

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД «УНІВЕРСИТЕТ «КРОК»  
Фаховий коледж Університету «КРОК»

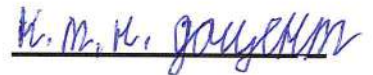
ДИПЛОМНА РОБОТА

за темою

«Розробка та створення програмного додатку з продажів на  
платформі Android»

Студент 4 курсу групи КН-20к

Керівник дипломної роботи



(посада керівника)

Вейдемюлер Олег Володимирович

(прізвище, ім'я та по-батькові студента)

Чернозубкін Ігор Олександрович

(прізвище, ім'я та по-батькові керівника)

До захисту

(резолуція «До захисту»)



(підпис студента)

10.06.2024

(дата)



(підпис викладача)

## Скорочення

**AR** (Augmented reality) - доповнена реальність. Термін, що позначає всі проєкти, спрямовані на доповнення реальності будь-якими віртуальними елементами.

**VR** (virtual reality) – віртуальна реальність. Створений технічними засобами світ, що передається людині через її відчуття: зір, слух, дотик та інші. Віртуальна реальність імітує як вплив, так і реакції на вплив.

**AI** (Artificial intelligence) – штучний інтелект. Розділ комп'ютерної лінгвістики та інформатики, який швидко розвивається, і зосереджений на розробці інтелектуальних машин, здатних виконувати завдання, які зазвичай потребують людського інтелекту.

**OS** (operating system) – операційна система. Це базовий комплекс програм, що виконує керування апаратною складовою комп'ютера або віртуальної машини; забезпечує керування обчислювальним процесом і організовує взаємодію з користувачем.

**A/B-тестування** (також відоме як спліт-тестування або bucket-тестування) - це методологія порівняння двох версій веб-сторінки або додатку одна з одною, щоб визначити, яка з них працює краще.

**IDE** (англ. Integrated Development Environment) – інтегрована середовище розробки. система програмних засобів, що використовується програмістами для розроблення програмного забезпечення.

**IOS** (до 24 червня 2010 року - iPhone OS) - мобільна операційна система для смартфонів, електронних планшетів, портативних програвачів, яку розробляє і випускає американська компанія Apple.

**ML модель** (англ. machine learning model) - Модель машинного навчання. Це програма, яка може знаходити закономірності або приймати рішення на основі раніше небаченого набору даних.

**ІОТ** (англ. Internet of Things) – павутина речей. описує мережу фізичних об'єктів - «речей», в які вбудовані датчики, програмне забезпечення та інші технології з метою підключення та обміну даними з іншими пристроями та

системами через Інтернет.

**Google I/O** - щорічна конференція, орієнтована на розробників, що проводиться компанією Google для обговорення розвитку відкритих технологій і сервісів Google. Проводиться в Moscone Center у травні або червні з 2008 р.

**IAP** (англ. in-app purchase) - будь-яка плата на додаток до вартості завантаження програми на смартфон або планшет. IAP дозволяють користувачам купувати додаткові функції, контент або послуги в додатку.

**ADT** (Android Development Tools) - це плагін для IDE Eclipse, призначений для надання вам потужного, інтегрованого середовища для створення додатків для Android.

**IntelliJ IDEA** - інтегроване середовище розробки програмного забезпечення для багатьох мов програмування, зокрема Java, JavaScript, Python, розроблене компанією JetBrains.

**UI** (англ. user interface) - Інтерфейс користувача. Інтерфейс, що забезпечує передачу інформації між користувачем-людиною і програмно-апаратними компонентами комп'ютерної системи.

**CI** (continuous integration) - практика розробки програмного забезпечення, яка полягає в постійному злитті робочих копій у загальну основну гілку розроблення та виконанні частих автоматизованих збірок проєкту для якнайшвидшого виявлення потенційних дефектів і розв'язання інтеграційних проблем.

**CD** (Continuous delivery) - Безперервна доставка. Є розширенням безперервної інтеграції, оскільки вона автоматично розгортає всі зміни коду в тестовому та/або виробничому середовищі після етапу збірки.

**ASO** (App store optimization) - це набір практик, спрямованих на підвищення видимості та конвертованості сторінки продукту мобільного додатку відповідно до конкретних технічних вимог певного ринку (наприклад, App Store або Google Play) та вподобань користувачів.

**CPU** (central processing unit) - електронний блок або інтегральна схема, що виконує машинні інструкції (код програм), головна частина апаратного забезпечення комп'ютера або програмованого логічного контролера. Іноді цей компонент називають просто процесором.

## Зміст

Вступ .....	5
1. Аналіз покупок в історії людства та сьогодні.....	6
1.1 Покупки у минулому та сьогодні.....	6
1.2 Задачі програми списку покупок .....	19
2 Середовище розробки та технічні рішення .....	20
2.1 Операційна система, платформа, засоби розробки .....	20
2.2 Принципи розробки додатків із застосуванням Android Studio .....	40
2.3 Додаток «Список покупок» ... ..	47
Висновок .....	51
Література .....	53
Додаток.....	55

## Вступ

Існує ситуація, з якою ми всі стикалися: повернувшись додому після закінчення робочого дня у продуктовому магазині, ви розумієте, що забули щонайменше три речі. Це не дуже приємно! Створення списку продуктів може здатися ще однією рутинною справою, але насправді це може заощадити вам час і гроші на бензин, і вам не доведеться двічі відвідувати магазин.

Ви будете зосереджені та продуктивні під час шопінгу, маючи список продуктів. Без нього легко забути про найважливіші речі або перевитрати гроші на те, що ви забули, що вже купили. Таким чином, як скласти список покупок, які методи є найефективнішими, щоб швидко і ефективно завершити найменш улюблені покупки?

Багато з нас мають певну систему купівлі продуктів протягом більшої частини свого життя. Але величезна кількість варіантів може легко збити нас з пантелику, оскільки різноманітність вибору та асортимент брендів постійно збільшуються. Іноді ці альтернативи є кращими, ніж те, що ми зазвичай купуємо, наприклад, яйця вищої якості за розумною ціною. Крім того, існує багато варіантів, які пропонують веселі доповнення, щоб урізноманітнити наше життя. Незважаючи на те, що кавун смачний, він, можливо, не потрібен.

В інших ситуаціях різноманітність може бути надзвичайно приголомшливою, і ви можете бути розгубленими. Це часто призводить до того, що ви повертаєтесь додому з порожніми руками або з неправильними товарами. Занадто багато речей можуть призвести до переосмислення та зупинки процесу прийняття рішень. З іншого боку, це може призвести до протилежного ефекту: всі варіанти можуть відволікати вашу увагу, що призведе до перекупівлі, що, у свою чергу, може призвести до переплати.

## **1. Аналіз покупок в історії людства та сьогодні**

### **1.1 Покупки у минулому та сьогодні**

Для того щоб зрозуміти важливість теми дослідження доведеться зануритись у історію шопінгу, щоб зрозуміти як він проявлявся у минулому і сьогодні.

Впродовж століть основний компонент сучасного життя, шопинг, зазнав значних змін. Історія шопінгу — це захоплююча історія про зміну, адаптацію та інновації, починаючи з простого і закінчуючи розкішним роздрібним досвідом сучасності.

Шопінг був рідкісною справою в стародавніх цивілізаціях і обмежувався місцевими ринками та базарами. Найдавніші згадки про шопінг можна знайти приблизно в 3000 році до нашої ери в стародавній Шумерії, де на жвавих ринках люди обмінювалися товарами. Зі зростанням міст складність торгівлі зростає, і з'явилися спеціалізовані купці, торговці та ринкові ятки. Старі греки та римляни покращили мистецтво шопінгу, створивши монети як засіб обміну та створивши постійні ринки.

У Середньовіччі з'явилися ярмарки та ринки, де торговці продавали свої товари охочим покупцям. Під час промислової революції, яка почалася з появою універмагів у 1800-х і 1900-х роках, концепція шопінгу, яку ми знаємо зараз, почала формуватися. Ці відомі магазини, такі як Harrods у Лондоні та Le Bon Marché у Парижі, об'єднали широкий вибір товарів і змінили досвід шопінгу. Запровадження фіксованих цін, реклами та обслуговування клієнтів змінило роздрібну торгівлю, зробивши шопінг більш приємним і зручним.

20 століття стало свідком поширення приміських торгових центрів і молів, що стало прямою відповіддю на зростання субурбанізації та зростання середнього класу. У цей період також з'явилися дисконтні магазини, супермаркети та мережеві магазини, які ще більше трансформували ландшафт роздрібною торгівлі. Епоха після Другої світової війни спричинила споживчий бум з появою кредитних карток, планів розстрочки та легкого кредитування, що дозволило людям легше потурати своїм бажанням.

У другій половині двадцятого століття почалася цифрова ера, яка значно вплинула на шопінг. Люди можуть зараз купувати речі, не виходячи з дому, завдяки розвитку електронної комерції, онлайн-маркетплейсів і цифрових платіжних систем. Поява таких гігантів роздрібною торгівлі, як Amazon, eBay та Alibaba, перевернула роздрібну торгівлю, пропонуючи неперевершену зручність, вибір і доступність.

Шопінг завжди мав значний вплив на суспільство, впливаючи на культурні норми, соціальну поведінку та економічні системи. Хоча він стимулює економічне зростання та розвиток, шопінг створив нові можливості для зайнятості, підприємництва та інновацій. Наші цінності, прагнення та ідентичність були створені завдяки йому, що сприяло розвитку культури споживання.

У 21 столітті світ шопінгу зазнав кардинальних змін. Минули часи виснажливого розглядання вітрин, довжелезних черг та обмеженого вибору товарів. Сьогодні шопінг перетворився на захоплюючий досвід, який поєднує в собі технології, зручність та розваги.

Сьогодні шопінг є складним, багатоканальним досвідом, який безперешкодно поєднує онлайн і офлайн канали. Розвиток смартфонів і соціальних мереж дав ритейлерам нові можливості взаємодіяти з клієнтами, надаючи їм індивідуальний досвід, унікальні пропозиції та зворотній зв'язок у режимі реального часу. Зростання експірієнс-ритейлу, де шопінг поєднується з розвагами, ресторанами та розвагою, ще більше змінило ландшафт роздрібною торгівлі. Велике різноманіття товарів, доступних для покупки, є однією з найважливіших переваг сучасного шопінгу. Люди мають безліч варіантів, від нішевих бутиків до великих універмагів. Кожен знайде щось для себе, будь то дорогі бренди, екологічно чиста мода чи ручна робота. Розвиток швидкої моди призвів до того, що модний одяг став дешевим і доступним, а малі підприємства та ремісники отримали можливість просувати свої унікальні вироби через онлайн-маркетплейси.

Досвід шопінгу значно покращився завдяки технологіям. Використовуючи доповнену реальність (AR) і віртуальну реальність (VR), клієнти можуть приміряти віртуальний одяг, дивитися меблі в своїх будинках і навіть відвідувати віртуальні магазини.

Алгоритми машинного навчання та штучний інтелект (AI) полегшують покупцям відкриття нових продуктів і брендів завдяки персоналізованим рекомендаціям.

Сучасний шопінг має враження, а не просто зручність і різноманітність. Торгові центри зараз перетворюються на розважальні центри з кінотеатрами, ресторанами та іншими розважальними заходами. Поява поп-ап магазинів і тимчасових торгових майданчиків підвищила несподіванку та ексклюзивність, що спричинило ажіотаж навколо нових товарів і брендів. Навіть онлайн-продавці вкладають гроші в фізичні магазини, щоб поєднати досвід покупок онлайн і офлайн.

