

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення

комп'ютерних технологій

Наталія СТЕФУРАК / _____ /

Підпис

«___» _____ 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи
освітньо-професійного ступеня «фаховий молодший бакалавр»
зі спеціальності 122 «Комп'ютерні науки»

на тему: «Мобільний застосунок для підбору комп'ютерних комплектуючих»

Студент групи КН-41

Віталій БОДНАР

(підпис)

Керівник роботи

Ольга ПОСВЯТОВСЬКА

(підпис)

Консультанти:

з техніко-економічного

обґрунтування

Любов МЕЛЕНЧУК

(підпис)

нормоконтролер

Надія ГАВРИШКІВ

(підпис)

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділення

комп'ютерних технологій

Наталія СТЕФУРАК / _____ /

підпис

«___» _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»

студенту Боднару Віталію Тарасовичу

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

1. Тема роботи Мобільний застосунок для підбору комп'ютерних
комплектуючих

затверджена наказом по коледжу від “25” Листопада 2024 р., №253а-н

2. Термін здачі студентом завершеної роботи “24” Червня 2025 р.

3. Вихідні дані до роботи: результати аналізів мобільних застосунків для
підбору комп'ютерних комплектуючих.

4. Перелік питань, які повинні бути розроблені:

а) основна частина: аналіз предметної області та постановка завдань;
проектування системи; реалізація та тестування системи.

б) техніко-економічне обґрунтування: аналіз ринку; розрахунок витрат на
проектування; обґрунтування необхідності розробки.

5. Перелік графічного матеріалу: схема взаємодії; діаграма використання;
структурна схема.

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
З техніко-економічного обґрунтування	Любов МЕЛЕНЧУК	20.04.2025	20.05.2025

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1	Пошук теми роботи, вивчення вимог для кваліфікаційної роботи	16.11.2024	25.11.2024
2	Аналіз існуючих рішень та визначення їхніх переваг і недоліків	26.11.2024	30.11.2024
3	Вивчення технологій та інструментів реалізації.	01.12.2024	03.12.2024
4	Формування архітектури застосунку, визначення функціональних вимог.	04.12.2024	11.12.2024
5	Підготовка та налаштування програмного середовища	12.12.2024	29.12.2024
6	Проектування структури застосунку, бази даних та інтерфейсу користувача	30.12.2024	21.01.2025
7	Реалізація основної логіки застосунку	22.01.2025	12.04.2025
8	Оптимізація коду та вдосконалення окремих модулів	13.04.2025	22.04.2025
9	Тестування застосунку та усунення помилок	23.04.2025	02.05.2025
10	Аналіз ринку збуту, визначення користувачів і оцінка економічної доцільності	03.05.2025	13.05.2025
11	Оформлення пояснювальної записки.	14.05.2025	12.06.2025
12	Попередній захист кваліфікаційної роботи	13.06.2025	13.06.2025
13	Підготовка до захисту кваліфікаційної роботи	14.06.2025	23.06.2025
14	Захист кваліфікаційної роботи	24.06.2025	24.06.2025

Дата видачі “ ___ ” _____ 202_ р. Керівник _____ / Посвятовська О. Б.

Завдання прийняв до виконання _____ / Боднар В.Т.

Реферат

Мобільний застосунок для підбору комп'ютерних комплектуючих

Кваліфікаційна робота. Боднар Віталій Тарасович. Галицький фаховий коледж імені В'ячеслава Чорновола, відділення комп'ютерних технологій. Спеціальність 122 «Комп'ютерні науки», 2025. Сторінок – 67, рисунків – 30, таблиць – 3, додатків – 5, джерел – 9.

Об'єктом кваліфікаційної роботи є процес підбору комп'ютерних комплектуючих за допомогою мобільного застосунку.

Метою роботи є створення зручного та функціонального інструменту, що допомагає користувачам формувати конфігурації ПК з урахуванням сумісності компонентів.

У роботі реалізовано мобільний застосунок «Build PC», що включає підбір комп'ютерних комплектуючих, перевірку їх сумісності, реєстрацію користувачів, збереження конфігурації, перегляд деталей компонентів та зручний інтерфейс.

Застосунок створено з використанням Android Studio, Java, SQLite, а дизайн розроблено у Figma.

Програмний продукт орієнтовано на українських користувачів, протестовано на практичну придатність та готово до впровадження.

МОБІЛЬНИЙ ЗАСТОСУНОК, КОМПЛЕКТУЮЧІ, СУМІСНІСТЬ, КОНФІГУРАЦІЯ ПК, ANDROID, SQLITE, JAVA, ІНТЕРФЕЙС КОРИСТУВАЧА.

Abstract

Mobile Application for Selecting Computer Components

Qualification work. Bodnar Vitalii Tarasovych. Halych Vocational College named after Vyacheslav Chornovil, Department of Computer Technologies. Specialty 122 "Computer Science", 2025. Pages – 67, figures – 30, tables – 3, appendices – 5, sources – 9.

The object of the qualification work is the process of selecting computer components using a mobile application.

The aim of the work is to create a convenient and functional tool that helps users build PC configurations taking into account component compatibility.

The "Build PC" mobile application has been implemented in the project, featuring the selection of computer components, compatibility checking, user registration, configuration saving, component detail viewing, and a user-friendly interface.

The application was developed using Android Studio, Java, and SQLite, with the UI/UX design created in Figma.

The product is intended for Ukrainian users, has been tested for practical usability, and is ready for deployment.

MOBILE APPLICATION, COMPONENTS, COMPATIBILITY, PC CONFIGURATION, ANDROID, SQLITE, JAVA, USER INTERFACE.

ЗМІСТ

Вступ	8
1 Аналіз предметної області та постановка завдань.....	9
1.1 Опис предметної області.....	9
1.2 Аналіз наявних рішень	10
1.3 Аналіз вимог до програмного засобу та постановка завдання	19
2 Проєктування мобільного застосунку	20
2.1 Проєктування структурної схеми мобільного застосунку	20
2.2 Проєктування бази даних мобільного застосунку	22
2.3 Функціональна структура мобільного застосунку	26
2.4 Проєктування інтерфейсу мобільного застосунку	27
3 Розробка мобільного застосунку	33
3.1 Розробка інтерфейсу застосунку	33
3.2 Розробка функціоналу застосунку.....	47
3.3 Тестування застосунку Build PC.....	51
4 Техніко-економічне обґрунтування	60
4.1 Аналіз ринку	60
4.2 Розрахунок витрат на розробку	61
4.3 Обґрунтування необхідності розробки.....	64
Висновки	66
Перелік джерел посилання.....	67
Додатки	68

					КР.КН 25.584.03.000 ПЗ			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Мобільний застосунок підбір комп'ютерних комплектуючих	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
Розроб.		Боднар В. Т.						
Перевір.		Посвятовська О.					6	67
Рценз		Гавришків Н.				ГФК. ВКТ. КН-41		
Н.контр.		Гавришків Н.						
Зав. від.		Стефурак Н.						

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

RAM – Random Access Memory.

CPU – Central Processing Unit.

GPU – Graphics Processing Unit.

DB – Database

UI – User Interface.

UX – User Experience.

ID – Identifier.

MB – Motherboard.

ПК – Персональний комп'ютер.

ПЗ – Програмне забезпечення.

СУБД – Система управління базами даних.

ЄСВ – Єдиний соціальний внесок.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

У сучасному світі комп'ютерні комплектуючі відіграють ключову роль у забезпеченні продуктивної роботи користувачів, незалежно від того, чи йдеться про офісні завдання, ігрові системи чи професійні робочі станції. Вибір оптимальних компонентів для комп'ютера може бути складним завданням через велику кількість доступних моделей та технічних характеристик.

Саме тому, є затребуваними зручні та функціональні інструменти, які допомагають користувачам швидко та безпомилково підібрати сумісні комплектуючі відповідно до своїх потреб та бюджету, зокрема у вигляді мобільних застосунків. Дані мобільні застосунки використовуються з метою спрощення процесу створення підбору сумісних комп'ютерних комплектуючих.

Актуальні застосунки орієнтуються як на досвідчених користувачів, які вже мають знання у сфері ПК, так і на новачків, яким потрібна допомога в підборі конфігурацій. Такі програмні продукти дозволяють зручно переглядати базу даних комплектуючих, отримувати рекомендації, перевіряти сумісність між компонентами, зберігати збірки та працювати з ними в офлайн-режимі.

Популярністю користуються мобільні застосунки, в яких створено інтуїтивно зрозумілий інтерфейс користувача, що забезпечує швидкий доступ до основних функцій та зручну навігацію.

Розробка такого застосунку вимагає аналізу предметної області, моделювання бази даних, реалізації алгоритмів сумісності, дизайну та інтеграції всіх компонентів у єдину систему, а також створення інтуїтивного інтерфейсу користувача.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАНЬ

1.1 Опис предметної області

Ринок комп'ютерних комплектуючих стрімко розвивається і постійно змінюється, що створює як нові можливості, так і виклики для користувачів. З появою нових моделей процесорів, відеокарт, материнських плат, пам'яті та інших пристроїв вибір найбільш відповідного набору компонентів перетворюється у складну задачу, яка вимагає врахування широкого спектру технічних характеристик і параметрів сумісності.

Для звичайного користувача, особливо без досвіду в цій сфері, процес вибору комплектуючих може виявитися занадто складним. Наприклад, невідповідність сокету процесора й материнської плати або неврахування потужності блока живлення може зробити зібрану систему нефункціональною. Також часто користувачі не зважають на фізичні розміри компонентів чи стандарти інтерфейсів, що призводить до несумісності або навіть ушкодження техніки.

Традиційно користувачі змушені шукати допомогу на тематичних форумах, у відеооглядах, чи консультантів магазинів або ж збирати комп'ютери самостійно, ризикуючи марно витратити кошти. Проте такі джерела часто надають суб'єктивну або застарілу інформацію, не враховуючи потреб конкретного користувача.

Саме тому, розробка мобільного застосунку, що дозволяє автоматизувати процес підбору комп'ютерних комплектуючих, є актуальною. Такий застосунок здатний надати інтелектуальні рекомендації, перевірити сумісність компонентів та допомогти сформувати оптимальну конфігурацію ПК та побажань користувача.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

Застосунок має стати доступним інструментом для широкого кола користувачів – від новачків до досвідчених людей. Він повинен мінімізувати кількість помилок під час вибору комплектуючих, підвищити ефективність складання ПК та забезпечити зручний, швидкий і сучасний підхід до створення персоналізованих конфігурацій комп'ютерів.

Отже, предметна область охоплює широкий спектр знань з апаратного забезпечення, логіки сумісності компонентів, тенденцій ринку, потреб користувачів та інформаційних технологій, що використовуються для розробки програмного забезпечення. Глибоке розуміння цієї області є критично важливим для створення якісного продукту, який дійсно полегшить життя користувачам та дозволить уникати технічних помилок під час складання ПК.

1.2 Аналіз наявних рішень

Перш ніж розпочати проєктування програмного виробу, варто проаналізувати вже існуючі або подібні програмні рішення. Аналіз існуючих програмних засобів у сфері мобільних застосунків для підбору комп'ютерних комплектуючих є важливим етапом у визначенні необхідних функцій та особливостей майбутнього застосунку. У цьому розділі буде проведено огляд та порівняльний аналіз трьох популярних та ефективних інструментів:

- PC Creator 2.
- PC Builder.
- Gaming PC Builder.

Кожен із цих застосунків має свої позитивні сторони, а також певні недоліки або обмеження, які варто врахувати при розробці нового продукту. Отримані результати аналізу дозволять зосередитися на розробці функціоналу, що відповідає сучасним вимогам ринку, та допоможуть уникнути створення продукту з численними недоліками заздалегідь.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

PC Creator 2 – це інноваційний мобільний застосунок, що виступає у ролі персонального консультанта при формуванні конфігурацій комп'ютерів. Застосунок дозволяє користувачам створювати індивідуальні збірки, враховуючи технічні вимоги, бюджет та сумісність компонентів. Основні особливості PC Creator 2:

- Динамічний підбір компонентів. Кожна конфігурація формується з урахуванням технічних характеристик, сумісності та фінансових обмежень користувача.

- Інтерактивна візуалізація. Застосунок надає можливість перегляду побудованої конфігурації у вигляді схематичних зображень, що допомагає краще розуміти розташування та сумісність компонентів.

- Інтелектуальні рекомендації. На основі аналізу технічних даних та відгуків користувачів застосунок пропонує оптимальні варіанти комплектуючих.

- Регулярне оновлення бази даних. Завдяки постійним оновленням інформації, PC Creator 2 завжди містить актуальні дані про новітні моделі та ринкові ціни.

PC Creator 2 можна знайти на Google Play або App Store. Переглянути інтерфейс застосунок можна на рисунку 1.1



Рисунок 1.1 – Застосунок PC Creator 2

Цей застосунок допомагає уникнути помилок при підборі компонентів і забезпечує швидкий та зручний процес формування оптимальної конфігурації ПК.

PC Builder – мобільний застосунок, який надає користувачам можливість детально підбирати комп'ютерні комплектуючі за допомогою розширених фільтрів і налаштувань. Основні функції застосунку:

- Розширений пошук та фільтрація. Користувач може здійснювати пошук компонентів за характеристиками, брендами, цінами та сумісністю.

- Порівняння конфігурацій. Застосунок дозволяє порівнювати декілька варіантів збірок, що сприяє прийняттю обґрунтованого рішення.

- Інтуїтивно зрозумілий інтерфейс. Простий і зручний дизайн забезпечує швидке освоєння інструменту як для новачків, так і для досвідчених користувачів.

- Інтеграція з онлайн-магазинами. Можливість перевірки актуальності цін та наявності компонентів у популярних інтернет-магазинах допомагає оптимізувати процес покупки.

Цей застосунок стане у нагоді користувачам, які шукають комплексний та гнучкий інструмент для формування власної конфігурації ПК. Переглянути інтерфейс застосунку можна на рисунку 1.2

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

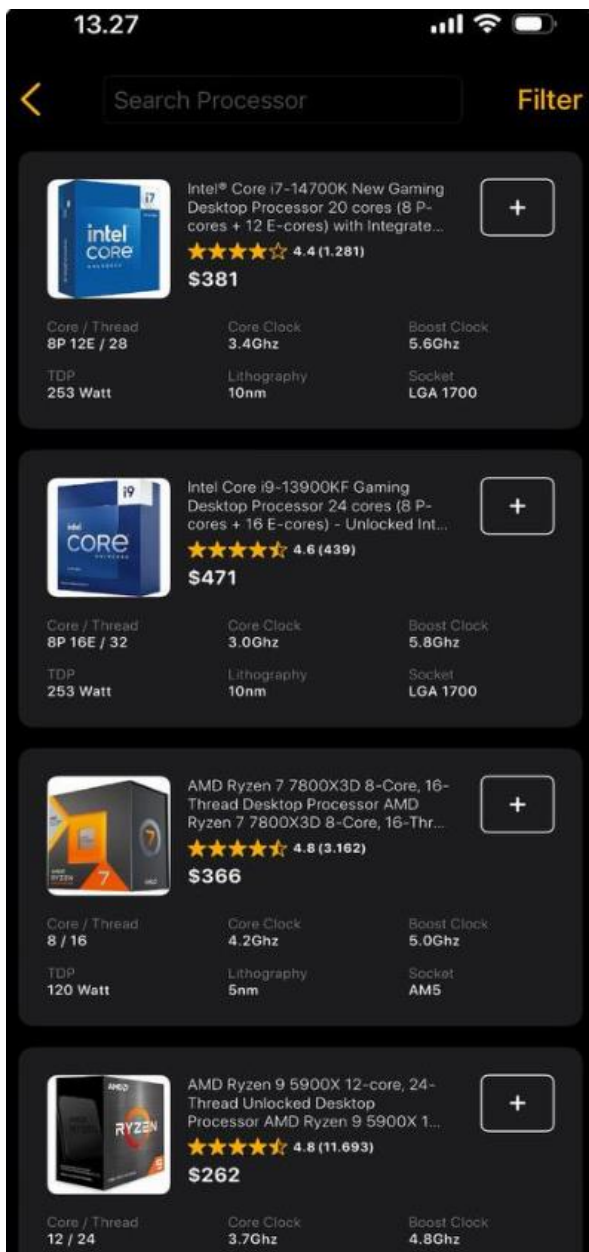


Рисунок 1.2 – Застосунок PC Builder

Цей застосунок стане у нагоді користувачам, які шукають комплексний та гнучкий інструмент для формування власної конфігурації ПК.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Gaming PC Builder – це інноваційний мобільний застосунок, розроблений спеціально для геймерів, які прагнуть створити високопродуктивні комп'ютерні збірки для сучасних ігор. Застосунок допомагає підібрати сумісні компоненти з урахуванням високих вимог до продуктивності, бюджету та естетики, дозволяючи легко створювати оптимальні конфігурації для ігрових ПК. Основні можливості застосунку Gaming PC Builder:

– Попередньо налаштовані ігрові збірки. Застосунок містить готові конфігурації, розроблені з урахуванням вимог сучасних ігор, що дозволяє швидко обрати оптимальний варіант під ваш бюджет та потреби.

