

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

## **ВИПУСКНА РОБОТА**

**на тему:**

**«Інформаційна технологія проєктування  
адаптивної системи онлайн-замовлень продуктів  
харчування»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Берест О.Б.**

**Студента групи ІК.м – 01**

**Полушкіної І.О.**

**СУМИ 2021**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

**Кафедра комп'ютерних наук**

Затверджую \_\_\_\_\_

Зав. кафедрою Довбиш А.С.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 г.

**ЗАВДАННЯ  
до випускної роботи**

Студента другого курсу, групи ІК.м-01 спеціальності “Інформаційно-комунікаційні технології” денної форми навчання Полушкіної Інни Олександрівни.

**Тема:** “ Інформаційна технологія проєктування адаптивної системи онлайн-замовлень продуктів харчування”.

Затверджена наказом по СумДУ

№ \_\_\_\_\_ от \_\_\_\_\_ 2021 г.

**Зміст пояснювальної записки:** 1) огляд аналогічних систем; 2) огляд типу архітектури системи, вибір типу програмного забезпечення; 3) вибір технологій для розробки проєкту, серед яких: мова програмування, база даних, фреймвокри;5) результат розробки проєкту.

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 2021 г.

Керівник випускної роботи \_\_\_\_\_ Берест О.Б.

Завдання прийняла до виконання \_\_\_\_\_ Полушкіна І.О.

## РЕФЕРАТ

**Записка:** 51 стор., 22 рис., 1 табл., 20 джерел.

**Об'єкт дослідження** — інформаційна технологія проєктування адаптивної системи онлайн-замовлень продуктів харчування.

**Мета роботи** — розробка інформаційної технології проєктування адаптивної системи онлайн-замовлень продуктів харчування.

**Результати** — розроблено веб-додаток інформаційної технології проєктування адаптивної системи онлайн-замовлень продуктів харчування, що дає можливість користувачам створювати онлайн замовлення на основі даних про заклади харчування, меню закладів та попередніх замовлень, користувачі можуть переглядати список попередніх замовлень, може відмінити активне замовлення за бажанням.

**Ключові слова:** інформаційна технологія проєктування, система, веб-додаток, проєкт, фреймворк, база даних, алгоритм аналізу.

## Зміст

<b>Вступ</b> .....	5
<b>1 Огляд аналогів та технологій</b> .....	6
<b>2 Вибір методів рішення та постановка задачі</b> .....	8
<b>2.1 Архітектура</b> .....	8
<b>2.2 Вибір типу програмного забезпечення</b> .....	10
<b>2.3 Постановка задачі</b> .....	12
<b>3 Проєктування додатку</b> .....	13
<b>3.2 Алгоритм аналізу</b> .....	19
<b>4 Огляд технологій реалізації проєкту</b> .....	22
<b>4.1 Аналіз мов програмування</b> .....	24
<b>4.2 Вибір фреймворку для розробки веб-додатку</b> .....	26
<b>4.3 Вибір системи управління базаю даних</b> .....	28
<b>5 Розробка інформаційної технології</b> .....	31
<b>5.1 Створення бази даних</b> .....	31
<b>5.2 Розробка компонентів додатку</b> .....	37
<b>5.3 Результати реалізації веб-додатку</b> .....	40
<b>Висновки</b> .....	49
<b>Список використаної літератури</b> .....	51

## Вступ

У сучасному світі є актуальною тема замовлення їжі через інтернет. Через швидкий темп життя не всі люди мають достатньо часу для звершення покупок продуктів харчування або ж для приготування смачної та корисної їжі. Тому онлайн замовлення та доставка продуктів харчування є актуальними послугами.

Подібні системи мають багато переваг для користувачів, оскільки дозволяють користуватися послугами ресторанів або інших закладів харчування невиходячи з дому.

Метою проєкту є створення інформаційної технології проєктування адаптивної системи онлайн-замовлень продуктів харчування, яка буде зручна та проста в експлуатації.

В першу чергу для створення такої технології слід проаналізувати аналогічні працюючі рішення для того, щоб побачити загальну картину сучасних технологій на створити систему, що буде актуальною для користувачів.

Для успішного створення інформаційної технології необхідно обрати архітектуру, оскільки вона є основою будь-якої системи, та обрати найбільш зручний та актуальний тип програмного забезпечення майбутньої системи. Своєчасне проєктування дозволить розширювати систему в майбутньому.

## 1 Огляд аналогів та технологій

У сучасному світі є актуальною тема замовлення їжі через інтернет. Через швидкий темп життя не всі люди мають достатньо часу для звершення покупок продуктів харчування або ж для приготування смачної та корисної їжі. Тому онлайн замовлення та доставка продуктів харчування є актуальними послугами.

Розглянемо популярні в Україні онлайн системи замовлення та доставки продуктів харчування.

**Glovo** – один із популярних сервісів онлайн замовлення та доставки продуктів харчування не тільки в Україні, а й в багатьох країнах Європи. Сервіс доставки Glovo має власну онлайн систему через яку клієнти можуть робити замовлення, ця система працює за допомогою мобільного додатку, який можна завантажити через Google Play або App Store, також ця система має зручний та приємний у користування веб-додаток.

За допомогою даного онлайн систем Glovo користувач може доставку товарів невеликого розміру, частіше за все їжа. Для замовлення користувач може використовувати сайт або мобільний додаток. Сервіс дозволяє обрати кафе, ресторан чи супермаркет, у якому можна замовити їжу, переглянути меню закладу та зробири замовлення та його оплату. Після того як користувач сторив замовлення, сервіс дає можливість відстежувати рух кур'єра по карті міста в режимі реального часу.

**Mister Am** – сервіс доставки їжі, через який також можна здійснити замовлення через сайт або мобільний додаток. Данний сервіс також має онлайн систему для створення замовлення, яка працює за допомогою веб-додатку.

Веб-додаток данного сервісу дає можливість користувачу переглянути у зручному форматі список ресторанів, розглянути меню кожного ресторану та замовити бажане блюдо. Після оформлення замовлення, додаток відправляє замовнику повідомлення підтвердження про успішне створення замовлення та орієнтований час доставки. Також користувач може обрати бажаний спосіб оплати: онлайн або готівкою.

Ще однією особливою функцією онлайн системи сервісу Mister Am є політика отримання кешбеку за кожне замовлення. Після кожного замовлення користувач отримує кешбек, який потім можна використати для часткової оплати наступного замовлення.

**Rocket** – це український сервіс доставки їжі, який все більше набуває популярності останнім часом. Сервіс пропонує робити замовлення з багатьох кафе, ресторанів або ж магазинів та супермаркетів. Для створення замовлення даний сервіс пропонує клієнтам використовувати мобільний або веб додатки.

Онлайн система дозволяє обрати необхідні страви з ресторану або ж продукти харчування з супермаркету. Система розраховує вартість всього замовлення, за умови можливих акцій в закладах харчування. Також додаток пропонує обрати клієнту зручний для нього спосіб оплати, відсилає повідомлення про успішне створення замовлення, зміну статусу замовлення та місцеперебування кур'єра.

Вищеописані системи мають великий ряд переваг, серед яких можливості для клієнтів зручно та швидко створити необхідне замовлення продуктів харчування, обрати спосіб оплати, отримувати інформацію про статус замовлення та багато. Але в них не наявної функція аналізу історичних даних попередніх замовлень користувачів та прогнозування можливих замовлень користувачів на основі їх попередніх. Отже пропонуємо створити власну інформаційну систему онлайн замовлень для продуктів харчування.

## **2 Вибір методів рішення та постановка задачі**

Інформаційна система онлайн-замовлень продуктів для хачування має одночасно оброблювати запити багатьох клієнтів та надавати їм необхідну інформацію, а також фіксувати створені ними нові замовлення або відміну замовлень. Це означає, що багато клієнтів матимуть одночасний доступ до ресурсів системи.

Інформаційна технологія повинна оброблювати великий об'єм історичних даних про попередні замовлення. На основі історичних даних система має робити аналіз в залежності від категорії або закладу в якому було створено більшість попередніх замовлень. Після аналізу історичних даних інформаційна онлайн система повинна запропонувати користувачу варіанти можливих замовлень, що можуть бути йому цікавими.

### **2.1 Архітектура**

Від архітектури програмного забезпечення залежить зручність використання звичайними користувачами, рівень якості інформаційної системи, що розробляється, успішна робота системи протягом тривалого часу.

Архітектура - це основна організація системи, втілена в її компонентах, зовнішніх та внутрішніх відносинах у системі, а також принципи, що визначають проектування та розвиток системи. [1]

Правильно та своєчасно спланована архітектурна модель полегшить не лише подальше використання онлайн-інформаційної системи, а й процес розробки, впровадження та модифікації системи за необхідності.

Архітектура клієнт-сервер найкраще підходить для створення подібних онлайн-інформаційних систем.



Сучасний клієнт-серверний підхід включає сервер, який в основному надає доступ до ресурсів (зазвичай зв'язок між клієнтом і сервером передається в XML або JSON) у відповідь на запити клієнта. Асинхронні серверні запити часто використовуються у сучасній моделі клієнт-сервер. Сервер спочатку підтримує сторінки з невеликою кількістю даних. Сторінки у відповідь на дії користувача роблять асинхронні запити до сервера і сервер просто відповідає на ці повідомлення, що викликає оновлення поточної сторінки. [2]

Згодом технології розвиваються, а потреби користувачів змінюються, що змушує розробників змінювати архітектурні моделі. Сьогодні веб-розробник повинен використовувати інструменти та підхід до розробки цього відповідно до сучасної веб-сцени. В даний час все більше і більше розробників використовують підхід клієнт-сервер, оскільки він дозволяє їм зробити архітектуру програми більш гнучкою. Завдяки цьому, якщо потрібні зміни - редагування або додавання, можна зробити набагато швидше. [3]

З розвитком нашої системи внесення змін буде значно простіше вносити зміни у додаток, який створений на основі архітектурної моделі клієнт-сервер. Це дає можливість клієнтській частині робити обмін повідомленнями з серверною частиною не вникаючи у подробиці реалізації, а також дає можливість вносити зміни тільки для однієї сторони без внесення змін в іншу частину та реалізацію механізму обробки та обміну повідомленнями.

Явною аргументом для використання цього підходу є перспектива розділити код на логічні частини, що дасть вищу згуртованість як для оригінальної конструкції, так і в подальшій підтримці будь-якої системи. Чіткий поділ між серверною та клієнтською частинами зробить реалізацію коду модульною та швидко керованою. Окрім того, розмітка відображення дані зможуть бути чітко відокремлені. Повторне використання та гнучкість

коду - логічний результат правильної організації коду. Повторне використання коду більш можливе, коли є чіткі компоненти. Як мінімум, ті самі REST API можуть бути використані для відображення та показу даних у різноманітних браузерах та на мобільних пристроях.[3]

Архітектурну модель клієнт-сервер доцільно використати для створення інформаційної адаптивної система онлайн-замовлень продуктів для харчування. Така модель, за допомогою розділу на серверну та клієнтську частини, дає можливість користувачу швидко отримати необхідну інформацію про заклади харчування, переглянути меню кожного закладу та замовити необхідні блюда. Завдяки використанню даного підходу під час створення інформаційної системи, в результаті буде отримано структурований та легкокерований код, в який буде легко вносити подальші зміни.

## **2.2 Вибір типу програмного забезпечення**

Інформаційна адаптивна система для онлайн замовлень продуктів харчування має будувати простою та зручно у використанні та керувати даними закладів харчування та замовлень клієнтів.

Для сучасної людини є звичайними щодня використовувати різні інформаційні системи для керування різними даними. Доступ до різних інформаційних систем зараз можна отримати через мобільний, веб або десктоп додатки.

