

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка мобільного застосунку-гри для підтримки процесів відновлення когнітивних функцій людини мовою програмування Java для платформи Android»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Микита НАУМОВ
(підпис)

Виконав: здобувач вищої освіти групи ПД-44

_____ Микита НАУМОВ

Керівник: _____ Оксана ЗОЛОТУХІНА
к.т.н., доцент

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерія програмного забезпечення

Ірина ЗАМРІЙ

“ ____ ” _____ 2024 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Наумову Микиті Сергійовичу

1. Тема кваліфікаційної роботи: «Розробка мобільного застосунку-гри для підтримки процесів відновлення когнітивних функцій людини мовою програмування Java для платформи Android»

керівник кваліфікаційної роботи к.т.н. доцент Оксана ЗОЛОТУХІНА,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 року № 36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи опис методів та засобів відновлення когнітивних функцій; науково-технічна література з розробки ігрових застосунків мовою Java для платформи Android.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження та аналіз предметної галузі відновлення когнітивних функцій людини та існуючих застосунків для підтримки процесів відновлення когнітивних функцій людини.

2. Аналіз та розробка вимог до мобільного застосунку-гри для підтримки процесів відновлення когнітивних функцій людини.

3. Проектування ігрових рівнів з урахуванням завдань підтримки відновлення

когнітивних функцій людини.

4. Реалізація геймплейних механік за допомогою платформи Android Studio та мови програмування Java.

5. Тестування застосунку.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до додатку.
3. Програмні засоби реалізації.
4. Діаграма класів.
5. Діаграма діяльності.
6. Алгоритм роботи застосунку.
7. Екранні форми.
8. Апробація результатів дослідження.

6. Дата видачі завдання «28» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Визначення актуальності підтримки процесів відновлення	28.02.-06.03.2024	
2.	Дослідження та аналіз предметної галузі відновлення когнітивних функцій людини та існуючих застосунків для підтримки процесів відновлення когнітивних функцій людини.	07.03-13.03.2024	
3.	Аналіз та розробка вимог до мобільного застосунку-гри для підтримки процесів відновлення когнітивних функцій людини.	14.03-29.03.2024	
4.	Проектування та розробка мобільного застосунку для підтримки процесів відновлення когнітивних функцій людини.	30.03-30.04.2024	
5.	Тестування застосунку	20.04-05.05.2024	
7.	Оформлення роботи: вступ, висновки, реферат	29.04-05.05.2024	
8.	Розробка демонстраційних матеріалів	06.05-12.05.2024	
9.	Попередній захист роботи	13.05-31.05.2024	

Здобувач вищої освіти

(підпис)

Микита НАУМОВ

Керівник
кваліфікаційної роботи

(підпис)

Оксана ЗОЛОТУХІНА

РЕФЕРАТ

Текстова частина кваліфікаційної роботи: 74 сторінок, 37 рисунки, 4 таблиці та 20 джерел.

Об'єкт дослідження – процеси відновлення когнітивних функцій людини.

Предмет дослідження – програмне забезпечення для підтримки процесів відновлення когнітивних функцій людини.

Методи дослідження – опрацювання літератури за даною темою, аналіз експлуатаційної документації.

У роботі досліджено проблему підтримки процесів відновлення когнітивних функцій людини після різних захворювань та ушкоджень мозку з використанням мобільного застосунку, що має ігровий формат представлення вправ для відновлення. Виявлено, що використання мобільних додатків для підтримки процесів відновлення когнітивних функцій може значно збільшити ефективність терапії. Такі додатки можуть включати різноманітні завдання, ігри та тренувальні вправи, спрямовані на покращення пам'яті, уваги, мовлення, мислення та інших когнітивних функцій. Вони дозволяють індивідуалізувати програму відновлення відповідно до потреб конкретного користувача, а також відстежувати й аналізувати його прогрес у реальному часі. В роботі спроектовано та реалізовано ігрові завдання та відповідні ігрові механіки для підтримки відновлення когнітивних функцій: уважність, пам'ять, сприйняття та відтворення побаченої інформації. Застосунок розроблено з використанням платформи Android Studio та мови програмування Java. Проведено тестування застосунку з використанням інструментованого тесту, локальних юніт-тестів та мануальних тестів

Галузь використання – реабілітаційні заходи для підтримки та поліпшення когнітивних функцій.

КЛЮЧОВІ СЛОВА: JAVA, ОС ANDROID, МОБІЛЬНИЙ ЗАСТОСУНОК, КОГНІТИВНІ ФУНКЦІЇ, ВІДНОВЛЕННЯ.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ КОГНІТИВНИХ ПРОЦЕСІВ ТА ЇХ ВІДНОВЛЕННЯ	11
1.1 Аналіз видів когнітивних функцій	11
1.2 Огляд методів та засобів відновлення когнітивних функцій.....	15
1.3 Аналіз мобільного додатку, як предмету дослідження	15
1.4 Порівняльний аналіз додатків для відновлення когнітивних функцій	17
Висновки до першого розділу	24
2 ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДТРИМКИ ПРОЦЕСІВ ВІДНОВЛЕННЯ КОГНІТИВНИХ ФУНКЦІЙ.....	25
2.1. Функціональні та нефункціональні вимоги до застосунку. Варіанти використання	25
2.2 Проектування інтерфейсу користувача.....	27
Висновки до другого розділу	33
3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДТРИМКИ ПРОЦЕСІВ ВІДНОВЛЕННЯ КОГНІТИВНИХ ФУНКЦІЙ.....	34
3.1 Обґрунтування вибору платформи та технологій розробки застосунку	34
3.2 Розробка архітектури застосунку	35
3.3 Опис реалізованих класів.....	36
3.3 Опис функціонування програми.....	55
Висновки до третього розділу	59
4 ТЕСТУВАННЯ ЗАСТОСУНКУ ДЛЯ ВІДНОВЛЕННЯ КОГНІТИВНИХ ФУНКЦІЙ.....	60
4.1 Обґрунтування вибору тест-кейсів	60
4.2 Розробка та виконання тест-кейсів	61
Висновки до четвертого розділу	67
ВИСНОВКИ	69
ПЕРЕЛІК ПОСИЛАНЬ	70
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	73
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ	81

ВСТУП

Актуальність дослідження. Актуальність теми полягає в тому, що в сучасному світі зростає зацікавленість у здоровому способі життя та підтримці психічного здоров'я. Захворювання та ушкодження мозку, такі як травми голови, інсульти та інші захворювання, можуть призвести до порушень когнітивних функцій людини, таких як пам'ять, увага, мовлення та інші.

Розробка мобільного додатку для підтримки процесів відновлення когнітивних функцій може значно полегшити життя людей, які стикаються з цими проблемами. Додаток буде містити різноманітні завдання та вправи, спрямовані на покращення пам'яті, уваги та інших когнітивних функцій. Додаток має враховувати обмеження інклюзивності для людей з вадами. Використання мови програмування Java для платформи Android дозволяє створити зручний та доступний додаток для широкого кола користувачів.

Вище сказане визначає актуальність теми даної кваліфікаційної роботи, отже, розробка мобільного додатку для підтримки процесів відновлення когнітивних функцій є актуальною та корисною ініціативою, яка може покращити якість життя людей з порушеннями когнітивних функцій.

Об'єкт дослідження – процеси відновлення когнітивних функцій людини.

Предмет дослідження – програмне забезпечення для підтримки процесів відновлення когнітивних функцій людини.

Мета роботи – підтримка процесів відновлення когнітивних функцій людини мовою засобами мобільного застосунку-гри для платформи Android.

Для досягнення поставленої мети визначено наступні задачі дипломної роботи:

1. Дослідження та аналіз предметної галузі відновлення когнітивних функцій людини та існуючих застосунків для підтримки процесів відновлення когнітивних функцій людини.
2. Аналіз та розробка вимог до мобільного застосунку-гри для підтримки

процесів відновлення когнітивних функцій людини.

3. Проектування ігрових рівнів з урахуванням завдань підтримки відновлення когнітивних функцій людини.
4. Реалізація геймплейних механік за допомогою платформи Android Studio та мови програмування Java.
5. Тестування застосунку.

Методи дослідження – опрацювання літератури за даною темою, аналіз експлуатаційної документації.

Практичне значення одержаних результатів: розроблено мобільного додатку для підтримки процесів відновлення когнітивних функцій людини мовою програмування Java для платформи Android.

1 АНАЛІЗ КОГНІТИВНИХ ПРОЦЕСІВ ТА ЇХ ВІДНОВЛЕННЯ

1.1 Аналіз видів когнітивних функцій

Когнітивні функції відносяться до первинних розумових здібностей, які дозволяють нам виконувати будь-які завдання. Вони формують те, як людина розуміє світ і сприймає свої дії на оточення. Згадані процеси, які ми здійснюємо у повсякденному житті, є механізмами того, як ми вчимося, запам'ятовуємо, плануємо свої дії, вирішуємо проблеми чи приймаємо рішення. Хоча здорова молода людина навіть не помічає, що використовує ці здібності, вони слабшають з віком або внаслідок травми, коли частинка нашого особливого "я" може бути втрачена. Це може перешкоджати нашому повсякденному функціонуванню, і ми можемо відчувати себе менш корисними та невпевненими. З багатьох наукових праць відомо, що тренування когнітивних навичок може покращити та підтримувати когнітивну компетентність людей похилого віку, як зазначає доктор філософії Круазіл у 2006 році [1]. Ці тренінги навіть перекладають на комп'ютер або мобільні тренінги як один з корисних допоміжних засобів, які в сучасному світі літні люди використовують з легкістю і без відрази. Оскільки вчені так і не дійшли повної згоди щодо виділених груп когнітивних здібностей, існує багато різних підходів до визначення основних функцій. У цій статті ми розділимо когнітивні функції на п'ять основних груп: пам'ять, увага, візуально-просторова обробка, мова, мовленнєві навички та виконавчі функції, описані Клуска у 2009 році [2].

1. Пам'ять - це психічна функція, яка допомагає нам запам'ятовувати всіляку інформацію, досвід і сприйняття. Процес дозволяє нам отримувати, зберігати та відновлювати інформацію з минулого. Це важливо для створення та розвитку нашої особистості. Окрім цього, добре працююча пам'ять впливає на багато інших важливих когнітивних функцій, наприклад, на виконавчі. У минулому вчені вважали, що для різних частин інформації існує лише один тип пам'яті. Зараз переважає думка, що пам'ять поділяється на кілька категорій, що вперше було

описано в 1968 році Аткинсоном і Шиффріном (рис. 1.1) [3]. Модель Аткинсона-Шиффріна також відома як модель кількох магазинів. Сенсорна пам'ять представляє інформацію, яку ми відчуваємо в часовому вікні в кілька сотень мс. Для цього типу пам'яті ідеальним прикладом є видима наполегливість. З точки зору довжини пам'яті для зберігання, ми розрізняємо два інші основні сегменти.



Рис. 1.1 Модель Аткинсона-Шиффріна

Короткочасна пам'ять зберігає інформацію протягом короткого періоду часу (секунд). Наприклад, запам'ятовування усних номерів телефонів, поки вони не будуть записані на папері. Короткочасна пам'ять має обмежену кількість об'єктів, які може вмістити середня людина, і це називають «магічним числом» із діапазоном 7 ± 2 , як зазначив Міллер Г. А. у 1956 р. [4]. Кожна людина може самостійно дізнатися номер, переписавши щойно побачений, наприклад, список покупок на інший папірець. Це число більше, коли людина логічно поєднує будь-які предмети, які потім стають єдиним цілим.

Довготривала пам'ять містить спогади, які зберігаються протягом тривалого періоду часу (хвилини-роки). З короткочасної пам'яті в довготривалу пам'ять переноситься лише деяка значна або повторювана інформація. На процес передачі позитивно впливає те, як його розуміють і пов'язують із раніше набутим досвідом. Довготривала пам'ять не є єдиною сутністю, а складається з кількох різних компонентів [5].

Ця пам'ять вважається імпліцитною пам'яттю, яка зберігає глибоко вкорінені навички, які ми виконуємо автоматично і більше про них не усвідомлюємо. Це стосується вивчення та збереження інформації шляхом повторення та практики, наприклад, знання того, як їздити на велосипеді, грати на фортепіано чи готувати

за рецептом. Він бере знання з процедурної попередньо збереженої пам'яті та формує нові спогади без зусиль. Чудовий приклад, коли у вашій голові застряг текст популярної пісні, яка звучить на радіо.

2. Увага – це здатність вибирати і концентрувати увагу в певному напрямку. Ця функція йде рука об руку з пам'яттю, і вона має вирішальне та важливе значення в нашому повсякденному житті. У той час як величезна кількість подразників постійно заливає нас, увага захищає нас, вибираючи більш важливі в даний момент. Існує п'ять основних типів уваги на основі клінічної моделі, заснованої на роботі Солберга та Матеєра:

- зосереджена увага;
- постійна увага;
- вибіркова увага;
- поперемінна увага;
- розділена увага.

3. Візуально-просторові навички допомагають людині збирати інформацію з навколишнього світу, орієнтуватися в просторі, сприймати об'єкти навколо себе та організовувати цю інформацію у візуальну сцену. З цієї візуальної сцени людина може зрозуміти просторові відносини між об'єктами та візуалізувати об'єкти, яких фізично немає. Базові відомі та популярні ігри Кубик Рубіка і Танграм, які використовуються для тренування нашої зорово-просторової функції, показані на рисунку 1.2.

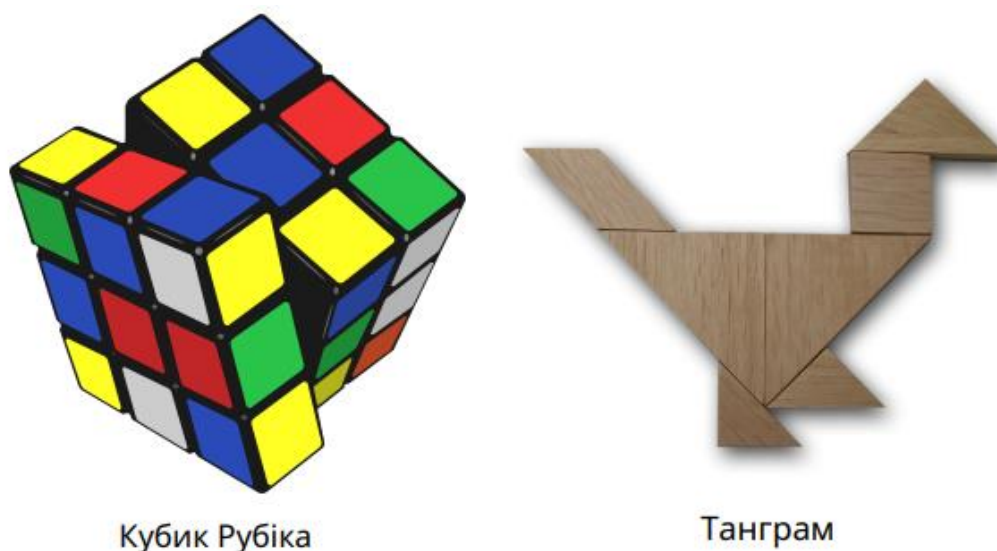


Рис. 1.2 Популярні ігри для тренування зорово-просторових навичок

4. Мова є основним інструментом спілкування і властивістю людини. Ми використовуємо його в читанні, письмі та розмові, щоб розуміти інших і виражати себе. Мовлення — це специфічна мовна навичка для створення звуку завдяки дуже складній координації рухів губ, язика, внутрішньої порожнини рота та голосових зв'язок. За допомогою цієї здатності ми намагаємося створити сенс, будуючи речення, використовуючи відповідні слова разом відповідно до застосовних правил граматики. Ми розрізняємо дві основні категорії під час використання мови: продукування та розуміння. Під час розмови речення утворюються шляхом перетворення думки в слова. Після цього вони об'єднуються, щоб утворити речення. Нарешті створюється звук, який потім передає думку слухачеві. З розумінням відбувається зворотна процедура, за якої звуки, що чуються, починають перетворюватися їх на слова зі значенням, з'єднуючи їх у речення й, нарешті, одержуючи повідомлення про те, що говорить [7].

5. Виконавчі функції - це набір основних навичок, необхідних для життя. Це включає в себе функції планування діяльності, вирішення проблем, виконання більшої кількості дій одночасно, логічне міркування та координацію наших словесних міркувань [8]. Наприклад, планування діяльності допомагає нам визначити план дій або розділити завдання на менші завдання з пріоритетами. На основі загального консенсусу виконавчі функції поділяються на три основні секції:

оновлення, перемикання та гальмування.

З виконавчими функціями та мисленням тісно пов'язане одне з іншим поняттям, яке називається креативністю, яке відноситься до робочої пам'яті. Саме здатність створювати нові думки та ідеї покращує якість нашого життя, створюючи щось нове, оригінальне, несподіване і водночас корисне.

1.2 Огляд методів та засобів відновлення когнітивних функцій

Відновлення когнітивних функцій у людини може відбуватися за допомогою різних методів та засобів, які можна класифікувати за способами впливу на когнітивні процеси. Деякі з них включають:

Комп'ютерні програми та ігри для тренування мозку – ці програми спрямовані на відновлення та підтримку різних когнітивних функцій, таких як пам'ять, увага, мовлення тощо. Вони можуть використовувати різні завдання та вправи, що сприяють покращенню функцій мозку.

Комплексна реабілітація – включає в себе фізичні та реабілітаційні вправи, спрямовані на відновлення когнітивних функцій, разом з психологічною та соціальною підтримкою пацієнта.

Фармакологічні препарати – деякі ліки можуть використовуватися для покращення когнітивних функцій у людей з різними захворюваннями мозку.

Психологічна терапія – психотерапевтичні методи можуть допомагати в управлінні стресом та емоціями, що може позитивно підтримувати когнітивні функції.

Ці методи можуть бути використані окремо або у поєднанні один з одним, залежно від індивідуальних потреб пацієнта та характеру його захворювання. В кваліфікаційній роботі буде розглядатися підтримка процесів відновлення когнітивних функцій у людини за допомогою ігрофікації.

1.3 Аналіз мобільного додатку, як предмету дослідження

Мобільні додатки - це програмні продукти, призначені для роботи на мобільних пристроях, таких як смартфони та планшети. Основні мобільні додатки або попередньо встановлюються на пристрої виробником, або їх можна завантажити з різних інтернет-магазинів мобільних додатків безкоштовно, за щомісячну або одноразову плату. Спочатку мобільні додатки використовувалися для швидкої перевірки електронної пошти, але високий попит призвів до їх використання в інших сферах, таких як мобільні ігри, GPS, зв'язок, перегляд відео, створення розважального контенту, прослуховування музики та користування Інтернетом. Мейнстрімінг. Багато мобільних пристроїв мають веб-браузер (Google Chrome, Safari), поштовий клієнт (Gmail), календар (наприклад, Google Calendar), додатки для купівлі та прослуховування музики (Youtube Music, Spotify, Deezer), мобільний Ряд додатків продається з попередньо встановленими програмами (у випадку з пристроями на базі Android, зазвичай встановлюється ряд додатків Google). Деякі попередньо встановлені програми користувач може видалити з пристрою за допомогою звичайного процесу видалення, щоб звільнити місце в пам'яті для інших (необхідних) програм.

Мобільні додатки, встановлені на пристроях користувачів, є інструментами для прямого контакту з користувачем і ідеально підходять для частого та багаторазового використання. За умови правильної реалізації такі продукти задовольняють конкретні потреби користувачів і підвищують лояльність аудиторії. Переваги мобільних додатків.

Значний рівень інтерактивності, користувачі мають можливість взаємодіяти з додатком різними способами;

Швидко завантажуються та адаптуються до загального користувацького інтерфейсу певної операційної системи, оскільки користувацький досвід може бути збережений.

Вони також можуть використовуватися в автономному режимі без підключення до Інтернету.

Недоліки мобільних додатків:

– Високі вимоги до сумісності(кожного року виходить велика низка різних смартфонів з ОС Android,але з різними надбудовами) та під різні розміри дисплеїв; Вартість розробки, підтримки та обслуговування буде набагато вища, ніж у випадку з мобільним сайтом;

Мобільні додатки класифікуються на:

- Платні – купуєш та отримуєш повний функціонал;
- Безкоштовні – “добрий розробник” дає можливість використовувати свій додаток безкоштовно;
- Умовно безкоштовно – певний час користувач має можливість спробувати функціонал мобільного додатку, а після завершення пробного періоду виріши для себе купувати/платити щомісячно за функціонал представлений розробником

1.4 Порівняльний аналіз додатків для відновлення когнітивних функцій

Існує багато застосунків подібних видів когнітивного навчання. Більшість із них доступні безкоштовно, але значна частина вмісту заблокована. Ці частини можна розблокувати, сплативши преміум-акаунт або щомісячний платіж. Це стосується майже всіх популярних програм із кількістю завантажень понад 1 мільйон. Для решти додатків часто є реклама, яку можна видалити, заплативши меншу суму одиниць євро. Нижче наведено результати аналізу вибраних популярних програм, доступних у Google Play і App Store.

