

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
Факультет кібербезпеки, комп'ютерної та програмної інженерії  
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

\_\_\_\_\_ Аліна САВЧЕНКО

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

# ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

*ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ*

**“МАГІСТРА”**

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ “ІНФОРМАЦІЙНІ  
УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ”

**Тема: “Мобільний додаток інтернет магазину”**

**Виконавець:** Велігоненко Сергій Олександрович

**Керівник:** проф. Моржов Володимир Іванович

**Нормоконтролер:** \_\_\_\_\_ Ігор РАЙЧЕВ

**Київ - 2021**

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12  
“Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні  
управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Аліна САВЧЕНКО

« \_\_\_\_ » \_\_\_\_\_ 2021р.

## ЗАВДАННЯ

**на виконання дипломної роботи студентки**

Велігоненка Сергія Олександровича

(прізвище, ім'я, по батькові)

- 1. Тема роботи:** «Мобільний додаток інтернет магазину» затверджена наказом ректора від 12.10.2021 за № 2228/ст.
- 2. Термін виконання роботи:** з 12.10.2021 по 31.12.2021.
- 3. Вихідні дані до роботи:** теоретичні відомості та основи проектування мобільних додатків, множина відомих архітектур та підходів до розробки програмних компонентів і мобільних додатків, множина відомих мобільних додатків
- 4. Зміст пояснювальної записки:** вступ, огляд теоретичної бази, аналіз мов програмування та підходів для розробки.
- 5. Перелік обов'язкового ілюстративного матеріалу:** слайди, презентація.

## 6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Отримання завдання на дипломну роботу та побудова плану-графіку виконання робіт.	12.10.2021 – 15.10.2021	
2.	Огляд та аналіз відомих програмних застосунків	16.10.2021 – 19.10.2021	
3.	Огляд та створення архітектури та дизайну.	20.10.2021 – 24.10.2021	
4.	Проектування	25.10.2021 – 31.10.2021	
5.	Написання Розділу 1 дипломної роботи.	01.11.2021 – 07.11.2021	
6.	Розробка та реалізація. Написання Розділу 2 дипломної роботи.	08.11.2021 – 17.11.2021	
7.	Написання Розділу 3 дипломної роботи. Завершення створення пояснювальної записки дипломної роботи.	18.11.2021 – 01.12.2021	
8.	Оформлення та друк пояснювальної записки.	02.12.2021 – 11.12.2021	
9.	Створення презентації, доповіді та підготовка до захисту дипломної роботи	12.12.2021 – 20.12.2021	

**7. Дата видачі завдання:** 12.10.2021р.

**Керівник дипломної роботи** \_\_\_\_\_ **Володимир Моржов**  
(підпис керівника)

**Завдання прийняла до виконання** \_\_\_\_\_ **Сергій Велігоненко**  
(підпис випускниці)

## РЕФЕРАТ

Пояснювальна записка до дипломної роботи “Мобільний додаток інтернет магазину” складається із вступу, трьох розділів, загальних висновків, списку використаних джерел і містить 12 сторінок тексту, 4 рисунків.

**Метою** дипломної роботи є розгляд теоретичних та практичних засад створення мобільних додатків та програмних систем на основі наборів загальних патернів і архітектур, що реалізують різні модулі.

**Предметом дослідження** оцінювання альтернативних додатків, розробка архітектури та підходу до розробки мобільного додатку

**Об’єктом дослідження** є структура мобільного додатку інтернет магазину, яка складається з клієнтської та серверної частини

**Ключові слова:** МОБІЛЬНИЙ ДОДАТОК, ІНТЕРНЕТ-МАГАЗИН, МОВИ ПРОГРАМУВАННЯ, КЛІЄНТ, ВЕБ-СЕРВІС

## ЗМІСТ

ВСТУП.....	6
1. Аналіз предметної області.....	8
1.1. Електронна комерція .....	8
1.2. Система керування сайтами Joomla .....	10
1.3. Система керування сайтами WordPress .....	13
1.4. Структура інтернет-магазину .....	15
1.5. Методи оплати .....	16
1.6. Система інтернет платежів .....	21
2. Огляд сучасних засобів та технологій розробки інтернет магазинів та мобільних додатків.....	23
2.1. Переваги мобільного додатку.....	23
2.2. Аналіз області. Огляд сучасних технологій.....	25
2.3. Вибір засобів та технологій розробки інтернет-магазину .....	27
2.4. Вибір середовища розробки.....	32
2.5. Вибір мови програмування .....	34
2.6. База даних .....	40
2.7. Структура даних, що надсилаються.....	46
2.8. Архітектура мобільного додатка.....	47
ВИСНОВКИ .....	50
Список бібліотечних джерел .....	51

## ВСТУП

На сьогоднішній день електронна комерція являється однією з найуспішніших сфер економіки України та щороку показує високі темпи розвитку. Діючим законодавством України поняття «електронна комерція» визначається як відносини, спрямовані на отримання прибутку, що виникають під час вчинення правочинів щодо набуття, зміни або припинення цивільних прав та обов'язків, здійснені дистанційно з використанням інформаційно-телекомунікаційних систем, внаслідок чого в учасників таких відносин виникають права та обов'язки майнового характеру. В результаті аналізу показників діяльності суб'єктів даної галузі за минулий рік було зроблено висновок, що однією з основних тенденцій вітчизняного ринку e-commerce стало зростання мобільного трафіку.

Зростання мобільного трафіку спостерігається у всьому світі. Цьому передувало збільшення територій, які покриває мобільний інтернет, покращення якості та збільшення швидкості мобільного інтернету а також доступність смартфонів різних виробників на ринку. До 2018 року більшість запитів надходила зі стаціонарних комп'ютерів, а після – з мобільних.

Також можна спостерігати й збільшення інтернет-користувачів. За останні 5 років їх кількість у всьому світі збільшилась до 4,5 млрд. Кожна людина проводить приблизно 7 годин в інтернеті. Це велика аудиторія, яку бізнес та маркетинг не може ігнорувати. То ж це спричинило вибух розробки мобільних застосунків та веб-сайтів. Ігри, соціальні мережи, стрімінгові сервіси для перегляду фільмів та серіалів, прослуховування музики, магазини – і це далеко не весь список.

Також сучасний світ змінився завдяки епідемії COVID-19. Всі, хто робив бізнес виключно в оф лайн, були змушені швидко пристосуватись до нових вимог. Зараз можна спостерігати що багато магазинів створили свої сайти та мобільні додатки, що допомогли їм не тільки пережити карантин, а навіть вийти на нову аудиторію і збільшити прибутки з продаж.

Виходячи з поставленої мети, було визначено завдання:

- обґрунтувати необхідність створення інтернет-магазину та мобільного
- додатки;
- Провести аналіз існуючих мобільних платформ;
- проаналізувати технології створення інтернет-магазинів;
- розробити зручний та ефективний графічний користувацький інтерфейс інтернет-магазину;
- Розробка дизайну, що робить знаходження користувача на сайті
- максимально комфортним;
- реалізувати програмний модуль для роботи інтернет-магазину;
- реалізувати програмний модуль для роботи мобільного додатку;
- Забезпечити зручний перегляд всього списку товарів;
- забезпечити зручний та ефективний інтерфейс службовців;
- оцінити ефективність виконаної розробки

# 1. Аналіз предметної області

## 1.1.Електронна комерція

Про процеси, що призводять до формування інформаційної економіки, почали говорити ще у 70-х роках 20 століття. Основа цих процесів – бурхливе зростання інформаційних та комунікаційних технологій, перетворення комп'ютера на домінуючу зброю праці та поширення мереж передачі даних. Підприємства у всьому світі змінюють способи організації та управління бізнесом. Відбувається відмова від старої ієрархічної структури, зникають бар'єри між відділеннями компанії, дедалі більше співробітників переходять на віддалену роботу. Спрощується взаємодія між компанією, її замовниками та постачальниками. Бізнес-процеси перебудовуються та виходять за старі кордони, створюються віртуальні підприємства та вдосконалюються технології аутсорсингу. Підйом економіки розвинених країн, що спостерігається сьогодні, не може бути обумовлений лише кількісним зростанням використовуваних комп'ютерів – ключове значення набуло об'єднання їх у мережи.



Рис.1.1

<b>Кафедра КІТ (47)</b>				<b>НАУ 21.23.91.000 ПЗ</b>			
<b>Виконав</b>	Велігоненко			<i>Мобільний додаток інтернет магазину</i>	<b>Лім.</b>	<b>Арк.</b>	<b>Аркушів</b>
<b>Керівник</b>	Моржов В.І.					7	54
<b>Консульт.</b>					<b>УС-211М 122</b>		
<b>Н. Контр.</b>	Райчев І.Е.						



Сьогодні електронна комерція надає найширші можливості як постачальникам, так і клієнтам. Серед цих можливостей:

самостійна реєстрація покупця на веб-сайті постачальника. Це створює додаткові зручності в обслуговуванні для клієнтів, а постачальникам дозволяє здійснювати адресну рекламу своїх товарів та послуг та маркетингові дослідження у процесі продажів;

оформлення замовлень через інтернет за допомогою електронних каталогів та прайс-листів. Дана можливість забезпечує значну економію часу при пошуку необхідного товару чи послуги та порівнянні цін різних постачальників;

електронна обробка замовлення, включаючи перевірку наявності товару складі, розрахунок можливих термінів поставки. Ця можливість є невід'ємною функціональною частиною логістичної системи підприємства;

прийом оплати за покупку через інтернет. Оплата при цьому може здійснюватися за допомогою банківських карток через захищені платіжні термінали, а також за допомогою локальних або міжнародних платіжних інтернет систем.

Електронна комерція має всі можливості для подальшого розвитку. Людині, яка працює 6 днів на тиждень з 10 до 19, їй буває ніколи сходити в магазин. Інтернет-магазин дозволяє зробити покупку не виходячи з офісу у будь-який час, а вибір та замовлення товару займе у нього кілька хвилин, якщо він точно знає, що хоче купити. Служба доставки інтернет-магазину доставить обраний товар у зручний час та місце. Крім цього вибір та оцінка властивостей товару відбувається в інтернет-магазині набагато швидше та зручніше ніж у звичайному магазині. Уявіть, що ви прийшли до звичайного магазину побутової техніки і хочете порівняти параметри 20-30 радіотелефонів. Для цього ви повинні вивчити всі ціни: запам'ятати характеристики, ціни та назви моделей. Можна звернутися до продавця-консультанта, якщо він не

зайнятий, з проханням розповісти про всі ці товари. Але звичайно жоден продавець не має стільки вільного часу для роботи з одним клієнтом, і немає гарантії, що він має всю інформацію.

Крім того, асортимент інтернет-магазину нічим не обмежений (як, наприклад, асортимент звичайного магазину обмежений площею торгового павільйону). А якщо передбачено пошук за параметрами, то Ви просто можете вказати характеристики, яким повинен відповідати товар, а потім уже вибирати зі списку моделей, які відповідають вашому запиту. І ще одне дуже важливе зауваження — жоден менеджер з продажу не може пам'ятати стільки інформації про товар, скільки може надати Вам інтернет-магазин. Крім того, інтернет-магазин може видати Вам таку інформацію, як купівельний рейтинг, поради та відгуки про товар, статті про товар, які можуть зумовити Ваш вибір.