Веб-додаток – це додаток, створений на основі клієнт-серверної архітектурної моделі, у якому сервер та клієнт взаємодіють за допомогою веб-бузера. Робота веб-додатку логічно розподілена між клієнтом і сервером, обмін даними між клієнтською та серверною частинами виконується за допомогою комп'ютерної мережі. Основну частину даних зберігає база даних, доступ до якої має сервер, також частина даних може зберігатися на

сервері, який в свою чергу буде взаємодіяти з клієнтською частиною та надавати інформацію клієнту у відповідь на запити. Тобто клієнтський пристрій може зберігати лише мінімальну кількість необхідних даних, що є значною перевагою для клієнта, оскільки використання пам'яті пристроєм є мінімальним. Це дає додатку ще одну перевагу – незалежність від операційної системи, тобто не має необхідності створювати декілька додатків для різноманітних операційних систем, а буде достатнім створення лише одного додатку.

Мобільний додаток – це програма, призначена для роботи з мобільних пристроїв. Додатки такого типу програмного забезпечення потрібно розроблювати окремо для конкретної платформи. Мобільні додатки також створюють беручи за основу клієнт-серверну архітектуру, але в них значну частину даних зберігає клієнтський пристрій. Мобільний додаток зазвичай використовує більший об'єм ресурсів клієнтського пристрою ніж веб-додаток. Частіше за все, такий програмного забезпечення необхідний для реалізації системи, яка повинна часто взаємодіяти з користувачем, наприклад, для системи миттєвого обміну повідомленнями.

Десктоп додаток – це програмне забезпечення, яке встановлене на клієнтському комп'ютері. Також зазвичай має, клієнт-серверну архітектуру, яка взаємодіє за допомогою мережі, але не завжди таким додаткам потрібен доступ до мережі, бо вони можуть самостійно виконувати роботу. Даний вид додатку використовує вагомую частину клієнтських ресурсів. Під час проектування потрібно знайти рішення проблеми залежності додатку від платформи.

Для створення нашої інформаційної системи краще обрати веб-додаток. Цей варіант має ряд вищеписаних переваг перед іншими, саме він буде найбільш зручним у використанні.

## 2.3 Постановка задачі

Отже постала задача спроектувати та розробити схоже програмне забезпечення, яке буде :

- 1) Web-додатком;
- 2) Дозволятиме клієнтам обирати заклад(кафе, ресторан, супермаркет) для замовлення продукту харчування;
- 3) Робити замовлення в обраному закладі;
- 4) Аналізувати попередні замовлення користувачів та прогнозувати їх наступні замовлення;
- 5) Використовувати базу даних для зберігання інформації про клієнтів на їх замовлення;
- 6) Мати простий та зрозумілий інтерфейс користувача, який не потребує додаткових технічних знань .

### 3 Проєктування додатку

Одним з перших кроків в процесі розробки будь-якої програми є побудова різноманітних допоміжних діаграм, які в подальшому допомагають спростити весь процес розробки. Для проєктування інформаційної технології будемо використовувати загальноприйнятну мову моделювання UML.

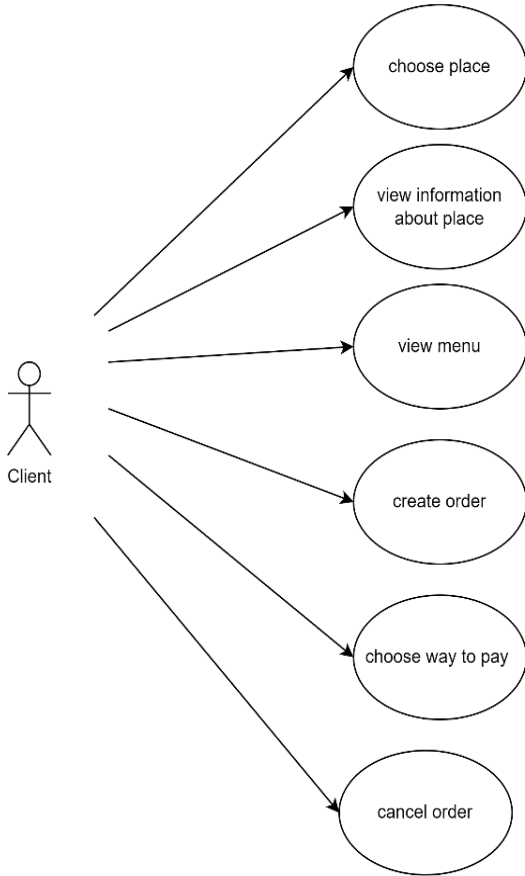
Уніфікована мова моделювання (UML) – це родина графічних нотацій, єдина метамодель є їх основою. [4] Ця мова допомагає в проєктуванні та описанні інформаційних систем, особливо для створення систем, які використовують в реалізації об'єктно-орієнтовані властивості.

Для зображення дій сервера або клієнта в системі побудуємо use case діаграму. Суть даної діаграми полягає в розділенні системи у вигляді множин акторів та сутностей, які відображають варіанти використання (use case). Актором називають клієнтську сутність, що може взаємодіяти з системою. Наприклад, програма або технічний пристрій за допомогою якого людина може взаємодіяти з системою. При цьому, варіант використання (use case) відповідає за опис дій сервісів, які система дає актору. Тобто, кожен варіант використання представляє деякий набір дій, які робить система під час комунікації актором. При цьому взаємодія акторів з системою може бути реалізована багатьма різними способами.[5]

Даний вид діаграм показує відношення між прецедентами (варіанти дій) та акторами системи.

На Рисунку 3.1 зображено use case діаграму інформаційної системи.

## Client side



## Server side

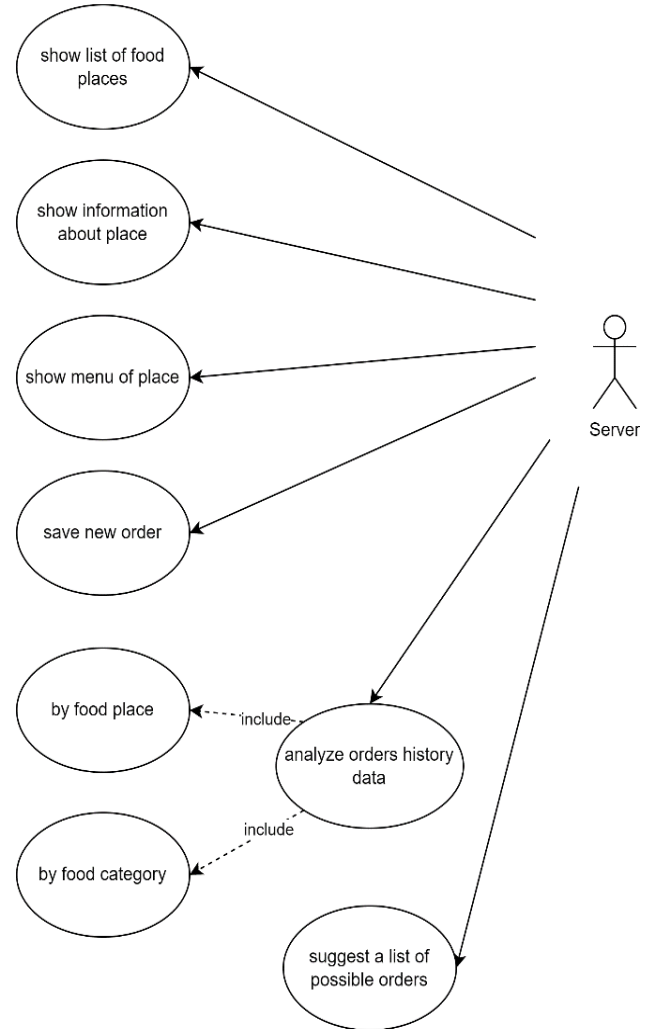


Рисунок 3.1.1 – Use Case Diagram

Актори системи:

- client – авторизований користувач системи;
- server – сервер інформаційної системи.

Сервер оброблює запити клієнтів, контролює процеси обміном даними, надаючи клієнтам необхідну інформацію. Клієнт, в залежності від свого статусу, може посилати відповідні запити на сервер та отримувати від нього відповіді.

Клієнти можуть виконувати такі дії, як :

- обрати заклад для замовлення;
- переглянути інформацію про заклад;
- переглянути меню закладу;
- створити замовлення;
- обрати зручний спосіб оплати;
- відмінити замовлення.

Сервер може виконувати такі дії:

- показати список закладів харчування;
- показати інформацію про заклад;
- показати меню закладу;
- зберегти нове замовлення, створене клієнтом;
- зробити аналіз історичних даних для користувача в залежності від категорії замовлення або категорії закладу;
- зробити проноз наступного замовлення користувача на основі попереднього аналізу.

Діаграма класів є основним будівельним блоком об'єктно-орієнтованого моделювання. Він використовується як для загального концептуального моделювання систематики програми, так і для детального моделювання, що переводить модель в код програми. Діаграми класів також можуть бути використані для моделювання даних. Класи в діаграмі класів представляють як основні елементи, що взаємодіють в додатку, так і класи, які потрібно запрограмувати.[6]

Діаграму класів зображено на Рисунку 3.2. Діаграму було розділено із використанням патерну Model-View-Controller.

Класи зображені на діаграмі:

- Model:
  - User – містить інформацію про користувача та методи її обробки;
  - Shop – містить інформацію про заклад харчування та методи її обробки;
  - Category – містить інформацію про категорію закладу харчування та методи обробки;
  - Dish – містить інформацію про блюдо або товар харчування та методи для її обробки;
  - DishType – містить інформацію про тип блюда або товару харчування та методи її обробки;
  - Order – містить інформацію про замовлення користувача та методи її обробки;
  - Employee – містить дані про працівника доставки та методи обробки даних;
  - Dao – інтерфейс взаємодії з базою даних на класовому рівні, який містить в собі методи для знаходження та керування інформацією в базі даних;
  - OracleDao – клас, що імплементує інтерфес Dao для Oracle Data Base;
  - Analyzer – інтерфейс, що містить методи для роботи алгоритму аналізу історичних даних та прогнозування замовлень користувача;
  - AnalyzerImpl – клас, що реалізує інтерфес Analyzer для алгоритму, що описаний в пункті 3.1;
- Controller:



- ServerController – приймає запити та оброблює запити від клієнтів, використовуючи методи моделі та відправляє користувачу результат на веб-інтерфейс;
- View:
  - ClientView – використовується для показу інформації користувачу.

Для створення інформаційної технології адаптивної системи були створені діаграми класів та варіантів використання. Діаграми були створені за допомогою онлайн ресурсу [draw.io](https://draw.io).

