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 веб-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 ведення щоденних витрат, щоб забезпечити користувачам простий, зручний і надійний інструмент для контролю своїх фінансових витрат. Він має допомогти користувачам визначити основні джерела витрат та ефективно розподіляти ці витрати. Це дозволить їм краще розуміти свої фінансові витрати та здійснювати розумніші фінансові рішення, що допоможе їм досягти своїх фінансових цілей. Основною метою цього проекту є створення простого, ефективного та інтуїтивно зрозумілого інструменту, який дозволяє користувачам легко створювати, редагувати, видаляти та відслідковувати свої витрати. В ході розробки додатку будуть використані сучасні технології та практики розробки веб-додатків, зокрема мова програмування Java з використанням фреймворку Spring Framework та його розширенням Spring Boot, Spring Data JPA, шаблонизатор Thymeleaf та архітектура MVС.

ABSTRACT

The course work is devoted to the development of a web application for keeping records of personal expenses. The application is designed to keep track of daily expenses to provide users with a simple, convenient and reliable tool for controlling their financial expenditures. It should help users to identify the main sources of expenses and allocate them efficiently. This will allow them to better understand their financial expenses and make smarter financial decisions, which will help them achieve their financial goals. The main goal of this project is to create a simple, efficient and intuitive tool that allows users to easily create, edit, delete and track their expenses. The application will be developed using modern web application development technologies and practices, including the Java programming language using the Spring Framework and its Spring Boot extension, Spring Data JPA, Thymeleaf templating engine and MVC architecture.

ВСТУП

Зважаючи на зростаючу важливість фінансової грамотності та управління особистими фінансами, веб-додатки для ведення обліку особистих витрат стають дедалі популярнішими. Такі додатки надають користувачам зручні інструменти для відстеження та аналізу своїх фінансових витрат.

Управління витратами є важливою складовою фінансового благополуччя. Веб-додаток для ведення щоденних витрат дозволить користувачам точно відстежувати, куди йдуть їх гроші, та аналізувати свої витрати. Це допоможе їм більш свідомо планувати свої фінанси, знаходити можливості для економії та приймати розумні фінансові рішення.

Веб-додаток для ведення витрат має надавати простий та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам з легкістю вносити та категоризувати свої витрати. Додаток повинен також надавати зручні функції аналізу та статистики, які допоможуть користувачам отримати цінну інформацію про свої фінанси.

Використання сучасних технологій та розробка веб-додатку на базі стабільних та ефективних фреймворків дозволять забезпечити надійну та швидку роботу додатку. Програмування на веб-платформі надає можливість забезпечити доступ до даних з будь-якого пристрою з підключенням до Інтернету, що робить додаток зручним та доступним для користувачів в будь-який час і в будь-якому місці.

Розробка веб-додатка для ведення щоденних витрат вимагає уважного планування, дизайну та програмування. Налагодження зв'язку з базою даних, реалізація функцій керування витратами та статистики, а також забезпечення безпеки даних - це важливі етапи процесу розробки.

Зрозуміння основних потреб користувачів і розробка зручного веб-додатку для ведення особистих витрат допоможуть користувачам краще контролювати свої фінанси, планувати витрати та досягати своїх фінансових цілей.

Для досягнення цієї мети необхідно виконати наступні задачі:

* здійснити огляд інших додатків для створення користувацьких заміток;
* проаналізувати та обрати сучасні технології для створення веб додатку;
* провести проєктування веб додатку своїх фінансових витрат;
* виконати програмну реалізацію спроєктованого веб 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 веб 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 ВЕБ-додатку для ведення обліку особистих витрат

## Мета та задачі веб-додатку

Мета веб-додатку для ведення обліку особистих витрат полягає в створенні зручного, ефективного і надійного інструменту для користувачів, що дозволить їм легко відстежувати та контролювати свої фінансові витрати. Основною метою додатку є надання користувачам зрозумілого та зручного інтерфейсу, що спрощує процес ведення обліку та аналізу особистих витрат. Він повинен бути доступним для використання всім, незалежно від рівня їхньої фінансової грамотності. Кожен користувач має мати свій окремий обліковий запис зі списком його витрат. Додаток має забезпечувати можливість додавання, редагування та видалення усіх категорій витрат для всіх користувачів. Додаток повинен також надавати зручний та інтуїтивно зрозумілий інтерфейс для введення даних про витрати. Користувачі повинні мати змогу переглядати статистику своїх витрат та доходів за певний період часу, яка повинна бути представлена у зручному та зрозумілому вигляді.

Задачі веб-додатку можуть включати:

* Реєстрація та авторизація користувачів: додаток повинен забезпечувати можливість реєстрації нових користувачів та авторизацію вже існуючих користувачів для доступу до особистого обліку витрат.
* Додавання та відстеження витрат: користувачі повинні мати можливість додавати нові записи про свої витрати, вказуючи категорію, суму та дату. Додаток повинен зберігати ці записи та надавати зручний спосіб їх подальшого перегляду та аналізу.
* Категоризація витрат: додаток може надавати можливість створення різних категорій витрат (наприклад, їжа, транспорт, розваги) для більш зручного організування та фільтрації записів про витрати.
* Аналіз та статистика витрат: веб-додаток може надавати функціонал для аналізу та візуалізації витрат, дозволяючи користувачам переглядати статистику за певний період часу, визначати основні джерела витрат та встановлювати фінансові цілі.
* Нагадування та планування: додаток може надавати можливість встановлення нагадувань про платежі, регулярні витрати або фінансові цілі, що допомагатимуть користувачам вчасно планувати та керувати своїми фінансами.
* Забезпечення безпеки даних: веб-додаток повинен забезпечувати захист особистих фінансових даних користувачів, використовуючи відповідні методи шифрування та заходи безпеки.

В цілому, мета веб-додатку для ведення обліку особистих витрат полягає в наданні користувачам зручного та ефективного інструменту для контролю своїх фінансових витрат, а задачі додатку включають створення зручного інтерфейсу, збереження та аналіз записів про витрати, категоризацію, аналіз та планування фінансів, а також забезпечення безпеки даних користувачів.

## 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.

Акторами веб-додатку є «Користувач» та «Гість»:

* Користувач: може додавати нові категорії витрат, додавати нові витрати, переглядати свої витрати та доходи за певний період, редагувати та видаляти свої записи, переглядати статистику тощо, редагувати налаштування.
* Гість: має доступ лише до деяких функцій додатку, таких, наприклад, як ознайомлення з можливостями додатку та можливість реєстрації як користувача в додатку.

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.