Витрати на роботу інтернет-магазину, включаючи доставку, значно нижчі ніж у звичайного. На відміну від звичайного магазину, інтернет-магазин здатний обслужити кілька сотень клієнтів одночасно. Хоча практично такого не зустрінеш. Крім того, якщо покупець мешкає в іншому місті, він отримує можливість не витратитися на міжміські телефонні консультації. Всю інформацію він може знайти на сторінках інтернет-магазину.

До недоліків інтернет-магазинів для покупця належать:

- не можна "помацати", не можна дізнатися більше, ніж написано (приклад: меблі, одяг);
- проблеми гарантії, супроводу;
- часто довга доставка (до 6 тижнів), особливо якщо ми говоримо про доставку до інших країн

## **1.2. Система керування сайтами Joomla**

Joomla є набір скриптів, написаних мовою програмування PHP. Ця мова була спеціально розроблена для написання веб-додатків.



Рис.1.2

У web-додатках існують дві сторони виконання коду – серверна та клієнтська. До клієнтської частини відносяться HTML, CSS, Javascript, а серверної ASP, JAVA, PHP і т.д. Виходячи з цього, для роботи (виконання) скриптів Joomla, потрібна наявність web-сервера з підтримкою PHP і MySQL (рекомендується Apache версії 1.3 і старше) і web-браузера у користувача (найпопулярніші - Internet Explorer, Mozilla Firefox, Opera).

Характеристики Joomla:

- повністю заснований на БД двигун із використанням PHP/MySQL;
- модуль безпеки для багаторівневої аутентифікації користувачів/адміністраторів;
- секції новин, продукції або послуг легко редаговані та керовані;
- розділи тем можуть бути додані авторами;
- схеми розташування елементів, що повністю настроюються, включаючи лівий, правий і центральний блоки меню;

- завантаження зображень за допомогою браузера у вашу власну бібліотеку для подальшого використання з будь-якого місця сайту;
- форум/опитування/голосування для ефективного зворотного зв'язку;
- робота під Linux, FreeBSD, MacOSX, Solaris, AIX, SCO, WinNT, Win2K.

Можливості адміністрування:

можливість створювати практично необмежену кількість сторінок;

- чітка організація структури сайту;
- для кожної динамічної сторінки можна створити свій опис та ключові слова для підвищення рейтингу в пошукових системах;
- початок та закінчення публікації будь-яких матеріалів можна запрограмувати за календарем;
- можливість обмежити доступ до певних розділів сайту лише для зареєстрованих користувачів;
- секції новин, продукції або послуг легко редаговані та керовані;
- схеми розташування елементів по п'яти областях, що повністю настроюються, що дозволяє зробити оригінальний дизайн;
- завантаження зображень за допомогою браузера у вашу власну бібліотеку для подальшого використання з будь-якого місця сайту;
- опитування та голосування для ефективного зворотного зв'язку;
- можливість підключення різних модулів (наприклад, останні новини, лічильник відвідувань, докладна статистика відвідувань, гостьова книга, форум тощо);
- можливість створення однієї, а кількох форм зворотний зв'язок кожного контакту;

- зміна порядку об'єктів, включаючи новини, питання, статті тощо;
- генератор показу випадкової новини;
- модуль прийому від віддалених авторів новин, статей та посилань;
- ієрархія об'єктів — кількість секцій, розділів, підрозділів та сторінок є настроюваною;
- менеджер розсилки новин;
- менеджер архіву, де можна зберігати зняті з публікації статті;
- можливість роздрукувати або надіслати по e-mail будь-яку статтю із сайту;
- вибір із кількох візуальних редакторів, що спрощує редагування матеріалів рівня редагування тексту у програмі Word;
- можливість попереднього перегляду перед остаточним розміщенням;
- можливість легкої зміни дизайну;
- економне використання місця на сервері за рахунок використання бази даних MySQL;
- можливість використання адрес сторінок адаптованих для кращої індексації всіма пошуковими системами

На сьогоднішній день можна знайти більше 3000 модулів і компонентів, які задовольняють вимоги будь-якого розробника сайту. Система підходить як для невеликого, так і великого корпоративного сайту або інформаційного проекту.

### **1.3. Система керування сайтами WordPress**

Система управління сайтами WordPress одна з найпопулярніших систем в інтернеті. І для цього є кілька причин, які можна поділити на дві категорії – SEO та користувальницькі.



Рис.1.3

Причини популярності WordPress серед SEO - спеціалістів - це насамперед чудова індексація пошуковими системами. Також, тому що WordPress безкоштовно існує безліч шаблонів і плагінів для неї у безкоштовному завантаженні. Ще один приємний момент системи WordPress це те, що php код можна редагувати прямо з «адмініки», тобто Вам не потрібно заходити на ftp сервер Вашого віддаленого сервера. Система має інтуїтивно зрозумілий інтерфейс, величезну кількість шаблонів, що легко настраюються. Саме тому блоги на WordPress такі популярні. Але, на жаль, керувати за допомогою WordPress легко лише блогами та невеликими порталами, для управління великими порталами потрібно вручну переробляти адмінку, а це складно і довго. Тому для великих порталів краще вибрати інший CMS.

#### Можливості WordPress:

- інформаційна система дозволяє створювати публікації за допомогою сторонніх програм;
- встановлення та налаштування не становлять жодної складності (див. розділ вище);
- підтримує функції створення списків оновлень RSS, Atom, trackback, pingback;

- унікальна простота системи взаємодії з кодом модулів (плагінів), що підключаються;
- реалізація «тем», як наборів файлів – шаблонів (на PHP), дуже помітно збільшує швидкість та гнучкість;
- існує можливість легко змінювати зовнішній вигляд та способи виведення даних за рахунок підтримки «тем»;
- невичерпні обсяги бібліотек «тем» та «плагінів»;
- доступна реалізація складних рішень з допомогою закладеного потенціалу архітектури;
- пропонується використання людино-зрозумілих URL;

#### **1.4. Структура інтернет-магазину**

Основне призначення інтернет-магазину – це продаж товару або послуг, а значить, отримання прибутку. При цьому переваги електронних торгових майданчиків перед реальними цілком очевидні.

Для споживача це:

- зручна навігація та пошук за каталогом наданої продукції;
- детальна інформація, характеристики та особливості запропонованого продукту;
- можливість порівняльного аналізу;
- автоматичний підрахунок суми замовлення, знижки;
- Історія замовлень;
- доставка додому;
- вибір найзручнішого способу оплати.

Важливо, щоб головна сторінка сайту задовольняла вимогам відвідувачів:

- знайти необхідний товар;
- Швидко його купити, не відволікаючись на зайві дії.

В інтернет-магазинах головна сторінка формується з певними особливостями, пов'язаними з Інтернетом:

- Яскрава або тематична шапка;
- Зручне та зрозуміле меню;
- вітальна або вступна інформація про компанію.

Більшість замовників інтернет-магазинів мають стереотип, що шаблонні або стандартні рішення – це погано. Насправді, це не завжди так. Часто буває навпаки, коли такі сайти виходять якіснішими та працездатними, ніж унікальні, нестандартні рішення. І відбувається це з таких причин.

Стандартна сітка розміщення елементів сайту. Для шаблонних сайтів зазвичай використовується перевірена на тисячах сайтів сітка розміщення елементів (меню, шапка сайту, інформаційний зміст). Користувач, приходячи на сайт, вже інтуїтивно розуміє, де потрібно шукати якусь інформацію.

Стандартні програмні модулі протестовані багато разів на різних проектах і з великою ймовірністю коректно працюватимуть і на будь-якому новому сайті, чого не можна з упевненістю сказати про нестандартні програмні рішення.

Так, для стандартного варіанту є обмеження щодо дизайну, але якщо для покупця вирішальним є ціна або просто наявність товару чи послуги, оригінальність дизайну відходить на другий план. Звичайно, у будь-якому випадку, сайт має бути охайним і приємним оку.

При розробці стандартного сайту немає тривалого етапу проектування, вигадкування оригінальної графічної ідеї та нестандартної подачі матеріалу. Все робиться за стандартами, при цьому все робиться швидко та добре. В результаті сайти виходять інтуїтивно зрозумілими всім категоріям користувачів.

### **1.5.Методи оплати**

Традиційні методи оплати, включаючи готівку, банківські перекази, чеки, пластикові картки, винайдені задовго до електронної комерції. Тому немає



нічого дивного в тому, що вони не повністю відповідають її потребам. Безумовно, при купівлі звичайного "фізичного" товару, гроші можна взяти з покупця при доставці. Але якщо купується "цифровий" товар або інформація, обов'язково має бути спосіб сплатити покупку прямо "на місці", тобто на сайті продавця. Ось чому в електронній комерції надзвичайно гостро стоїть питання розробки універсального способу оплати покупок в Інтернет, який дозволив би робити дешеві та безпечні платежі в режимі реального часу.

До способів оплати покупок в Інтернет, які проводяться електронним шляхом, можна віднести такі:

1. Оплата з рахунку мобільного телефону (m-commerce);
2. Оплата за картами передоплати
3. Оплата електронними грошима;
4. Оплата платіжними картками;
5. Оплата з рахунку мобільного телефону
6. Оплата готівкою після отримання товару



Рис.1.4

Мобільна комерція це використання мобільних портативних пристроїв для спілкування, розваги, отримання та передачі інформації, здійснення транзакцій через громадські та приватні мережі.

Скретч-карти є картами передоплати, оскільки, купуючи їх, ви робите передоплату сервісу або товару, який збираєтеся замовити. Свою назву вони отримали від англійського слова "scratch", що означає дряпати. Саме наявність особливого захисного шару приховує буквено-цифровий набір є відмінною особливістю карт передоплати. Дуже важливо відзначити, що кожна картка передоплати має свій унікальний серійний номер, який використовуватиметься для оплати послуг.

Карти передоплати випускаються різними фірмами, бувають різного номіналу та різного призначення. Як правило, скретч-карта має обмежений термін дії, який може змінюватись в залежності від номінальної вартості картки

або інших умов. Сама карта передоплати являє собою пластикову картку з розміщеною на ній інформацією щодо фірми виробника, сфери її застосування та терміну її дії.

Карти передоплати набули досить широкого поширення, оскільки є зручним способом оплати послуг та покупок. Їх активно використовують компанії, що займаються стільниковим зв'язком, наданням послуг Інтернет-провайдерів тощо.

Цифрові гроші є основою платіжної системи Інтернету. Вони можуть бути класифіковані за видами організації їхнього функціонування, за рівнем безпеки, а також за способом розрахунку.

Цифрову готівку можна уявити як файли-жетони, що замінюють готівку. Продавці та покупці можуть вільно обмінюватися цими "монетами" через мережу, оплачуючи ними товари та послуги. Для зазначеної мети учасники системи встановлюють у себе на комп'ютерах особливу програму - "електронний гаманець", який забезпечує облік та передачу жетонів, а також перевірку їхньої справжності. Цифрові гроші можуть необмежено довго звертатися в мережі, але також можуть будь-коли бути обмінені на реальні гроші в організаторів системи або в банках, що беруть участь у ній.

