Державний вищий навчальний заклад

«Чернівецький політехнічний коледж»

Відділення інформаційних технологій

Циклова комісія інженерії програмного забезпечення

**Пояснювальна записка**

до дипломного проекту

молодшого спеціаліста

на тему: Електронне меню для закладів громадського харчування

Виконав: студент IV курсу, групи 541

спеціальність 5.05010301

«Розробка програмного забезпечення»

Гудан Ю.В.

Керівник Серьогіна І.В.

Рецензент «невідомо»

Чернівці – 2018 року**зміст**

[ВСТУП 5](#_Toc516528477)

[РОЗДІЛ 1. ПЕРЕДПРОЕКТНЕ ДОСЛІДЖЕННЯ 6](#_Toc516528478)

[1.1 Аналіз предметної області 6](#_Toc516528479)

[1.2 Аналіз існуючого програмного забезпечення 7](#_Toc516528480)

[1.3 Аналіз сучасного стану та перспективи розвитку 8](#_Toc516528481)

[1.4 Постановка задачі 10](#_Toc516528482)

[1.5 Маркетингове дослідження 11](#_Toc516528483)

[РОЗДІЛ 2. ТЕХНІЧНИЙ ПРОЕКТ 12](#_Toc516528484)

[2.1 Технічне завдання 12](#_Toc516528485)

[2.2 Моделювання програмного забезпечення 12](#_Toc516528486)

[2.3 Моделювання даних 13](#_Toc516528487)

[2.4 Проектування інтерфейсу 13](#_Toc516528488)

[РОЗДІЛ 3. робочий проект 14](#_Toc516528489)

[3.1 Засоби розробки 14](#_Toc516528490)

[3.2 Керівництво програміста 15](#_Toc516528491)

[3.3 Керівництво користувача 15](#_Toc516528492)

[3.4 Тестування програмного забезпечення 15](#_Toc516528493)

[Розділ 4. ОХОРОНА ПРАЦІ 17](#_Toc516528494)

[ВИСНОВКИ 18](#_Toc516528495)

[перелік джерел 19](#_Toc516528496)

[Додатки 20](#_Toc516528497)

# ВСТУП

Вступ виконують з нової пронумерованої сторінки із заголовком Вступ посередині рядка. Текст вступу розкриває сутність і стан задачі та її значущість, підстави і вихідні дані для розробки теми.

У вступі висвітлюється: актуальність теми, практична значущість, аналіз вивченості проблеми вітчизняними та зарубіжними дослідниками, об’єкт і предмет дослідження, мета і завдання дослідження та загальну структуру пояснювальної записки. Приклади оформлення вступу наведено в додатку Г.

У вступі і далі по тексту не дозволяється використовувати скорочені слова, терміни, крім загальноприйнятих.

*Актуальність теми*: шляхом критичного аналізу та порівняння з відомими розв’язаннями проблеми обґрунтовується актуальність та доцільність роботи для розвитку відповідної сфери інформаційних технологій.

*Предмет дослідження*: знаходиться в межах об’єкта і містить проблему, що не дозволяє на виході процесу отримати результат, який задовольнив би конкретного споживача.

*Об’єкт дослідження*: процес або явище, що проходять в процесах інформатизації і мають на виході конкретного споживача.

Об’єкт і предмет дослідження співвідносяться між собою як загальне і часткове. В об’єкті виділяється та його частина, яка є предметом дослідження. Саме на нього спрямована основна увага слухача, оскільки предмет дослідження визначає тему ДП.

*Мета і завдання дослідження*: формулюють мету проекту і завдання, які необхідно вирішити для досягнення поставленої мети. Не слід формулювати мету як “Дослідження...”, “Вивчення...”, тому що ці слова вказують на засіб досягнення мети, а не на саму мету.

Мета дослідження повинна розглядатись як модель майбутнього результату ДП і в значній мірі може співпадати з темою ДП.

*Практичне значення* отриманих результатів: в пояснювальній записці треба подати відомості про практичне застосування протягом виконання проекту одержаних результатів або рекомендації щодо їх використання.

# РОЗДІЛ 1. ПЕРЕДПРОЕКТНЕ ДОСЛІДЖЕННЯ

## 1.1 Аналіз предметної області

Ресторанне господарство – це вид економічної діяльності суб'єктів господарської діяльності з надання послуг для задоволення потреб споживачів у харчуванні з організацією дозвілля або без нього. Суб’єкти господарювання провадять діяльність у ресторанному господарстві через підприємства (заклади) ресторанного господарства.

Заклад ресторанного господарства – підприємство, призначене для виробництва кулінарної продукції, борошняних, кондитерських і булочних виробів, їх реалізації та організації їх споживання.

Однією із основних вимог сучасної індустрії ресторанного господарства є швидке і якісне обслуговування споживачів. Заклади ресторанного господарства, що займаються торгівлею конкурентоспроможною продукцією, необхідної широкому споживачеві, стискаються з проблемою збільшення кількості клієнтів. А так, як при ручному веденні документації потрібного багато часу, трудовитрат і матеріалів, необхідна автоматизація процесів заповнення, поновлення, зберігання і обробки документації – 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. Якісне обслуговування в ресторані – це суть послуг ресторану й умова його успішності. Асортимент, смакові якості страв, ціни, дизайн, імідж, місце розташування важливі, але вони ніколи не компенсують в очах клієнта неуважного і недбалого ставленням з боку персоналу. З іншого боку, ресторан, що зумів надати клієнтам відмінний сервіс, одержує найсильнішу конкурентну перевагу.