– Детальний опис та порівняння компонентів. Кожен елемент обладнання супроводжується докладним описом, характеристиками, відгуками користувачів та можливістю порівняння з іншими моделями.

– Інтерактивна візуалізація. За допомогою 3D-моделей та схем ви зможете побачити розташування компонентів у збірці, що допомагає оцінити естетичний вигляд та функціональність.

– Інтеграція з онлайн-магазинами. Застосунок дозволяє перевірити актуальні ціни та наявність компонентів у популярних інтернет-магазинах, що полегшує прийняття рішень щодо покупки.

– Персоналізовані рекомендації. На основі заданих параметрів та бюджету застосунок пропонує оптимальні варіанти комплектуючих, що дозволяють створити ігровий ПК, максимально адаптований до ваших потреб.

З Gaming PC Builder ви отримуєте потужний інструмент для створення ідеального ігрового комп'ютера

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Переглянути застосунок можна на рисунку 1.3.

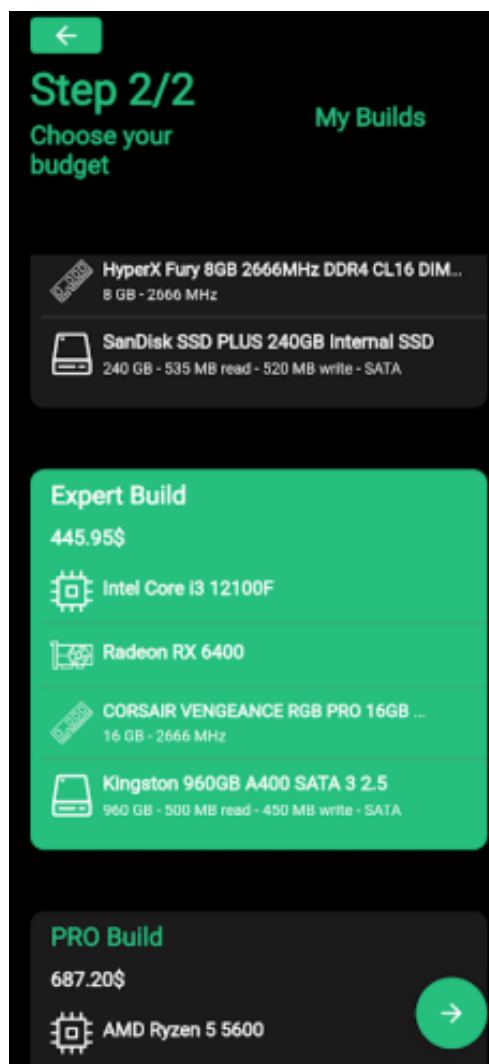


Рисунок 1.3 – Застосунок Gaming PC Builder

Він дозволяє зосередитися на улюблених іграх, не витрачаючи зайвого часу на пошук сумісних компонентів та прийняття складних технічних рішень.

Для того, щоб провести всебічне порівняння існуючих мобільних застосунків для підбору комп'ютерних комплектуючих, необхідно розглянути низку ключових аспектів, що впливають на користувацький досвід та функціональність програмних рішень. Серед основних критеріїв оцінки варто виділити:

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

– Інтерфейс користувача. Оцінюється зручність, інтуїтивність та естетичність графічного інтерфейсу. Важливо, щоб дизайн також був адаптивним для різних пристроїв.

– Автоматизація процесів. Здатність застосунку автоматично підбирати сумісні комплектуючі з урахуванням технічних характеристик, бюджету та індивідуальних вимог користувача. Чим більше завдань автоматизовано, тим менша потреба у ручному втручанні.

– Фільтрація даних. Наявність можливості застосовувати розширені фільтри для відбору релевантної інформації за різними критеріями, такими як бренд, технічні характеристики, ціна, сумісність та інші параметри, що дозволяє користувачеві швидко знаходити потрібні компоненти.

– Підтримка протоколів. Оцінюється спектр мережевих та інших протоколів, які застосунок здатен розпізнавати та обробляти. Це може включати інтеграцію з API виробників, онлайн-магазинами та базами даних, що забезпечує актуальність інформації про комплектуючі.

– Зручність використання. Простота освоєння програми та її інтуїтивність для кінцевого користувача незалежно від рівня його технічної підготовки. Важливо, щоб навіть користувач без спеціальних знань міг легко орієнтуватися у функціях застосунку.

– Споживання ресурсів. Ефективність використання обчислювальних ресурсів пристрою, таких як оперативна пам'ять, процесорний час та мережевий трафік. Оптимізація цих показників забезпечує стабільну роботу застосунку навіть на пристроях із середніми характеристиками.

– Спеціалізовані функції. Наявність унікальних інструментів, які відрізняють програму від інших рішень на ринку. До таких функцій можна віднести автоматичну перевірку сумісності компонентів, інтерактивну візуалізацію збірок, персоналізовані рекомендації та можливості порівняння різних конфігурацій.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

– Кросплатформеність. Здатність застосунку працювати на різних операційних системах (Android та iOS) без втрати функціональності та зручності користування.

Нижче представлено порівняльну таблицю мобільних застосунків для підбору комп'ютерних комплектуючих (таб. 1.1). Вона містить оцінку основних критеріїв, які впливають на функціональність, зручність використання та можливості кожного застосунку.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Таблиця 1.1 – Порівняльна характеристика мобільних застосунків

Критерії	PC Creator 2	PC Builder	Gaming PC Builder
Інтерфейс	Інтуїтивний сучасний дизайн	Деталізований орієнтований на технічний аналіз	Акцент на ігрових конфігураціях
Фільтрація даних	Є, базові фільтри	Розширені можливості налаштування	Фільтрація під ігри
Підтримка протоколів	Інтеграція з онлайн - магазинами	API з виробниками	Мінімальна підтримка зовнішніх сервісів
Зручність використання	Легка для початківців	Для досвідчених користувачів	Орієнтована на геймерів
Споживання ресурсів	Оптимізована для середніх пристроїв	Середнє навантаження	Підтримка слабких пристроїв
Спеціалізовані функції	Автоматична перевірка сумісності	Порівняння характеристик	3D-візуалізація збірок
Кросплатформеність	Android	Android, iOS	Android

Ця порівняльна таблиця надає можливість оцінити переваги та недоліки кожного застосунка для вибору комп'ютерних комплектуючих, що дозволяє працювати над подальшим покращенням їхньої функціональності.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

1.3 Аналіз вимог до програмного засобу та постановка завдання

На основі аналізу існуючих рішень було визначено функціональні та нефункціональні вимоги до майбутнього мобільного застосунку для підбору комп'ютерних комплектуючих. Функціональні вимоги, які представлені нижче:

- Реєстрація та авторизація користувачів.
- Перегляд доступних комплектуючих. Наприклад користувач може ознайомитися з детальною інформацією про процесори, назва, частота, кількість ядер, тип сокету, а також орієнтовна ціна в гривнях.
- Підбір та збереження конфігурацій ПК.
- Сортування компонентів.
- База даних користувачів. Реалізована таблиця користувачів та таблиця конфігурацій, яка зберігає обрані компоненти для кожного зареєстрованого користувача.

Нижче вказано нефункціональні вимоги для мобільного застосунку:

- Зручність та інтуїтивність інтерфейсу.
- Висока продуктивність та оптимізація ресурсів.
- Адаптивність на різних пристроях.
- Можливість оновлення та розширення функціоналу.
- Підтримка та безпека даних.

Основними завданнями для розробки є створення мобільного застосунку для підбору та налаштування конфігурацій ПК, що дозволить користувачам зручно обирати компоненти, порівнювати різні конфігурації та зберігати свої налаштування.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

2 ПРОЄКТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

2.1 Проєктування структурної схеми мобільного застосунку

Створення мобільного застосунку для підбору комп'ютерних комплектуючих передбачає кілька основних етапів, які впливатимуть на його ефективність та успішність у кінцевому підсумку.

З метою забезпечення впізнаваності мобільного застосунку обрано назву «Build PC» що відображає його основне функціональне призначення – підбір та складання персонального комп'ютера.

За результатами аналізу предметної області спроектовано схему взаємодії користувачів з системою (рис. 2.1).

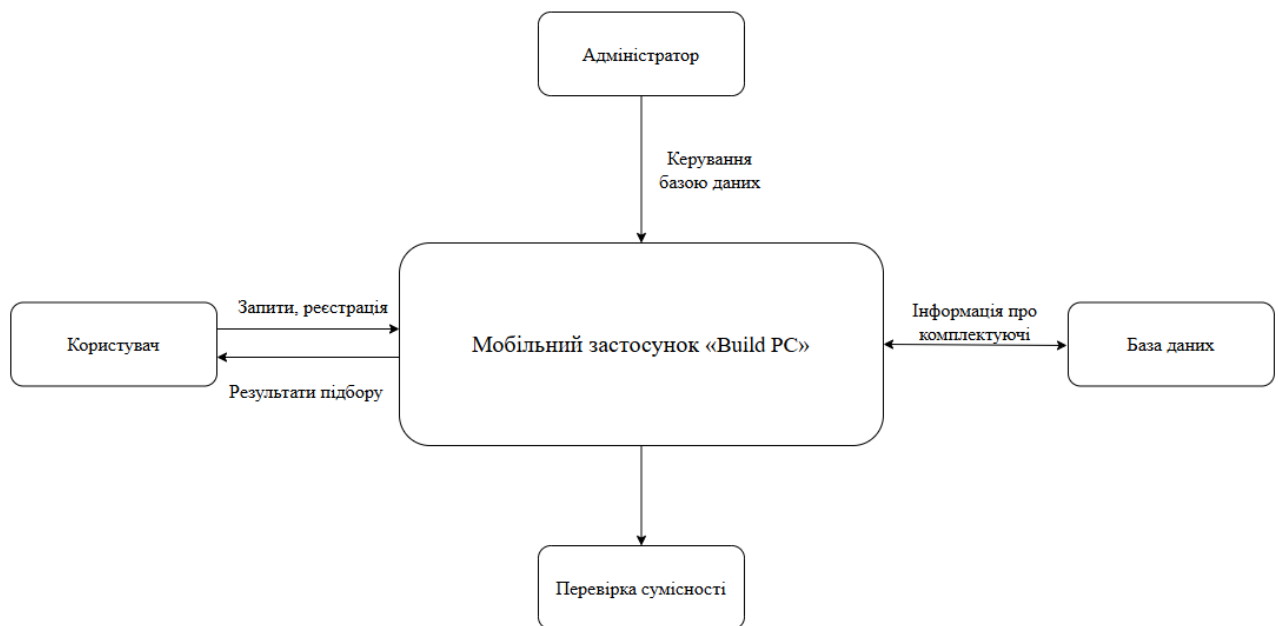


Рисунок 2.1 – Схема взаємодії користувачів з застосунком

На схемі зображено взаємодію мобільного застосунку «Build PC» із зовнішніми суб'єктами та компонентами системи. Користувач надсилає запити, проходить реєстрацію, а також отримує результати підбору комплектуючих.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Далі була побудована діаграма варіантів використання, яка відображає функціональні можливості, що доступні користувачу в рамках роботи із системою, що показує логічну структуру дій, які він може виконувати. Діаграму варіантів використання зображено на рисунку 2.2.

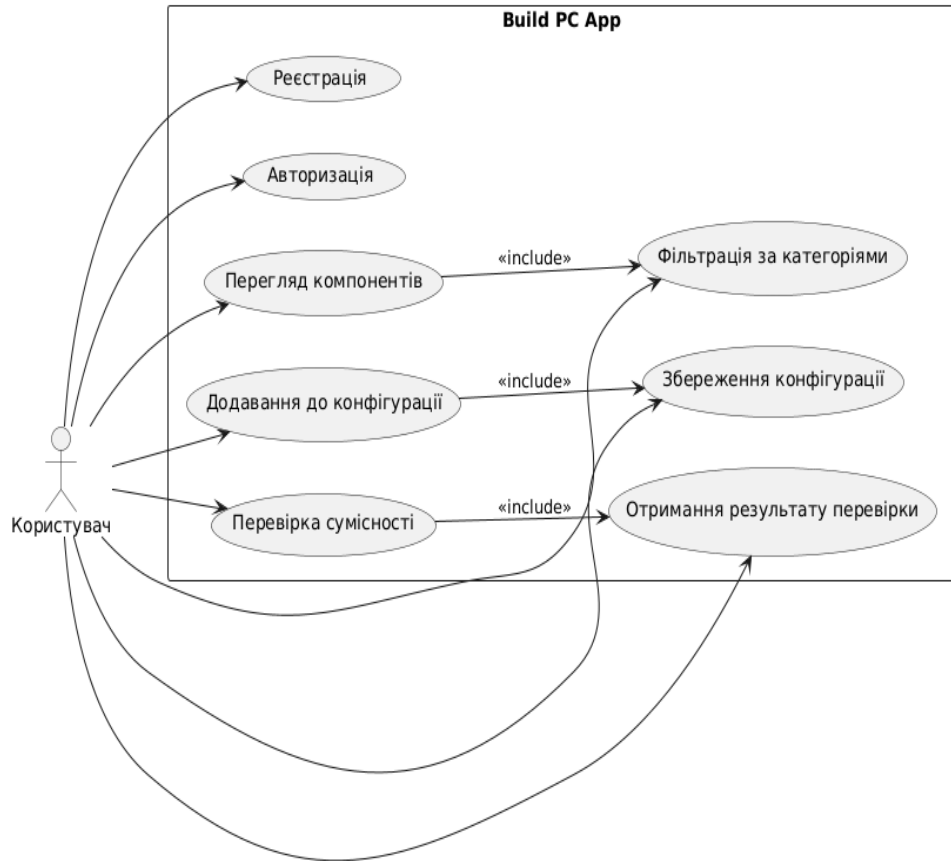


Рисунок 2.2 – Діаграма варіантів використання

В діаграмі подано зв'язки типу «include», що свідчить про залежність певних варіантів використання від інших (наприклад, перегляд компонентів включає фільтрацію; перевірка сумісності включає отримання результатів тощо).

Ця діаграма є важливою для розуміння логіки роботи системи з точки зору кінцевого користувача та дозволяє оцінити повноту реалізованого функціоналу.

2.2 Проєктування бази даних мобільного застосунку

У результаті проєктування бази даних для мобільного застосунку «Build PC» було визначено ключові інформаційні об'єкти, їх атрибути та логічні зв'язки між ними. Побудована структура забезпечує ефективне зберігання даних про комплектуючі, користувачів, конфігурації та інші сутності, необхідні для реалізації функціоналу застосунку з інтерфейсною частиною. Таке проєктне рішення створює надійну основу для подальшої розробки, тестування та використання застосунку.

База даних мобільного застосунку включає таблицю `users`, яка зберігає унікальний ідентифікатор кожного користувача, його електронну пошту та хешований пароль. Для фіксації вибраної конфігурації створено таблицю `user_config`, що містить зовнішні ключі на всі основні компоненти, обрані користувачем: процесор, відеокарту, материнську плату, оперативну пам'ять, накопичувач, блок живлення, корпус і систему охолодження.

Окремі таблиці бази даних відповідають за зберігання інформації про кожен тип комплектуючих. Зокрема, таблиця `cpu` містить дані про процесори – назву, частоту, кількість ядер, сокет та ціну. Таблиця `gpu` містить відомості про відеокарти: обсяг пам'яті, кількість ядер, інтерфейс підключення та вартість.

