

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота
за освітнім ступенем «бакалавр»
на тему:

**МОБІЛЬНИЙ ЗАСТОСУНОК ДЛЯ ТРЕНУВАННЯ КОГНІТИВНИХ
НАВИЧОК**

Виконав:

здобувач IV курсу
групи ПЗ-41
спеціальності 121 «Інженерія
програмного забезпечення»
Луцик Віталій Миколайович

Науковий керівник:

к.п.н., доцент Петренко С. В.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ ДЛЯ ТРЕНУВАННЯ КОГНІТИВНИХ НАВИЧОК	6
1.1. Когнітивні навички як об’єкт цифрового тренування	6
1.2. Аналіз мобільних застосунків для розвитку пам’яті, уваги та мислення	11
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ	17
2.1. Постановка задачі та сценарії використання	17
2.2. Функціональні й нефункціональні вимоги до мобільного застосунку	20
2.3. Архітектура, компоненти та логіка взаємодії модулів застосунку	24
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ	32
3.1. Обґрунтування вибору технологій та організація Android-проєкту	32
3.2. Реалізація функціональних модулів та користувацького інтерфейсу застосунку	36
3.3. Тестування, оцінювання результатів роботи та напрями вдосконалення застосунку	41
ВИСНОВКИ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47

ВСТУП

Актуальність дослідження. У сучасних умовах швидкого зростання інформаційних потоків, постійного використання цифрових сервісів та підвищення вимог до швидкості прийняття рішень особливого значення набуває розвиток когнітивних навичок людини. Увага, пам'ять, логічне мислення, швидкість реакції та здатність до аналізу інформації впливають на ефективність навчання, професійної діяльності й повсякденного життя. Саме тому цифрові інструменти, спрямовані на тренування когнітивних процесів, стають актуальним напрямом розробки сучасного програмного забезпечення. Мобільні застосунки є зручним засобом організації коротких регулярних тренувань, оскільки вони доступні користувачеві в будь-який час, мають інтерактивний формат і можуть поєднувати навчальні, тренувальні та ігрові елементи. На ринку вже існує низка рішень для розвитку пам'яті, уваги та мислення, зокрема Lumosity, Elevate, Peak, CogniFit, NeuroNation. Водночас значна частина таких застосунків має обмежений безкоштовний функціонал, перевантажений інтерфейс або надмірно складну структуру взаємодії з користувачем. Це створює потребу в розробці простого, доступного та зрозумілого мобільного застосунку, який дає змогу виконувати базові когнітивні вправи у зручному форматі.

Актуальність теми також зумовлена необхідністю практичного застосування знань з інженерії програмного забезпечення: аналізу предметної області, формування вимог, проектування архітектури, розробки інтерфейсу, реалізації програмних модулів і тестування готового продукту. У межах кваліфікаційної роботи розробляється Android-застосунок для тренування когнітивних навичок, що поєднує вправи на пам'ять, увагу та логічне мислення, має зрозумілу навігацію, систему рівнів складності й механізм відображення результатів користувача.

Мета дослідження полягає у проектуванні та реалізації мобільного застосунку для тренування когнітивних навичок користувача на платформі

Android із використанням сучасних засобів розробки, зручного інтерфейсу та базових механізм оцінювання результатів виконання вправ.

Для досягнення поставленої мети необхідно виконати такі **завдання дослідження**:

1. Проаналізувати предметну область розвитку когнітивних навичок та існуючі мобільні застосунки для тренування пам'яті, уваги й логічного мислення.
2. Сформулювати функціональні та нефункціональні вимоги до мобільного застосунку з урахуванням зручності використання, простоти навігації та можливості подальшого розширення.
3. Спроекувати структуру застосунку, основні сценарії взаємодії користувача із системою та логіку роботи тренувальних модулів.
4. Реалізувати мобільний застосунок на платформі Android із використанням Kotlin, Android Studio та XML-розмітки інтерфейсу.
5. Провести тестування роботи застосунку, перевірити коректність навігації, виконання вправ, підрахунку результатів і стабільність функціонування основних екранів.

Об'єктом дослідження є процес цифрового тренування когнітивних навичок користувача засобами мобільного програмного забезпечення.

Предметом дослідження є методи, моделі та програмні засоби проєктування і реалізації Android-застосунку для тренування пам'яті, уваги та логічного мислення.

Методи дослідження. У роботі використано методи аналізу та узагальнення для вивчення предметної області й наявних програмних рішень; методи порівняння для визначення переваг і недоліків існуючих мобільних застосунків; методи моделювання для опису структури системи, сценаріїв використання та логіки роботи модулів; методи проєктування програмного забезпечення для побудови архітектури застосунку; методи програмної реалізації для створення Android-застосунку засобами Kotlin та Android Studio; методи тестування для перевірки коректності роботи основних функцій.

Практичне значення одержаних результатів полягає у створенні мобільного застосунку, який може використовуватися як інструмент для самостійного тренування базових когнітивних навичок. Розроблений застосунок має навчальну та прикладну цінність, оскільки демонструє повний цикл створення програмного продукту: від аналізу предметної області й формування вимог до реалізації інтерфейсу, програмної логіки та тестування. Отримані результати можуть бути використані як основа для подальшого розширення функціоналу, додавання нових вправ, системи статистики, персоналізації тренувань і збереження прогресу користувача.

Апробація результатів дослідження. Основні положення та результати кваліфікаційної роботи були представлені під час звітної студентської науково-практичної конференції. Подано тези доповіді на XIX Всеукраїнська науково-практичну конференцію здобувачів вищої освіти і молодих учених «Наука, освіта, суспільство очима молодих» (14 травня 2026 р.).

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі розглянуто теоретичні основи розвитку когнітивних навичок, проведено аналіз існуючих мобільних застосунків і визначено основні проблеми таких рішень. У другому розділі сформульовано вимоги до застосунку, описано його структуру, основні сценарії використання та логіку взаємодії користувача із системою. У третьому розділі подано обґрунтування вибору технологій, описано реалізацію основних екранів і функціональних модулів, а також наведено результати тестування розробленого мобільного застосунку.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ ДЛЯ ТРЕНУВАННЯ КОГНІТИВНИХ НАВИЧОК

1.1. Когнітивні навички як об'єкт цифрового тренування

Когнітивні навички є сукупністю психічних процесів, які забезпечують сприймання, обробку, збереження, відтворення та практичне використання інформації людиною. До них належать увага, пам'ять, мислення, мовлення, уява, здатність до розв'язання проблем, прийняття рішень, орієнтація в нових ситуаціях та швидкість обробки інформації [22]. У повсякденному житті ці навички визначають здатність людини навчатися, виконувати професійні завдання, швидко реагувати на зміни, аналізувати інформацію та ефективно взаємодіяти з навколишнім середовищем. Саме тому розвиток когнітивних навичок є актуальним не лише для дітей і здобувачів освіти, а й для дорослих користувачів, які прагнуть підтримувати концентрацію, пам'ять, логічне мислення та загальну розумову активність.

У контексті інженерії програмного забезпечення когнітивні навички можуть розглядатися як об'єкт цифрового тренування, тобто як сукупність характеристик, які можна підтримувати або розвивати за допомогою спеціально спроектованих інтерактивних вправ. Цифрове тренування не замінює повноцінного психологічного чи медичного супроводу, однак може бути корисним інструментом для регулярної самостійної практики. Мобільний формат є особливо зручним, оскільки користувач може виконувати короткі вправи у зручний час, отримувати миттєвий результат і поступово відстежувати власний прогрес.

Однією з базових когнітивних функцій є увага. Вона забезпечує здатність людини зосереджуватися на певному об'єкті або завданні, відокремлювати важливу інформацію від другорядної та підтримувати концентрацію протягом визначеного часу [24]. У цифровому середовищі увага

може тренуватися через вправи на пошук об'єктів, визначення відмінностей, вибір правильного елемента серед кількох варіантів, реагування на зміну кольору, форми або положення об'єктів. Такі завдання мають бути короткими, зрозумілими та візуально не перевантаженими, оскільки надмірна складність інтерфейсу може не розвивати увагу, а навпаки створювати зайве когнітивне навантаження.

Пам'ять є не менш важливою складовою когнітивної діяльності. Вона пов'язана зі здатністю зберігати, відтворювати та використовувати інформацію. У межах когнітивного тренування особливе значення мають короткочасна, робоча та довготривала пам'ять. Короткочасна пам'ять дає змогу утримувати невеликий обсяг інформації протягом короткого часу, робоча пам'ять забезпечує активне опрацювання даних під час виконання завдання, а довготривала пам'ять пов'язана з накопиченням досвіду та знань [22]. У мобільному застосунку ці процеси можуть бути реалізовані через вправи на запам'ятовування послідовностей, розташування елементів, повторення комбінацій, відтворення раніше показаних об'єктів або вибір правильного варіанта після короткої паузи.

Мислення є складною когнітивною функцією, що охоплює аналіз, порівняння, узагальнення, встановлення закономірностей, причинно-наслідкових зв'язків та пошук рішень. Логічне мислення особливо важливе для навчання, професійної діяльності, програмування, роботи з інформацією та прийняття рішень [1]. Для мобільного застосунку це створює підстави для включення вправ у формі лабіринтів, логічних задач, пошуку правильного шляху, вибору закономірностей, сортування елементів або побудови послідовностей. Такі вправи повинні мати чіткі правила, зрозумілий результат і можливість поступового ускладнення.

Швидкість реакції також може розглядатися як складова когнітивного тренування, оскільки вона пов'язана з оперативним сприйняттям інформації та прийняттям рішень в умовах обмеженого часу. У цифрових вправах ця навичка може тренуватися через завдання, де користувач має швидко

натиснути на правильний об'єкт, уникнути помилки, відреагувати на появу стимулу або виконати дію до завершення таймера. Водночас використання обмеження часу має бути помірним, щоб не створювати надмірного стресу для користувача. Для навчального або тренувального застосунку доцільно передбачати кілька рівнів складності, щоб користувач міг поступово адаптуватися до швидшого темпу виконання вправ.

Розвиток когнітивних навичок може розглядатися з позицій різних психолого-педагогічних підходів. Когнітивний підхід акцентує увагу на способах обробки інформації, розвитку стратегій запам'ятовування, самоконтролю, концентрації та поступового ускладнення завдань [24]. Для програмного продукту це означає необхідність передбачити логічно структуровані вправи, систему рівнів, повторення завдань та зрозумілий зворотний зв'язок після виконання. Користувач має не лише пройти вправу, а й побачити результат, зрозуміти власну помилку та мати можливість повторити завдання [15].