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, зменшує «людський фактор» (помилки) під час приймання замовлення. Меню представлене у вигляді візуально дуже схожому на паперовий прототип, що сприяє легкому сприйняттю інтерфейсу.

На практиці експлуатації такого програмного забезпечення приведе до вирішення таких завдань як: фізичний 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 існуючого програмного забезпечення

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

Перший аналог «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.

Інтерфейс програмного забезпечення зображено на рисунку 1.



Рисунок 1 – Інтерфейс 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.

Другий аналог «Ultra Ресторан» – це програма для ресторанів, яка впорядкує облік і дозволить поліпшити систему обслуговування клієнтів.

Інтерфейс програмного забезпечення зображено на рисунку 2.



Рисунок 2 – Інтерфейс ПЗ «Ultra Ресторан»

Дане програмне забезпечення має можливості: захист від зловживання з боку персоналу, доступ до інформації в залежності від рівня доступу, внесення нововведених пунктів за декілька натискань, автоматизація тарифікації часу перебування клієнта, структурування бази клієнтів і активне застосування дисконтних програм, постійний доступ до максимально повної інформації щодо тарних і фінансових запасів.

## 1.3 Аналіз сучасного стану та перспективи розвитку

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%;
* збільшення кількості клієнтів (за рахунок підвищення популярності та пізнаваності);
* 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 (див. рис. 3).



Рисунок 3 – 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.

## 1.4 Постановка задачі

Завданням дипломного проекту є створити програмне забезпечення «Електронне меню» для закладів громадського харчування під операційну систему Windows 7 і вище, що забезпечить обліку інформації про страви, інгредієнти, напої, замовлення, цінові характеристики, розрахунок вартості замовлень, виведення меню в електронному вигляді. Звернути увагу, що кількість інгредієнта вимірюється в різних одиницях виміру (штуки, кілограми і літри).

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

Вхідні дані:

* інформація про страви та напої: фотографія страви, назва страви, категорія, собівартість, ціна, націнка, відомості;
* інформація про склад страви: назва інгредієнту, собівартість, одиниця виміру;
* інформація про замовлення: дата замовлення, прізвище та ініціали офіціанта, номер стола, оплата замовлення, дисконтна картка, дата відкриття та закриття замовлення, активність замовлення.

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

Функціональна специфікація:

* доступ до інформації в залежності від доступу до системи: адміністратора, персонал та відвідувач;
* пошук для перегляду інформації за різними критеріями;
* формування вихідних даних, тобто рахунки з інформацією про замовлення, звіти для перегляду: рейтинг страв та напоїв, рейтинг працівників, денного заробітку, витрат інгредієнтів, рейтинг клієнтів;
* візуально в інтерактивному режимі ознайомитися з меню закладу та наочно й оперативно побачити високоякісні фотографії і детальний опис кожної страви;
* розрахунок ціни та ваги за допомогою інгредієнтів для страви;
* реплікація бази даних.

Розроблена постановка задачі описує необхідні вхідні дані, які мають бути введені, та вихідні дані для виводу відповідної інформації, та функціональні можливості системи.

Специфікація користувача

## 1.5 Маркетингове дослідження

# РОЗДІЛ 2. ТЕХНІЧНИЙ ПРОЕКТ

## 2.1 Технічне завдання

Найменування програмного забезпечення: «eMenu» – електронне меню для закладів громадського харчування. Підставою до виконання програмного забезпечення є наказ по коледжу № від 12 березня 2018 року та індивідуальне завдання видане цикловою комісією програмування та інформаційних технологій.

Терміни розробки:

* початок – 25 квітня 2018 року;
* закінчення – 17 червня 2018 року.

Застосування та вимоги до програмного забезпечення для закладів громадського харчування досить різноманітні, тому існують різні версії, які можуть не містити стандартні функцій, проте мати певні специфічні додаткові функції, необхідні конкретній категорії користувачів.

Призначення розробки: розробити програмне забезпечення для обліку замовлень в закладах громадського харчування під управління ОС Windows.

Програмне забезпечення, що розробляється повинен вирішувати наступні задачі: принципові, функціональні, сервісні.

До принципових задач відносять:

* забезпечення збереження та обробка інформації меню та ефективний пошук даних за ключовими словами;
* мінімізація можливості виникнення помилок у роботі з програмним забезпеченням;
* контроль операцій, що використовуються для пошуку;
* зручний, простий та зрозумілий інтерфейс;
* контроль операцій, які можуть привести до змінення або пошкодження даних.

До функціональних даних відносять:

* збереження зміни даних;
* можливість керуванням рівнем доступу, для обходу зловживання з боку персоналу;
* можливість перегляду звітності за різний період часу;
* можливість керуванням активності відображення інформації для клієнтів.

До сервісних задач відносяться:

* організація головного меню та адміністративної частити;
* організація рівня доступу до програмного забезпечення;
* інформування користувача про причину відмови у виконанні тієї чи іншої операції;
* забезпечення маніпуляції з даними та синхронність даних;

Системні й апаратні вимоги які необхідні для встановлення програми наведено у таблиці 2.1.