Крім того, реалізовано таблиці для інших апаратних компонентів: `motherboards` – материнські плати (чипсет, сокет, форм-фактор, підтримка пам'яті), `ram` – оперативна пам'ять (тип, розмір, частота, напруга), `storages` – накопичувачі (тип, обсяг, інтерфейс, форм-фактор), `psus` – блоки живлення (потужність, форм-фактор, модульність, ефективність), `cases` – корпуси (форм-фактор, підтримка материнських плат, довжина відеокарти, підтримка вентиляторів), `coolings` – системи охолодження (тип, розмір вентилятора, сумісність із сокетом, рівень шуму, ціна).

Також реалізовано таблицю `accessories`, яка містить додаткові компоненти, що можуть бути додані до конфігурації. Вона включає назву аксесуара, характеристики та вартість. Для зв'язку між користувачами та

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

аксесуарами використовується таблиця user_accessories, яка зберігає відповідність між конкретним користувачем та вибраними аксесуарами.

Структура бази даних забезпечує зручну взаємодію з усіма компонентами системи, дозволяє швидко отримувати доступ до інформації та ефективно використовувати її під час формування персоналізованої комп'ютерної збірки. Візуальне представлення структури наведено у таблиці 2.1.

Таблиця 2.1 – Аналіз інформаційних об'єктів та їх атрибутів сутності

Інформаційний об'єкт	Атрибут сутності	Ідентифікатор	Ключ
users	id	id	P-key
	email	email	–
	password	password	–
cpus	id	id	P-key
	name	name	–
	frequency	frequency	–
	cores	cores	–
	socket	socket	–
	price	price	–
gpus	id	id	P-key
	name	name	–
	memory	memory	–
	cores	cores	–
	interface	interface	–
	price	price	–
motherboards	id	id	P-key
	name	name	–
	chipset	chipset	–
	socket	socket	–

Продовження таблиці 2.1

	form_factor	form_factor	–
	ram_type	ram_type	–
	price	price	–
user_config	id	id	P-key
	user_id	user_id	F-key
	cpu_id	cpu_id	F-key
	gpu_id	gpu_id	F-key
	motherboard_id	motherboard_id	F-key
	ram_id	ram_id	F-key
	storage_id	storage_id	F-key
	psu_id	psu_id	F-key
	case_id	case_id	F-key
	cooling_id	cooling_id	F-key
accessories	id	id	P-key
	name	name	–
	type	type	–
	feature1	feature1	–
	feature2	feature2	–
	feature3	feature3	–
	price	price	–
user_accessories	id	id	P-key
	user_id	user_id	F-key
	accessory_id	accessory_id	F-key
ram	id	id	P-key
	name	name	–
	size	size	–
	type	type	–
	speed	speed	–

Продовження таблиці 2.1

	voltage	voltage	—
	price	price	—
storages	id	id	P-key
	name	name	—
	type	type	—
	capacity	capacity	—
	interface	interface	—
	form_factor	form_factor	—
	price	price	—
psus	id	id	P-key
	name	name	—
	wattage	wattage	—
	form_factor	form_factor	—
	efficiency	efficiency	—
	modular	modular	—
	price	price	—
cases	id	id	P-key
	name	name	—
	form_factor	form_factor	—
	mb_support	mb_support	—
	gpu_length	gpu_length	—
	fan_support	fan_support	—
	price	price	—
coolings	id	id	P-key
	name	name	—
	type	type	—
	fan_size	fan_size	—
	socket_compatibility	socket_compatibilit	—

2.3 Функціональна структура мобільного застосунку

Функціональна структура мобільного застосунку включає кілька основних модулів, кожен з яких виконує специфічні завдання, що забезпечують повний цикл підбору комп'ютерних комплектуючих.

Інтерфейс користувача – це візуальна частина застосунку, яка дозволяє зручно взаємодіяти з системою, обирати компоненти, переглядати характеристики та формувати власні збірки.

Модуль авторизації та реєстрації відповідає за створення облікових записів, вхід до системи та збереження індивідуальних налаштувань користувача.

Модуль перевірки сумісності аналізує відібрані компоненти та повідомляє користувача про можливі конфлікти у збірці.

Важливою частиною системи є база даних комплектуючих, яка зберігає інформацію про всі доступні товари, яка постійно оновлюється адміністратором через спеціальний модуль керування.

Усі модулі взаємодіють між собою, формуючи цілісну, зручну та функціональну систему для підбору ПК.

Деталізація роботи системи на рівні внутрішніх процесів демонструє структурну схему застосунку, що зображена на рисунку 2.3.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

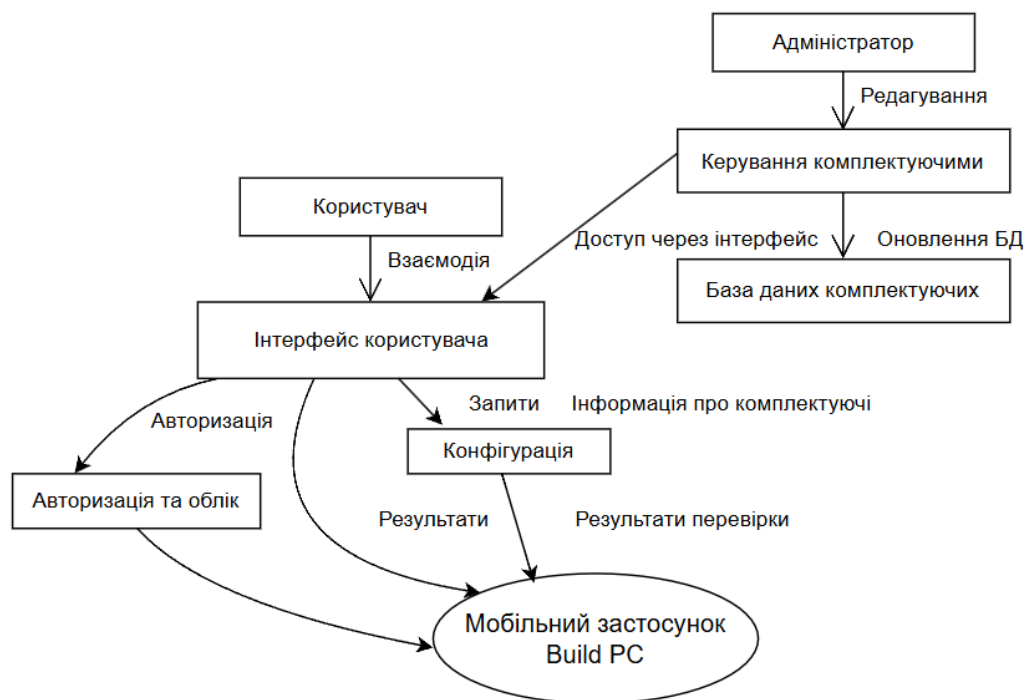


Рисунок 2.3 – Функціональна структура застосунку

2.4 Проектування інтерфейсу мобільного застосунку

Проектування інтерфейсу користувача (UI) є одним із ключових етапів розробки мобільного застосунку, оскільки саме через нього відбувається основна взаємодія користувача з програмною системою. У застосунку «Build PC» інтерфейс виконує не лише візуальну функцію, але й забезпечує логіку керування, швидкий доступ до інструментів підбору комплектуючих, та загалом – комфортну роботу із застосунком.

Перед створенням макетів інтерфейсу було визначено цільову аудиторію, її потреби, очікування та технічний рівень підготовки. Основними користувачами є молодь, студенти, ентузіасти ПК, а також користувачі, які не мають поглиблених технічних знань, але хочуть самостійно зібрати персональний комп'ютер. Це зумовлює необхідність створення інтуїтивного, зрозумілого та візуально збалансованого інтерфейсу.

Одним з головних завдань на етапі UI-проектування було забезпечення зручного переходу між ключовими етапами використання застосунку – авторизація, вибір категорії, перегляд характеристик, перевірка сумісності, збереження конфігурації. Інтерфейс мав бути лаконічним, легким для сприйняття, та водночас функціонально наповненим. Було вирішено використовувати сучасні підходи до UI/UX, враховуючи принципи мінімалізму, доступності та адаптивності.

На етапі концептуального планування також були визначені ключові блоки взаємодії користувача: стартова навігація, екран вибору компонентів, модуль перевірки сумісності, конфігураційний блок та загальна система повідомлень. Кожен із цих елементів матиме відповідне візуальне втілення згідно з логікою роботи застосунку.

Проектування інтерфейсу стало результатом послідовного аналізу функціональних вимог, структури бази даних та можливих сценаріїв взаємодії з системою. Створення макетів планується здійснити з використанням інструментів прототипування, що дозволить ефективно перевірити зручність та логіку інтерфейсу до початку його реалізації в Android Studio.

Макети мобільного застосунку були створені за допомогою онлайн-інструменту Figma, який широко використовується для UI/UX-дизайну.

Перший макет інтерфейсу мобільного застосунку зображений на рисунку 2.4. Він був створений у Figma у стилі мінімалізму з акцентом на зручність користування. На екрані розміщені два поля вводу: для електронної пошти та пароля. Нижче розташовані дві кнопки – «Увійти» та «Зареєструватися», які забезпечують відповідний перехід у застосунку. Такий дизайн дозволяє користувачу швидко пройти процес входу або створити обліковий запис без зайвих елементів та у зрозумілому інтерфейсі.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

В результаті проєктування мобільного застосунку було створено схему взаємодії користувачів із застосунком, структурну схему системи, а також діаграму варіантів використання, які дозволили детально візуалізувати логіку функціонування мобільного застосунку «Build PC». Ці графічні моделі дають змогу визначити основні ролі користувачів, функціональні можливості застосунку та архітектурну побудову системи.

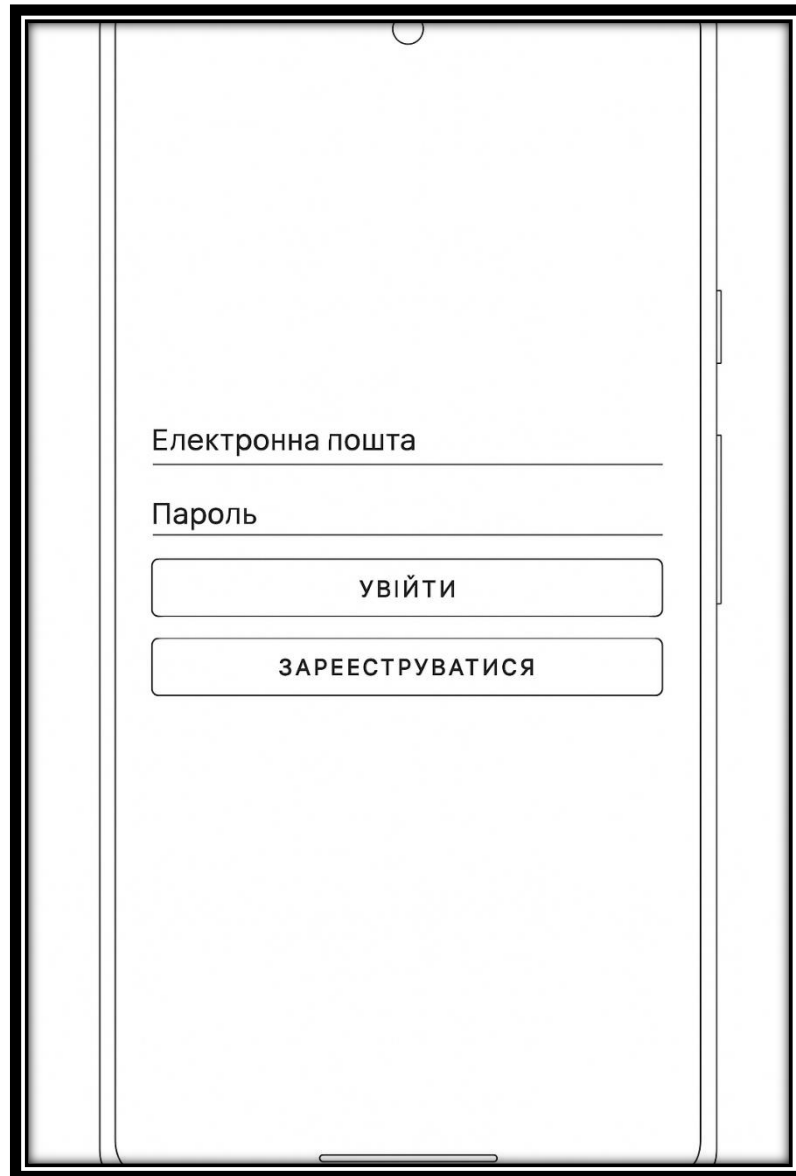


Рисунок 2.4 – Макет авторизації та реєстрації

Далі було спроектовано головну сторінку мобільного застосунку як центральний елемент навігації, з якого користувач розпочинає взаємодію з системою. У верхній частині екрана розміщено логотип із назвою Build PC, що

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

підкреслює бренд і призначення застосунку. Нижче розташовані основні функціональні кнопки: «Почати збірку», «Моя конфігурація», «Перевірити сумісність», «Налаштування» та «Вихід». Кожна кнопка веде до відповідного розділу застосунку, що дозволяє швидко орієнтуватися та переходити до потрібних дій. Дизайн головної сторінки є лаконічним, інтуїтивно зрозумілим і зручним як для новачків, так і для досвідчених користувачів. Макет зображено на рисунку 2.5.

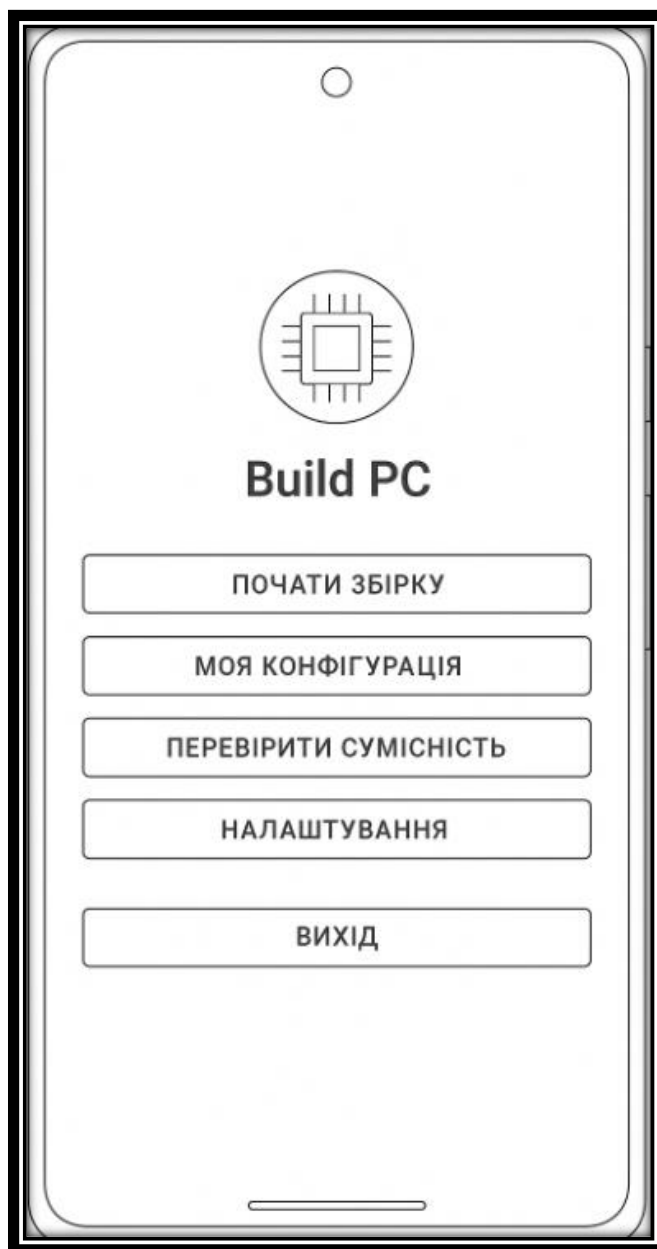


Рисунок 2.5 – Макет головної сторінки

Після вибору певної категорії на екрані «Обери компонент», користувач переходить до наступного етапу – перегляду списку доступних

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

комплектуючих. Цей макет відображає перелік товарів певного типу (наприклад, процесори), які відповідають обраній категорії. Його зображено на рисунку 2.6

Інтерфейс реалізовано у вигляді прокручуваного списку карток, кожна з яких містить назву моделі, основні технічні характеристики (наприклад, частота, кількість ядер, тип сокету), зображення компонента, а також кнопку «Обрати»

Такий підхід дозволяє користувачу швидко ознайомитися з характеристиками, порівняти різні моделі та зробити оптимальний вибір. Макет зберігає єдиний стиль застосунку – лаконічний дизайн, зручна навігація та акцент на функціональність.