Різновидом електронних грошей є системи цифрових готівки, що базуються на використанні технології smart-card.

Сучасна смарт-карта – це маленький комп'ютер зі своїм процесором, пам'яттю, програмним забезпеченням та системою введення/виведення інформації. У чіпі такої картки зберігається не що інше, як електронні гроші в описаному вище сенсі.

Поки що смарт-карта використовується як звичайна дебетова карта (називається електронним гаманцем), в яку вносяться записи про списання грошей, або просто інформація про клієнта. Готівкові цифрові гроші на базі смарт-карток не тільки можуть забезпечити необхідний рівень конфіденційності та анонімності, але й не вимагають зв'язку з центром для

підтвердження оплати на відміну від подібних систем на базі персонального комп'ютера. У зв'язку з цим вартість транзакції прагнуть до нуля, за винятком поповнення картки або переведення в готівку грошей. Єдина незручність полягає в тому, що для переказу електронних грошей з картки на комп'ютер або назад необхідний спеціальний пристрій для читання карток, рідер, що приєднується до комп'ютера.



Рис.1.5

Кредитна картка - іменний платіжно-розрахунковий документ у вигляді пластикової картки, що видається банком своїм вкладникам для безготівкової оплати ними товарів та послуг у роздрібній торговельній мережі, з комп'ютерними пристроями, що передають запит на оплату товару до банку.

Кредитна картка засвідчує наявність у її власника поточного рахунку у банку. У різних країнах використовують кредитні картки американських кредитно-фінансових груп типу Visa, MasterCard, American Express і т.д.

Дебетові картки можуть використовуватися при оплаті товарів та послуг через Інтернет у режимі онлайн так само, як при отриманні готівки в банкоматі: для здійснення платежу клієнт повинен запровадити номер картки та PIN-код.

Оскільки використання класичної кредитної картки для розрахунків в Інтернет є небезпечним, було розроблено віртуальні картки типу VISA E-c@rd та Virtual MasterCard, які призначені виключно для розрахунків в Інтернет.

VISA E-c@rd розроблена за технологією міжнародної платіжної системи VISA Int. Virtual MasterCard розроблена за технологією міжнародної платіжної системи MasterCard Int.

Випуском подібних карток сьогодні займаються різні банки своєчасно оцінивши їх переваги та потенціал. VISA E-c@rd та Virtual MasterCard застосовується виключно для оплати через Інтернет товарів та послуг в електронних магазинах по всьому світу, а також послуг операторів мобільного зв'язку, провайдерів Інтернету, туристичних фірм та готелів.

### **1.6. Система інтернет платежів**

У основі кредитних платіжних систем лежить використання кредитних карток. При разових покупках на "електронному ринку" картка діє так само, як і при звичайній покупці в магазині: клієнт купує товар або послугу і передає продавцю для оплати номер своєї кредитної картки, тільки відбувається все через Інтернет. Такий спосіб передачі даних зобов'язує до застосування додаткових заходів безпеки (шифрування обміну повідомленнями, цифровий підпис тощо).

Схема проведення платежів через Інтернет має такий вигляд:

1. Покупець на сторінці електронного магазину вибирає товар чи послугу, формує «кошик» покупок та вибирає спосіб оплати "кредитна картка".
2. Реквізити картки (номер, ім'я власника, дата закінчення дії) передаються платіжній системі Інтернет для авторизації. При цьому параметри кредитної картки можуть вводитись як на сайті магазину (після цього вони будуть передані на сервер платіжної системи), так і безпосередньо на сервері платіжної системи. Для покупця другий спосіб вважається безпечнішим. У цьому випадку знижується ризик витоку відомостей про карту.
3. Платіжна система Інтернет передає запит на авторизацію в процесинговий центр платіжної системи, який і здійснює авторизацію картки.
4. Результат авторизації передається платіжній системі Інтернет.
5. Продавець та Покупець отримують результат авторизації.

б. За позитивного результату авторизації:

магазин надає послугу або відвантажує товар;

процесинговий центр передає до розрахункового банку відомості про скоєну транзакцію. Гроші з рахунку покупця у банку-емітенті перераховуються через розрахунковий банк на рахунок магазину у банку-еквайєрі.

Безперечними перевагами кредитної схеми платежів є:

Звичність для клієнтів та правова визначеність;

Досить висока захищеність конфіденційної інформації з допомогою використання протоколу <https>. Відповідно до цього протоколу номер картки, що передається через мережу, шифрується. Дешифрування зможуть здійснювати лише уповноважені банки та процесингові компанії. Цей протокол має забезпечити захист клієнтів від недобросовісних продавців та захист продавців від шахрайства за допомогою підроблених чи крадених карток.

Однією з основних проблем, пов'язаних із платежами за кредитними картками, є питання безпеки. Транзакція за кредитною картою через Інтернет з погляду платіжних систем мереж є "транзакцією без фізичної присутності власника", і як наслідок того, підвищується ризик можливості перехоплення особистої та банківської інформації під час транзакції.

Серед інших недоліків цих систем можна назвати:

Необхідність перевірки кредитоспроможності клієнта та авторизації картки, що підвищує витрати для проведення транзакції та робить системи непристосованими для мікроплатежів;

Відсутність анонімності і, як наслідок, нав'язливий сервіс із боку торгових структур.

Дебетні схеми платежів через Інтернет побудовані за аналогією з їх офлайнними попередниками - чековими та звичайними готівковими розрахунками. У розрахунках беруть участь емітент та користувач. Емітент випускає деякі електронні одиниці (цифрові еквіваленти грошей на рахунках

банках чи паперових чеків). Користувач здійснює платежі через Інтернет, використовуючи дані електронні одиниці.

Електронні чеки є аналогом стандартних паперових чеків, тобто. по суті — це розпорядження платника своєму банку перерахувати гроші зі свого рахунку на рахунок одержувача платежу. Власник чекової книги підписує електронний чек електронним підписом на необхідну суму та передає його будь-яким доступним способом одержувачу. Одержувач пред'являє цей чек до оплати платіжної системи. Якщо перевірка підтверджує справжність чека, платник отримує товар чи послугу, а гроші перераховуються з рахунку платника з цього приводу одержувача.

Подібні схеми платежів прості і давно застосовуються за кордоном (NetCash, NetCheq, NetCheque), але для України вони поки що не надто актуальні, тому що, насамперед, відсутня широка практика використання чеків навіть за офлайнових розрахунків. Однією з перших ластівок у цій сфері електронних платежів у нашій країні є система PAYMER, в якій як розрахунковий засіб використовуються цифрові чеки.

Системи цифрової готівки базуються на використанні електронних грошей. Вони є грошовими знаками, які емітовані у формі електронних документів і звертаються до Інтернету. Цифрові гроші функціонують так само, як і їх аналог – банкноти та монети, лише у вигляді файлів, а емітуються як банками, так і небанківськими організаціями. Електронні гроші можуть зберігатися, переноситися та використовуватися за допомогою спеціальних пристроїв та/або програмних засобів на звичайних персональних комп'ютерах.

## **2. Огляд сучасних засобів та технологій розробки інтернет магазинів та мобільних додатків**

### **2.1. Переваги мобільного додатку**

У сучасному світі важко представити людину без смартфона. Цей девайс перетворився з предмету розкоші в предмет побутового використання. Сучасні смартфони дозволяють використовувати їх у якості телефону, камери, будильника, записної книжки, калькулятора, засобу для перегляду відео та

інтернет сторінок. Завдяки програмам ми можемо контролювати свою рухову активність, з'являється можливість стежити за харчуванням і заносити всі дані в спеціальну програму. Люди проводять більше часу в смартфоні, а розумна контекстна реклама дозволяє краще підбирати актуальну рекламу для кожної людини особисто, а сповіщення нагадують користувачеві про нові надходження товарів та розпродажі. Недооцінювати мобільний сегмент ринку однозначно не варто. Згідно з останніми отриманими показниками кількість користувачів, які віддають перевагу мобільному додатку замість сайту, перевищує 70%.



Рис.2.1

Користувачі, які завантажили мобільний додаток вашого магазину, вже є більш лояльними до вашої компанії і готові до покупок. У мобільних додатках клієнт витрачає в середньому в 3 рази більше часу ніж на сайтах. Маючи іконку вашого магазину на своєму телефоні, клієнт найімовірніше віддасть перевагу саме вашому магазину, не витрачаючи час на пошуки потрібного товару в інтернеті. Що, як показує статистика, змушує клієнта знову зробити у вас покупку набагато частіше, ніж якби він користувався сайтом.

Мобільний додаток завдяки своїй адаптивності на iOS і Android завантажується набагато швидше на всіх пристроях. У той час як зайшовши на сайт через телефон, він буде чекати кілька секунд і швидше за все закриє вкладку вашого магазину.



## 2.2. Аналіз області. Огляд сучасних технологій

Необхідність створення мобільного додатка та інтернет-магазину для пекарні обумовлено тим, що дані пристрої дозволяють охопити більше велику частину ринку, тобто отримати більше прибутку, що є з думкою для всіх типів підприємств. Хоча за останніми даними дослідницької компанії Strategy Analytics операційна система iOS знаходиться на підйомі, у той час як Android поступається своїми позиціями, саме операційною системою від Google користуються 81,2% користувачів на глобальному мобільному ринку за підсумками минулого року. Але крім Google Android та iOS існують і інші мобільні операційні системи:

- OS Symbian офіційно є власністю Nokia. Це означає, що будь-яка інша компанія повинна буде отримати дозвіл від Nokia перед використанням цієї операційної системи. Nokia, як і раніше, є гігантом у недорогому сегменті ринку мобільних телефонів, тому Java Symbian найбільш часто використовувалася в мобільних телефонах пару років назад. Тим не менш, Symbian широко використовується в бюджетних апаратах, але попит ними постійно знижується. Оновлення мобільної OS Symbian зробило його здатним працювати ефективно на смартфонах;
- BlackBerry OS є власністю компанії RIM (Research In Motion) і вперше була випущена у 1999 році. RIM розробила цю операційну систему для своєї BlackBerry лінії смартфонів. BlackBerry сильно відрізняється від інших операційних систем. Стильний інтерфейс, а також оригінальний дизайн телефону та клавіатура QWERTY. Як і Apple, BlackBerry OS не доступна для будь-яких інших виробників;
- Як і інші Samsung також володіє операційною системою, яка відома як BADA. Bada є зручною та ефективною операційною системою, так само, як Android, але, на жаль, Samsung не використовує OS Bada в великих масштабах з невідомих причин;
- Palm OS була розроблена Palm Inc у 1996 році спеціально для кишенькових комп'ютерів (Personal Digital). Palm OS була в основному призначена для

роботи на сенсорному інтерфейсі. Кілька років по тому він був використаний у деяких смартфонах. На жаль, він не зміг закріпитися на ринку, і в даний час не використовується

Вибір платформи для мобільного додатка залежатиме від кількості проданих пристроїв на даній платформі

