

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування  
Кафедра інформаційних технологій

**Кваліфікаційна робота бакалавра**

на тему: Аналіз вимог UI/UX до сучасних мобільних додатків

---

Виконав студент групи К-21і  
спеціальності 122 Комп'ютерні науки  
Преснова Тетяна Василівна

---

Керівник ст. викладач  
Штефан Наталія Зінов'ївна

---

Консультант док. техн. наук,  
професор  
Казакова Надія Феліксівна

---

Рецензент к.т.н., доцент  
Сергієнко Андрій  
Володимирович

---

Одеса 2023

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ .....	5
ВСТУП.....	6
1 АНАЛІТИЧНИЙ РОЗДІЛ.....	7
1.1 Інтерфейс користувача (ІК) .....	7
1.2 Труднощі проектування ІК .....	10
1.3 Етапи проектування ІК, орієнтованого на користувача .....	14
1.4 Аналітичний огляд аналогів .....	16
1.5 Інструментальні засоби розробки.....	20
2 РОЗРОБКА ТЕХНІЧНОГО ПРОЕКТУ .....	27
2.1 Вимоги до мобільних додатків .....	27
2.2 Проектування.....	30
2.3 Відмінності дизайну під iOS від дизайну під Android.....	38
3 РОЗРОБКА UI.....	41
ВИСНОВКИ .....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	58

## **ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ**

ІК – інтерфейс користувача

ПЗ – програмного забезпечення

UX – User Experience

UI – User Interface

Android – операційна система

iOS – операційна система

Figma – сервіс для проектування макетів

Java – мова програмування

Material Design концепцію дизайну інтерфейсу користувача

Swift – мова програмування

## ВСТУП

На одному смартфоні у середньому встановлено 40 додатків. А 80% часу користування смартфоном займають саме мобільні застосунки. Тому кожного дня розробники намагаються максимально покращити і полегшити весь мобільний інтерфейс, щоб користувачу було як можна краще і скоріше розібратися у тому чи іншому додатку. Мобільна розробка неухильно йде слідом за веб-розробкою. Таким чином, роль шаблонів дизайну мобільного інтерфейсу користувача стає більш помітною.

В даній роботі розглянемо саме вимоги до інтерфейсу мобільних додатків. Інтерфейс це перше із чим користувач зіштовхується у додатку і немає різниці наскільки корисний цей додаток, якщо інтерфейс у ньому незрозумілий та незручний у використанні.

Акцентувати увагу будемо на розгляді:

- різниці між мобільним додатком та сайтом;
- видів додатків;
- UA/UX дизайну в цілому;
- відмінностей дизайну під iOS та Android;
- найважливіших нюансів розробки інтерфейсу мобільних додатків.

Не вивчивши того, що з погляду користувача, повинна робити система, неможливо сформуванати реалістичний погляд на проектування ефективного інтерфейсу, тому дипломний проект націлений на вивчення саме візуалу додатку. Виходячи з цього, метою дипломної роботи є розробка інтерфейсу мобільного додатку, який буде розкривати усі потреби користувача та ефективно на нього впливати.

# 1 АНАЛІТИЧНИЙ РОЗДІЛ

## 1.1 Інтерфейс користувача (ІК)

Інтерфейс користувача (UI, від англ. User Interface) – це спосіб взаємодії користувача з комп'ютерною програмою, пристроєм або системою. Він є набором елементів і компонентів, які дозволяють користувачеві взаємодіяти з програмним або апаратним забезпеченням, робити вибори, вводити дані та отримувати інформацію.

Інтерфейс користувача включає різні елементи, такі як:

1. Графічні елементи: кнопки, меню, поля введення, перемикачі, списки тощо. Вони надають користувачеві спосіб взаємодії з програмою чи пристроєм.

2. Інформаційні елементи: текстові позначки, іконки, індикатори стану тощо. Вони надають інформацію користувачеві про поточний стан системи або виконання операцій.

3. Навігаційні елементи: посилання, вкладки, меню навігації та ін. Вони дозволяють користувачеві переміщатися за різними розділами та функціями програми.

4. Мультимедіа елементи: зображення, відео, звукові ефекти і т.д. Вони збагачують досвід користувача і надають додаткові засоби комунікації та взаємодії.

Мета інтерфейсу користувача – зробити взаємодію користувача з програмою або пристроєм максимально інтуїтивним, зручним і ефективним. Добре спроектований інтерфейс користувача дозволяє користувачам легко розуміти функціональність і виконувати необхідні дії без зайвої праці або плутанини.

Інтерфейс користувача комп'ютерного додатку включає різноманітні компоненти, які забезпечують сприйняття та взаємодію користувача з програмою. Це включає засоби для відображення інформації, такі як графічні елементи, текстові поля та інші формати та коди, які дозволяють відображати дані. Також важливими елементами є командні режими та мова взаємодії, які

дозволяють користувачу взаємодіяти з програмою через команди та інструкції.

Для введення та виведення даних використовуються різні пристрої та технології, такі як клавіатура, миша, сенсорний екран або голосові команди. Діалоги та взаємодія між користувачем та комп'ютером відбуваються через взаємодійні елементи, які дозволяють користувачу виконувати операції, вводити дані та отримувати відповіді від системи. Зворотній зв'язок з користувачем, наприклад, повідомлення про помилки або підтвердження виконання дії, також є важливою складовою інтерфейсу.

Інтерфейс користувача також може надавати підтримку прийняття рішень в конкретній предметній області, надаючи користувачу необхідну інформацію та функціональність для виконання завдань. Також важливим є порядок використання програми та документації, щоб користувач міг ефективно навігувати та використовувати програму.

Отже, інтерфейс користувача – це не лише зовнішній вигляд програми, але й всі елементи, що дозволяють користувачу сприймати та взаємодіяти з програмою в цілому. [1].

В дійсності, інтерфейс користувача (ІК) об'єднує в собі всі елементи і компоненти програми, які здатні впливати на взаємодію користувача з програмним забезпеченням. На загальному рівні проектування ІК здається простою справою. Спроекувати будь-який фрагмент ІК не складно, однак мало хто може впоратись з розробкою повного інтерфейсу.

Процес проектування ІК – це складний, нелінійний, недетермінований і неортогональний процес. Складність – звичайна властивість програмного забезпечення (ПЗ), але в ще більшому степені це стосується ІК – через численні фактори та невизначеності, які впливають на його розробку.

Процес проектування інтерфейсу користувача (UI) включає кілька етапів, які допомагають створити ефективний і зручний інтерфейс для користувачів. Ось загальний огляд етапів проектування UI:

1. Дослідження та розуміння користувачів: Початковим кроком у проектуванні UI є вивчення цільової аудиторії та розуміння її потреб, цілей,

завдань та контексту використання. Це може включати проведення досліджень, збирання зворотного зв'язку від користувачів, створення персонажів користувачів та сценаріїв використання.

2. Визначення функціональності та структури: На цьому етапі визначаються основні функції та структура інтерфейсу. Створюється інформаційна архітектура, яка визначає організацію інформації та навігацію у додатку чи веб-сайті. Також визначаються основні елементи інтерфейсу, такі як меню, кнопки та поля введення.

3. Створення концепції та макетування: Проектувальники UI створюють концептуальні макети, які відображають загальний вигляд і відчуття інтерфейсу. Це може включати створення нарисів, проведення візуальних досліджень та розробку прототипів інтерфейсу. На цьому етапі зазвичай розглядається композиція елементів інтерфейсу, колірна схема, типографіка та іконки.

4. Деталізація та розробка: Коли затверджена концепція, проектувальники UI приступають до створення детальних макетів, які включають точні розміри, кольори, шрифти та інші характеристики елементів інтерфейсу. Розробка UI може виконуватися за допомогою графічних редакторів, таких як Photoshop або Sketch.

5. Тестування та ітерації: Створений інтерфейс тестується за участю реальних користувачів. Це може включати проведення тестів користувача, збір зворотного зв'язку та аналіз результатів. На основі отриманих даних проектувальники UI проводять ітерації, вносять поліпшення і оптимізують інтерфейс для досягнення кращого досвіду користувача.

6. Розробка та документування стилів інтерфейсу: Після затвердження та остаточної версії інтерфейсу створюються стилі та правила його застосування. Це включає визначення типографіки, колірних схем, стилів кнопок, форм та інших елементів інтерфейсу. Документування стилів та створення посібника з використання інтерфейсу допомагає забезпечити послідовність зовнішнього вигляду у різних частинах системи.

7. Впровадження та супровід: Після завершення процесу проектування та розробки UI інтерфейс впроваджується в остаточну систему чи продукт. Тут важливо забезпечити правильну інтеграцію інтерфейсу із основним функціоналом. Після впровадження інтерфейсу рекомендується його подальше супроводження та оновлення з урахуванням зворотного зв'язку користувачів та змін вимог.

8. Процес проектування інтерфейсу користувача є ітеративним і вимагає співпраці між дизайнерами, розробниками та користувачами. Мета полягає у створенні інтуїтивного, ефективного та приємного для використання інтерфейсу, який відповідає потребам користувачів та сприяє досягненню їх цілей.

## **1.2 Труднощі проектування ІК**

Сучасні вимоги до розробки програмного забезпечення включають рішення багатьох проблем. Важливо знизити витрати і скоротити терміни розробки, створити передбачувані плани і забезпечити якісні розв'язки. Крім того, програмне забезпечення повинно бути легким у використанні та навчанні, а також потребувати постійного оволодіння новими технологіями. Конкуренентоспроможність програмного продукту також є важливим аспектом.

Задоволення користувача програмним продуктом великою мірою залежить від якості інтерфейсу користувача, швидкості реакції програми, надійності, легкості встановлення, наявності інформаційної підтримки, можливості супроводження та інших факторів. Головна увага користувачів зосереджується на швидкості, простоті та надійності виконання роботи за допомогою автоматизованих засобів, які полегшують виконання завдань. [2].

Проектування інтерфейсу користувача може зустріти ряд труднощів, ось деякі з них:

1. Однією з головних труднощів є повне розуміння користувачів, їх



потреб, цілей та способів взаємодії з інтерфейсом. Іноді буває складно отримати достатню інформацію про користувачів і це може призвести до неоптимальних рішень у проектуванні.

2. Складність простоти : створення простого та інтуїтивно зрозумілого інтерфейсу може бути складним завданням. Виникають питання про те, як подати інформацію та функціональність таким чином, щоб користувачі могли легко зрозуміти та використовувати інтерфейс без зайвої напруги або плутанини. Баланс між простотою та функціональністю часто вимагає ретельного аналізу та тестування.

3. Різноманітність платформ та пристроїв: Сучасні інтерфейси повинні бути адаптовані до різних платформ та пристроїв, таких як комп'ютери, смартфони, планшети та телевізори. Кожна платформа має свої обмеження та особливості, і розробникам UI необхідно враховувати ці відмінності та створювати адаптивні та чуйні інтерфейси.

4. Підтримка різних цілей користувача та рівнів досвіду: В одній системі може бути безліч користувачів з різними цілями, рівнями досвіду та навичок. Проектувальникам UI необхідно враховувати різні потреби користувачів та створювати інтерфейс, який буде доступним та задовольняє потреби як новачків, так і досвідчених користувачів.

5. Взаємодія з даними та складність операцій: Якщо інтерфейс взаємодіє з великим обсягом даних або включає складні операції, то проектувальникам UI потрібно приділити увагу зручності роботи з даними та спрощенню складних процесів. Керування даними та подання інформації повинні бути інтуїтивними, щоб користувачі могли легко знаходити та обробляти потрібну інформацію.

6. Постійна еволюція та зміна вимог: Вимоги та переваги користувачів постійно змінюються. Це означає, що проектувальникам UI потрібно бути гнучкими та готовими до змін. Регулярні оновлення, тестування та збір зворотного зв'язку від користувачів допомагають адаптуватися до змін, що змінюються, та покращувати інтерфейс з часом.

Проектування інтерфейсу – це складний процес, що вимагає уваги до

деталей, розуміння користувачів і обліку різних факторів. Вирішення цих труднощів вимагає командної роботи, досліджень, тестування та постійного вдосконалення.

Існує кілька ключових умов, які дозволяють стверджувати, що проект ведеться в орієнтованому на користувача стилі.

У проекті проводяться дослідження та аналіз потреб та цілей користувачів. Вивчення цільової аудиторії, її потреб, контексту використання та поведінки допомагає зрозуміти, яким чином проект може найкращим чином задовольнити їх очікування.

Користувальні сценарії визначають, як користувачі будуть взаємодіяти з продуктом чи системою. Вони допомагають уявити та відтворити реальні ситуації використання, а також виявити потреби та вимоги користувачів.

Орієнтація на користувача передбачає створення прототипів та проведення їх тестування з реальними користувачами. Цей процес дозволяє перевірити ефективність та зручність інтерфейсу, виявити слабкі місця та внести необхідні зміни для покращення користувацького досвіду.

Орієнтація на користувача передбачає постійний зворотний зв'язок із користувачами на різних етапах проекту. Врахування їхньої думки, отримання відгуків та побажань допомагає вдосконалювати інтерфейс та відповідати їх потребам.

Дизайн інтерфейсу, спрямований на користувача, включає створення легкого у використанні та інтуїтивно зрозумілого інтерфейсу. Важливо представляти інформацію та функціональність таким чином, щоб вони були логічними і зрозумілими для користувачів. Також необхідно забезпечити зручну та ефективну навігацію.

При проектуванні інтерфейсу важливо враховувати контекст використання продукту, включаючи різні пристрої та платформи, умови використання та технічні обмеження. Це дозволяє оптимізувати інтерфейс і забезпечити зручну взаємодію з користувачами.[3]

Ключовим аспектом є забезпечення швидкості, легкості та надійності виконання завдань за допомогою автоматизованих засобів. Користувачі

сподіваються на зручну та задоволену роботу з програмним продуктом, яка полегшує їх завдання.

У процесі проектування інтерфейсу використовуються різні підходи(табл. 1.1), і найбільш оптимальним є еволюційний ітеративний підхід, який орієнтований на досягнення користувацьких цілей та практичності продукту. Розробка програмного забезпечення з використанням нових технологій та в умовах конкурентних факторів може бути складною задачею, особливо для складних проектів з обмеженими ресурсами. Загалом, інтерфейс, спрямований на користувача, має забезпечувати зручну, ефективну та задоволену взаємодію з програмним продуктом, враховуючи потреби та вимоги користувачів.

