

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

на тему: «Розробка мобільного додатку "Доступний менеджер покупок" на базі ОС Android»

Виконав: студент 4 курсу групи Іт-41

Спеціальності 126 «Інформаційні системи та технології»

(шифр і назва)

Драгула Роман Васильович

(Прізвище та ініціали)

Керівник: к.е.н., доц. Шувар Б.І.

(Прізвище та ініціали)

Рецензент: к.т.н., доц. Шарібур А.О.

(Прізвище та ініціали)

ДУБЛЯНИ-2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

д.т.н., професор, Тригуба А. М.

(вч. звання, прізвище, ініціали)

“ _____ ” _____ 202__ року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Драгула Роман Васильович

(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка мобільного додатку "Доступний менеджер покупок" на базі ОС Android»

керівник роботи к. е. н., доцент, Шувар Б.І.

(наук. ступінь, вч. звання, прізвище, ініціали)

затверджені наказом Львівського НУП № 453 / к - с від 30.12.2022 р

2. Строк подання студентом роботи 09.06.2023 р.

3. Вихідні дані: алгоритми розробки мобільного застосунку; використання бібліотек для реалізації засобів використання на основі базової документації науково-технічна і довідкова література.

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)

Вступ

1. Постановка задачі дослідження

2. Вибір засобів розробки мобільного додатку.

3. Розробка мобільного додатку «Доступний менеджер покупок».

4. Охорона праці.

Висновки.

Список використаної літератури.

5. Перелік графічного матеріалу

Графічний матеріал подається у вигляді презентації

6. Консультанти розділів

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3	<i>к.е.н., доцент Шувар Б.І.</i>			
4	<i>к.т.н., доцент Городецький І.М.</i>			

7. Дата видачі завдання 30.12.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Відмітка про виконання
1	<i>Отримання завдання. Вивчення рекомендованої літератури по темі роботи. Написання аналітичного огляду предметної області.</i>	<i>30.12.2022 – 03.02.2023</i>	
2	<i>Проектування та опис технічного завдання, функціональних вимог та технічної сторони реалізації проекту (написання проектної частини).</i>	<i>06.02.2023 – 03.03.2023</i>	
3	<i>Програмна реалізація проекту (вибір мови програмування, розробка сайту, тестування його роботи, розрахунок ефективності проекту)</i>	<i>06.03.2023 – 14.04.2023</i>	
4	<i>Розгляд питань з охорони праці та безпеки у надзвичайних ситуаціях</i>	<i>17.04.2023 – 05.05.2023</i>	
5	<i>Завершення оформлення основної частини, написання висновків та підготовка презентаційного матеріалу</i>	<i>08.05.2023 – 19.05.2023</i>	
6	<i>Завершення роботи в цілому. Підготовка до захисту кваліфікаційної роботи</i>	<i>22.05.2023 – 09.06.2023</i>	

Студент

(підпис)

Драгула Р.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Шувар Б.І.

(прізвище та ініціали)

УДК 004.75

Розробка мобільного додатку «Доступний менеджер покупок» на базі ОС Android.

Драгула Р.В. Кафедра ІТ – Дубляни, Львівський НАУ, 2023.

Кваліфікаційна робота: ббс. текст. част., 12 рис., 4 табл., 12 арк.

ілюстраційного матеріалу, 38 джерел.

Подано теоретичні основи розробки мобільних додатків. Виконано опис предметної області. Здійснено огляд і аналіз існуючих аналогів.

Наведено особливості вибору засобів розробки мобільного додатка. Виконано вибір мови програмування Java. Подані характеристики існуючих інструментів розробника на Android. Підібрано компоненти android додатку. Проаналізовано життєвий цикл Activity. Здійснено опис та особливості роботи з базою даних Sqlite. Подано особливості передачі даних в базі даних за допомогою формату Json. Наседено особливості використання Tesseract для оптичного розпізнавання символів.

Описано основні етапи створення додатку «Доступний менеджер покупок». Здійснено розробку інтерфейсу користувача. Створено код розмітки для графічного інтерфейсу користувача (Gui) на платформі Android. Створено коду для формування списку завдань (to-do list) мобільного додатку Android.

Запропоновано заходи із охорони праці під час розробки мобільного додатку.

Ключові слова: Мобільний додаток, проектування, покупки, Android, дані.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Теоретичні основи розробки мобільних додатків.....	9
1.2. Опис предметної області	14
1.3. Огляд і аналіз існуючих аналогів	17
РОЗДІЛ 2. ВИБІР ЗАСОБІВ РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ ..	24
2.1. Особливості вибору засобів розробки мобільного додатка.....	24
2.2. Мова програмування Java.....	26
2.3. Характеристика існуючих інструментів розробника на Android	27
2.4. Компоненти android додатку.....	31
2.5. Життєвий цикл Activity	32
2.6. Опис та особливості роботи з базою даних Sqlite	35
2.7. Передача даних в базі даних за допомогою формату Json	36
2.8. Особливості використання tesseract для оптичного розпізнавання символів	38
РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ «ДОСТУПНИЙ МЕНЕДЖЕР ПОКУПОК».....	39
3.1. Основні етапи створення додатку «доступний менеджер покупок»	39
3.2. Розробка інтерфейсу користувача	41
3.3. Створення коду розмітки для графічного інтерфейсу користувача (gui) на платформі android.....	42
3.4. Створення коду для формування списку завдань (to-do list) мобільного додатку Android	46
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	49
4.1. Аналіз умов праці під час розроблення мобільного додатку	49
4.2. Організаційні та конструктивно-технологічні заходи для зниження впливу шкідливих виробничих факторів.....	51
4.3. Аналіз інформаційних небезпек під час розроблення мобільного	

додатку.....	52
4.4. Пожежна безпека.....	54
4.5. Заходи з охорони праці під час розробки мобільного додатку	55
ВИСНОВКИ І ПРОПОЗИЦІЇ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТКИ	63

ВСТУП

Сьогодні більшість населення світу активно використовують мобільні додатки в буденній діяльності для вирішення ряду завдань. Однак на самому початку їх появи мобільні додатки надавали інформаційну підтримку загального призначення та обмежені інформаційні послуги в глобальній мережі, такі як електронна пошта, календар, фондовий ринок, списки та інформацію про погоду [18].

Зростання попиту користувачів мобільних пристроїв разом із технічними можливостями розробки мобільних додатків обумовили поширення їх практичного застосування на інші категорії, такі як мобільні ігри, автоматизація виробництва, GPS, особисте здоров'я, фінансові послуги, сільське господарство, торгівля, перевезення тощо. Вибухове зростання кількості та різноманітності додатків охопило великі та різноманітні сфери. Це призвело до того, що сьогодні складно уявити ряд послуг та галузей без використання мобільних додатків [22]. Завдяки ним визначення місцезнаходження та інтернет-банкінг, відстеження та купівля квитків, мобільних медичних послуг, сигнали оповіщення повітряної тривоги, керування дронами і т.д. стали звичними речами.

Метою кваліфікаційної роботи є створення зручного та комфортного мобільного додатку «Доступний менеджер покупок», який може мати практичне застосування як для споживачів товарів і послуг, так і для представників бізнесу.

Для виконання кваліфікаційної роботи були визначені наступні завдання:

1. Дослідити та узагальнити теоретичні та методологічні аспекти, основні тенденції та підходи у створенні мобільних додатків.
2. Дослідити прикладну область, вивчення основні вимоги до мобільного додатку, вибрати технологію для його розробки.

3. Практично реалізувати проект по розробці мобільного додатку для здійснення покупок, протестувати його на коректність роботи та оцінити ефективність і перспективи застосування.

Загалом кваліфікаційна робота складається з чотирьох розділів. Перший розділ присвячений теоретичним основам розробки мобільних додатків, описано предметну область дослідження, здійснено аналіз існуючих аналогів. У другому розділі здійснено проектування логічної структури мобільного додатку, описані технології для розробки та особливості платформи. В третьому розділі розкрито питання програмої реалізації проекту з розробки мобільного додатку «Доступний менеджер покупок». У четвертому розділі описані питання охорони праці.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Теоретичні основи розробки мобільних додатків

Масовий доступ до мобільного інтернету та поширення використання смартфонів, планшетів призвели до того, що в повсякденне життя більшості користувачів активно увійшли мобільні додатки, які активно використовуються в різних сферах життя пересічної людини та бізнесі.

Мобільні додатки отримали свій розвиток від інтегрованих програмних систем, які традиційно є на персональних комп'ютерах або ноутбуках.

Найпростіші мобільні програми на початку брали додатки на базі ПК і інтегрували їх на мобільний пристрій. Оскільки мобільні додатки розвиваються, ця стратегія є виявилася недосконалою. Сучасніша методологія передбачає розробку безпосередньо для мобільного середовища, використовуючи його обмеження та переваги.

Загалом сьогодні під мобільними додатками розуміють програмний додаток, розроблений спеціально для використання на невеликих бездротових обчислювальних пристроях, таких як смартфони та планшети.

Експерти ІТ-галузі виділяють такі три основні типи мобільних програм (додатків), якщо їх класифікувати за технологією, яка була використана для кодування [1]:

1. Власні додатки: створюються для однієї конкретної платформи чи операційної системи.

2. Веб-додатки – це адаптивні версії веб-сайтів, які можуть працювати на будь-якому мобільному пристрої чи операційній системі, оскільки вони доступні за допомогою мобільного браузера.

3. Гібридні додатки – це комбінації як нативних, так і веб-програм, але включені в нативну програму, що дає їй можливість мати власну піктограму або завантажуватися з магазину програм.

В таблиці 1.1. представлено порівняльний аналіз основних типів мобільних додатків, виходячи із використаної технології.

Таблиця 1.1 – Порівняльний аналіз мобільних додатків на основі технологій розробки

№ п/п	Назва типу мобільного додатку	Технологія	Сильні сторони мобільного додатку	Слабкі сторони мобільного додатку
1	Native Apps (власні додатки)	Java, Kotlin, Python, Swift, Objective-C, C++, React.	Швидші та надійніші з точки зору продуктивності. Надають оптимізовану взаємодію з клієнтами	Більші витрати на розробку. Складності у підтримці та оновленні кодової бази для кожної версії.
2	Web Apps (веб-додатки)	HTML5, CSS, JavaScript, Ruby	Немає необхідності налаштовувати додаток під платформу чи ОС. Менші витрати на розробку. Користувачам не потрібно завантажувати оновлення в магазині програм.	Веб-додатки повністю залежать від браузера, який використовується на пристрої. Не працюватимуть повністю в автономному режимі.
3	Hybrid Apps (гібридні додатки)	Використовують суміш веб-технологій і власних API. Ionic, Objective C, Swift, HTML5	Створення гібридної програми набагато швидше та економніше. Гібридна програма може бути мінімально життєздатним продуктом. Ці додатки швидко завантажуються, ідеально підходять для використання в країнах із повільнішим Інтернет-з'єднанням і надають користувачам узгоджену взаємодію	Може бракувати потужності та швидкості.

Джерело: узагальнено на основі [1]

Власні додатки часто написані для використання на певній платформі чи пристрої. Узагальнені дані говорять, що на iOS від Apple і Android від Google припадає 99% мобільних пристроїв [2]. Через це власні додатки (рідні програми) можуть використовувати переваги апаратного та програмного забезпечення пристрою (наприклад, камери). Зазвичай вони високопродуктивні та пропонують мобільному користувачеві багатий досвід. Для роботи мобільних веб-додатків потрібен лише веб-браузер. На відміну від власних програм, прогресивні веб-програми працюватимуть на будь-якій мобільній платформі.

Гібридний додаток поєднує в собі аспекти рідних і веб-програм. Фактично гібридні додатки є веб-програмою, поміщеною в оболонку (або контейнер) рідної програми.

В таблиці 1.2. представлені найбільш поширені види мобільних додатків та типом використання.

Таблиця 1.2 – Порівняльний аналіз типів мобільних додатків

Назва типу	Опис
Ігрові програми	Еквівалент комп'ютерних відеоігор. Вони є одними з найпопулярніших типів програм. На них припадає одна третина всіх завантажень додатків і три чверті всіх споживчих витрат.
Програми для підвищення продуктивності	Зосереджені на підвищенні ефективності бізнесу, спрощуючи різні завдання, такі як надсилання електронних листів, відстеження прогресу роботи, бронювання готелів і багато іншого.
Програми для способу життя та розваг	Охоплюють багато аспектів особистого способу життя та спілкування, таких як побачення, спілкування в соціальних мережах, а також обмін (і перегляд) відео. Деякі з найвідоміших програм, таких як Netflix, Facebook або TikTok, потрапляють до цієї категорії.
Додатки мобільної комерції	Використовуються для покупки товарів в Інтернеті, наприклад Amazon або eBay.
Додатки для подорожей	Допомагають мандрівникам багатьма способами (бронювання турів і квитків, знаходження шляху за допомогою карт і геолокації, щоденники подорожей тощо).
Службові програми	Програми, які виконують вузькопрофільні функції (наприклад для здоров'я та сканери штрих-кодів).

