Міністерство освіти і науки України

Національний університет «Одеська політехніка»

Інститут комп’ютерних систем

Кафедра інформаційних систем

Іщенко Олександр Геннадійович,

студент групи АІ-212

ДИСЦИПЛІНА

Об’єктно-орієнтоване програмування

КУРСОВА РОБОТА

Розробка веб-додатку для ведення обліку особистих витрат

Спеціальність:

122 Комп’ютерні науки

Освітня програма:

Комп’ютерні науки

Керівник:

Годовиченко Микола Анатолійович,

кандидат технічних наук, доцент

Одеса – 2023

ЗМІСТ

Анотація 4

Вступ 5

[1 Огляд систем-аналогів та технологій їх розробки 6](#_Toc137028404)

[1.1 Особливості використання мобільних технологій для ведення справ та особистої продуктивності 6](#_Toc137028405)

[1.2 Огляд додатків для ведення заміток 7](#_Toc137028406)

[1.2.1 Додаток Remember The Milk 8](#_Toc137028407)

[1.2.2 Додаток ColorNote 9](#_Toc137028408)

[1.3 Формування вимог до основних функцій мобільного додатку 10](#_Toc137028409)

[1.4 Огляд інформаційних технологій для розробки мобільного додатку 11](#_Toc137028410)

[1.4.1 Фреймворк Jetpack Compose 11](#_Toc137028411)

[1.4.2 Архітектурні компоненти Android 13](#_Toc137028412)

[1.5 Висновки до першого розділу 14](#_Toc137028413)

[2 проєктування мобільного додатку для ведення заміток 16](#_Toc137028414)

[2.1 Мета та задачі мобільного додатку 16](#_Toc137028415)

[2.2 Визначення функціональних вимог до мобільного додатку 17](#_Toc137028416)

[2.3 Формування користувацьких історій мобільного додатку 19](#_Toc137028417)

[2.4 Визначення нефункціональних вимог до мобільного додатку 21](#_Toc137028418)

[2.5 Ідентифікація архетипу мобільного додатку 21](#_Toc137028419)

[2.6 Проектування навігаційного графу мобільного додатку 21](#_Toc137028420)

[2.7 Проектування користувацького інтерфейсу мобільного додатку 22](#_Toc137028421)

[2.8 Висновки до другого розділу 26](#_Toc137028422)

[3 ПРОГРАМНА РЕАЛІЗАЦІЯ мобільного додатку для ведення заміток 27](#_Toc137028423)

[3.1 Структура серверного програмного проєкту 27](#_Toc137028424)

[3.2 Діаграма класів веб-ресурсу 29](#_Toc137028425)

[3.3 Керування вихідним кодом веб додатку 29](#_Toc137028426)

[3.4 Функціональне тестування розробленого веб-ресурсу 31](#_Toc137028427)

[3.5 Інструкція користувача веб-ресурсу 34](#_Toc137028428)

[3.6 Вихідний код веб-ресурсу 41](#_Toc137028429)

[3.7 Висновки до третього розділу 41](#_Toc137028430)

[ВИСНОВКИ 43](#_Toc137028431)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 44](#_Toc137028432)

АНОТАЦІЯ

Курсова робота присвячена розробці веб-додатку для ведення обліку особистих витрат. Додаток призначений для зручного створення та управління користувальницькими замітками на мобільних пристроях. Основною метою роботи є створення простого, ефективного та інтуїтивно зрозумілого інструменту, який дозволяє користувачам легко створювати, редагувати, видаляти та організовувати свої замітки на їх мобільних пристроях. В ході розробки додатку будуть використані сучасні технології та практики розробки мобільних додатків, зокрема мова програмування Kotlin для розробки на платформі Android, бібліотека Jetpack Compose та архітектура MVVM.

ABSTRACT

The course work is devoted to the development of a mobile application for creating notes. The application is designed to conveniently create and manage user notes on mobile devices. The main goal of the work is to create a simple, efficient and intuitive tool that allows users to easily create, edit, delete and organize their notes on their mobile devices. The application will be developed using modern technologies and practices of mobile application development, including the Kotlin programming language for Android development, the Jetpack Compose library and the MVVM architecture.

ВСТУП

У сучасному світі, де мобільні пристрої стали невід'ємною частиною нашого повсякденного життя, зростає потреба в ефективних та зручних інструментах для організації наших справ, ідей та нагадувань. Одним з найпоширеніших і водночас простих засобів є замітки. Вони дозволяють нам швидко зафіксувати важливу інформацію, зберегти цінні думки та спростити наше щоденне життя.

З урахуванням цієї потреби, було вирішено зосередитися на розробці мобільного додатку для заміток, який дозволить користувачам зручно створювати, організовувати та управляти своїми замітками прямо зі своїх мобільних пристроїв. Мета роботи полягає у створенні інтуїтивно зрозумілого інструменту, який буде доступний на платформі Android, і забезпечуватиме зручний і швидкий доступ до заміток незалежно від місця та часу.

Розробка такого додатку вимагає використання сучасних технологій, програмування та архітектурних принципів, що дозволяють створити стійкий, ефективний і надійний продукт. Додаток повинен мати інтуїтивний інтерфейс, що дозволить користувачам швидко засвоїти його функціонал і легко використовувати його у повсякденному житті.

Враховуючи викладені обставини, актуальною є тема розробки мобільного додатку для створення заміток. Такий додаток дозволяв би створювати та переглядати замітки зі збереженням їх на локальному сховищі, щоб мати можливість перегляді їх при наступних запусках додатку.

Метою курсової роботи є розробка мобільного додатку для управління замітками.

Для досягнення цієї мети необхідно виконати наступні задачі:

* здійснити огляд інших додатків для створення користувацьких заміток;
* проаналізувати та обрати сучасні технології для створення мобільного додатку;
* провести проєктування мобільного додатку заміток;
* виконати програмну реалізацію спроєктованого мобільного додатку.

# Огляд систем-аналогів та технологій їх розробки

## Особливості використання мобільних технологій для ведення справ та особистої продуктивності

Мобільні технології відіграють значну роль у поліпшенні особистої продуктивності та веденні справ. Вони надають користувачам широкі можливості доступу до інформації, організації задач і спрощення повсякденних обов'язків. Ось декілька особливостей використання мобільних технологій у цьому контексті:

* мобільний доступ – завдяки мобільним пристроям, ми можемо мати постійний доступ до наших завдань, розкладу, заміток та інших важливих даних незалежно від місця перебування. Це дозволяє нам бути організованими і ефективними навіть у рухливому режимі;
* синхронізація – багато мобільних додатків для ведення справ підтримують синхронізацію даних через хмарні сервіси. Це означає, що ми можемо оновлювати, редагувати або видаляти задачі на одному пристрої і автоматично синхронізувати зміни на всіх інших підключених пристроях. Це забезпечує консистентність даних і уникнення втрати інформації;
* нагадування та сповіщення – мобільні пристрої дозволяють нам встановлювати нагадування та отримувати сповіщення про надходження важливих подій або завдань. Це допомагає нам не пропустити терміни, зустрічі або інші важливі події, що сприяє підвищенню продуктивності та плануванню часу;
* можливості організації – мобільні додатки для ведення справ надають широкі можливості організації завдань, проектів і заміток. Ми можемо створювати категорії, мітки, теги або групи для легкого сортування і пошуку необхідної інформації. Також, можна використовувати функції пріоритетування, календаря і дедлайнів для кращого керування своїм часом та завданнями;
* мобільні сповіщення – однією з переваг мобільних технологій є можливість отримувати сповіщення навіть без потреби активного відкриття додатку. Це дозволяє нам отримувати нагадування, повідомлення або важливі оновлення безпосередньо на екрані свого мобільного пристрою, що сприяє оперативності і швидкій реакції на події;
* можливості синтезу голосу та розпізнавання письма – деякі мобільні додатки для ведення справ мають функції синтезу голосу та розпізнавання письма. Це дозволяє нам записувати аудіонотатки або набирати текст, використовуючи голосові команди. Це особливо корисно, коли ми знаходимося в русі або не можемо вводити текст руками.

Використання мобільних технологій для ведення справ та особистої продуктивності дозволяє нам бути організованими, керувати часом і завданнями ефективніше, а також забезпечує постійний доступ до нашої важливої інформації. Важливо вибирати додатки, які найкраще відповідають нашим потребам та вимогам, забезпечують зручність використання та надійність.

## Огляд додатків для ведення заміток

Для розробки мобільного додатку веб-ресурсу продуктової крамниці необхідно з'ясувати, які функції повинні бути доступні в ньому. Для цього потрібно дослідити та проаналізувати інші додатки, які також дозволяють вести замітки. Це дозволить встановити особливості роботи цих додатків та з'ясувати позитивні та негативні моменти їх використання. Також цей аналіз допоможе краще зрозуміти потреби користувачів в цій області. Після проведення пошуку в Інтернеті, були знайдені найбільш популярні та відомі додатки, які будуть використовуватися як аналоги в даній роботі:

* додаток Remember The Milk [3];
* додаток ColorNote [4];

Далі потрібно провести аналіз цих додатків з метою встановлення особливостей їх роботи та з'ясування позитивних та негативних моментів їх використання. Це допоможе краще зрозуміти, які функції повинні бути доступні мобільному додатку для створення заміток, та відповісти на потреби користувачів в цій області.

### Додаток Remember The Milk

Додаток Remember The Milk є зручним додатком для ведення справ для зайнятих людей. Ви ніколи не забудете придбати молоко (або що-небудь ще) ще раз (рис. 1.2) [3].

|  |  |  |
| --- | --- | --- |
| Зображення, що містить текст, знімок екрана, програмне забезпечення, Операційна система  Автоматично згенерований опис | Зображення, що містить текст, знімок екрана, програмне забезпечення, Операційна система  Автоматично згенерований опис | Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма  Автоматично згенерований опис |

Рисунок 1.2 – Знімки екрану додатку Remember The Milk

Основний функціонал додатку:

* можливість створювати списки справ;
* можливість отримати нагадування через електрону пошту, мобільний пристрій чи популярні месенджери;
* можливість ділитися своїми списками з іншими людьми;
* синхронізація справ на всіх пристроях;
* можливість додавати пріоритети, терміни закінчення, кількість повторів справи, мітки тощо;
* пошук справ та нотаток за текстом, збереження справ та нотаток в розумних списках;
* можливість бачити справи, які фізично розташовані найближче та у найбільш ефективний спосіб.