Lumosity – один з найвідоміших додатків у цій категорії, пропонує користувачам ігри, що націлені на тренування різних аспектів когнітивних функцій, включаючи пам'ять, увагу, гнучкість мислення, та швидкість обробки інформації. За статистикою Lumosity є найпопулярнішим додатком для тренування мозку. Він доступний для завантаження з 2007 року, і з тих пір його зареєстрували понад 85 мільйонів користувачів із 182 країн. Додаток доступний для безкоштовного завантаження на настільні комп'ютери, Android та iOS (рис 1.3), тому його завантажено понад 35 мільйонів на мобільних пристроях Android та iOS.

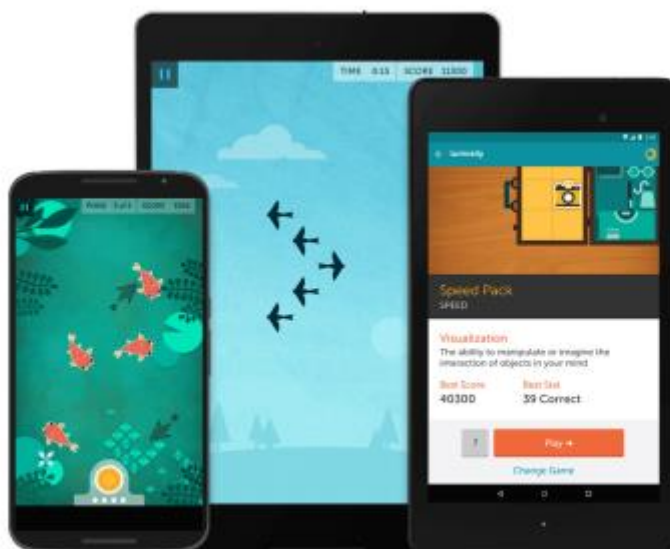


Рис. 1.3 Екранні форми мобільного додатку Lumosity

Щодня три різні міні-гри пропонуються для безкоштовного навчання. Кожна з доступних ігор перевіряє іншу частину когнітивної функції, тобто дві гри в один день не повторюють тренування тієї самої когнітивної функції. Користувач може отримати доступ до всіх ігор (50+), а також переглянути статистику прогресу навчання, придбавши преміум-акаунт. Преміум-акаунт оплачується щомісяця, а ціна коливається в межах 5,6-13,6 євро.

CogniFit – це медична компанія, заснована в 1999 році, зосереджена на оцінці та покращенні когнітивного здоров'я. Точні дані про кількість користувачів або статистичні дані відсутні, але кількість користувачів, які використовують запропоновані програми, обчислюється мільйонами. Компанія також пропонує оцінки когнітивних здібностей або навчальні програми для цільових груп. [8] Як і Lumosity, цей мобільний додаток пропонує три різні гри щодня. У разі гри в декілька ігор або повторів у певний день, це вимагає щомісячної оплати за преміум-акаунт, вартість якого становить від 4,2 до 22,99 євро. Також є можливість придбати оцінку цільової програми за 19,95 євро (наприклад, тест на дислексію, тест на хворобу Паркінсона).

Додаток Peak працює тільки на пристроях iOS і Android. Детальна статистика чи інформація про програму не є загальнодоступною, тому можна порівняти лише

кількість завантажень із Google Play понад 10 мільйонів. Візуалізацію вмісту програми можна побачити на рисунку 1.4.

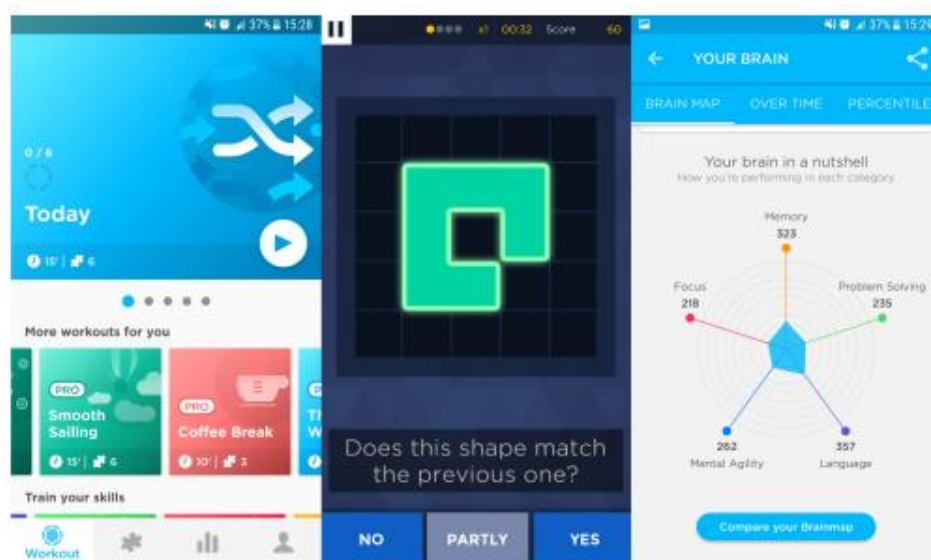


Рис 1.4 Екранні форми додатку Peak

На відміну від інших програм, він пропонує безкоштовні 4-6 міні-гри рівно через 24 години після останньої гри минулого дня. Як і в попередніх випадках, придбання преміум облікового запису дає користувачам змогу розблокувати різні цілі, індивідуальні тренування, розширену статистику, доступ до всіх ігор і можливість порівнювати з іншими користувачами. Ціна за місяць коливається від 1-2 євро. Однак існує можливість одноразової покупки на все життя за 22,99 євро. На даний момент існує лише один поглиблений спеціалізований тренінг, присвячений перевірці пам'яті.

Програма Elevate призначена в основному для вдосконалення комунікативних і аналітичних навичок. Він був запущений у 2014 році, і з того часу його завантажили понад 25 мільйонів користувачів у Google Play та App Store. Доступно лише на мобільних пристроях. [8] Більшість міні-ігор засновані на читанні та письмі лише англійською мовою. Три безкоштовні ігри доступні щодня, щоб розблокувати всі 35+ ігор. Варіанти налаштування, докладні звіти про навчання та навчальні матеріали для покращення когнітивних здібностей вимагаються, щоб заплатити 42,55 євро на рік за обліковий запис "PRO".

Понад 12 мільйонів користувачів використовують програми NeuroNation, що робить її одним із провідних постачальників додатків для тренування мозку в Європі. Заснована в 2011 році, коли вона представила онлайн-курси безпосередньо на своєму веб-сайті, а потім у 2014 році запустила мобільний додаток App Store і Google Play. [9]

NeuroNation використав інший підхід, а саме можливість грати у вісім доступних безкоштовних ігор. Однак вони заохочують користувача зіграти в «сеанс» із 4 рекомендованих ігор, зосереджених на одному когнітивному аспекті. Індивідуальною відмінністю є оцінка статистики користувачів, яка відбувається лише після восьми сесій. Порівняно з іншими іграми, вони дозволяють користувачам вибирати ігри для свого навчання, які найбільше підходять, і не залишають їх випадковими іграми на один день. Звичайно, пропонується преміум-акаунт, який коштує 3-18,18 євро за один місяць. З преміум-акаунтом розблоковано всі ігри (27), загалом 250 рівнів і персоналізований досвід навчання на основі характеристик користувача. Їхня зосередженість на більш чистому графічному дизайні зображена на рисунку 1.5.



Рис. 1.5 Екранні форми додаток NeuroNation

Основні деталі описаних вище програм зведені в таблицю 1.1. Сортуювання додатків за популярністю здійснюється шляхом вибору відповідної колонки, яка є кількістю завантажень. Згадана таблиця є уже відсортованою за цим стовпцем, і тому можна зрозуміти, що з цього порядку менш популярні програми, швидше за все, створять просунуті методи для залучення більшої кількості людей.

Таблиця 1.1

Сортуювання додатків за популярністю

	Завантаження	Доступно	Примітки
Lumosity	35 млн.+	Інтернет, Android, iOS	—
Elevate	25 млн.+	Android, iOS	—
NeuroNation	12 млн.+	Інтернет, Android, iOS	—
Peak	10 млн.+	Android, iOS	Розширений спеціалізований тренінг з перевірки пам'яті (7,28 євро)
CogniFit	~1 млн.+	Інтернет, Android, iOS	Цільове спеціалізоване навчання для тестування різних дисфункцій (19,95 євро за кожен)

Таблиця 1.2 зосереджена на аспектах ціноутворення та функцій. Сортуювання за кількістю завантажень для цієї таблиці залишається незмінним. Тут можна помітити, що кожен окремий елемент має певну преміальну версію, яка відкриває більше переваг. Вибираючи з найнижчого діапазону преміум-ціноутворення, де це можливо, можна побачити, що Lumosity є найдорожчим із 5,6 євро, а Peak є другим найдешевшим із 1 євро, але обидва забезпечують однакові переваги преміум-класу. Можливо, можна сказати, що різниця в мінімальних платежах є більшою залежно від популярності програми.

Таблиця 1.2

Результати аналізу ціноутворення та функцій застосунків

	Безкоштовні функції	Преміум (ціна / місяць) о	Преміум переваги
Lumosity	Тренування по 3 гри щодня	5,6 - 13,6 євро	Розблокувати всі ігри, детальну статистику, розблокувати досягнення, порівняти з іншими користувачами
Elevate	Тренування по 3 гри щодня	~3,51 євро	Розблокуйте всі ігри, параметри налаштування навчання, детальну статистику та новини, навчальні матеріали
NeuroNation	Необмежене навчання з 8 іграми	3 - 18,18 євро	Розблокуйте всі ігри, виберіть персоналізоване тренування
Peak	Щоденні тренування по 4-6 ігор	1 - 2 євро (22,99 євро назавжди)	Розблокувати всі ігри, детальну статистику, розблокувати досягнення, порівняти з іншими користувачами
CogniFit	Тренування по 3 гри щодня	4,2 - 22,99 євро	Розблокувати всі ігри, розблокувати статистику

Основні відомості про характеристику вищезгаданих додатків наведено в таблиці 1.3. Ранжування додатків за популярністю здійснюється шляхом вибору відповідного стовпчика (кількість завантажень). Оскільки наведена вище таблиця вже відсортована за цим стовпчиком, з цього рейтингу можна зробити висновок, що менш популярні додатки, швидше за все, розробили складніші методи для залучення більшої кількості користувачів.

Таблиця 1.3

Зведені результати аналізу характеристик додатків
для відновлення когнітивних функцій

Показник	Lumosity	Elevate	NeuroNation	Peak	CogniFit	Rehabilitation
Робота без підключення до мережі	Обмежено	Обмежено	-	-	Обмежено	+
Складність геймплею	Середня	Середня	Середня	Середня	Середня	Низька
Дизайн враховує обмеження інклюзивності для людей з вадами зору, слуху	+	-	-	+	-	+
Можливість відправити результати сесії на e-mail	+	+	-	-	+	+
Статистика для наглядача	-	-	-	-	-	+
Підтримка локалізації (українська мова)	-	-	+	-	-	+

Використовуючи представлені додатки для тренування мозку у режимі офлайн, деякі функції можуть бути обмежені, але основні функції залишаються доступними. Наприклад, ви зазвичай зможете виконувати тренування, які ви вже завантажили раніше, переглядати свій прогрес та отримувати результати. Однак для завантаження нового контенту, оновлень програми чи синхронізації даних буде потрібне підключення до Інтернету.

Щодо української локалізації, деякі додатки можуть не мати повної підтримки української мови, що ускладнює їх використання для користувачів із України. Проте, це може бути вирішено шляхом додавання української локалізації, яка включатиме переклад усіх інтерфейсів, повідомлень та інших елементів додатку на українську мову. Це робить додаток більш доступним і зручним для

українських користувачів, зокрема для літніх людей або тих, хто не має досконалого володіння іноземними мовами. Додавання української локалізації може стати одним з пріоритетів розробників цих додатків для полегшення їх використання українськими користувачами.

В більшості представлених додатків дизайн не враховує обмеження інклюзивності для людей з вадами зору, слуху.

Висновки до першого розділу

Проведено короткий огляд основних когнітивних функцій людини, таких як пам'ять, увага, мовлення, мислення та інші. Було визначено, що ці функції відіграють важливу роль у повсякденному житті людини та їх відновлення є актуальною проблемою сучасної медицини та психології.

Розглянуто різні методи та засоби відновлення когнітивних функцій у людини. Було виявлено, що такі методи, як комп'ютерні програми, фізичні вправи, фармакологічні препарати та психологічна терапія, можуть бути ефективними у відновленні та підтримці когнітивних функцій. Кожен з цих методів має свої переваги та обмеження, тому вибір конкретного методу повинен залежати від індивідуальних потреб кожного пацієнта. Зазначено що в роботі буде розглядатися тільки метод когнітивного відновлення.

Проведений порівняльний аналіз різних когнітивних додатків для мобільних пристроїв. Було виявлено, що кожен з додатків має свої переваги та недоліки, і вибір конкретного додатку повинен залежати від потреб користувача та його індивідуальних можливостей.

Аналіз існуючих програм когнітивного навчання довів, що найпопулярніші з них мають дуже комерційну основу. Це підтверджує, що ці програми здебільшого створюють команди, які роблять це заради фінансової вигоди, хоча зрозуміло, що команди, які створюють ці високоякісні програми, не будуть робити це безкоштовно. Одночасно жоден із додатків не підтримує українську мову, що може відштовхнути використання додатків для когнітивного навчання.

2 ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДТРИМКИ ПРОЦЕСІВ ВІДНОВЛЕННЯ КОГНІТИВНИХ ФУНКЦІЙ

2.1. Функціональні та нефункціональні вимоги до застосунку. Варіанти використання

Аналіз особливостей процесу відновлення когнітивних функцій та функціональних і нефункціональних можливостей мобільних додатків-аналогів сформовано нижченаведені вимоги до застосунку.

Функціональні вимоги:

1. Запуск та завершення сесії тренування.
2. Застосування вправ для стимулювання різних когнітивних функцій, таких як пам'ять, увага, розуміння та швидкість обробки інформації.
3. Вибір завдання для тренування.
4. Відстеження прогресу тренувань.
5. Налаштування параметрів тренування.
6. Налаштування статистики для наглядача.
7. Можливість відправити результати сесії на e-mail.

Нефункціональні вимоги:

1. Сумісність з операційною системою на базі Android.
2. Робота додатку без доступу в Інтернет.
3. Підтримка налаштувань для персоналізації досвіду користувача.
4. Дизайн застосунку повинен враховувати обмеження інклюзивності для людей з вадами зору, слуху.

Для моделювання вимог використано діаграму варіантів використання – вона є корисним інструментом в процесі проектування для розуміння та візуалізації того, які кроки виконуються під час роботи користувача з застосунком (рис.2.1).

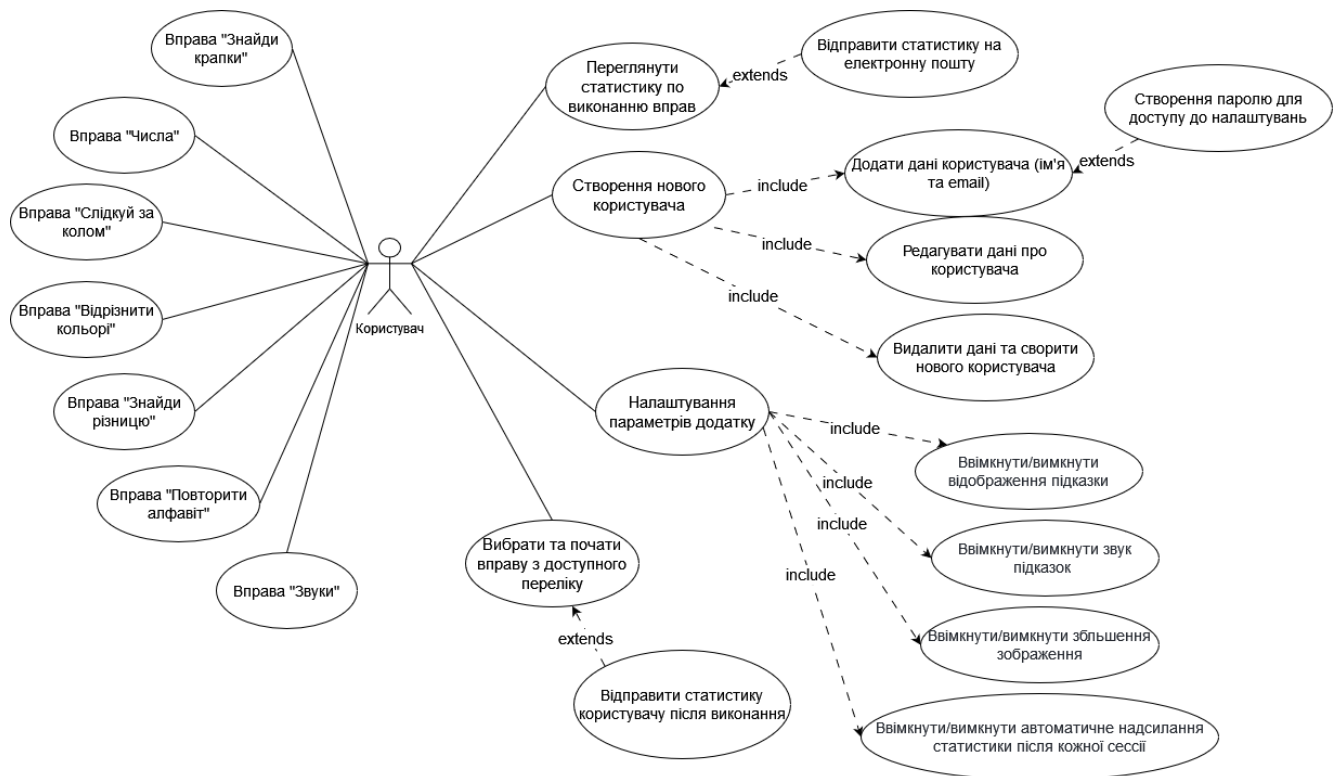


Рис. 2.1 Діаграма варіантів використання застосунку

Прецеденти, які доступні для використання на головному екрані включають наступні:

- вибір вправи з списку:
 - «Знайди крапки»;
 - «Числа»;
 - «Слідкуй за колом»;
 - «Відрізнити кольори»;
 - «Знайди різницю»;
 - «Повторити алфавіт»;
 - «Звуки»;
- перехід до перегляду та відправки статистики;
- вихід та надсилення статистики.

Процес налаштування включає в себе:

- ввімкнути/вимкнути відображення підказки;
- ввімкнути/вимкнути звук підказок;
- ввімкнути/вимкнути збільшення зображення;

- ввімкнути/вимкнути автоматичне надсилення статистики.

Процес створення нового користувача включає прецеденти:

- додати дані користувача (ім'я) та email;
- створення паролю для доступу до налаштувань;
- редагувати дані.

Алгоритм тренувальної сесії користувача, який відображає порядок виконання описаних прецедентів, наведено на рис.2.2.

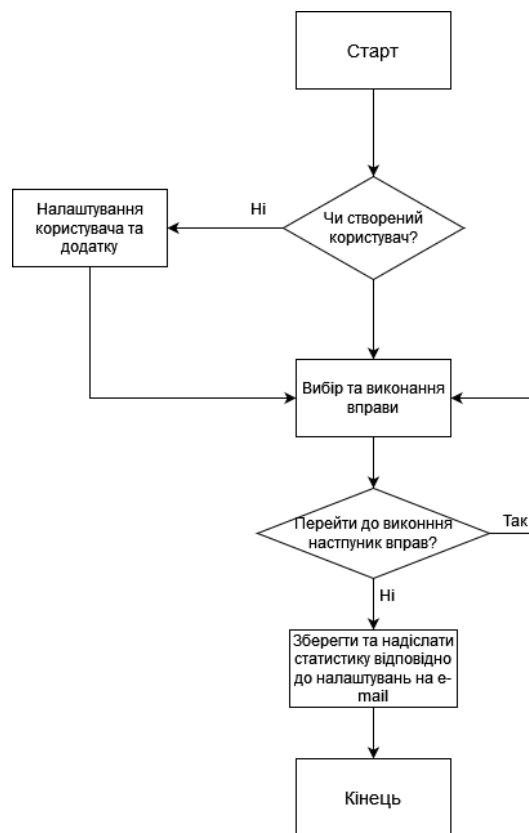


Рис. 2.2 Алгоритм тренувальної сесії користувача

2.2 Проектування інтерфейсу користувача

Прототипування - це процес створення макета або моделі майбутнього мобільного додатка з метою перевірки його структури, функціональності та загальної концепції програми. Макет дозволяє уникнути непорозумінь між різними

учасниками проекту, такими як керівник, дизайнер, програміст та клієнт, і відсіяти помилки ще на початкових етапах розробки. Прототипування також допомагає структурувати думки та уникнути виконання зайвої роботи.