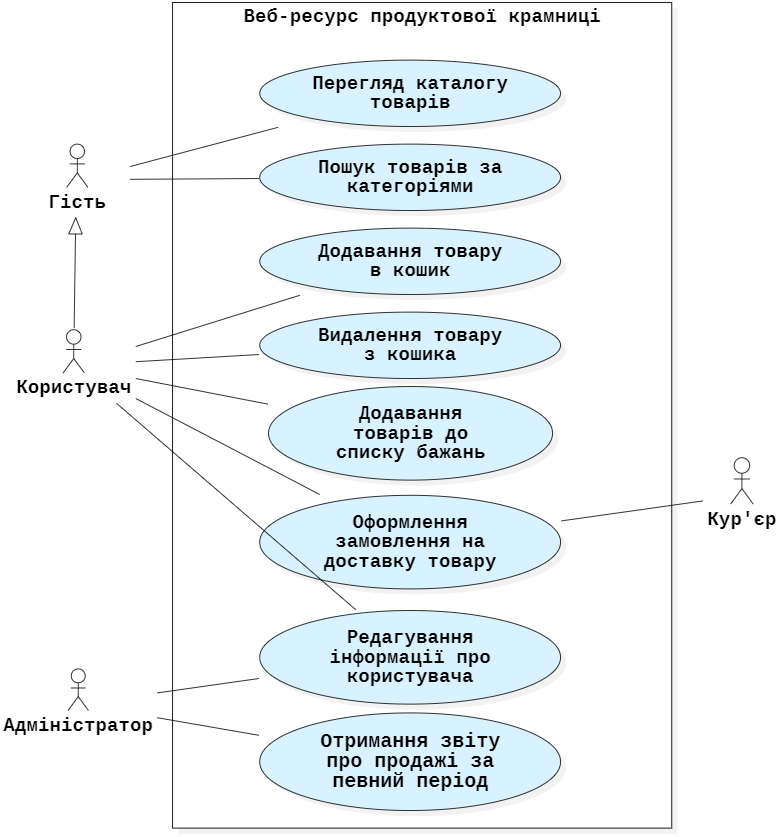


Рисунок 2.1 – Діаграма варіантів використання мобільного додатку

## Формування користувацьких історій веб-додатку

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

US1 Як гість, я хочу мати можливість ознайомитись з можливостями додатку, щоб зрозуміти, чи цей додаток відповідає моїм потребам.

Критерії прийнятності:

* На вітальній сторінці присутній перелік можливостей додатку з коротким описом кожної з них.
* Користувач може детальніше ознайомитись з кожною можливістю, натиснувши на неї.
* Після ознайомлення з можливостями додатку користувач може перейти до реєстрації або авторизації.

Деталі історії:

* Гість, відкриває веб-додаток та потрапляє на вітальну сторінку.
* Гість бачить перелік можливостей додатку та короткий опис кожної з них.
* Гість може детальніше ознайомитись з кожною з можливостей, натиснувши на неї.
* Після ознайомлення з можливостями додатку, гість може перейти до реєстрації або авторизації для доступу до повного функціоналу додатку.

US2 Я, як гість, хочу мати можливість зареєструвати свій аккаунт в додатку.

Критерії прийнятності:

* Користувач може ввести свої особисті дані (ім'я, електронну пошту та пароль);
* Система перевіряє правильність введених даних;
* Користувач отримує підтвердження про успішну реєстрацію та може увійти до свого аккаунту.

Деталі історії:

* Користувач запускає додаток;
* Користувач вибирає опцію реєстрації;
* Користувач вводить свої особисті дані (ім'я, електронну пошту та пароль);
* Система перевіряє правильність введених даних;
* Якщо дані введені правильно, система створює новий аккаунт;
* Користувач отримує підтвердження про успішну реєстрацію та може увійти до свого аккаунту.

US3 Я, як користувач, хочу мати можливість корегувати свій аккаунт в додатку.

Критерії прийнятності:

* Користувач може змінювати свої особисті дані (ім'я, електронну пошту та пароль);
* Система перевіряє правильність введених даних;
* Користувач отримує підтвердження про успішне оновлення даних.

Деталі історії:

* Користувач входить у свій аккаунт;
* Користувач обирає опцію "Змінити дані";
* Користувач вводить нові дані;
* Система перевіряє валідність введених даних та їх унікальність;
* Після успішної зміни даних користувача система перенаправляє його на домашню сторінку.

US4 Я, як користувач, хочу мати можливість додавати нові витрати до свого списку.

Критерії прийнятності:

* Користувач може ввести назву витрати та її вартість;
* Введені дані зберігаються в базі даних додатку;

Деталі історії:

* Користувач входить у свій обліковий запис у додатку;
* Користувач вибирає опцію "Додати витрати";
* Користувач вибирає дату.
* Користувач вводить назву витрати та її вартість;
* Після введення даних, користувач натискає кнопку "Додати";
* Система зберігає дані у базі даних;

US5 Я, як користувач, хочу мати можливість редагувати існуючі витрати у своєму списку.

Критерії прийнятності:

* Користувач може редагувати будь-яку витрату у своєму списку.
* Користувач може змінити назву, дату, суму та опис витрати.
* Зміни повинні бути збережені у базі даних і відображені в списку витрат.

Деталі історії:

* Користувач входить у свій обліковий запис у додатку та переходить до списку витрат.
* Користувач обирає витрату, яку бажає відредагувати та натискає на неї.
* Користувач змінює необхідні поля та натискає на кнопку "Зберегти зміни".
* Система перевіряє коректність введених даних та зберігає зміни у базі даних.
* Список витрат оновлюється та показує нові дані про відредаговану витрату.

US6 Я, як користувач, хочу мати можливість видаляти витрати зі свого списку.

Критерії прийнятності:

* Користувач може видалити будь-яку витрату зі свого списку
* Підтвердження видалення відображається користувачеві перед остаточним видаленням витрати
* Після видалення витрати, вона більше не відображається у списку витрат користувача.

Деталі історії:

* Користувач відкриває список своїх витрат у додатку.
* Користувач виділяє витрату, яку бажає видалити зі списку.
* Додаток показує повідомлення для підтвердження видалення витрати.
* Користувач натискає на кнопку "Підтвердити видалення".
* Витрата видаляється зі списку витрат користувача і більше не відображається у додатку.

US7 Я, як користувач, хочу мати можливість переглядати свій список витрат за певний період.

Критерії прийнятності:

* Користувач може вибрати період для перегляду витрат (наприклад, за останній місяць, тиждень, день тощо);
* Список витрат відображається вірно і зрозуміло, включаючи назву, суму витрат, дату тощо;
* Користувач може переглядати свій список.

Деталі історії:

* Користувач входить до свого облікового запису в додатку;
* Користувач вибирає опцію "Переглянути список витрат" або подібну опцію на головному екрані додатку;
* Додаток показує список всіх витрат, які були занесені користувачем;
* Користувач може використовувати фільтри, щоб переглядати витрати за конкретний період (наприклад, за тиждень, за місяць, за рік тощо);
* Користувач може сортувати витрати за різними параметрами, наприклад, за датою або за сумою;
* Користувач може переглядати деталі кожної витрати, такі як назва, дата, сума тощо;
* Користувач може закрити список витрат та повернутися на головний екран додатку.

US8 Я, як користувач, хочу мати можливість налаштовувати бюджетні обмеження та ліміти.

Критерії прийнятності:

* Користувач може встановлювати бюджетні обмеження та ліміти на період.
* Встановлені бюджетні обмеження та ліміти відображаються на дашборді користувача.
* Користувач може змінювати встановлені бюджетні обмеження та ліміти.

Деталі історії:

* Користувач входить в додаток і переходить до налаштувань.
* В налаштуваннях користувач вибирає опцію "Ліміти витрат".
* Користувач вводить максимальну суму, яку він може витрачати за день, тиждень, місяць або інший встановлений період.
* Користувач вибирає, яким чином він отримуватиме сповіщення, якщо його витрати перевищують встановлені ліміти (повідомлення в додатку, електронна пошта, повідомлення на телефоні тощо).
* Встановлені бюджетні обмеження та ліміти відображаються на дашборді користувача.
* Користувач може змінювати встановлені бюджетні обмеження та ліміти на сторінці "Налаштування".