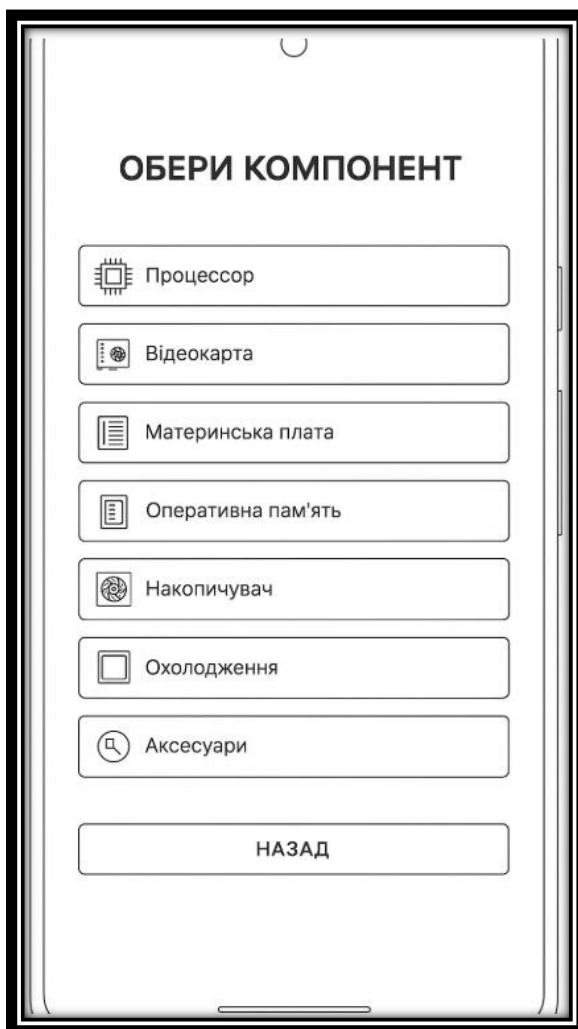


Рисунок 2.6 – Макет переліку компонентів

В результаті проектування мобільного застосунку було створено схему взаємодії користувачів із застосунком, структурну схему системи, а також діаграму варіантів використання, які дозволили детально візуалізувати логіку функціонування мобільного застосунку «Build PC». Ці графічні моделі дають змогу визначити основні ролі користувачів, функціональні можливості застосунку та архітектурну побудову системи.

Отримані результати стали надійною основою для реалізації застосунку в середовищі Android Studio та переходу до технічної розробки і тестування функціональності.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

3 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Розробка інтерфейсу застосунку

Інтерфейс користувача є критично важливою складовою будь-якого мобільного застосунку, адже саме він формує перше враження та забезпечує зручність взаємодії користувача із системою. На цьому етапі розробки особлива увага буде приділена простоті, логічності та доступності елементів інтерфейсу, що дозволяє зробити застосунок інтуїтивно зрозумілим навіть для недосвідчених користувачів.

Після завершення етапу проектування графічного інтерфейсу та створення інтерактивних прототипів у середовищі Figma, було розпочато реалізацію інтерфейсу безпосередньо в Android Studio. Для розмітки графічних елементів застосовано мову XML, що дозволяє задавати структуру, стилі та поведінку елементів інтерфейсу. Для реалізації логіки переходів між екранами, обробки натискань кнопок, валідації введених даних та зв'язку з базою – використано мову програмування Java.

Одним із перших реалізованих екранів став екран авторизації, який забезпечує ідентифікацію користувача в системі та слугує точкою входу до функціоналу застосунку. Даний екран містить два текстові поля для введення електронної пошти та пароля, а також дві кнопки – «Увійти» та «Зареєструватися».

Функціонал кнопки «Увійти» передбачає перевірку введених даних: у разі правильного введення електронної пошти та пароля, користувач авторизується та переходить до головного екрана застосунку. У випадку помилки – виводиться повідомлення з відповідним поясненням. Якщо користувач ще не має акаунту, натискання кнопки «Зареєструватися» відкриває форму реєстрації, яка додатково перевіряє унікальність електронної пошти у базі даних.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

Інтерфейс реалізовано відповідно до принципів Material Design з урахуванням темної колірної схеми. Це підвищує комфорт користування у темний час доби та відповідає сучасним естетичним стандартам.

Сторінки зображено на рисунках 3.1 та 3.2.

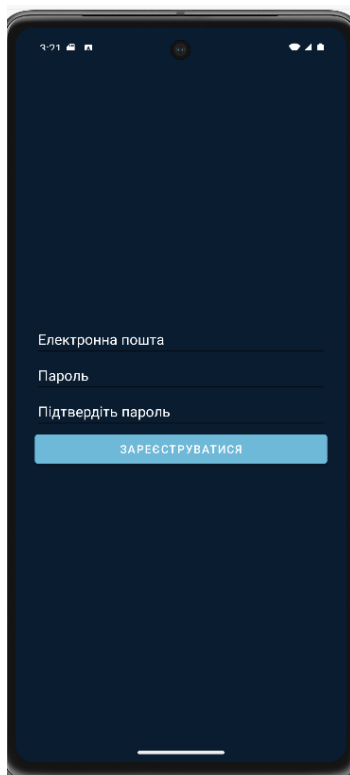


Рисунок 3.1 – Сторінка реєстрації

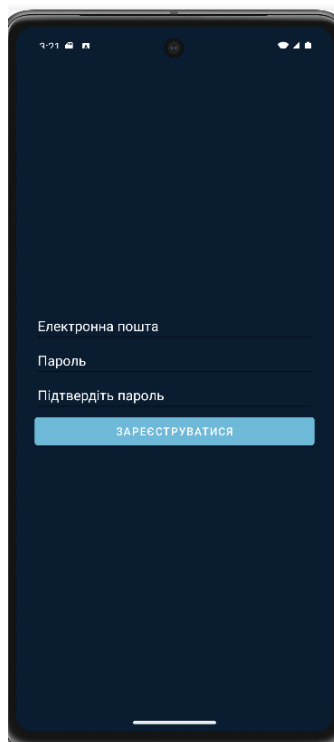


Рисунок 3.2 – Сторінка для входу

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Збереження інформації про поточного користувача здійснюється через окремий механізм сесій, що дозволяє надалі взаємодіяти із персональними конфігураціями в системі.

Далі було розроблено головну сторінку мобільного застосунку Build PC. Вона є ключовим елементом взаємодії користувача з системою. Саме з неї починається основний робочий процес, тому її реалізації було приділено особливу увагу як з точки зору дизайну, так і функціональності. Основна мета головної сторінки – забезпечити швидкий і зручний доступ до всіх основних функцій застосунку, водночас створивши позитивне перше враження та комфортну навігацію.

Інтерфейс головної сторінки побудований за принципами мінімалізму та зручності. У верхній частині екрана розміщено заголовок із назвою застосунку Build PC, який виконує роль логотипа та орієнтира для користувача. У її центрі розміщено основні функціональні кнопки: «Почати збірку», «Моя конфігурація», «Перевірити сумісність» та «Налаштування».

Також було реалізовано підтримку адаптивного інтерфейсу. Розмітка головної сторінки автоматично підлаштовується під розміри екрана пристрою, забезпечуючи зручність користування на смартфонах з різною діагоналлю. Всі елементи розміщені з урахуванням правил просторової гармонії – достатні відступи, зрозуміла ієрархія елементів та зручна зона натискання.

Крім функціональних переваг, головна сторінка виконує також естетичну роль. Її візуальна простота дозволяє уникнути інформаційного перевантаження, що є важливим аспектом для користувачів без технічного досвіду. Водночас логічна структура дає змогу досвідченим користувачам швидко переходити до потрібного розділу без зайвих дій.

Таким чином, головна сторінка у застосунку Build PC є повноцінним навігаційним центром, який поєднує зручність, простоту, візуальну привабливість та високу функціональність. Її реалізація відповідає всім вимогам до сучасного мобільного UI-дизайну та закладає основу для ефективного використання застосунку в цілому.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

Сторінка зображена на рисунку 3.3

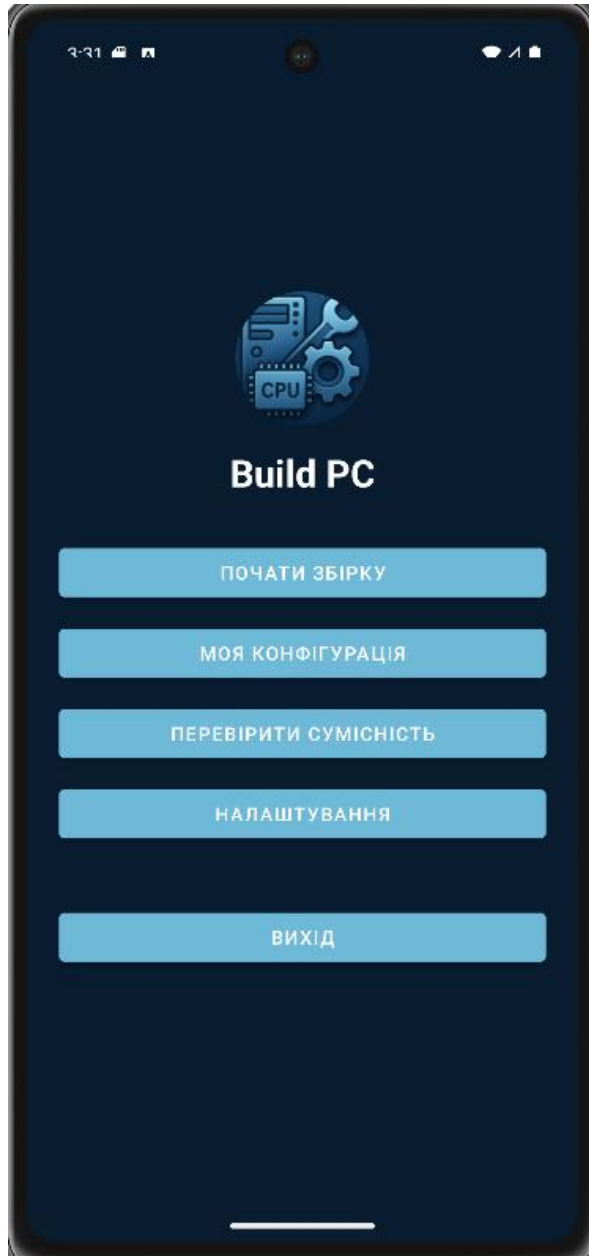


Рисунок 3.3 – Головна сторінка застосунку

Далі Екран «Почати збірку» відкривається після натискання відповідної кнопки на головній сторінці й слугує відправною точкою для формування персонального складу комплектуючих комп'ютера.

Його головне призначення – забезпечити покроковий вибір усіх основних компонентів, з яких складатиметься майбутня конфігурація.

Інтерфейс реалізовано у вигляді вертикального списку кнопок із назвами категорій комплектуючих, таких як: процесор, відеокарта, материнська плата,

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

оперативна пам'ять, накопичувач, блок живлення, корпус, система охолодження та аксесуари. Кожен елемент списку містить іконку та текстову назву компонента, що значно полегшує візуальне сприйняття та орієнтацію в інтерфейсі.

Після вибору відповідної категорії, користувач переходить до наступного екрану з переліком компонентів цієї групи, де може ознайомитися з характеристиками та цінами та зробити вибір. У нижній частині екрана розміщена кнопка «Назад», яка дозволяє повернутися на головну сторінку без втрати поточних дій.

Візуально екран витримано у темній кольоровій гамі із контрастними блакитними кнопками, що відповідає загальному дизайну застосунку. Уся структура побудована логічно, щоб користувач міг легко й швидко перейти до вибору потрібного компонента. Сторінку зображено на рисунку 3.4.

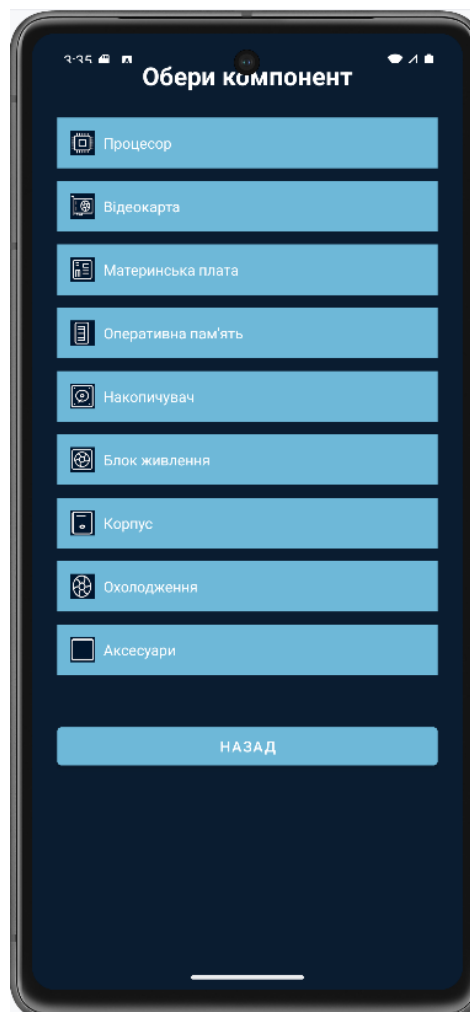


Рисунок 3.4 – Сторінка «Почати збірку»

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

Екран «Вибір процесора» є одним із найважливіших функціональних етапів у процесі складання комп'ютера, оскільки саме процесор визначає базовий рівень продуктивності системи. Даний інтерфейс відкривається після вибору категорії «Процесор» на екрані «Почати збірку» та відображає перелік доступних моделей CPU, які користувач може обрати для своєї конфігурації.

Інтерфейс реалізовано у вигляді вертикального списку карток. Кожна картка містить назву моделі процесора, тактову частоту, максимальну частоту в режимі Boost, кількість ядер і потоків, тип сокету, а також вартість компонента у гривнях. Для зручності сприйняття, кожна позиція супроводжується піктограмою процесора, що сприяє швидкому візуальному орієнтуванню.

Праворуч від кожної картки розміщено кнопку із символом «+», яка дозволяє додати обраний процесор до поточної конфігурації. Після додавання вибраного компонента система зберігає вибір у базу даних і повертає користувача до попереднього етапу складання. Сторінку зображено на рисунку 3.5.

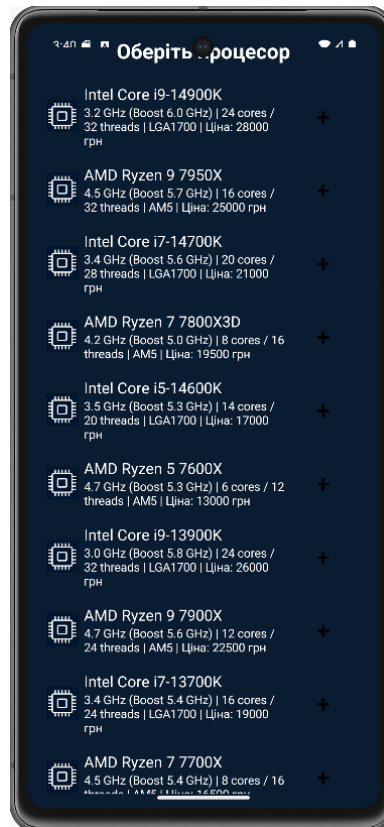


Рисунок 3.5 – Сторінка процесорів

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Екран «Вибір відеокарти» є важливим компонентом інтерфейсу мобільного застосунку Build PC, оскільки відеокарта (GPU) відіграє ключову роль у загальній продуктивності системи, особливо для ігор, графічного дизайну, 3D-моделювання та відеомонтажу. Цей екран відкривається після вибору категорії «Відеокарта» на попередньому етапі складання ПК.