Джерело: узагальнено на основі [3]

Розповсюдження мобільних додатків. Після створення мобільного додатку (гібридного та рідного), його необхідно розмістити в магазині програм. Двома основними магазинами мобільних додатків є Google Play і Apple App Store. Також Amazon (для своїх пристроїв Kindle), і Samsung мають магазини мобільних додатків, однак вибір у них значно менший.

На рис. 1.1. представлено найбільш популярні мобільних додатків за категоріями Apple App Store за часткою доступності у 2021 році.

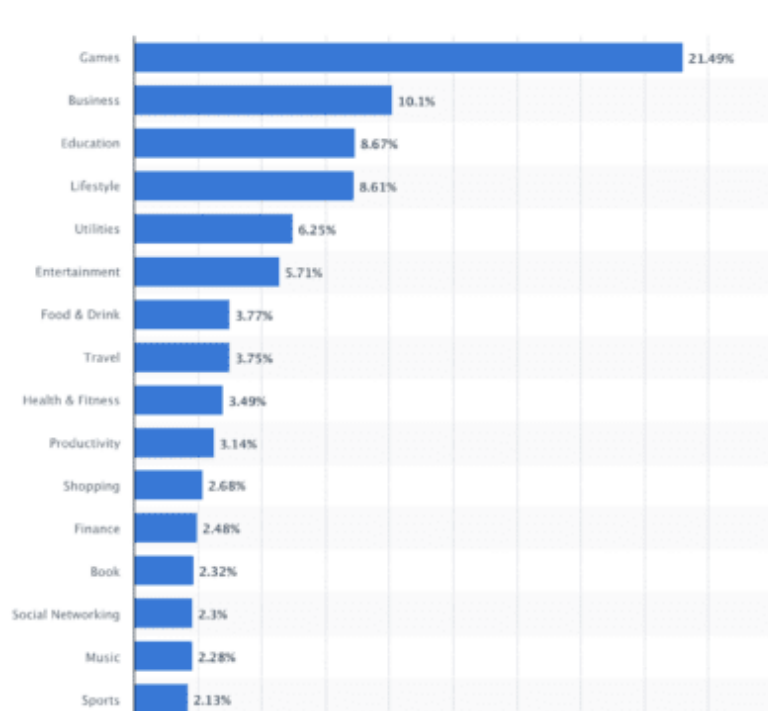


Рисунок 1.1 – Розподіл мобільних додатків за категоріями Apple App Store, 2021 р [5]

Популярність мобільних платформ та мобільних додатків постійно зростає, що призводить у виникненні потреби в постійному оновленні ринку мобільних додатків, які мають відповідати сучасним вимогам.

Мобільні додатки допомагають компаніям безпосередньо спілкуватися зі своїми клієнтами за допомогою додаткових цінностей на кожному етапі бізнес-процесу. Керівництво та ТОП-менеджмент компаній може відстежувати етапи розвитку проекту із проектування та розробки мобільного додатку, полегшувати комунікацію співробітників, незалежно від того, де вони

перебувають. Зазвичай компанія прагне використати досвід клієнтів, пропонуючи каталоги продажів для замовлення або перегляду продукції.

Мобільні додатки можуть допомагати компаніям покращувати сервісне обслуговування, адже вони полегшують користувачам зв'язок зі службою підтримки за допомогою різних інструментів.

Широке впровадження мобільних пристроїв серед споживачів призвело до нових маркетингових стратегій мобільних додатків.

Процес створення мобільного додатку для покупок переважно передбачає виконання наступних етапів [15]:

1. Дослідження ринку для тематичного мобільного додатку.
2. Вибір платформи та стеку технологій.
3. Розподіл ролей в ІТ-команді, яка буде працювати над проектом.
4. Визначення функції для мінімально життєздатного продукту.
5. Визначення додаткових функцій, щоб які будуть актуальні в майбутньому.
6. Створення макету мобільного додатку.
7. Реалізація процесу розробки мобільного додатку для покупок.
8. Тестування додатку для покупок.
9. Запуск додаток для покупок в комерційне використання.

Додатки для покупок мають ряд переваг в порівнянні з комерційними веб-сайтами і можуть застосовуватися для мобільних пристроїв, націлюючись на отримання прибутку, конверсій та задоволеності клієнтів.

Вони пропонують набагато більше функцій, ніж традиційний веб-сайт. Можна виділити відмінностей, яким чином мобільний додаток може змінити бізнес-процеси компанії з продажу:

- ✓ В мобільних додатках покупки для клієнтів є швидкі та зручні;
- ✓ Більш чуйний і зручний для навігації, ніж веб-сайт;
- ✓ Наявність знижок та ексклюзивних пропозицій лише для мобільних додатків;
- ✓ Інтеграція із соціальними медіа;

- ✓ Ефективний маркетинговий канал;
- ✓ Краще обслуговування клієнтів;
- ✓ Наявність відгуків клієнтів;
- ✓ Залученість нових клієнтів.

Поява мобільних додатків для покупок фактично полегшили зв'язок покупців і продавців практично в усіх галузях. Завдяки насиченому способу життя, смартфонам і зручності додатків споживачі все частіше обирають електронну комерцію, а не покупки в магазині, особливо коли вони в дорозі.

1.2. Опис предметної області

Статистика мобільних додатків за 2023 рік свідчить, що в Apple Store доступно 1,96 мільйона додатків і 2,87 мільйона в Google Play Store [6]. Від шкіль до транснаціональних компаній, усі схиляються до створення власних мобільних додатків, щоб бути більш зв'язаними в усьому світі та полегшити людям доступ до всього, що вони хочуть, швидким натисканням на екран свого смартфона.

Загалом можна виділити наступні переваги використання мобільних додатків для бізнесу [6]:

1. Підвищує пізнаваність бренду. Наявність фірмового мобільного додатку дає змогу здійснювати маркетингову політику та поширювати інформацію про свій бізнес одним клацанням миші. Це дає змогу користувачам більше інформації про компанію, товари та послуги, які компанія з продажів пропонує на ринку. До того ж мобільні додатки привносять можливість зміцнювати емоційні зв'язки зі своїми клієнтами. Мобільний додаток поширює потенційним клієнтам інформацію про компанію та дозволяє їм легко вести з нею бізнес лише кількома дотиками або кліками.

2. Дозволяє персоналізувати. За допомогою мобільного додатку компанія може персоналізувати свої послуги відповідно до цільового ринку.

Наприклад, завдяки проведенню тестування або опитування користувачів додатку дізнатися, що подобається вашим клієнтам, на основі чого пізніше реалізувати стратегію персоналізації, коригування вмісту і сповіщення у формі, більш релевантній для них. Така практика дозволить докласти менше зусиль до створення персоналізованого досвіду, завдяки якому можна заохотити споживачів частіше використовувати мобільний додаток, здійснювати операції купівлі товару та спонукатиме більше людей використовувати даний мобільний додаток.

Персоналізований підхід через застосування мобільного додатку дасть також необхідну інформацію про те, який механізм взаємодії користувачів з даним програмним забезпеченням, використовуючи яку компанія може покращити свої маркетингові стратегії та дізнатися більше про свою клієнтську базу.

3. Економія часу. Мобільний додаток може просто заощадити час ваших клієнтів, надаючи їм швидкий і легкий доступ до послуг компанії з продажу товарів чи послуг. Що стосується оплати через додаток, за оцінками експертів ІТ-галузі, платежі через мобільні додатки до 52,2% у 2023 році.

4. Кращий рівень залучення. За допомогою ефективного та цільового мобільного додатку представники бізнесу можуть охопити ширшу цільову аудиторію. Мобільний додаток може спростити для клієнтів пошук компаній-виробників і використання товарів або послуг. Фірмам уже немає необхідності дотримуватися обмежень фізичного розташування; завдяки додатку компанії можуть охопити ширшу аудиторію, незалежно від того, де вони фізично знаходяться у світі.

5. Зміцнення бізнесу. Мобільний бізнес-додаток також може допомогти посилити та зміцнити компанію, створивши актуальні та довірливі відносини між бізнесом і клієнтом. Використання мобільного додатку дозволить надавати зручну та корисну інформацію про клієнтів, визначати популярність окремих функцій додатку у клієнтів, планувати та розробляти стратегію розвитку

мобільного додатку як елементу сервісу. Мобільний додаток часто є найкращим способом для отримання інформації та відгуків від клієнтів.

6. Маркетингова допомога. За допомогою додатку представники компаній можуть керувати контактною інформацією, спеціальними повідомленнями та навіть інструментами взаємодії. В мобільному додатку можна використовувати фірмові кольори, логотипи та слогани. Ще одна перевага полягає в тому, що мобільний додаток є економічно ефективним з точки зору маркетингу. Це дає змогу заощаджувати гроші за допомогою свого бізнес-дodatку, оскільки він усуває потребу в матеріальних маркетингових товарах і позитивно впливає на зменшення витрати на маркетингові кампанії в реальному часі.

Мобільні додатки безсумнівно можуть бути корисні різним типам бізнесу та комерційної діяльності, у т.ч. і торгівлі. На рис. 1.2. наведені основні типи популярних мобільних додатків для бізнесу та клієнтів.

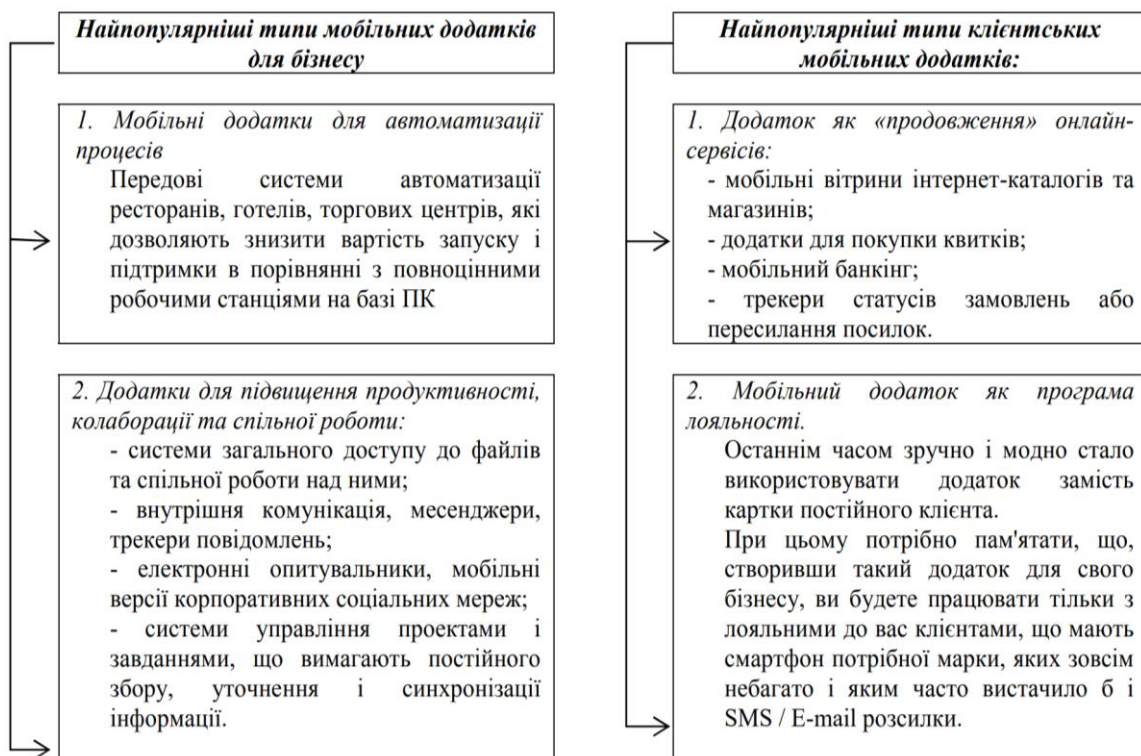


Рисунок 1.2 – Типи мобільних додатків для бізнесу та клієнтів [5]

Виділяють сім переваг використання мобільної реклами для розвитку бізнесу [10]: безпосередність (простота у створенні та користуванні);

доступність (часто безкоштовні); поширення (можливість використання та контакту із людьми з усього світу); ефективність (середній показник ефективності охоплення та поширення інформації 15% проти 2–3% для інших методів поширення інформації); досяжність (досить активна аудиторія активність споживачів реклами без використання телебачення, радіо); вау-фактор (відносна свіжість ринку мобільної реклами); вірусний потенціал (зручність та легкість поширення).

Підприємства можуть застосовувати різні креативні способи привернення уваги існуючих і потенційних клієнтів, шляхом розширення комерційних пропозицій, асортиментної лінійки за допомогою мобільних додатків, спонукаючи користувачів здійснювати покупку товару чи послуги.

73% загальної частки світового ринку електронної комерції припадає на мобільну комерцію. 79% користувачів мобільних пристроїв купували щось онлайн з мобільного пристрою протягом останніх шести місяців [15].

Інформаційні технології постійно розвиваються та оновлюються, зазнають зміни і очікування та вимоги клієнтів, що обумовлено конкуренцією на споживчому ринку та ринку технологій. Як ніколи важливо підтримувати конкурентоспроможність свого мобільного додатку, що дасть змогу задовольняти потреби клієнтів, залучати нових клієнтів, розвивати та масштабувати бізнес.