Конструктивістський підхід розглядає навчання як активний процес, у якому користувач самостійно формує знання через власний досвід. У цифровому середовищі це може реалізовуватися через інтерактивні вправи, у яких користувач не пасивно сприймає інформацію, а виконує дії, перевіряє припущення, отримує результат і коригує свою стратегію. Наприклад, у вправі на проходження лабіринту користувач має самостійно обирати напрям руху, аналізувати помилки та поступово знаходити правильний шлях. Такий формат є корисним для розвитку логічного мислення, просторової орієнтації та планування дій [20].

Нейропсихологічний підхід підкреслює значення вправ, спрямованих на активізацію окремих психічних функцій: пам'яті, уваги, просторового сприйняття, моторної координації та швидкості реакції [1]. Для мобільного застосунку це означає доцільність використання коротких вправ з чіткими правилами, простими візуальними стимулами, поступовим ускладненням та можливістю багаторазового повторення. Наприклад, вправи на

запам'ятовування розташування елементів можуть підтримувати робочу пам'ять, а завдання на швидкий вибір правильного об'єкта можуть тренувати увагу й реакцію.

Інтерактивний підхід передбачає активну участь користувача, швидку реакцію системи на його дії та підтримання мотивації через зворотний зв'язок. У мобільному застосунку це реалізується через кнопки вибору, анімації, повідомлення про успішне виконання вправи, систему балів, рівнів або результатів. Важливо, щоб зворотний зв'язок був зрозумілим і не перевантажував користувача. Наприклад, після проходження вправи доцільно показувати кількість набраних балів, правильні й неправильні відповіді, а також коротку рекомендацію щодо повторення або переходу до складнішого рівня.

Гуманістичний підхід у контексті цифрового тренування пов'язаний із врахуванням індивідуальних можливостей користувача, підтримкою позитивного досвіду та уникненням ситуацій, які можуть знижувати мотивацію. Застосунок не повинен створювати відчуття невдачі через складні або незрозумілі завдання. Навпаки, важливо забезпечити поступовий перехід від простих вправ до складніших, використовувати доброзичливі повідомлення та надавати користувачеві можливість повторити завдання. Це особливо важливо, якщо потенційною аудиторією застосунку можуть бути діти, люди похилого віку або користувачі з різним рівнем цифрової грамотності.

Компетентнісний підхід дає змогу розглядати когнітивне тренування не як ізольоване виконання окремих вправ, а як формування практичної здатності використовувати увагу, пам'ять і мислення у реальних ситуаціях. Наприклад, вправа на пам'ять може бути пов'язана не лише з механічним запам'ятовуванням елементів, а й із розвитком уміння помічати закономірності. Вправа на логіку може не лише перевіряти правильність відповіді, а й формувати здатність планувати послідовність дій. Саме тому під

час розробки мобільного застосунку важливо враховувати не тільки окремі когнітивні функції, а й взаємозв'язок між ними.

Окрему увагу потрібно приділити принципу регулярності. Когнітивні тренування є ефективнішими тоді, коли виконуються не одноразово, а систематично. Для мобільного застосунку це означає доцільність реалізації коротких тренувальних сесій, які користувач може проходити щодня або кілька разів на тиждень. Такий формат краще відповідає реальним сценаріям використання мобільних пристроїв, оскільки користувач не завжди готовий витратити багато часу на одну вправу. Натомість короткі завдання тривалістю кілька хвилин можуть бути зручними, доступними та менш втомлюваними.

Важливою характеристикою цифрового когнітивного тренажера є адаптивність. Користувачі можуть мати різний рівень підготовки, різний вік, різну швидкість реакції та різний досвід використання мобільних застосунків. Тому система має передбачати рівні складності або поступове ускладнення вправ залежно від результатів. На початковому рівні завдання повинні бути простими й пояснювати базову механіку. На середньому рівні можна збільшувати кількість елементів, часове обмеження або складність логічних зв'язків. На складному рівні доцільно поєднувати кілька когнітивних функцій, наприклад пам'ять і швидкість реакції або увагу й логічне мислення.

Ще одним важливим аспектом є доступність інтерфейсу. Оскільки когнітивне тренування саме по собі створює певне навантаження на користувача, інтерфейс застосунку має бути простим, передбачуваним і візуально зрозумілим. Кнопки повинні мати достатній розмір, тексти інструкцій мають бути короткими, а кольори та графічні елементи не повинні заважати виконанню завдання. Надмірна кількість елементів, складна навігація або нечіткі правила можуть негативно впливати на якість тренування, оскільки користувач витратиме зусилля не на виконання вправи, а на розуміння інтерфейсу [23].

З погляду розробки програмного забезпечення когнітивні навички можна перетворити на набір функціональних модулів майбутнього

застосунку. Наприклад, модуль пам'яті може містити вправи на запам'ятовування послідовностей, модуль уваги – вправи на пошук або швидке реагування, модуль логіки – лабіринти, задачі на закономірності або вибір правильного рішення. Такий поділ дає змогу побудувати зрозумілу структуру застосунку, спростити навігацію та забезпечити можливість подальшого розширення функціоналу.

Отже, когнітивні навички є доцільним об'єктом цифрового тренування, оскільки значна частина таких навичок може підтримуватися через повторювані, короткі та інтерактивні вправи. Аналіз теоретичних підходів показує, що мобільний застосунок для розвитку когнітивних навичок повинен мати просту структуру, зрозумілі правила, різні рівні складності, візуальний зворотний зв'язок, можливість повторення вправ і поступове ускладнення завдань. Саме ці положення формують теоретичне підґрунтя для подальшого аналізу існуючих мобільних рішень і проєктування власного застосунку.

1.2. Аналіз мобільних застосунків для розвитку пам'яті, уваги та мислення

Ринок мобільних застосунків для розвитку когнітивних навичок активно розвивається завдяки поширенню смартфонів, зростанню інтересу користувачів до саморозвитку та популярності коротких інтерактивних форматів навчання. Такі застосунки зазвичай пропонують набір вправ, спрямованих на тренування пам'яті, уваги, логічного мислення, швидкості реакції, мовних навичок або математичних операцій. Їхньою перевагою є доступність: користувач може виконувати тренування у зручний час, отримувати миттєвий результат і поступово відстежувати динаміку власного прогресу.

Водночас аналіз таких рішень потребує обережності, оскільки не всі заяви розробників щодо впливу застосунків на когнітивний розвиток мають достатнє наукове підтвердження. У науковій літературі підкреслюється, що

когнітивні тренування можуть покращувати виконання конкретних завдань, однак питання перенесення цих результатів на ширші життєві або навчальні ситуації залишається дискусійним [7, 21]. Тому в межах цієї роботи мобільні застосунки доцільно розглядати не як медичні або терапевтичні засоби, а як інструменти для організації регулярних вправ, підтримання мотивації користувача та розвитку окремих когнітивних дій у цифровому середовищі.

Для аналізу було обрано кілька популярних мобільних рішень, які представлені на ринку когнітивних тренажерів: Lumosity, Elevate, Peak, CogniFit та NeuroNation. Вони мають схожу загальну мету, але відрізняються за структурою вправ, рівнем персоналізації, складністю інтерфейсу, цільовою аудиторією та способом подання результатів. Огляд цих застосунків дає змогу виокремити корисні практики для подальшого проєктування власного мобільного застосунку.

Lumosity є одним із найвідоміших застосунків у сфері цифрових когнітивних тренувань. Його концепція базується на коротких ігрових вправах, що охоплюють пам'ять, увагу, швидкість мислення, гнучкість і розв'язання проблем [11]. Перевагою Lumosity є велика кількість мініігор, зрозумілий формат щоденних тренувань, система прогресу та можливість швидкого залучення користувача до виконання завдань. Застосунок добре демонструє, що когнітивне тренування в мобільному форматі має бути коротким, візуально привабливим і зрозумілим без довгого навчання.

Разом із тим Lumosity має і певні обмеження. Частина функцій доступна лише за умови підписки, що може зменшувати доступність для широкої аудиторії. Крім того, історія цього сервісу демонструє важливість коректного формулювання очікувань користувача: розробники не повинні подавати когнітивні ігри як засіб лікування або гарантованого покращення інтелектуальних здібностей без належних доказів [6]. Для власного застосунку це означає необхідність уникати перебільшених тверджень і позиціонувати продукт як тренувальний та освітній інструмент, а не як медичний засіб.

Elevate орієнтований не лише на класичне тренування пам'яті та уваги, а й на практичні навички: читання, письмо, мовлення, математичні операції та швидкість обробки інформації [5]. Особливістю застосунку є персоналізована програма тренувань, яка адаптується до результатів користувача. Такий підхід є важливим з погляду користувацького досвіду, оскільки різні користувачі мають різний рівень підготовки, різні цілі та різний темп виконання завдань.

Сильною стороною Elevate є поєднання когнітивних вправ із практично корисними навичками. Застосунок не обмежується абстрактними іграми, а намагається пов'язати тренування з реальними діями, наприклад роботою з текстом або числовою інформацією. Водночас для теми цієї кваліфікаційної роботи Elevate є частково ширшим за потрібний напрям, оскільки він більше фокусується на продуктивності, мовних і математичних уміннях, ніж на окремих вправах для пам'яті, уваги та логічного мислення. Проте з нього доцільно запозичити ідею персоналізації, коротких сесій і поступового ускладнення завдань.

Peak також належить до популярних застосунків для тренування когнітивних навичок. Він пропонує набір ігор, спрямованих на пам'ять, увагу, розв'язання проблем, мовні навички та інші когнітивні процеси [17]. Важливою особливістю Peak є виражена гейміфікація: користувач отримує вправи у форматі коротких ігор, може відстежувати прогрес, порівнювати результати й проходити щоденні тренування. Такий підхід підвищує мотивацію, оскільки користувач сприймає тренування не як складне навчальне завдання, а як коротку ігрову активність.

Недоліком Peak може бути надмірна ігрова насиченість для користувачів, які очікують максимально простого інструменту без зайвих елементів. Велика кількість режимів, показників і візуальних компонентів може створювати додаткове навантаження на користувача. Для власного застосунку важливо врахувати цю особливість і зберегти баланс між ігровою привабливістю та простотою інтерфейсу. Гейміфікація має підтримувати тренування, але не повинна ускладнювати основний сценарій взаємодії.

