

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ

Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту

Завідувач кафедри _____ Олена ОЛЬХОВСЬКА
(підпис)

«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

«РОЗРОБКА ANDROID ДОДАТКУ КУР'ЄРСЬКОЇ СЛУЖБИ»

зі спеціальності 122 «Комп'ютерні науки»
освітня програма «Комп'ютерні науки»
ступеня бакалавр

Виконавець роботи Писаренко Владислав Вадимович
_____ «___» _____ 2024 р.
(підпис)

Науковий керівник к.пед. н., доцент Кошова Оксана Петрівна
_____ «___» _____ 2024 р.
(підпис)

Рецензент _____

ПОЛТАВА 2024

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ Олена ОЛЬХОВСЬКА

« ____ » вересня 202_ р

**ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК
ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему «Розробка Android додатку кур'єрської служби»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня бакалавр

Прізвище, ім'я, по батькові Писаренко Владислав Вадимович

Затверджена наказом ректора № 204-Н від «27» жовтня 2023 р

Термін подання студентом роботи «__» _____ 202_ р.

Вихідні дані до кваліфікаційної роботи: публікації з теми, рисунки роботи додатку.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

РОЗДІЛ 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Теоретичні основи

1.2 Огляд та аналіз наявного програмного забезпечення

РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА

2.1 Обрані технології та інструменти розробки

2.2. Опис архітектури додатку кур'єрської служби

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

3.1. Обґрунтування вибору мови програмування

3.2. Створення UI/UX

3.3. Робота з базою даних

3.4 Програмна реалізація елементів алгоритму

РОЗДІЛ 4. ТЕСТУВАННЯ РОБОТИ ЗАСТОСУНКУ

ВИСНОВКИ

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

ДОДАТОК А

Перелік графічного матеріалу: 1 таблиця, 15 ілюстрацій

Консультанти розділів кваліфікаційної роботи

Розділ	ПІБ, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постанова задачі	Кошова О.П.		
Інформаційний огляд	Кошова О.П.		
Теоретична частина	Кошова О.П.		
Практична реалізація	Кошова О.П.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій і стандартів та звіт керівнику		
3. Постановка завдання		
4. Інформаційний огляд джерел бібліотек та Інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доопрацювання (за необхідності), рецензування		

Дата видачі завдання «__» _____ 202__ р.

Здобувач вищої освіти _____ Писаренко Владислав Вадимович
(підпис)

Науковий керівник _____ к.пед.н., доц. Кошова О.П.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «_____» _____ 2024 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

Затверджую
Зав. кафедрою _____
к.ф.-м.н. Олена ОЛЬХОВСЬКА
« ____ » _____ 202__ р.

Погоджено
Науковий керівник _____
к.пед.н., Оксана КОШОВА
« ____ » _____ 202__ р.

План

кваліфікаційної роботи на тему
«Розробка Android додатку кур'єрської служби»
зі спеціальності 122 Комп'ютерні науки
освітня програма 122 «Комп'ютерні науки»
ступеня бакалавр
Прізвище, ім'я, по батькові Писаренко Владислав Вадимович

ВСТУП

РОЗДІЛ 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Теоретичні основи

1.2 Огляд та аналіз наявного програмного забезпечення

РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА

2.1 Обрані технології та інструменти розробки

2.2. Опис архітектури додатку кур'єрської служби

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

3.1. Обґрунтування вибору мови програмування

3.2. Створення UI/UX

3.3. Робота з базою даних

3.4 Програмна реалізація елементів алгоритму

РОЗДІЛ 4. ТЕСТУВАННЯ РОБОТИ ЗАСТОСУНКУ

ВИСНОВКИ

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

ДОДАТОК А

Здобувач вищої освіти _____ Владислав ПИСАРЕНКО
« ____ » _____ 202__ р.

РЕФЕРАТ

Записка: 65 с., 1 таблиця, 15 рисунків, 1 додаток, 19 джерел.

КУР'ЄРСЬКА СЛУЖБА ДОСТАВКИ, ANDROID-ДОДАТОК, KOTLIN,
БАЗА ДАНИХ ROOMDB .

Об'єкт розробки – Android додаток кур'єрської служби.

Мета роботи – Розробка додатку для кур'єрської служби за допомогою доступних для студента інструментів та матеріалів.

Методи дослідження – методи теорії систем числення, середовище візуальної розробки програм JetBrains IntelliJ IDEA та об'єктно-орієнтована мова програмування Kotlin.

Зроблено огляд відомого програмного забезпечення для кур'єрської доставки їжі та товарів, виявлено позитивні та негативні сторони. Розроблено алгоритм Android додатку для замовлення їжі.

Програмно реалізовано елементи Android додатку для використання користувачами.

У результаті тестування виявлено, що додаток працює тільки з підключенням до мережі Інтернет.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
ПОСТАНОВКА ЗАВДАННЯ	9
РОЗДІЛ 1. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	11
1.1 Теоретичні основи.....	11
1.2 Огляд та аналіз наявного програмного забезпечення.....	12
РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА.....	34
2.1 Обрані технології та інструменти розробки.....	34
2.2. Опис архітектури додатку кур'єрської служби.....	42
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА.....	44
3.1. Обґрунтування вибору мови програмування	44
3.2. Створення UI/UX	45
3.3. Робота з базою даних	46
3.4 Програмна реалізація елементів алгоритму	47
РОЗДІЛ 4. ТЕСТУВАННЯ РОБОТИ ЗАСТОСУНКУ.....	49
ВИСНОВКИ	54
СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.....	55
ДОДАТОК А. Код програми.....	56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення

SQL – structured query language

ОС(OS) – операційна система

ОЗУ(RAM) – оперативний запам'ятовуючий пристрій

DOS – disk operation system

BIOS – basic input/output system

БД(DB) – база даних

СУБД – система управління базами даних

ІС – інформаційна система

API(Application Programming Interface) - прикладний програмний інтерфейс

XML(Extensible Markup Language) - розширювана мова розмітки

.kt - розширення файлу мови програмування Kotlin

IDE(Integrated development environment) - Інтегроване середовище розробки

ВСТУП

Актуальність. У наш час, в епоху стрімкого розвитку Інтернету та сучасних технологій, більшість людей стали активними користувачами онлайн-сервісів та використовують смартфони для різноманітних потреб. Це особливо актуально у зв'язку з глобальною пандемією COVID-19, яка спровокувала масовий перехід до роботи та навчання на відстані. Люди дистанційно займаються своїми справами, працюючи та навчаючись зручно з власного дому.

Ця тенденція також знайшла своє втілення в збільшенні популярності послуг з доставки товарів додому. Замість того, щоб витратити час на поїздки до магазину або ресторану, люди віддають перевагу замовленню продуктів та страв через Інтернет і отриманню їх безпосередньо до свого дверного порогу. Це дає можливість економити час, який можна витратити на більш важливі або приємні справи, а також зменшує ризик зараження вірусом.

Не лише пандемія, але і політичні конфлікти та повномасштабне вторгнення росії на територію України, змусили багатьох людей переглянути свій спосіб життя та пристосуватися до нових умов. Забезпечений доступ до послуг доставки додому став необхідністю для багатьох, надаючи зручність і безпеку у виборі та отриманні необхідних товарів і послуг.

Метою роботи є розробка додатку для кур'єрської служби за допомогою доступних для студента інструментів та матеріалів.

Об'єктом розробки в цій роботі є Android додаток кур'єрської служби.

Предметом розробки є елементи ПЗ Android додатку кур'єрської служби.

Завдання – Розробка елементів ПЗ Android додатку кур'єрської служби, з вигаданим меню та базою даних для замовлень. Аналіз наявних кур'єрських додатків, аналіз потрібних інструментів, фреймворків та вибір їх для розробки свого додатку, розробити UI/UX для додатку, розробити базу даних та реалізувати алгоритм додатку.

У ході реалізації додатку використано мову програмування Kotlin, базу даних від RoomDB, середовище розробки IntelliJ IDEa, фреймворки від Google.

Робота складається з вступу, у першому розділі інформаційний огляд, у другому розділі теоретична частина, у третьому розділі практична частина, висновок, список використаних джерел та додатки.

Обсяг пояснювальної записки: 65 стор., джерела – 19 назв., таблиці – 1, рисунки – 15.

ПОСТАНОВКА ЗАВДАННЯ

У сучасному світі концепція розробки мобільних додатків для кур'єрської служби набула особливої актуальності та значущості для суспільства. Завдяки розширенню впливу Інтернету та зростаючій популярності смартфонів, з'явилося безліч можливостей для спрощення та ефективного вирішення повсякденних завдань. Один із таких рішень - це зручний і швидкий доступ до кур'єрських послуг через мобільний додаток.

Ця ідея заснована на потребах сучасного споживача, який цінує свій час і шукає способи оптимізувати свої рутинні обов'язки. Додаток кур'єрської служби в смартфоні дозволяє замовити продукти з магазину або страви з ресторану чи кафе за декілька клацань екрану, уникнувши необхідності особисто відвідувати ці заклади. Це забезпечує зручність та ефективність в процесі покупок, а також дозволяє заощадити дорогоцінний час, який можна використати для інших цікавих або корисних справ.

Важливо враховувати, що розвиток цього типу додатків сприяє не лише комфорту споживача, але і створенню нових робочих місць у сфері кур'єрських послуг та підтримці місцевих підприємств, таких як магазини та ресторани.

Таким чином, мобільні додатки для кур'єрської служби в смартфоні відіграють важливу роль у вдосконаленні способів взаємодії між споживачем та підприємствами, сприяючи ефективному функціонуванню сучасного ринку послуг.

Мета даного завдання полягає у розробці повнофункціонального кур'єрського додатку для платформи Android. Цей додаток буде призначений для оптимізації процесу замовлення та доставки товарів, щоб забезпечити користувачам зручність та ефективність у вирішенні повсякденних завдань. Для досягнення цієї мети необхідно виконати ряд завдань. Спочатку потрібно провести аналіз наявних кур'єрських додатків, щоб визначити їхні сильні та слабкі сторони, а також виявити можливість покращення та інновацій. Цей аналіз допоможе зрозуміти потреби та очікування користувачів.

Після аналізу наявних додатків необхідно визначити потрібні інструменти та фреймворки для розробки свого додатку. Вибір правильних інструментів буде ключовим етапом, оскільки вони повинні відповідати вимогам проекту та забезпечити ефективну розробку.

Наступним етапом є розробка UI/UX для додатку. Інтерфейс повинен бути зрозумілим та не завантаженим, а взаємодія з додатком - інтуїтивно зрозумілою для користувачів. Процес проектування інтерфейсу включатиме створення wireframes, макетів та прототипів для перевірки та оптимізації користувацького досвіду.

Після розробки UI/UX переходимо до створення бази даних та реалізації взаємодії з нею. База даних повинна ефективно зберігати та організовувати дані про користувачів, замовлення, товари та інше. Забезпечення правильної взаємодії з базою даних є ключовим аспектом для успішної роботи додатку.

Далі необхідно реалізувати алгоритми додатку, включаючи процеси замовлення, оплати, вибору кур'єра та доставки. Код додатку має бути чистим, організованим та ефективним, щоб забезпечити його стабільну та швидку роботу.

Останнім етапом є тестування додатку для перевірки його функціональності та надійності. Тестування допоможе виявити та виправити будь-які помилки чи недоліки перед випуском додатку на ринок.

Загальною вимогою до проекту є створення зрозумілого та ергономічного інтерфейсу, який забезпечить комфортну взаємодію користувачів з додатком.

Такий додаток має забезпечувати ефективне вирішення повсякденних завдань та підвищувати якість життя користувачів.

РОЗДІЛ 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Теоретичні основи

Кур'єрські служби є важливою складовою сучасного бізнесу та економіки. Вони забезпечують доставку товарів та послуг від відправника до одержувача з мінімальними зусиллями з боку клієнтів. З розвитком технологій мобільні додатки стали незамінною частиною бізнес-екосистеми кур'єрських служб.

Вони дозволяють клієнтам легко розміщувати замовлення, відстежувати їх статус та спілкуватися з кур'єрами.

