**Міністерство освіти і науки України**

**ВСП «ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ**

**ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»**

Циклова комісія математичних дисциплін та інформаційних технологій

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему:«РОЗРОБКА ТА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПЕРЕВІРКИ І ОЦІНЮВАННЯ ЗНАНЬ»

|  |  |
| --- | --- |
| Виконав: | здобувач освіти 4 курсу, групи К121-20 |
| Спеціальність | 121 Iнженерiя |
|  | програмного забезпечення |
|  | (шифр i спеціальність) |
|  | Володимир КОРОТЕНКО |
| (ім’я та ПРIЗВИЩE) | |
| Керівник | Альона ХОДАКОВСЬКА |
|  | (ім’я та ПРIЗВИЩE) |
|  | |
| Рецензент | завідувач кафедри програмної інженерії ЗНУ, доцент, к.ф.-м.н., Андрій ЛІСНЯК |
|  | (посада, вчене звання, науковий ступінь, ім’я прізвище) |
|  | |

Запоріжжя

2024

|  |  |  |  |
| --- | --- | --- | --- |
| ЗAПOРIЗЬКИЙ НAЦIOНAЛЬНИЙ УНIВEРСИТEТ | | | |
|  | | | |
| ВСП «Економіко-правничий фаховий коледж ЗНУ» | | | |
| Освітньо-професійний ступінь | | | фаховий молодший бакалавр |
| Спеціальність | 121 Iнженерiя програмного забезпечення | | |
|  | | (шифр i назва) | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ЗAТВEРДЖУЮ**  Голова циклової комісії математичних дисциплін та інформаційних технологій | | | | | |
|  | | | | Тетяна СМОЛЯНКОВА | |
| (підпис) | | | |  | |
|  | | | | | |
| “ | 14 | ” | червня | | 2024 р. |

# ЗAВДAННЯ НA КВАЛІФІКАЦІЙНУ РОБОТУ СТУДEНТУ

|  |
| --- |
| КОРОТЕНКА Володимира Володимировича |

(прізвище, ім’я, по батькові)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Тема роботи | Розробка та реалізація мобільного додатку для перевірки і | | | | | | | |
| оцінювання знань | | | | | | | | |
|  | | | | | | | | |
| Керівник роботи | PhD, ХОДАКОВСЬКА Альона Валеріївна | | | | | | | |
|  | (науковий ступінь, вчене звання, прізвище, ім’я, по батькові) | | | | | | | |
|  | | | | | | | | |
| Затверджено наказом ВСП EПФК ЗНУ від | | | « | 30» | | листопада | 2023 р | № 2004-с |  |
|  | | | | | | | | |
| 2. Строк подання студентом роботи | | | | | 01.06.2024 | | | |
|  | | | | | | | | |
| 3. Вихідні дані до роботи | | 1. Постановка задачі. | | | | | | |
| 2. Перелік літератури. | | | | | | | | |
|  | | | | | | | | |
| 4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | | | | | | | | |  | |
| 1.Сучасні інформаційні системи | | | | | | | | |
| 2. Розробка проєкту інформаційної системи | | | | | | | | |
| 3. Програмна реалізація інформаційної системи | | | | | | | | |
| 4. Перелік графічного матеріалу | | | | | | | |  |
| презентація до захисту | | | | | | | | |

6. Кoнсyльтaнти рoздiлiв роботи

|  |  |  |  |
| --- | --- | --- | --- |
| **Рoздiл** | **Прiзвищe, iнiцiaли тa пoсaдa кoнсyльтaнтa** | **Пiдпис, дaтa** | |
| **Зaвдaння видaв** | **Завдання прийняв** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| 7. Дaтa видaчi зaвдaння | 01.12.2023 |

**КAЛEНДAРНИЙ ПЛAН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Нaзвa eтaпiв кваліфікаційної роботи** | **Стрoк викoнaння eтaпiв роботи** | **Примiткa** |
| 1. | Рoзрoбкa плaнy рoбoти: |  |  |
|  | Пoстaнoвкa зaдaчi | Грудень 2023 | викoнaнo |
| 2. | Збiр вихiдних дaних, oбрoбкa мeтoдичних |  |  |
|  | тa тeoрeтичних джeрeл | Сiчень 2024 | викoнaнo |
| 3. | Рoзрoбкa пeршoгo рoздiлy: |  |  |
|  | Огляд інформаційних систем та технологій для їхнього створення | Лютий 2024 | викoнaнo |
| 4. | Рoзрoбкa дрyгoгo рoздiлy: |  |  |
|  | Розробка проєкту інформаційної системи та вибір технологій | Квiтeнь 2024 | викoнaнo |
| 5. | Рoзрoбкa трeтьoгo рoздiлy: |  |  |
|  | Рoзрoбкa застосунку | Трaвeнь 2024 | викoнaнo |
| 6. | Oфoрмлeння i нoрмoкoнтрoль кваліфікаційної роботи тa пeрeвiркa нa плaгiaт | Червень 2024 | викoнaнo |
| 7. | Зaхист кваліфікаційної роботи | 21.06.2024 | викoнaнo |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Здобувач освіти | |  |  | Володимир КОРОТЕНКО |
|  |  | (пiдпис) |  | (ім’я ПРIЗВИЩE) |
|  |  | | | |
| Керівник роботи | |  |  | Альона ХОДАКОВСЬКА |
|  |  | (пiдпис) |  | (ім’я ПРIЗВИЩE) |

**Нoрмoкoнтрoль прoйдeнo**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Нормоконтролер | |  |  | Юлія БOРИСOВСЬКA |
|  |  | (пiдпис) |  | (ім’я ПРIЗВИЩE) |

# РЕФЕРАТ

Кваліфікаційна робота: 44 сторінки, 7 рисунків, 9 джерел.

Об’єкт дослідження: процес розробки, впровадження та використання мобільних додатків для перевірки і оцінювання знань в освітньому процесі.

Предмет дослідження: мобільний додаток для перевірки і оцінювання знань.

Мета роботи: розробка та реалізація мобільного додатка, який надасть можливість перевіряти та оцінювати знання у певній предметній області.

Метод дослідження: аналітичний, експериментальний, порівняльний аналіз.

У кваліфікаційній роботі проведено детальний аналіз та розроблено цілісний процес створення мобільного додатка. Використовуючи теоретичні дані процесу розробки додатків для Android, а також мову програмування Java та мову макетів XML, було розроблено функціональний додаток з назвою «Turbo НМТ».

В рамках дослідження було ретельно спроєктовано та реалізовано інтерфейс додатка, що повністю задовольняє вимоги користувачів та забезпечує ефективну взаємодію з мовою програмування та мовою макетів. З використанням інтегрованого середовища розробки Android Studio, мови програмування Java та мови макетів XML було успішно створено додаток «Turbo НМТ».

Цей додаток є результатом глибокого вивчення процесу розробки Android-додатків, а його функціональність спрямована на надання користувачам можливості вдосконалювати свої знання з дисциплін Національного мультипредметного тесту. Робота демонструє не лише знання теоретичних аспектів розробки мобільних додатків, але й вміння їх успішно реалізувати в практичному застосуванні.

У результаті проведеного дослідження було досягнуто поставленої

мети, а саме розроблено та реалізовано мобільний додаток, який забезпечує перевірку та оцінювання знань користувачів. Цей додаток може бути використаний для підвищення ефективності освітнього процесу та поліпшення рівня знань у відповідній предметній області.

ANDROID STUDIO, JAVA, МОВА МАКЕТІВ XML, MATERIAL DESIGN, НМТ.

# SUMMARY

Qualification Work: 44 pages, 7 figures, 9 references.

Object of Study: The process of developing, implementing, and using mobile applications for knowledge testing and assessment in the educational process.

Subject of Study: Mobile application for knowledge testing and assessment.

Purpose of the Work: To develop and implement a mobile application that enables the testing and assessment of knowledge in a specific subject area.

Research Methods: Analytical, experimental, comparative analysis.

In the qualification work, a detailed analysis and comprehensive process for creating a mobile application were conducted. Utilizing theoretical data on the application development process for Android, as well as the Java programming language and XML layout language, a functional application named "Turbo HMT" was developed.

As part of the study, the interface of the application was carefully designed and implemented to fully meet user requirements and ensure effective interaction with the programming and layout languages. Using the integrated development environment Android Studio, the Java programming language, and the XML layout language, the "Turbo HMT" application was successfully created.