Платна версія додатку має додатковий функціонал:

* підзадачі – можна розбивати задачі на менші частини;
* можливість ділитися задачами без обмежень;
* додатковий параметр кольору для задач;
* нагадування про необхідність виконання задачі через мобільний пристрій;
* синхронізація з Microsoft Outlook;
* безлімітне сховище для даних.

### Додаток ColorNote

ColorNote – це простий блокнот. Він надасть можливість легкого і простого користування блокнотом при написанні заміток, нагадувань, email, повідомлень, переліків справ і покупок (рис 1.3) [4]. З ColorNote створення заміток простіше, ніж з будь-якими іншими блокнотами і органайзерами. Розглянемо можливості додатку:

* організація заміток за кольором;
* віджет стікерів;
* списки справ і покупок;
* організація розкладу в календарі;
* блокування замітки паролем;
* захищене резервне копіювання заміток на SD карту – перед вивантаженням замітки будуть зашифровані за стандартом AES, який використовується банками для захисту даних клієнтів;
* підтримка онлайн-синхронізації і резервного копіювання;
* синхронізація заміток між телефоном і планшетом;
* нагадування в статус-барі;
* нагадування про нотатки;
* можливість ділитися нотатками за допомогою SMS, електроної пошти чи твітеру;
* можливість кольорової організації нотаток.

|  |  |  |
| --- | --- | --- |
| Зображення, що містить текст, знімок екрана, Шрифт  Автоматично згенерований опис | Зображення, що містить текст, знімок екрана, число, програмне забезпечення  Автоматично згенерований опис | Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма  Автоматично згенерований опис |

Рисунок 1.3 – Знімки екрану додатку ColorNote

## Формування вимог до основних функцій мобільного додатку

Аналіз додатків для ведення справ та нотаток дозволив виділити їх переваги та недоліки та сформувати вимоги до власного мобільного додатку.

Основний функціонал мобільного додатку для ведення заміток може включати наступні можливості:

* створення заміток – користувач може створювати нові замітки, вводячи текстовий контент. Додаток повинен забезпечувати простий та зручний спосіб створення нових заміток;
* редагування та форматування – користувач може редагувати вміст заміток, включаючи видалення, редагування або переміщення тексту. Додаток може також надавати можливість форматування тексту, таке як жирний, курсив, маркери списку, нумерація тощо, для зручності організації і вигляду заміток;
* категорії та мітки – додаток може дозволяти користувачу створювати категорії або мітки для заміток, що допомагає організувати їх за певними темами, проектами або контекстами. Це дозволяє швидко знаходити потрібну інформацію і забезпечує більшу структурованість;
* пошук та фільтрація – користувач може шукати замітки за ключовими словами або фільтрувати їх за певними критеріями, такими як дата створення, пріоритет, категорія тощо. Це допомагає знаходити необхідні замітки швидко і ефективно;
* безпека даних – з метою забезпечення конфіденційності та безпеки, додаток може пропонувати захист заміток за допомогою паролю, відбитка пальця або інших методів аутентифікації. Це гарантує, що тільки авторизовані користувачі можуть отримати доступ до приватних даних.

## Огляд інформаційних технологій для розробки мобільного додатку

### Фреймворк Jetpack Compose

Jetpack Compose є сучасним фреймворком для розробки інтерфейсу користувача (UI) для мобільних додатків на платформі Android. Ось деякі переваги та недоліки використання Jetpack Compose:

Розглянемо переваги фреймворку Jetpack Compose:

* декларативний підхід – Jetpack Compose використовує декларативний підхід до побудови UI, що означає, що ви описуєте, як повинен виглядати ваш UI, а не як його побудувати крок за кроком. Це спрощує процес розробки та розуміння коду;
* легше утримувати – завдяки декларативному підходу Jetpack Compose дозволяє писати більш зрозумілий та компактний код. Це полегшує розуміння та утримання кодової бази, особливо великих проектів;
* живий попередній перегляд – Jetpack Compose надає можливість відразу бачити результати змін у реальному часі без необхідності компіляції або запуску додатка. Це прискорює процес розробки та дозволяє швидко перевірити вигляд та поведінку UI;
* легка навігація – Jetpack Compose надає простий та зрозумілий API для навігації між екранами додатка. Він спрощує реалізацію функціоналу переходу між фрагментами або активностями, забезпечуючи зручну інтеграцію з іншими компонентами додатка;
* багата функціональність – Jetpack Compose має велику кількість вбудованих компонентів та функцій, які полегшують створення різноманітних UI-елементів, включаючи кнопки, тексти, списки, анімації тощо. Він також має добре підтримувану документацію та активну спільноту, що сприяє розробці та розширенню функціональності.

Розглянемо недоліки фреймворку Jetpack Compose:

* іматурність – Jetpack Compose - це нова технологія, яка все ще знаходиться на етапі активного розвитку. Це означає, що вона може мати деякі недоліки та обмеження, а також потребує постійного оновлення для вирішення проблем та додавання нового функціоналу;
* потреба в оновленні – використання Jetpack Compose вимагає оновлення вашого проекту на новішу версію Android та використання сучасних компонентів, таких як AndroidX. Це може бути вимогливим і вимагати додаткових зусиль для перенесення існуючих проектів на Jetpack Compose;
* навчання – Jetpack Compose відрізняється від традиційного підходу до розробки Android-інтерфейсу, що означає, що розробникам може знадобитися деякий час для вивчення нових концепцій та API. Це може бути викликом для команд, які вже працюють зі старішими підходами;

Незважаючи на деякі недоліки – Jetpack Compose швидко розвивається і вже стає все більш популярним фреймворком для розробки інтерфейсу користувача на платформі Android. Він пропонує простоту, ефективність та розширюваність, що робить його привабливим вибором для розробників.

### Архітектурні компоненти Android

Використання архітектурних компонентів в Android є важливим аспектом при розробці додатків, оскільки вони допомагають забезпечити структурованість, модульність, перевикористовуваність та тестованість коду. Ось деякі ключові переваги використання архітектурних компонентів.

Масштабованість: Архітектурні компоненти, такі як ViewModel, LiveData і Room, надають підхід, який сприяє масштабуваності додатка. Вони допомагають розділити бізнес-логіку від інтерфейсу користувача і забезпечують зручний спосіб управління даними та станом додатка.

Розподілення обов'язків: Архітектурні компоненти, такі як Model-View-ViewModel (MVVM) або Model-View-Presenter (MVP), допомагають чітко розподілити обов'язки між компонентами додатка. Це полегшує розуміння та утримання кодової бази, а також сприяє зручній співпраці в командному середовищі.

Тестованість: Архітектурні компоненти підтримують тестування додатків, оскільки вони розділяють бізнес-логіку від інтерфейсу користувача. Це дозволяє розробникам легко писати автоматизовані тести для перевірки функціональності та надійності додатка.

Легка утримуваність: Архітектурні компоненти пропонують структуровану організацію коду, що спрощує утримання проекту. Це дозволяє розробникам швидко зорієнтуватися в коді, вносити зміни та вдосконалювати додаток з мінімальними зусиллями.

Перевикористання: Використання архітектурних компонентів сприяє перевикористанню коду. Вони дозволяють виділити загальну функціональність у відокремлені компоненти, які можна використовувати в різних частинах додатка або навіть в інших проектах.

Спільність розробки: Архітектурні компоненти забезпечують зручну інтеграцію та спільну роботу в командному середовищі. Це полегшує спільне управління кодовою базою та співпрацю між розробниками.

Загалом, використання архітектурних компонентів в Android рекомендується для покращення якості, швидкості розробки та легкості утримання додатків. Вони допомагають розділити обов'язки, спростити тестування та забезпечити модульність, що робить їх важливою складовою частиною процесу розробки Android-додатків.

## Висновки до першого розділу

В першому розділі курсової роботи був проведений огляд додатків для створення заміток та технологій їх розробки.

Було розглянуто питання впливу мобільних технологій на задачу ведення справ та особистої продуктивності. Було визначено, що мобільні технології відіграють значну роль у поліпшенні особистої продуктивності та веденні справ. Вони надають користувачам широкі можливості доступу до інформації, організації задач і спрощення повсякденних обов'язків.

Далі був проведений огляд додатків-аналогів, який дозволив визначити основні переваги та недоліки існуючих мобільних додатків для ведення заміток, встановити їх основні функції та можливості. На базі проведеного огляду були визначені основні вимоги до власного веб-ресурсу, а також наданий детальний опис цих вимог.

Був проведений огляд інформаційних технологій для розробки мобільного додатку для ведення заміток. У якості мобільної операційної системи виступає ОС Android. У якості фреймворку для розробки користувацького інтерфейсу було вирішено обрати фреймворк Jetpack Compose. Залучення цього фреймворку дозволяє використати декларативний підхід до створення користувацького інтерфейсу, що суттєво полегшує процес його створення. У якості технології для розробки архітектури додатку були обрані архітектурні компоненти. Вони допомагають забезпечити структурованість, модульність, перевикористовуваність та тестованість коду.

# проєктування мобільного додатку для ведення заміток

## Мета та задачі мобільного додатку

Мета мобільного додатку Android для ведення заміток полягає в наданні користувачам зручного та організованого інструменту для зберігання, організації та керування їх замітками. Головною метою додатку є полегшення процесу ведення особистих або професійних записів, щоб користувачі могли легко зберігати важливу інформацію та нагадування.

Задачі мобільного додатку для ведення заміток можуть включати:

* створення нових заміток – користувачі повинні мати можливість швидко створювати нові записи заміток з використанням тексту, фотографій або голосових нотаток;
* організація заміток – додаток повинен надати можливість створення категорій або міток для заміток, щоб користувачі могли легко групувати та організовувати свої записи;
* редагування та видалення заміток – користувачам необхідно мати можливість змінювати вміст заміток, редагувати або видаляти їх в разі потреби;
* пошук та фільтрація заміток – додаток повинен забезпечувати швидкий доступ до заміток за допомогою пошуку або застосування фільтрів за категоріями, тегами чи ключовими словами;
* синхронізація та резервне копіювання – важливо, щоб додаток забезпечував можливість синхронізувати замітки між різними пристроями користувача і забезпечував регулярне резервне копіювання для запобігання втраті даних;
* сповіщення та нагадування – додаток може надавати можливість встановлення нагадувань та сповіщень для заміток, щоб користувачі не пропустили важливі події або завдання;

Призначення мобільного додатку для ведення заміток полягає в полегшенні організації та керування інформацією для користувачів. Він надає зручний спосіб зберігання і використання заміток, дозволяючи легко доступатися до них, редагувати, організовувати та забезпечувати їх безпеку. Мобільний додаток для ведення заміток сприяє покращенню особистої продуктивності, організації ідей, плануванню завдань та важливих подій, а також виконанню рутинних завдань швидко та ефективно.