1.3. Огляд і аналіз існуючих аналогів

Мобільний додаток для електронного бізнесу, комерції та торгівлі, є одним з дієвих інструментів підвищення продажів для підприємств із здійснення торгівельних операцій, що обумовлено тим, що він є завжди доступним, може користуватися попитом серед постійних та потенційних клієнтів для пошуку і покупки потрібних товарів.

Проведене дослідження показало, що на даний час придбання товарів та замовлення послуг через мобільний додаток є швидкою і зручною альтернативою для споживачів, тому більшість підприємств звертають на цю технологію увагу і прагнуть покращити взаємодію з клієнтами, розробляючи та пропонуючи додатки із розширеним переліком функцій, які можуть бути корисними і цікавими для споживачів.

В цілому, використовуючи мобільні додатки для замовлення товарів і послуг в мережі Інтернет, користувачі отримують низку переваг [12]:

- можливість купувати в будь-який час (здійснення покупок в один клік);
- використання ресурсу без прив'язки до ПК;
- інтерактивне джерело отримання інформації про товари і послуги;
- можливість порівняння товарів та цін;
- можливість переглянути відгуки про товар або послугу;
- інструмент для здійснення покупок, який завжди під рукою;
- можливість мати програму лояльності на своєму мобільному пристрої.

Незалежно від того, чи компанія продає квіти в Інтернеті чи пропонує послуги спа та салону, клієнтам потрібен ефективний засіб зв'язку з бізнесом. Мобільний маркетинг має важливу перевагу, оскільки надає компаніям перспективний потенціал для взаємодії з клієнтами в режимі реального часу, використовуючи місцезнаходження клієнта та повну інформацію про профіль споживача. Використовуючи ефективно цю інформацію, компанії зможуть монетизувати усі канали залучення клієнтів, перетворивши їх у джерела доходу за допомогою маркетингу мобільних додатків.

Гарним прикладом залучення клієнтів є iFood Assistant від Kraft, який надає приблизно 2000 рецептів, використовуючи продукти Kraft з різними купонами, пропозиціями та списками покупок, щоб надати переваги покупцям.



Рисунок 1.3 – Вигляд вікна мобільного додатку iFood Assistant

Даний мобільний додаток підтримує можливості обміну в соціальних мережах та дозволяє клієнтам ділитися своїми улюбленими рецептами зі своїми друзями в соціальних мережах.

Серед усіх товарів, які споживачі купують в Інтернеті, очолює список одяг. Програми для покупок одягу полегшили покупцям можливість купувати модний одяг, взуття, сумки та аксесуари, не виходячи з дому.

Перспективним моментом для комерційного застосування мобільного додатку для покупок є використання таких функцій, як відгуки клієнтів, деталізацію інформації про продукт, наявність упорядкованих категорій товарів, посібники та довідники з визначення розмірів, наявність безкоштовної доставки, унікальних пропозицій та знижок. Вважається, що додатки для одягу споживачам часто здаються більш привабливими варіантами, ніж звичайні магазини.

Мобільний додаток ASOS застосовується для здійснення модних покупок, адже в ньому представлено понад 850 брендів і 85 000 товарів у таких категоріях, як одяг, макіяж, взуття й аксесуари [14].

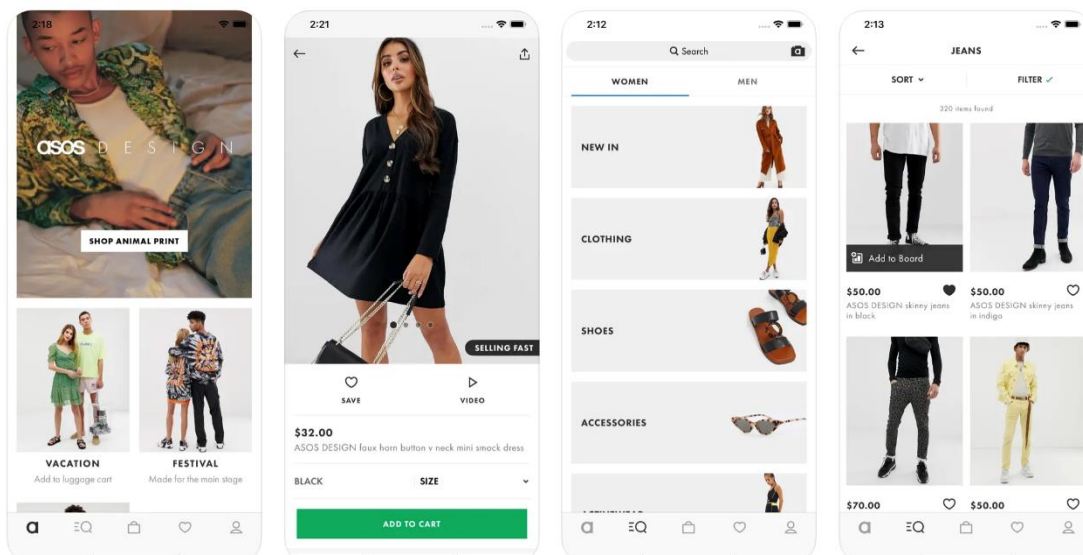


Рисунок 1.4 – Вікна мобільного додатку ASOS

На українському ринку представлений широкий спектр мобільних додатків для придбання товарів і послуг. На рис. 1.5 показано вигляд мобільного додатку бренду Інтертоп, який реалізовує продукцію за категоріями взуття, одяг, аксесуари.

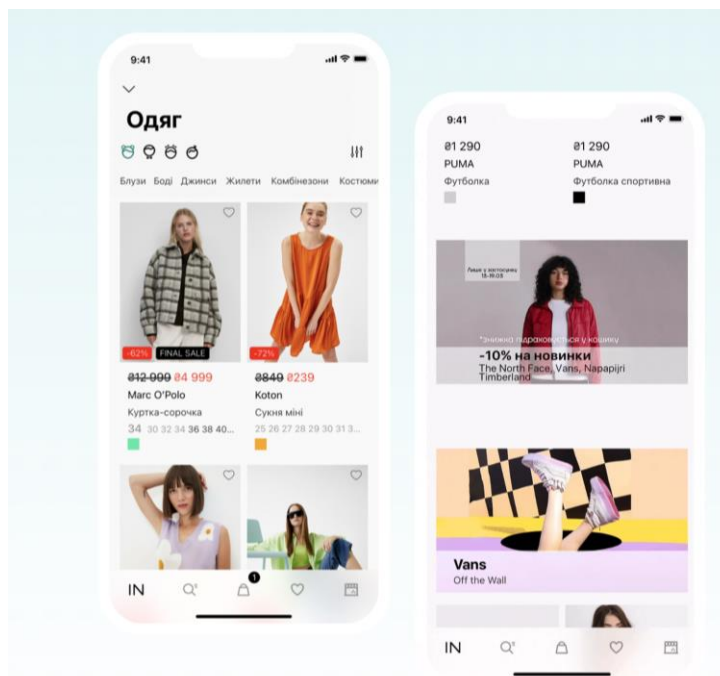


Рисунок 1.5 – Вікна мобільного додатку Інтертоп

Мобільні додатки можуть мати ряд привабливих і цікавих моментів для потенційних користувачів. Наприклад мобільні додатки Cashback Shopping

винагороджують покупців за покупку товарів і послуг. Вони зазвичай мають партнерські стосунки з брендами та платять клієнтам готівкою або балами лояльності за покупку через партнерське посилання.

Більшість додатків для повернення грошей можна завантажити та використовувати безкоштовно, і вони можуть допомогти споживачам заощадити купу грошей на щоденних покупках у довгостроковій перспективі.

Мобільні додатки для покупок продуктів допомагають споживачам купувати продукти онлайн за допомогою наступних опцій: доставки додому, самовивозу з магазину або обох способів. Ці додатки привабливі для споживачів в багатьох країнах, оскільки мають такі функції, як віртуальні списки покупок, можливість переглядати інформацію про харчування та дозволяють обирати бренди і продукти.

Продуктові мобільні додатки дозволяють людям, що працюють, економити час, оскільки вони позбавляють покупців необхідності їздити в великі торговельні мережі або ринки, заходити у звичайні магазини. Виходячи з персоналізованого підходу, в продуктових мобільних додатках можна отримати ексклюзивні знижки (досить часто такі знижки діють лише на покупки онлайн), набори та пропозиції, щоб допомогти покупцям заощадити гроші на предметах, які вони регулярно купують.

Amazon має свій мобільний додаток для покупок, доступний як для систем Android, так і для iOS, пропонуючи переваги лише для додатків. Основна мета даного мобільного додатку полягала в тому, щоб зробити покупки з мобільного пристрою швидшими та простішими, ніж покупки на комп'ютері.

В таблиці 1.3. наведено порівняльний аналіз основних функцій найбільш популярних мобільних додатків для покупок.

Таблиця 1.3 – Порівняльна характеристика мобільних додатків для покупок

Amazon	AliExpress	eBay
Порівняння товарів	Пошук зображень	Ексклюзивні
Перегляд і сканування товару	Інтерактивні та веселі ігри (ви можете отримати винагороду за гру в них)	розпродажі додатків Безпечний і швидкий розрахунок
Пошук товару (просто сфотографуйте або відскануйте штрих-код)	Персоналізовані рекомендації	Відстеження замовлень у реальному часі
Сповіщення про відстеження та доставку в реальному часі	Списки з елементами найвищого рангу	Продаж простих речей
Повний огляд продукту 3600	Захист покупця	Пошук зображень
Push-повідомлення, коли товар у продажу	Управління замовленнями та автоматичне сповіщення про відвантаження	Сканер штрих-кодів
Цілодобова підтримка в чаті	Цілодобовий довідковий центр	Засіб для видалення фонового зображення

Одним з найпопулярніших мобільних додатків у категорії «Доставка» в Україні є мобільний додаток Glovo (рис. 1.6).

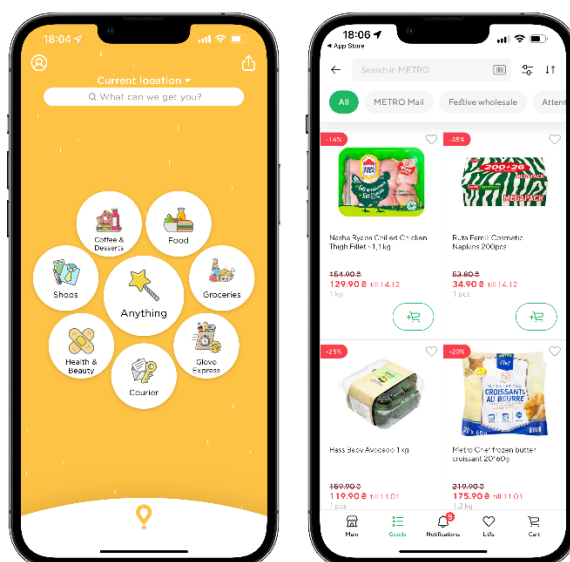


Рисунок 1.6 – Вигляд мобільного додатку Glovo

У Glovo можна замовити все, що завгодно, – доставка виконується досить швидко, фактично за лічені хвилини. Замовлення їжі та інших товарів можна оформити у будь-якому закладі великих міст України: в магазині, супермаркеті, аптеці, ресторані, кафе.

РОЗДІЛ 2.

ВИБІР ЗАСОБІВ РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ

2.1. Особливості вибору засобів розробки мобільного додатка

При виборі засобів розробки мобільного додатка важливо враховувати різні фактори, такі як функціональність, продуктивність, доступність ресурсів, розширюваність і зручність розробки [22]. Проаналізуємо деякі популярні засоби, які можна використовувати для розробки мобільних додатків.



Рисунок 2.1 – Найбільш розповсюджені засоби для проектування мобільних додатків

Android Studio – це офіційне інтегроване середовище розробки (IDE) для платформи Android. Воно надає широкі можливості для розробки, тестування та накладання мобільних додатків на базі ОС Android. Android Studio має потужні інструменти, такі як редактор коду, емулятори пристроїв, системні збірки та дебагер.

React Native – це відкрита платформа для розробки мобільних додатків, що дозволяє використовувати JavaScript для створення крос-платформних додатків для Android та iOS. React Native надає можливість перевикористовувати код і компоненти, що спрощує розробку для обох платформ.

Flutter – це відкритий фреймворк, розроблений Google, для створення крос-платформних мобільних додатків. Він використовує мову програмування Dart і дає можливість швидко розробити високоякісні інтерфейси користувача з відмінною продуктивністю.

Xamarin – це фреймворк, що базується на мові програмування C#, для розробки крос-платформних мобільних додатків. Xamarin дозволяє перевикористовувати код між платформами, що зменшує зусилля при розробці для Android та iOS.

Ionic – це веб-фреймворк для розробки крос-платформних мобільних додатків з використанням веб-технологій, таких як HTML, CSS і JavaScript.

NativeScript – це фреймворк, який дозволяє розробляти крос-платформні мобільні додатки, використовуючи нативні компоненти для створення інтерфейсу користувача. Він підтримує JavaScript, TypeScript та Angular, що надає розробникам гнучкість при створенні додатків для Android.