CogniFit відрізняється від попередніх застосунків сильнішим акцентом на когнітивне оцінювання, індивідуальні показники та тренування різних когнітивних доменів. У матеріалах сервісу зазначено, що застосунок пропонує вправи для пам'яті, уваги, концентрації, швидкості обробки інформації, реакції та інших когнітивних здібностей [11]. Також CogniFit використовує підхід, за якого користувач може отримувати показники за різними когнітивними напрямками та проходити персоналізовані тренування.

Перевагою CogniFit є структурований підхід до когнітивних функцій і спроба поєднати тренування з оцінюванням. Це може бути корисним для користувачів, які хочуть не лише виконувати вправи, а й бачити більш детальну картину власних результатів. Водночас така модель може бути складнішою для пересічного користувача, особливо якщо інтерфейс містить багато показників або спеціалізованих термінів. Для власного застосунку варто використати ідею відображення результатів, але реалізувати її у спрощеній формі: наприклад, показувати кількість балів, успішність проходження рівня, коротку статистику або рекомендацію щодо повторення.

NeuroNation позиціонується як застосунок для персоналізованого тренування мозку, що містить вправи для пам'яті, концентрації та мислення [17]. Серед його сильних сторін можна виокремити персоналізовані плани, регулярність тренувань, візуальне подання прогресу та орієнтацію на короткі заняття. Для мобільного застосунку такий підхід є практично доцільним, оскільки користувачі частіше повертаються до сервісу, якщо бачать зрозумілу мету, простий сценарій і поступовий прогрес.

Водночас NeuroNation, як і інші подібні рішення, значною мірою залежить від моделі підписки та повного набору функцій, доступного не всім користувачам. Крім того, складні системи персоналізації потребують більшої кількості даних про користувача, що актуалізує питання приватності та прозорості обробки інформації.

Порівняння розглянутих застосунків показує, що більшість сучасних рішень мають спільні функціональні ознаки: поділ вправ за когнітивними

напрямами, короткі тренувальні сесії, використання гейміфікації, підрахунок балів, візуальне відображення прогресу та певний рівень персоналізації. Ці елементи є важливими для підтримання інтересу користувача та формування звички до регулярного тренування. Водночас аналіз виявляє і типові недоліки: обмеження безкоштовної версії, складні або перевантажені інтерфейси, надмірні маркетингові твердження, недостатню прозорість щодо використання даних і не завжди очевидний зв'язок між вправами та заявленими когнітивними ефектами. Для зручності результати аналізу узагальнено у таблиці 1.1.

Таблиця 1.1

Порівняльна характеристика мобільних застосунків для розвитку
когнітивних навичок

Застосунок	Основна спрямованість	Сильні сторони	Обмеження	Ідеї для застосунку
Lumosity	Пам'ять, увага, швидкість мислення, розв'язання проблем	Велика кількість мініігор, короткі тренування, зрозуміла система прогресу	Частина функцій платна, ризик завищених очікувань користувача	Короткі вправи, щоденні тренування, проста подача результатів
Elevate	Пам'ять, фокус, мовні навички, математика, продуктивність	Персоналізація, адаптація програми, практична спрямованість	Не всі вправи прямо пов'язані з пам'яттю, увагою та логікою	Адаптивність, поступове ускладнення, короткі сесії
Peak	Пам'ять, увага, логіка, мова, розв'язання проблем	Гейміфікація, сучасний дизайн, відстеження прогресу	Можлива перевантаженість режимами та показниками	Ігрова мотивація, рівні, візуальний прогрес
CogniFit	Когнітивне оцінювання і тренування різних доменів	Структурований підхід, персоналізовані вправи, показники результатів	Складніший інтерфейс, спеціалізована термінологія	Спрощена статистика, оцінювання результатів, когнітивні категорії
NeuroNation	Персоналізоване тренування пам'яті, концентрації та мислення	Короткі регулярні заняття, персональні плани, прогрес	Частина можливостей залежить від підписки, потреба в даних користувача	Регулярність, простий план тренувань, мінімальна персоналізація

Аналіз таблиці показує, що найбільш придатними для використання у власному проєкті є не окремі функції конкретного застосунку, а загальні

принципи побудови когнітивного тренажера. Насамперед це поділ вправ за типом когнітивної навички, прості правила, короткий час проходження, система балів, рівні складності та зрозумілий зворотний зв'язок. Саме ці елементи можна реалізувати в Android-застосунку без надмірного ускладнення архітектури.

Окремої уваги потребує питання доступності. Більшість популярних застосунків мають професійний дизайн, але не завжди є однаково зручними для користувачів із різним рівнем цифрової грамотності. Для застосунку доцільно зробити інтерфейс максимально простим: головне меню, вибір типу вправи, вибір рівня складності, проходження завдання та екран результату. Така структура зменшує кількість зайвих переходів і дає користувачеві змогу швидко зрозуміти логіку роботи програми.

Ще одним важливим висновком є необхідність обмеженого та прозорого використання даних. Оскільки застосунки для когнітивного тренування можуть зберігати результати виконання вправ, статистику, рівень складності або інші показники користувача, варто дотримуватися принципу мінімізації даних. Результати доцільно зберігати локально без передавання персональної інформації на зовнішні сервери. Це спрощує реалізацію, зменшує ризики приватності та відповідає навчальному характеру розробки.

Таким чином, аналіз мобільних застосунків для розвитку пам'яті, уваги та мислення показав, що сучасні рішення поєднують когнітивні вправи, ігрову мотивацію, персоналізацію та відстеження прогресу. Водночас їхні недоліки пов'язані з платними обмеженнями, складністю інтерфейсу, недостатньою прозорістю наукових тверджень і потенційними питаннями приватності. Для розроблюваного мобільного застосунку доцільно зосередитися на простій структурі, базових когнітивних модулях, коротких вправах, рівнях складності, локальному збереженні результатів і зрозумілому інтерфейсі. Це дасть змогу створити доступний навчальний продукт, який враховує сильні сторони існуючих рішень, але не повторює їхніх основних недоліків.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

2.1. Постановка задачі та сценарії використання

Розвиток когнітивних навичок за допомогою цифрових інструментів потребує створення простих, доступних та ефективних рішень, які поєднують науковий підхід із сучасними технологіями. У межах цієї роботи ставиться задача розробити прототип мобільного застосунку, що дозволить користувачам тренувати базові когнітивні процеси – увагу, пам'ять, логічне мислення та швидкість реакції – у зручному інтерактивному форматі

Актуальність розробки такого застосунку пов'язана з тим, що сучасний користувач дедалі частіше взаємодіє з навчальними, розважальними та тренувальними сервісами саме через мобільні пристрої. Мобільний формат є зручним для коротких регулярних занять, не потребує спеціального обладнання та дає змогу виконувати вправи у будь-який зручний час [23]. Тому під час проєктування застосунку важливо врахувати не лише наявність окремих тренувальних завдань, а й простоту навігації, зрозумілість правил, швидкий запуск вправ і можливість отримання результату після завершення тренування.

Цільовим користувачем застосунку є особа, яка прагне підтримувати або розвивати базові когнітивні навички за допомогою коротких інтерактивних вправ. Передбачається, що користувач не повинен мати спеціальних знань для роботи із застосунком. Усі дії мають бути інтуїтивно зрозумілими: запуск програми, вибір напрямку тренування, вибір рівня складності, виконання вправи та перегляд результату. Такий підхід відповідає загальній ідеї когнітивних тренувань, у яких важливе значення мають регулярність, поступове ускладнення завдань і збереження мотивації користувача.

Основна мета розробки полягає у створенні мобільного застосунку, який демонструє ключові функції тренування когнітивних навичок і забезпечує взаємодію користувача з різними типами вправ. Застосунок має відображати структуру інтерфейсу, логіку переходів між екранами, базові механіки тренувань, вибір рівня складності та систему підрахунку результатів.

З погляду інженерії програмного забезпечення задача розробки полягає не лише у створенні окремих екранів, а й у формуванні узгодженої структури програмного продукту. Застосунок повинен мати логічно пов'язані компоненти: головне меню, модулі тренувань, екран вибору складності, екран виконання вправи, екран результатів і допоміжні налаштування. Такий поділ спрощує реалізацію, тестування та подальше розширення функціональності [9].

Типовий сценарій використання застосунку передбачає таку послідовність дій: користувач запускає програму, потрапляє на головний екран, обирає категорію вправи, визначає рівень складності, виконує запропоноване завдання, після чого отримує результат у вигляді балів або повідомлення про завершення рівня. За потреби користувач може повернутися до головного меню, повторити вправу або перейти до іншого тренувального модуля. Такий сценарій є базовим і визначає основну логіку взаємодії користувача із застосунком.

У межах роботи передбачається реалізація кількох груп вправ, орієнтованих на різні когнітивні процеси. Логічні вправи спрямовані на розвиток мислення, пошуку рішень і послідовного аналізу дій. Вправи на увагу мають тренувати концентрацію, швидкість сприйняття та здатність помічати потрібні об'єкти серед інших елементів. Вправи на пам'ять орієнтовані на запам'ятовування послідовностей, образів або розташування елементів. Додаткові візуальні завдання можуть використовувати роботу з кольорами, формами та просторовим сприйняттям.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- визначити функціональні та нефункціональні вимоги до мобільного застосунку;
- описати основні сценарії взаємодії користувача із системою;
- сформувати архітектурну модель, що описує основні компоненти застосунку та їх взаємодію;
- розробити діаграми прецедентів, послідовності та структурні схеми;
- описати логіку роботи окремих тренувальних модулів;
- визначити підхід до збереження результатів, налаштувань і вибраного рівня складності;
- підготувати основу для подальшої технічної реалізації та тестування застосунку.

Межі розроблюваної системи визначаються тим, що застосунок орієнтований на локальну роботу на мобільному пристрої без обов'язкового підключення до віддаленого сервера. Це дає змогу зменшити складність реалізації, підвищити швидкодію та уникнути зайвого збирання персональних даних користувача [14]. У перспективі архітектура застосунку може бути розширена за рахунок додавання повнішої статистики, персоналізованих рекомендацій, адаптивного добору складності та синхронізації результатів між пристроями.

Очікуваним результатом виконання поставленої задачі є мобільний застосунок, який забезпечує базовий цикл когнітивного тренування: вибір вправи, виконання завдання, оцінювання результату та повернення до наступної дії користувача. Розроблена структура має бути достатньо простою для реалізації в межах бакалаврської кваліфікаційної роботи, але водночас відкритою для подальшого розвитку та додавання нових тренувальних модулів.

