

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
Харківський національний університет імені В.Н.Каразіна  
Факультет математики і інформатики  
Кафедра теоретичної та прикладної інформатики

**Кваліфікаційна робота**  
**бакалавр**

на тему: “Проектування та Розробка застосунку для користування послугами  
foodbank”

Виконала: студентка 4 курсу, групи  
МФ-41  
спеціальність 122 «Комп’ютерні науки»  
освітньо-професійна програма  
«Інформатика»  
Пугачова Анна Романівна

Керівник: ст. викладач Бережна Наталія Ігорівна

Рецензент \_\_\_\_\_  
(прізвище та ініціали)

Харків – 2023 року

## **ЗМІСТ**

1. ВСТУП
  - 1.1. Мета проєкту
  - 1.2. Постановка задачі
  - 1.3. Пояснення актуальності теми
  - 1.4. Аналіз існуючих додатків
  - 1.5. Огляд та обґрунтування використаних технологій
2. ОСНОВНА ЧАСТИНА
  - 2.1. Розробка архітектури додатку
    - 2.1.1. Опис функціональності системи
    - 2.1.2. Моделювання бази даних
    - 2.1.3. Опис структури класів та зв'язків
  - 2.2. Розробка UI/UX дизайну
    - 2.2.1. Вимоги до інтерфейсу користувача
    - 2.2.2. Процес проєктування користувацького досвіду (UX)
    - 2.2.3. Процес проєктування інтерфейсу користувача (UI)
  - 2.3. Майбутні перспективи розвитку проєкту
3. ВИСНОВКИ
4. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

## 5. ВСТУП

Україна переживає важкі часи: люди втрачають дохід та помешкання. Щодня запити на гуманітарну допомогу ростуть, але деякі не мають можливості бути реалізовані через брак сервісів. Додаток food bank може бути корисним для того, щоб допомогти вимушеним переселенцям та людям з низьким доходом, які мають проблеми зі здобуттям достатньої кількості харчів.

Такий додаток також може бути корисним для користувачів, які можуть відвідувати різні місця для отримання благодійної допомоги в їжі. За допомогою додатку, вони можуть знайти найближчу до них точку збору їжі та оформити замовлення на продукти, що задовільняють їх потреби та уподобання.

Магазинам та підприємствам, які займаються виготовленням та розподілом харчів, цей додаток також буде корисним. Вони можуть надавати невикористані та надлишкові продукти у місця видачі, таким чином зменшуючі кількість відходів та звільняючи склади від надлишкових товарів.

Знизити кількість відходів в Україні є однією з цілей створення додатку, бо продукти матимуть другий шанс бути використаними до закінчення терміну придатності.

Отже, розробка додатку food bank може бути корисною для багатьох людей та може допомогти зменшити проблему голоду серед населення та загальну проблему забруднення навколишнього середовища.

## 1.1. Мета Проєкту

Метою свого проєкту я бачу:

- Спростити роботу центрів гуманітарної допомоги: звільнити працівників від реєстрації відвідувачів у режимі офлайн
- Налогодити зручний процес реєстрації та консультування онлайн
- Надати користувачам доступний та легкий спосіб покрити свої споживчі потреби
- Запропонувати нову, не доступну до цього, модель пошуку та отримання необхідних продуктів

## 1.2. Постановка задачі

Описання задачі:

Необхідно розробити веб-додаток або його прототип для користування послугами банків їжі, що надасть користувачам зручний інструментарій для реєстрації та формування замовлень, а також будь-яку потрібну додаткову інформацію.

Основні функції, що реалізовані у додатку:

1. Створення акаунту замовника
2. Пошук необхідної інформації: адреси пунктів видачі та години їх роботи, інформація про партнерів проєкту, доступ до соціальних мереж для консультування, інформація стосовно роботи сервісу та контактна інформація
3. Налаштування особистих потреб
  - 3.1. Зазначення алергій
  - 3.2. Зазначення дієтичних обмежень
4. Можливість сформування замовлення
  - 4.1. Доступ до перегляду товарів
    - 4.1.1. Інформація про їх склад
  - 4.2. Можливість видалення товарів
  - 4.3. Отримання номеру замовлення, для легкого одержання пакету продуктів на місці

### 1. 3. Пояснення актуальності теми

Актуальність створення нового додатку food bank в Україні, я вважаю, базується на декількох факторах.

По-перше, Україна є країною з високим рівнем бідності та соціальних проблем, що зумовлює значну кількість людей, які потребують допомоги з отриманням продуктів харчування. Згідно з даними Міжнародного Валютного Фонду, у 2020 році близько 25% населення України жило за межею бідності.

По-друге, створення нового додатку food bank може стати важливим інструментом у боротьбі з втратою їжі, яка становить значну проблему в Україні. За даними Міністерства розвитку економіки, торгівлі та сільського господарства України, у 2019 році втрати їжі в країні становили близько 9 мільйонів тонн, що складало 14% загального обсягу виробництва продуктів харчування. Україна входить в 10-ку країн, де виробляють найбільшу кількість органічних відходів

По-третє, у зв'язку з пандемією COVID-19 зросла потреба у дистанційних сервісах, у тому числі і у доступі до продуктів харчування. Створення нового додатку може стати важливим кроком у напрямку підтримки потребуючих у цей складний час.