Можливість тестування майбутнього додатка за допомогою фокус-групи дозволяє отримати цінну інформацію від потенційних користувачів.

Для даного мобільного додатка було використано концепцію "швидкого прототипування", що дозволило швидко намалювати основні вікна майбутнього програмного продукту. Макет було розроблено за допомогою стандартних засобів Android Studio [15].

Під час його розробки основними критеріями були наступні пункти:

- елементи мають бути виконані в єдиному стилі;
- елементи не повинні заважати відтворенню один одному;
- продумати навігацію, яка буде інтуїтивно-зрозумілою та враховувати обмеження інклюзивності користувача.

Створені макети дизайну представлені на рис. 2.4 – 2.8.

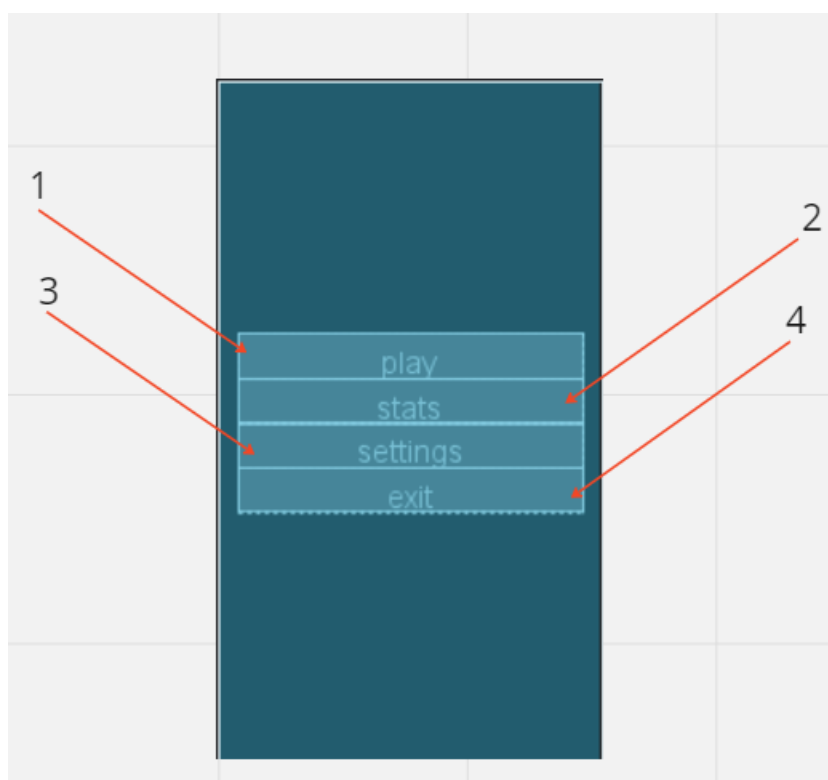


Рис. 2.3 Макет головного вікна додатку

Стрілками та позначеннями до них показані основні елементи екранів мобільного додатку, які відносяться до категорії типових в даному класі додатків [15].

Роз'яснення до елементів головного екрану наведено нижче:

1. Кнопка яка відповідає за початок роботи над вибраними вправами.
2. Кнопка яка відповідає за статистику, вправи які були пройдені конкретним або різними користувачами.
3. Кнопка яка відповідає за налаштування вибрані користувачем (зокрема персональна інформація)
4. Кнопка виходу з застосунку.

На рисунку 2.4 показано макет activity виконання одної з вибраних користувачем вправ.

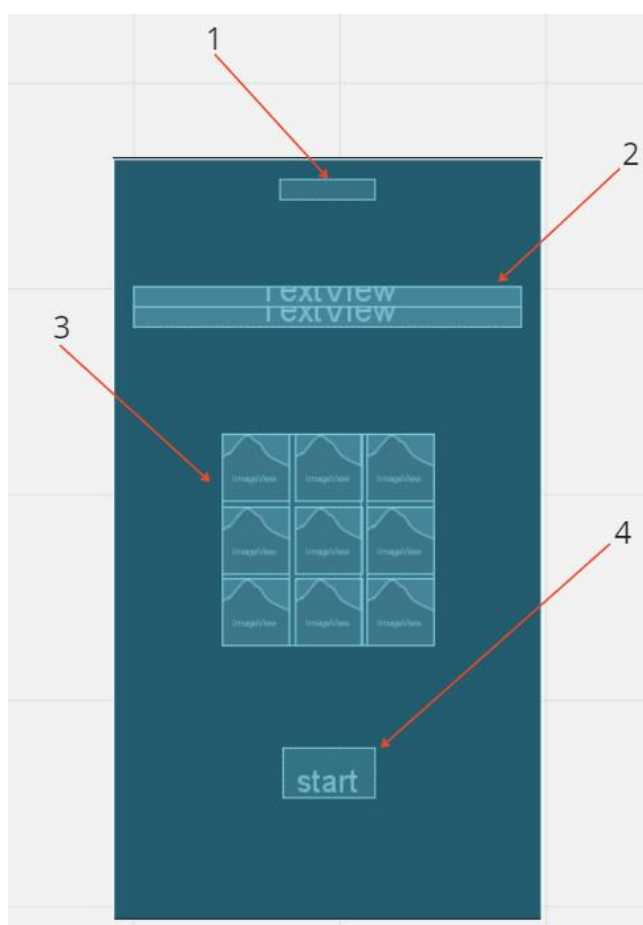


Рис. 2.4 Макет вікна виконання вправи

Використані на рис.2.4 позначення:

1. Елемент який відображає назву вправи.
2. Текстовий елемент який відповідає за покрокове виконання дій для вдалого завершення завдання.
3. Елементи які відображають інтерфейс об'єктів які змінюються (такий інтерфейс притаманний не кожній вправі).
4. Кнопка запуску вправи [15].

На рис. 2.5 представлено, як результати користувача будуть відображатися в статистиці, а саме:

- Скільки часу було витрачено на виконання вправи;
- Рівень складності;
- Час початку виконання.

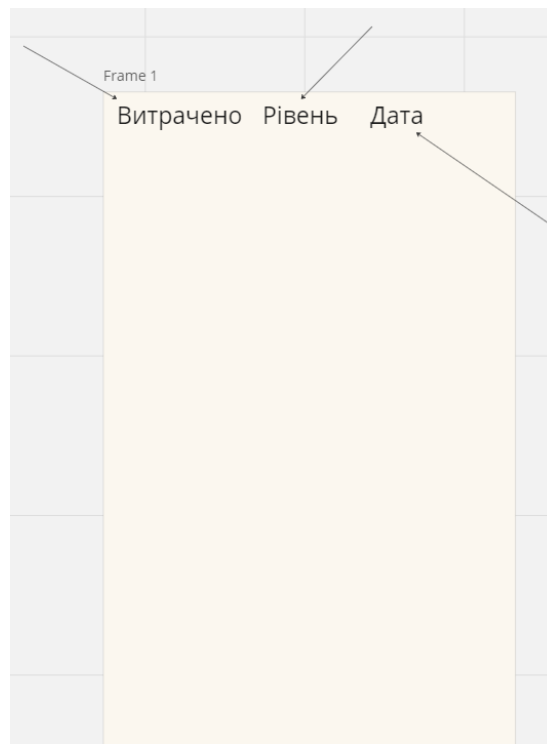


Рис. 2.5 Макет вікна статистики

Кнопка налаштувань для того щоб користувач мав змогу вказати свої персональні дані для правильного відображення в статистиці та змінити налаштування (підказок, звуку та збільшення вигляду вправ) (рис.2.6) [15].

1. Ім'я користувача.
2. Встановлення імені вручну. (Встановити)
3. Поточний E-mail.
4. Встановити E-mail. (Встановити)
5. Поля вибору (Підказки ввімкнено, Звук підказок ввімкнено, Збільшений вигляд).

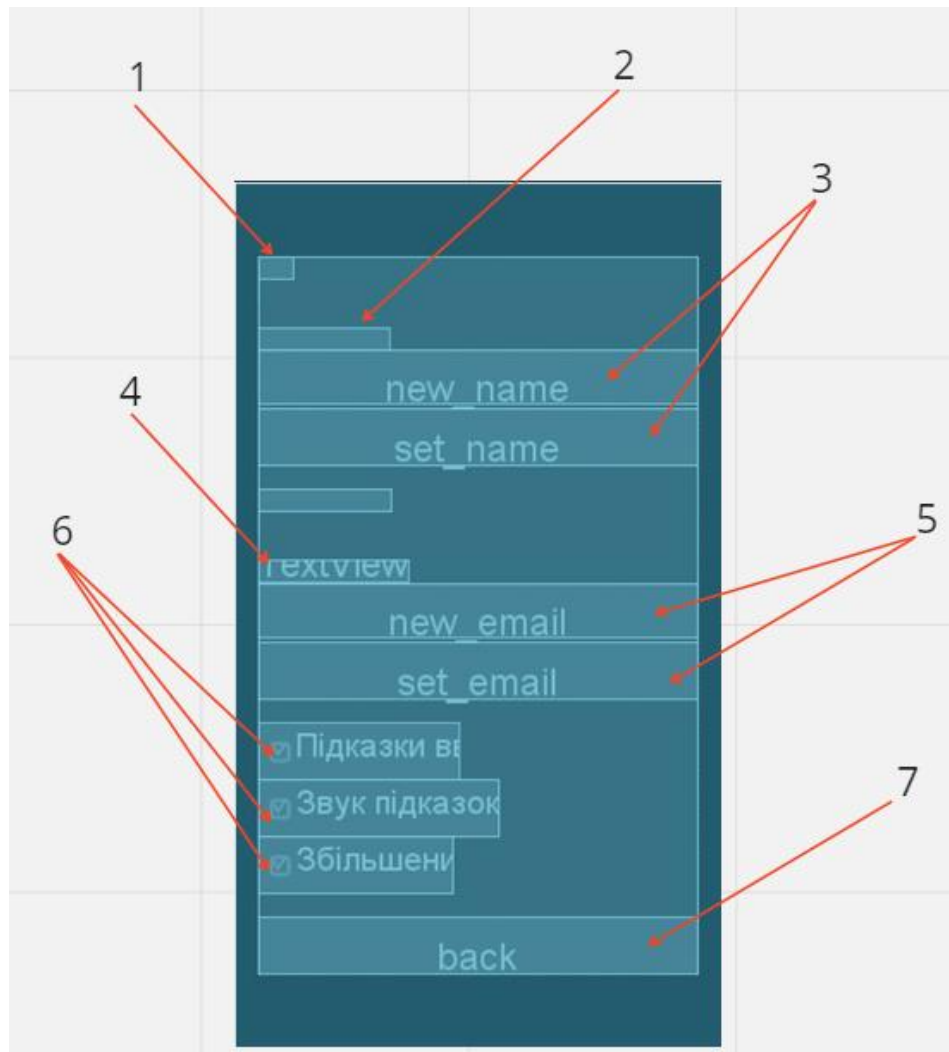


Рис. 2.6 Макет вікна налаштування

Макет екрану з полями вибору для виконання вправи (кожна наступна з поглибленою складністю) наведено на рис.2.7.

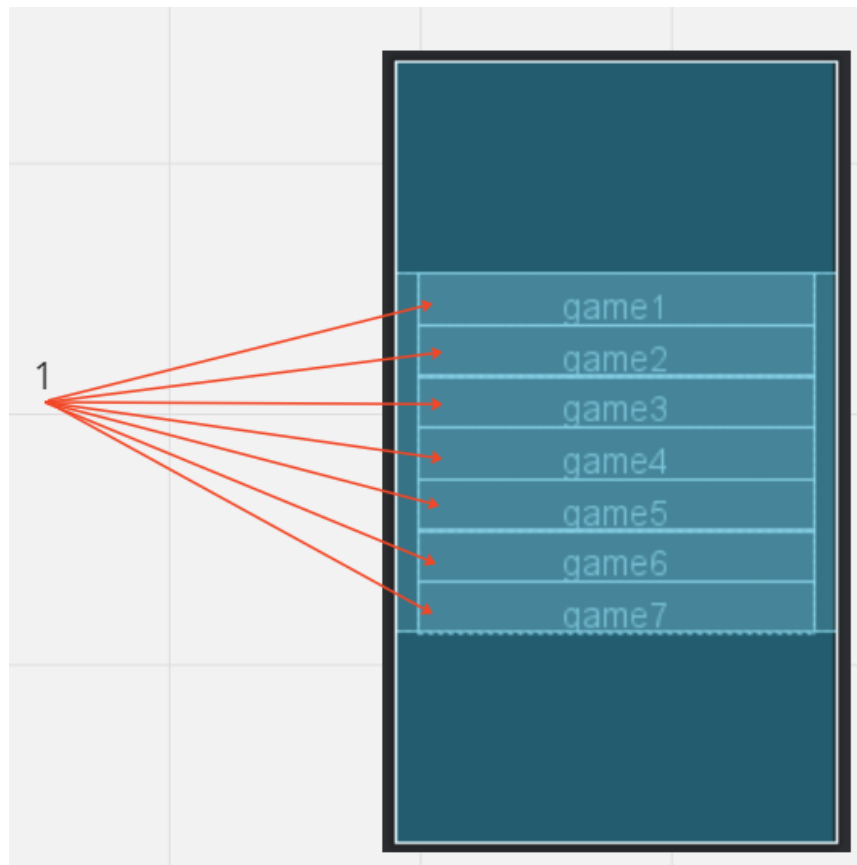


Рис. 2.7 Макет вікна вибору вправи

Для мобільного додатку «Відновлення когнітивних функцій» було створено логотип [16].

Даний логотип відповідає наступним вимогам, які передбачені Google Play:

- підсумковий розмір: 512 x 512 пікселів;
- формат: 32-розрядний PNG;
- кольорова палітра: RGB;
- максимальний розмір файлу: 1024 КБ;
- форма: повний квадрат.

На рис. 2.8 показано контури об'єкта, який лежить всередині квадрата.

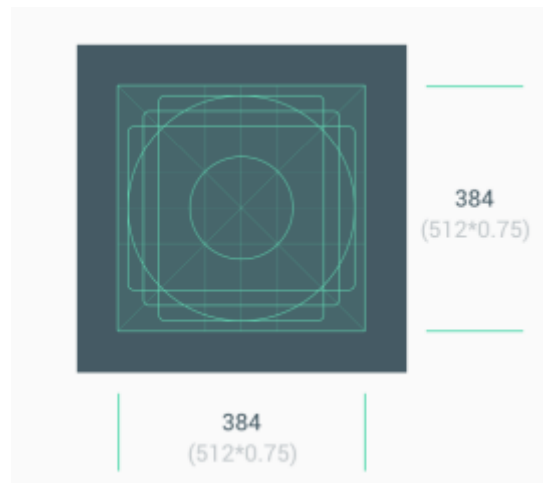


Рис. 2.8 Контури значка продукту

За основу логотипу на даний момент було вибрано шаблонний варіант Android Studio з мінімалістичним стилем за трьома кольорами: Сірий, білий та зелений (рис.2.9).

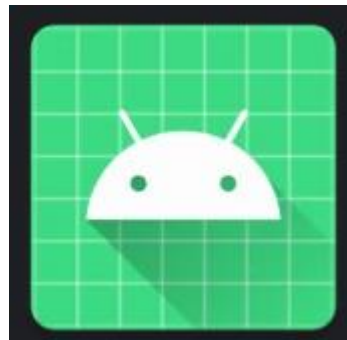


Рис. 2.9 Логотип мобільного додатку

Висновки до другого розділу

Відповідно до аналізу аналогів, дослідження потреб цільової аудиторії були розроблені функціональні та нефункціональні вимоги до застосунку.

Створено діаграму використання застосунку та блок схему сесії користувача, що описує взаємодію застосунку з користувачем.

З урахування потреб цільової аудиторії виконано прототипування та створено макети інтерфейсу, який допоможе користувачам легко користуватися додатком та досягати їхніх цілей у відновленні когнітивних функцій.

3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДТРИМКИ ПРОЦЕСІВ ВІДНОВЛЕННЯ КОГНІТИВНИХ ФУНКЦІЙ

3.1 Обґрунтування вибору платформи та технологій розробки застосунку

В даній роботі розробляється додаток під платформу Android. В якості мови розробки вибрано Java. Це одна з найбільш поширених мов програмування, яка дозволяє розробляти різні види додатків на різних платформах Android, MacOS, Windows, Linux, а також веб-служби, бази даних, клієнт-серверне програмне забезпечення та інше. Java є сучасною мовою, яка підтримує асинхронне програмування, що дозволяє відокремити потік обробки користувацького інтерфейсу від задач бізнес-логіки додатку.

Автоматичне управління пам'яттю засобами Garbage collection дозволяє гнучке використання складних структур даних. Мультипарадигмальність мови забезпечує підтримку об'єктно-орієнтованого програмування, функціональне програмування, в тому числі, лямбда-вирази, яка дозволяє використовуючи синтаксис запиту виконувати операції фільтрації, упорядкування та групування джерел даних з мінімумом коду.

Java на платформі Android є дуже хорошим варіантом вибору з кількох причин. По-перше, вона має широку підтримку від офіційної документації Android, що робить її основним вибором для багатьох розробників. Крім того, Java є високорівневою мовою програмування з простим синтаксисом, що полегшує розробку додатків для Android [17, 18].

Також важливою перевагою є велика кількість доступних бібліотек та інструментів для розробки на Java, які спрощують роботу розробників та роблять процес більш ефективним. Прозора сумісність з нативним кодом надає можливості написати обгортку для будь-якої нативної бібліотеки і використовувати її можливості. Java є кросплатформною мовою і крім мобільних пристроїв, додатки,

написані мовою Java працюють на багатьох вбудованих, десктопних і серверних платформах.

Оскільки програма була спроектована під мобільну розробку, а саме на пристрої на базі Android, враховуючи вибір мови, вирішено було розробляти на платформі Android Studio. Android Studio - це інтегроване середовище розробки (IDE) для створення мобільних додатків для платформи Android. Воно базується на відкритому програмному забезпеченні та надає розробникам широкий спектр інструментів для створення високоякісних додатків. Android Studio використовує мову програмування Java або Kotlin для розробки додатків [18].

Платформа Android Studio надає широкий набір інструментів, таких як редактор коду з підсвічуванням синтаксису, інструменти для розладки, візуальний редактор інтерфейсу користувача та інші. Крім того, Android Studio має вбудовану підтримку для різних пристроїв Android, що дозволяє розробникам ефективно тестувати свої додатки на різних пристроях та версіях операційної системи.

3.2 Розробка архітектури застосунку

Файлова структура застосунку Rehabilitation представлена на рис. 3.1 та 3.2. У вкладені data ми можемо побачити такі елементи, як network, який містить в собі клас GMailSender який використовується для надсилання електронних листів через Gmail SMTP-сервер. Він містить метод send, який приймає параметри отримувача, теми, повідомлення, колбек-функцію для виконання після відправлення та додаткові вкладення (якщо необхідно). StatisticsSender відправляє електронний лист із статистикою користувача. Він спочатку отримує дані з бази даних, витягуючи дані вправ з об'єкту StatisticsDao, та читає електронну пошту користувача із налаштувань за допомогою Prefs.