Таблиця 1.1 – Підходи до проектування ПЗ

Підхід до проектування	Опис підходу
Проектування «ззовні-всередину» (outside-in)	Спрямований на інтерфейс та доступні користувачу властивості програмного продукту.
Проектування «зсередини-назовні» (inside-out)	Починається з розгляду внутрішніх властивостей системи.
Однократне проектування (без ітерацій)	Проектування без встановлення планованого обсягу робіт з конструювання продукту та ІК.
Багатократне (ітераційне) проектування	Концентрується на ітераційній побудові ІК та його основних факторах практичності.
Проектування з використанням теорії «великого вибуху»	Спроба розробити «все або нічого», тобто ПЗ проектується і реалізується паралельно.
Еволюційне проектування	Зосереджене на побудові продукту з покроковим нарощуванням і уточненням можливостей.

Сьогодні для розробки інтерфейсів користувача та додатків необхідно мати різноманітні спеціальні навички в різних областях. Вони включають розробку програмного забезпечення, тестування та забезпечення якості функціонування ПЗ, графіку та візуалізацію, навчання моделям та системам, розробку довідкових систем, підтримку експлуатації, урахування людських факторів, бізнес-планування, управління потоками робіт, бізнес-реінжиніринг, управління проектами та змінами.

Ці навички є важливими для створення стильних інтерфейсів та додатків, які відповідають потребам користувачів. У сучасному світі важливо мати розуміння і вміння в різних областях, щоб забезпечити успішну розробку та впровадження програмних продуктів. Враховуючи широкий спектр спеціалізованих навичок, фахівці можуть працювати над створенням високоякісних інтерфейсів та додатків, які задовольняють потреби користувачів і досягають поставлених цілей.[4]

Успішність побудови ІК та програмного продукту полягає в професіоналізмі розробників, які повинні володіти наступними здібностями:

- працювати з різними людьми як в організаціях, які займаються розробкою, так і в комерційних організаціях;
- розуміти існуючі запити користувачів, а також їх майбутні потреби;
- вміти уточнювати й розширяти до необхідного ступеня деталізації неясно сформульовані вимоги;
- використовувати засоби й методи, які не обов'язково призначені для розробки ПЗ;
- швидко й ефективно оцінювати й відновлювати проект та реалізацію.

### **1.3 Етапи проектування ІК, орієнтованого на користувача**

1. Планування: Створення детального плану розробки продукту, який фокусується на побудові інформаційної системи та забезпеченні її

практичності. План включає календарні терміни для кожного етапу процесу, визначає основні ризики, об'єднує різні методи та встановлює цілі та критерії для інформаційної системи.

2. Встановлення вимог: Визначення вимог до системи, включаючи опис користувачів, постановку їхніх задач, оцінку поточного рівня практичності, аналіз можливостей інформаційної системи та дослідження тенденцій.

3. Концептуальне проектування: Розробка концептуального проекту, який містить високорівневі описи, абстракції та загальну інформацію про програмний продукт, його структуру та інформаційну систему. Це дає розробникам та кінцевим користувачам загальне уявлення про продукт.

4. Детальне проектування: Розробка детального проекту інформаційної системи, який включає характеристики програми, що сприймаються користувачем, такі як вхідні сигнали, взаємодія користувача та відповідь системи на цю взаємодію.

5. Прототипування: Створення прототипів та моделей системи, що є ефективними засобами для ранньої оцінки проекту. Прототип є матеріалізацією проекту з використанням передбачуваної платформи реалізації, включаючи обладнання, операційну систему, мови програмування та засоби реалізації.

6. Специфікація: Це документ, який конкретизує проект програмного продукту і описує дії користувачів, а також вигляд і поведінку програмного забезпечення в різних ситуаціях.

7. Конструювання: Цей етап включає написання коду програми та проведення автономних тестів для перевірки його працездатності.

8. Оцінка: Включає різні методи оцінки програмного продукту, які залучають потенційних користувачів. Це може включати збір зворотного зв'язку, проведення опитувань або тестування залученими користувачами.

9. Ітеративний підхід: Використовується для досягнення поставлених цілей розробки інформаційної системи. Робота може вимагати багато повторних ітерацій, де кожна ітерація додає нові функціональності або виправляє помилки на основі отриманого зворотного зв'язку.

10. Розгортка: Після того, як програмний продукт задовольняє вимогам та потребам користувачів, його розгортають для використання. [6]

З цього моменту починається оцінка продукту залученими користувачами, виконання завдань, які не були передбачені під час розробки, та здійснення подальших вдосконалень.

#### **1.4 Аналітичний огляд аналогів**

Завдяки зростанню кількості людей, які використовують смартфони та планшети для доступу до Інтернету, розробка мобільних додатків стає дуже привабливою та перспективною галуззю. Наприклад, у США близько 67% власників смартфонів щодня використовують їх для підключення до мережі. Існуючі дослідження також показують зростання кількості завантажень мобільних додатків, і очікується, що доходи від них досягнуть 65 мільярдів доларів. Одна з причин цього успіху полягає у постійному збільшенні продажів смартфонів та планшетів, що забезпечує велику базу потенційних користувачів для мобільних додатків.

«Sprinkle»[7] – це платформа замовлення тортів, доступна онлайн, яка пропонує широкий вибір смачних тортів для різних подій. Це зручний додаток, що дозволяє клієнтам налаштовувати свої замовлення та відстежувати доставку (див. рис. 1). Інтерфейс додатку приємний для користувача і використовує концепцію Material Design.

Люди, які використовують смартфони і планшети, постійно перебувають в русі, що означає, що вони не завжди мають можливість зайти на мобільний веб-сайт. Крім того, мобільні веб-сайти зазвичай спроектовані для зручного читання та навігації, а не для керування різними процесами.

Мобільні додатки дозволяють користувачам миттєво отримувати простий та функціональний доступ до інформації, продуктів, послуг та процесів, які їм потрібні. Вони оптимізовані для зручної взаємодії в режимі реального часу.

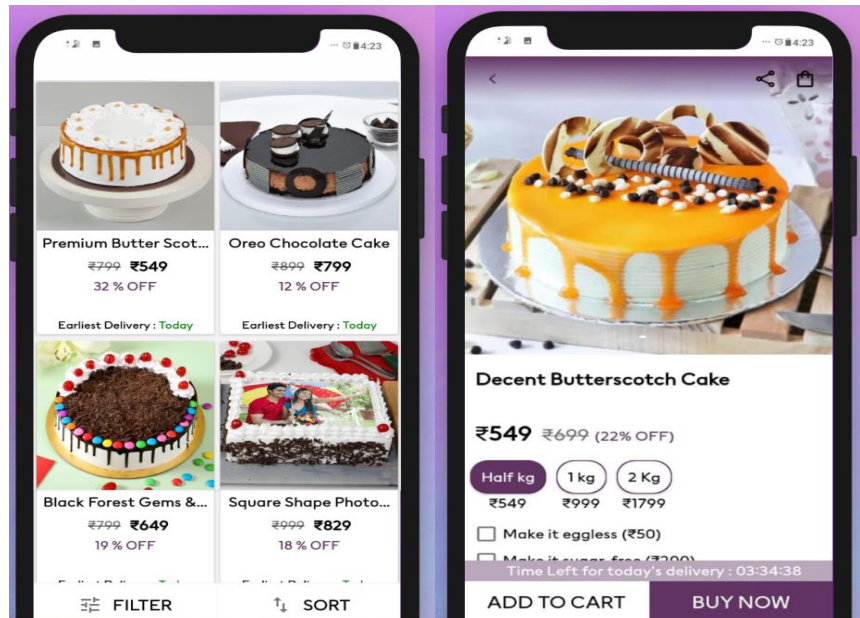


Рисунок 1– Sprinkle – «Order Cake Online»

«Kneaders Bakery Ordering» – це зручний мобільний додаток, який надає можливість клієнтам замовляти свіжу випічку з найближчого до них магазину Kneaders Bakery (рис. 2). Цей додаток пропонує простий процес замовлення та різноманітні параметри налаштування, що дозволяє з легкістю задовольнити ваші бажання. Він є кросплатформеним, тому ви можете відкрити його не тільки на мобільному пристрої, але і на планшеті. Будучи мобільним додатком, він надає зручний доступ до послуг і продуктів Kneaders Bakery у режимі реального часу, забезпечуючи зручну та ефективну взаємодію з клієнтами.

Крім того, завдяки своїм кросплатформеним можливостям, додаток може бути використаний на різних пристроях, що забезпечує його універсальність та доступність для широкого кола користувачів.

«Swiggy» – це мобільний додаток для доставки їжі, який співпрацює з різними пекарнями та кондитерськими закладами по всій Індії з метою доставки свіжої випічки прямо до вашого дверей (див. рис. 3). Цей додаток має зручний і легкий у використанні інтерфейс, що дозволяє користувачам з легкістю та швидкістю замовляти їжу.

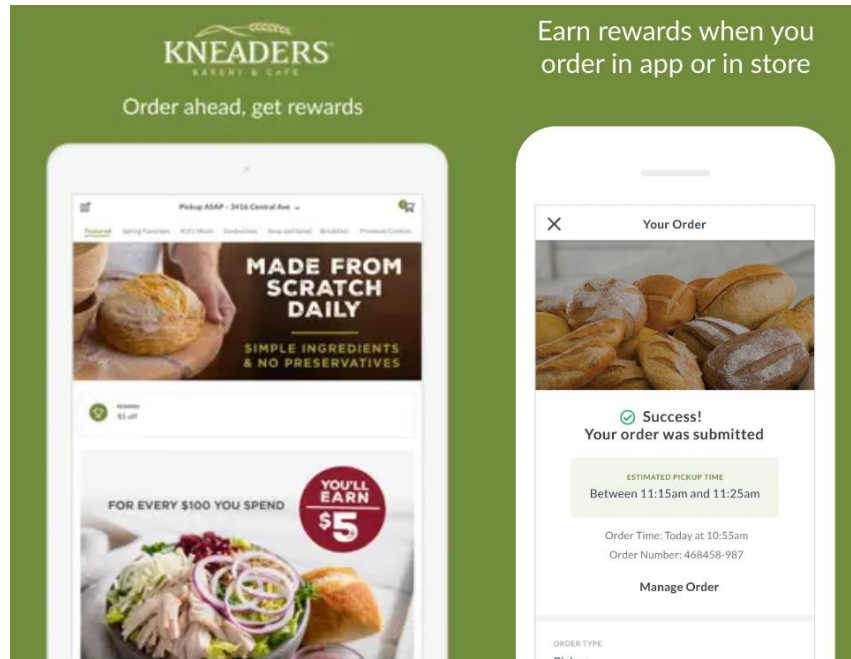


Рисунок 2 – «Kneaders Bakery Ordering»

Ви можете зробити свій вибір з різноманітного асортименту пекарських виробів і кондитерських виробів та замовити їх зручною для вас способом доставки. Будучи мобільним додатком, «Swiggy» надає зручний доступ до широкого вибору страв і десертів з різних закладів, забезпечуючи швидку та зручну доставку їжі прямо до ваших дверей.

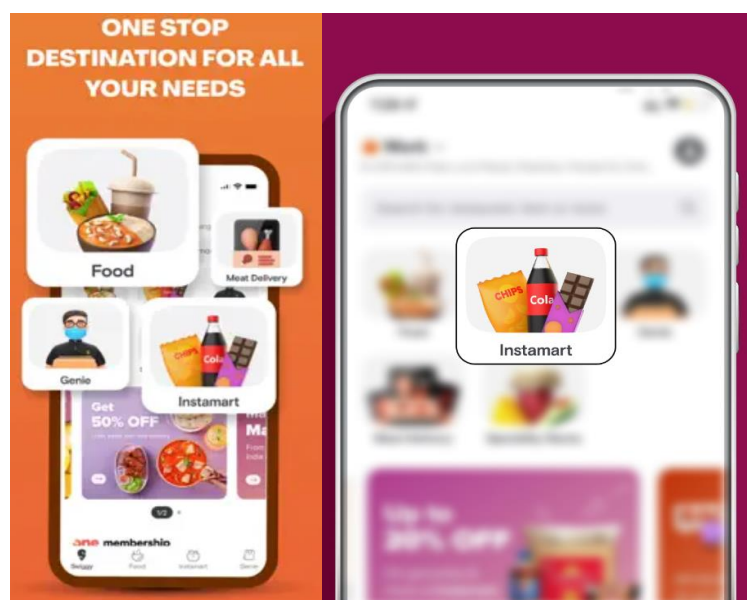


Рисунок 3 – «Swiggy»



«CakeFizz» – це найкращий додаток для пекарні для замовлення апетитних тортів. Завдяки широкому асортименту смаків і дизайнів, зручному інтерфейсу та швидкій доставці CakeFizz перетворює кожне святкування насолоду (рис.4).