This application is the result of an in-depth study of the Android application development process, and its functionality is aimed at providing users with the opportunity to improve their knowledge in the subjects of the National Multisubject Test. The work demonstrates not only the theoretical knowledge of mobile application development but also the ability to successfully implement it in practical use.

As a result of the research, the objective was achieved: a mobile application that ensures the testing and assessment of users' knowledge was developed and implemented. This application can be used to enhance the efficiency of the

educational process and improve the level of knowledge in the relevant subject area.

ANDROID STUDIO, JAVA, XML LAYOUT LANGUAGE, MATERIAL DESIGN, NMT.

**ЗМІСТ**

[Зaвдaння нa кваліфікаційну роботу студeнту 2](#_Toc167977894)

[Реферат 4](#_Toc167977895)

[Summary 6](#_Toc167977896)

[Вступ 9](#_Toc167977897)

[1 Основи розробки Android-додатку 11](#_Toc167977898)

[1.1 Поняття Android-додатку та їх класифікація 12](#_Toc167977899)

[1.2 Огляд існуючих аналогів 13](#_Toc167977900)

[1.3 Функціональні можливості існуючих рішень 13](#_Toc167977901)

[1.4 Висновок до розділу 1 14](#_Toc167977902)

[2 Проєктування мобільного додатка 16](#_Toc167977903)

[2.1 Визначення технічних вимог та інструментів розробки 16](#_Toc167977904)

[2.2 Проєктування інтерфейсу користувача та макетів інтерфейсу 19](#_Toc167977905)

[2.3 Проєктування функціональної структури додатка 22](#_Toc167977906)

[2.4 Висновок до розділу 2 24](#_Toc167977907)

[3 Розробка та реалізація мобільного додатка «Turbo НМТ» 25](#_Toc167977908)

[3.1 Порядок кроків та реалізація додатка «Turbo НМТ» 25](#_Toc167977909)

[3.2 Результати розробки та тестування додатку 37](#_Toc167977910)

[3.3 Висновок до розділу 3 41](#_Toc167977911)

[Висновки 42](#_Toc167977912)

[Перелік використаних джерел 44](#_Toc167977913)

# ВСТУП

Мобільні додатки стали невід’ємною частиною сучасного життя, забезпечуючи користувачів широким спектром можливостей: від спілкування та розваг до навчання та професійного розвитку. У зв’язку з цим, розробка мобільних додатків стає все більш актуальною та затребуваною сферою діяльності. Однією з ключових задач, що стоять перед розробниками, є створення зручних, ефективних та функціональних додатків, які б задовольняли потреби користувачів та відповідали сучасним стандартам якості.

Дана кваліфікаційна робота присвячена розробці мобільного додатку «Turbo НМТ», призначеного для підготовки користувачів до Національного мультипредметного тесту (далі – НМТ). Метою цього додатку є надання зручного та ефективного інструменту для самостійної підготовки до тестування з різних предметів, включаючи українську мову, англійську мову, математику та історію України.

Вступна частина роботи охоплює основні етапи та аспекти розробки мобільного додатку для платформи Android.

У першому розділі проводиться огляд існуючих інформаційних систем, технологій їх створення та можливостей використання для розробки мобільних додатків. Аналіз вимог користувачів, огляд аналогічних рішень та аналіз функцій. Важливим компонентом є визначення технічних вимог до додатку, а також функціональних та нефункціональних вимог.

Другий розділ розкриває процес проєктування додатку, етапи визначення технічних вимог, проєктування інтерфейсу користувача, вибір технологій розробки (таких як мова програмування Java, мова розмітки XML та інтегроване середовище розробки Android Studio) та функціональну структуру додатку. Проєктування інтерфейсу користувача охоплює створення зручного, привабливого та інтуїтивно зрозумілого інтерфейсу, який відповідає сучасним тенденціям дизайну та забезпечує зручність взаємодії з додатком. Функціональна структура додатку описує основні функціональні модулі, їх взаємодію та порядок виконання операцій, а також способи збереження та обробки даних.

У третьому розділі представлена програмна реалізація мобільного застосунку «Turbo НМТ». Описується процес розробки програмного забезпечення, включаючи реалізацію інтерфейсу, обробку даних, інтеграцію з базами даних та інші аспекти. Описується тестування та можливості вдосконалення додатку в майбутньому.

У висновку підводяться підсумки виконаної роботи. Результатом цієї кваліфікаційної роботи є розробка мобільного додатку «Turbo НМТ», який дозволяє користувачам ефективно підготуватися до НМТ, надаючи зручний та функціональний інструмент для самостійного навчання. Додаток реалізовано на основі сучасних технологій та інструментів, що забезпечує його високу продуктивність, стабільність та зручність використання.

У підсумку робота демонструє повний цикл створення додатку від визначення вимог до реалізації та тестування, а також забезпечує корисний інструмент для широкої аудиторії користувачів.

# 1 ОСНОВИ РОЗРОБКИ ANDROID-ДОДАТКУ

Основи розробки Android-додатку зосереджуються на ключових аспектах створення мобільних застосунків для платформи Android, які включають розгляд понять, класифікацію, аналіз потреб користувачів, аналіз конкурентів та огляд засобів розробки, що використовуються у процесі розробки Android-додатків.

Поняття Android-додатку та їх класифікація представлені в області мобільної розробки, розглядаючи сутність додатків для платформи Android та їх класифікацію за різними критеріями, такими як тип, функціональність та призначення. Дослідження інформаційних потреб у системі мобільного застосунку розглядає процес аналізу потреб користувачів та визначення вимог до мобільного додатку, що є важливим етапом перед розробкою. Для розуміння аспектів та подальшої розробки конкретного додатку з тестування потрібно зробити декілька кроків.

Першим кроком є проведення детального огляду аналогів майбутнього додатку, що включає визначення їх основних функціональних можливостей. Розуміння потреб цільової аудиторії та очікувань користувачів дозволить створити додаток, який буде відповідати їхнім потребам.

Другим кроком є визначення кращих практик та уникнення поширених помилок при розробці нового додатку.

Аналіз цих аспектів дозволяє створити додаток з широким спектром функцій, які підвищують зручність та ефективність його використання.

Огляд основних аспектів розробки Android-додатку допоміг зрозуміти процес мобільної розробки та її подальшого практичного застосування, а також перші кроки для проєктування застосунку.

## 1.1 Поняття Android-додатку та їх класифікація

Поняття Android-додатку охоплює програмне забезпечення, призначене для використання на пристроях, що працюють під управлінням операційної системи Android. Ці додатки можуть забезпечувати різноманітні функції, від ігор та розважальних додатків до корисних інструментів для роботи, освіти та бізнесу. Мобільні додатки для перевірки та оцінювання знань стають все більш популярними серед користувачів різного віку та рівня освіти. Ці додатки надають користувачам можливість ефективно готуватися до тестування, пропонуючи інтерактивні методи навчання та перевірки знань.

Класифікація Android-додатків може бути здійснена за кількома критеріями:

1. За функціональністю:

* ігри та розважальні додатки;
* програми для навчання та освіти;
* додатки для роботи та продуктивності;
* соціальні мережі та комунікації;
* утиліти та інструменти.

1. За метою використання:

* особисті додатки;
* корпоративні додатки;
* додатки для розваг та розвитку.

1. За способом розповсюдження:

* додатки з Google Play Store;
* сторонні додатки, завантажені з інших джерел.

1. За типом доступу:

* безкоштовні додатки;
* платні додатки;
* додатки з внутрішніми покупками.

Розуміння цих понять і класифікація додатків допомагає розробникам та користувачам краще розуміти різноманітність та функціональність Android-додатків.

## 1.2 Огляд існуючих аналогів

Аналіз аналогів додатку дозволив вивчити та порівняти існуючі додатки схожих за функціональністю та цілям і які відповідають запитам користувачів. Для визначення найбільш ефективних та корисних підходів до розробки мобільного додатку «Turbo НМТ» було проведено детальний огляд існуючих програмних рішень у цій області. Спочатку було проведено аналіз ринку мобільних додатків для перевірки та оцінювання знань. Було переглянуто найпопулярніші додатки на платформі Android такі як «Просте ЗНО» та «Тести ЗНО і НМТ».

Було вивчено функціональні можливості таких додатків, методи навчання та оцінювання, а також способи адаптації під специфічні вимоги цих тестів.

## 1.3 Функціональні можливості існуючих рішень

Дослідження функціональних можливостей існуючих мобільних додатків для перевірки та оцінювання знань дозволило з’ясувати різноманітні аспекти, які важливі для користувача під час використання таких програм. Перш за все, були детально проаналізовані можливості створення тестів у різних форматах. Це включає різноманітні типи питань (текстові, зображення), можливість формулювання питань у різних стилістиках та складностях.