Таблиця 2.1 – Системні й апаратні вимоги для встановлення програми

|  |  |  |
| --- | --- | --- |
| Компонент | Мінімальні вимоги | Рекомендовані вимоги |
| Процесор | Pentium / Celeron, AMD K6 / Athlon і інші, 1.1 ГГц | Pentium IV, 1.5ГГц |
| Дисковий простір | 300 Мб | 500 Мб |
| Оперативна пам’ять | 256 Мб | 512 Мб |
| Операційна система | Windows 7, 8, 8.1, 10 | |
| Додатки | ODBC-драйвери для SQL Server | |

Програмне забезпечення повинно підтримувати спільний режим з операційною системою Windows 7 та вище. Вимоги до мови програмування не передбачаються.

У процесі розробки програмного забезпечення можливе використання стандартних програмних засобів, які використовуються для розробки програмного забезпечення під управління операційної системи Windows.

Етапи розробки програмного забезпечення можуть уточнюватися згідно календарного плану робіт по узгодженню між замовником та виконавцем (див. табл. 2.2).

Таблиця 2.2 – Етапи роботи над дипломним проектом

| Етапи виконання роботи | Термін виконання та приблизний обсяг робіт | Звітні матеріали |
| --- | --- | --- |
| Передпроектні дослідження | 25.04-06.05  Аналіз предметної області та існуючих аналогів, вивчення технологій, структури даних, методів рішення тощо | Опис предметної області, аналіз аналогів, вибору методів рішення завдання та засобів розробки, оформлення технічного завдання |
| Технічний проект | 10.05-16.05  Проектування програмного забезпечення. Розробка алгоритму, визначення форми представлення даних, архітектури програми | Специфікація вимог (опис алгоритму, правил, функціональних вимог, критеріїв якості тощо), опис концептуальної, інформаційної та функціональної моделей |
| Робочий проект | 17.05-11.06  Реалізація програмного забезпечення | Опис засобів розробки, розробка документів на супроводження ПЗ (інструкції програмісту та користувачу), опис плану тестування (розробка тестів, аналіз результатів тестування) |

Оцінка результатів розробки і доцільність її продовження здійснюється замовником по представленню наступних матеріалів:

* встановлене програмне забезпечення на комп’ютерному обладнанню замовника;
* перелік файлів на резервному носії;
* стислий опис роботи додатку та опис всіх файлів, які необхідні для його роботи;
* перелік документів: технічне завдання, пояснювальна записка, додаток та методика тестування, керівництва користувача та програміста.

Склад і зміст дипломного проекту: передпроектні дослідження: аналіз предметної області та існуючих аналогів, постановка задачі, маркетингове дослідження інновацій розробки мобільного додатку, технічний проект: технічне завдання специфікація вимог, концептуальна, логічна та фізична моделі, проектування інтерфейсу, робочий проект: засоби розробки, документи на супроводження додатку (інструкції програмісту та користувачу), план тестування (розробка тестів та аналіз результатів тестування), охорона праці та техніка безпеки, висновки, додатки.

Все програмне забезпечення та супроводжуюча технічна документація повинні відповідати наступним ГОСТам:

* ІСО 5807-85 ГОСТ на розробку програмних документів, схем алгоритмів програм, даних та систем.
* ГОСТ 34.602-89 – Комплекс стандартів на автоматизовані системи. Технічне завдання на створення автоматизованої системи.
* ГОСТ 19.201-78 – Єдина система програмної документації. Технічне завдання. Вимоги до змісту та оформлення.
* ГОСТ 19.781-74 – Вимоги до розробки програмного забезпечення.
* ГОСТ 19.101-77 – Держстандарт на розробку програмної документації, видів програм та програмних документів.
* ГОСТ 29.401-78 – Текст програми. Вимоги до змісту та оформлення.
* ГОСТ 19.106-78 – Вимоги до програмної документації.
* ГОСТ 7.1-84 та ДСТУ 3008-95 – Розробка технічної документації.

Тестування виконується відповідно до «Програми та методики тестування», яка розробляється виконавцем та затверджується замовником.

У процесі розробки програмного забезпечення виконуються наступні види випробувань:

* тестування елементів системи;
* тестування системи у цілому;
* дослідна експлуатація.

Після закінчення відповідного етапу робіт формується відповідний комплект документації.

Завершення етапу розробки фіксується відповідним протоколом захисту дипломних проектів.

## 2.2 Моделювання програмного забезпечення

Після дослідження предметної області, можна побудувати концептуальну модель даних, у рамках якої інформація предметної області виражається деякими абстрактними засобами.

Концептуальна модель – це абстрактна модель, що визначає структуру модельованої системи, властивості її елементів і причинно-наслідкові зв’язки, властиві системі та суттєві для досягнення мети моделювання. Для зображення концептуальної моделі використовується діаграма прецедентів.

Діаграма прецедентів – діаграма, на якій зображено відношення між акторами та прецедентами в системі (див. рис. 2.1).



Рисунок 2.1 – Діаграма прецедентів

Опис діаграми прецедентів:

1. Актор «Користувач» – даний актор поділяється на рівні доступу до програмного забезпечення. Після авторизації для звичайного користувач, з доступом до програмного забезпечення є: «Офіціант» – при взаємодії з програмним забезпеченням який має доступ до: додавання та редагування інформації про замовлення та перегляду звітності. Актор «Адміністратор» – даний актор при взаємодії з програмним забезпеченням має повний доступ до маніпуляції над інформацією;
2. Актор «Клієнт» – даний актор при взаємодії з програмним забезпеченням, має доступ для перегляду інформації про страви та напої в електронному вигляді, для вибори під час оформлення замовлення.