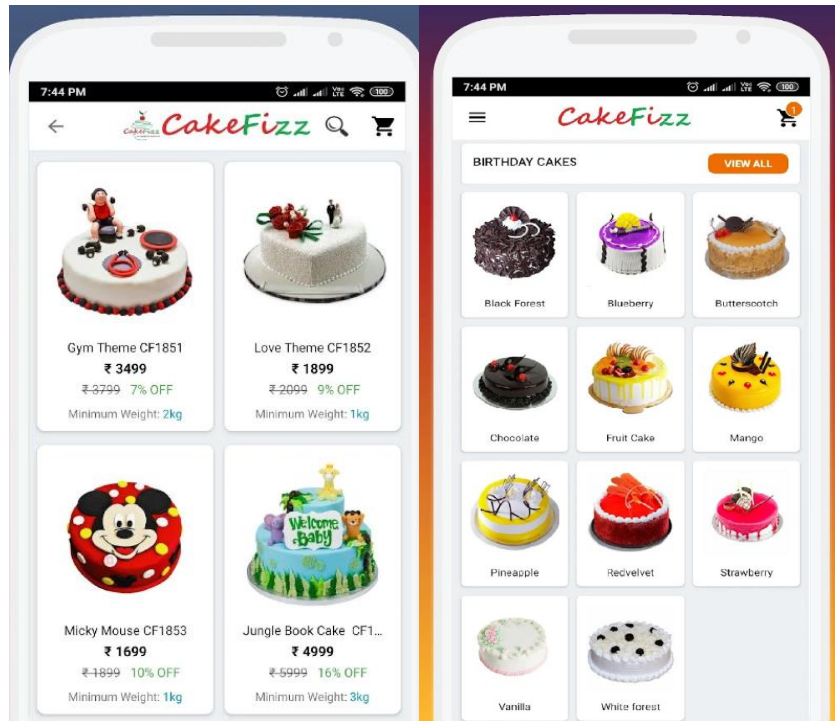


Рисунок 4 – «CakeFizz»

«CakeFizz» – це винятковий мобільний додаток для замовлення смачних тортів у пекарні. Завдяки широкому вибору смаків і оригінальних дизайнів, зручному інтерфейсу та швидкій доставці, CakeFizz перетворює будь-яке святкування на незабутню гастрономічну подорож (рис. 4). Цей додаток дарує клієнтам можливість насолодитись неперевершеними смаками тортів, вибрати дизайн, який відповідає їхнім вимогам, та отримати вишуканий десерт просто й швидко. Вибір і замовлення торта за допомогою CakeFizz – це шлях до задоволення смакових ковтків та створення незабутнього настрою на будь-якому святкуванні.

Проте, існує один важливий недолік – додаток доступний лише на платформі Android. Це обмежує аудиторію та відсікає велику кількість

користувачів, які використовують інші операційні системи.

Крім того, при відкритті додатку користувач не може повернути екран або збільшити картинку жестами пальців. Ці недоліки можуть суттєво погіршити користувацький досвід, оскільки деякі деталі можуть бути непомітними для користувача. Також, відсутня функція фільтрації пошуку в меню додатку, що ускладнює пошук потрібного товару для потенційних клієнтів.

Розгляд цих обмежень є важливим для поліпшення загального користувацького досвіду та розширення кола користувачів додатку.[8]

## 1.5 Інструментальні засоби розробки

Графічні редактори поділяються на два типи: векторні та растрові. Векторні редактори використовуються для створення масштабованих малюнків, розмітки сторінок, логотипів, технічних ілюстрацій, діаграм та блок-схем. Деякі популярні векторні редактори включають Adobe Illustrator, CorelDRAW та Figma.

Растрові редактори, з свого боку, призначені для обробки та ретушування фотографій, створення фотореалістичних ілюстрацій, колажів та ручних малюнків за допомогою графічних планшетів. Деякі популярні растрові редактори включають Adobe Photoshop та GIMP.

Розберемо детальніше засоби розробки:

1. Figma є векторним онлайн-сервісом, який призначений для розробки інтерфейсів та прототипування. Він дозволяє користувачам спільно працювати над проектами і має можливість зберігати файли онлайн. Figma (рис.5) доступний у двох форматах: як веб-додаток у браузері та як клієнтський додаток для роботи на комп'ютері. Цей редактор особливо підходить для створення прототипів, дизайн-систем і складних проектів, таких як мобільні додатки та портали.

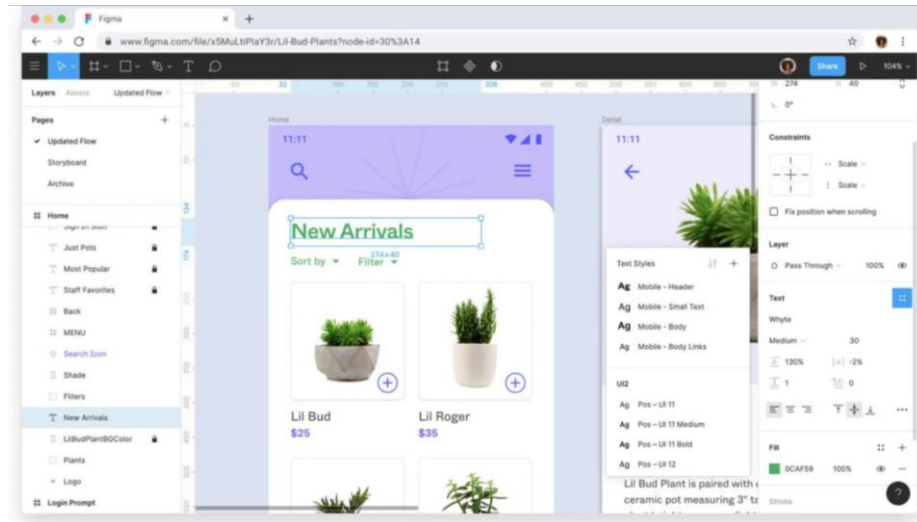


Рисунок 5 – «Figma»

2. Sketch є програмою для векторного дизайну, спеціально розробленою для платформи Mac. Вона призначена для дизайну екранів і зазвичай використовується дизайнерами, які створюють веб-сайти, значки та інтерфейси користувачів для настільних і мобільних пристроїв. Sketch – потужний і легкий у використанні додаток, який відповідає потребам як початківців, так і професіоналів у галузі дизайну. Він дозволяє користувачам фокусуватися на своїй творчості, пропонуючи інструменти для створення найкращих дизайнів. Додаток доступний за ціною 99 доларів США.

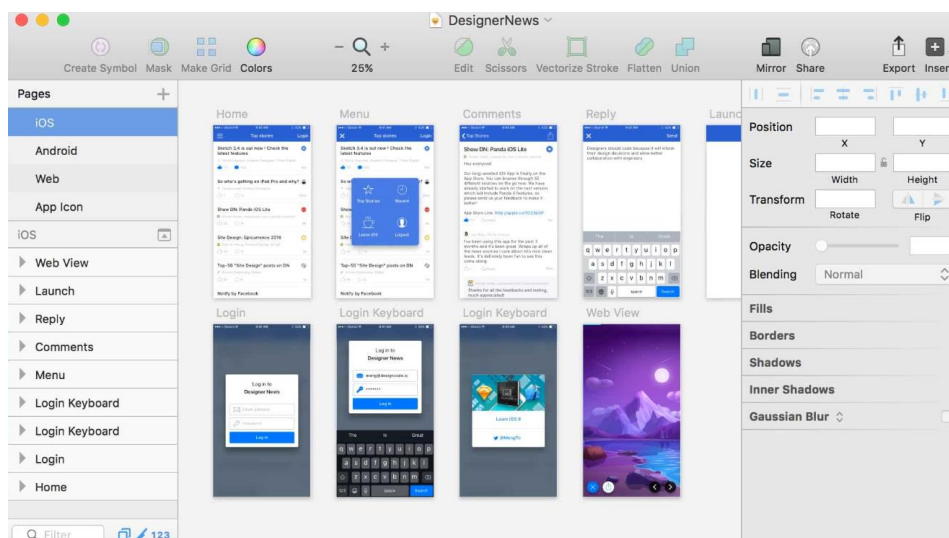


Рисунок 6 – «Sketch»

3. Inkscape є редактором векторної графіки з відкритим вихідним кодом. Він надає можливості, подібні до Illustrator, CorelDraw або Xara X, і підтримує стандартний формат файлів W3C Scalable Vector Graphics (SVG). Inkscape дозволяє створювати і редагувати векторні зображення, включаючи логотипи, розмітку сторінок, ілюстрації та інші графічні елементи. Завдяки відкритому вихідному коду, Inkscape є безкоштовним і доступним для широкого кола користувачів.(рис.7)

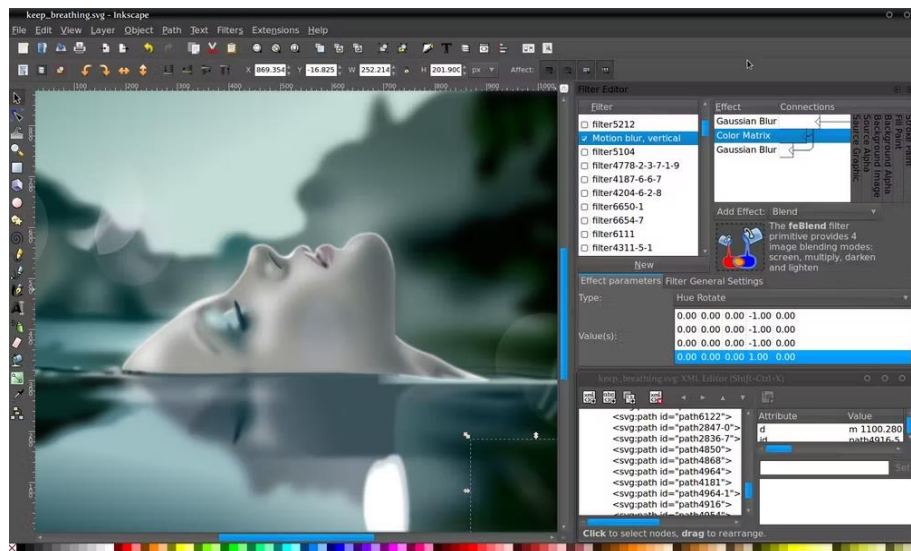


Рисунок 7 – «Inkscape»

Inkscape володіє багатьма розширеними можливостями формату SVG, такими як використання маркерів, клонування об'єктів, альфа-змішування тощо. Редактор приділяє велику увагу оптимізації інтерфейсу, щоб забезпечити зручну роботу з програмою.

У Inkscape дуже зручно редагувати вузли, виконувати складні операції з кривими і здійснювати трасування растрових зображень. Це лише деякі з функцій, які роблять Inkscape потужним інструментом для створення та редагування векторної графіки.

4. Appy Pie – це хмарний сервіс, який дозволяє створювати мобільні додатки без необхідності програмування. Цей інструмент дозволяє користувачам без технічних навичок створювати додатки для різних –

платформ і публікувати їх. Вам не потрібно нічого встановлювати або завантажувати – ви можете просто перетягувати сторінки, щоб створити свій власний мобільний додаток онлайн. Appy Pie надає простий і зручний спосіб для невишколених користувачів створювати свої власні додатки без необхідності залучення програмістів або розробників.

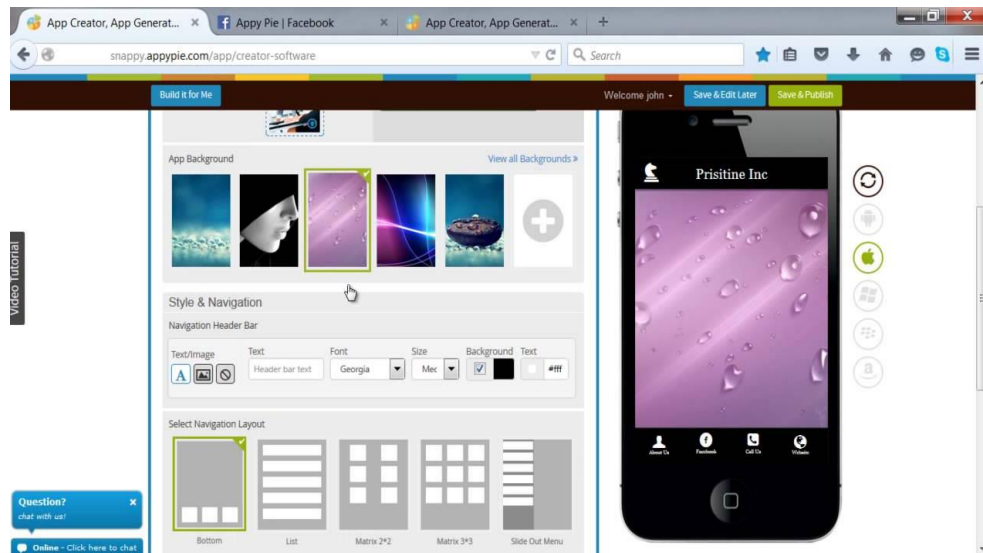


Рисунок 8 – «Appy Pie»

Після завершення використання Appy Pie ви отримуєте гібридну програму, розроблену на основі HTML5, яка працює на різних платформах, включаючи iOS, Android, Windows і прогресивні програми. Ви можете переглядати програму в режимі реального часу та отримувати push-повідомлення. Інструмент також надає можливість монетизувати додаток за допомогою реклами, відстежувати аналітику в реальному часі та використовувати GPS для відстеження місцезнаходження.

Ви можете інтегрувати соціальні мережі, блоги, веб-сайти, аудіо, радіо і багато іншого. Платформа надає різні теми, що підходять для різних видів бізнесу, і має інструмент планування зустрічей, особливо корисний для салонів, лікарів та спа-центрів. Ціна для створення такого додатку складає приблизно 50 доларів.

Розглянемо також засоби реалізації [9] прототипів:

Android Studio – це офіційне інтегроване середовище розробки (IDE) для розробки програм для Android. (рис.9) Заснована на потужному редакторі коду та інструментах розробника IntelliJ IDEA, Android Studio пропонує ще більше функцій, що підвищують вашу продуктивність при створенні програм для Android, таких як:

1. Гнучка система збирання на основі Gradle.
2. Швидкий та багатofункціональний емулятор.
3. Єдине середовище, в якому ви можете розробляти для всіх пристроїв Android.
4. Live Edit для оновлення складових об'єктів у емуляторах та фізичних пристроях у режимі реального часу.
5. Великі інструменти та фреймворки для тестування.
6. Інструменти Lint для виявлення проблем із продуктивністю, зручністю використання, сумісністю версій та іншими проблемами.

Вбудована підтримка Google Cloud Platform, що спрощує інтеграцію Google Cloud Messaging та App Engine.