# Model

# Controller

# View

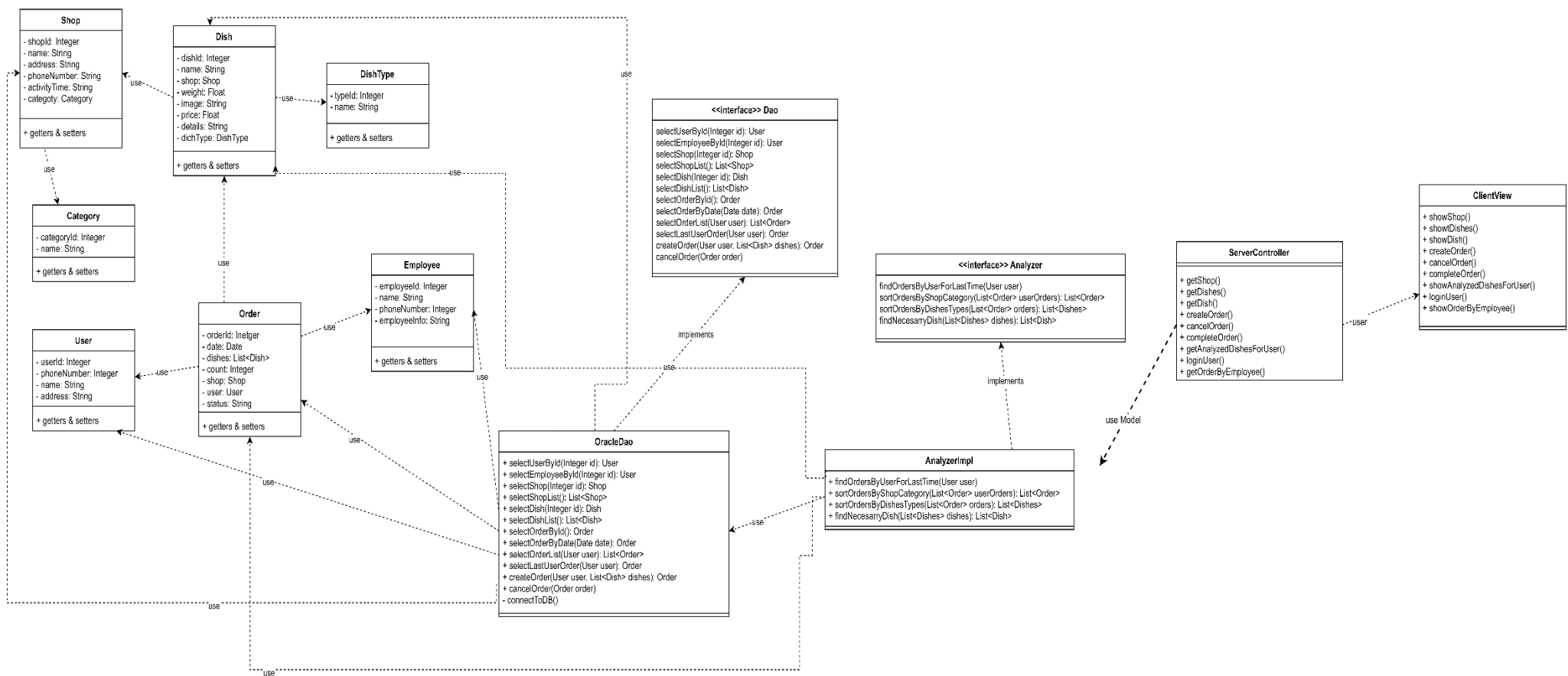


Рисунок 3.2 – Class Diagram

## 3.2 Алгоритм аналізу

Однією з основних переваг створюваної інформаційної системи є аналіз історичних даних попередніх замовлень користувачів та прогнозування. Подібний функціонал все більш набуває популярності серед багатьох сервісів онлай-замовлень їжі та онлайн-магазинів, тому що завдяки такому функціоналу можна швидко підвищити кількість онлайн-продажів та замовлень. А на сьогоднішній день підвищення кількості онлайн-замовлень є актуальною темою для ресторанного та харчового бізнесу.

Одним із методів аналізу великих об'ємів даних є кластерний аналіз. Розглянемо його детальніше та визначемо чи зможемо ним скористатися.

Кластерний аналіз (від англ. cluster - потік, пучок, гроно) - це один із способів класифікації об'єктів. Він дозволяє розглядати досить великий обсяг інформації, стискаючи його та роблячи компактними та простими для розуміння. Головне призначення кластерного аналізу – розбиття множини досліджуваних об'єктів та ознак на однорідні у відповідному розумінні групи або кластери. Це означає, що вирішується завдання класифікації даних та виявлення відповідної структури у ній. Велика перевага кластерного аналізу в тому, що він дозволяє робити розбиття об'єктів не за одним параметром, а за цілим набором ознак. Крім того, кластерний аналіз на відміну більшості математико-статистичних методів не накладає жодних обмежень на вигляд об'єктів, що розглядаються, і дозволяє розглядати множину вхідних даних практично довільної природи. Тобто це метод аналізу може використовуватися для аналізу великого об'єму даних онлайн-замовлень.

Розв'язанням задачі кластерного аналізу є розбиття, що задовольняють деякому критерію оптимальності. Цей критерій може бути деякий функціонал,

що виражає рівні бажаності різних розбиття та угруповань, що називають цільовою функцією.

Метод кластерного аналізу даних є складним для розробки та не підходять для бізнес-логіки створюваною інформаційної онлайн системи. Але ми можна взяти його за основу та створити власний алгоритм, який буде простим для розуміння та програмування та буде актуальним для області створення інформаційної системи.

Оскільки інформаційна система буде працювати для онлайн-замовлення харчових продуктів, то вона обов'язково має зберігати в базі даних таку інформацію:

- заклад харчування;
- категорія закладу(кафе, ресторан, супермаркет, піцерія);
- меню закладу;
- тип страви або продукту;
- дані користувачів;
- дані про всі замовлення.

Для більш ефективної роботи нашого алгоритму скористаймося відомим принципом програмування «розділяй та пануй» та розділемо весь алгоритм на невеликі підзадчі. Для здійснення аналізу виберемо такі критерії: категорія закладу та тип страви або продукту.

Для обробки, аналізу попередніх замовлень та прогнозування наступних створимо API, яке буде приймати на вхід дані користувача, який вже робив кілька замовлень, знаходити інформацію про попередні замовлення за допомогою запита в базу даних. Після чого API буде аналізувати з яких категорій закладів користувач частіше за все робив замовлення, потім вибере типи страв та продуктів, які частіше за все замовляв користувач, та створить

список. На основі створеного списку типів товарі API може обрати будь-який товар та запропонувати користувачу замовити його.

Алгоритм зображений на Рисунку 3.1.2

Алгоритм аналізу та прогнозування

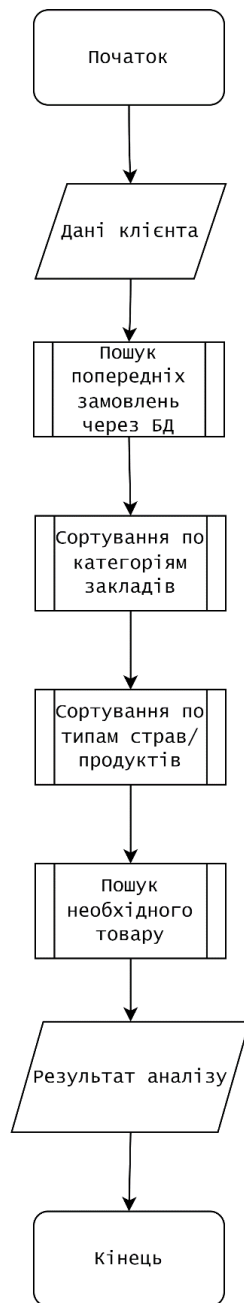


Рисунок 3.1.2 – Алгоритм аналізу та прогнозування замовлень

## 4 Огляд технологій реалізації проекту

Під час проектування було обрано тип для створення інформаційної технології проектування адаптивної системи онлайн замовлень продуктів харчування – веб-додаток. Основною перевагою веб-додатку є незалежність від операційної системи, це дає можливість, при необхідності внесення змін у додаток, не перероблювати його для багатьох операційних систем, а буде достатнім внесення змін лише один раз.

Архітектурна модель проекту буде розділена на серверну та клієнтську частини, тому що такий вид моделі дозволить забезпечити швидкий обмін інформацією між користувачами та додатком. За допомогою цього підходу в результаті матимемо легкокеровану, структуровану систему, до якої буде легко вносити зміни з мінімальними регресійними наслідками.

Для створення інформаційної технології проектування адаптивної системи доцільно застосувати патерн проектування MVC(Model-View-Controller).

Патерн проектування – це загальне рішення певної проблеми в дизайні архітектури. Патерн представляє собою не конкретний код, а загальну концепцію або приклад рішення, що може бути налаштоване під потреби конкретної програми.[4]

Патерн проектування Model View Controller(MVC) – шаблон проектування, його основна ідея полягає в розділенні роботи проекту на окремі компоненти – Модель, Контролер та Представлення, завдяки такому поділу подальші модифікації компонентів можуть бути здійснені незалежно, зміни можуть бути внесені тільки до одного з компонентів.[5]

Модель втілює бізнес логіку проекту, відповідає за збереження та обробку даних, є незалежною від Контролера та Представлення. Контролер

отримує запити від користувача, оброблює їх, використовуючи класи, в яких реалізовано бізнес логіку Моделі, надсилає відповіді на запити користувача. Представлення втілює інтерфейс користувача, який буде віддавати Контролеру інформацію про дії користувача, буде відображати користувачу відповіді Контролера на запити.

Отже, створимо інформаційну технологію проектування адаптивною системою онлайн-замовлень на основі патерну MVC. Це дозволить розділити проєкт інформаційної технології на три компоненти, які будуть мати свій окремий функціонал. Також, завдяки використанню даного патерну дані відображення та їх обробка будуть відокремленими та незалежними один від одного.

Беручи за основу раніше спроектовану клієнт-серверну архітектуру оберемо технології для подальшої розробки.

Розділемо технології, які потрібно оглянути для реалізації проєкту, на три частини:

- мова програмування;
- провайдера бази даних;
- фреймворки, що можна використати для швидкого процесу розробки.

## 4.1 Аналіз мов програмування

Вибір мови програмування – один із важливих аспектів аналізу під час проектування. Мова програмування – це штучностворена мова, яка використовується для надання команд комп'ютеру. Це система для реалізації та запуску алгоритмів та структур даних, які створюються для роботи програмної системи. Вибір мови програмування впливає на основні критерії роботи стороюваної інформаційної технології, а саме на швидкість, гнучкість до модифікацій, розширення системи з мінімальними регресійними наслідками.

Для створення інформаційної технології проектування адаптивної системи онлайн-замовлень продуктів харчування було вирішено реалізувати систему у вигляді веб-додатку. Розглянемо топ мов веб програмування у 2021 році[7]:

1. JavaScript.
2. Python.
3. Java.
4. PHP.

Докладніше розглянемо вищеописані мови.

JavaScript – одна з найпопулярніших мов програмування для розробки у всьому світі та одна з основних технологій, що лежать в основі всесвітньої мережі. JavaScript – це «інтерфейсна мова», призначена для розробки інтерактивних інтерфейсних додатків.

JavaScript має гнучкий синтаксис, зручний для початківців та працює у всіх основних браузерах. Крім того, відмінні фреймворки та бібліотеки, React.JS, Node.JS, AngularJS і т. д., містять JavaScript за лаштунками. Однак серед функцій, завдяки яким JavaScript отримує ці нагороди, слід зазначити, що він може працювати як стороні сервера, так і стороні клієнта. Як? Node.JS



- це середовище виконання на основі JavaScript, при використанні разом з JavaScript робить одну мову програмування достатньою для серверних і клієнтських сценаріїв.

Тому більшість розробників віддають перевагу цій мові іншим, наприклад, для веб-серверів, ігор, програм для розумного годинника та програмного забезпечення. Крім того, такі веб-сайти, як Amazon, Facebook, YouTube, Google мають у своєму інтерфейсі JavaScript.

Python – популярна мова веб розробки. В даний час Python використовується для веб-розробки, обробки даних, написання сценаріїв та розробки масштабованих веб-додатків на основі її та машинного навчання.

Python - універсальна, швидка і одна з найпростіших мов для вивчення програмування. Вона проста у використанні та містить безліч корисних інструментів, фреймворків та бібліотек; Django, Flask і т. д. Крім того, ще однією перевагою є велика підтримка спільноти.

Python може працювати як Front-End, так і для Back-end розробки. Окрім того є серверною мовою, враховуючи її читабельність, а також та бібліотеки та фреймворки. Крім того, Python відомий своїми можливостями динамічної типізації, що дозволяють досить легко створювати великі та складні веб-сайти (динамічні). Такі сайти як Facebook, YouTube, Instagram, Netflix, Dropbox, Mozilla, також частково використовують Python.