Розроблена діаграма прецедентів описує структуру взаємодії між співробітником, клієнтом та програмним забезпеченням.

Діаграма взаємодій – діаграма, на якій зображено етапи взаємодії користувача з програмним забезпеченням (див. рис. 2.2).



Рисунок 2.2 – Діаграма взаємодії

Діаграма послідовності – діаграма якій описується послідовність дій під час оформлення замовлення. Коли клієнт вибирає пункти меню, офіціант оформляє замовлення, подає готове замовлення на кухню, очікує оплату, після чого офіціант віддає замовлення та рахунок клієнту (див. рис. 2.3)



Рисунок 2.3 – Діаграма послідовності

Отже, проаналізувавши предметну область було спроектовано концептуальну модель, діаграму взаємодії та послідовності.

## 2.3 Моделювання даних

Створена модель відображає концептуальні аспекти побудови об’єктної моделі програмного забезпечення і відносяться до логічного рівня представлення. Основне значення логічного представлення складається із аналізу структурних і функціональних зв’язків між елементами моделі.

Однак, для створення конкретної фізичної схеми необхідно певним чином реалізувати всі елементи логічного представлення в конкретні матеріальні сутності. Для опису таких реальних сутностей призначений другий аспект модельного представлення, а саме фізичне представлення об’єктної моделі. Для зберігання даних спроектована база даних, яка містить 8 таблиць зв’язки мiж якими представлені на рисунку 2.4.



Рисунок 2.4 – Логічна модель бази даних

В якості бази даних використовується SQL Server. База даних складається з наступних таблиць: OrderMenu, ListOrdeMenu, Food, Personal, Discounts, ListIngredientFood, ListTable, Category.

У таблиці «OrderMenu» зберігається основна інформація про замовлення. Структура таблиці вказана в таблиці 2.3.

Таблиця 2.3 –«OrderMenu»

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Признак ключа | Примітки |
| id\_order | INT | PK | Унікальне значення |
| date\_open\_order | DATETIME |  | Оформлення замовлення |
| date\_close\_order | DATETIME |  | Закриття замовлення |
| kod\_table | INT | FK | Ідентифікатор столів. Вказує на запис в таблиці ListTable |
| close\_order | BIT |  | Активність замовлення |
| payment | NUMERIC |  | Сума оплати за замовлення |
| kod\_personal | INT | FK | Ідентифікатор персоналу. Вказує на запис таблиці Personal |
| kod\_discount | INT | FK | Ідетнифікатор знижки. Вказує на запис таблиці Discounts |

У таблиці «ListOrderMenu» зберігається інформація про вміст замовлення. Структура таблиці вказана в таблиці 2.3.

Таблиця 2.3 –«ListOrderMenu»

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Признак ключа | Примітки |
| id\_listordermenu | INT | PK | Унікальне значення |
| kod\_order | INT | FK | Ідентифікатор замовлення. Вказує на запис таблиці OrderMenu |
| kod\_food | INT | FK | Ідентифікатор страв або напоїв. Вказує на запис таблиці Food |
| counts | NUMERIC |  | Кількість страв або напоїв |

У таблиці «Food» зберігається інформація про страви та напої. Структура таблиці вказана в таблиці 2.4.

Таблиця 2.4 –«Food»

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Признак ключа | Примітки |
| id\_food | INT | PK | Унікальне значення |
| name\_food | VARCHAR(50) |  | Найменування страви або напою |
| picture | VARBINARY |  | Зображення |
| kod\_category | INT | FK | Ідентифікатор категорії. Вказує на запис таблиці Category |
| weight\_food | NUMERIC |  | Кількість (вага) для страви або напою |
| unit\_food | VARCHAR(10) |  | Одиниця виміру |
| cost\_price\_food | NUMERIC |  | Собівартість страви або напою |
| mark\_up | INT |  | Націнка до собівартості страви або напою |
| price\_food | NUMERIC |  | Ціна для продажу страви або напою |
| data\_food | VARCHAR |  | Опис страви або напою |
| visible | BIT |  | Відображення страви або напою в електронному меню |

У таблиці «Personal» зберігається інформація про користувача та доступ до програмного забезпечення. Структура таблиці вказана в таблиці 2.5.

Таблиця 2.5 –«Personal»

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Признак ключа | Примітки |
| id\_personal | INT | PK | Унікальне значення |
| pib\_personal | VARCHAR(50) |  | Прізвище ініціали користувача |
| logins | VARCHAR(50) |  | Логін для доступу до програмного забезпечення |
| passwords | VARCHAR(50) |  | Пароль для доступу до програмного забезпечення |
| access | BIT |  | Рівень доступу (адміністратор, користувач) |
| activity | BIT |  | Активність користувача |
| telefon | VARCHAR(20) |  | Номер телефону |
| address | VARCHAR(50) |  | Адреса проживання |
| birthday | DATE |  | Дата народження |
| data\_of\_work | DATETIME |  | Дата обліку |
| release\_data | DATETIME |  | Дата звільнення |

У таблиці «Discounts» зберігається інформація про знижки та додаткова інформація про клієнта. Структура таблиці вказана в таблиці 2.6.