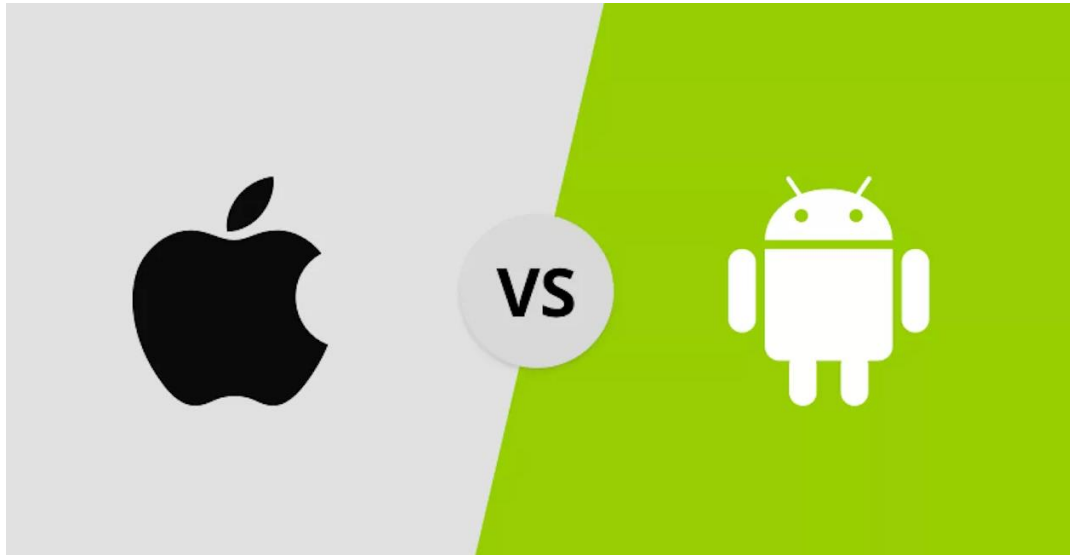


Рис.2.2

На таблиці рис.2.2 видно, що лідером продажів є операційна система від Google Android. На кінець 2021 року ця операційна система займала 71% ринку.

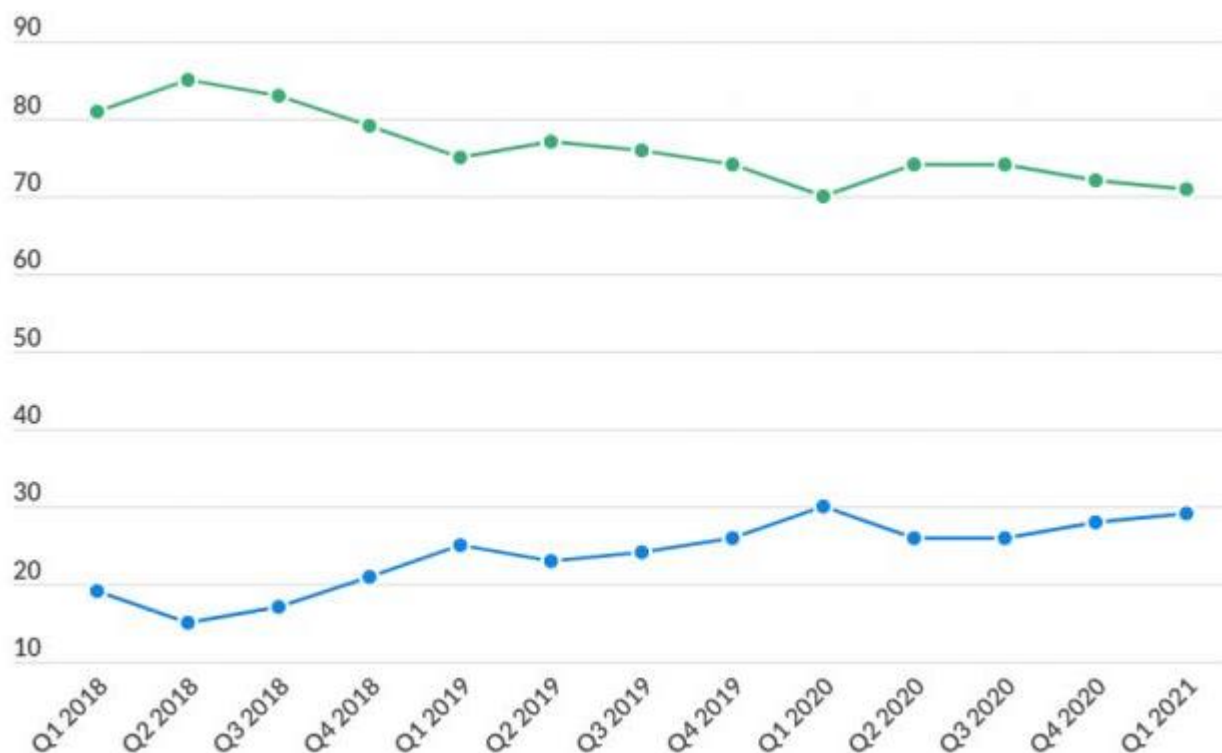


Рис.2.3

### 2.3. Вибір засобів та технологій розробки інтернет-магазину

Вибір засобів розробки залежатиме від веб-сервера на якому буде розміщуватися сайт інтернет-магазину. Тобто всі кошти будуть обрані в зв'язку з тим, чи є у даного хостингу ця технологія. Вибір хостингу відбуватиметься серед серверів, що знаходяться у ТОП – 10. Далі будуть виділені такі показники:

- надійність сайту є одним з найважливіших факторів, вибору хороший веб-хостинг. Найкраще подивитися відгуки про хостинг на незалежних ресурсах, перед тим як розміщувати на ньому сайт;

- швидкість роботи хостингу безпосередньо впливає на роботу розміщеного на ньому сайту. Оцінити швидкість роботи хостингу до його використання можна за відгуками на незалежних ресурсах;

- можливості хостингу полягають у тому, наскільки сервер повно підтримує різні засоби. Поки що кількість трафіку і місце на диску є ключовими факторами порівняння функціональності, інші можливості, пропонувані хостингом, теж не повинні залишатися нерозглянутими;

– техпідтримка – це сервісна структура, що вирішує проблеми користувачів є стандартом для багатьох компаній. Безперечним плюсом вважається наявність онлайн тикетів на хостингу

На основі цих показників було прийняте рішення використовувати хостінг Google із використанням Firebase.

Firebase – це платформа розробки мобільних додатків з величезним функціоналом. Починалася вона як стартап, а сьогодні її використовують при розробці кращих кроссплатформених додатків. Головне достоїнство платформи в тому, що вона дозволяє розробнику не відволікатися на створення бекенд, тобто прихованої від користувача програмної частини проекту, наприклад, серверного коду. І це спрощує і прискорює створення мобільних додатків, дає можливість повністю зосередитися саме на UX / UI, тобто, на призначеному для користувача інтерфейсі і досвіді. Саме зв'язка Firebase з фреймворком Flutter дозволяє програмістам компанії AVADA MEDIA створювати швидкі програми для Android і iOS, що дозволяють вирішувати найрізноманітніші завдання.



Рис.2.4

Firebase надає такий функціонал:

Firebase Analytics – безкоштовне рішення для оцінки застосунків, яке дає змогу ознайомитись із використанням застосунків та залученням користувачів

Firebase Cloud Messaging (FCM) – це крос-платформове рішення для повідомлень і нотифікацій для Android, iOS та веб-застосунків, які наразі можна використовувати безкоштовно

Firebase Auth – це служба, яка може аутентифікувати користувачів, використовуючи лише код на стороні клієнта. Він підтримує соціальні логін-провайдери Facebook, GitHub, Twitter і Google (і Google Play Games). Крім того, вона включає в себе систему управління користувачами, за допомогою якої розробники можуть увімкнути автентифікацію користувача за допомогою входу з електронної пошти та пароля, що зберігаються в Firebase

Firebase надає в режимі реального часу базу даних та бекенд як службу. Ця служба надає розробникам застосунків API, який дозволяє синхронізувати дані застосунків між клієнтами та зберігати їх у хмарі Firebase. Компанія також надає клієнтські бібліотеки, які дозволяють інтеграцію із застосунками Android, iOS, JavaScript / Node.js, Java, Objective-C, Swift. База даних також доступна через REST API та прив'язки до декількох сценаріїв JavaScript, таких як AngularJS, React, Ember.js та Backbone.js. REST API використовує протокол подій із сервером, який є інтерфейсом для створення HTTP-з'єднань для отримання push-повідомлень від сервера. Розробники, які використовують Realtime Database, можуть захищати свої дані за допомогою правил безпеки, що застосовуються на сервері

Firebase Storage забезпечує надійне завантаження та вивантаження файлів для застосунків Firebase незалежно від якості мережі. Розробник може використовувати його для зберігання зображень, аудіо-, відео- чи іншого вмісту, створеного користувачами. Зберігання Firebase підтримується Google Cloud Storage

Firebase Hosting – це статичний та динамічний веб-хостинг, який було запущено 13 травня 2014 року. Він підтримує хостинг статичних файлів, таких як CSS, HTML, JavaScript та інші файли, а також динамічну підтримку Node.js через Cloud Functions. Служба передає файли через мережу доставки контенту

(CDN) за допомогою протоколу HTTPS та шифрування SSL. Firebase підтримує Fastly, CDN, щоб забезпечити підтримку CDN Firebase Hosting. Компанія стверджує, що хостинг Firebase виріс із запитів клієнтів; розробники використовували Firebase для своєї бази даних в режимі реального часу, але вони потребували місця для розміщення їхнього вмісту.

ML Kit – це мобільна система машинного навчання для розробників, яка була запущена в режимі бета-тестування 8 травня 2018 року під час Google I/O 2018. ML Kit API містить різноманітні інструменти, серед яких розпізнавання тексту, розпізнавання облич, сканування баркодів, створення опису для зображень та розпізнавання наземних об'єктів. Наразі вона доступна для iOS та Android розробників. Також можливий імпорт власних моделей TensorFlow. API можна використовувати у пристрої або у хмарі.