Таким чином, постановка задачі визначає основні напрями подальшого проектування: формування вимог до застосунку, опис сценаріїв використання, побудову архітектури, моделювання компонентів і визначення логіки роботи

тренувальних вправ. Саме ці аспекти є основою для подальшого опису функціональних вимог, структури застосунку та його практичної реалізації.

2.2. Функціональні й нефункціональні вимоги до мобільного застосунку

Визначення вимог є одним із ключових етапів проектування програмного забезпечення, оскільки саме на цьому етапі встановлюється, які функції має виконувати система, які обмеження повинні враховуватися під час її реалізації та якими критеріями оцінюватиметься якість готового продукту. У сучасній інженерії програмного забезпечення вимоги розглядаються як формалізований опис потреб користувачів, умов експлуатації та очікуваної поведінки системи. Згідно з підходами ISO/IEC/IEEE 29148, інженерія вимог охоплює процеси виявлення, аналізу, специфікації та перевірки вимог упродовж життєвого циклу програмної системи [9].

Для мобільного застосунку, призначеного для тренування когнітивних навичок, вимоги доцільно поділити на дві основні групи: функціональні та нефункціональні. Функціональні вимоги описують конкретні дії, які система має виконувати у відповідь на дії користувача. Нефункціональні вимоги визначають якісні характеристики застосунку: зручність використання, швидкодію, надійність, розширюваність, безпеку та доступність. Такий поділ відповідає загальноприйнятій практиці специфікації вимог до програмних систем, де функціональні можливості відображають поведінку продукту, а нефункціональні характеристики задають умови його якісної роботи [18].

Під час формування вимог до розроблюваного застосунку враховано специфіку мобільного середовища. Користувач взаємодіє із системою через сенсорний екран, тому інтерфейс має бути простим, зрозумілим і не перевантаженим зайвими елементами. Також важливо забезпечити швидкий запуск вправ, мінімальну кількість проміжних дій і зрозумілий результат після завершення тренування. Рекомендації Android Developers щодо якості

мобільних застосунків наголошують на важливості інтуїтивного користувацького досвіду, стабільної роботи, адаптації до різних пристроїв і коректної поведінки застосунку в типових сценаріях використання [2-4].

Функціональні вимоги до застосунку визначають основні можливості, які мають бути реалізовані для забезпечення повного циклу когнітивного тренування: вибір напрямку вправи, вибір рівня складності, виконання завдання, перевірка результату та повернення до подальших дій користувача.

Таблиця 2.1

Функціональні вимоги до мобільного застосунку

Код вимоги	Назва вимоги	Опис вимоги
FR-1	Запуск застосунку	Система має забезпечувати запуск мобільного застосунку та перехід користувача до головного меню.
FR-2	Вибір напрямку тренування	Користувач має змогу обрати тип вправи: логіка, увага, пам'ять або візуальне тренування.
FR-3	Вибір рівня складності	Перед початком виконання вправи користувач має змогу обрати рівень складності: легкий, середній або складний.
FR-4	Запуск тренувального модуля	Після вибору категорії та рівня складності система має відкрити відповідний екран вправи.
FR-5	Виконання вправи	Система має відображати завдання, приймати дії користувача та забезпечувати проходження вправи відповідно до її правил.
FR-6	Перевірка результату	Після завершення вправи система має визначити правильність виконання завдання або факт проходження рівня.
FR-7	Підрахунок балів	Застосунок має нараховувати користувачу бали або інший показник успішності виконання вправи.
FR-8	Відображення результату	Після завершення тренування користувач має отримати повідомлення про результат виконання вправи.
FR-9	Навігація між екранами	Система має забезпечувати переходи між головним меню, екраном вибору складності, тренувальним модулем та екраном результату.
FR-10	Завершення роботи	Користувач має змогу завершити роботу із застосунком або повернутися до головного меню.

Окрему групу становлять функціональні можливості, які можуть бути реалізовані як напрями подальшого розвитку застосунку. До них належать збереження повної історії тренувань, персоналізовані рекомендації, адаптивна зміна складності, нагадування про регулярні заняття та розширена система гейміфікації. Такі можливості не є критичними для базової роботи застосунку,

однак вони можуть підвищити мотивацію користувача та зробити тренувальний процес більш персоналізованим.

Нефункціональні вимоги визначають не окремі дії системи, а якість її роботи. Для мобільного застосунку особливо важливими є зручність використання, швидкодія, стабільність, візуальна узгодженість і можливість подальшого розширення. У моделі якості ISO/IEC 25010 якість програмного продукту розглядається через характеристики функціональної придатності, продуктивності, сумісності, зручності використання, надійності, безпеки, супроводжуваності та переносимості [8]. Саме ці характеристики доцільно враховувати під час формування нефункціональних вимог до застосунку.

Таблиця 2.2

Нефункціональні вимоги до мобільного застосунку

Код вимоги	Характеристика	Опис вимоги
NFR-1	Зручність використання	Інтерфейс має бути простим, інтуїтивно зрозумілим і придатним для користувачів без спеціальної підготовки.
NFR-2	Швидкодія	Перехід між екранами, запуск вправ і відображення результатів мають виконуватися без помітних затримок.
NFR-3	Надійність	Застосунок має коректно працювати під час повторного запуску екранів, повернення назад і завершення вправи.
NFR-4	Розширюваність	Структура застосунку має давати змогу додавати нові вправи та модулі без повної перебудови архітектури.
NFR-5	Візуальна узгодженість	Екрани застосунку мають бути оформлені в єдиному стилі з однаковою логікою розміщення елементів керування.
NFR-6	Доступність	Основні елементи інтерфейсу мають бути достатньо зрозумілими, помітними та зручними для натискання на сенсорному екрані.
NFR-7	Безпека даних	Застосунок не повинен збирати зайві персональні дані користувача; результати тренувань доцільно зберігати локально.
NFR-8	Підтримуваність	Код і структура проєкту мають бути організовані так, щоб спростити подальше тестування, виправлення помилок і розвиток функціональності.

Для застосунку особливо важливою є вимога розширюваності. Оскільки тренувальні вправи можуть мати різну логіку, бажано, щоб нові модулі додавалися без зміни загальної навігації та базової структури

застосунку. Наприклад, після реалізації вправ на логіку можна додати нові вправи на пам'ять або увагу, використовуючи подібний підхід до вибору рівня складності, запуску завдання та відображення результату.

Не менш важливою є вимога зручності використання. Когнітивний тренажер має бути орієнтований на швидку взаємодію користувача із системою. Якщо для початку вправи потрібно виконати занадто багато дій або інтерфейс є перевантаженим, це знижує мотивацію користувача та ускладнює регулярне використання застосунку. Тому під час проєктування інтерфейсу доцільно дотримуватися принципу мінімальної кількості кроків: головне меню, вибір напрямку тренування, вибір складності, виконання вправи, результат.

Вимога швидкодії також є суттєвою, оскільки мобільний застосунок повинен забезпечувати миттєву реакцію на дії користувача. Затримки під час переходу між екранами або виконання вправ можуть негативно впливати на користувацький досвід, особливо у вправах, пов'язаних із концентрацією, швидкістю реакції та увагою. У межах поточної реалізації ця вимога забезпечується локальною роботою застосунку без обов'язкового підключення до віддаленого сервера.

Вимога безпеки у межах цього застосунку пов'язана насамперед із мінімізацією обробки персональних даних. Оскільки застосунок виконує тренувальну функцію і не потребує обов'язкової реєстрації користувача, доцільно зберігати лише технічні або навчальні дані: вибраний рівень складності, кількість балів, результати проходження вправ і налаштування інтерфейсу. Такий підхід зменшує ризики, пов'язані з конфіденційністю, і спрощує архітектуру системи.

Отже, сформульовані функціональні та нефункціональні вимоги визначають основу для подальшого проєктування структури мобільного застосунку. Функціональні вимоги описують основні сценарії взаємодії користувача із системою, а нефункціональні – задають критерії якості, яких необхідно дотримуватися під час реалізації. У сукупності ці вимоги формують

підґрунтя для побудови архітектури застосунку, визначення його компонентів і моделювання логіки роботи тренувальних модулів.

2.3. Архітектура, компоненти та логіка взаємодії модулів застосунку

Архітектура мобільного застосунку визначає загальну організацію його програмних компонентів, принципи взаємодії між екранами, логіку обробки дій користувача та спосіб збереження результатів. Для розроблюваного застосунку обрано просту модульну структуру, яка відповідає функціональним вимогам, сформульованим у попередньому підрозділі, і водночас залишає можливість для подальшого розширення функціональності.

Оскільки застосунок призначений для тренування когнітивних навичок, його архітектура має забезпечувати швидкий доступ користувача до тренувальних модулів, зрозумілу навігацію між екранами, коректну обробку результатів виконання вправ і можливість додавання нових типів завдань. Тому систему доцільно розглядати як сукупність кількох взаємопов'язаних рівнів: рівня інтерфейсу користувача, рівня логіки тренувань і рівня даних. Застосунок побудований за принципом простої модульної архітектури, що включає три основні рівні [12].

1. UI Layer – рівень інтерфейсу користувача. Цей рівень відповідає за відображення екранів застосунку, обробку натискань користувача та організацію переходів між основними частинами програми. До нього належать екрани головного меню, вибору напрямку тренування, вибору рівня складності, виконання вправи, перегляду результату та налаштувань.

Рівень інтерфейсу складається з таких екранів:

- головне меню;
- логіка;
- увага;

- пам'ять;
- візуальні вправи;
- налаштування;
- екран результатів;
- екран вибору рівня складності.

Основна вимога до UI Layer полягає в тому, щоб користувач міг швидко зрозуміти, яку дію потрібно виконати далі. Тому інтерфейс має бути мінімалістичним, із чіткими кнопками, зрозумілими підписами та логічною послідовністю переходів. Наприклад, після вибору розділу «Логіка» користувач переходить до екрана вибору рівня складності, а після завершення вправи – до екрана результатів [19].

2. Logic Layer – рівень логіки тренувальних модулів. Рівень логіки відповідає за правила виконання вправ, генерацію завдань, перевірку дій користувача, зміну складності та підрахунок результатів. Саме цей рівень визначає, як працює кожен тренувальний модуль і яким чином система реагує на дії користувача.

Logic Layer містить алгоритми для кожного типу вправ:

- логічні задачі;
- вправи на увагу;
- вправи на пам'ять;
- візуальні вправи;
- систему підрахунку балів;
- механізм вибору або зміни складності.