Але сучасний шопінг має свою ціну через зручність і доступність. Стрес щодо впливу швидкої моди на навколишнє середовище, відходів упаковки та викидів вуглекислого газу при транспортуванні є надзвичайно актуальним. Як споживачі, ми повинні бути свідомими, коли купуємо речі, вибираючи екологічні бренди, зменшуючи кількість відходів і, коли це можливо, переробляючи їх.

Навіть у наші складні часи ситуація зазнала мінімальних змін.

«Як свідчать дані щомісячного дослідження, що проводиться серед членів Асоціації ритейлерів України, через 12 місяців повномасштабної війни вітчизняні FMCG-оператори відновили роботу 96% своїх мереж. Ця частина критичної інфраструктури та найчисельніша галузь ритейлу за кількістю гравців та торгових точок не тільки вистояла, але й продовжує активно розвиватися.

Попри скорочення загальної кількості торгових точок на 122 магазини порівняно із довоєнною добою, лідером за цим показником залишається мережа АТБ (Дніпро). Станом на березень 2023 року тут працювало 1176 магазинів, що на 12% більше за аналогічний показник березня 2022 року. Слідом за АТБ як і раніше йде компанія Fozzy Group (Київ), яка об'єднує мережі Сільпо, Фора, Fozzy Cash&Carry,

Le Silpo і Thrash!Траш!, що сумарно дають 726 магазинів проти 785, що працювали перед війною. З березня минулого року компанії вдалося відновити роботу майже 41% магазинів і це найбільша динаміка відновлення серед трьох основних гравців.

Третій ключовий гравець за кількістю торгових точок – компанія VolWest Retail (Луцьк), що об'єднує мережі Наш Край та Spar, теж не уникла втрат і тепер налічує всього 258 магазинів проти 267, які працювали до війни. Відносно невеликий відсоток відновлення роботи торгових точок (+5,7%) продиктований, перш за все, географією переважного розташування їх в центральних та західних регіонах країни.

В першу п'ятірку лідерів входить також мережа КОЛО (Київ), яка перед війною нараховувала 249 магазинів, розташованих у Київському та Одеському регіонах. На жаль, ритейлер, що колись дуже динамічно розвивав мережу магазинів «коло дому», тепер вимушено скорочує кількість торгових точок задля збереження ефективності роботи бізнесу. Порівняно із довоєнним періодом кількість діючих точок зменшилася майже на 11% і наразі становить 222 магазини. Проте динаміка відновлення вражає: з березня минулого року кількість діючих торгових точок збільшилася на 141,3% (з 92 до 222 магазинів).

В той же час компанія МХП, яка знаходиться на етапі трансформації в кулінарну компанію, активно продовжує розвиток свого ритейл-проекту – мережі магазинів М'ясомаркет (Київ). Якщо напередодні повномасштабного вторгнення мережа налічувала 179 торгових точок, то вже у березні 2022-го працювало тільки трохи більше половини. Але за наступні 12 місяців відкритих М'ясомаркетів побільшало майже на 153%. І тепер їх більше на 66 точок ніж до великої війни. Багато в чому це обумовлено вибраною моделлю розвитку по франшизі та активною підтримкою материнської компанії. Як результат – четверте місце за кількістю магазинів.

Успіх М'ясомаркету витіснив із п'ятірки лідерів львівську мережу магазинів Рукавичка, навіть попри те, що вона масштабувалась зі 185 магазинів до 200 (з приростом у 8%).

Водночас на другому місці серед найспритніших є луцький ритейлер Сім23, який має вже 179 магазинів у західних регіонах країни і майже щомісяця відкриває нові. Це на 47 точок більше, ніж було до війни і на 57% більше за показники березня минулого року. Ювілейний 150-й магазин мережі маркетів Сім23, яка працює в форматі convenience store, відкрився восени минулого року в Луцьку – місті, з якого ритейлер починав свій розвиток.

Ще один ритейлер, який демонструє гарну динаміку розвитку – мережа Файно маркет з Кропивницького яка наростила мережу до 148 магазинів або майже на 14%. Замикають рейтинг мережі Делві (Київ) та ЕКО маркет, які дещо зменшили кількість магазинів за останні 12 місяців.»[1]

Для зручності дані також представлено у вигляді картинки (Рис.1.1.)

Топ-10 продуктових мереж України за кількістю працюючих магазинів і темпами відкриттів								
№	Компанія	Мережі	Лого	Місто	Кількість т.ч., березень, 2023	Кількість т.ч., березень, 2022	Приріст 2023-2022 рр. %	Кількість торгових точок до військових дій
1	ТОВ "АТБ-маркет"	АТБ, АТБ express		Дніпро	1176	1050	+12%	1298
2	Fozzy Group	Сільпо, Thrashi, Фора, Fozzy, Le Silpo, Favore		Київ	726	515	+40,9%	785
3	VolWest Retail	Наш Край, SPAR		Луцьк	258	244	+5,7%	267
4	ПАО «МХП»	М'ясомаркет		Київ	245	97	152,6%	179
5	ТОВ "АРИТЕЙЛ"	КОЛО		Київ	222	92	+141,3	249
6	ТОВ ТПК "Львівхолод"	Рукавичка, Під боком		Львів	200	185	+8,1%	185
7	ТОВ «Клевер Сторс»	Сім23		Луцьк	179	114	+57%	132
8	ТОВ «Вересень плюс»	Файно маркет, Соціальний магазин, Varto,		Кропивницький	148	130	+13,8%	130
9	ТОВ "Оптторг-15", ТОВ НМСМ "Делві"	Делві		Київ	145	148	-2%	148
10	ТОВ «ЕКО»	ЕКО маркет, Sympatic		Київ	124	125	-0,8%	135

Рис.1.1. Кількість магазинів та темпи відкриттів в Україні

Станом на 2023 рік у світі налічується 2,64 мільярда онлайн-покупців. Це становить понад 33% від загальної кількості населення планети.[2]

Тепер, коли ми дізналися про історію явища і продивились деякі статистичні дані, можна переходити до причини необхідності списку покупок – сучасному дизайну магазинів.

Коли ми проходимо крізь розсувні скляні двері супермаркету, нас одразу огортає світ спокусливих ароматів, яскравих кольорів і стратегічно розміщених продуктів. Непомітно для себе ми потрапляємо у ретельно продумане середовище, створене для того, щоб розлучити нас з нашими важко заробленими грошима. Супермаркети опанували мистецтво психологічної маніпуляції, застосовуючи безліч тактик, щоб спонукати нас витратити більше, ніж ми планували.

Перш за все, супермаркети використовують наш нюх - потужний тригер емоцій та спогадів. Привабливий аромат свіжоспеченого хліба, що доноситься з пекарні при магазині, є цілеспрямованим прийомом, який стимулює наш апетит і збільшує наші витрати. Йдучи на непереборний аромат, ми опиняємося у відділі хлібобулочних виробів, оточені асортиментом апетитних ласощів. Ми не встигаємо озирнутися, як додаємо до кошика кілька імпульсивних покупок, піддавшись спокусі свіжої випічки.

Ще одна хитра тактика, яку застосовують супермаркети, - це стратегічне розміщення продуктів. Продукти першої необхідності, такі як молоко та хліб, часто розташовані в глибині магазину, що змушує нас пробиратися крізь лабіринт спокусливих товарів, щоб дістатися до них. Таке навмисне планування гарантує, що ми бачимо безліч продуктів, що збільшує ймовірність імпульсивних покупок. Коли ми блукаємо проходами, наші погляди привертають барвисті упаковки, рекламні акції, що привертають увагу, та стратегічно розташовані знаки знижок - все це покликане привернути нашу увагу і відірвати нас від наших грошей. (Рис.1.2.)

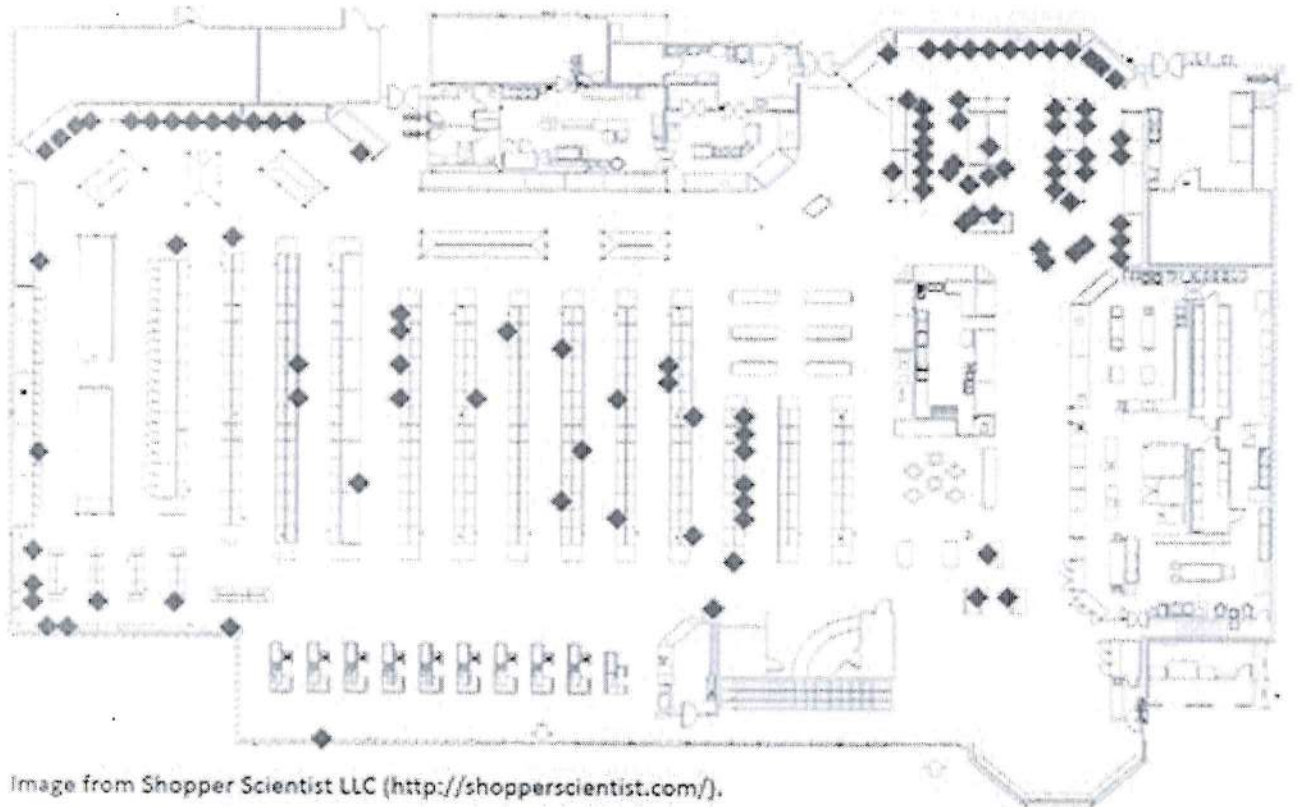


Рис.1.2. Приклад розташування товарів у супермаркеті

Супермаркети також експлуатують нашу емоційну вразливість, створюючи відчуття терміновості та дефіциту. Пропозиції з обмеженим терміном дії, акції «купи один - отримай другий безкоштовно» та «поки є запаси» створюють хибне відчуття терміновості, спонукаючи нас до імпульсивних покупок, щоб уникнути уявних втрат. Така тактика грає на нашому страху щось упустити (FOMO), спонукаючи нас витратити більше, ніж нам потрібно. Крім того, ретельне розміщення часто куплюваних продуктів, таких як шоколад і цукерки, на касах використовує нашу ослаблену силу волі, що призводить до додаткових, незапланованих покупок.