US9 Я, як користувач, хочу мати можливість отримувати сповіщення, якщо я перевищую ліміти.

Критерії прийнятності:

* Користувач може налаштувати ліміти для своїх витрат.
* Користувач може вказати, яким чином отримувати сповіщення про перевищення лімітів (повідомлення в додатку, електронна пошта, повідомлення на телефоні тощо).
* Користувач отримує сповіщення, якщо його витрати перевищують встановлені ліміти.
* Оповіщення повинні бути достатньо частими, щоб користувач міг вчасно реагувати на перевищення ліміту.

Деталі історії:

* Якщо користувач витратив більше встановленого ліміту, він отримує сповіщення відповідно до обраного способу повідомлення.
* Користувач може переглянути статистику своїх витрат за певний період, щоб визначити, чому він перевищив ліміт і як може заощадити гроші у майбутньому.
* Може відображати кольором наближення витрат до лимітів

## 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.

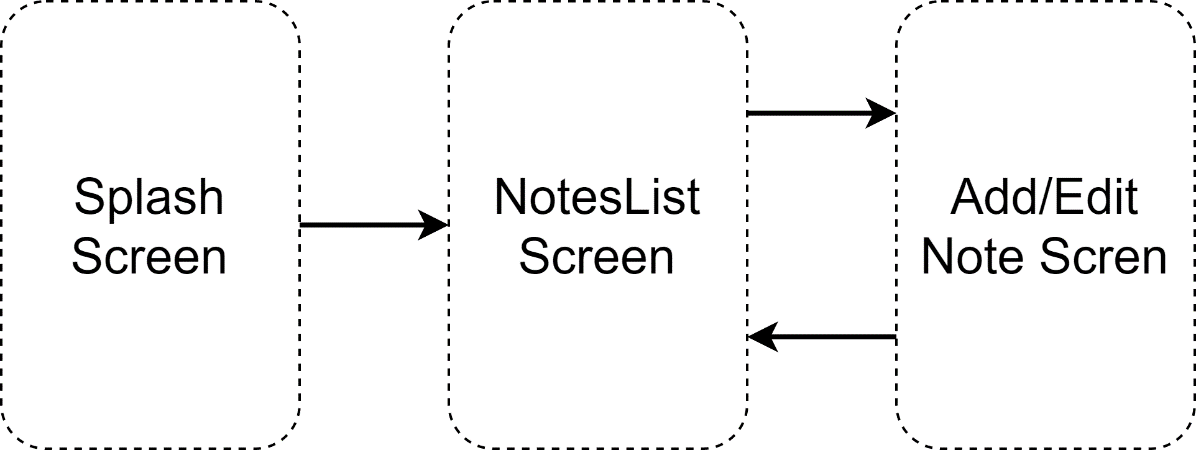


Рисунок 2.2 – Навігаційний граф мобільного додатку

## 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.

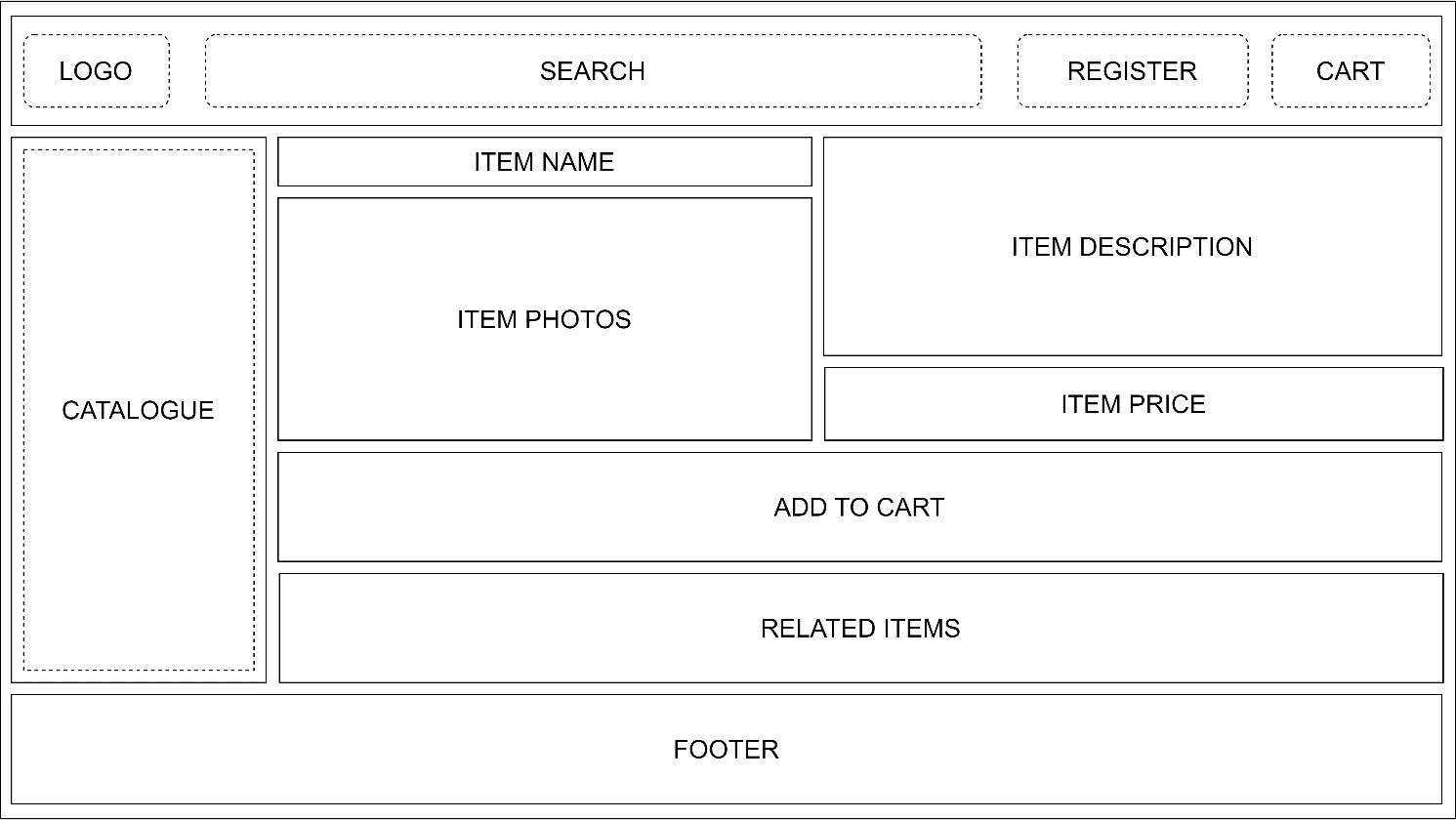


Рисунок 2.5 – Макет сторінки товару веб-ресурсу

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.