Мобільні додатки спрощують процеси доставки та дозволяють покращити обслуговування клієнтів. Також через пандемію в 2020 році а тепер і через повномасштабне вторгнення росії на територію України ринок кур'єрських служб став досить великий, та успішно процвітає, звісно під час вторгнення багато компаній понесли втрати, але в більшості своїй вони змогли реабілітуватись та продовжити свою роботу.

По оцінкам експертів з сервісу «Glovo», український ринок доставки товарів виріс у 2020 році у 6-7 разів у порівнянні з минулим роком, а у 2023 році за даними компанії «Statista» прогнозують, що прибуток виросте до 0,91 трильйона доларів, коли до війни, у 2021 році ринок їжі досягав 300млн.

Кур'єрські служби використовуються в різних сферах, включаючи торгівлю, логістику, медицину, харчову індустрію та багато інших. З розвитком сучасних технологій та мобільних пристроїв стали популярними мобільні додатки для управління та відстеження кур'єрською службою. Ці додатки надають зручний і швидкий спосіб взаємодії з кур'єрами та відстеженням замовлень. Вони спрощують процес замовлення, зменшують час очікування та забезпечують зручний інтерфейс для користувачів.

З урахуванням обміну чутливими даними, такими як адреси, особиста інформація та дані платежів, забезпечення безпеки є критично важливою частиною розробки кур'єрського додатку. Заходи безпеки включають

шифрування даних, двофакторну аутентифікацію та захист від несанкціонованого доступу.

Для розробки мобільних додатків для платформи Android використовуються різноманітні технології та інструменти. Основні з них включають мову програмування Java або Kotlin, середовище розробки Android Studio, бібліотеки та фреймворки для роботи з інтерфейсом користувача та доступом до функціональних можливостей пристроїв.

Додаток складається з таких компонентів:

- Вікно реєстрації/входу;
- Список продуктів на замовлення;
- Пошук товарів;
- Вікно інформації про замовлення;

Звісно є можливість розробити більше компонентів, зробити додаток для кур'єра, за потреби, але на мою думку такі проекти потрібно робити для комерційного застосування та при наявності команди програмістів, тестувальників, дизайнерів тощо, та і в рамках цієї роботи розглядається тільки додаток для користувачів. Тому зупинимось на наявних компонентах для додатку. Звісно може здаватись, що це додаток не повноцінний, але розробка двох додатків може займати більше часу, ніж виділено на цю роботу, за умови що цим буде займатися один розробник.

1.2 Огляд та аналіз наявного програмного забезпечення

На сьогодні існує багато різних конкурентних сервісів, які надають послуги доставки товарів на замовлення, кожен з яких займає свою нішу на ринку. Але варто зауважити, що деякі з них виявилися надзвичайно популярними на українському ринку. Серед найбільш популярних сервісів на сьогодні можна виділити такі, як: «Нова Пошта», «Glovo», «Mister.am», «Bolt Food», «Silpo». Тому ми їх і розглянемо.

Glovo UA – лідер ринку в Україні:

"Glovo" - це програма, яка дозволяє користувачам отримувати різні товари, які можуть бути поміщені в кошик кур'єра, у своєму місті протягом короткого періоду часу. Цей сервіс створений для об'єднання користувачів, компаній та кур'єрів з метою забезпечення зручності та швидкості у процесі замовлення різних товарів.

Головна ідея полягає в укладанні контрактів із ресторанами та супермаркетами та створенні комфортних умов для праці кур'єрів. Проект був створений з метою зміни способу, яким користувачі отримують потрібні їм речі в місті, роблячи процес покупок більш швидким та доступним. "Glovo" має за мету надати кожному легкий доступ до будь-яких товарів у своєму місті та суттєво вплинути на економіку, суспільство та навколишнє середовище[13,15].

Реєстрація в додатку проводиться шляхом входу через соціальну мережу "Facebook" або за допомогою номера телефону, і вона є обов'язковою для всіх користувачів. Усі переходи між різними вікнами супроводжуються плавними анімаціями, що забезпечує приємний користувацький досвід.



Рисунок 1.1 Інтерфейс додатку "Глово"

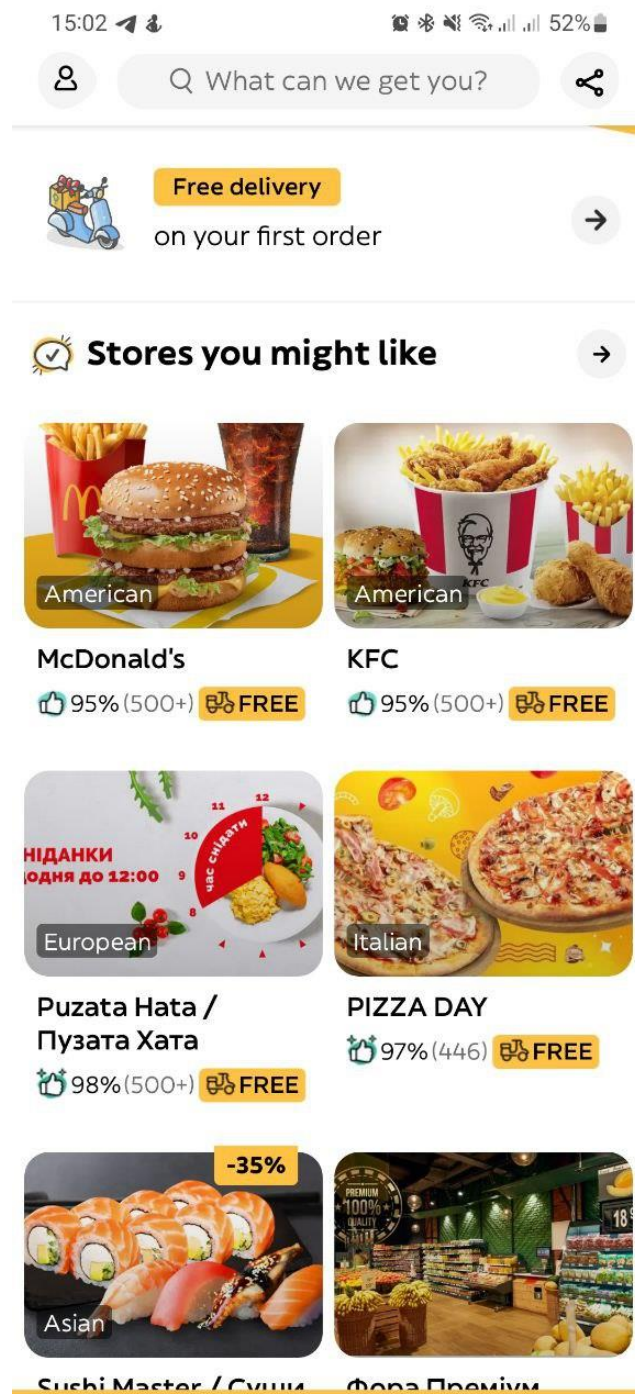


Рисунок 1.2 Список закладів "Глово"

Згідно з рисунками(Рисунки 1.1-1.2), додаток Glovo має простий інтерфейс, що дозволяє швидко та просто користувачам взаємодіяти з ним.

Ось опис його основних елементів:

1. Головний екран: Після запуску додатка користувач потрапляє на головний екран, де розташовуються такі елементи:

- Пошукове поле: Дозволяє здійснювати пошук перекладів за ключовими словами або фразами.
- Кнопка акаунта: Користувач може зайти в меню свого акаунту та проглянути замовлення, інформацію, промокоди, видалити свої дані та вийти з акаунту. Ця опція розташована в лівому верхньому куті екрану з картинкою людини.
- Кнопка “Поділитися”: Дозволяє користувачеві поділитися додатком Glovo з друзями, знайомими, колегами, тощо.

2. Результати пошуку: Після введення запиту в пошукове поле користувач отримує список результатів товарів та закладів. Ці результати можуть включати страви в закладах, назви закладів, ціни на товари та інші корисні відомості.

3. Меню акаунту: В меню можуть бути доступні додаткові функції, такі як:

- Мої замовлення: Дозволяє побачити список минулих або теперішніх замовлень та продивитись їх статус.
- Моя інформація: Користувач може побачити інформацію, яку він надав додатку. Це: Ім’я, електронна пошта, можливість зміни паролю, номера телефону, налаштування способів оплати, перегляд та зміна інформації для доставки та налаштування приватності(отримування рекламних пропозицій).
- Промокоди: Користувач може додати надані йому промокоди та використати їх у подальших замовленнях.
- Довідка: Надає інформацію про користування додатком, правила використання, контактні дані підтримки тощо.
- Видалення інформації про акаунт: Надає можливість попади заявку на видалення акаунту, якщо користувач більше не потребує використання додатку.

- Вихід: Користувач може вийти з свого аккаунту Glovo.

Mister.am:

"Mister.am" - сервіс доставки їжі, який пропонує клієнтам зручний спосіб отримати смачну їжу з різних ресторанів та закладів харчування, не накладаючи додаткових витрат. Співпраця із найкращими закладами міста дозволяє гарантувати високу якість та різноманітність страв. Основний принцип роботи - швидка доставка їжі додому, при цьому ціни залишаються доступними для всіх клієнтів.

Сервіс пропонує різні зручні способи замовлення їжі, включаючи онлайн-замовлення через їх веб-сайт. Користувачі можуть вибирати страви із різних категорій та ресторанів, зручно платити за них та отримувати їх доставку прямо до дверей. Залишаючи замовлення, клієнти отримують підтвердження через SMS та можуть насолоджуватися смачною їжею в затишку свого дому чи офісу.

Один із ключових переваг Mister.am - система кешбеку, яка надає клієнтам можливість зекономити гроші на майбутні замовлення. Кожне замовлення збільшує кешбек, який можна використовувати для сплати частини суми наступних покупок.

Цей сервіс працює в різних містах України, і надає можливість користуватися різноманітними ресторанами та кафе, не обмежуючись географічними рамками. Mister.am надає можливість насолоджуватися смачною їжею з найкращих закладів, роблячи процес замовлення та доставки максимально зручним для клієнтів[16].