Інтерфейс реалізовано у вигляді прокручуваного списку, де кожна позиція представлена у формі інформативної картки. У кожній картці виводиться назва моделі відеокарти, обсяг відеопам'яті, тип пам'яті (наприклад, GDDR6X), кількість обчислювальних ядер, інтерфейс підключення (наприклад, PCIe 4.0), а також актуальна ціна в гривнях. Це представлено на рисунку 3.6.



Рисунок 3.6 – Сторінка відеокарт

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Екран «Вибір материнської плати» є одним з ключових у процесі складання ПК, адже саме плата визначає сумісність з іншими компонентами. Інтерфейс представлено у вигляді зручного списку, де для кожного елемента вказано модель, чіпсет, сокет, форм-фактор, тип оперативної пам'яті та актуальну ціну. Завдяки зрозумілому оформленню та кнопці вибору можна швидко знайти і додати до конфігурації відповідну материнську плату. Зображення представлено на рисунку 3.7

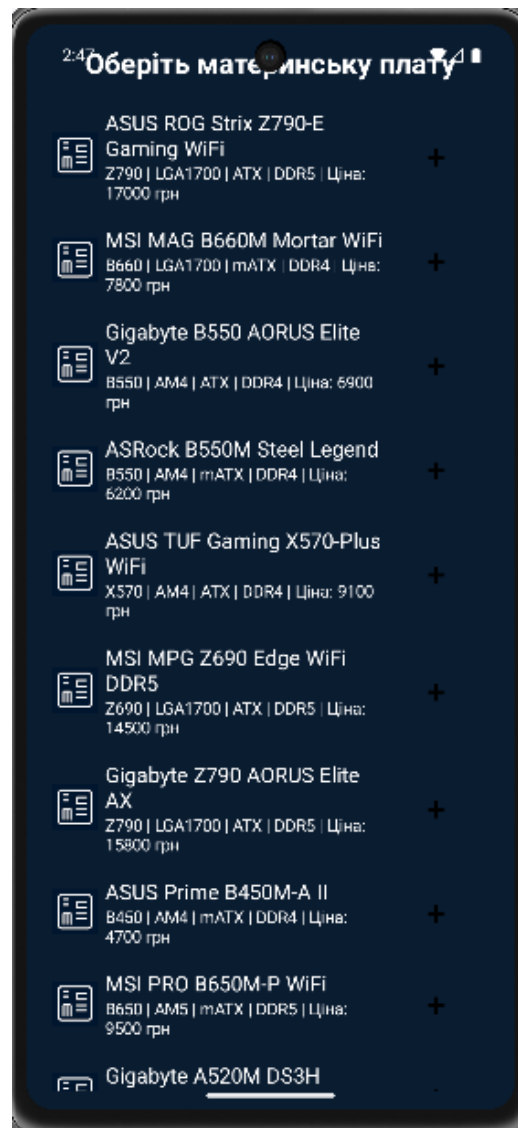


Рисунок 3.7 – Сторінка материнських плат

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

Далі сторінка «Оберіть оперативну пам'ять» дозволяє підібрати модулі RAM відповідно до вимог системи та бюджету. У вигляді прокручуваного списку відображаються назва моделі, обсяг пам'яті, тип (наприклад, DDR4 або DDR5), тактова частота, напруга живлення та вартість. Це представлено на рисунку 3.8



Рисунок 3.8 – Сторінка оперативної пам'яті

Екран вибору накопичувача дозволяє додати до збірки SSD відповідного типу, обсягу та формату підключення. У списку відображається назва моделі, ємність, інтерфейс, форм-фактор (M.2 або 2.5") та вартість у гривнях. Інформація подається стисло та зручно для порівняння, а кнопка з правого боку дає змогу швидко додати обраний варіант до конфігурації.



Рисунок 3.9 – Сторінка накопичувачів

Нижче на рисунку 3.10 представлено блоки живлення, з-поміж яких можна обрати оптимальний варіант для своєї збірки. Кожна позиція містить назву моделі, потужність у ватах, форм-фактор (ATX), сертифікацію ефективності (наприклад, 80+ Gold або Bronze), тип, а також вартість.



Рисунок 3.10 – Сторінка блоків живлення

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

На рисунку 3.11 користувач можна вибрати комп'ютерний корпус. У кожному блоці вказано тип (наприклад, Mid Tower), підтримувані формати (ATX, Micro-ATX, Mini-ITX), максимальна довжина відеокарти, кількість і розташування вентиляторів, а також ціна. Завдяки такому поданню користувач може легко підібрати корпус, який відповідатиме як технічним вимогам, так і бюджету.

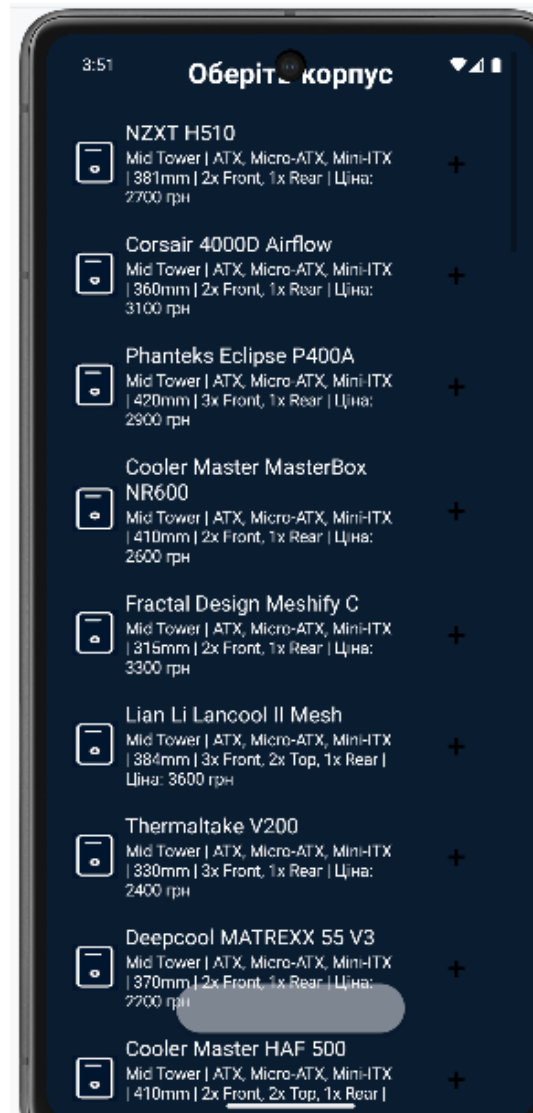


Рисунок 3.11 – Сторінка корпусів

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Також показано системи охолодження, серед яких доступне повітряне або рідинне рішення відповідно до платформи та потреб у тепловідведенні. Для кожної моделі вказано тип охолодження, розміри вентилятора або радіатора, сумісні сокети (наприклад, AM4, LGA1700), рівень шуму в децибелах і вартість. Це дозволяє легко порівняти параметри та підібрати оптимальне рішення для стабільної роботи процесора. Сторінку зображено на рисунку 3.12



Рисунок 3.12 – Сторінка охолодження

На екрані аксесуарів представлено додаткові пристрої, які можуть стати частиною ПК: клавіатури, мишки, монітори. Дані згруповані за категоріями для зручності перегляду. Для кожного аксесуара вказано тип, технічні характеристики (наприклад, DPI, розмір екрана), а також вартість. Інтерфейс дозволяє швидко додати потрібний аксесуар до конфігурації за допомогою кнопки з правого боку. Це зображено на рисунку 3.13.

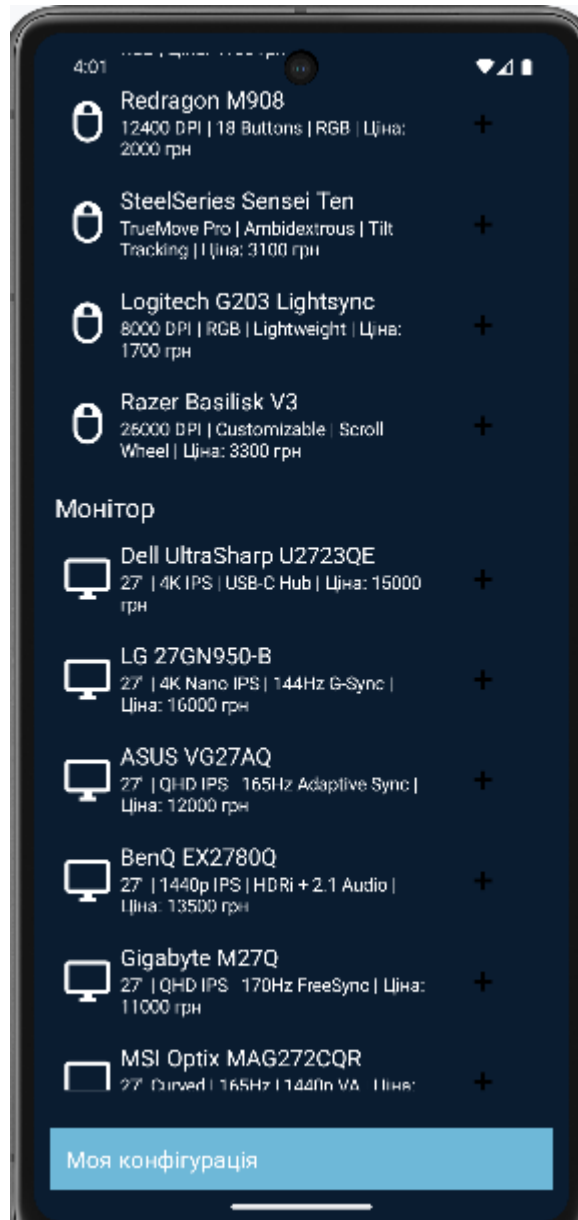


Рисунок 3.13 – Сторінка аксесуарів

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

3.2 Розробка функціоналу застосунку

У застосунку реалізовано механізм хешування паролів за допомогою алгоритму SHA-256, що дозволяє зберігати лише криптографічні хеші замість відкритого тексту. Це суттєво знижує ризик компрометації паролів у разі несанкціонованого доступу до бази даних. Навіть у випадку отримання доступу до збережених даних, розшифрування пароля стає практично неможливим, оскільки SHA-256 є одностороннім хеш-функціоналом.

Розробка функціональної частини мобільного застосунку Build PC охоплює реалізацію основних можливостей, які забезпечують повноцінну роботу системи та взаємодію користувача з даними. Функціонал застосунку реалізовано за допомогою мови програмування Java та інтегрованої бази даних SQLite, що дає змогу зберігати, обробляти та відображати необхідну інформацію без постійного підключення до Інтернету.

Клас DatabaseHelper.java відповідає за створення бази даних, таблиць та основні операції з ними – реєстрація користувача, перевірка пароля, отримання ID користувача, додавання конфігурацій тощо.

Фрагмент коду представлено в лістингу 3.1.

Лістинг 3.1 – Ініціалізація локальної бази даних SQLite

```
package com.example.buildpc;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "BuildPC.db";
    private static final int DATABASE_VERSION = 1;

    public static final String TABLE_USERS = "users";
    public static final String COL_ID = "id";
    public static final String COL_EMAIL = "email";
    public static final String COL_PASSWORD = "password";

    public DatabaseHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}
```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

Далі реалізовано виведення списку процесорів з таблиці `cpus`. Після відкриття екрана виконується SQL-запит до локальної бази даних, а результати зберігаються у курсор для подальшого відображення.

Фрагмент коду представлено в лістингу 3.2.

Лістинг 3.2 – Ініціалізація `CPUActivity` та завантаження даних

```
public class CPUActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cpu);
        Toast.makeText(this, "Екран: Вибір процесора",
            Toast.LENGTH_SHORT).show();
        LinearLayout cpuListContainer =
            findViewById(R.id.cpuListContainer);
        DatabaseHelper dbHelper = new DatabaseHelper(this);
        SQLiteDatabase db = dbHelper.getReadableDatabase();
```

Інформація про відеокарти зчитується за допомогою SQL-запиту, а далі – виводиться у вигляді списку на екрані. Отримання даних виконується через `DatabaseHelper`, який забезпечує підключення до бази даних.

Фрагмент коду представлено в лістингу 3.3.

Лістинг 3.3 – Ініціалізація `GPUActivity` та завантаження даних про відеокарти з `SQLite`

```
public class GPUActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gpu); // або інша
        відповідна розмітка з контейнером gpuListContainer

        Toast.makeText(this, "Екран: Вибір відеокарти",
            Toast.LENGTH_SHORT).show();
        LinearLayout container =
            findViewById(R.id.gpuListContainer);
        DatabaseHelper dbHelper = new DatabaseHelper(this);
        SQLiteDatabase db = dbHelper.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT id, name, memory, cores,
            interface, price FROM gpus", null);
```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

У цій активності реалізовано виведення списку материнських плат із таблиці motherboards. Дані зчитуються за допомогою SQL-запиту та готуються до відображення у вказаному контейнері.

Фрагмент коду представлено в лістингу 3.4.

Лістинг 3.4 – Ініціалізація MotherboardsActivity та завантаження даних

```
public class MotherboardsActivity extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_motherboards);
        Toast.makeText(this, "Екран: Вибір материнської плати",
        Toast.LENGTH_SHORT).show();
        LinearLayout container =
        findViewById(R.id.motherboardListContainer);
        DatabaseHelper dbHelper = new DatabaseHelper(this);
        SQLiteDatabase db = dbHelper.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT id, name, chipset,
        socket, form_factor, ram_type, price FROM motherboards", null);
```

Наступний фрагмент XML визначає користувацький інтерфейс екрана, де відображається зібрана конфігурація ПК. Основу становить ScrollView, що дозволяє прокручувати вміст у вертикальному напрямку, а також заголовок «Моя конфігурація», вирівняний по центру. Фрагмент коду представлено в лістингу 3.5.

Лістинг 3.5 – Розмітка інтерфейсу

```
<ScrollView
    android:id="@+id/scrollView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_above="@+id/bottomButtons"
    android:padding="16dp">
    <LinearLayout
        android:id="@+id/myConfigLayout"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="start">
```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

```

<TextView
    android:id="@+id/titleText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Моя конфігурація"
    android:textSize="24sp"
    android:textStyle="bold"
    android:textColor="@color/text_white"
    android:gravity="center"
    android:paddingBottom="16dp" />
</LinearLayout>
</ScrollView>

```

У цьому фрагменті здійснюється зчитування інформації про охолоджувальні системи з бази даних SQLite. Отримані дані формуються у вигляді текстових елементів, які динамічно додаються до інтерфейсу за допомогою `LinearLayout`. Фрагмент коду представлено в лістингу 3.6.