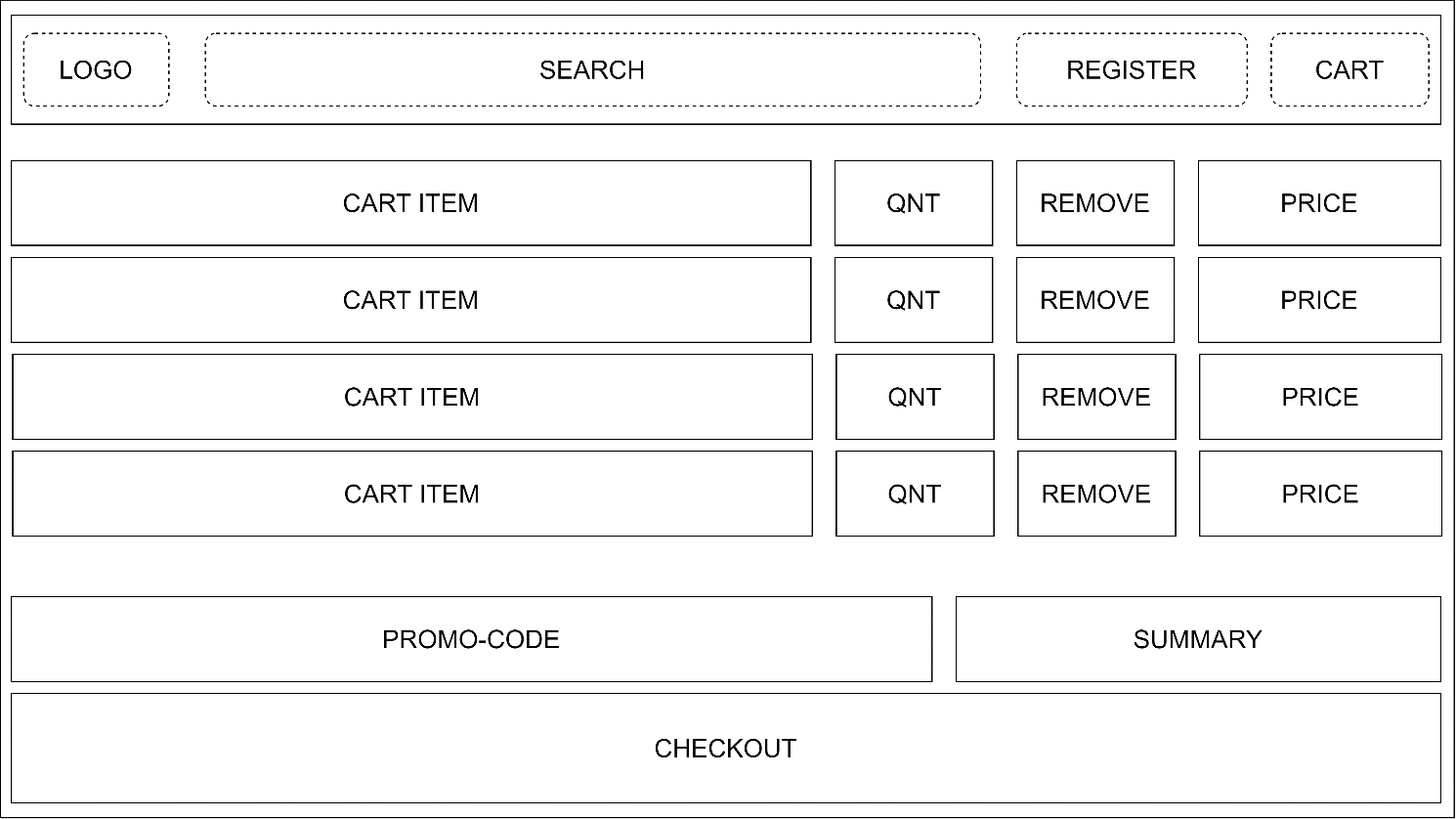


Рисунок 2.6 – Макет вікна кошику товарів веб-ресурсу

## 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 ВЕБ-додатку для ведення щоденних витрат

## Структура серверного програмного проекту

У процесі розробки додатку використані такі сучасні технології та практики розробки веб-додатків, як мова програмування Java, фреймворк Spring Framework з використанням розширення Spring Boot, Spring Data JPA, шаблонізатор Thymeleaf та архітектурний шаблон MVС. Використання стеку Spring Boot дозволить нам швидко розробляти та підтримувати веб-додаток. Ми будемо використовувати Spring Data JPA для зберігання та отримання даних з бази даних, а шаблонізатор Thymeleaf дозволить нам легко створювати динамічні HTML сторінки. Застосування архітектурного шаблону MVС дозволить нам розділити логіку додатку на моделі, представлення та контролери для забезпечення кращої організації коду та розширюваності.

Такий стек технологій є популярним та добре підтримується, тому я впевнений, що наприкінці вийде потужний та надійний додаток для наших користувачів.

Структура проекту DailyExpenses має наступний вид:

\dailyexpenses

\controllers

CategoryController.java

ExpensesController.java

UserController.java

\models

Category.java

Expense.java

User.java

\repositories

CategoryRepository.java

ExpenseRepository.java

UserRepository.java

\services

CategoryService.java

CategoryServiceImpl.java

ExpenseService.java

ExpenseServiceImpl.java

UserService.java

UserServiceImpl.java

DailyExpensesApplication.java

Компоненти додатку включають контролери, які відповідають за обробку запитів користувачів та повернення відповідей у форматі HTML сторінок. У проекті вже присутні такі контролери:

* ExpensesController.java, який відповідає за роботу з витратами, їх додавання, видалення та редагування, а також за відображення домашньої сторінки веб-сайту;
* UserController.java, який відповідає за роботу з користувачами, тобто відображення сторінок входу та реєстрації, редагування профілю користувача та інші функції, що пов'язані з аутентифікацією та авторизацією користувачів.
* CategoryController.java, який відповідає за роботу з категоріями витрат.

Моделі включають об'єкти, які представляють дані, що використовуються в системі. В проекті присутні такі моделі:

* Category.java, який представляє категорію витрат.
* Expense.java, який представляє витрату.
* User.java, який представляє користувача.

Репозиторії використовуються для зберігання та отримання даних з бази даних. У проекті вже присутні такі репозиторії:

* CategoryRepository.java, який відповідає за збереження та отримання даних про категорії витрат.
* ExpenseRepository.java, який відповідає за збереження та отримання даних про витрати.
* UserRepository.java, який відповідає за збереження та отримання даних про користувачів.

Сервіси використовуються для виконання бізнес-логіки та обробки даних. У проекті вже присутні такі сервіси:

* CategoryService.java та CategoryServiceImpl.java, які відповідають за логіку категорій витрат.
* ExpenseService.java та ExpenseServiceImpl.java, які відповідають за логіку витрат.
* UserService.java та UserServiceImpl.java, які відповідають за логіку користувачів.

Основним файлом додатку є DailyExpensesApplication.java, який містить точку входу до додатку.

## Інструкція користувача веб-ресурсу

На рисунку 3.1 зображена екранна форма початкової сторінки веб-додатку. Вона містить ознайомлювальну інформацію щодо додатка.