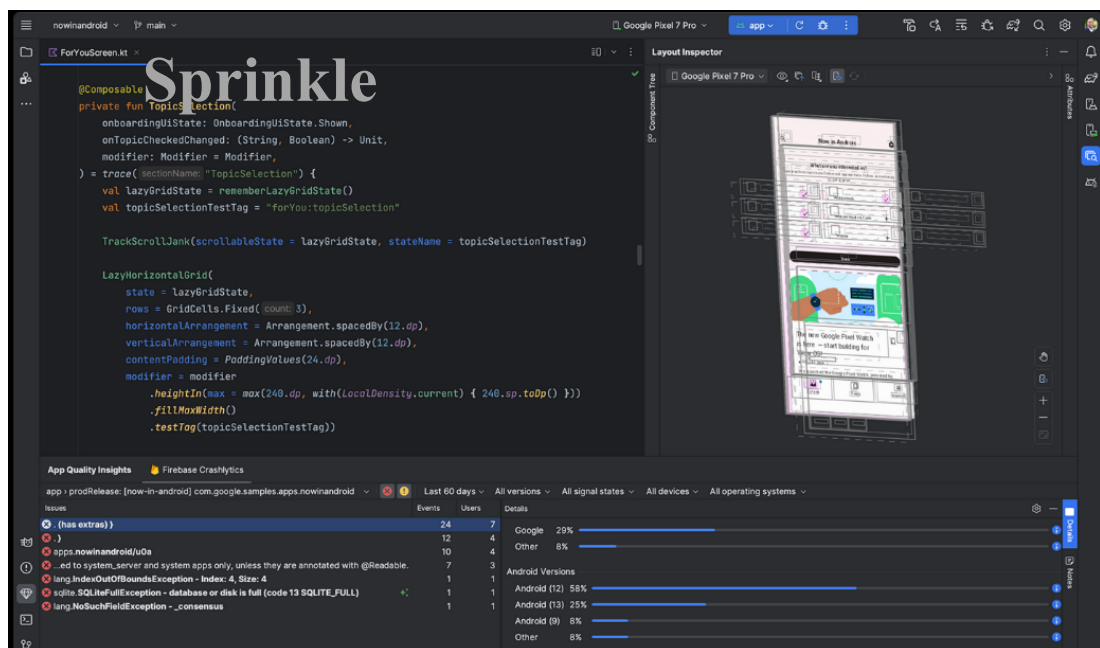


Рисунок 9 – середовище розробки «Android Studio»

Visual Studio Code – [10] це оптимізований редактор коду з підтримкою таких операцій розробки, як налагодження, виконання завдань та контроль версій. Він спрямований на те, щоб надати розробнику тільки ті інструменти, які необхідні розробнику для швидкого циклу складання коду та налагодження, і залишає складніші робочі процеси повнофункціональнішим IDE, таким як Visual Studio IDE . Безкоштовний для приватного чи комерційного використання.

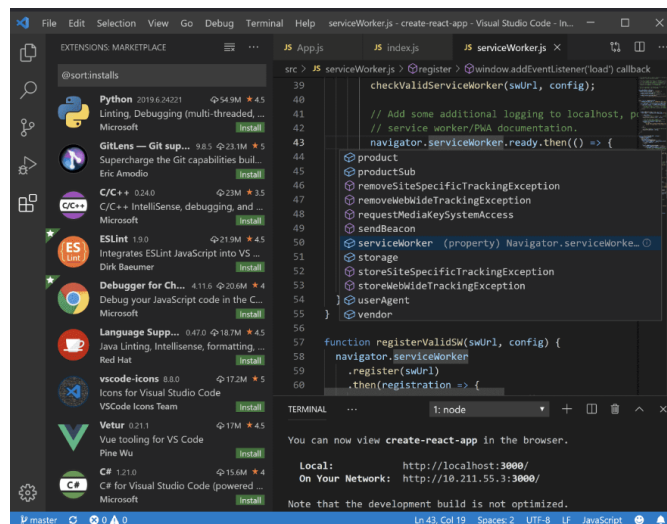


Рисунок 10 – «Visual Studio Code»

Xcode включає все, що потрібно розробникам для створення відмінних додатків для Mac, iPhone, iPad, Apple TV і Apple Watch. Xcode надає розробникам єдиний робочий процес для проектування інтерфейсу користувача, кодування, тестування та налагодження. Xcode IDE у поєднанні з мовою програмування Swift робить розробку додатків простою та захоплюючою.

Xcode включає Xcode IDE, компілятори Swift та C/C++/Objective-C, інструмент аналізу Instruments, симулятори, новітні SDK та сотні потужних функцій:

– Інноваційні інструменти допоможуть вам створювати чудові програми.

- Swift – це інтуїтивно зрозуміла мова програмування, безпечна, швидка та сучасна.
- SwiftUI [11]— це революційна платформа для створення інтерфейсів користувача з декларативним синтаксисом Swift.
- Ігрові майданчики – це цікавий спосіб експериментувати та взаємодіяти з кодом Swift.
- Налаштування перегляду показує тривимірний стек всіх шарів перегляду інтерфейсу користувача вашої програми під час виконання.
- Розділяйте редактори практично необмеженими способами, показуйте прев'ю або вибирайте помічника для перегляду пов'язаного контенту.
- Поточні завдання відображають помилки при введенні, а функція Fix-it покращує ваш код одним клацанням миші.
- Навігатор системи керування версіями та інтеграція служб допомагають керувати кодом у команді.

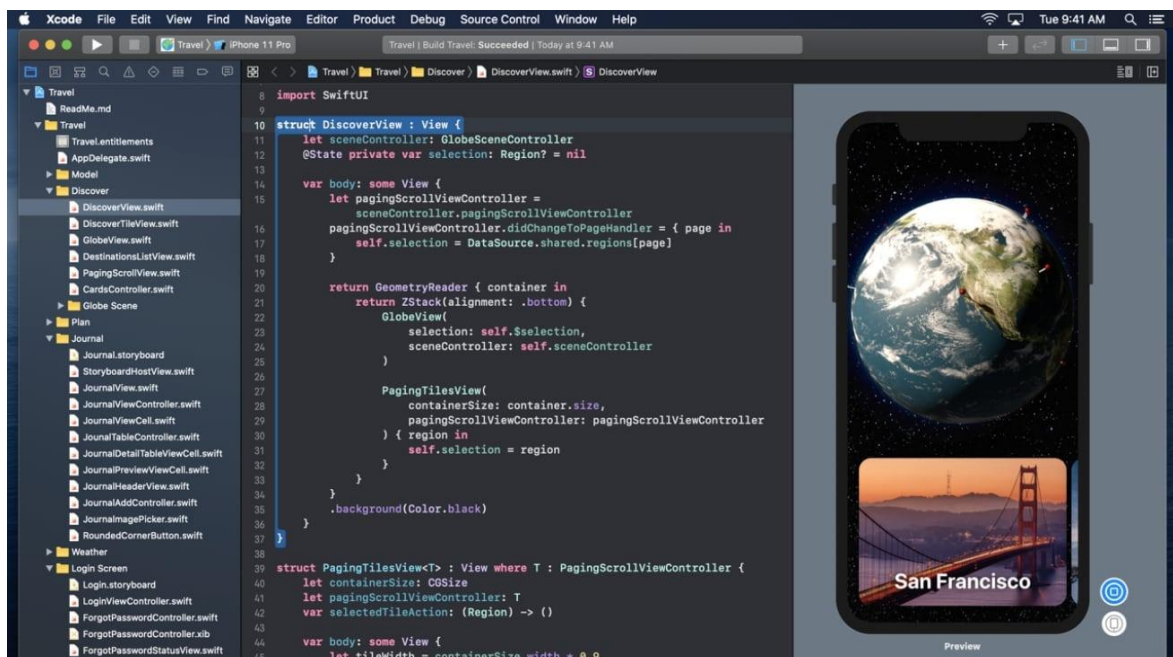


Рисунок 11 – «SwiftUI»



## **2 РОЗРОБКА ТЕХНІЧНОГО ПРОЕКТУ**

### **2.1 Вимоги до мобільних додатків**

У сучасному світі розробка мобільних додатків стала надзвичайно популярною в галузі інформаційних технологій. Це пов'язано з тим, що мобільні пристрої стали невід'ємними частинами нашого життя і виконують багато функцій, таких як обмін даними, швидкий доступ до актуальної інформації, мобільні платіжні сервіси, контроль бізнес-процесів, аналітика та інше.

Мобільний додаток – це програма, яка встановлюється на мобільний пристрій і має певний функціонал, що дозволяє користувачеві виконувати певні дії. Виділяють такі типи додатків: нативні, веб-додатки і гібридні додатки. Нативні додатки розробляються спеціально для певної платформи (наприклад, iOS або Android) і мають повний доступ до функціональності пристрою. Веб-додатки використовують веб-технології і працюють у веб-браузері на мобільному пристрої. Гібридні додатки поєднують в собі переваги нативних і веб-додатків, дозволяючи розробникам використовувати загальний код для різних платформ.

Ця розробка стала необхідною для задоволення потреб користувачів в мобільних сервісах та створення зручного інструменту для виконання різних завдань на мобільних пристроях. Розглянемо їх більш детально.

Нативний додаток – це додаток, який повністю побудований з використанням технологій, що відповідають конкретній операційній системі, такі як Android, iOS, Windows, Blackberry і т.д. Для розробки нативних додатків для Android зазвичай використовується мова програмування Java, а для iOS – Objective C або Swift.

Розробка дизайну та інтерфейсу мобільного додатку є критичним етапом в процесі створення програмного продукту. Від цієї роботи залежить сприйняття користувачами додатку, його популярність та зручність використання.

Дизайн мобільного додатку повинен бути максимально простим і зрозумілим. Оскільки мобільні пристрої мають обмежений розмір екрану, при проектуванні додатків необхідно враховувати особливості цих пристроїв. Важливо створити інтерфейс, який буде зручним для використання на малих екранах та забезпечуватиме зручний доступ до функцій додатку.

Основні принципи для дизайну інтерфейсу мобільних додатків:

1. Простота та ефективність: Мінімізувати кількість елементів інтерфейсу, зберігаючи в той же час функціональність. Створити лаконічний та дружній інтерфейс, який ефективно виконує потрібні завдання.

2. Адаптація до сенсорних екранів: Врахувати, що користувачі взаємодіють з екраном за допомогою чутливого дотику. Уникати дрібних елементів інтерфейсу, оскільки площа дотику пальця більша, ніж покажчика комп'ютерної миші або стилуса.

3. Читабельність тексту: Забезпечити достатній розмір шрифту, щоб користувач міг зручно прочитати текст з відстані не менше 30 см.

4. Виділення головних елементів: Розташовувати найважливіші та часто використовувані елементи в центральній частині екрану, надаючи їм достатньо великий розмір, щоб вони виділялись серед інших.

5. Оптимізація для мобільних платформ: При перенесенні додатків з ПК на мобільні пристрої, враховувати обмежене просторове та функціональне середовище. Зменшувати кількість елементів, групуючи їх за функціональністю. Застосовувати «віконний» підхід, де елементи інтерфейсу замінюють один одного на екрані, забезпечуючи навігацію між ними за допомогою графічних елементів або перетягуванням пальцем.

6. Культурні особливості: При проектуванні мобільних додатків для конкретного регіону враховувати його культурні особливості, такі як напрям читання (справа наліво) та вибір колірної палітри.

7. Зручність використання: Створити інтуїтивний дизайн, який використовує всі можливості мобільного пристрою. Враховувати особливості популярних мобільних платформ, таких як iPhone і Android.

8. Цікавість та корисність: Забезпечити цікавість користувача,

включаючи елементи розваги. Додаток повинен бути корисним та задовольняти потреби користувача.

9. Структура діалогу: Визначити логічну структуру діалогу між користувачем і мобільним пристроєм, забезпечуючи зрозумілу та послідовну взаємодію.

10. Оновлення та зворотний зв'язок: Забезпечити можливість оновлення додатка та отримання зворотного зв'язку від користувачів, щоб постійно покращувати його функціональність та відповідати потребам аудиторії.

Цей список враховує основні принципи та вимоги до дизайну інтерфейсу мобільних додатків, забезпечуючи оптимальну взаємодію між користувачем та програмним забезпеченням.

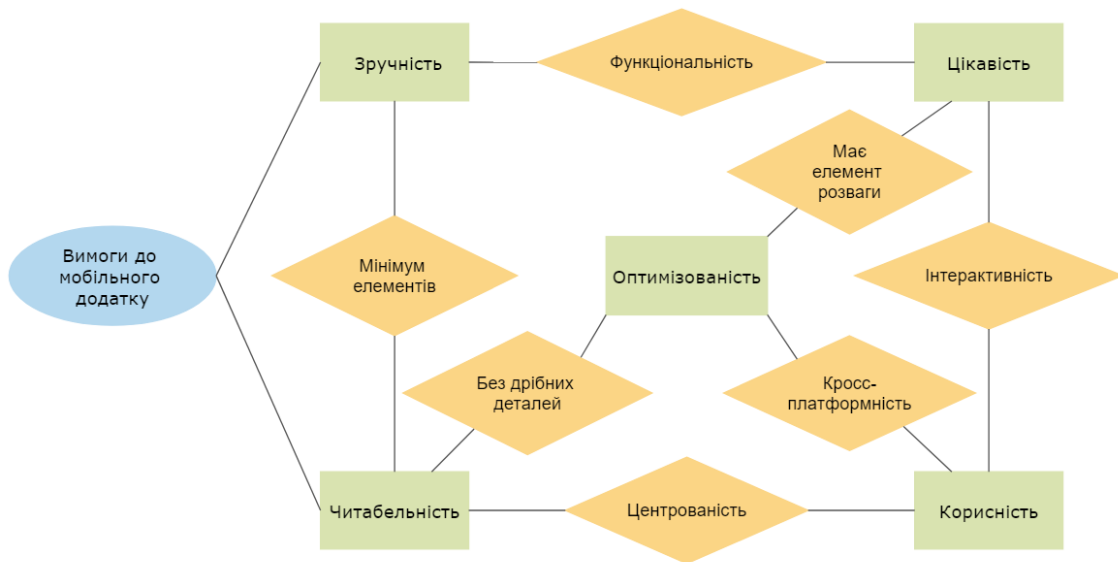


Рисунок 12 – Діаграма «Вимоги до мобільного додатку»

Мобільні додатки мають численні переваги. Вони забезпечують високий рівень інтерактивності та мають доступ до різноманітних функцій пристрою, таких як акселерометр, GPS-навігація і фотокамера. Мобільні додатки також надають кращий користувацький досвід, швидко завантажуються і відповідають стандартному інтерфейсу операційної

системи. Завдяки можливості зберігати персоналізований контент і проводити складні обчислення, мобільні додатки забезпечують високий рівень персоналізації. Крім того, вони можуть працювати в автономному режимі без Інтернет-підключення.