PhoneGap/Cordova – це фреймворк, який дозволяє розробляти крос-платформні мобільні додатки, використовуючи веб-технології, такі як HTML, CSS і JavaScript. PhoneGap/Cordova дозволяє отримати доступ до функцій пристрою за допомогою набору плагінів і підтримує розробку для різних платформ, включаючи Android.

Вибір засобів розробки мобільного додатку залежить від ваших потреб,

навичок, вимог проекту та особистих вподобань. Рекомендується провести детальний аналіз кожного іншого, враховуючи його функціональність, підтримку спільного використання коду, доступність документації та спільних розробників. Також важливо врахувати масштабність проекту та часові рамки, щоб обрати оптимальний засіб розробки мобільного додатка.

2.2. Мова програмування Java

Java - це об'єктно-орієнтована мова програмування, яка була розроблена компанією Sun Microsystems (зараз володіє Oracle Corporation). Вона відома своєю платформонезалежністю, високою надійністю, безпекою та простотою використання [15]. Особливістю цієї мови є платформонезалежність. Java використовує віртуальну машину Java (JVM), яка перетворює код Java в байт-код, що може бути виконаний на будь-якій підтримуваний платформі, що має встановлену JVM. Це дозволяє розробляти програми на Java для різних операційних систем, таких як Windows, macOS і Linux, без необхідності змінювати вихідний код.

Також Java повністю підтримує об'єктно-орієнтовану парадигму програмування. Всі елементи в мові, включаючи змінні, функції та класи, є об'єктами. Це дозволяє розробникам створювати модульний і легко зрозумілий код.

Дана мова програмування має вбудовану систему управління пам'яттю, яка автоматично відстежує виділення та звільнення пам'яті. Це допомагає уникнути проблем, пов'язаних з витоками пам'яті або неправильним використанням пам'яті, що можуть призвести до сбоїв програми.

На безпековому рівні Java має вбудовану систему безпеки, яка дозволяє запускати програми в обмеженому середовищі, відомому як "пісочниця" (sandbox). Це забезпечує відокремлення виконуваного коду від операційної системи, що запобігає небажаному впливу програм на систему.

Java підтримує багатопоточність, що дозволяє виконувати одночасно кілька ниток (потоків) в межах одного програмного процесу. Це корисно для розробки швидких і ефективних програм, які використовують паралельну обробку даних.

Велика стандартна бібліотека: Java поставляється з великою стандартною бібліотекою класів, яка містить багато готових рішень для різних завдань програмування. Це включає роботу з мережами, роботу з базами даних, графічний інтерфейс користувача та багато іншого. Велика кількість сторонніх бібліотек також доступна для розширення можливостей мови.

Вона має простий синтаксис, подібний до інших мов програмування C-подібного сімейства, таких як C++ і C#. Це дозволяє розробникам швидко оволодіти мовою і писати чистий і зрозумілий код. А також підтримує можливість розширення мови шляхом створення власних класів і бібліотек. Це дозволяє розробникам створювати власні рішення та використовувати їх у власних програмах.

Тут нами описані лише кілька особливостей мови Java, які роблять її популярною серед розробників. Java широко використовується для розробки різноманітних програм, веб-додатків, мобільних додатків і серверних застосунків.

2.3. Характеристика існуючих інструментів розробника на Android

Оскільки, розробка Android-додатків здійснюється на мові Java. Тому, в першу чергу для початку розробки необхідно встановити Java Development Kit (JDK).

Java Development Kit (JDK) – це набір програмних інструментів, який необхідний для розробки, компіляції та виконання програм на мові Java. JDK включає в себе компілятор Java (javac), виконавче середовище Java (JRE), бібліотеки класів Java, документацію і додаткові утиліти.

До основних компонентів JDK можна віднести:

Java Compiler (javac): Це інструмент, який використовується для компіляції (перетворення у виконуваний код) файлів з розширенням .java в байт-код Java з розширенням .class. Компілятор перевіряє синтаксичну і семантичну правильність програми та генерує виконуваний код.

Java Virtual Machine (JVM): Це виконавче середовище, яке відповідає за виконання байт-коду Java. JVM інтерпретує байт-код та забезпечує його виконання на різних операційних системах та апаратних платформах.

Java Class Library: JDK включає багату бібліотеку класів, яка містить готові компоненти і функціональні можливості для розробки Java-програм. Ці класи надають різноманітні функції, такі як робота з мережею, робота з файлами, обробка рядків, робота з графікою та багато іншого.

Документація: JDK постачається з документацією, яка містить опис класів, інтерфейсів, методів та інших складових Java API. Ця документація надає розширені пояснення та приклади використання, що допомагають розробникам у розумінні та використанні різних функцій.

Додаткові утиліти: JDK також містить додаткові інструменти, такі як отладчик (jdb), профілер (jvisualvm), генератор документації (javadoc) і багато інших, які полегшують розробку, тестування і налагодження Java-програм.

Android Software Development Kit (SDK) містить багато інструментів і утиліт для створення і тестування додатків. Наприклад, за допомогою SDK Manager можна встановити Android API будь-якої версії (рис. 2.2), а також перевірити репозиторій на наявність доступних, але ще не встановлених пакетів і архівів.

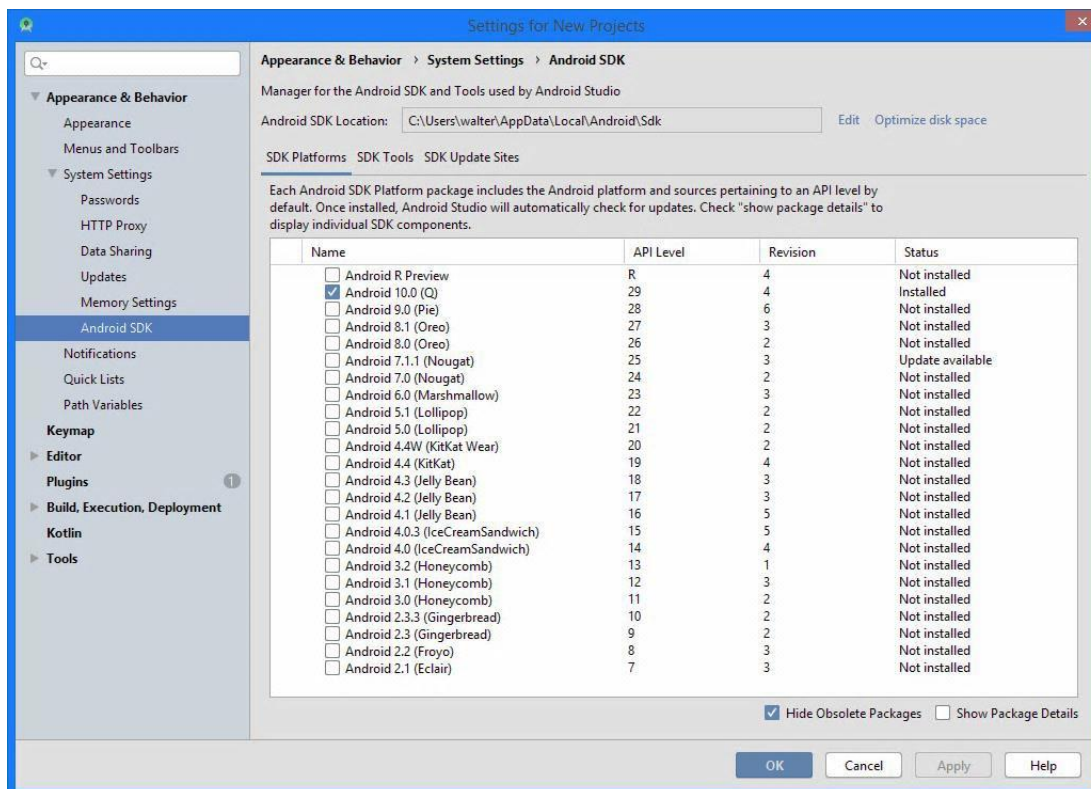


Рисунок 2.2 – Вікно Android SDK

Основні компоненти Android SDK:

1. Android SDK Tools – це набір утиліт командного рядка, які включають такі інструменти, як архіватор (adb), компілятор ресурсів (aapt), генератор налагоджувальних файлів (aidl) та інші. Ці інструменти дозволяють розробникам взаємодіяти з пристроями, компілювати ресурси, генерувати код та виконувати інші операції, пов'язані з розробкою Android-додатків.

2. Android Platform Tools – це набір утиліт командного рядка, які допомагають у взаємодії з Android-пристроями, такими як смартфони та планшети. До цих утиліт входять adb (Android Debug Bridge) для підключення до пристрою, fastboot для режиму завантаження та інші допоміжні інструменти.

3. Android SDK Platform – це колекція пакетів, які містять Android API та інші компоненти, необхідні для розробки Android-додатків. Кожен пакет представляє собою версію Android, на якій буде працювати додаток. Розробники можуть встановлювати різні версії платформ для тестування та сумісності своїх додатків.

4. Android Virtual Device (AVD) Manager – це інструмент, який дозволяє розробникам створювати та керувати віртуальними пристроями Android для емуляції різних конфігурацій та версій Android. AVD дозволяє тестувати додатки на різних пристроях без необхідності володіння фізичними пристроями.

5. Android Debug Bridge (ADB) – це інструмент, який дозволяє розробникам взаємодіяти з підключеними Android-пристроями для налагодження, копіювання файлів, встановлення додатків та інші дії.

6. SDK також включає додаткові компоненти, такі як бібліотеки підтримки, засоби розробки UI, інструменти для профілювання та тестування, документацію та приклади коду.

7. Android SDK надає розробникам повний спектр інструментів, необхідних для розробки високоякісних та ефективних Android-додатків. Він сприяє швидкому розвитку, тестуванню та розгортанню додатків на платформі Android.

Найпопулярніші середовища розробки під Android наступні.

Eclipse – це безкоштовне модульне інтегроване середовище розробки програмного забезпечення. Розроблено та підтримується Eclipse Foundation.

IntelliJ IDEA – комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP тощо), запущене JetBrains.

Android Studio – це середовище розробки Android на основі IntelliJ IDEA. Надає інтегровані засоби розробки та налагодження. На додаток до всього, чого ви очікуєте від IntelliJ, Android Studio реалізує:

- підтримку процесів складання програм на основі Gradle;
- рефакторинг для Android і швидке виправлення помилок;
- інструменти для визначення продуктивності, зручності використання, сумісності версій та інших проблем;
- функції Proguard (утиліти для скорочення, оптимізації та обфускації коду) і підписи програм;
- майстри для створення загальних дизайнів і компонентів Android на

основі шаблонів;

➤ редактор Wysiwyg працює на різних розмірах і розширених екранах, а вікно попереднього перегляду може відображати запущену програму в режимі реального часу на кількох пристроях одночасно;

➤ вбудована підтримка google cloud platform.

Intel XDK - середовище для розробки кросплатформних мобільних додатків; містить інструменти для створення, налагодження та побудови програмного забезпечення, а також емулятори пристроїв; підтримує розробку для Android, Apple iOS, Microsoft Windows 10, Tizen; та такі мови розробки, як HTML5 і JavaScript .

Intel Beacon Mountain – ще одне середовище розробки від Intel. Надає всі інструменти, необхідні для проектування, розробки, налагодження та оптимізації програм Android. Підтримує розробку цільових платформ на базі процесорів Intel Atom і ARM. Beacon Mountain створено на основі IDE Android (Eclipse, Android ADT, Android SDK).

Marmalade SDK – це кросплатформний SDK від Ideaworks3D Limited. Являє собою набір бібліотек, зразків коду, інструментів і документації, необхідних для розробки, тестування та розгортання програм для мобільних пристроїв. В основному використовується для розробки ігор.

2.4. Компоненти Android додатку

Програми написані для Android використовують у своїй роботі вікна, схожі на вікна Windows, але в цій системі вікна мають іншу назву - Activity. Як і операційна система Windows, кожне вікно має свої особливості та життєвий цикл [15]. Як правило, діяльність пов'язана з окремими екранами або вікнами програми, а також переходи між вікнами відбуватимуться як переходи від однієї діяльності до іншої. Програма може мати одну або кілька дій. Усі програми для Android мають суворий життєвий цикл системного типу. Коли

користувач запускає програму, система надає їй високий пріоритет. Кожна окрема програма виконується як окремий процес, що дозволяє системі надавати певним програмам вищий пріоритет, ніж іншим. Так, наприклад, ви не можете блокувати вхідні дзвінки під час використання програми Android.

Після завершення роботи програми система звільняє всі пов'язані з програмою ресурси, надає їй низькопріоритетний стан і при необхідності вимикає її. Усі об'єкти Activity у програмі керуються системою в стеку, який називається заднім стеком. Коли починається нова дія, вона розміщується на вершині стека та відображається на екрані пристрою, доки не з'явиться нова дія. Коли користувач завершує роботу (наприклад, закриває програму), вона видаляється зі стеку, а друга дія в стеку відновлює свою роботу.

2.5. Життєвий цикл Activity

Загальну схему роботи Activity представлено на рисунку 2.2.

Після запуску activity проходить через декілька особливих подій, які повинна обробити система. Для контролю цього процесу обробки існують спеціальні методи.