Таблиця 2.6 –«Discounts»

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Признак ключа | Примітки |
| id\_discount | INT | PK | Унікальне значення |
| pib\_client | VARCHAR(50) |  | Прізвище ініціали клієнта |
| birthday | DATETIME |  | Дата народження клієнта |
| address | VARCHAR(50) |  | Адреса проживання клієнта |
| telefon | VARCHAR(20) |  | Номер телефону клієнта |
| discount | INT |  | Знижка |

У таблиці «ListIngredientFood» зберігається інформація інгредієнти пов’язані зі стравою. Структура таблиці вказана в таблиці 2.7.

Таблиця 2.7 –«ListIngredientFood»

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Признак ключа | Примітки |
| id\_listingrfood | INT | PK | Унікальне значення |
| kod\_food | INT | FK | Ідентифікатор страви. Вказує на запис таблиці Food |
| counts | NUMERIC |  | Кількість (вага) інгредієнта |
| name\_ingredient | VARCHAR(50) |  | Найменування інгредієнта |
| unit | VARCHAR(10) |  | Одиниця виміру інгредієнта |
| price | NUMERIC |  | Ціна за одиницю виміру |

У таблиці «ListTable» зберігається інформація про найменування місць для обслуговування клієнтів. Структура таблиці вказана в таблиці 2.8.

Таблиця 2.8 –«ListTable»

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Признак ключа | Примітки |
| id\_table | INT | PK | Унікальне значення |
| name\_table | VARCHAR(50) |  | Найменування місця для обслуговування |
| occupation | BIT |  | Зайнятість |

У таблиці «Category» зберігається інформація про найменування місць для обслуговування клієнтів. Структура таблиці вказана в таблиці 2.9.

Таблиця 2.9 –«Category»

|  |  |  |  |
| --- | --- | --- | --- |
| Назва поля | Тип даних | Признак ключа | Примітки |
| id\_category | INT | PK | Унікальне значення |
| name\_category | VARCHAR(50) |  | Найменування категорії |

Зв’язок між таблицями «OrderMenu» і «ListOrderMenu» – один-до-багатьох, так як замовлення може містити декілька записів вибраних страв або напоїв, зв’язок між таблицями «Food» і «ListOrderMenu» – один-до-багатьох, так як в таблиці страви можна вибрати один запис страви або напою для списку замовлення, зв’язок між таблицями«Food» і «ListIngredientFood» – один-до-багатьох, так як одна страва може містити багато записів про інгредієнти, зв’язок між таблицями «Category» і «Food» – один-до-багатьох, так як одна категорія містить багато записів про страви або напої, зв’язок між таблицями «OrderMenu» і «Discounts» – один-до-одного, так як замовлення може містити лише один запис пов’язаний з знижкою, зв’язок між таблицями «OrderMenu» і «Personal» – один-до-одного, так як замовлення може містити лише один запис пов’язаний з працівником, який обслуговував клієнта, зв’язок між таблицями «OrderMenu» і «ListTable» – один-до-одного, так як одне замовлення містить лише один запис про місце обслуговування клієнта.

Для реалізації поставленої задачі використано принцип об’єктно-орієнтованого програмування та було спроектована ієрархію класів, яка представлена у вигляді діаграми класів (див. додаток А).

## 2.4 Проектування інтерфейсу програмного забезпечення

Взаємодія між користувачем і програмним забезпечення відбувається за допомогою інтерфейсу. Найчастіше ефективність програмного забезпечення визначають у більшому ступені тим, як побудований його інтерфейс. Тому інтерфейс програмного забезпечення має бути простий, зручний та інтуїтивно-зрозумілий.

Для реалізації інтерфейсу програмного забезпечення необхідно спроектувати декілька вікон, для зручності роботи з програмним забезпеченням враховано, що адміністратор та офіціант, та клієнт маються різні можливості, так як у них є різний рівень доступу.

Інтерфейс вікна «Головне вікно» програмного забезпечення зображено на рисунку 2.5. Дане вікно призначено для перегляду вмісту інформації меню в електронному вигляді, який можна переглядати при виборі категорії меню, в якому буде відображено повну інформацію пов’язану з даною категорію. Також реалізовано пошук за допомогою якого, можна знайти потрібно інформацію яка пов’язана з стравами або напоями. Також дане вікно використовується для оформлення замовлення, після авторизації в систему, відображаються скриті компоненти, які надаються можливість офіціанту вибрати страви та напої для оформлення замовлення (див. рис. 2.6).



Рисунок 2.6 – Вікно «Головне вікно»

Інтерфейс вікна «Замовлення» програмного забезпечення зображено на рисунку 2.7. Дане вікно призначено для офіціанта, для якого буде реалізовано, управління даними тобто: оформлення замовлення, закриття замовлення, додавання та видалення вибраних страв або напоїв з замовлення, знижки для клієнтів, друк рахунку, перегляд вільних місць, інформація про замовлення.



Рисунок 2.7 – Вікно «Замовлення»

Інтерфейс вікна «Адміністратор» програмного забезпечення зображено на рисунку 2.8. Дане вікно призначено для адміністратора, який має повний доступ до програмного забезпечення, тобто: обробка даних, пошук інформації за різним ключем, створення звітності за різними критеріями, управління користувачами та доступ до програмного забезпечення.



Рисунок 2.8 – Вікно «Адміністратор»

Для авторизації користувача під різний рівень доступу спроектовано інтерфейс вікна «Авторизація», яке зображено на рисунку 2.9. Дане вікно призначено для вводу особистих даних та вибору необхідної дії.