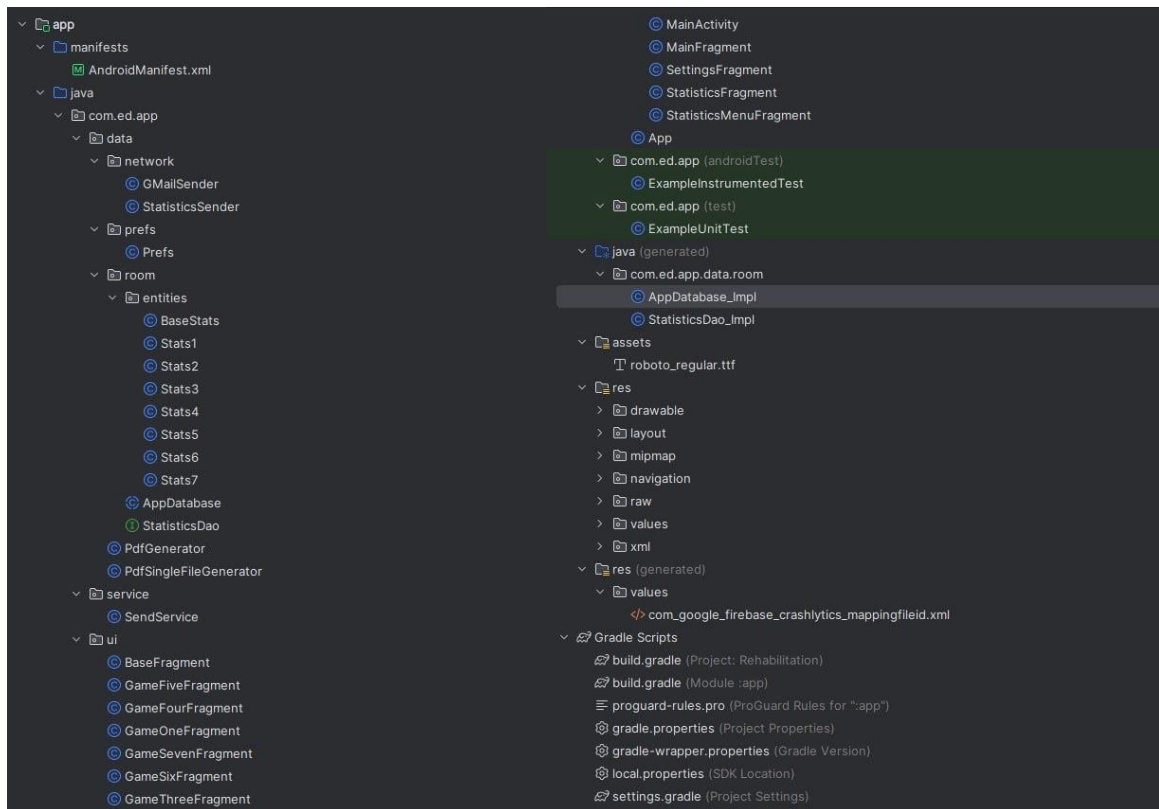


Рис. 3.1 Файлова структура розробленого застосунку

3.3 Опис реалізованих класів

Для роботи з статистикою реалізовано клас `GMailSender` який використовується для надсилання електронних листів через Gmail SMTP-сервер. Він містить метод `send`, який приймає параметри отримувача, теми, повідомлення, колбек-функцію для виконання після відправлення та додаткові вкладення (якщо необхідно). Використовуючи конфігураційні властивості, він створює сеанс `JavaMail`, ініціалізує повідомлення, додає вкладення (якщо вони є) та відправляє лист. Якщо відправлення не вдалося, викидається виняток з відповідним повідомленням про помилку, яке також виводиться через `Toast` на головному потоці `UI`.

Клас `StatisticsSender` відправляє електронний лист із статистикою користувача. Він спочатку отримує дані з бази даних, витягуючи дані вправ з об'єкту `StatisticsDao`, та читає електронну пошту користувача із налаштувань за допомогою `Prefs`. Якщо адреса електронної пошти відсутня, виводиться

повідомлення Toast із проханням встановити пошту у налаштуваннях. Інакше, викликається метод `generatePdf` для генерації PDF-файлу зі статистикою користувача, а потім відправляється електронний лист із згенерованим файлом вкладенням.

Клас `Prefs` використовується для роботи з налаштуваннями додатка. Він містить ряд статичних констант для ключів, що представляють різні налаштування, такі як допомога, звук допомоги, розширений перегляд, адреса електронної пошти тощо. Конструктор отримує контекст і ініціалізує об'єкт `SharedPreferences` для доступу до налаштувань за допомогою іменованої відділеної області `"prefs"`. Клас також має методи для запису та читання різних типів даних з/у налаштування, таких як `boolean`, `String`, `int` та `long`, які зберігаються в об'єкті `SharedPreferences`.

`BaseStats` який є базовим для об'єктів, що будуть зберігатися в базі даних за допомогою бібліотеки `Room Persistence` в `Android`. Поле `id` є унікальним ідентифікатором із автогенерацією значень (`autoGenerate = true`), що дозволяє автоматично створювати унікальний ідентифікатор для кожного об'єкта. Поле `usedTime` вказує на час використання, `level` - рівень, а `dateTime` - дата та час, коли були збережені ці статистичні дані.

Розглядається клас `AppDatabase`, який представляє базу даних для додатка. Він використовує анотацію `@Database` для вказівки, які сутності включати до бази даних та версії бази даних. Клас також розширює `RoomDatabase`, що є базовим класом для `Room Persistence Library` в `Android`. Анотація `@Database` містить перелік сутностей (`entities`), які будуть збережені в базі даних. У цьому випадку зазначені сутності `Stats1`, `Stats2`, `Stats3`, `Stats4`, `Stats5`, `Stats6` та `Stats7`, які, очевидно, відображають статистичні дані для різних категорій. Крім того, вказана версія бази даних. Також у класі визначений абстрактний метод `getDao()`, який повертає об'єкт типу `StatisticsDao`. Цей метод дозволяє отримати доступ до об'єкту DAO (`Data Access Object`), який використовується для взаємодії з базою даних.

`SendService` визначає служба, яка відправляє статистику через мережу. Після старту служби створюється канал сповіщень для відображення повідомлень. Створюється сповіщення про процес відправки статистики, і служба переходить у

фоновий режим. Після цього відбувається перевірка наявності електронної пошти користувача для відправки статистики. Якщо адреса відсутня, відображається сповіщення з пропозицією встановити пошту. Якщо адреса присутня, викликається метод `send()`, який ініціює відправку статистики. Після відправлення відображається сповіщення про успішну відправку, і служба завершує роботу.

Генерація реалізації бази даних `AppDatabase_Impl`, яка використовує `Room`. Вона містить методи для створення та оновлення таблиць, перевірки схеми бази даних, очищення таблиць, а також отримання об'єкту доступу до даних (DAO). Метод `createOpenHelper()` створює об'єкт `SupportSQLiteOpenHelper` для доступу до бази даних, а метод `createInvalidationTracker()` створює відслідковувач невалідності для таблиць бази даних. Метод `getDao()` повертає об'єкт `StatisticsDao`, який використовується для взаємодії з базою даних.

Розглянемо класи `GMailSender` та `StatisticsSender` з їх методами більш детально.

Клас GMailSender:

1. Опис класу: Клас `GMailSender` відповідає за надсилання електронних листів через Gmail SMTP-сервер.

2. Метод `send`:

- Приймає параметри: адреса отримувача (to), тема листа (subject), текст повідомлення (body), колбек-функція для виконання після відправлення (callback) та, при потребі, додаткові вкладення у вигляді файлів (attachments).

- У структурі методу виконується логіка надсилання листа через Gmail SMTP-сервер. Він використовує дані аутентифікації користувача Gmail, такі як адреса електронної пошти та пароль, для відправлення листа.

3. Взаємодія з іншими класами:

- Цей клас може використовуватись іншими компонентами програми для надсилання електронних повідомлень. Наприклад, він використовується в класі `StatisticsSender` для відправлення статистики користувача.

```

1 package com.ed.app.data.network;
2
3 > import ...
4 usages
5
6 public class GMailSender extends javax.mail.Authenticator {
7     2 usages
8     public void send(
9         String recipient,
10        String subject,
11        String msg,
12        Runnable finishCallback,
13        File... attachments
14    ) {
15
16        final String username = "mykyta.naumov03@gmail.com";
17        final String password = "omms bhqk nbun yyae";
18
19        Properties props = new Properties();
20        props.put("mail.smtp.auth", "true");
21        props.put("mail.smtp.starttls.enable", "true");
22        props.put("mail.smtp.host", "smtp.gmail.com");
23        props.put("mail.smtp.port", "587");
24
25        Session session = Session.getInstance(props,
26            getPasswordAuthentication() -> {
27                return new PasswordAuthentication(username, password);
28            });
29
30        try {
31            Message message = new MimeMessage(session);
32            message.setRecipients(Message.RecipientType.TO,
33                InternetAddress.parse(recipient));
34            message.setSubject(subject);
35            message.setText(msg);
36
37            if (attachments != null) {
38                Multipart multipart = new MimeMultipart();

```

Рис. 3.2 Структура класу GMailSender

Клас StatisticsSender:

- Опис класу: Цей клас відповідає за відправлення електронних листів із статистикою користувача.
 - Метод send:
 - Приймає параметри: контекст програми (context) та колбек-функцію для виконання після успішного відправлення (finishCallback).
 - У структурі методу виконується наступна логіка:
 - Отримання електронної пошти користувача з налаштувань.
 - Отримання даних статистики користувача з бази даних.
 - Виклик методу send з класу GMailSender для надсилання електронного листа із статистикою.
 - Виконання колбек-функції після відправлення.
3. Взаємодія з іншими класами:

– Цей клас взаємодіє з класом `GMailSender` для відправлення електронних повідомлень. Також він використовує клас `Prefs` для отримання електронної пошти користувача з налаштувань.

– Класи утворюють функціонал для надсилання електронних листів зі статистикою користувача через Gmail SMTP-сервер.

```

1 package com.ed.app.data.network;
2
3 > import
14
6 usages
15 public class StatisticsSender {
16     no usages
17     private Prefs prefs;
18
19     3 usages
20     public void send(
21         Context context,
22         Runnable finishCallback
23     ) {
24         StatisticsDao dao = App.instance.db.getDao();
25         Prefs prefs = new Prefs(context);
26         String email = prefs.readString(Prefs.EMAIL, defaultValue: null);
27         String userName = prefs.readString(Prefs.NAME, defaultValue: "не встановлено");
28         if (email == null) {
29             Toast.makeText(
30                 context,
31                 text: "Вкажіть пошту у налаштуваннях",
32                 Toast.LENGTH_SHORT
33             ).show();
34         } else {
35             File file1 = PdfGenerator.generatePdf(
36                 context,
37                 "Вправа 1",
38                 "Вправа 1",
39                 "Ім'я: " + userName,
40                 dao.get1()
41             );
42             File file2 = PdfGenerator.generatePdf(
43                 context,
44                 "Вправа 2",
45                 "Вправа 2",
46                 "Ім'я: " + userName
47             );
48         }
49     }
50 }

```

Рис. 3.3 Структура класу `StatisticsSender`

Клас `Prefs`:

1. Опис класу:

- Клас `Prefs` відповідає за роботу з налаштуваннями за допомогою `SharedPreferences` в Android.

- Він має методи для збереження і отримання різних типів даних у налаштуваннях.

2. Поля класу:

- `prefs`: Об'єкт `SharedPreferences`, що використовується для доступу до налаштувань.

3. Конструктор:

- `public Prefs(Context context)`: Конструктор, який ініціалізує об'єкт `SharedPreferences` за допомогою контексту.

4. Методи для запису значень:

- `write(String key, boolean value)`: Записує значення типу `boolean` у налаштування за ключем `key`.

- `write(String key, String value)`: Записує значення типу `String` у налаштування за ключем `key`.

- `write(String key, int value)`: Записує значення типу `int` у налаштування за ключем `key`.

- `write(String key, long value)`: Записує значення типу `long` у налаштування за ключем `key`.

5. Методи для читання значень:

- `readBoolean(String key, boolean defaultValue)`: Повертає значення типу `boolean` з налаштувань за ключем `key`, або значення за замовчуванням `defaultValue`, якщо значення не знайдено.

- `readString(String key, String defaultValue)`: Повертає значення типу `String` з налаштувань за ключем `key`, або значення за замовчуванням `defaultValue`, якщо значення не знайдено.

- `readInt(String key, int defaultValue)`: Повертає значення типу `int` з налаштувань за ключем `key`, або значення за замовчуванням `defaultValue`, якщо значення не знайдено.

- `readLong(String key, long defaultValue)`: Повертає значення типу `long` з налаштувань за ключем `key`, або значення за замовчуванням `defaultValue`, якщо значення не знайдено.

6. Загальний принцип роботи:

- Клас `Prefs` дозволяє зручно зберігати та отримувати різні типи даних у налаштуваннях за допомогою `SharedPreferences`.

- Методи запису `write` використовують редактор `SharedPreferences.Editor`, щоб зберегти значення у налаштуваннях.

- Методи читання `read` використовують методи `getBoolean`, `getString`, `getInt`, `getLong` об'єкта `SharedPreferences` для отримання значень з налаштувань.

```

1 package com.ed.app.data.prefs;
2
3 > import ...
4
5
6
7 public class Prefs {
8     public static final String HELP = "HELP";
9     public static final String HELP_SOUND = "HELP_SOUND";
10    public static final String ENLARGED_VIEW = "ENLARGED_VIEW";
11    public static final String EMAIL = "EMAIL";
12
13    public static final String NAME = "NAME";
14    public static final String SETTINGS_PASSWORD = "SETTINGS_PASSWORD";
15    public static final String AUTO_SEND_STATISTICS = "AUTO_SEND_STATISTICS";
16    public static final String AUTO_SEND_STATISTICS_PERIOD = "AUTO_SEND_STATISTICS_PERIOD";
17    public static final String AUTO_SEND_STATISTICS_LAST_DATE = "AUTO_SEND_STATISTICS_LAST_DATE";
18    public static final String AUTO_SEND_STATISTICS_AFTER_SESSION = "AUTO_SEND_STATISTICS_AFTER_SESSION";
19    private final SharedPreferences prefs;
20    @ public Prefs(Context context) {
21        prefs = context.getSharedPreferences( name: "prefs", Context.MODE_PRIVATE);
22    }
23
24    public void write(
25        String key,

```

Рис. 3.4 Структура класу `Prefs`

У вкладені `room` ми можемо побачити елемент `entities` який містить такі класи та інтерфейс:

Клас `BaseStats`:

1. Опис класу:

- `BaseStats` є базовим класом для об'єктів, які будуть зберігатися в базі даних за допомогою бібліотеки `Room Persistence` в `Android`.

- Він містить поля, які характеризують статистичні дані: `id`, `usedTime`, `level`, `dateTime`.

2. Поля класу:

- `id`: Унікальний ідентифікатор із автогенерацією значень (`autoGenerate = true`), що дозволяє автоматично створювати унікальний ідентифікатор для кожного об'єкта.

- usedTime: Час використання.
- level: Рівень.
- dateTime: Дата та час, коли були збережені ці статистичні дані.

3. Похідні класи:

Класи Stats1 до Stats7 є похідними від BaseStats. Вони використовуються для конкретизації статистичних даних для різних типів вправ або сценаріїв.

4. Анотації:

- @Entity: Вказує, що клас є сутністю бази даних Room.
- @PrimaryKey: Вказує, що поле id є первинним ключем.

```

1 package com.ed.app.data.room.entities;
2
3 import androidx.room.PrimaryKey;
4
5 7 inheritors
6 public class BaseStats {
7     @PrimaryKey(autoGenerate = true)
8     public int id;
9     38 usages
10    public String usedTime;
11    public int level;
12    38 usages
13    public String dateTime;
14 }

```

Рис. 3.5 Структура класу BaseStats

Класи Stats1 до Stats7:

1. Опис класів:

- Класи Stats1 до Stats7 є конкретними реалізаціями статистичних даних для різних типів вправ або сценаріїв.
- Вони наслідують базовий клас BaseStats, щоб успадкувати його поля та функціональність.

2. Анотації:

- @Entity: Вказує, що клас є сутністю бази даних Room.
- @PrimaryKey: Вказує, що поле id є первинним ключем.
- @Entity(tableName = "table1"): Вказує на ім'я таблиці, у яку будуть зберігатися об'єкти цього класу.

```

1 package com.ed.app.data.room.entities;
2
3 > import ...
4
5
6
7 13 usages
8 @Entity(tableName = "table1")
9 public class Stats1 extends BaseStats {
10
11 }

```

Рис. 3.6 Структура класу Stats

Клас AppDatabase:

1. Опис класу:

- AppDatabase є абстрактним класом, що розширює RoomDatabase.
- Він визначає базу даних Room для додатку та включає сутності, які будуть зберігатися в цій базі даних, та версію бази даних.

2. Анотації:

- @Database: Вказує, що клас є базою даних Room. Визначає список сутностей та версію бази даних.

```

1 package com.ed.app.data.room;
2
3 > import ...
4 4 usages 1 inheritor
5
6 @Database(
7     entities = {
8         Stats1.class,
9         Stats2.class,
10        Stats3.class,
11        Stats4.class,
12        Stats5.class,
13        Stats6.class,
14        Stats7.class,
15    },
16    version = 1
17 )
18 public abstract class AppDatabase extends RoomDatabase {
19     4 usages 1 implementation
20     public abstract StatisticsDao getDao();
21 }
22
23
24

```

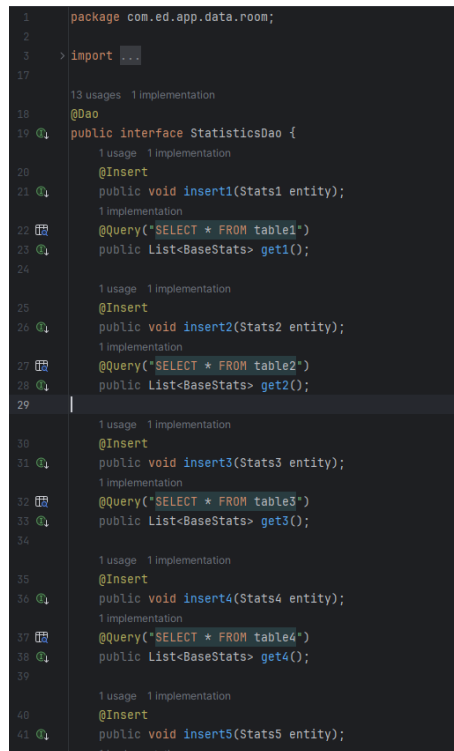
Рис. 3.7 Структура класу BaseStats

Інтерфейс StatisticsDao:

1. Опис інтерфейсу:

- StatisticsDao є DAO (Data Access Object), який визначає методи для доступу до даних в базі даних Room.

- Включає методи для вставки даних та отримання даних для кожного з конкретних типів статистичних даних.



```

1 package com.ed.app.data.room;
2
3 > import ...
4
57
58 @Dao
59 public interface StatisticsDao {
60     1 usage 1 implementation
61     @Insert
62     public void insert1(Stats1 entity);
63     1 implementation
64     @Query("SELECT * FROM table1")
65     public List<BaseStats> get1();
66
67     1 usage 1 implementation
68     @Insert
69     public void insert2(Stats2 entity);
70     1 implementation
71     @Query("SELECT * FROM table2")
72     public List<BaseStats> get2();
73
74     1 usage 1 implementation
75     @Insert
76     public void insert3(Stats3 entity);
77     1 implementation
78     @Query("SELECT * FROM table3")
79     public List<BaseStats> get3();
80
81     1 usage 1 implementation
82     @Insert
83     public void insert4(Stats4 entity);
84     1 implementation
85     @Query("SELECT * FROM table4")
86     public List<BaseStats> get4();
87
88     1 usage 1 implementation
89     @Insert
90     public void insert5(Stats5 entity);
91     1 implementation
  
```

Рис. 3.8 Структура інтерфейс StatisticsDao

Класи PdfGenerator та PdfSingleFileGenerator:

1. Опис класів:

- PdfGenerator та PdfSingleFileGenerator відповідають за генерацію PDF-файлів на основі статистичних даних.

- Вони приймають список статистичних даних та генерують PDF-документ з цими даними.

2. Робота з PDF:

- Обидва класи використовують бібліотеку iText для створення PDF-документів.

- Вони створюють таблиці зі статистичними даними та вставляють їх у документ.

- Клас PdfSingleFileGenerator може генерувати PDF-файл для кожного типу вправ окремо, створюючи відповідні розділи у документі.

```

1 package com.ed.app.data;
2
3 > import ...
4
5 2 usages
6
7 public class PdfGenerator {
8     no usages
9     @
10     public static File generatePdf(
11         Context context,
12         String fileName,
13         String title,
14         String subtitle,
15         List<BaseStats> baseStats
16     ) {
17         File dir = context.getFilesDir();
18         File file = new File(dir, child.fileName + ".pdf");
19
20         try {
21             PdfWriter writer = new PdfWriter(file);
22             PdfDocument pdfDoc = new PdfDocument(writer);
23             Document doc = new Document(pdfDoc);
24
25             doc.setFont(PdfFontFactory.createFont("fontProgram: assets/roboto_regular.ttf", PdfEncodings.
26
27             Table table = new Table( numColumns: 3);
28             table.setHorizontalAlignment(HorizontalAlignment.LEFT);
29             table.setVerticalAlignment(VerticalAlignment.MIDDLE);
30             table.addCell( content: "Витрачено");
31             table.addCell( content: "Рівень");
32             table.addCell( content: "Дата");
33
34             for (BaseStats s : baseStats) {
35                 table.addCell(s.usedTime);
36                 table.addCell(String.valueOf(s.level));
37                 table.addCell(s.dateTime);
38             }
39         }
40     }
41 }

```

Рис. 3.9 Структура класу PdfGenerator та PdfSingleFileGenerator

Клас SendService:

1. Опис класу:

- SendService є службою Android, яка відповідає за відправку статистичних даних з додатку.