Проте у мобільних додатків є й деякі недоліки. Вони вимагають відповідності до конкретної операційної системи, що може створювати проблеми з сумісністю. Розробка, підтримка і обслуговування мобільних додатків також можуть бути витратними порівняно з мобільними сайтами. Також для створення мобільного додатка потрібно більше часу, а користувачам потрібно встановити додаток на свій пристрій перед його використанням.

Враховуючи всі ці вимоги та обмеження, можна приступати до проектування мобільного додатка з метою створення оптимального користувацького досвіду.

## **2.2 Проектування мокапів**

Для розробки моделі мобільного додатку був обраний веб-ресурс Figma. Кожна вкладка прототипу була створена індивідуально відповідно до вимог замовника.

Вікна привітання та реєстрації надають користувачам можливість взаємодіяти з додатком. На цьому етапі користувач має можливість пропустити реєстрацію, щоб спочатку ознайомитись з додатком, а потім, за бажанням, зареєструватись. Для тих, хто обрав «Продовжити», доступні різні способи реєстрації, такі як електронна пошта, або альтернативні варіанти.

Вікно реєстрації та вітання в мобільному додатку має кілька основних цілей. Воно дозволяє новим користувачам створити обліковий запис у додатку. Користувачі можуть ввести свої дані, такі як електронна пошта, номер телефону або скористатись входом через соціальні мережі. Реєстрація необхідна для ідентифікації користувачів та надання їм доступу до функцій та контенту програми.

Для зареєстрованих користувачів вікно привітання дозволяє виконати вхід до їх облікового запису. Користувачі можуть використовувати свої дані, такі як ім'я користувача та пароль, для отримання доступу до додатку. Це забезпечує персоналізований досвід використання програми та доступ до збережених даних та налаштувань.

Крім того, вікно привітання може використовуватись для представлення інформації про програму, її функції та можливості. Це може включати короткий опис програми, скріншоти або відеопрезентацію, що допомагають користувачам зрозуміти, як використовувати програму та які переваги вона пропонує.

Також, вікно привітання може пропонувати користувачеві налаштування повідомлень та дозволів для додатку. Наприклад, користувачеві може бути запропоновано дозволити додатку надсилати повідомлення, отримувати доступ до контактів або використовувати геолокацію. Це допомагає взаємодіяти з користувачем та надавати персоналізований контент.

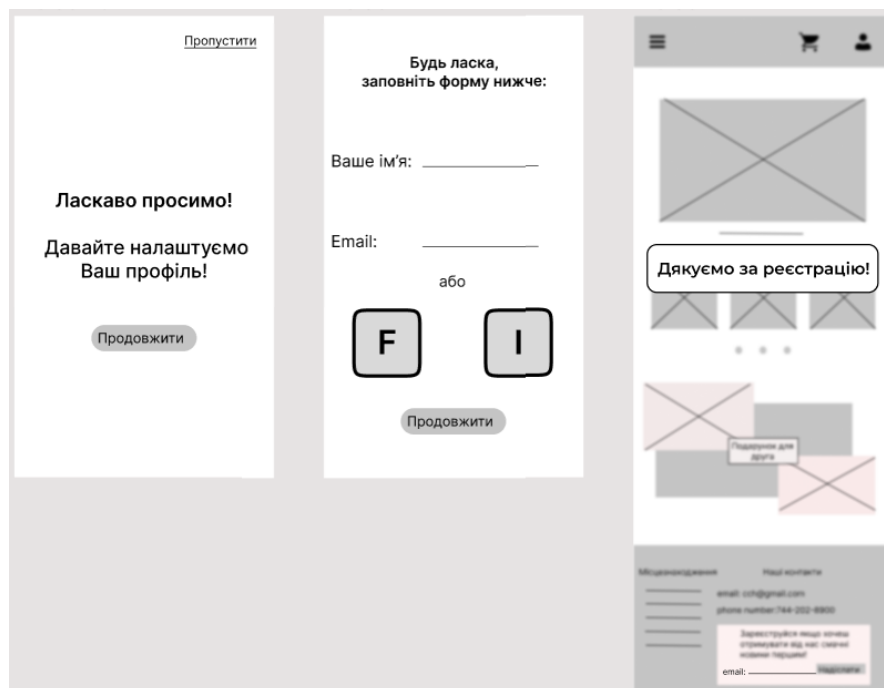


Рисунок 13 – «Вікна привітання»

Після успішної реєстрації або пропуску реєстраційного кроку, користувач буде перенаправлений на домашню сторінку мобільного додатку. На цій сторінці він зможе побачити актуальні знижки та найпопулярніші пропозиції. Також верхня панель міститиме всю контактну інформацію про кондитерську.

Домашня сторінка може надавати користувачу персоналізовані рекомендації, враховуючи його попередню взаємодію з додатком, його уподобання та аналіз даних. Ці рекомендації можуть включати контент, продукти, послуги або акції, які можуть бути цікавими для користувача. Такий підхід допомагає поліпшити користувацький досвід та задовольнити його індивідуальні потреби.

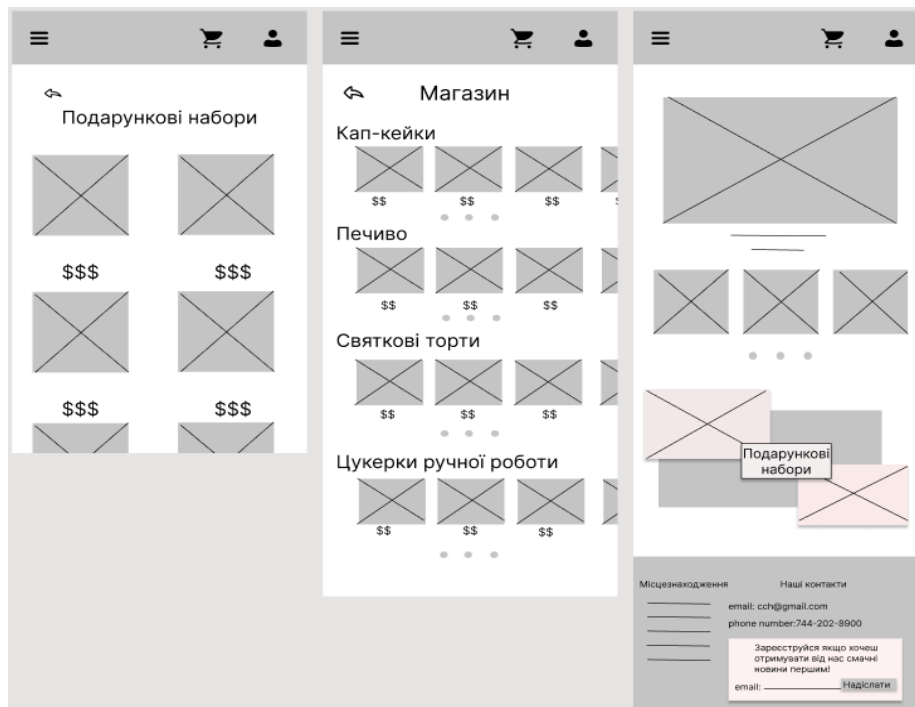


Рисунок 14 – «Вікна домашньої сторінки і магазину»

У вкладці «Магазин» можна побачити весь асортимент розділений на категорії. Перейшовши по категорії відкривається сторінка на якій зручно вибирати товар. Вікно магазину надає користувачеві можливість переглядати та купувати продукти або послуги, пропонувані додатком.

Також воно може відображати актуальні акції, знижки та спеціальні пропозиції. Це може стимулювати користувача до здійснення покупки та надати йому вигідні умови для придбання продукту чи послуги.



Рисунок 15 – «Панель управління»

Всі вкладки супроводжує верхня панель керування. На ній можна знайти три основних кнопки управління: «Бургер меню», «Кошик», та «Профіль». Дякуючи цим кнопкам користувач зможе швидко знайти потрібну вкладку. Також на верхній панелі керування можна розміщувати логотип завдяки якому компанія буде краще запам'ятовуватися.

Верхня панель керування може містити значки або кнопки для швидкого доступу до основних функцій програми.

Наприклад, це можуть бути значки для переходу до домашньої сторінки, пошуку, профілю користувача, повідомлень, налаштувань та інших важливих розділів. Верхня панель керування може відображати заголовок поточного екрана або розділ програми. Вона допомагає користувачеві орієнтуватися у додатку та розуміти, на якій сторінці він знаходиться.

Натискаючи на «бургер меню» відкривається панель управління із повним списком інформації яка потрібна користувачу. Бургер-меню використовується для розміщення основного набору функцій та розділів програми.

При натисканні на іконку бургер-меню відкривається меню, де користувач може вибирати потрібні розділи, переходити на інші екрани або виконувати навігаційні дії. Таким чином натиснувши на кнопку «Профіль», користувач може з легкістю перейти на її вкладку.

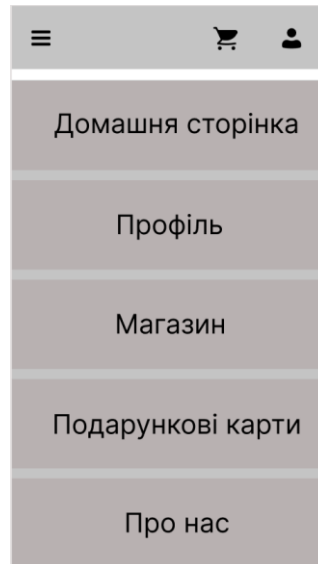


Рисунок 16 – «Вікно бургер-меню»

У вкладці профіль можна додати або змінити адресу та спосіб оплати, також редагувати данні і побачити що користувач замовляв. Для більшої комунікації в Профіль можна також завантажувати фото.

Профілі користувачів надають актуальну інформацію про окремих клієнтів стосовно всієї вашої клієнтської бази. Посилання на профіль користувача допомагає відповісти на запитання про те, хто ваші клієнти та як вони взаємодіють із вашим брендом.

Вікно профілю дозволяє користувачеві переглядати та редагувати інформацію, пов'язану з його профілем. Це може включати ім'я, фотографію, електронну адресу, контактні дані, біографію та іншу персональну інформацію. Також це вікно може надавати інформацію про активність користувача та статистику.

Вкладка «Про нас» дає змогу користувачам більше познайомитися із кондитерською в якій вони будуть замовляти продукт. Таким чином вибудовуються відносини між компанією та клієнтом, або будується підґрунття до співпраці. Розділ «Про нас» є другою за популярністю сторінкою сайту, одразу після головної сторінки. На ній потрібно розмістити історію компанії таким чином щоб показати її основні цілі і особливості.



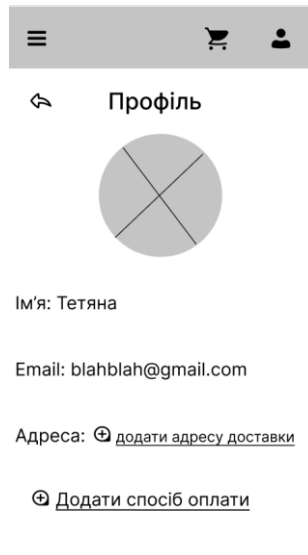


Рисунок 17– «Сторінка профілю»

Доречно описати місію та цінності, історію компанії, лідерство та команду, відносини з інвесторами, новини компанії, інформацію про кар'єру в організації, соціальну позицію та іншу інформацію, яка може бути цікавою для клієнтів.

У вікні «Про нас» надається інформація про програму, її призначення, цілі, функціональність та особливості. Це може включати опис можливостей, сфери застосування, ключові особливості та інші деталі, які допомагають користувачам краще зрозуміти і використовувати програму.



Рисунок 18 – «Вкладка про нас»

У вкладці кошик користувач може побачити всі позиції які він раніше додав до листу, проставити кількість товару, побачити сумму всього замовлення і замовити усе разом.

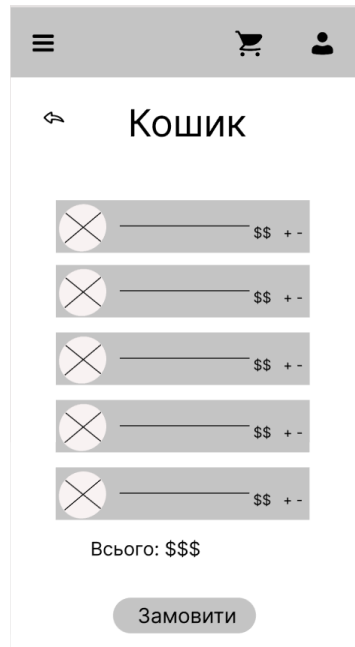


Рисунок 19 – «Кошик»

Повторне натискання на кнопку «До кошика» не повинно додавати товар до кошика. Повторне натискання на кнопку має здійснювати перехід на сторінку кошика. Якщо розрахувати вартість доставки на сторінці кошика неможливо, поряд із сумою замовлення має бути вказана приблизна вартість доставки або стояти позначка про те, що сума вказана без вартості доставки.

Натиснувши на будь яку пропозицію товару користувач переходить до вкладки описання товару де можна побачити ціну, опис, відрегулювати кількість, додати до кошику. Вікно описання товару дозволяє користувачеві отримати детальну інформацію про продукти або послуги, включаючи опис, характеристики, ціни, знижки, відгуки та зображення. Це допомагає користувачеві прийняти поінформоване рішення про покупку, оцінити відповідність продукту до його потреб і порівняти різні варіанти.

Для того щоб спонукати користувача до більшої кількості замовлень, також додано віджет «Схоже за вашим запитом» де користувач може побачити нові варіанти товару. Також є кнопка «Перейти до кошику» для зручної навігації по додатку.



Рисунок 20 – «Опис товару»

Перед завершенням замовлення користувач має змогу перевірити всі свої данні і після цього замовити товар. У вікні перевірки замовлення відображається зведення всіх вибраних товарів або послуг, їх кількість, ціна та інші деталі. Це дозволяє користувачеві переконатися, що обрані позиції вірні та відповідають його намірам.

Вікно перевірки замовлення може містити інформацію про спосіб доставки або отримання замовлення. Це може включати адресу доставки, вибраний пункт самовивезення або інші відповідні деталі. Користувач може перевірити, чи вказана інформація вірна і необхідні параметри доставки вказані.

