МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»	**«До захисту допущено»**
Директор інституту(декан факультету)	Завідувач кафедри
_____ ___Андрій ФОРСЮК__	_____ _Сергій ГРИБКОВ__
(підпис) (ім'я та прізвище)	(підпис) (ім'я та прізвище)
«10» __червня_____ 2024р.	«10» ____червня_____ 2024р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності 122 «Комп'ютерні науки»
(код та назва спеціальності)

освітньо-професійної програми __Інформаційні системи та штучний інтелект

на тему: «Розроблення мобільного додатку для продажу ігрових девайсів»

Виконав: здобувач _4_ курсу, групи __КН-4-4___

_____Попов Іван Владиславович_____ _____
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник ___Мазуренко Ольга Олександрівна_____ _____
(прізвище , ім'я та по батькові повністю) (підпис)

Консультанти _____ _____
(ім'я та прізвище) (підпис)

_____ _____
(ім'я та прізвище) (підпис)

_____ _____
(ім'я та прізвище) (підпис)

Рецензент _____ _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач_____
(підпис)

Київ - 2024р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки
Освітній ступінь бакалавр
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма Інформаційні системи та штучний інтелект

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інформаційних технологій, штучного інтелекту і кібербезпеки
Сергій ГРИБКОВ

" 15 " квітня 2024 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Попов Іван Владиславович
(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення мобільного додатку для продажу ігрових девайсів»
керівник роботи Мазуренко Ольга Олександрівна, ст.вик, к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від 15 квітня 2024 року № 279-кс
2. Строк подання здобувачем роботи 03.06.2024 р.
3. Вихідні дані до роботи
Розроблений додаток для iOS, написаного на мові Swift з використанням фреймворку SwiftUI та Firebase.
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
   1) Обґрунтування доцільності розробки додатку
   2) Постановка задачі на розробку мобільного додатку
   3) Моделювання та конструювання програмного забезпечення
   4) Опис реалізованих функцій додатку "Funkeys"
   5) Розроблення інструкції користувача для розробленого проекту
   6) Заходи охорони праці
   7) Висновки про розроблений мобільний застосунок
5. Перелік графічного матеріалу
   1) Організаційна структура
   2) Функціональні моделі бізнес-процесів
   3) Схема бази даних
   4) Приклади роботи програмного додатку та фрагменти його коду

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Мазуренко О. О., ст.вик, к.т.н.	20.04.2024	25.04.2024
2	Мазуренко О. О., ст.вик к.т.н.	20.04.2024	01.05.2024
3	Мазуренко О. О., ст.вик, к.т.н.	20.04.2024	06.05.2024
4	Мазуренко О. О., ст.вик, к.т.н.	20.04.2024	10.05.2024

7. Дата видачі завдання_____15 квітня 2024 року_____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної області	08.04.24-15.04.24	
2	Розробка концептуальної моделі системи	15.04.24-20.04.24	
3	Вибір технологій та інструментів розробки	20.04.24-24.04.24	
4	Розробка технічного завдання	24.04.24-28.04.24	
5	Розробка функціональності додатку	28.04.24-14.05.24	
6	Оформлення пояснювальної записки	14.05.24-20.05.24	

Здобувач _____ Попов І.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Мазуренко О. О.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з 9 таблиць, 32 рисунків та 17 джерел, загалом 68 сторінки.

Дипломний проєкт присвячений розробці мобільного застосунку для продажу ігрових девайсів для цифрової компанії “Funkeys”.

Метою даного дипломного проєкту є створення мобільного застосунку, який забезпечує зручний і ефективний процес вибору та покупки ігрових девайсів.

Об’єкт дослідження: інтерактивний процес вибору та покупки ігрових девайсів у мобільному середовищі.

Предмет дослідження: мобільний застосунок для цифрової копмпанії “Funkeys”, який включає функціонал реєстрації, навігації по категоріях товарів, кастомізації продуктів, управління замовленнями та інтеграцію з Firebase для збереження даних.

Дипломний проєкт включає розробку системи, яка дозволяє користувачам переглядати, кастомізувати та замовляти ігрові девайси. Застосунок пропонує інтуїтивно зрозумілий інтерфейс з можливістю реєстрації, перегляду товарів за категоріями, додавання товарів до кошика з відображенням загальної ціни замовлення та кастомізацій. Розробка виконана з використанням SwiftUI та Firebase, що забезпечує надійне зберігання даних та управління замовленнями.

КЛЮЧОВІ СЛОВА: МОБІЛЬНИЙ ЗАСТОСУНОК, ІГРОВІ ДЕВАЙСИ, IOS, XCODE, SWIFTUI, SWIFT, GOOGLE FIREBASE, ІНТЕРФЕЙС КОРИСТУВАЧА, FUNKEYS

ABSTRACT

The explanatory note of the diploma project consists of 9 tables, 32 figures, and 17 sources, a total of 68 pages.

The graduation project is dedicated to the development of a mobile application for the sale of gaming devices for the digital company Funkeys.

The purpose of this diploma project is to create a mobile application that provides a convenient and efficient process for choosing and buying gaming devices.

Research object: an interactive process of choosing and buying gaming devices in a mobile environment.

The subject of the study: a mobile application for the digital campaign ‘Funkeys’, which includes the functionality of registration, navigation through product categories, product customisation, order management and integration with Firebase for data storage.

The graduation project includes the development of a system that allows users to browse, customise and order gaming devices. The application offers an intuitive interface with the ability to register, browse products by category, add items to the cart with the display of the total order price and customisations. The development was carried out using SwiftUI and Firebase, which ensures reliable data storage and order management.

KEYWORDS: MOBILE APPLICATION, GAMING DEVICES, IOS, XCODE, SWIFTUI, SWIFT, GOOGLE FIREBASE, USER INTERFACE, FUNKEYS

ЗМІСТ

ВСТУП

У сучасному цифровому світі, де люди проводять значну частину свого часу за персональними комп’ютерами, вибір правильної клавіатури стає ключовим для комфорту та продуктивності. Незалежно від професії — чи то програмісти, геймери, чи блогери — якість тактильних відчуттів та швидкість реакції клавіатури можуть значно покращити досвід користувача. Подібно до впливу комфортного ліжка на якість сну, ідеально підібрана клавіатура може зробити робочий процес не лише ефективнішим, але й приємнішим.

З огляду на стрімкий розвиток технологій, актуальність інформатизації установ стає все більш важливою. Це особливо стосується індустрії електронної комерції, де зручність та швидкість є ключовими факторами успіху. Компанія “Funkeys”, яка спеціалізується на продажу ігрових девайсів, не є винятком. Інформатизація таких установ дозволяє не лише підвищити ефективність роботи, але й забезпечити кращий досвід для користувачів.

Актуальність теми розробки мобільного застосунку для “Funkeys” полягає у необхідності адаптації до змінюваних потреб ринку та забезпечення зручного доступу до продуктів для широкого кола користувачів. Мобільні застосунки відкривають нові можливості для бізнесу, дозволяючи залучити більше клієнтів. Розробка такого застосунку є важливим кроком у розвитку компанії та її адаптації до сучасних умов ведення бізнесу.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ САЙТУ “FUNKEYS”

1.1. Загальна характеристика “Funkeys”

Інтернет магазин механічних клавіатур "Funkeys" [1] є першою цифровою компанією в Україні, спеціалізованим магазином, що пропонує унікальні клавіатури для геймерів та ентузіастів. "Funkeys" створений у 2019 році людьми, які хочуть отримувати задоволення від своїх пристроїв, якими вони користуються щодня. Компанія, яка має власний асортимент та склад у м.Київ, а також є офіційним представником багатьох світових виробників механічних клавіатур, таких як Keychron, Leopold, Ducky, Varmilo, Meletrix та інші. Вони відрізняється від своїх конкурентів високою якістю продукції, швидкою доставкою, гарантією та сервісом, та має свою спільноту в соціальних мережах, де він ділиться новинами, порадами, відгуками та фотографіями своїх продуктів.

Магазин "Funkeys" спеціалізується на клавіатурах, також продають кейкапи, аксесуари для модингу клавіатур і перемикачі, також продають килими та миші при наявності. Також вони надають можливість замовити індивідуальні клавіатури з вибором типу перемикачів, корпусу та інших параметрів.

1.2. Організаційна структура “Funkeys”, роль і взаємодія підрозділів

1.2.1. Загальна схема організаційної структури

Цифрова компанія “Funkeys” немає філій, а є тільки офіс та інтернет-магазин для розповсюдження товарів. В контексті організаційних структур таку компанію можна описати як плоску організаційну структуру чи централізовану структуру. У плоскій організаційній структурі менше рівнів управління, що сприяє швидкому прийняттю рішень та гнучкості. Таким чином всі ключові рішення приймаються у головному офісі, оскільки відсутні окремі філії.

Організаційна структура цифрової компанії виглядає наступним чином:

1. Головний офіс: центральне місце управління та координації всіх операцій компанії.
2. Інтернет-магазин: основний канал продажу та взаємодії з клієнтами.

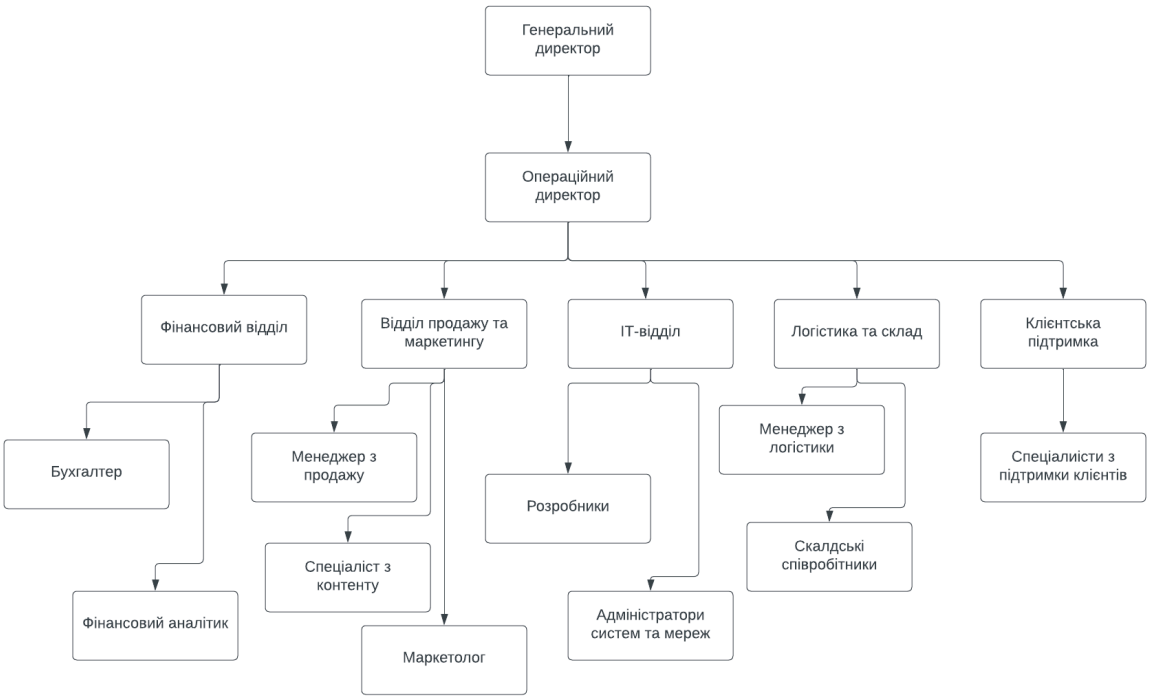


Рисунок 1.1 - Організаційна схема цифрової фірми “Funkeys”

Таблиця 1.1. Основні відділи та їх функції

Керівництво	• Генеральний директор (CEO): відповідає за стратегічне управління та загальні напрями розвитку компанії. • Операційний директор (COO): керує повсякденними операціями та координує роботу відділів.
IT-відділ	• Розробники: підтримують та розвивають платформу інтернет-магазину, забезпечують її працездатність та безпеку. • Адміністратори систем та мереж: забезпечують технічну підтримку та управління IT-інфраструктурою компанії.

Продовження таблиці 1.1.

Відділ продажу та маркетингу	• Менеджери з продажу: відповідають за взаємодію з клієнтами, обробку замовлень та підтримку клієнтів. • Маркетологи: розробляють та реалізують маркетингові кампанії, просувають товари через інтернет, аналізують ринок та потреби клієнтів. • Спеціалісти з контенту: створюють та керують контентом для сайту, включаючи описи товарів, блоги та рекламні матеріали.
Логістика та склад	• Менеджери з логістики: координують постачання товарів, управління запасами та доставку продукції клієнтам. • Складські співробітники: займаються прийманням, зберіганням та відправкою товарів.
Фінансовий відділ	• Бухгалтери: ведуть облік фінансових операцій, контролюють бюджет та складають фінансові звіти. • Фінансові аналітики: аналізують фінансові показники та допомагають у прийнятті стратегічних рішень.
Клієнтська підтримка	• Спеціалісти з підтримки клієнтів: обробляють запити та скарги клієнтів, надають допомогу з питань використання продуктів та рішень.

Ця структура дозволяє компанії ефективно управляти своїми операціями, забезпечувати високий рівень обслуговування клієнтів та швидко адаптуватися до змін на ринку.

Майже всі відділи пов’язані з інтеренет-магазином та виконують наступні основні задачі:

- IT-відділ: відповідає за технічний стан та функціональність сайту, його стабільну роботу та безпеку.
- Відділ маркетингу та продажів: відповідає за залучення клієнтів, покращення користувальницького досвіду та контент сайту.
- Відділ клієнтської підтримки: вирішує питання та проблеми клієнтів, пов'язані з використанням інтернет-магазину.
- Логістичний відділ: управляє обробкою замовлень та їх доставкою клієнтам.

Також є склади, де зберігаються товари перед відправкою клієнтам. Склади відіграють ключову роль у забезпеченні своєчасної обробки замовлень та ефективної логістики.

Таким чином, за супровід інтернет-магазину відповідає насамперед IT-відділ, але успішна робота потребує координації та взаємодії з іншими відділами.

1.3. Аналіз нинішнього стану комп’ютеризації

Цифрові компанії такі як “Funkeys” (без філій та з інтернет-магазином), як правило, сильно комп'ютеризована. Комп'ютерні системи та технології використовуються практично у всіх аспектах бізнесу, від розробки та підтримки сайту до управління замовленнями, маркетингу та обслуговування клієнтів. Це дозволяє компанії ефективно працювати, забезпечувати високий рівень обслуговування клієнтів та швидко реагувати на зміни ринку. Ось основні аспекти її комп'ютеризації:

Інтернет магазин

- Платформа інтернет-магазину: використання веб-сервера та програмного забезпечення для управління каталогом товарів, кошиком покупок, оплатою та обробкою замовлень.
- Контент-менеджмент: системи управління контентом (CMS) для створення та оновлення описів товарів, блогів, зображень та іншого контенту.

Інфраструктура та IT

- Сервери та хостинг: розміщення інтернет-магазину на власних або орендованих серверах, забезпечення їхньої безпеки та доступності.
- Хмарні технології: використання хмарних сервісів для зберігання даних, резервного копіювання та масштабування ресурсів.

Управління замовленнями та логістика

- Системи керування замовленнями (OMS): програмне забезпечення для відстеження замовлень, керування запасами та координації доставки.
- Інтеграція з кур'єрськими службами: автоматизоване створення накладних, відстеження посилок та повідомлення клієнтів.

Маркетинг та аналітика

- Інструменти веб-аналітики: використання сервісів (наприклад, Google Analytics) для аналізу поведінки користувачів, ефективності маркетингових кампаній та оптимізації сайту.
- Автоматизація маркетингу: системи керування розсилками, таргетованої рекламою та персоналізованими пропозиціями.

Фінансові та бухгалтерські системи

- Автоматизація бухгалтерії: програмне забезпечення для обліку фінансових операцій, виставлення рахунків, управління податками та підготовки звітності.
- Платіжні системи: інтеграція з онлайн-платежами для безпечної та зручної оплати замовлень.

Використовувані технології та інструменти:

- E-commerce платформа: Shopify.

- Система аналітики: Google Analytics.
- Платіжні системи: Monobank еквайринг.

1.4. Розроблення функціональної моделі та аналіз існуючих бізнес-процесів

1.4.1. Функціональна модель “Funkeys”.

Модель бізнес-процесів, створена у Bizagi Modeler [3], відображає ключові етапи взаємодії клієнта з цифровою фірмою “Funkeys”, починаючи з відвідування вебсайту і закінчуючи доставкою товару. Основні блоки моделі включають:

- Формування та оплата замовлення.
- Перевірка наявності товару.
- Обробка замовлення.
- Зберігання на складі.
- Перевірка необхідності кастомізації.
- Кастомізація товару (за потреби).
- Передача товару компанії доставки.

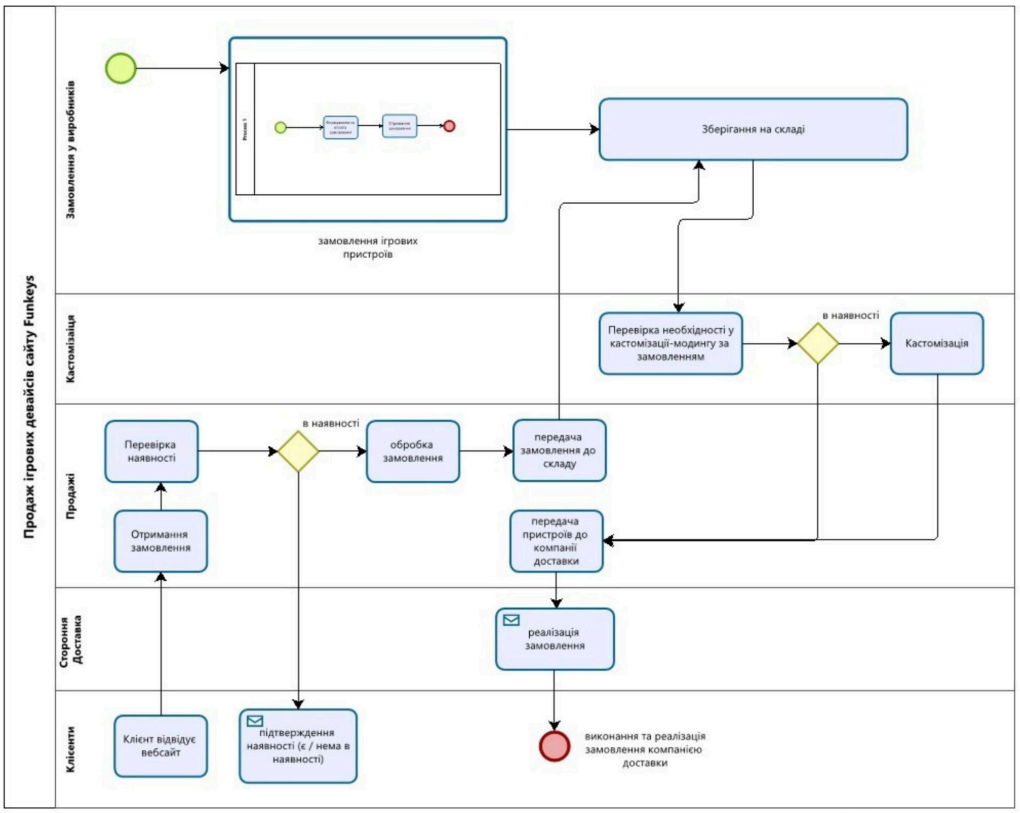


Рисунок 1.2 - Модель бізнес-процесу онлайн-продажу ігрових пристроїв

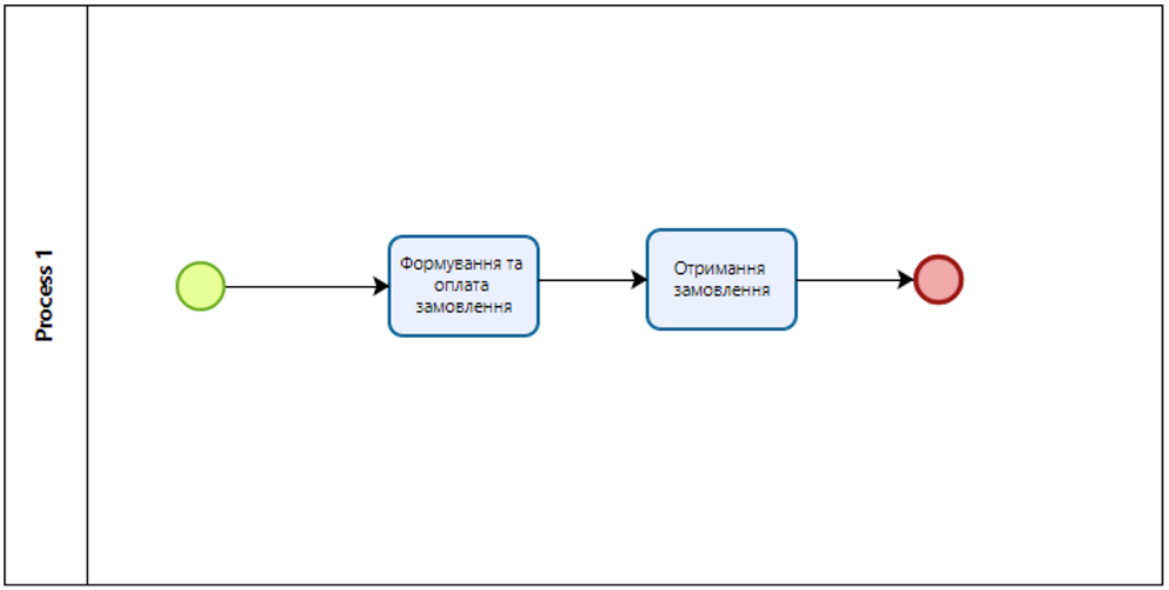


Рисунок 1.3 - Модель підпроцессу «Замовлення ігрових пристроїв»

1.4.2. Виявлені проблеми в існуючих бізнес-процесах цифрової фірми “Funkeys”

Аналізуючи продажі фірми можна виявити декілька ключових проблем, які впливають на ефективність та задоволення потреб клієнтів. Ці проблеми можуть мати корінь у різних діяльностях магазину, від логістики до взаємодії з клієнтами. Розглянемо виявлені проблеми:

Комунікація з клієнтами

Комунікація з клієнтами також потребує уваги. Наявна система не забезпечує автоматичного повідомлення клієнтів про статус їх замовлень, що може призвести до невпевненості та зниження довіри до магазину. Важливо, щоб клієнти мали можливість відстежувати свої замовлення в реальному часі та отримувати оновлення про будь-які зміни або затримки.

Відсутність інтегрованої системи акаунтів

Відсутня інтегрована система акаунтів, яка б дозволяла клієнтам створювати особисті профілі для відстеження замовлень та управління персоналізованим налаштуванням. Впровадження подібної системи на вебсайті може значно покращити досвід користувачів.

1.4.3. Задачі автоматизації

У контексті сучасного ринку, “Funkeys” стоїть перед необхідністю оптимізації своїх бізнес-процесів через автоматизацію. Це стосується як внутрішніх операцій, так і взаємодії з клієнтами. Розглянемо ключові задачі автоматизації, які можуть підвищити ефективність та зручність користування, а також полегшити роботу інших відділів.

Перш за все, важливо зосередитися на автоматизації процесу перевірки наявності товарів. Впровадження системи управління запасами, яка інтегрована з онлайн-платформою магазину. Така система дозволить автоматично оновлювати інформацію про наявність товарів, зменшуючи час на обробку замовлень та знижуючи ризик помилок, пов’язаних з ручним введенням даних.

Розробка інтерактивного інструменту для кастомізації товарів на сайті “Funkeys” може значно покращити досвід користувачів. Клієнти зможуть в режимі реального часу вибирати додаткові опції для товарів, що не тільки збільшить задоволеність клієнтів, але й оптимізує роботу відділу кастомізації.