Лістинг 3.6 – Завантаження даних про системи охолодження

```

Cursor cursor = db.rawQuery("SELECT id, name, type,
fan_size, socket_compatibility, noise_level, price FROM
coolings", null);
if (cursor.moveToFirst()) {
    do {
        String name = cursor.getString(1);
        String specs = cursor.getString(2) + " | " +
cursor.getString(3) + " | " + cursor.getString(4) + " | " +
cursor.getString(5) + " | Ціна: " + cursor.getInt(6) + " грн";
        TextView nameView = new TextView(this);
        nameView.setText(name);
        TextView specsView = new TextView(this);
        specsView.setText(specs);
        LinearLayout itemLayout = new LinearLayout(this);
        itemLayout.setOrientation(LinearLayout.VERTICAL);
        itemLayout.addView(nameView);

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

```

        itemLayout.addView(specsView);
        coolingListContainer.addView(itemLayout);
    } while (cursor.moveToNext());
}
cursor.close();

```

Розробка функціональної та візуальної частини мобільного застосунку Build PC була успішно реалізована відповідно до поставлених вимог. Усі основні модулі – авторизація, реєстрація, початок збірки, вибір компонентів, збереження конфігурації, перегляд поточного складу ПК – повністю функціонують і пройшли первинне тестування. Інтерфейс користувача створено на базі Android-компонентів із дотриманням принципів адаптивності, читабельності та зручності навігації. Усі екрани мають логічну структуру, узгоджену візуальну стилістику та інтуїтивно зрозумілу поведінку елементів. Робота з базою даних реалізована засобами SQLite, що забезпечує швидкий доступ до інформації, без потреби у зовнішньому з'єднанні. У результаті реалізації цього етапу було створено стабільний прототип застосунку, який дозволяє ефективно виконувати завдання з підбору комп'ютерних комплектуючих, забезпечуючи користувачеві простоту та зручність використання.

3.3 Тестування застосунку Build PC

Після завершення етапу реалізації основного функціоналу буде проведено тестування мобільного застосунку Build PC з метою перевірки коректності роботи системи, виявлення можливих помилок та забезпечення стабільності в різних умовах використання.

Тестування охоплює такі основні аспекти:

- Функціональне тестування.
- Реєстрація нового користувача з перевіркою унікальності email.
- Авторизація з правильними та неправильними даними.
- Перегляд зібраної конфігурації.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

– Перевірка сумісності компонентів.

Проведення тестування застосунку:

Для початку потрібно у полях вказати електронну пошту та пароль.

Це зображено на рисунку 3.14

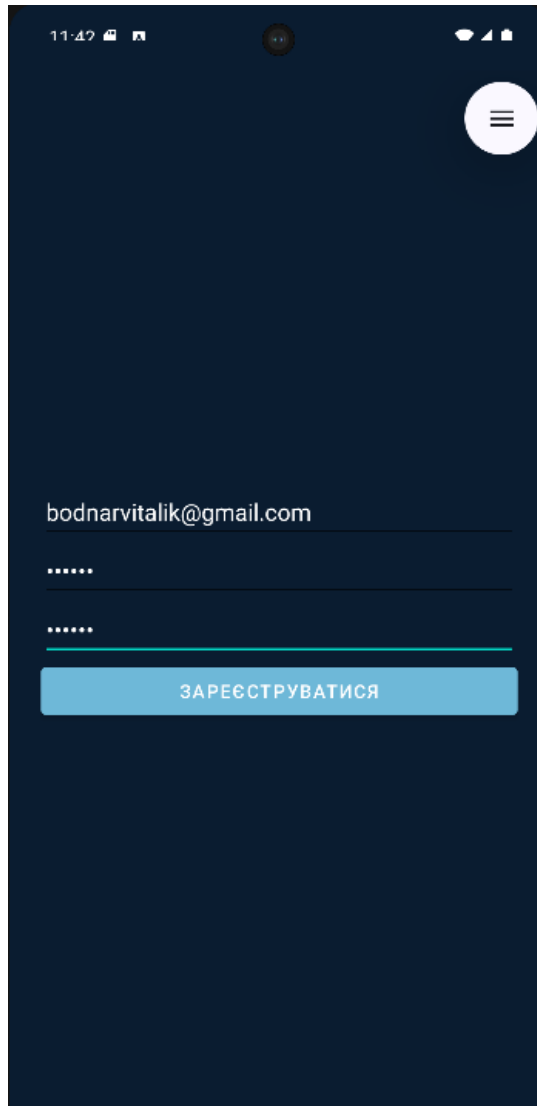


Рисунок 3.14 – Форма реєстрації

Далі реалізовано реєстрацію нового користувача з перевіркою унікальності введеної електронної пошти. Це підтверджено на рисунку 3.15.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

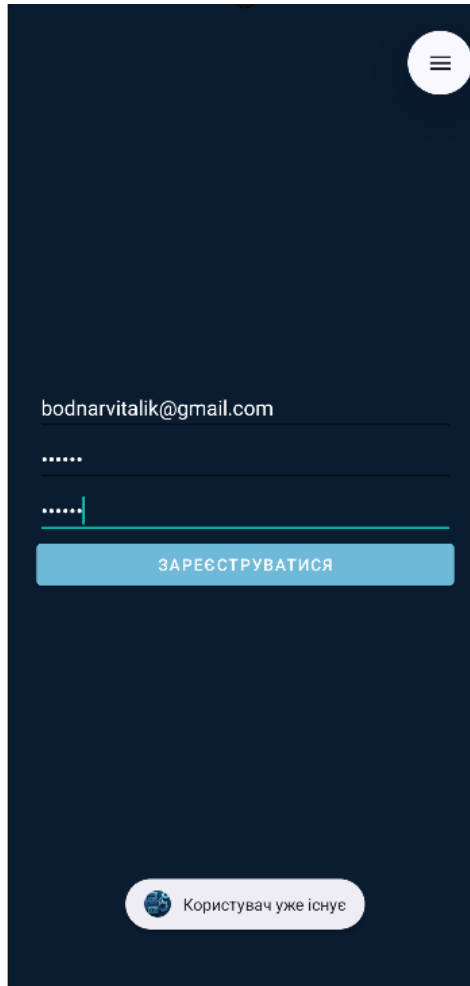


Рисунок 3.15 – Повідомлення про унікальність

Також була впроваджена система авторизації, яка дозволяє користувачу увійти до застосунку, ввівши коректні дані. Це показано на рисунку 3.16.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

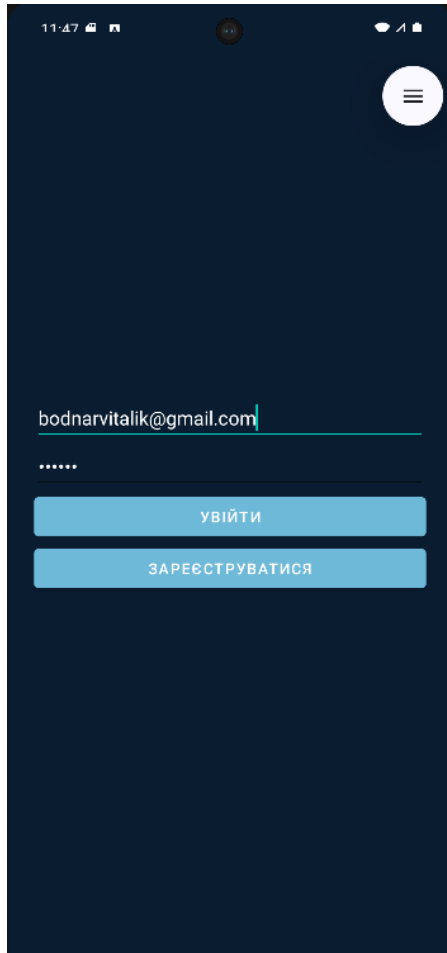


Рисунок 3.16 – Забезпечення коректності введених даних

Наступним кроком було протестовано правильне відображення елементів інтерфейсу на різних пристроях та екранах із різною щільністю пікселів. Оцінено зручність навігації, розміщення кнопок, списків, полів вводу. Усі основні елементи були адаптовані до мобільного формату. Це зображено на рисунку 3.17.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

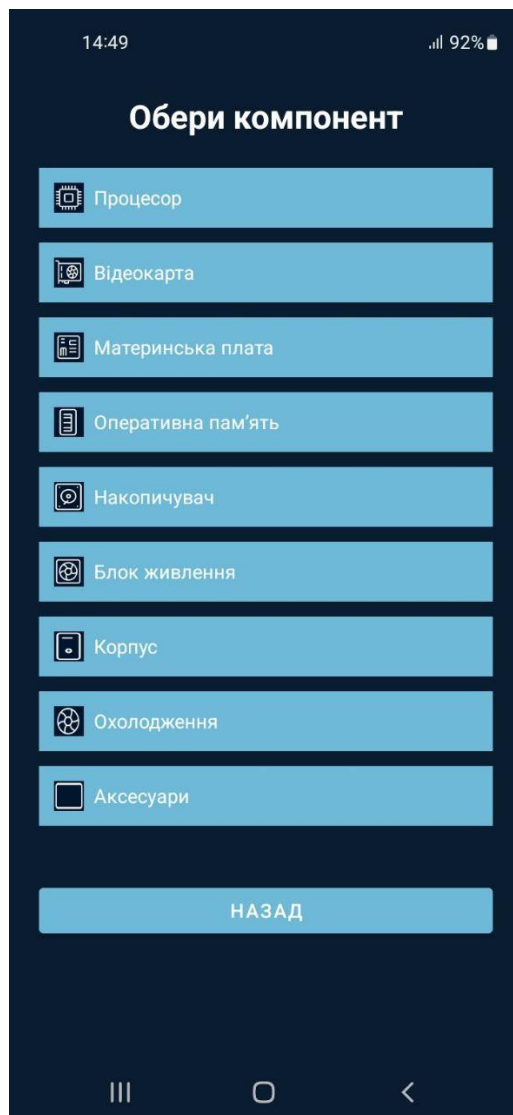


Рисунок 3.17 – Адаптовано до мобільного пристрою

Далі біля кожного компонента можна натиснути на значок “+”, після чого він додається до збірки конфігурації, це зображено на рисунку 3.18.



Рисунок 3.18 – Додавання компоненту

Аналогічно було проведено таке тестування з іншими вкладками і різними компонентами.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

Після цього компонент автоматично був доданий в конфігурацію збірки, де потім можна видалити будь який доданий елемент, натиснувши на значок “-”. Це показано на рисунку 3.19



Рисунок 3.19 – Перевірка конфігурації

Аналогічно було проведено таке тестування з іншими компонентами.

Після повної збірки виводиться повідомлення про те, що конфігурація завершена та оголошено загальну вартість. Це показано на рисунку 3.20

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

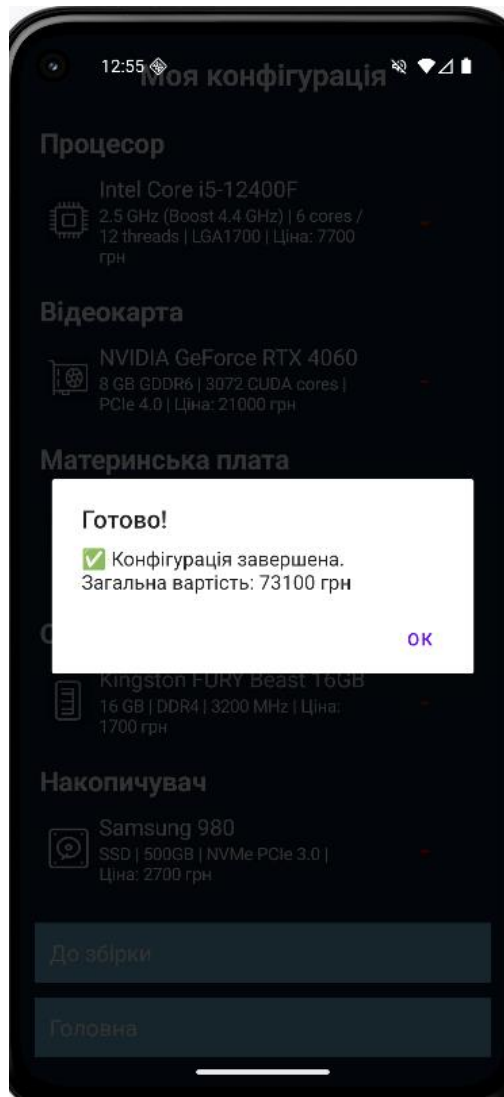


Рисунок 3.20 – Завершена конфігурація

Після збірки всіх складових ПК, можна перевірити сумісність між ними за допомогою кнопки “Перевірити сумісність”. Для цього потрібно перейти на головну сторінку. Результат показано на рисунку 3.21

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

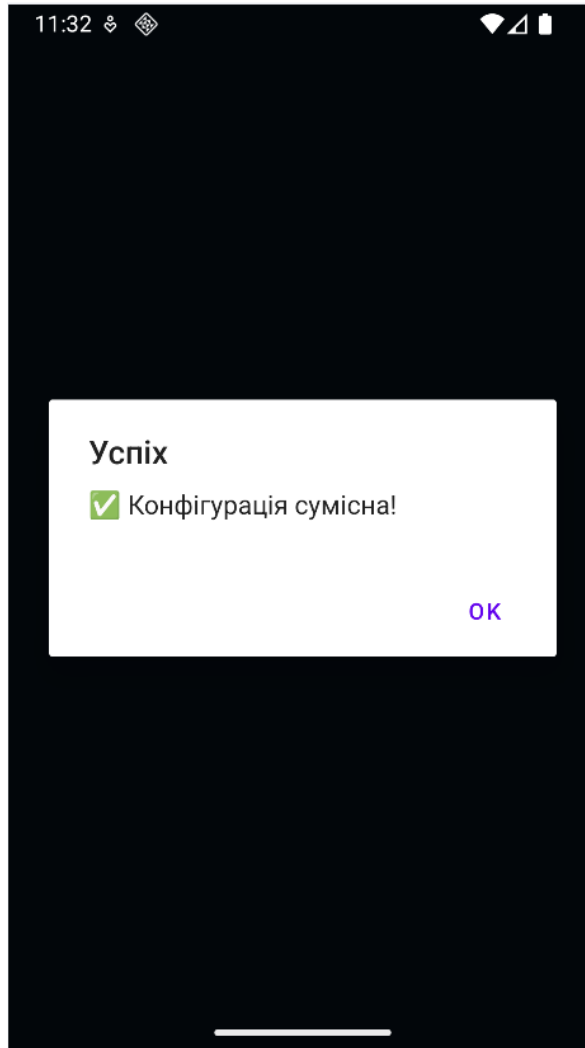


Рисунок 3.21 – Перевірка сумісності

У результаті проведеного тестування було встановлено, що мобільний застосунок працює стабільно за стандартних сценаріїв використання.

Реалізовано перевірку унікальності електронної адреси при реєстрації нового користувача. Такий підхід запобігає створенню дублюючих облікових записів і забезпечує цілісність даних. У разі спроби зареєструватися з уже існуючим email, користувач отримує відповідне повідомлення про помилку, що також підвищує зручність взаємодії з системою.

Усі основні функціональні можливості, включаючи реєстрацію, авторизацію, збереження облікових даних, створення конфігурацій та перевірку сумісності компонентів, виконуються коректно та згідно з поставленими вимогами.

					КР.КН 25.584.03.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Застосунок продемонстрував високу надійність, готовність до можливого використання іншими користувачами та потенціал для подальшого вдосконалення. У майбутньому планується додати нові рівні безпеки, такі як двофакторна автентифікація, та розширити систему прав доступу, щоб забезпечити ще вищий рівень захисту персональної інформації. У перспективі планується також реалізація функції динамічного оновлення інформації про комплектуючі – ціни, наявність та технічні характеристики зможуть автоматично завантажуватися з відкритих джерел або за допомогою API-підключення до партнерських онлайн-магазинів.

Усі ці вдосконалення спрямовані на підвищення зручності, функціональності та адаптивності застосунку до потреб сучасного користувача.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

4.1 Аналіз ринку

Мобільний застосунок «Build PC» є програмним продуктом, призначеним для автоматизованого підбору комп'ютерних комплектуючих з урахуванням їх сумісності. Основними техніко-економічними характеристиками продукту є: зручний адаптивний інтерфейс, система перевірки сумісності компонентів, можливість створення, збереження та редагування збірок, робота в режимі офлайн, мінімальні вимоги до ресурсів смартфона. Застосунок має високу швидкість обробки даних, підтримує найпоширеніші категорії комплектуючих (процесори, материнські плати, відеокарти, оперативна пам'ять, накопичувачі тощо), а також дозволяє зберігати конфігурації для подальшого аналізу.

Програмний продукт не є абсолютно новим, але має суттєві відмінності від існуючих сервісів, таких як Telemart, Hotline або Rozetka. Основна відмінність полягає в тому, що «Build PC» не є комерційною платформою, а фокусується саме на підборі та перевірці сумісності комплектуючих, без нав'язування конкретного продавця. Це робить його зручним не лише для кінцевих покупців, а й для студентів, консультантів, технічних спеціалістів, які хочуть швидко перевірити конфігурацію без потреби аналізувати пропозиції на різних сайтах.

Потенційними користувачами застосунку є люди, які хочуть зібрати свій комп'ютер. Географія реалізації охоплює передусім ринок України, однак застосунок може бути представлений і в інших країнах, якщо буде локалізований англійською мовою.

Очікуваний попит на застосунок оцінюється як стабільний і поступово зростаючий, оскільки самостійне складання ПК стає все більш популярним серед молоді. Основною формою реалізації продукту є безкоштовне

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

розміщення в Google Play з можливістю монетизації через підписку на додаткові функції або вбудовану рекламу.

Сервісне обслуговування включає онлайн-підтримку через email, FAQ-розділ у застосунку та можливість звернення через месенджери. Післяпродажна підтримка буде зосереджена на оновленнях застосунку та підтримці актуальності бази даних комплектуючих.

Основними конкурентами на ринку є інтернет-магазини з модулем підбору комплектуючих (наприклад, Hotline, Brain, Telemart), проте жоден із них не пропонує автономний мобільний застосунок зі зручним офлайн-доступом і функціоналом перевірки сумісності без прямої прив'язки до товарного каталогу.

Конкурентні переваги «Build PC» полягають у простоті використання, універсальності, відсутності комерційного тиску, підтримці української мови та можливості використання в навчальних цілях. За рахунок цього застосунок має всі шанси зайняти нішу серед безкоштовних інструментів для підбору ПК.

4.2 Розрахунок витрат на розробку

Розробка мобільного застосунку «Build PC» передбачала реалізацію повного комплексу робіт – від формування технічного завдання до створення структурної бази даних, моделювання взаємодії компонентів, розробки інтерфейсу користувача та оптимізації роботи програми. Всі етапи виконувалися індивідуально, що дозволяє чітко структурувати склад витрат, які виникли в процесі проектування.

Найбільшу частку витрат становить заробітна плата виконавця, який виконував увесь обсяг проектно-розробних робіт. Загальна тривалість проектування склала 160 годин, погодинна оплата – 100 грн. Це дозволяє оцінити загальні витрати на заробітну плату на рівні 16 000 грн. В рамках цієї суми враховано аналіз ринку та функціональних потреб, розробку структури застосунку, дизайн інтерфейсу, базові тести та налагодження системи.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

До заробітної плати додаються нарахування на соціальне страхування відповідно до чинного законодавства. Єдиний соціальний внесок (ЄСВ), який сплачується у розмірі 22% від зарплати, становить 3 520 грн.

Витрати на програмне забезпечення включають платну підписку на сервіс Figma Pro, що використовувався для побудови інтерактивних макетів інтерфейсу користувача та демонстрації логіки навігації. Протягом трьох місяців використання витрати склали 1 350 грн. Решта інструментів, таких як Android Studio та Firebase, використовувалися у базовій безкоштовній версії, тому не входять до розрахунку.

До витрат на обладнання віднесено часткову амортизацію ноутбука, що був задіяний під час розробки. З урахуванням періоду активної експлуатації для потреб проекту та розподілу вартості пристрою, витрати на амортизацію оцінюються в 2 000 грн.

Окремо враховано витрати на електроенергію, необхідну для забезпечення роботи обладнання. При середній потужності ноутбука в 150 Вт, тривалості роботи близько 80 годин та тарифі 6 грн/кВт·год, витрати склали 72 грн.

Інші витрати охоплюють побутові й організаційні потреби, що виникають у процесі реалізації проекту. До них належать канцелярські витрати (папір, зошити, ручки), витрати на інтернет-зв'язок, мобільний зв'язок, а також транспортні витрати, пов'язані з поїздками на консультації та проведенням опитувань. Загальна сума за цією статтею – 1 850 грн.

Всі витрати були узагальнені в підсумковій таблиці 4.1. Це дозволяє наочно оцінити розподіл фінансових ресурсів і глибше проаналізувати загальну економічну картину витрат на проектування.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

Таблиця 4.1 – Підсумок витрат

Категорія	Назва	Кількість	Вартість за одиницю (грн)	Сума (грн)
Заробітна плата	Робота виконавця (160 год)	1	16 000	16 000
ЄСВ	Єдиний соціальний внесок (22%)	1	3 520	3 520
	Figma (3 міс)	1	1 350	1 350
Обладнання	Амортизація ноутбука	1	2 000	2 000
Електроенергія	Споживання під час роботи	1	72	72
Інше	Канцелярія + інтернет + транспорт	1	1 850	1 850

Отже загальні витрати на розробку становлять: $16\,000 + 3\,520 + 1\,350 + 2\,000 + 72 + 1\,850 = 24\,792$ грн

Таким чином, сукупні витрати на проектування мобільного застосунку «Build PC» становлять 24 792 грн. Вони включають усі необхідні ресурси, починаючи від оплати праці та технічного забезпечення, закінчуючи організаційними та інфраструктурними витратами. Отримане значення стане базою для наступного етапу – оцінки ефективності розробки, терміну її окупності та загального економічного обґрунтування доцільності впровадження програмного продукту.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

4.3 Обґрунтування необхідності розробки

У сучасних умовах високої динаміки розвитку інформаційних технологій користувачі все частіше прагнуть самостійно формувати конфігурації персональних комп'ютерів відповідно до власних потреб. Однак через складність технічних характеристик, велику кількість компонентів на ринку та необхідність врахування сумісності між ними, процес підбору комплектуючих часто є складним і потребує професійної підготовки. Саме тому розробка мобільного застосунку «Build PC» є технічно та економічно доцільною.

Цей програмний продукт покликаний автоматизувати процес створення конфігурацій ПК, забезпечити зручну перевірку сумісності обраних компонентів і зменшити ризик помилок. Застосунок орієнтований на широку аудиторію користувачів: початківців, студентів технічних спеціальностей, ентузіастів та консультантів з продажу комп'ютерної техніки.

Застосунок «Build PC» дозволяє значно спростити процес вибору комплектуючих, зменшити навантаження на персонал магазинів і сервісних центрів, а також підвищити загальну якість обслуговування. Його функціонал забезпечує швидке формування збірки, автоматичну перевірку на відповідність, збереження конфігурацій, можливість повернення до попередніх варіантів, що робить взаємодію із застосунком зручною та ефективною.

Також продукт має потенціал для використання в освітньому середовищі – як візуальний інструмент для навчання основам комп'ютерної інженерії, обробки даних, логіки систем і взаємодії компонентів ПК.

З економічної точки зору, розробка застосунку не потребує надмірних витрат, а витрати на його реалізацію швидко окупаються за рахунок зекономленого часу й ресурсів. Це підтверджено розрахунком терміну окупності, який становить менше чотирьох місяців. Завдяки поєднанню практичної цінності, простоти використання та потенціалу масштабування,

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

розробка мобільного застосунку «Build PC» є обґрунтованим та перспективним рішенням як у навчальному, так і в прикладному середовищі.

У процесі виконання економічної частини було здійснено повний розрахунок витрат на розробку мобільного застосунку «Build PC», визначено економічний ефект від його впровадження, розраховано термін окупності та коефіцієнт ефективності.

Загальні витрати на розробку застосунку склали 24 792 грн, що включає оплату праці, нарахування ЄСВ, витрати на програмне забезпечення, електроенергію, амортизацію обладнання та супутні організаційні витрати. Усі розрахунки здійснено згідно з методичними рекомендаціями та реалістичними умовами розробки.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

ВИСНОВКИ

У процесі розробки мобільного застосунку «Build PC» було реалізовано програмний засіб, призначений для зручного та ефективного підбору комп'ютерних комплектуючих з урахуванням їх сумісності. Застосунок дозволяє користувачу самостійно формувати конфігурацію ПК, зберігати її, змінювати, перевіряти на сумісність і працювати з нею в офлайн-режимі. Завдяки інтуїтивно зрозумілому інтерфейсу, застосунок орієнтований як на новачків, так і на технічно підготовлених користувачів.

У ході реалізації було виконано повний цикл створення застосунку: проведено аналіз аналогів і предметної області, сформульовано вимоги, розроблено базу даних, створено графічний інтерфейс, реалізовано функціонал перевірки сумісності, протестовано працездатність окремих модулів та всього застосунку в цілому.

Було також проведено техніко-економічне обґрунтування доцільності створення застосунку. Визначено потенційну аудиторію, конкурентні переваги, сферу застосування та перспективи масштабування.

Окрему увагу було приділено забезпеченню адаптивності дизайну, оптимізації швидкодії та можливості подальшого розширення функціоналу. Завдяки підтримці української мови, автономності та простоті використання, застосунок може ефективно застосовуватися не лише в особистих цілях, але й у навчальному процесі або як допоміжний інструмент для консультантів у сфері комп'ютерних технологій.

Таким чином, мобільний застосунок «Build PC» відповідає поставленим цілям, виконує необхідні функції, має чітку практичну цінність та є готовим до розповсюдження серед цільової аудиторії.

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Download Android Studio & App Tools - Android Developers. Android Developers. URL: <https://developer.android.com/studio> (дата звернення: 08.05.2025).

2. SQLite Documentation. Офіційна документація SQLite. URL: <https://www.sqlite.org/docs.html> (дата звернення: 10.05.2025).

3. Figma Design Tutorials. Figma Community. URL: <https://www.figma.com/community/design-tutorials/figma> (дата звернення: 12.05.2025).

4. Основи програмування на Java. Курс Prometheus. URL: <https://prometheus.org.ua/prometheus-free/fundamentals-of-java-programming/> (дата звернення: 12.05.2025).

5. Основи Android програмування. Навчальний посібник. Запорізький національний університет. URL: <https://files.znu.edu.ua/files/Bibliobooks/Inshi70/0051178.pdf> (дата звернення: 15.05.2025).

6. Бойко О. В., Мельник В. М. Android - програмування: основи та практика : навч. посіб. Львів : Видавництво Львівської політехніки, 2022. 350 с. (дата звернення: 17.05.2025)

7. Oracle Java SE Documentation. Java Documentation. URL: <https://docs.oracle.com/javase/> (дата звернення: 20.05.2025).

8. Архітектура мобільного застосунку: усе, що потрібно знати. Brander. URL: <https://brander.ua/blog/arkhitektura-mobilnoho-zastosunku-use-shcho-potribno-znaty> (дата звернення: 26.05.2025).

9. Ковальчук А. М., Петренко С. В. Сучасні технології мобільного програмування : монографія. Харків : Фоліо, 2023. 200 с. (дата звернення: 29.05.2025)

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

ДОДАТКИ

Додаток А

Лістинг програмного коду екрану авторизації