Для кожного типу вправи можуть використовуватися різні правила. Наприклад, у логічному модулі користувач може проходити лабіринт або розв'язувати задачу на послідовність дій. У модулі уваги завдання може полягати у пошуку потрібного об'єкта серед інших елементів. У модулі пам'яті користувач може запам'ятовувати порядок об'єктів або їх розташування. Незважаючи на різну логіку вправ, усі модулі мають спільний цикл роботи:

запуск завдання, виконання дії користувачем, перевірка результату, нарахування балів або перехід до завершення рівня.

3. Data Layer – рівень даних. Рівень даних призначений для збереження інформації, необхідної для коректної роботи застосунку. У межах поточної реалізації основний акцент зроблено на локальному збереженні даних, що дає змогу застосунку працювати без обов’язкового підключення до віддаленого сервера.

Data Layer може містити:

- локальне збереження балів;
- збереження вибраного рівня складності;
- збереження налаштувань користувача;
- збереження параметрів інтерфейсу, наприклад теми або звуку.

Для локального збереження даних у мобільному Android-застосунку можуть використовуватися SharedPreferences або Room. SharedPreferences доцільно застосовувати для простих налаштувань, таких як вибрана тема, звук або останній рівень складності. Room є доцільнішим у випадку, якщо потрібно зберігати історію результатів, статистику тренувань або складніші структури даних.

Такий поділ на рівні дає змогу розмежувати відповідальність між компонентами. Рівень інтерфейсу відповідає за взаємодію з користувачем, рівень логіки – за правила роботи вправ, а рівень даних – за збереження результатів і налаштувань. Це спрощує розробку, тестування й подальше розширення застосунку.

Таблиця 2.3

Особливості прецедентів

№	Назва прецеденту	Опис
1	Обрати розділ	Користувач обирає тип вправ: логіка, увага, пам’ять або візуальні вправи.
2	Обрати рівень складності	Користувач вибирає рівень складності вправи: легкий, середній або складний.
3	Виконати вправу	Користувач проходить вибрану вправу відповідно до її правил.

4	Отримати бали	Система нараховує бали або визначає результат після виконання вправи.
5	Переглянути результати	Користувач переглядає результат проходження вправи або попередні результати.
6	Змінити налаштування	Користувач змінює параметри застосунку, зокрема тему, звук або мову.
7	Вийти з програми	Користувач завершує роботу із застосунком.

Для формалізації основних сценаріїв взаємодії користувача із застосунком використано діаграму прецедентів. Вона показує, які дії може виконувати користувач і які функції повинна підтримувати система. Головним актором є користувач, який запускає застосунок, обирає розділ, вибирає рівень складності, виконує вправу, отримує результат, переглядає статистику або змінює налаштування.

Діаграма прецедентів відображає загальну функціональну модель застосунку. Вона не деталізує внутрішню реалізацію окремих модулів, але показує, які можливості мають бути доступні користувачу. Така схема є корисною для перевірки повноти функціональних вимог: кожен основний сценарій користувача має бути представлений відповідним прецедентом.

Наступним етапом моделювання є визначення основних компонентів застосунку. Компонентна модель описує логічні частини системи, їх призначення та роль у загальній роботі застосунку. У межах розроблюваної системи доцільно виділити компоненти, пов'язані з екранами, тренувальними модулями, керуванням складністю, підрахунком балів, налаштуваннями та збереженням даних.

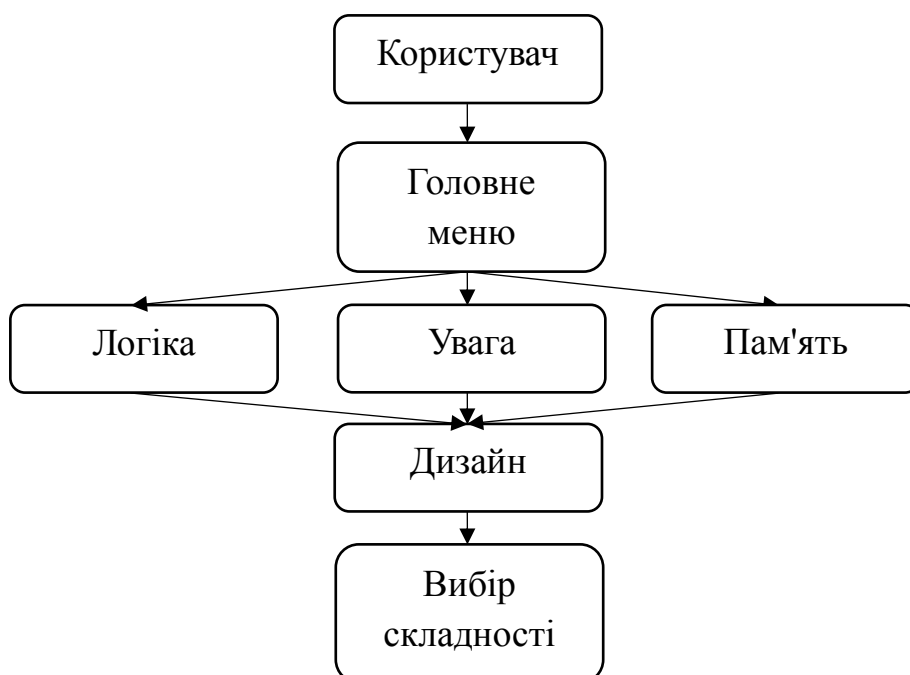




Рисунок 2.1. Схема діаграми прецедентів

Компонент MenuScreen виконує роль центральної точки входу до основних функцій застосунку. Через нього користувач переходить до тренувальних модулів або налаштувань. LogicModule, AttentionModule, MemoryModule і VisualModule відповідають за конкретні групи вправ. DifficultyManager забезпечує вибір рівня складності, а ScoreManager відповідає за обчислення результатів після завершення завдання.

Компонент Storage забезпечує збереження даних між запусками застосунку. У найпростішому варіанті він може зберігати лише останні результати або налаштування користувача. У подальшому цей компонент може бути розширений для збереження повної історії тренувань, побудови статистики та формування рекомендацій.

Логіка взаємодії компонентів під час виконання вправи може бути подана у вигляді діаграми послідовності. Така діаграма демонструє порядок обміну повідомленнями між користувачем, інтерфейсом застосунку, модулем тренування, менеджером результатів і сховищем даних.

Типовий сценарій проходження вправи включає такі етапи:

1. користувач обирає тренувальний модуль;
2. система відкриває екран вибору складності;

3. користувач вибирає рівень складності;
4. відповідний модуль генерує завдання;
5. користувач виконує вправу;
6. система перевіряє результат;
7. ScoreManager нараховує бали або фіксує завершення рівня;
8. Storage зберігає результат;
9. користувач бачить екран завершення вправи.

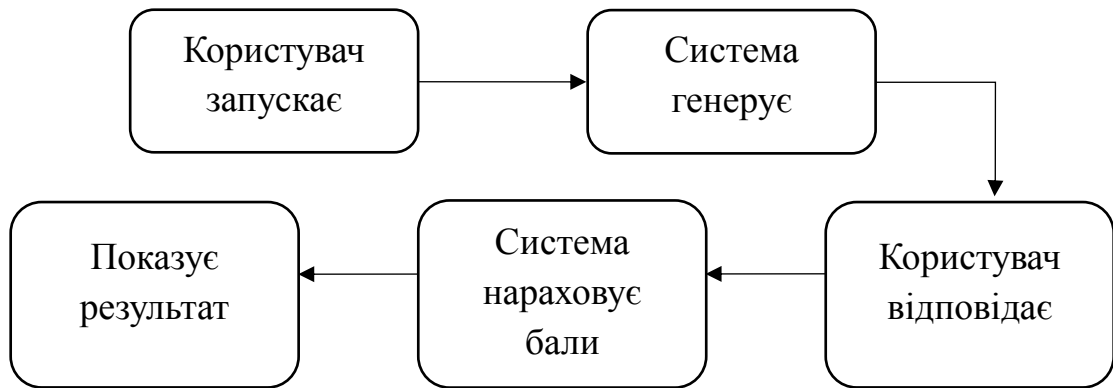


Рисунок 2.2. Схема діаграми послідовності

Діаграма послідовності деталізує базовий сценарій «Пройходження вправи». Вона показує, що взаємодія користувача із застосунком не обмежується лише натисканням кнопок. За кожною дією користувача відбувається внутрішня обробка: генерація завдання, перевірка відповіді, підрахунок результату та відображення підсумку. Такий підхід дає змогу краще зрозуміти, які компоненти повинні бути реалізовані в програмному кодї та як вони мають взаємодіяти між собою.

Окрему увагу під час проєктування потрібно приділити логіці навігації між екранами. Для мобільного застосунку важливо, щоб користувач завжди розумів, де він перебуває і яку дію може виконати далі [23]. Основною точкою навігації є головне меню. З нього користувач переходить до вибору типу вправи, далі – до вибору складності, після чого відкривається екран виконання вправи. Після завершення тренування користувач потрапляє на екран результатів, звідки може повернутися до головного меню або повторити вправу.

Ескізи інтерфейсу використовуються для попереднього опису структури основних екранів. Вони не є остаточним дизайном, але визначають розташування ключових елементів і загальну логіку взаємодії користувача із застосунком.

Головне меню містить кнопки для переходу до основних розділів: логіка, увага, пам'ять, візуальні вправи, налаштування та вихід. Цей екран має бути максимально простим, оскільки саме з нього починається більшість сценаріїв використання.

Екран вибору складності містить три основні варіанти: легкий, середній і складний рівень. Такий підхід дає змогу користувачу самостійно обрати навантаження відповідно до власного досвіду або бажаного рівня складності.

Екран вправи містить саме завдання та мінімальний набір елементів керування. Його структура залежить від типу тренувального модуля. Для логічної вправи це може бути ігрове поле або лабіринт, для вправи на увагу – набір об'єктів для пошуку, для вправи на пам'ять – послідовність елементів або зображень.

Екран результатів відображає підсумок проходження вправи: кількість балів, факт завершення рівня, кількість правильних або неправильних відповідей, а також коротке повідомлення для користувача. Такий екран завершує тренувальний цикл і дає користувачу зрозумілий зворотний зв'язок.

Екран налаштувань може містити параметри звуку, теми інтерфейсу та мови. У базовій реалізації ці налаштування можуть бути мінімальними, однак сама наявність окремого модуля налаштувань робить структуру застосунку більш гнучкою для подальшого розвитку.