Комунікація з клієнтами є ще однією важливою сферою для автоматизації. Автоматичні повідомлення про статус замовлення забезпечать клієнтів актуальною інформацією та зменшать навантаження на службу підтримки.

Ще одним аспектом, який потребує автоматизації, є введення системи акаунтів для користувачів на вебсайті. Створення особистих акаунтів на сайті дозволить клієнтам зберігати історію замовлень, відстежувати статуси та отримувати персоналізовані пропозиції. Це не тільки підвищить лояльність клієнтів, але й забезпечує збір цінної аналітичної інформації для маркетингових цілей.

Отже, розглянуті задачі автоматизації зосереджені на ефективності, зручності користування клієнтами та полегшенні та оптимізації роботи інших відділів. Під час аналізу наявних варіантів вирішення проблем “Funkeys” важливо звернути увагу на наступні можливості:

Розширення функціоналу вебсайту

Впровадження додаткових інструментів для персоналізації досвіду користувачів є критично важливим для підвищення залученості та задоволення клієнтів. Це може включати рекомендаційні системи, які аналізують попередні покупки та перегляди, щоб пропонувати товари, які найбільше відповідають інтересам користувача. Також, інтеграція інтерактивних елементів, таких як віртуальний примірочний зал або 3D-перегляд товарів, може значно підвищити взаємодію з сайтом та сприяти більш обдуманим покупкам.

Інтеграція з соціальними мережами

Автоматизація публікації оновлень та акцій на сторінках магазину в соціальних мережах є важливою для підтримання постійного зв’язку з клієнтами. Використання інструментів для планування та аналітики дозволить “Funkeys” ефективно управляти своєю присутністю в соціальних мережах, відстежувати вплив рекламних кампаній та залучати нових клієнтів через цілеспрямовані рекламні акції.

Впровадження мобільного додатку

Розробка мобільного додатку, який забезпечить зручний доступ до товарів та послуг “Funkeys”, є ключовим аспектом стратегії цифрової трансформації. Мобільний додаток може включати функції швидкої оплати, персоналізованих сповіщень та відстеження замовлень. Це не тільки полегшить процес покупки для клієнтів, але й відкриє нові канали збору даних для аналітики поведінки користувачів.

Зважаючи на виявлені проблеми в компанії, розгляд існуючих рішень може стати основою для подальшої розробки мобільного додатку, спрямованого на вирішення потреб фірми “Funkeys”. Мобільний додаток може стати центральним елементом у взаємодії з клієнтами, забезпечуючи їм зручний інтерфейс для взаємодії з магазином, що в кінцевому підсумку може підвищити продажі та покращити користувацький досвід.

1.5. Обґрунтування доцільності проектування й розроблення мобільного додатку “Funkeys”

Цифрова фірма “Funkeys” вже зарекомендувала себе як популярний ресурс серед любителів ігрових пристроїв. Проте, у світі, де цифрові технології постійно еволюціонують, важливо не лише підтримувати поточний успіх, а й розвиватися, щоб задовольняти зростаючі потреби та очікування клієнтів. У цьому контексті, розробка мобільного додатку може стати стратегічним кроком, який відкриває нові горизонти для “Funkeys”.

Розробка мобільного додатку для продажу ігрових девайсів є актуальною і перспективною роботою, яка відповідає сучасним технологічним та ринковим тенденціям, оскільки кількість користувачів мобільних пристроїв в Україні є досить високою. Згідно зі статистикою проникнення інтернету в Україні на 2019 рік, яку провела "Fractum Group" [4], типи доступу "регулярних" інтернет-користувачів налічують 66 відсотків серед мобільних пристроїв (зображено на рис. 1.4). Тенденція зростання користувачів інтернету через мобільні телефони з кожним роком збільшується (додаток Г), що робить мобільні додатки все більш важливими для бізнесу [5].

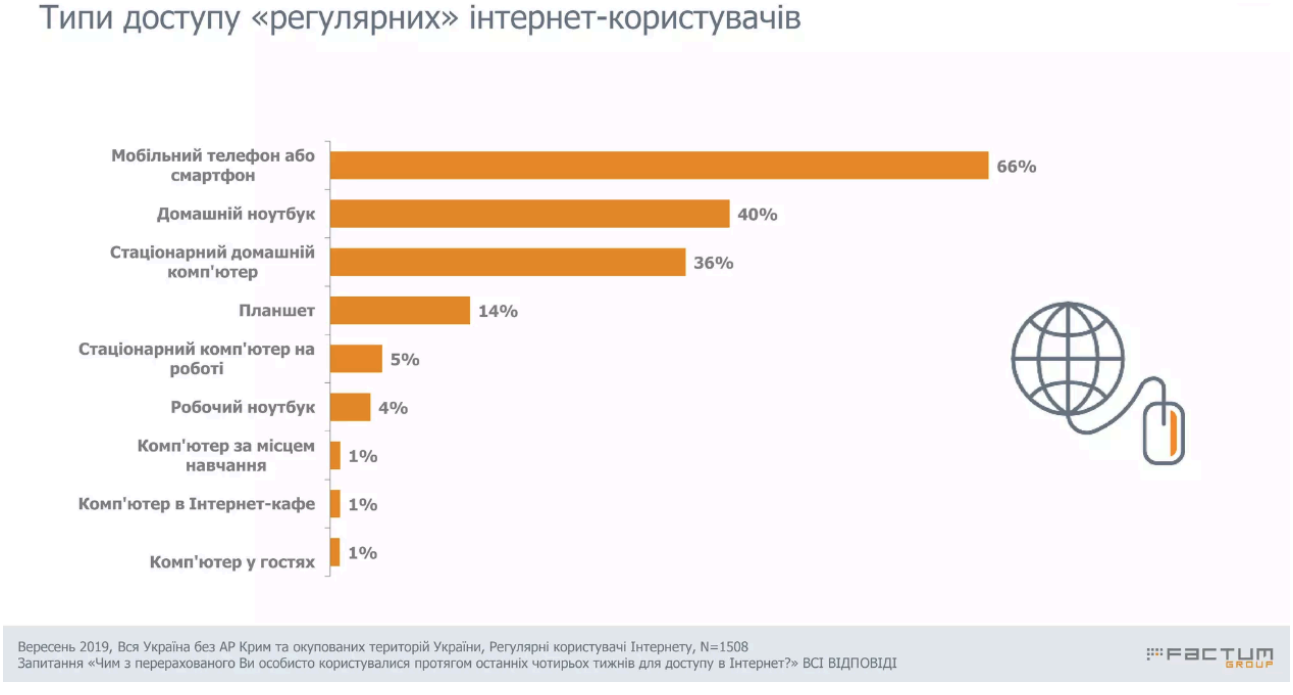


Рисунок 1.4 - Типи доступу інтернет-користувачів

Сайт магазину, хоча і має адаптивний дизайн, але не завжди зручний для користування з мобільних пристроїв. Також сайт магазину не має функції повідомлень, яка б дозволяла інформувати клієнтів про стан замовлення.

Мобільний додаток для продажу ігрових девайсів би вирішив ці проблеми та додав би нові функції, які б покращили користувацький досвід та збільшили продажі, він би мав такі переваги:

- Мобільний додаток забезпечить користувачам швидкий доступ до асортименту товарів, збереження замовлень, а також спростить процесс покупки завдяки оптимізованому інтерфейсу.
- Більша взаємодія та залученість: Мобільний додаток би дозволяв надсилати повідомлення та сповіщення клієнтам про новинки та статус замовлень, та надасть можливість здійснювати мобільні платежі, популярність яких зростає.
- Додаток може включати систему бонусів, знижок та привілеїв для постійних клієнтів. Це спонукає користувачів повертатися і здійснювати покупки.
- Мобільний додаток може адаптуватися до індивідуальних потреб користувачів. Наприклад, він може рекомендувати товари на основі попередніх покупок або переглядів.
- Наявність додатку б дала перевагу над іншими магазинами, які не мають такого додатку, а також би демонстрував сучасність та прогресивність магазину, який йде в ногу з технологічними трендами.

На основі аналізу, викладеного в пунктах 1.3, 1.4.2 та тексту наведеного вище у цьому пункті можна зробити висновок, що розроблення мобільного додатку “Funkeys” є не тільки доцільним, але й необхідним кроком для подальшого розвитку цифрової фірми. Так як фірма є цифровою та комп’ютеризованою, то самостійна розробка мобільного додатку є більш вигідною та менш фінансово затратною (додаток Г). Враховуючи сучасні тенденції зростання мобільного інтернету та зміну споживацьких звичок,

мобільний додаток стане важливим інструментом для забезпечення зручності, швидкості та забезпечення сталого зростання магазину в цифрову епоху.

1.6. Концептуальна модель системи

Модель системи була оптимізована для забезпечення більшої зручності та ефективності, використовуючи сучасні технології та платформи для мобільного застосунку.

Для опису бізнес-процесів для мобільного додатку використовується BPMN модель у Lucidchart [2] (рис. 1.5-1.9)

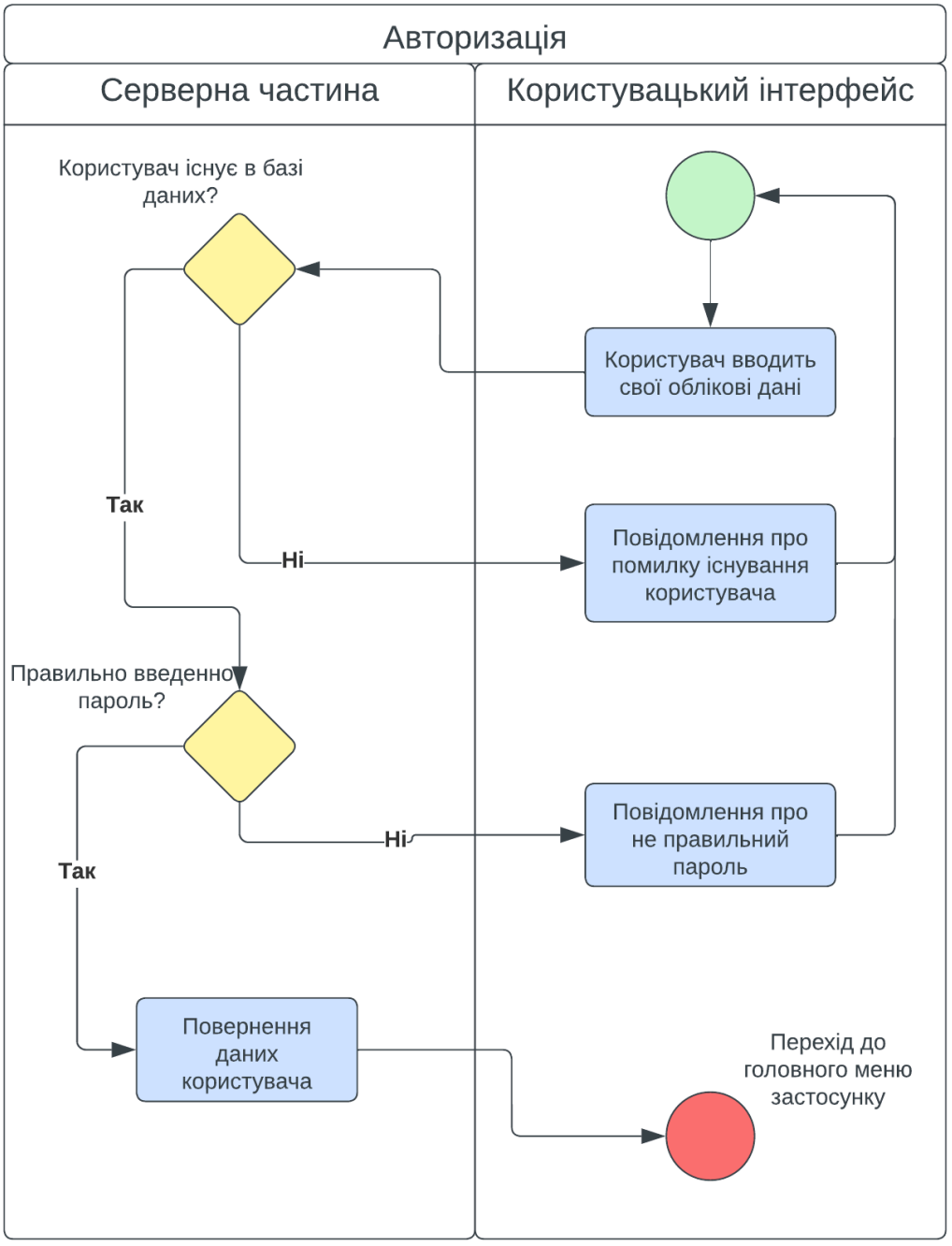


Рисунок 1.5 - Модель бізнес-процесу авторизації користувача

Опис процесу авторизації:

- Користувач відкриває додаток.
- Користувач вказує свої облікові дані (логін та пароль).
- Система передає введені дані на сервер для перевірки та підтвердження їхньої вірності.
- Сервер проводить перевірку логіну та паролю та надсилає клієнту загальну інформацію про користувача.
- Після успішної авторизації, клієнт переходить на головний екран додатку.

Рисунок 1.6 - Модель бізнес-процесу реєстрації користувача

Опис процесу реєстрації:

- Користувач відкриває додаток.
- Користувач вводить бажану пошту та пароль, які буде використовувати для подальшої авторизації у додатку.

- Додаток передає ці дані у зашифрованому вигляді на сервер.
- Сервер реєструє нового користувача та зберігає його дані аутентифікації у базі даних.
- У випадку успішної реєстрації, сервер повертає повідомлення про успішне створення облікового запису користувача.
- Після успішної реєстрації, користувач автоматично переходить на головне меню додатку.

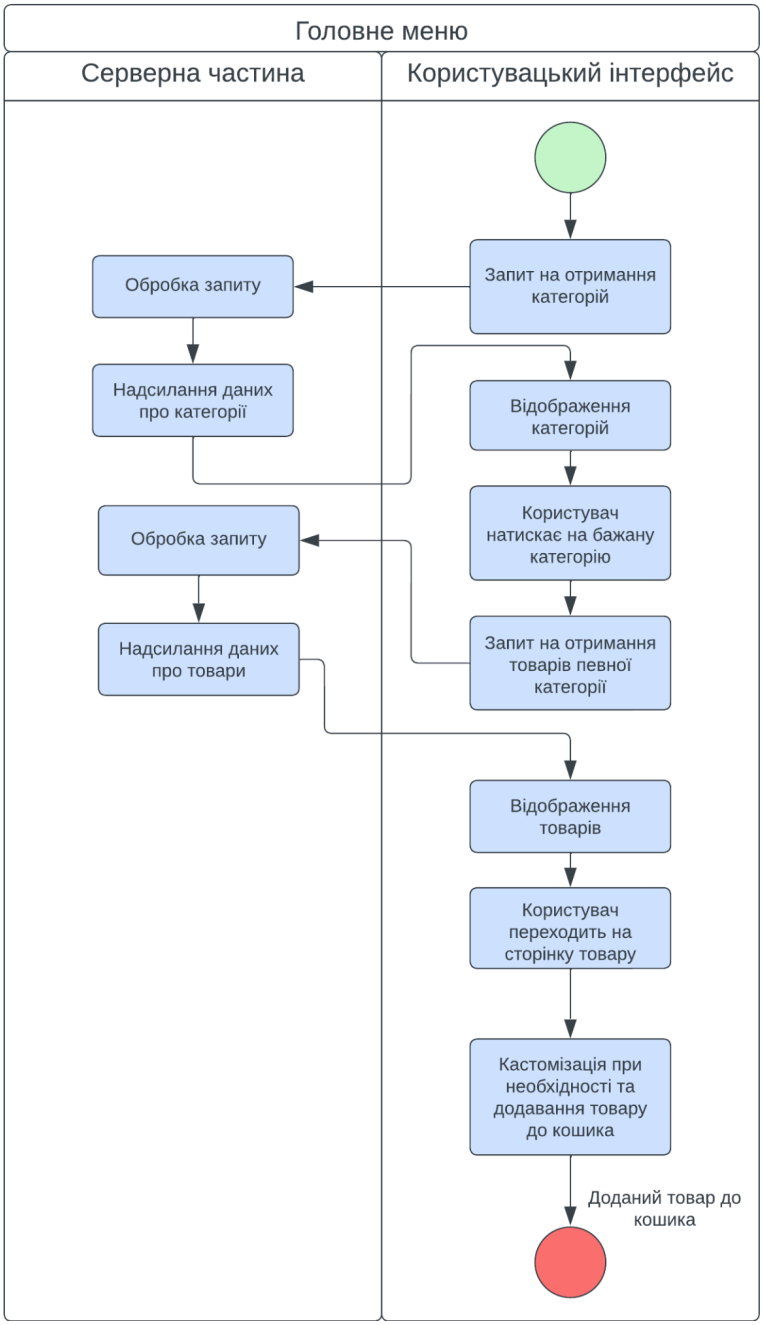


Рисунок 1.7 - Модель бізнес-процесу взаємодії клієнта з головним екраном застосунку

Опис бізнес-процесу відображення головного екрану застосунку:

- Користувач переглядає список категорій на головному екрані додатку.
- Користувач обирає конкретну категорії для огляду.
- Користувацький інтерфейс формує запит до серверної частини для отримання деталей про товари певної категорії.
- Клієнт переглядає товари, після чого товар і переходить на сторінку товару та має можливість змінювати параметри кастомізації.
- Після внесення змін користувач бачить зміну ціни та має можливість додати товар до свого кошика за допомогою відповідної кнопки

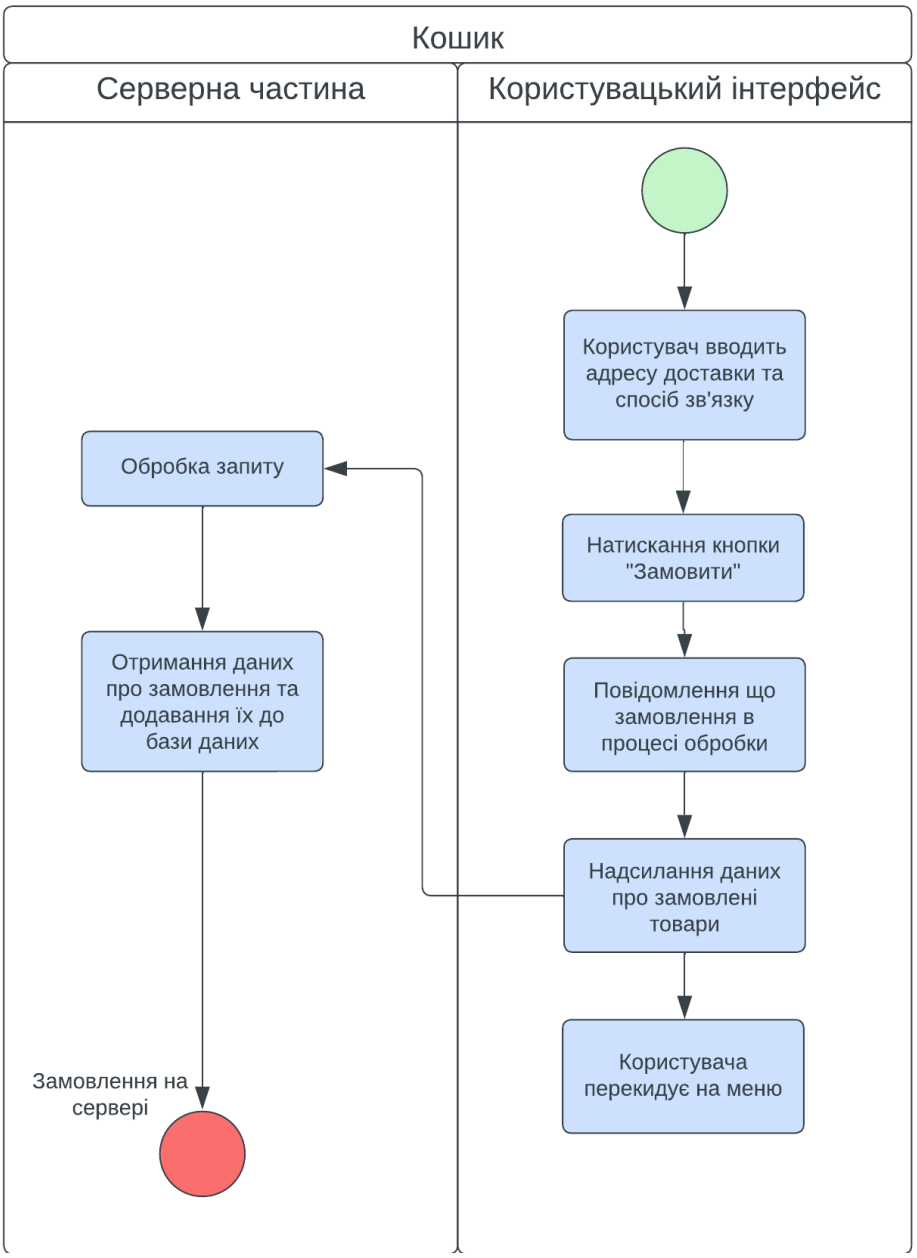


Рисунок 1.8 - Модель бізнес-процесу взаємодії клієнта з кошиком

Опис бізнес-процесу кошика:

- Клієнт переглядає товари, які додав до кошика
- Натискає кнопку замовити
- Додаток передає дані введені на сторінці кошика до серверу
- Сервер зберігає дані замовлення у базі даних.

Рисунок 1.9 - Модель бізнес-процесу взаємодії клієнта зі сторінкою замовлень

Опис бізнес-процесу отримання замовлень:

- Клієнт відкриває замовлення у додатку
- Система передає запит на сервер для отримання даних
- При наявності замовлень за авторизованим користувачем сервер передає дані до додатку

- Користувач бачить своє замовлення з відповідним статусом

Отже, основні зміни в моделі бізнес-процесів для мобільного додатку:

- Впровадження системи реєстрації та входу в додаток дозволяє створювати персоналізований досвід для користувачів, зберігаючи їхні дані, переваги та історію покупок.
- Організація головного меню з можливістю перегляду категорій та товарів покращує навігацію та спрощує процес вибору товарів.
- Впровадження профілю з інформацією про користувача та статусом замовлень підвищує довіру та зручність використання додатку.

Перспективи використання розробки в майбутньому:

- Інтеграція платіжних систем: У майбутньому можливе додавання функціоналу для внутрішніх платежів, що зробить процес покупки ще більш зручним та автоматизованим.
- Розширення функціоналу: Постійне оновлення та додавання нових функцій, таких як рекомендаційні системи та персоналізовані пропозиції, можуть значно підвищити залученість користувачів.
- Аналітика та маркетинг: Збір та аналіз даних про поведінку користувачів дозволить “Funkeys” оптимізувати маркетингові стратегії та підвищити ефективність рекламних кампаній.
- Масштабування: Завдяки використанню масштабованих хмарних рішень, “Funkeys” зможе легко адаптуватися до зростаючого числа користувачів та збільшення обсягу даних.

Концептуальна модель системи мобільного додатку “Funkeys” створює міцну основу для розроблення додатку, який відповідатиме потребам сучасного ринку та забезпечить високий рівень задоволеності клієнтів.

1.7. Економічний ефект від впровадження мобільного додатку

Визначення економічного ефекту від впровадження мобільного додатку є дуже важливим моментом, адже в його основі лежить техніко-економічне обґрунтування розробки автоматизованої системи.

Джерелами прибутку від впровадження додатку для "Funkeys" можуть бути такі фактори:

- Збільшення кількості замовлень,
- Автоматизація процесів продажу
- Додаткові послуги
- Зменшення витрат на обслуговування замовлень
- Зменшення часу комунікування із клієнтом

Визначаємо ознаку – продаж ігрових девайсів.

Ступінь новизни розроблюваних задач — "В" — використання типових проектних рішень за умови їх змін, розробка проектів, що мають аналогічні рішення. Група складності алгоритму — 2.

Узагальнені дані вхідної та вихідної інформації для мобільного додатку за видами вхідної та вихідної інформації таблиці 1.2.