Також були вивчені можливості категоризації предметів та тестів, що дозволяє користувачам обрати конкретний предмет. Різноманітність тем по кожному предмету забезпечується різноманіттям питань у базі даних додатка. Особлива увага приділена методу рандомізації вибірки питань для кожного нового тесту, що забезпечує унікальність кожного тестування та мінімізує можливість запам’ятовування правильних відповідей.

Іншим критичним аспектом є наявність функції обмеження часу на виконання тесту, що дозволяє моделювати реальні умови тестування на НМТ та підвищує дисциплінованість користувачів. Важливою є також функція отримання результатів тестування після кожного тесту, що дозволяє користувачам відразу дізнаватися свій результат та визначати слабкі місця в знаннях.

Завдяки детальному аналізу стали зрозумілими очікування користувачів від додатку. Та сформульовано конкретні вимоги до функціональності мобільного додатку «Turbo НМТ». Це дозволило визначити ключові функції, які необхідно реалізувати для створення зручного та ефективного інструменту для перевірки та оцінювання знань.

## 1.4 Висновок до розділу 1

Розуміння основ розробки Android-додатків є важливим етапом у створенні мобільних застосунків. Цей розділ розглядає ключові поняття та класифікацію Android-додатків, огляд аналогів - «Просте ЗНО» та «Тести ЗНО і НМТ», та їхніх функціональних можливостей. Аналіз всіх цих пунктів дає розуміння очікувань користувачів від додатку.

Огляд існуючих аналогів дозволяє визначити найкращі практики, оцінити їхні переваги та недоліки, а також уникнути типових помилок під час розробки. Особлива увага приділяється функціональним можливостям додатків для підготовки до НМТ, таким як різноманітні типи питань, категоризація предметів, рандомізація тестів, обмеження часу та відображення результатів.

Ретельний аналіз цих аспектів та розуміння очікувань користувачів створює міцну основу для розробки ефективного мобільного додатку «Turbo НМТ», який відповідатиме потребам вступників та забезпечуватиме якісну підготовку до НМТ.

# 2 ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКА

Проєктування мобільного додатка – це важливий етап у розробці програмного забезпечення, що передбачає створення зручного та ефективного інтерфейсу користувача та розробку функціональних можливостей, що відповідають потребам цільової аудиторії. У цьому розділі буде детально розглянуто процес проєктування мобільного додатка «Turbo НМТ», визначено технічні вимоги та інструменти розробки.

Основні етапи та ключові аспекти процесу розробки мобільного застосунку для платформи Android включають проєктування вебзастосунку та створення макету додатку. Обговорення проєкту вебзастосунку передбачає визначення структури додатку, його інтерфейсу, функціональності та взаємодії з користувачем. Проєктування дозволяє заздалегідь спланувати архітектуру додатку щоб вона була зручна у використанні та відповідала очікуванням користувачів.

## 2.1 Визначення технічних вимог та інструментів розробки

Визначення технічних вимог у процесі розробки мобільного додатка «Turbo НМТ» включає в себе конкретизацію технічних параметрів та функціональних вимог, необхідних для успішної реалізації проєкту. Ця частина проєкту є ключовою, оскільки визначає базові параметри, на основі яких проводиться подальший процес розробки.

1. Вибір інструментів розробки.

Мова програмування Java**:**

Мобільний додаток «Turbo НМТ» створено на мові програмування Java, яка є однією з найпопулярніших мов для розробки мобільних додатків під Android-платформи. Java говорить з Android на рідній мові, має велике співтовариство розробників, а також багато готових бібліотек та ресурсів для спрощення розробки. Java забезпечує високий рівень сумісності та стабільності з платформою Android [1].

Мова макетів XML:

З Java треба обов’язково використовувати мову макетів XML для створення інтерфейсу користувача, що є стандартом для розробки додатків під Android-платформу. XML забезпечує чітку і зрозумілу структуру для опису компонентів інтерфейсу, таких як кнопки, текстові поля та інші елементи. Використання XML дозволяє відокремити логіку додатка від його візуальної частини, що спрощує розробку та підтримку. Крім того, XML-файли легко змінювати та підтримувати, що робить їх незамінним інструментом для розробників Android-додатків [2].

Поєднання Java та XML в розробці Android забезпечує високий рівень продуктивності, зручності розробки та гарантує безпеку та читабельність коду. Ця комбінація є стандартом галузі та забезпечує розробників всіма необхідними інструментами для створення сучасних та ефективних мобільних додатків [3].

Інтегроване середовище розробки Android Studio:

Android Studio є офіційним інтегрованим середовищем розробки (IDE) для платформи Android, рекомендованим компанією Google. Використання офіційного інструменту сприяє зручності в розробці, тестуванні та розгортанні додатків. Середовище об’єднує всі необхідні елементи для створення Android-додатків на Java. Вершина досконалості в Android   
Studio – це емулятор Android. Це не просто емуляція програми, це багатогранний інструмент, який трансформує стек коду в живий досвід. Емулятор відтворює безліч пристроїв з різними характеристиками, дозволяючи розробнику тестувати свою програму в різних сценаріях. Це не просто вікно з екраном, це віртуальна лабораторія, де ідеї проходять вогонь та воду реального світу. Також прекрасно реалізовано інструменти для обробки подій та відлагоджування. Інтегровані відлагоджувачі дозволяють розробнику стежити за кожним кроком коду, розкриваючи таємниці його виконання та виявляючи можливі помилки. Android Studio має вбудовану підтримку «Material Design», яка полегшує створення сучасного, приємного та зручного інтерфейсу користувача.

Ці інструменти розробки дозволяють розробити додаток, та забезпечити широкі можливості розширення функціоналу та вдосконалення додатку в майбутньому [4].

1. Технічні вимоги до інтерфейсу користувача та функціональності.

Інтерфейс користувача:

Стиль «Material Design» дозволяє створювати сучасний, легкий для сприйняття, привабливий та зручний у використанні інтерфейс.

Елементи управління: Розташування кнопок, текстових полів, списків та інших елементів управління на екранах додатка повинно бути інтуїтивно зрозумілим та зручним для користувачів [5].

Функціональні можливості:

Тести з різних предметів. Додаток надає можливість користувачам проходити тести з різних предметів, таких як українська мова, англійська мова, математика та історія України. Користувач має можливість вибрати тест, проходити його та отримати результати.

Випадковість вибірки питань з бази даних. Ця функція покращує ефективність навчання та користувацький досвід, забезпечуючи різноманітність завдань і допомагаючи користувачам більш ґрунтовно покращувати свої знання.

Таймер. Вбудований таймер, який відлічує час, відведений на проходження тесту, та автоматично завершує тест після закінчення цього часу. Це дозволяє користувачам ефективно контролювати свій час під час виконання завдань.

Результати тестів. Після завершення тесту користувач отримує результати, які включають кількість правильних відповідей та оцінку. Результати можуть бути збережені для подальшого аналізу та відстеження прогресу.

Мінімальні вимоги до операційної системи: для забезпечення коректної роботи додатка «Turbo НМТ» на різних пристроях, необхідно визначити мінімальні вимоги до операційної системи. Додаток повинен підтримуватися на всіх сучасних версіях Android, починаючи з Android 7.0 та вище. Це забезпечить широкий охоплення аудиторії користувачів та сумісність з більшістю мобільних пристроїв.

Визначення цих технічних вимог є основою для подальших етапів проєктування та розробки мобільного додатка «Turbo НМТ». Точне дотримання цих вимог забезпечить створення функціонального, зручного та стабільного додатка, який відповідатиме потребам користувачів.

## 2.2 Проєктування інтерфейсу користувача та макетів інтерфейсу

Проєктування інтерфейсу користувача в контексті розробки мобільного додатку «Turbo НМТ» охоплює процес створення зручного, привабливого та легко зрозумілого інтерфейсу для користувачів. Це ключовий етап розробки, оскільки інтерфейс користувача визначає враження та зручність взаємодії з додатком. Макет інтерфейсу додатку - це візуальна концепція, що відображає вигляд та організацію елементів програмного інтерфейсу на мобільному пристрої. Це детальний план того, як користувач буде взаємодіяти з додатком, включаючи розташування кнопок, меню, текстових полів та інших елементів.