На додаток до цієї тактики, супермаркети опанували мистецтво психології ціноутворення. Ціни, що закінчуються на .99 або .95, створюють психологічну ілюзію, роблячи продукти більш доступними, ніж вони є насправді. Ця цінова стратегія, відома як зачаровуюче ціноутворення, покликана змусити нас сприймати товари як більш цінні, ніж вони є насправді, що призводить до збільшення витрат. Крім того, використання «збиткових товарів» - товарів, що продаються зі збитками для залучення покупців, - заманює нас до магазину, де ми

неминуче купуємо більше, ніж планували.

Більше того, супермаркети стали вправними у зборі та використанні даних про наші купівельні звички, що дозволяє їм пристосовувати свої маркетингові стратегії до наших індивідуальних уподобань. Програми лояльності, які пропонують винагороди та персоналізовані знижки, створюють відчуття лояльності та заохочують нас витратити більше в гонитві за винагородою. Ці програми також надають супермаркетам цінну інформацію про наші купівельні звички, що дозволяє їм орієнтуватися на нас за допомогою спеціальних акцій та пропозицій.

Нарешті, супермаркети стали майстрами у створенні захоплюючого шопінгу, покликаного утримати нас у магазині якомога довше. Зручне освітлення, заспокійлива музика, зручності, такі як кафе та зони відпочинку, - все це сприяє створенню гостинної атмосфери, яка спонукає нас затриматися і витратити більше.

Розуміючи ці тактики, ми можемо стати більш уважними до своїх звичок у витрачанні коштів і приймати більш обґрунтовані рішення про покупки. Однак не можна заперечувати, що вони стали експертами в мистецтві переконання, тому протистояти їхній спокусі стає дедалі складніше.

Тож, дізнавшись про методи маніпуляції що використовують ретейлери по всьому світу ми повертаємось до нашого початкового питання: «як цьому протистояти?»

Відповідь очевидна: створити список покупок.

Навряд чи вийде зробити це з першого разу. Але коли такий «список по науці» стане звичкою, ви здивуєтесь, скільки грошей вдалося заощадити. Таким чином, трохи науки. Наш мозок не адаптований до сучасного життя, у якому є багато їжі і ніхто не хоче тебе з'їсти. Це неможливо з точки зору еволюції, і наш мозок є результатом цієї самої еволюції. У наших мізках є набагато більше тваринного, ніж людського.

Таким чином, сучасні люди вважають, що наш мозок поділений на дві системи. Одна система називається підсвідомістю, яка швидка, автоматична та споживає мало енергії; інша система називається Свідомістю, яка повільна, неавтоматична та споживає багато енергії. Щоб прийняти свідомі рішення, мозок втомлюється швидко, коли свідомість вимикається, і всі рішення передаються підсвідомості. Маркетологи також відмінно знають, як працює підсвідомість.

Навіть найпростіший список покупок може допомогти вам заощадити гроші, час і нерви. Як наслідок, найпростіші правила складання списку покупок, а також причини, чому це робити саме так:

Починайте планувати свої покупки заздалегідь, а не перед самим походом. Якщо ви пишете список похапцем, підсвідомість завжди нагадає вам, що ви «щось забули». Упорядкування списку того, що закінчилося або може закінчитися в будинку, є ідеальним способом позбутися нав'язливості. Друга перевага заздалегідь складеного списку полягає в тому, що в разі, якщо хтось із близьких чи знайомих подзвонить до магазину і запитає, що купити, достатньо просто поглянути в нього.

Ви повинні розрахувати суму покупки та взяти трохи більше грошей. У вас просто не буде грошей на покупку чогось, чого немає в списку. Якщо це неможливо, наприклад, якщо ви користуєтеся кредитною картою, уявіть, що на ній немає грошей, крім зазначеної суми. абсолютно.

Якщо у вас обмежена сума, визначте, які покупки є найважливішими. Ціни останнім часом різко зросли. Крім того, приймати рішення щодо того, що прибрати з повного кошика в черзі на касі не буде найкращим доповненням до вашого дня.

Заздалегідь плануйте маршрут і позначте черговість придбання товарів, якщо магазин знайомий. Крім того, це зменшує ймовірність непередбачених покупок.

Ось приклад, як може виглядати корисний список:

«фрукти: яблука, мандарини, грейпфрути, авокадо, апельсини, банани, лимони, ананас;

овочі: броколі, спаржа, цибуля, шпинат, перець, кабачки, картопля, морква, цвітна капуста, солодкий перець, зелень, цибуля-порей, гриби;

круп: рис, гречка, пшоно, вівсянка, червона сочевиця, ячмінь, горох;

білки: яйця, риба, курячі грудки, фарш з індички, тофу;

молочні продукти: молоко, йогурт, сир "Фета", "Чеддер";

приправи: оливки, в'ялені томати, заправка для салату, оливкова та соняшникова олія, соуси;

напої: бутильована вода, кава, чай;

заморожені продукти: креветки, суміш фруктів;

різне: сухофрукти, темний шоколад.» [3]

Існуючі рішення:

#### 1. Listonic

Listonic є одним із найпопулярніших програм для складання списків покупок, і його основні функції доступні для безкоштовного використання. На додаток до списку можна включити кількість продуктів і ціни, щоб отримати приблизну загальну вартість. Функція синхронізації, доступна після реєстрації та авторизації в системі, дозволяє переглядати списки з будь-якого підключеного пристрою, включаючи веб-сайт. Можливість голосового додавання позицій і відображення найпопулярніших варіантів у вікні введення скоротили час при складанні списку покупок.

Оскільки Listonic підтримує функцію обміну даними та підтримує зміну списків у режимі реального часу, він дозволяє організовувати покупки для всієї родини. Крім того, додаток має можливість сортувати продукти, що полегшує пошук товарів у самому магазині. Для зручності ви можете створювати безліч

списків. Це може бути корисно, наприклад, плануючи покупки на вихідні або свята. Мова інтерфейсу російська, але у вас є можливість змінити оформлення інтерфейсу: ви можете вибрати тему, розмір тексту та стиль шрифту.

## 2. Bring!

Більше додаток для складання списків покупок-Це Bring! Однією з особливостей програми є вбудована колекція рецептів, яка регулярно поповнюється. Кожен рецепт можна зберегти собі і вибрати зі сторінки опису відсутні інгредієнти, які потрібно купити. Імпортувати рецепти можна і з інтернету, вибравши функцію «поділитися» . Підтримується формування відразу декількох списків, припустимо продуктів або ліків. Кожен з них приємно візуалізований, а намальованих значків продуктів налічується понад 350.

Додаток включає нагадування про відвідування магазину та синхронізацію з хмарним сховищем. У Bring! є багато налаштувань для налаштування, наприклад темний і світлий режими. Ви також можете змінити зовнішній вигляд списку, який може бути плиткою або списком з місцем для опису. Крім того, додаток пропонує відповідні товари під час планування покупок: наприклад, якщо вам потрібно купити спагетті, Bring! запропонує також купити соус. Усі функції є безкоштовними.

## 3. Купи батон!

Додаток для списку покупок на Андроїд Купи батон! відрізняється простим і приємним інтерфейсом в мінімалістичному виконанні. При введенні назви продукту відобразиться перелік варіантів, позначених різними кольорами в залежності від типу товарів, наприклад овочі або хлібобулочні вироби. Користувач може або вибрати шаблон, або створити нову позицію, визначивши для нього категорію (колір). Доступно створення відразу декількох списків і швидке перемикання між ними. Після створення облікового запису всі дані можуть бути доступні і на інших пристроях.

У головному меню, що викликається натисканням на три горизонтальні смужки у верхньому кутку зліва, відображаються списки покупок з декількома позиціями з нього, перехід в розділ з настройками і показ керівництва по використанню ( «допомога» ). На цьому функції базової версії Програми закінчуються, але саме в цьому його головна перевага – в простому інтерфейсі і інтуїтивному управлінні. У Купи батон! PRO можна створювати необмежену кількість списків, передбачається більше колірних груп і відсутні обмеження для спільного використання.

#### 4. SoftList

Простий і зручний список покупок SoftList має багато функцій, включаючи підтримку кількох списків, можливість ділитися створеними списками, зберігання історії покупок та багато іншого. У програмі є функція розпізнавання товарів за допомогою сканування штрихкодів, а зображення позицій можна налаштувати самостійно. Завдяки вбудованій функції формування звітів можна відстежувати ціни.

Всі позиції автоматично розподіляються за категоріями (наприклад, хлібобулочні чи молочні вироби), а інтерфейс SoftList є повністю російською мовою. Крім можливості створювати нові шаблони товарів і категорії, кожен з них можна змінити або видалити. Сформовані дані можна не тільки поділитися, але й відправити на друк негайно. Додаткові можливості, такі як конвертер одиниць вимірювання, податок з продажів і купони, доступні в розділі налаштувань. У преміум-версії доступна хмарна синхронізація, аналітика витрат і створення списків товарів, розділених по магазинах.

#### 5. «Ваш список покупок»

Програма Твій список покупок дозволяє швидко створювати списки, змінювати їх, ділитися, клонувати та виконувати інші функції. Воно постійно працює офлайн і має вбудовану велику базу товарів, яка включає широкий асортимент продуктів. Усі категорії представлені за допомогою кольорів і зображень, що полегшує швидке орієнтування в списку покупок під час відвідування магазину. Введіть назву продукту, і нижче будуть показані

подібні варіанти. Просто напишіть цифру після пробілу, щоб вказати кількість товару. Голосове введення є можливим.

Для швидкого створення списків можна використовувати історію, оскільки товари, які найчастіше купуються, зберігаються в окремому розділі. Крім того, є можливість створити книгу рецептів, вводячи рецепти вручну за допомогою вбудованого редактора або використовуючи посилання з Інтернету. Користувач може змінити категорії продуктів, які вже є, і додати свої. Видалення, копіювання та очищення списків можна зробити одним натисканням. Стиль і колір елементів інтерфейсу, світлий і темний фон, розмір шрифту — це деякі з налаштувань, доступних для налаштування. У додатку всі можливості доступні безкоштовно, і в ньому немає реклами.

## 1.2 Задачі програми списку покупок

Дослідження користувацького досвіду (UX): Дослідження Google показало, що користувачі віддають перевагу візуально простим і зрозумілим сайтам і додаткам. Складний дизайн і щільний макет часто призводять до збільшення показника відмов. Ця тенденція поширюється і на мобільні додатки, де захаращеність інтерфейсу може призвести до поганого утримання користувачів і зниження їхньої залученості.

Галузеві тенденції: Тенденція до мінімалізму в дизайні добре задокументована. Такі компанії, як Apple і Google, відстоюють простий, чистий дизайн у своїх рекомендаціях для iOS і Android. Ці рекомендації підкреслюють важливість простоти, зручності навігації та чіткої візуальної ієрархії, впливаючи на розробників і дизайнерів, щоб вони прийняли ці принципи.

Хоча конкретні цифрові дані про те, скільки людей віддають перевагу простому дизайну, можуть відрізнитися, опитування та дослідження часто показують, що значна більшість користувачів віддають перевагу простоті. Наприклад, опитування, проведене компанією Clutch, показало, що 68% користувачів з більшою ймовірністю скористаються послугою, якщо вона має зручний і зрозумілий мобільний додаток.[4]

Простота в дизайні - це не просто перевага, а поведінкове очікування. Користувачі, як правило, витрачають більше часу і демонструють вищу зацікавленість у додатках, які пропонують бездоганний, зрозумілий досвід. Складний дизайн може призвести до розчарування користувачів і закриття програми, саме тому багато успішних додатків надають перевагу простим та інтуїтивно зрозумілим інтерфейсам.