Таблиця 3.2. Узагальнені дані для вхідної та вихідної інформації

Вид інформації	**Позначення**	**К-сть наборів даних**
Змінна інформація	ЗІ	m=4
Нормативно – довідкова інформація	НДІ	n=2
Банк(база) даних	БД	p=1
Обробка в режимі реального часу	РЧ	Так

Визначаються витрати часу на розроблення макету проекту (предметне дослідження) Т1 та технічного завдання Т2 за такими даними:
Макет проекту, Т1 = 38. Технічне завдання, Т2 = 20.

Визначимо витрати часу на стадіях «технічний проект», «рабочий проект» і «впровадження».

Вхідними даними для визначення є:

- кількість форм вхідної інформації 5;

- кількість форм вихідної інформації 3;
- базове значення витрат часу для стадії «Технічний проект» ТБ3 = 70
- базове значення витрат часу для стадії «Робочий проект» ТБ4 = 110
- базове значення витрат часу для стадії «Впровадження» ТБ5 = 45

Визначення затрат часу для стадії «Технічний проект» (Т3)

$k_П$ – коефіцієнт трудомісткості робіт на стадії «Технічний проект» розраховується за формулою 1.1.

$$T_З = T_{Б3} * k_П * k_0 \quad (1.1)$$

$$k_П = (k1*m+k2*n+k3*p)/(m+n+p) \quad (1.2)$$

Таблиця 3.3. Коефіцієнти ступеню новизни проекту (k_O)

Стадія розробки системи	Вид обробки	Ступінь новизни
		В
Технічний проект	РЧ	1.26
Робочий проект	РЧ	1.32
Впровадження	РЧ	1.21

На стадії «Технічний проект», коефіцієнти k1, k2, та k3 встановлені таким чином:

- k1 (для змінної інформації, ЗІ) — 1.0;
- k2 (для нормативно-довідкової інформації, НДІ) — 0.72;
- k3 (для банку даних, БД) — 2.08.

$k_П = (1*5+0.72*3+2.08*1) / (5+3+1) = 9.24/9 = 1.027$

$T_З = 70 * 1.027 * 1.26 = 90.58$

На стадії «Робочий проект», коефіцієнти k1, k2, та k3 встановлені таким чином:

- k1 (для змінної інформації, ЗІ) — 1.1;
- k2 (для нормативно-довідкової інформації, НДІ) — 0.58;
- k3 (для банку даних, БД) — 0.48.

Визначення витрат часу на стадії «Робочий проект» (Т4) розраховується за формулою 1.3.

$$T_4 = T_{Б4} * k_П * k_0 * k_C \quad (1.3)$$

$k_П$ – коефіцієнт трудомісткості робіт на стадії «Робочий проект» розраховується за формулою 1.2.

Для знаходження для формули необхідно ідентифікувати складність контролю вхідної та вихідної інформації. Тобто $k_C = 1.0$

$k_П = (1.1*5+0.68*3+0.48*1) / (5+3+1) = 8.02/9 = 0.88$

$T_4 = 110 * 0.88 * 1.32 * 1.0 = 90.58 = 127.78$

Визначення витрат часу на стадії «Впровадження» (Т5) розраховується за формулою 1.4

$$T_5 = T_{Б5} * k_П * k_0 * k_C \quad (1.4)$$

$T_4 = 45 * 0.88 * 1.21 * 1.0 = 47.92$

Отже, загальних затрати часу на розроблення проекту (складають за формулою (1.5):

$$T_Σ = T_1 + T_2 + T_3 + T_4 + T_5 \quad (1.5)$$

$T_Σ = 38 + 20 + 90.58 + 127.78 + 47.92 = 324.28$ годин

Визначення чисельність виконавців (Ч) за формулою (1.6).

$$Ч = T∑Ф \quad (1.6)$$

На розробку проекту виділено Ф, днів: 80;

Кількість місяці на розробку (М): 3;

Отже, для виконання такого проекту потрібно така чисельність виконавців Ч, яка обраховується за формулою: Ч = 324/80 = 4 виконавців.

Приймемо розмір заробітної плати програміста - 25000 грн, тоді загальна сума заробітних плат програмістів складає:

$$V_1 = Ч * М * ЗП = 4 * 3 * 25000 = 300000 \text{ грн} \quad (1.7)$$

Витрати на підготовку ПК та серверу під час розробки V2 = 0

Витрати на підготовку приміщення V3 = 0 грн

В середньому навчання персоналу праці з триватиме 1 місяць , тому можна вважати, що V4 = 20000 грн.

Загальна вартість розробки і впровадження системи вираховується за формулою (1.8):

$$V\sum = V1+V2+V3+V4 = 300000 + 0 + 0 + 20000 = 320000 \text{ грн} \quad (1.8)$$

Оскільки норма амортизаційних втрат для комп'ютерних систем НА = 5, то для обрахування річного економічного ефекту слід брати до розгляду величину (формула 1.9):

$$Vp = V\sum НА \quad (1.9)$$

$$Vp = 320000 / 5 = 64000 \text{ грн}$$

Термін окупності розробки визначається Ток = 1 / Кеф, де Кеф = Пр/Vp, де річний прибуток Пр від впровадження системи буде досягнуто за рахунок збільшення продаж і, орієнтовно, складатиме 50 000 грн на рік.

$$Кеф = Пр/Vp = 50000/64000 = 0.78125$$

Отже, термін окупності додатку розраховується за формулою (1.10):

$$Ток = 1 / 0.78125 = 1.28 \text{ роки} \quad (1.10)$$

Мобільний додаток, який окуповується всього за 1.28 року, є гарною інвестицією. Це свідчить про високу ефективність та потенціал зростання, які він пропонує "Funkeys". З таким коротким терміном окупності, додаток не лише швидко повертає вкладені кошти, але й продовжує генерувати прибуток, перевищуючи первісні інвестиції. Завдяки підвищенню продуктивності, автоматизації продажів, додатковим послугам, зменшенню витрат на обслуговування та оптимізації часу спілкування з клієнтами, мобільний додаток стає ключовим активом у стратегії розвитку "Funkeys". Він не тільки спрощує процеси та підвищує задоволеність клієнтів, але й відкриває нові можливості для розширення бізнесу.

У ході системного аналізу сайту "Funkeys" було встановлено, що компанія є цифровим лідером у сфері продажу ігрових пристроїв. Завдяки чіткій організаційній структурі, кожен підрозділ компанії ефективно виконує свої функції, сприяючи досягненню загальної мети. Аналіз поточного стану комп'ютеризації виявив основні технологічні платформи та інструменти, які використовуються в компанії, що дозволяє зрозуміти її технічну оснащеність.

Детальний аналіз бізнес-процесів та розробка функціональної моделі допомогли виявити шляхи оптимізації роботи “Funkeys”, підвищуючи її продуктивність. Це стало основою для обґрунтування необхідності розробки мобільного додатку, який забезпечить більшу зручність для користувачів та розширить можливості продажу.

Концептуальна модель системи, яка була розроблена, включає ключові компоненти та їх взаємодію, що є важливим для забезпечення функціональності мобільного додатку. Оцінка економічного ефекту від впровадження додатку показала, що він має значний потенціал для збільшення доходів компанії через зростання продажів та поліпшення клієнтського досвіду.

Таким чином, проведений системний аналіз підтвердив, що розробка мобільного додатку “Funkeys” є доцільною та необхідною для того, щоб компанія могла успішно конкурувати на ринку ігрових пристроїв та задовольняти потреби своїх клієнтів.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1. Найменування проекту та галузь розробки

Назва проекту: «Мобільний додаток для продажу ігрових девайсів».

Галузь застосування: Наведене технічне завдання поширюється на розробку програмного застосунку для платформи iOS, котрий використовується для продажу ігрових пристроїв.

Підстава для розробки

Підставою для розробки мобільного додатку для продажу ігрових девайсів є необхідність розвитку фірми для збільшення аудиторії, клієнтів та підвищення обсягів продажу. Створення мобільного додатку сприятиме покращенню взаємодії з клієнтами, наданню зручного доступу до товарів та послуг, а також зміцненню позицій фірми на ринку.

2.2. Ціль та призначення розробки

2.2.1. Призначення розробки

Метою розробки мобільного додатку є вирішення проблем та покращення функціональності фірми “Funkeys”. Даний додаток буде сприяти зручності для клієнтів та забезпечить більшу взаємодію між магазином та користувачами.

2.2.1. Ціль створення додатку

Застосунок призначений для підвищення продажів ігрових девайсів через зручність та швидкість здійснення покупок, покращення користувацького досвіду за допомогою інтуїтивного інтерфейсу та рекомендацій, залучення нових клієнтів, яка активно користується мобільними пристроями.

2.3. Вимоги до програмного забезпечення

2.3.1 Вимоги до програмного забезпечення в цілому

Додаток повинен забезпечувати виконання наступних функцій:

2.3.2. Інтерфейсу користувача:

- Реєстраційна форма, що дозволяє внести інформацію про користувача, включаючи ім’я, електронну адресу, пароль та інші дані;

- Форма входу в систему, який вимагає введення логіна користувача для ідентифікації;
- Додаток має мати простий та зрозумілий інтерфейс, який дозволяє користувачам легко орієнтуватись та знаходити потрібні ігрові пристрої;
- Додаток має мати можливість для перегляду замовлень;
- Додаток має можливість кастомізувати товари в залежності від їх типу;
- Додаток має можливість пошуку за назвою.

2.3.3. Для користувача:

- Простий процес реєстрації та входу в додаток;
- Перегляд історії купівель;
- Можливість переглядати товар;
- Можливість обрати перемикачі, наявність гравіювання та інші послуги на сторінці з товаром;
- Функціональний кошик, який дозволяє користувачам зберігати товари та здійснювати покупки;
- Перевірка стану замовлення
- Інтерфейс для введення пошуку;
- Відображення результатів пошуку.

2.3.4. Вимоги до надійності

Передбачити контроль введення інформації та захист від некоректних дій користувача. Забезпечити цілісність інформації котра знаходиться в базі даних та валідацію даних.

2.3.5. Умови експлуатації

Умови експлуатації згідно ДСанПІН 3.3.2.007-98. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин

2.3.6. Вид Обслуговування

Вимоги до виду обслуговування не висуваються.

2.3.7. Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не висуваються.

2.3.8. Вимогу до складу і параметрів технічних засобів

Програмне забезпечення повинно функціонувати на мобільних пристроях з операційною системою iOS.

Мінімальна конфігурація технічних засобів:

- Apple A11 Bionic;
- Об’єм ОЗП: 2 Гб;
- Підключення до мережі Інтернет зі швидкістю від 20 мегабіт;
- Операційна система iOS версії 16 та вище.

Рекомендована конфігурація технічних засобів :

- Apple A13 Bionic;
- Об’єм ОЗП: 4 Гб;
- Підключення до мережі Інтернет зі швидкістю від 50 мегабіт;

2.3.9. Вимоги до інформаційної та програмної сумісності

Програмне забезпечення повинно працювати під управлінням операційних систем iOS версії 16.0 та більше. Проведення розробки повинне бути виконане на платформі macOS Sonoma

2.3.10. Вимоги до вхідної та вихідної інформації

Таблиця 2.1. Вхідна та вихідна інформація

Функції системи	**Вхідна інформація**	**Вихідна інформація**
Реєстрація користувача	Реєстраційні дані (ім'я, прізвище, електронна пошта, пароль)	Підтвердження реєстрації, персональний профіль користувача
Обробка замовлень	Товари у кошику, метод оплати	Повідомлення про успішне замовлення, відображення наявності замовлення у профілі

Продовження таблиці 2.1.

Функції системи	**Вхідна інформація**	**Вихідна інформація**
Реєстрація користувача	Реєстраційні дані (ім'я, прізвище, електронна пошта, пароль)	Підтвердження реєстрації, персональний профіль користувача
Обробка замовлень	Товари у кошику, метод оплати	Повідомлення про успішне замовлення, відображення наявності замовлення у профілі
Обмін інформацією з клієнтами	Сповіщення статус замовлення	Повідомлення про зміни в статусі замовлення
Пошук серед товарів	Запит користувача на пошук	Форма з результатами пошуку
Кастомізації продуктів	Вибір перемикачів, гравіювання та змащення клавіатур.	Товар зі змінними характеристиками та ціною на сторінці товару та у кошику

2.3.11. Вимоги до мови розробки

Розробку виконати на мові програмування Swift [7], з використанням фреймворку SwiftUI [10] для створення інтерфейсу користувача

2.3.12. Вимоги до середовища розробки

Розробку виконати за допомогою середовища Xcode [6]

2.3.13. Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

2.3.14. Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

2.3.15. Спеціальні вимоги

Спеціальні вимоги не висуваються

2.4. Вимоги до програмної документації

2.4.1. Попередній склад програмної документації

У склад супроводжувальної документації повинні входити наступні документи на аркушах формату А4:

- Пояснювальна записка;
- Технічне завдання;
- Текст програми;
- Програма;
- Керівництво користувача.

Графічна частина повинна бути виконана на аркушах формату А4 та містити наступні документи:

- Схема бази даних;

2.4.2. Спеціальні вимоги до програмної документації

Програмні модулі, котрі розробляються, повинні бути задокументовані (тексти програм повинні містити всі необхідні коментарі).

2.5. Стадії і етапи розробки

Таблиця 2.2 - Стадії та етапи розробки

№	Найменування робіт	Строки виконання робіт
1	Вивчення літератури за тематикою проекту	24.03.2024
2	Розробка технічного завдання	14.04.2024
3	Аналіз вимог та уточнення специфікацій	16.04.2024
4	Проектування структури програмного забезпечення	20.04.2024
5	Програмна реалізація програмного	04.05.2024

	забезпечення	
6	Тестування розробленого проекту	06.05.2024
7	Розробка матеріалів текстової частини проекту	10.05.2024
8	Розробка матеріалів графічної частини проекту	12.05.2024
9	Оформлення технічної документації	17.05.2024

2.6. Порядок контролю та приймання

2.6.1. Загальні положення

- Контроль якості продукту здійснюється на всіх етапах розробки;
- Приймання робіт виконується замовником за результатом перевірки відповідності функціональних та технічних вимог.

2.6.2. Етапи контролю

- Забезпечення повноти та правильності технічної документації;
- Перевірка коду на відповідність вимогам розробки та виявлення потенційних помилок;
- Проведення функціонального тестування в реальних умовах для виявлення можливих проблем та оптимізації функції.

2.6.3. Приймання роботи

- Підтвердження відповідності програмного забезпечення встановленим критеріям якості;
- Підтвердження замовником готовності продукту до експлуатації.

2.7. Джерела розробки

2.7.1 При розробленні технічного завдання на систему використано наступні документи:

ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;

ДСТУ ISO/IEC 25010:2016 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання.

РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1. Інформаційне забезпечення системи

Для створення додатку продажу ігрових девайсів для цифрової фірми “Funkeys” на основі процесів моделі (наведених у Додатку А) була розроблена модель структури даних (Додаток Б) за допомогою Lucidchart [2].

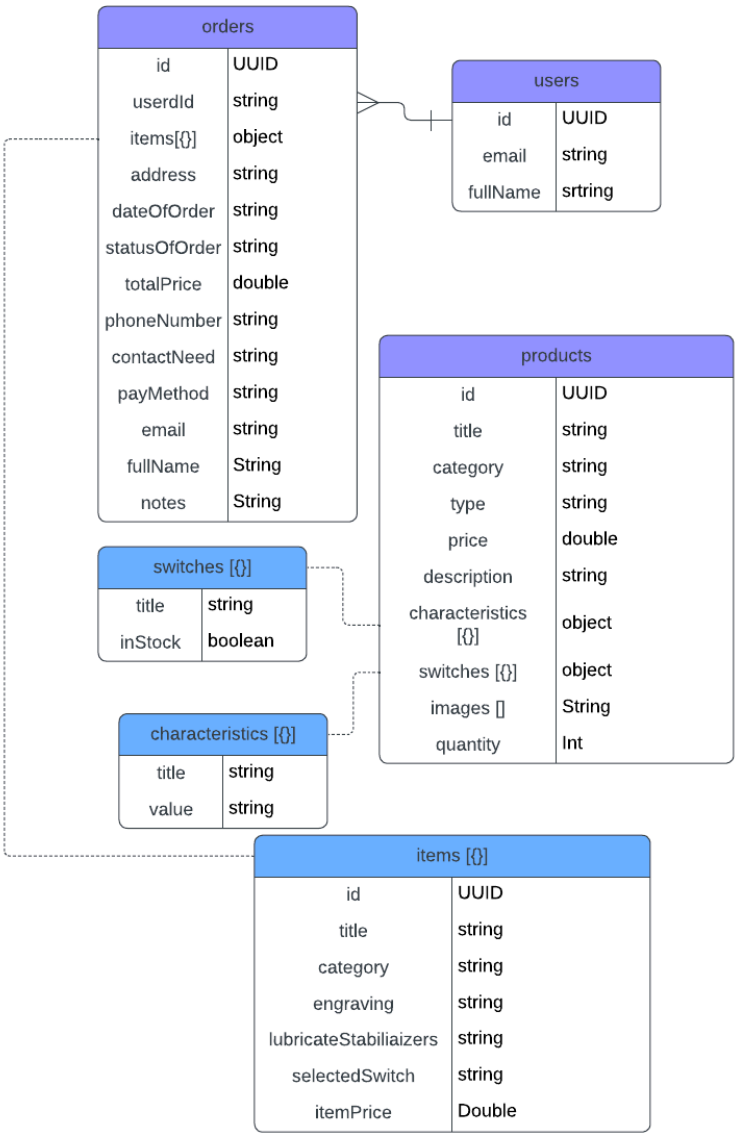


Рисунок 3.1 - Структура бази даних

База даних, створена для мобільного додатку, базується на даних, що вже використовуються на веб сайті фірми, але з деякими покращеннями для забезпечення кращої інтеграції та продуктивності в мобільному середовищі.

Нова база даних була адаптована для оптимізації роботи в мобільному додатку, включаючи можливість збереження інформації про користувачів і замовлення, що забезпечує більш персоналізований та ефективний користувацький досвід.

Опис користувацьких сутностей бази даних наведено у таблицях 3.1-2.4:

Таблиця 3.1. Опис таблиці "users"

Назва таблиці	users
Призначення	Відповідає за збереження та відображення даних користувача, також використовується для замовлень
Опис атрибутів таблиці	
id	Відповідає за збереження унікального ідентифікатора користувача
email	Відповідає за збереження поштової адреси користувача при реєстрації
fullName	Відповідає за збереження Ім'я та прізвище користувача при реєстрації

Таблиця 3.2. Опис таблиці "products"

Назва таблиці	products
Призначення	Відповідає за збереження даних про товари, їх тип, категорію, ціну, характеристики та інше
Опис атрибутів таблиці	

Продовження таблиці 3.2.

id	Відповідає за збереження унікального ідентифікатора товару
title	Відповідає за збереження назви товару
category	Категорія товару
type	Тип товару
price	Ціна товару
description	Опис товару
characteristics	Масив даних, що містить в собі характеристики товару
switches	Масив даних, що містить в собі перемикачі клавіатур.
images	Масив посилань на картинки товару, що взято з API “Funkeys”
quantity	Кількість наявного товару

Таблиця 3.3. Опис таблиці “orders”

Назва таблиці	orders
Призначення	Відповідає за збереження замовлення товарів зроблене у кошику
Опис атрибутів таблиці	

Продовження таблиці 3.3.

id	Відповідає за збереження унікального ідентифікатора замовлення
userId	Унікальний ідентифікатор користувача котрий зробив замовлення
items	Масив унікальних товарі котрі були замовлені
address	Введена адреса користувача для замовлення
dateOfOrder	Дата коли було зроблене замовлення
statusOfOrder	Статус замовлення для відстежування його у додатку
totalPrice	Загальна ціна замовлення
phoneNumber	Номер телефону замовника
fullName	Ім’я та Прізвище на яке робиться замовлення
email	Поштова скринька
contactNeed	Варіант зв’язку з клієнтом
payMethod	Спосіб оплати
notes	Примітки до замовлення

Таблиця 3.4. Опис таблиці “items”

Назва таблиці	items
Призначення	Відповідає за збереження даних про замовлені товари та їх унікальні параметри
Опис атрибутів таблиці	
id	Відповідає за збереження унікального ідентифікатора товару
title	Відповідає за збереження назви товару
category	Категорія товару
engraving	Відповідає за необхідність гравіювання клавіатури або кейкапів
lubricateStabilizators	Відповідає за необхідність змащування клавіатури
selectedSwitch	Відповідає за обрані перемикачі на клавіатурі
itemsPrice	Відповідає за ціну котра вийшла в залежності від кастомізованих параметрів та знижки на момент замовлення

3.2. Алгоритмізація та реалізація комплексу задач автоматизації

Розроблений додаток працює на основі клієнт-серверної моделі. У цій моделі, клієнт (що використовує мобільний додаток) та сервер взаємодіють

через інтернет, відповідно клієнт надсилає запити, а сервер відповідає на ці запити надаючи запитувані дані.

У цьому проекті мобільний додаток забезпечує інтерфейс для користувача та відображення інформації, тоді як серверна частина відповідає за збереження, та надання даних котрі необхідні клієнту

Користувацький інтерфейс створено згідно архітектури MVVM (Model-View-ViewModel), який включає наступне:

View (Вид): Цей компонент інтерфейсу користувача є тим, що користувачі сприймають візуально та з чим вони безпосередньо взаємодіють, включаючи екрани, форми та інші елементи UI.

ViewModel (Модель-Вид): Виконує роль медіатора між View та Model, керуючи взаємодією користувача з програмою. Він обробляє вхідні дані від View, здійснює необхідні операції з даними та визначає логіку реагування на дії користувача, такі як натискання кнопок.

Model (Модель): Є ядром програми, де зосереджена основна бізнес-логіка та обробка даних. Він відповідає за зберігання, вилучення, обробку даних та інші операції, що забезпечують функціонування програми відповідно до заданих вимог. Шаблон архітектури Swift MVVM наведений на рисунку 3.2.

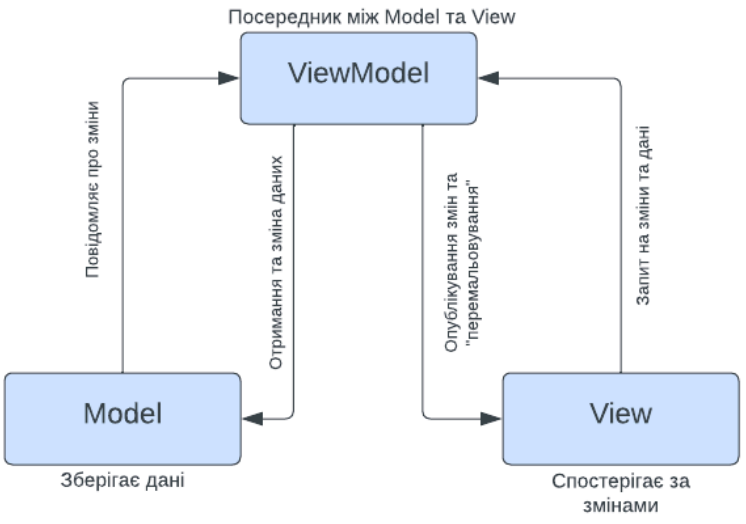


Рисунок 3.2 - Архітектура MVVM

Файлова структура (рисунок 3.3) мобільного додатку була організована з використанням директорій для кращої структури та орієнтування по проєкту, також зроблені окремо компоненти, що часто використовуються для зменшення кількості коду та для кращого читання коду і його зменшення.