І на останок, у результаті вторгнення РФ, ще більше людей втратили роботу та домівку і, скоріш за все, ці числа будуть лише збільшуватися. На даний момент неможливо оцінити загальний вплив війни на життя населення. Так на складах залишається надлишкова їжа, що не може бути доставлена у магазини через проблеми транспортної розв'язки, бомбардування або окупації.

#### 1.4. Аналіз Існуючих Додатків

В Україні існує декілька подібних додатків, натомість вони не виконують функцію, яку я хочу реалізувати.

Так існує:

- Charitable Organization "Kyiv City Charitable Foundation" Food Bank ("КМБФ "ФУДБЕНК")

<http://www.foodbank.com.ua/index.php?lang=2>

- UKRAINIAN FOOD FOUNDATION

Ці додатки збирають матеріальні пожертви та мають загальний опис своїх послуг. Тим не менш не надають можливості користувачам знайти типи продуктів, що доступні, а також не мають інструментів для реєстрації людей у чергу онлайн

Найбільшу схожість я знайшла у сервісі “Тарілка”, що відкрився у Львові в середині 2021го року та працює лише на території міста. Організація “рятує їжу”, термін придатності якої спливає, і надає доступ до неї людям. Але у гугл-формі, за допомогою якої надають доступ до реєстрації, зазначені обмежені категорії людей, що можуть звернутися по допомогу, а також після заповнення форми треба чекати на зворотній зв’язок.

Таким чином лише маленький відсоток людей мають доступ до подібних сервісів і звертаються за допомогою у порядку “живої черги”, що значно обмежує можливості та швидкість обслуговування у подібних закладах.

За кордоном такі сервіси більш розвинені, серед прикладів можна навести американські служби:

Feeding America, <https://www.feedingamerica.org/>

Greater Chicago Food Depository, <https://www.chicagosfoodbank.org/>

Los Angeles Regional Food Bank, <https://www.lafoodbank.org/>

канадські:

Food Banks Canada, <https://www.foodbankscanada.ca/>

Second Harvest, <https://secondharvest.ca/>

та деякі інші:

The Trussell Trust, <https://www.trusselltrust.org/>

Foodbank Australia, <https://www.foodbank.org.au/>

## 1.5. Огляд та обґрунтування використаних технологій

Java:

Існує декілька причин, зважаючи на які, варто використовувати мову Java для програмування веб-додатків, серед них треба виділити такі:

Кросплатформовість:

Java є кросплатформовою мовою програмування, що означає, що веб-додатки, написані на Java, можуть працювати на різних операційних системах, таких як Windows, Linux та macOS. Це дає вам більшу гнучкість і забезпечує, що ваш додаток буде доступним для широкого кола користувачів.

Масштабованість:

Фреймворки, такі як Spring і Java EE, допомагають побудувати масштабовані веб-додатки з використанням різних компонентів та інструментів, що спрощують роботу з базами даних, керуванням сесіями, безпекою, розподіленими системами та багато іншого.

Безпека:

Java має вбудовану систему безпеки, що дозволяє створювати безпечні веб-додатки. Вона включає механізми для управління доступом, шифрування даних, автентифікації та іншого. Це дозволяє забезпечити високий рівень безпеки для користувачів та їх даних.

Багатофункціональність:

Можливість легко інтегрувати різноманітні бібліотеки, сервіси та інші компоненти допомагає розширювати функціональність додатків. Java також має вбудовану підтримку для обробки мультимедіа, мережевих операцій, роботи з XML та іншого.

Spring Boot:

Spring Boot є популярним фреймворком для розробки додатків на мові Java, його використання має численні переваги, наприклад, швидкість

розробки, зручне керування залежностями, вбудований сервер і проста конфігурація.

Spring Boot Starter Web включає в себе підтримку веб-стандартів, це дозволяє створювати веб-додатки, які дотримуються стандартів та легко взаємодіють з іншими веб-технологіями.

Ці фактори та безліч інших роблять Spring Boot вдалим вибором для розробки веб-додатків на Java.

Вибір бази даних H2:

H2 є вбудовуваною базою даних, яка може бути використана в додатках на Java та Spring Boot.

H2 надає простий та зручний інтерфейс для роботи з базою даних, її можна швидко налаштувати та підключити до веб-додатку без значних зусиль.

H2 є швидкою та ефективною базою даних. Вона підтримує оптимізацію запитів, кешування та індексування, що дозволяє забезпечити високу продуктивність додатку.

H2 підтримує різні режими роботи, включаючи режим в пам'яті, файловий режим та режим сервера.

Можна легко перенести дані на іншу базу даних, якщо це необхідно, бо H2 підтримує стандартні SQL-запити.

Використання H2 як бази даних для додатку на Java та Spring Boot дозволяє забезпечити простоту використання, ефективність, гнучкість та сумісність з іншими базами даних.