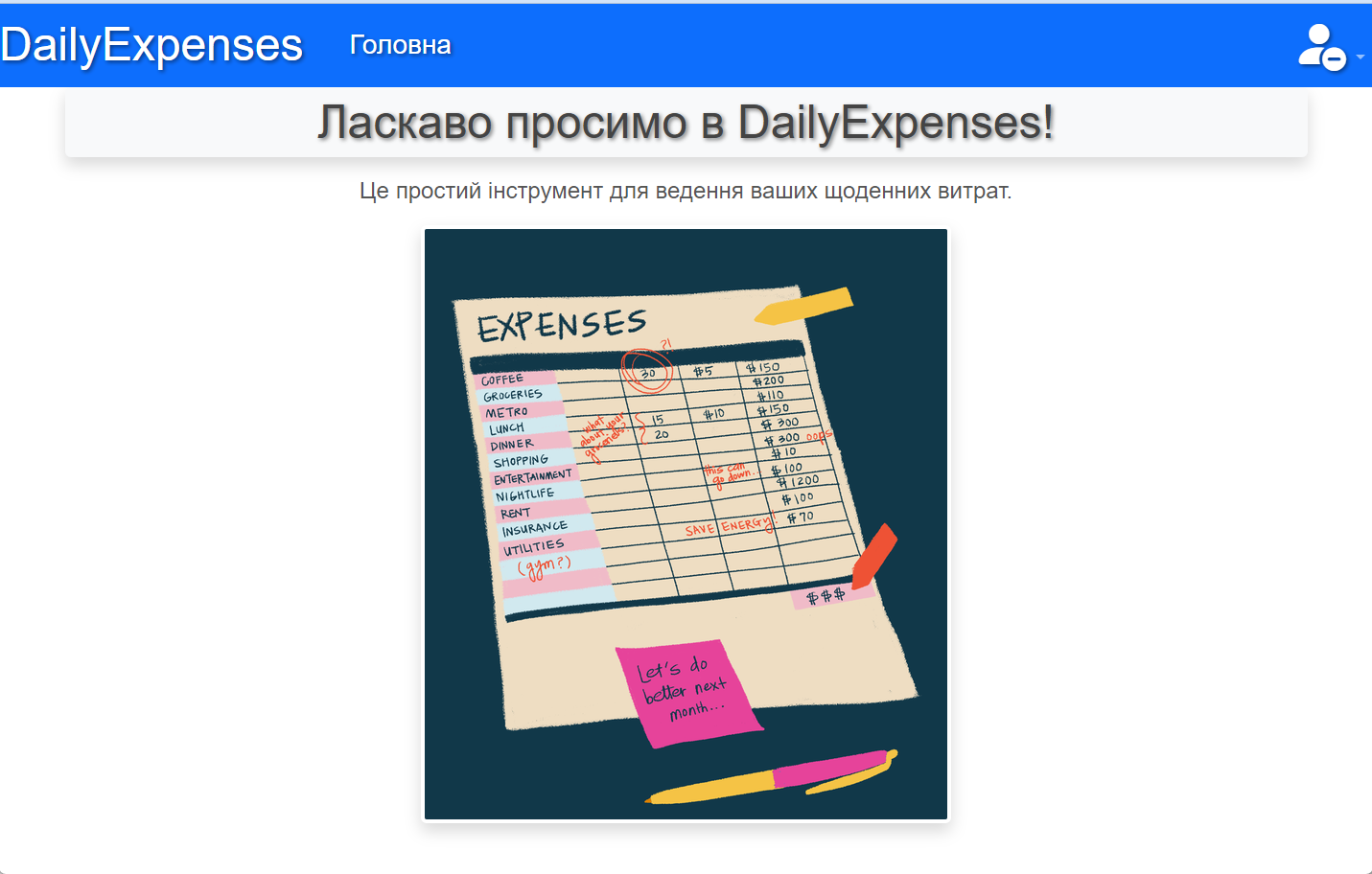
Шапка сайту містить логотип додатку та початкове навігаційне меню, яке дозволить користовачу або увійти до додатку, або зареєструватись.

Рисунок 3.1 – Екранна форма початкового вікна веб-додатку

На рисунку 3.2 зображено меню неавторизованого користувача, яке дозволяє перейти на сторінки авторізації або реєстрації користувача.

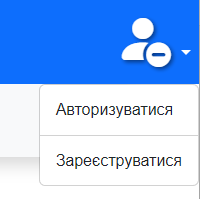


Рисунок 3.2 – меню неавторизованого користувача

На рисунку 3.3 зображена сторінка з формою авторизації користувача, яка дозволяє увійти до додатку під своїм логіном.

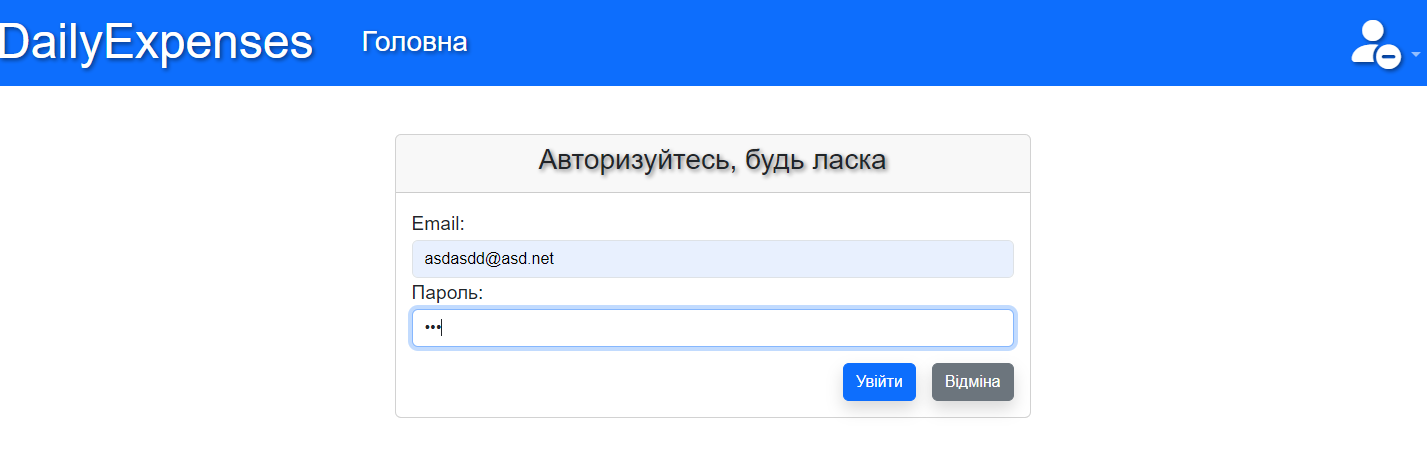


Рисунок 3.3 – сторінка з формою авторизації користувача

На рисунку 3.4 зображена сторінка з формою реєстрації користувача, яка дозволяє зареєструватися в додатку як новий користувач.