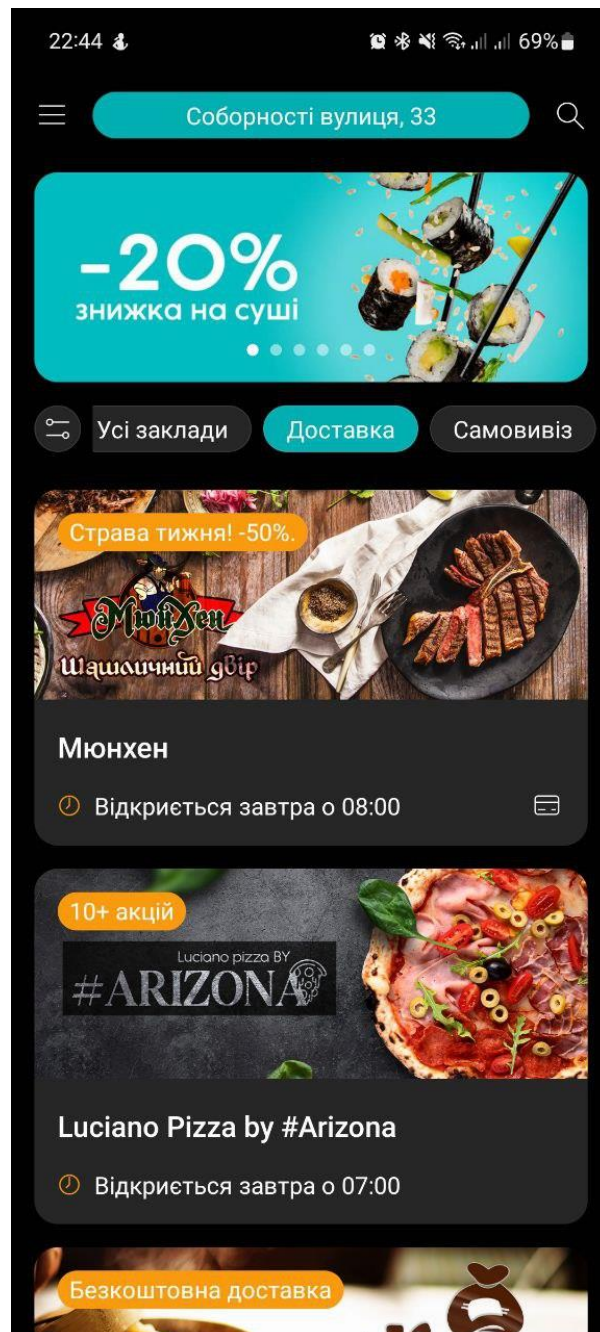


Рисунок 1.3 Головна додатку Mister.Am

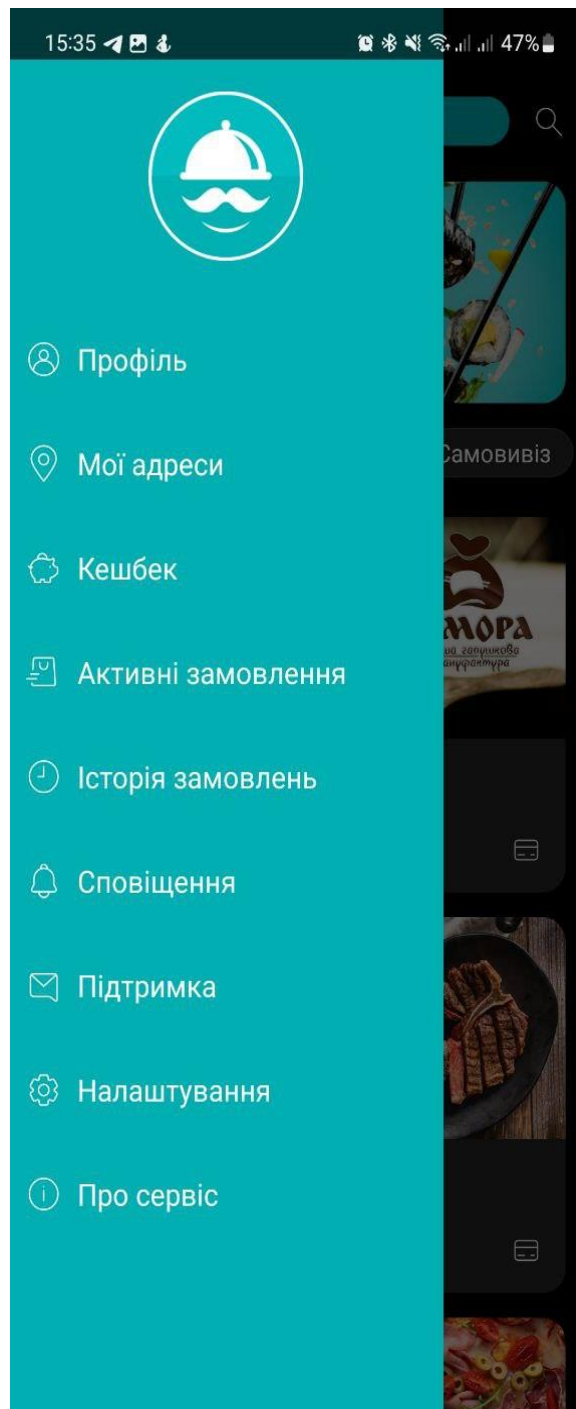


Рисунок 1.4 Меню користувача додатку Mister.Am

Згідно з рисунків(Рисунки 1.3-1.4) додаток Mister.Am має:

1. Головний екран:

- Користувачі бачать список доступних ресторанів або кафе, з яких можна замовити їжу.
- На головному екрані також рекомендовані страви або акції від обраних ресторанів.

2. Сторінка ресторану:

- При виборі ресторану користувачі переходять на сторінку зі списком доступних страв та їхніми цінами.
- Є зображення страв, їх вага, ціна та рейтинг ресторану.

3. Кошик:

- Користувачі можуть додавати страви до кошика, переглядати їхню сумарну вартість та кількість.
- Вони можуть редагувати кількість страв, видаляти їх або продовжувати замовлення.

4. Форма замовлення:

- Після вибору страв користувачі переходять на сторінку оформлення замовлення, де вони вводять інформацію про доставку, контактні дані тощо.
- Тут також буде відображена інформація про загальну вартість замовлення та способи оплати.

5. Меню:

- Користувачі можуть переглядати історію своїх замовлень, змінювати особисті дані, переглянути сповіщення тощо.
- Користувачі можуть зайти в свій профіль, що змінити свій номер телефону або ім'я.
- Користувачі в додатку мають можливість додавати адреси та редагувати їх.
- Користувачі можуть відкрити підтримку, щоб задати запитання через Телеграм або Viber.
- Користувачі мають можливість переглянути свій кешбек.

- Користувачі можуть відкрити “Активні замовлення”, щоб мати можливість проглядати замовлення під час доставки, та мати змогу переглянути історію замовлень в цілому.
- Користувачі можуть зайти в налаштування теми інтерфейсу, якщо вони бажають змінити наявну, ще є інформація про сервіс.
- Користувачі може відкрити “Про сервіс” та має змогу дізнатися соціальні мережі розробників додатку та їх контактну інформацію.

Bolt Food:

Bolt Food, молода компанія на ринку кур'єрських послуг української доставки їжі, почала свою діяльність лише 29 жовтня 2020 року. Основна спеціалізація - доставка страв з різних супермаркетів та ресторанів, кількість яких налічує приблизно 150. Під час запуску послуг компанія пропонує безкоштовну доставку, але обіцяє, що в подальшому вартість буде на 10-15% нижчою, ніж у конкурентів. На даний момент Bolt Food працює лише у Києві, але у планах розширити діяльність на декілька інших великих міст країни. Під час карантину компанія здійснювала доставку виключно безконтактним способом, а платіжні операції приймалися лише за допомогою банківських карток. Під час повномасштабного вторгнення росії в Україну у 2022-2024 роках компанія Bolt Food продовжувала свою діяльність, працюючи в умовах складної соціально-політичної ситуації. З метою забезпечення безпеки клієнтів та кур'єрів, у періоди активних бойових дій та обмежень на в'їзд-виїзд у деякі райони, компанія вживала додаткові заходи безпеки. Доставка продовжувалася виключно за безконтактною схемою, з дотриманням всіх вимог безпеки та гігієни. Оплата за послуги також проводилася електронними засобами, з метою мінімізації фізичного контакту.

Незважаючи на виклики, пов'язані зі складною ситуацією у країні, Bolt Food продовжувала залучати нових партнерів серед ресторанів та супермаркетів, щоб розширювати асортимент доступних продуктів для клієнтів. Крім того, компанія активно працювала над поліпшенням функціональності

мобільного додатку, впроваджуючи нові зручні сервіси та інструменти для користувачів.

У цей важкий період Bolt Food продовжувала демонструвати свою здатність адаптуватися до змінних умов та викликів, надавати надійні послуги та забезпечувати зручні та безпечні умови для своїх клієнтів.

Мобільний додаток Bolt Food має зручний та зрозумілий інтерфейс, який дозволяє легко користуватися ним. Навігаційна панель і великі кнопки спрощують навігацію по додатку. Пошук та фільтрація продуктів здійснюється швидко та ефективно, що дозволяє знайти потрібні товари за їхнім типом чи категорією. Під час першого запуску додатку вимагається реєстрація, під час якої користувач повинен ввести номер телефону, прізвище, ім'я та адресу електронної пошти.

У лютому 2022 року, після російського вторгнення в Україну, естонська батьківська компанія Bolt прийняла рішення видалити всі товари, вироблені в Росії або пов'язані з російськими компаніями, з Bolt Market. Крім того, компанія припинила свою діяльність в Білорусі через підтримку цієї країни російському вторгненню.

24 лютого 2023 року Bolt разом з Visa здійснили пожертву до фонду "Повернись живим" в розмірі 1,5 мільйона гривень. [13,17]



Популярні категорії



Американська кухня



Українська кухня



Фаст-фуд



Піца



Бургери



Салат



Європейська кухня



Сніданок



Суші



Риба



М'ясо



Алкоголь



Азіатська кухня



Рисунок 1.5 Головне меню “Bolt Food”

11:06

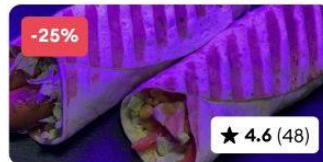
45%

Уолкер

Restaurants and stores

New on Bolt Food

All >



Лупаш Лаваш вул. Кашт...
% 52,00 ₴ · 82,00 ₴ ⌚ 30–45 min



Боулінг FlashBowl ТЦ
% 60,00 ₴ · 90,00 ₴ ⌚ 50–с

Have breakfast with Bolt Food

Find promo up to 20% and order breakfasts!



Promotional offers

All >



Акція Бургер
Шухевича



Юджин
Бургер Шух...



KFC
Блокбастер



Євразія
яковського

Save on delivery 🛵

All >



Рисунок 1.6 Популярні категорії

11:38

41%

Vlad

+380

 Способи оплати

 Промокоди

 Профіль

 Параметри

 Про нас

 Підтримка

Ставай кур'єром

Отримуй дохід за власним графіком



Рисунок 1.7 Вигляд меню користувача

Інтерфейс додатку(Рисунки 1.5-1.7) Bolt Food відзначається мінімалістичним та зрозумілим дизайном, спроектованим для забезпечення зручності та швидкості замовлення їжі. Основні елементи інтерфейсу виглядають так:

1. **Головний екран:** Головний екран додатку Bolt Food містить список доступних ресторанів або кулінарних закладів у вашому районі. Кожен ресторан відображається з вказанням назви, категорії кухні, оцінки та ціни за доставку.

2. **Пошук:** Функція пошуку дозволяє користувачам швидко знаходити ресторани за назвою або категорією кухні.

3. **Меню ресторану:** При виборі конкретного ресторану, користувач може переглянути його меню, де представлені страви та напої з вказанням цін.

4. **Мої замовлення:** Після вибору бажаних страв користувач може переглянути свої замовлення, де відображається загальна сума замовлення та список обраних елементів. Тут також можна ввести зміни до замовлення, змінити кількість страв або видалити їх.

5. **Оплата:** Після завершення вибору страв та перевірки кошика користувач може перейти до екрану оплати, де вказується метод оплати та адреса доставки.

6. **Статус замовлення:** Після підтвердження замовлення користувач може відстежувати його статус, від виконання до доставки. Інформація про статус замовлення зазвичай відображається в розділі "Мої замовлення" або на окремому екрані.

7. **Профіль користувача:** У розділі "Профіль" користувач може переглянути свої особисті дані, способи оплати, параметри, промокоди та звернутися в підтримку і почитати про додаток.

Нова Пошта:

Група "Нова Пошта" пропонує широкий спектр логістичних та пов'язаних з ними послуг як бізнесу, так і приватним особам. До складу Групи входять українські та зарубіжні компанії, зокрема "Нова Пошта", "НП Логістик", "NovaPay" і "Нова Пошта Глобал".

"Нова Пошта" є провідним гравцем на логістичному ринку та надає зручну доставку для всіх клієнтів, незалежно від способу отримання - до відділення, поштомату або за адресою. Крім того, цей сервіс дозволяє тисячам підприємців розвивати свій бізнес не лише в Україні, а й за її межами. Мережа компанії налічує понад 9300 відділень та 14000 поштоматів по всій Україні, а кількість відправлень тільки за 2022 рік перевищила 315 мільйонів. 30 грудня 2022 року компанія відправила 1,9 млн відправлень за добу, і це стало новим рекордом Нової пошти. Для користування усіма функціями додатку, включаючи створення нових замовлень, реєстрація не є обов'язковою, але для доступу до повного функціоналу необхідно авторизуватися. Додаток має простий і зрозумілий інтерфейс, що легко освоюється, і відзначається мінімалістичним оформленням.