Рисунок 2.9 – Вікно «Авторизація»

Також спроектовані вікна для редагування інформації

# РОЗДІЛ 3. робочий проект

## 3.1 Засоби розробки

Для створення дієвого і простого у використанні програмного продукту було обрано середовище Embarcadero RAD Studio 10.2 Architect 2017.

Embarcadero RAD Studio 10.2 – це комплексне рішення для швидкої розробки додатків Window, .NET, веб-додатків і додатків баз даних. Embarcadero RAD Studio 10.2 об’єднує Delphi 10.2 і С++ Builder 10.2 в єдину інтегровану середу розробки. У даній версії також реалізована підтримка таких технологій: .NET, як WinForms, WPF, ADO.NET, ASP.NET і LINQ.

Додаткова підтримка фреймворку Mono забезпечує можливості створення крос-платформних додатків, які працюють під операційними системами Windows, Linux і Mac OS X.

Спочатку розроблявся компанією Borland Software, а потім її підрозділом CodeGear, що на сьогоднішній день належить компанії Embarcadero Technologies.

Embarcadero RAD Studio 10.2 об’єднує в собі комплекс об’єктних бібліотек (STL, VCL, CLX, MFC та ін.), компілятор відкладчик, редактор коду та багато інших компонентів.

Можливості Embarcadero RAD Studio 10.2:

* підтримка Windows 7 API і Direct2D;
* підтримка жестів і технології multi-touch для Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8-8.1 та Windows 10;
* IDE Insight – миттєвий доступ до будь-якої функції компонентів, налаштувань;
* понад 120 удосконалень для підвищення продуктивності;
* візуалізатори налагодження;
* підтримка Firebird.

Класичний інтерфейс Delphi 7 і C++ Builder 6 – RAD Studio Architect включає всю функціональність версії Enterprise, а також потужні можливості моделювання та проектування баз даних:

* зворотне проектування, аналіз і оптимізація баз даних;
* автоматична генерація коду баз даних на основі моделей для прямого проектування;
* створення логічних і фізичних моделей на основі відомостей, витягнутих з баз даних і файлів сценаріїв;
* робота із зручними для читання і навігації діаграмами;
* вдосконалене порівняння і об’єднання за допомогою двонаправлених операцій порівняння і об’єднання моделей і структур баз даних.

Основні можливості для розробки додатків Windows:

* інтегроване середовище розробки Delphi і С++ Builder з функцією перетягування курсором;
* більше 250 компонентів в бібліотеці VCL;
* вбудована підтримка торкань і жестів (більше 30 готових жестів);
* dbExpress з підтримкою 9 основних баз даних, включаючи InterBase;
* DataSnap з підтримкою JSON, REST, HTTP, COM і XML;
* Можливість налагодження багато поточних додатків;
* UML / аудит коду та облік показників;
* підтримка Windows 2000 та вище.

## 3.2 Інструкція програміста

## 3.3 Інструкція користувача

## 3.4 Тестування програмного забезпечення

# Розділ 4. ОХОРОНА ПРАЦІ

**Четвертий розділ** повинен складатися з:

* правових та організаційних питань охорони праці;
* організації роботи з охорони праці на підприємствах;
* електробезпеки, дії електричного струму на організм людини, засоби захисту від враження електричним струмом;
* санітарно-гігієнічних вимог до обладнання приміщень операторів ПК та організації робочих місць;
* параметрів мікроклімату повітря робочої зони;
* вимоги техніки безпеки до організації робочого місця оператора ПК;
* пожежної безпеки, правил пожежної безпеки для приміщень операторів ПК, вогнегасильні речовини та засоби гасіння пожеж.

# ВИСНОВКИ

**Висновки** оформляють з нової пронумерованої сторінки із заголовком Висновки посередині рядка.

Висновки є заключною частиною, підсумком прийнятого рішення виконаного проекту із зазначенням досягнутих результатів, визначенням перспектив їх покращення, пропозиціями реалізації проекту.

# перелік джерел

**Перелік джерел** оформляють з нової пронумерованої сторінки із заголовком ПЕРЕЛІК джерел посередині рядка.

Перелік літературних джерел, на які є посилання в пояснювальній записці, подають на окремих аркушах, крім того, у відповідних місцях пояснювальної записки повинні бути посилання на подані джерела інформації.

Список джерел слід розміщувати в алфавітному порядку прізвищ перших авторів. Бібліографічний опис джерел складають відповідно до чинних стандартів (ДСТУ ГОСТ 7.1:2006. Бібліографічний запис. Загальні вимоги та правила складання).

На всі наведені джерела мають бути посилання в тексті курсового проекту (в місцях цитування).Посилання на джерела необхідно вказувати порядковим номером переліку джерел, відокремлюючи квадратними дужками. Наприклад:

Для вирішення багатьох задач по розкрою найбільш прийнятними є евристичні методи [4, с. 38-41].

Приклади оформлення переліку джерел наведено в додатку Е.

# Додатки

**Додатки** оформляють з нової пронумерованої сторінки із заголовком Додатки посередині сторінки.

До додатків пояснювальної записки доцільно включати допоміжний матеріал, необхідний для повноти сприйняття ДП:

* таблиці допоміжних цифрових даних;
* графічні матеріали;
* допоміжні ілюстрації.

Додатки оформляють як продовження ПЗ у вигляді окремої частини, розміщуючи їх у порядку появи посилань у тексті ПЗ.

Додатки слід позначати послідовно великими літерами української абетки, за винятком літер Г, Є, І, Ї, Й, О, Ч, Ь, наприклад, Додаток А.