Тож, наша програма має бути простою і інтуїтивно зрозумілою. Користувача ніщо не має відволікати, а додаток має працювати швидко і ефективно, щоб будь які дії можливо було зробити у декілька рухів.

## **2. Середовище розробки та технічні рішення**

### **2.1 Операційна система, платформа, засоби розробки**

Платформа Android демонструє постійну еволюцію від ідеї молодого стартапу до найпоширенішої мобільної операційної системи в усьому світі, що робить її візитною карткою сучасних мобільних технологій.

Виникнення та заснування (2003-2005): Енді Рубін, Річ Майнер, Нік Сірс і Кріс Вайт заснували Android Inc. у Пало-Альто, штат Каліфорнія, у жовтні 2003 року. Компанія спочатку планувала створити операційну систему для цифрових камер, але побачила потенціал ринку мобільних телефонів і перейшла на розробку операційної системи для мобільних телефонів. Позиціонування Android як конкурента таким відомим конкурентам, як Symbian та Windows Mobile, було важливим стратегічним здвигом.

У липні 2005 року компанія Google придбала Android Inc. за приблизно 50 мільйонів доларів.[5] Це придбання привело Енді Рубіна та його команду під парасольку Google, що стало важливою віхою. Величезні ресурси та інноваційне середовище Google створили сприятливий ґрунт для розвитку Android. Команда задумала створити мобільну ОС з відкритим вихідним кодом, яка могла б запропонувати гнучкість і можливість кастомізації для широкого спектру пристроїв.[6]

#### Створення Open Handset Alliance та ранні версії (2007-2009)

5 листопада 2007 року Google оголосив про створення Open Handset Alliance (ОНА), консорціуму технологічних компаній, до якого увійшли виробники пристроїв, мобільні оператори, напівпровідникові компанії та розробники програмного забезпечення. Метою ОНА є просування відкритих стандартів для мобільних пристроїв, центральним елементом яких є Android. Ця ініціатива підкреслила прихильність Android до відкритості та співпраці.[7]

Перший офіційний реліз Android версії 1.0 відбувся у вересні 2008 року.

Вона мала такі основні функції, як веб-браузер, підтримка камери, інтеграція з Gmail та доступ до Android Market (пізніше перейменованого на Google Play Store). HTC Dream (T-Mobile G1), перший комерційно доступний смартфон під управлінням Android, був випущений у жовтні 2008 року, що ознаменувало початок присутності Android на споживчому ринку.[8]

### Швидкий розвиток і проникнення на ринок (2009-2012)

Перші роки існування Android характеризувалися швидким розвитком та частими оновленнями, кожне з яких містило значні вдосконалення та нові функції:

- Android 1.5 Cupcake (квітень 2009): Впроваджено екранну клавіатуру, можливість запису відео та підтримку віджетів. Ці функції розширили функціональність Android та покращили користувацький досвід.[9]

- Android 1.6 Donut (вересень 2009): Додана підтримка різних розмірів і роздільної здатності екранів, оновлений інтерфейс камери та покращена функціональність пошуку. Це зробило Android більш універсальним і зручним для користувачів.[10]

- Android 2.0/2.1 Eclair (жовтень 2009): Значні покращення інтерфейсу, навігація по картах Google Maps та підтримка декількох облікових записів. Ці оновлення значно підвищили зручність використання та привабливість пристроїв Android.

Ці випуски допомогли Android набути популярності, і такі виробники, як Samsung, HTC та Motorola, перейшли на цю ОС. Відкритий характер Android дозволив виробникам налаштовувати ОС відповідно до своїх уподобань, сприяючи інноваціям та диференціації на конкурентному ринку.

У 2010 році в Android 2.2 Froyo з'явилася оптимізація продуктивності, підтримка Adobe Flash, а також покращена швидкість роботи та управління пам'яттю. Android 2.3 Gingerbread, випущений у грудні 2010 року, зосередився на вдосконаленні інтерфейсу користувача, покращенні ігрової продуктивності та кращому управлінні акумулятором. Того ж року було започатковано

програму Nexus з Nexus One, що пропонує чистий досвід роботи з Android і слугує еталонним пристроєм для інших виробників.[11]

#### Шлях до домінування (2012-2015)

До 2012 року Android стала найпоширенішою мобільною операційною системою у світі, випередивши iOS від Apple. У цей період було випущено кілька ключових версій, які ще більше зміцнили домінування Android:

- Android 4.0 Ice Cream Sandwich (жовтень 2011): Об'єднала версії Android для телефонів і планшетів, запровадивши сучасну мову дизайну, відому як «Ноло». З'явилися такі функції, як розблокування за допомогою розпізнавання обличчя, покращена багатозадачність та вдосконалений користувацький інтерфейс.[12]

- Android 4.1-4.3 Jelly Bean (липень 2012 - липень 2013): Впроваджено Google Now, Project Butter для більш плавної роботи та розширені сповіщення, що покращило загальний користувацький досвід та швидкість реагування ОС.

Успіху Android сприяла його адаптивність і поширення пристроїв за різними ціновими категоріями. Магазин Google Play Store також швидко розширювався, пропонуючи величезну екосистему додатків, які задовольняли широкий спектр потреб користувачів.[13]

У 2014 році в Android 5.0 Lollipop з'явився Material Design, нова мова дизайну, що підкреслює чистий, плоский інтерфейс з плавною анімацією. Цей випуск також приніс значні зміни в базову архітектуру, такі як перехід на середовище виконання ART для кращої продуктивності та збільшення часу автономної роботи.[14]

#### Розширення та зрілість Android (2016-2019)

Протягом цього періоду Android продовжував розвиватися і розширювати свої можливості:

- Android 6.0 Marshmallow (жовтень 2015): Зосереджена на покращенні

взаємодії з користувачем завдяки таким функціям, як Google Now on Tap, режим очікування для збільшення часу роботи від акумулятора та гранульовані дозволи для додатків.[15]

- Android 7.0 Nougat (серпень 2016): Впроваджено багатозадачність з розділеним екраном, покращені сповіщення та API Vulkan для кращої графічної продуктивності.[16]

У 2017 році в Android 8.0 Oreo з'явився режим «картинка в картинці», точки сповіщень та Project Treble, спрямований на полегшення оновлення ОС для виробників.[17] Наступного року в Android 9.0 Pie з'явилася навігація жестами, адаптивний заряд батареї та яскравість, а також інструменти Digital Wellbeing.[18]

Сучасна епоха та перспективи на майбутнє (2020 - теперішній час)

Останні версії Android продовжують розвивати сильні сторони платформи, водночас вирішуючи нові завдання:

- Android 10 (вересень 2019): Відзначила кінець десертної схеми іменування, зосередившись на покращенні конфіденційності, загальносистемному темному режимі та підтримці складних пристроїв.[19]

- Android 11 (вересень 2020): Впроваджено бульбашки розмов, покращено контроль конфіденційності та покращено підтримку 5G.[20]

- Android 12 (жовтень 2021): Має значний редизайн під назвою «Material You», що дозволяє користувачам широко кастомізувати зовнішній вигляд і функціональність своїх пристроїв.

Станом на 2024 рік Android продовжує лідирувати на світовому ринку мобільних ОС з мільярдами активних пристроїв у всьому світі. Платформа вийшла за рамки смартфонів і використовується на планшетах, смарт-годинниках (Wear OS), телевізорах (Android TV) і навіть автомобільних системах (Android Automotive), демонструючи свою універсальність і широке розповсюдження.

## Виклики та інновації

Незважаючи на свій успіх, Android зіткнувся з кількома проблемами протягом багатьох років. Фрагментація залишається значною проблемою, оскільки багато пристроїв працюють на застарілих версіях ОС, що створює ризики для безпеки та проблеми сумісності для розробників. Google докладає зусиль для вирішення цієї проблеми за допомогою таких ініціатив, як Project Treble та Android One, які мають на меті впорядкувати оновлення та забезпечити узгоджений користувацький досвід на всіх пристроях.

Безпека також була основною проблемою, оскільки відкрита природа Android робить її мішенню для шкідливого програмного забезпечення. Компанія Google впровадила різні заходи для посилення безпеки, такі як регулярні оновлення безпеки, Google Play Protect та суворіші дозволи для додатків. Ці зусилля допомогли зменшити ризики та покращити загальний стан безпеки платформи.

Інновації продовжують залишатися рушійною силою для Android. Платформа використовує нові технології, такі як штучний інтелект (AI), доповнена реальність (AR) і машинне навчання. Інтеграція компанією Google функцій штучного інтелекту в Android, таких як Google Assistant і розширені можливості камери, встановила нові стандарти для мобільних пристроїв. Крім того, поява технології 5G відкрила нові можливості для швидшого та більш чутливого мобільного досвіду.

## Android на різних ринках і пристроях

Універсальність Android призвела до його впровадження на різних ринках, окрім смартфонів. Android TV, представлений у 2014 році, переніс платформу на смарт-телевізори, забезпечивши уніфікований розважальний досвід з доступом до додатків, ігор та потокових сервісів. Android Auto, запущений того ж року, поширив платформу на автомобільну промисловість, дозволивши водіям підключати свої Android-пристрої до автомобілів для навігації, прослуховування музики та гучного зв'язку.

На ринку пристроїв, що носяться, Wear OS (раніше Android Wear) є важливим гравцем з моменту її запуску в 2014 році. Розроблена для смарт-годинників та інших натільних пристроїв, Wear OS інтегрується з сервісами Google і підтримує широкий спектр додатків від сторонніх розробників. Платформа постійно оновлюється та вдосконалюється, що робить її популярним вибором для виробників пристроїв.

#### Глобальний вплив і частка ринку

Глобальний вплив Android незаперечний. До 2013 року Android стала домінуючою мобільною операційною системою, захопивши понад 80% світового ринку. Це домінування зберігається протягом багатьох років, і станом на 2019 рік Android працює на понад 2,5 мільярдах активних пристроїв.

Успіх платформи можна пояснити її відкритим вихідним кодом, що сприяло широкому впровадженню виробниками та розробниками. Гнучкість і можливості кастомізації, які пропонує Android, дозволили їй задовольнити різноманітні потреби та вподобання користувачів, що робить її кращим вибором для широкого спектру пристроїв і додатків.

#### Роль магазину Google Play Store

Магазин Google Play відіграє вирішальну роль в екосистемі Android, надаючи користувачам доступ до мільйонів додатків, ігор, фільмів, книг та іншого цифрового контенту. Запущений як Android Market у 2008 році, Play Store зростав у геометричній прогресії, ставши одним з найбільших магазинів додатків у світі.

Успіх Play Store можна пояснити потужною підтримкою розробників, комплексним процесом перевірки додатків та інтеграцією з сервісами Google. Розробники мають доступ до потужних інструментів і ресурсів, таких як Android Studio, для створення високоякісних додатків для платформи. Крім того, акцент Google на безпеці та користувацькому досвіді допоміг зберегти репутацію Play Store як надійного джерела цифрового контенту.

Android дивиться в майбутнє, але перед ним стоять як можливості, так і виклики. Постійний розвиток мобільних технологій, поява нових форм-факторів, таких як складні пристрої, а також зростаюче значення конфіденційності та безпеки визначатимуть майбутнє платформи Android.

Тепер – про Google Play Store

Google Play Store, який спочатку був запущений як Android Market у 2008 році, перетворився на одну з найбільших і найвпливовіших платформ цифрової дистрибуції у світі. Будучи основним магазином додатків для операційної системи Android, Play Store пропонує широкий вибір додатків, ігор, фільмів, книг, музики та іншого цифрового контенту.