Вікно перевірки замовлення надає інформацію про обраний спосіб оплати та пов'язану з ним інформацію. Це може бути номер кредитної

картки, дані про рахунок PayPal або інші деталі. Користувач може переконатися, що інформація про платіж правильна та необхідні дані вказані.

У вікні перевірки замовлення відображається загальна сума замовлення, включаючи вартість товарів або послуг, податки, доставку та будь-які інші застосовні збори. Це дозволяє користувачеві оцінити остаточну вартість і переконатися, що всі правильні розрахунки.

У вікні перевірки замовлення зазвичай є кнопка або елемент для фінального підтвердження замовлення. Після перевірки всіх деталей користувач може натиснути цю кнопку, щоб розмістити замовлення та перейти до процесу оплати або підтвердження. По завершенню користувач отримує сповіщення про те що замовлення обробляється і може спокійно продовжувати пошук потрібних йому товарів.

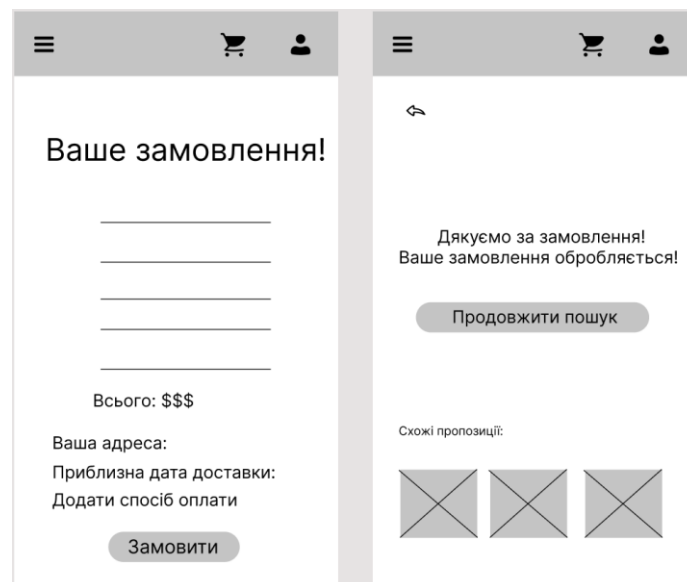


Рисунок 21 – «Вікна підтвердження замовлення»

### 2.3 Відмінності дизайну під iOS від дизайну під Android

Дизайн під iOS та Android має свої особливості та відмінності відповідно до гайдлайнів та рекомендацій кожної платформи. Ось деякі з

основних відмінностей у дизайні між iOS та Android:

1. Візуальний стиль та іконки: iOS та Android мають різні візуальні стилі та іконки, що відображають загальну естетику кожної платформи. iOS зазвичай має більш плоский і гладкий дизайн з використанням округлих кутів і тіней, в той час як Android часто має більш реалістичний та глибокий дизайн із більш гострими кутами та градієнтами.

2. Розміщення елементів інтерфейсу користувача: Розміщення елементів інтерфейсу користувача на iOS і Android може відрізнитися. Наприклад, в iOS загальноприйнятою практикою є розміщення основного навігаційного меню в нижній частині екрану (Tab Bar), тоді як в Android це може бути верхнє меню (Toolbar) або бічне висувне меню (Navigation Drawer).

3. Іконки навігації: iOS використовує іконку «Назад» у верхньому лівому куті для переходу на попередній екран, а Android використовує іконку «Назад» у верхньому лівому куті або фізичну кнопку «Назад» на пристрої. Крім того, на Android може використовуватися іконка «Додому» для переходу на головний екран програми.

4. Компоненти інтерфейсу користувача: iOS і Android мають набір своїх унікальних компонентів інтерфейсу користувача. Наприклад, iOS має такі компоненти, як Navigation Bar, Tab Bar, Segmented Control, а Android пропонує Toolbar, Bottom Navigation, Drawer Navigation та інші.

5. Колірна схема та типографіка: iOS та Android мають різні рекомендації щодо колірної схеми та типографіки. iOS зазвичай надає перевагу більш яскравим і насиченим кольорам, а Android може мати більш налаштовану і гнучку кольорову палітру. Типографіка також може відрізнитися відповідно до переваг кожної платформи.

6. Анімації та переходи: iOS та Android можуть мати різні стандартні анімації та переходи між екранами. iOS часто використовує плавні переходи із застосуванням анімації перегортання або виїзду екрану, у той час як Android може мати більш динамічні анімації, такі як переходи з використанням обертання чи переміщення елементів.

Враховуючи ці відмінності в дизайні, розробники мобільних програм намагаються адаптувати свої додатки до стандартів кожної платформи, щоб забезпечити користувачеві більш якісний і консистентний досвід використання на iOS або Android.

### 3 РОЗРОБКА UI

Для реалізації інтерфейсу під різні операційні системи дизайн вкладок буде виконуватись попарно.

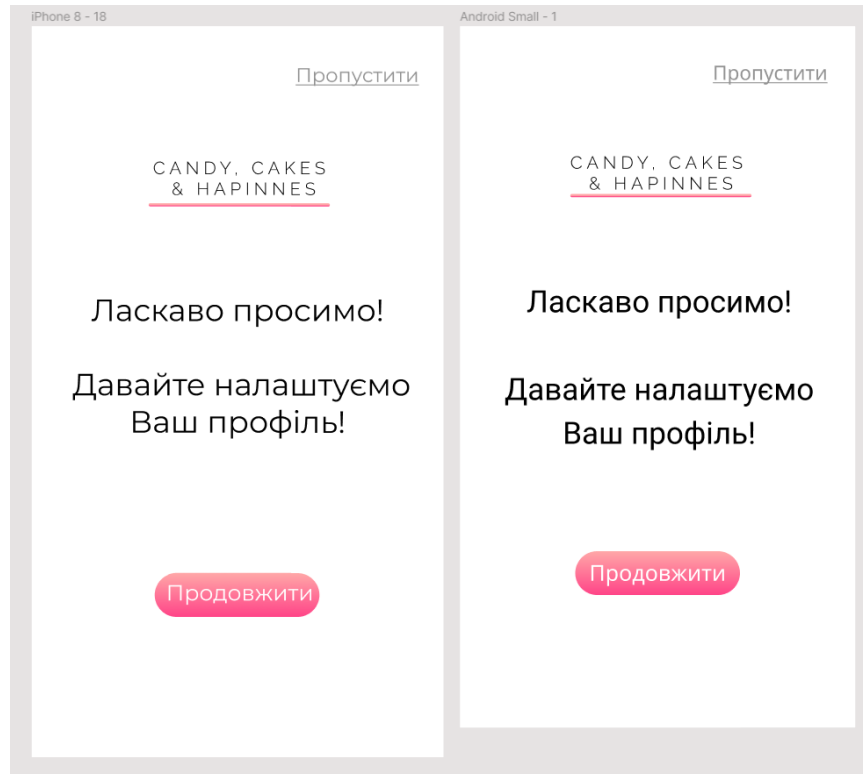


Рисунок 22 – «Вікно привітання»

Для кнопки «Продовжити» використовувалися такі параметри позиціонування:

```
position: absolute;
width: 150px;
height: 40px;
left: calc(50% - 150px/2 - 0.5px);
top: calc(50% - 40px/2 + 182.5px);
background: linear-gradient(180deg, rgba(255, 153, 153, 0.85) 0%, rgba(255, 28, 109, 0.82) 100%);
border-radius: 40px;
```

Та для Android використовуються xlm-теги:

```

<View
  android:id="@+id/rectangle_4"
  android:layout_width="150dp"
  android:layout_height="40dp"
  android:layout_alignParentLeft="true"
  android:layout_marginLeft="105dp"
  android:layout_alignParentTop="true"
  android:layout_marginTop="456dp"
  android:background="@drawable/rectangle_4"/>

<item
  android:color="#D9FF9999"
  android:offset="0"/>
<item
  android:color="#D1FF1C6D"
  android:offset="1"/>

```

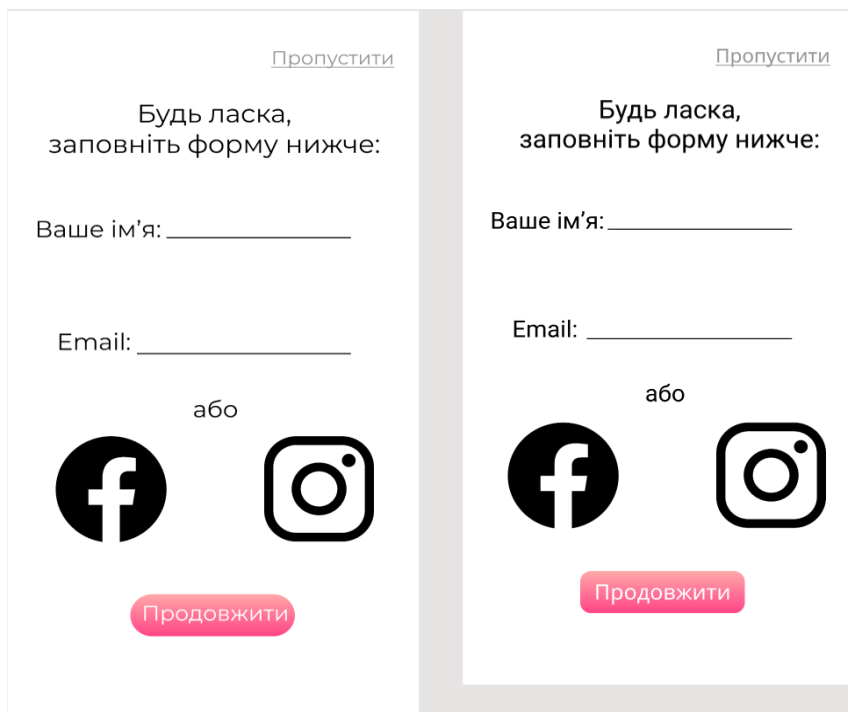


Рисунок 23 – «Вікно реєстрації»

Поля для вводу реалізовано наступним чином:

```

input[type="text"] {
  width: 300px;
  height: 40px;
}

```



```
padding: 8px;
font-size: 16px;
border: 1px solid #ccc; }
input[type="text"]:focus {
border-color: #007bff;
outline: none; }
```

Для Андроид:

```
<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Введіть текст"
    android:inputType="text"
    android:maxLines="1"
    android:singleLine="true" />
```

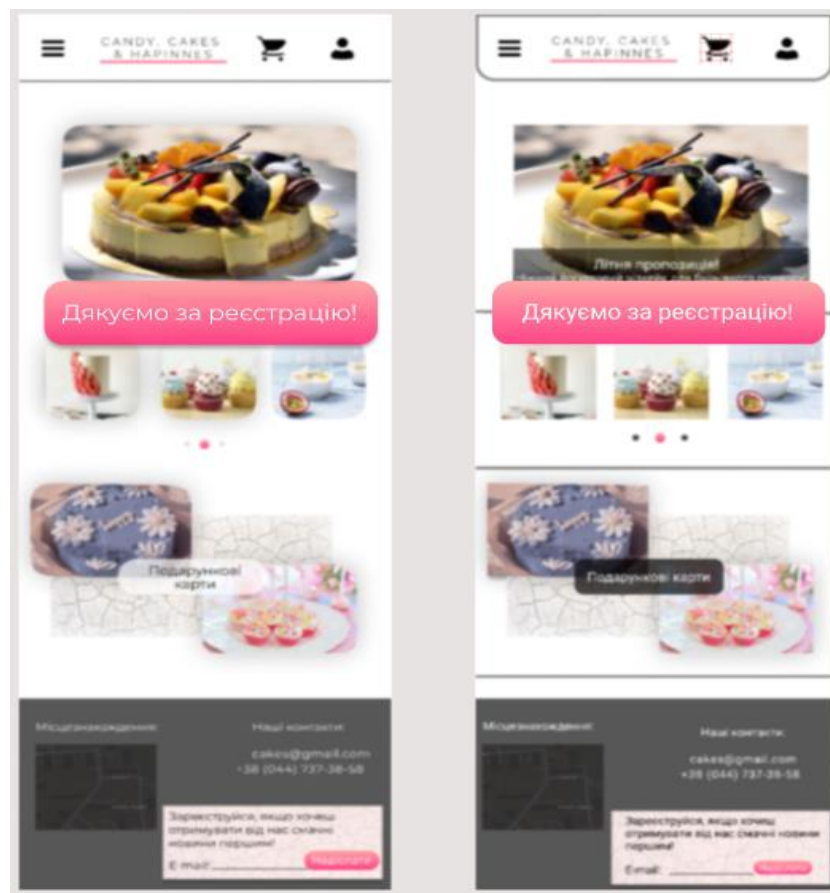


Рисунок 24 –Вікно «Дякуємо за реєстрацію»

Для iOS повідомлення «Дякуємо за реєстрацію»:

```

position: absolute;
width: 335px;
height: 72px;
left: 26px;
top: 298px;
background: linear-gradient(180deg, rgba(255, 153, 153,
0.85)0%, rgba(255, 28, 109, 0.82) 100%);
mix-blend-mode: normal;
box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);
border-radius: 20px;
/* Дякуємо за реєстрацію! */
position: absolute;
width: 301px;
height: 29px;
left: 20px;
top: 21px;
font-family: 'Montserrat';
font-style: normal;
font-weight: 400;
font-size: 24px;
line-height: 29px;
color: #FFFFFF;

```

Під ОС Android реалізація цієї функції має наступний вигляд:

```

<!-- Дякуємо за реєстрацію! -->
<TextView
    android:id="@+id/some_id"
    android:layout_width="279dp"
    android:layout_height="26dp"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="30dp"
    android:layout_alignParentTop="true"
    android:layout_marginTop="21dp"
    android:text="@string/some_id"
    android:textAppearance="@style/some_id"
    android:gravity="top"/>
<!--
    Font family: Roboto
    Line height: 28sp
-->
<!-- styles.xml -->
<style name="some_id">
    <item name="android:textSize">24sp</item>
    <item name="android:textColor">#FFFFFF</item>

```

```

</style>
<!-- strings.xml -->
    <string name="some_id">
</string>

```

Меню «Бургер меню» реалізовано на окремому активіті, складається з 5ти підпунктів, кожен з яких викликає відповідну сторінку мобільного додатку (рис. 25).