Макет інтерфейсу допомагає забезпечити однорідність та спростити розробку додатку, оскільки він дає змогу візуально спланувати всі аспекти користувацького досвіду перед початком програмування. Крім того, макет дозволяє провести тестування інтерфейсу з потенційними користувачами та внести необхідні зміни ще до завершення розробки додатку [6].

Основні аспекти проєктування інтерфейсу включають:

1. Дизайн і стиль інтерфейсу.

Визначення загального вигляду та стилю інтерфейсу додатку**:**

Для додатку «Turbo НМТ» обрано дизайн у стилі «Material Design», який відповідає сучасним тенденціям і забезпечує зручність взаємодії. «Material Design» надає чіткі правила та інструменти для створення візуально привабливих і функціональних інтерфейсів. Це легкий для сприйняття дизайн з приємними кольорами, який дозволяє зрозуміти, як користуватися додатком, з першого погляду. Інтерфейс має бути візуально привабливим та зручним для користувачів, що включає використання чіткого шрифту, зручного розташування елементів дизайну. Привабливий інтерфейс сприяє кращому сприйняттю додатку користувачами та підвищує їх задоволеність від його використання. Він сприяє створенню естетично приємних і інтуїтивно зрозумілих додатків, які забезпечують позитивний користувацький досвід.

1. Розташування елементів інтерфейсу.

Організація та розміщення елементів на екрані додатку:

Розташування елементів інтерфейсу є критично важливим для зручності користування додатком. На головній сторінці «Turbo НМТ» розташовані іконки для вибору предмету тесту та кнопка почати, що спрощує навігацію для користувача. Важливо забезпечити, щоб усі елементи були розміщені логічно та інтуїтивно зрозуміло. Кнопки навігації мають бути легко доступними, а текст має бути читабельним на будь-якому пристрої.

1. Навігація.

Проєктування системи навігації в додатку:

Ефективна система навігації забезпечує зручний та логічний перехід між екранами та функціями додатка. У «Turbo НМТ» використовується проста навігація за допомогою іконок предметів тесту на головній сторінці, та можливість повернутися назад жестом. Користувачі повинні мати змогу легко знайти необхідні функції та переходити між різними розділами додатка без плутанини.

1. Взаємодія з елементами.

Розробка інтерфейсу, який дозволяє користувачам легко взаємодіяти з елементами додатку. Взаємодія з елементами інтерфейсу має бути простою та інтуїтивною. Наприклад, відображення підказок чи повідомлень при неправильному виборі варіанту відповіді під час проходження тесту допомагає користувачам зрозуміти свої помилки та виправити їх. Також важливо забезпечити швидкий відгук на дії користувача, що покращує загальне враження від використання додатку [7].

Після розгляду головних аспектів проєктування інтерфейсу були розроблені макети сторінок додатку, які показані на рисунку 2.1.

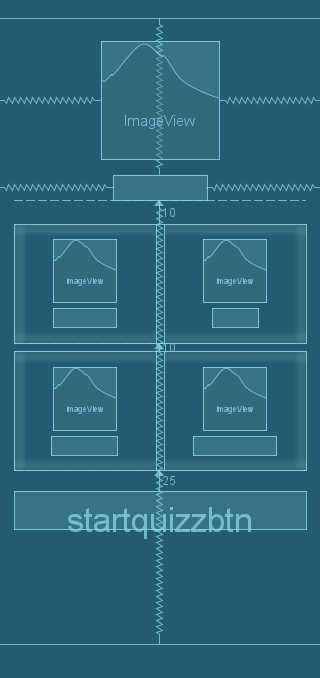


Рисунок 2.1 – Макети головних сторінок додатку

## 2.3 Проєктування функціональної структури додатка

Проєктування функціональної структури мобільного додатка «Turbo НМТ» включає в себе опис функціональних модулів, їх взаємодію, порядок виконання операцій, способи обробки даних, а також можливості розширення функціональності. Ця структура визначає набір функціональних можливостей, які реалізовані в додатку для забезпечення його коректної роботи та задоволення потреб користувачів.

Основні елементи функціональної структури «Turbo НМТ» включають:

1. Вибір предмету тестування.

Користувач може обрати предмет тестування з чотирьох доступних опцій:

* українська мова;
* історія України;
* математика;
* англійська мова.

Цей етап є першим кроком, де користувач обирає предмет, з якого хоче пройти тестування. Інтерфейс забезпечує зручний вибір та підсвічування обраного предмета щоб надати швидкий візуальний відгук для зручності користувача.

1. Кнопка «Почати тестування»:

Після натискання на кнопку відбувається перехід до проходження тесту лише у випадку, якщо користувач вже вибрав предмет тесту. У протилежному випадку, користувач отримає повідомлення про необхідність вибору предмету перед переходом до тестування.

1. Процес тестування:

Додаток надає користувачеві питання та можливість обирати варіант відповіді. Для кожної нової спроби випадковим чином обираються 20 питань які беруться з бази даних додатку. Тестовий інтерфейс відзначається не лише інтуїтивно зрозумілим дизайном, але й забезпечує швидкий та зручний процес тестування. При виборі правильної відповіді обраний варіант підсвічується зеленим кольором, в той час як при неправильній відповіді він виділяється червоним. Незалежно від рішення, після відповіді користувач має можливість переходу до наступного питання. Проте важливо зазначити, що перед переходом обов’язково потрібно обрати відповідь; у випадку її відсутності і спроби перейти далі, з’являється повідомлення, що необхідно надати відповідь, щоб перейти до наступного завдання.

1. Таймер:

Під час проходження тесту на екрані відображається таймер, що показує залишений час до завершення тестування. Цей таймер стає відмінним помічником користувачам у розподілі часу під час тестування. У випадку, якщо час закінчується до того, як користувач встигне відповісти на всі питання та натиснути кнопку «Закінчити тест», додаток автоматично завершить тестування після вичерпання відведеного часу. Всі тести, які користувач не встиг пройти, будуть враховані як неправильні.

1. Перегляд результатів:

Після завершення тесту додаток обробляє відповіді користувача, підраховує правильні та неправильні відповіді, і відображає результати. Алгоритм обробки враховує всі можливі типи питань і забезпечує точний підрахунок результатів. Користувач може переглянути результати свого проходження, включаючи кількість правильних та неправильних відповідей. Це дозволяє аналізувати свої помилки та готуватися до наступних тестів. Також на цій сторінці є кнопка повернення на головну сторінку.

1. Розширення функціоналу:

Крім того, можна сказати, що всі елементи додатку не лише обслуговують поточні потреби користувачів, а й мають потужний резерв для розширення функціональності та вдосконалення. Особливо легко можна розширити базу даних питань, що надає безмежні можливості для подальшого збагачення змісту та покращення досвіду користувачів.

## 2.4 Висновок до розділу 2

У цьому розділі було детально розглянуто процес проєктування мобільного додатка «Turbo НМТ». На початку були визначені технічні вимоги та інструменти для розробки додатка. Зокрема, обрано мову програмування Java, мову макетів XML та інтегроване середовище розробки Android Studio, що забезпечить зручність, сумісність та широкі можливості для вдосконалення додатку в майбутньому.

Наступним кроком стало проєктування інтерфейсу користувача та створення макетів інтерфейсу. Було обрано стиль «Material Design» для забезпечення сучасного, привабливого та зручного дизайну. Особлива увага приділялась розміщенню елементів інтерфейсу, навігації та взаємодії з користувачем. Розроблені макети головних сторінок додатку візуалізують концепцію інтерфейсу та допомагають спланувати користувацький досвід.

Окрім цього, було спроєктовано функціональну структуру додатка, що охоплює основні модулі та їх взаємодію. Детально описано порядок виконання операцій, починаючи від вибору предмета тестування, проходження тесту, відображення результатів та збереження прогресу користувача. Також передбачено можливість розширення функціональності додатка в майбутньому.

Загалом, ретельне проєктування є ключовим етапом у розробці успішного мобільного додатка. Цей розділ заклав міцну основу для подальшої реалізації «Turbo НМТ», забезпечуючи зручний та ефективний користувацький досвід, а також гнучкість для майбутніх оновлень і вдосконалень.

# 3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКА «TURBO НМТ»

Розробка Android-додатку є ключовим етапом у процесі створення мобільних програм для платформи Android. Цей етап включає конкретну імплементацію задуманої функціональності та дизайну, програмування, тестування та оптимізацію додатку для відповідності вимогам користувачів і стандартам розробки.

Під час розробки ми використовуємо інтегроване середовище розробки Android Studio, яке забезпечує зручний інтерфейс для програмування, налагодження та оптимізації додатків. Android-додатки розробляються мовою Java для створення функціональності та взаємодії зі складовими Android-платформи. Інтерфейс користувача створюється за допомогою XML-розмітки та ресурсів, що дозволяє чітко визначити вигляд та поведінку додатку.