Загальна логіка взаємодії модулів застосунку побудована навколо циклу «вибір – виконання – результат». Користувач обирає тип тренування, система визначає відповідний модуль, відкриває потрібний екран, запускає вправу, обробляє результат і повертає користувачу підсумкову інформацію. Така

модель є зрозумілою для користувача та зручною для реалізації, оскільки кожен модуль має чітко визначену відповідальність.

Запропонована архітектура відповідає поставленим вимогам до застосунку. Вона забезпечує просту навігацію, розділення функціональності на окремі модулі, можливість локального збереження результатів і перспективу подальшого розширення. У наступному розділі ця модель використовується як основа для опису практичної реалізації мобільного застосунку, вибору технологій, структури проєкту та програмної реалізації основних екранів і тренувальних модулів.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

3.1. Обґрунтування вибору технологій та організація Android-проєкту

Практична реалізація мобільного застосунку для тренування когнітивних навичок передбачала вибір технологічного стеку, який забезпечує зручну розробку інтерфейсу, стабільну роботу на мобільних пристроях, можливість подальшого розширення функціональності та підтримку інтерактивної взаємодії з користувачем. Оскільки розроблюваний застосунок орієнтований на виконання коротких когнітивних вправ, важливими критеріями вибору технологій були швидкодія, простота реалізації користувацьких сценаріїв, підтримка анімацій, можливість збереження локальних даних і доступність інструментів тестування.

Для реалізації застосунку було обрано платформу Android. Такий вибір зумовлений широким поширенням Android-пристроїв, відкритістю платформи, наявністю розвиненої екосистеми інструментів розробки та достатніми можливостями для створення інтерактивних мобільних застосунків. Android дає змогу реалізовувати застосунки з різними типами екранів, підтримувати обробку натискань, переходи між активностями, анімаційні ефекти, локальне збереження даних і роботу з графічними ресурсами. Для застосунку, метою якого є тренування пам'яті, уваги та логічного мислення, ці можливості є достатніми для створення повноцінного навчально-тренувального середовища [4].

Основною мовою програмування було обрано Kotlin. Kotlin є сучасною мовою програмування, яка активно використовується для Android-розробки та добре інтегрується з Android SDK [3]. Її перевагами є лаконічний синтаксис, підтримка об'єктно-орієнтованого підходу, безпечніша робота з null-

значеннями, зручна обробка подій інтерфейсу та сумісність із наявними бібліотеками Android. Для проєкту Kotlin є доцільним вибором, оскільки дає змогу зменшити кількість шаблонного коду й зосередитися на реалізації логіки застосунку, поведінки екранів і сценаріїв взаємодії користувача [10].

Середовищем розробки було обрано Android Studio, яке є основним інструментом для створення Android-застосунків. Воно містить редактор коду, засоби роботи з XML-макетами, емулятор мобільних пристроїв, інструменти налагодження, перегляд структури проєкту, систему керування ресурсами та підтримку Gradle-збірки. Використання Android Studio спростило процес створення екранів застосунку, підключення графічних ресурсів, запуску програми на емуляторі або фізичному пристрої, а також перевірки коректності роботи окремих активностей.

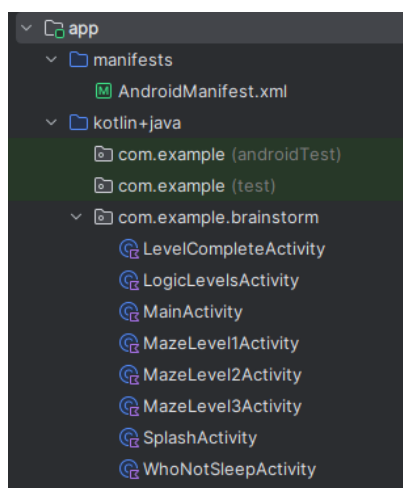


Рисунок 3.1. Структура програмного коду Android-проєкту

Для побудови інтерфейсу користувача було використано XML-розмітку. Такий підхід є класичним для Android-розробки та дає змогу окремо описувати візуальну структуру екранів і програмну логіку їхньої роботи. XML-макети застосовуються для визначення розташування кнопок, текстових елементів, зображень, фонів та інших компонентів інтерфейсу. У межах розробленого застосунку XML-розмітка використовується для опису стартового екрана, головного меню, екрана вибору рівнів, екранів ігрових вправ та екрана завершення рівня.

Архітектурно застосунок побудовано як набір взаємопов'язаних активностей, кожна з яких відповідає за окремий екран або функціональний сценарій. Такий підхід є зрозумілим, оскільки застосунок має порівняно невелику кількість функціональних модулів і не потребує складної серверної частини. Водночас структура проєкту залишає можливість подальшого розвитку: додавання нових вправ, реалізації модуля статистики, розширення налаштувань, збереження прогресу користувача або підключення локальної бази даних.

У межах роботи було використано спрощений підхід до організації логіки, близький до MVVM, однак без надмірного ускладнення архітектури. Основна логіка взаємодії з користувачем зосереджена в активностях, а візуальна частина винесена в XML-макети. Така структура відповідає навчальному характеру проєкту та дає змогу чітко простежити зв'язок між екранами, подіями інтерфейсу та результатами дій користувача. У подальшому, за потреби масштабування застосунку, логіку тренувань можна винести в окремі класи або ViewModel-компоненти, а збереження результатів організувати через Room або інше локальне сховище.

Проєкт організовано в межах основного модуля app, який містить програмний код, ресурси інтерфейсу, графічні матеріали, конфігураційні файли та опис активностей. У структурі коду реалізовано кілька основних активностей: `SplashActivity`, `MainActivity`, `LogicLevelsActivity`, `MazeLevel1Activity`, `MazeLevel2Activity`, `MazeLevel3Activity`, `WhoNotSleepActivity` та `LevelCompleteActivity`. Кожна з них виконує окрему функцію в межах загальної логіки застосунку. Наприклад, `SplashActivity` відповідає за стартовий екран і початкову анімацію, `MainActivity` реалізує головне меню, `LogicLevelsActivity` забезпечує вибір рівня логічної гри, а активності `MazeLevel1Activity`, `MazeLevel2Activity` і `MazeLevel3Activity` відповідають за проходження окремих рівнів.

Ресурсна частина проєкту розміщена в каталозі `res`, який містить кілька важливих підкаталогів. У каталозі `layout` зберігаються XML-макети екранів,

зокрема `activity_main.xml`, `activity_splash.xml`, `activity_logic_levels.xml`, `activity_maze_level1.xml`, `activity_maze_level2.xml`, `activity_who_not_sleep.xml` та `activity_level_complete.xml`. У каталозі `drawable` розміщено графічні ресурси, фони, елементи лабіринту, зображення та XML-описи візуальних компонентів. Каталог `miptar` містить адаптивні іконки застосунку для різних щільностей екрана. У каталозі `values` зберігаються файли `colors.xml`, `strings.xml`, `styles.xml` і `themes.xml`, які визначають кольори, текстові ресурси, стилі та теми інтерфейсу.

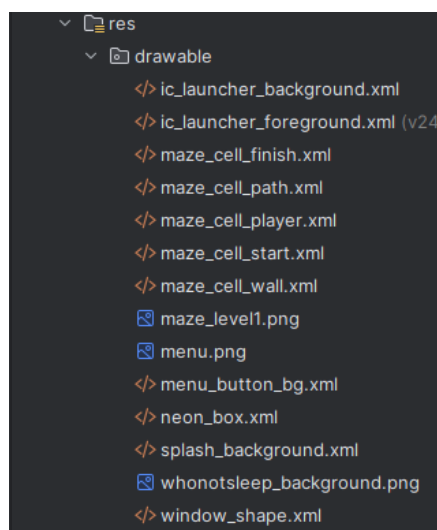


Рисунок 3.2. XML-макети та конфігураційні ресурси інтерфейсу

Окрему роль у структурі застосунку відіграють графічні ресурси. Вони використовуються для формування візуального стилю, побудови ігрових екранів, оформлення кнопок, фонів і елементів вправ. Застосування окремих ресурсних файлів дає змогу відокремити графічне оформлення від програмної логіки. Це спрощує підтримку проєкту, оскільки зміна зовнішнього вигляду окремих елементів не потребує істотного переписування Kotlin-коду [10].

Загальна організація Android-проєкту забезпечує зрозумілу структуру та логічний поділ між програмним кодом, макетами інтерфейсу й ресурсами. Такий підхід є доцільним для реалізації мобільного застосунку когнітивного спрямування, оскільки він поєднує простоту розробки, достатню функціональність і можливість подальшого розширення. Обраний технологічний стек дає змогу реалізувати інтерактивні вправи, організувати

навігацію між екранами, використовувати анімації, зберігати базові результати користувача та поступово розвивати застосунок до повноцінного тренувального інструменту.

3.2. Реалізація функціональних модулів та користувацького інтерфейсу застосунку

Реалізація мобільного застосунку для тренування когнітивних навичок здійснювалася шляхом поетапного створення окремих функціональних модулів, кожен із яких відповідає за певний сценарій взаємодії користувача із системою. Основу застосунку становить набір активностей Android, пов'язаних між собою через механізм переходів Intent. Такий підхід дає змогу розділити програму на окремі логічні екрани, спростити підтримку коду та забезпечити зрозумілу навігацію між основними частинами застосунку [12].

Функціональна структура застосунку охоплює стартовий екран, головне меню, модуль вибору логічних рівнів, модуль проходження вправ, модуль тренування уваги та екран завершення рівня. Кожна активність має відповідний XML-макет, у якому визначено розташування елементів інтерфейсу, а також Kotlin-клас, що описує логіку роботи екрана. Такий розподіл дає змогу відокремити візуальне представлення від програмної поведінки застосунку.

Стартовий модуль реалізовано за допомогою `SplashActivity`. Його призначення полягає у відображенні початкового екрана застосунку, формуванні першого візуального враження та плавному переході до головного меню. На цьому екрані використано назву застосунку, фонове зображення та анімаційні ефекти. Зокрема, реалізовано появу заголовка за допомогою анімації прозорості та появу нижнього текстового елемента з одночасним рухом угору. Після короткої затримки застосунок автоматично відкриває `MainActivity`, що відповідає за головне меню.

Така реалізація стартового екрана має не лише декоративне, але й функціональне значення. Вона дає змогу приховати короткий проміжок початкового завантаження застосунку, підготувати користувача до взаємодії з програмою та надати застосунку завершеного вигляду. Для навчально-тренувального застосунку, орієнтованого на щоденне використання, це важливо, оскільки перший екран створює загальне враження про якість програмного продукту.