## 2. ОСНОВНА ЧАСТИНА

### 2.1. Розробка архітектури додатку

#### 2.1.1. Опис функціональності системи

Користувачами системи будуть: Гість (Незареєстрований Користувач), Замовник, Менеджер та Адміністратор.

Гість матиме доступ до головної сторінки додатку, де описана інформація про сервіс, список міст, де послуги є доступними, контактні дані та функції реєстрації та авторизації.

Реєстрація користувачів: Система надає користувачам можливість зареєструватися з використанням номеру телефону.

Метою Замовника є:

Редагування персональних даних: Система надає можливість редагувати персональні данні, де окрім контактних даних Замовник може зазначати алергії та дієтичні обмеження.

Перегляд інформації стосовно адрес та розкладів пунктів видачі у місті: Система дає Замовнику можливість знайти необхідний пункт видачі та переглянути доступну про нього інформацію.

Формування замовлення: Система дає Замовнику можливість сформувати та оформити замовлення на отримання продуктового пакету, що включає в себе автоматичне виключення товарів, що не підходять Замовнику, зважаючи на обмеження, зазначені у персональних даних; та можливість видалення товарів, що не є потрібними Замовнику на даний момент часу.

Приклад Use-case діаграми Замовника приведений на рис. 1.



Метою Адміністратора є:

Додавання або видалення елементів: Система надає Адміністратору можливість додавати та видаляти відділення, продукт та категорію продуктів.

Додавання або видалення менеджерів: Система надає Адміністратору можливість видаляти або додавати Менеджерів.

Приклад Use-case діаграми Адміністратора наведений на рисунку 3.

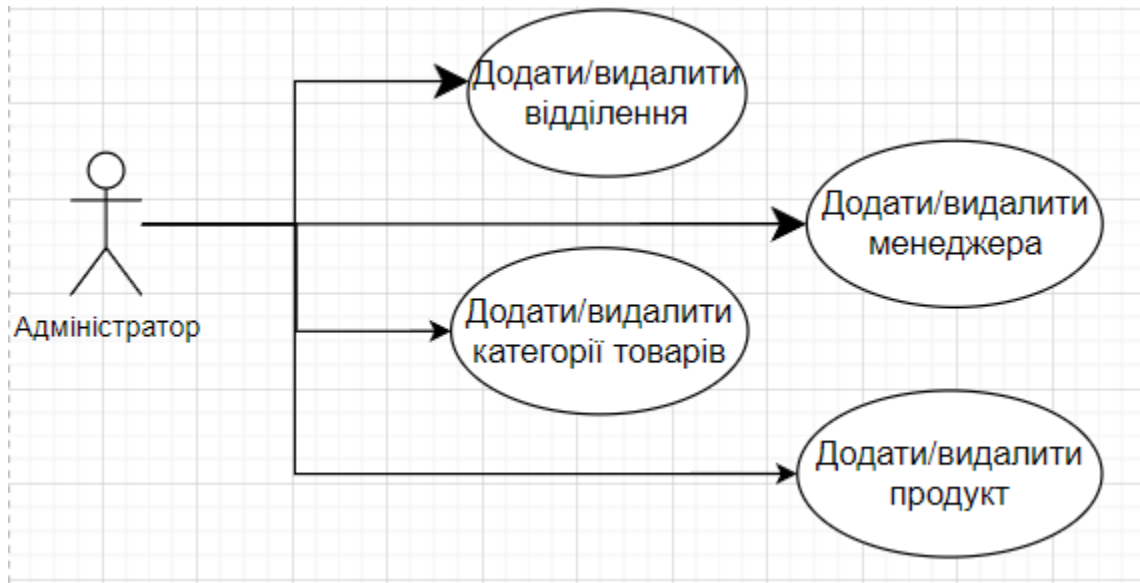


Рисунок 3 - Use-case діаграма Адміністратора

## 2.1.2. Моделювання Бази Даних

Для реалізації функціональності цього додатку підійде реляційна база даних. Використання реляційної моделі бази даних для цього додатку має кілька переваг: Структуровані дані; Цілісність даних; Запити та звіти; Масштабованість; Безпека.

Перший крок моделювання бази даних: зрозуміти, якими будуть основні таблиці, та які типи зв'язку будуть між ними.

Основними таблицями моєї бази даних будуть:

1. Замовник
2. Алергени
3. Продукти
4. Замовлені продукти
5. Пункт видачі
6. Замовлення

Поєднаємо таблицю “Замовник” з таблицею “Алергени”, у замовника може бути декілька алергенів, тож зв'язок буде один до багатьох. Додамо у таблицю “Алергени” строку “id\_client”, що буде посилатися на первинний ключ у таблиці “Замовник”. Такий підхід надасть легкий доступ до користувача через екземпляр “Алергени”, що дозволить спростити подальший процес сортування замовлених продуктів.

Далі створимо таблицю “Продукти” – таблиця для створення екземплярів продуктів з вказанням таких характеристик, як назва, склад, термін використання та категорія. У таблиці “Продукти” буде міститися рядок “elements” що є строкою з перерахуванням складу продукту, ця строка допоможе виявити алергени у продукті.