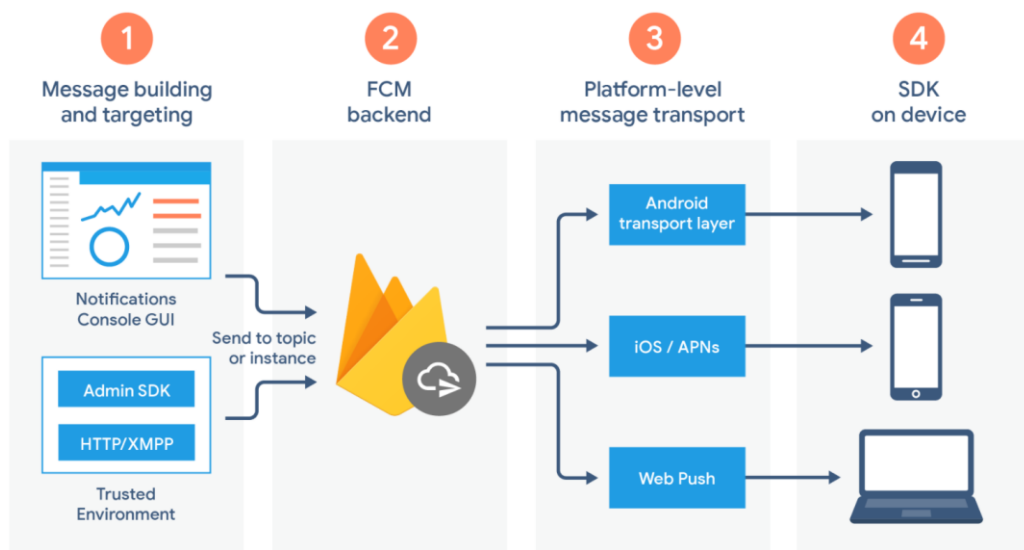


Рис.2.5

Firebase – це платформа розробки мобільних додатків з величезним функціоналом. Починалася вона як стартап, а сьогодні її використовують при розробці кращих кроссплатформених додатків. Головне достоїнство платформи в тому, що вона дозволяє розробнику не відволікатися на створення бекенд, тобто прихованої від користувача програмної частини проекту, наприклад, серверного коду. І це спрощує і прискорює створення мобільних додатків, дає можливість повністю зосередитися саме на UX / UI, тобто, на

призначеному для користувача інтерфейсі і досвіді. Саме зв'язка Firebase з фреймворком Flutter дозволяє програмістам компанії AVADA MEDIA створювати швидкі програми для Android і iOS, що дозволяють вирішувати найрізноманітніші завдання.

Також Firebase виступає в якості сховища файлів. Firebase Storage забезпечує надійну завантаження і вивантаження файлів для додатка. Хмарне зберігання файлів відео, аудіо або будь-якого іншого типу підтримується Google Cloud Storage. Вміст хмарного сховища надійно захищене власною системою безпеки.

При створенні нового мобільного застосування в AVADA MEDIA взагалі багато уваги приділяється питанням безпеки. Створювати систему аутентифікації кожен раз з нуля досить затратно, причому витрати ці найчастіше не виправдані. Чи впорається з більшістю викликів дозволяє система аутентифікації Firebase Auth, в якій можлива аутентифікація користувача додатки по паролю і електронній пошті. Підтримує Firebase Auth також відкритий протокол авторизації OAuth 2.0, який використовується Google, Twitter, Facebook. Система аутентифікації Firebase інтегрується безпосередньо в базу даних.

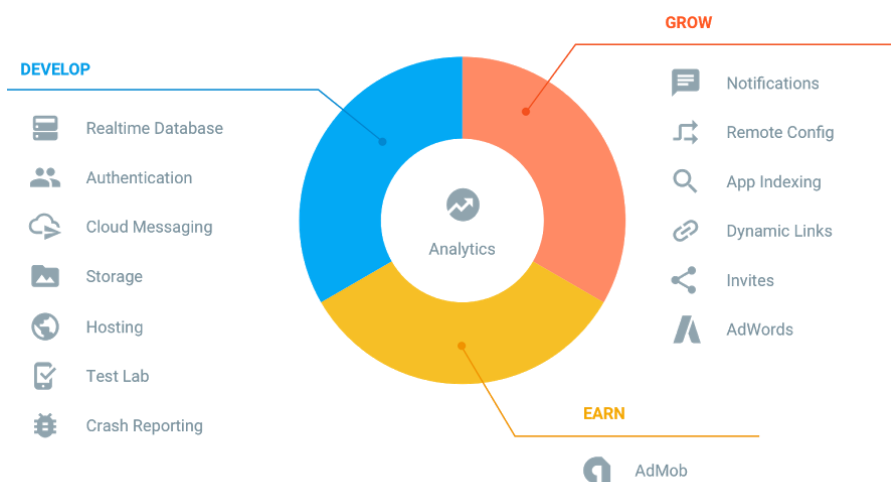


Рис.2.6

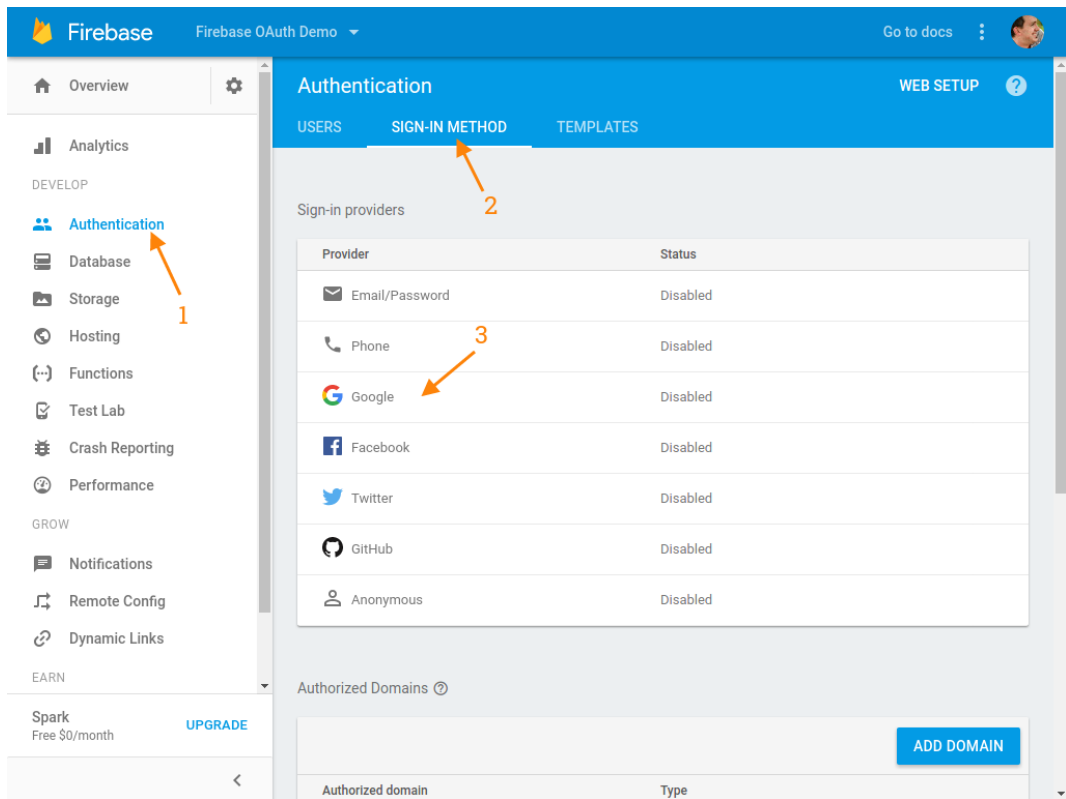


Рис.2.7

## 2.4. Вибір середовища розробки

Серед технологій розробки мобільних додатків виділяється без умовний лідер Android Studio(рис.2.3) – це нове та повністю інтегроване середовище розробки додатків, нещодавно випущена компанією Google для операційної системи Android. Цей продукт покликаний забезпечити розробників новими інструментами для створення додатків, а також надати альтернативу Eclipse, що є в даний час найбільш популярним середовищем розробки.



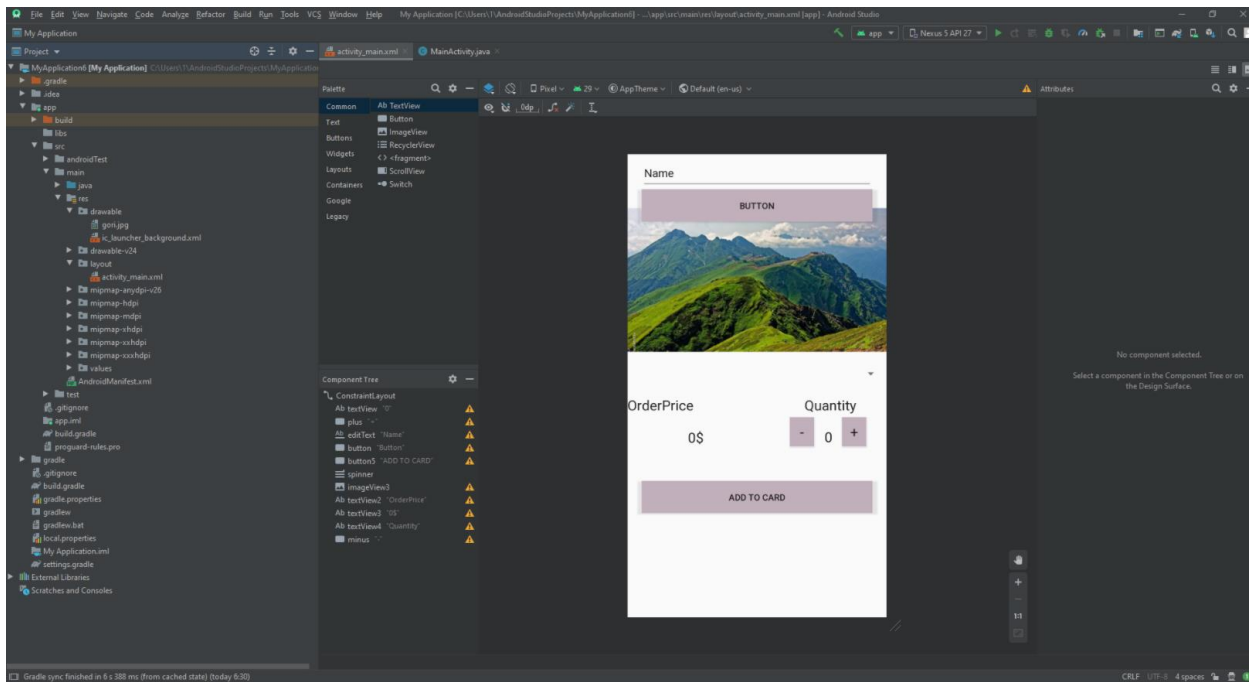


Рис.2.8

При створенні нового проекту в Android Studio, буде показано структуру проекту з усіма файлами, що містяться у каталозі SDK. Цей перехід до системи управління Gradle надає процесу розробки ще більшої гнучкості. Android Studio і дозволяє побачити будь-які візуальні зміни, які виробляються реальному часі у додатку. Також можна побачити, як додаток буде одночасно виглядати на різних пристроях під керуванням Android, з різними налаштуваннями та роздільною здатністю екрану.