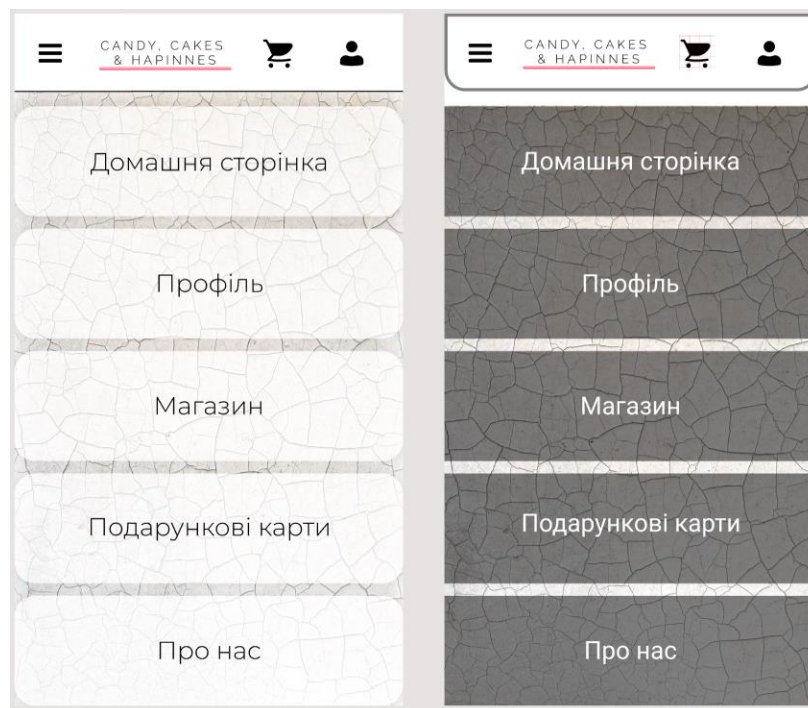


Рисунок 25 – Вікно «Бургер-меню»

Для iOS код кнопки в цьому меню реалізовано наступним чином:

```

position: absolute;
width: 375px;
height: 106px;
left: 0px;
top: 89px;
background: rgba(255, 255, 255, 0.7);
box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.15);
border-radius: 20px;

/* Домашня сторінка */
position: absolute;

```

```

width: 242px;
height: 28px;
left: 67px;
top: 128px;
font-family: 'Montserrat';
font-style: normal;
font-weight: 300;
font-size: 24px;
line-height: 29px;
text-align: center;
color: #000000;

```

Під ОС Андроїд реалізація відмінна, а саме:

```

<View
    android:id="@+id/rectangle_4"
    android:layout_width="360dp"
    android:layout_height="106dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="65dp"
    android:background="@drawable/rectangle_4"/>

<path
    android:pathData="M0 0V106H360V0"
    android:fillColor="#80000000"/>
</group>

<!-- Домашня сторінка -->
<TextView
    android:id="@+id/some_id"
    android:layout_width="229dp"
    android:layout_height="27dp"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="65dp"
    android:layout_alignParentTop="true"
    android:layout_marginTop="104dp"
    android:text="@string/some_id"
    android:textAppearance="@style/some_id"
    android:gravity="center_horizontal|top"/>
<!--
    Font family: Roboto
    Line height: 28sp
-->
<!-- styles.xml -->
<style name="some_id">

```

```

        <item name="android:textSize">24sp</item>
        <item name="android:textColor">#FFFFFF</item>
</style>
<!-- strings.xml -->
        <string name="some_id">
</string>

```

Шапка мобільного додатку буде містити в собі іконки для швидкого переходу до меню, корзини та профілю покупця.



Рисунок 26 – «Панель управління»

Під ОС Андроїд слід прописати у файлі-маніфесту наступний код, який буде описувати всі вищезазначені елементи:

```

<View
    android:id="@+id/rectangle_1"
    android:layout_width="360dp"
    android:layout_height="75dp"
    android:background="@drawable/rectangle_1"/>
    android:strokeWidth="6"
    android:strokeColor="#80000000"/>
</group>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/andro
id"
    android:id="@+id/user"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clipToOutline="true"
    android:background="@drawable/user"/>

```

Лейаути під корзину користувача та меню мають наступний код:

```

<!-- Shopping Cart -->
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/andro
id"
    android:id="@+id/shopping_ca"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clipToOutline="true"
    android:background="@drawable/shopping_ca"/>

<!-- Menu -->
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/andro
id"
    android:id="@+id/menu"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clipToOutline="true"
    android:background="@drawable/menu"/>
<!-- drawable/menu.xml -->

```

Посилання до головної сторінки додатку розміщено посередені хедера для зручності користувача:

```

<!-- CANDY, CAKES & hapinnes -->
<TextView
    android:id="@+id/candy_cakes"
    android:layout_width="172dp"
    android:layout_height="39dp"
    android:text="@string/candy_cakes"
    android:textAppearance="@style/candy_cakes"
    android:gravity="center_horizontal|top"/>
<!--
    Font family: Raleway
    Line height: 14sp
-->

<!-- styles.xml -->
<style name="candy_cakes">
    <item name="android:textSize">12sp</item>
    <item name="android:letterSpacing">0.25</item>
    <item name="android:textColor">#000000</item>
</style>
<!-- strings.xml -->

```

```

    <string name="candy_cakes">
      CANDY, CAKES \n& hapinnes
    </string>

```

Туж саму реалізацію слід прописати для іншої операційної системи:

```

position: absolute;
width: 375px;
height: 75px;
border-bottom: 1px solid #000000;

/* User */
position: absolute;
width: 35px;
height: 35px;

/* Shopping Cart */
position: absolute;
width: 35px;
height: 35px;

/* Menu */
position: absolute;
width: 35px;
height: 35px;

/* CANDY, CAKES & hapinnes */
position: absolute;
width: 172px;
height: 39px;
font-family: 'Raleway';
font-style: normal;
font-weight: 300;
font-size: 12px;
line-height: 14px;
text-align: center;
letter-spacing: 0.25em;
text-transform: uppercase;
color: #000000;

```

На домашній сторінці присутня картка продукту «Літня пропозиція», яка має різний дизайн відповідно до вимогам ОС:

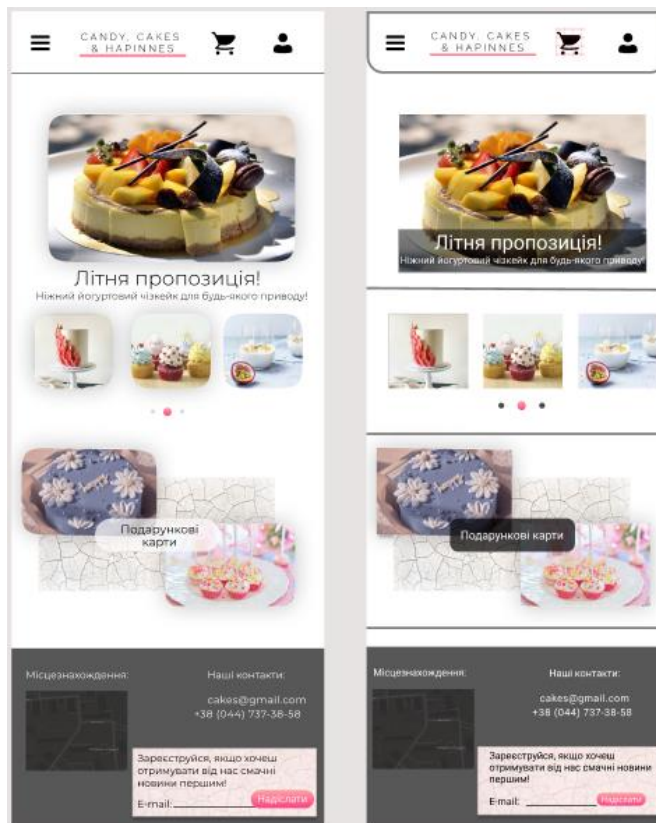


Рисунок 27 – «Домашня сторінка»

Так наприклад для iOS:

```

position: absolute;
width: 296px;
height: 177px;
left: 44px;
top: 124px;
background: url(image.png);
/* Літня пропозиція! */
    position: absolute;
    width: 224px;
    height: 29px;
    left: 75px;
    top: 306px;
    font-family: 'Montserrat';
    font-style: normal;
    font-weight: 300;
    font-size: 24px;
    line-height: 29px;
    text-align: center;
    color: #000000;
/* Ніжний йогуртовий чізкейк для будь-якого приводу! */

```



```

    position: absolute;
    width: 326px;
    height: 15px;
    left: 29px;
    top: 334px;
    font-family: 'Montserrat';
    font-style: normal;
    font-weight: 300;
    font-size: 12px;
    line-height: 15px;
/* identical to box height */
text-align: center;
color: #000000;

```

Для порівняння, наведемо частину коду під ОС Андроїд:

```

<View
android:id="@+id/image_1"
    android:layout_width="296dp"
    android:layout_height="177dp"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="44dp"
    android:layout_alignParentTop="true"
    android:layout_marginTop="100dp"
    android:background="@drawable/image_1"
    android:elevation="30dp"/>
<!-- Літня пропозиція! -->
<TextView
    android:id="@+id/some_id"
    android:layout_width="224dp"
    android:layout_height="29dp"
    android:layout_alignParentLeft="true"
    android:layout_marginLeft="75dp"
    android:layout_alignParentTop="true"
    android:layout_marginTop="282dp"
    android:text="@string/some_id"
    android:textAppearance="@style/some_id"
    android:gravity="center_horizontal|top"/>
<!--
Font family: Montserrat
Line height: 29sp
-->

```

```

<!-- styles.xml -->
  <style name="some_id">
    <item name="android:textSize">24sp</item>
    <item name="android:textColor">#000000</item>
  </style>
<!-- strings.xml -->
  <string name="some_id">
</string>

```

Реалізовані сторінки «Профіль користувача» наведено на рисунку 28:

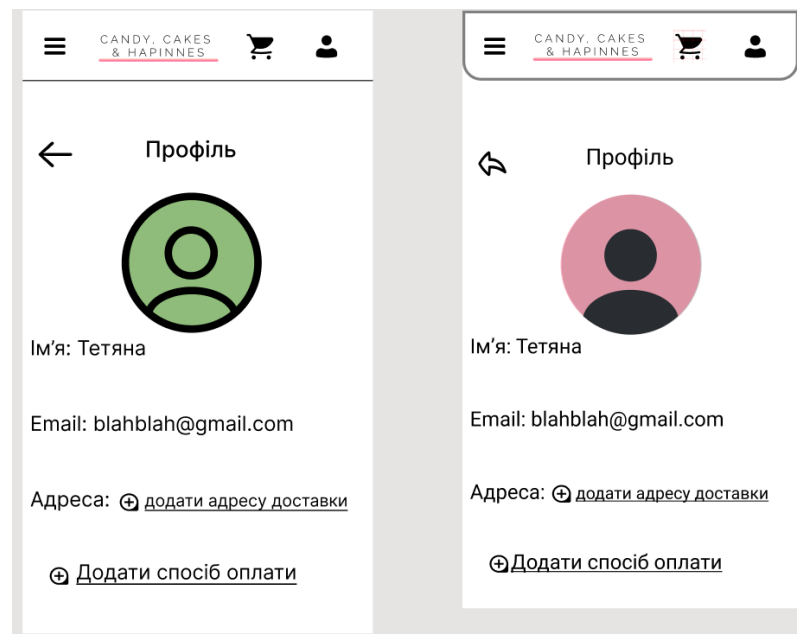


Рисунок 28 – «Профіль»

На кожній основній сторінці також є кнопка повернення до попередньої сторінки:

```

//Айос:
position: absolute;
left: 4.53%;
right: 85.87%;
top: 21.06%;
bottom: 74.86%;
background: #000000;

//під Андроид
<View

```

```

android:id="@+id/vector"
android:layout_width="0dp"
android:layout_height="0dp"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentTop="true"
android:layout_alignParentBottom="true"
android:background="@drawable/vector"/>
<!-- drawable/vector.xml -->

```

Скріншоти сторінок «Магазин», де буде відображено весь асортимент виробів, представлено на наступному рисунку:

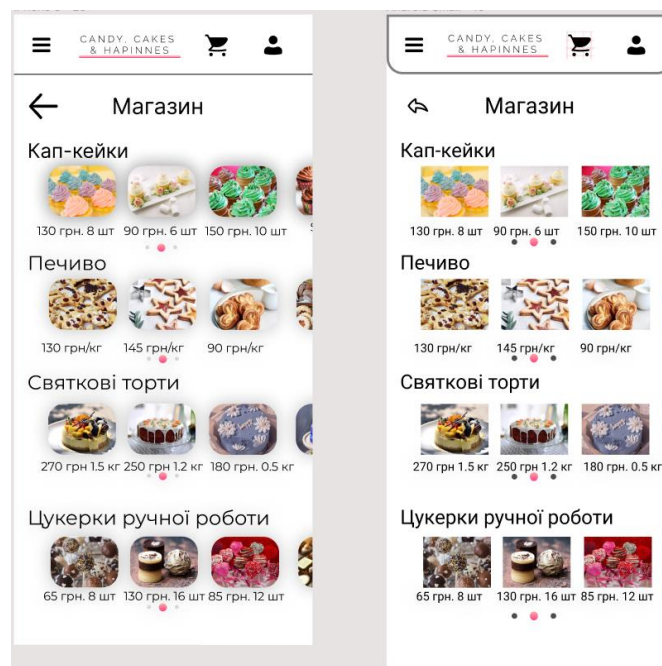


Рисунок 29 – Скрін сторінки «Магазин»

Крім цього, у дипломному проекті, відповідно до вимог, реалізовані сторінки для відображення асортименту «Подарункові набори» (рис. 30) та сторінка з інформацією про компанію (рис. 31).