Java — це високорівнева, заснована на класах, об'єктно-орієнтована мова програмування, яка розроблена так, щоб мати якомога менше залежностей реалізації. Це мова програмування загального призначення, призначена для того, щоб програмісти могли писати один раз, запускати де завгодно. Java є однією з самих стабільних мов програмування, оскільки вона тримається в верхній частині рейтингу мов програмування більш ніж 20 років.

Основними перевагами мови є кросплатформенність, багатопотоковість та динамічний розподіл пам'яті.

Java є одночасно і мовою програмування, і платформою. Програмуючи на Java, фаши вихідного коду з розширенням `.java`, які пише розробник, компілюються за допомогою компілятора `javac` в файли з розширенням `.class` в так звани байт-код. Після цього Java інтерпретатор JIT(Just in Time) перетворює байт-код в машинний код в залежності від платформи – таким чином, релізується принцип кросплатформенності. А оскільки JVM доступна для всіх існуючих операційних систем, то програмісти мають можливість написати код лише один раз і він буде успішно працювати на різних платформах. Для розробки на Java достань встановити стандартний пакет JDK.

Кросплатформенність не лише одна з переваг мови Java, також ця мова автоматично керує пам'ятю через роботу віртуальною джава машини, має можливість багатопотокового виконання задач одночасно, записуючи багато паралельних процесів.

Веб-платформи, що використовують Java: `linkedin.com`, `ebay.com`, `aliexpress.com`, `aws.amazon.com`, `bitbucket.org` [7].

Грунтуючись на вищеописаних перевагах на недоліках мов програмування, для розробки інформаційної технології проєктування адаптивної системи онлайн-замовлень продуктів харчування є доцільним обрати мову програмування Java. Ця мова широко використовується та зарекомендувала себе стабільною протягом більш ніж 20 років. Також ця мова має зручні фреймворки та API, які можна використати під час розробки.

## **4.2 Вибір фреймворку для розробки веб-додатку**

Сучасне веб-програмування майже неможливе без використання фремворків. Фреймворком називають платформу для розробки програмних

додатків. Він забезпечує фундаментальну основу, використовуючи яку, розробники програмного забезпечення будують програми для певної платформи. Фреймворк може містити вже готові до використання класи та методи, які можуть бути використані для того щоб кожен раз не розробляти великий об'єм базового функціоналу.

Оскільки, в процесі створення архітектури біло вирішено взяти за основу паттерн Model-View-Controller, то для розробки інформаційної технології доцільно використати сучасний фреймворк Spring MVC.

Spring MVC — це фреймворк мови програмування Java, який використовується для створення веб-додатків. Він відповідає шаблону проектування Model-View-Controller. Він реалізує всі основні функції основної пружинної структури, як-от інверсія керування, ін'єкція залежності. Інфраструктура Spring model-view-controller (MVC) розроблена на основі DispatcherServlet, який надсилає запити обробникам з налаштовуваними зіставленнями обробників, роздільною здатністю представлення, мовним стандартом і роздільною здатністю теми, а також підтримкою завантаження файлів. Обробник за замовчуванням заснований на анотаціях @Controller та @RequestMapping, пропонуючи широкий спектр методів гнучких обробки.

У Spring Web MVC можна використовувати будь-який об'єкт як об'єкт команди або підтримки форми; вам не потрібно реалізовувати специфічний для платформи інтерфейс чи базовий клас. Зв'язування даних Spring дуже гнучке: наприклад, він розглядає невідповідність типів як помилки перевірки, які можна оцінити додатком, а чи не як системні помилки. Таким чином, вам не потрібно дублювати властивості бізнес-об'єктів у вигляді простих, нетипізованих рядків в об'єктах форми, просто для обробки неприпустимих відправлень або для правильного перетворення рядків. Натомість часто краще виконувати пряму прив'язку до бізнес-об'єктів.

Дозвіл огляду Spring надзвичайно гнучкий. Контролер зазвичай відповідає за підготовку карти моделі з даними та вибір імені подання, але він також може писати безпосередньо в потік відповіді та виконувати запит. Дозвіл імені представлення легко налаштовується за допомогою розширення файлу або узгодження типу вмісту заголовка Ассерт, за допомогою імен компонентів, файлу властивостей або навіть реалізації ViewResolver.[12]

Spring MVC фреймворк може бути використаний для створення інформаційної технології проектування адаптивної системи онлайн-замовлень продуктів харчування. Даний фреймворк створений на основі кросплатформенної мови програмування Java, також за його допомогою можна логічно розділити весь проєкт різні за призначенням частини, в результаті матимемо проєкт, в який буде легко вносити зміни для подальшого розширення системи. За допомогою цього фреймворку зможемо розробити систему на основі клієнт-серверної архітектурної моделі. Також Spring MVC буде створювати більшу частину об'єктів у системі та створить між ним необхідні зв'язки. Отже, фреймворк створить каркас онлайн-системи і залишає нам лише створення та налаштування нашої функціональності.

### **4.3 Вибір системи управління базаю даних**

Інформаційна технологія проектування адаптивної системи онлайн-замовлень продуктів харчування буде працювати з великою кількістю даних закладів харчування, створення замовлень, аналізом даних попередніх замовлень. Для збереження та обробки такої кількості інформації будемо використовувати базу даних.

Базовою даних називають, пов'язаний між собою за певною ознакою чи властивістю, набір даних, який зазвичай зберігається в електронному вигляді в комп'ютерній системі. Основним призначенням баз даних є управління

великими об'ємами інформації шляхом збереження, отримання та керування даними. Зазвичай управління базою даних контролюється за допомогою системи управління базами даних(СУБД).

Однією з головних переваг використання баз даних є швидкість внесення, оновлення та отримання необхідної інформації. У роботі баз даних використовуються спеціальні алгоритми, завдяки яким знаходження даних відбувається легко за декілька секунд. Бази даних, які використовуються для збереження інформації сайтів, зазвичай мають вигляд таблиць пов'язаних між собою. База даних зберігає всю необхідну та корисну інформацію для функціонування сайту.[13]

Для керування БД, що зберігає дані в таблицях використовують мову структурованих запитів SQL.

SQL (Structured Query Language) — декларативна мова структурованих запитів, комп'ютерна мова для збереження та керування даними, що зберігаються в реляційній БД. [14]

База даних зазвичай потребує комплексного програмного забезпечення, яке називають системою управління базою даних(СУБД). СУБД працює як інтерфейс між базою даних та її користувача або програмами та дозволяє користувачам діставати, обовлювати та керувати організацією та оптимізацією даних. СУБД також полегшує контроль над базою даних, дозволяючи виконувати різноманітні адміністративні операції, такі як моніторинг продуктивності, налаштування, резервне копіювання та відновлення даних.[15]

Сучасна СУБД Oracle широко використовується для управління реляційними базами даних. Ця система дає можливість створити структури бази даних ефективним способом, за допомогою створення ієрархічної

структури, оскільки СУБД Oracle має можливість виконання ієрархічних запитів.

Базу даних Oracle можна підключити до багатьох популярних мов програмування, а також для середовищ розробки, включаючи мову програмування Java. Також існують безкоштовні інструменти розробки від компанії Oracle, наприклад SQL Developer. Oracle Database добре підходить як для маленьких та і для великих користувачів.[16]

Отже, СУБД Oracle підходить для інформаційної технології проектування адаптивної системи онлайн-замовлень продуктів харчування на основі Spring MVC фреймворку.

## 5 Розробка інформаційної технології

Процес розробки веб-додатку для інформаційної технології проектування адаптивної системи онлайн-замовлень продуктів харчування розділемо на такі етапи:

- Проектування та створення бази даних.
- Розробка додатку, за допомогою Spring MVC фреймворку.

За допомогою діаграм зобразимо потоки даних в системі, на основі яких будемо створювати концептуальну модель даних.

### 5.1 Створення бази даних

Створювана інформаційна технологія керує даними замовлень клієнтів та працівників. Основні інформаційні процеси системи зобразимо у вигляді Data Flow Diagram(DFD) – діаграми потоків даних.

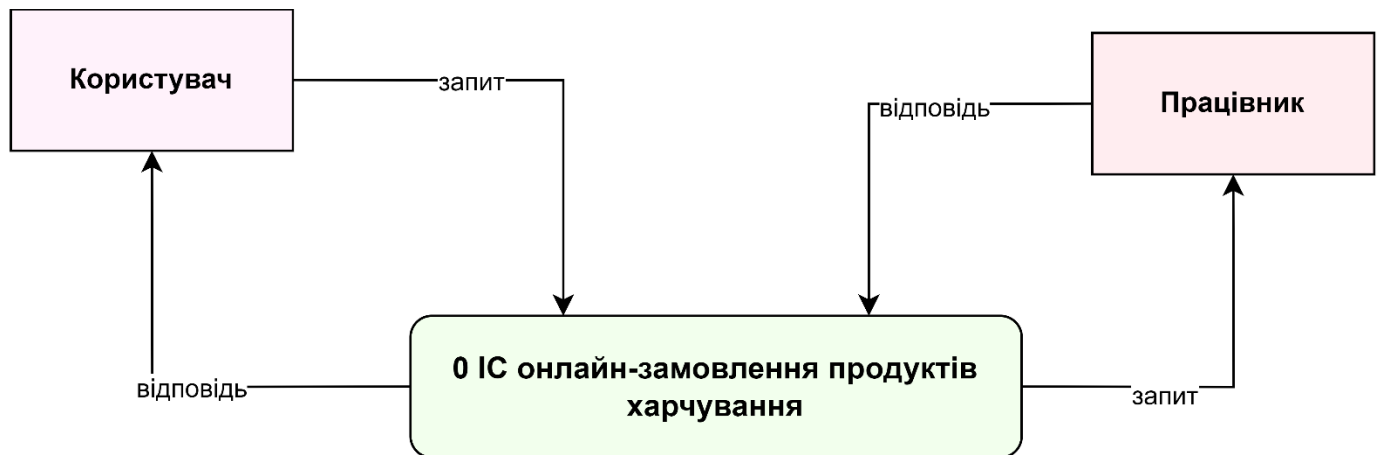


Рисунок 5.1.1 – DFD 0-го рівня ІС для онлайн-замовлення продуктів харчування

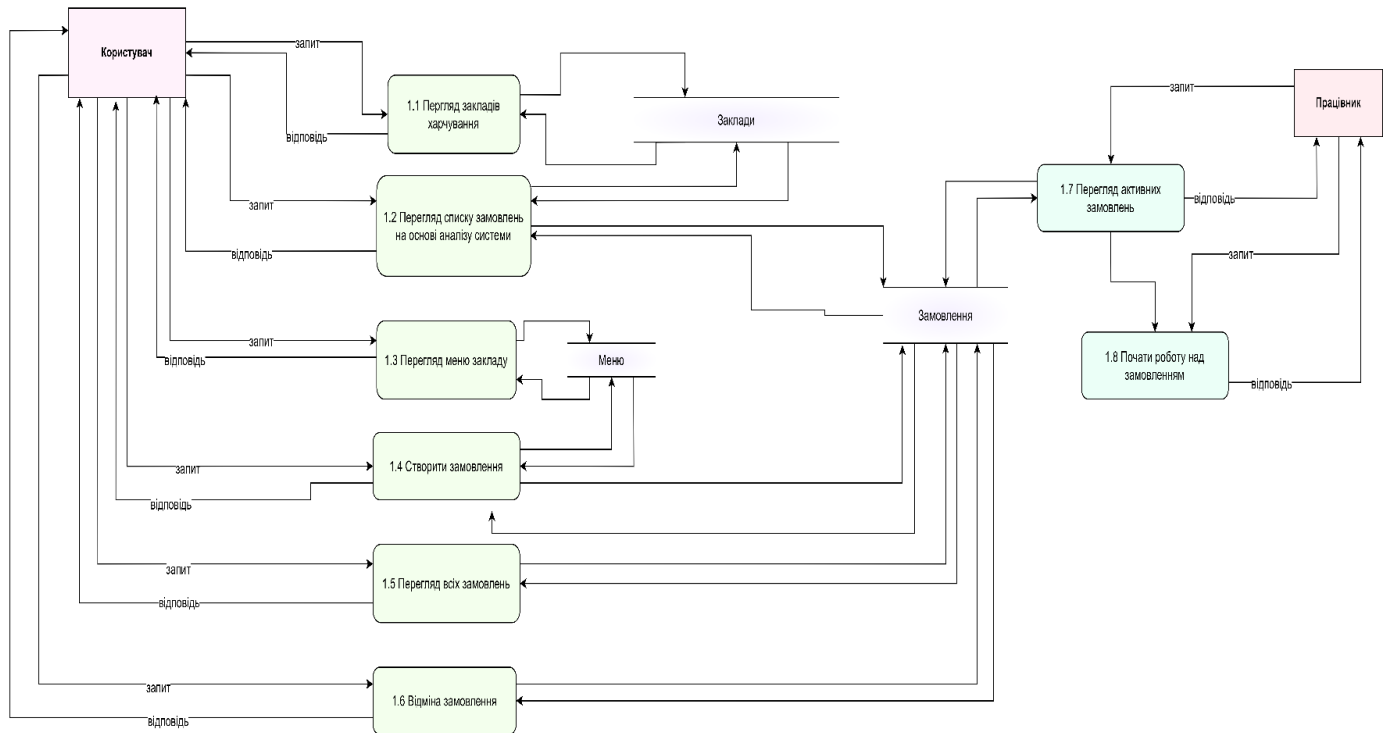


Рисунок 5.1.2 – DFD 1-го рівня ІС для онлайн-замовлення продуктів харчування

Коцептуальна модель має наступні сховища даних: «Заклади», «Меню», «Замовлення». Всі запити від користувача розділені на процеси:

- «Перегляд закладів харчування» - надає користувачу інформацію про всі заклади харчування.
- «Перегляд списку замовлень на основі аналізу системи» - надає користувачу список замовлень на основі роботи алгоритму аналізу історичних даних попередніх замовлень та прогнозування нових.
- «Перегляд меню закладів» - надає користувачу меню відповідного закладу харчування.
- «Створити замовлення» - оброблює запити користувачів на створення замовлень.
- «Перегляд всіх замовлень» - система надає користувачу список всіх його замовлень активних даних момент часу та всіх попередніх.



- «Відміна замовлення» - система відмінює замовлення та переводить його в неактивний статус.

- «Перегляд активних замовлень» - надає працівнику доставки список активних замовлень.

- «Почати роботу на замовленням» - працівник доставки бере замовлення на виконання.

Процеси отримують запити від користувачів та працівників доставки, оброблюють їх, формують та відсилають відповідь.

Отримані зв'язки в системі зобразимо за допомогою ERD.

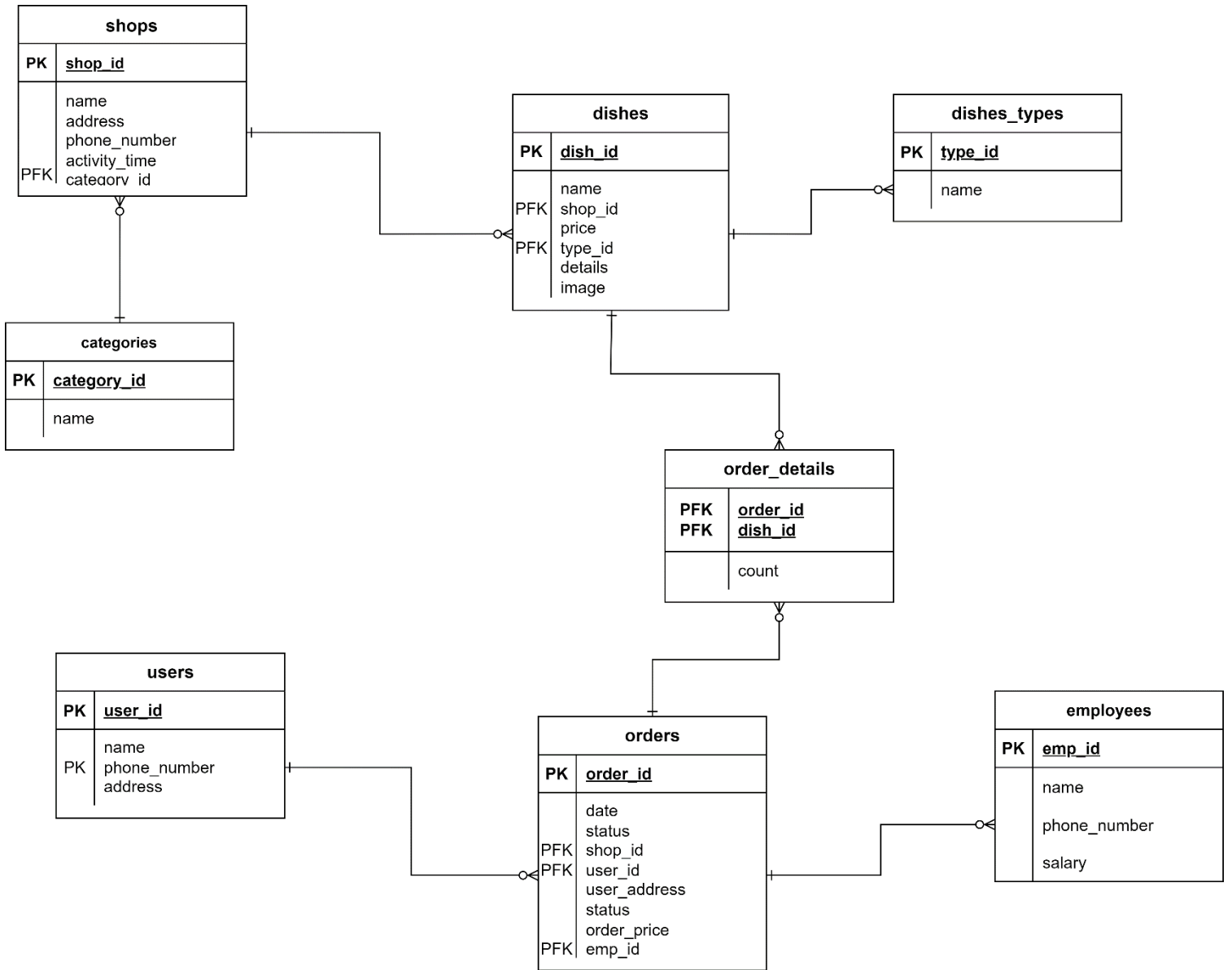


Рисунок 5.1.3 - ERD IC для онлайн-замовлення продуктів харчування

Детальний опис таблиць, атрибутів, їх типів та призначення, а також обмеження зображено у вигляді талиці(Табл. 5.1.1).

Таблиця 5.1.1 – Детальний опис сутностей БД

Таблиця	Поле	Зміст	Ключі	Тип	Обмеження
users	user_id	Ідентифікатор користувача	PK	NUMBER(10)	Не пустий
	name	Ім'я користувача		VARCHAR(20)	Не пустий
	phone_number	Номер телефону	PK	NUMBER(10)	Не пустий
	address	Адреса останнього замовлення		VARCHAR(50)	Не пустий
employees	emp_id	Ідентифікатор працівника доставки	PK	NUMBER(10)	Не пустий
	name	Повне ім'я працівника		VARCHAR(50)	Не пустий
	phone_number	Номер телефону		NUMBER(10)	Не пустий
	salary	Заробітна плата		NUMBER(5)	Не пустий
orders	order_id	Ідентифікатор замовлення	PK	NUMBER(10)	Не пустий
	date	Дата замовлення		DATE	Не пустий
	status	Статус замовлення (Активне, завершене, відмінене)		VARCHAR(10)	Не пустий
	shop_id	Ідентифікатор закладу харчування	PFK	NUMBER(10)	Не пустий
	user_id	Ідентифікатор користувача	PFK	NUMBER(10)	Не пустий
	user_address	Адреса доставки замовлення		VARCHAR(50)	Не пустий
	order_price	Загальна ціна за все замовлення		NUMBER(10)	Не пустий
	emp_id	Ідентифікатор працівника доставки	PFK	NUMBER(10)	Не пустий
order_details	order_id	Ідентифікатор замовлення	PFK	NUMBER(10)	Не пустий
	dish_id	Ідентифікатор блюда	PFK	NUMBER(10)	Не пустий
	count	Кількість блюд		NUMBER(2)	Не пустий
dishes	dish_id	Ідентифікатор блюда	PK	NUMBER(10)	Не пустий
	name	Назва блюда		VARCHAR(20)	Не пустий
	shop_id	Ідентифікатор закладу харчування	PFK	NUMBER(10)	Не пустий

	price	Ціна за одну одинцю		NUMBER(5)	Не пустий
	details	Деталі		VARCHAR(100)	Не пустий
	image	Зображення готового блюда		VARCHAR(15)	Не пустий
dishes_types	type_id	Тип блюда	PK	NUMBER(10)	Не пустий
	name	Назва типу блюда		VARCHAR(20)	Не пустий
shops	shop_id	Ідентифікатор закладу харчування	PK	NUMBER(10)	Не пустий
	name	Назва закладу харчування		VARCHAR(20)	Не пустий
	address	Адреса закладу		VARCHAR(50)	Не пустий
	phone_number	Номер телефону		NUMBER(10)	Не пустий
	activity_time	Робочі часи		VARCHAR(50)	Не пустий
	category_id	Ідентифікатор категорії закладу	PFK	NUMBER(10)	Не пустий
categories	category_id	Ідентифікатор категорії закладу	PK	NUMBER(10)	Не пустий
	name	Назва категорії закладу		VARCHAR(20)	Не пустий

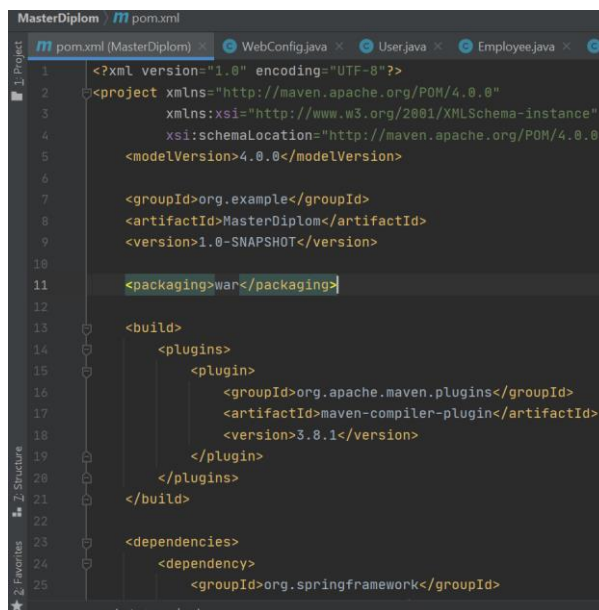
## 5.2 Розробка компонентів додатку

Веб-додаток для інформаційної технології розроблюємо на мові програмування Java та основі Spring MVC фреймворку, оскільки Java є кросплатформенною, а використання Spring MVC фреймворку спростиить процес розробки веб-додатку.

Для автоматизації роботи проєкту, автоматичного підключення сторонніх бібліотек використовуємо систему збірки проєкту Apache Maven.