- Виконує операцію відправлення в фоновому режимі, щоб не блокувати виконання основного потоку додатку.

2. Методи життєвого циклу служби:

- onStartCommand(): Викликається при запуску служби. Створює канал сповіщень, створює та запускає foreground-службу, перевіряє наявність електронної пошти для відправки.

3. Приватні методи:

- createNotificationChannel(): Створює канал для сповіщень.

- createNotification(): Створює сповіщення про відправку статистики.

- send(): Відправляє статистику через StatisticsSender. При успішній відправці встановлює час останньої відправки, відображає сповіщення про успішне відправлення та завершує foreground-службу.

4. Створення сповіщень:

- showSuccessNotification(): Створює сповіщення про успішну відправку статистики.

5. Інші методи:

- onBind(): Повертає null, оскільки служба не підтримує зв'язок.

- stopForeground(true): Зупиняє foreground-режим служби.

Цей клас відповідає за ініціацію та керування процесом відправлення статистичних даних.

```

1 package com.ed.app.service;
2
3 > import ...
17
18 </> public class SendService extends Service {
19     4 usages
20     private static final String CHANNEL_ID = "REHABILITATION_CHANNEL";
21
22     @Nullable
23     @Override
24     public IBinder onBind(Intent intent) { return null; }
25
26
27     @Override
28     public int onStartCommand(Intent intent, int flags, int startId) {
29         createNotificationChannel();
30         Notification notification = createNotification();
31         startForeground(100, notification);
32
33         Prefs prefs = new Prefs(context, this);
34         String email = prefs.readString(Prefs.EMAIL, defaultValue: null);
35
36         if (email != null) {
37             send();
38         } else {
39             Notification errorNotification = new NotificationCompat.Builder(context, this, CHANNEL_ID)
40                 .setContentTitle("Rehabilitation")
41                 .setContentText("Вкажіть e-mail для відправки статистики")
42                 .setSmallIcon(R.drawable.baseline_send_24)
43                 .setPriority(NotificationCompat.PRIORITY_HIGH)
44                 .build();
45             NotificationManager manager = getSystemService(NotificationManager.class);
46             manager.notify(0, errorNotification);
47
48             stopForeground(removeNotification: true);
49             stopSelf();
50         }
51
52         return START_NOT_STICKY;

```

Рис. 3.10 Структура класу SendService

Якщо переглянути розділ ui, ми побачимо класи розроблені для інтерфейсу користувача:

BaseFragment - це базовий клас фрагмента для Android додатка. Він надає загальні методи та функціональність, що використовуються у багатьох фрагментах.

Клас містить змінні для роботи з потоками, озвученням тексту та збереженням налаштувань користувача. Також він забезпечує збереження статистики гри.

GameOneFragment, тут реалізована функція, яка змінює положення елемента на вказаному полі, для тренування уважності користувача. Його задача запам'ятати положення елемента та в залежності від рівня складності натиснути на поле де елемент було продемонстровано в останнє. Кожен з наступних рівнів додає все більше елементів на поля, для ускладнення завдання, також змінюється швидкість розстановки на позиції. Після відтворення послідовності перевіряється правильність вибору користувача.

GameTwoFragment, демонструє функцію основна задача якої, представляти для користувача перелік випадкових чисел, в якому необхідно виділене число з представлених, в разі правильно вибору, збільшується рівень завдання та його складність, випадкових елементів на вибір стане на один більше. Після вибору поля з числом перевіряється правильність позиції та перехід на наступний рівень складності.

У фрагменті **GameThreeFragment**, реалізована ігрова функція, яка дозволяє користувачеві навчитись відрізняти відтінки кольорів. Всього представлено 9 полів, які містять або зовсім різні відтінки, або однорідні до вказаного. Якщо користувач робить правильний вибір, то кожен наступний рівень трохи посилює контрастність кольору, що ускладнює завдання.

GameFourFragment, представляє собою ігрову функцію, основний мотив якої це розвинення мовної функції користувача, шляхом відговоренням окремих простих або складних звуків, що в свою чергу обумовлено подальшим покращенням діалекту. В процесі виконання вправи, якщо у користувача все вдало виходить, звуки стають складнішими.

У класі **GameFiveFragment**, реалізована ігрова функція, що дозволяє користувачеві вивчати алфавіт та відтворювати звуки кожної букви. При створенні

фрагмент ініціалізує елементи інтерфейсу, такі як кнопки для відтворення звуку та переходу до наступної букви. Кожна буква алфавіту відображається на екрані з можливістю відтворення її звуку за допомогою TextToSpeech.

У фрагменті **GameSixFragment** реалізована ігрова функція, в якій користувач повинен торкатися рухомого об'єкта, щоб відповісти на його переміщення. При створенні фрагменту ініціалізуються елементи інтерфейсу та обчислюються розміри екрану. Коли користувач торкається екрану, об'єкт відображається, і розпочинається процес перевірки та реагування на торкання. Щоб визначити правильність торкання, обчислюється відстань між об'єктом та точкою торкання. Після кожного переміщення об'єкта генеруються нові координати для наступного пункту призначення, і між ними малюються пунктирні лінії.

LineView - це власний клас перегляду (View), який використовується для малювання ліній на екрані. У конструкторах встановлюються значення параметрів, які використовуються для малювання ліній. В методі onDraw встановлюється колір та товщина лінії, а також малюється лінія між двома заданими точками.

GameSevenFragment, ця ігрова функція представляє собою 9 полів які містять в собі об'єкт не схожий на інші. Користувачеві потрібно проаналізувати представлені поля, та знайти невідповідність в об'ємі контуру об'єкта. Після вдалого проходження рівня, складність гри залишається незмінною, але швидкість виконання завдання збільшується, що дає змогу користувачеві швидше вибрати правильне поле.

Фрагмент **HomeFragment** відповідає за відображення домашнього екрану додатка та навігацію до інших фрагментів гри. При створенні фрагменту налаштовується інтерфейс та додаються обробники подій для кнопок, які відповідають за перехід до різних ігрових фрагментів. Кожен обробник подій навігує користувача до відповідного фрагменту за допомогою `Navigation.findNavController(view).navigate()`, передаючи ідентифікатор ресурсу кожного фрагменту.

У класі **MainActivity** наслідується від `AppCompatActivity`. У методі `onCreate()` встановлюється макет `activity_main`, який містить головний контейнер для фрагментів.

У класі **MainFragment** реалізовано функціонал головного екрану додатка.

При кліку на кнопку "play" або "stats" відбувається перехід до відповідних фрагментів. Кнопка "settings" відкриває фрагмент налаштувань, проте якщо встановлено пароль, спочатку виводиться діалогове вікно для введення пароля.

Кнопка "exit" запускає відправку статистики та закриває додаток.

У фрагменті **SettingsFragment** реалізовано налаштування додатку, включаючи зміну паролю, імені користувача, електронної пошти та інші параметри. Пароль, ім'я та електронна пошта зберігаються в `SharedPreferences`. Після введення нового паролю відправляється лист на вказану електронну адресу. Крім того, користувач може увімкнути або вимкнути автоматичне надсилання статистики, встановити період надсилання та змінити інші налаштування.

У фрагменті **StatisticsFragment** відображаються статистичні дані збережені в базі даних. Фрагмент використовує адаптер для відображення даних у вигляді рядків. Параметр `MODE` використовується для вибору режиму статистики (наприклад, рівень гри або інший параметр). Залежно від значення режиму, відповідний запит виконується до бази даних, і результати відображаються у вигляді рядків в таблиці.

Фрагмент **StatisticsMenuFragment** відповідає за відображення меню статистики та обробку відповідних подій користувача. При натисканні кнопки "Надіслати статистику", викликається метод `send()` класу `StatisticsSender`, який відправляє статистику на вказану електронну пошту.

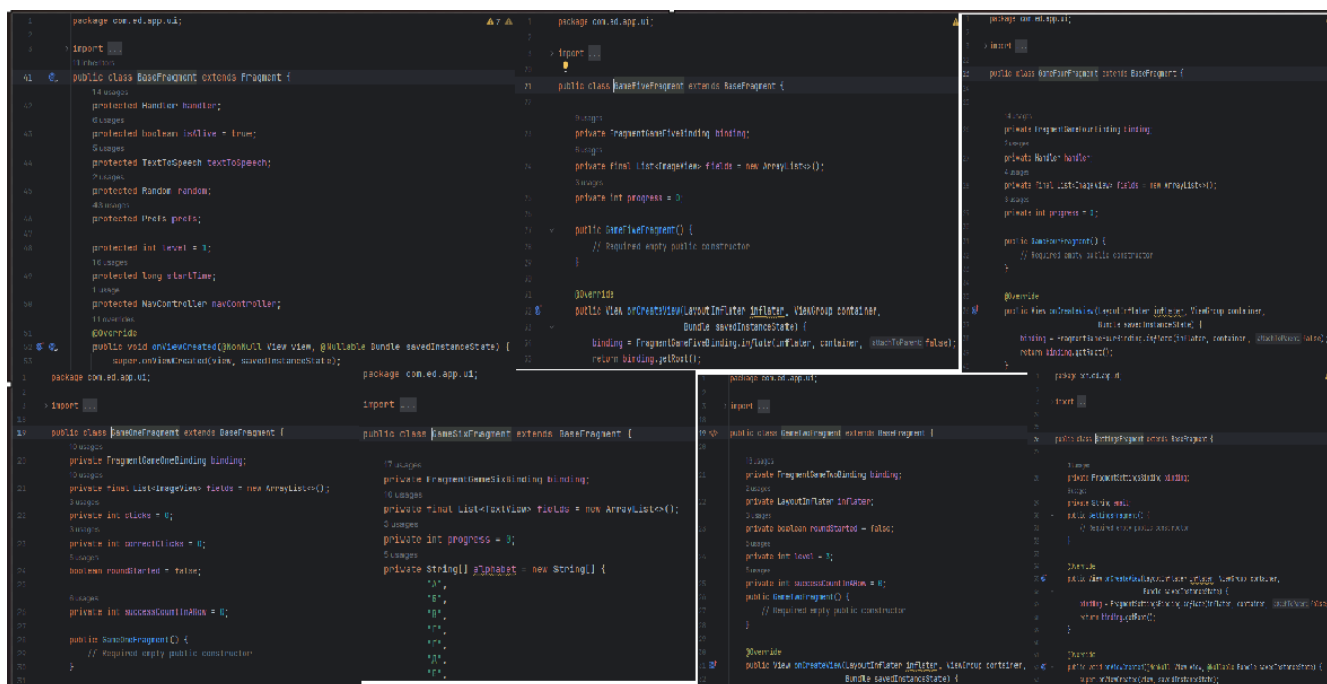


Рис. 3.11 Структура класів елементу User Interface

Пакет Database містить клас AppDatabase_Impl, який є реалізацією бази даних AppDatabase. Основні складові пакету:

1. AppDatabase_Impl: Це реалізація бази даних, яка наслідує клас RoomDatabase. В цьому класі здійснюються операції з базою даних, такі як відкриття та закриття з'єднання, створення та видалення таблиць, очищення всіх таблиць, отримання об'єкту DAO тощо.

2. createOpenHelper(): Цей метод створює об'єкт 'SQLiteOpenHelper', який відповідає за створення та управління базою даних SQLite.

3. createInvalidationTracker(): Цей метод створює об'єкт 'InvalidationTracker', який відслідковує зміни в базі даних та надсилає сповіщення при необхідності.

4. clearAllTables(): Цей метод виконує очищення всіх таблиць в базі даних. Він викликається, наприклад, при перезапуску додатка або при видаленні всіх даних.

5. getRequiredTypeConverters(): Цей метод повертає мапу, яка вказує, які конвертери типів необхідні для використання в базі даних.

6. getRequiredAutoMigrationSpecs(): Цей метод повертає набір специфікацій автоматичної міграції, які необхідні для міграції бази даних між версіями.

7. `getAutoMigrations()`: Цей метод повертає список міграцій, які необхідно виконати автоматично за допомогою специфікацій автоматичної міграції.

8. `getDao()`: Цей метод повертає об'єкт DAO (об'єкт доступу до даних), який дозволяє виконувати операції з базою даних, такі як вставка, оновлення, видалення та запити даних.

Крім того, в цьому пакеті також містяться імпорти необхідних класів та інтерфейсів, таких як `'RoomDatabase'`, `'SupportSQLiteOpenHelper'`, `'Migration'`, `'TableInfo'` та інші.

```

1 package com.ed.app.data.room;
2
3 > import ...
4
5 /unchecked, deprecation/
6 public final class AppDatabase_Impl extends AppDatabase {
7     1 usage
8     @Override
9     @NonNull
10    protected SupportSQLiteOpenHelper createOpenHelper(@NonNull final DatabaseConfiguration config) {
11        final SupportSQLiteOpenHelper.Callback _openCallback = new RoomOpenHelper(config, new RoomOpenHel
12        2 usages
13        @Override
14        public void createAllTables(@NonNull final SupportSQLiteDatabase db) {
15            db.execSQL("CREATE TABLE IF NOT EXISTS 'table1' ('id' INTEGER PRIMARY KEY AUTOINCREMENT NO
16            db.execSQL("CREATE TABLE IF NOT EXISTS 'table2' ('id' INTEGER PRIMARY KEY AUTOINCREMENT NO
17            db.execSQL("CREATE TABLE IF NOT EXISTS 'table3' ('id' INTEGER PRIMARY KEY AUTOINCREMENT NO
18            db.execSQL("CREATE TABLE IF NOT EXISTS 'table4' ('id' INTEGER PRIMARY KEY AUTOINCREMENT NO
19            db.execSQL("CREATE TABLE IF NOT EXISTS 'table5' ('id' INTEGER PRIMARY KEY AUTOINCREMENT NO
20            db.execSQL("CREATE TABLE IF NOT EXISTS 'table6' ('id' INTEGER PRIMARY KEY AUTOINCREMENT NO
21            db.execSQL("CREATE TABLE IF NOT EXISTS 'table7' ('id' INTEGER PRIMARY KEY AUTOINCREMENT NO
22            db.execSQL("CREATE TABLE IF NOT EXISTS room_master_table (id INTEGER PRIMARY KEY,identity_
23            db.execSQL("INSERT OR REPLACE INTO room_master_table (id,identity_hash) VALUES(42, '1d050e
24        }
25
26        1 usage
27        @Override
28        public void dropAllTables(@NonNull final SupportSQLiteDatabase db) {
29            db.execSQL("DROP TABLE IF EXISTS 'table1'");
30            db.execSQL("DROP TABLE IF EXISTS 'table2'");
31            db.execSQL("DROP TABLE IF EXISTS 'table3'");
32            db.execSQL("DROP TABLE IF EXISTS 'table4'");
33            db.execSQL("DROP TABLE IF EXISTS 'table5'");
34            db.execSQL("DROP TABLE IF EXISTS 'table6'");
35        }

```

Рис. 3.12 Структура класу AppDataBase

Всередині конструктора він ініціалізує декілька об'єктів **EntityInsertionAdapter**, по одному для кожного класу сутностей (від `Stats1` до `Stats7`). Ці адаптери використовуються для виконання операцій вставки у відповідні таблиці бази даних. Він реалізує методи для вставки даних у кожну таблицю (від `insert1` до `insert7`) шляхом виклику відповідного методу `insert` відповідного `EntityInsertionAdapter`.

Реалізовано методи для отримання даних з кожної таблиці (від get1 до get7) шляхом виконання сирих SQL-запитів та відображення результатів в об'єкти BaseStats.

Метод getRequiredConverters повертає порожній список, вказуючи на те, що для цього DAO не потрібні конвертери типів.

```

1 package com.ed.app.data.room;
2
3 > import ...
26
27 /unchecked, deprecation/
28 public final class StatisticsDao_Impl implements StatisticsDao {
29     43 usages
30     private final RoomDatabase __db;
31
32     2 usages
33     private final EntityInsertionAdapter<Stats1> __insertionAdapterOfStats1;
34
35     2 usages
36     private final EntityInsertionAdapter<Stats2> __insertionAdapterOfStats2;
37
38     2 usages
39     private final EntityInsertionAdapter<Stats3> __insertionAdapterOfStats3;
40
41     2 usages
42     private final EntityInsertionAdapter<Stats4> __insertionAdapterOfStats4;
43
44     2 usages
45     private final EntityInsertionAdapter<Stats5> __insertionAdapterOfStats5;
46
47     2 usages
48     private final EntityInsertionAdapter<Stats6> __insertionAdapterOfStats6;
49
50     2 usages
51     private final EntityInsertionAdapter<Stats7> __insertionAdapterOfStats7;
52
53     1 usage
54     public StatisticsDao_Impl(@NonNull final RoomDatabase __db) {
55         this.__db = __db;
56         this.__insertionAdapterOfStats1 = new EntityInsertionAdapter<Stats1>(__db) {
57             1 usage
58             @Override
59             @NonNull

```

Рис. 3.13 Структура класу StatisticsDao

Пакет res містить структуру папок ресурсів програми. Містить такі папки:

А) папка drawable;

Для зображень. Зображення розраховані на відповідні роздільну здатність екрана мобільного пристрою.

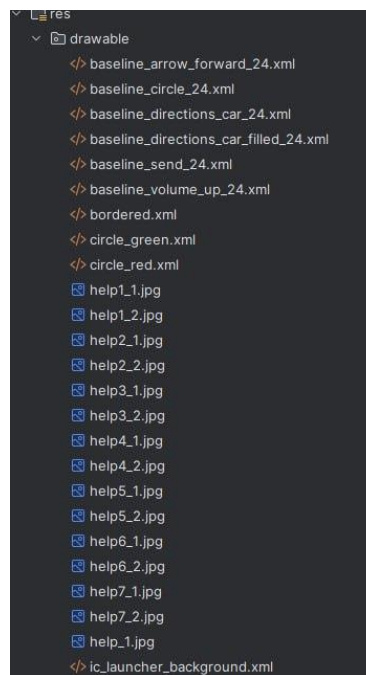


Рис. 3.14 Вміст папки drawable

Б) папка layout;

Для XML-файлів компонування (компонування графічних елементів керування/розмітка).

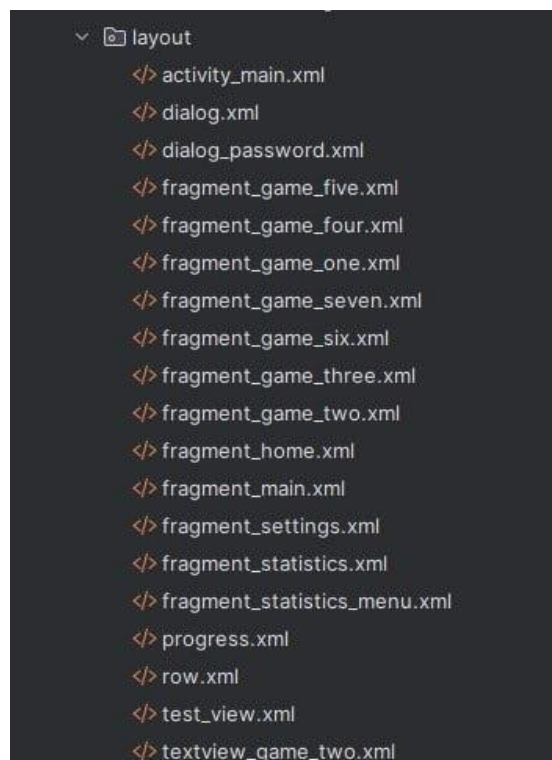


Рис. 3.15 Вміст папки layout

На рисунку 3.16 представлена діаграма класів та порядок їх взаємодії між собою.