Історичний огляд [21]

Виникнення та ранній розвиток (2008-2012)

Магазин Google Play розпочав свій шлях як Android Market, який був запущений у жовтні 2008 року. Початкова версія ринку була базовою, з обмеженим вибором додатків і простим інтерфейсом. Основна мета полягала в тому, щоб надати користувачам Android платформу для легкого пошуку та завантаження додатків, подібно до App Store від Apple, який був запущений кількома місяцями раніше.

У перші роки свого існування Android Market швидко зростав. До 2009 року він налічував понад 2300 додатків, а до 2010 року - понад 100 000 додатків. Такому зростанню сприяла зростаюча популярність смартфонів на базі Android та відкритість платформи, яка приваблювала велику кількість розробників.

У березні 2012 року компанія Google провела ребрендинг Android Market на Google Play Store, об'єднавши під єдиним брендом різні сервіси цифрової дистрибуції - Google Music, Google Books і Google Movies. Цей крок був спрямований на створення єдиної екосистеми для всього цифрового контенту, щоб полегшити користувачам пошук і доступ до широкого спектру медіа з одного місця.

## Розширення та вдосконалення функцій (2012-2016)

Після ребрендингу Google Play Store продовжував розширювати свої пропозиції та вдосконалювати функції. Основні події цього періоду включали

- Запровадження Сервісів Google Play: У 2012 році Google запустив Google Play Services, пакет фонових сервісів та API, який розширив функціональність додатків та покращив користувацький досвід. Це дозволило розробникам легше інтегрувати сервіси Google у свої додатки.

- Служби підписки: У 2012 році Play Store запровадив тарифікацію за підпискою, що дозволило розробникам пропонувати періодичну оплату за свої додатки та сервіси. Це було важливим доповненням, особливо для постачальників контенту та сервісних додатків.

- Сімейна бібліотека: Запущена у 2016 році функція «Сімейна бібліотека» дозволила користувачам ділитися придбаними додатками, іграми, фільмами, телепередачами та книгами з п'ятьма членами сім'ї. Ця функція заохочувала спільне використання та підвищила загальну ціннісну пропозицію Play Store.

Ці вдосконалення допомогли Play Store зміцнити свої позиції як комплексної платформи цифрової дистрибуції, що задовольняє різноманітні потреби своєї зростаючої бази користувачів.

Розмір ринку та економічний вплив[22]

Глобальне охоплення та база користувачів

Станом на 2023 рік Google Play Store налічує понад 3,5 мільйона додатків та ігор, що робить його найбільшим магазином додатків за обсягом доступного контенту. Щомісяця він обслуговує понад 2,5 мільярда активних користувачів у більш ніж 190 країнах. Широке охоплення Play Store є свідченням глобального домінування операційної системи Android, на якій працює приблизно 72,9% всіх смартфонів у світі.

## Генерування доходів

Google Play Store є значним джерелом доходу як для Google, так і для розробників додатків. У 2022 році дохід магазину Play Store склав приблизно 48 мільярдів доларів США, що є значною частиною загального доходу Google від цифрового контенту та послуг. Основними джерелами цього доходу є

- Купівля додатків та підписки: Прямі покупки додатків і підписки в додатках роблять значний внесок у доходи Play Store. Популярні додатки на основі підписки, такі як Spotify, Netflix, а також різні інструменти для фітнесу та продуктивності, генерують значний постійний дохід.
- Покупки в додатку (IAP): Багато додатків та ігор пропонують внутрішні покупки, що дозволяє користувачам купувати додатковий контент, функції або віртуальні товари. Ця модель особливо прибуткова в ігровому секторі: такі ігри, як «Clash of Clans» та «Candy Crush Saga», принесли мільйони доларів завдяки IAP.
- Реклама: Play Store також слугує платформою для мобільної реклами, дозволяючи розробникам монетизувати свої безкоштовні додатки за допомогою реклами. Платформа AdMob від Google відіграє в цьому вирішальну роль, допомагаючи розробникам безперешкодно інтегрувати рекламу в свої додатки.

## Економічний вплив на розробників

Магазин Google Play має значний вплив на світову екосистему розробки додатків. Він надає розробникам прибуткову платформу для охоплення широкої аудиторії та отримання прибутку. За останніми оцінками, з моменту свого заснування Play Store виплатив розробникам понад 120 мільярдів доларів США. Ця економічна діяльність підтримує мільйони робочих місць по всьому світу, від незалежних розробників до великих софтверних компаній.

Більше того, політика та інструменти Play Store полегшили розробникам монетизацію їхніх додатків. Впровадження таких функцій, як динамічна доставка, що дозволяє зменшити початковий обсяг завантаження, а додатковий

вміст завантажувати за потреби, покращило користувацький досвід і підвищило показники утримання користувачів. Постійні інвестиції Google у підтримку розробників, включаючи вичерпну документацію, навчальні ресурси та конференції для розробників, ще більше зміцнили екосистему додатків.

#### Ключові функції та інновації

##### Користувацький досвід та інтерфейс

Інтерфейс Google Play Store постійно вдосконалюється, щоб покращити зручність користування та покращити досвід користувачів. Основні функції включають:

- Персоналізовані рекомендації: Використовуючи алгоритми машинного навчання, Play Store надає персоналізовані рекомендації щодо додатків та контенту на основі поведінки та вподобань користувачів. Це значно покращило пошук контенту та залучення користувачів.

- Вміст для всієї родини: Play Store пропонує спеціальний розділ «Сім'я», в якому представлений спеціально підібраний контент для дітей та сімей. Цей розділ включає засоби батьківського контролю, рекомендації відповідно до віку та функцію «Сімейна бібліотека», яка сприяє спільному використанню матеріалів усіма членами сім'ї.

- Покращений пошук і фільтрація: Розширені можливості пошуку та фільтрації допомагають користувачам ефективніше знаходити програми та контент. Користувачі можуть фільтрувати результати пошуку за рейтингом, релевантністю та іншими критеріями, щоб швидко знайти потрібний вміст.

##### Інструменти та підтримка для розробників

Google надає розробникам потужний набір інструментів і ресурсів для ефективного створення, керування та монетизації їхніх додатків:

- Android Studio: Це інтегроване середовище розробки (IDE) надає комплексні інструменти для створення високоякісних програм для Android. Такі функції, як редагування коду, налагодження та аналіз продуктивності, допомагають розробникам оптимізувати свої програми.

- Firebase: Платформа Firebase від Google пропонує набір внутрішніх сервісів, включаючи хмарне сховище, бази даних у режимі реального часу, автентифікацію та аналітику. Ці сервіси дозволяють розробникам легко створювати масштабовані та багатофункціональні додатки.

- Консоль Google Play: Play Console надає розробникам детальну аналітику, відгуки користувачів та інструменти для управління випусками та оновленнями додатків. Вона також включає функції для бета-тестування, A/B-тестування та моніторингу продуктивності.

#### Заходи безпеки та конфіденційності

Компанія Google впровадила численні заходи безпеки та конфіденційності, щоб захистити користувачів і зберегти цілісність Магазину Play:

- Google Play Захист: Запущений у 2017 році, Google Play Protect - це комплексна служба безпеки, яка сканує додатки на наявність шкідливого програмного забезпечення та інших загроз. Вона виконує безперервне сканування, щоб забезпечити безпеку додатків на пристроях користувачів.

- Процес перевірки додатків: В Play Маркеті застосовується суворий процес перевірки нових програм та оновлень. У цьому процесі беруть участь як автоматизовані системи, так і люди, які виявляють і запобігають розповсюдженню шкідливих або зловмисних програм.

- Покращення конфіденційності: Google впровадив кілька функцій, орієнтованих на конфіденційність, таких як деталізовані дозволи для додатків і контроль місцезнаходження. Ці функції дають користувачам більше контролю над своїми даними та покращують загальний рівень конфіденційності.

Конкурентне середовище[23][24][25]

Основні конкуренти

Google Play Store працює на висококонкурентному ринку, стикаючись зі значною конкуренцією з боку кількох великих гравців:

- Apple App Store: Як основний магазин додатків для пристроїв iOS, Apple App Store є найсильнішим конкурентом Google Play. Він відомий своїм суворим процесом перевірки додатків, високими стандартами якості та прибутковою базою користувачів, особливо на розвинених ринках.

- Amazon Appstore: Amazon's Appstore - це альтернативна платформа для розповсюдження додатків для користувачів Android, зокрема тих, хто використовує пристрої Amazon's Fire. Він пропонує добірку додатків з особливим акцентом на контенті для сімейного перегляду.

- Samsung Galaxy Store: Власний магазин додатків Samsung, попередньо встановлений на пристроях компанії, пропонує додатки, теми та ексклюзивний контент, розроблений спеціально для користувачів Samsung. Він слугує додатковою платформою до Play Store для власників пристроїв Samsung.

#### Диференціація та конкурентні переваги

Незважаючи на конкуренцію, Google Play Store має кілька унікальних переваг, які допомагають йому утримувати лідируючі позиції:

- Широка екосистема додатків: Величезна бібліотека додатків Play Store, що налічує понад 3,5 мільйона додатків, надає користувачам широкий вибір, гарантуючи, що кожен знайде щось для себе.

- Глобальне охоплення: Доступність Play Store у понад 190 країнах світу та підтримка багатьох мов робить його доступним для різноманітної глобальної аудиторії.

- Бездоганна інтеграція з сервісами Google: Інтеграція Play Маркету з екосистемою Google, включаючи такі сервіси, як Google Диск, Gmail та Google Фото, покращує користувацький досвід та зручність.

#### Виклики та критика

#### Фрагментація та проблеми сумісності

Однією з головних проблем, з якими стикається Google Play Store, є фрагментація екосистеми Android. Зважаючи на велику кількість виробників

пристроїв та різні технічні характеристики, забезпечення сумісності та стабільної роботи на всіх пристроях є значним викликом. Фрагментація може призвести до суперечливого користувацького досвіду і труднощів для розробників в оптимізації своїх додатків для різних пристроїв.

#### Безпека та шкідливе програмне забезпечення

Незважаючи на зусилля Google, спрямовані на покращення безпеки, Play Store зазнає критики через наявність шкідливих програм. Випадки, коли шкідливе програмне забезпечення, рекламне ПЗ та інші шкідливі програми прослизали крізь процес перевірки, викликали занепокоєння щодо заходів безпеки платформи. Google Play Protect та інші ініціативи з безпеки певною мірою пом'якшили ці проблеми, але Play Store повинен постійно адаптуватися до нових загроз.

#### Регуляторні перевірки та антимонопольні питання

Компанія Google зіткнулася з регуляторними перевітками та антимонопольними розслідуваннями в різних регіонах, зокрема, щодо її практики, пов'язаної з магазином додатків Play Store. Такі питання, як плата за користування магазином додатків, преференційне ставлення до власних додатків Google та обов'язкове включення сервісів Google на пристроях Android, були предметом розгляду.

Android Studio, офіційне інтегроване середовище розробки (IDE) для операційної системи Android від Google, докорінно змінило спосіб створення, тестування та розгортання Android-додатків. З моменту своєї появи Android Studio стала незамінним інструментом для розробників, надаючи повний набір функцій та інструментів, призначених для спрощення процесу розробки додатків.

Тепер – про середовище розробки яке буде використовуватися для цієї роботи: Android Studio.

Історичний огляд[26][27]

Ранній початок та еволюція

Android Studio було вперше анонсовано на конференції Google I/O 16 травня 2013 року як заміну Eclipse Android Development Tools (ADT). Початковий попередній реліз був заснований на спільнотній версії IntelliJ IDEA від JetBrains, яка забезпечила надійну основу для нового IDE.

- Initial Preview (2013): Рання версія Android Studio включала базові функції, такі як система збірки на основі Gradle, нова структура проекту та інструменти Lint для аналізу коду.