Продукт також володіє новими інструментами для упаковки та марки коду. Це дозволяє не загубитися в проекті, коли є справа з великою кількістю коду. У програмі також задіяна функція перетягування, завдяки якій можна переміщати компоненти за допомогою інтерфейсу користувача. До того ж, нове середовище розробки має функцією Google Cloud Messaging, яка дозволяє надсилати дані з сервера на Android-пристрої через хмару. Це відмінний спосіб посилати push повідомлення додатків. Також можна за допомогою програми локалізувати програми. Це дозволяє програмувати і при цьому зберігати контроль над додатком. З усіх вищеперелічених середовищ розробки більш підходящою є Android Studio, так як:

– є надійним та простим середовищем розробки;

- легко перевіряє продуктивність програми на різних типах пристроїв;
- присутні помічники та шаблони для загальних елементів програмування для Android;
- є повнофункціональний редактор з безліччю додаткових інструментів, що сприяють прискоренню розробки програм;
- є шаблони основних макетів і компонентів Android;
- вбудована підтримка Google Cloud Platform, яка включає в себе інтеграцію із сервісами Google Cloud Messaging та App Engine;
- підтримує Android N Preview SDK, а це означає, що розробники зможуть розпочати роботу зі створення програми для нової програмної платформи;
- Є розширений редактор макетів: WYSIWYG, здатність працювати з UI компонентами за допомогою Drag-and-Drop, функція до перегляду макета на декількох конфігураціях екрана

## **2.5. Вибір мови програмування**

Для розробки мобільного додатку потрібно обрати мову програмування. Але постає питання для роботи на популярних сьогодні ОС Android та IOS. Вони дуже відрізняються і потрібно використовувати різні мови програмування для створення мобільного додатку для цих платформ. Обрати лише одну платформу означає залишитись без значної кількості потенційних покупців. Раніше розробляли дві версії одного додатку для двох платформ. Це збільшувало час та ціну розробки, т.к. як є спільні частини, які можна використовувати в обох версіях, але багато чого треба було робити виключно під платформу.

Ця проблема довго турбувала розробників і вже були створені реактивні мови програмування, які дозволяли розробляти один догадок одразу для двох платформ, але вони мали більше недоліків ніж переваг. Але в 2018 році Google представила світові свій продукт під назвою Flutter. Flutter – це платформо-

незалежна технологія для створення додатків для мобільних, десктопних та веб-платформ. Як рендерить віджети Flutter(рис.1.1):

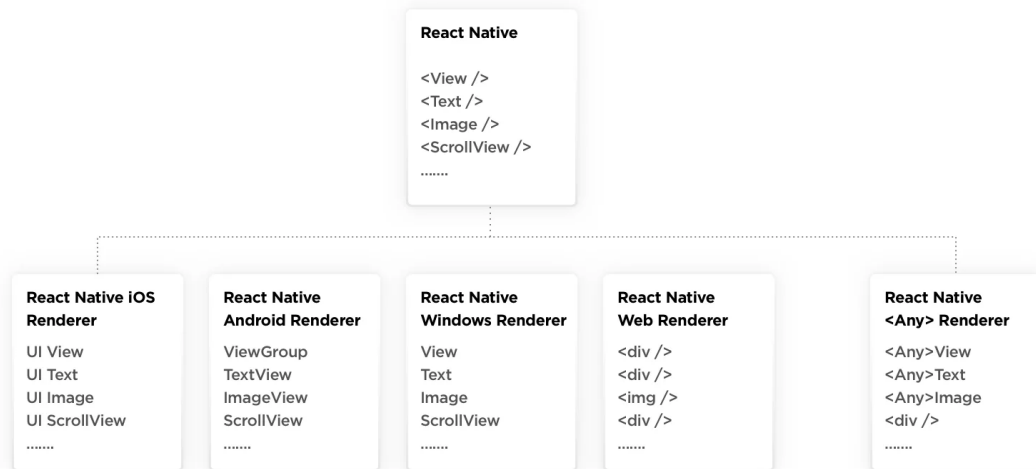


Рис.2.9

Такий вид процесу рендерингу спрощує створення програми, яка виглядає нативно на будь-якій платформі. Але, як завжди, всі складнощі криються в деталях. Використання компонентів, що залежать від платформи, для рендерингу викликає потребу в шарі зіставлення властивостей віджета платформи і синхронізації даних віджета платформи. Це те, що вимагає порівняння кожної анімації з викликом віджета для конкретної платформи. Набагато складніше, ніж має бути.

Навпаки, Flutter не потребує будь-яких компонентів інтерфейсу користувача для конкретної платформи для візуалізації інтерфейсу користувача. Єдине, що потрібно Flutter для відображення інтерфейсу програми, - це полотно, на якому можна малювати. А ось як це виглядає:

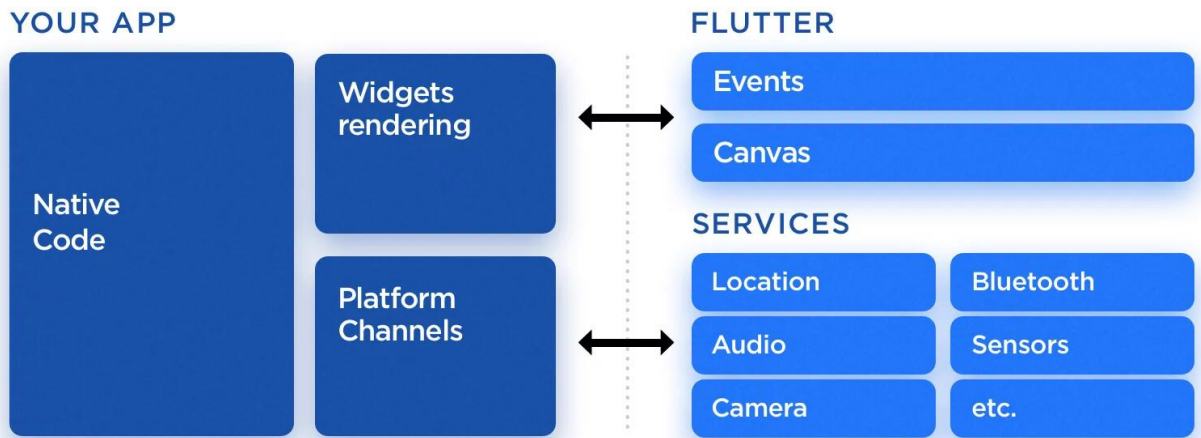


Рис.2.10

Flutter знижує ризики для вашого бізнесу:

проблем із розробкою продукту не виникне, тому що Google створює Fuchsia OS, яка відмінно працює з Flutter

використання Flutter не викличе проблем із пошуком інженерів, тому що серед розробників Android у спільноті вже дуже багато шанувальників (обговорення можна знайти тут і тут)

простота інтеграції: список глобальних компаній, що використовують Flutter, постійно поповнюється такими представниками як Alibaba, Google Ads, AppTree, Reflectly та My Leaf, що є доказом безкомпромісної якості Flutter.

Dart – мова програмування, яку розробляє компанія Google, позиціонуючи як мову структурованого програмування для Веб. Розробники вважали, що в довгостроковій перспективі Dart може стати прогресивною заміною JavaScript, котрий потерпає від наявних в даний час проблем з розширюваністю, продуктивністю і підтримкою розробки складних застосунків. Мова має схожий на Java синтаксис, не вимагає явного визначення типів і її можна використовувати для створення серверних та клієнтських застосунків.

У березні 2015 компанія Google представила оновлену стратегію просування Dart, у котрій вирішено не прив'язувати Dart до браузеру і

відмовитися від ідеї інтеграції віртуальної машини Dart у Chrome. Розробку буде зосереджено на застосуванні Dart як проміжної мови, скомпільованої в JavaScript. Розвиток Dart як окремої мови, альтернативної JavaScript і безпосередньо підтримуваної у браузерях, визнано недоцільним. Замість цього Dart рухатиметься у бік якіснішої інтеграції з JavaScript і генерації оптимального JavaScript-коду. При цьому розробку віртуальної машини Dart VM буде продовжено, але вона позиціонуватиметься в основному для створення серверних і мобільних застосунків.

Мова має схожий на Java синтаксис, не вимагає явного визначення типів і може використовуватися для створення серверних і клієнтських застосунків. Для запуску всередині браузера код мовою Dart може бути перетворений в JavaScript-подання або запущений безпосередньо під управлінням спеціального JavaScript-інтерпретатора Dartboard. Підтримується вбудовування коду мовою Dart в HTML-сторінки, використовуючи MIME тип "application/dart". На стороні сервера застосунок на мові Dart може бути виконаний всередині спеціальної віртуальної машини, яка забезпечує продуктивність виконання близьку до скомпільованих в машинний код мов. Віртуальну машину Dart планують інтегрувати в майбутні версії браузера Chrome, що дозволить виконувати застосунки мовою Dart без компіляції в JavaScript.

Мова підходить як для розробки одним програмістом невеликих скриптів без жорсткої структури, так і для створення високо масштабованих великих модульних проектів, підтримуваних великим колективом з потребою більш явної типізації для того, щоб уникнути плутанини і помилок. При цьому явне задання типів не обов'язкове, наприклад, можна почати розробку без вказання типів, а надалі при необхідності додати їх (наприклад, спочатку написати "var x", а потім замінити на "num x"). Код Dart завжди виконується тільки в рамках одної потоку, для організації паралельного виконання пропонується використовувати класи з атрибутом `isolate`. У кожному скрипті використовується власний простір імен, для використання зовнішніх об'єктів, функцій або змінних слід їх явно імпортувати за допомогою конструкції

"import". Всі змінні, початково, діють тільки в межах поточного скрипту і не експортуються глобально.

Для спрощення розробки мовою Dart поставляється SDK, який включає в себе компілятор dart2js, віртуальну машину Dart VM, пакетний менеджер pub, статичний аналізатор коду dart\_analyzer, і набір бібліотек. Для виконання і відлагодження застосунків на мові Dart, без компіляції в JavaScript, поширюється Dartium – складання браузера Chromium з інтегрованою віртуальною машиною Dart VM.

До складу SDK входить Dart Editor – спеціалізоване інтегроване середовище розробки мовою Dart. Dart Editor з одного боку досить легковаговий і не вимогливий до ресурсів, а з іншого підтримує такі елементи сучасних IDE, як рефакторинг, автодоповнення коду, перехід на визначення, контекстні підказки, містить вбудований відлагоджувач і аналізатор коду. Плагіни з підтримкою Dart також підготовлені для IntelliJ IDEA, WebStorm, Eclipse, Emacs і Vim.

#### Особливості мови Dart:

Звичний і простий для вивчення синтаксис, природний для програмістів на JavaScript, C і Java;

Забезпечення швидкого запуску і високої продуктивності для всіх сучасних веб-браузерів і різних типів оточень, від портативних пристроїв до потужних серверів;

Можливість визначення класів і інтерфейсів, що дозволяють використовувати інкапсуляцію і повторно використовувати існуючі методи і дані;

Необов'язкове вказування типів, використовувати чи ні статичні типи вирішує розробник. Вказування типів дозволяє спростити виявлення помилок, робить код яснішим і читаним, спрощує його доопрацювання та аналіз сторонніми розробниками.

Серед підтримуваних типів: різні види хешів, масивів і списків, черги, числові і рядкові типи, типи для визначення дати і часу, регулярні вирази (RegExp). Можливо створення своїх типів;

Для організації паралельного виконання пропонується використовувати класи з атрибутом `isolate`, код яких виконується повністю в ізольованому просторі в окремій області пам'яті, взаємодіючи з основним процесом через відправку повідомлень;

Підтримка використання бібліотек, що спрощують підтримку і зневадження великих веб-проектів. Сторонні реалізації функцій можуть підключатися у вигляді поділюваних бібліотек. Застосунки можна розбити на частини і доручити розробку кожної з частин окремій команді програмістів;

Набір готових інструментів для підтримки розробки мовою Dart, включаючи реалізацію засобів динамічної розробки та зневадження з виправленням коду на льоту ("edit-and-continue");

Можливість створювати однорідні системи, що охоплюють як клієнтську, так і серверну частину. Використання однієї мови та інструментарію для клієнтських і серверних компонентів спрощує процес кодування і позбавляє від постійної зміни контексту.