Для опису програмного проєкту, який потрібно побудувати (*build*), Maven використовує конструкцію відому як Project Object Model (POM), залежності від зовнішніх модулів, компонентів та порядку побудови. Виконання певних, чітко визначених задач — таких, як компіляція коду та пакування відбувається шляхом досягнення задалегідь визначених цілей (*targets*).[17]

Налаштування збірки проєкту знаходяться в файлі `pom.xml`, його фрагмент зображений на рисунку 5.2.1.



```
MasterDiplom pom.xml
pom.xml (MasterDiplom) x WebConfig.java x User.java x Employee.java x S
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>org.example</groupId>
8     <artifactId>MasterDiplom</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11     <packaging>war</packaging>
12
13     <build>
14       <plugins>
15         <plugin>
16           <groupId>org.apache.maven.plugins</groupId>
17           <artifactId>maven-compiler-plugin</artifactId>
18           <version>3.8.1</version>
19         </plugin>
20       </plugins>
21     </build>
22
23     <dependencies>
24       <dependency>
25         <groupId>org.springframework</groupId>
```

Рисунок 5.2.1 – Файл проєктної збірки `pom.xml`

Результатом збірки проекту буде веб-архів типу .war . Такий архів буде містити повністю готовий архів для запуску на веб-сервері.

Під час розробки проекту та для тестових запусків веб-додатку використовуємо веб-контейнер Apache Tomcat, який підтримує реалізацію багатьох Java технологій. Також цей веб-контейнер є простим та легким у використанні та має значну перевагу над іншими веб-контейнерами завдяки мінімальному використанню ресурсів комп'ютера.

Підключення до бази даних здійснено з використанням безкоштовного драйверу ojdbc8. Він реалізує підключає проєкт до бази даних.

З базою даних процюють інтерфейс Dao та його імплементація OracleDao, як зображено на діаграмі класів (Рис. 3.2).

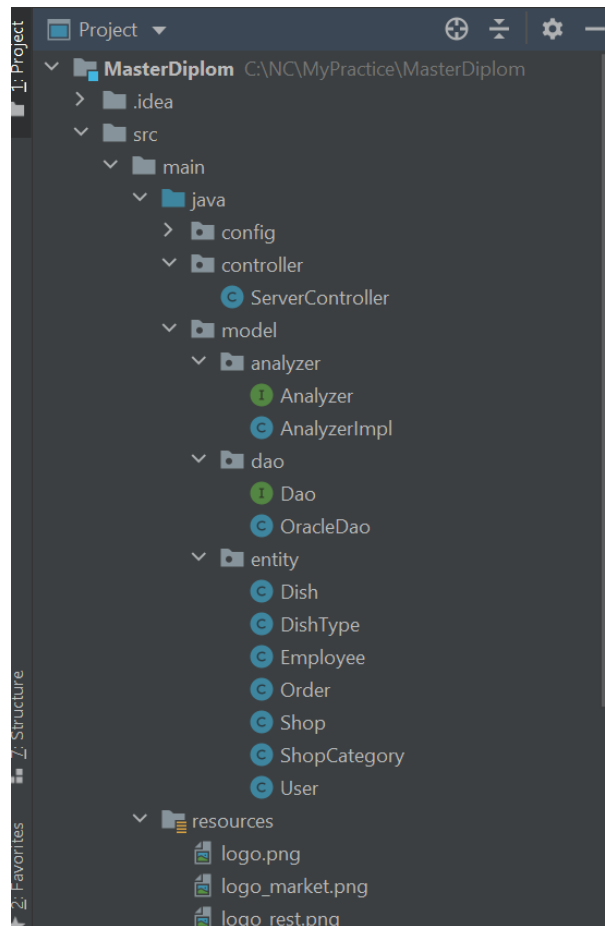


Рисунок 5.2.2 – Розташування файлів та пакетів в каталозі проекту

Кореневий каталог проєкту містить такі директорії:

- `java` – містить пакети структури пакетів, в яких знаходяться `java` класи, на яких працює проєкт.
- `resources` – містить інші проєктні файли, наприклад рисунки.

JSP (JavaServer Pages) - технологія, за допомогою якої можна створювати динамічний та статичний зміст для HTML сторінок.[19]

Також для створення динамічного контенту використовуємо стандартну бібліотеку тегів JSTL (англ. JavaServer Pages Standard Tag Library, JSTL).

Алгоритм аналізу з пункту 3.2 реалізований через інтерфейс `Analyzer` та клас, що його імплементує `AnalyzerImpl`.

### 5.3 Результати реалізації веб-додатку

Для того щоб користувач зміг отримати доступ на перегляд даних веб-додатку інформаційної технології проектування адаптивної системи онлайн-замовлень продуктів харчування створити своє перше замовлення, потрібно натиснути на кнопку «Почати !» (Рис. 5.3.1).

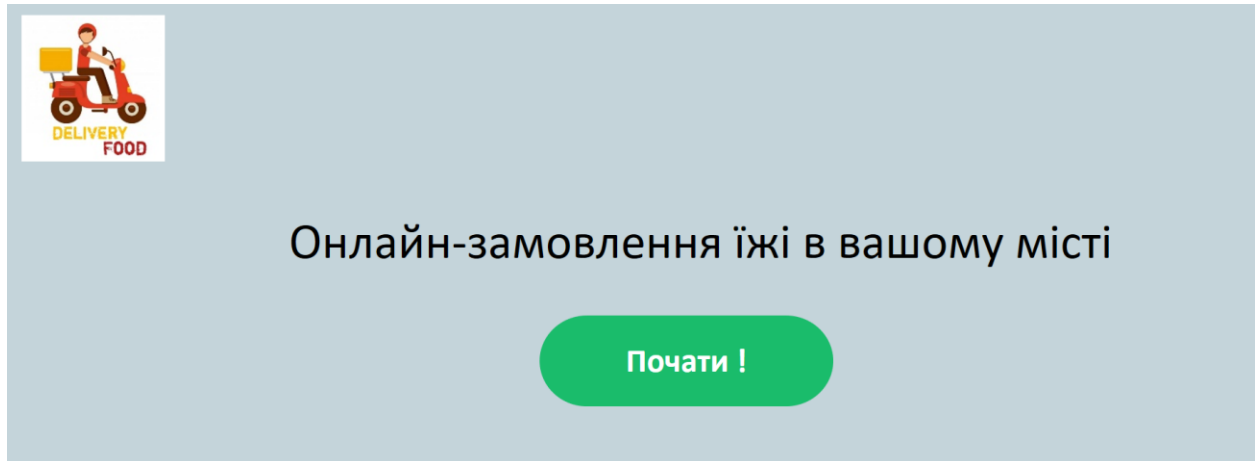



Рисунок 5.3.1 – Сторінка початку роботи

Починаючи роботу з додатком, користувач отримає форму для введення даних, що потрібні для створення замовлень (Рис. 5.3.2).





Для початку будь-ласка введіть ваші дані:

Онлайн місто

Ім'я

Номер телефону


Почати !

Ваша адреса

Підтвердити дані !

Рисунок 5.3.2 – Пуста форма введення початкових даних користувача

Після введення даних користувач може їх підтвердити (Рис. 5.3.3)



Для початку будь-ласка введіть ваші дані:

Онлайн місто

Іван

095 20 21 000

Почати !

вулиця Римського-Корсакова, 2

Підтвердити дані !

Рисунок 5.3.3 – Заповнена форма введення початкових даних

Підтвердивши свої дані у системі, можна обрати категорію закладу харчування (Рис. 5.3.4).



Рисунок 5.3.4 – Сторінка вибору категорії закладу харчування

Після вибору категорії, бачимо список закладів. Для прикладу обрано категорію «Ресторани» (Рис. 5.3.5).

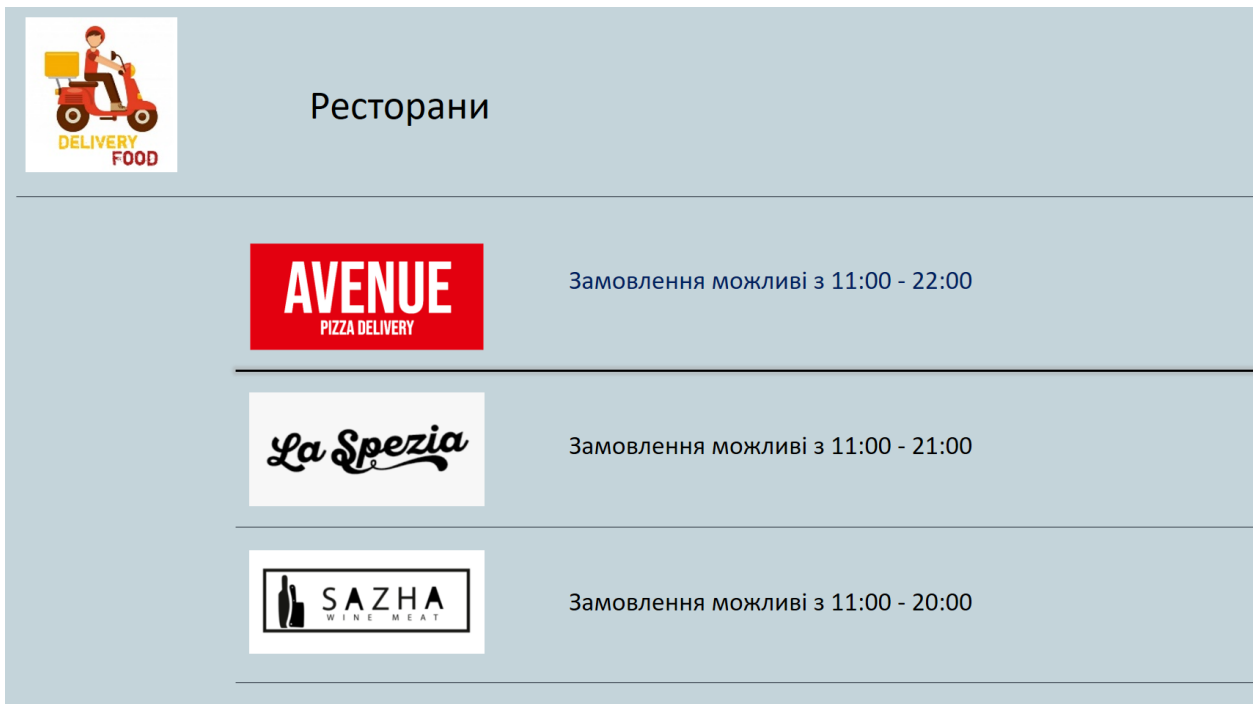


Рисунок 5.3.5 – Список закладів харчування категорії Ресторани

Для тестування оберемо ресторан Avenue (Рис. 5.3.6). Після цього можемо переглянути його меню та обрати блюдо для замовлення.

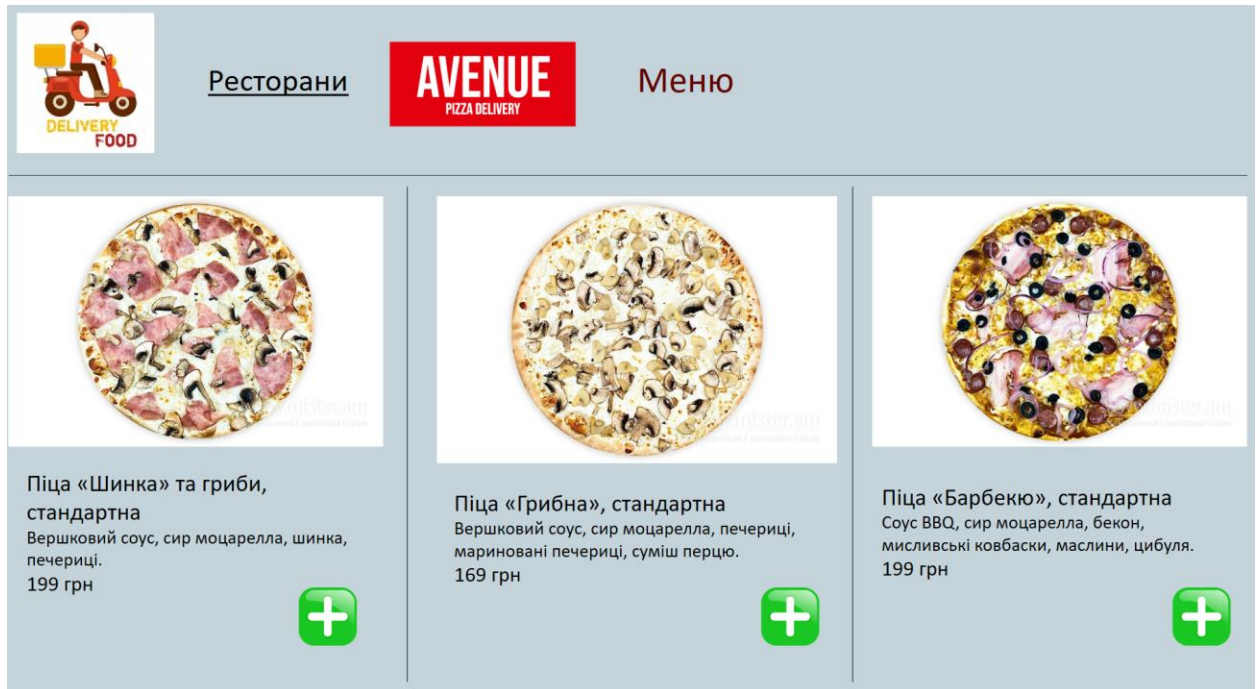


Рисунок 5.3.6 – Меню ресторану Avenue

Обравши блюдо, стандартну піцу, користувач отримає форму підтвердження замовлення з його подробицями(Рис. 5.3.7).