Oncreate() – це найперший метод з якого і починається життєвий цикл кожної activity. Під час виконання цього методу activity переходить у стан Created. Цей метод повинен обов'язково бути переназначений у класі, який відповідає за activity. За допомогою цього методу також роблять перше налаштування додатку, а саме створюють об'єкти візуального інтерфейсу, та пов'язують їх унікальними ідентифікаторами.

Метод приймає, у якості параметра, об'єкт типу Bundle, який здійснює збереження у собі попереднього стану activity, якщо цей стан було збережено. Якщо activity буде створена заново, то цей об'єкт повинен набути значення null. Якщо ж activity вже була створена та знаходиться в призупиненому стані, то об'єкт bundle зберігатиме у собі інформацію про стан activity.

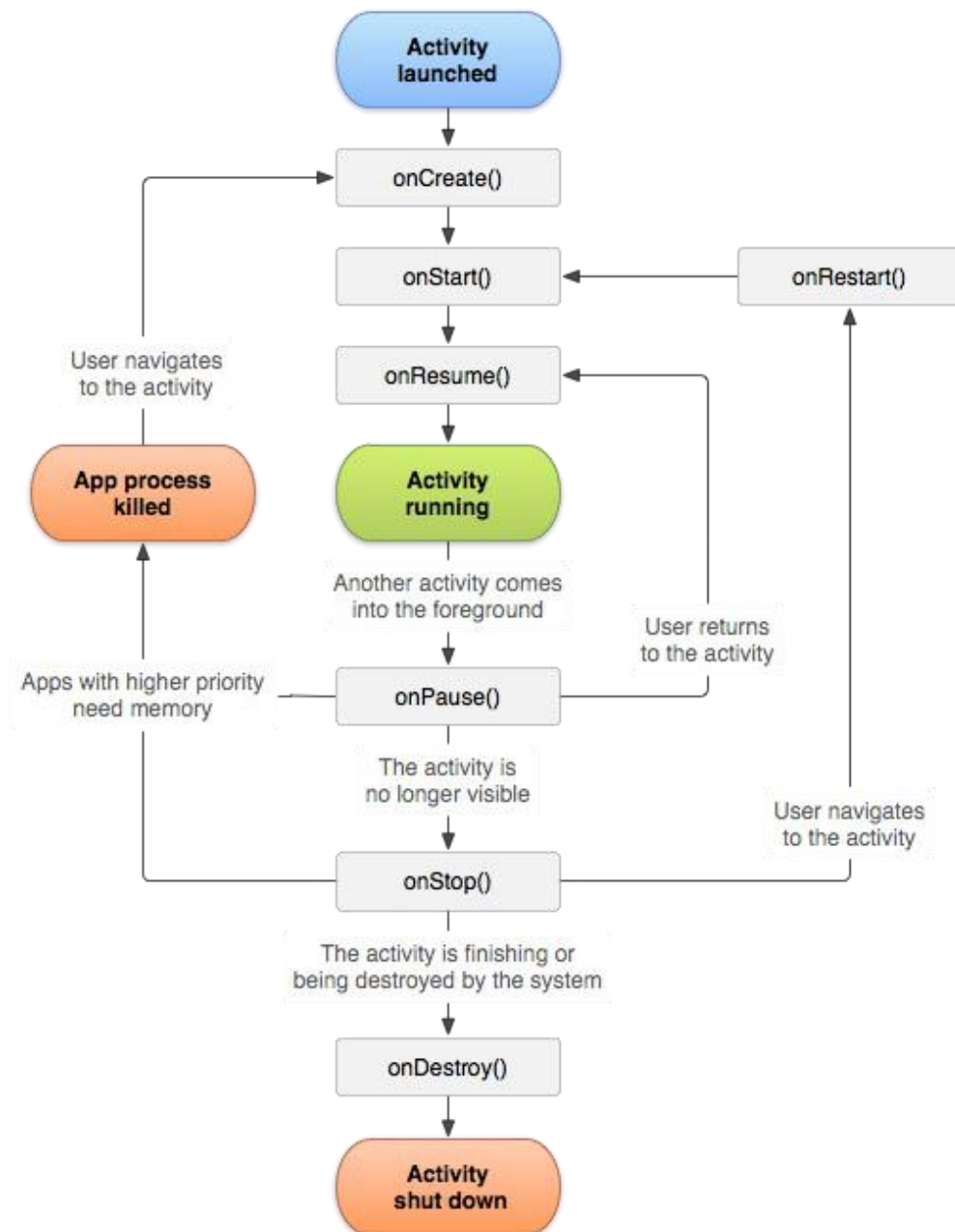


Рисунок 2.3 – Життєвий цикл Activity

Після закінчення роботи методу `onCreate()`, activity переходить у наступний свій стан, а саме у стан `Started` та викликає метод `onStart()`. В цьому методі проводиться підготовка до виводу activity на екран гаджету. Як правило, даний метод не має потреби у перевизначенні, тому що майже всю роботу робить влаштований код. Після завершення роботи цього методу activity може бути відображена на екрані користувача, а після цих дій проводиться виклик методу `onResume()`, а сама activity переходить до стану `Resumed`.

Після того як цей метод почав свою роботу `activity` з'являється на екрані, а користувач вже може починати з нею взаємодіяти. У цьому стані `activity` залишається доти поки не втратить свій фокус.

Коли користувач прийме рішення перейти до іншої `activity` то система автоматично викличе метод `onPause()`, а сама активність переходить до стану `Paused`. У цьому стані можна призупиняти процеси, наприклад такі як анімацій, роботу камери, аудіо та відео. А також є можливість звільнити ресурси які використовує додаток, для підвищення продуктивності системи. Проте треба мати на увазі, що у цьому стані `activity` все ще видно на екрані, а тому на обробку цього методу залишається дуже мало часу. У цьому методі не потрібно здійснювати зберігання будь-яких яких даних, особливо якщо вони пов'язані з роботою мережі, або з базою даних. Такі дії найкраще виконувати за допомогою методу `onStop()`.

Після виконання методу `onStop()`, `activity` стає невидимою, вона не показується на екрані, проте вона все ще активна. Якщо користувач все ж таки прийме рішення повернутися до цієї `activity` то система Android викличе метод `onResume()` і `activity` знову з'явиться на екрані.

Проте система іноді може і сама завершити роботу `activity`, якщо вона побачить що для роботи активних додатків треба виділити більше пам'яті.

Під час стану `Stopped` `activity` залишається у пам'яті гаджету, вона зберігає стан усіх своїх елементів інтерфейсу.

Якщо після виклику методу `onStop()` користувач знову вирішить повернутися до цієї `activity` то система викличе метод `onRestart()`. Якщо `activity` завершить свою роботу, то виконуватиметься метод `onDestroy()`.

Цей метод здійснює завершення життєвого циклу `activity`. Він викликається також якщо система вирішила завершити роботу `activity` через причини конфігурації гаджету.

Також слід пам'ятати, що при зміні орієнтацій екрану, система буде здійснювати завершення `activity`, а потім створювати її заново викликаючи перший метод – `onCreate()`.

2.6. Опис та особливості роботи з базою даних SQLite

В середовищі Android Studio для розробки мобільних додатків для платформи Android за замовчуванням використовується SQLite як система управління базами даних (СУБД). SQLite є компактною і вбудованою СУБД, яка надає локальне зберігання та обробку даних на пристрої. Вона підтримується безпосередньо в Android-платформі і вбудована в бібліотеки розробки Android SDK.

SQLite - це вбудована реляційна база даних, яка надає легку у використанні і просту встановлення систему управління базами даних. Особливістю SQLite є те, що вона не працює як окремий сервер бази даних, а зберігає дані безпосередньо у файлової системі на пристрої, на якому вона використовується. Це робить SQLite відмінним вибором для мобільних додатків, вбудованих систем та невеликих проєктів, де не потрібен розподілений сервер бази даних.

Особливості роботи з SQLite:

- Легка у використанні: SQLite проста у встановленні та налаштуванні. Для роботи з нею не потрібно окремої настройки сервера або створення користувачів та доступів.
- Вбудована: SQLite можна використовувати безпосередньо у вбудованих додатках, що дозволяє знизити залежність від зовнішніх серверів баз даних.
- Не потребує окремого сервера: SQLite не вимагає окремого сервера баз даних для роботи. Запити до бази даних обробляються прямо на пристрої, де вона встановлена.
- Малий розмір: Бібліотека SQLite має невеликий розмір, що робить її популярним вибором для мобільних додатків, де розмір файлу має значення.
- Безсерверна: SQLite не має окремого процесу сервера. Всі дані зберігаються в одному файлі, який можна переносити і копіювати на різні пристрої.

- Підтримка стандартних SQL-запитів: SQLite підтримує багато стандартних SQL-запитів і операцій, таких як SELECT, INSERT, UPDATE і DELETE, що робить його зручним для роботи з базами даних.
- Транзакційна безпека: SQLite підтримує транзакції з ACID-властивостями (атомарність, консистентність, ізолюваність, довіреність), що дозволяє забезпечити цілісність даних та безпеку операцій.
- Підтримка багатьох мов програмування: SQLite можна використовувати з різними мовами програмування, такими як C/C++, Python, Java, PHP і багатьма іншими. Існує багато бібліотек інтерфейсів, що спрощують роботу з SQLite у різних середовищах програмування.

Загалом, SQLite - це проста, легка та ефективна система управління базами даних, яка чудово підходить для малих та середніх проектів, де потрібне вбудоване рішення зі зручним SQL-інтерфейсом.

2.7. Передача даних в базі даних за допомогою формату JSON

JSON (Java Script Object Notation) – це найпоширеніший формат обміну даним. Він дуже зручний для написання та читання як людиною так і комп'ютером. JSON був оснований на підмножині такої мови програмування, як JavaScript визначеного у стандарті ECMA-262 3rd Edition. У 1999 році JSON був остаточно визнаний незалежним мів мовної реалізації. Він використовує угоди, які нагадують C-подібні мови програмування, такі як C, C++, Perl та Java. Саме ці властивості роблять JSON ідеальною мовою для обміну даними.

Незважаючи на походження від мови JavaScript, формат JSON вважається незалежним та може використовуватися майже з усіма мовами програмування. Для багатьох мов існує готовий код, для створення даних у форматі JSON та їх подальшої обробки.

Завдяки своїй лаконічності, у порівнянні із мовою XML, формат JSON може бути більш вигідним для серіалізації складних структур. Йому також

знайшлося використання у веб-додатках, його використовують для обміну даними між сервером та браузером, а також проста між серверами.

JSON використовується для представлення даних у вигляді пар "ключ-значення". Дані можуть бути представлені у вигляді рядків, чисел, логічних значень, масивів, об'єктів та нульових значень. Основна структура JSON полягає в тому, що дані зберігаються у вигляді набору пар "ключ-значення", розділених комами і заключених у фігурні дужки {}.

Передача даних за допомогою формату JSON відбувається наступним чином:

Сервер або джерело даних створює об'єкт або структуру даних в своїй програмі. Ці дані конвертуються в рядок JSON за допомогою методу серіалізації. Більшість сучасних мов програмування мають вбудовані функції або бібліотеки для серіалізації об'єктів в формат JSON.

Рядок JSON передається через мережу до отримувача. Це може відбуватися через HTTP, WebSocket або інші протоколи передачі даних. Отримувач отримує рядок JSON і використовує метод десеріалізації, щоб перетворити його назад у структуру даних, яку можна легко обробляти в його програмі. Отримані дані можуть бути використані для подальшого аналізу, відображення на веб-сторінці або взаємодії з іншими системами.

JSON є популярним форматом для передачі даних через мережу, оскільки його легко читати і записувати як людям, так і комп'ютерам. Він широко використовується веб-серверами, веб-додатками та мобільними додатками для обміну даними з клієнтами.

2.8. Особливості використання Tesseract для оптичного розпізнавання символів

Tesseract - це відкрита бібліотека для оптичного розпізнавання символів (OCR). Вона розроблена на базі Google, і починаючи з версії 2.0, її підтримкою займається команда Tesseract OCR на GitHub.

На базі Tesseract реалізовано підтримку багатомовного OCR, що дозволяє розпізнавати текст у різних мовах, включаючи латиницю, кирилицю, азійські та багато інших мовних складностей. Вона також підтримує розпізнавання тексту змішаних мов.

Бібліотека відома своєю високою точністю розпізнавання. Завдяки постійним поліпшенням та оптимізаціям, Tesseract може добре справлятися з різноманітними типами тексту, включаючи структурований текст, надрукований текст, текст на фотографіях тощо.

Tesseract надає розширені можливості конфігурації, що дозволяє визначати параметри розпізнавання, такі як мову, розпізнавальні правила, алгоритми обробки зображень та інші параметри. Це дозволяє налаштувати Tesseract для оптимальної роботи з конкретними типами тексту та задачами OCR. А також він може працювати з різними форматами зображень, включаючи TIFF, JPEG, PNG, BMP та інші, здатен розпізнавати текст, який подано у вигляді зображення або у форматі PDF.