Головне меню реалізовано в MainActivity. Цей екран є центральною точкою навігації, з якої користувач може перейти до різних типів вправ або завершити роботу із застосунком. У межах наявної реалізації головне меню містить кнопки для переходу до тренування логіки, уваги, пам'яті, налаштувань, а також кнопку виходу. Натискання відповідної кнопки обробляється через слухачі подій `setOnClickListener`, після чого створюється об'єкт `Intent` і відкривається потрібна активність.

Особливістю головного меню є його простота та орієнтація на швидкий вибір дії. Користувач не повинен проходити складну авторизацію або налаштовувати профіль перед початком тренування. Це відповідає концепції коротких когнітивних вправ, які мають бути доступними без зайвих підготовчих кроків. Такий підхід також зменшує когнітивне навантаження на користувача, оскільки всі основні дії представлені у вигляді зрозумілих кнопок [13].

Рисунок 3.5. Головне меню застосунку

Модуль логічних вправ починається з екрана `LogicLevelsActivity`. Його основне завдання полягає у відображенні доступних рівнів логічної гри та організації переходу до конкретного рівня. У застосунку передбачено кілька рівнів гри “Лабіринт”, кожен із яких реалізований окремою активністю: `MazeLevel1Activity`, `MazeLevel2Activity` та `MazeLevel3Activity`. Такий підхід дає змогу поетапно ускладнювати завдання та забезпечувати користувачеві відчуття поступового прогресу.

Логіка екрана вибору рівнів передбачає ініціалізацію кнопок для кожного рівня, обробку натискань і відкриття відповідної активності. Додатково може використовуватися перевірка доступності рівнів залежно від попереднього прогресу користувача. Наприклад, перший рівень може бути доступним одразу, а наступні – лише після успішного проходження попередніх. Такий механізм створює базову ігрову мотивацію та підтримує послідовність проходження вправ.

Рівні гри “Лабіринт” реалізують логічно-просторову вправу, у якій користувач має пройти шлях від початкової точки до завершення. У межах кожної активності рівня визначаються елементи ігрового поля, графічні ресурси клітинок, правила переміщення та умови завершення вправи. Для побудови візуальної частини використано окремі графічні елементи, що позначають стартову клітинку, шлях, стіни, позицію гравця та фініш. Це дає змогу зробити вправу зрозумілою для користувача без додаткових текстових інструкцій.

Механіка лабіринту спрямована на тренування логічного мислення, просторового орієнтування та планування дій. Користувач має аналізувати доступні напрями руху, уникати помилкових шляхів і знаходити оптимальну траєкторію до фінішу. Поступове ускладнення рівнів дає змогу формувати ефект навчання: після проходження простішого рівня користувач краще розуміє логіку гри та може ефективніше взаємодіяти зі складнішими завданнями.

Окремий функціональний модуль реалізовано в WhoNotSleepActivity. Він орієнтований на тренування уваги та пам’яті. За своїм призначенням цей модуль відрізняється від лабіринту, оскільки акцент робиться не на просторовому плануванні, а на концентрації, спостережливості та швидкому розпізнаванні потрібних елементів. Така вправа урізноманітнює зміст застосунку й демонструє можливість поєднання різних типів когнітивних тренувань у межах одного мобільного продукту.

Екран завершення рівня реалізовано за допомогою `LevelCompleteActivity`. Його призначення полягає у відображенні результату після виконання вправи та наданні користувачеві можливості перейти далі або повернутися до попереднього меню. Наявність такого екрана є важливою з погляду користувацького досвіду, оскільки завершення вправи повинно мати чітке візуальне підтвердження. Користувач має розуміти, що завдання виконано, а його прогрес зафіксовано.

У застосунку передбачено збереження базового прогресу користувача за допомогою `SharedPreferences`. Це локальний механізм збереження простих даних, який доцільно використовувати для фіксації стану проходження рівнів, відкриття наступних завдань або збереження окремих параметрів роботи застосунку. Для поточного етапу реалізації такого підходу достатньо, оскільки застосунок не потребує складної бази даних, облікових записів або синхронізації з сервером.

Користувацький інтерфейс застосунку побудовано на основі XML-макетів. Для кожної активності створено окремий файл розмітки, що описує структуру відповідного екрана. Наприклад, `activity_splash.xml` відповідає за стартовий екран, `activity_main.xml` – за головне меню, `activity_logic_levels.xml` – за вибір рівнів, а `activity_maze_level1.xml`, `activity_maze_level2.xml` та інші файли – за екрани конкретних вправ. Такий підхід спрощує внесення змін до інтерфейсу, оскільки макети можна редагувати незалежно від основної логіки Kotlin-класів.

Під час проєктування інтерфейсу було враховано принцип простоти. Екрани не перевантажені зайвими елементами, а основні дії користувача представлені у вигляді великих і зрозумілих кнопок. Це особливо важливо для застосунків когнітивного спрямування, оскільки інтерфейс не повинен відволікати від виконання вправи. Візуальна структура має бути інтуїтивною, а користувач повинен швидко розуміти, яку дію потрібно виконати.

Графічні ресурси застосунку розміщені в каталозі `drawable`. До них належать фонові зображення, елементи лабіринту, графічне оформлення

кнопок, зображення для окремих вправ та XML-описи форм. Використання окремих ресурсів дає змогу підтримувати єдиний стиль застосунку й уникати дублювання графічних елементів у різних частинах проєкту. Наприклад, елементи лабіринту можуть повторно використовуватися в кількох рівнях, що спрощує розширення гри новими завданнями.

Для оформлення застосунку також використовуються ресурси з каталогів `values` і `ipmap`. У файлах `colors.xml`, `strings.xml`, `styles.xml` і `themes.xml` задаються кольори, текстові ресурси, стилі та загальна тема інтерфейсу. Винесення таких параметрів в окремі файли є важливим для підтримки проєкту, оскільки дає змогу централізовано змінювати візуальні характеристики застосунку. Каталог `ipmap` використовується для зберігання іконок застосунку, адаптованих до різної щільності екранів Android-пристроїв.

Загалом реалізація функціональних модулів побудована так, щоб кожен екран виконував чітко визначену роль у взаємодії користувача із застосунком. `SplashActivity` забезпечує старт застосунку, `MainActivity` організовує основну навігацію, `LogicLevelsActivity` відповідає за вибір рівнів, активності лабіринту реалізують логічні вправи, `WhoNotSleepActivity` забезпечує вправу на увагу, а `LevelCompleteActivity` завершує сценарій проходження завдання. Така структура є достатньо простою для реалізації, але водночас дає змогу масштабувати застосунок шляхом додавання нових активностей, вправ і сценаріїв.

Розроблений інтерфейс і функціональні модулі забезпечують базову цілісність мобільного застосунку. Користувач може запустити програму, перейти з початкового екрана до головного меню, обрати тип вправи, виконати завдання, отримати підтвердження завершення рівня та продовжити взаємодію із застосунком. Це підтверджує, що реалізований продукт не є лише статичним макетом, а функціонує як робочий мобільний застосунок із навігацією, інтерактивними елементами, графічними ресурсами та локальним збереженням окремих даних.

3.3. Тестування, оцінювання результатів роботи та напрями вдосконалення застосунку

Після реалізації основних функціональних модулів мобільного застосунку було проведено тестування його працездатності. Метою тестування була перевірка коректності запуску застосунку, відображення інтерфейсу, навігації між екранами, роботи інтерактивних елементів, проходження вправ і збереження базового прогресу користувача. Оскільки застосунок має локальний характер і не використовує серверну частину, основну увагу було приділено функціональному тестуванню та перевірці користувацького інтерфейсу.

Тестування проводилося в середовищі Android Studio з використанням емулятора Android та ручної перевірки основних сценаріїв взаємодії. Такий підхід є доцільним для поточного етапу розробки, оскільки застосунок має обмежену кількість екранів і чітко визначені сценарії роботи: запуск, перехід до головного меню, вибір вправи, проходження рівня, завершення завдання та повернення до меню.

Основні напрями тестування охоплювали:

1. запуск застосунку, відображення стартового екрана та перехід до головного меню;
2. коректність навігації між основними екранами застосунку;
3. запуск різних категорій когнітивних вправ;
4. виконання вправ, обробку дій користувача та завершення завдань;
5. відображення результатів після проходження вправ;
6. збереження базового прогресу користувача;
7. коректність відображення графічних ресурсів, кнопок, фонів та XML-макетів;
8. стабільність роботи застосунку під час повторного запуску вправ і повернення до меню.

Таблиця 3.1

Результати функціонального тестування мобільного застосунку

№	Тестовий сценарій	Очікуваний результат	Фактичний результат	Статус
1	Запуск мобільного застосунку	Застосунок запускається без помилок, відображається стартовий екран	Стартовий екран відображається коректно	Пройдено
2	Перехід зі стартового екрана до головного меню	Після завершення стартової анімації відкривається головне меню	Перехід до головного меню виконується коректно	Пройдено
3	Відображення головного меню	Користувач бачить основні розділи застосунку та кнопки переходу до вправ	Елементи головного меню відображаються коректно	Пройдено
4	Перехід до категорій когнітивних вправ	Після натискання відповідної кнопки відкривається вибрана категорія вправ	Перехід до доступних категорій виконується без помилок	Пройдено
5	Запуск вправ різних типів	Вибрана вправа відкривається та готова до виконання	Вправи запускаються відповідно до вибору користувача	Пройдено
6	Виконання інтерактивних дій у вправах	Застосунок реагує на натискання, вибір елементів або інші дії користувача	Інтерактивні елементи працюють коректно	Пройдено
7	Завершення вправи	Після виконання завдання користувач переходить до екрана завершення або отримує результат	Завершення вправ відбувається коректно	Пройдено
8	Відображення результатів проходження	Користувач бачить повідомлення про завершення вправи або результат виконання	Результати відображаються коректно	Пройдено
9	Збереження базового прогресу	Після проходження вправи прогрес або стан користувача зберігається локально	Збереження через SharedPreferences працює коректно	Пройдено
10	Повторний запуск вправи	Користувач може повторно відкрити вправу без аварійного завершення застосунку	Повторний запуск виконується стабільно	Пройдено
11	Повернення до попередніх екранів	Користувач може повернутися до меню або екрана вибору вправ	Навігація назад працює коректно	Пройдено
12	Відображення графічних ресурсів	Фони, кнопки, іконки та ігрові елементи відображаються без спотворень	Графічні ресурси відображаються коректно	Пройдено
13	Відображення XML-макетів на екрані	Елементи інтерфейсу розміщені коректно та не перекривають один одного	Макети відображаються коректно	Пройдено
14	Завершення роботи із застосунком	Після натискання кнопки виходу застосунок завершує роботу	Завершення роботи виконується коректно	Пройдено

Результати тестування засвідчили, що реалізовані функціональні модулі працюють відповідно до поставлених вимог. Застосунок коректно запускається, відображає стартовий екран, виконує перехід до головного меню, забезпечує навігацію між основними екранами та дає змогу запускати реалізовані вправи. Графічні ресурси відображаються коректно, а XML-макети забезпечують цілісне візуальне представлення інтерфейсу.