Для реалізації різноманітних функцій ми використовуємо різні бібліотеки та інструменти, включаючи роботу з мережею, базами даних, мультимедійними ресурсами та іншими. Розробка Android-додатку вимагає великої уваги до деталей, а також ретельного тестування та оптимізації для забезпечення високої якості та продуктивності програми.

Цей розділ надає уявлення про процес розробки Android-додатку та ключові кроки, необхідні для створення успішної мобільної програми.

## 3.1 Порядок кроків та реалізація додатка «Turbo НМТ»

Спочатку було встановлено та запущено середовище розробки Android Studio.

Ми обрали пункт «Створити новий проєкт» та вибрали шаблон «Empty Views Activity».

Далі ми налаштували параметри додатку.

Опціонально визначили назву додатку та пакету, шлях до директорії.

Мову програмування обрали Java.

Мінімальну версію Android встановили на 7.0.

Це показано на рисунку 3.1.

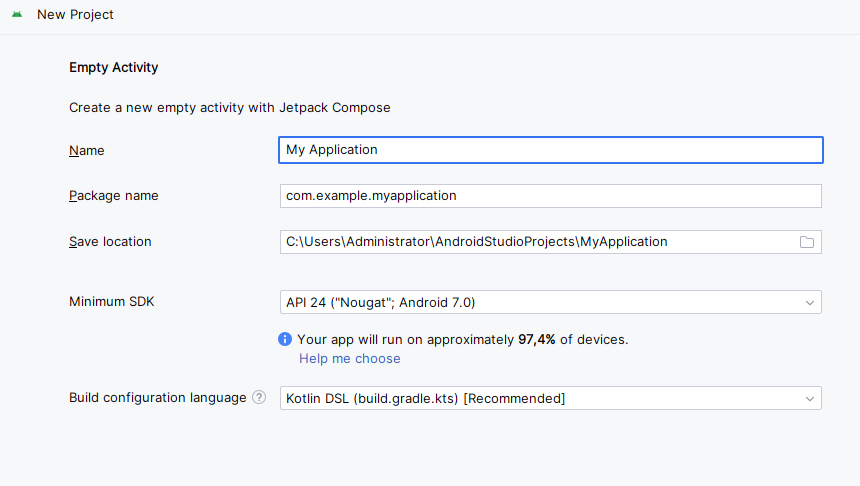


Рисунок 3.1 – Вікно параметрів проєкту

Ієрархія директорій створеного проєкту показано на рисунку 3.2.

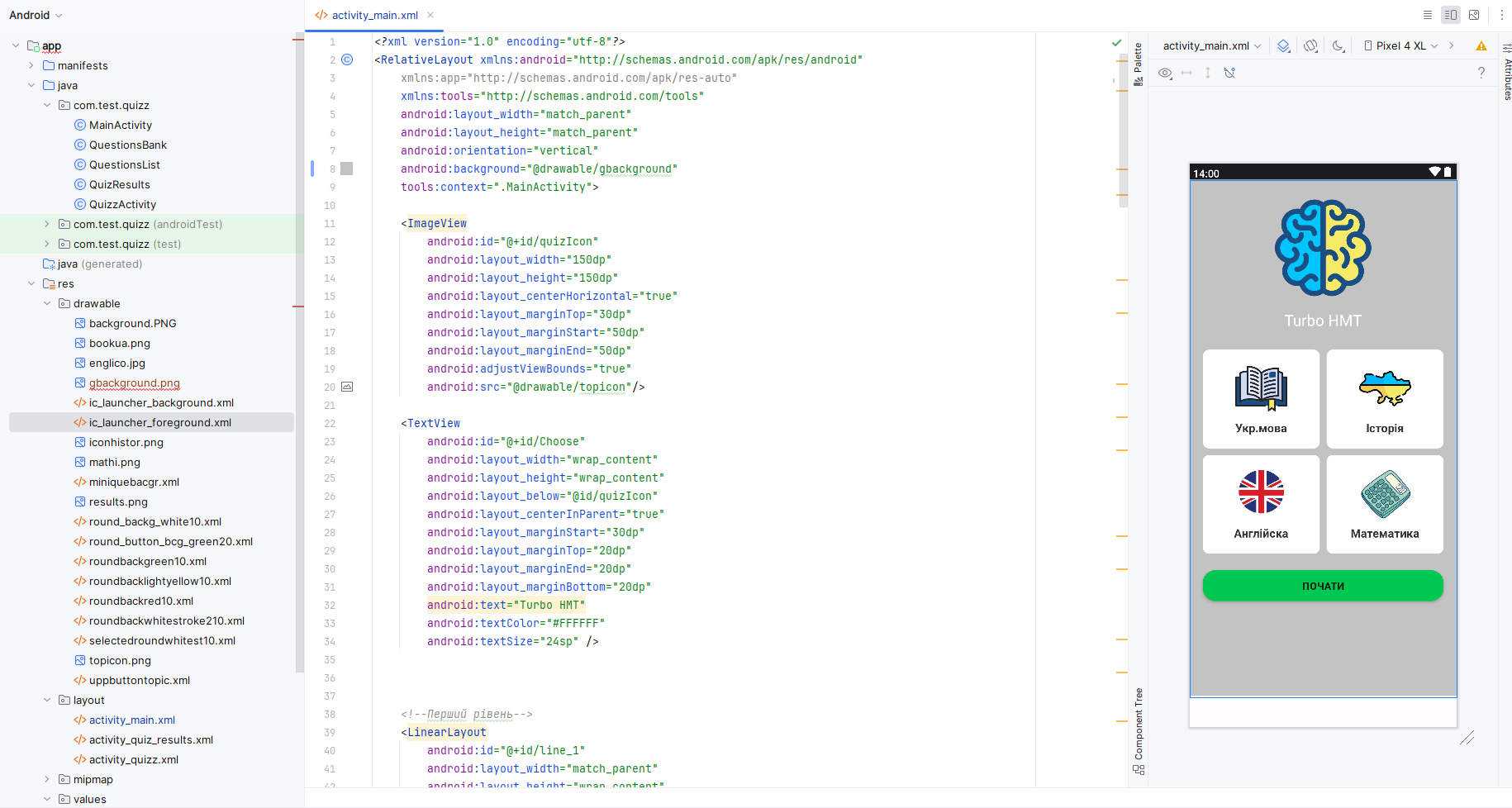


Рисунок 3.2 – Ієрархія директорій створеного проєкту

Використовуючи макети основних сторінок, які були спроєктовані раніше, починаємо реалізовувати інтерфейс.

У файлі проєкту activity\_main.xml змінюємо “<androidx.constraintlayout.widget.Constraint Layout” на “<LinearLayout”, а далі вже починаємо формувати сторінку за макетом [8].

Додамо XML-макет activity\_main.xml призначений для головної активності (MainActivity) додатка.

RelativeLayout:

* android:layout\_width="match\_parent" та аndroid: layout\_height="match\_parent" - визначають розміри родинного контейнера, які займають весь екран;
* android:orientation="vertical" - вказує вертикальну орієнтацію;
* android:background="@drawable/background" - встановлює фон зображенням.

ImageView (quizIcon):

* android:id="@+id/quizIcon" - унікальний ідентифікатор;
* android:layout\_width та android:layout\_height - розміри зображення;
* android:layout\_centerHorizontal="true" - розташовує зображення по горизонталі по центру;
* android:layout\_marginTop="30dp" - задає верхній відступ;
* android:src="@drawable/topicon" - встановлює зображення для ImageView.

TextView (Choose):

* android:id="@+id/Choose" - унікальний ідентифікатор;
* android:layout\_below="@id/quizIcon" - розташовує текст нижче зображення;
* android:layout\_centerInParent="true" - центрує текст відносно батьківського контейнера по вертикалі;
* android:text="Turbo НМТ" - встановлює текст.

Перший рівень (line\_1):

Дочірні елементи (movaLayout та historyLayout):

* два LinearLayout з ідентифікаторами movaLayout та historyLayout;
* кожен з них має свою іконку, текст та фон.

Другий рівень (line\_2):

Дочірні елементи (englLayout та matchLayout):

* два інші LinearLayout з ідентифікаторами englLayout та matchLayout;
* також містять іконки, тексти та фони.