**Додаток А**

**Приклад складення анотації**

**Анотація**

Пояснювальна записка складається з 139 ст., 27 рис., 8 таблиць, 5 діаграм, 3 додатків, 18 джерел.

Об’єкт проектування – автоматизована інформаційна система.

Мета проектування – проаналізувати специфіку діяльності таксопарку, створити автоматизовану інформаційну систему «Таксопарк» із детальною розробкою робочої станції диспетчера.

Метод проектування – платформа .NET, система розробки об’єктно-орієнтованих продуктів Microsoft Visual Studio 2008, мова програмування С#, програмні продукти Framework 3.0 і Crystal Report 2008, а також система керування базами даних MS SQL Server 2005.

У результаті дипломного проектування створена база даних автоматизованої інформаційної системи і робочої станції диспетчера таксопарку, яка дозволяє вести облік основних засобів таксопарку, облік усіх видів робіт, виконаних співробітниками, облік витрат пально – мастильних матеріалів, облік витрат на утримання будівель і території тощо.

Прогнозні припущення щодо розвитку об’єкта дослідження – реалізація мережевої автоматизованої системи.

Ключові слова: ПРОГРАМНА СИСТЕМА, БАЗА ДАНИХ, ТАКСОПАРК, MS SQL SERVER, РОБОЧА СТАНЦІЯ ДИСПЕТЧЕРА.

Додаток Б

**Приклад змісту**

**ЗМІСТ**

ВСТУП 3

Розділ 1. Передпроектні дослідження

1.1 Аналіз предметної області

1.2 Аналіз існуючих аналогів

1.3 Маркетингове дослідження інновацій

Розділ 2. Технічний проект

2.1 Специфікація вимог

2.1.1 Опис алгоритму

2.1.2 Опис функціональних вимог

2.1.3 Опис критеріїв якості

2.2 Технічне завдання

2.3 Концептуальна модель

2.4 Функціональна модель

2.3 Проектування інтерфейсу

Розділ 3. Робочий проект

3.1 Засоби розробки

3.2 Документи на супроводження ПЗ

3.2.1 Інструкції програмісту

3.2.2 Інструкції користувачу

3.3 План тестування

3.3.1 Розробка тестів

3.3.2 Аналіз результатів тестування

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

ВИСНОВКИ

Перелік джерел

ДОДАТКИ

Додаток В

**Приклад оформлення переліку умовних позначень та скорочень**

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

БД – база даних

ІС – інформаційна система

ІТ – інформаційні технології

ООП – об’єктно-орієнтоване програмування

ОС – операційна система

ПЗ – програмне забезпечення

СУБД – система управління базами даних

ШІ – штучний інтелект

Додаток Г

**Приклад оформлення вступу**

**ВСТУП**

**[Актуальність роботи]**

На сучасному етапі розробка інформаційних інтелектуальних систем – є однією з фундаментальних задач у області аналізу даних і технології Data Mining [1]. Список прикладних областей широкий: маркетинг, боротьба з плагіатом, прогнозування, аналіз текстів і багато інших.

Одним з перспективних напрямків є створення комп’ютерних ІС і онтологізованних E-learning………..

Тому в останнє десятиліття ведуться активні дослідження у області розробки програмного забезпечення (ПЗ) інформаційних інтелектуальних систем і, зокрема розробки ПЗ обробки онтології.

……………………….

Як буде вказано нижче питання пов’язані із використанням онтології добре піддаються комп’ютеризації. Першим етапом при машинній обробці є порівняння онтології. Наявне ПЗ (Protеgе…) не має графічного інтерфейсу для злиття онтології. Необхідно розробити ПЗ, яке забезпечує графічний інтерфейс обробці і зокрема злиття онтології. Для цього необхідно розробити теоретично обґрунтовані вибори набора операцій для порівняння онтології і реалізації його у вигляді класів в конкретній мові.

…………………………………………………………………………

У проекті був проведений аналіз існуючих класів для понятійного порівняння онтологий та вибраний і реалізований найбільш підходящий алгоритм для порівняння онтологій, що показує досить високу ефективність порівняно з іншими подібними алгоритмами.

**[Мета і задачі роботи]**

Метою дипломного проекту є розробка ПЗ для понятійного порівняння онтологій на прикладі предметної області БД. ПЗ повинно робити порівняння даних, представлених у вигляді онтології.

Досягнення мети включало розв’язання таких задач:

* аналіз алгоритмів штучного інтелекту огляд абстрактної алгебри;
* вивчення онтологізованих ІС;
* опис класів для понятійного порівняння онтологій;
* вибір релевантного алгоритму та обґрунтування доцільності його використання;
* реалізація алгоритму.

Об’єктом дослідження є дані предметної області, представлені у вигляді онтології.

Предметом дослідження є технологія порівняння даних, представлених у вигляді онтології.

**[Практичне значення одержаних результатів]**

Одержана реалізація алгоритму, що дозволяє виконувати порівняння даних, представлених у вигляді онтології. А це дозволяє спростити подальшу обробку даних.

Основним результатом є:

* розроблені класи для понятійного порівняння онтологій;
* приведена реалізація алгоритму, що виконує порівняння даних, представлених у вигляді онтології.

Пояснювальна записка дипломного проекту складається із вступу, …., економічного розділу, розділу охорони праці, висновків, переліку посилань на джерела та додатків.

У першому розділі наведено …..