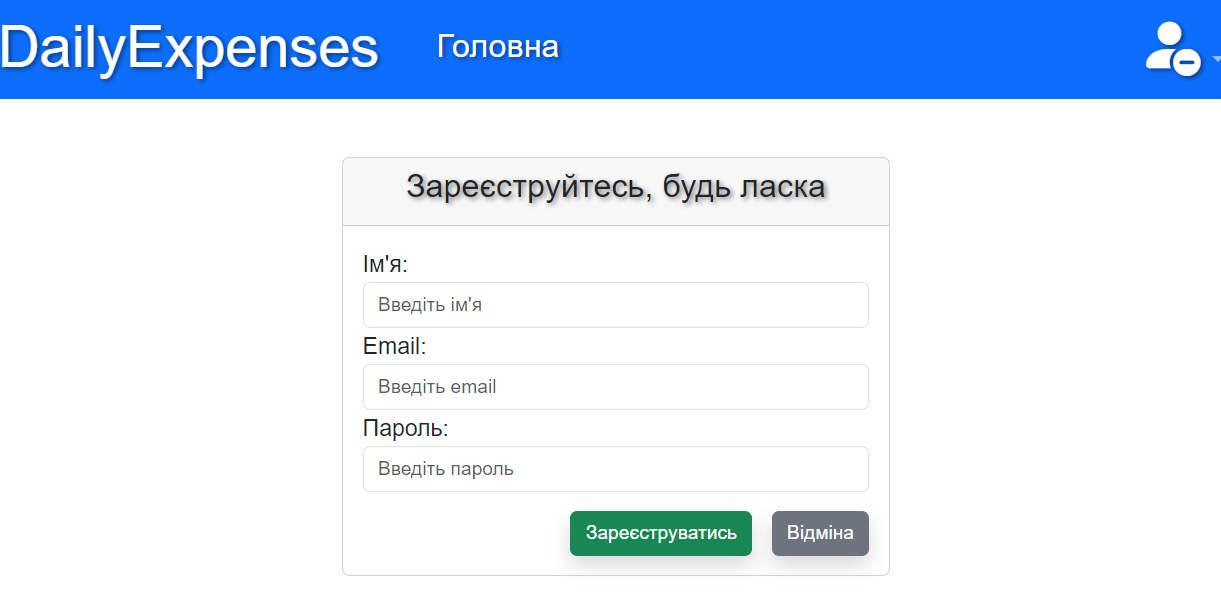


Рисунок 3.4 – меню неавторизованого користувача

На рисунку 3.5 зображена екранна форма початкової сторінки вже авторизованого користувача. Вона має повноцінне меню роботи з додатком, що дозволяє перейти на сторінки роботи з витратами, налаштуваннями додатку та користувача.

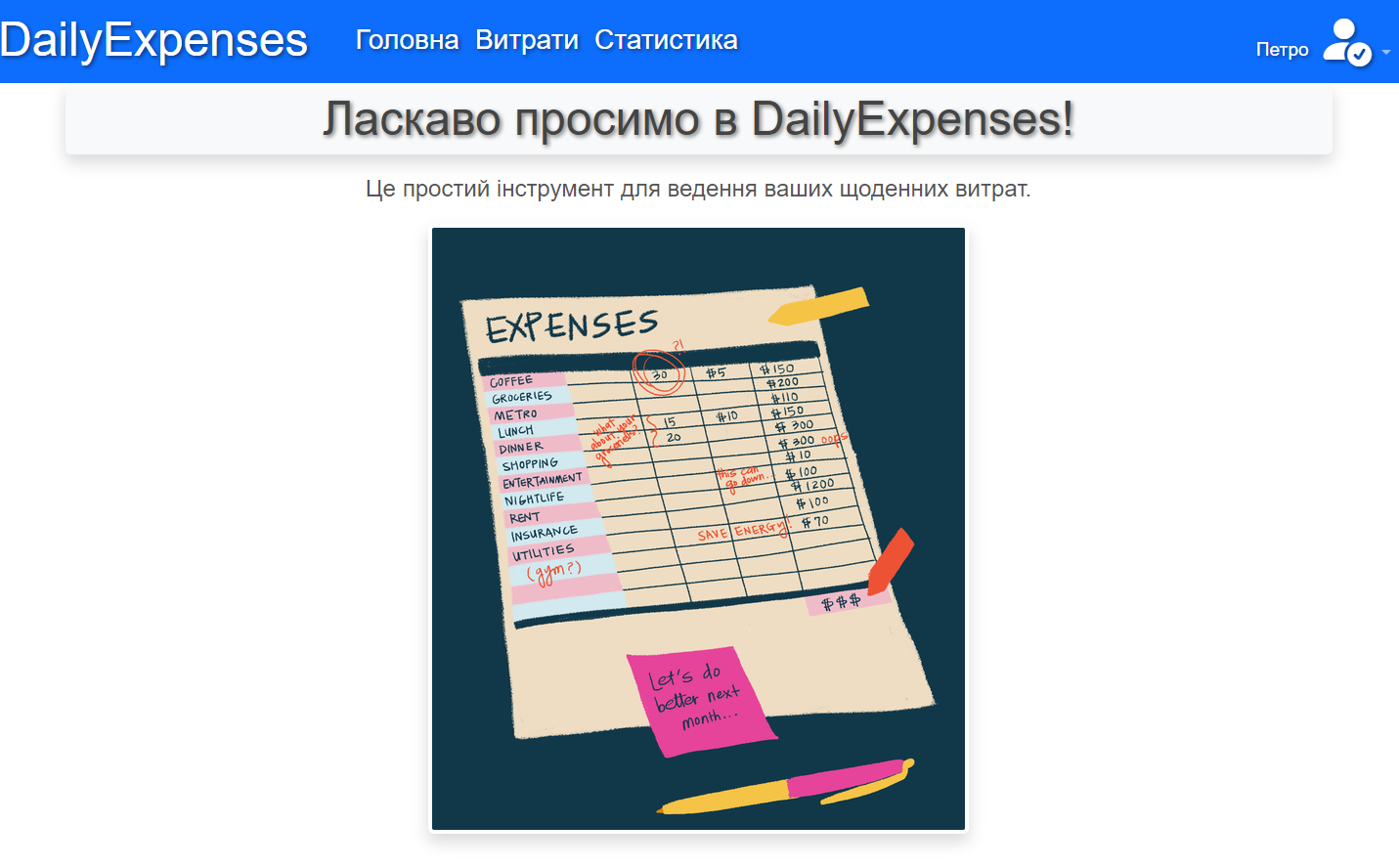


Рисунок 3.5 –початкова сторінка авторизованого користувача

На рисунку 3.6 зображено меню авторизованого користувача, яке дозволяє перейти на сторінки корегування даних користувача та налаштування додатку.

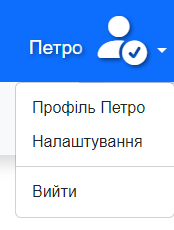


Рисунок 3.6 – меню авторизованого користувача

На рисунку 3.7 зображена сторінка з формою редагування користувача, яка дозволяє зареєструватися в додатку як новий користувач.

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.

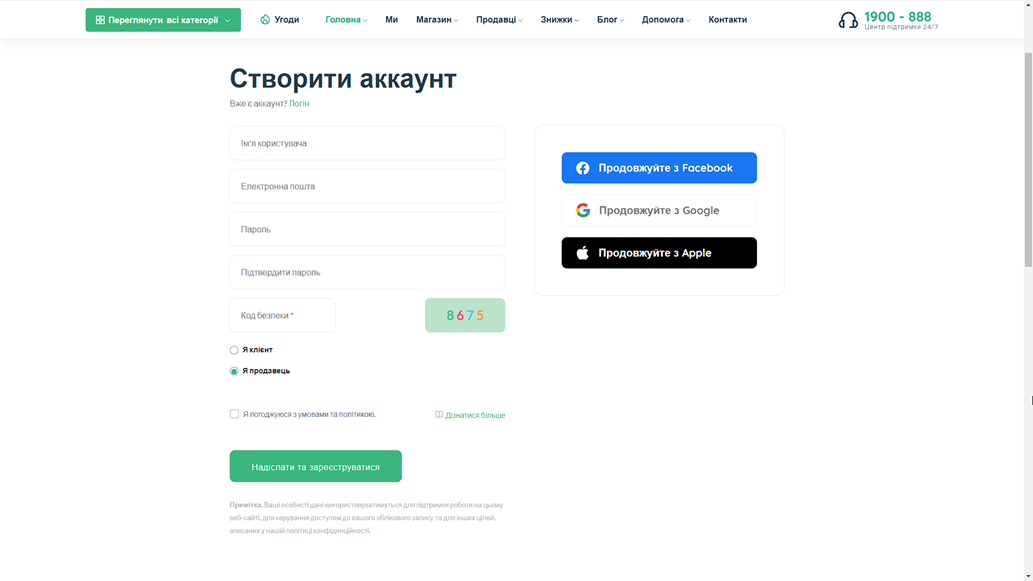


Рисунок 3.2 – Екранна форма створення нового акаунту

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.

