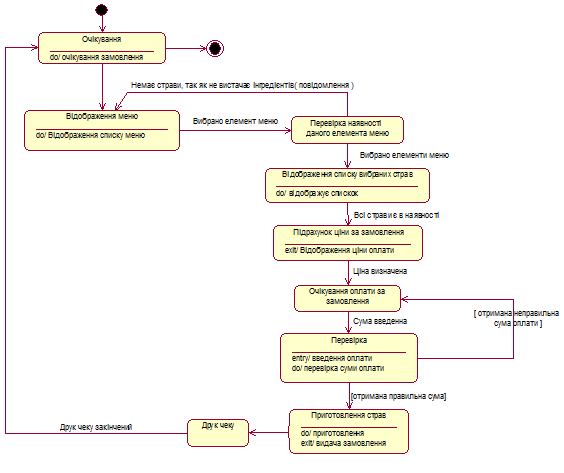


Рисунок 15 – Діаграма стану

На даній діаграмі зображено стан програмного забезпечення, коли користувач взаємодіє з ПЗ під час оформлення замовлення.

В даному розділі було представлено реалізацію діаграм послідовності, стану, класів, взаємодії, описано кожну дію. Також були описані фізичні та логічні моделі даних.

***Всі моделі повинні бути описані!!!!!***

# Розділ 3. Робочий проект

## 3.1 Засоби розробки

Обґрунтування вибору програмних засобів для реалізації програмного забезпечення (інтерфейсу, роботи з базою даних тощо)

Мова програмування RAD Studio Delphi XE7 – мова програмування, який використовується в однойменній середовищі розробки і є комбінацією декількох найважливіших технологій:

* високопродуктивний компілятор в машинний код;
* об’єктно-орієнтована модель компонентів;
* візуальна (а, отже, і швидкісний) побудова додатків з програмних прототипів;
* масштабовані засоби для побудови баз даних.

Спочатку мова називався Object Pascal. Починаючи з середовища розробки Delphi 7.0, в офіційних документах Borland стала використовувати назву Delphi для позначення мови Object Pascal.

Основу Delphi становить не тільки саму мову, але і RAD (Rapid Application

Development) – середовище швидкої розробки програм. Завдяки візуальному програмуванню, а також досить велика бібліотека візуальних RAD Studio Delphi XE7 дозволяє створювати програми найбільш швидко і ефективно, приймаючи на себе основну роботу, і залишаючи програмісту творчий процес. Зрозуміло, можливість швидкого створення професійних додатків для Windows робить RAD Studio Delphi XE7– програмістів затребуваними у всіх галузях людської діяльності.

## 3.2 Інтерфейс програмного забезпечення

Містить опис усіх реалізованих форм (екранів), меню, підменю тощо

## 3.3 Розробка документів на супроводження програмного забезпечення

### 3.3.1 Інструкція програмісту

Опис основної програми і користувацьких програмних модулів у порядку їх оголошення. Містить опис класів та їх методів, функцій обробки даних тощо (може бути побудований у вигляді тексту або таблиць, що містять відповідні описи, згруповані за належністю до певного хідер файлу. Обов’язково потрібно надати опис тих даних, розуміння призначення яких може вплинути на розуміння функціонування важливої частини або програми загалом)

### 3.4.2 Інструкція користувачеві

Містить опис усіх можливостей використання програмного забезпечення

# Розділ 4. Тестування програмного забезпечення

Містить інформацію про результати тестування програмного забезпечення та порівняльну характеристику з існуючими аналогами, виявлені недоліки та рекомендації щодо використання (виплавлення). Може містити ілюстрації певних ігрових ситуацій, станів інформаційно-пошукової системи або інших результатів роботи програмного забезпечення.

## 4.1 Тестування адміністраторської частини

## 4.2Тестування клієнтської частини

## 4.3 Тестування користувацького інтерфейсу

## 4.4 Тестування роботи бази даних

# Висновки

Містить висновки по розробленому програмному забезпеченні: що і як вдалося реалізувати, переваги та недоліки розробки, можливі шляхи подальшого покращення (обсяг 1-2 сторінки)

# Перелікджерел

1. Страуструп Б. Язык программирования С++, спец.изд. /Пер. с англ – М.: «Издательство БИНОМ», 2002. – 1099с.
2. Буч Г. Объектно-ориентированное проектирование с примерами применения: Пер. с англ. – М.: Конкорд, 2010. – 519с.,ил.
3. Збірка бібліотек для робот из RAD STUDIO XE5 С++BUILDER <http://docwiki.embarcadero.com/Libraries/XE5/en/Main_Page>
4. Короткий довідник sql запитів <http://databases.about.com/od/development/>
5. Форум з зразками sql та с++ функцій http://www.cyberforum.ru
6. Практичне використання реактивного програмування в програмуванні систем <http://habrahabr.ru/post/140719/>.

# додатки

**Додаток А**

**Тексти програмного коду**

А.1 Файл …