Таблиця “Замовлені Продукти” буде проміжною між користувачем та замовленням та включатиме в себе `id_client` та `id_product`, ця таблиця допоможе не лише зібрати усі продукти у список, що потім передасться у “Пункт Видачі”, а й надасть додатковий функціонал, наприклад, можливість вилучити продукт зі списку за бажанням замовника.

Таблиця “Пункт Видачі” міститиме інформацію про пункт видачі. Поєднаємо її з замовником, додавши зовнішній ключ “`id_pickupPoint`” до таблиці “Замовник”.

І на сам кінець, створимо таблицю “Замовлення”, що буде формуватися методом поєднання попередніх таблиць та матиме такі рядки:

- “`composition`” – список назв продуктів, що потрібні клієнту
- “`time`” – час, коли замовлення було сформоване
- “`state`” – стан замовлення
- “`id_client`” – зовнішній ключ Замовника для прив’язання таблиці

Замовлення до таблиці Замовника

- “`id_pickupPoint`” – зовнішній ключ Пункта Видачі

ER-діаграма цієї бази даних наведена на рис. 4.

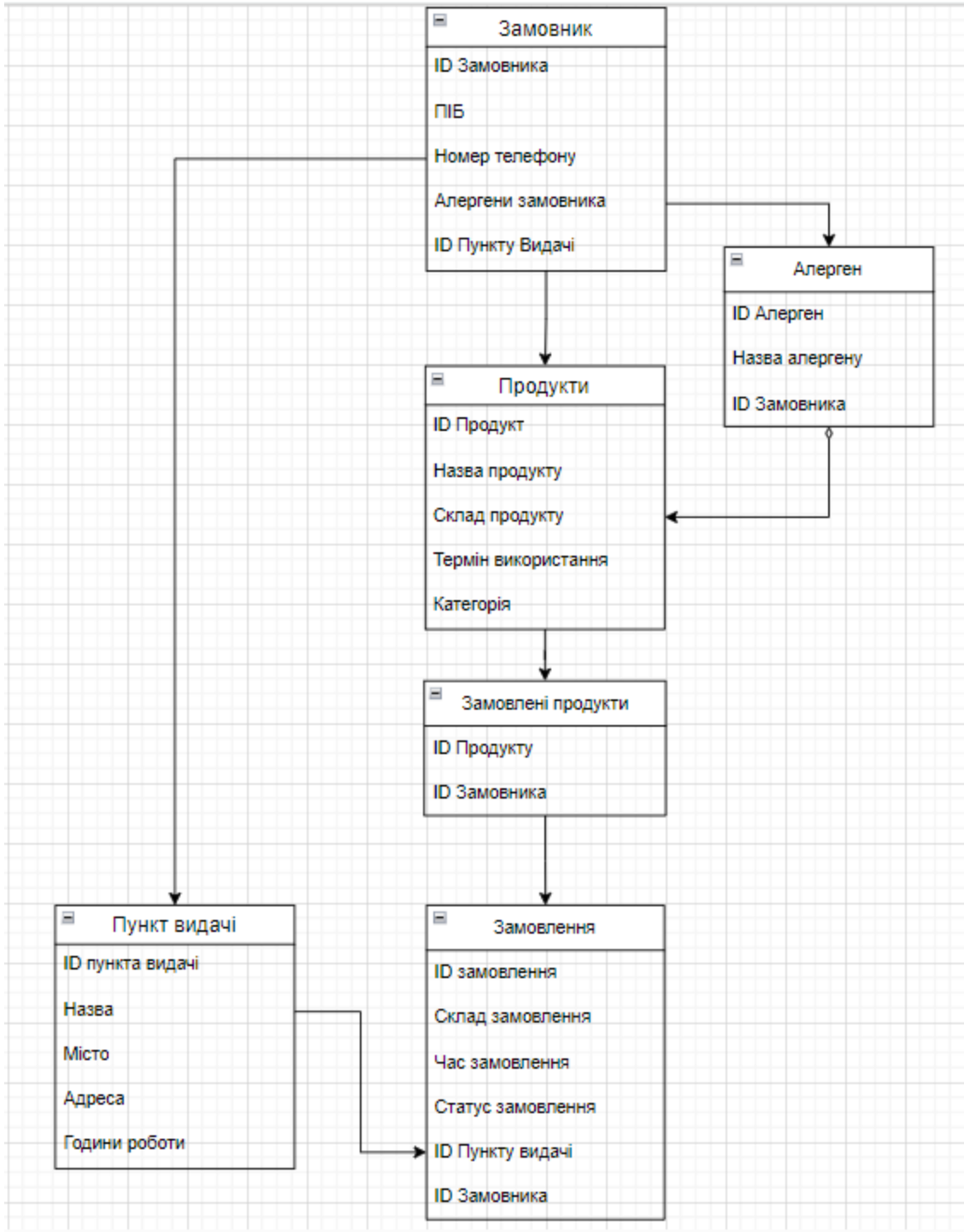


Рисунок 4 – ER-діаграма

### 2.1.3. Опис структури класів та зв'язків

#### Клас "Замовник":

Поля: id, ім'я, номер телефону

Зв'язки:

- One-to-One з класом "Пункт видачі" через поле "id пункту видачі"
- One-to-Many з класом "Алерген" через поле "id замовника"
- One-to-Many з класом "Замовлені продукти" через поле "id замовника"

#### Клас "Алерген":

Поля: id, назва алергену

Зв'язки:

- Many-to-One з класом "Замовник" через поле "id замовника"

#### Клас "Продукт":

Поля: id, назва продукту, склад, термін використання, категорія

Зв'язки:

- Many-to-Many з класом "Замовник" через клас "Замовлені продукти"

#### Клас "Пункт видачі":

Поля: id, назва, адреса, години роботи

Зв'язки:

- One-to-One з класом "Замовник" через поле "id пункту видачі"
- One-to-Many з класом "Замовлення" через поле "id пункту видачі"

#### Клас "Замовлення":

Поля: id, час створення

Зв'язки:

- Many-to-One з класом "Пункт видачі" через поле "id пункту видачі"

- Many-to-One з класом "Замовник" через поле "id замовника"
- Many-to-Many з класом "Продукт" через клас "Замовлені продукти"

Клас "Замовлені продукти":

Поля: id продукту, id замовника

Зв'язки:

- Many-to-Many з класом "Продукт" через поле "id продукту"
- One-to-Many з класом "Замовник" через поле "id замовника"

На рис. 5 наведено приклад діаграми класів:

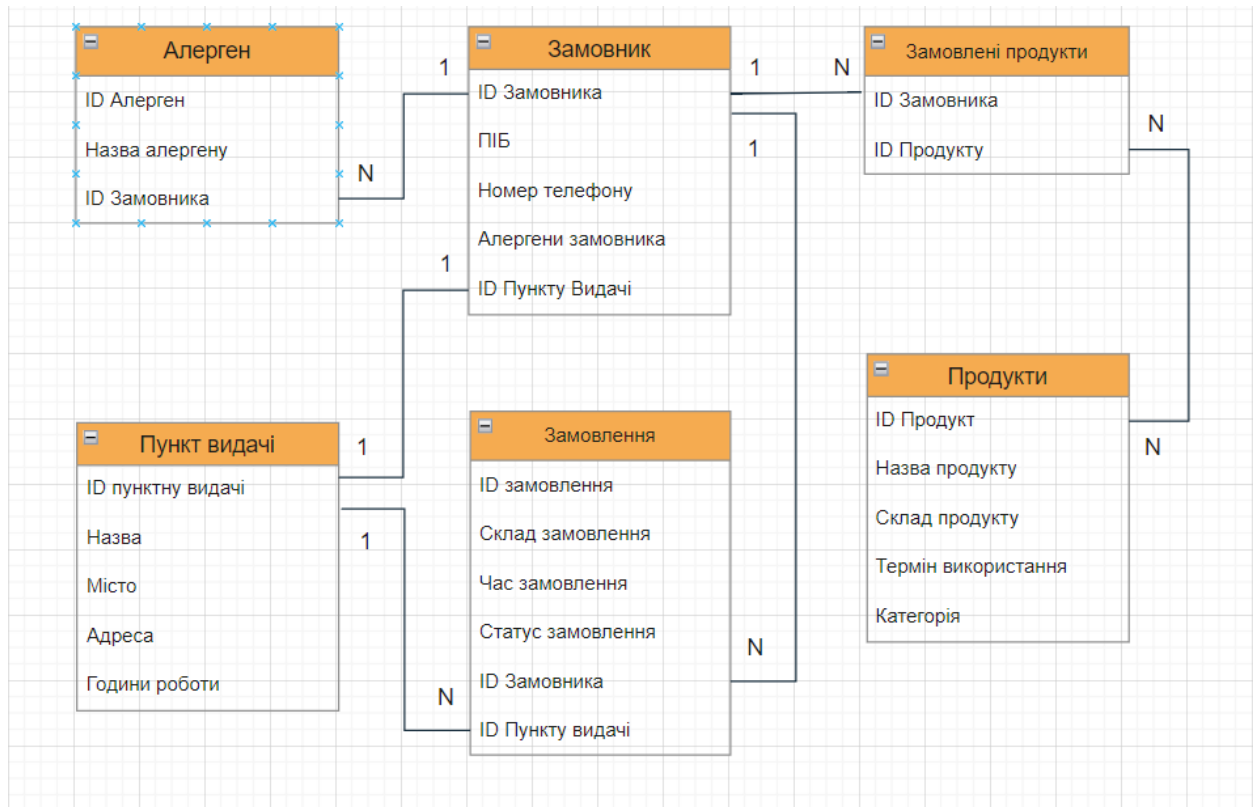


Рисунок 5 – Приклад діаграми класів

## 2.2. Розробка UX/UI дизайну

### 2.2.1. Вимоги до інтерфейсу користувача

Інтуїтивний та простий у використанні – саме таким має бути інтерфейс додатку банку їжі.