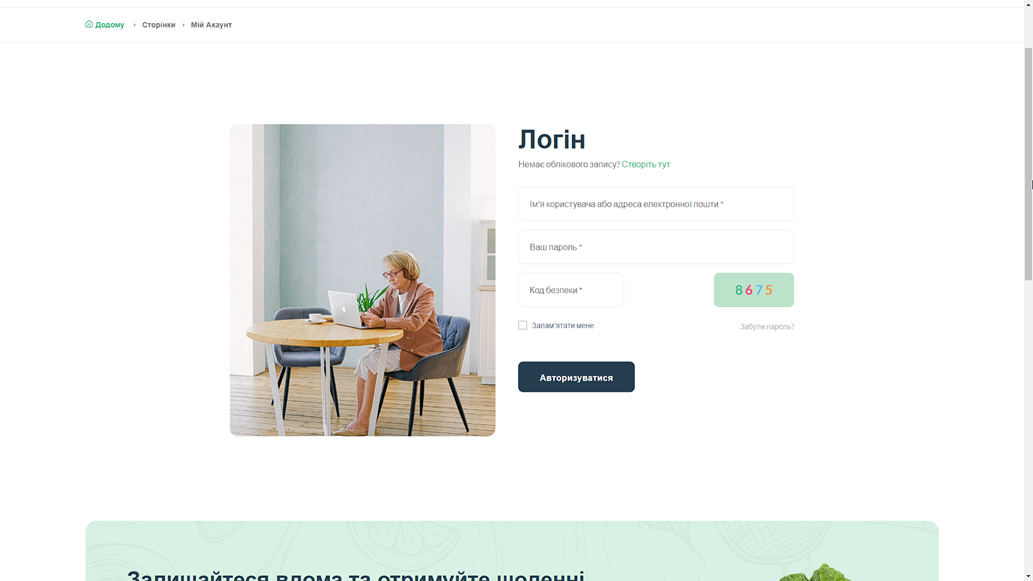


Рисунок 3.3 – Екранна форма логіну

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.

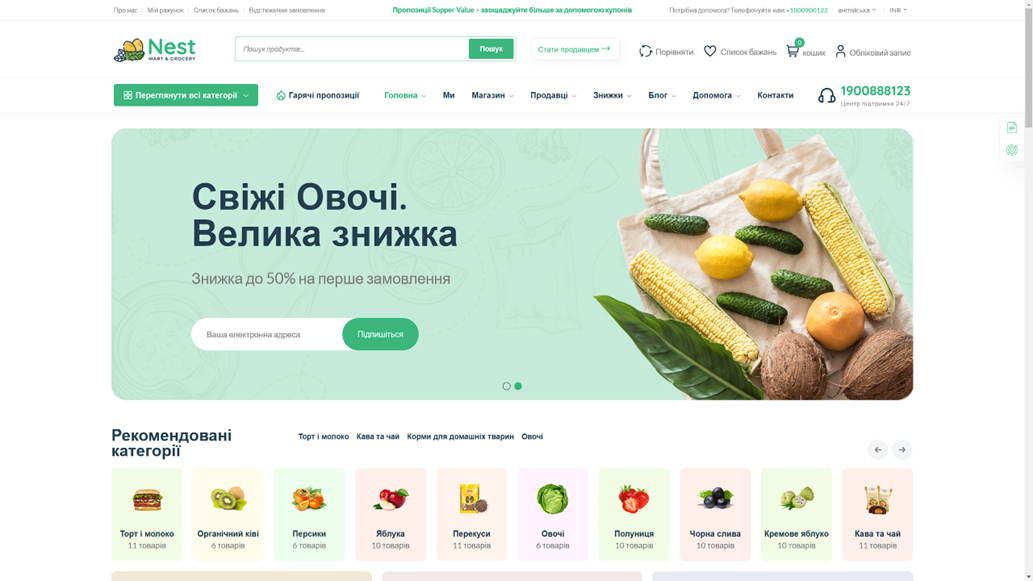


Рисунок 3.4 – Екранна форма головного вікна веб-ресурсу

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.

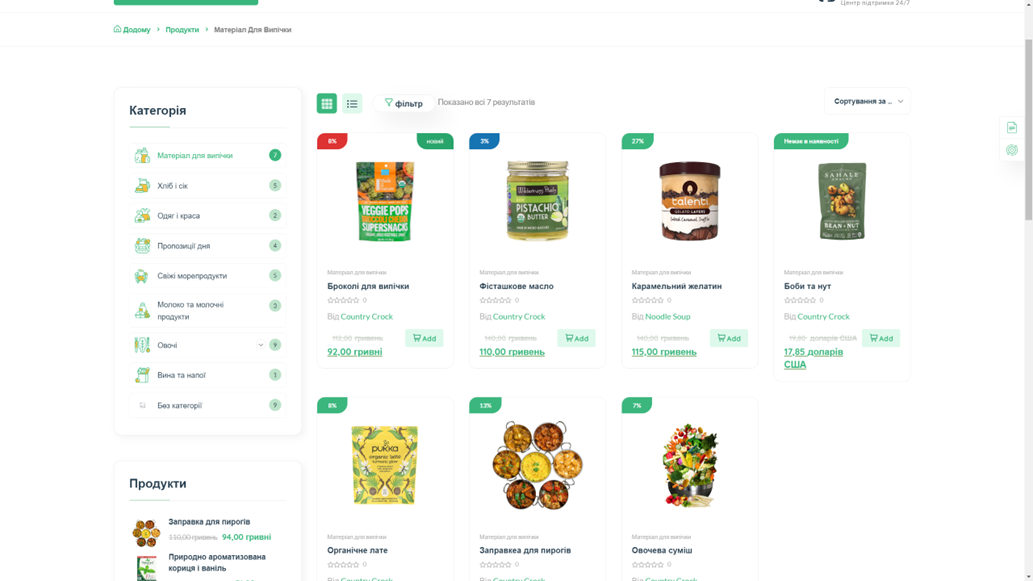


Рисунок 3.5 – Екранна форма переліку продуктів певної категорії

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.