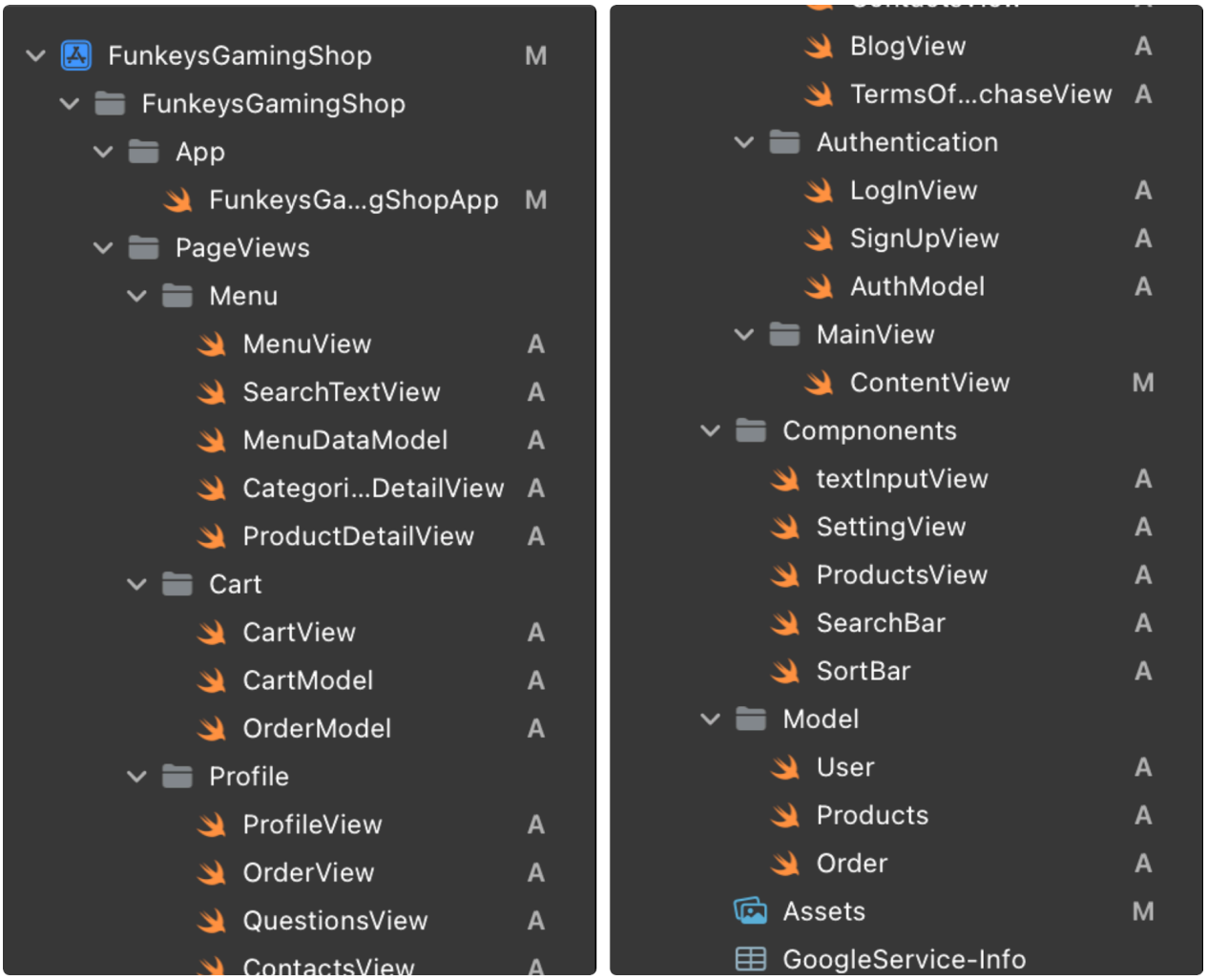


Рисунок 3.3 - Файлова структура мобільного додатку

Зазначені нижче компоненти є основними складовими розробленого мобільного додатку:

PageViews є основною директорією яка містить відображення мобільного додатку, у ній знаходиться директорія MainView, а у ній ContentView. ContentView у SwiftUI [10] є основним файлом який визначає основну структуру та навігацію в проекті.

Рисунок 3.4 - Код “ContentView.swift”

У цьому файлі, якщо користувач успішно увійшов у систему, то відображається "TabView" з трьома вкладками (наведені у Додатку В):

- MenuView: Відображає меню додатку.
- CartView: Відображає кошик користувача і показує кількість елементів у кошику.
- ProfileView: Відображає профіль користувача.

Якщо користувач не авторизований, то він переходить до LogInView

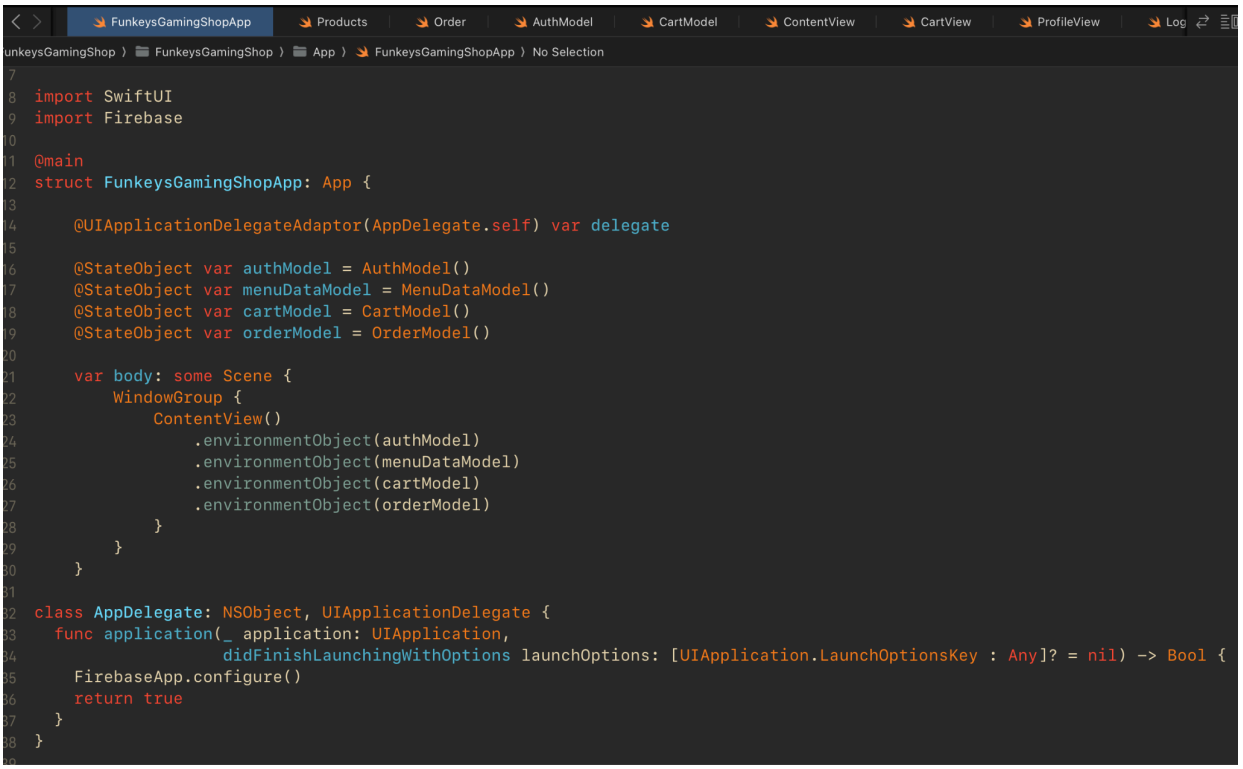


Рисунок 3.5 - Код файлу “FunkeysGamingShopApp.swift”

“FunkeysGamingShopApp.swift” є головним файлом проекту, який відповідає за запуск програми, та за запуск головного вікна при вході у додаток. Також у ньому ми підключаємо додаток до Firebase, у котрому використовуємо Firebase Authentication (для аутентифікації) та Firebase Firestore [8] (для збереження даних), для підключення використовуємо код котрий наданий у інструкції підключення Firebase до додатку розробленого на фреймворку SwiftUI (інструкція наведена у додатку В).

Після успішного входу в систему через ContentView, користувач отримує доступ до основних функцій додатку, таких як MenuView, CartView та ProfileView. Процес аутентифікації керується “AuthModel.swift”, який є ключовим компонентом у забезпеченні безпеки та персоналізації досвіду користувача.

“AuthModel” є центральним компонентом системи аутентифікації мобільного додатку, який відповідає за управління станом авторизації користувачів. Він використовує сервіси Firebase для реєстрації нових користувачів, входу та виходу існуючих користувачів, а також для отримання даних про поточного користувача.

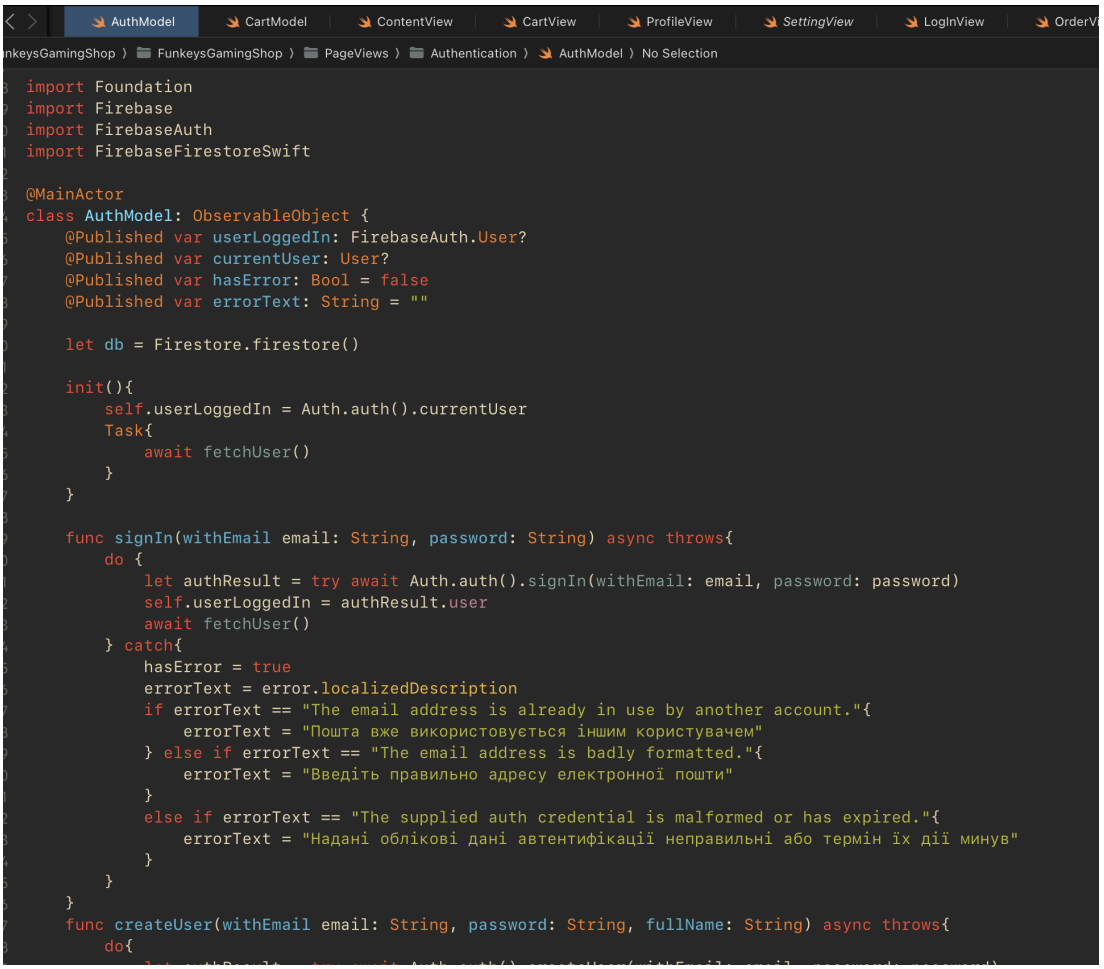


Рисунок 3.6 - Частина коду файлу AuthModel.swift

Функції “AuthModel”:

- signIn: Дозволяє користувачам входити в систему, використовуючи електронну пошту та пароль. У разі помилки, модель встановлює повідомлення про помилку, яке може бути відображене користувачу.
- createUser: Реєструє нового користувача з вказаним ім’ям, електронною поштою та паролем. Після створення облікового запису, дані користувача зберігаються в базі даних Firestore.
- signOut: Виконує вихід користувача з системи, очищаючи всі локальні дані про поточного користувача.
- fetchUser: Отримує дані про поточного користувача з бази даних Firestore, використовуючи унікальний ідентифікатор користувача.

“AuthModel” також включає обробку помилок, яка інформує користувачів про проблеми, що можуть виникнути під час аутентифікації, наприклад, якщо електронна адреса вже використовується або неправильно введена. “AuthModel”

не зберігає паролі локально, замість цього використовуючи токени для управління сесіями користувачів.

“OrderModel.swift” є ключовим компонентом у системі управління замовленнями мобільного додатку, який відповідає за створення та відстеження замовлень користувачів. Використовуючи Firebase Firestore як хмарну базу даних, ця модель дозволяє зберігати та отримувати інформацію про замовлення в реальному часі.

Рисунок 3.7 - Код файлу “OrderModel.swift”

Основні функції “OrderModel”:

- addOrder: Додає нове замовлення до бази даних, використовуючи інформацію про замовлення та ідентифікатор користувача. Після додавання замовлення, модель автоматично оновлює список замовлень.

- fetchOrders: Отримує всі замовлення з бази даних, які належать конкретному користувачу. Це дозволяє користувачам переглядати історію своїх покупок та статус поточних замовлень.

“OrderModel” тісно інтегрований з “AuthModel”, оскільки він використовує ідентифікатор користувача для асоціації замовлень з їхніми власниками. Це забезпечує персоналізований досвід користувача та дозволяє забезпечити безпеку та конфіденційність інформації про замовлення.

3.3. Інструкція користувача

Якщо користувач не авторизований, то як тільки він входить у мобільний додаток, то потрапляє на форму авторизації

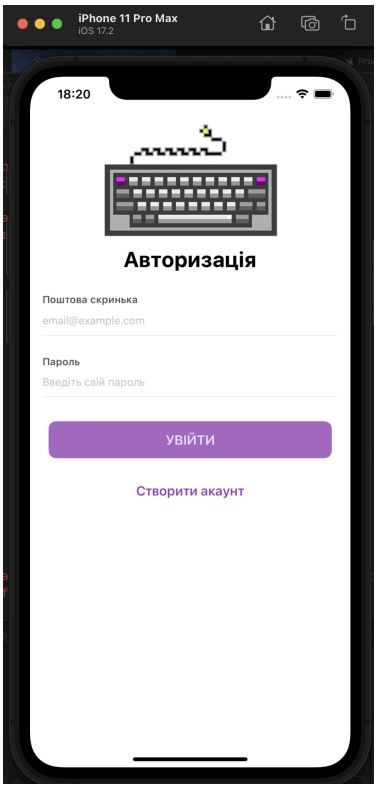


Рисунок 3.8 - Авторизація користувача

Якщо поля є пустими та не відповідають правилам валідації, то кнопка “Увійти” є неактивною. Якщо користувач введе неправильний пароль, або неправильно введе поштову скриньку, то з’явиться відповідна помилка

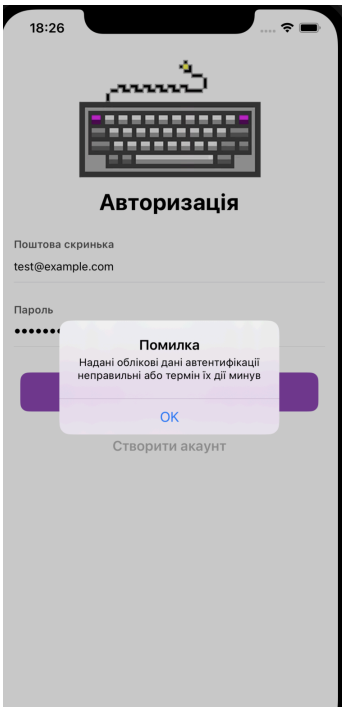


Рисунок 3.9 - Повідомлення про помилку під час авторизації

Також на формі авторизації є кнопка “Створити акаунт”, якщо у користувача ще немає акаунту.

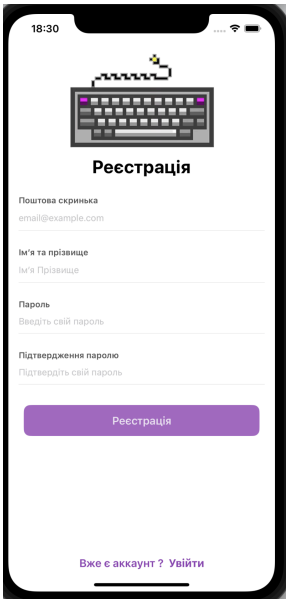


Рисунок 3.10 - Реєстрація користувача

Відповідно як і на попередній формі, якщо поля є пустими та не відповідають правилам валідації, то кнопка “Реєстрації” є неактивною. Якщо користувач введе неправильне підтвердження паролю, то на полі виводиться

про це повідомлення. Також, якщо користувач помилково натиснув кнопку реєстрації, То є відповідна кнопка для повернення до авторизації.

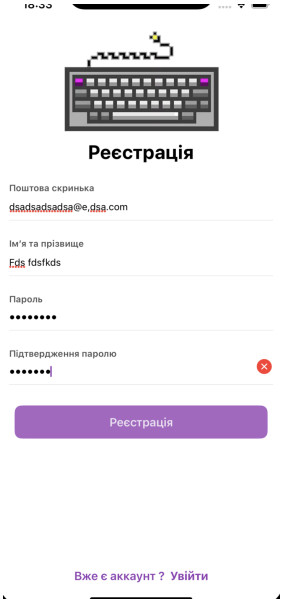


Рисунок 3.11 - Неправильно підтверджений пароль

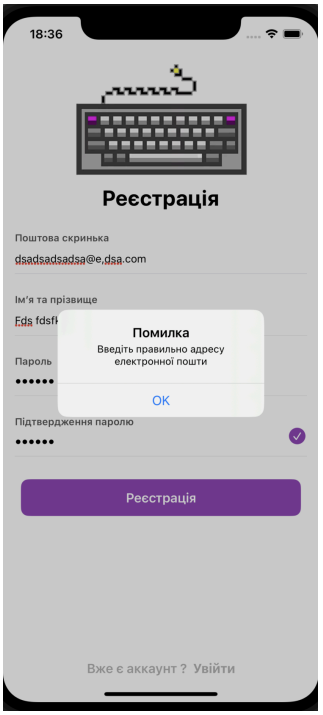


Рисунок 3.12 - Повідомлення про помилку під час реєстрації користувача

Після авторизації/реєстрації користувач потрапляє на головну сторінку меню де може подивитися на категорії товарів, пролистувати їх та натисканні на них. Також наявний пошук який знаходиться над категоріями

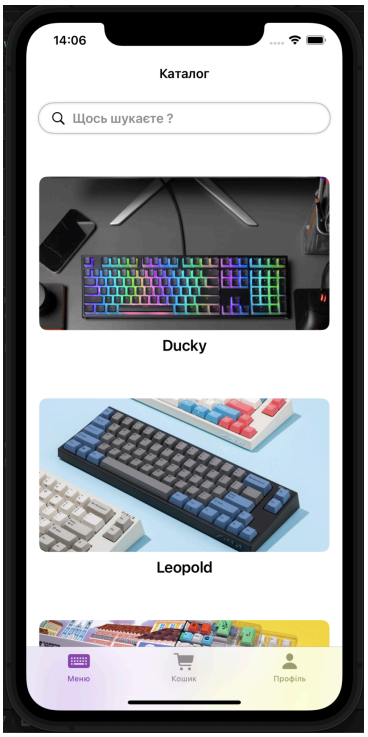


Рисунок 3.13 - Зовнішний вигляд головного меню застосунку

При натисканні на пошук, користувач переходить до введення запиту

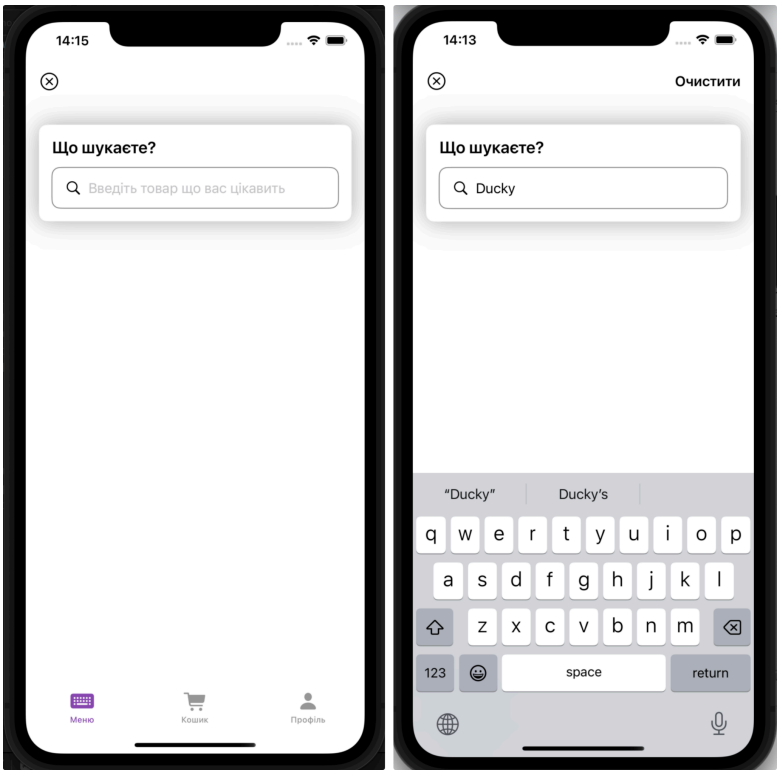


Рисунок 3.14 - Зовнішний вигляд пошуку

Після натискання “Введення”, користувач переходить на сторінку з виконаним пошуком, користувач бачить товари, видані за пошуком, їх першу картинку, та ціну. Також з’являється можливість сортування за ціною та назвою.

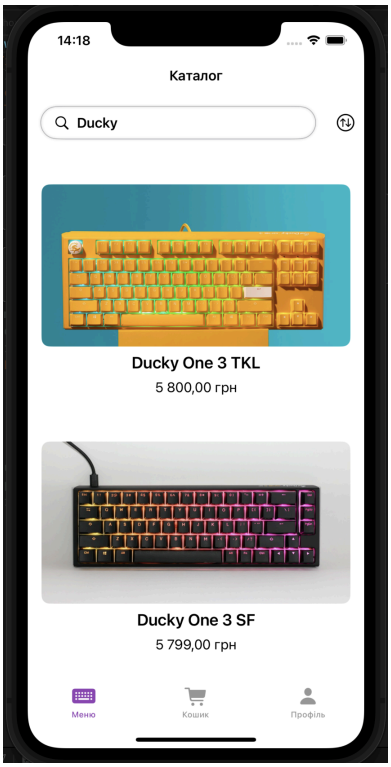


Рисунок 3.15 - Виконаний пошук по товарам

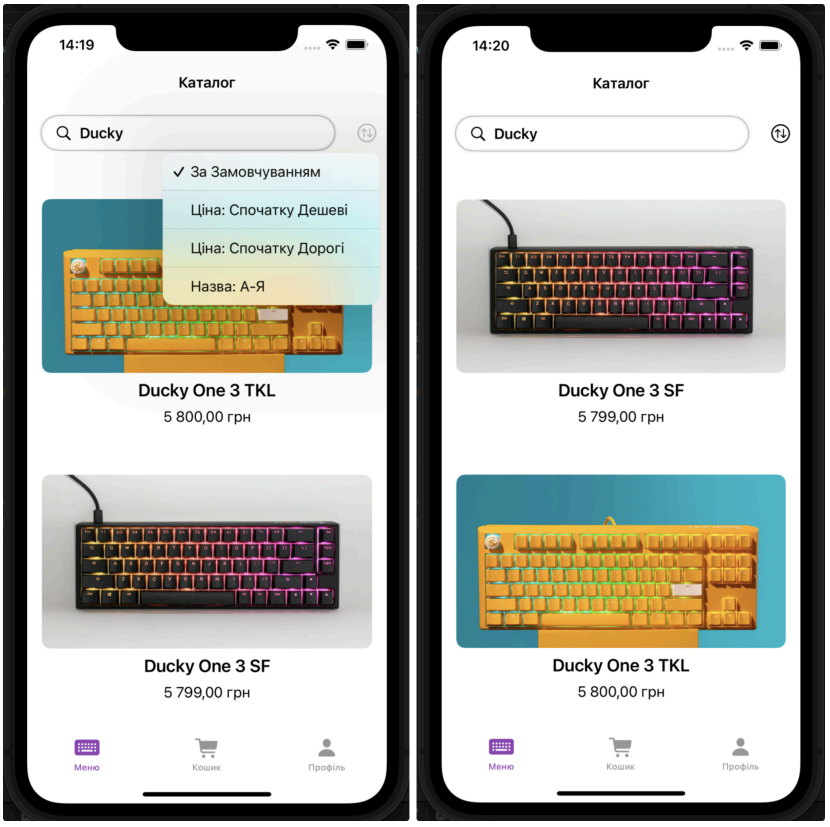


Рисунок 3.16 - Виконане сортування

Після натискання на відповідну категорію (до виконаного пошуку), користувач бачить товари цієї категорії, їх першу картинку, та ціну. Також зверху наявна можливість сортування.