Додаток "Нова Пошта" є ключовим інструментом для користувачів в Україні, які шукають зручний спосіб відстеження вантажів, розрахунку вартості послуг та замовлення доставки. Завдяки своїй широкій функціональності та інтуїтивному інтерфейсу, додаток став невід'ємною частиною щоденного життя багатьох людей.

На головному екрані додатку користувачі мають доступ до різноманітних функцій, таких як відстеження вантажу, розрахунок вартості доставки, пошук відділення або відправлення нового вантажу. Інтуїтивно зрозумілий інтерфейс дозволяє швидко здійснювати необхідні дії, навіть для тих, хто користується додатком вперше.

У розділі відстеження вантажів користувачі можуть ввести номер накладної та перевірити статус відправлення в режимі реального часу. Крім того, додаток надає повну інформацію про маршрут вантажу, включаючи дату та час доставки.

Додаток також пропонує зручний сервіс розрахунку вартості доставки, де користувачі можуть обрати тип вантажу, його вагу та розміри, а також маршрут доставки. На основі цих даних додаток автоматично розраховує вартість послуги та надає розгорнуту інформацію про доступні тарифи. Ще додаток має

надійний захист, що не дозволяє робити знімки та запис екрану при користування, це забезпечує зниження ризиків втрати особистої інформації користувачів, особливо при втраті смартфона або використання даних шахраями для отримання посилок.

Окрім цього, "Нова Пошта" дозволяє зручно шукати найближчі відділення компанії за допомогою GPS, переглядати робочі години та послуги, які доступні в кожному відділенні.

Загалом, додаток "Нова Пошта" відповідає вимогам сучасних користувачів, надаючи широкий спектр послуг та зручний інтерфейс для їх використання[14].

Silpo:

Silpo - це одна з провідних мереж супермаркетів в Україні, яка пропонує розмаїття продуктів харчування та товарів для дому. Заснована у 1998 році, мережа Silpo налічує понад 300 супермаркетів у всій країні. У мережі представлена продукція «Власного імпорту» з більш ніж 80 країн світу, якісно відібрані товари українських крафтярів «Лавка традицій», власні торгові марки, а також товари власного виробництва – від запашної випічки «Крафтяр» до солодошів «Власної кондитерської». У Silpo також приділяють велику увагу враженням гостей, тому кожен новий магазин оформлений в індивідуальному дизайні. Цей бренд відомий своїм широким вибором свіжих продуктів, високою якістю обслуговування та постійними інноваціями.[18]

Мобільний додаток Silpo - це передова платформа, яка дозволяє клієнтам мережі Silpo робити покупки онлайн, отримувати знижки та бонуси, а також користуватися іншими зручними сервісами. Ось декілька ключових особливостей додатка:

1. Онлайн-замовлення: Користувачі можуть оглядати асортимент товарів, скласти список покупок та робити замовлення безпосередньо через свій смартфон. Це допомагає економити час та зручно планувати покупки, особливо для зайнятих людей.

2. Доставка додому: Додаток Silpo пропонує можливість доставки продуктів прямо до дверей клієнта. Це особливо зручно для тих, хто не має можливості відвідати супермаркет або має обмежений час.

3. Програми лояльності: Користувачі можуть зареєструвати свою картку лояльності в додатку та отримувати знижки, бонуси та інші привілеї за покупки. Це заохочує постійних клієнтів і підвищує їхню залученість до мережі Silpo.

4. Пошук товарів: У додатку реалізована зручна система пошуку, яка допомагає користувачам швидко знаходити потрібні товари за ключовими словами або категоріями.

5. Оплата онлайн: Крім замовлення продуктів, користувачі можуть здійснювати оплату безпосередньо через додаток. Це забезпечує швидку та безпечну оплату покупок, що робить процес ще зручнішим для клієнтів.

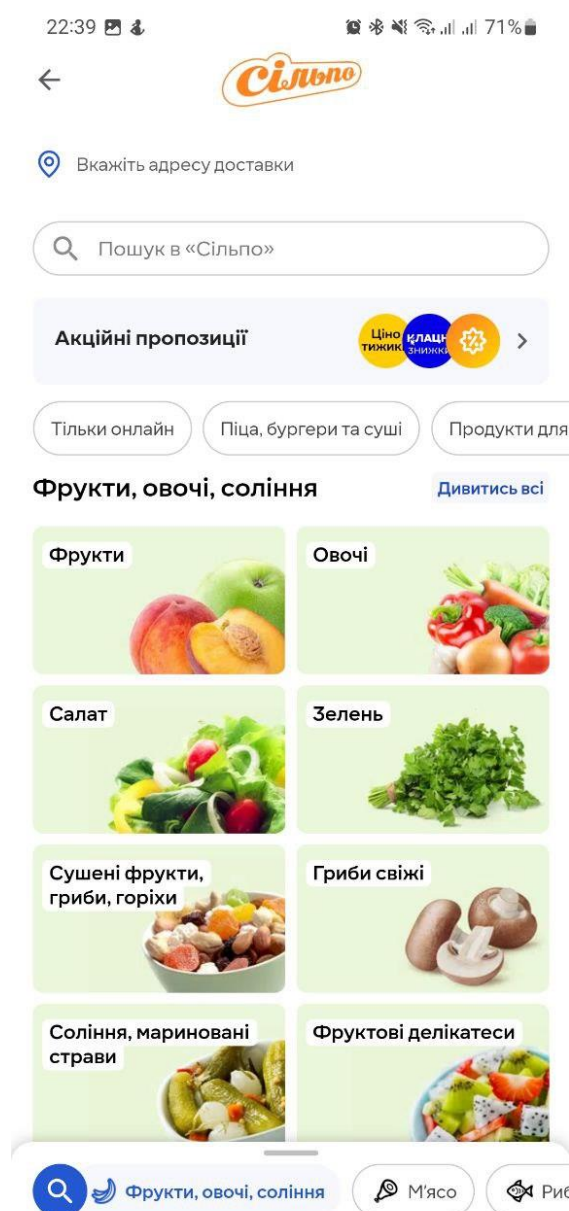


Рисунок 1.8 Інтерфейс пошуку додатку "Сільпо"

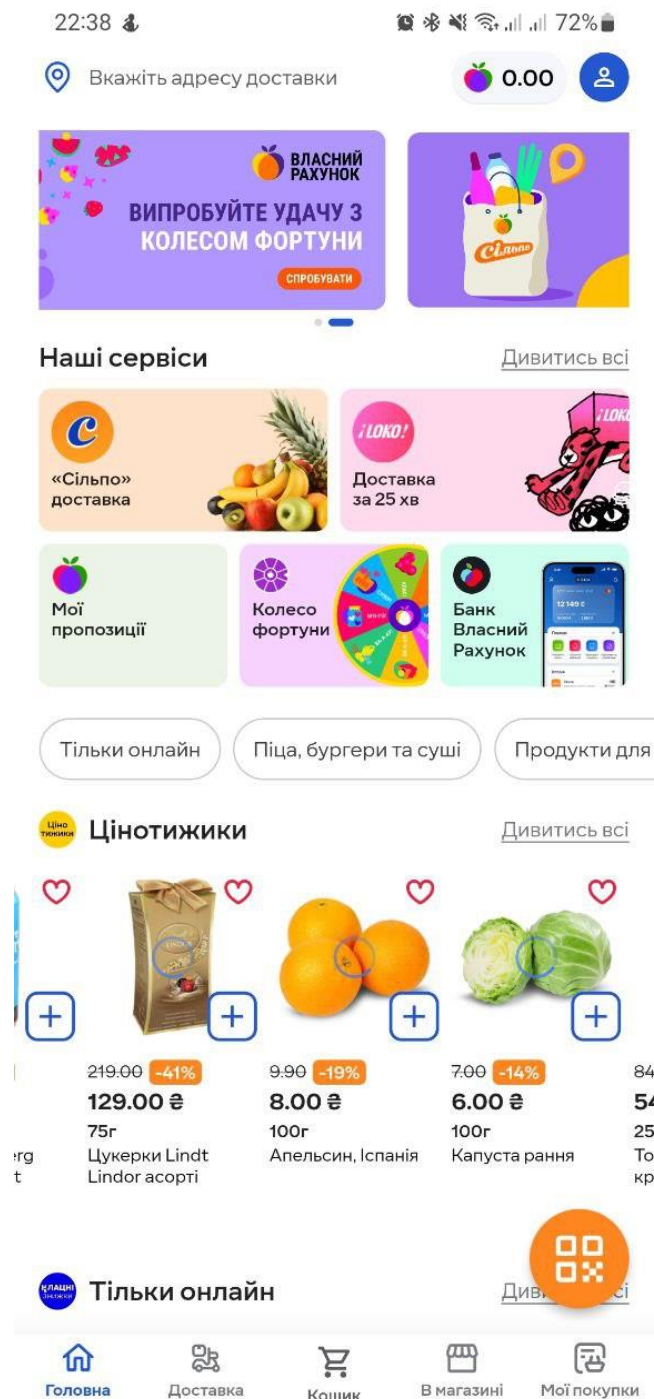


Рисунок 1.9 Інтерфейс головної сторінки додатку "Сільпо"

[← Кошик](#)

Ой, а тут порожньо 🙄

Ваш кошик порожній! Нумо виправляти
ситуацію ❤️

До покупок

Рисунок 1.10 Вид кошику в додатку



Корисності «Сільпо»



Колесо
фортун



Списки
Улюблене



Ск
Др

Історія покупок



Чеки



Замовлення



Йой, тут поки жодної покупочки!
Нумо виправляти ситуацію❤️



Головна



Доставка



Кошик



В магазині



Мої покупки

Рисунок 1.11 Вид меню «Мої покупки»

На рисунках(Рисунки 1.8-1.11) з додатку ми бачимо:

1. **Головний екран:** Головний екран додатку магазину "Сільпо" привітно вітає користувача та пропонує швидкий доступ до різних функцій та розділів додатку.

2. **Меню навігації:** У нижній частині екрана розташоване меню навігації, яке дозволяє перейти до основних розділів додатку, таких як "Головна", "Доставка", "Кошик", "В магазині", "Мої покупки".

3. **Пошук продуктів:** Функція пошуку дозволяє користувачеві швидко знаходити необхідні товари за їхніми назвами або категоріями.

4. **Перегляд продуктів:** Користувач може детально ознайомитися з інформацією про кожен продукт, включаючи ціну, опис, фото та наявність у магазині.

5. **Онлайн-замовлення:** Функція онлайн-замовлення дозволяє користувачеві зручно скласти кошик покупок, обирати зручний час доставки та вказувати адресу.

6. **Кошик покупок:** У розділі "Кошик" користувач може переглянути список обраних товарів, змінити їх кількість або видалити з кошика.

7. **Оплата та доставка:** Після оформлення замовлення користувачу пропонується вибрати метод оплати та вказати адресу доставки.

8. **Програма лояльності:** Додаток може мати інтегровану програму лояльності, за допомогою якої користувачі отримують бонуси та знижки за покупки.

9. **Повідомлення та акції:** Користувач може отримувати повідомлення про акції, нові надходження товарів та інші важливі повідомлення від магазину через спеціальний розділ або повідомлення в додатку.

Додаток Silpo розроблено з урахуванням потреб сучасного споживача та надає широкий спектр сервісів для комфортних та ефективних покупок.

У цьому розділі було розглянуто важливі аспекти, пов'язані з програмним забезпеченням для доставки товарів на замовлення в сучасному світі. Була

надана статистика та інформація щодо росту та використання таких сервісів серед населення України.

Далі розглянуто сучасний ринок доставки товарів на замовлення, на якому виділилися декілька основних лідерів, які є серйозними конкурентами один одному. Однак, слід відзначити, що ринок цих послуг в Україні є досить молодим, і це створює перспективи для створення нових стартапів та завоювання частки цього ринку.

Отже, можливість створення конкурентного аналогу для доставки товарів на замовлення в сучасному світі є надзвичайно актуальною та перспективною темою. У цьому розділі було визначено ключові аспекти, які слід врахувати при розробці такого проекту, щоб забезпечити його успішну реалізацію та конкурентоспроможність на ринку.