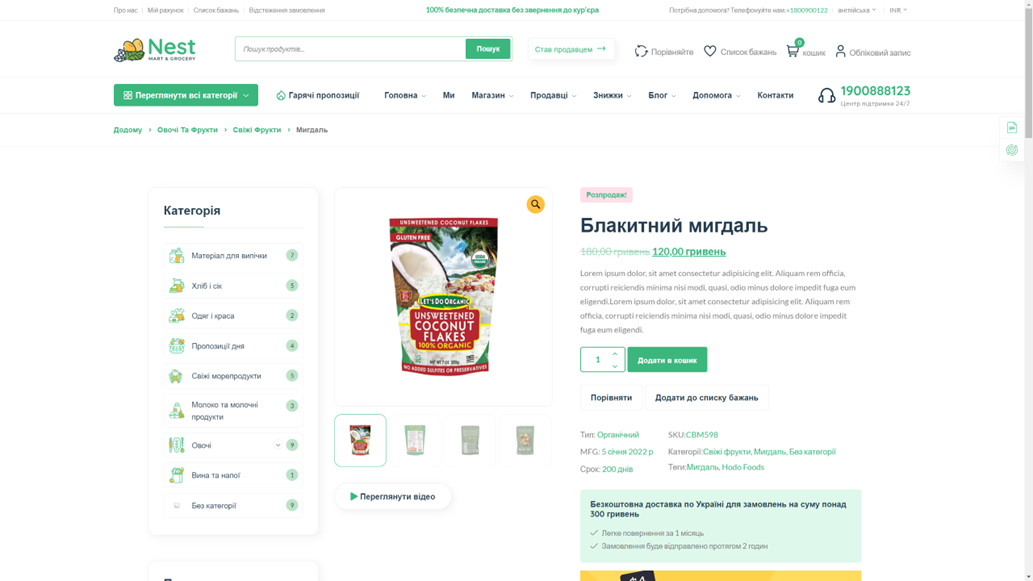


Рисунок 3.6 – Екранна форма вікна продукту

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.

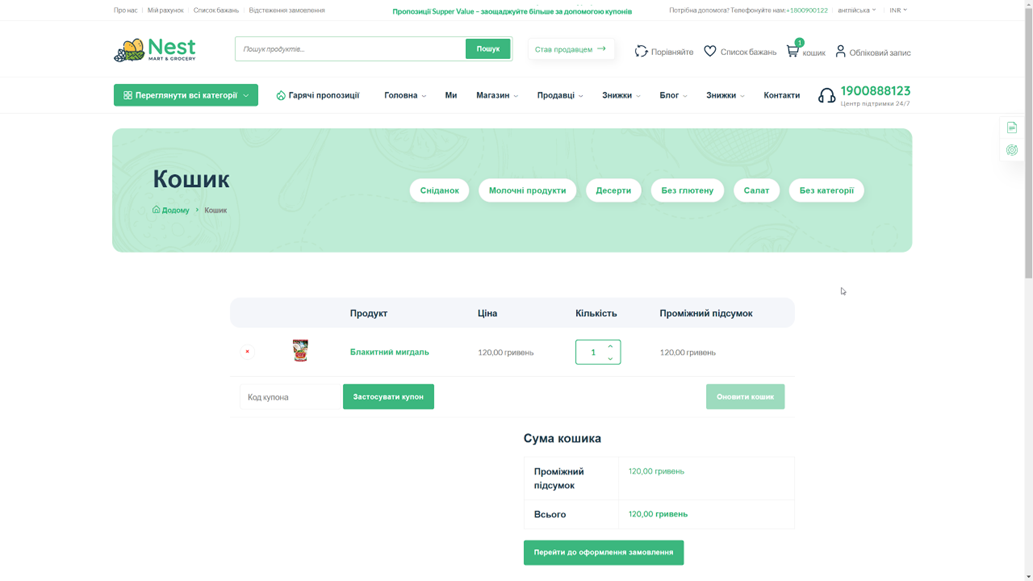


Рисунок 3.7 – Екранна форма вікна кошику товарів

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.

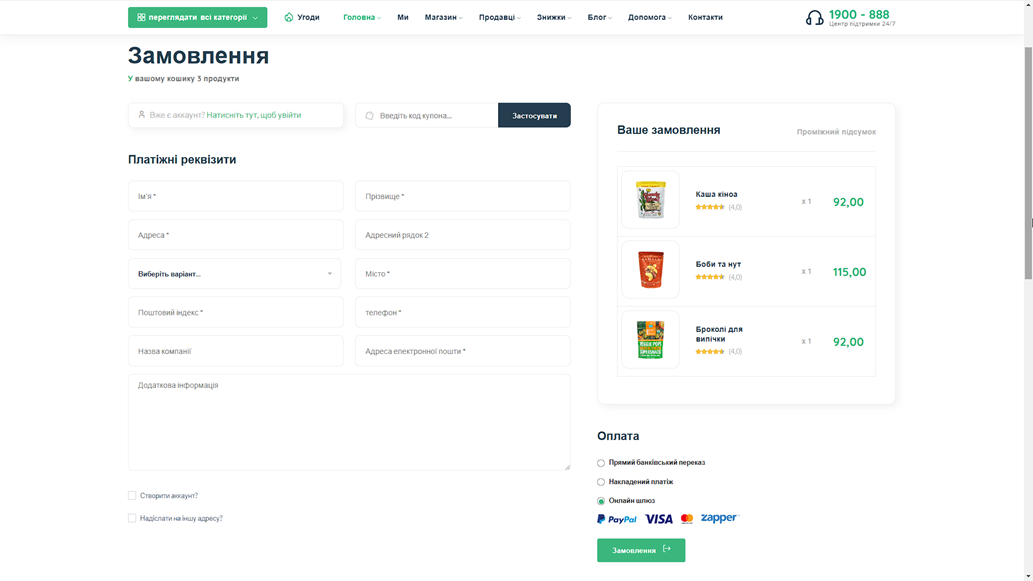


Рисунок 3.8 – Екранна форма оформлення замовлення

## 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. Уоллс К. «Spring в действии.» 6-е изд./ пер. с англ.А. Н. Киселева. – М.: ДМКПресс, 2022.
2. Гутьеррес Фелипе «Spring Boot 2: лучшие практики для профессионалов» СПб.: Питер, 2020 - 464 с.: ил. — (Серия «Библиотека программиста»).
3. Хеклер Марк «Spring Boot по-быстрому» СПб.: Питер, 2022. — 352 с.: ил. — (Серия «Бестселлеры O’Reilly»).
4. Spring Framework guides. URL: <https://spring.io/guides> (дата звернення 25.04.2023).
5. Tutorial: Thymeleaf + Spring. URL: https://www.thymeleaf.org/doc/tutorials/3.0/thymeleafspring.html (дата звернення 25.04.2023).
6. Учебник: Использование Thymeleaf. URL: <https://habr.com/ru/articles/350862/> (дата звернення 26.04.2023)
7. Spring Data JPA. URL: <https://habr.com/ru/articles/435114/> (дата звернення 28.04.2023)
8. Lombok. Полное руководство. URL: <https://habr.com/ru/companies/piter/articles/676394/> (дата звернення 28.04.2023)
9. Thymeleaf Page Layouts. URL: <https://www.thymeleaf.org/doc/articles/layouts.html> (дата звернення 30.04.2023)
10. Современный базовый стек приложений Java Spring. URL: <https://coderlessons.com/articles/java/sovremennyi-bazovyi-stek-prilozhenii-java-spring> (дата звернення 25.04.2023)
11. Spring MVC — основные понятия и архитектура. <https://highload.today/spring-mvc/#1> (дата звернення 28.04.2023)