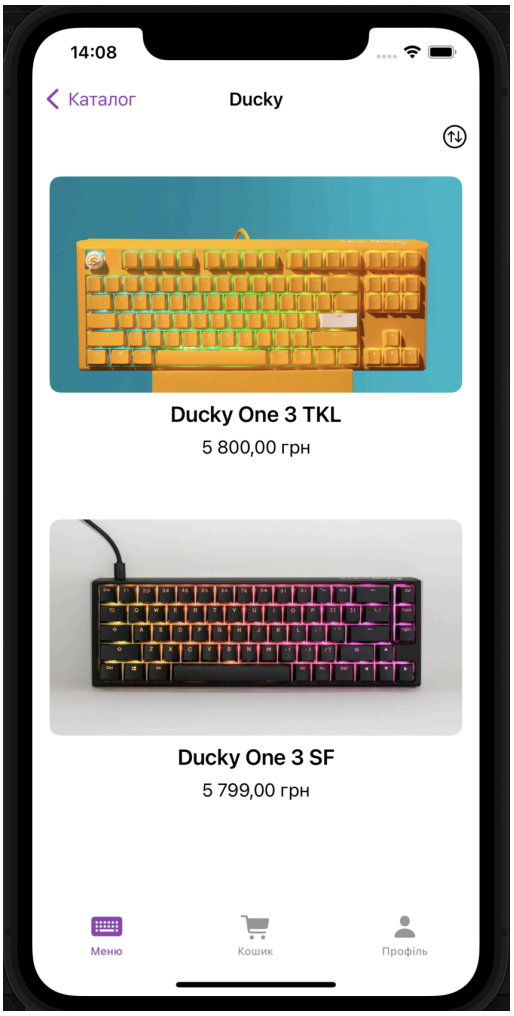


Рисунок 3.17 - Зовнішній вигляд після переходу до категорії товарів

Якщо користувач натискає на відповідний товар то потрапляє на сторінку товару, де можна побачити різні фото товару, кастомізований вибір (котрий є різним для кожного типу товару) , в залежності від якого змінюється ціна товару. Під кастомізацією товару є кнопка “Додати в корзину”. Під інтерфейсом з яким можна взаємодіяти знаходяться характеристики, та опис товару, які можна прочитати пролиснувши екран нижче.

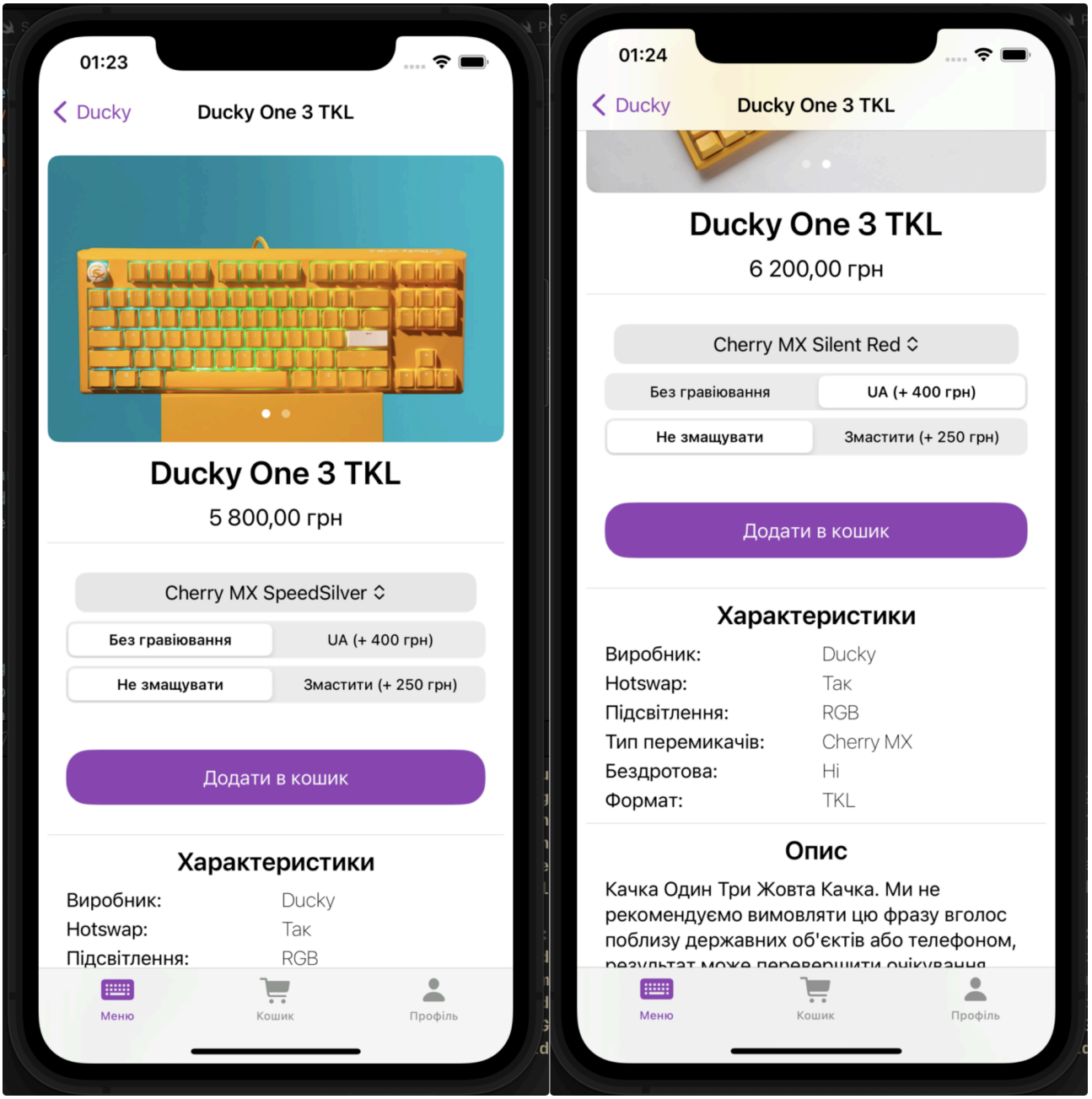


Рисунок 3.18 - Зовнішній вигляд сторінки клавіатур

Якщо товар не додано до кошику то внизу екрану (де знаходяться наші вкладки) не зображено над кошиком кількості товарів доданих до кошику, то відповідно кошик пустий.

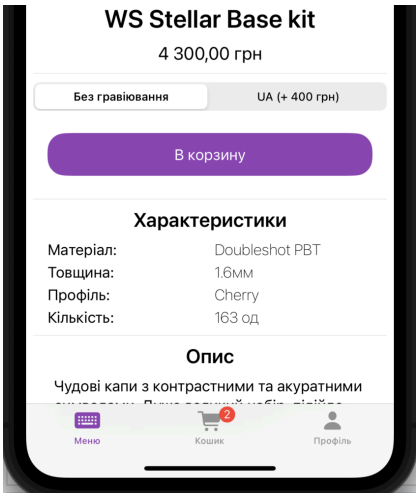


Рисунок 3.19 - Відображена кількість доданих товарів до кошику

На сторінці кошика ми можемо побачити додані товари, нижче від них ми можемо побачити параметри кастомізації котрі обрав користувач, справа від кожного товару є кнопка видалення. Під доданими товарами виводиться загальна сума замовлення.

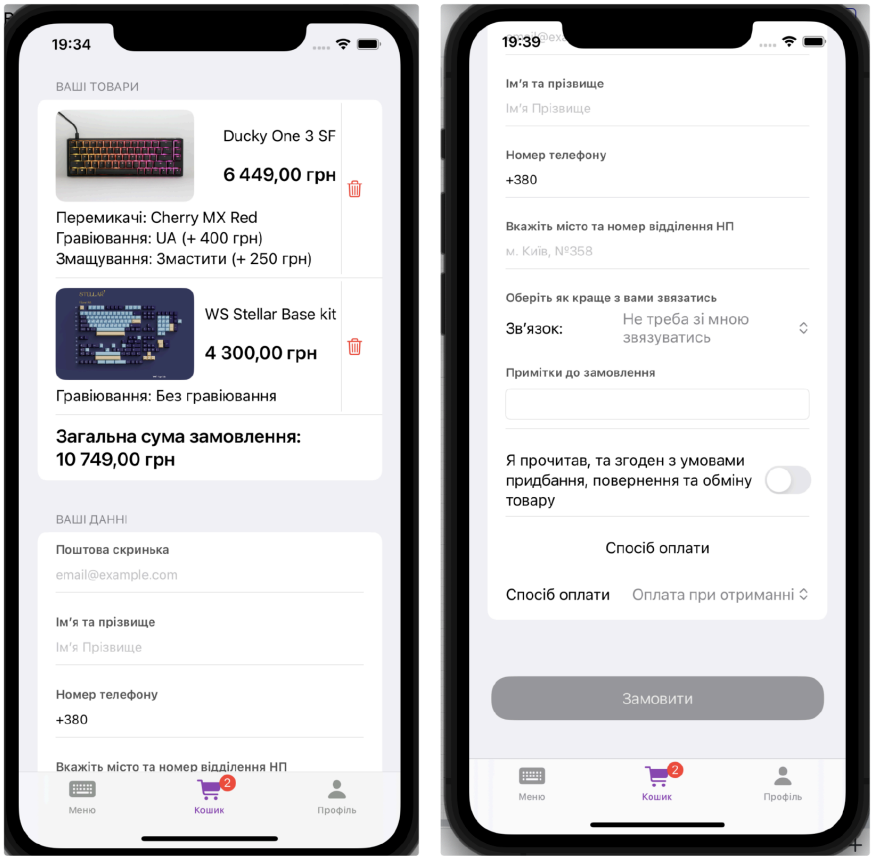


Рисунок 3.20 - Зовнішній вигляд кошику

Під доданими товарами до кошику користувач має заповнити всі поля (крім приміток) для оформлення замовлення. Якщо поля є пустими та не відповідають правилам валідації, то кнопка “Замовити” буде неактивною. Користувач може обрати спосіб зв’язку та обрати спосіб оплати. Після заповнених даних з’являється можливість замовити товар, після чого буде повідомлення що речі користувача готуються до відправки. Після натискання кнопки “Ок” наш кошик очищується.

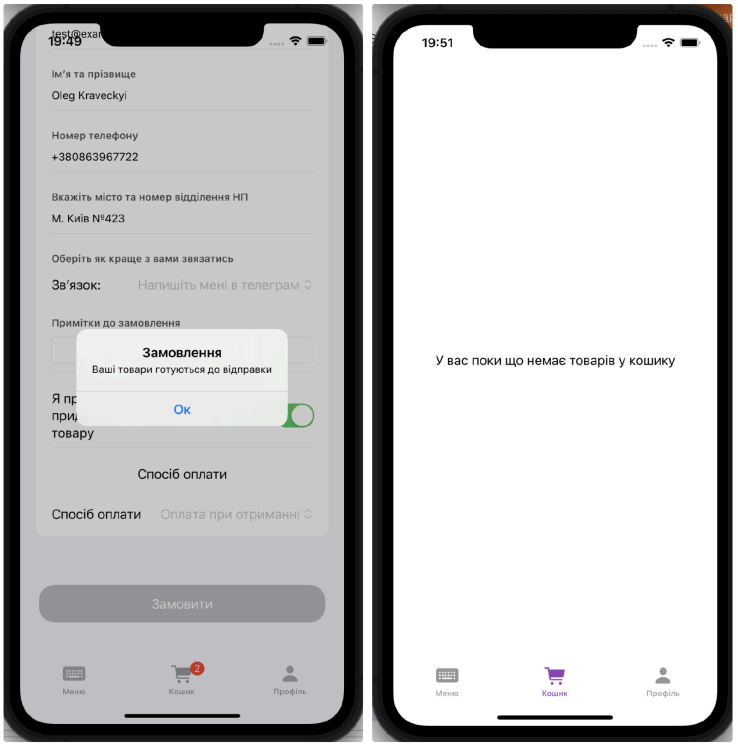


Рисунок 3.21 - Повідомлення про замовлення товару

На сторінці профілю користувач може вийти зі свого акаунту та подивитися свої замовлення

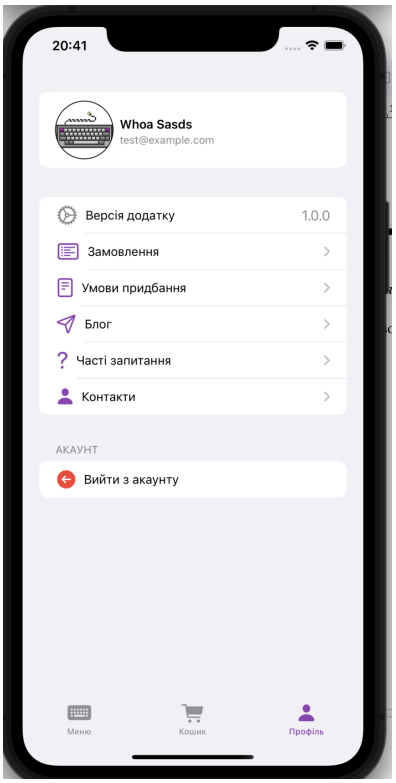


Рисунок 3.22 - Профіль користувача

При наявності замовлень, користувач може подивитися коли та що він замовив, та статус у якому знаходиться замовлення.

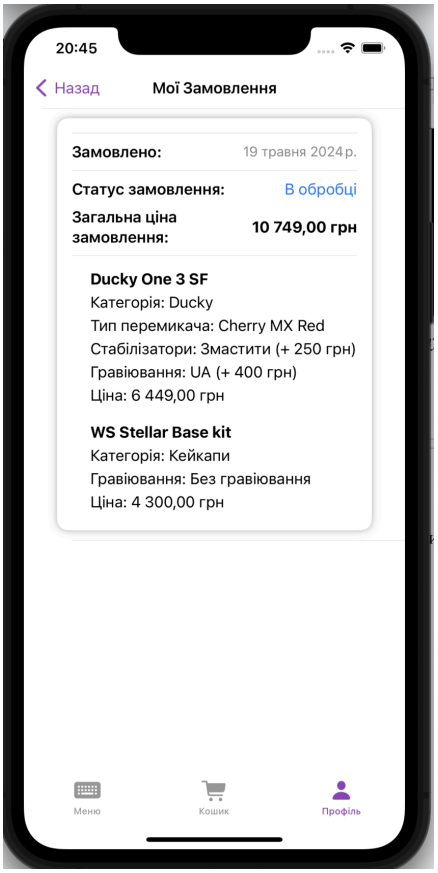


Рисунок 3.23 - Зовнішній вигляд сторінки замовлень

3.4. Технічне та системне забезпечення розробки

3.4.1. Обґрунтування вибору технічних засобів

Для продажу ігрових девайсів “Funkeys” на мобільному додатку було критично важливо вибрати такі технічні засоби, які б забезпечили надійність, масштабованість та високу продуктивність. Нижче наведені основні технічні засоби, використані у проекті, та обґрунтування їх вибору:

Вибір Операційної системи

Для реалізації мобільного додатку було обрано платформу iOS з кількох причин. По-перше, покупці "Funkeys", які купують кастомні клавіатури, часто є більш платоспроможними, а значна частина цієї аудиторії користується пристроями Apple. Це робить розробку додатку для iOS стратегічно вигідною, оскільки він буде відповідати потребам основного сегменту клієнтів.

По-друге, розробка додатку на нативній мові для iOS є легшою в підтримці. Використання SwiftUI [10] дозволяє створювати більш стабільні та ефективні додатки з високим рівнем користувацького досвіду. Крім того, платформа iOS забезпечує високу безпеку та продуктивність, що є важливим для електронної комерції. Таким чином, вибір iOS як платформи для розробки мобільного додатку “Funkeys” є обґрунтованим і стратегічно правильним рішенням.

Клієнтська частина

Для розробки додатку була обрана мова програмування Swift [7]. Swift - це високопродуктивна мова програмування, спеціально розроблена для платформи iOS. Вона забезпечує швидку та ефективну роботу додатків та має має зручний синтаксис, за допомогою якого підтримувати та розробляти додатки ефективніше.

Для інтерфейсу користувача додатку було обрано фреймворк SwiftUI, оскільки він є сучасною технологією для розробки інтерфейсу користувача, що дозволяє створювати складні UI швидше та ефективніше. SwiftUI пропонує більш простий і зрозумілий синтаксис у порівнянні з UIKit, що прискорює процес розробки.

Вибір бази даних та бекенд-сервісу

Для збереження даних користувачів, замовлень та товарів було обрано Firebase. Це рішення було прийнято через декілька причин:

- Firebase надає можливість синхронізації даних у реальному часі, що є критично важливим для додатків з високим рівнем інтерактивності;
- Firebase надає готові SDK, які легко інтегруються з мобільними додатками, зокрема з SwiftUI;
- Firebase може автоматично масштабуватися в залежності від навантаження, що дозволяє обслуговувати велику кількість користувачів без збоїв;
- Firebase пропонує вбудовані механізми безпеки, включаючи аутентифікацію користувачів, що дозволяє легко керувати доступом до даних.

Вибір Swift, SwiftUI та Firebase був обумовлений їх високою продуктивністю та сучасними можливостями. Цей набір технічних засобів дозволяє створити зручний, швидкий та безпечний мобільний додаток для цифрової фірми "Funkeys" .

3.4.2. Обґрунтування вибору протоколу обміну даними

Для забезпечення безпечної передачі даних у мобільному додатку для "Funkeys" було обрано протокол HTTPS (HyperText Transfer Protocol Secure) [9]. Під час процесу реєстрації та авторизації в додатку передача облікових даних користувача здійснюється саме через HTTPS. Цей протокол забезпечує шифрування інформації, що перешкоджає її перехопленню або компрометації під час трансляції до серверів Firebase.

Додаток використовує Firebase у реальному часі, хмарну базу даних NoSQL [11] яка підтримує захищене з’єднання через HTTPS. Це гарантує безпечну передачу даних між додатком та базою даних, захищаючи їх від можливого перехоплення або несанкціонованої модифікації. Використання Firebase дозволяє швидко синхронізувати дані у реальному часі, що є важливим для інтерактивності та актуальності інформації в додатку.

3.4.3. Обґрунтування стратегії адміністрування системи

Адміністрування системи для мобільного додатку "Funkeys" є важливою складовою успішного функціонування та підтримки додатку. Оскільки додаток використовує Firebase для збереження даних, передбачається інтеграція з існуючою системою адміністрування магазину, що забезпечить ефективне управління товарними позиціями, замовленнями та користувачами.користувацькими даними, замовленнями та іншими аспектами бізнесу.

3.4.4. Заходи захисту від несанкціонованого доступу до системи

Firebase Authentication є ключовим компонентом у забезпеченні безпеки додатків, оскільки вона управляє процесом автентифікації користувачів без необхідності зберігання чутливих даних, таких як паролі, локально в додатку. Замість цього, Firebase Authentication використовує JSON Web Tokens (JWT), які є стандартом для безпечної передачі інформації між двома сторонами. JWT містить всю необхідну інформацію для автентифікації та може бути перевірений для підтвердження його справжності.

Коли користувач успішно аунтефікувався, Firebase Authentication повертає JWT, який потім використовується для здійснення запитів до сервера. Сервер може перевірити цей токен, щоб підтвердити, що запит прийшов від автентифікованого користувача. Це дозволяє забезпечити безпечний обмін даними та захистити систему від несанкціонованого доступу. Додаток, створений на основі такої системи, обмежує зберігання особистих даних до мінімуму.

Використання Firebase для аутентифікації та управління базою даних, а також строга перевірка введених даних, сприяють мінімізації ризику витоку інформації або несанкціонованого доступу. Firebase також дозволяє здійснювати аутентифікації за допомогою Apple, Google та інших засобів, що забезпечує надійний захист користувацьких даних.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці включає систему заходів, спрямованих на збереження здоров'я і працездатності працівників у процесі їх трудової діяльності. Важливість дотримання загальних вимог охорони праці не можна недооцінювати, оскільки це дозволяє знизити ризики виробничих травм та професійних захворювань.

Основні принципи охорони праці в контексті розробки програмного забезпечення включають:

1. Дотримання законодавчих вимог: Розробники повинні бути ознайомлені з діючими нормативними актами і стандартами з охорони праці.
2. Навчання і інструктаж: Регулярні тренінги з питань охорони праці та техніки безпеки для всіх працівників.
3. Контроль за дотриманням вимог: Постійний моніторинг і контроль умов праці, виконання вимог безпеки.
4. Забезпечення безпеки робочого місця: Організація робочих місць відповідно до вимог безпеки та ергономіки.
5. Використання засобів індивідуального захисту (ЗІЗ): За необхідності, забезпечення працівників засобами індивідуального захисту.

Розробка програмного забезпечення часто проводиться в офісних умовах, де основними ризиками є недостатня ергономіка робочих місць, неадекватне освітлення, відсутність належної вентиляції та інші фактори, що можуть впливати на здоров'я працівників.

Техніка безпеки при розробці програмного забезпечення охоплює комплекс заходів, спрямованих на запобігання нещасним випадкам та професійним захворюванням. Особлива увага приділяється забезпеченню безпечних умов праці для розробників, які часто працюють за комп'ютерами тривалий час.

Основні заходи техніки безпеки включають:

1. Правильна організація робочого місця: Забезпечення зручних і безпечних робочих місць, які відповідають ергономічним вимогам. Робоче місце повинно бути обладнане регульованими стільцями, належним освітленням та моніторами, які можна налаштовувати по висоті і куту нахилу.
2. Регулярні перерви: Встановлення регламентованих перерв для уникнення перенапруження очей та м’язів. Рекомендується робити короткі перерви кожні 60 хвилин роботи.
3. Освітлення: Забезпечення належного рівня освітлення на робочому місці. Використання як природного, так і штучного освітлення, щоб уникнути відблисків на моніторах.
4. Вентиляція та кондиціонування: Забезпечення якісної вентиляції та кондиціонування повітря для підтримання комфортних температурних умов і свіжого повітря.
5. Захист від електромагнітного випромінювання: Використання моніторів з низьким рівнем випромінювання та регулярне перевіряння технічного стану обладнання.

Дотримання цих заходів сприяє збереженню здоров'я працівників та підвищенню їх продуктивності.

Ергономіка відіграє важливу роль у забезпеченні комфортних і безпечних умов праці для розробників програмного забезпечення. Вона включає в себе розробку та налаштування робочих місць таким чином, щоб мінімізувати фізичне навантаження та ризик виникнення професійних захворювань.

Основні принципи ергономіки робочого місця:

1. Розташування обладнання: Робоче місце повинно бути організоване так, щоб всі необхідні інструменти та матеріали були легко доступні, що дозволяє уникнути надмірного руху та натягування.
2. Регульовані меблі: Стільці і столи повинні бути регульованими по висоті та куту нахилу, щоб забезпечити правильну поставу та зменшити навантаження на спину і шию.