Кнопка почати тест (startquizzbtn):

* android:id="@+id/startquizzbtn" - унікальний ідентифікатор;
* android:layout\_below="@id/line\_2" - розташовує кнопку нижче другого рівня;
* android:layout\_marginTop="25dp" - верхній відступ;
* android:text="Почати" - встановлює текст кнопки.

Результат головної сторінки показано на рисунку 3.3.

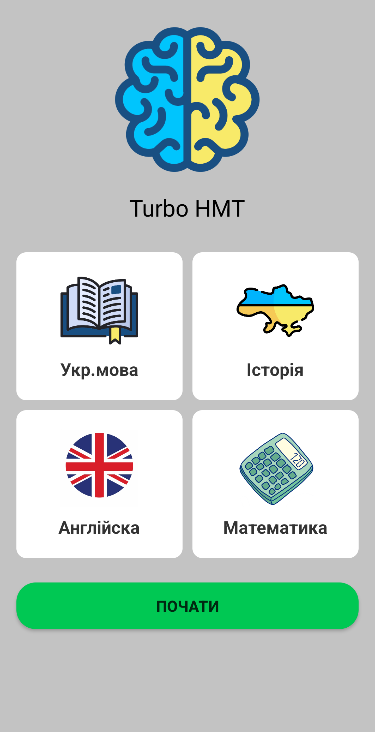


Рисунок 3.3 – Головна сторінка

Далі у MainActivity.java було написано код для забезпечення взаємодії елементів інтерфейсу та їх функціональності згідно з закладеним у макеті. [9]

У методі `onCreate()`:

* виклик `super.onCreate(savedInstanceState)`;
* встановлення макету активності `setContentView(R.layout.activity\_ main)`;
* встановлення кольору статус-бару за допомогою `getWindow().setStatusBarColor(ContextCompat.getColor(MainActivity.this, R.color.statbar))`.

Ініціалізація `LinearLayout` та `Button` за їх `id`:

* `mova`, `history`, `english`, `match` - для вибору теми тестування;
* `startquizzbtn` — для початку тестування.

Реалізовано обробники подій `onClick` для кожного `LinearLayout`, що змінюють обрану тему тесту та фонове забарвлення відповідного `Layout`:

* при натисканні на `mova`, `history`, `english`, або `match`, відповідно, змінюється значення змінної `selectedTopic` та фон вибраного елемента встановлюється на `selectedroundwhitest10`, а інші елементи скидаються на `round\_backg\_white10`.

Для кнопки «Почати тест» написано обробник, який:

* перевіряє, чи обрана тема;
* якщо так, створює `Intent` для переходу до `QuizzActivity`, передаючи обрану тему через `intent.putExtra("selectedTopic", selectedTopic)` і запускає нову активність;
* інакше виводить повідомлення про необхідність вибору теми за допомогою `Toast`.

Таким чином, забезпечено взаємодію елементів інтерфейсу, можливість вибору та запуску тесту з обраної теми відповідно до макета та логіки роботи додатку.

Потім було додано XML-макет `activity\_quizz.xml`, який визначає розмітку для активності квізу (QuizzActivity).

RelativeLayout:

Атрибути:

* `android:layout\_width="match\_parent"` та `android:layout\_height=" match\_parent"` встановлюють розміри контейнера, які займають весь екран;
* `android:background="#1C1B1F"` встановлює колір фону для всього макету;
* `tools:context=".QuizzActivity"` вказує на клас QuizzActivity для зв'язку з кодом.

Верхня панель (RelativeLayout):

Дочірні елементи:

* - `ImageView` з ідентифікатором `backbtn` представляє кнопку "Назад" зі зображенням;
* `textView` з ідентифікатором `selectedTopicName` відображає ім'я обраної теми.
* `textView` з ідентифікатором `timer` показує лічильник часу для кожного питання.

NestedScrollView та LinearLayout:

Атрибути NestedScrollView:

* `android:id="@+id/nestedScrollView"` має унікальний ідентифікатор;
* `android:layout\_marginTop="70dp"` забезпечує відступ для уникнення перекриття верхньої панелі.

Дочірні елементи LinearLayout:

* `textview` з ідентифікатором `questions` вказує на поточне питання у форматі "1/20";
* `textview` з ідентифікатором `question` відображає саме питання;
* чотири `AppCompatButton` (опції відповідей) з ідентифікаторами `option1`, `option2`, `option3`, `option4`;
* `AppCompatButton` з ідентифікатором `nextBtn` вказує на кнопку "Далі" для переходу до наступного питання.

Стилізація та ресурси:

* задано фони, кольори тексту та інші стилі для покращення зовнішнього вигляду інтерфейсу.

Таким чином, макет забезпечує зручне розташування елементів інтерфейсу для квізу, відповідно до функціональних вимог та дизайну додатку. Результат сторінки проходження тесту показано на рисунку 3.4.

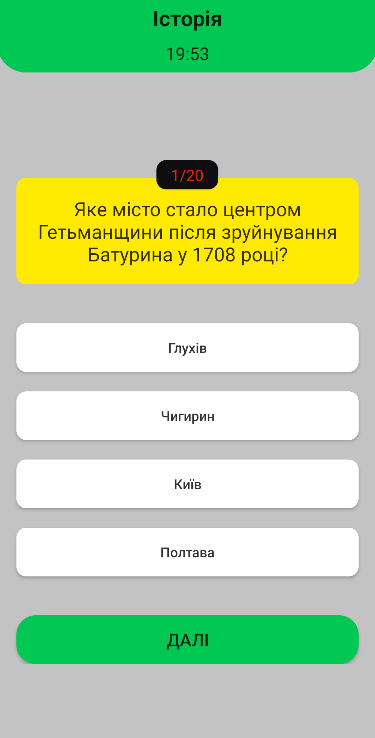


Рисунок 3.4 – Результат сторінки проходження тесту

Далі створили QuizzActivity.java та написали код. Він включає обробку елементів інтерфейсу, таймера, вибору відповідей користувачем та переходу між питаннями.

onCreate() Метод:

* виклик суперметоду: `super.onCreate(savedInstanceState)`.

Встановлення макету активності: `setContentView (R.layout.activity\_quizz)`.

Ініціалізація елементів інтерфейсу:

* imageview: `backBtn`;
* textview: `timer`, `selectedTopicName`, `questions`, `question`;
* appcompatbutton: `option1`, `option2`, `option3`, `option4`, `nextBtn`.

Отримання обраної теми:

* `getIntent().getStringExtra("selectedTopic")`.

Встановлення тексту обраної теми: `selectedTopic Name.setText(getSelectedTopic)`.

Отримання списку питань:

* `questionsList = QuestionsBank.getQuestions(getSelectedTopic)`;
* запуск таймера: `startTimer(timer)`.

Обробка натискань на кнопки:

* back Button: повернення до головної активності;
* option Buttons: вибір відповіді та виявлення правильної відповіді;
* next Button: перехід до наступного питання або завершення квізу.

Таймер:

* метод startTimer(): запускає таймер, який відраховує час до завершення квізу;
* оновлення часу: таймер оновлює текстове поле `timerTextView` кожну секунду.

Рандомізація питань: `Collections.shuffle(questionsList)` для перемішування питань.

Обробка відповідей:

* методи getCorrectAnswers() та getInCorrectAnswers(): підраховують кількість правильних та неправильних відповідей;
* метод revealAnswer(): виявляє правильну відповідь, змінюючи фон та колір тексту відповідної кнопки.

Перехід між питаннями:

* метод changeNextQuestion(): змінює питання, обнуляє вибір користувача, оновлює текст та стилі кнопок відповідей.

Таким чином, код забезпечує взаємодію користувача з квізом, включаючи вибір відповідей, обробку результатів та управління часом на кожне питання.

Далі було створено activity\_quiz\_results.xml, тобто код інтерфейсу результатів тесту

Код визначає макет `QuizResults` з наступними елементами:

* relativelayout: основний контейнер з фоном `@drawable /gbackground`;
* imageview: відображає іконку результатів (`@drawable/results`);
* textview (resultTexst): текст «Ваш Результат» великим шрифтом;
* linearlayout: вертикальний контейнер для текстів правильних та неправильних відповідей;
* textview (correctAnswers): показує кількість правильних відповідей;
* textview (incorrectAnswers): показує кількість неправильних відповідей;
* appcompatbutton (homeBtn): кнопка повернення до головного меню.

Результат сторінки результатів тесту показано на рисунку 3.5.



Рисунок 3.5 – Результат сторінки результатів тесту

До QuizResults.java було написано код для відображення результатів пройденого тесту та забезпечення переходів між активностями.