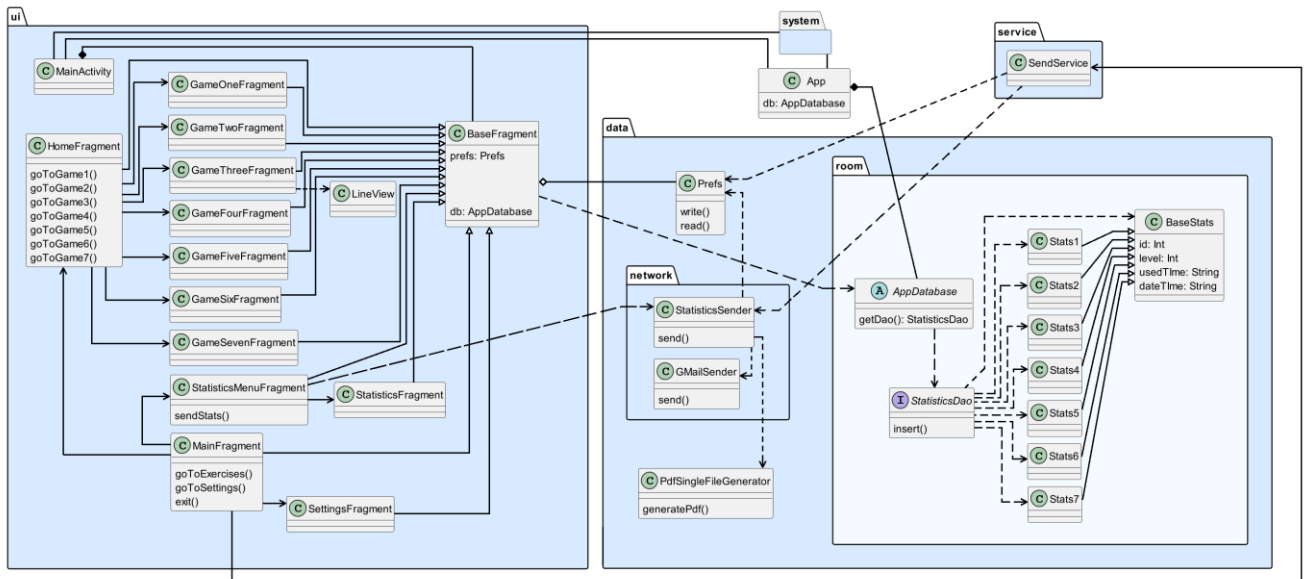


Рис 3.16 Діаграма класів

3.3 Опис функціонування програми

При першому завантаженні програми з'являються головна активність із списком активних елементів. Ця активність представлена на рисунку 3.17.

Для того щоб користувач міг зареєструвати себе, потрібно перейти в налаштування та додати свої персональні дані, задати пароль, зробити можливість відправлення результатів на Email та відкоригувати налаштування відносно «автоматичної відправки», «увімкнення підказок», «увімкнення звуку».

Якщо користувач заповнив всі поля у вікні «Налаштування» (рис.3.18), він може переходити до вибору необхідної для себе вправи в першій активності «Головного меню» додатку, всі представлені для вирішення завдання показані на рисунку 3.19.

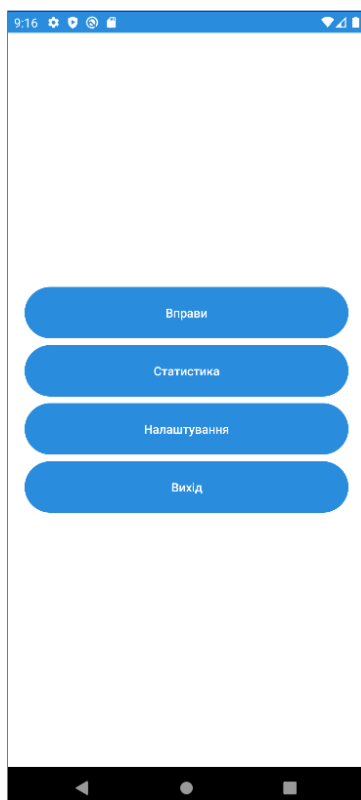


Рис. 3.17 Головна активність додатку при першому запуску

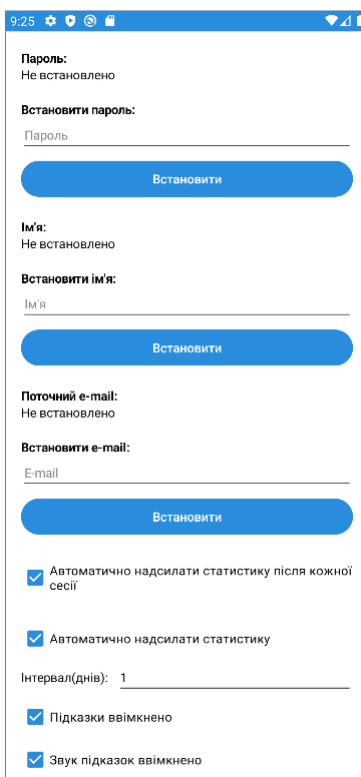


Рис. 3.18 Активність вікна Налаштування

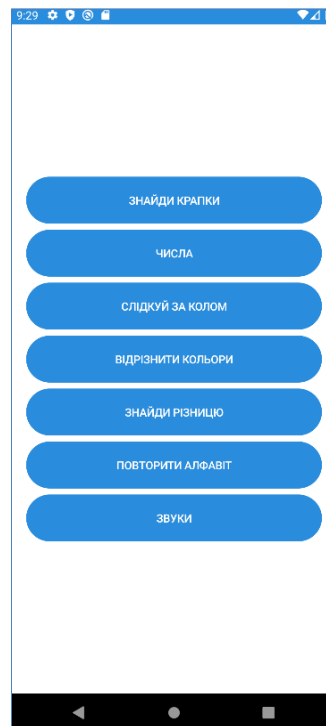


Рис. 3.19 Активність вікна Вправи

В залежності від проблем користувача відносно відновлення когнітивних функцій, можна або вибирати вправи за власним бажанням, або ж пройти всі послідовно для відпрацювання всіх можливих варіантів перевірки поточного стану. Перелік вправ в додатку представлено на рис. 3.20.

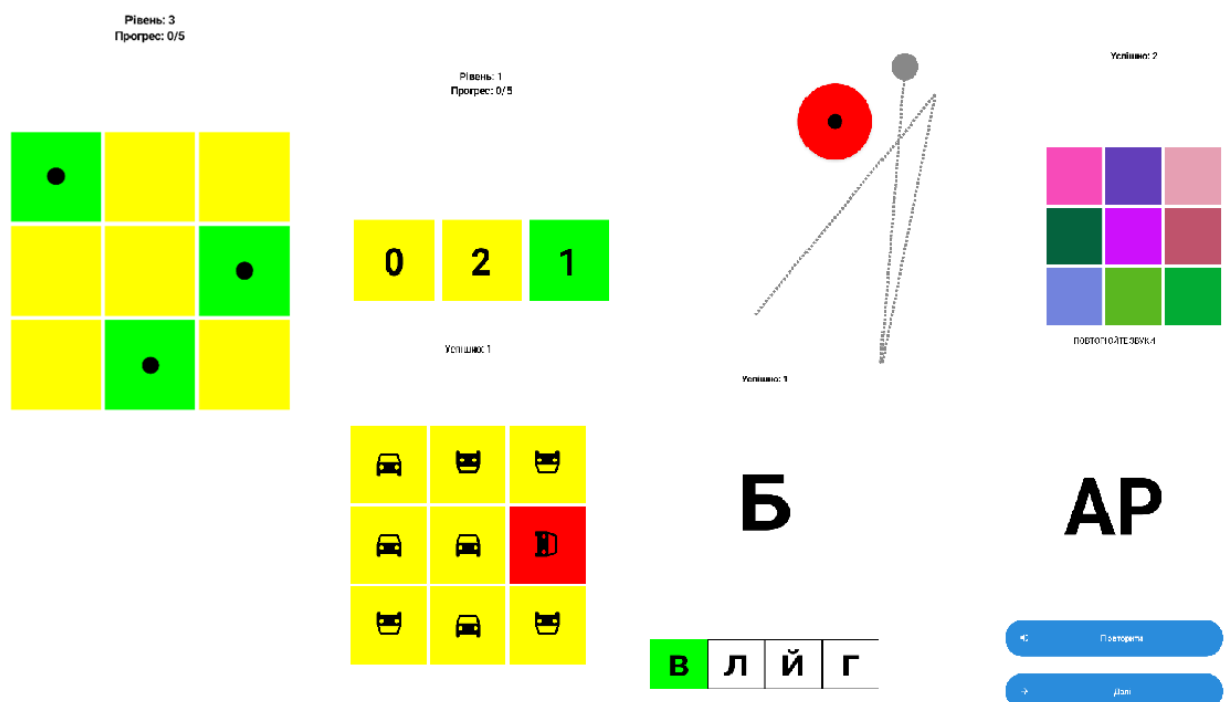


Рис. 3.20 Список активних вправ для тренування

Після завершення виконання вправ користувачем, він може переглянути свої результати у вікні активності «Статистика», в ній можна побачити свою статистику по кожній з виконаних користувачем вправ, після перегляду її можна відправити на Email, який був вказаний в налаштуваннях користувача (рис.3.21).



Рис. 3.21 Список активностей вікна Статистика

Зайшовши сюди користувач може переглянути свої статистику по всім виконаним вправам, а саме – час виконання, рівень та дату. Результати кожного з пройдених завдань показано на рисунку 3.22.

Витрачено 152 с	Рівень 3	Дата 10-05-2024 09:36	Вправа - 1
Витрачено 32 с	Рівень 1	Дата 10-05-2024 09:36	Вправа - 2
20 с	1	10-05-2024 09:37	
Витрачено 25 с	Рівень 1	Дата 08-05-2024 18:22	Вправа - 3
29 с	4	10-05-2024 09:38	
Витрачено 36 с	Рівень 2	Дата 08-05-2024 18:22	Вправа - 4
25 с	2	10-05-2024 09:38	
Витрачено 18 с	Рівень 1	Дата 10-05-2024 09:39	Вправа - 5
54 с	1	10-05-2024 09:40	
Витрачено 24 с	Рівень 2	Дата 10-05-2024 09:40	Вправа - 6
Витрачено 245 с	Рівень 1	Дата 08-05-2024 18:21	Вправа - 7
111 с	1	10-05-2024 09:42	

Рис. 3.22 Статистика пройдених вправ

Висновки до третього розділу

Представлено опис реалізації програми, включаючи вибір платформи та технологій, а також розробку архітектури додатку. Цей аспект розділу дозволив отримати уявлення про технічні рішення, прийняті під час розробки, та вибрані інструменти для досягнення бажаних цілей.

Розроблено діаграму класів, що відображає структуру програми та обґрунтовувала алгоритм основної механіки. Це дозволило краще зрозуміти організацію коду та логіку роботи додатку з точки зору програмування.

У розділі також було описано функціонування програми, включаючи діаграму діяльності. Це надало чітке уявлення про те, як саме працює програма в реальному часі та які операції виконуються при взаємодії користувача з додатком.

4 ТЕСТУВАННЯ ЗАСТОСУНКУ ДЛЯ ВІДНОВЛЕННЯ КОГНІТИВНИХ ФУНКЦІЙ

4.1 Обґрунтування вибору тест-кейсів

1. ExampleInstrumentedTest:

- Контекст Android додатку: Цей тест перевіряє, чи правильно встановлюється пакет для Android додатку. Він використовує InstrumentationRegistry, щоб отримати контекст додатку, а потім перевіряє, чи повертає метод `getPackageName()` очікуване значення (в даному випадку "com.ed.app").
- Важливість пакету для додатку: Пакет додатку є одним з основних ідентифікаторів додатку в Android. Його правильність встановлення важлива для багатьох аспектів, таких як компоненти, дозволи, конфігурація та інше. Тому важливо перевірити, чи встановлено пакет додатку правильно.
- Основний тест на рівні системи: Цей тест виконується на реальному Android пристрої або симуляторі, тому він дає найбільш достовірні результати щодо роботи додатку в реальних умовах.

2. ExampleUnitTest:

- Простий приклад функціональності: Тест перевіряє, чи працює операція додавання двох чисел. Це один з найпростіших тестів, але він важливий, оскільки перевірка основних математичних операцій є важливою частиною тестування програм.
- Юніт-тест на рівні модуля: Цей тест виконується локально на вашій розробницькій машині, що робить його швидшим у виконанні та менш залежним від середовища. Він дозволяє вам швидко перевірити, чи функція додавання працює правильно, без необхідності запускати додаток на реальному пристрої або емуляторі.

– Покриття базових сценаріїв: Хоча цей тест є дуже простим, він перевіряє базову функціональність, яка може бути використана в багатьох частинах програми. Такі прості тести допомагають виявити проблеми в базових компонентах програми та їх взаємодії, що дозволяє запобігти виникненню більш складних проблем у майбутньому.

Обидва ці тест-кейси обираються через їх важливість для забезпечення правильної роботи додатку на різних рівнях (системному та модульному), а також через їх спроможність перевірити основні аспекти функціональності додатку.

4.2 Розробка та виконання тест-кейсів

ExampleInstrumentedTest являється інструментованого тесту в Android додатку. Розглянемо його детальніше:

1. Package і Imports: Починаємо з визначення пакету `com.ed.app` і імпортів необхідних класів, таких як `Context`, `InstrumentationRegistry`, `AndroidJUnit4`, `Test` і т.д.

2. Анотація `@RunWith(AndroidJUnit4.class)`: Ця анотація позначає клас, який буде виконувати тест. У цьому випадку використовується `AndroidJUnit4`, який дозволяє запускати тести на Android пристроях або симуляторах.

3. Метод `useAppContext()`: Цей метод є фактичним тестом. Він перевіряє, чи повертає метод `getPackageName()` контексту додатку очікуване значення. У цьому випадку очікується, що пакетом додатку є `"com.ed.app"`.

4. Отримання контексту додатку: Для того, щоб виконати тест, нам потрібно отримати контекст додатку. Це робиться за допомогою `InstrumentationRegistry.getInstrumentation().getTargetContext()`, який повертає контекст додатку.

5. Перевірка: В даному випадку використовується метод `assertEquals()`, щоб порівняти фактичне значення пакету з очікуваним значенням.

6. Коментарі у коді пояснюють призначення класу та методів, що допомагає зрозуміти, що робить цей код.

ExampleInstrumentedTest вважається доволі простим, але важливим, оскільки він перевіряє основну функціональність - чи правильно встановлюється пакет для додатку. Такі тести є першим кроком у впевненості, що базові складові вашого додатку працюють належним чином.

```

1  package com.ed.app;
2
3  > import ...
12
13  /**
14   * Instrumented test, which will execute on an Android device.
15   *
16   * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
17   */
18  @RunWith(AndroidJUnit4.class)
19  public class ExampleInstrumentedTest {
20
21      @Test
22      public void useAppContext() {
23          // Context of the app under test.
24          Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
25          assertEquals("com.ed.app", appContext.getPackageName());
26      }
27  }

```

Рис. 4.1 – Реалізація Unit-тесту ExampleInstrumentedTest

ExampleUnitTest є локальним юніт-тестом для Java-класу ExampleUnitTest.

Давайте розглянемо кожен елемент цього тесту:

Package і Imports: Клас належить пакету com.ed.app, імпортовано лише необхідний клас Test і метод assertEquals з org.junit. Метод addition_isCorrect(): Цей метод є фактичним тестом. Він перевіряє, чи є сума двох чисел ($2 + 2$) рівною 4.

Це робиться за допомогою методу assertEquals(), який порівнює очікуване значення (у цьому випадку 4) з фактичним результатом виразу ($2 + 2$). Коментарі: Коментарі в коді пояснюють призначення класу та методів.

Цей юніт-тест виконує дуже просту перевірку, а саме, чи правильно працює додавання чисел. Це типовий приклад тесту одиничної функціональності, який допомагає перевірити, чи працює певний фрагмент коду, як очікувалося. Такі тести особливо корисні під час розробки, коли потрібно переконатися, що функції виконуються правильно до інтеграції з іншими частинами програми.

```
1 package com.ed.app;
2
3 > import ...
4
5
6
7 /**
8  * Example local unit test, which will execute on the development machine (host).
9  *
10  * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
11  */
12 public class ExampleUnitTest {
13     @Test
14     > public void addition_isCorrect() { assertEquals( expected: 4, actual: 2 + 2); }
15
16
17 }
```

Рис. 4.2 – Реалізація Unit-тесту ExampleInstrumentedTest

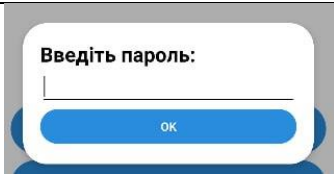
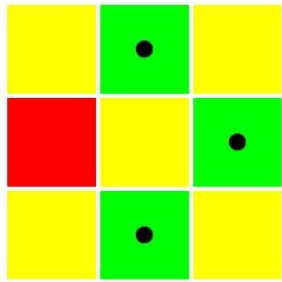
Таблиця 4.1

Тестування повного алгоритму роботи додатку

№	Вимоги	Опис тестового кейсу	Очікувані результати	Реальний результат
1	1. Запустити програму. 2. Натиснути налаштування. 3. Заповнити поля «не встановлено». 4. Натиснути «встановити»	Нам потрібно встановити такі параметри користувача, як: пароль, ім'я та e-mail. Це необхідно для встановлення статистики та відправки результатів.	Зміна поля кожного з зазначених параметрів на «встановлено». Зазначено, що користувач, створений та можна відправляти статистику.	

Продовження Таблиці 4.1

Тестування повного алгоритму роботи додатку

№	Вимоги	Опис тестового кейсу	Очікувані результати	Реальний результат
2	1. Після збереження повернутись в налаштування. 2. Ввести секретний пароль.	Після збереження персональних даних нашого користувача, повертаємося в меню налаштування.	При натисканні на поле нам повинно відобразитись вікно введення секретного паролю для доступу до даних користувача.	
3	1. Перейти у вікно налаштування. 2. Вибрати елемент перемикач. (представлено 5 елементів)	В цьому тест-кейсі нам потрібно перевірити роботу перемикачів, в самому вікні ми зробимо першу перевірку ввімкнувши «Збільшений вигляд»	Під час натискання на елемент розмір нашого інтерфейсу повинен збільшитись.	Встановити e-mail: crechet.rbbt@gmail.com Встановити Автоматично надсилати ☒ статистику після кожної сесії ☒ Автоматично надсилати статистику Інтервал(днів): 2 ☐ Підказки ввімкнено ☐ Звук підказок ввімкнено ☒ Збільшений вигляд Повернутися
4	1. Повернутись до головного меню. 2. Перейти у вкладку «Вправи». 3. Вибрати необхідні вправи для тестування.	Потрібно вибрати перелік вправ для тестування. Перевірка відображення, оптимізації та швидкодії.	Виконуючи вправу нам потрібно пройти декілька рівнів, щоб побачити наскільки багато рівнів ми зможемо пройти, та чи не зміниться швидкість роботи додатку в залежності від важчого рівню завдання.	Рівень: 6 Прогрес: 5/5 

Продовження Таблиці 4.1

Тестування повного алгоритму роботи додатку

№	Вимоги	Опис тестового кейсу	Очікувані результати	Реальний результат																																																																					
				Рівень: 2 Прогрес: 4/5 120																																																																					
5	1. Повернутись в Головне меню. 2. Перейти у вкладку «Статистика». 3. Вибрати вправу з переліку.	При переході на вкладку «Статистика», потрібно вибрати одну з пройдених раніше вправ користувачем, щоб побачити статистику.	Коли буде відкрито вибрана нами вправа, ми повинні побачити 3 поля: 1. Скільки часу витрачено. 2. Якого рівня досягнуто. 3. Коли була виконана вправа.	<table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>32 c</td><td>1</td><td>10-05-2024 09:36</td></tr><tr><td>20 c</td><td>1</td><td>10-05-2024 09:37</td></tr><tr><td>79 c</td><td>1</td><td>13-05-2024 20:30</td></tr></table> <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>36 c</td><td>2</td><td>08-05-2024 18:22</td></tr><tr><td>25 c</td><td>2</td><td>10-05-2024 09:38</td></tr><tr><td>41 c</td><td>11</td><td>13-05-2024 20:28</td></tr></table> <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>36 c</td><td>2</td><td>08-05-2024 18:22</td></tr><tr><td>25 c</td><td>2</td><td>10-05-2024 09:38</td></tr><tr><td>41 c</td><td>11</td><td>13-05-2024 20:28</td></tr></table> <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>18 c</td><td>1</td><td>10-05-2024 09:39</td></tr><tr><td>54 c</td><td>1</td><td>10-05-2024 09:40</td></tr><tr><td>73 c</td><td>11</td><td>13-05-2024 20:27</td></tr></table> <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>24 c</td><td>2</td><td>10-05-2024 09:40</td></tr><tr><td>146 c</td><td>27</td><td>13-05-2024 20:26</td></tr></table> <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>245 c</td><td>1</td><td>08-05-2024 18:21</td></tr><tr><td>111 c</td><td>1</td><td>10-05-2024 09:42</td></tr><tr><td>16 c</td><td>1</td><td>13-05-2024 20:26</td></tr></table>	Витрачено	Рівень	Дата	32 c	1	10-05-2024 09:36	20 c	1	10-05-2024 09:37	79 c	1	13-05-2024 20:30	Витрачено	Рівень	Дата	36 c	2	08-05-2024 18:22	25 c	2	10-05-2024 09:38	41 c	11	13-05-2024 20:28	Витрачено	Рівень	Дата	36 c	2	08-05-2024 18:22	25 c	2	10-05-2024 09:38	41 c	11	13-05-2024 20:28	Витрачено	Рівень	Дата	18 c	1	10-05-2024 09:39	54 c	1	10-05-2024 09:40	73 c	11	13-05-2024 20:27	Витрачено	Рівень	Дата	24 c	2	10-05-2024 09:40	146 c	27	13-05-2024 20:26	Витрачено	Рівень	Дата	245 c	1	08-05-2024 18:21	111 c	1	10-05-2024 09:42	16 c	1	13-05-2024 20:26
Витрачено	Рівень	Дата																																																																							
32 c	1	10-05-2024 09:36																																																																							
20 c	1	10-05-2024 09:37																																																																							
79 c	1	13-05-2024 20:30																																																																							
Витрачено	Рівень	Дата																																																																							
36 c	2	08-05-2024 18:22																																																																							
25 c	2	10-05-2024 09:38																																																																							
41 c	11	13-05-2024 20:28																																																																							
Витрачено	Рівень	Дата																																																																							
36 c	2	08-05-2024 18:22																																																																							
25 c	2	10-05-2024 09:38																																																																							
41 c	11	13-05-2024 20:28																																																																							
Витрачено	Рівень	Дата																																																																							
18 c	1	10-05-2024 09:39																																																																							
54 c	1	10-05-2024 09:40																																																																							
73 c	11	13-05-2024 20:27																																																																							
Витрачено	Рівень	Дата																																																																							
24 c	2	10-05-2024 09:40																																																																							
146 c	27	13-05-2024 20:26																																																																							
Витрачено	Рівень	Дата																																																																							
245 c	1	08-05-2024 18:21																																																																							
111 c	1	10-05-2024 09:42																																																																							
16 c	1	13-05-2024 20:26																																																																							
6	1. Повернутись в головне меню вкладки «Статистика». 2. Натиснути на поле «Надіслати статистику».	Перевіривши статистику всіх пройдених користувачем вправ нам необхідно надіслати її на E-mail.	Натиснувши поле «Надіслати статистику», на вказану раніше користувачем пошту повинні надійти результати виконання вправ в форматі pdf.																																																																						