3. Розташування монітора: Монітор повинен бути розташований на рівні очей або трохи нижче, на відстані витягнутої руки, щоб уникнути напруги очей і шиї.
4. Положення рук та зап'ясть: Клавіатура та мишка повинні бути розташовані таким чином, щоб руки і зап'ястя знаходилися в природному положенні, що зменшує ризик виникнення тунельного синдрому.
5. Освітлення: Освітлення повинно бути достатнім, але не надто яскравим, щоб уникнути відблисків на екрані монітора. Рекомендується використання комбінованого освітлення (природного та штучного).
6. Перерви та фізична активність: Розробники повинні робити регулярні перерви для розминки та зміни положення тіла. Це допомагає уникнути перенапруження м’язів і покращує кровообіг.

Дотримання цих принципів сприяє підвищенню продуктивності праці, зменшенню втоми і ризиків професійних захворювань серед розробників програмного забезпечення.

Пожежна безпека є критично важливою складовою забезпечення безпечних умов праці. Вона включає в себе комплекс заходів, спрямованих на попередження виникнення пожеж та забезпечення швидкого і безпечного реагування у разі їх виникнення.

Основні заходи пожежної безпеки:

1. Встановлення пожежної сигналізації: Установка сучасних систем пожежної сигналізації, які включають в себе димові детектори, теплові датчики та системи оповіщення, що дозволяють швидко виявляти та реагувати на пожежу.
2. Обладнання засобами пожежогасіння: Робочі приміщення повинні бути обладнані вогнегасниками, пожежними кранами та іншими засобами первинного пожежогасіння. Важливо регулярно перевіряти їхню справність і проводити навчання працівників щодо їх використання.

3. Евакуаційні шляхи: Забезпечення вільного доступу до евакуаційних виходів та шляхів евакуації. Всі евакуаційні шляхи повинні бути чітко визначені, освітлені та утримуватися у вільному від перешкод стані.
4. План евакуації: Розробка та регулярне оновлення планів евакуації, які включають маршрути виходу з приміщення у разі пожежі. Проведення регулярних тренувань з евакуації для працівників.
5. Контроль за електрообладнанням: Регулярна перевірка стану електрообладнання та електропроводки. Важливо запобігати використанню несправних електроприладів, які можуть стати джерелом загоряння.
6. Навчання працівників: Проведення регулярних інструктажів та тренінгів з питань пожежної безпеки для всіх працівників. Це допоможе підвищити рівень обізнаності та готовності до дій у разі пожежі.

Виконання цих заходів забезпечує високий рівень пожежної безпеки, що значно знижує ризик виникнення пожеж та забезпечує безпеку працівників у разі надзвичайної ситуації.

Електробезпека також є важливою складовою охорони праці, особливо у сучасних офісах, де використовується велика кількість електронного обладнання. Правильне використання та обслуговування електричних пристроїв дозволяє запобігти виникненню пожеж та зменшити ризик ураження електричним струмом.

Основні заходи електробезпеки:

1. Регулярна перевірка електрообладнання: Усі електричні пристрої повинні регулярно перевірятися на справність. Це включає огляд кабелів, розеток, вимикачів та інших елементів електричної системи.
2. Використання заземлення: Усі електричні прилади повинні бути правильно заземлені, щоб уникнути ризику ураження електричним струмом у випадку несправностей.

3. Уникнення перевантаження електромережі: Використання розеток та подовжувачів повинно бути обмежене, щоб уникнути перевантаження електромережі, що може призвести до короткого замикання та пожежі.
4. Використання засобів захисту: Встановлення автоматичних вимикачів, запобіжників та пристроїв захисного відключення (ПЗВ) для забезпечення безпеки електромережі.
5. Інструктаж та навчання персоналу: Працівники повинні бути ознайомлені з правилами безпечного використання електроприладів, а також з діями у разі виникнення аварійних ситуацій.
6. Правильне розташування електроприладів: Електричне обладнання повинно бути розташоване таким чином, щоб уникнути контакту з водою та іншими рідинами. Розетки та вимикачі повинні бути встановлені на безпечній відстані від джерел вологи.

ВИСНОВКИ

Дипломна робота на тему "Розроблення мобільного додатку для продажу ігрових девайсів" мала на меті створення функціонального, зручного та ефективного інструменту для продажу товарів фірми “Funkeys”. У процесі виконання роботи було досягнуто кілька ключових цілей, які підтверджують успішність та доцільність проведених досліджень і розробок.

Було проведено детальний аналіз вимог до мобільного додатку, визначено основні функціональні та нефункціональні вимоги, що дозволило розробити чіткий план проекту. Визначено ключові функціональні можливості додатку, такі як реєстрація користувачів, перегляд товарів, додавання їх до кошика та здійснення замовлень.

Запропонована архітектура мобільного додатку базується на використанні сучасних технологій та підходів до розробки, таких як SwiftUI та Firebase. Розроблено структуру додатку, яка включає основні модулі: аутентифікація, головне меню, кошик та профіль користувача.

У додатку реалізовано всі основні функції, передбачені технічним завданням. Користувачі мають можливість зареєструватися, входити в систему, переглядати та обирати товари за категоріями, додавати їх до кошика, а також здійснювати замовлення. Використання Firebase для зберігання даних забезпечує надійність та безпеку інформації.

Особлива увага приділена зручності використання додатку. Інтерфейс розроблений з урахуванням принципів ергономіки та зручності для користувача, що дозволяє легко та швидко здійснювати покупки. Використання SwiftUI забезпечує високу якість графічного інтерфейсу та його адаптивність до різних розмірів екрану.

Проведено тестування додатку на різних пристроях для виявлення та усунення можливих помилок. Здійснено оптимізацію коду та поліпшення продуктивності додатку.

Підсумовуючи, можна зазначити, що розроблений мобільний додаток для цифрової фірми “Funkeys” повністю відповідає поставленим цілям і завданням.

Він є зручним та ефективним інструментом для продажу ігрових девайсів, який забезпечує високу якість обслуговування користувачів та підвищує конкурентоспроможність компанії на ринку. Дослідження та розробки, проведені в рамках дипломної роботи, можуть бути використані як основа для подальшого вдосконалення та розвитку додатку, а також для створення аналогічних програмних продуктів у сфері електронної комерції.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інтернет-магазин Funkeys – [Електронний ресурс]. – Режим доступу до ресурсу: https://funkeys.com.ua
2. LucidChart – [Електронний ресурс]. – Режим доступу до ресурсу: https://lucid.app/documents
3. Bizagi Modeler – [Електронний ресурс]. – Режим доступу до ресурсу: https://www.bizagi.com/en/platform/modeler
4. Factum Group – [Електронний ресурс]. – Режим доступу до ресурсу: https://factum-ua.com
5. Цифрова інформація 2024: Україна – [Електронний ресурс]. – Режим доступу до ресурсу: https://datareportal.com/reports/digital-2024-ukraine
6. Xcode – [Електронний ресурс]. – Режим доступу до ресурсу: https://developer.apple.com/xcode/
7. Swift Documentation – [Електронний ресурс]. – Режим доступу до ресурсу: https://www.swift.org/documentation/
8. Firebase Documentation – [Електронний ресурс]. – Режим доступу до ресурсу: https://firebase.google.com/docs/firestore/quickstart
9. Про клієнт-серверну архітектуру – [Електронний ресурс]. – Режим доступу до ресурсу: https://intellipaat.com/blog/what-is-client-server-architecture/?US
10. SwiftUI Documentation – [Електронний ресурс]. – Режим доступу до ресурсу: https://developer.apple.com/documentation/swiftui/
11. Побудова моделі даних електронної комерції NoSQL – [Електронний ресурс]. – Режим доступу до ресурсу: https://fabric.inc/blog/commerce/nosql-ecommerce-data-model
12. Управління охорони праці (OSHA) – [Електронний ресурс]. – Режим доступу до ресурсу: https://www.osha.gov/
13. Методичні рекомендації до виконання випускної кваліфікаційної роботи на здобуття освітнього ступеня «Бакалавр» спеціальності 122 «Комп’ютерні науки» освітньо-професійної програми «Інформаційні

системи та штучний інтелект» ден. форми навчання [Електрон. ресурс] / уклад. О. М. М’якшило, М. П. Костіков. – К.: НУХТ, 2022. – 34 с.

14. ДСТУ ISO/IEC TR 15504. Інформаційні технології. Оцінювання процесів життєвого циклу програмних засобів. – 315 с.
15. ДСТУ ISO/IEC 25010:2016 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання.
16. ДСТУ ISO 6309:2007. Пожежна безпека. Загальні вимоги
17. ДСТУ ISO/IEC 15910:2012. Інформаційні технології. Документування програм. Документація користувача.

ДОДАТКИ

Додаток А. «Організаційна схема та бізнес-процеси»

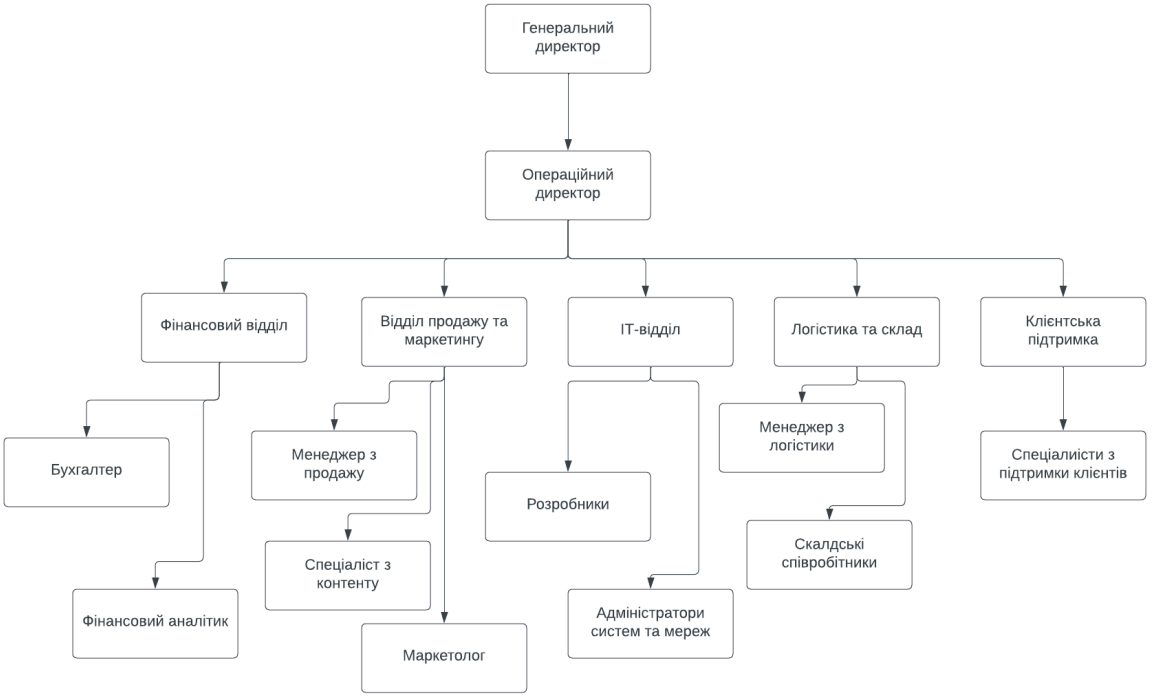


Рисунок А.1 - Організаційна схема цифрової фірми “Funkeys”

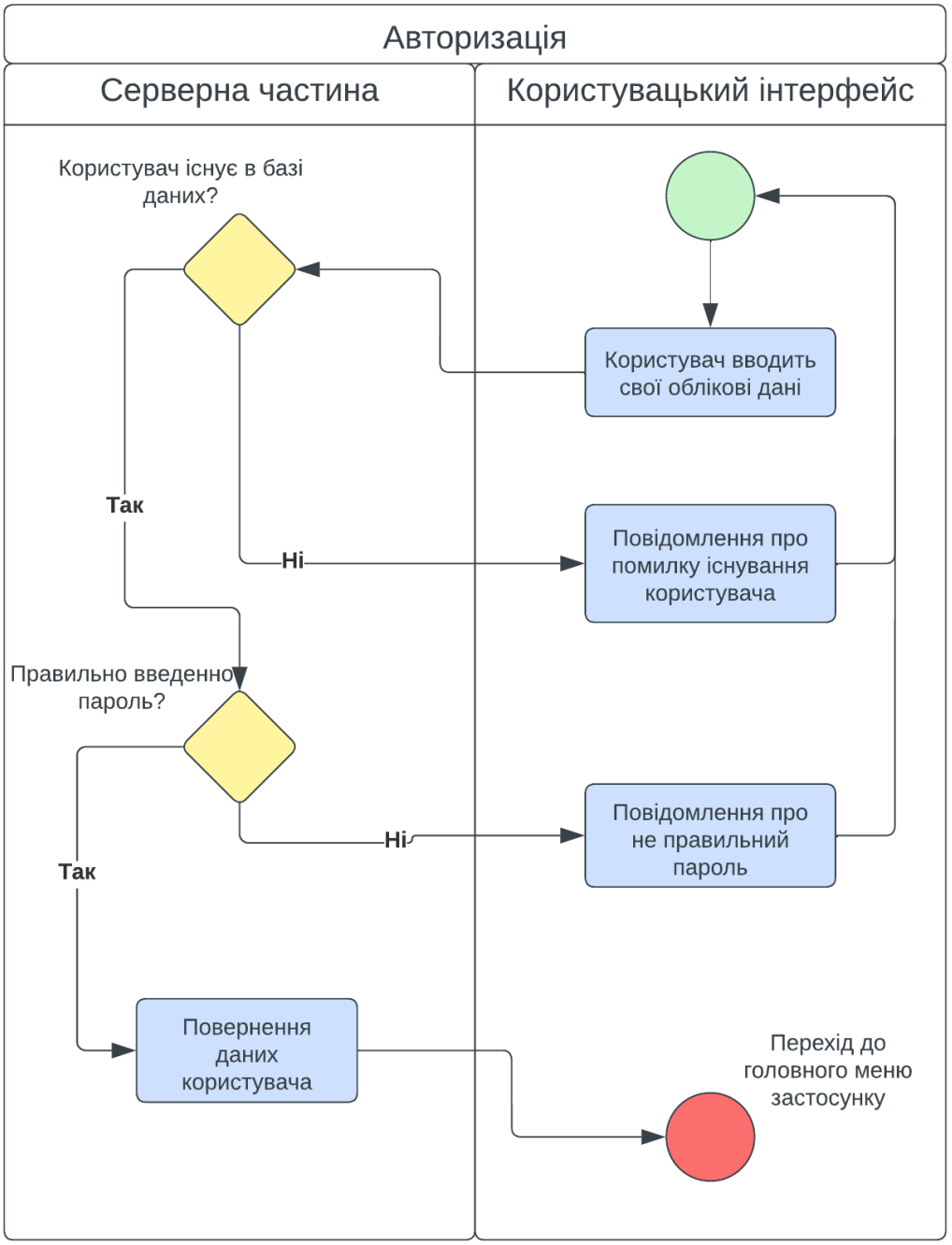


Рисунок А.2 - Модель бізнес-процесу авторизації користувача

Рисунок А.3 - Модель бізнес-процесу реєстрації користувача

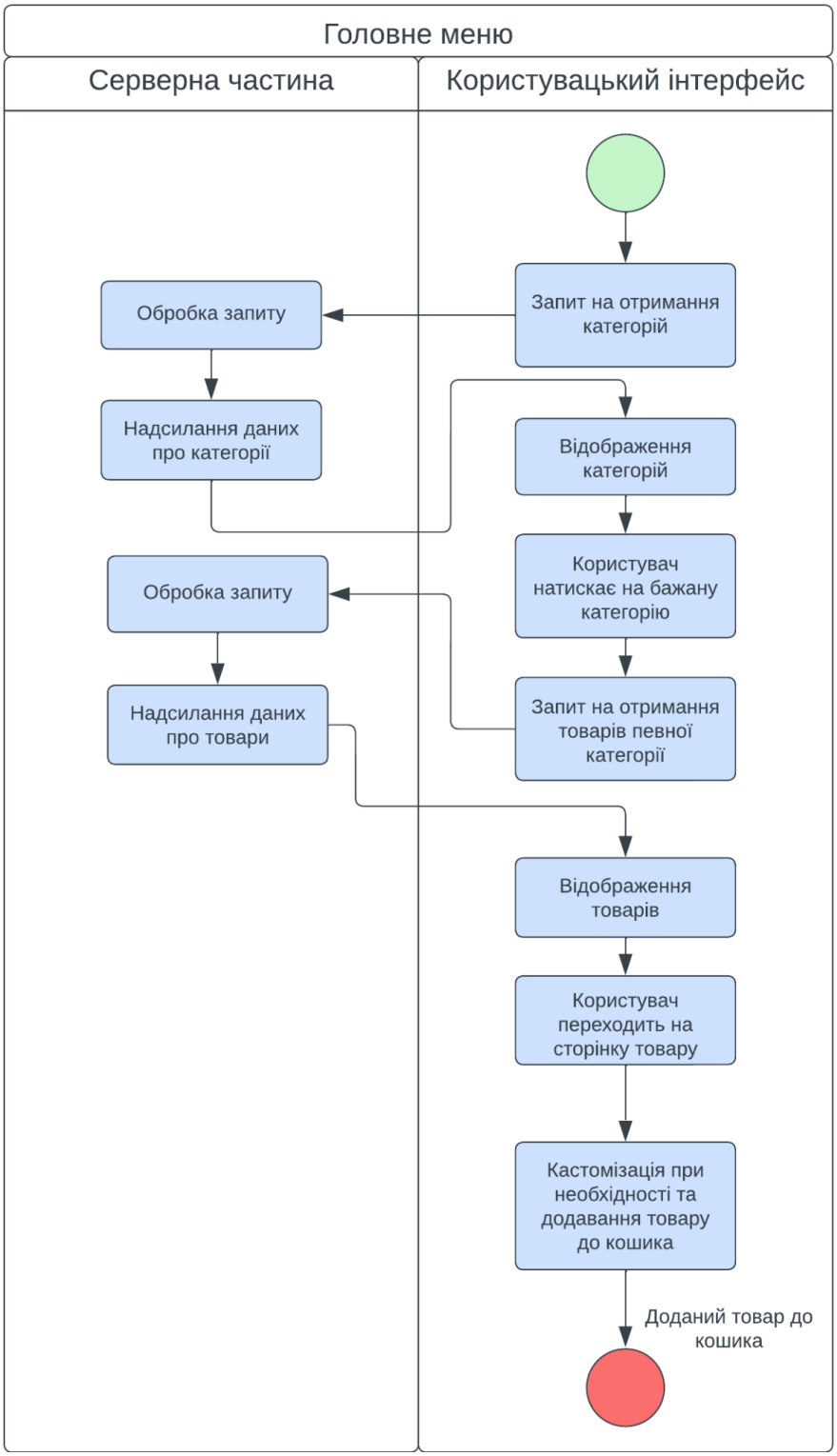


Рисунок А.4 - Модель бізнес-процесу взаємодії клієнта з головним екраном застосунку

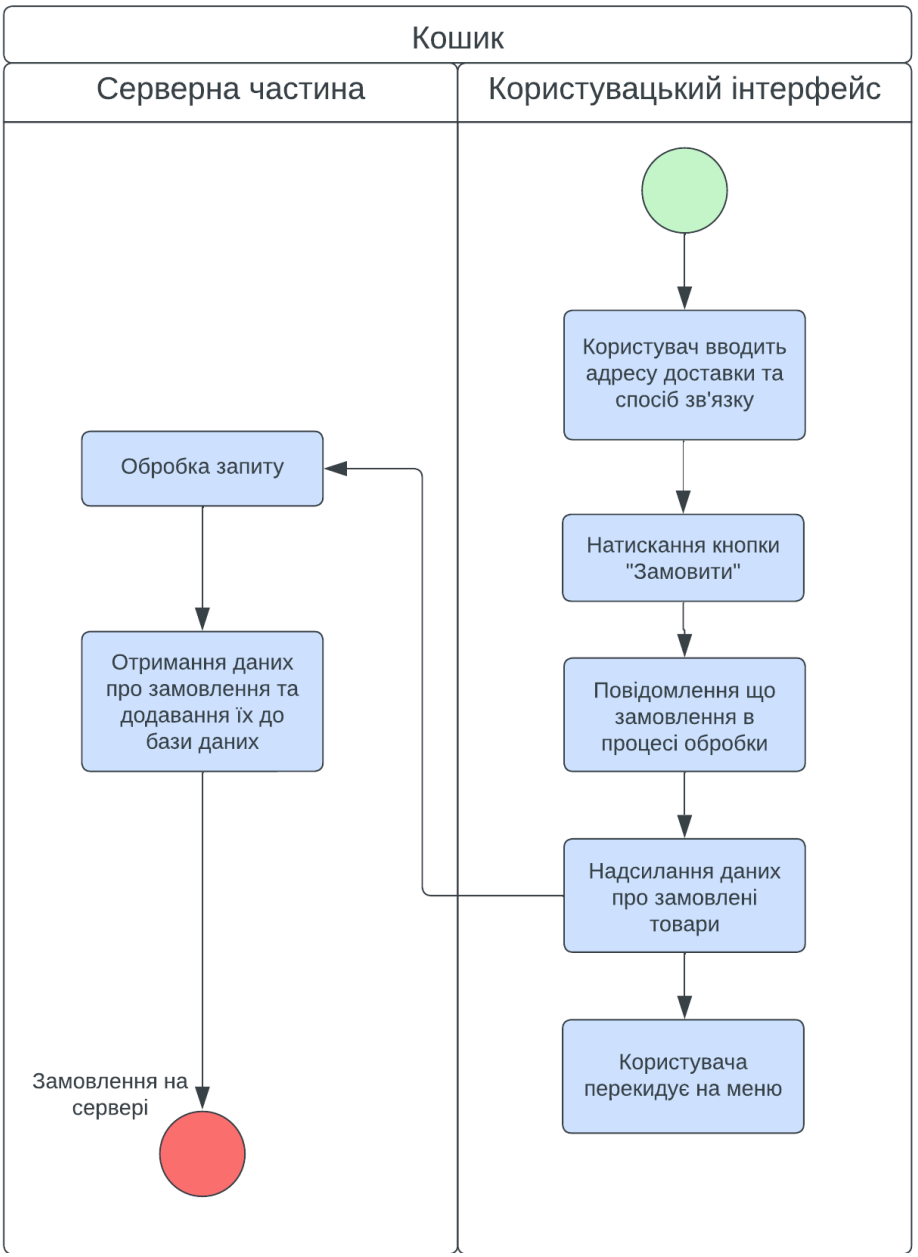


Рисунок А.5 - Модель бізнес-процесу взаємодії клієнта з кошиком

Рисунок А.6 - Модель бізнес-процесу взаємодії клієнта зі сторінкою замовлень

Додаток Б. «Структура бази даних»