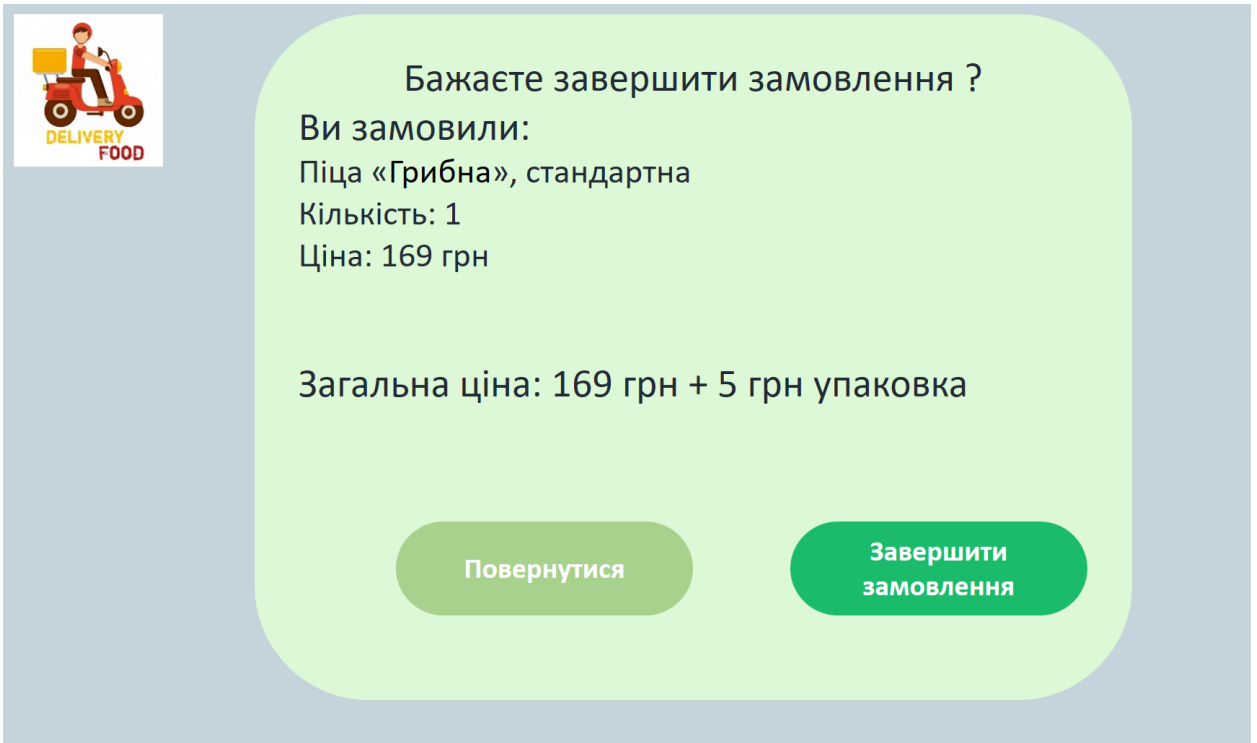


Рисунок 5.3.7 – Сторінка підтвердження замовлення користувача

Підтверджуючи замовлення, за допомогою кнопки «Завершити замовлення» користувачу буде відображене повідомлення про те, що його замовлення було успішно прийняте до обробки(Рис. 5.3.8) та кнопка «Перейти до моїх замовлень», натиснувши на яку, можна перестити до списку всіх замовлень (Рис. 5.3.9).

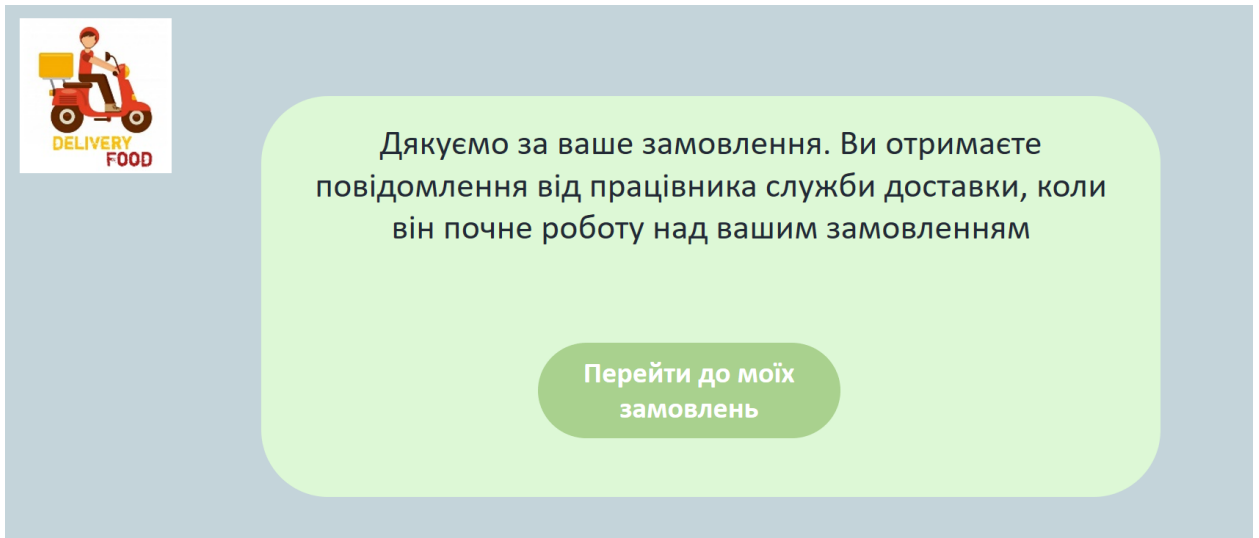


Рисунок 5.3.8 – Повідомлення про іспішно створене замовлення

На сторінці всіх замовлень, можна скасувати активне замовлення за допомогою кнопки (Рис. 5.3.9). Настснувши на кнопку скасування, користувач отримає форму підтвердження(Рис. 5.3.10). Якщо, користувач підтвердить скасування замовлення, то статус замовлення буде змінено на «Скасовано» (Рис. 5.3.11).