## Визначення функціональних вимог до мобільного додатку

Визначення функціональних вимог є важливим етапом у процесі створення мобільного додатку для ведення заміток. Функціональні вимоги визначають, які конкретні функції та можливості повинен мати додаток. Вони встановлюють чітку спрямованість розробки, допомагають уникнути неоднозначностей та непорозумінь.

Крім того, визначення функціональних вимог дозволяє зосередитися на потребах та вимогах користувачів. Вони допомагають врахувати, які функції та можливості будуть найбільш корисними для користувачів додатку.

Також, функціональні вимоги слугують основою для комунікації між розробниками, дизайнерами та іншими учасниками проекту. Вони допомагають зрозуміти, що саме потрібно реалізувати та які очікувані результати.

Встановлення функціональних вимог дозволяє визначити обсяг роботи та потребу в ресурсах для реалізації додатку. Це допомагає планувати час, бюджет та ресурси проекту ефективно.

Єдиним актором мобільного додатку є актор «користувач»:

* користувач – користувач мобільного додатку. Має доступ до всіх функцій додатку та несе всю відповідальність за його роботу.

З метою визначення користувацьких історій та нефункціональних вимог до веб-ресурсу, було розроблено діаграму сценаріїв використання мобільного додатку (рис. 2.1). Діаграма сценаріїв UML (Unified Modeling Language) - це графічний інструмент для опису функціональної взаємодії між користувачами та системою. Вона складається з акторів, сценаріїв та взаємодії між ними.

Ця діаграма містить основних акторів системи та описує сценарії їх взаємодії з системою. Вона допомагає проаналізувати залежності між акторами та можливими варіантами використання системи. Також ця діаграма швидко демонструє основні функції системи для розробників.

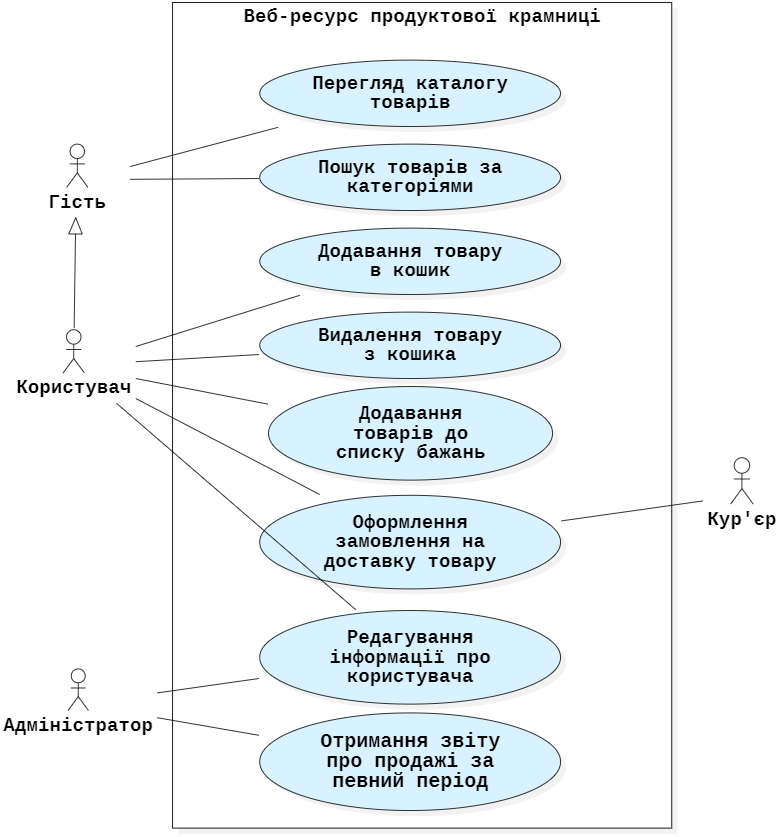


Рисунок 2.1 – Діаграма варіантів використання мобільного додатку

## Формування користувацьких історій мобільного додатку

Проектування діаграми прецедентів дозволяє визначити такі користувацькі історії до мобільного додатку для ведення заміток.

US1 Як користувач, я хочу створити нову замітку, щоб зафіксувати важливу інформацію.

Сценарій користувацької історії виглядає таким чином:

* користувач відкриває додаток для ведення заміток;
* він натискає кнопку «Створити нову замітку»;
* користувач вводить текстове повідомлення або нотатку;
* після завершення введення, користувач натискає кнопку «Зберегти»;
* нова замітка зберігається в додатку і з'являється у списку заміток;

US2 Як користувач, я хочу редагувати вміст замітки, щоб вносити зміни або доповнення до інформації.

Сценарій користувацької історії виглядає таким чином:

* користувач відкриває конкретну замітку;
* він натискає на кнопку «Редагувати» або вибирає опцію редагування;
* користувач вносить необхідні зміни або доповнення до тексту замітки;
* після завершення редагування, користувач натискає кнопку «Зберегти» або подібний елемент, щоб зберегти оновлену версію замітки.

US3 Як користувач, я хочу видаляти замітки, які більше не потрібні, для підтримки чистоти та організованості списку заміток.

Сценарій користувацької історії виглядає таким чином:

* користувач відкриває конкретну замітку або виділяє багато заміток у списку;
* він натискає кнопку «Видалити» або вибирає опцію видалення;
* користувач підтверджує видалення замітки або заміток;
* замітки видаляються з додатку та зникають зі списку заміток.

US4 Як користувач, я хочу синхронізувати свої замітки з обліковим записом, щоб мати доступ до них з різних пристроїв.

Сценарій користувацької історії виглядає таким чином:

* користувач відкриває налаштування облікового запису в додатку;
* він натискає на опцію «Синхронізація» або «Увійти в обліковий запис»;
* користувач вводить свої облікові дані або реєструється новим користувачем;
* після успішної аутентифікації, замітки автоматично синхронізуються з обліковим записом, що дозволяє доступ до них з інших пристроїв.

US5 Як користувач, я хочу додавати зображення до заміток, щоб доповнювати текст ілюстраціями або фотографіями.

Сценарій користувацької історії виглядає таким чином:

* користувач відкриває конкретну замітку або створює нову;
* Він натискає на кнопку «Додати зображення» або вибирає опцію для додавання медіафайлу;
* користувач вибирає зображення зі свого пристрою або робить фотографію в додатку;
* після завершення, зображення додається до замітки і відображається разом з текстом.

US6 Як користувач, я хочу встановити пріоритети для заміток, щоб виділити найважливіші завдання.

Сценарій користувацької історії виглядає таким чином:

* користувач відкриває конкретну замітку або створює нову;
* він натискає на опцію «Встановити пріоритет» або подібний елемент;
* користувач вибирає пріоритетний рівень для замітки (наприклад, високий, середній, низький);
* після збереження, замітка отримує позначення пріоритету, що допомагає користувачеві виділити найважливіші завдання.

## Визначення нефункціональних вимог до мобільного додатку

Після визначення функціональних вимог, необхідно сформувати нефункціональні вимоги, які висуваються до мобільного додатку для ведення заміток.

NFR1 Версія операційної системи – додаток може підтримувати певний діапазон версій, наприклад, Android 8.0 або новіше.

NFR2 Процесор – мінімальний процесор з відповідною архітектурою (наприклад, ARM або x86), який підтримується операційною системою.

NFR3 Оперативна пам'ять (RAM) – мінімальний обсяг оперативної пам'яті, необхідний для плавної роботи додатку. Зазвичай, це може бути від 1 ГБ і більше.

NFR4 Внутрішня пам'ять – додаток може вимагати певний обсяг внутрішньої пам'яті для збереження даних заміток і налаштувань. Це може бути від кількох мегабайт до кількох гігабайт.

NFR5 Роздільна здатність екрану – мінімальна роздільна здатність екрану, яка забезпечує належний візуальний досвід користувача.

NFR6 Інтернет-підключення – додаток може вимагати наявності активного Інтернет-підключення для синхронізації заміток, резервного копіювання, відправки сповіщень тощо.

## Ідентифікація архетипу мобільного додатку

Інформаційна система відноситься до архетипу Mobile Application (MA) –класичний додаток, який запускається на мобільному пристрою під управлінням операційної системи Android.

## Проектування навігаційного графу мобільного додатку

Навігаційний граф (Navigation Graph) – це компонент фреймворку Android Jetpack, який використовується для візуалізації і управління навігацією між екранами (фрагментами) в додатку Android. Він визначає структуру навігації та взаємозв'язки між різними екранами (рис. 2.2).

За допомогою навігаційного графа розробник може легко організувати навігацію між різними екранами в додатку. Він дозволяє встановлювати точки входу (destination) для кожного екрану, визначати перехідні анімації, задавати параметри переходів (наприклад, передачу даних), а також встановлювати специфічні правила навігації, такі як повернення до попереднього екрану (back navigation) або виконання визначених дій перед показом нового екрану.