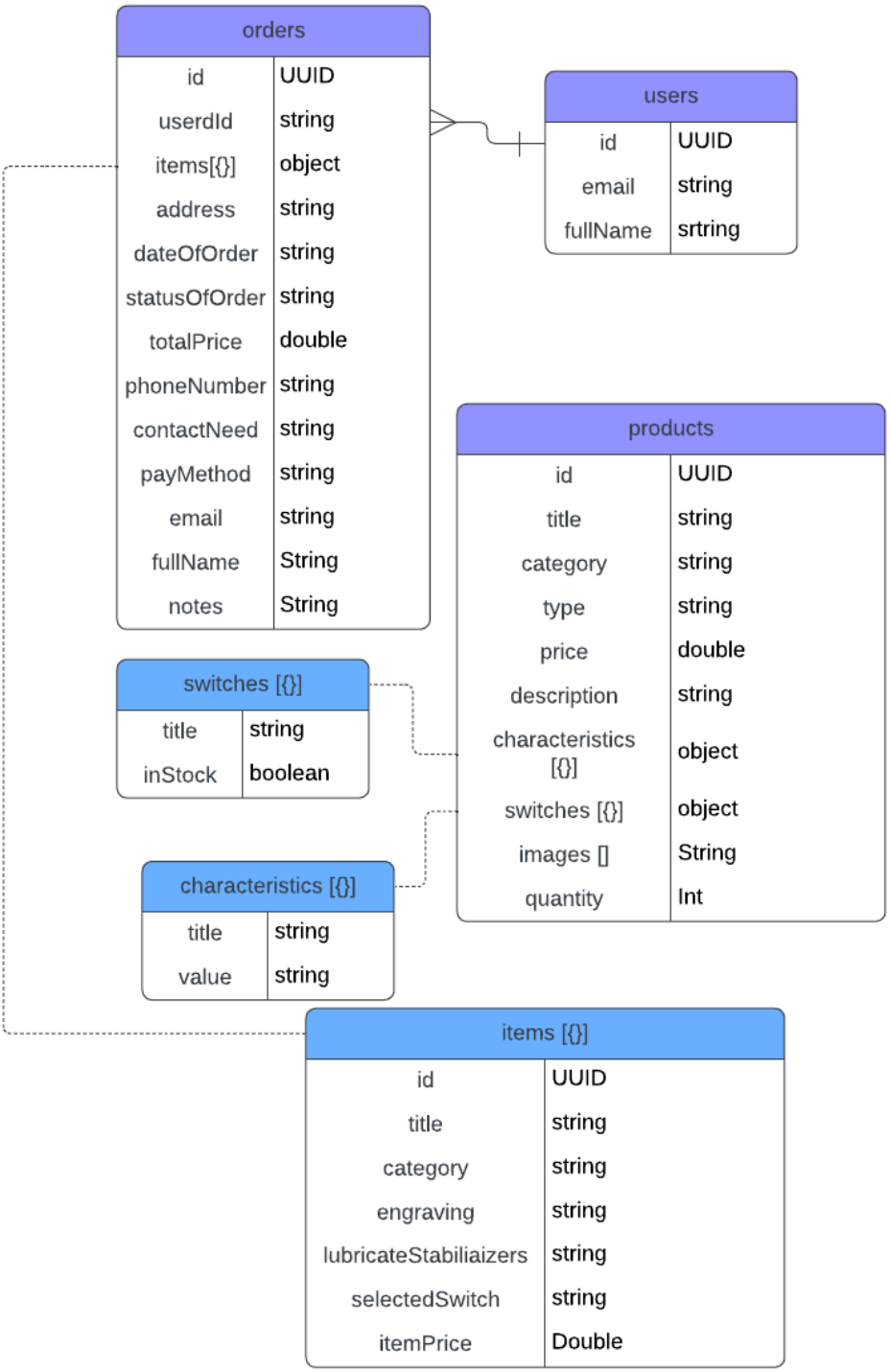


Рисунок Б.1 - Структура бази даних

Додаток В. «Фрагменти коду програми»

Рисунок В.1 - Вид “ContentView.swift”

FunkeysGamingShopApp | Products | Order | AuthModel | CartModel | ContentView | CartView | ProfileView | Log

unkeysGamingShop ⟩ FunkeysGamingShop ⟩ App ⟩ FunkeysGamingShopApp ⟩ No Selection

```swift
import SwiftUI
import Firebase

@main
struct FunkeysGamingShopApp: App {

    @UIApplicationDelegateAdaptor(AppDelegate.self) var delegate

    @StateObject var authModel = AuthModel()
    @StateObject var menuDataModel = MenuDataModel()
    @StateObject var cartModel = CartModel()
    @StateObject var orderModel = OrderModel()

    var body: some Scene {
        WindowGroup {
            ContentView()
                .environmentObject(authModel)
                .environmentObject(menuDataModel)
                .environmentObject(cartModel)
                .environmentObject(orderModel)
            }
        }
    }

class AppDelegate: NSObject, UIApplicationDelegate {
  func application(_ application: UIApplication,
                   didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey : Any]? = nil) -> Bool {
    FirebaseApp.configure()
    return true
  }
}
```

Рисунок В.2 - Код файлу “FunkeysGamingShopApp.swift”

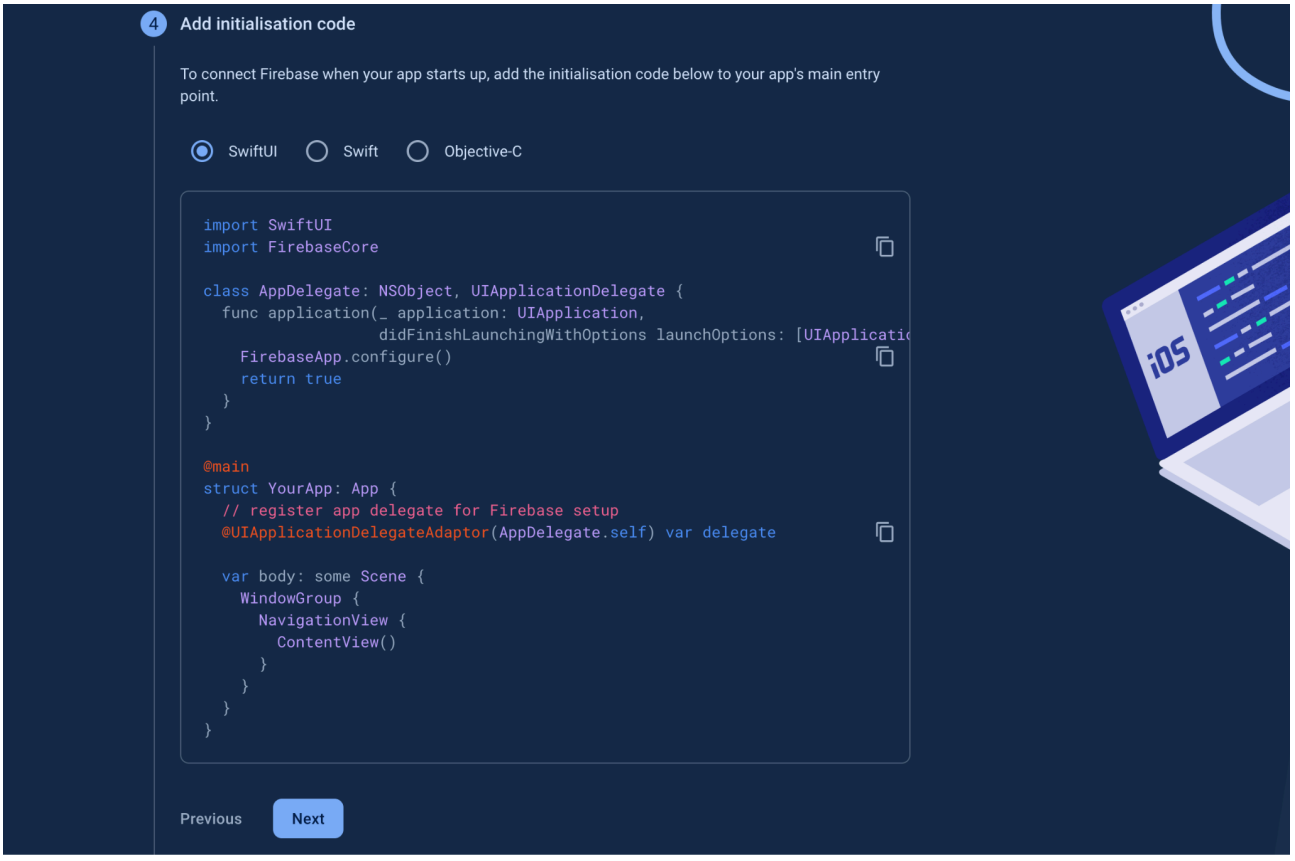


Рисунок В.3 - Інструкція по підключенню Firebase до проекту

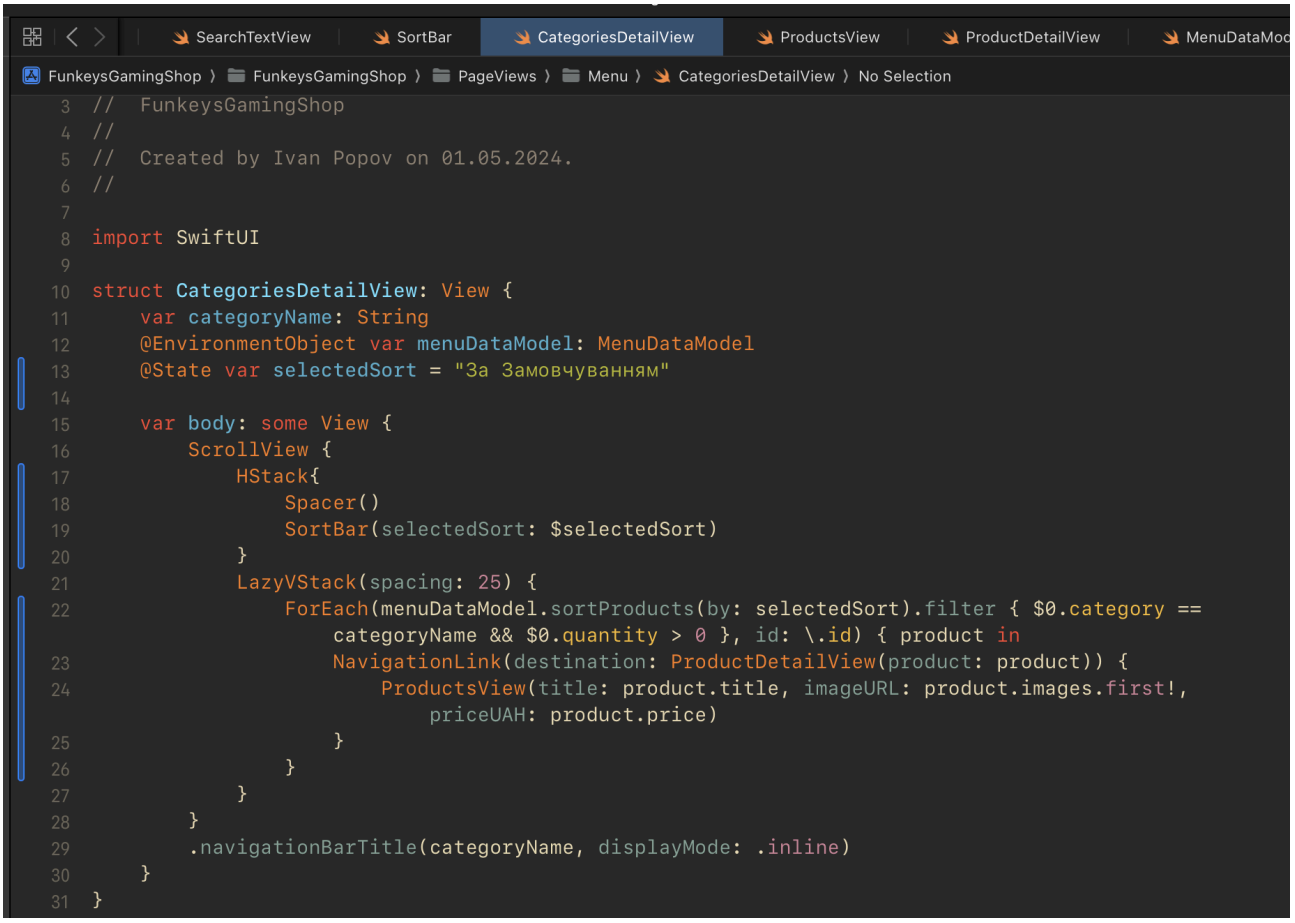
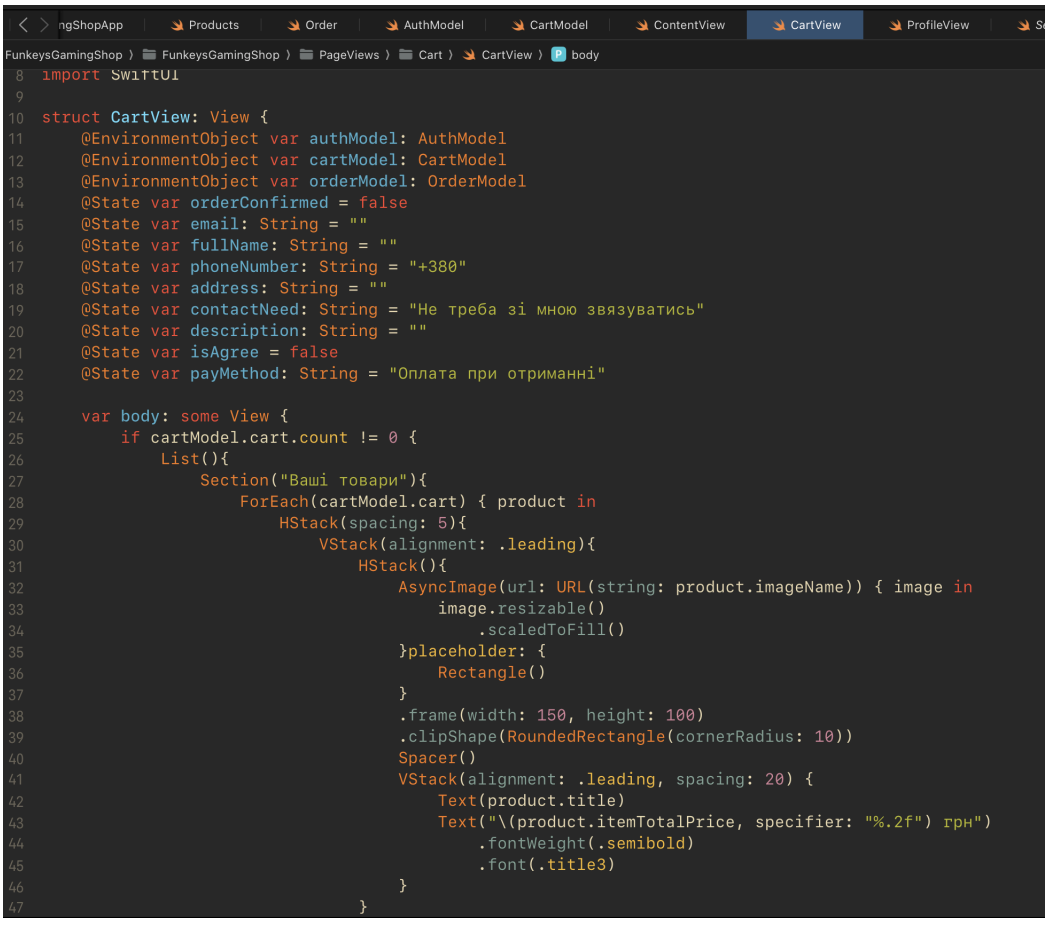
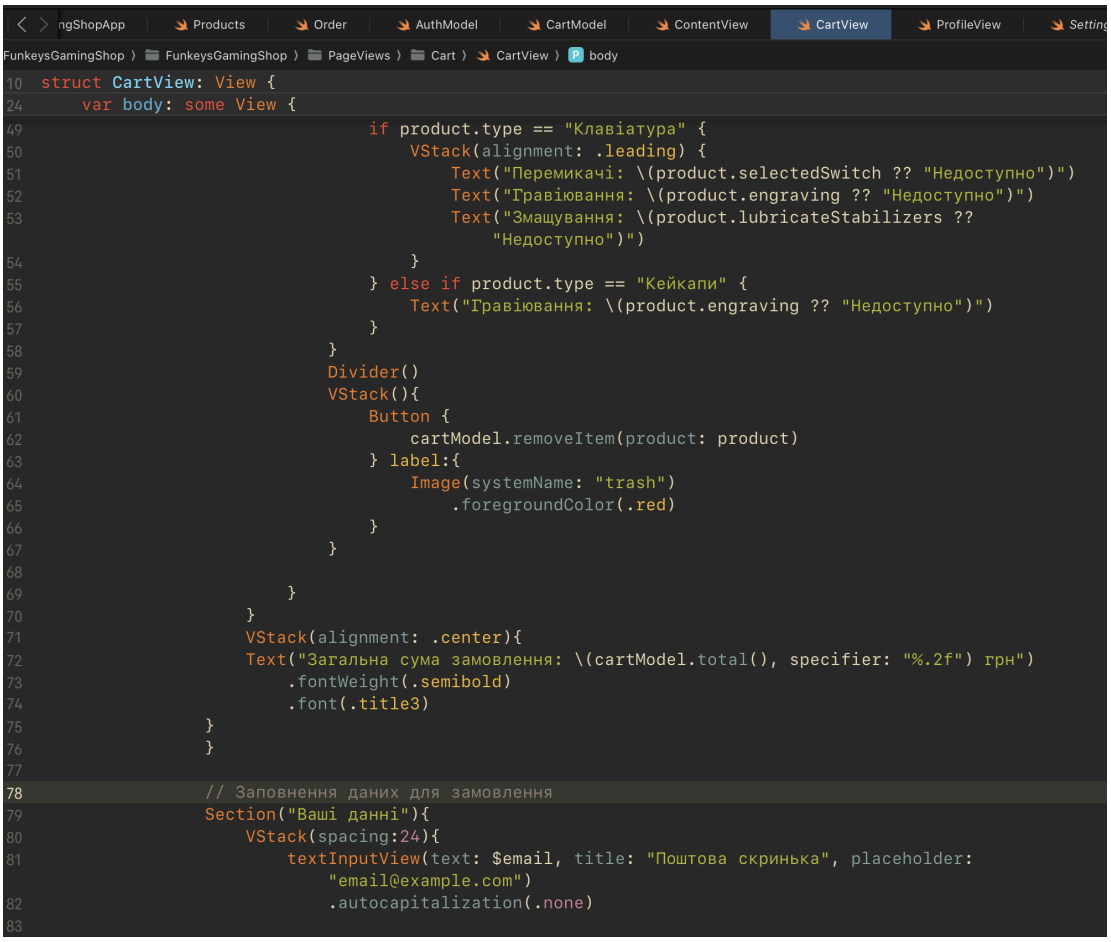
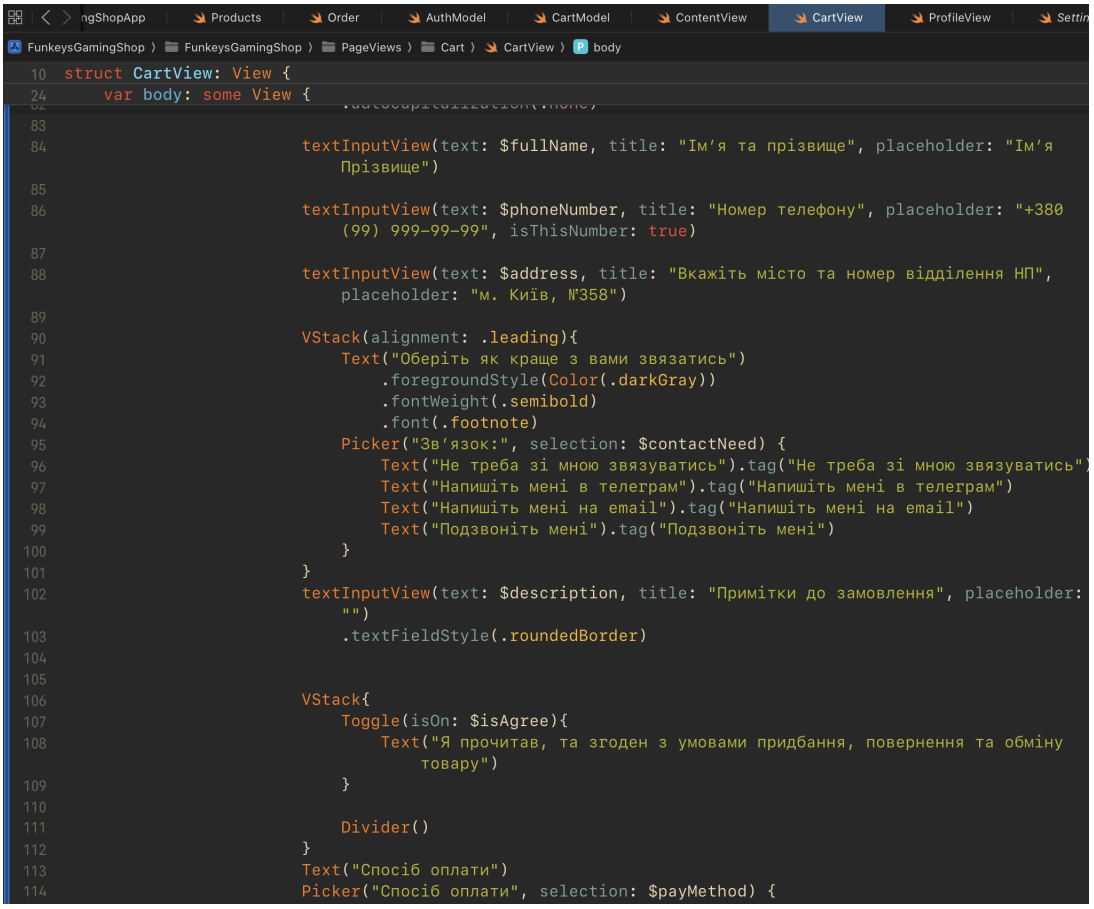


Рисунок В.4 - Код файлу “CategoriesDetailView.swift”