Загалом Tesseract є стабільною та доведеною бібліотекою OCR, яка знайшла широке застосування у багатьох проектах та додатках. Вона активно підтримується та оновлюється командою розробників, її можна використовувати як самостійну бібліотеку для розпізнавання тексту власноруч розробленими програмами або використовувати готові інтерфейси та зв'язки (bindings), доступні для різних мов програмування, таких як Python, Java, C++, .NET та інші.

РОЗДІЛ 3.

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ «ДОСТУПНИЙ МЕНЕДЖЕР ПОКУПОК»

3.1. Основні етапи створення додатку «Доступний менеджер покупок»

Створення мобільного додатка вимагає виконання кількох основних етапів, які забезпечують успішну розробку, тестування та впровадження програмного продукту. У цьому розділі розглянемо основні етапи створення мобільного додатку «Доступний менеджер покупок».

Насамперед виконується визначення вимог. При цьому першим етапом виникає потреба у зборі та аналізі вимог до мобільного додатку. Завданням розробників є встановлені функціональні та нефункціональні вимоги, визначення цільової та цільової аудиторії використання мобільного додатку. Для мобільного додатку «Доступний менеджер покупок» можна було б включати створення інтуїтивно зрозумілого інтерфейсу, забезпечення доступності для користувачів із безкоштовними потребами та надання функцій для управління покупками та списками товарів.

Після цього виконується проектування інтерфейсу. Цей другий етап розробки проекту та виконання проектування інтерфейсу мобільного додатка. Важливо врахувати потреби цільової аудиторії, забезпечити зручність використання та доступність функціональності. Дизайнери створюють розміщення макетів екранів, вирішують питання щодо елементів управління, вибору кольорової гами та шрифтів. Для мобільного додатку «Доступний менеджер покупок» необхідно врахувати принципи універсального доступу, які дозволяють користувачам з особливостями використовувати додаток без перешкод.

Одним із важливих етапів є розробка функціоналу. На цьому етапі команда розробників створює функціонал мобільного додатку «Доступний менеджер покупок». Цей етап включає програмування логіки додатку, розробку

бази даних, інтеграцію із зовнішніми сервісами та реалізацію всіх деяких функцій.

Наприклад, для «Доступного менеджера покупок» можуть бути розроблені такі функції:

- створення списків покупок – користувач може створювати нові списки покупок, додавати товари та встановлювати їх кількість;
- додавання товарів – користувач може швидко додавати товари до списку покупок, використовуючи функцію пошуку або сканування штрих-кодів товарів;
- управління списками – користувач може редагувати, переміщувати, копіювати або видаляти списки покупок;
- оповіщення та нагадування – додаток може надсилати користувачеві спільність про наближення до терміну покупки, нагадувати про необхідність купити певні товари;
- категорії товарів – додаток може пропонувати користувачам розділити товари за категоріями, наприклад, продукти харчування, косметика, побутова хімія тощо, для більш зручного управління списком покупок;
- пошук та фільтрація товарів – користувач може шукати конкретні товари за назвою або замінити фільтри для швидкого продажу товарів за певними характеристиками, такими як ціна, бренд, вага тощо.
- історія покупок – додаток збереження може зберегти історію покупок користувача, що дозволяє переглядати попередні списки покупок, повторювати покупки або створювати новий список на основі попередніх покупок;
- ділитися списками покупок – користувач може поділитися своїми списками покупок з іншими користувачами, надіславши їм запрошення до спільного використання або надіславши список покупок через повідомлення або електронну пошту;
- тестування та відладка – на цьому етапі цього тесту виявлення роботи всіх функцій, перевірку наявності помилок та виявлення ваших проблем. Тестування може бути проведене як вручну, так і автоматизовано, зокрема, для

перевірки функціональності, сумісності з іншими пристроями та операційними системами;

➤ поліпшення та оптимізація – після завершення тестування команда розробників аналізує результати та вносить покращення до додатку. Це може включати виправлення помилок, оптимізацію продуктивності, покращення інтерфейсу та забезпечення відповідності всім вимогам та стандартам;

➤ впровадження та реліз – якщо мобільний додаток готовий до використання, він підлягає впровадженню. Це включає підготовку додатків для публікації в магазинах додатків (наприклад, Google Play або App Store).

3.2. Розробка інтерфейсу користувача

Графічний інтерфейс користувача мобільного додатку «Доступний менеджер покупок» має відповідати наступним вимогам:

- ✓ зрозумілість;
- ✓ відповідність вимогам;
- ✓ виразність інформації;
- ✓ послідовність подання інформації.

Для забезпечення цих вимог був розроблений інтерфейс користувача мобільного додатку «Доступний менеджер покупок» (рис. 3.1).

Запропонований інтерфейс користувача включає в себе декілька елементів. Зокрема, до нього належить головне меню, сторінка вибору, налаштування, контакти та довідник з користування системою. Коли користувач відриває додаток перед ним з'являється головне меню в якому він може проглядати список покупок.

Всі покупки розташовані у окремому порядку. Після того як користувач обирає покупку, він переходить на вікно покупки, в якому відображається вся інформація про товар, а також його характеристики.

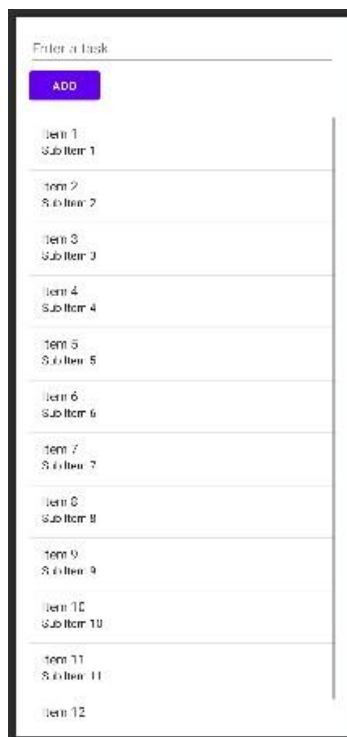


Рисунок 3.1 – Інтерфейс користувача мобільного додатку «Доступний менеджер покупок»

Допоміжне вікно налаштувань дозволяє користувачу персоналізувати додаток для зручності використання. Вікно контактів надає користувачу інформацію про способи зв'язку з розробниками. Вікно довідника створено для надання інформації та інструкції з використання системи.

3.3. Створення коду розмітки для графічного інтерфейсу користувача (GUI) на платформі Android

Запропонований нами код є розміткою для графічного інтерфейсу користувача (GUI) на платформі Android (рис. 3.2). Нижче подано аналіз та пояснення основних елементів коду.

Заголовок XML-документа прописується наступним чином, як представлено на рис. 3.2.

```
1 <?xml version="1.0" encoding="utf-8"?>
```

Рисунок 3.2 – Фрагмент коду заголовка XML-документа

Цей рядок вказує, що файл є XML-документом версії 1.0 з кодуванням UTF-8.

Кореневий елемент `LinearLayout` прописується наступним чином, як представлено на рис. 3.3.

```
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:orientation="vertical"
7   android:padding="16dp"
8   tools:context=".MainActivity">
```

Рисунок 3.3 – Фрагмент коду кореневого елемента `LinearLayout`

Кореневий елемент `LinearLayout` створює контейнерний об'єкт, який використовує інші елементи у вертикальному спрямуванні. В атрибутах цього елемента задаються розміри та вигляд GUI-компонентів.

Атрибут `xmlns:android` – вказує на простір імен Android і дозволяє використовувати вбудовані атрибути та елементи Android.

Атрибут `xmlns:tools` – вказує на простір імені засобу розробника, який дозволяє використовувати деякі атрибути, що підтримуються інтерфейсом розробника Android Studio.

Атрибут `android:layout_width="match_parent"` встановлює ширину елемента такою ж, як ширина батьківського контейнера.

Атрибут `android:layout_height="match_parent"` – встановлює висоту елемента такою ж, як висота батьківського контейнера.

Атрибут `android:orientation="vertical"` – встановлює орієнтацію контейнера по вертикалі, що означає, що елементи розташовуються вертикально один за одним.

Атрибут `android:padding="16dp"` – встановлює відступи для всіх сторінок контейнера, що додає простір навколо вмісту.

Після цього прописуємо код елемента `EditText` (рис. 3.4).

```
10     <EditText
11         android:id="@+id/taskEditText"
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:hint="Enter a task" />
```

Рисунок 3.4 – Фрагмент коду елемента `EditText`

Елемент `EditText` – виберіть поле для введення тексту, де користувач може ввести дані. В атрибутах цього елемента задаються його розміри та властивості.

Атрибут `android:id="@+id/taskEditText"` – встановлює унікальний ідентифікатор для елементів, що дозволяє підтримувати доступ до нього з кодом програми.

Атрибут `android:layout_width="match_parent"` – встановлює ширину елемента такою ж, як ширина батьківського контейнера.

Атрибут `android:layout_height="wrap_content"` автоматично встановлює висоту елемента, щоб вмістити його вміст.

Атрибут `android:hint="Enter a task"` встановлює текст-підказку, яка відображається в полі введення, щоб нагадати користувачам, яку інформацію ввести.

Після цього прописуємо код елемента `Button` (рис. 3.5).

```
16     <Button
17         android:id="@+id/addButton"
18         android:layout_width="wrap_content"
19         android:layout_height="wrap_content"
20         android:text="Add" />
```

Рисунок 3.5 – Фрагмент коду елемента `Button`

Елемент `Button` забезпечує вибір кнопки, на яку користувач може натиснути для виконання певних дій.

Атрибут `android:id="@+id/addButton"` – встановлює унікальний ідентифікатор для кнопок.

Атрибут `android:layout_width="wrap_content"` – автоматично встановлює ширину кнопки, щоб вмістити текст кнопки.

Атрибут `android:layout_height="wrap_content"` – автоматично встановлює висоту кнопки, щоб вмістити текст кнопки.

Атрибут `android:text="Add"` – встановлює текст, який буде відображатися на кнопці.

Після цього прописуємо код елемента `ListView` (рис. 3.6).

```

22     <ListView
23         android:id="@+id/taskListView"
24         android:layout_width="match_parent"
25         android:layout_height="match_parent"
26         android:layout_marginTop="16dp" />

```

Рисунок 3.6 – Фрагмент коду елемента `ListView`

Елемент `ListView` забезпечує вибір списку, у якому можна відобразити набір елементів. Атрибут `android:id="@+id/taskListView"` – встановлює унікальний ідентифікатор для списку.

Атрибут `android:layout_width="match_parent"` – встановлює ширину списку такою ж, як ширина батьківського контейнера.

Атрибут `android:layout_height="match_parent"` – встановлює висоту списку такою ж, як висота батьківського контейнера.

Атрибут `android:layout_marginTop="16dp"` – встановлює зовнішній відступ білого списку, що додає простір між верхнім краєм контейнера і списком.

Це основні елементи коду, які запускають розмітку графічного інтерфейсу користувача для `Android`-додатку. `LinearLayout` використовується для розташування `EditText`, `Button` і `ListView` у вертикальному порядку. З

використанням EditText користувач може ввести текстову інформацію, Button використовується як кнопка для додавання завдань, а також ListView використовується для відображення списку завдань.

3.4. Створення коду для формування списку завдань (to-do list) мобільного додатку Android

Розробка програми для мобільного додатку «Доступний менеджер покупок» включає у себе створення списку завдань (to-do list). При цьому насамперед слід визначити назву пакета (рис. 3.7).

```
8 package com.example.myapplication12;  
9  
10 import android.os.Bundle;  
11 import android.view.View;  
12 import android.widget.Button;  
13 import android.widget.EditText;  
14 import android.widget.ListView;  
15 import android.widget.AdapterView;  
16  
17 import androidx.appcompat.app.AppCompatActivity;  
18  
19 import java.util.ArrayList;
```

Рисунок 3.7 – Фрагмент коду для імпортування потрібних бібліотек

Зокрема, пакет com.example.myapplication12 визначає назву пакета, до якого належить клас MainActivity.

На наступному кроці виконуються імпортибібліотек:

- android.os.Bundle;
- android.view.View;
- android.widget.Button;
- android.widget.EditText;
- android.widget.ListView;
- android.widget.AdapterView.

Отже, описаний вище код використовують для імпорту останніх класів із фреймворка Android.

Після цього виконують імпорт AppCompatActivity, який є базовим класом для діяльності (активності) в Android і надає підтримку для старших версій Android.