Користувачі з будь-яким рівнем користувацького досвіду мають швидко адаптуватися до інтерфейсу та ефективно задовольняти свої споживацькі потреби.

Також варто враховувати користувачів, що можуть мати вади зору.

Отже можна вивести загальні вимоги до інтерфейсу додатку:

- Легкість у використанні
- Мінімальна кількість елементів на екрані
- Легка, інтуїтивно-зрозуміла механіка
- Відсутність зайвих кроків
- Однозначність тексту та позначок на сторінці
- Чітко-видимі, контрасні об'єкти

## 2.2.2. Процес проєктування користувацького досвіду (UX)

Дослідження користувачів:

Цільовою аудиторією додатку є люди 16+, що мають фінансові проблеми. Так у цю категорію входять, як студенти так і люди похилого віку з можливими вадами слуху та зору.

Ціллю користувачів є: зареєструватися у додатку та зробити замовлення на отримання продуктового набору.

Проблеми: обмежений доступ до інтернет-ресурсу. Труднощі з адаптацією до нових пристроїв та сервісів. Можливі вади слуху та зору.

Очікування від додатку: зручний, інтуїтивно зрозумілий та стабільний інтерфейс, до якого можна швидко призвичаїтися.

У процесі проєктування користувацького досвіду важливу роль відіграє створення персонажів користувачів, тобто опису портретів цільової аудиторії.

Створення персонажів користувачів:

Іван - Студент

Вік: 17 років

Стать: Чоловік

Освіта: Студент коледжу

Проблеми: Низький дохід, не зручні умови для зберігання продуктів

Мотивація: Отримати швидкий доступ до дешевої їжі у кількості, зручній для швидкого споживання

Потреби: Швидке замовлення та легке сортування продуктів

Марія - Зайнята мати

Вік: 35 років

Стать: Жінка

Робота: Вчителька

Мотивація: Знайти швидке та зручне рішення для отримання продуктів, враховуючи алергії своїх дітей та маленький бюджет

Потреби: Персоналізовані рекомендації з огляду на алергії дітей, можливість збереження персональних налаштувань для скорочення часу оформлення замовлення, спрощений процес досягання пункту видачі

Петро - Вегетаріанець, еко-активіст

Вік: 28 років

Стать: Чоловік

Освіта: Бакалавр

Мотивація: Знайти продукти, які задовольняють його вегетаріанський спосіб життя та зменшити кількість продуктових відходів

Потреби: Легкий пошук вегетаріанських продуктів, фільтри для виключення продуктів, що містять м'ясо, можливість перегляду складу та харчової цінності продуктів

Ольга - Пенсіонерка

Вік: 70 років

Стать: Жінка

Робота: Пенсіонерка

Проблеми: Відсутність постійного місця проживання, низький дохід, проблеми зору, цукровий діабет, обмежений доступ до інтернет-ресурсу.

Мотивація: Знайти легкий спосіб отримання продуктів, щоби нагодувати себе та свого чоловіка

Потреби: Легкий пошук продуктів, фільтри для виключення продуктів, що мають у складі цукор, стабільний та інтуїтивно зрозумілий інтерфейс додатку

Інформаційна архітектура є важливим етапом в розробці додатків, оскільки вона визначає структуру та організацію контенту для забезпечення зручного використання користувачами.

Головна сторінка:

Навігаційна панель: Зверху сторінки розміщена навігаційна панель з логотипом додатка, а також кнопками “Реєстрація” та “Вхід”, а зліва розміщені розділи для швидкого отримання необхідної інформації про опис та доступність послуг.

Короткий опис: На головній сторінці відображається новинна стрічка додатка та кнопки для реєстрації та авторизації, а також вікна перегляду необхідної інформації.

Розділи додатка:

Навігаційне меню: Зліва сторінки будуть розміщені посилання на інформаційні розділи додатку. Зверху – панель з основними навігаційними кнопками, такими як “Повернутися”, “Допомога”, “Завершити вибір” і т.п. Знизу – панель з посиланнями на соціальні мережі та додатковою контактною інформацією.

Вміст сторінок: Кожний розділ додатка має свою власну сторінку, на якій відображається відповідний контент та функціональність. Наприклад, на сторінці "Вибір продуктів" мають бути перелік категорій та назв продуктів з можливістю відміни або підтвердження вибору.

Деталі сторінок:

Елементи навігації: На кожній сторінці розміщуються елементи навігації, які допомагають користувачеві переміщатися між розділами, завершувати дію або повернутися до головної сторінки.

Контент: Кожна сторінка має відповідний контент, що відображає необхідну інформацію для користувача. Наприклад, на сторінці "Замовлення" можуть бути відображені списки замовлень з інформацією про кожне замовлення.

### 2.2.3. Процес проєктування інтерфейсу користувача (UI)

Кольорова гама:

Приглушені кольори фону та контрасні кольори тексту зробили інтерфейс читким та таким, що відмінюють проблему нерозбірливості тексту та переходів між розділами.