По-перше, розділ про архітектуру додатку визначає структуру та організацію компонентів системи, що є фундаментом для подальшого розроблення.

Користувацький інтерфейс відіграє ключову роль у забезпеченні зручності та доступності для кінцевого користувача, тоді як бізнес-логіка визначає стратегії та процедури, необхідні для ефективної роботи системи. База даних відповідає за збереження та управління інформацією, необхідною для оптимального функціонування додатку.

Крім того, в цьому розділі було розглянуто інші важливі аспекти, такі як безпека, швидкодія, масштабованість та інші технічні вимоги, які також відіграють ключову роль у успішному впровадженні та експлуатації додатку.

Узагальнюючи, цей розділ надає комплексний огляд важливих аспектів розробки конкурентоспроможного додатку для доставки товарів на замовлення, що дозволяє розробникам та командам проекту краще розуміти вимоги та виклики цієї галузі та ефективніше працювати над створенням успішного продукту.

РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА

2.1 Обрані технології та інструменти розробки

Для виконання свого проекту було обрано мову програмування Kotlin, базу даних RoomDB за допомогою якої будуть зберігатися всі дані користувачів додатку, проект буде розроблятися на Android платформу використовуючи Android Studio IDE від Google за допомоги інструментів від JetBrains.

RoomDB - це інноваційна технологія баз даних для Android, що забезпечує розробникам зручний та потужний інструмент для роботи з даними в їхніх додатках. Вона дозволяє ефективно управляти інформацією, забезпечуючи високу продуктивність та надійність. Однією з найбільших переваг RoomDB є її вбудована реалізація бази даних SQLite, що робить роботу з даними більш простою та ефективною. Ось деякі ключові особливості та переваги використання RoomDB:

1. Об'єктно-реляційне відображення (ORM): RoomDB надає можливість прив'язати об'єкти Java або Kotlin до таблиць бази даних SQLite, що дозволяє розробникам працювати з базою даних, використовуючи звичайні об'єктно-орієнтовані підходи.

2. Анотації для мапінгу даних: RoomDB використовує анотації для визначення сутностей бази даних, що дозволяє зберігати, оновлювати та видаляти дані в базі даних без написання складних SQL-запитів.

3. Схема бази даних на рівні компіляції: Однією з ключових переваг RoomDB є те, що вона перевіряє правильність структури бази даних на етапі компіляції, що дозволяє виявляти помилки ще до виконання додатку.

4. Підтримка LiveData та RxJava: RoomDB інтегрується з архітектурою компонентів Android, зокрема з LiveData, що дозволяє автоматично оновлювати інтерфейс користувача при зміні даних в базі даних. Також вона підтримує RxJava для асинхронної роботи з даними.

5. Міграції бази даних: RoomDB надає зручний механізм для міграції баз даних при зміні їх структури, що дозволяє зберігати користувацькі дані при оновленні додатку.

6. Використання DAO (Data Access Object): RoomDB використовує DAO для виконання операцій з базою даних, що робить код більш структурованим та зручним для розробників.

7. Підтримка асинхронних операцій: RoomDB підтримує асинхронні операції з базою даних, що дозволяє виконувати запити до бази даних в окремих потоках та уникати блокування основного потоку виконання.

У підсумку, RoomDB відкриває перед розробниками безліч можливостей для створення потужних та ефективних Android-додатків з використанням баз даних. З її допомогою можна легко зберігати та обробляти дані, забезпечуючи при цьому високу продуктивність та надійність роботи додатка. RoomDB - це не просто інструмент для роботи з базами даних, а справжня основа для створення сучасних та високоякісних Android-додатків.

Android Studio є офіційною інтегрованою середою розробки (IDE) для написання додатків під операційну систему Android. Ця IDE базується на платформі IntelliJ IDEA і надає розробникам ряд корисних функцій і інструментів, що поліпшують продуктивність під час створення Android-додатків:

- Гнучка система зборки на основі Gradle: Android Studio використовує Gradle для збирання проектів, що дозволяє зручно керувати залежностями та конфігурацією додатка.
- Швидкий та багатоцільовий емулятор: Інструменти емуляції Android Studio дозволяють розробникам тестувати свої додатки на різних віртуальних пристроях.
- Підтримка для різних пристроїв: Android Studio є єдиною IDE, в якій можна писати код для різних пристроїв під управлінням Android.
- Швидка зміна коду без перезапуску додатка: Розробники можуть вносити зміни у програмний код додатка без перезавантаження його.

- Інструменти для тестування: Android Studio має велику кількість інструментів і фреймворків для тестування працездатності додатків.
- Інструменти Lint: Використовуються для виявлення потенційних проблем в коді, таких як продуктивність, зручність використання, сумісність версій і інші.
- Підтримка C++ та NDK: Android Studio надає можливість розробляти додатки на мові програмування C++ і використовувати NDK (Native Development Kit).
- Вбудована підтримка Google Cloud Platform: Це спрощує роботу з такими сервісами, як Google Cloud Messaging та App Engine.

Загалом, Android Studio - це потужна інструментальна система, яка допомагає розробникам створювати високоякісні Android-додатки з більшою продуктивністю і зручністю.

Android Studio використовує систему збірки Gradle з плагіном Android для Gradle як основний інструмент для компіляції та збірки Android-додатків. Ця система збірки надає багато можливостей та гнучкість для налаштування та розширення проектів Android. Основні особливості використання Gradle в Android Studio включають таке:

- **Гнучкість налаштувань:** Gradle дозволяє розробникам налаштовувати збірку додатків на різних етапах процесу. Це включає налаштування залежностей, налаштування різних варіантів збірки для різних пристроїв та конфігурацій.
- **Підтримка кількох смуг:** За допомогою Gradle можна створювати декілька APK-файлів для одного і того ж додатка з різними функціями або налаштуваннями. Це дозволяє розробникам створювати різні версії свого додатка для різних цільових аудиторій.
- **Гнучка конфігурація ресурсів:** Ресурси (зображення, рядки тощо) можуть бути налаштовані для різних конфігурацій, таких як розмір екрану, орієнтація та інші параметри. Gradle дозволяє легко керувати різними варіантами ресурсів у вашому проекті.

- **Вбудована мовна підтримка:** Gradle використовує мову програмування Groovy для налаштування збірки, що робить конфігурацію проекту більш зрозумілою та доступною для розробників.

- **Підтримка плагінів:** Gradle має багато плагінів, включаючи той, який використовується для розробки під Android. Розробники можуть використовувати додаткові плагіни для розширення можливостей свого проекту.

Файли збірки для проектів Android Studio називаються `build.gradle`. Це текстові файли, в яких використовується синтаксис мови програмування Groovy для налаштування різних параметрів збірки та залежностей проекту. Користувачі можуть редагувати ці файли, щоб налаштувати збірку своїх додатків під свої потреби.

Android - це операційна система для мобільних пристроїв, яка була спроектована компанією Android Inc., а пізніше була придбана Google.

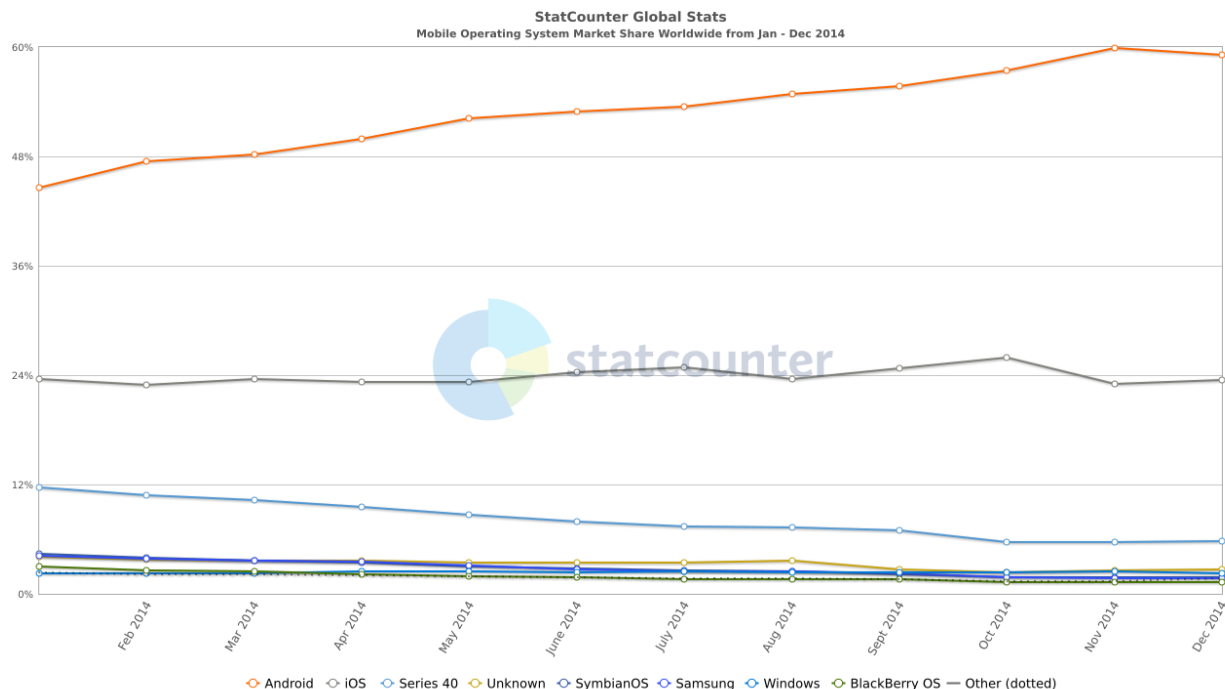
Операційна система базується на ядрі Linux та має власну реалізацію віртуальної машини Java, розроблену компанією Google. Ця платформа була вперше представлена широкому загалу у 2007 році та з того часу стала однією з найпоширеніших операційних систем для смартфонів, планшетів, годинників та інших мобільних пристроїв.

Основні характеристики Android:

- **Ядро Linux:** Android використовує ядро Linux як базову операційну систему, що забезпечує низькорівневий функціонал.
- **Java та Google Libraries:** Додатки для Android розробляються в основному мовою програмування Java та використовують бібліотеки, розроблені Google, для взаємодії з функціоналом пристрою.
- **Open Handset Alliance (ОНА):** Google створила альянс Open Handset Alliance для підтримки та розвитку платформи Android. Цей альянс об'єднує виробників апаратного забезпечення, операторів мобільного зв'язку та розробників програмного забезпечення.

- Android Native Development Kit (NDK): Для розробки додатків, написаних на мовах C та C++, Android пропонує Native Development Kit, який дозволяє переносити бібліотеки та компоненти на низькому рівні [7].

Статистика з <https://gs.statcounter.com/>:



Таблиця 1 Статистика популярності Android

У 2014 році Android була встановлена на 60% усіх проданих смартфонів у світі, але у 2024 році вже на 70.79% усіх смартфонів.

На конференції розробників в травні 2017 року Google заявила, що було активовано понад 2 мільярди пристроїв під управлінням Android за всю історію платформи.

Android визнана однією з найважливіших та найпопулярніших операційних систем для мобільних пристроїв у світі.

Справді, на початку свого існування Android ставки були низькі, це було зумовлено важкою конкуренцією на ринку мобільних операційних систем. У січні 2009 року, коли Android був тільки на початку свого шляху, він займав лише 0,66% ринку смартфонів. На той момент конкурентами були операційні системи, такі як SymbianOS від Nokia та iOS від Apple, які мали велику частку ринку.

Проте, Android швидко розвивався і випустив декілька оновлень, таких як Cupcake, Donut і Eclair, протягом 2009 року. Ці оновлення додали багато нових корисних функцій, включаючи підтримку Bluetooth 2.1, синхронізацію через різні сервіси, можливість додавати декілька облікових записів Google, підтримку спалаху камери, цифровий зум та апаратну підтримку графіки.

Ці оновлення допомогли Android утримати та посилити свої позиції на ринку. До січня 2011 року частка Android зросла до 14,61% ринку смартфонів, і він продовжував стрімко розвиватися. На сьогодні Android є однією з найпоширеніших операційних систем для мобільних пристроїв у світі [19].