```
package com.example.buildpc;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class LoginActivity extends AppCompatActivity {
    EditText editEmail, editPassword;
    Button btnLogin, btnRegister;
    DatabaseHelper db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        editEmail = findViewById(R.id.editEmail);
        editPassword = findViewById(R.id.editPassword);
        btnLogin = findViewById(R.id.btnLogin);
        btnRegister = findViewById(R.id.btnRegister);
        db = new DatabaseHelper(this);
        btnLogin.setOnClickListener(v -> {
            String email = editEmail.getText().toString();
            String password = editPassword.getText().toString();
            if (db.loginUser(email, password)) {
                SharedPreferences prefs = getSharedPreferences("UserPrefs",
                MODE_PRIVATE);
                prefs.edit().putBoolean("logged_in", true).apply();

                startActivity(new Intent(LoginActivity.this,
                MainActivity.class));
                finish();
            } else {
                Toast.makeText(this, "Невірна пошта або пароль",
                Toast.LENGTH_SHORT).show();
            }
        });
        btnRegister.setOnClickListener(v -> {
            startActivity(new Intent(LoginActivity.this,
            RegisterActivity.class));
        });
    }
}
```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

Додаток Б

Лістинг програмного коду екрану вибору процесора

```
package com.example.buildpc;

import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.os.Bundle;
import android.view.Gravity;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class GPUActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gpu);

        Toast.makeText(this, "Екран: Вибір відеокарти",
            Toast.LENGTH_SHORT).show();
        LinearLayout container =
            findViewById(R.id.gpuListContainer);
        DatabaseHelper dbHelper = new DatabaseHelper(this);
        SQLiteDatabase db = dbHelper.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT id, name, memory, cores,
            interface, price FROM gpus", null);
        if (cursor.moveToFirst()) {
            do {
                int id = cursor.getInt(0);
                String name = cursor.getString(1);
                String memory = cursor.getString(2);
                String cores = cursor.getString(3);
                String iface = cursor.getString(4);
                int price = cursor.getInt(5);

                LinearLayout itemLayout = new LinearLayout(this);
                itemLayout.setOrientation(LinearLayout.HORIZONTAL);
                itemLayout.setPadding(16, 16, 16, 16);
                itemLayout.setGravity(Gravity.CENTER_VERTICAL);
                itemLayout.setLayoutParams(new LinearLayout.LayoutParams (
                    ViewGroup.LayoutParams.MATCH_PARENT,
                    ViewGroup.LayoutParams.WRAP_CONTENT
                ));

                ImageView icon = new ImageView(this);
```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

```

        icon.setImageResource(R.drawable.ic_gpu);
        LinearLayout.LayoutParams imageParams = new
LinearLayout.LayoutParams(100, 100);
        icon.setLayoutParams(imageParams);
        itemLayout.addView(icon);

        LinearLayout textLayout = new LinearLayout(this);
        textLayout.setOrientation(LinearLayout.VERTICAL);
        textLayout.setPadding(16, 0, 0, 0);
        textLayout.setLayoutParams(new LinearLayout.LayoutParams(0,
ViewGroup.LayoutParams.WRAP_CONTENT, 1));

        TextView nameView = new TextView(this);
        nameView.setText(name);
        nameView.setTextColor(getResources().getColor(R.color.text_w
hite));
        nameView.setTextSize(18);
        textLayout.addView(nameView);

        TextView specsView = new TextView(this);
        specsView.setText(memory + " | " + cores + " | " + iface + "
| Ціна: " + price + " грн");
        specsView.setTextColor(getResources().getColor(R.color.text_
white));
        specsView.setTextSize(14);
        textLayout.addView(specsView);

        itemLayout.addView(textLayout);
        Button addButton = new Button(this);
        addButton.setText("+");
        addButton.setTag(id);
        addButton.setTextColor(Color.BLACK);
        addButton.setTextSize(28f);
        addButton.setBackgroundColor(Color.TRANSPARENT);
        addButton.setPadding(24, 12, 24, 12);

        addButton.setOnClickListener(v -> {
            int gpuId = (int) v.getTag();
            String email = UserSession.getEmail();
            if (email == null || email.isEmpty()) {
                Toast.makeText(this, "Користувач не авторизований",
Toast.LENGTH_SHORT).show();
            }
            return;
        });

        SQLiteDatabase writeDb = dbHelper.getWritableDatabase();
        int userId = dbHelper.getUserIdByEmail(writeDb, email);

        if (userId != -1) {
            Cursor checkCursor = writeDb.rawQuery("SELECT id FROM
user_config WHERE user_id = ?", new
String[]{String.valueOf(userId)});
            ContentValues values = new ContentValues();

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

Додаток В

Лістинг програмного коду екрану реєстрації користувача

```
package com.example.buildpc;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import android.util.Patterns;
public class RegisterActivity extends AppCompatActivity {
    EditText editEmail, editPassword, editConfirm;
    Button btnSignUp;
    DatabaseHelper db;

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

        editEmail = findViewById(R.id.editEmail);
        editPassword = findViewById(R.id.editPassword);
        editConfirm = findViewById(R.id.editConfirm);
        btnSignUp = findViewById(R.id.btnSignUp);
        db = new DatabaseHelper(this);

        btnSignUp.setOnClickListener(v -> {
            String email = editEmail.getText().toString();
            String pass = editPassword.getText().toString();
            String confirm = editConfirm.getText().toString();

            if (email.isEmpty() || pass.isEmpty() || confirm.isEmpty())
            {
                Toast.makeText(this, "Будь ласка, заповніть усі поля",
                Toast.LENGTH_SHORT).show();
                return;
            }

            if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
                Toast.makeText(this, "Невірний формат електронної пошти",
                Toast.LENGTH_SHORT).show();
                return;
            }

            if (!isPasswordValid(pass)) {
                Toast.makeText(this, "Пароль має містити щонайменше 8
                символів, включаючи літери та цифри", Toast.LENGTH_LONG).show();
                return;
            }

            if (!pass.equals(confirm)) {
```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

```

        Toast.makeText(this, "Паролі не співпадають",
Toast.LENGTH_SHORT).show();
        return;
    }

    if (db.registerUser(email, pass)) {
        Toast.makeText(this, "Реєстрація успішна",
Toast.LENGTH_SHORT).show();
        startActivity(new Intent(this, LoginActivity.class));
        finish();
    } else {
        Toast.makeText(this, "Користувач уже існує",
Toast.LENGTH_SHORT).show();
    }
});
}

private boolean isPasswordValid(String password) {
    String passwordPattern = "^(?=.*[A-Za-z])(?=.*\\d)[A-Za-
z\\d]{8,}$";
    return password.matches(passwordPattern);
}
}

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

Додаток Г

Лістинг програмного коду екрану вибору охолодження