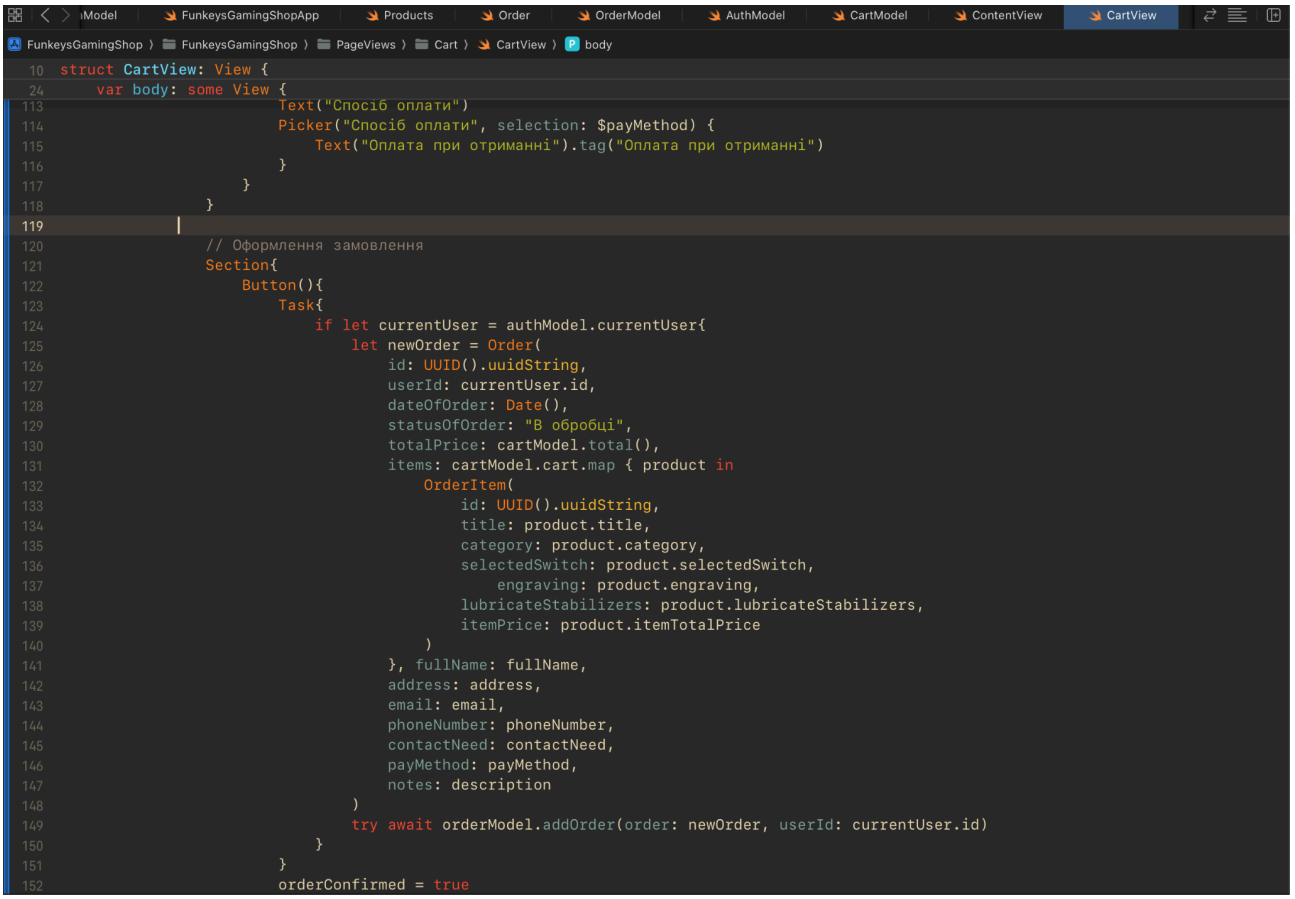




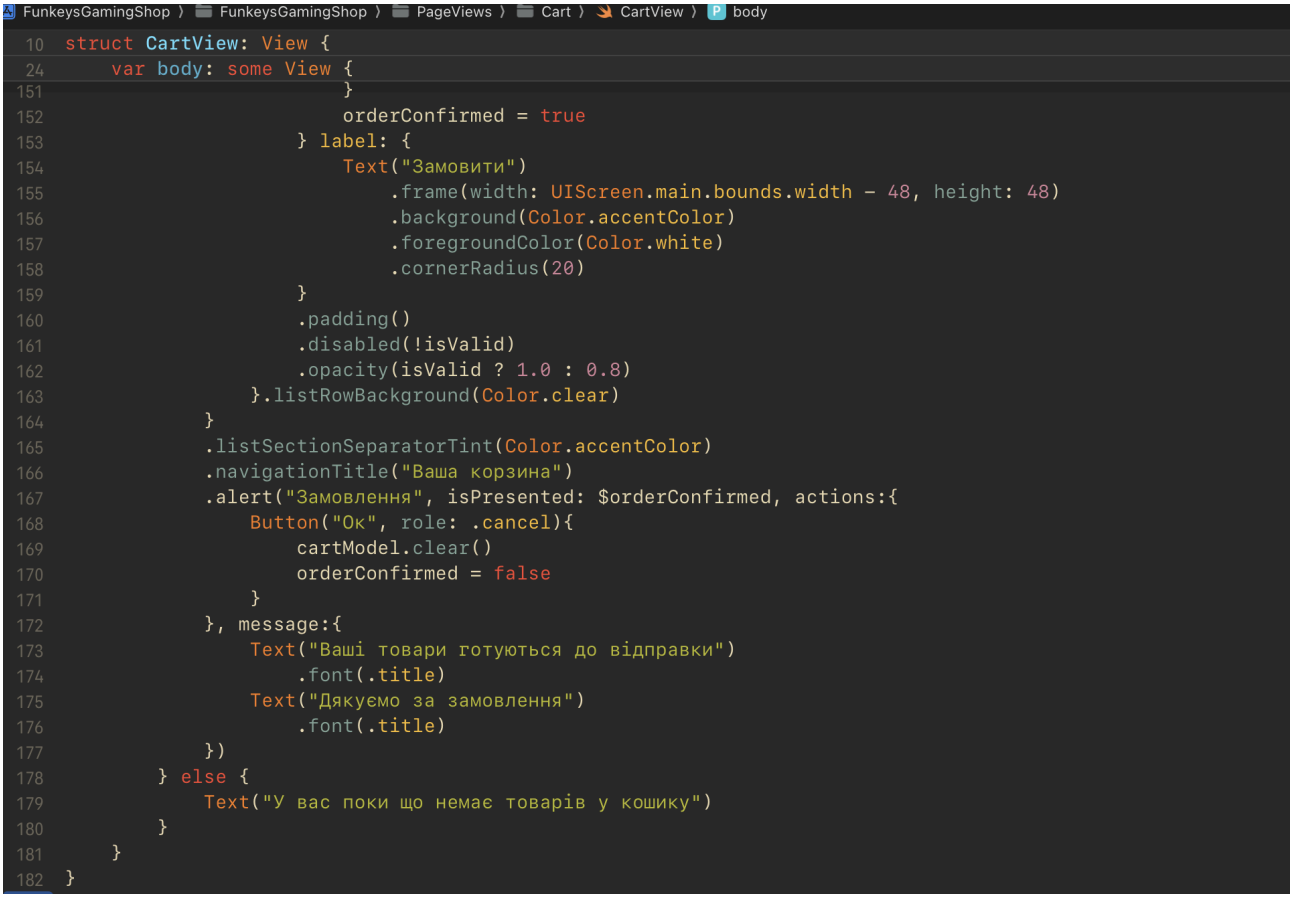
```
struct CartView: View {
    var body: some View {
                                if product.type == "Клавіатура" {
                                    VStack(alignment: .leading) {
                                        Text("Перемикачі: \(product.selectedSwitch ?? "Недоступно")")
                                        Text("Гравіювання: \(product.engraving ?? "Недоступно")")
                                        Text("Змащування: \(product.lubricateStabilizers ??
                                            "Недоступно")")
                                    }
                                } else if product.type == "Кейкапи" {
                                    Text("Гравіювання: \(product.engraving ?? "Недоступно")")
                                }
                            }
                            Divider()
                            VStack(){
                                Button {
                                    cartModel.removeItem(product: product)
                                } label:{
                                    Image(systemName: "trash")
                                        .foregroundColor(.red)
                                }
                            }

                        }
                    }
                    VStack(alignment: .center){
                    Text("Загальна сума замовлення: \(cartModel.total(), specifier: "%.2f") грн")
                        .fontWeight(.semibold)
                        .font(.title3)
                }
                }

                // Заповнення даних для замовлення
                Section("Ваші данні"){
                    VStack(spacing:24){
                        textInputView(text: $email, title: "Поштова скринька", placeholder:
                            "email@example.com")
                            .autocapitalization(.none)

```



```
struct CartView: View {
    var body: some View {
                            .autocapitalization(.none)

                        textInputView(text: $fullName, title: "Ім'я та прізвище", placeholder: "Ім'я
                            Прізвище")

                        textInputView(text: $phoneNumber, title: "Номер телефону", placeholder: "+380
                            (99) 999-99-99", isThisNumber: true)

                        textInputView(text: $address, title: "Вкажіть місто та номер відділення НП",
                            placeholder: "м. Київ, №358")

                        VStack(alignment: .leading){
                            Text("Оберіть як краще з вами звязатись")
                                .foregroundStyle(Color(.darkGray))
                                .fontWeight(.semibold)
                                .font(.footnote)
                            Picker("Зв'язок:", selection: $contactNeed) {
                                Text("Не треба зі мною звязуватись").tag("Не треба зі мною звязуватись")
                                Text("Напишіть мені в телеграм").tag("Напишіть мені в телеграм")
                                Text("Напишіть мені на email").tag("Напишіть мені на email")
                                Text("Подзвоніть мені").tag("Подзвоніть мені")
                            }
                        }
                        textInputView(text: $description, title: "Примітки до замовлення", placeholder:
                            "")
                            .textFieldStyle(.roundedBorder)


                        VStack{
                            Toggle(isOn: $isAgree){
                                Text("Я прочитав, та згоден з умовами придбання, повернення та обміну
                                    товару")
                            }

                            Divider()
                        }
                        Text("Спосіб оплати")
                        Picker("Спосіб оплати", selection: $payMethod) {
```



```swift
struct CartView: View {
    var body: some View {
                        Text("Спосіб оплати")
                        Picker("Спосіб оплати", selection: $payMethod) {
                            Text("Оплата при отриманні").tag("Оплата при отриманні")
                        }
                    }
                }

                // Оформлення замовлення
                Section{
                    Button(){
                        Task{
                            if let currentUser = authModel.currentUser{
                                let newOrder = Order(
                                    id: UUID().uuidString,
                                    userId: currentUser.id,
                                    dateOfOrder: Date(),
                                    statusOfOrder: "В обробці",
                                    totalPrice: cartModel.total(),
                                    items: cartModel.cart.map { product in
                                        OrderItem(
                                            id: UUID().uuidString,
                                            title: product.title,
                                            category: product.category,
                                            selectedSwitch: product.selectedSwitch,
                                                engraving: product.engraving,
                                            lubricateStabilizers: product.lubricateStabilizers,
                                            itemPrice: product.itemTotalPrice
                                        )
                                    }, fullName: fullName,
                                    address: address,
                                    email: email,
                                    phoneNumber: phoneNumber,
                                    contactNeed: contactNeed,
                                    payMethod: payMethod,
                                    notes: description
                                )
                                try await orderModel.addOrder(order: newOrder, userId: currentUser.id)
                            }
                        }
                        orderConfirmed = true
```



```swift
struct CartView: View {
    var body: some View {
                            }
                            orderConfirmed = true
                        } label: {
                            Text("Замовити")
                                .frame(width: UIScreen.main.bounds.width - 48, height: 48)
                                .background(Color.accentColor)
                                .foregroundColor(Color.white)
                                .cornerRadius(20)
                        }
                        .padding()
                        .disabled(!isValid)
                        .opacity(isValid ? 1.0 : 0.8)
                    }.listRowBackground(Color.clear)
                }
                .listSectionSeparatorTint(Color.accentColor)
                .navigationTitle("Ваша корзина")
                .alert("Замовлення", isPresented: $orderConfirmed, actions:{
                    Button("Ок", role: .cancel){
                        cartModel.clear()
                        orderConfirmed = false
                    }
                }, message:{
                    Text("Ваші товари готуються до відправки")
                        .font(.title)
                    Text("Дякуємо за замовлення")
                        .font(.title)
                })
            } else {
                Text("У вас поки що немає товарів у кошику")
            }
        }
    }
```

Рисунок В.5-9 - Код файлу “CartView.swift”

```swift
import SwiftUI

struct SettingView: View {
    let imageName: String
    let title: String
    let imageColor: Color

    var body: some View {
        HStack{
            Image(systemName: imageName)
                .imageScale(.small)
                .font(.title)
                .foregroundColor(imageColor)

            Text(title)
                .font(.subheadline)
                .foregroundColor(.black)
        }
    }
}
```

Рисунок В.10 - Код файлу компоненту для профілю “SettingView.swift”

Рисунок В.11 - Код файлу компоненту для товарів та категорій “ProductsView.swift”

```swift
import SwiftUI

struct SearchBar: View {
    @State var searchText = ""

    var body: some View {
        HStack{
            Image(systemName: "magnifyingglass")
                if searchText.isEmpty {
                    Text("Щось шукаєте ?")
                        .fontWeight(.semibold)
                        .foregroundColor(.gray)
                }
                Text(searchText)
                    .fontWeight(.semibold)
            Spacer()
        }
        .padding(.horizontal)
        .padding(.vertical, 10)
        .overlay{
            Capsule()
                .stroke(lineWidth: 0.5)
                .foregroundStyle(Color(.systemGray))
                .shadow(color: .black.opacity(0.4), radius: 1)
        }
        .padding()
    }
}
```

Рисунок В.12. - Код файлу компоненту для пошуку

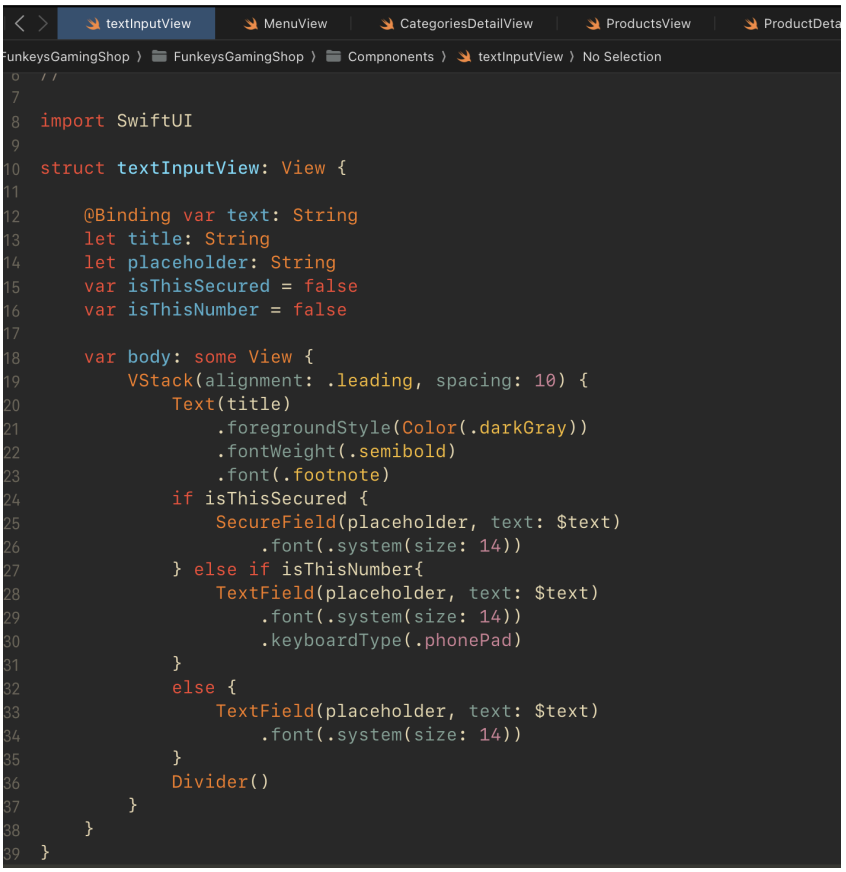
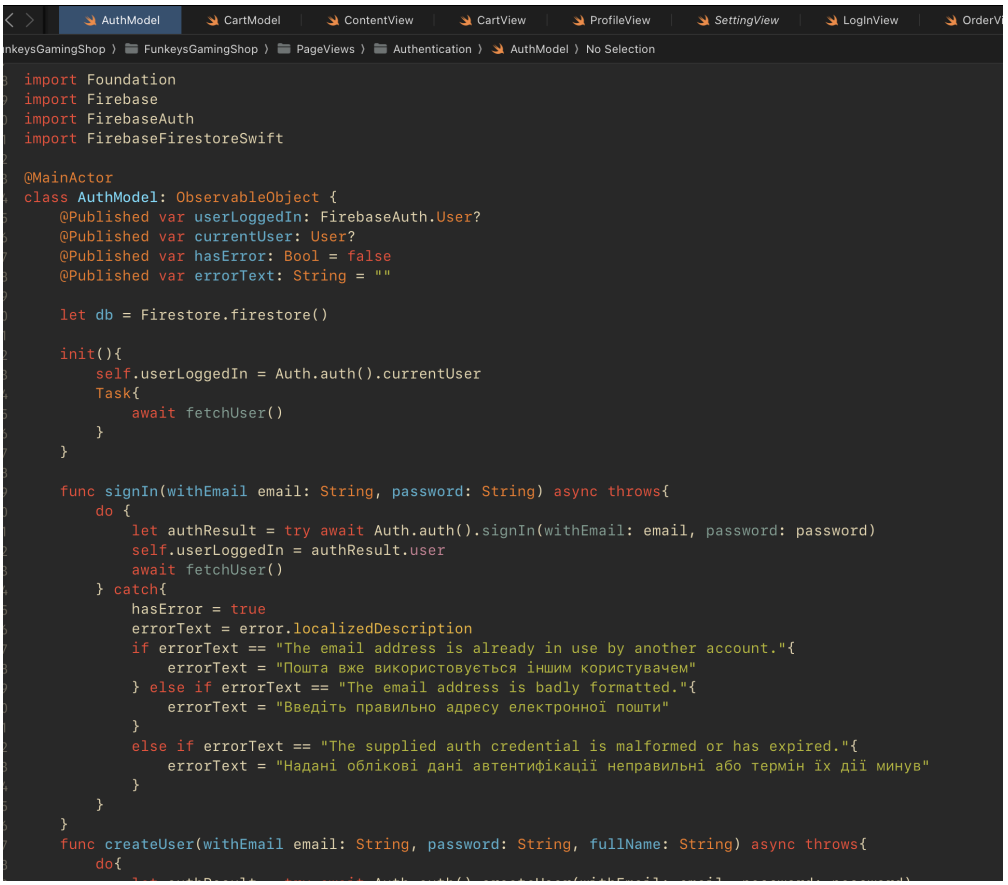


Рисунок В.13 - Код файлу компоненту для введення даних



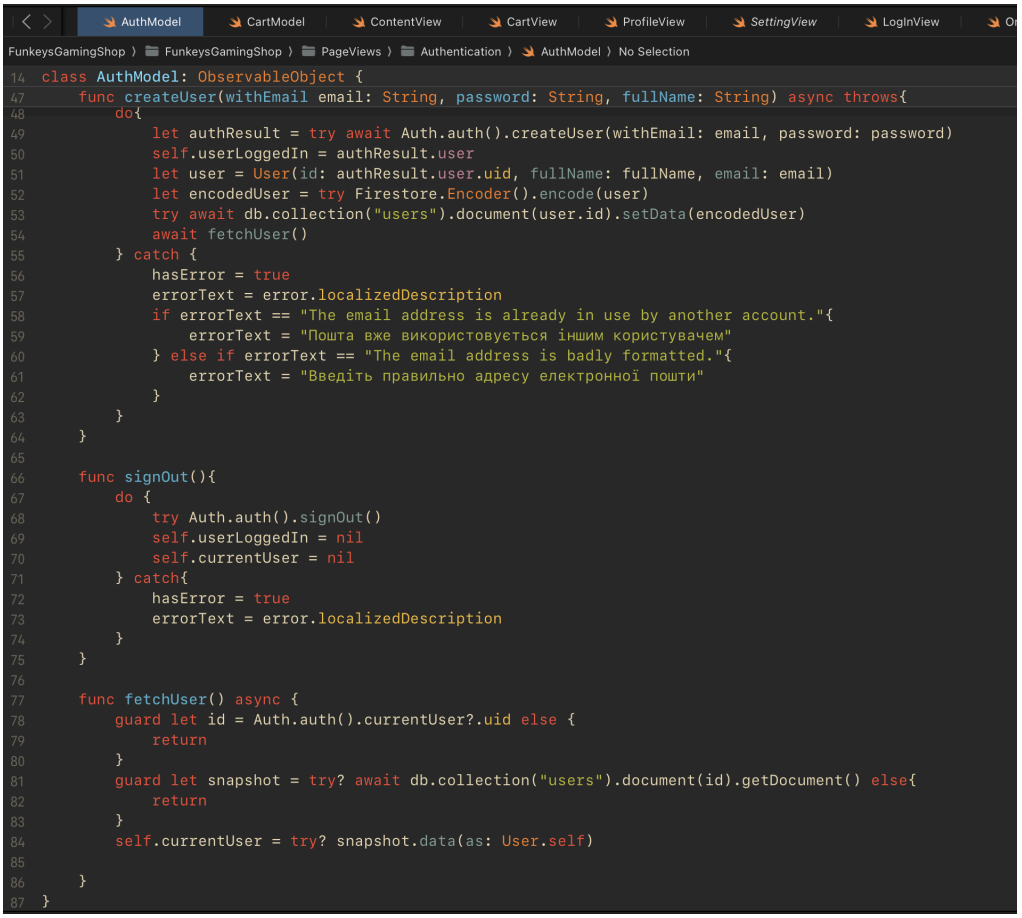


Рисунок В.14-15 - Код файлу AuthModel.swift

Рисунок В.15 - Код файлу “OrderModel.swift”

Додаток Г. «Додаткові матеріали»

Таблиця Г.1 - Порівняльна характеристика шляхів розробки

Критерій	Конструктори додатків	Фірми, фрилансери	Розробка самою фірмою
Якість	Низька, стандартна, неоригінальна	Залежить від репутації та досвіду, висока, але не гарантована	Висока, відповідає вашим вимогам та стандартам
Ціна	Низька, але містить додаткові витрати на підписку, комісію, рекламу тощо	Висока, але може бути обговорена та знижена	Середня, але є можливість контролювати бюджет та витрати
Надійність	Обмежена, використання тільки тих можливостей, які надає платформа	Розширена, є змога вказати свої потреби та бажання, але не завжди отримати те, що хочете	Висока, є можливість забезпечити безперебійну роботу, інтеграцію, підтримку та оновлення додатку
Функціональність	Обмежена, використання тільки тих можливостей, які надає платформа	Розширена, є змога вказати свої потреби та бажання	Розширена, є можливість реалізувати будь-які функції
Дизайн	Стандартний, наявний вибір з готових шаблонів та кольорів	Індивідуальний, є змога вказати свої преференції та вимоги, але не завжди отримати те, що очікуєте	Унікальний, є можливість створити власний дизайн та модифікувати його у майбутньому

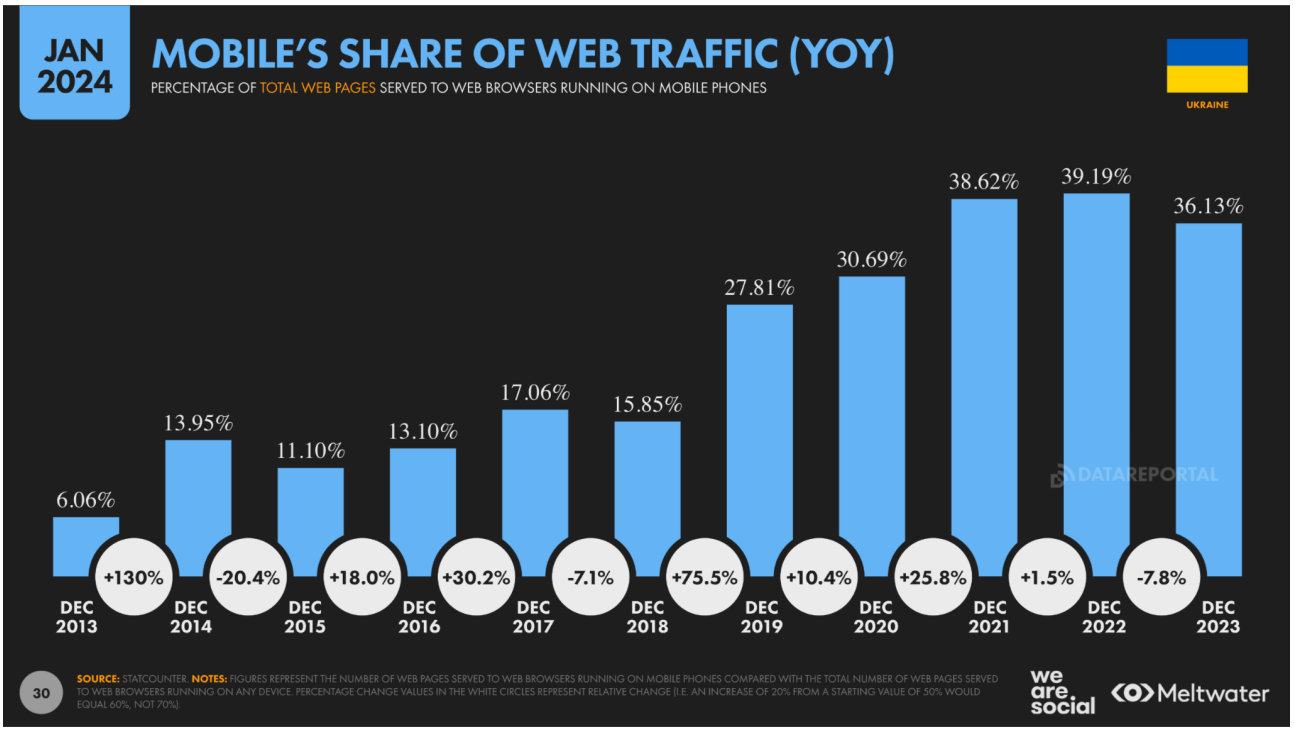


Рисунок Г.2 - Частка мобільного веб-трафіку