Android має численні переваги, які варто враховувати:

- Інтернет-серфінг та інтеграція з сервісами Google: Android добре справляється із інтернет-серфінгом та інтеграцією з сервісами Google, що робить його популярним серед користувачів, які використовують ці послуги.
- Відкрита платформа: Android є відкритою платформою, що означає, що він доступний для багатьох розробників. Це сприяє розвитку великої спільноти розробників та створенню різноманітних додатків.
- Можливість вимкнення обмеження встановлення додатків з неперевірених джерел: Користувачі можуть вимкнути функцію, яка обмежує встановлення додатків з неперевірених джерел. Це дозволяє встановлювати програми з окремих джерел, що може бути корисним для тестування та розробки додатків.
- Підтримка різних апаратних платформ: Android підтримує різні апаратні платформи, включаючи ARM, MIPS і x86, що дозволяє його використовувати на різних типах пристроїв.
- Альтернативні магазини додатків: Окрім офіційного магазину Google Play, існують інші альтернативні магазини додатків, такі як Amazon Appstore, Uptodown App Store, Opera Mobile Store, GetApps, Mobogenie, F-Droid та багато інших. Це дає користувачам більше варіантів для завантаження додатків.

Android є важливою платформою для розробки додатків та має великий вплив на мобільний світ завдяки своїм перевагам [5].

Обравши операційну систему Android для подальшої розробки та створення сервісів та додатків, ви здійснили розумне рішення, оскільки Android має численні переваги, які ви вже навели. Його велика поширеність, простота розробки та тестування, а також можливість вибору з різних альтернативних магазинів додатків роблять його привабливим вибором для розробників.

Android також надає можливість створення різноманітних додатків, починаючи від ігор і закінчуючи корисними інструментами для щоденного використання. Ця платформа відкриває широкі можливості для реалізації творчості та ідей.

Kotlin - це сучасна мова програмування, яка була розроблена компанією "JetBrains" для роботи на платформі Java Virtual Machine (JVM). Її назву отримала на честь острова Котлін у фінському заливі. Метою розробників було створення мови, яка б була безпечнішою і більш простою в порівнянні з Java, але при цьому мала б бути сумісною з нею [3].

Основні переваги мови Kotlin включають:

- **Безпека:** Kotlin надає більшу безпеку програм завдяки статичній типізації та іншим механізмам, що допомагають уникнути типових помилок на етапі компіляції.
- **Простота та виразність:** Kotlin дозволяє писати чистий та лаконічний код. Він прибирає деякі зайві конструкції, які були властиві Java.
- **Сумісність з Java:** Kotlin повністю сумісний з Java, що означає, що ви можете використовувати існуючий Java-код та бібліотеки в проектах на Kotlin без проблем.
- **Висока продуктивність:** Мова має швидку компіляцію, що полегшує розробку та відлагодження коду.
- **Активна підтримка:** Kotlin активно підтримується та оновлюється, а також має широке спільноту розробників.
- **Підтримка Android:** Kotlin офіційно підтримується Android, і він став все більш популярним для розробки мобільних додатків на цій платформі.

- Розширена функціональність: Kotlin має ряд функцій, які полегшують роботу з колекціями, роблять код більш ефективним та зрозумілим.

Kotlin - мова програмування з відкритим вихідним кодом, що означає, що всі можуть вільно користуватися та вносити зміни у її код. Ця відкритість є великою перевагою, оскільки будь-які помилки можна швидко виявити і повідомити розробникам.

Синтаксис Kotlin вражає своєю інноваційністю. Він поєднує різні елементи з інших мов, таких як JavaScript, Pascal, TypeScript, F#, Go, Scala, C ++, Java, C#.

У Kotlin, типи даних вказуються після назви змінних і параметрів за допомогою двокрапки. Крапка з комою як роздільник операторів не завжди обов'язкова, і в більшості випадків для завершення виразу досить перенесення рядка.

Окрім об'єктно-орієнтованого підходу, Kotlin також підтримує процедурний стиль програмування з використанням функцій. Точкою входу в програму є функція "main", яка приймає масив параметрів командного рядка.

З цих причин Kotlin став однією з переваг для розробників Android-додатків та інших проектів, що використовують JVM. Ця мова набирає популярність і має потенціал стати однією з провідних мов програмування в майбутньому.

Отже, було розглянуто обрані технології для розробки програмної системи підтримки доставки товарів на замовлення. Почнемо з Firebase, яка надає широкий спектр інструментів для розробки додатків, включаючи аналітику, аутентифікацію, зберігання даних, роботу з конфігурацією та інші сервіси. Firebase спрощує роботу розробників, роблячи багато типових завдань легкими та швидкими завдяки хмарному зберіганню та масштабуванню.

Після цього розглянули Android Studio, офіційну інтегровану середовище розробки (IDE) для створення додатків для операційної системи Android. Android Studio пропонує розробникам потужний редактор коду, гнучку систему збірки на основі Gradle, швидкий емулятор та багато інших інструментів, які покращують продуктивність розробки.

Далі ми розглянули саму операційну систему Android та її переваги порівняно з іншими операційними системами. Android є популярною ОС для мобільних пристроїв та користується великою популярністю по всьому світу.

Ми проаналізували статистичні дані з порталу statcounter.com, які свідчать про широке поширення Android серед користувачів.

Потім ми зосередили увагу на мові програмування Kotlin. Ми виявили, що Kotlin - це сучасна мова програмування, яка швидко набирає популярність, особливо серед розробників Android-додатків. Мова має багато переваг і підтримується активною спільнотою розробників.

У заключенні, ми розглянули інтегровану середу розробки Android Studio, яка є стандартним інструментом для розробки додатків для Android. Android Studio надає розробникам багато інструментів та функцій для зручної та продуктивної роботи над проектами.

2.2. Опис архітектури додатку кур'єрської служби

Архітектура додатку кур'єрської служби є фундаментальним складовим його успішної розробки та функціонування. Вона визначає структуру, взаємозв'язки та взаємодію між різними компонентами системи, що дозволяє забезпечити ефективність, надійність та зручність використання додатку як для користувачів, так і для адміністраторів.

Архітектура додатку кур'єрської служби складається з декількох ключових компонентів:

1. Користувацький інтерфейс:

Інтерфейс в додатку відіграє ключову роль у спрощенні взаємодії між користувачами та додатком. Він слугує мостом між людиною та функціоналом програми, сприяючи комфортній та інтуїтивно зрозумілій взаємодії.

Користувацький інтерфейс є основним механізмом взаємодії між користувачем та функціоналом додатку. Він забезпечує зручність та доступність

всіх функцій програми, зробивши процес роботи з додатком ефективним та приємним для користувачів.

2. Бізнес-логіка

Бізнес-логіка в додатку має важливу роль у реалізації основних операцій та функцій, пов'язаних з організацією доставки та управлінням замовленнями.

Отже, бізнес-логіка в цьому додатку визначає стратегії та процедури, які керують ключовими бізнес-процесами, забезпечуючи оптимальне функціонування та ефективність.

3. База даних

Для збереження даних про замовлення, користувачів, кур'єрів та іншої інформації використовується база даних. Зазвичай використовуються реляційні бази даних або NoSQL рішення, в залежності від вимог проекту. Ці дані взаємодіють з різними частинами додатку та іншими компонентами системи, надаючи можливість користувачам та адміністраторам доступ до необхідної інформації для оптимального функціонування.

Ці компоненти взаємодіють між собою, утворюючи стабільну та продуктивну систему доставки, яка задовольняє потреби як користувачів, так і провайдерів послуг. Вони спільно працюють для забезпечення якісної та надійної роботи додатку кур'єрської служби.

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

3.1. Обґрунтування вибору мови програмування

Вибір мови програмування Kotlin для створення додатку кур'єрської служби має кілька обґрунтованих переваг, про які було сказано раніше:

1. **Безпека та Простота:** Kotlin є статично типізованою мовою програмування, що дозволяє виявляти помилки на етапі компіляції, забезпечуючи високу рівень безпеки в програмі. Також, вона пропонує чистий та лаконічний синтаксис, який робить процес розробки менш складним.

2. **Інтероперативність з Java:** Kotlin повністю сумісна з Java, що означає, що можна використовувати бібліотеки Java безпосередньо в коді на Kotlin та навпаки. Це дає можливість поступово переходити з Java на Kotlin, не переписуючи весь код разом.

3. **Підтримка Google:** Kotlin отримує активну підтримку від Google, що відображається у високому рівні документації, зручності інтеграції з Android Studio, а також впровадженні нововведень.

4. **Швидкість розробки:** Один з ключових аргументів - швидкість розробки на Kotlin. Ця мова має короткий код, менш кількість шаблонного коду, що дозволяє розробникам швидше писати програми та проводити рефакторинг.

5. **Активне співтовариство:** Kotlin має велике та активне співтовариство, яке надає корисні бібліотеки та відповіді на питання, що полегшує процес розробки та вирішення можливих проблем.

Загалом, вибір Kotlin для створення додатку кур'єрської служби визначається його безпекою, простотою, швидкістю розробки та підтримкою на рівні платформи Android, що сприяє створенню надійного та ефективного програмного забезпечення.

3.2. Створення UI/UX

У цьому розділі наведено висновки стосовно проектування і створення інтерфейсу користувача та взаємодії з користувачем в додатку кур'єрської служби. Описано процес розробки UI/UX, включаючи вимоги до функціоналу, концепцію дизайну, розробку wireframes та mockups, проектування графічного інтерфейсу та його реалізацію в коді.

Перед проектуванням дизайну важливо ретельно визначити потреби та очікування користувачів. Проведений аналіз дозволив з'ясувати основні функціональні вимоги та необхідні елементи для забезпечення зручного та ефективного використання додатку.

Етапи створення UI/UX:

1. Розроблення Концепції та Макету (Wireframes):

Перший етап полягає в створенні концепції додатку та його користувацького інтерфейсу. Важливо визначити структуру додатку, план розміщення елементів та його функціональність. Це може бути представлено через wireframes, або прототипи екранів додатку.

2. Графічний дизайн та Реалізація:

Далі слід розробляти графічний дизайн елементів інтерфейсу. Графічний дизайн користувацького інтерфейсу (UI) вимагає ретельного проектування та реалізації компонентів, які використовуються в додатку. У розробці додатків для Android застосовують різні ресурси, такі як XML-файли, які визначають макети, стилі, ресурси зображень тощо.

3. Інтерактивність та Функціональність:

Створення взаємодії між графічними елементами та функціональністю. Це включає визначення реакцій на події користувача, наприклад, обробку натискання кнопок, заповнення полів тощо. Створення інтерактивного та привабливого інтерфейсу для користувача є однією з ключових задач. Інтерфейс забезпечує зручність взаємодії користувача з додатком. Функціональність

визначається набором можливостей, які пропонуються користувачам. У цьому додатку, функціональність обумовлена концепцією кур'єрської доставки.

4. Тестування:

Після створення дизайну створюються інтерактивні прототипи, які дозволяють тестувати функціональність та взаємодію інтерфейсу. Тестування здійснюється з допомогою фокус-груп або тестування з реальними користувачами для збору зворотного зв'язку та виявлення можливих проблем. В даному випадку тестування буде проходити без окремо створених прототипів та під час розробки й після.

3.3. Робота з базою даних

У цьому кур'єрському додатку база даних є важливою складовою, яка забезпечує ефективне зберігання та організацію даних про замовлення, користувачів, адреси доставки та іншу інформацію, необхідну для функціонування сервісу доставки.

Основною структурою даних у базі є таблиці, які відображають основні сутності додатку, такі як замовлення, користувачі, адреси та інші. Кожна таблиця має відповідний набір полів, що визначають типи даних та взаємозв'язки між ними.

RoomDB дозволяє легко визначати схему бази даних за допомогою анотацій у класах сутностей та інтерфейсах доступу до даних. Це дозволяє швидко та зручно створювати та модифікувати схему бази даних за необхідності, що дозволяє легко адаптувати додаток під змінні потреби.