Продовження Таблиці 4.1

Тестування повного алгоритму роботи додатку

№	Вимоги	Опис тестового кейсу	Очікувані результати	Реальний результат																																																																		
				Ім'я: TestUser Вправа 1 <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>152 с</td><td>3</td><td>10-05-2024 09:36</td></tr><tr><td>224 с</td><td>6</td><td>13-05-2024 20:21</td></tr></table> Вправа 2 <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>32 с</td><td>1</td><td>10-05-2024 09:36</td></tr><tr><td>20 с</td><td>1</td><td>10-05-2024 09:37</td></tr><tr><td>79 с</td><td>1</td><td>13-05-2024 20:30</td></tr></table> Вправа 3 <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>25 с</td><td>1</td><td>08-05-2024 18:22</td></tr><tr><td>29 с</td><td>4</td><td>10-05-2024 09:38</td></tr><tr><td>45 с</td><td>13</td><td>13-05-2024 20:29</td></tr></table> Вправа 4 <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>36 с</td><td>2</td><td>08-05-2024 18:22</td></tr><tr><td>25 с</td><td>2</td><td>10-05-2024 09:38</td></tr><tr><td>41 с</td><td>11</td><td>13-05-2024 20:28</td></tr></table> Вправа 5 <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>18 с</td><td>1</td><td>10-05-2024 09:39</td></tr><tr><td>54 с</td><td>1</td><td>10-05-2024 09:40</td></tr><tr><td>73 с</td><td>11</td><td>13-05-2024 20:27</td></tr></table> Вправа 6 <table><tr><th>Витрачено</th><th>Рівень</th><th>Дата</th></tr><tr><td>24 с</td><td>2</td><td>10-05-2024 09:40</td></tr><tr><td>146 с</td><td>27</td><td>13-05-2024 20:26</td></tr></table> Вправа 7	Витрачено	Рівень	Дата	152 с	3	10-05-2024 09:36	224 с	6	13-05-2024 20:21	Витрачено	Рівень	Дата	32 с	1	10-05-2024 09:36	20 с	1	10-05-2024 09:37	79 с	1	13-05-2024 20:30	Витрачено	Рівень	Дата	25 с	1	08-05-2024 18:22	29 с	4	10-05-2024 09:38	45 с	13	13-05-2024 20:29	Витрачено	Рівень	Дата	36 с	2	08-05-2024 18:22	25 с	2	10-05-2024 09:38	41 с	11	13-05-2024 20:28	Витрачено	Рівень	Дата	18 с	1	10-05-2024 09:39	54 с	1	10-05-2024 09:40	73 с	11	13-05-2024 20:27	Витрачено	Рівень	Дата	24 с	2	10-05-2024 09:40	146 с	27	13-05-2024 20:26
Витрачено	Рівень	Дата																																																																				
152 с	3	10-05-2024 09:36																																																																				
224 с	6	13-05-2024 20:21																																																																				
Витрачено	Рівень	Дата																																																																				
32 с	1	10-05-2024 09:36																																																																				
20 с	1	10-05-2024 09:37																																																																				
79 с	1	13-05-2024 20:30																																																																				
Витрачено	Рівень	Дата																																																																				
25 с	1	08-05-2024 18:22																																																																				
29 с	4	10-05-2024 09:38																																																																				
45 с	13	13-05-2024 20:29																																																																				
Витрачено	Рівень	Дата																																																																				
36 с	2	08-05-2024 18:22																																																																				
25 с	2	10-05-2024 09:38																																																																				
41 с	11	13-05-2024 20:28																																																																				
Витрачено	Рівень	Дата																																																																				
18 с	1	10-05-2024 09:39																																																																				
54 с	1	10-05-2024 09:40																																																																				
73 с	11	13-05-2024 20:27																																																																				
Витрачено	Рівень	Дата																																																																				
24 с	2	10-05-2024 09:40																																																																				
146 с	27	13-05-2024 20:26																																																																				

Всі тести, зазначені в таблиці 4.1 було завершено з результатом «Тест пройдено».

Висновки до четвертого розділу

У цьому розділі було розглянуто два різні тест-кейси для Android додатку: інструментований тест (ExampleInstrumentedTest) і локальний юніт-тест (ExampleUnitTest). Обидва тестування є важливими для забезпечення правильної роботи додатку на різних рівнях і в різних умовах.

Інструментований тест дозволяє перевірити функціональність додатку на реальному Android пристрої або емуляторі, забезпечуючи найбільш достовірні

результати. Він особливо важливий для перевірки основних параметрів додатку, таких як пакет та інші, які можуть впливати на взаємодію з системою Android.

Локальний юніт-тест дозволяє швидко та ефективно перевірити окремі фрагменти коду без необхідності використання реального пристрою або емулятора. Він особливо корисний для перевірки базових функцій та їх взаємодії в межах додатку.

Разом обидва ці типи тестів допомагають забезпечити якість та надійність Android додатку, покриваючи різні аспекти його функціональності на різних рівнях. Це дозволяє впевнитися, що додаток працює належним чином як на розробничій машині, так і на реальних пристроях або емуляторах.

Було проведено мануальне тестування роботи додатку для перевірки коректної роботи всіх наявних функцій. Було перевірено форму заповнення персональних даних користувача (введення секретного паролю, введення імені та e-mail), все працює коректно. Також активні елементи, які розташовані в нижній половині форми виконують свою задачу, перевірено елементом «Збільшений вигляд». Форма вправи містить в собі 7 різноманітних завдань, ми виконали всі вправи, щоб протестувати збільшення рівнів та складності завдання. Цей тест також пройдений успішно. В фінальному етапі, коли всі вправи було завершено, перейшовши в форму статистика можна побачити її для кожного з завдань. Після перевірки статистики і натиснувши на поле «Відправити статистику», ми отримали її на вказану раніше пошту у форматі pdf від автора додатку.

ВИСНОВКИ

1. Проведено аналіз предметної галузі відновлення когнітивних функцій людини та існуючих застосунків для підтримки цих процесів. Ключові проблеми, які можна виділити, пов'язані з відсутністю в застосунках важливих для процесу реабілітації функцій, зокрема, персоналізація підходів до відновлення, процес відстеження наглядачем результатів тренувань.
2. Спроектовано ігрові завдання для підтримки відновлення когнітивних функцій.
3. Реалізовано механіки виконання завдань на уважність, пам'ять, сприйняття та відтворення побаченої інформації.
4. Розроблено застосунок з використанням платформи Android Studio та мови програмування Java.
5. Проведено тестування застосунку з використанням інструментованого тесту, локальних юніт-тестів та мануальних тестів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Croisile, B. Memory stimulation. Which scientific benefits? Which exercises? [online]. La Revue de Gériatrie (French Geriatrics Journal), 2006. [cit. 2020/03/09]. URL: <http://www.happy-neuron.com/rsc/hn4/docs/French%20Geriatrics%20Journal%20%20Dr.%20B%20Croisile%20Publication.pdf> (дата звернення: 20.03.2024).
2. Klucká, J. – Volfová, P. Kognitivní trénink v praxi. Grada Publishing, 2009. ISBN 978-80-247-5580-9 (дата звернення: 22.03.2024).
3. Atkinson, R. C. – Shiffrin, R. Human Memory: A Proposed System and its Control Processes [online]. Elsevier, 1968. [cit. 2020/02/28]. URL: <https://www.sciencedirect.com/science/article/pii/S0079742108604223> (дата звернення: 22.03.2024).
4. Miller, G. A. The magical number seven, plus or minus two: some limits on our capacity for processing information. [online]. Psychological Review, 1956. [cit. 2020/03/20]. URL: <https://psycnet.apa.org/record/1957-02914-001> (дата звернення: 24.03.2024).
5. Squire, L. R. – Knowlton, B. – Musen, G. The Structure and Organization of Memory [online]. Annual Review of Psychology, 1993. [cit. 2020/04/09]. URL: <https://www.annualreviews.org/doi/pdf/10.1146/annurev.ps.44.020193.002321> (дата звернення: 24.03.2024).
6. Clifton, C. et al. Language Comprehension and Production [online]. Comprehensive handbook of psychology, 2013. [cit. 2020/04/09]. URL: https://www.researchgate.net/publication/269700690_Language_Comprehension_and_Production (дата звернення: 25.03.2024).
7. Diamond, A. Executive Functions [online]. Annual Review of Psychology, 2013. [cit. 2020/04/12]. URL: <https://doi.org/10.1146/annurev-psych-113011-143750> (дата звернення: 25.03.2024).
8. About Elevate [online]. Elevate, 2019. [cit. 2019/08/20]. URL:

<https://www.elevateapp.com/about> (дата звернення: 05.04.2024).

9. Press info about NeuroNation [online]. Berliner Synaptikon GmbH, 2019. [cit. 2019/08/20]. URL: <https://sp.neuronation.com/en/press/> (дата звернення: 05.04.2024).

10. CogniFit [online]. CogniFit, 2019. [cit. 2019/08/20]. URL: <https://www.cognifit.com> (дата звернення: 05.04.2024).

11. Kurschl, W.; Augstein, M.; Burger, T.; Pointner, C. User modeling for people with special needs. *Int. J. Pervasive Comput. Commun.* 2014, 10, 313–336. [CrossRef]

<https://pdfs.semanticscholar.org/519d/c560904561f0a6fef3190d42ab653912f863.pdf>

12. Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5(4), 297–323. <https://doi.org/10.1007/BF00122574>

13. Weber, E. U., & Johnson, E. J. (2009). Mindful judgment and decision making. *Annual Review of Psychology*, 60, 53–85. <https://doi.org/10.1146/annurev.psych.60.110707.163633>

14. Wundt, W. M. (1911). *Grundzüge der physiologischen Psychologie* [Basics of physiological psychology]. Leipzig, Germany: Wilhelm Engelmann. Retrieved from <https://archive.org/details/grundzgederphys03goog>

15. Zimmermann, H. J. (1980). Testability and meaning of mathematical models in social sciences. *Mathematical Modelling*. [https://doi.org/10.1016/0270-0255\(80\)90012-3](https://doi.org/10.1016/0270-0255(80)90012-3)

16. J. R. Hajdukiewicz, K. J. Vicente, D. J. Doyle, P. Milgram, and C. M. Burns. Modeling a medical environment: an ontology for integrated medical informatics design. *Int J Medical Informatics*, 62(1):79–99, June 2001.

17. Greg A. Jamieson and Kim J. Vicente. Ecological interface design for petrochemical applications: supporting operator adaptation, continuous learning, and distributed, collaborative work. *Computers & Chemical Engineering*, 25(7-8):1055–1074, August 2001.

18. B. van Dongen, A. de Medeiros, H. Verbeek, A. Weijters, and W. van der

Aalst, Applications and Theory of Petri Nets 2005, 2005, ch. The ProM Framework: A New Era in Process Mining Tool Support, pp. 444–454.

19. Наумов М.С., Золотухіна О.А. Аналіз вимог до мобільного додатку для підтримки відновлення когнітивних функцій: матеріали Всеукраїнської науково-технічна конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». 24.04.2024, ДУІКТ, м. Київ. С. 331.

20. Наумов М.С., Золотухіна О.А. Мобільні додатки для стимулювання мозкової активності: технології, підходи до реалізації та практичне застосування: матеріали «V Всеукраїнська науково-практична конференція молодих вчених та здобувачів вищої освіти присвячена Дню науки СУЧАСНА МОЛОДЬ В СВІТІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ». 17.05.2024. С. 175.

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ-ГРИ ДЛЯ ПІДТРИМКИ ПРОЦЕСІВ ВІДНОВЛЕННЯ КОГНІТИВНИХ ФУНКЦІЙ ЛЮДИНИ МОВОЮ ПРОГРАМУВАННЯ JAVA ДЛЯ ПЛАТФОРМИ ANDROID

Виконав студент 4 курсу
 Групи ПД-44
 Наумов Микита Сергійович
 Керівник роботи

К.т.н., доц., доцент кафедри ІПЗ Золотухіна Оксана Анатоліївна
 Київ – 2024

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – підтримка процесів відновлення когнітивних функцій людини мовою засобами мобільного застосунку-гри для платформи Android.
- **Об'єкт дослідження** – процеси відновлення когнітивних функцій людини.
- **Предмет дослідження** – програмне забезпечення для підтримки процесів відновлення когнітивних функцій людини.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Дослідження та аналіз предметної галузі відновлення когнітивних функцій людини та існуючих застосунків для підтримки процесів відновлення когнітивних функцій людини.
2. Аналіз та розробка вимог до мобільного застосунку-гри для підтримки процесів відновлення когнітивних функцій людини.
3. Проектування ігрових вправ з урахуванням завдань підтримки відновлення когнітивних функцій людини.
4. Реалізація геймплейних механік за допомогою платформи Android Studio та мови програмування Java.
5. Тестування застосунку.



3

АНАЛІЗ АНАЛОГІВ

Показник	Lumosity	Elevate	NeuroNation	Peak	CogniFit	Rehabilitation
Робота без підключення до мережі	Обмежено	Обмежено	-	-	Обмежено	+
Складність геймплею	Середня	Середня	Середня	Середня	Середня	Низька
Дизайн враховує обмеження інклюзивності для людей з вадами зору, слуху	+	-	-	+	-	+
Можливість відправити результати сесії на e-mail	+	+	-	-	+	+
Статистика для наглядача	-	-	-	-	-	+



4

ВИМОГИ ДО ДОДАТКУ

Функціональні вимоги:

1. Запуск та завершення сесії тренування.
2. Застосування вправ для стимулювання різних когнітивних функцій, таких як пам'ять, увага, розуміння та швидкість обробки інформації.
3. Вибір завдання для тренування.
4. Відстеження прогресу тренувань.
5. Налаштування параметрів тренування.
6. Налаштування статистики для наглядача.
7. Можливість відправити результати сесії на e-mail.

Нефункціональні вимоги:

1. Сумісність з операційною системою на базі Android.
2. Робота додатку без доступу в Інтернет.
3. Підтримка налаштувань для персоналізації досвіду користувача.
4. Дизайн застосунку повинен враховувати обмеження інклюзивності для людей з вадами зору, слуху.



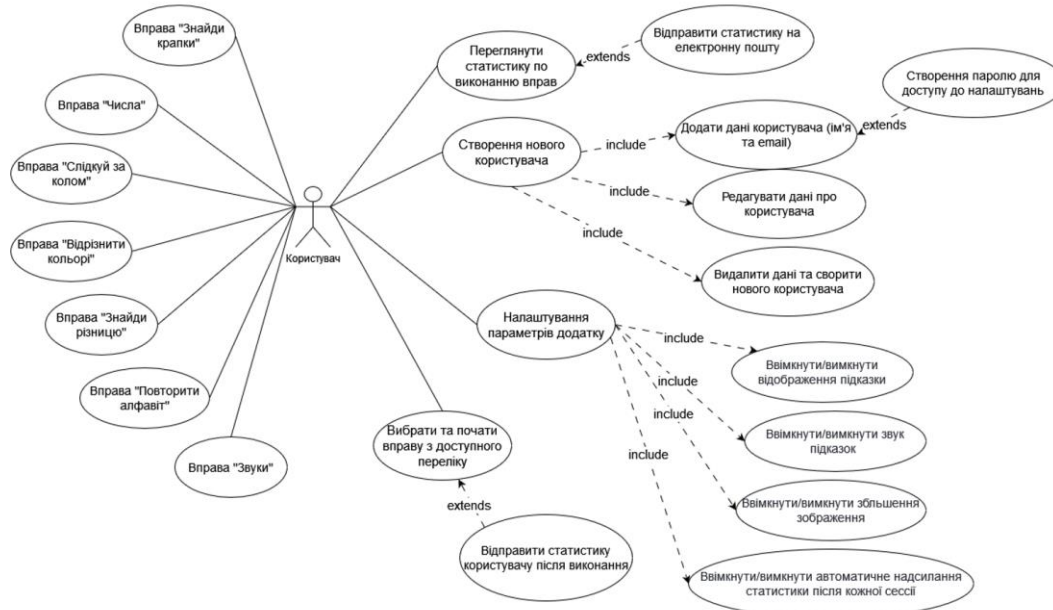
5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



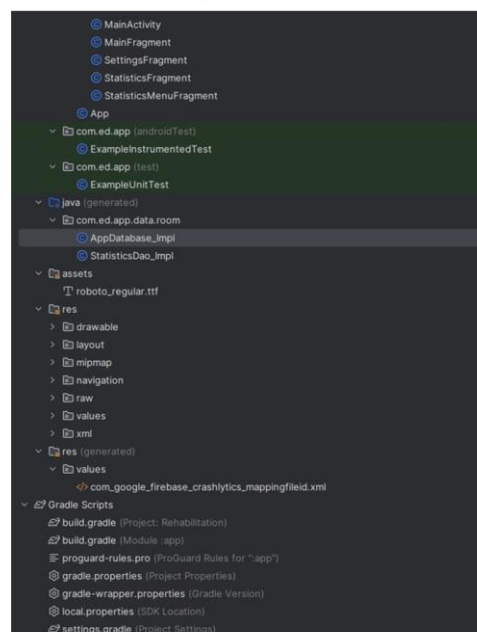
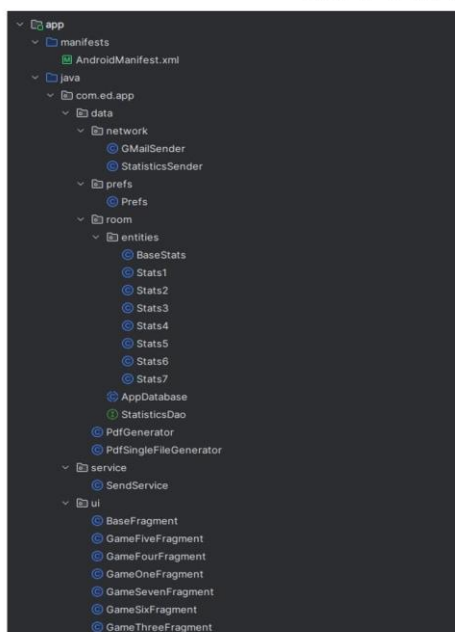
6