Окремо було перевірено логіку переходів між активностями. Використання механізму Intent забезпечило стабільну навігацію між стартовим екраном, головним меню, екраном вибору рівнів, ігровими екранами та екраном завершення рівня. Під час тестування не було виявлено критичних помилок, які унеможлилювали б проходження основних сценаріїв використання.

Перевірка користувацького інтерфейсу показала, що структура екранів є зрозумілою для користувача. Основні дії представлені у вигляді кнопок, а перехід між екранами не потребує додаткових інструкцій. Це відповідає початковій вимозі щодо простоти й доступності застосунку. Для мобільного застосунку когнітивного спрямування така характеристика є важливою, оскільки інтерфейс не повинен відволікати користувача від виконання вправ.

Оцінювання результатів роботи застосунку дає підстави зробити висновок, що створений програмний продукт реалізує базову функціональність мобільного тренажера когнітивних навичок. У застосунку реалізовано стартовий екран, головне меню, логічні вправи, вправи на увагу та пам'ять, екран завершення рівня, графічні ресурси та локальне збереження окремих результатів. Це підтверджує, що розробка не обмежується статичним макетом, а має ознаки робочого Android-застосунку.

Водночас застосунок має низку обмежень, які визначають напрями його подальшого розвитку. На поточному етапі частина запланованих функцій реалізована базово або потребує розширення. Зокрема, доцільно вдосконалити модуль пам'яті, додати повноцінний розділ налаштувань, розширити кількість

вправ, реалізувати детальнішу статистику результатів і додати адаптивну зміну складності залежно від успішності користувача.

Перспективними напрямками вдосконалення застосунку є:

1. Розширення набору когнітивних вправ для пам'яті, уваги, логіки та швидкості реакції.
2. Реалізація повноцінної статистики проходження вправ.
3. Додавання адаптивної складності, що змінюється відповідно до результатів користувача.
4. Розробка модуля налаштувань із вибором теми, звуку та мови інтерфейсу.
5. Удосконалення дизайну екранів і покращення адаптивності для різних розмірів дисплеїв.
6. Перехід від SharedPreferences до Room у разі потреби зберігати детальнішу історію результатів.
7. Додавання автоматизованих тестів для перевірки стабільності ключових сценаріїв.

Таким чином, проведене тестування підтвердило працездатність основних модулів мобільного застосунку. Реалізований продукт забезпечує базові сценарії тренування когнітивних навичок, має зрозумілий інтерфейс, стабільну навігацію та може бути використаний як основа для подальшого розвитку. Подальше вдосконалення доцільно спрямувати на розширення функціональності, покращення аналітики результатів і підвищення якості користувацького досвіду.

ВИСНОВКИ

У кваліфікаційній роботі було розглянуто питання проєктування та реалізації мобільного застосунку для тренування когнітивних навичок користувача. Актуальність теми зумовлена зростанням ролі цифрових інструментів у повсякденному житті, навчанні та професійній діяльності, а також потребою у доступних і зручних програмних рішеннях для розвитку пам'яті, уваги, логічного мислення та швидкості реакції.

Відповідно до першого завдання було проаналізовано предметну область розвитку когнітивних навичок та існуючі мобільні застосунки для тренування пам'яті, уваги й логічного мислення. Встановлено, що сучасні рішення пропонують широкий набір інтерактивних вправ, проте часто мають певні недоліки: обмежений безкоштовний функціонал, перевантажений інтерфейс, складну навігацію або надмірну залежність від платних підписок. Проведений аналіз дав змогу визначити доцільність створення простого Android-застосунку, орієнтованого на базові когнітивні тренування та зручну взаємодію користувача із системою.

У межах другого завдання було сформовано функціональні та нефункціональні вимоги до мобільного застосунку. До основних функціональних можливостей віднесено вибір типу тренування, запуск вправ, вибір рівня складності, виконання завдань, підрахунок результатів і перехід між основними екранами. Серед нефункціональних вимог визначено простоту використання, стабільність роботи, зрозумілу навігацію, візуальну привабливість інтерфейсу та можливість подальшого розширення функціоналу.

Відповідно до третього завдання було спроектовано загальну структуру застосунку, основні сценарії взаємодії користувача із системою та логіку роботи тренувальних модулів. Архітектура застосунку передбачає поділ на інтерфейсний рівень, рівень логіки вправ і рівень збереження даних. Було визначено основні екрани застосунку: стартовий екран, головне меню, екрани вибору розділів і рівнів складності, екрани виконання вправ та екран

завершення рівня. Такий підхід забезпечив логічну організацію програмного продукту та спростив його подальшу реалізацію.

У межах четвертого завдання було реалізовано мобільний застосунок на платформі Android із використанням мови програмування Kotlin, середовища Android Studio та XML-розмітки інтерфейсу. У процесі розробки створено основні активності застосунку, реалізовано переходи між екранами, графічні ресурси, логіку виконання вправ та відображення результатів. Розроблений застосунок є завершеним програмним продуктом, який може бути встановлений і використаний на Android-пристроях. Програмний код проєкту буде розміщено у репозиторії GitHub, що забезпечить можливість його перегляду, збереження, подальшого розвитку та демонстрації результатів виконаної роботи.

Відповідно до п'ятого завдання було проведено тестування основних функцій застосунку. Перевірено коректність запуску програми, роботу навігації між екранами, відкриття тренувальних модулів, вибір рівнів складності, виконання вправ і відображення результатів. За результатами тестування встановлено, що реалізований застосунок працює стабільно, основні сценарії використання виконуються коректно, а інтерфейс є зрозумілим для користувача.

Отже, мету кваліфікаційної роботи досягнуто: спроектовано та реалізовано мобільний застосунок для тренування когнітивних навичок користувача на платформі Android. Практичне значення роботи полягає у створенні програмного продукту, який може використовуватися для самостійного тренування пам'яті, уваги та логічного мислення, а також як основа для подальшого розширення. У перспективі застосунок можна доповнити системою збереження статистики, персоналізованими рекомендаціями, новими типами вправ, профілем користувача, адаптивною складністю та синхронізацією результатів із хмарним сховищем.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Anderson J. R. Cognitive Psychology and Its Implications. 8th ed. New York : Worth Publishers, 2014. 544 p.
2. Android Developers. Core app quality. Google. URL: <https://developer.android.com/docs/quality-guidelines/core-app-quality> (дата звернення: 01.12.2025).
3. Android Developers. Develop Android apps with Kotlin. Google. URL: <https://developer.android.com/kotlin> (дата звернення: 17.12.2025).
4. Android Developers. Guide to app architecture. Google. URL: <https://developer.android.com/topic/architecture> (дата звернення: 08.01.2026).
5. Elevate Labs. Elevate: Brain Training Games. URL: <https://www.elevateapp.com/> (дата звернення: 24.05.2026).
6. Federal Trade Commission. Lumosity to Pay \$2 Million to Settle FTC Deceptive Advertising Charges for Its “Brain Training” Program. 2016. URL: <https://www.ftc.gov/news-events/news/press-releases/2016/01/lumosity-pay-2-million-settle-ftc-deceptive-advertising-charges-its-brain-training-program> (дата звернення: 21.11.2025).
7. Hampshire A. et al. A large-scale, cross-sectional investigation into the efficacy of brain training. *Frontiers in Human Neuroscience*. 2019. Vol. 13. Article 221. DOI: 10.3389/fnhum.2019.00221.
8. ISO/IEC 25010:2023. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model. Geneva : ISO, 2023.
9. ISO/IEC/IEEE 29148:2018. Systems and software engineering — Life cycle processes — Requirements engineering. Geneva : ISO, 2018.
10. Kotlin Foundation. Kotlin documentation. URL: <https://kotlinlang.org/docs/home.html> (дата звернення: 05.11.2025).

11. Lumos Labs. Lumosity: Brain Training. URL: <https://www.lumosity.com/> (дата звернення: 27.10.2025).
12. Material Design. Designing usable mobile interfaces. Google. URL: <https://m3.material.io/> (дата звернення: 15.02.2026).
13. Nielsen Norman Group. Mobile User Experience. URL: <https://www.nngroup.com/topic/mobile-and-tablet-design/> (дата звернення: 12.03.2026).
14. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data. General Data Protection Regulation. URL: <https://eur-lex.europa.eu/eli/reg/2016/679/oj> (дата звернення: 24.05.2026).
15. Simons D. J. et al. Do “brain-training” programs work? Psychological Science in the Public Interest. 2016. Vol. 17, No. 3. P. 103–186. DOI: 10.1177/1529100616661983.
16. Sternberg R. J., Sternberg K. Cognitive Psychology. 7th ed. Boston : Cengage Learning, 2016. 640 p.
17. Synaptikon GmbH. NeuroNation Brain Training. URL: <https://www.neuronation.com/> (дата звернення: 11.11.2025).
18. Winters T., Manshreck T., Wright H. Software Engineering at Google: Lessons Learned from Programming Over Time. Sebastopol : O'Reilly Media, 2020. 602 p.
19. World Wide Web Consortium. Web Content Accessibility Guidelines (WCAG) 2.2. W3C Recommendation, 2023. URL: <https://www.w3.org/TR/WCAG22/> (дата звернення: 26.11.2025).
20. Белан В. Формування культури цифрової етики у здобувачів професійної освіти. Київ : НАПН України, 2024.
21. Ворожбіт-Горбатюк В. В. Цифрова етика: компоненти у сфері освіти і педагогічної діяльності. Професійна педагогіка. 2025.

22. Дерев'янка С. П. Когнітивна психологія : навчально-методичні рекомендації. Чернігів : НУЧК імені Т. Г. Шевченка, 2021. 80 с.
23. Ічанська Н. В., Улько С. І. Основні аспекти створення мобільних додатків та вибір інструментів їх розробки. Полтава : Національний університет «Полтавська політехніка імені Юрія Кондратюка».
24. Коч Н. В. Когнітивна лінгвістика : навчальний посібник для студентів вищих навчальних закладів. Миколаїв : Іліон, 2021. 252 с.