Однією з важливих особливостей RoomDB є його підтримка асинхронного доступу до даних. Це означає, що операції зчитування та запису в базу даних відбуваються асинхронно, що дозволяє уникнути блокування інтерфейсу користувача та забезпечує плавну реакцію додатку на дії користувача.

Крім того, RoomDB має вбудовану підтримку транзакцій, яка забезпечує консистентність даних при виконанні групи пов'язаних операцій. Це особливо важливо для операцій, які вимагають атомарності та цілісності даних, таких як оновлення статусу замовлення або обробка платежів.

Усі ці особливості роблять RoomDB ідеальним вибором для забезпечення ефективної та надійної роботи бази даних у кур'єрському додатку. Він дозволяє швидко розробляти, масштабувати та підтримувати додаток, забезпечуючи при цьому високу продуктивність та надійність.

3.4 Програмна реалізація елементів алгоритму

В цьому пункті буде розглянуто програмну реалізацію елементів алгоритму системи підтримки доставки товарів на замовлення та елементів інтерфейсу. Один з ключових аспектів цього алгоритму - це взаємодія із сервером та обмін даними між різними частинами системи.

Для забезпечення цього використовується RoomDB, яка є локальною базою даних для зберігання та управління інформацією про замовлення, користувачів та інші сутності, необхідні для роботи додатку. RoomDB є частиною архітектурного компонента Jetpack і надає простий та зручний спосіб робити операції з базою даних без прямого використання SQL.

Реалізація взаємодії БД з додатком з RoomDB включає такі кроки:

- Створення та налаштування бази даних: Для використання RoomDB у додатку потрібно спочатку створити базу даних та визначити сутності, які будуть зберігатися в цій базі. Кожна сутність відображається як таблиця в базі даних, а поля об'єктів стають колонками цих таблиць.
- Реалізація функцій доступу до даних: Після створення сутностей потрібно реалізувати Data Access Object (DAO), який надає методи для виконання операцій з базою даних, таких як вставка, оновлення, видалення та отримання даних. Ці методи будуть викликатися з інших частин додатку для взаємодії з базою даних.

- Інтеграція з додатком: Після написання DAO потрібно підключити RoomDB до додатку. Це включає створення базового класу додатку, який містить посилання на базу даних та встановлення необхідних налаштувань для RoomDB.
- Використання в додатку: Після налаштування RoomDB можна почати використовувати його в додатку. Це означає виклик методів DAO для зберігання, отримання та оновлення даних в базі даних.

RoomDB дозволяє легко працювати з базою даних в Android-додатках, забезпечуючи при цьому високу продуктивність та надійність операцій з даними. Використання RoomDB у кур'єрському додатку дозволяє ефективно керувати інформацією про замовлення, користувачів та інші сутності, забезпечуючи швидкий доступ до неї та масштабованість системи.

Реалізація інтерфейсу користувача:

- На головному екрані додатку користувач має доступ до основних функцій системи. Інтерфейс включає інтуїтивно зрозумілі елементи керування, такі як кнопки "Розпочати доставку", "Мої замовлення", "Профіль". Головний екран також відображає важливі повідомлення та акції. При переході до деталей замовлення, користувач може переглядати інформацію про товар, адресу доставки та інші деталі. Інтерфейс містить кнопку "Почати доставку", яка ініціює процес доставки. Екран профілю містить основну інформацію про користувача, його історію замовлень, статус та інші персональні налаштування. Користувач може змінювати свої особисті дані та перевіряти статус своїх доставок.

РОЗДІЛ 4. ТЕСТУВАННЯ РОБОТИ ЗАСТОСУНКУ

Для забезпечення надійності та стабільності роботи кур'єрського додатку було проведено ретельне тестування різних його аспектів. Тестування включає в себе перевірку функціональності, інтерфейсу користувача, а також взаємодії з базою даних та зовнішніми сервісами.

1. Функціональне тестування:

Першочергово були протестовані всі функції додатку згідно з їхнім призначенням. Це включало у себе:

- Тестування процесу реєстрації та входу користувачів у систему (Рисунок 4.1).

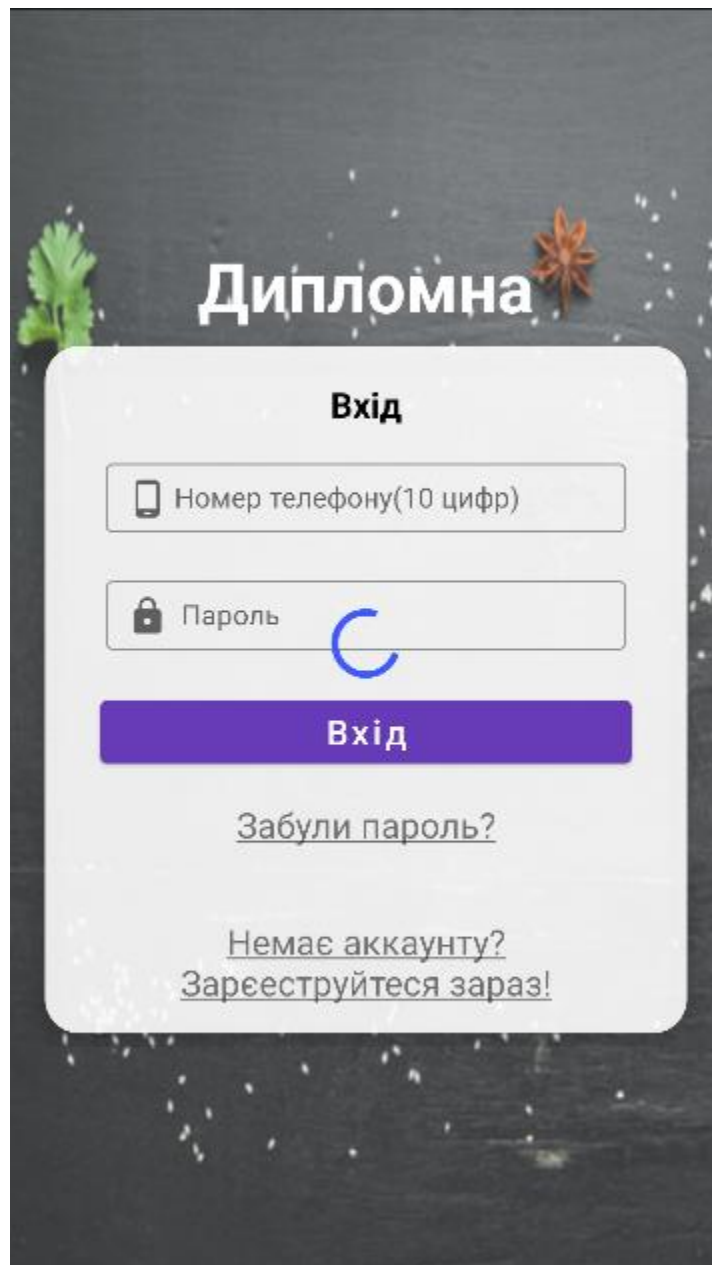


Рисунок 4.1 Вхід в застосунок

- Перевірку можливості створення та відправлення замовлень користувачами(Рисунок 4.2).

Замовлення з : 'Назва ресторану'

🍴 Картопля по-селянські € 60

€ Всього: 60 [Замовити](#)

Рисунок 4.2 Замовлення

- Перевірку роботи алгоритмів обробки замовлень (Рисунок 4.3).

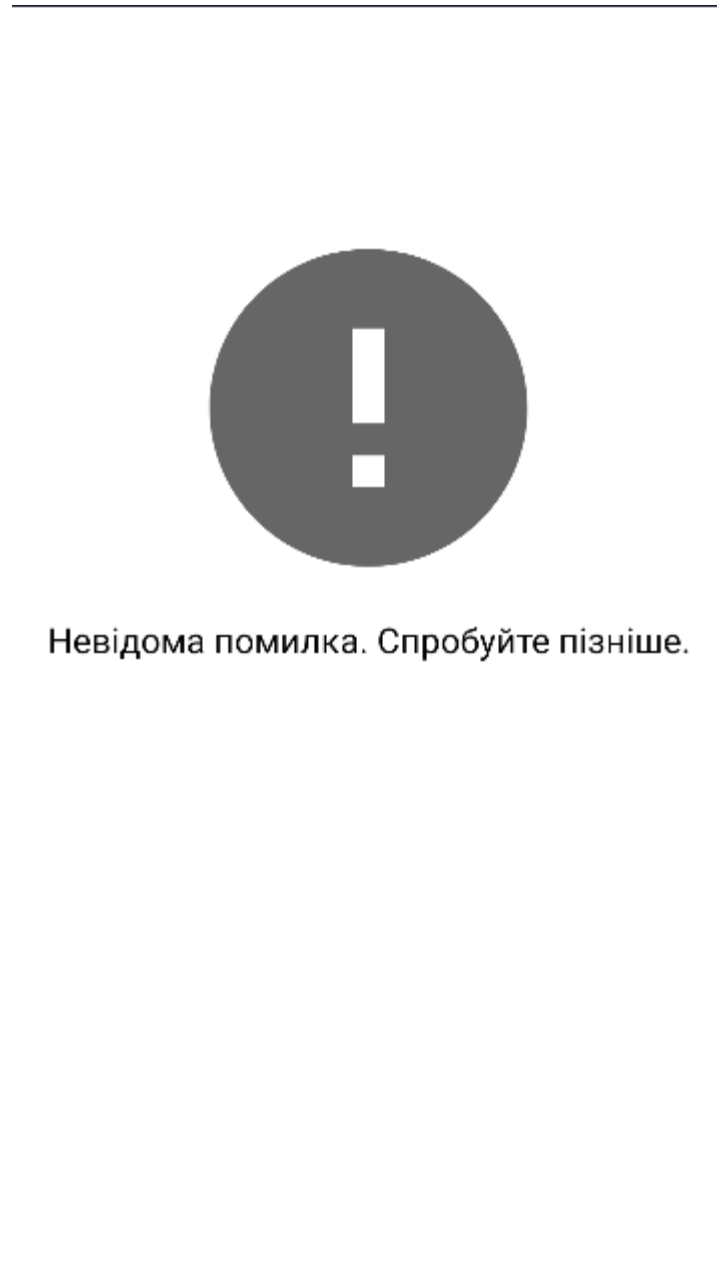


Рисунок 4.3 Помилка, якщо замовлення не оброблене

Якщо замовлення обробиться правильно, то алгоритм все опрацює і покаже повідомлення(Рисунок 4.4).

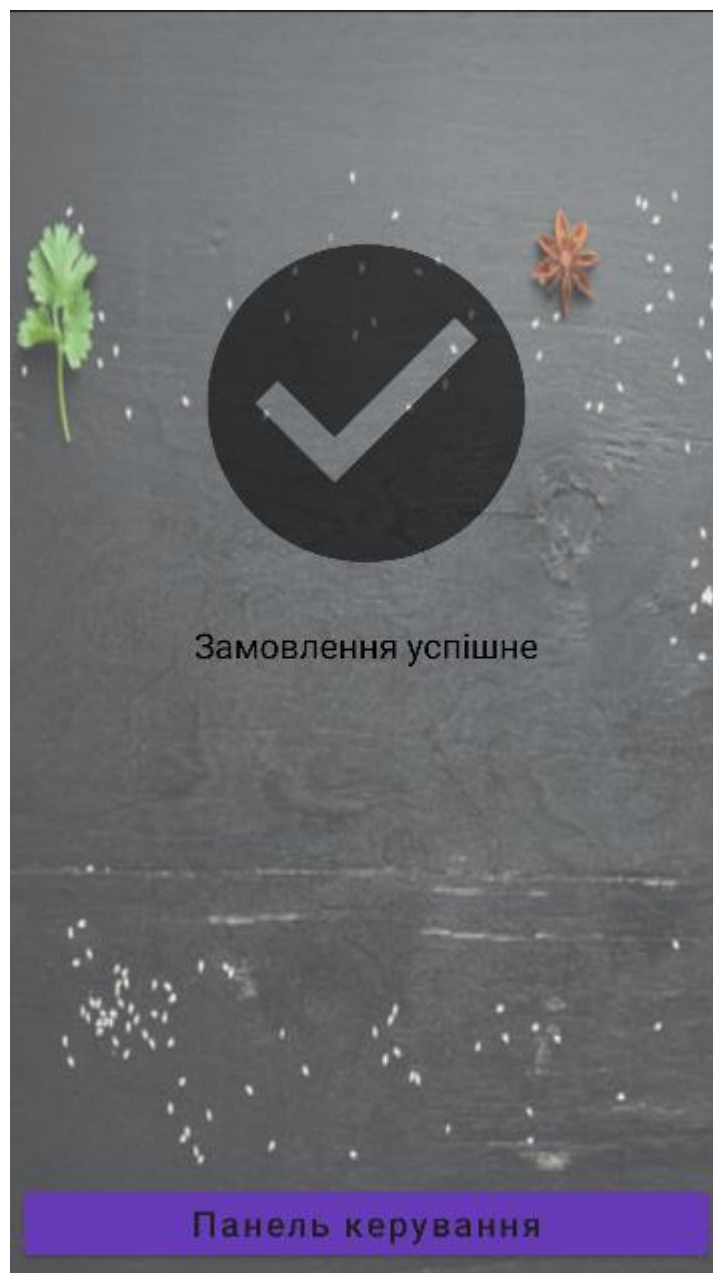


Рисунок 4.4 Успішність замовлення

2. Тестування інтерфейсу користувача:

Було проведено тестування інтерфейсу додатку з метою перевірки його зручності та інтуїтивності для користувачів.