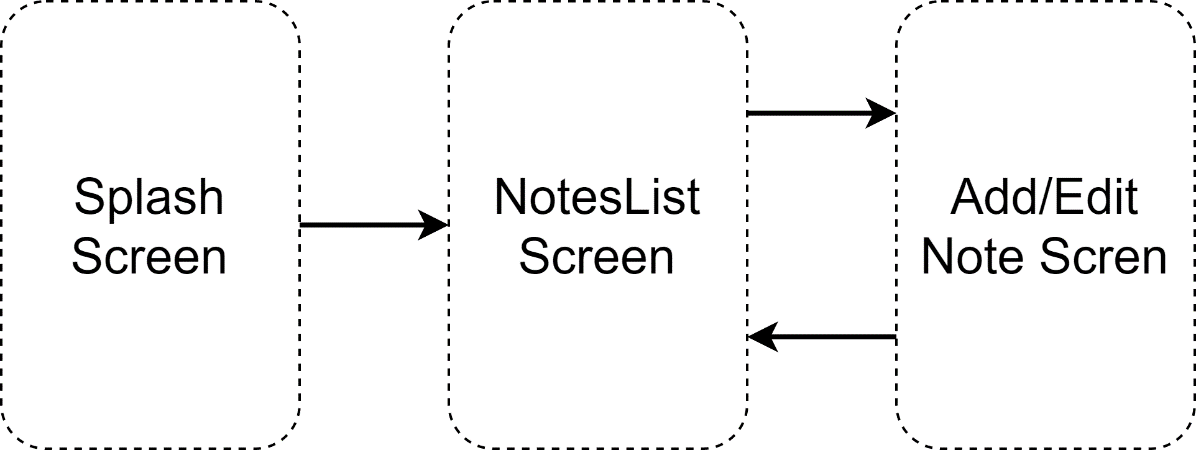


Рисунок 2.2 – Навігаційний граф мобільного додатку

## Проектування користувацького інтерфейсу мобільного додатку

На основі вимог до веб-ресурсу, які включають функціональні та нефункціональні вимоги, можна приступити до проектування користувацького інтерфейсу веб-додатку. Це вимагає розробки макетів ключових вікон веб-сайту, визначення сценаріїв взаємодії користувачів з цими вікнами, складання стислого опису кожного макету та визначення їх ролі в досягненні визначеної мети в цій кваліфікаційній роботі.

На рисунку 2.4 зображений макет головного вікна веб-ресурсу для продуктової крамниці. Головне вікно веб-ресурсу продуктової крамниці містить різні елементи. Елемент шапки включає логотип, пошукову строку, кнопку входу/реєстрації, кнопку кошика. Меню розміщується під шапкою і містить посилання на головні розділи веб-ресурсу, такі як «Продукти», «Акції», «Доставка та оплата», «Контакти» тощо.

Банер розміщується вище або нижче меню і містить зображення з рекламою акційних товарів або послуг. Категорії продуктів можуть бути розміщені на головній сторінці і містити візуальні зображення і назви категорій продуктів, таких як «Овочі та фрукти», «М'ясо та птиця», «Кондитерські вироби» тощо.

Блоки з акційними пропозиціями можуть бути розміщені в боковому меню або на головній сторінці, щоб привернути увагу клієнтів до акційних товарів. Найбільш популярні товари можуть бути розміщені на головній сторінці, щоб привернути увагу клієнтів до найбільш продаваних товарів. Відгуки клієнтів можуть бути розміщені на головній сторінці для стимулювання покупців залишати відгуки про товари та послуги. Кнопка «Показати більше» може бути розміщена на головній сторінці, щоб дозволити користувачам переглядати більше товарів безпосередньо на сторінці.

Футер містить посилання на додаткові розділи веб-ресурсу, контактні дані, логотипи платіжних систем тощо.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.4 – Макет сторінки головного вікна веб-ресурсу

На рисунку 2.5 зображений макет вікна товару веб-ресурсу продуктової крамниці. Даний макет містить назву товару, зображення товару, ціну, опис товару, що містить характеристики, переваги, рецензії від інших покупців тощо. Також в макеті міститься кнопка "Додати в кошик", що дозволяє користувачам додавати товар до свого кошика.

Крім того, у вікні знаходиться посилання на розділ, де можна знайти інші товари того ж бренду або категорії. Також на сторінці знаходиться посилання на обрані товари користувача (список бажань або кошик)

Макет повинен мати привабливий дизайн і забезпечувати користувачам легку навігацію і доступ до усієї необхідної інформації про товар. Також, повинен бути доданий функціонал рекомендацій на основі попередніх покупок, щоб користувачі могли знайти товари, які їм можуть сподобатися.

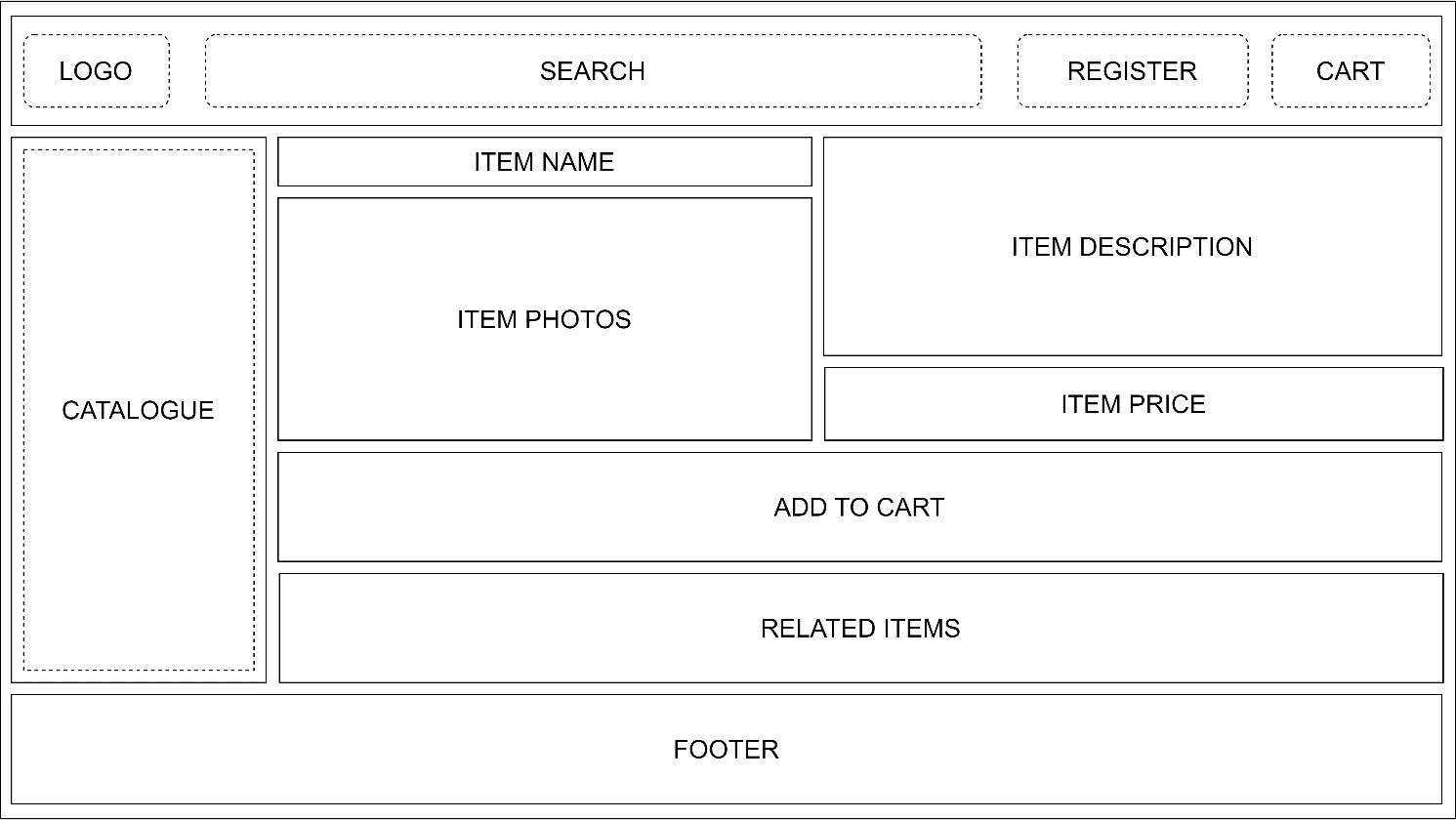


Рисунок 2.5 – Макет сторінки товару веб-ресурсу

На рисунку 2.6 зображений макет вікна кошику товарів веб-ресурсу продуктової крамниці. Макет містить список товарів, які клієнт додав до свого кошика. Для кожного товару вказується його назва, зображення, ціна, кількість, а також загальна вартість товару (ціна на одиницю товару помножена на кількість). Також у макеті є різні кнопки та поля, наприклад:

* кнопка "Видалити товар" – дозволяє видалити товар з кошика;
* кнопка "Оформити замовлення" – дозволяє перейти до оформлення замовлення на покупку товарів з кошика;
* поле для введення промо-коду – дозволяє ввести промо-код для отримання знижки на покупку товарів;
* поле для введення купону на безкоштовну доставку – дозволяє ввести купон на безкоштовну доставку товарів;
* інформація про загальну вартість товарів у кошику – показує загальну вартість всіх товарів у кошику;
* інформація про стан замовлення – показує статус замовлення, наприклад, "в обробці", "відправлено", "доставлено".

Макет кошика повинен бути зручним та інтуїтивно зрозумілим для користувачів, щоб вони могли легко знайти необхідну інформацію та керувати своїм замовленням.