Я вирішила обрати ось таку палітру ( рис. 6, перерахування ведеться зліва-направо), де:

- Перші три кольори слугуватимуть кольорами фону. Приємні, достатньо контрасні та спокійні кольори допоможуть призвичаїтися до взаємодії з додатком.
- Останній темний колір виступатиме кольором тексту, його контрастність з кольорами фону гратиме важливу роль у покращенні користувацького досвіду.

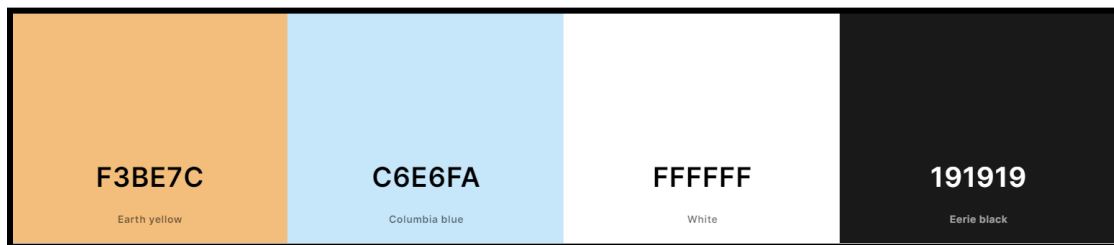


Рисунок 6 – Палітра інтерфейсу додатку

Логотип та назва:

Легкий, розбірливий логотип та зрозуміла, коротка назва – одні з головних атрибутів додатку.



Рисунок 7 – Логотип додатку

Умовні позначення:

Контрастні та прості у розумінні умовні позначення, можуть замінити текст, коли це необхідно

Візуальні характеристики:

Великий, розбірливий текст, інтуїтивне розташування кнопок – ці та інші характеристики спрощують взаємодію користувача з додатком.

Перехід між сторінками зі сторони замовника може бути виконана у такій послдовності: Початкова сторінка → Сторінка перегляду інформації про доступні послуги → Сторінка реєстрації або входу → Головна сторінка замовника → Сторінка заповнення персональних даних → Сторінка вибору відділення → Сторінка дієтичних та алергічних обмежень → Сторінка вибору продуктів замовлення → Сторінка перегляду повної інформації замовлення → Сторінка успіху формування замовлення → (Вихід) → Початкова сторінка.

Розглянемо приклади деяких сторінок додатку:

На рисунку 8 представлена початкова сторінка, яку бачить кожен гість сайту.



Рисунок 8 – Початкова сторінка

Сторінка виконана у мінімалістичному стилі, вміщує в себе лише найнеоюхіднішу інформацію, не перегружаючи композицію. Інформація розміщена відповідно до звичних моделей сайтів, так контактна інформація розміщена знизу, а новини – у центрі сторінки, що не залишає шансу лишитися непроінформованим стосовно оновлень сервісу та майбутніх подій.

Рисунок 9 відображає сторінку вибору продуктів:

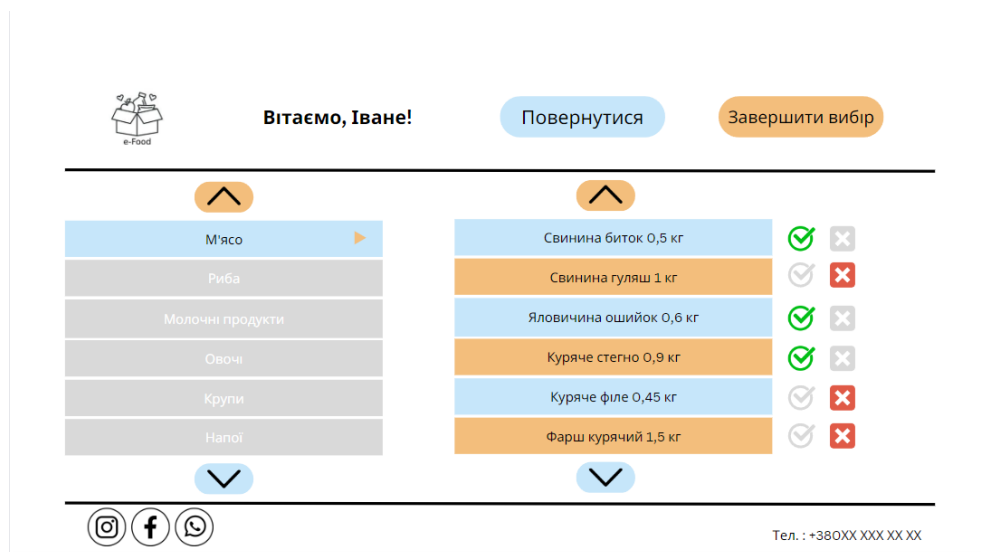


Рисунок 9 – Сторінка вибору продуктів

Так зліва бачимо список категорій продуктів (вибрана категорія підсвічена блакитним), перехід між категоріями здійснюється за допомогою стрілочок. А справа відображені продукти обраної категорії. Біля кожного продукту бачимо зрозумілі символи “обрати” та “відхилити”. Панель контактної інформації збережена. Наступним кроком після цього вікна можуть бути натискання кнопки “Повернутися”, що призведе до повернення користувача на головну сторінку замовника, або “Завершити вибір”, що переведе користувача до вікна підтвердження замовлення.