Додаток Д

**Приклад оформлення таблиць та ілюстрацій**

**Д.1 Оформлення таблиць**

Всі таблиці нумерують в межах розділу. Номер складається з номера розділу і порядкового номера таблиці, розділених крапкою, наприклад; «Таблиця 2.2» (друга таблиця другого розділу). Над продовженням таблиці на новій сторінці пишуть «Продовження табл. 2.2» (без лапок), тематичний заголовок не повторюють. На всі таблиці повинні бути посилання в тексті, наприклад: «... у таблиці 3.2», «... (див. таблиця 3.2) ...».

Приклад подання таблиць.

Системні й апаратні вимоги які необхідні для встановлення програми наведено у таблиці 3.2.

Таблиця 3.2 – Системні й апаратні вимоги для встановлення програми

|  |  |  |
| --- | --- | --- |
| Компонент | Мінімальні вимоги | Рекомендовані вимоги |
| Процесор | Pentium / Celeron, AMD K6 / Athlon / Duron і інші, 550МГц | Pentium IV, 1ГГц |
| Дисковий простір | 30 Мб | 500 Мб |
| Оперативна пам’ять | 256 Мб | 512 Мб |
| Операційна система | Windows Xp | |
| Додатки | Microsoft Office 2003 або пізнішої версії (або ODBC-драйвери) | |

*Вимоги до тексту в таблицях:*

* шрифт – Times New Roman;
* розмір – 12 пт;
* інтервал між рядками – 1.0;
* відступи не допускаються.

Заголовки граф повинні починатися з великих літер. Висота рядків повинна бути не меншою 8мм. Графу з порядковими номерами рядків до таблиці включати не треба.

**Д.2 Оформлення ілюстрацій**

Ілюстрації (рисунки, схеми, тощо) необхідно розміщати безпосередньо після першого нагадування про них у тексті, усі вони нумерують й підписують знизу. Ілюстрації, розміри якої більше половини формату А4, можна винести у додатки.

Приклад посилання на рисунки: «... на рис. 1.3 ...» або «... на рисунку 1.3 ...», «... (див. рисунок 1.3) ...» або «... (рис. 1.3) ...».

Приклад 1.

Термінатор – символ відображає вихід в зовнішнє середовище та вхід із зовнішнього середовища (рис. 3.10, б).

б

а

в

Рисунок 3.10 – Спеціальні символи: а – з’єднувач;

б – термінатор; в – коментар

Приклад 2.

Схеми програм відображають послідовність операцій в програмі (див. рисунок 3.21).

Зупинка програми. По ручному перезапуску повернутися до місця з’явлення помилки і попробувати ще раз

Початок роботи

Арифметична обробка в ЕОМ

Передача результату в запам. пристрій

Перезапуск

Передача успішна

Ні

Помилка

Передача результату в запам. пристрій

Передача успішна

Ні

Контр-на передача

Так

Так

Зупинка

***Рисунок 3.21 – Схема програми***

Схеми роботи системи відображають управління операціями і потік даних в системі.

**Д.3 Оформлення формул**

Формули і рівняння розташовують безпосередньо після їхнього нагадування у тексті, посередині сторінки. Нумеруються формули та рівняння в межах розділу. Номер включає в себе номер розділу та порядковий номер формули (рівняння), розділених крапкою, наприклад: (2.3) – третя формула другого розділу. Номер формули (рівняння) вказують у тому ж рядку у крайньому правому положенні.

Пояснення значення кожного символу і числового коефіцієнта необхідно давати з нового рядка. Перший рядок пояснення починають з абзацу словом «де» без двокрапки. Наприклад:

Питома провідність плазми

 (2.3)

де *n* – концентрація електронів;

*е* – заряд електрона;

*m* – його маса;

*τ* – інерційній час.

Посилання на формули вказують порядковим номером формули в дужках, наприклад «... у формулі (3.1)», «... за формулою (3.1) ...», «... у рівняннях (1.3)-(1.5) ...».

Додаток Е

**Приклади оформлення списку використаних джерел**

**ПЕРЕЛІК ДЖЕРЕЛ**

*Приклад – статті з журналу*

1. Юрчишин В.М. Методика формування баз знань експертних систем для аналізу режимів роботи нафтових родовищ. – Івано-Франківськ. // Нафтогазова енергетика, 2007. – №1. – 22 с.

*Приклад – книги*

1. Шилдт Г. Java. Руководство для начинающих / Г. Шилдт; пер. с англ. – М.: ООО «И.Д. Вильямс», 2012 – 624 с.

*Приклад – патентні документи*

1. Патент UA 85387 A / Україна. Спосіб вимірювання покладів геотермальних вод на території Прикарпатського регіону / Юрчишин В.М. Опубл. 25.03.2007, Бюл. №35.

*Приклад – стандарти*

1. ДСТУ 2293-93. Система стандартів безпеки праці. Терміни та визначення.

*Приклад – дисертація*

1. Самохіна Н.Ф. Фонд наукової бібліотеки: модель обігу документів, засоби рівневої організації: дис.: канд.. техн. наук 05.13.06 – захищена 24.06.2007.

*Приклад – інтернет-посилання (посилання на статтю)*

1. Точилкина Т.Е. Создание организационной диаграммы в All Fusion Process Modeler. – [www.interface.ru/fset.asp?Url=/ca/sso.htm&anchor=2](http://www.interface.ru/fset.asp?Url=/ca/sso.htm&anchor=2)