Особлива увага приділялася:

- Перевірці навігації між різними екранами та функціональними блоками додатку.
- Оцінці візуального оформлення та взаємодії з елементами інтерфейсу.
- Перевірці відповідності інтерфейсу стандартам дизайну для забезпечення однорідного вигляду на різних пристроях.
- Перевірці інтерфейсу на наявність лексичних помилок та проблем з відображенням тексту.

3. Тестування взаємодії з базою даних:

Для перевірки коректності роботи з базою даних RoomDB було проведено наступні види тестів:

- Тестування процесу збереження, оновлення та видалення даних у базі.
- Перевірка відновлення даних після перезапуску додатку.
- Тестування відповідності структури бази даних вимогам та специфікаціям проекту.

ВИСНОВКИ

Отже, вибір технологій та інструментів був обґрунтований на основі їхньої ефективності та спроможностей у реалізації функціоналу додатку. Робота пройшла апробацію, по ній було написано тези та у ході виконання даної дипломної роботи були реалізовані наступні завдання розробки програмної системи для підтримки доставки товарів на замовлення, а саме:

1. Зроблено огляд відомого програмного забезпечення для кур'єрської доставки їжі та товарів, виявлено позитивні та негативні сторони.
2. Розроблено алгоритм Android додатку для замовлення їжі.
3. Програмно реалізовано елементи Android додатку для використання користувачами.
4. У результаті тестування виявлено, що додаток працює тільки з підключенням до мережі Інтернет.

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Kotlin for Android Developers: Learn Kotlin the Easy Way While Developing an Android App by Antonio Leiva March 21, 2016.
2. Kotlin URL: <https://kotlinlang.org/>.
3. Kotlin Docs URL: <https://kotlinlang.org/docs/home.html/>.
4. Java URL: <https://www.java.com/en/>.
5. RoomDB URL: <https://developer.android.com/reference/android/arch/persistence/room/RoomDatabase/>.
6. Android docs URL: <https://developer.android.com/docs/>.
7. Android Studio URL: <https://developer.android.com/studio/>.
8. Gradle URL: <https://gradle.org/>.
9. Gradle Android URL: <https://gradle.org/guides/?q=Android/>.
10. Android NDK URL: <https://developer.android.com/ndk/>.
11. GS-Statcounter URL: <https://gs.statcounter.com/>.
12. Russian invasion of Ukraine URL: <https://war.ukraine.ua/>.
13. Інтеграція Glovo та Bolt Food URL: <https://posteat.ua/news/integraciya-glovo-ta-bolt-food-chomu-lideri-rinku-roblyat-ce-same-zaraz/>.
14. Нова Пошта News URL: <https://novaposhta.ua/news/>.
15. Glovo URL: <https://glovoapp.com/ua/uk/>.
16. Mister.am URL: <https://misteram.com.ua/>.
17. Bolt Food URL: <https://bolt.eu/uk-ua/food/>.
18. Silpo News URL: <https://silpo.ua/press-center/press-releases/merezha-silpo-znovu-1-sered-supermarketiv-v-ukrayini-u-fmcg/>.
19. Android Statistics URL: <https://www.bankmycell.com/blog/how-many-android-users-are-there/>.

ДОДАТОК А. Код програми

SplashActivity.kt:

```
class SplashActivity : AppCompatActivity() {

    private lateinit var sharedPreferences: SharedPreferences

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        sharedPreferences = loadSharedPreferences()

        setContentView(R.layout.activity_splash)
        val isLoggedIn = sharedPreferences.getBoolean(isLoggedInKey, false)

        Handler(Looper.getMainLooper()).postDelayed({
            navigateFromSplash(isLoggedIn)
        }, 2000)
    }

    private fun navigateFromSplash(isLoggedIn: Boolean) {
        val intentFromSplash: Intent = if (isLoggedIn) {
            Intent(this@SplashActivity, DashboardActivity::class.java)
        } else {
            Intent(this@SplashActivity, LoginActivity::class.java)
        }
        startActivity(intentFromSplash)
        ActivityCompat.finishAffinity(this@SplashActivity)
    }
}
```

RegistrationActivity.kt:

```

class RegistrationActivity : AppCompatActivity() {

    private lateinit var sharedPreferences: SharedPreferences

    private lateinit var progressBar: ProgressBar
    private lateinit var etName: TextInputEditText
    private lateinit var etEmail: TextInputEditText
    private lateinit var etMobileNumber: TextInputEditText
    private lateinit var etDeliveryAddress: TextInputEditText
    private lateinit var etPassword: TextInputEditText
    private lateinit var etConfirmPassword: TextInputEditText
    private lateinit var btnRegister: Button

    private lateinit var networkTaskListener: NetworkTask.NetworkTaskListener

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        sharedPreferences = loadSharedPreferences()

        setContentView(R.layout.activity_registration)
        title = "Register Yourself"
        supportActionBar?.setDisplayHomeAsUpEnabled(true)
        supportActionBar?.setDisplayShowHomeEnabled(true)

        progressBar = findViewById(R.id.progressBar)
        progressBar.hide()
        etName = findViewById(R.id.etName)
        etEmail = findViewById(R.id.etEmail)
        etMobileNumber = findViewById(R.id.etMobileNumber)
    }
}

```

```

etDeliveryAddress = findViewById(R.id.etDeliveryAddress)
etPassword = findViewById(R.id.etPassword)
etConfirmPassword = findViewById(R.id.etConfirmPassword)
btnRegister = findViewById(R.id.btnRegister)

btnRegister.setOnClickListener {
    tryRegister()
}

}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    if (item.itemId == android.R.id.home) {
        navigateToLoginActivity()
    }
    return super.onOptionsItemSelected(item)
}

override fun onBackPressed() {
    super.onBackPressed()
    navigateToLoginActivity()
}

private fun tryRegister() {
    val name = etName.text.toString()
    val email = etEmail.text.toString()
    val mobile = etMobileNumber.text.toString()
    val address = etDeliveryAddress.text.toString()
    val password = etPassword.text.toString()
    val confirmPassword = etConfirmPassword.text.toString()

```

```

when (checkValidInputs(name, email, mobile, address, password,
confirmPassword)) {
    InputState.INVALID_NAME -> {
        etName.error = "Name should contain minimum 3 characters."
    }
    InputState.INVALID_EMAIL -> {
        etEmail.error = "Invalid Email."
    }
    InputState.INVALID_MOBILE -> {
        etMobileNumber.error = "Invalid mobile number."
    }
    InputState.INVALID_ADDRESS -> {
        etDeliveryAddress.error = "Field should not be empty."
    }
    InputState.INVALID_PASSWORD -> {
        etPassword.error = "Password should contain minimum 6 characters."
    }
    InputState.PSW_NO_MATCH -> {
        etConfirmPassword.error = "Passwords do not match!"
    }
    InputState.OKAY -> {
        sendNetworkRequest(name, mobile, password, address, email)
    }
    else -> {
        showToast("Unknown Input state.")
    }
}
}

```

```

private fun checkValidInputs(
    name: String, email: String, mobile: String,
    address: String, password: String, confirmPassword: String
): InputState {
    return when {
        name.trim().length < 3 -> {
            InputState.INVALID_NAME
        }
        !email.matches(emailRegex) -> {
            InputState.INVALID_EMAIL
        }
        mobile.trim().length != 10 -> {
            InputState.INVALID_MOBILE
        }
        address.isEmpty() -> {
            InputState.INVALID_ADDRESS
        }
        password.length < 6 -> {
            InputState.INVALID_PASSWORD
        }
        password.compareTo(confirmPassword) != 0 -> {
            InputState.PSW_NO_MATCH
        }
        else -> {
            InputState.OKAY
        }
    }
}

```

```

private fun sendNetworkRequest(

```

```

        name: String, mobileNumber: String, password: String, address: String, email:
String
    ) {
        if (ConnectionManager().checkConnectivity(this@RegistrationActivity)) {
            progressLayout.show()
            btnRegister.disable()

            setupNetworkTaskListener()

            val jsonParams = JSONObject()
            jsonParams.put("name", name)
            jsonParams.put("mobile_number", mobileNumber)
            jsonParams.put("password", password)
            jsonParams.put("address", address)
            jsonParams.put("email", email)

            NetworkTask(networkTaskListener).makeNetworkRequest(
                this@RegistrationActivity, Request.Method.POST, REGISTER,
jsonParams
            )
        } else {
            noInternetDialog(this@RegistrationActivity)
        }
    }

private fun setupNetworkTaskListener() {
    networkTaskListener =
        object : NetworkTask.NetworkTaskListener {
            override fun onSuccess(result: JSONObject) {
                progressLayout.hide()

```

```

try {
    val returnObject = result.getJSONObject("data")
    val success = returnObject.getBoolean("success")

    if (success) {
        val data = returnObject.getJSONObject("data")
        val userId = data.getString("user_id")
        val userName = data.getString("name")
        val userEmail = data.getString("email")
        val userMobileNumber = data.getString("mobile_number")
        val userAddress = data.getString("address")
        saveToPreferences(
            userId, userName, userEmail, userMobileNumber, userAddress
        )
        // Since, we are navigating from RegistrationActivity and stopping
the activity(finish())
        // after navigation, we do not need to enable the disabled Register
button.

        navigateToDashboardActivity()
    } else {
        btnRegister.enable()
        val errorMessage = returnObject.getString("errorMessage")
        showToast(errorMessage)
    }
} catch (e: Exception) {
    btnRegister.enable()
    showToast("Error: ${e.localizedMessage}")
}
}

```

```

        override fun onFailed(error: VolleyError) {
            btnRegister.enable()
            showToast("Error: ${error.localizedMessage}")
        }
    }
}

```

```

private fun saveToPreferences(
    userId: String, name: String, email: String, mobile: String, address: String
) {
    sharedPreferences.edit()
        .putBoolean(isLoggedInKey, true)
        .putString(userIdKey, userId)
        .putString(userNameKey, name)
        .putString(userMobileKey, mobile)
        .putString(userEmailKey, email)
        .putString(userAddressKey, address)
        .apply()
}

```

```

private fun navigateToLoginActivity() {
    val intent = Intent(this@RegistrationActivity, LoginActivity::class.java)
    startActivity(intent)
    ActivityCompat.finishAffinity(this@RegistrationActivity)
}

```

```

private fun navigateToDashboardActivity() {
    val intent = Intent(this@RegistrationActivity, DashboardActivity::class.java)
    startActivity(intent)
}

```

```
        ActivityCompat.finishAffinity(this@RegistrationActivity)
    }
}
```