На рисунку 10 бачимо сторінку успіху формування замовлення.



Рисунок 10 – Замовлення сформоване

На цій сторінці збережене звичне форматування сторінки, але окрім звичних елементів маємо блакитну кнопку “Вийти”, що допоможе замовнику швидко вийти зі свого акаунту для збереження безпеки та конфіденційності. Ця функція може бути корисна у випадку, якщо замовник використовує не власний ПК.

### 2.3. Майбутні перспективи розвитку проєкту

Додаток банку їжі є перспективним проєктом, який може бути вдосконалено та розширено. Деякі можливості для майбутнього розвитку проєкту включають:

Забезпечити можливості зробити грошову пожертву:

На жаль, у наші часи важко вести проєкт лише на волонтерських заставах, тож будь-яка грошова підтримка постійних спонсорів або небайдужих людей значно допоможе підтримати проєкт, надати можливість розширення функціональності та територіального охоплення, збільшити кількість працівників.

Розширення функціональності:

Неперервне розширення функціональних можливостей додатку приведе до покращення користувацького досвіду та надання їм більшої корисної інформації та послуг.

Взаємодія з користувачем:

Розробити способи взаємодії з користувачем, такі як анімація, переходи між сторінками, реакції на дії користувача методом підсвітлення тексту та впливаючих повідомлень, нотаток. Забезпечити додаткові можливості для людей з порушеннями зору та слуху, наприклад: виділення та збільшення тексту, голосовий супровід, озвучення очікуваних дій, розпізнавання слів та авто виправлення.

Розширити діапазон пристроїв, на яких відбуватиметься коректне відображення змісту та повного функціоналу додатку

Доставка замовлень:

Пандемія та війна, знижують або унеможлиблюють користування громадським транспортом, а доступ до авто мають не кожні, це може стати на заваді отримання послуг сервісу. Дієвим вирішенням цієї проблеми стане сервіс доставки продуктів. Так можна допомогти ще більшому колу людей, наприклад батькам-одинакам та людям з недієздатними родичами, хто не можуть покинути опіканців та людям, що мають проблеми переміщення або ослаблений імунітет.

Замовлення:

Зазначення пріоритетності товарів у списку. Так, якщо у пункті видачі закінчаться деякі товари, їх могли замінити іншими, або, у зворотньому випадку, працівники розуміли, кому необхідніший той чи інший товар.

Нагальна потреба:

Додати пріоритетність замовників, що буде вираховувати система спираючись на такі факти, як: матеріальне становище, продуктові потреби, мотивація та інше.

Можливість оформлення замовлення на родину або групу людей:

Щоб зменшити кількість зареєстрованих користувачів та кількість відвідувачів офлайн, а також розширити клієнтську базу, варто додати функцію отримання та/або оформлення замовлення однією людиною на декількох потребуючих.

Діалог допомоги:

Для отримання підтримки стосовно користування послугами сервісу або інтерфейсом додатку можна додати можливість комунікувати з чат-ботом або живою людиною у режимі онлайн.

Перегляд історії замовлень:

Користувачеві може бути корисною функція перегляду історії замовлень, наприклад, для розуміння, які продукти найчастіше є у наявності.

### **3. ВИСНОВКИ**

В ході дипломної роботи було розроблено макет додатку food bank, який може покращити роботу центрів видачі продуктової допомоги, забезпечуючи легкий доступ населення.

Під час роботи було обґрунтовано актуальність обраної теми, проаналізовано наявні аналоги (як вітчизняні, так і зарубіжні), а також сформульовано постановку задачі. Також проведено огляд та обґрунтування сучасних технологій, які дозволяють реалізувати поставлені задачі. Основними результатами дипломної роботи є розробка архітектури додатку, опис функціональності системи, моделювання бази даних, сформульовані вимоги до інтерфейсу користувача, на основі яких розроблений UI/UX дизайн додатку. Також окреслені майбутні перспективи розвитку проєкту.

Таким чином, поставлені задачі виконані повністю.

#### 4. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Поняття ER-моделі [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bestprog.net/uk/>
2. Дідковська М. Проєктування програмного забезпечення / Н. Степанова [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/hnizy>
3. Що таке UI/UX дизайн [Електронний ресурс] – Режим доступу до ресурсу: <https://prjctr.com/mag/uxui-questions>
4. Статті про дизайн українською [Електронний ресурс] – Режим доступу до ресурсу: <https://ux.pub/t/ua>
5. Java Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.oracle.com/en/java/>
6. Spring Boot Reference Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
7. H2 Database Engine [Електронний ресурс] – Режим доступу до ресурсу: <https://www.h2database.com/html/main.html>
8. How to Build a Web App [Електронний ресурс] – Режим доступу до ресурсу: <https://kissflow.com/application-development/how-to-create-a-web-application/>
9. Website vs. Web Application [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/hnjem>