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



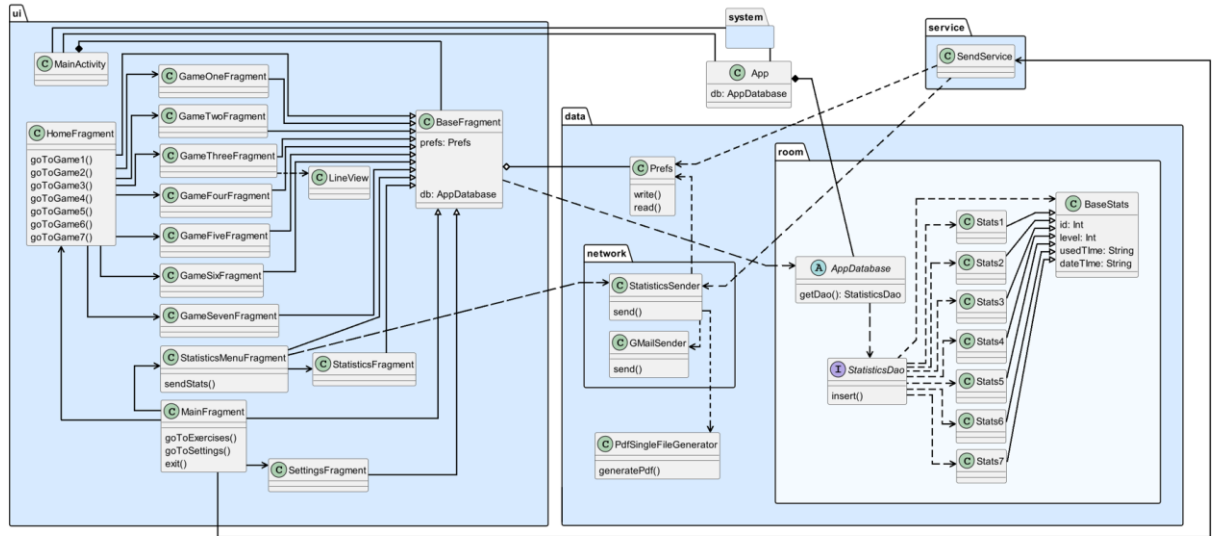
7

СТРУКТУРА ПРОЄКТУ



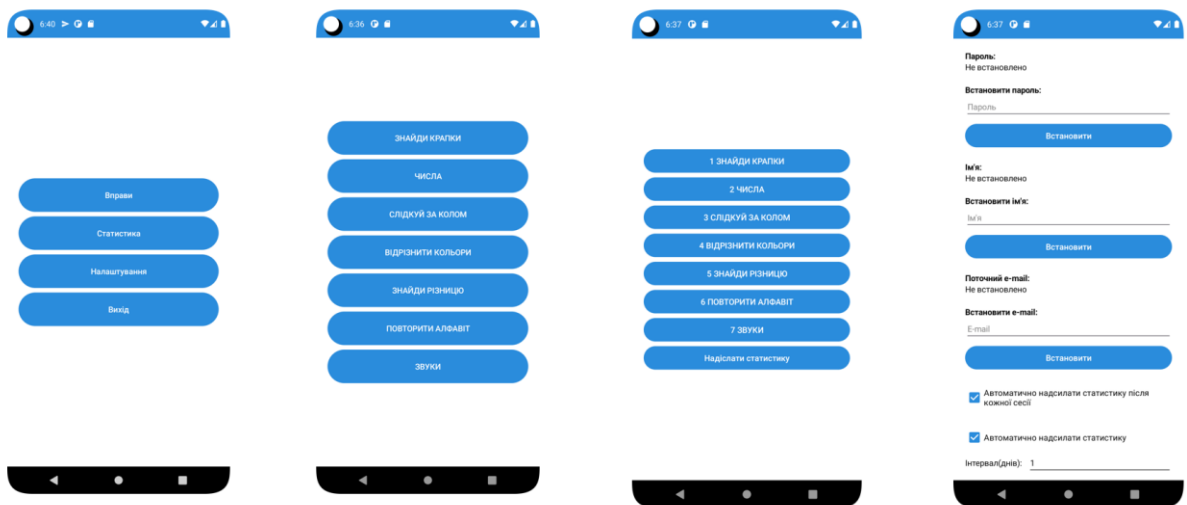
8

ДІАГРАМА КЛАСІВ



9

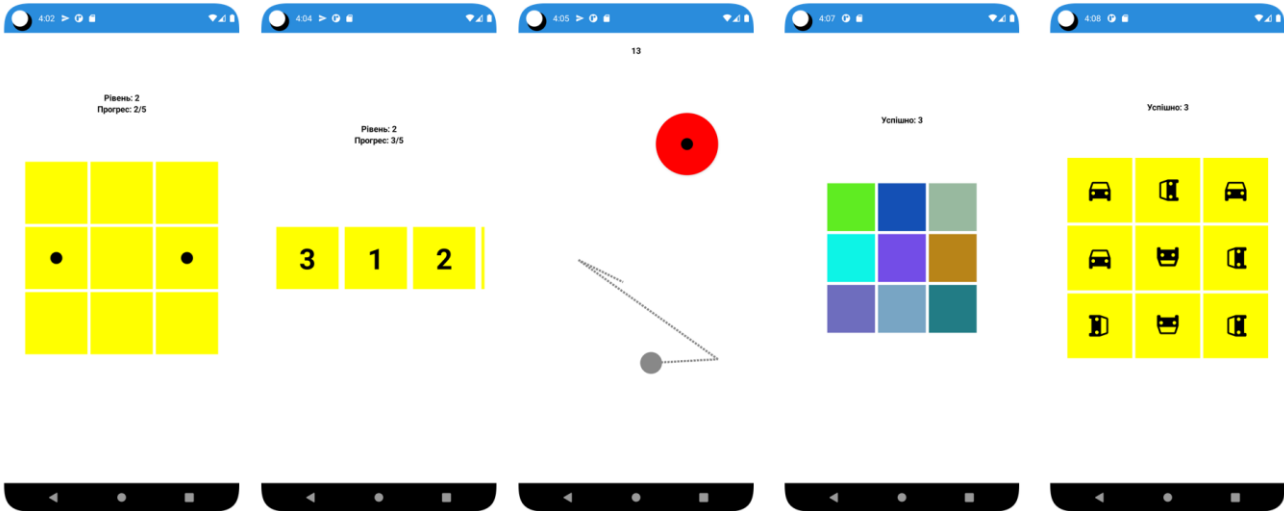
ЕКРАННІ ФОРМИ



Головне меню та його параметри

10

ЕКРАННІ ФОРМИ



Інтерфейс когнітивних вправ

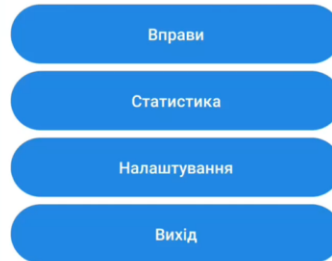
ЕКРАННІ ФОРМИ



Інтерфейс когнітивних вправ

Демонстрація налаштувань та статистики

ДЕМОНСТРАЦІЯ РОБОТИ ЗАСТОСУНКУ



АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Наумов М.С., Золотухіна О.А. Аналіз вимог до мобільного додатку для підтримки відновлення когнітивних функцій: матеріали Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». 24.04.2024, ДУІКТ, м.Київ с. 331
2. Наумов М.С., Золотухіна О.А. Мобільні додатки для стимулювання мозкової активності: технології, підходи до реалізації та практичне застосування: матеріали «V Всеукраїнська науково-практична конференція молодих вчених та здобувачів вищої освіти присвячена Дню науки СУЧАСНА МОЛОДЬ В СВІТІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ». 17.05.2024, с. 175

ВИСНОВКИ

1. Проведено аналіз предметної галузі відновлення когнітивних функцій людини та існуючих застосунків для підтримки цих процесів. Ключові проблеми, які можна виділити, пов'язані з відсутністю в застосунках важливих для процесу реабілітації функцій, зокрема, персоналізація підходів до відновлення, процес відстеження наглядачем результатів тренувань.
2. Спроековано ігрові завдання для підтримки відновлення когнітивних функцій.
3. Реалізовано механіки виконання завдань на уважність, пам'ять, сприйняття та відтворення побаченої інформації.
4. Розроблено застосунок з використанням платформи Android Studio та мови програмування Java.
5. Проведено тестування застосунку.

ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

```
?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/andr
oid"
xmlns:tools="http://schemas.android.com/tools">
<uses-permission
android:name="android.permission.INTERNET"/>
<!-- <uses-permission
android:name="android.permission.READ_EXTERNAL
_STORAGE" />-->
<!-- <uses-permission
android:name="android.permission.WRITE_EXTERNA
L_STORAGE" />-->
<uses-permission
android:name="android.permission.POST_NOTIFICATI
ONS" />
<uses-permission
android:name="android.permission.FOREGROUND_SE
RVICE" />
<application
android:name=".App"
android:allowBackup="true"

android:dataExtractionRules="@xml/data_extraction_rul
es"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="true"
    android:theme="@style/Theme.Rehabilitation"
    tools:targetApi="31">
<service android:name=".service.SendService"
    android:enabled="true"
    />
<activity
    android:name=".ui.MainActivity"
    android:exported="true">
<intent-filter>
<action
android:name="android.intent.action.MAIN" />
<category
android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<provider

android:name="androidx.core.content.FileProvider"

android:authorities="${applicationId}.fileprovider"
    tools:replace="android:authorities"
    android:exported="false"
    android:grantUriPermissions="true">
<meta-data

android:name="android.support.FILE_PROVIDER_PAT
HS"
    android:resource="@xml/file_paths"
```

```
tools:replace="android:resource"
/>
</provider>
</application>
<queries>
<intent>
<action
android:name="android.intent.action.TTS_SERVICE" />
</intent>
</queries>
</manifest>
package com.ed.app.data.network;
import android.content.Context;
import android.os.Handler;
import android.os.Looper;
import android.util.Log;
import android.widget.Toast;
import com.ed.app.App;
import java.io.File;
import java.security.Security;
import java.util.Properties;
import javax.activation.DataHandler;
import javax.activation.DataSource;
import javax.activation.FileDataSource;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Multipart;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;
public class GMailSender extends
javax.mail.Authenticator {
    public void send(
        String recipient,
        String subject,
        String msg,
        Runnable finishCallback,
        File... attachments
    ) {
        final String username =
"mykyta.naumov03@gmail.com";
        final String password = "omms bhgk nbun yyae";
        Properties props = new Properties();
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.host", "smtp.gmail.com");
        props.put("mail.smtp.port", "587");
        Session session = Session.getInstance(props,
            new javax.mail.Authenticator() {
                protected PasswordAuthentication
getPasswordAuthentication() {
                    return new
PasswordAuthentication(username, password);
                }
            }
        );
    }
}
```

```

    });
    try {
        Message message = new MimeMessage(session);

        message.setRecipients(Message.RecipientType.TO,
            InternetAddress.parse(recipient));
        message.setSubject(subject);
        message.setText(msg);
        if (attachments != null) {
            Multipart multipart = new MimeMultipart();

            for (File f : attachments) {
                String path = f.getAbsolutePath();
                String fileName =
                    path.substring(path.lastIndexOf("/") + 1);
                MimeBodyPart messageBodyPart = new
                    MimeBodyPart();
                DataSource source = new FileDataSource(f);
                messageBodyPart.setDataHandler(new
                    DataHandler(source));
                messageBodyPart.setFileName(fileName);
                multipart.addBodyPart(messageBodyPart);
            }
            message.setContent(multipart);
        }
        Transport.send(message);
        finishCallback.run();
    } catch (Exception e) {
        new Handler(Looper.getMainLooper()).post() ->
    {
        Toast.makeText(
            App.instance,
            e.getMessage(),
            Toast.LENGTH_LONG
        ).show();
    });
    throw new RuntimeException(e.getMessage() +
        e.toString());
    }
}
}
package com.ed.app.data.prefs;

import android.content.Context;
import android.content.SharedPreferences;

public class Prefs {
    public static final String HELP = "HELP";
    public static final String HELP_SOUND =
        "HELP_SOUND";
    public static final String ENLARGED_VIEW =
        "ENLARGED_VIEW";
    public static final String EMAIL = "EMAIL";

    public static final String NAME = "NAME";
    public static final String SETTINGS_PASSWORD =
        "SETTINGS_PASSWORD";
    public static final String AUTO_SEND_STATISTICS
        = "AUTO_SEND_STATISTICS";

```

```

    public static final String
        AUTO_SEND_STATISTICS_PERIOD =
            "AUTO_SEND_STATISTICS_PERIOD";
    public static final String
        AUTO_SEND_STATISTICS_LAST_DATE =
            "AUTO_SEND_STATISTICS_LAST_DATE";
    public static final String
        AUTO_SEND_STATISTICS_AFTER_SESSION =
            "AUTO_SEND_STATISTICS_AFTER_SESSION";
    private final SharedPreferences prefs;
    public Prefs(Context context) {
        prefs = context.getSharedPreferences("prefs",
            Context.MODE_PRIVATE);
    }

    public void write(
        String key,
        boolean value
    ) {
        SharedPreferences.Editor editor = prefs.edit();
        editor.putBoolean(key, value);
        editor.apply();
    }

    public void write(
        String key,
        String value
    ) {
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString(key, value);
        editor.apply();
    }

    public void write(
        String key,
        int value
    ) {
        SharedPreferences.Editor editor = prefs.edit();
        editor.putInt(key, value);
        editor.apply();
    }

    public void write(
        String key,
        long value
    ) {
        SharedPreferences.Editor editor = prefs.edit();
        editor.putLong(key, value);
        editor.apply();
    }

    public boolean readBoolean(
        String key,
        boolean defaultValue
    ) {
        return prefs.getBoolean(key, defaultValue);
    }

    public String readString(
        String key,
        String defaultValue
    ) {

```

```

        return prefs.getString(key, defaultValue);
    }

    public int readInt(
        String key,
        int defaultValue
    ) {
        return prefs.getInt(key, defaultValue);
    }

    public long readLong(
        String key,
        int defaultValue
    ) {
        return prefs.getLong(key, defaultValue);
    }
}
package com.ed.app.data;

import android.content.Context;

import com.ed.app.data.room.entities.BaseStats;
import com.itextpdf.io.font.PdfEncodings;
import com.itextpdf.kernel.font.PdfFontFactory;
import com.itextpdf.layout.Document;
import com.itextpdf.layout.element.Paragraph;
import com.itextpdf.layout.element.Table;
import
com.itextpdf.layout.property.HorizontalAlignment;
import com.itextpdf.layout.property.TextAlignment;
import com.itextpdf.layout.property.VerticalAlignment;
import com.itextpdf.kernel.pdf.PdfDocument;
import com.itextpdf.kernel.pdf.PdfWriter;
import java.io.File;
import java.io.IOException;
import java.util.List;

public class PdfGenerator {
    public static File generatePdf(
        Context context,
        String fileName,
        String title,
        String subtitle,
        List<BaseStats> baseStats
    ) {
        File dir = context.getFilesDir();
        File file = new File(dir, fileName + ".pdf");

        try {
            PdfWriter writer = new PdfWriter(file);
            PdfDocument pdfDoc = new
PdfDocument(writer);
            Document doc = new Document(pdfDoc);

            doc.setFont(PdfFontFactory.createFont("assets/roboto_re
gular.ttf", PdfEncodings.IDENTITY_H));

            Table table = new Table(3);

            table.setHorizontalAlignment(HorizontalAlignment.LEF
T);

```

```

            table.setVerticalAlignment(VerticalAlignment.MIDDLE
);
            table.addCell("Витрачено");
            table.addCell("Рівень");
            table.addCell("Дата");

            for (BaseStats s : baseStats) {
                table.addCell(s.usedTime);
                table.addCell(String.valueOf(s.level));
                table.addCell(s.dateTime);
            }

            doc.add(new Paragraph(title).setBold());
            doc.add(new Paragraph(subtitle));
            doc.add(table);
            doc.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

        return file;
    }
}
package com.ed.app.data;

import android.content.Context;

import com.ed.app.data.room.entities.BaseStats;
import com.itextpdf.io.font.PdfEncodings;
import com.itextpdf.kernel.font.PdfFontFactory;
import com.itextpdf.kernel.pdf.PdfDocument;
import com.itextpdf.kernel.pdf.PdfWriter;
import com.itextpdf.layout.Document;
import com.itextpdf.layout.element.Paragraph;
import com.itextpdf.layout.element.Table;
import
com.itextpdf.layout.property.HorizontalAlignment;
import com.itextpdf.layout.property.VerticalAlignment;

import java.io.File;
import java.io.IOException;
import java.util.List;

public class PdfSingleFileGenerator {
    public static File generatePdf(
        Context context,
        String fileName,
        String title,
        String subtitle,
        List<BaseStats>... baseStatsArray
    ) {
        File dir = context.getFilesDir();
        File file = new File(dir, fileName + ".pdf");

        try {
            PdfWriter writer = new PdfWriter(file);
            PdfDocument pdfDoc = new
PdfDocument(writer);
            Document doc = new Document(pdfDoc);

```

```

doc.setFont(PdfFontFactory.createFont("assets/roboto_regular.ttf", PdfEncodings.IDENTITY_H));

doc.add(new Paragraph(title).setBold());
doc.add(new Paragraph(subtitle));

for (int i = 0; i < baseStatsArray.length; i++) {

    List<BaseStats> baseStatsList =
baseStatsArray[i];
    doc.add(new Paragraph("Вправа " + (i + 1)));

    Table table = new Table(3);

table.setHorizontalAlignment(HorizontalAlignment.LEFT);

table.setVerticalAlignment(VerticalAlignment.MIDDLE);

    table.addCell("Витрачено");
    table.addCell("Рівень");
    table.addCell("Дата");

    for (BaseStats s : baseStatsList) {
        table.addCell(s.usedTime);
        table.addCell(String.valueOf(s.level));
        table.addCell(s.dateTime);
    }

    doc.add(table);
}

doc.close();
} catch (IOException e) {
    e.printStackTrace();
}

return file;
}
}
package com.ed.app.service;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.util.Log;

import androidx.annotation.Nullable;
import androidx.core.app.NotificationCompat;

import com.ed.app.R;
import com.ed.app.data.network.StatisticsSender;
import com.ed.app.data.prefs.Prefs;

```

```

public class SendService extends Service {
    private static final String CHANNEL_ID =
"REHABILITATION_CHANNEL";

    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int
startId) {
        createNotificationChannel();
        Notification notification = createNotification();
        startForeground(100, notification);

        Prefs prefs = new Prefs(this);
        String email = prefs.readString(Prefs.EMAIL, null);

        if (email != null) {
            send();
        } else {
            Notification errorNotification = new
NotificationCompat.Builder(this, CHANNEL_ID)
                .setTitle("Rehabilitation")
                .setText("Вкажіть e-mail для
відправки статистики")
                .setSmallIcon(R.drawable.baseline_send_24)
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .build();
            NotificationManager manager =
getSystemService(NotificationManager.class);
            manager.notify(0, errorNotification);

            stopForeground(true);
            stopSelf();
        }

        return START_NOT_STICKY;
    }

    private void createNotificationChannel() {
        NotificationChannel serviceChannel = new
NotificationChannel(
            CHANNEL_ID,
            "Rehabilitation App Service Channel",
            NotificationManager.IMPORTANCE_HIGH
        );

        NotificationManager manager =
getSystemService(NotificationManager.class);

manager.createNotificationChannel(serviceChannel);
    }

    private Notification createNotification() {
        NotificationCompat.Builder builder = new
NotificationCompat.Builder(this,
CHANNEL_ID)

```

```

        .setTitle("Rehabilitation")
        .setSmallIcon(R.drawable.baseline_send_24);
        return builder.build();
    }

    private void send() {
        Prefs prefs = new Prefs(this);

        new StatisticsSender().send(
            this,
            () -> {
                Log.d("MYLOGCAT", "sent");
            }
        );
    }

    private void showSuccessNotification() {
        Notification notification = new
        NotificationCompat.Builder(this, CHANNEL_ID)
            .setSmallIcon(R.drawable.baseline_send_24)
            .setPriority(NotificationCompat.PRIORITY_HIGH)
            .build();
        NotificationManager manager =
        getSystemService(NotificationManager.class);
        manager.notify(0, notification);
    }
}

package com.ed.app;
import android.app.Application;
import androidx.room.Room;
import com.ed.app.data.room.AppDatabase;
public class App extends Application {
    public static App instance;
    public AppDatabase db;
    @Override
    public void onCreate() {
        super.onCreate();
        instance = this;
        db = Room.databaseBuilder(
            this,
            AppDatabase.class,
            "database"
        )
        .allowMainThreadQueries()
        .build();
    }
}

```

```

}
package com.ed.app;

import android.content.Context;

import
androidx.test.platform.app.InstrumentationRegistry;
import androidx.test.ext.junit.runners.AndroidJUnit4;

import org.junit.Test;
import org.junit.runner.RunWith;

import static org.junit.Assert.*;

/**
 * Instrumented test, which will execute on an Android
 * device.
 *
 * @see <a
 * href="http://d.android.com/tools/testing">Testing
 * documentation</a>
 */
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
    @Test
    public void useAppContext() {
        // Context of the app under test.
        Context appContext =
        InstrumentationRegistry.getInstrumentation().getTargetContext();
        assertEquals("com.ed.app",
        appContext.getPackageName());
    }
}

```