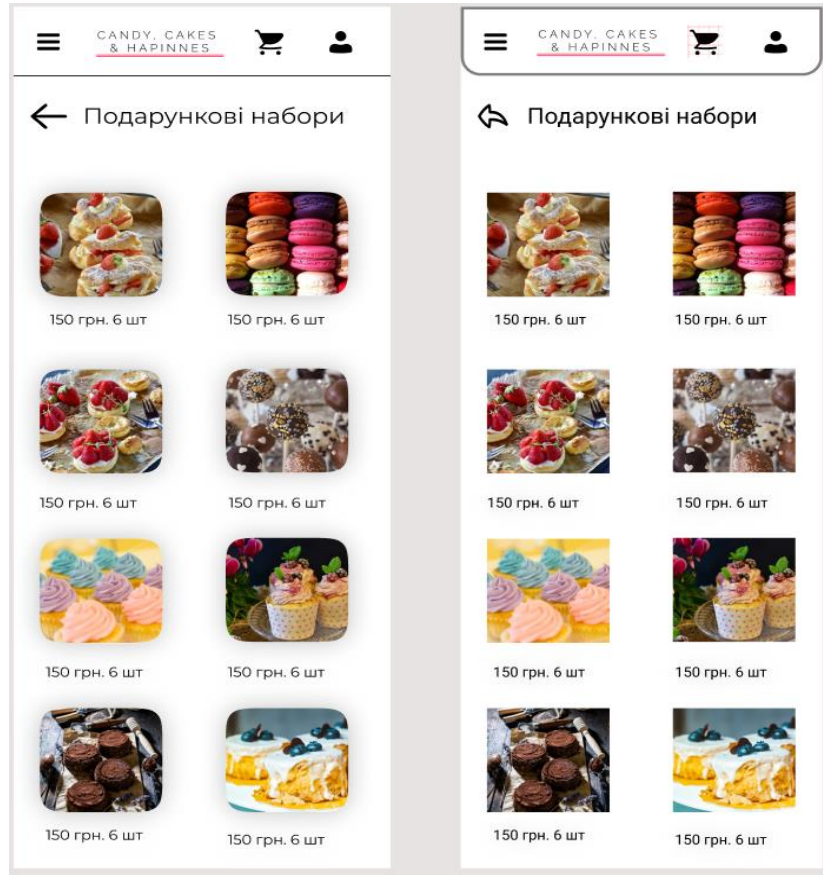


Рисунок 30 – «Подарункові набори»

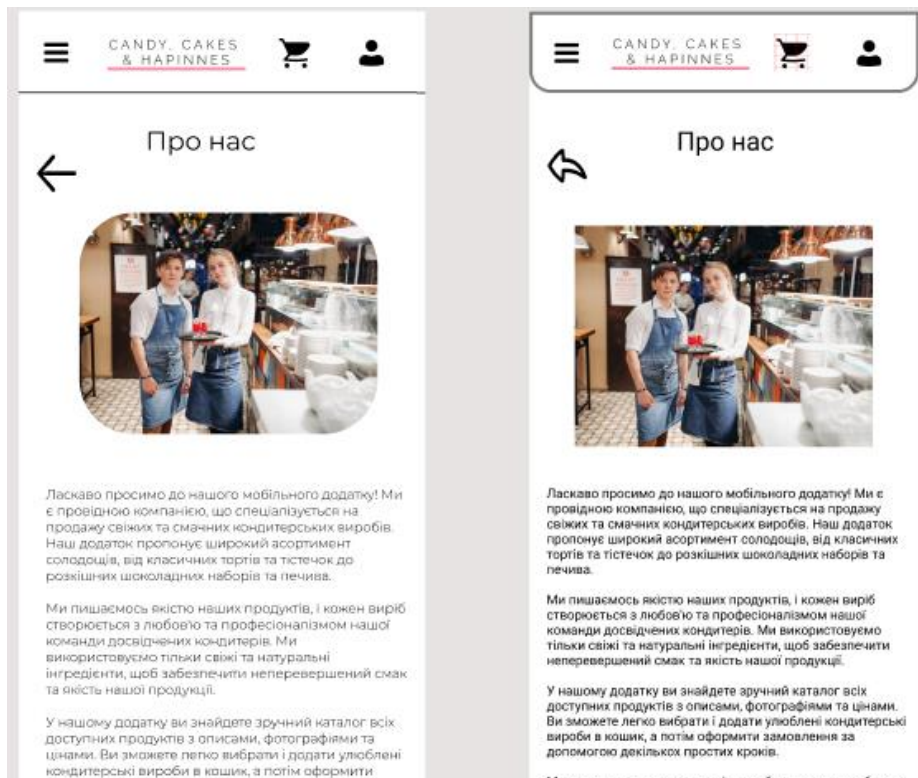


Рисунок 31 – «Про нас»

Якщо користувач хоче ознайомитись більш детально з обраним виробом, він переходить до відповідної сторінки (рис. 32). Після цього можна сформуванати замовлення (рис. 33).

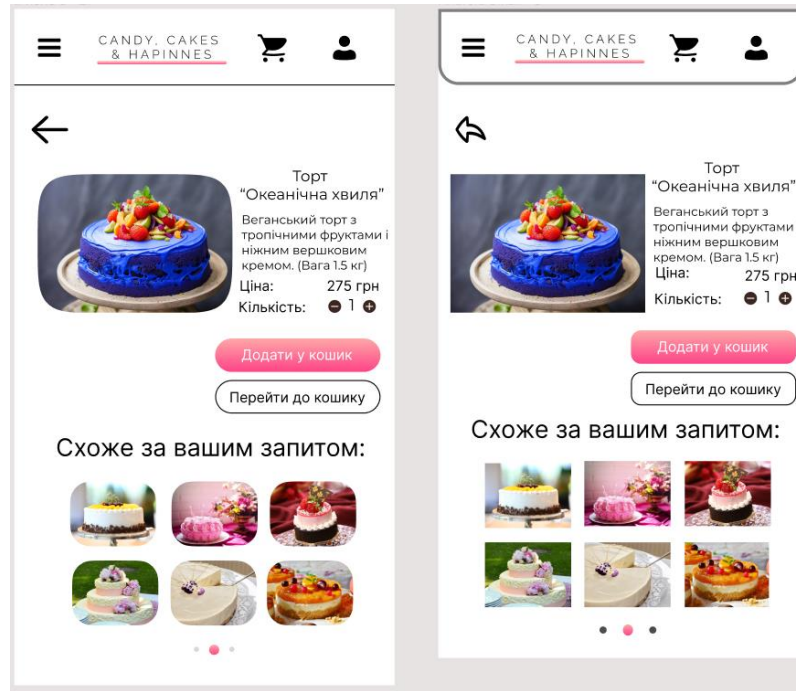


Рисунок 32 – «Опис товару»

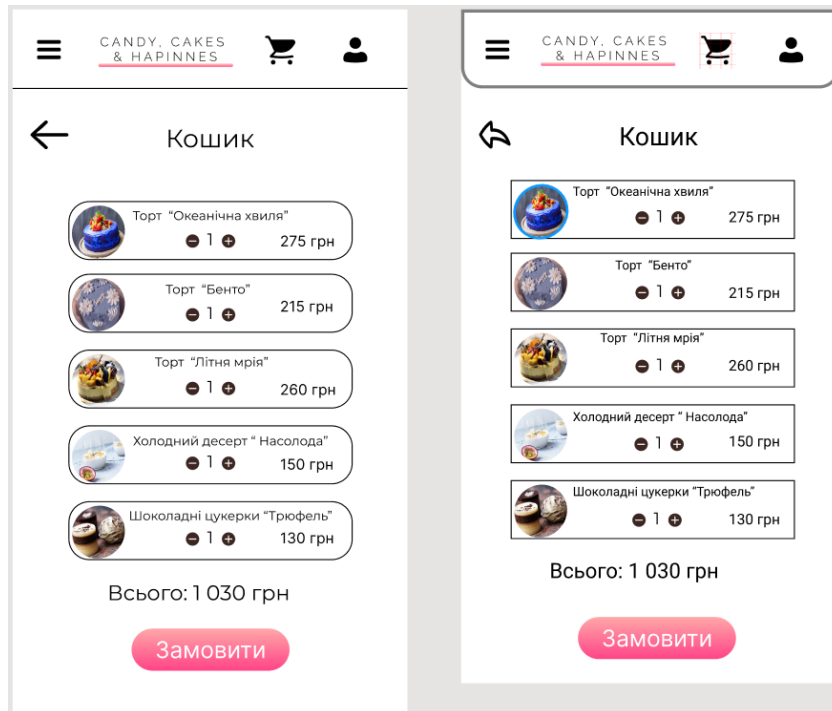


Рисунок 33 – «Кошик»

Після натискання кнопки «Замовити» користувачу відкривається сторінка з детальним описом (рис. 34):

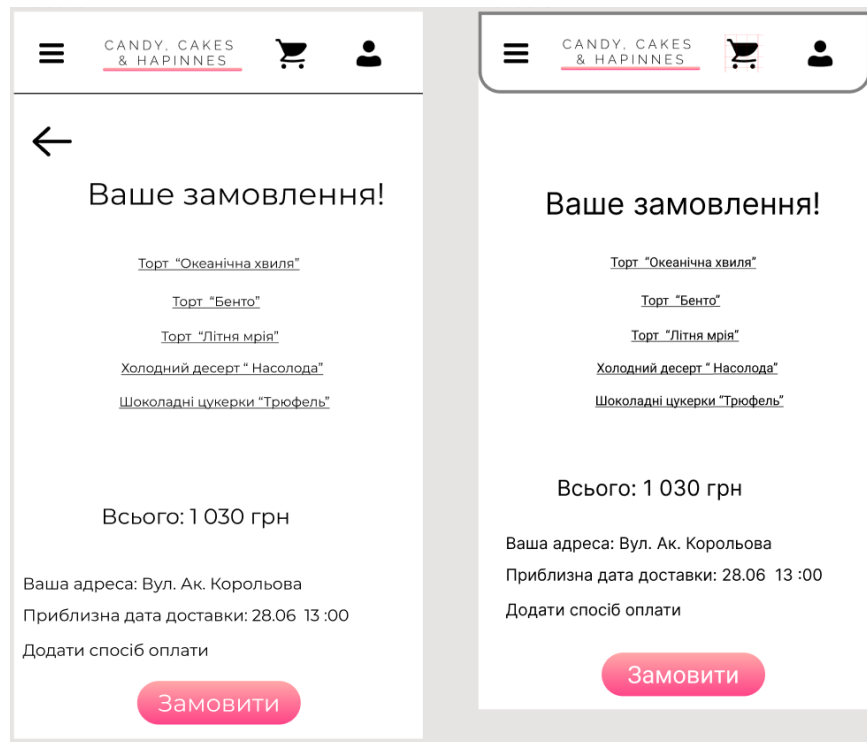


Рисунок 34 – «Підтвердження замовлення»

## ВИСНОВКИ

На даний час найчастіше замовники просять розробити кросплатформний мобільний додаток, щоб максимально охопити потенціальну цільову аудиторію. Таким чином, проектувальники та розробники повинні володіти двома напрямками, тобто нюансами розробки під обидві операційні системи.

У роботі було розглянуто:

- найважливіші аспекти розробки інтерфейсу мобільного додатка;
- види додатків;
- тенденції та вимоги UA/UX дизайну в цілому;
- відмінності дизайну під iOS та Android;
- найважливіших нюансів розробки інтерфейсу мобільних додатків.

В ході виконання дипломної роботи був отриманий повнофункціональний інтерфейс користувача, повністю готовий до застосування. Інтерфейс створеного додатку зручний, простий, відображає її можливості. Процес проектування виявився легким, якщо спочатку розібрати основні правила його створення.

Актуальність роботи полягає в тому, що кожний інтерфейс повинен бути зручний для користувача та задовільняти всі його потреби. В ході виконання проекту був отриманий готовий інтерфейс мобільного додатку, який може конкурувати на ринку праці.

Розроблений інтерфейс мобільного додатку задовольняє всім вимогам, поставленим на етапі постановки завдання.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Поняття користувачього інтерфейсу. URL: [https://studopedia.su/12\\_23280\\_ponyattya-koristuvatskogo-interfeysu.html](https://studopedia.su/12_23280_ponyattya-koristuvatskogo-interfeysu.html) (дата звернення 10.04.2023)
2. Дослідження етапів розробки програмного забезпечення на основі інформаційної моделі URL: <https://ua-referat.com/uploaded/doslidjennya-etapiv-rozrobki-programnogo-zabezpechennya/index1.html> (дата звернення 10.04.2023)
3. Підходи до проектування URL: [https://msn.khmnu.edu.ua/pluginfile.php/273378/mod\\_resource/content/3/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F1.pdf](https://msn.khmnu.edu.ua/pluginfile.php/273378/mod_resource/content/3/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F1.pdf) (дата звернення 17.04.2023)
4. Моделі життєвого циклу, принципи і методології розробки програмного забезпечення (ПЗ) URL: [vergreens.com.ua/ua/articles/software-development-metodologies.html](http://vergreens.com.ua/ua/articles/software-development-metodologies.html) (дата звернення 17.04.2023)
5. Інтерфейс користувача. Командні рядки. Меню. Графічний інтерфейс користувача URL: [https://msn.khmnu.edu.ua/pluginfile.php/277742/mod\\_resource/content/2/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F4.pdf](https://msn.khmnu.edu.ua/pluginfile.php/277742/mod_resource/content/2/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F4.pdf) (дата звернення 06.05.2023)
6. Лекція №1. Користувачький інтерфейс URL: <https://pandia.ru/text/79/519/25522.php> (на українській мові) (дата звернення 23.04.2023)
7. Sprinkle URL: [https://play.google.com/store/apps/details?id=blueappsoftware.sprinklecake&hl=en\\_US&pli=1](https://play.google.com/store/apps/details?id=blueappsoftware.sprinklecake&hl=en_US&pli=1) (дата звернення 10.05.2023)
8. 10 Best Bakery Apps URL: <https://devtechnosys.com/top-platforms/top-bakery-apps.php> (дата звернення 27.05.2023)
9. Meet Android Studio URL: <https://developer.android.com/studio/intro> (дата звернення 16.05.2023)
10. Visual Studio Code FAQ URL: <https://code.visualstudio.com/docs/>



supporting/faq (дата звернення 16.05.2023)

11. SwiftUI URL: <https://developer.apple.com/xcode/swiftui/> (дата звернення 21.05.2023)