- Стабільний випуск (2014): Перша стабільна версія, Android Studio 1.0, була випущена в грудні 2014 року. У ній було впроваджено значні покращення, включаючи новий інтерфейс, покращене редагування коду та інтеграцію з хмарною платформою Google.

Основні віхи розвитку

- Android Studio 2.0 (2016): Впроваджено миттєвий запуск, який дозволив розробникам бачити зміни у своєму додатку без його перезапуску. У цій версії також з'явився новий емулятор Android, який був швидшим та більш функціональним.

- Android Studio 3.0 (2017): Додана підтримка Kotlin, сучасної мови програмування, повністю сумісної з Java. У цій версії також з'явилася розширена підтримка миттєвих додатків - функції, що дозволяє користувачам запускати програми без їх встановлення.

- Android Studio 4.0 (2020): Зосереджена на інструментах дизайну, додано новий редактор руху та вдосконалено інспектор макетів. У цій версії також вдосконалено аналізатор збірок, щоб допомогти розробникам зрозуміти та усунути проблеми з конфігураціями їхніх збірок.

## . Безперервний розвиток

Протягом багатьох років Android Studio зазнала численних оновлень, кожне з яких додавало нові функції та покращення. IDE розвивається, щоб задовольнити зростаючі потреби спільноти розробників Android, враховуючи відгуки користувачів і технологічний прогрес. Регулярні оновлення гарантують, що Android Studio відповідає останнім тенденціям і вимогам мобільної розробки.

### Основні можливості Android Studio[28][29]

#### Інтелектуальний редактор коду

Однією з визначних особливостей Android Studio є інтелектуальний редактор коду. Редактор коду, що працює на основі IntelliJ IDEA від JetBrains, пропонує ряд функціональних можливостей, призначених для підвищення продуктивності та якості коду:

- Завершення коду: Android Studio забезпечує контекстно-орієнтоване завершення коду, пропонуючи відповідні пропозиції під час введення коду. Ця функція прискорює написання коду та зменшує кількість помилок.

- Рефакторинг: IDE включає в себе надійні інструменти рефакторингу, які дозволяють розробникам перейменовувати змінні, методи і класи, а також витягувати методи та інтерфейси. Ці інструменти допомагають підтримувати чисті та керовані кодові бази.

- Підсвічування синтаксису та виявлення помилок: Android Studio підсвічує синтаксичні помилки і забезпечує виявлення помилок в режимі реального часу, допомагаючи розробникам виявляти і виправляти проблеми на ранній стадії процесу розробки.

#### Редактор макетів та макет обмежень

Редактор макетів в Android Studio - це потужний інструмент візуального дизайну, який дозволяє розробникам з легкістю створювати складні макети:

- Інтерфейс перетягування: Розробники можуть використовувати

інтерфейс перетягування для додавання компонентів інтерфейсу користувача до макетів, зменшуючи потребу в ручному кодуванні XML.

- Макет обмежень: Представлений в Android Studio 2.2, Constraint Layout дозволяє розробникам створювати адаптивні та гнучкі макети з використанням обмежень. Цей менеджер макетів спрощує створення складних користувацьких інтерфейсів, які адаптуються до різних розмірів та орієнтації екрану.

### Система збірки на основі Gradle

Система збірки Android Studio базується на Gradle, універсальному інструменті автоматизації збірки. Система збірки на основі Gradle має декілька переваг:

- Гнучкі конфігурації збірки: Розробники можуть визначати власні конфігурації збірки для різних середовищ (наприклад, налагодження, випуск) і створювати варіанти збірки з різними налаштуваннями і ресурсами.

- Управління залежностями: Gradle спрощує управління залежностями, дозволяючи розробникам легко додавати бібліотеки та плагіни і керувати ними.

- Автоматизація збірки: Система збірки підтримує автоматизацію таких завдань, як компіляція коду, тестування та пакування, спрощуючи робочий процес розробки.

### Інтегрований емулятор та тестування пристроїв

Тестування є важливим аспектом розробки додатків для Android, і Android Studio пропонує комплексні інструменти для тестування та налагодження:

- Емулятор Android: Інтегрований емулятор дозволяє розробникам тестувати свої програми на різних віртуальних пристроях, імітуючи різні розміри екранів, роздільну здатність і версії Android. Емулятор підтримує розширені функції, такі як симуляція GPS, мультисенсорний ввід та зміна

орієнтації пристрою.

- Тестування на реальних пристроях: Android Studio підтримує тестування на реальних пристроях, підключених через USB або по мережі. Розробники можуть розгортати і тестувати свої програми на фізичних пристроях, щоб забезпечити сумісність і продуктивність.

- Автоматизоване тестування: IDE включає інструменти для написання і запуску модульних тестів, інтеграційних тестів і тестів інтерфейсу користувача. Бібліотека підтримки тестування для Android надає API для створення надійних тестів.

### Інтеграція з Firebase

Firebase - це комплексна платформа для розробки мобільних і веб-додатків, а Android Studio забезпечує безперешкодну інтеграцію з сервісами Firebase:

- Помічник Firebase: Помічник Firebase в Android Studio допомагає розробникам додавати сервіси Firebase до своїх додатків лише кількома кліками. Можна легко інтегрувати такі сервіси, як аналітика, автентифікація, хмарне сховище Firestore та Crashlytics.

- База даних у режимі реального часу: База даних Firebase в режимі реального часу дозволяє програмам синхронізувати дані між усіма клієнтами в режимі реального часу. Ця функція особливо корисна для додатків для спільної роботи та додатків, які потребують оновлення в режимі реального часу.

### Розширені інструменти профілювання

Android Studio містить розширені інструменти профілювання, які допомагають розробникам оптимізувати продуктивність своїх програм:

- CPU Profiler: Профайлер процесора дає уявлення про використання процесора у програмі, дозволяючи розробникам виявляти вузькі місця у продуктивності та оптимізувати виконання коду.

- Профайлер пам'яті: Профайлер пам'яті допомагає виявити витічки

пам'яті та надмірне використання пам'яті, забезпечуючи ефективне управління пам'яттю.

- Профайлер мережі: Профайлер мережі відстежує мережеву активність, надаючи детальну інформацію про дані, надіслані та отримані додатком. Цей інструмент необхідний для оптимізації використання мережі та підвищення продуктивності програми.

### Вплив на екосистему розробки Android

#### Прийняття розробниками та підтримка спільноти[30]

Android Studio був широко прийнятий спільнотою розробників Android, ставши стандартом де-факто для розробки додатків для Android. Широкий набір функцій, регулярні оновлення та потужна підтримка спільноти сприяли її популярності. Прагнення Google вдосконалювати Android Studio та реагувати на відгуки розробників ще більше зміцнило її позиції як найкращого IDE для Android розробників.

#### Освіта та навчання

Android Studio також відіграє важливу роль в освіті та навчанні. Багато університетів і таборів кодування використовують Android Studio як основний інструмент для навчання розробці додатків для Android. Обширна документація, навчальні посібники та онлайн-курси Google надають цінні ресурси для розробників усіх рівнів кваліфікації, від початківців до досвідчених професіоналів.

#### Внесок у розвиток відкритого вихідного коду

Android Studio базується на відкритому вихідному коді IntelliJ IDEA, і компанія Google зробила значний внесок у розвиток спільноти відкритого вихідного коду, розробивши Android Studio. Відкритий характер IDE заохочує до співпраці та інновацій, дозволяючи розробникам робити свій внесок у її розвиток та покращувати її можливості.

#### Інтеграція зі службами та API Google

Безшовна інтеграція Android Studio зі службами та API Google полегшує розробникам використання повного потенціалу платформи Android. Незалежно від того, чи це інтеграція Google Maps для сервісів на основі визначення місцезнаходження, використання сервісів Google Play для автентифікації та хмарного зберігання даних, чи включення Google Ads для монетизації, Android Studio надає інструменти та ресурси, необхідні для створення багатофункціональних та масштабованих додатків.

#### Вирішення проблеми фрагментації

Однією з постійних проблем в екосистемі Android є фрагментація, оскільки численні пристрої працюють під управлінням різних версій операційної системи Android. Android Studio зробила значний крок у вирішенні цієї проблеми, надавши інструменти для тестування на різних пристроях і конфігураціях. Однак, забезпечення сумісності та продуктивності на різноманітних пристроях Android залишається складним завданням для розробників.

#### Підвищення продуктивності та зручності використання

Оскільки Android Studio продовжує розвиватися, підвищення продуктивності та зручності використання залишається пріоритетом. Розробники часто прагнуть скоротити час збірки, підвищити стабільність і створити більш інтуїтивно зрозумілі інтерфейси. Постійні зусилля Google, спрямовані на оптимізацію IDE, зменшення споживання ресурсів та покращення користувацького досвіду, мають вирішальне значення для збереження лідерства у спільноті розробників.

#### Впровадження нових технологій

Швидкий темп технологічного прогресу створює як можливості, так і виклики для Android Studio. Нові технології, такі як штучний інтелект (AI), доповнена реальність (AR) та інтернет речей (IoT), змінюють ландшафт розробки додатків. Android Studio повинна продовжувати адаптуватися і надавати інструменти та фреймворки, які дозволять розробникам ефективно використовувати ці технології.

## Розширення крос-платформних можливостей

Оскільки попит на крос-платформну розробку зростає, зростає тиск на Android Studio щодо підтримки безперешкодної інтеграції з іншими платформами. Такі інструменти, як Flutter, що дозволяє розробляти нативно скомпільовані додатки для мобільних, веб та настільних комп'ютерів з єдиної кодової бази, набувають все більшої популярності. Хоча Android Studio вже підтримує розробку на Flutter, розширення цих можливостей і поліпшення інтеграції може залучити більше розробників, які шукають уніфіковане середовище розробки.

## Задоволення потреб розробників та зворотній зв'язок

Оперативне реагування Google на відгуки розробників стало наріжним каменем успіху Android Studio. Постійна взаємодія зі спільнотою розробників за допомогою форумів, опитувань і таких заходів, як Google I/O, має вирішальне значення для розуміння і задоволення мінливих потреб розробників. Функції, які покращують співпрацю, оптимізують робочі процеси та спрощують виконання складних завдань, залишатимуться важливими.

## Розширені функції та вдосконалення

### Інтеграція машинного навчання та штучного інтелекту

Оскільки штучний інтелект і машинне навчання стають все більш невід'ємною частиною розробки додатків, Android Studio почала включати інструменти для підтримки цих технологій:

- ML Kit: ML Kit, частина Firebase, надає функції машинного навчання, які можна легко інтегрувати в додатки для Android. Він пропонує API для розпізнавання тексту, розпізнавання облич, сканування штрих-кодів тощо.
- TensorFlow Lite: Для більш просунутих моделей машинного навчання TensorFlow Lite надає фреймворк для запуску моделей ML на мобільних пристроях. Android Studio підтримує TensorFlow Lite, що дозволяє розробникам створювати та розгортати власні моделі ML у своїх додатках.

## 2.2. Принципи розробки додатків із застосуванням Android Studio

Хоч більшість із них не є важливими для нашого завдання через його просту структуру та відсутню необхідність виходу на ринок, я вважаю необхідним у ході дослідження дізнатись та розповісти про основні принципи розробки додатків із застосуванням Android Studio.

Створення успішного додатку для Android вимагає більше, ніж просто навичок програмування. Це передбачає глибоке розуміння потреб користувачів, принципів дизайну, найкращих практик кодування, ефективних методів тестування та ефективних стратегій розгортання. Android Studio, офіційне інтегроване середовище розробки (IDE) для платформи Android від Google, надає повний набір інструментів для полегшення цього процесу.