## 2.6.База даних

Метою розробки будь-якої бази даних є зберігання та використання інформації про будь-яку предметну область. Для реалізації цієї мети є такі інструменти:

- Реляційна модель даних – зручний спосіб подання даних предметної галузі;
- Мова SQL – універсальний спосіб маніпулювання такими даними.

Однак очевидно, що для однієї і тієї ж предметної області реляційні відносини можна спроектувати безліччю різних способів. Наприклад, можна спроектувати кілька відносин з великою кількістю атрибутів, або навпаки, рознести всі атрибути за великою кількістю дрібних відносин. При розробці бази даних зазвичай виділяється кілька рівнів моделювання, за допомогою яких відбувається перехід від предметної області до конкретної реалізації бази даних засобами конкретної СКБД. Можна, можливо виділити такі рівні:

- Сама предметна область;
- Модель предметної області;
- Логічна модель даних;
- Фізична модель даних;
- Власне база даних та додатки.

Предметна область – це частина реального світу, дані про який необхідно відобразити в базі даних. Предметна область нескінченна і містить як істотно важливі поняття та дані, так і малозначущі або взагалі не значущі дані. Модель предметної області – це знання предметної області. Знання можуть бути як у вигляді неформальних знань, так і виражені формально з допомогою будь-яких коштів. Як такі кошти можуть виступати текстові описи предметної галузі, набори посадових інструкцій, правила ведення справ у компанії тощо. Досвід показує, що текстовий спосіб представлення моделі предметної області вкрай



неефективний. Набагато більш інформативними та корисними при розробці баз даних є описи предметної області, виконані за допомогою спеціалізованих графічних нотацій. Є велика кількість методик опису предметної галузі. З найбільш відомих можна назвати методику структурного аналізу SADT і засновану на ньому IDEF0, діаграми потоків даних Гейна-Сарсона, методику об'єктно-орієнтованого аналізу UML, та ін. Модель предметної області описує швидше процеси, що відбуваються в предметній області та дані, що використовуються цими процесами. Від того, Наскільки правильно змодельована предметна область, залежить успіх подальшої розробки додатків.

Логічна модель даних. На наступному, нижчому рівні знаходиться логічна модель даних предметної області. Логічна модель описує поняття предметної галузі, їх взаємозв'язок, а також обмеження на дані, що накладаються предметною областю. Логічна модель даних є початковим зразком майбутньої бази даних. Логічна модель будується в терміни інформаційних одиниць, але без прив'язки до конкретної СУБД. Більше того, логічна модель даних необов'язково має бути виражена засобами саме реляційної моделі даних. Основним засобом розробки логічної моделі даних зараз є різні варіанти ER-діаграм (Entity-Relationship, діаграми сутність-зв'язок). Одну й ту саму ER-модель можна перетворити як на реляційну модель даних, так і в модель даних для ієрархічних та мережевих СУБД, або у постріляційну модель даних. Функціонал Firebase дозволяє використовувати свою базу даних.

Рішення, що були прийняті на кожному етапі моделювання та розробки бази даних, позначатимуться на подальших етапах. Тому особливу роль відіграє прийняття правильних рішень на ранніх етапах моделювання.

Атрибут – властивість певної сутності. Часто називається полем таблиці.

Домен атрибута – безліч допустимих значень, які можуть приймати атрибут.

Кортеж – кінцева множина взаємопов'язаних допустимих значень атрибутів, які разом описують деяку сутність (рядок таблиці).

Ставлення – кінцева множина кортежів (таблиця).

Схема відношення - кінцева безліч атрибутів, що визначають яку сутність. Іншими словами, це структура таблиці, що складається з конкретного набору полів.

Проекція – відношення, отримане із заданого шляхом видалення та (або) перестановки деяких атрибутів.

Функціональна залежність між атрибутами (множинами атрибутів)  $X$  і  $Y$  означає, що для будь-якого допустимого набору кортежів у даному відношенні: якщо два кортежі збігаються за значенням  $X$ , то вони збігаються за значення  $Y$ . Наприклад, якщо значення атрибута «Назва компанії» – Canonical Ltd, то значенням атрибуту «Штаб-квартира» у такому кортежі завжди буде Millbank Tower, London, United Kingdom. Позначення:  $\{X\} \rightarrow \{Y\}$ .

Нормальна форма - вимога, що пред'являється до структури таблиць теорії реляційних баз даних для усунення з бази надлишкових функціональних залежностей між атрибутами (полями таблиць). Метод нормальних форм (НФ) полягає у зборі інформації про об'єкти вирішення задачі в рамках одного відношення та подальшої декомпозиції цього відношення на кілька взаємопов'язаних відносин на основі процедур

нормалізації відносин. Мета нормалізації: виключити надмірне дублювання даних, яке є причиною аномалій, що виникли при додаванні, редагуванні та видаленні кортежів (рядок таблиці).

Аномалією називається така ситуація у таблиці БД, яка призводить до суперечності в БД або значно ускладнює обробку БД. Причиною є зайве дублювання даних у таблиці, яке викликається наявністю функціональних залежностей від не ключових атрибутів. Аномалії модифікації виявляються в тому, що зміна одних даних може спричинити перегляд всієї таблиці і відповідна зміна не якихось записів таблиці. Аномалії видалення – при видаленні будь-якого кортежу з таблиці може пропасти інформація, яка не пов'язана безпосередньо з записом, що видаляється. Аномалії додавання

виникають, коли інформацію до таблиці не можна помістити, поки вона не повна, або вставка запису вимагає додаткового перегляд таблиці.

Перша нормальна форма (1НФ). Відношення знаходиться в 1НФ, якщо всі його атрибути є простими, всі використовувані домени повинні містити лише скалярні значення. Не повинно бути повторень рядків у таблиці. Згідно з визначенням відносин, будь-яке відношення автоматично вже знаходиться у 1НФ. Властивості стосунків це і будуть властивості 1НФ:

- щодо немає однакових кортежів;
- Кортежі не впорядковані;
- атрибути не впорядковані та розрізняються за найменуванням;
- Усі значення атрибутів атомарні.

Друга нормальна форма (2НФ). Відношення знаходиться в 2НФ, якщо воно знаходиться в 1НФ і кожен не ключовий атрибут невідомо залежить від Первинний Ключ (ПК). Неприводимість означає, що у складі потенційного ключа відсутня менше підмножина атрибутів, якого можна також вивести цю функціональну залежність.

Третя нормальна форма (3НФ). Відношення знаходиться в 3НФ, коли знаходиться у 2НФ і кожен не ключовий атрибут нетранзитивно залежить від первинного ключа. Простіше кажучи, друге правило вимагає виносити все не ключові поля, вміст яких може належати до кількох записів таблиці на окремі таблиці.

Нормальна форма Бойса-Кодда (НФБК) (приватна форма третьої нормальної форми). Визначення 3НФ не зовсім підходить для таких відношень:

- Відношення має два або більше потенційних ключа;
- два і більше потенційні ключі є складовими;
- вони перетинаються, тобто. мають хоча б один атрибут.

Для відносин, що мають один потенційний ключ (первинний), НФБК є ЗНФ.

Ставлення знаходиться в НФБК, коли кожна нетривіальна і неприводима зліва функціональна залежність має потенційний ключ як детермінанти.

Четверта нормальна форма (4НФ). Відношення знаходиться в 4НФ, якщо воно знаходиться в НФБК і всі нетривіальні багатозначні залежності фактично є функціональними залежностями від її потенційних ключів. Щодо  $R(A, B, C)$  існує багатозначна залежність  $R.A \rightarrow R.B$  в тому і тільки в тому випадку, якщо безліч значень  $B$  відповідне парі значень  $A$  і  $C$  залежить тільки від  $A$  і не залежить від  $C$ . П'ята нормальна форма (5НФ). Відносини знаходяться у 5НФ, якщо воно знаходиться в 4НФ і відсутні складні залежні сполуки між атрибутами. Якщо "Атрибут залежить від "Атрибута\_2", а "Атрибут\_2" у свою чергу залежить від "Атрибута\_3", а "Атрибут\_3" залежить від "Атрибута\_1", то всі три атрибути обов'язково входять в один кортеж.

П'ята нормальна форма (5НФ). Відносини знаходяться у 5НФ, якщо воно знаходиться в 4НФ і відсутні складні залежні сполуки між атрибутами. Якщо "Атрибут залежить від "Атрибута\_2", а "Атрибут\_2" у свою чергу залежить від "Атрибута\_3", а "Атрибут\_3" залежить від "Атрибута\_1", то всі три атрибути обов'язково входять в один кортеж. Це дуже жорстка вимога, яку можна виконати лише за додаткових умов. На практиці важко знайти приклад реалізації цього вимоги у чистому вигляді. Доменно-ключова нормальна форма. Змінна стосунки перебуває в ДКНФ тоді і лише тоді, коли кожне накладене на неї обмеження є логічним наслідком обмежень доменів та обмежень ключів, накладених на цю змінну відносини. Обмеження домену – обмеження, що наказує використовувати для певного атрибута значення лише з певного заданого домену. Обмеження за своєю сутністю є завданням переліку (або логічного еквівалента переліку) допустимих значень типу та оголошенням про те, що зазначений атрибут має даний тип. Обмеження ключа – обмеження, яке стверджує, що певний атрибут або комбінація атрибутів є потенційним ключем.

Будь-яка змінна відносини, що у ДКНФ, обов'язково перебуває у 5НФ. Однак не будь-яку змінну відношення можна привести до ДКНФ.