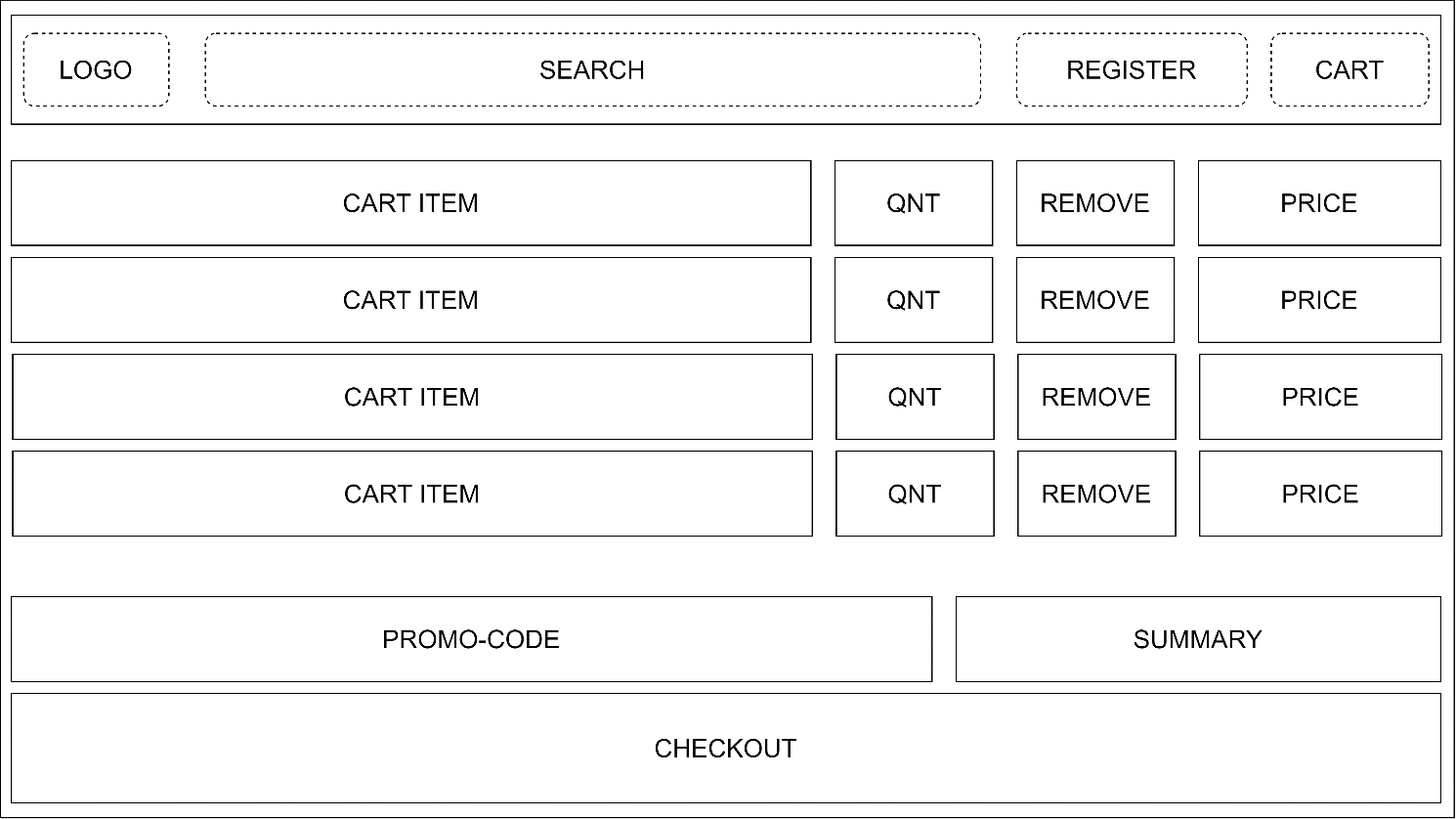


Рисунок 2.6 – Макет вікна кошику товарів веб-ресурсу

## Висновки до другого розділу

У другому розділі цієї курсової роботи було проведено проектування веб-ресурсу продуктової крамниці на базі фреймворку Angular.

У процесі проектування була визначена мета сайту, його потенційна аудиторія та основні можливості. На основі цих даних були встановлені основні вимоги до веб-ресурсу, включаючи функціональні та нефункціональні. Також була розроблена діаграма сценаріїв використання веб-ресурсу. Крім того, була розроблена діаграма інформаційних потоків та визначені вхідні та вихідні інформаційні потоки системи.

З метою проектування інтерфейсу користувача були розроблені макети ключових вікон веб-ресурсу, а також діаграма станів системи. Крім того, була розроблена діаграма логічного уявлення системи, а також діаграма розгортання веб-ресурсу. Для побудови веб-ресурсу була визначена схема даних системи, а також ключові технології, які були використані в процесі проєктування та реалізації веб-ресурсу.

# ПРОГРАМНА РЕАЛІЗАЦІЯ мобільного додатку для ведення заміток

## Структура серверного програмного проєкту

Пакет програмного проекту Spring для веб-ресурсу продуктової крамниці може містити різні компоненти. Визначено ці компоненти та визначимо, які класи реалізують ці компоненти.

Першим компонентом є контролери. Вони відповідають за обробку запитів, які приходять від користувачів та повернення відповіді у формі HTML сторінки. В рамках серверного проєкту були створені наступні контролери:

* контролер для домашньої сторінки (*HomeController.java*) – відповідає за відображення домашньої сторінки веб-сайту;
* контролер для категорій продуктів (*CategoryController.java*) – відповідає за відображення списку категорій продуктів, їх додавання, видалення та редагування;
* контролер для продуктів (*ProductController.java*) – відповідає за відображення списку продуктів, їх додавання, видалення та редагування;
* контролер для кошика (*CartController.java*) – відповідає за відображення вмісту корзини, додавання та видалення продуктів з неї;
* контролер для оформлення замовлення (*OrderController.java*) – відповідає за обробку замовлення, введення адреси доставки та підтвердження оплати;
* контролер для аутентифікації та авторизації користувачів (*AuthController.java*) – відповідає за відображення сторінок входу, реєстрації, відновлення паролю та інші функції, що пов'язані з аутентифікацією та авторизацією користувачів.

Компонент сервісів відповідає за логіку бізнесу та операції, які пов'язані з даними. Сервіси можуть включати методи, які повертають дані з бази даних, змінюють їх та зберігають зміни. В рамках серверного проєкту були створені наступні сервіси:

* *ProductService.java* – сервіс, що забезпечує роботу з продуктами, такі як створення, оновлення, видалення та отримання даних про продукти;
* *OrderService.java* – сервіс, що забезпечує роботу з замовленнями, такі як створення, оновлення, видалення та отримання даних про замовлення;
* *UserService.java* – сервіс, що забезпечує роботу з користувачами, такі як створення, оновлення, видалення та отримання даних про користувачів;
* *CartService.java* – сервіс, що забезпечує роботу з кошиком покупця, такі як додавання, оновлення та видалення продуктів з кошика;
* *PaymentService.java* – сервіс, що забезпечує обробку оплати за замовлення, такі як створення та відправлення запитів на оплату, перевірку стану оплати та оновлення стану замовлення в залежності від результатів оплати;
* *ShippingService.java* – сервіс, що забезпечує роботу з доставкою замовлення, такі як створення та відправлення запитів на доставку, відстеження стану доставки та оновлення стану замовлення в залежності від результатів доставки;
* *AuthenticationService.java* – сервіс, що забезпечує автентифікацію та авторизацію користувачів на веб-ресурсі;
* *ImageService.java* – сервіс, що забезпечує роботу з зображеннями продуктів, такі як завантаження, збереження та отримання даних про зображення.

Також, в проєкті містяться такі компоненти:

* репозиторії– відповідають за збереження та отримання даних з бази даних. Репозиторії можуть містити методи, які зберігають, оновлюють та видаляють дані з бази даних;
* моделі – представляють сутності, які використовуються в системі. Моделі можуть містити поля, які відображають дані з бази даних, та методи, які допомагають взаємодіяти з цими даними;
* конфігураційні файли – відповідають за конфігурування та налаштування Spring Framework. Конфігураційні файли можуть містити налаштування зв'язку з базою даних, налаштування безпеки та інші параметри, які необхідні для правильної роботи системи.

## Діаграма класів веб-ресурсу

Діаграма класів є одним з видів UML-діаграм і представляє зовнішній вигляд класів, інтерфейсів та їх зв'язків у системі або додатку. Діаграма класів допомагає візуалізувати структуру системи та відношення між її компонентами.

Діаграма класів веб-ресурсу продуктової крамниці на базі фреймворку Angular наведена на рисунку 3.1.

## Керування вихідним кодом веб додатку

Використання системи контролю версій є дуже важливим для будь-якого програмного проекту, в тому числі і для веб-ресурсу продуктової крамниці. Використання системи контролю версій дозволяє зберігати історію всіх змін, які були внесені до веб-ресурсу.

Це дозволяє в разі потреби повернутися до попередніх версій веб-ресурсу та відновити попередній стан, який був на момент попередньої версії.

Також, система контролю версій дозволяє контролювати версії веб-ресурсу, що значно полегшує роботу з веб-ресурсом та дозволяє підтримувати його в актуальному стані.

Основні метрики репозиторію веб-ресурсу наведені в таблиці 3.1.

Таблиця 3.1 – Метрики керування програмним кодом додатку

|  |  |  |  |
| --- | --- | --- | --- |
| Веб-ресурс | Кількість комітів | Кількість pull-реквестів | Кількість гілок у репозиторії |
| Клієнтська частина | 28 | 11 | 3 |
| Серверна частина | 32 | 8 | 4 |

|  |  |
| --- | --- |
|  | Рисунок 3.1 – Діаграма класів веб-ресурсу |

## Функціональне тестування розробленого веб-ресурсу

Функціональне тестування веб-ресурсу продуктової крамниці є дуже важливим процесом, оскільки воно дозволяє перевірити правильність функціонування різних функцій та можливостей цього ресурсу перед його введенням в експлуатацію.

Функціональне тестування дозволяє перевірити, що всі функції та можливості веб-ресурсу продуктової крамниці працюють правильно, як очікувалося, що забезпечує коректну роботу веб-сайту та надійність для його використання.

Також функціональне тестування дозволяє виявити та виправити помилки веб-ресурсу, що можуть негативно впливати на досвід користувачів, підвищуючи їх задоволеність та вірогідність повернення на сайт в майбутньому.

Для проведення функціонального тестування необхідно розробити протокол тестування.

Протокол тестування – це документ, який містить опис кроків, процедур та результатів тестування програмного продукту, системи чи окремої їх частини. Протокол тестування має на меті систематично задокументувати процес тестування та отримані результати, щоб забезпечити максимальну об'єктивність та повторюваність результатів.

Для функціонального тестування веб-ресурсу продуктової крамниці був розроблений наступний протокол тестування.

TC1 Тест-кейс для додавання нового товару:

* відкрити програму для крамниці;
* обрати опцію «Додати товар»;
* ввести інформацію про товар, таку як ім'я, опис, ціна, категорія тощо;
* натиснути кнопку «Зберегти»;
* переконатись, що новий товар з'явився в списку товарів у програмі.

TC2 Тест-кейс для редагування існуючого товару:

* відкрити програму для крамниці;
* обрати товар, який потрібно змінити;
* натиснути кнопку «Редагувати»;
* змінити потрібні поля (наприклад, ціну або опис);
* натиснути кнопку «Зберегти»;
* переконатись, що зміни збереглися і відображаються правильно.

TC3 Тест-кейс для видалення товару:

* відкрити програму для крамниці;
* обрати товар, який потрібно видалити;
* натиснути кнопку «Видалити»;
* підтвердити, що ви дійсно хочете видалити товар;
* переконатись, що товар більше не відображається в програмі.