У методі onCreate():

* виклик super.onCreate(savedInstanceState) для ініціалізації базового класу;
* встановлення макету активності setContentView (R.layout.activity\_quiz\_results);
* встановлення кольору статус-бару;
* ініціалізація кнопки та текстових полів за їх ідентифікаторами;
* отримання даних з попередньої активності.

Також реалізовано:

* відображення кількості правильних/неправильних відповідей у текстових полях;
* обробник натискання кнопки «Home», який переходить до MainActivity;
* перехід до MainActivity при натисканні кнопки «Назад».

Таким чином забезпечено:

* відображення результатів тесту та переходи між активностями QuizResults та MainActivity відповідно до логіки роботи додатку;
* клас QuestionsList представляє собою модель для зберігання та обробки інформації про окреме питання в квізі. Даний клас виконує важливу функцію у структурі програми, забезпечуючи організацію та управління даними.

Ініціалізація даних через конструктор:

* клас має конструктор, який приймає всі необхідні дані для створення об'єкта питання (питання, варіанти відповідей, правильна відповідь та вибір користувача).

Отримання даних питання:

* публічні методи getOption1(), getOption2(), getOption3(), getOption4(), getQuestion(), getAnswer() надають доступ до відповідних полів класу, щоб отримати інформацію про питання та його варіанти.

Отримання та зміна вибору користувача:

* методи getUserSelectedAnswer() та setUserSelectedAnswer(String userSelectedAnswer) дозволяють отримати значення вибору користувача та встановити нове значення.

Далі було створено клас бази даних додатку QuestionsBank.java, який відповідає за створення та управління списками питань для різних тем у квізі.

Конструктор класу `QuestionsList`.

Конструктор класу `QuestionsList` приймає всі необхідні дані для створення об'єкта запитання:

* текст запитання (`question`);
* варіанти відповідей (`option1`, `option2`, `option3`, `option4`);
* правильну відповідь (`answer`);
* вибір користувача (`userSelectedAnswer`).

Колекції запитань за темами.

Методи для кожної теми повертають список запитань:

* `englishQuestions()` - список запитань з англійської мови;
* `historyQuestions()` - список запитань з історії;
* `matchQuestions()` - список запитань з математики;
* `movaQuestions()` - список запитань з української мови.

Отримання запитань за темою.

Метод `getQuestions(String selectedTopicName)`:

* повертає список запитань залежно від обраної теми.

Метод для випадкового вибору запитань.

Метод `getRandomQuestions(String selectedTopicName, int numberOfQuestions)`:

* отримує список запитань для обраної теми за допомогою `getQuestions()`;
* перемішує запитання за допомогою `Collections.shuffle()`;
* повертає підсписок з необхідною кількістю випадкових запитань.

Для кожної теми підготовлено 100 запитань, а також створені умови для зручного розширення їх кількості, щоб з часом вдосконалювати додаток.

Цей клас дозволяє зручно зберігати запитання для різних тем і забезпечує ефективний механізм для їх випадкового вибору.

Також для функціонування додатку було створено клас QuestionsList.java представляє собою модель для зберігання та обробки інформації про окреме питання в квізі. Даний клас виконує важливу функцію у структурі програми, забезпечуючи організацію та управління даними.

Опис коду `QuestionsList.java`.

Конструктор приймає всі необхідні дані для створення об’єкта питання:

* текст питання (`question`);
* варіанти відповідей (`option1`, `option2`, `option3`, `option4`);
* правильна відповідь (`answer`);
* вибір користувача (`userSelectedAnswer`).

Отримання даних питання.

Публічні методи надають доступ до відповідних полів класу:

* `getOption1()`, `getOption2()`, `getOption3()`, `getOption4()` - отримання варіантів відповідей;
* `getQuestion()` - отримання тексту питання;
* `getAnswer()` - отримання правильної відповіді.

Отримання та зміна вибору користувача.

Методи для роботи з вибором користувача:

* `getUserSelectedAnswer()` — отримання вибору користувача;
* `setUserSelectedAnswer(String userSelectedAnswer)` — встановлення вибору користувача.

Метод для випадкового вибору питань.

Метод `getRandomQuestions(List<QuestionsList> allQuestions, int numberOfQuestions)`:

* створює копію списку всіх питань;
* перемішує список за допомогою `Collections.shuffle()`;
* повертає підсписок з необхідною кількістю випадкових питань.

В цілому цей клас забезпечує зручне зберігання питань для квізу та ефективний механізм для їх випадкового вибору.

## 3.2 Результати розробки та тестування додатку

В цьому розділі представлені результати розробки мобільного додатку «Turbo НМТ» для перевірки і оцінювання знань.

Розроблено додаток, в якому користувач може обрати предмет тестування з чотирьох доступних опцій:

* українська мова;
* Історія України;
* математика;
* англійська мова.

Інтерфейс забезпечує зручний вибір та підсвічування обраного предмета, щоб надати швидкий візуальний відгук для зручності користувача. Після натискання на кнопку «Почати тестування» відбувається перехід до проходження тесту лише у випадку, якщо користувач вже вибрав предмет тесту. У протилежному випадку користувач отримає повідомлення про необхідність вибору предмету перед переходом до тестування.

В процесі тестування додаток надає користувачеві питання та можливість обирати варіант відповіді. Для кожної нової спроби випадковим чином обираються 20 питань, які беруться з бази даних додатку. Випадковість питань з великої бази даних забезпечує неповторність кожної спроби та вищу зацікавленість користувача. Тестовий інтерфейс забезпечує швидкий та зручний процес тестування. При виборі правильної відповіді обраний варіант підсвічується зеленим кольором, а при неправильній - червоним. Перед переходом до наступного питання обов’язково потрібно обрати відповідь; у випадку її відсутності і спроби перейти далі, з’являється повідомлення, що необхідно надати відповідь. Зверху є лічильник питань, а також під час проходження тесту на екрані відображається таймер, що показує залишений час до завершення тестування. У випадку, якщо час закінчується до того, як користувач встигне відповісти на всі питання та натиснути кнопку «Закінчити тест», додаток автоматично завершить тестування після вичерпання відведеного часу. Всі тести, які користувач не встиг пройти, будуть враховані як неправильні.

Після завершення тесту додаток обробляє відповіді користувача, підраховує правильні та неправильні відповіді і відображає результати. Користувач може переглянути результати свого проходження, включаючи кількість правильних та неправильних відповідей. Це дозволяє аналізувати свої помилки та готуватися до наступних тестів. Також на цій сторінці є кнопка повернення на головну сторінку.

Додаток було розроблено з врахуванням потужного резерву для розширення функціоналу та вдосконалення. Особливо легко можна розширити базу даних питань, що надає необмежені можливості для подальшого збагачення змісту та поліпшення досвіду користувачів.

У підсумку можна зазначити, що додаток «Turbo НМТ» забезпечує ефективний та зручний процес тестування та підвищує мотивацію до навчання.

Скриншоти фінального вигляду додатку наведено на рисунку 3.6.

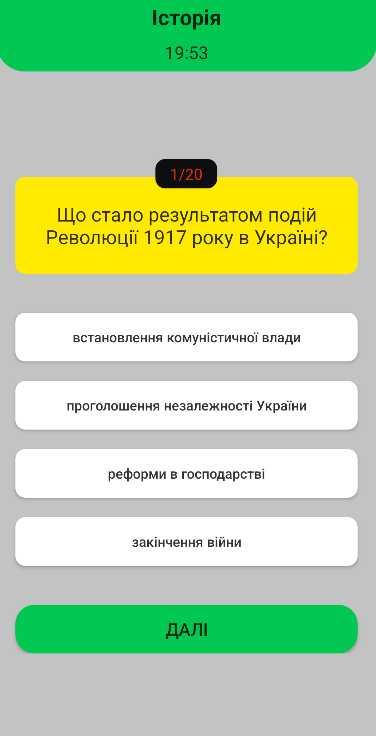
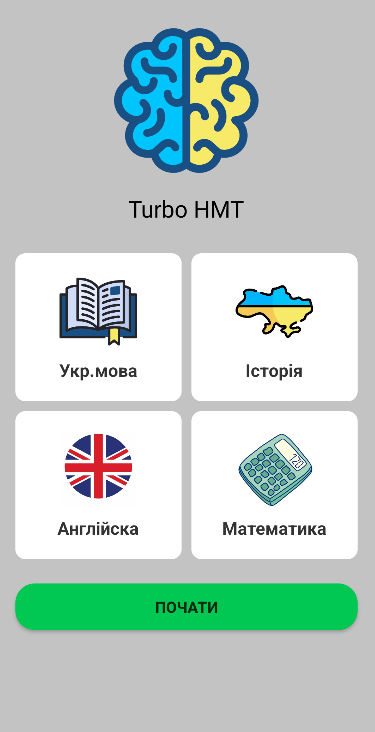


Рисунок 3.6 – Фінальний вигляд додатку

Також у результаті самостійного вивчення Java та XML у середовищі Android Studio отримані глибокі знання, які знаходять практичне застосування в розробці мобільних додатків.