На наступному кроці прописується код, який є основним шаблоном для активності у вигляді класу «MainActivity» у додатку (рис. 3.8).

```

21 public class MainActivity extends AppCompatActivity {
22
23     private EditText taskEditText;
24     private Button addButton;
25     private ListView taskListView;
26     private ArrayList<String> taskList;
27     private ArrayAdapter<String> adapter;
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         setContentView(R.layout.activity_main);
33
34         // Знаходження елементів інтерфейсу за допомогою їх ідентифікаторів
35         taskEditText = findViewById(R.id.taskEditText);
36         addButton = findViewById(R.id.addButton);
37         taskListView = findViewById(R.id.taskListView);
38
39         // Ініціалізація списку завдань та адаптера для ListView
40         taskList = new ArrayList<>();
41         adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, taskList);
42         taskListView.setAdapter(adapter);
43
44         // Встановлення слухача подій для кнопки "Додати"
45         addButton.setOnClickListener(new View.OnClickListener() {
46             @Override
47             public void onClick(View v) {
48                 // Отримання тексту з EditText
49                 String task = taskEditText.getText().toString();
50                 if (!task.isEmpty()) {
51                     // Додавання нового завдання до списку
52                     taskList.add(task);
53                     // Оновлення адаптера
54                     adapter.notifyDataSetChanged();
55                     // Очищення поля введення
56                     taskEditText.setText("");
57                 }
58             }
59         });
60     }
61 }

```

Рисунок 3.8 – Фрагмент коду класу «MainActivity»

На наступному кроці створюється об'єкт taskList типу ArrayList<String>, який буде використаний для зберігання завдань. Також створити об'єкт адаптер типу ArrayAdapter<String>, який використовується для відображення списку завдань у taskListView.

Нами використовується метод `setAdapter` для `taskListView`, в який встановлюється адаптер `adapter`. Це дозволяє відображати елементи зі списку завдань у `ListView`.

Встановлюється отримувач подій для кнопки «Додати» (`addButton`). При натисканні на кнопку виконується код у методі `onClick`.

У методі `onClick` забезпечується отримання тексту з `taskEditText` за допомогою методу `getText().toString()`. Якщо текст не є пустим, то він додається до `taskList` за допомогою методу `add()`.

Після цього адаптера переходять до змін за допомогою `adapter.notifyDataSetChanged()`, щоб повторно відтворити список. Нарешті, поле введення `taskEditText` очищується за допомогою `setText("")`, щоб готове поле для введення нового завдання.

Цей код створює функціональну складову мобільного додатку «Доступний менеджер покупок», де користувач може виконувати завдання, і вони будуть відображатися у списку. При цьому можна додати додатковий функціонал, такий як видалення завдань, редагування, визначення завдань як виконаних та інше, незалежно від ваших потреб.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1. Аналіз умов праці під час розроблення мобільного додатку

Проектування мобільних додатків виконують у проектному відділі, в якому наявні комп'ютери та принтер. У даному приміщенні температура повітря у теплий період року становить 30°C, використовується природне та штучне освітлення. Штучне освітлення виконано у вигляді переривчастих ліній світлодіодних світильників. Рівень шуму в приміщенні становить 54 дБ, а згідно з Державними санітарними нормами [13] не повинен перевищувати 50 дБ.

Робоче місце проєктанта розташоване так, щоб природне світло падало з лівої сторони, при цьому відстань зі світлом до робочого місця – 1 м. Висота робочої поверхні столу над підлогою 750 мм, глибина столу – 800 мм, ширина столу 1300мм. Робочий стіл має простір для ніг висотою 650 мм та шириною 600 мм.

Перелік шкідливих та небезпечних виробничих чинників.

Створення сприятливих умов праці, в роботі інженера-проектувальника, має велике значення як для полегшення, так і для підвищення продуктивності праці.

Відповідно до [8] шкідливими виробничими факторами є:

- ✓ підвищена температура робочого приміщення;
- ✓ недостатня освітленість робочої поверхні;
- ✓ виробничий шум;
- ✓ електромагнітні випромінювання радіочастотного діапазону;
- ✓ іонізуючі випромінювання.

Відповідно до [13] робота інженера-проектувальника у приміщенні з енерговитратами 90-120 ккал/год. відносяться до категорії легких фізичних

робіт Ia (роботи, що виконуються сидячи і не потребують фізичного напруження) (табл. 4.1-4.2).

Таблиця 4.1 – Оптимальні величини температури

Період року	Категорія робіт	Температура повітря, °С
Холодний період року	Легка Ia	22-24
Теплий період року		23-25

Таблиця 4.2 – Допустимі величини температури на постійних робочих місцях

Період року	Категорія робіт	Температура повітря, °С	
		Верхня межа	Нижня межа
Холодний період року	Легка Ia	25	21
Теплий період року		28	22

У проектному відділі температура повітря становить 30°C в теплий період року, що перевищує допустиму на 2°C. Забезпечили температуру приміщення 23°C, за допомогою механічної вентиляції з вентилятором VORTICE VARIO, повітрообмін якого становить 680 м³/год.

В приміщенні встановлені персональні комп'ютери, присутнє природне та штучне освітлення. За вимогами [9], величина коефіцієнта природної освітленості повинна бути не менше 1.5%. В проектному відділі порушенні вимоги, освітленість робочої поверхні складає 370 лк, а коефіцієнт освітленості складає 1.2%. Природне світло проникає у приміщення через бічні світло прорізи. Вікна мають жалюзі. Штучне освітлення виконано у вигляді переривчастих ліній світлодіодних світильників, розташованих паралельно лінії зору інженера-проектувальника. Для місцевого освітлення використовувати галогенні лампи розжарювання

Шум на робочому місці створюється: комп'ютером та периферійним пристроєм. Допустимі рівні звукового тиску на робочому місці повинні відповідати вимогам [13] (табл. 4.3).

Таблиця 6.3 – Санітарні норми виробничого шуму, ультразвуку та інфразвуку

Вид трудової діяльності, робоче місце	Рівні шуму та еквівалентні рівні шуму, ДБА, дБАекв
Конструювання та проектування.	50

Реальний рівень шуму в проектному відділі становить 54 дБ, що перевищує допустимий рівень. Для зменшення рівня шуму рекомендується використовувати місцеву та загальну звукоізоляцію, шумопоглинаючі екрани, поглинаючі фільтри.

4.2. Організаційні та конструктивно-технологічні заходи для зниження впливу шкідливих виробничих факторів

Для створення й автоматичної підтримки в ІТ відділі незалежно від зовнішніх умов оптимальних значень температури, вологості, чистоти і швидкості руху повітря, у холодний час року використовується водяне опалення, у теплий час року застосовується кондиціонування повітря [8].

Під час аналізу освітлення на робочому місці програміста було встановлено, що воно не відповідає встановленим нормам, тому для покращення умов праці рекомендуємо збільшити рівень загальної освітленості приміщення шляхом встановлення 5 додаткових світильників, щоб загальна кількість лам відповідала розрахованому вище значенню, а саме 36 світлодіодних ламп. Також для підтримки запроектованого освітлення у

чистому виді необхідно скласти графік, де передбачити очищення віконних блоків і світильників не менше 2 разів на рік [13].

Електробезпеність у приміщенні ІТ відділу пропоную забезпечити наступними технічними способами і засобами захисту:

✓ для зменшення накопичення статичної електрики застосовувати зволожувачі і нейтралізатори, антистатичне покриття підлоги;

✓ забезпечити приєднання металевих корпусів устаткування до жили, що заземлює. Заземлення корпусу ПК забезпечити підведенням жили, що заземлює, до розеток. Опір заземлення 4 Ом, згідно (ПУЕ) для електроустановок з напругою до 1000 В.

А також організаційними заходами – своєчасне проведення інструктажів з техніки безпеки [9].

Після проведення аналізу робочого місця програміста в відділі проектування було з'ясовано, що воно відповідає встановленим вимогам. Виходячи з результатів аналізу важкості та напруженості праці пропоную скоротити час роботи за комп'ютером, робити перерви сумарний час яких повинен складати 50 хвилин при 8-ми годинному робочому дні [8].

4.3. Аналіз інформаційних небезпек під час розроблення мобільного додатку

Аналіз небезпечних і шкідливих факторів під час розробки мобільного додатку є важливим складним процесом забезпечення якості та безпеки продукту. Ось деякі небезпечні і шкідливі фактори, які можуть виникнути під час розробки мобільного додатка і вимагають уваги.

Недостатня безпека даних є важливою складовою, яку слід враховувати під час розробки мобільних додатків. Зокрема, під час розробки мобільного додатку «Доступний менеджер покупок» на базі ОС Android до процесів отримання інформації необхідно забезпечити заходи безпеки для захисту

конфіденційності інформаційних користувачів, таких як паролі, особливості даних та фінансів. Важливо досягти кращої практики шифрування даних та міцності систем автентифікації.

Вразливість програмного забезпечення безпеки має важливе значення. Під час розробки додатку необхідно перевірити код наявності наявних вразливостей на безпеки, таких як Cross-Site Scripting (XSS) або SQL Injection. Важливо проводити регулярні аудиторські перевірки безпеки для виявлення та усунення можливих загроз.

Розробка мобільного додатка не завершується з моменту його випуску. Важливо забезпечити регулярне оновлення, виправлення помилок та підтримку нових версій операційної системи. Відсутність таких оновлень може призвести до проблем безпеки та несумісності з новими функціями ОС.

Додаток повинен працювати ефективно, не заважаючи нормальній роботі пристрою користувача. Неможливе програмування або неправильне управління ресурсами може призвести до експоненційного зростання споживання батареї заряду батареї, перегріву пристрою та зниження продуктивності. Під час розроблення добавки необхідно оптимізувати його роботу, враховуючи обмежені ресурси мобільних пристроїв.

Розробники повинні збільшити різноманітність моделей мобільних пристроїв та версій операційних систем Android. Додаток повинен бути протестований на різних пристроях, щоб переконатися, що він працює правильно та не має проблем зі сумісністю.

Під час розробки додатку необхідно важливо перевірити, чи не відбувається витік конфіденційної інформації через незахищені канали передачі даних або неправильного управління кешем. Необхідно використовувати безпечні методи передачі та збереження даних.

Недостатнє або неправильне тестування може призвести до виявлення помилок та після недоліків у додатку його випуску. Важливо проводити ретельне тестування, включаючи функціональні тести, тести безпеки, тести на стійкість та тестування сумісності, щоб забезпечити високу якість продукту.

Врахування цих небезпечних і шкідливих факторів під час розробки мобільного додатка «Доступний менеджер покупок» забезпечує безпеку, якість та задоволення користувачів. Важливо аналізувати ці фактори і приділяти їм належну увагу в процесі розробки таких додатків.

4.4. Пожежна безпека

Відповідно до [9] дане приміщення відноситься до категорії В по вибуховопожежній та пожежній небезпеці із-за використання у ньому твердих горючих матеріалів з температурою спалаху понад 61°C.

Проектний відділ оснащено двома безпроводними датчиками детектування диму SD-02 (оповіщає при задимленні приміщення; площа обслуговування до 20 м²). Окрім того є два порошкові вогнегасники ВП-5 (для приміщення категорії В за відсутності горючих газів і рідин, площею до 50 м² і масою вогнегасної речовини – 5 кг, мінімальна кількість порошкових вогнегасників 2).

LifeSOS LS-30LR бездротова пожежно-охоронна система (при детектуванні вторгнення, датчики передають на центральний блок сигнал тривоги по радіоканалу без проводів. Централь приймає сигнал від датчиків, включає сирену, відправляє інформацію на пульт централізованого нагляду, дзвонить на зазначені телефонні номери та відправляє SMS повідомлення з повідомленнями про тривогу.)

Для попередження виникнення пожеж проводяться організаційно-технічні заходи пожежної безпеки, які включають:

- включення питань пожежної безпеки у всі інструкції по техніці безпеки;
- виконання встановленого режиму експлуатації електричних мереж та обладнання;
- заборона куріння в недозволеному місці;

- видання необхідних інструктажів, планів евакуації (рис. 4.1).

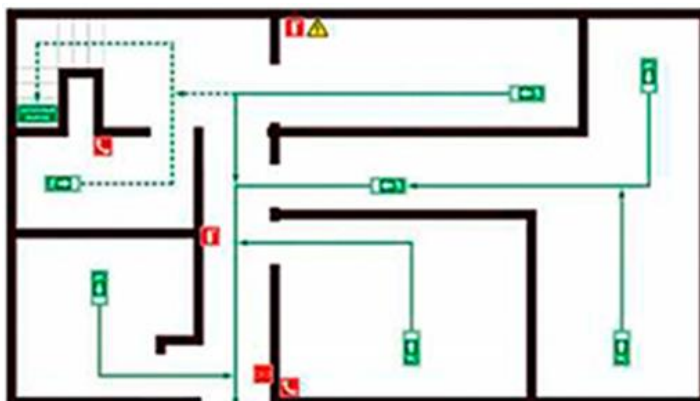


Рисунок 4.1 – План евакуації на випадок пожежі

План евакуації у випадку пожежі складається з графічної і текстової частин. Графічна частина являє собою схематичний план поверху (рис. 4.1), в якому зеленими суцільними стрілками вказують шляхи евакуації, що ведуть до основних евакуаційних виходів, а пунктирними зеленими стрілками - до аварійних виходів. Двері на шляху евакуації відчиняються назовні у напрямку виходу з будівлі. На плані евакуації умовними знаками показано розміщення вогнегасників, пожежних гідрантів, телефонів, аптечок медичної допомоги, електрощитів, датчиків диму, системи охоронно-пожежної сигналізації.

4.5. Заходи з охорони праці під час розробки мобільного додатку

Інструкція з охорони праці під час роботи з персональним комп'ютером має на меті забезпечити безпеку, здоров'я та комфорт працівника. Нами пропонуються деякі рекомендації, які варто дотримуватися.