Шоста нормальна форма (6НФ). Змінна відносини знаходиться в шостій нормальній формі тоді і лише тоді, коли вона задовольняє всім нетривіальним залежностям з'єднання. З визначення слідує, що змінна знаходиться в 6НФ тоді і тільки тоді, коли вона неприведена, тобто не може бути піддана подальшій декомпозиції без втрат. Кожна перемінна відношення, яка знаходиться в 6НФ, також знаходиться і в 5НФ. Ідея «декомпозиції до кінця» висувалась до початку досліджень у області хронологічних даних, але не знайшла підтримки. Однак для хронологічних баз даних максимально можлива декомпозиція дозволяє боротися з надмірністю і спрощує підтримку цілісності бази даних. Для хронологічних баз даних визначено U\_оператори, які розпаковують відносини за вказаними атрибутами, виконують відповідну операцію та упаковують отриманий результат. У цьому прикладі з'єднання проєкцій зв'язку повинно проводитись за допомогою оператора U\_JOIN

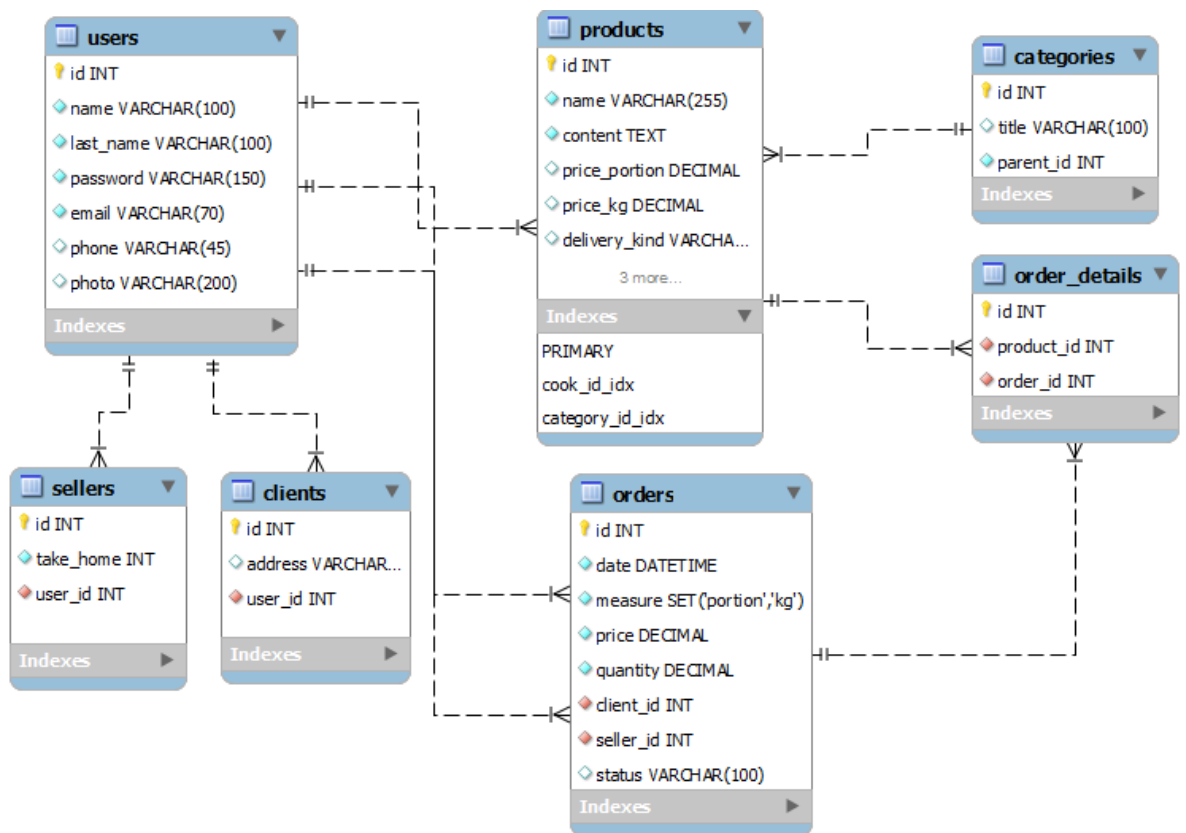


Рис.2.11

## 2.7. Структура даних, що надсилаються

JSON (JavaScript Object Notation) – простий формат обміну даними, зручний для читання та написання як людиною, так і комп'ютером. Це текстовий формат, повністю незалежний від мови реалізації, але він використовує угоди, знайомі програмістам С-подібних мов, таких як С, С++, С#, Java, JavaScript, Perl, Python та багато інших. Ці властивості роблять JSON ідеальною мовою обміну даними. Колекція пар ключ/значення. У різних мовах, ця концепція реалізована як об'єкт, запис, структура, словник, хеш, іменованний список або асоціативний масив. Упорядкований перелік значень. У більшості мов це реалізовується але як масив, вектор, список або послідовність. Це є універсальні структури даних. Майже всі сучасні мови програмування підтримують їх у будь-якій формі. Логічно припустити, що формат даних, незалежний від мови програмування, має бути заснований цих структурах. У нотації JSON виглядає так, як описано нижче.

Об'єкт – невпорядкований набір пар ключ/значення. Об'єкт, починається з { (відкриває фігурної дужки) і закінчується} (закриває фігурною дужкою). Кожне ім'я супроводжується: (двокрапкою), пари ключ/значення поділяються, (комою).

Масив – впорядкована колекція значень. Масив починається з [ (відкриває квадратної дужки) і закінчується ] (закриває квадратною дужкою). Значення розділені, (комою).

- Дані у форматі JSON є:
- JavaScript-об'єкти { ... };
- масиви [...];
- Значення одного з типів:
- рядки у подвійних лапках;
- Число;
- Логічне значення true / false;

- Null.

Перераховані вище структури можуть бути вкладеними

Рядок – колекція нуля або більше символів Unicode, що міститься в подвійні лапки, використовуючи \ (зворотну косу межу) як символ екранування. Символ представляється як один символний рядок. Схожий синтаксис використовується в C та Java.

Число представляється так само, як C або Java, крім того, що використовується лише десяткова система числення.

- Формування JSON файлу в мобільному додатку:
- рахувати файл з даними на сервері;
- Вибрати запис;
- додати запис до JSON об'єкта;
- перейти до кроку 2 доки не закінчаться записи;
- надіслати об'єкт JSON.

## **2.8.Архітектура мобільного додатка**

Поширеним підходом для створення сучасних мобільних Додатками є використання шаблону Model-View-Controller (MVC). MVC – це конструкційний шаблон, який описує спосіб побудови структури нашого додатка, сфери відповідальності та взаємодія кожної з частин у даній структурі. Враховуючи рекомендації для розробників від корпорації Google і загальноприйняті практики, при побудові архітектури програми буде використовуватися шаблон MVC. MVC – це не шаблон проекту, це конструкційний шаблон визначає спосіб побудови структури нашого додатка, сфери відповідальності та взаємодію кожної з частин у цій структурі. Шаблон проектування MVC передбачає поділ даних програми, інтерфейсу користувача і керуючої логіки на три окремих компоненти: Модель, Подання та Контролер – таким чином, що модифікація кожного компонента може здійснюватися незалежно.

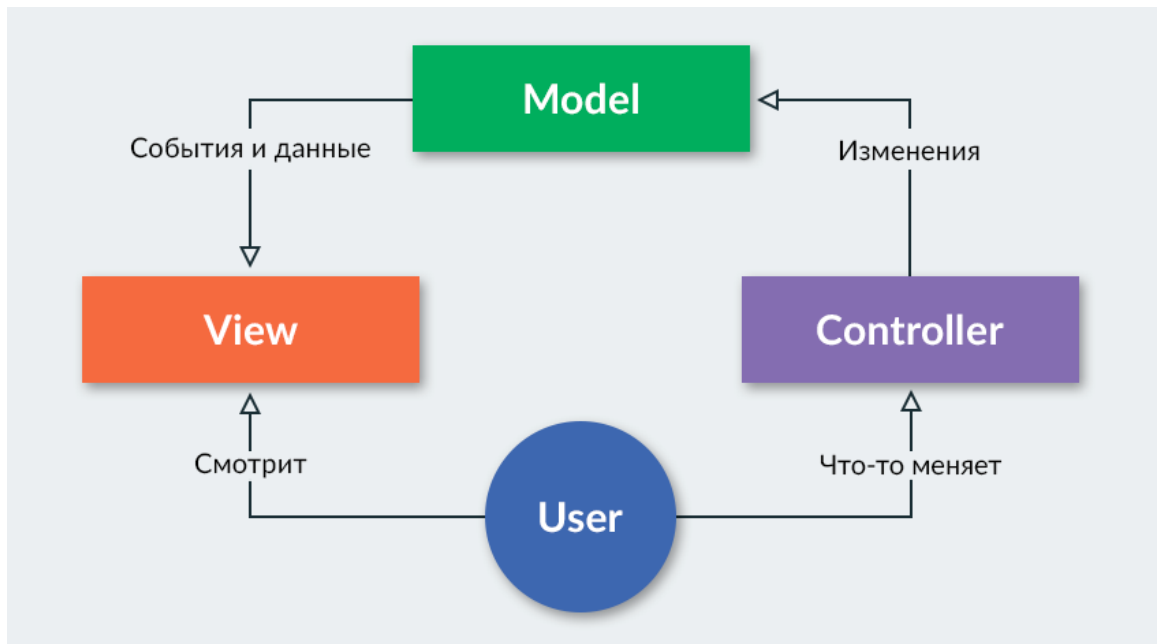


Рис.2.12

#### Преваги MVC:

- відокремлює бізнес-логіку від користувацького інтерфейсу та роботи з базою даних;
- дозволяє розташовувати різні частини в різних місцях, що сприятливо позначається на підтримці та обслуговуванні

Контролер (Controller) управляє запитам користувача (отримувани у вигляді запитів HTTP GET або POST, коли користувач натискає на елементи інтерфейсу для виконання різних дій). Його основна функція – викликати і координувати дію необхідних ресурсів та об'єктів, необхідних для виконання дій, що задаються користувачем. Зазвичай контролер викликає відповідну модель для завдання та вибирає відповідний вигляд.

Модель (Model) – це дані та правила, що використовуються для роботи з даними, що представляють концепцію управління додатком. У будь-якому додатку вся структура моделюється як дані, які обробляються певним чином. Тільки дані, які мають бути оброблені відповідно до правил (дата не може вказувати в майбутнє, e-mail повинен бути у певному форматі, ім'я не може бути довшим за X символи, і так далі). Модель дає контролеру подання даних, які запросив Користувач. Модель даних буде однаковою, незалежно від того, як



необхідно представляти їх користувачеві. Модель містить найважливішу частину логіки програми, логіки, яка вирішує завдання. Контролер містить в основному організаційну логіку для самого додатка.

Вид (View) забезпечує різні способи представлення даних, які отримані з моделі. Він може бути шаблоном, який заповнюється даними. Можливо кілька різних видів, і контролер вибирає, який найкраще підходить для поточної ситуації.

Веб-додаток зазвичай складається з набору контролерів, моделей і видів. Контролер може бути влаштований як основний, який отримує все за проси та викликає інші контролери для виконання дій залежно від ситуації

Завдяки цьому підходу реалізується концепція поділу відповідальності, у зв'язку з чим легше побудувати роботу над окремими компонентами. Крім того, внаслідок цього додаток має кращу тестованість. І якщо, допустимо, важлива візуальна частина або front end, то ми можемо тестувати уявлення незалежно від контролера. Або ми можемо зосередитись на back end та тестувати контролер.

Конкретні реалізації та визначення даного патерну можуть відрізнитися, але через свою гнучкість і простоту він став дуже популярним в останній час, особливо у сфері веб-розробки

## **ВИСНОВКИ**

Зростання мобільного трафіку спостерігається у всьому світі. Збільшення мобільного трафіку створила попит на розробку мобільних додатків. Бізнес також зреагував на це зростання і виникла потреба у зручному і швидкому процесу розробки мобільних додатків. Одним із таких рішень є використання Flutter. Зараз спостерігається збільшений попит на розробників на цій мові програмування, а це говорить про зручний підхід для сторони бізнесу. Метою цієї роботи була розробка мобільного додатку для інтернет магазину.

### **Список бібліотечних джерел**

1. Robert C. Martin Clean Code: A Handbook of Agile Software Craftsmanship
2. <https://flutter.dev/learn>
3. <https://www.tutorialspoint.com/flutter/index.htm>
4. Роберт Мартин Чистая архитектура. Искусство разработки программного обеспечения