TC4 Тест-кейс для пошуку товару:

* відкрити програму для крамниці;
* ввести ключове слово або фразу в поле пошуку;
* натиснути кнопку «Пошук»;
* переконатись, що усі товари, що містять введене ключове слово або фразу, відображаються на екрані.

TC5 Тест-кейс для відображення інформації про товар:

* відкрити програму для крамниці;
* обрати товар, про який потрібно отримати інформацію;
* переконатись, що на екрані відображається інформація про товар, така як ім'я, опис, ціна, категорія тощо.

TC6 Тест-кейс для перевірки наявності товару:

* відкрити програму для крамниці;
* обрати товар, для якого потрібно перевірити наявність;
* переконатись, що на екрані відображається інформація про кількість товару на складі і доступність для покупки.

TC7 Тест-кейс для створення нового замовлення:

* відкрити програму для крамниці;
* обрати товар, який потрібно придбати;
* натиснути кнопку «Додати до кошика»;
* повторити попередні кроки для додавання інших товарів до кошика;
* натиснути кнопку «Оформити замовлення»;
* ввести інформацію про клієнта та адресу доставки;
* переконатись, що інформація про замовлення відображається на екрані.

TC8 Тест-кейс для відстеження стану замовлення:

* відкрити програму для крамниці;
* обрати замовлення, стан якого потрібно відстежувати;
* переконатись, що на екрані відображається інформація про стан замовлення (наприклад, чи він був оплачений, який етап доставки тощо).

TC9 Тест-кейс для додавання нової категорії товарів:

* відкрити програму для крамниці;
* обрати опцію «Додати категорію»;
* ввести інформацію про категорію, таку як ім'я та опис;
* натиснути кнопку «Зберегти»;
* переконатись, що нова категорія з'явилась в списку категорій у програмі.

TC10 Тест-кейс для редагування існуючої категорії товарів:

* відкрити програму для крамниці;
* обрати категорію, яку потрібно змінити;
* натиснути кнопку «Редагувати»;
* змінити потрібні поля (наприклад, опис категорії);
* натиснути кнопку «Зберегти»;
* переконатись, що зміни збереглися і відображаються правильно.

Результати функціонального тестування веб-ресурсу продуктової крамниці наведені в таблиці 3.2.

Як можна побачити з результатів тестування, всі функціональні тести пройдені успішно, що свідчить про те, що фактична поведінка веб-ресурсу співпадає з очікуваною поведінкою, визначеною у функціональних вимогах.

Таблиця 3.2 – Протокол функціонального тестування веб-ресурсу

|  |  |  |  |
| --- | --- | --- | --- |
| Номер тест-кейсу | Очікуваний результат | Фактичний результат | Результат тестування |
| TC1 | Успішне додання нового товару | Новий товар успішно додається до бази даних товарів | Успішно |
| TC2 | Параметри існуючого товару успішно змінюються | Поля запису в базі даних змінюються успішно | Успішно |
| TC3 | Успішне видалення товару | Запис в базі даних успішно видаляється | Успішно |
| TC4 | Успішний пошук товару | Товар знаходиться успішно | Успішно |
| TC5 | Вірне відображення інформації про товар | Відображена інформація відповідає дійсності | Успішно |
| TC6 | Успішна перевірка наявності товару | Наявність товару успішно перевіряється | Успішно |
| TC7 | Успішно створюється нове замовлення | Новий запис до бази даних додається успішно | Успішно |
| TC8 | Отримання статусу замовлення | Статус замовлення показується успішно та відповідає дійсному стану | Успішно |
| TC9 | Успішне створення нової категорії товарів | Нова категорія створюється успішно в базі даних | Успішно |
| TC10 | Зміна існуючої категорії товарів | Категорія товару успішно змінюється | Успішно |

## Інструкція користувача веб-ресурсу

Для забезпечення успішного користувацького досвіду використання розробленого веб-ресурсу необхідно скласти інструкцію користувача, яка включає в себе знімки екранів та пояснювальний текст до кожного знімка. Це допоможе користувачам легко зорієнтуватись у функціоналі веб-ресурсу та ефективно використовувати його.

На рисунку 3.2 зображено вікно створення нового акаунту. Дизайн сторінки виконаний в світлих тонах з використанням контрастних елементів, що робить її зручною для використання.

На сторінці присутня форма реєстрації, яка містить поля для введення основної інформації, такої як ім'я, прізвище, адреса електронної пошти та пароль. Також є кнопка "Зареєструватися", яка відправляє заповнену форму на сервер для обробки. Кнопка входу знаходиться під формою реєстрації та дозволяє користувачам, які вже зареєструвалися, увійти на сайт.

Ця сторінка є важливою для користувачів, оскільки дозволяє їм зареєструватися на сайті та отримати доступ до послуг. Дизайн сторінки є зручним та легким для використання, що сприяє покращенню користувацького досвіду.

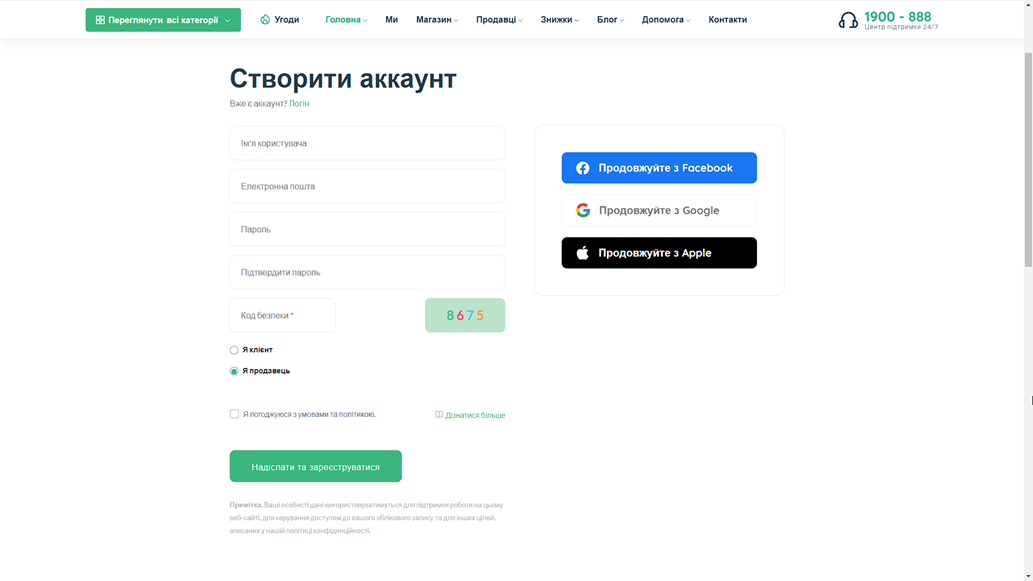


Рисунок 3.2 – Екранна форма створення нового акаунту

На рисунку 3.3 зображене вікно логіну веб-ресурсу. Вона містить форму для введення інформації користувача та кнопку "Увійти", щоб здійснити вхід на сайт.

На сторінці присутня форма входу, яка містить поля для введення ім'я користувача та пароля. Крім того, є кнопка "Увійти", яка дозволяє користувачам здійснити вхід на сайт.

Посилання на сторінку "Забули пароль?" дозволяє користувачам відновити свій пароль, якщо вони його забули. Посилання на сторінку реєстрації дозволяє користувачам зареєструватися на сайті та отримати доступ до послуг.

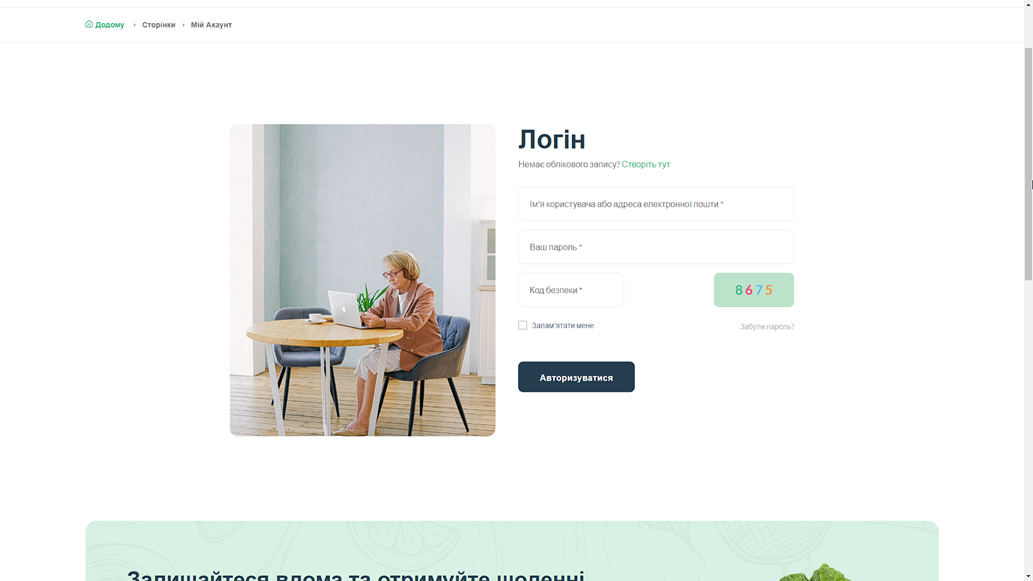


Рисунок 3.3 – Екранна форма логіну

На рисунку 3.4 зображена екранна форма головного вікна веб-ресурсу. Вона містить інформацію про продукти та надає користувачам можливість знайти потрібну інформацію та здійснити необхідні дії.

Шапка сайту містить логотип компанії, навігаційне меню та кнопку "Увійти", щоб здійснити вхід на сайт. Банер містить промо-контент компанії та посилання на послуги. Секції продуктів містять короткий опис та зображення продуктів, що продає компанія.

Секція "Про нас" містить інформацію про компанію та її місію. Секція "Наші клієнти" містить відгуки та рекомендації клієнтів компанії. Футер сайту містить навігаційне меню, контактну інформацію та посилання на соціальні мережі. Секція "Наші клієнти" містить відгуки та рекомендації клієнтів компанії. Секція "Останні новини" містить оновлення та новини компанії.

Ця сторінка є важливою для користувачів, оскільки вона надає інформацію про продукти компанії та дозволяє здійснити необхідні дії, такі як вхід на сайт, пошук інформації про продукти, зв'язок з компанією та інше.