```
package com.example.buildpc;
import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Button;
import android.widget.Toast;
import android.view.Gravity;
import android.view.ViewGroup;
import androidx.appcompat.app.AppCompatActivity;

public class CoolingActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cooling);

        Toast.makeText(this, "Екран: Вибір охолодження",
            Toast.LENGTH_SHORT).show();

        LinearLayout coolingListContainer =
            findViewById(R.id.coolingListContainer);

        DatabaseHelper dbHelper = new DatabaseHelper(this);
        SQLiteDatabase db = dbHelper.getReadableDatabase();

        Cursor cursor = db.rawQuery("SELECT id, name, type,
            fan_size, socket_compatibility, noise_level, price FROM
            coolings", null);

        if (cursor.moveToFirst()) {
            do {
                int id = cursor.getInt(0);
                String name = cursor.getString(1);
                String type = cursor.getString(2);
                String fanSize = cursor.getString(3);
                String socket = cursor.getString(4);
                String noise = cursor.getString(5);
                int price = cursor.getInt(6);

                LinearLayout itemLayout = new LinearLayout(this);
                itemLayout.setOrientation(LinearLayout.HORIZONTAL);
                itemLayout.setPadding(16, 16, 16, 16);
                itemLayout.setGravity(Gravity.CENTER_VERTICAL);
```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

```

        LinearLayout.LayoutParams params = new
LinearLayout.LayoutParams (
        ViewGroup.LayoutParams.MATCH_PARENT,
        ViewGroup.LayoutParams.WRAP_CONTENT
        );
        params.setMargins(0, 0, 0, 16);
        itemLayout.setLayoutParams(params);

        ImageView imageView = new ImageView(this);
        imageView.setImageResource(R.drawable.ic_cooling);
        imageView.setLayoutParams(new LinearLayout.LayoutParams(100,
100));
        itemLayout.addView(imageView);

        LinearLayout textLayout = new LinearLayout(this);
        textLayout.setOrientation(LinearLayout.VERTICAL);
        textLayout.setPadding(16, 0, 0, 0);
        textLayout.setLayoutParams(new LinearLayout.LayoutParams(0,
ViewGroup.LayoutParams.WRAP_CONTENT, 1));

        TextView nameView = new TextView(this);
        nameView.setText(name);
        nameView.setTextColor(getResources().getColor(R.color.text_w
hite));
        nameView.setTextSize(18);
        textLayout.addView(nameView);

        TextView specsView = new TextView(this);
        specsView.setText(type + " | " + fanSize + " | " + socket +
" | " + noise + " | Ціна: " + price + " грн");
        specsView.setTextColor(getResources().getColor(R.color.text_
white));
        specsView.setTextSize(14);
        textLayout.addView(specsView);

        itemLayout.addView(textLayout);

        Button addButton = new Button(this);
        addButton.setText("+");
        addButton.setTag(id);
        addButton.setTextColor(Color.BLACK);
        addButton.setTextSize(28f);
        addButton.setBackgroundColor(Color.TRANSPARENT);
        addButton.setPadding(24, 12, 24, 12);

        addButton.setOnClickListener(v -> {
        int coolingId = (int) v.getTag();
        Toast.makeText(this, "Обрано охолодження з ID: " +
coolingId, Toast.LENGTH_SHORT).show();

        String currentEmail = UserSession.getEmail();
        SQLiteDatabase writeDb = dbHelper.getWritableDatabase();

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

```

        int userId = dbHelper.getUserIdByEmail(writeDb,
currentEmail);
        if (userId != -1) {
            Cursor configCursor = writeDb.rawQuery("SELECT id FROM
user_config WHERE user_id = ?", new
String[]{String.valueOf(userId)});

            ContentValues values = new ContentValues();
            values.put("cooling_id", coolingId);
            if (configCursor.moveToFirst()) {
                int configId = configCursor.getInt(0);
                int rows = writeDb.update("user_config", values, "id = ?",
new String[]{String.valueOf(configId)});
                if (rows > 0) {
                    Toast.makeText(this, "Охолодження оновлено в конфігурації",
Toast.LENGTH_SHORT).show();
                }
            } else {
                values.put("user_id", userId);
                values.put("cpu_id", (Integer) null);
                values.put("gpu_id", (Integer) null);
                values.put("motherboard_id", (Integer) null);
                values.put("ram_id", (Integer) null);
                values.put("storage_id", (Integer) null);
                values.put("psu_id", (Integer) null);
                values.put("case_id", (Integer) null);

                long result = writeDb.insert("user_config", null, values);
                if (result != -1) {
                    Toast.makeText(this, "Охолодження додано до конфігурації",
Toast.LENGTH_SHORT).show();
                }
            }
            configCursor.close();
            writeDb.close();
            Intent intent = new Intent(CoolingActivity.this,
MyConfigActivity.class);
            startActivity(intent);
        } else {
            Toast.makeText(this, "Користувача не знайдено",
Toast.LENGTH_SHORT).show();
        }
    });

    itemLayout.addView(addButton);
    coolingListContainer.addView(itemLayout);
} while (cursor.moveToNext());
}
cursor.close();
db.close();
}
}

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

Додаток Г

Лістинг програмного коду екрану перегляду конфігурації

```
package com.example.buildpc;

import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.Gravity;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MyConfigActivity extends AppCompatActivity {
    private LinearLayout layout;
    private DatabaseHelper dbHelper;
    private int userId;
    private boolean allComponentsExist = false;
    private boolean hasAccessories = false;

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

        LinearLayout btnGoHome = findViewById(R.id.btnGoHome);
        btnGoHome.setOnClickListener(v -> {
            Intent intent = new Intent(MyConfigActivity.this,
MainActivity.class);
            startActivity(intent);
            finish();
        });

        LinearLayout btnStartBuild =
findViewById(R.id.btnStartBuild);
        btnStartBuild.setOnClickListener(v -> {
            Intent intent = new Intent(MyConfigActivity.this,
StartBuildActivity.class);
            startActivity(intent);
            finish();
        });
    }
}
```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

```

layout = findViewById(R.id.myConfigLayout);
dbHelper = new DatabaseHelper(this);

String currentEmail = UserSession.getEmail();
SQLiteDatabase db = dbHelper.getReadableDatabase();
userId = dbHelper.getUserIdByEmail(db, currentEmail);

if (userId == -1) {
    Toast.makeText(this, "Користувача не знайдено",
Toast.LENGTH_SHORT).show();
    return;
}

Cursor configCursor = db.rawQuery("SELECT cpu_id, gpu_id,
motherboard_id, ram_id, storage_id, psu_id, case_id, cooling_id
FROM user_config WHERE user_id = ?",
new String[]{String.valueOf(userId)});

if (configCursor.moveToFirst()) {
int cpuId = configCursor.getInt(0);
int gpuId = configCursor.getInt(1);
int motherboardId = configCursor.getInt(2);
int ramId = configCursor.getInt(3);
int storageId = configCursor.getInt(4);
int psuId = configCursor.getInt(5);
int caseId = configCursor.getInt(6);
int coolingId = configCursor.getInt(7);

allComponentsExist = cpuId > 0 && gpuId > 0 && motherboardId
> 0 && ramId > 0 && storageId > 0 && psuId > 0 && caseId > 0 &&
coolingId > 0;

loadComponent("Процесор", "cpus", cpuId, "name, frequency,
cores, socket, price", R.drawable.ic_cpu, new
String[]{"cpu_id"});
loadComponent("Відеокарта", "gpus", gpuId, "name, memory,
cores, interface, price", R.drawable.ic_gpu, new
String[]{"gpu_id"});
loadComponent("Материнська плата", "motherboards",
motherboardId, "name, chipset, socket, form_factor, price",
R.drawable.ic_motherboard, new String[]{"motherboard_id"});
loadComponent("Оперативна пам'ять", "ram", ramId, "name,
size, type, speed, price", R.drawable.ic_ram, new
String[]{"ram_id"});
loadComponent("Накопичувач", "storages", storageId, "name,
type, capacity, interface, price", R.drawable.ic_storage, new
String[]{"storage_id"});
loadComponent("Блок живлення", "psus", psuId, "name,
wattage, form_factor, efficiency, price", R.drawable.ic_psu, new
String[]{"psu_id"});
loadComponent("Корпус", "cases", caseId, "name, form_factor,
mb_support, gpu_length, price", R.drawable.ic_case, new
String[]{"case_id"});

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

```

        loadComponent("Охолодження", "coolings", coolingId, "name,
type, fan_size, socket_compatibility, price",
R.drawable.ic_cooling, new String[]{"cooling_id"});
    } else {
        Toast.makeText(this, "Конфігурація не знайдена",
Toast.LENGTH_SHORT).show();
    }

    configCursor.close();
    loadAccessories();

    if (allComponentsExist && hasAccessories) {
        int totalPrice = 0;

        Cursor priceCursor = db.rawQuery("SELECT " +
"c.price, g.price, m.price, r.price, s.price, p.price,
cs.price, cl.price " +
"FROM user_config uc " +
"LEFT JOIN cpus c ON uc.cpu_id = c.id " +
"LEFT JOIN gpus g ON uc.gpu_id = g.id " +
"LEFT JOIN motherboards m ON uc.motherboard_id = m.id " +
"LEFT JOIN ram r ON uc.ram_id = r.id " +
"LEFT JOIN storages s ON uc.storage_id = s.id " +
"LEFT JOIN psus p ON uc.psu_id = p.id " +
"LEFT JOIN cases cs ON uc.case_id = cs.id " +
"LEFT JOIN coolings cl ON uc.cooling_id = cl.id " +
"WHERE uc.user_id = ?", new
String[]{String.valueOf(userId)});

        if (priceCursor.moveToFirst()) {
            for (int i = 0; i < priceCursor.getColumnCount(); i++) {
                totalPrice += priceCursor.isNull(i) ? 0 :
priceCursor.getInt(i);
            }
        }
        priceCursor.close();

        Cursor accessoriesCursor = db.rawQuery("SELECT a.price " +
"FROM accessories a " +
"JOIN user_accessories ua ON a.id = ua.accessory_id " +
"WHERE ua.user_id = ?", new
String[]{String.valueOf(userId)});
        if (accessoriesCursor.moveToFirst()) {
            do {
                totalPrice += accessoriesCursor.getInt(0);
            } while (accessoriesCursor.moveToNext());
        }
        accessoriesCursor.close();

        new android.app.AlertDialog.Builder(this)
            .setTitle("Готово!")
            .setMessage("☑ Конфігурація завершена.\nЗагальна вартість:
" + totalPrice + " грн")

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78

```

        .setPositiveButton("OK", (dialog, which) ->
dialog.dismiss())
        .show();
    }

    db.close();
}

private void loadComponent(String title, String table, int
id, String columns, int iconRes, String[] updateKeys) {
    if (id <= 0) return;

    layout.addView(createHeader(title));

    SQLiteDatabase db = dbHelper.getReadableDatabase();
    Cursor cursor = db.rawQuery("SELECT " + columns + " FROM " +
table + " WHERE id = ?",
    new String[]{String.valueOf(id)});

    if (cursor.moveToFirst()) {
        String name = cursor.getString(0);
        String spec1 = cursor.getString(1);
        String spec2 = cursor.getString(2);
        String spec3 = cursor.getString(3);
        int price = cursor.getInt(cursor.getColumnCount() - 1);

        LinearLayout item = createItem(title, name, spec1 + " | " +
spec2 + " | " + spec3, price, iconRes);
        item.addView(createDeleteButton(() -> {
            ContentValues values = new ContentValues();
            for (String key : updateKeys) {
                values.putNull(key);
            }
            SQLiteDatabase writeDb = dbHelper.getWritableDatabase();
            writeDb.update("user_config", values, "user_id = ?", new
String[]{String.valueOf(userId)});
            Toast.makeText(this, title + " видалено",
Toast.LENGTH_SHORT).show();
            recreate();
        }));
        layout.addView(item);
    }
    cursor.close();
}

private void loadAccessories() {
    SQLiteDatabase db = dbHelper.getReadableDatabase();
    Cursor cursor = db.rawQuery("SELECT a.id, a.name, a.type,
a.feature1, a.feature2, a.feature3, a.price " +
"FROM accessories a JOIN user_accessories ua ON a.id =
ua.accessory_id " +
"WHERE ua.user_id = ? ORDER BY a.type", new
String[]{String.valueOf(userId)});

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

```

        if (cursor.moveToFirst()) {
            hasAccessories = true;
            layout.addView(createHeader("Аксессуары"));
            do {
                int id = cursor.getInt(0);
                String name = cursor.getString(1);
                String type = cursor.getString(2);
                String f1 = cursor.getString(3);
                String f2 = cursor.getString(4);
                String f3 = cursor.getString(5);
                int price = cursor.getInt(6);

                int icon = R.drawable.ic_accessories;
                if (type.equalsIgnoreCase("keyboard")) icon =
R.drawable.ic_keyboard;
                else if (type.equalsIgnoreCase("mouse")) icon =
R.drawable.ic_mouse;
                else if (type.equalsIgnoreCase("monitor")) icon =
R.drawable.ic_monitor;

                LinearLayout item = createItem(type, name, f1 + " | " + f2 +
" | " + f3, price, icon);
                item.addView(createDeleteButton(() -> {
                    SQLiteDatabase wdb = dbHelper.getWritableDatabase();
                    wdb.delete("user_accessories", "user_id = ? AND accessory_id
= ?", new String[]{String.valueOf(userId), String.valueOf(id)});
                    Toast.makeText(this, "Аксессуар удалено",
Toast.LENGTH_SHORT).show();
                    recreate();
                }));

                layout.addView(item);
            } while (cursor.moveToNext());
        }
        cursor.close();
    }

    private LinearLayout createItem(String title, String name,
String specs, int price, int iconRes) {
        LinearLayout layout = new LinearLayout(this);
        layout.setOrientation(LinearLayout.HORIZONTAL);
        layout.setPadding(16, 16, 16, 16);
        layout.setGravity(Gravity.CENTER_VERTICAL);
        LinearLayout.LayoutParams params = new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.WRAP_CONTENT);
        params.setMargins(0, 0, 0, 16);
        layout.setLayoutParams(params);

        ImageView imageView = new ImageView(this);
        imageView.setImageResource(iconRes);
        imageView.setLayoutParams(new LinearLayout.LayoutParams(100,
100));
        layout.addView(imageView);
    }

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		80

```

        LinearLayout textLayout = new LinearLayout(this);
        textLayout.setOrientation(LinearLayout.VERTICAL);
        textLayout.setPadding(16, 0, 0, 0);
        textLayout.setLayoutParams(new LinearLayout.LayoutParams(0,
        ViewGroup.LayoutParams.WRAP_CONTENT, 1));

        TextView nameView = new TextView(this);
        nameView.setText(name);
        nameView.setTextColor(getResources().getColor(R.color.text_w
        hite));
        nameView.setTextSize(18);
        textLayout.addView(nameView);

        TextView specsView = new TextView(this);
        specsView.setText(specs + " | Ціна: " + price + " грн");
        specsView.setTextColor(getResources().getColor(R.color.text_
        white));
        specsView.setTextSize(14);
        textLayout.addView(specsView);

        layout.addView(textLayout);
        return layout;
    }

    private LinearLayout createHeader(String title) {
        LinearLayout headerLayout = new LinearLayout(this);
        headerLayout.setLayoutParams(new
        LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
        ViewGroup.LayoutParams.WRAP_CONTENT));
        headerLayout.setPadding(0, 24, 0, 12);
        TextView textView = new TextView(this);
        textView.setText(title);
        textView.setTextColor(getResources().getColor(R.color.text_w
        hite));
        textView.setTextSize(22);
        textView.setTypeface(null, Typeface.BOLD);
        headerLayout.addView(textView);

        return headerLayout;
    }

    private Button createDeleteButton(Runnable action) {
        Button button = new Button(this);
        button.setText("-");
        button.setTextColor(Color.RED);
        button.setTextSize(28f);
        button.setBackgroundColor(Color.TRANSPARENT);
        button.setPadding(24, 12, 24, 12);
        button.setOnClickListener(v -> action.run());
        return button;
    }
}

```

					КР.КН 25.584.03.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		81