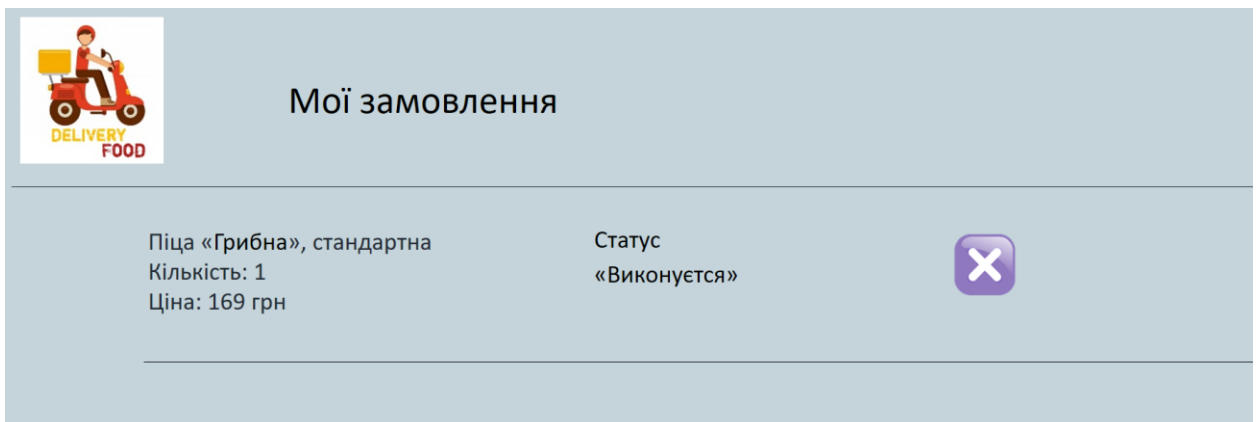


Рисунок 5.3.9 – Сторінка замовлень користувача

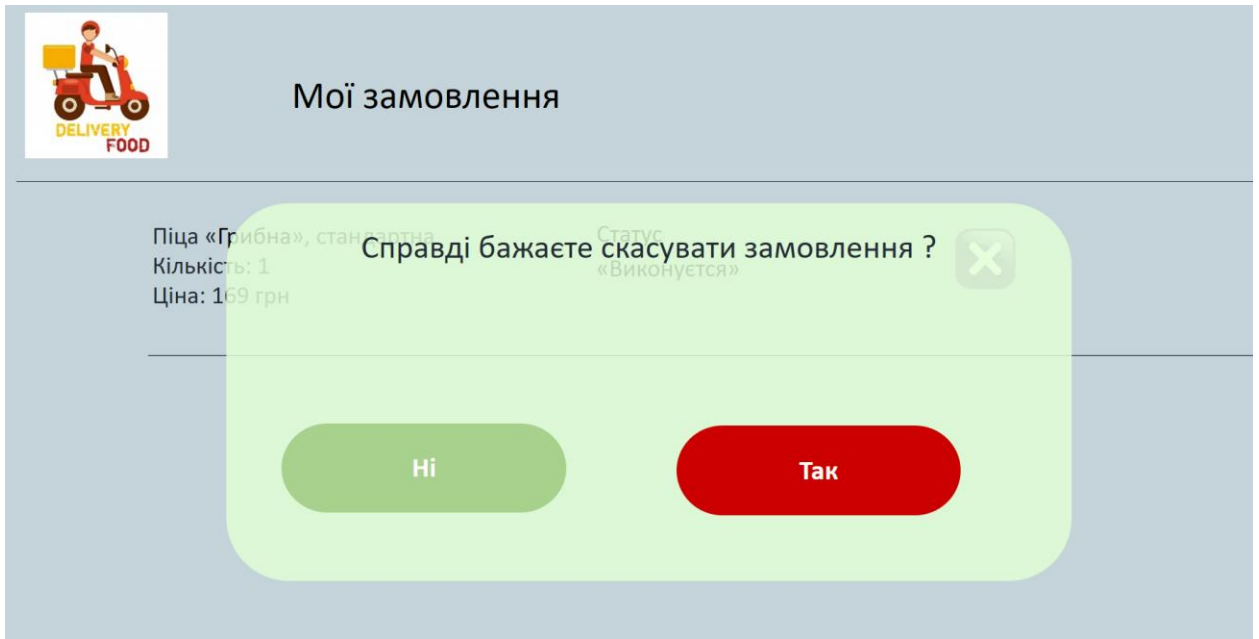


Рисунок 5.3.10 – Форма підтвердження скасування замовлення

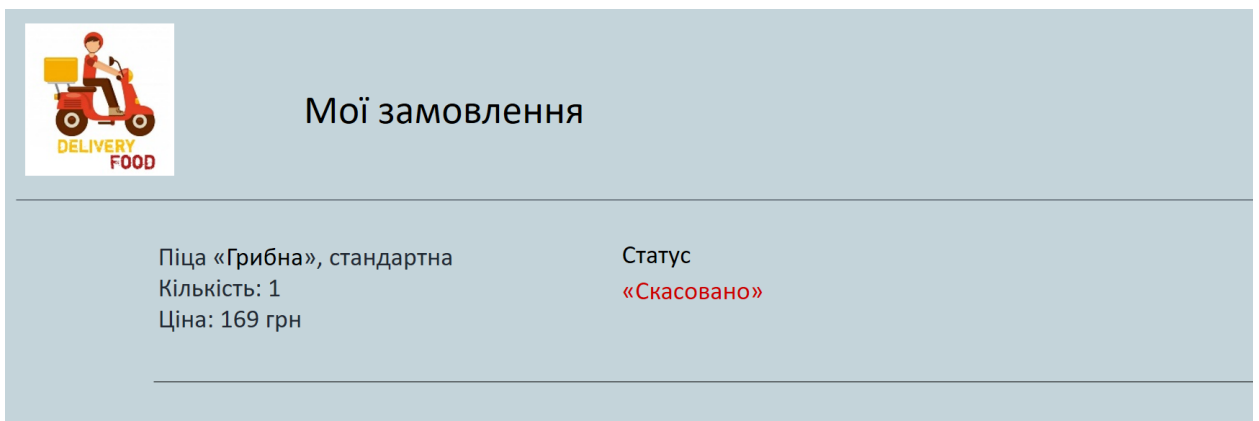


Рисунок 5.3.11 – Сторінка замовлень користувача після скасування замовлення

Після успішного виконання замовлення, його статус буде змінено системою на «Замовлення виконано» (Рис. 5.3.12).



## Мої замовлення

Піца «Грибна», стандартна  
Кількість: 1  
Ціна: 169 грн

Статус  
«Замовлення виконано»

Рисунок 5.3.12 – Сторінка замовлень користувача після успішного виконання замовлення

Наступного разу, коли користувач введе до системи з тими ж вхідними даними, що й минулого разу (Рис. 5.3.2), то веб-додаток запропонує йому результати алгоритму аналізу та відобразить сторінку зі списком можливих замовлень (Рис. 5.3.13).



Доброго дня!

Минулого разу ви замовляли Піца «Грибна», стандартна з закладу ресторан Avenue. Бажаєте замовити ще щось з цього ресторану. Наприклад, вам може сподобатись:



Піца «Барбекю», стандартна  
Соус BBQ, сир моцарелла, бекон,  
мисливські ковбаски, маслини, цибуля.  
199 грн

Ні, обрати інше

Так, хочу замовити

Рисунок 5.3.13 Форма з пропозицією замовлення

Якщо, користувач натисне кнопку «Так, хочу замовити», то система створить нове замовлення на основі запропонованих даних, а користувач отримає повідомлення про успішне створення замовлення(Рис. 5.3.8) та зможе побачити його в списку своїх замовлень(Рис 5.3.14).

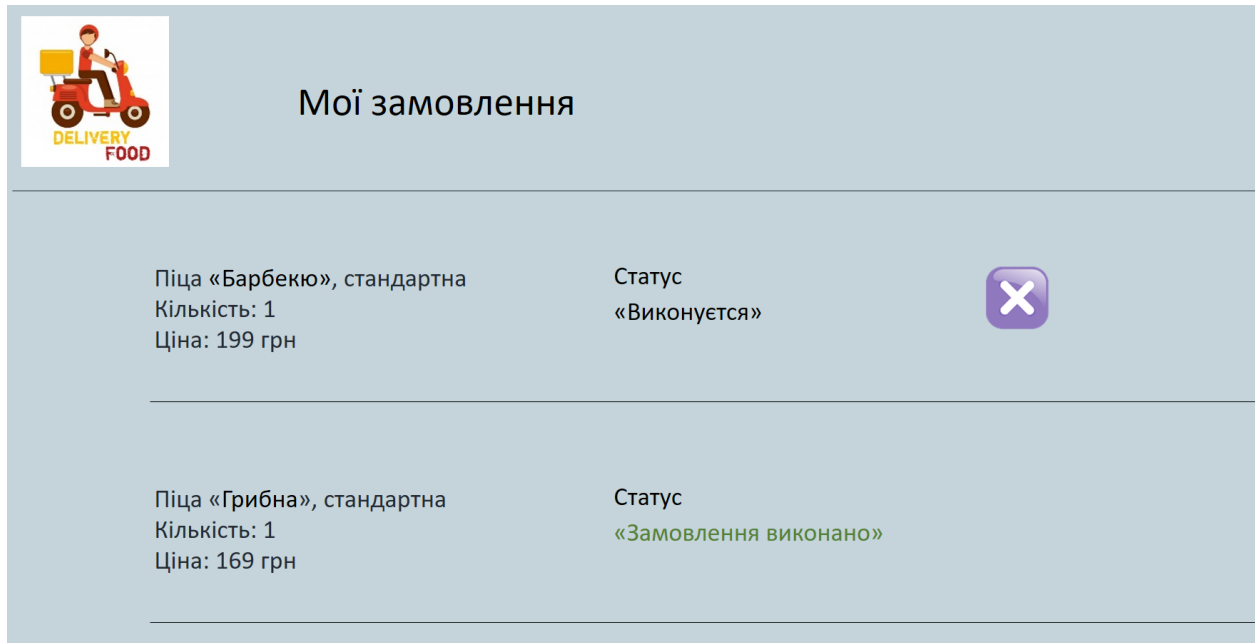


Рисунок 5.3.14 – Список всіх замовлень користувача



## Висновки

Дипломний проєкт інформаційної технології проєктування адаптивної системи онлайн-замовлень продуктів харчування є актуальним для сучасної людини, оскільки останнім часом стала дуже популярною тема замовлення доставки їжі до дому через інтернет.

Створення зручної адаптивної онлайн-системи за допомогою якої користувач може обирати заклад харчування, переглядати меню, створювати та скасовувати замовлення, робити замовлення на основі запропонованих системою варіантів. В цьому полягала основна ідея створюваного проєкту

Подібні системи мають багато переваг для користувачів, оскільки дозволяють користуватися послугами ресторанів або інших закладів харчування, залишаючись вдома.

Першим кроком, під час виконання дипломного проєкту було проаналізовано аналогічні рішення, на основі яких стало зрозуміло яким критеріям має відповідати система онлайн-замовлень, щоб вона могла стати популярною у користувачів. В результаті аналізу було вирішено створити власну інформаційну систему онлайн-замовлень.

Для проєктування інформаційної онлайн-системи обрали клієнт-серверну архітектуру. Дана архітектурна модель забезпечує швидкий обмін інформацією між системою та користувачем. Базуючи проєкт на основі даної архітектурної моделі, в результаті отримали добре структуровану, легку в управлінні систему, до якої легко можемо внести зміни.

Для створення проєкту було обрано веб-додаток як тип програмного забезпечення, оскільки він є максимально зручним та простим для користувачів.

Було спроектовано діаграму варіантів використання та класів. А також розроблено власний алгоритм аналізу та прогнозування замовлень на основі категорій закладів харчування та типів замовлень.

Технологіями розробки було обрано мову програмування Java та Spring MVC фреймворк, оскільки вони є кросплатформенними, веб-орієнтованими та зарекомендували себе як стабільні технології протягом більш ніж 20 останніх років.

В результаті виконання дипломного проєкту маємо веб-додаток для інформаційної технології проєктування адаптивної системи онлайн-замовлень продуктів харчування. За допомогою додатку, користувач може створювати замовлення з різних категорій закладів харчування, скасовувати замовлення за потреби, також створювати замовлення на основі пропозицій, які є результатом розробленого алгоритму аналізу.

## Список використаної літератури

1. Что такое архитектура программного обеспечения [Электронный ресурс] – 2015. – Режим доступа: [ibm.com/developerworks/ru/library/eeles/index.html](http://ibm.com/developerworks/ru/library/eeles/index.html) – Назва з екрану.
2. Fielding, R.T. Dissertation Architectural Styles and the Design of Network-based Software Architectures. – 20 p.
3. Saternos, C. Client-Server Web Apps with JavaScript and Java. – 12 с.
4. Швец, А. Погружение в паттерны проектирования. – 5 с.
5. Tsonev, Kh. React in Patterns. . – 13 p.
6. Head First Design Pattern Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra – O'Reilly Media, 2004. – 694 p.
7. BEST PROGRAMMING LANGUAGES FOR WEB DEVELOPMENT IN 2020 [Электронный ресурс] – 2020. – Режим доступа: [manifera.com/best-programming-languages-for-web-development-in-2020/](http://manifera.com/best-programming-languages-for-web-development-in-2020/) - Назва з екрану.
8. Python [Электронный ресурс] – 2020. – Режим доступа: [uk.wikipedia.org/wiki/Python](http://uk.wikipedia.org/wiki/Python) – Назва з екрану.
9. Н. Terrace, Н. Walk Java in easy steps – p. 15
10. Сервлет [Электронный ресурс] – 2015 –Режим доступу [uk.wikipedia.org/wiki/Сервлет](http://uk.wikipedia.org/wiki/Сервлет) – Назва з екрану.
11. 10 Best Programming Languages to Learn in 2020 [Электронный ресурс] – 2020. – Режим доступа: [hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future](http://hackr.io/blog/best-programming-languages-to-learn-2020-jobs-future) – Назва з екрану.
12. Spring [Электронный ресурс] – 2020. – Режим доступа: [docs.spring.io/spring/docs/3.0.x/spring-framework-reference/html/mvc.html](http://docs.spring.io/spring/docs/3.0.x/spring-framework-reference/html/mvc.html) – Назва з екрану.

13. Що таке база даних? [Електронний ресурс] – 2017. – Режим доступу: [apers.kpi.ua/shco-take-basa-danykh](http://apers.kpi.ua/shco-take-basa-danykh) – Назва з екрану.
14. SQL [Електронний ресурс] – 2020. – Режим доступу: [uk.wikipedia.org/wiki/SQL](http://uk.wikipedia.org/wiki/SQL) – Назва з екрану.
15. Система управління базами даних [Електронний ресурс] – 2006. – Режим доступу: [uk.wikipedia.org/wiki/Система\\_управління\\_базами\\_даних](http://uk.wikipedia.org/wiki/Система_управління_базами_даних) – Назва з екрану.
16. Free Oracle Database for Everyone [Електронний ресурс] – 2018. – Режим доступу: [oracle.com/database/technologies/appdev/xe.html](http://oracle.com/database/technologies/appdev/xe.html) – Назва з екрану.
17. Apache Maven [Електронний ресурс] – 2020. – Режим доступу: [uk.wikipedia.org/wiki/Apache\\_Maven](http://uk.wikipedia.org/wiki/Apache_Maven) – Назва з екрану.
18. Apache Tomcat Maven [Електронний ресурс] – 2020. – Режим доступу: [tomcat.apache.org/](http://tomcat.apache.org/) – Назва з екрану.
19. JSP [Електронний ресурс] – 2020. – Режим доступу: [uk.wikipedia.org/wiki/JSP](http://uk.wikipedia.org/wiki/JSP) – Назва з екрану.
20. JSTL [Електронний ресурс] – 2020. – Режим доступу: [en.wikipedia.org/wiki/JavaServer\\_Pages\\_Standard\\_Tag\\_Library](http://en.wikipedia.org/wiki/JavaServer_Pages_Standard_Tag_Library) – Назва з екрану.