Дизайн сторінки є зручним та легким для використання, що сприяє покращенню користувацького досвіду.

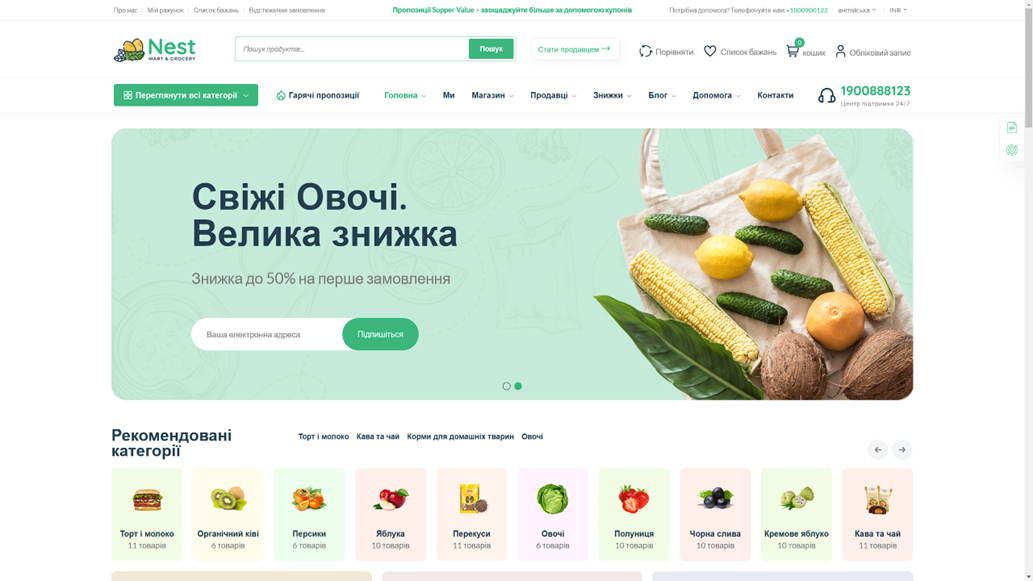


Рисунок 3.4 – Екранна форма головного вікна веб-ресурсу

На рисунку 3.5 зображена екранна форма переліку товарів певної категорії. На сторінці розміщена таблиця товарів, що дозволяє користувачам переглядати інформацію про товари, такі як назва товару, зображення, ціна та опис. Дизайн сторінки привабливий та легкий для сприйняття. На сторінці присутні наступні елементи:

* шапка сайту, яка містить логотип компанії, навігаційне меню та кнопку "Кошик", щоб здійснити перехід до кошика покупок;
* список категорій товарів, який містить список категорій товарів, які є доступними на сайті;
* фільтри товарів, які дозволяють користувачам фільтрувати товари за різними параметрами, такими як ціна, розмір, колір та інші;
* список товарів, який містить зображення, назву та ціну товарів.

Механізм пагінації дозволяє користувачам переходити до наступних сторінок, якщо на сайті доступний великий асортимент товарів. Футер сайту містить навігаційне меню, контактну інформацію та посилання на соціальні мережі. Крім того, на сторінці присутні фільтри, що дозволяють користувачам знайти товари, які відповідають їхнім потребам. Дизайн сторінки є зручним та легким для використання, що сприяє покращенню користувацького досвіду.

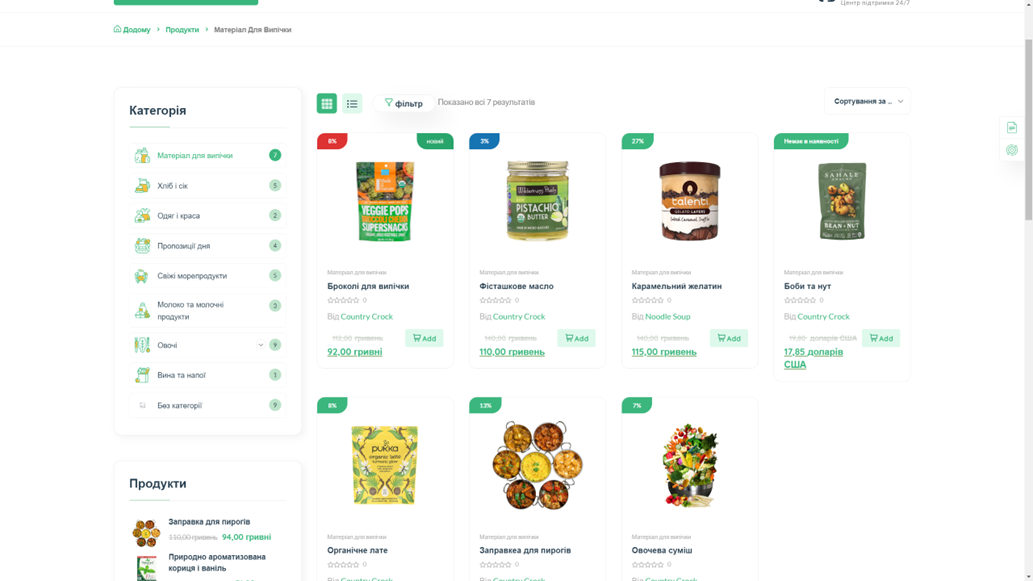


Рисунок 3.5 – Екранна форма переліку продуктів певної категорії

На рисунку 3.6 зображена екранна форма вікна продукту. На сторінці розміщена детальна інформація про обраний товар, така як назва, зображення, ціна, опис, характеристики та відгуки користувачів.

На сторінці присутні картка товару, яка містить зображення, назву, ціну, опис та характеристики товару. Крім того, на сторінці присутня можливість перегляду зображення товару в різних ракурсах за допомогою каруселі зображень. Це дозволяє користувачам отримати більш детальний огляд товару перед покупкою.

Кнопка "Додати у кошик" дозволяє користувачам додавати товар до кошика покупок. Секція "Відгуки користувачів" містить відгуки користувачів про товар, які допомагають іншим користувачам прийняти рішення про покупку. Футер сайту містить навігаційне меню, контактну інформацію та посилання на соціальні мережі.

Структура сторінки досить логічна та зрозуміла, що дозволяє користувачам легко зорієнтуватися та знайти необхідну інформацію. Дизайн сторінки є досить мінімалістичним, що сприяє увазі на основну інформацію та забезпечує більш зручний користувацький досвід.

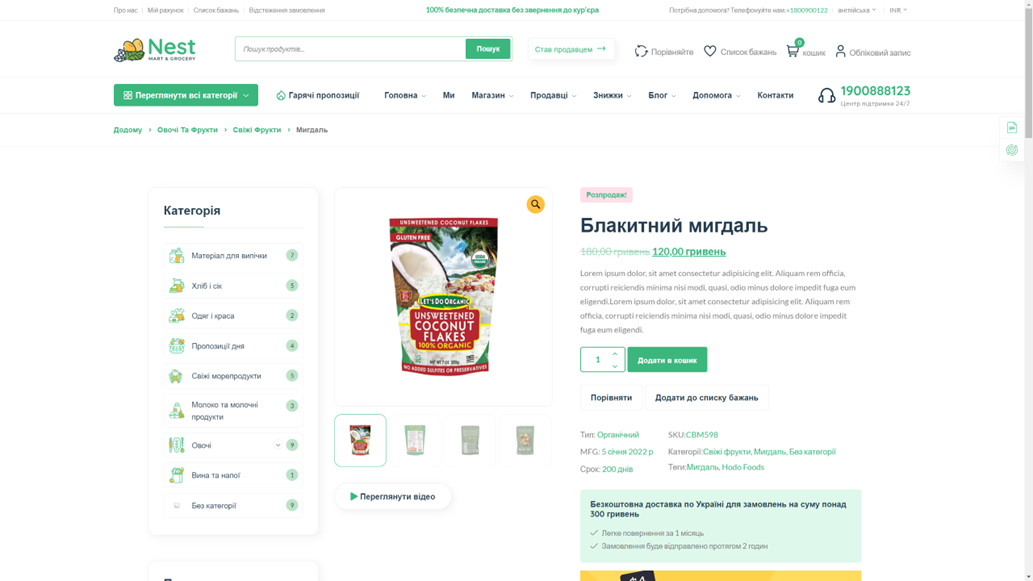


Рисунок 3.6 – Екранна форма вікна продукту

На рисунку 3.7 зображена екранна форма вікна кошику товарів. На цій сторінці користувач може переглянути всі товари, які він додав до кошика, зокрема їх зображення, назву, ціну, кількість та загальну вартість замовлення.

Крім того, на сторінці присутня можливість редагування кількості товару, видалення товару з кошика, а також використання промокодів та отримання знижок.

На сторінці присутні важливі елементи для успішного оформлення замовлення, такі як кнопка "Оформити замовлення", яка перенаправляє користувача на сторінку оформлення замовлення, а також інформаційний блок про доставку та оплату замовлення.

Сторінка має адаптивний дизайн, що дозволяє користувачам зручно переглядати її на різних пристроях та розмірах екранів.

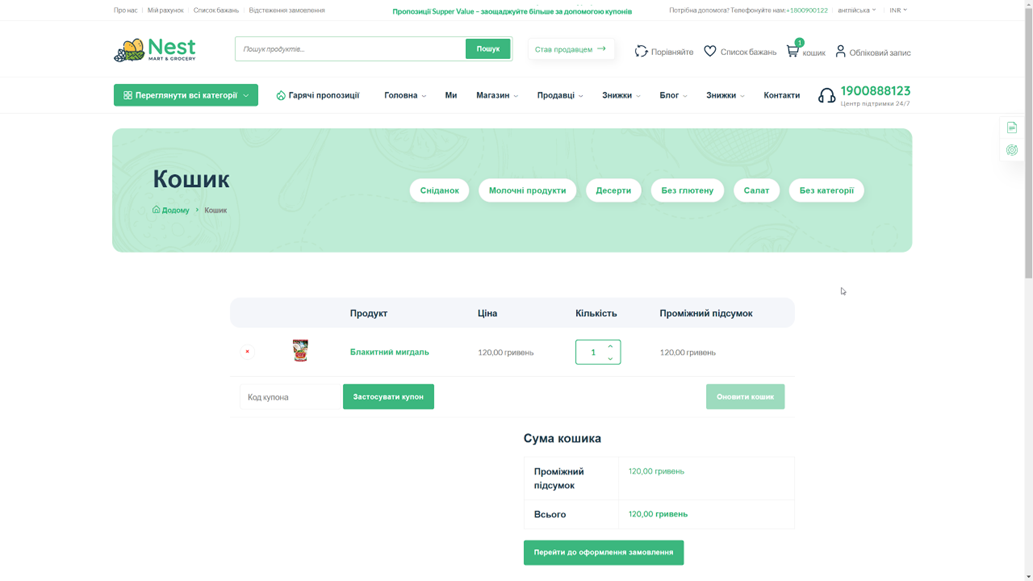


Рисунок 3.7 – Екранна форма вікна кошику товарів

На рисунку 3.8 зображена екранна форма вікна оформлення замовлення. На цій сторінці користувачі можуть оформити своє замовлення, заповнивши необхідну інформацію про себе та свої покупки.