Розміщення робочого місця має вагомий вплив на безпеку працівників. Правильне розташування робочого місця має велике значення для комфорту та безпеки. Переконайтеся, що стіл і є місце належним чином налаштовані.

Сидячи за столом, ваші ноги повинні стояти на підлозі, а коліна повинні бути в куті приблизно 90 градусів.

Розташуйте клавіатуру і мишу на достатній відстані від краю столу, щоб ваші зап'ястя були прямими під час введення даних. Забезпечте, щоб рука була в рівній площині злегка зігнута в лікті при роботі з мишкою.

Налаштуйте монітор таким чином, щоб верхній край екрана був на рівні очей або внизу. Це зменшити напруження на шії і очі. Регулюйте яскравість і контрастність монітора, щоб уникнути надмірного напруження очей.

Забезпечте достатнє освітлення, щоб уникнути напруження очей. Уникайте прямого світла, яке відбивається від монітора. Зручною альтернативою є природне освітлення, якщо це можливо.

Регулярно робіть перерви під час тривалої роботи з комп'ютером. Під час перерв виконуйте вправи для розтягування та зміцнення м'язів, особливо спини. Наприклад, підніміть і опустіть плечі, обертайте голову вліво і вправо, робіть коливні рухи пальцями і зап'ястями. Це допоможе попередити напругу м'язів і тому.

Уникайте тривалої непорушної позиції. Зробіть короткі перерви для зміни позиції та розтягування м'язів. Приймайте розумні позиції під час роботи, уникайте скручування тіла та напруження м'язів.

Забезпечте чистоту робочого місця і періодично витирайте клавіатуру, мишу та монітор. Використовуйте антибліковий фільтр на моніторі для зменшення відблисків і напруги на очах.

Здоровий спосіб життя передбачає займайтеся регулярною фізичною активністю поза робочим часом, правильно харчуйтеся і намагайтеся зберегти позитивний баланс між роботою і відпочинком. Загальна фізична та психологічна підтримка вашого здоров'я вплине на вашу продуктивність та добробут.

Ці рекомендації повністю забезпечити безпеку та комфорт під час розробки мобільного додатку.

ВИСНОВКИ І ПРОПОЗИЦІЇ

У нашій кваліфікаційній роботі було проведено детальне дослідження та розроблено мобільний додаток під назвою «Доступний менеджер покупок» на базі ОС Android. За результатами виконання кваліфікаційної роботи були отримані наступні висновки.

Розробка мобільного додатка «Доступний менеджер покупок» на базі ОС Android є актуальною та важливою задачею в сучасному світі, оскільки все більше користувачів переглядають покупки через мобільні пристрої. Проведений аналіз потреб та користувачів підтвердив доцільність створення додатка, що надає зручний та доступний інтерфейс для управління покупками.

У процесі розробки була використана ОС Android, яка є популярною платформою для розробки мобільних додатків. Це дозволило забезпечити широке охоплення користувачів та зручність використання додатка.

Додаток «Доступний менеджер покупок» розроблено з урахуванням принципів доступності та універсальності з позицій проектування. Це забезпечує зручність використання запропонованого мобільного додатка для користувачів.

У ході розробки додатку були використані сучасні інструменти та технології, що дозволили забезпечити якість та ефективність роботи додатка. У розробці було враховано особливості інтерфейсу ОС Android, що дозволило забезпечити гармонійну інтеграційну додатку з платформою та забезпечити зручний досвід використання.

Функціональне наповнення включає можливість створення списків покупок, відстеження витрат, пошуку та додавання товарів, нагадування про заплановані покупки та інші корисні функції. Це допомагає ефективному управлінню покупками та забезпечує фінансову свідомість.

Проведено тестування додатка з використанням реальних користувачів, які дозволили виявити та виправити помилки та недоліки. Отримані результати підтвердили ефективність та корисність мобільного додатка.

Розроблений додаток може бути використаний як для окремих користувачів під час планування та керування своїми покупками, так і бізнес-користувачами для ведення обліку товарів та контролю витрат.

Додаток має потенціал для подальшого розширення функціональності, наприклад, інтеграції з онлайн-магазинами, можливість поділитися списками покупок з іншими користувачами, використання технологій розпізнавання зображення для швидкого додавання товарів тощо.

У своїй роботі ми пропонуємо заходи щодо забезпечення безпеки праці під час розробки мобільних додатків, забезпечуючи тим самим покращення умов праці та мікроклімату робочого місця.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Барановський, А. В. Програмування мобільних додатків: навчальний посібник. А. В. Барановський, В. І. Горошко, В. О. Кондратюк. К.: НТУУ «КПІ», 2020. 234 с.
2. Берестянський, Д. Мобільна розробка на платформі Android. Одеса: Видавництво "Фенікс". 2019. 523с.
3. Геврик Є.О., Пешко Н.П. Гігієна праці на виробництві: Навчальний посібник для студентів вищих навчальних закладів. К.: Чельга, Ніка, Центр, 2004. 280 с.
4. Гейко А. В. Перспективи використання мобільної реклами. Сучасний менеджмент і економічний розвиток: матеріали Міжнародної науково-практичної інтернет-конференції, що постійно діє, 1 вересня 2015 р. – 31 серпня 2016 р. Суми, 2016. [Електронний ресурс] – Режим доступу: https://essuir.sumdu.edu.ua/bitstreamdownload/123456789/47974/1/Geiko_Mobilna_reklama.pdf.
5. Гулак, О. Розробка Android-додатків: від початку до публікації. Київ: Видавництво "КМ Академія". 2017. 126с.
6. Давидов М.В., Демчук А.Б., Лозинська О.В. Програмне забезпечення мобільних пристроїв: навчальний посібник. Львів: Видавництво «Новий Світ-2000» 2020. 218 с.
7. Дворецький М. Л., Нездолій Ю. О., Дворецька С. В., Кандиба І. О. Розробка мобільних застосунків для OS Android : навч. посіб. Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2021. 140 с.
8. Жидецький В.Ц. Охорона праці користувачів комп'ютерів. Львів: Афіша, 2000. 176 с.
9. Жидецький В.Ц., Джигерей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доп. Львів: Афіша, 2000. 350 с.
10. Захаров, В. Розробка Android-додатків на мові Kotlin. Київ: Видавництво "Графіка". 2019. 456с.

11. Ішанська Н, Улько С. Основні аспекти створення мобільних додатків та вибір інструментів їх розробки. Системи управління, навігації та зв'язку. Збірник наукових праць. Полтава: ПНТУ, 2020. Т. 1 (59). С. 74-78. doi:<https://doi.org/10.26906/SUNZ.2020.1.074>.
12. Капустян В., Шокалюк О. Розробка мобільних додатків на платформі Android. Львів: Видавництво Львівської політехніки. 2017. 147с.
13. Катренко Л.А., Катренко А.В., Охорона праці в галузі комп'ютерингу: Підручник. За науковою редакцією В.В. Пасічника. Львів: «Магнолія 2006», 2012. 544 с.
14. Корпанюк Т.М., Мулик Я.І. Застосування мобільних додатків в бізнесі та їх облік. Ефективна економіка. №3, 2018. [Електронний ресурс] – Режим доступу: http://www.economy.nauka.com.ua/pdf/3_2018/59.pdf
15. Лещенко В. Розробка мобільних додатків на платформі Android. Київ: Видавничий дім "Києво-Могилянська академія". 2019. 265с.
16. Лопатін, О. Мобільна розробка на платформі Android. Київ: Видавничий дім "Либідь". 2018. 364с.
17. Мина Ж.В., Кулічкова М.О. Формування незалежного інтернет-ринку та розроблення електронного каталогу в додатку Android для ефективного продажу товарів. Прикладні соціально-комунікаційні технології. Том 32 (71) № 5. Ч.2. 2021.С.237-242.
18. Поляков, В. Розробка мобільних додатків на платформі Android. Харків: Видавництво "Слово". 2018. 85с.
19. Столярчук А. Мобільна розробка на платформі Android. Львів: Видавництво Львівської політехніки. 2016. 354с.
20. Чаплінський Ю.П. Мобільні інформаційні системи підтримки прийняття рішень. Наукова-технічна інформація. № 1. 2003. С. 22-26.
21. Шкраба, В. Мобільна розробка на Android для початківців. Київ: Видавництво "Майстер-клас". 2018. 65с.
22. Якимович, О. Android-розробка: практичний підхід. Київ: Видавничий дім "Слово". 2016. 142с.

23. Яремко С.А., Кузьміна О.М., Бевз С.В. Удосконалення електронного бізнесу на базі мобільних платформ. Вісник Хмельницького національного університету. №1. 2020. С. 165-169.

24. Ярова І. Інтернет як інструмент просування продукції промислових підприємств. Економіка України. 2016. № 3. С. 48–54.

25. 21 best mobile apps for small business owners. [Електронний ресурс] – Режим доступу: <https://www.wix.com/blog/>

26. Bahr O. 6 Benefits of Having a Mobile App for Your Business [Електронний ресурс] – Режим доступу: <https://www.entrepreneur.com/growing-a-business/6-reasons-to-consider-making-a-mobile-app-for-your-business/443863>

27. Create a Shopping App. [Електронний ресурс] – Режим доступу: <https://buildfire.com/create-shopping-app/>

28. Goransson A. Efficient Android Threading: Asynchronous Processing Techniques for Android Applications. O'Reilly Media : 1-st ed. June 13, 2014. 280 p.

29. Meier R. Professional Android, 4-th ed. Wrox, 2018. 928 p.

30. Mobile App Download Statistics & Usage Statistics (2023) [Електронний ресурс] – Режим доступу: <https://buildfire.com/app-statistics/>

31. Mobile application: definition, technology types and examples 2023 [Електронний ресурс] – Режим доступу: <https://magenest.com/en/mobile-application/>

32. Phillips B., Stewart C., Marsicano K. Android Programming: The Big Nerd Ranch Guide : 3-rd ed. February 9, 2017. 624 p.

33. Rouse M. Mobile Application [Електронний ресурс] – Режим доступу: <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>

34. Sheich M. 14 best shopping apps of 2022. [Електронний ресурс] – Режим доступу: <https://sproutsocial.com/insights/shopping-apps/>

35. What Are the Different Types of Mobile Apps? And How Do You Choose? [Електронний ресурс] – Режим доступу: <https://clevertap.com/blog/types-of-mobile-apps/>

36. What Is a Mobile Application? [Электронный ресурс] – Режим доступа:
<https://www.outsystems.com/glossary/what-is-mobile-application/>
37. Why mobile online shopping matters or how to build a store app to increase your sales [Электронный ресурс] – Режим доступа:
<https://anywhere.epam.com/business/how-to-create-an-online-shopping-app>
38. Zechne M. Beginning Android Games : 3-rd ed. Apress, 2016. 619 p.

Додатки

Додаток А

Фрагменти програмного коду для формування списку завдань (to-do list) мобільного додатку Android

```

package com.example.myapplication12;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.AdapterView;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    private EditText taskEditText;
    private Button addButton;
    private ListView taskListView;
    private ArrayList<String> taskList;
    private ArrayAdapter<String> adapter;

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

        // Знаходження елементів інтерфейсу за допомогою їх
        ідентифікаторів
        taskEditText = findViewById(R.id.taskEditText);
        addButton = findViewById(R.id.addButton);
        taskListView = findViewById(R.id.taskListView);

        // Ініціалізація списку завдань та адаптера для ListView
        taskList = new ArrayList<>();
        adapter = new ArrayAdapter<>(this,
        android.R.layout.simple_list_item_1, taskList);
        taskListView.setAdapter(adapter);

        // Встановлення слухача подій для кнопки "Додати"
        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Отримання тексту з EditText
                String task = taskEditText.getText().toString();
                if (!task.isEmpty()) {
                    // Додавання нового завдання до списку
                    taskList.add(task);
                    // Оновлення адаптера
                    adapter.notifyDataSetChanged();
                    // Очищення поля введення
                    taskEditText.setText("");
                }
            }
        });
    }
}

```


Фрагменти програмного коду для створення розмітки графічного інтерфейсу користувача (GUI) для Android-додатку

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp"
  tools:context=".MainActivity">

  <EditText
    android:id="@+id/taskEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter a task" />

  <Button
    android:id="@+id/addButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add" />

  <ListView
    android:id="@+id/taskListView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="16dp" />

</LinearLayout>

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">

  <com.google.android.material.appbar.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/Theme.MyApplication12.AppBarOverlay">

    <androidx.appcompat.widget.Toolbar
      android:id="@+id/toolbar"
      android:layout_width="match_parent"
      android:layout_height="?attr/actionBarSize"
      android:background="?attr/colorPrimary"
      app:popupTheme="@style/Theme.MyApplication12.PopupOverlay"
    />

  </com.google.android.material.appbar.AppBarLayout>

  <include layout="@layout/content_main" />

```

```

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_marginEnd="@dimen/fab_margin"
    android:layout_marginBottom="16dp"
    app:srcCompat="@android:drawable/ic_dialog_email" />

```

```

</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_main">

    <fragment
        android:id="@+id/nav_host_fragment_content_main"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.gallery.GalleryFragment">

```

```

    <TextView
        android:id="@+id/text_gallery"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:textAlignment="center"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```