Орієнтований на користувача дизайн[31]

Розуміння потреб користувачів

Основою будь-якого успішного додатку є глибоке розуміння його цільових користувачів та їхніх потреб. Розробники повинні брати участь у масштабних дослідженнях ринку, проводити інтерв'ю з користувачами та юзабіліті-тестування, щоб отримати уявлення про вподобання та поведінку користувачів. Такий підхід, орієнтований на користувача, гарантує, що додаток вирішує реальні проблеми та пропонує ефективні рішення.

- Дослідження ринку: Визначення цільової аудиторії та розуміння її больових точок, вподобань та поведінки.

- Інтерв'ю та опитування користувачів: Збір якісних даних для глибшого розуміння потреб та очікувань користувачів.

- Юзабіліті-тестування: Спостереження за тим, як користувачі взаємодіють з додатком, щоб виявити проблеми юзабіліті та сфери для покращення.

Проектування для зручності та простоти використання

Зручність та простота використання є критично важливими факторами

задоволеності користувачів. Додаток повинен бути інтуїтивно зрозумілим, легким для навігації та послідовним у дизайні інтерфейсу. Android Studio надає різні інструменти для підтримки юзабіліті-дизайну:

- Рекомендації з матеріального дизайну: Рекомендації Google з матеріального дизайну пропонують всеосяжну основу для проектування користувацьких інтерфейсів, які є одночасно естетично привабливими і функціональними. Ці настанови охоплюють такі аспекти, як кольорові схеми, типографіку та принципи макетування.

- Редактор макетів: Редактор макетів в Android Studio дозволяє розробникам візуально створювати і переглядати макети інтерфейсу. Цей інструмент підтримує функцію перетягування, що полегшує розташування компонентів інтерфейсу та встановлення обмежень.

- Інтерактивне прототипування: Такі інструменти, як Sketch і Figma, можна використовувати разом з Android Studio для створення інтерактивних прототипів, які можна протестувати з реальними користувачами перед початком розробки.

#### Міркування щодо доступності

Забезпечення доступності програми для всіх користувачів, включаючи людей з обмеженими можливостями, є як юридичною вимогою, так і моральним обов'язком. Android Studio підтримує такі функції доступності, як

- Описи вмісту: Розробники можуть додавати описи вмісту до елементів інтерфейсу, щоб зробити їх доступними для зчитування з екрану.

- Інструменти тестування доступності: Android Studio включає такі інструменти, як сканер доступності, який допомагає виявити потенційні проблеми доступності в додатку.

- Настроювані компоненти інтерфейсу: Надання опцій для зміни розміру тексту, режимів високої контрастності та інших функцій доступності покращує користувацький досвід для всіх користувачів.

## Якість коду та зручність супроводу

Написання чистого коду, який легко підтримувати, має вирішальне значення для довгострокового успіху програми. Android Studio надає кілька функцій, які сприяють правильному кодуванню:

- Інструмент Lint: Інструмент Lint аналізує вихідний код, щоб виявити потенційні помилки і «неприємні запахи» коду. Він допомагає забезпечити дотримання стандартів і найкращих практик кодування.

- Інструменти рефакторингу: Android Studio включає потужні інструменти рефакторингу, які дозволяють розробникам реструктурувати код без зміни його поведінки. Це дуже важливо для підтримки чистоти кодової бази.

- Рецензування коду та парне програмування: Рецензування коду та парне програмування може ще більше підвищити якість коду та забезпечити дотримання стандартів кодування.

## Модульна архітектура[32]

Модульна архітектура покращує організацію коду та можливість його повторного використання. Розбиваючи додаток на менші, автономні модулі, розробники можуть ефективніше керувати складністю та полегшити паралельну розробку. Популярні архітектурні патерни для розробки Android включають

- Model-View-ViewModel (MVVM): MVVM відокремлює інтерфейс користувача (View) від бізнес-логіки (ViewModel) і даних (Model). Таке розділення покращує тестованість і супроводжуваність.

- Ін'єкція залежностей: Такі інструменти, як Dagger і Hilt, допомагають керувати залежностями за модульним принципом, роблячи код більш гнучким і легким для тестування.

- Чиста архітектура: Прийняття принципів чистої архітектури, таких як запропоновані Робертом Мартіном (Robert C. Martin), може ще більше покращити організацію коду та розподіл завдань.

### Використання сучасних практик розробки[33]

Використання новітніх практик розробки гарантує, що додаток використовує весь потенціал платформи Android. Деякі сучасні практики включають:

- Мова програмування Kotlin: Kotlin, сучасна мова програмування, повністю сумісна з Java, пропонує ряд переваг, таких як нульова безпека, функції розширення та підпрограми для асинхронного програмування. Android Studio надає повну підтримку Kotlin.

- Бібліотеки Jetpack: Android Jetpack - це набір бібліотек, які допомагають розробникам слідувати найкращим практикам, зменшувати шаблонний код та писати надійні додатки. Такі бібліотеки, як LiveData, Navigation та Room надають готові рішення для поширених завдань розробки.

- Реактивне програмування: Використання фреймворків, таких як RxJava або Kotlin Flow, для реактивного програмування може покращити швидкість реагування та масштабованість додатку.

### Автоматизоване тестування[34]

Автоматизоване тестування має важливе значення для забезпечення якості та надійності програми. Android Studio підтримує різні типи автоматизованих тестів:

- Юніт-тести: Юніт-тести перевіряють функціональність окремих компонентів. Такі інструменти, як JUnit і Mockito, зазвичай використовуються для модульного тестування в Android.

- UI-тести: UI-тести перевіряють користувацький інтерфейс програми та взаємодію з користувачем. Espresso - популярний фреймворк для написання надійних і підтримуваних UI-тестів в Android Studio.

- Інтеграційні тести: Інтеграційні тести гарантують, що різні компоненти програми працюють разом, як очікувалося. Вони мають вирішальне значення для виявлення проблем, які можуть бути неочевидними в ізольованих модульних тестах.

- Безперервне тестування: Інтеграція автоматизованих тестів в конвеєр безперервної інтеграції (CI) гарантує, що тести запускаються регулярно, виявляючи проблеми на ранній стадії і підтримуючи якість коду.

#### Безперервна інтеграція та безперервне розгортання (CI/CD)

Впровадження конвеєра CI/CD автоматизує процес створення, тестування та розгортання програми. Ця практика допомагає виявити проблеми на ранніх стадіях і гарантує, що додаток завжди знаходиться в стані, придатному для розгортання. Android Studio добре інтегрується з такими інструментами CI/CD, як Jenkins, CircleCI та GitHub Actions.

- Автоматизовані процеси збірки: Автоматизація процесу збірки забезпечує узгодженість і надійність.

- Конвеєри розгортання: Налаштування конвеєрів розгортання для автоматизації процесу випуску, від стадії розробки до виробництва.

- Моніторинг та оповіщення: Впровадження механізмів моніторингу та оповіщення для забезпечення працездатності конвеєра CI/CD.

#### Інструменти профілювання

Продуктивність є критично важливим аспектом користувацького досвіду. Android Studio містить кілька інструментів профілювання, які допомагають виявити та вирішити проблеми з продуктивністю:

- CPU Profiler: Аналізує використання процесора у програмі та допомагає виявити вузькі місця у продуктивності.

- Профайлер пам'яті: Відстежує використання пам'яті програми, щоб виявити витoki та оптимізувати управління пам'яттю.

- Профайлер мережі: Відстежує мережеві запити та відповіді, допомагаючи оптимізувати використання мережі та усунути несправності.

- Профайлер енергоспоживання: Відстежує енергоспоживання програми, допомагаючи виявити та усунути поведінку, що призводить до виснаження

акумулятора.

### Ефективне управління ресурсами

Ефективне керування ресурсами, такими як зображення, макети та рядки, може суттєво вплинути на продуктивність програми. Android Studio надає інструменти для оптимізації використання ресурсів:

- Векторні малюнки: Використання векторних рисунків замість растрових зображень зменшує розмір програми та покращує продуктивність на екранах різної щільності.

- Кваліфікатори ресурсів: Розробники можуть використовувати класифікатори ресурсів для надання альтернативних ресурсів для різних конфігурацій пристроїв, забезпечуючи оптимальну продуктивність на широкому діапазоні пристроїв.

- Оптимізація завантаження ресурсів: Впровадження механізмів лінивого завантаження та кешування для оптимізації завантаження ресурсів і підвищення продуктивності програми.

### Підготовка до випуску

Перш ніж випустити додаток, розробники повинні переконатися, що він відповідає стандартам якості та вимогам цільової платформи. Android Studio надає кілька інструментів, які допоможуть у цьому процесі:

- Варіанти збірки: Розробники можуть визначати різні варіанти збірки для різних середовищ (наприклад, налагодження, випуск) з різними налаштуваннями і ресурсами.

- ProGuard: ProGuard - це інструмент, який зменшує, оптимізує та заплутує код, що ускладнює реінжиніринг та зменшує розмір програми.

- Управління релізами: Впровадження практик управління релізами, таких як контроль версій та управління змінами, для забезпечення безперебійного процесу випуску.

Публікація додатку в Google Play складається з кількох кроків:

- Підписання додатку: Перед публікацією розробники повинні підписати додаток за допомогою ключа випуску. Android Studio надає інструменти для керування підписанням програми.

- Консоль Google Play: Консоль Google Play використовується для завантаження програми, управління випусками, а також для моніторингу продуктивності програми та відгуків користувачів.

- Бета-тестування: Проведення бета-тестування через Google Play дозволяє розробникам збирати відгуки від реальних користувачів і виявляти будь-які проблеми, що залишилися до повного релізу.

- Оптимізація App Store (ASO): Впровадження стратегій ASO для покращення видимості та пошуку додатку в магазині Google Play.

#### Моніторинг та оновлення після релізу

Моніторинг роботи додатку та відгуки користувачів після релізу мають вирішальне значення для подальшого успіху. Розробники повинні регулярно оновлювати додаток, щоб виправляти помилки, додавати нові функції та покращувати продуктивність. Android Studio інтегрується з Firebase Crashlytics для звітування та аналітики збоїв у режимі реального часу, що допомагає розробникам швидко вирішувати проблеми.

- Відгуки користувачів: Збір та аналіз відгуків користувачів для визначення областей для покращення та запитів на нові функції.

- Моніторинг продуктивності: Використання таких інструментів, як Firebase Performance Monitoring, для відстеження продуктивності програми в реальних умовах.

- Регулярні оновлення: Впровадження регулярного циклу оновлень для постійного вдосконалення програми та задоволення потреб користувачів.

### 2.3 Додаток «Список покупок»

Нарешті ми перейшли до створення додатку. У цьому розділі будуть пояснені основні рішення у процесі створення програми.

Для створення додатку я використовував мову програмування Java і середовище розробки Android Studio

Для більш гнучкого розташування об'єктів на екрані мною було використано «RelativeLayout»: Контейнер, що дозволяє розташовувати дочірні елементи відносно один одного або самого контейнера.

Для Відображення напису «Список покупок» у верхній частині екрану використовується «TextView»: Нерухоме текстове поле, яке використовується для відображення тексту.

Кнопки «Додати поле», «Видалити поле», «Закрити» та «Видалити» використовують функцію Button: Основна кнопка, яку можна натискати для виконання дії.

Поля, що створюються при натисканні кнопки «Додати поле» належать типу «EditText»: Редаговане текстове поле, яке дозволяє користувачам вводити текст.

Стандартний текст «Введіть назву товару» створений за допомогою підказки «hint». Коли поле вводу пусте, підказка відображається всередині поля у вигляді напівпрозорого тексту. Коли користувач починає вводити текст, підказка зникає.