Сторінка містить різні елементи, необхідні для успішного оформлення замовлення, такі як форма для введення інформації про замовлення, включаючи адресу доставки та спосіб оплати.

Крім того, сторінка містить блок з інформацією про вартість замовлення, включаючи вартість товарів, податки та доставку.

Користувачам також доступні різноманітні опції, такі як використання купонів та збереження адрес доставки для майбутніх покупок.

На сторінці є кнопка "Підтвердити замовлення", яка дозволяє користувачам підтвердити та оформити своє замовлення. Дизайн сторінки є простим та зрозумілим, зроблений в мінімалістичному стилі.

Після того, як користувач створив замовлення, він може відстежувати статус замовлення у відповідному вікні профілю користувача, до якого можна перейти з головного меню сайту, яке знаходиться у шапці кожної сторінки веб-ресурсу.

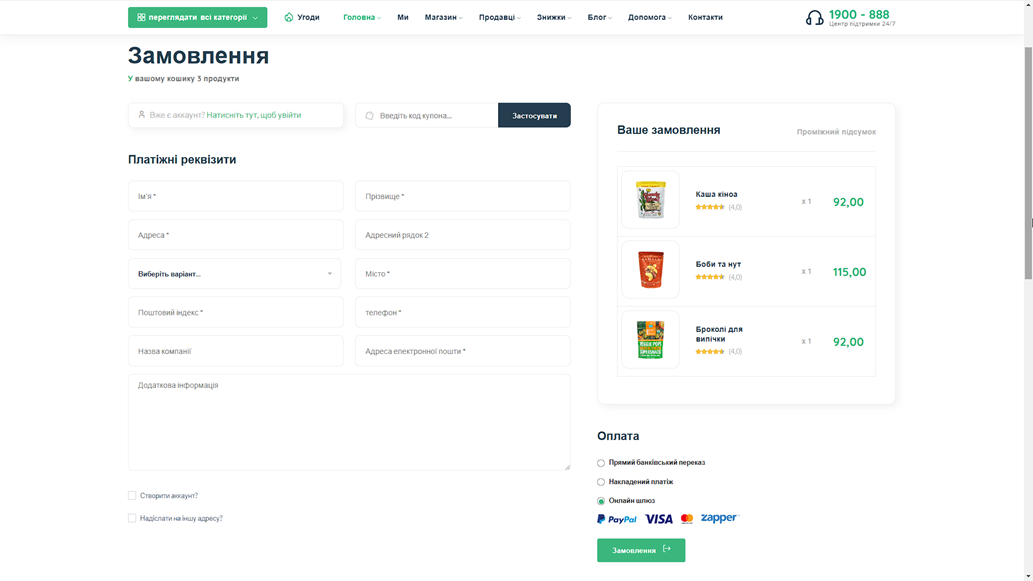


Рисунок 3.8 – Екранна форма оформлення замовлення

## Вихідний код веб-ресурсу

Вихідний код програмних класів розробленого веб-ресурсу наведений в Додатку А даної роботи.

## Висновки до третього розділу

В даному розділі курсової роботи було проведено програмну реалізацію веб-ресурсу продуктової крамниці на базі фреймворку Angular. Для того, щоб провести програмну реалізацію, був використаний проєкт веб-ресурсу, який був розроблений в другому розділі даної роботи.

Перш за все, була розглянута структура серверного програмного проєкту. Були визначені основні програмні компоненти, з яких складається програмний проєкт веб-ресурсу, а також наданий перелік основних класів, які відносяться до цих компонентів.

Далі була розроблена концептуальна діаграма класів, яка дозволяє визначити основні класи веб-ресурсу та визначити взаємовідносини між цими класами.

Окремо було розглянуто питання використання системи контроля версій для упорядкування розробки веб-ресурсу. Були визначені основні репозиторії проєкту та надані основні показники метрик цих репозиторіїв.

Далі було проведено функціональне тестування розробленого веб-ресурсу. Був розроблений протокол тестування у вигляді множини тест-кейсів та проведено тестування. Було визначено, що всі тест-кейси були пройдені, а отже фактична поведінка веб-ресурсу співпадає з очікуваною поведінкою, визначеною у функціональних вимогах.

Також була розроблена інструкція користувача у вигляді множини знімків екрану та пояснювального тексту, який визначає функціонал додатку у даному вікна та можливі дії користувача. Крім того, наданий вихідний код розробленого веб-ресурсу.

# ВИСНОВКИ

У даній курсовій роботі був розроблений веб-ресурс продуктової крамниці на базі фреймворку Angular.

Такий веб-ресурс дозволяє користувачам переглядати каталог товарів в магазині, робити онлайн-замовлення товарів, а також залишати відгуки. Таким чином, мета, поставлена перед даною курсовою роботою досягнута в повному обсязі. Для досягнення мети були вирішені наступні задачі.

У першому розділі даної курсової роботи була детально розглянута предметна область створення веб-ресурсу для продуктової крамниці з використанням фреймворку Angular. Були визначені основні завдання, які повинні бути вирішені в процесі розробки веб-ресурсу продуктової крамниці. Був проведений аналіз існуючих аналогів та визначені ключові вимоги до створення власного веб-ресурсу. У якості технологій для розробки клієнтської частини був обраний Angular, для розробки серверної частини – Java Spring. Також, було вирішено використовувати PostgreSQL в якості СУБД для даного веб-ресурсу.

У другому розділі роботи було проведено проектування веб-ресурсу. Була визначена мета сайту, його потенційна аудиторія та основні можливості. Визначені основні вимоги до веб-ресурсу, включаючи функціональні та нефункціональні. Також була розроблена діаграма сценаріїв, діаграма інформаційних потоків, діаграма станів. Крім того, була розроблена діаграма логічного уявлення системи, а також діаграма розгортання веб-ресурсу. Також була визначена схема даних системи та наданий опис ключовим технологіям.

В третьому розділі роботи було проведено програмну реалізацію веб-ресурсу. Розглянута структура серверного програмного проєкту, визначені основні програмні компоненти, розроблена діаграма класів. Розглянуто питання використання системи контроля версій для упорядкування розробки веб-ресурсу. Проведено функціональне тестування, розроблена інструкція користувача у вигляді множини знімків екрану та пояснювального тексту та наданий вихідний код розробленого веб-ресурсу.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The importance of information technology in retail. URL: https://www.retailcouncil.org/community/technology/the-importance-of-information-technology-in-retail/ (дата звернення 01.03.2023).
2. The Importance of Information Technology in Retail. URL: https://www.aeologic.com/blog/the-importance-of-information-technology-in-retail/ (дата звернення 01.03.2023).
3. Fozzyshop. URL: https://fozzyshop.ua/odesa/ (дата звернення 01.03.2023).
4. Інтернет-магазин продуктів «Таврія В». URL: https://www.tavriav.ua/ (дата звернення 01.03.2023).
5. Онлайн-супермаркет «Сільно». URL: https://silpo.ua/ (дата звернення 01.03.2023).
6. Angular is a platform for building mobile and desktop web applications. URL: https://angular.io/ (дата звернення 01.03.2023).
7. Angular components overview. URL: https://angular.io/guide/component-overview (дата звернення 01.03.2023).
8. The Good and the Bad of Angular Development. URL: https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/ (дата звернення 01.03.2023).
9. Node.js – an open-source, cross-platform JavaScript runtime environment. URL: https://nodejs.org/en (дата звернення 01.03.2023).
10. Django: The web framework for perfectionists with deadlines. URL: https://www.djangoproject.com/ (дата звернення 01.03.2023).
11. Ruby on Rails — A web-app framework. URL: https://rubyonrails.org/ (дата звернення 01.03.2023).
12. Laravel - The PHP Framework For Web Artisans. URL: https://laravel.com/ (дата звернення 01.03.2023).
13. Spring | Home. URL: https://spring.io/ (дата звернення 01.03.2023).
14. Express – fast, unopinionated, minimalist web framework for Node.js. URL: https://expressjs.com/ (дата звернення 01.03.2023).
15. Welcome to Flask. URL: https://flask.palletsprojects.com/ (дата звернення 01.03.2023).
16. ASP.NET – open-source web framework for .NET infrastructure. URL: https://dotnet.microsoft.com/en-us/apps/aspnet (дата звернення 01.03.2023).
17. Java spring framework – pros, cons, common mistakes. URL: https://bit.ly/3A1dS4w (дата звернення 01.03.2023).
18. Key Components and Internals of Spring Boot Framework. URL: https://bit.ly/3ojMdtb (дата звернення 01.03.2023).
19. MySQL Home. URL: https://www.mysql.com/ (дата звернення 01.03.2023).
20. PostgreSQL: The world's most advanced open source database. URL: https://www.postgresql.org/ (дата звернення 01.03.2023).
21. MongoDB Atlas Database - The Database For Modern Apps. URL: https://bit.ly/414ADjX (дата звернення 01.03.2023).
22. Industry-leading performance and security with SQL Server 2019. URL: https://www.microsoft.com/en-us/sql-server/sql-server-2019?rtc=1 (дата звернення 01.03.2023).
23. Database | Oracle. URL: https://www.oracle.com/database/ (дата звернення 01.03.2023).
24. SQLite Home Page. URL: https://sqlite.org/index.html (дата звернення 01.03.2023).
25. Amazon RDS Features. URL: https://aws.amazon.com/rds/features/?nc1=h\_ls (дата звернення 01.03.2023).
26. Firebase Home. URL: https://firebase.google.com/ (дата звернення 01.03.2023).