Java:

* основи мови: засвоєння ключових концепцій, таких як змінні, умови, цикли, масиви, функції та колекції;
* ООП: розуміння та використання принципів об’єктно-орієнтованого програмування, таких як створення класів, спадкування, поліморфізм та інтерфейси;
* робота з бібліотеками: компетентне використання бібліотек та фреймворків для полегшення процесу розробки.

XML:

* створення документів: навички роботи з XML, включаючи створення документів та роботу з їх елементами;
* використання в проєкті: здатність інтегрувати XML для зберігання та передачі даних у проєкті.

Огляд інструментів:

* засвоєння інструментів: опановування різноманітних інструментів та технологій, які можна використовувати для розробки в Android Studio;
* загалом, здобуті не лише конкретні технічні знання Java та XML, а й поглиблене розуміння того, як їх застосовувати в реальних проєктах у Android Studio.

Професійний ріст:

* розробка та вдосконалення додатків: навички розробки та вдосконалення Android додатків.
* поглиблення технічних знань: поглиблене розуміння принципів програмування на Java та роботи з XML.
* практичний досвід: отримання практичного досвіду роботи з Android Studio та інтеграції різноманітних бібліотек і фреймворків.
* аналіз та оптимізація: уміння аналізувати процеси тестування та оптимізувати їх для кращої взаємодії з користувачами.

Ці досягнення та знання сприяють не лише розробці функціональних додатків, але й професійному розвитку, що відкриває нові можливості для подальшого вдосконалення і кар’єрного зростання.

Тестування додатку відбувалося в Android Studio на кожному етапі розробки і всі помилки виправлялися оперативно.

У підсумку, додаток було протестовано на моєму особистому пристрої Android з версією операційної системи 13. Додаток працює стабільно і жодних додаткових помилок більше не виявлено.

Вихідний код та файли для встановлення додатку були завантажені до репозиторію на сервісі Github за посиланням <https://github.com/woldesj/TEST_NMT1>.

Тепер кожен бажаючий може завантажити, встановити та протестувати додаток на своєму пристрої.

## 3.3 Висновок до розділу 3

В цьому розділі було детально розглянуто процес розробки, реалізації та тестування мобільного додатку «Turbo НМТ».

Детально розглянуто архітектуру створеного додатку, який складається з декількох ключових компонентів: активностей, фрагментів та допоміжних класів. Кожен компонент виконує специфічні функції, забезпечуючи загальну працездатність та зручність використання додатку. Структура додатку була спроєктована з дотриманням принципів модульності та розширюваності, що дозволяє легко додавати нові функції у майбутньому.

Завдяки ретельному проєктуванню, правильному вибору технологій та грамотній реалізації вдалося створити повноцінний мобільний додаток «Turbo НМТ», який відповідає поставленим вимогам та задовольняє потреби користувачів у зручній підготовці до Національного мультипредметного тесту. Додаток має інтуїтивно зрозумілий інтерфейс, забезпечує зберігання результатів тестувань та надає зворотній зв’язок щодо рівня засвоєння матеріалу.

# ВИСНОВКИ

Результатом даної кваліфікаційної роботи стало створення мобільного додатку «Turbo НМТ», призначеного для перевірки та оцінювання знань користувачів. У процесі роботи були ретельно проаналізовані потреби цільової аудиторії, існуючі аналоги та сучасні підходи до розробки мобільних додатків, що дозволило визначити оптимальні технології та інструменти для реалізації проєкту.

Завдяки продуманому проєктуванню, правильному вибору технологій та грамотній реалізації, вдалося створити повноцінний мобільний додаток, який відповідає всім поставленим вимогам та задовольняє потреби користувачів у зручній та ефективній підготовці до Національного мультипредметного тесту. Застосунок «Turbo НМТ» має інтуїтивно зрозумілий інтерфейс та надає зворотний зв’язок щодо рівня знань.

Додаток працює стабільно, пройшов тестування на особистому пристрої з операційною системою Android 13 і всі виявлені помилки були своєчасно виправлені. Вихідний код та файли для встановлення додатку завантажені до репозиторію на сервісі GitHub, що дозволяє кожному бажаючому завантажити, встановити та протестувати додаток на своєму пристрої.

Додаток «Turbo НМТ» має значний потенціал для подальшого розвитку та може бути легко розширений завдяки модульній структурі. Це відкриває нові можливості для вдосконалення функціоналу та адаптації під інші освітні потреби. Дана робота не лише демонструє знання теоретичних аспектів розробки мобільних додатків, але й показує вміння їх успішно реалізувати на практиці, що сприяє підвищенню професійного рівня та відкриває нові перспективи для подальшого розвитку та кар’єрного зростання в сфері мобільних додатків.

Отримання практичного досвіду та нових знань у процесі проєктування та розробки Android-додатку «Turbo НМТ» стало важливим етапом мого професійного становлення. Цей проєкт дозволив мені оволодіти основними принципами розробки мобільних додатків для платформи Android та успішно втілити їх на практиці.

Під час роботи над додатком, я здобув безцінний досвід у використанні інтегрованого середовища розробки Android Studio, вивчив принципи мови програмування Java та мови макетів XML. Ці знання дозволили створити функціональний додаток, який відповідає вимогам та впроваджується в контекст навчання з Національного мультипредметного тесту.

Отриманий досвід у розробці додатку «Turbo НМТ» став важливим етапом у моєму професійному зростанні, дозволяючи відкрити нові горизонти для подальшого вдосконалення навичок та реалізації інноваційних ідей у сфері мобільної розробки. Цей проєкт зміцнив мою впевненість у здатності досягати професійних цілей та успішно розвиватися у сфері інформаційних технологій.

Під час розробки додатку, я зміг глибоко оцінити важливість усіх етапів навчання у ВСП «Економіко-правничий фаховий коледж ЗНУ» за спеціальністю 121 Інженерія програмного забезпечення. Це навчання сприяє підготовці висококваліфікованих фахівців, які можуть ефективно використовувати сучасні технології та методи розробки інформаційних систем. Весь процес навчання відкрив широкі перспективи для моєї майбутньої кар’єри в сфері програмування.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мак-Лахлін Бретт. Java та XML.: 2-е видання : Старого Лева, 2017. 547 с.

2. Рей Ерік. Вивчаємо XML. Символ-Плюс, 2017. 402 с.

3. Хантер Девід, Фаусетт Джо, Рафтер Джефф. XML: Базовий курс: Діалектика, 2009. 1344 с.

4. Офіційна документація Android, посiбник до Android Studio. URL : https://developer.android.com/studio/intro (дата звернення : 01.03.2024).

5. Стандарт дизайну URL : https://m3.material.io/ (дата звернення : 20.02.2024).

6. Bruce Eckel. Thinking in Java. Prentice Hall, 2006. 1150 с.

7. Cay Horstmann, Gary Cornell: Core Java Fundamentals - Volume I. 12-th edition: Prentice-Hall, 2016. 1008 с.

8. Велика спільнота розробників URL : https://stackoverflow.com/ (дата звернення : 12.04.2024).

9. Cay Horstmann, Gary Cornell: Core Java Advanced Topics - Volume II.: Prentice-Hall, 2016. 1088 с.

**Декларація**

**академічної доброчесності**

**здобувача освіти ВСП «Економіко-правничого фахового коледжу ЗНУ»**

Я, Коротенко Володимир Володимирович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, здобувач освіти 4 курсу, спеціальності/освітньо-професійної програми 121 Інженерія програмного забезпечення, групи К121-20, адреса електронної пошти frisb23j@gmail.com

- підтверджую, що написана мною кваліфікаційна робота на тему «Розробка та реалізація мобільного додатку для перевірки і оцінювання знань» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згоден на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата \_\_\_\_\_\_\_\_\_ Підпис \_\_\_\_\_\_\_\_\_ Коротенко В.В.

Дата \_\_\_\_\_\_\_\_\_ Підпис \_\_\_\_\_\_\_\_\_ Ходаковська А.В.