Для того щоб програма була однаково зручною і функціональною незалежно від кількості полів створених користувачем всі поля що створюються при натисканні кнопки «Додати поле» поміщені всередині контейнеру ScrollView. ScrollView - Контейнер, який дозволяє прокручувати вміст, якщо він не поміщається на екрані.

У середовищі Android studio визначення кольору об'єкту чи тексту, що він відображує вважається помилкою. Тому всі об'єкти для визначення цих

параметрів ссилаються на ім'я відповідного ресурсу у файлі «colors» для кольорів, або «strings» для тексту.

Приклади:

```
«android:text="@string/addbutton"»;
```

```
«android:backgroundTint="@color/green"»;
```

```
«android:textColor="@color/black"»
```

При натисканні на будь яку із кнопок спрацює функція «public void onClick(View v)» яка далі викликає функцію відповідну натиснутій кнопці, а саме:

Для кнопки «Додати поле»:

```
private void addNewEditText()
```

Ця функція при виклику створює нове редагуєме текстове поле з підказкою. В свою чергу кожне текстове поле має кнопку «Видалити», що при натисканні викликає функцію

```
public void onClick(View v) {container.removeView(newLayout); }
```

Ця функція видаляє лише поле на якому знаходилась сама кнопка.

Для кнопки «Видалити поле»:

```
private void removeLastEditText()
```

Ця функція видаляє останнє створене текстове поле.

І нарешті кнопка «Закрити» викликає функцію:

```
private void closeApp()
```

Ця функція закриває додаток, що також стирає всі поля і дані, що вписав користувач.

Приклади роботи додатку:

Тут можна побачити як виглядає додаток одразу після запуску, коли користувач ніяк з ним не взаємодіє(Рис.2.1.) та після створення пустих полів для введення тексту(Рис.2.2.).

Список покупок



Рис.2.1. головний екран додатку

Список покупок

Введіть назву товару

Введіть назву товару

Введіть назву товару

Введіть назву товару

Введіть назву товару

Введіть назву товару

Введіть назву товару

Введіть назву товару

Введіть назву товару

Введіть назву товару

Введіть назву товару

Введіть назву товару



Рис.2.2. створення пустих полів для введення тексту

Тут можна побачити роботу контейнеру `ScrollView`, що дозволяє створювати більше полів ніж може уміститися на екрані і продивлятися їх, прокручуючи список вертикально(Рис.2.3.),(Рис.2.4.).

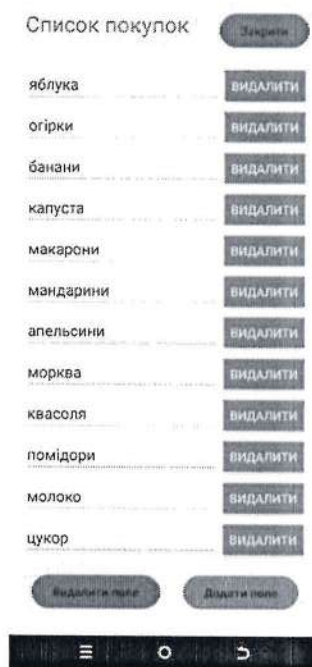


Рис.2.3. демонстрація роботи `ScrollView`. 1 частина.



Рис.2.4. демонстрація роботи `ScrollView`. 2 частина.

## Висновок

У сучасному швидкоплинному і часто хаотичному світі список покупок залишається життєво важливим інструментом з багатьох причин. По-перше, він забезпечує ефективність і організованість, дозволяючи людям систематично підходити до своїх завдань по покупкам. Це особливо важливо з огляду на широкий вибір доступних продуктів і щільний графік більшості людей. Завдяки добре підготовленому списку покупок покупці можуть уникнути пасток імпульсивних покупок і дотримуватися запланованих покупок, заощаджуючи таким чином час і гроші.

Крім того, списки покупок суттєво сприяють кращому управлінню фінансами. Вказуючи, що саме потрібно, вони допомагають запобігти непотрібним витратам і перекупкам. Така фінансова дисципліна особливо важлива в епоху, коли споживацтво процвітає, а маркетингова тактика постійно заохочує до незапланованих покупок. Детальний список покупок може слугувати захистом від цих впливів, сприяючи усвідомленому витрачання коштів та дотриманню бюджету.

Використання списків покупок також приносить користь здоров'ю та харчуванню. Заздалегідь плануючи харчування та потреби в продуктах, люди можуть робити більш свідомий вибір продуктів, які вони купують, підтримуючи здорові харчові звички. Такий проактивний підхід до покупок може зменшити спокусу купувати нездорові закуски та напівфабрикати, що в кінцевому підсумку сприяє покращенню загального самопочуття.

В довгостроковій перспективі списки покупок можуть допомогти зменшити кількість відходів. Ретельне планування покупок означає менше забутих речей, які будуть томитися невикористаними, і менше харчових відходів загалом. Це не лише приносить користь окремим домогосподарствам, але й сприяє ширшим зусиллям зі збереження довкілля, мінімізуючи утворення відходів.

Отже, список покупок - це простий, але потужний інструмент, який підвищує ефективність, управління фінансами, здоров'я та сталий розвиток. Його актуальність у сучасному суспільстві неможливо переоцінити, адже він допомагає людям легше та цілеспрямованіше орієнтуватися у складнощах повсякденного життя.

## Література

1. Топ-10 продуктових мереж України за кількістю працюючих магазинів у 2023 році – <https://rau.ua/novyni/top-10-produktovih-merezh-2023/>
2. 99 Online Shopping Statistics 2024 – Total Users, Spending – <https://www.yaguara.co/online-shopping-statistics/>
3. Як правильно ходити в продуктовий магазин і не купувати зайвого. Інструкція.– <https://life.pravda.com.ua/society/2021/11/20/246588/>
4. How Corporate Social Responsibility Influences Buying Decisions - <https://clutch.co/resources/how-corporate-social-responsibility-influences-buying-decisions>
5. Vance, Ashlee. "Google Buys Android for Its Mobile Arsenal." The New York Times, 2005
6. "Android History." Android Developers. - <https://developer.android.com>
7. "Open Handset Alliance." Open Handset Alliance. - <https://www.openhandsetalliance.com>
8. "HTC Dream." CNET. - <https://www.cnet.com>
9. "Android Cupcake." Engadget. - <https://www.engadget.com>
10. "Android Donut." TechCrunch. - <https://techcrunch.com>
11. "Nexus One." The Verge. - <https://www.theverge.com>
12. "Android Ice Cream Sandwich." Ars Technica. - <https://arstechnica.com>
13. "Android Jelly Bean." PCMag. - <https://www.pcmag.com>
14. "Android Lollipop." Wired. - <https://www.wired.com>
15. "Android Marshmallow." Android Authority. - <https://www.androidauthority.com>
16. "Android Nougat." Gizmodo. - <https://gizmodo.com>
17. "Android Oreo." The Guardian. - <https://www.theguardian.com>
18. "Android Pie." Forbes. - <https://www.forbes.com>
19. "Android 10." ZDNet. - <https://www.zdnet.com>
20. "Android 11." Digital Trends. - <https://www.digitaltrends.com>
21. Google's Official Blog - <https://blog.google/>
22. Mobile app usage - Statistics & Facts – Statista -

- <https://www.statista.com/topics/1002/mobile-app-usage/>
23. TechCrunch - <https://techcrunch.com/>
  24. IDC - Market Intelligence - <https://www.idc.com/>
  25. Gartner - Market Research - <https://www.gartner.com/en>
  26. Android Developer Blog - <https://android-developers.googleblog.com/>
  27. Android Studio Release Notes - <https://developer.android.com/studio/releases>
  28. Official Android Developer Website - <https://developer.android.com/>
  29. JetBrains IntelliJ IDEA - <https://www.jetbrains.com/idea/>
  30. Stack Overflow - Android Studio -  
<https://stackoverflow.com/questions/tagged/android-studio>
  31. Layout Editor - <https://developer.android.com/studio/write/layout-editor>
  32. Android developer guide -  
<https://developer.android.com/topic/architecture#recommended-app-arch>
  33. Kotlin - <https://kotlinlang.org/>
  34. Android testing documentation -  
<https://developer.android.com/training/testing/fundamentals>

## Код файлу activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:background="@color/white"
    >

    <TextView
        android:id="@+id/logo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:text="@string/logo"
        android:textAlignment="viewStart"

        android:textColor="@color/green"
        android:textSize="25sp" />

    <Button
        android:id="@+id/add_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/addbutton"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true"
        android:layout_marginBottom="10sp"
        android:layout_marginEnd="10sp"
        android:backgroundTint="@color/green"
        android:textColor="@color/black"
        />

    <Button
        android:id="@+id/remove_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/removebutton"
        android:layout_alignParentBottom="true"
        android:layout_alignParentStart="true"
        android:layout_marginStart="10sp"
        android:layout_marginBottom="10sp"
        android:backgroundTint="@color/red"
        android:textColor="@color/black"
        />

    <Button
        android:id="@+id/close_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/closebutton"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"

```

```
        android:backgroundTint="@color/red"
        android:textColor="@color/black"
    />
<ScrollView
    android:id="@+id/scroll_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/close_button"
    android:layout_alignBottom="@id/add_button"
    android:layout_marginBottom="60sp"
    android:layout_marginTop="16dp"
    android:contentDescription="Scrollable container for text fields">

    <LinearLayout
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" />
</ScrollView>
</RelativeLayout>
```

## Код файлу MainActivity.java

```
package com.example.myapplication;

import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private LinearLayout container;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        container = findViewById(R.id.container);
        Button addButton = findViewById(R.id.add_button);
        Button removeButton = findViewById(R.id.remove_button);
        Button closeButton = findViewById(R.id.close_button);

        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                addNewEditText();
            }
        });

        removeButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                removeLastEditText();
            }
        });

        closeButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                closeApp();
            }
        });
    }

    private void addNewEditText() {
        LinearLayout newLayout = new LinearLayout(this);
        newLayout.setOrientation(LinearLayout.HORIZONTAL);
        newLayout.setLayoutParams(new LinearLayout.LayoutParams(
```

```

        LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT));

    EditText newEditText = new EditText(this);
    newEditText.setLayoutParams(new LinearLayout.LayoutParams(
        0,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1));
    newEditText.setHint("Введіть назву товару");
    newEditText.setHintTextColor(Color.GRAY);
    newEditText.setTextColor(Color.BLACK);

    Button deleteButton = new Button(this);
    deleteButton.setText("Видалити");
    deleteButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            container.removeView(newLayout);
        }
    });

    newLayout.addView(newEditText);
    newLayout.addView(deleteButton);

    container.addView(newLayout);
}

private void removeLastEditText() {
    int childCount = container.getChildCount();
    if (childCount > 0) {
        container.removeViewAt(childCount - 1);
    }
}

private void closeApp() {
    finish();
    System.exit(0);
}
}

```

**Код файлу colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="black">#FF000000</color>
  <color name="my_light_primary">#FF000000</color>
  <color name="white">#FFFFFFFF</color>
  <color name="purple_200">#FFBB86FC</color>
  <color name="purple_500">#FF6200EE</color>
  <color name="purple_700">#FF3700B3</color>
  <color name="teal_200">#FF03DAC5</color>
  <color name="teal_700">#FF018786</color>
  <color name="background">#D3D3D3</color>
  <color name="green">#008000</color>
  <color name="red">#FF0000</color>
</resources>
```

**Код файлу strings.xml**

```
<resources>
  <string name="app_name">Список покупок</string>
  <string name="logo">Список покупок</string>
  <string name="addbutton">Додати поле</string>
  <string name="closebutton">Закрити</string>
  <string name="removebutton">Видалити поле</string>

</resources>
```

Код файлу AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.MyApplication"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```