

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____ Олена ОЛЬХОВСЬКА
(підпис)

«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему
**РОЗРОБКА НАВЧАЛЬНОГО АНДРОЇД ЗАСТОСУНКУ З ТЕМИ
«СОРТУВАННЯ ВСТАВКАМИ» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО
КУРСУ «АЛГОРИТМИ І СТРУКТУРИ ДАНИХ»**

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня магістра

Виконавець роботи Янко Олексій Миколайович
_____ «___» _____ 2023 р.
(підпис)

Науковий керівник к.пед.н., доцент, Кошова Оксана Петрівна
_____ «___» _____ 2023 р.
(підпис)

Рецензент

ПОЛТАВА 2023 р.

ЗАТВЕРДЖУЮЗавідувач кафедри _____ Олена ОЛЬХОВСЬКА
(підпис)

« ____ » _____ 202__ р.

**ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК
ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ****на тему «Розробка навчального андроїд застосунку з теми «Сортування вставками» дистанційного навчального курсу «Алгоритми і структури даних»»**

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня магістр

Прізвище, ім'я, по батькові Янко Олексій МиколайовичЗатверджена наказом ректора № 21-Н від «25» січня 2023 р.

Термін подання студентом роботи « ____ » _____ 2023 р.

Вихідні дані до кваліфікаційної роботи: статті та документації з теми розробки навчальних застосунків та сервісів для тестування.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП**РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ****РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ**

2.1. Навчальний додаток «Quizlet»

2.2. Навчальний додаток «Plant Parent»

2.3. Навчальний додаток «SoloLearn»

2.4. Навчальний додаток «Enki»

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Опис проектних рішень, інструментів та підходів до розробки для андроїд додатку.

3.2. Опис методології створення програмного забезпечення

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис та побудова діаграми прецедентів роботи з системою.

4.2. Інструкція з використання додатка

ВИСНОВКИ**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ****ДОДАТОК А**Перелік графічного матеріалу: 1 аркуш блок-схем, 24 ілюстрації.

Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Кошова О.П.		
Інформаційний огляд	Кошова О.П.		
Теоретична частина	Кошова О.П.		
Практична частина	Кошова О.П.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
Вступ		
Вивчення методичних рекомендацій та стандартів та звіт керівнику		
Постановка задачі		
Інформаційний огляд джерел бібліотек та інтернету		
Теоретична частина		
Практична частина		
Закінчення оформлення		
Доповідь студента на кафедрі		
Доробка (за необхідністю), рецензування		

Дата видачі завдання «__» _____ 2023 р.

Здобувач вищої освіти _____ Янко Олексій Миколайович
(підпис)Науковий керівник _____ к.пед.н., доц. Кошова О.П.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «__» _____ 2024 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

Затверджую

Зав. кафедри _____
 к.ф.-м.н. Олена ОЛЬХОВСЬКА
 « ____ » _____ 2023 р.

Погоджено

Науковий керівник _____
 к.пед.н., Оксана КОШОВА
 « ____ » _____ 2023 р.

План

дипломного проекту з фаху
 спеціальності 122 Комп'ютерні науки
 освітня програма 122 Комп'ютерні науки
 на тему «Розробка навчального андроїд застосунку з теми «Сортування вставками» дистанційного навчального курсу «Алгоритми і структури даних»»
 Прізвище, ім'я, по батькові Янко Олексій Миколайович

ВСТУП**РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ****РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ****2.1. Навчальний додаток «Quizlet»****2.2. Навчальний додаток «Plant Parent»****2.3. Навчальний додаток «SoloLearn»****2.4. Навчальний додаток «Enki»****РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА****3.1. Опис проектних рішень, інструментів та підходів до розробки для андроїд додатку.****3.2. Опис методології створення програмного забезпечення****РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА****4.1. Опис та побудова діаграми прецедентів роботи з системою.****4.2. Інструкція з використання додатка****ВИСНОВКИ****СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ****ДОДАТОК А**

Здобувач вищої освіти _____ О.М. Янко

(підпис)

« ____ » _____ 2023 р.

РЕФЕРАТ

Записка: 49 сторінок, 24 рисунків, 1 додаток, 14 літературних джерел.

Мета дипломної роботи розробка мобільного навчального додатку для Android на тему «Сортування вставками».

Об'єкт розробки - мобільний навчальний додаток для Android.

Методи дослідження – використання сучасних підходів для написання якісних додатків за принципами чистої архітектури. IDE ANDROID STUDIO для збірки проекту, Графічні інструменти Photoshop та Figma для розроблення інтерфейсу та мова програмування Kotlin спільно з фреймворком Android.

Здійснено аналіз застосунків для навчання, включаючи їх сильні та слабкі сторони. Опрацьовані основні проектні рішення, вибрані інструменти та методики для розробки Android-додатку у сфері дистанційної освіти. Встановлено конкретну методологію для розробки програмного забезпечення. Складено діаграму прецедентів для кращого розуміння взаємодії користувачів із системою. Розроблено інтерфейс додатку та підготовлено користувацьку інструкцію.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	9
РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ.....	12
2.1. Навчальний додаток «Quizlet»	12
2.2. Навчальний додаток «Plant Parent».....	13
2.3. Навчальний додаток «SoloLearn».....	14
2.4. Навчальний додаток «Enki».....	16
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА	18
3.1. Опис проектних рішень, інструментів та підходів до розробки для андроїд додатку.	18
3.2. Опис методології створення програмного забезпечення.....	25
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА	28
4.1. Опис та побудова діаграми прецедентів роботи з системою.	28
4.2. Інструкція з використання додатка.....	28
ВИСНОВКИ.....	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	37
ДОДАТОК А.....	39

ВСТУП

Актуальність. В епоху стрімкого технологічного розвитку, новітні тенденції в області Інтернет-технологій створюють унікальні умови для модернізації освітнього процесу. Інтернет є масштабною платформою для студентів, яка забезпечує доступ до безмежної кількості інформації. У цьому контексті, актуальним стає завдання адаптації освітньої системи до сучасних вимог, з метою підготовки фахівців, які будуть конкурентоспроможні в сучасному світі. Однією з можливих відповідей на це завдання є розробка мобільних навчальних додатків для Android. У порівнянні з традиційними формами навчання, мобільні додатки відкривають нові горизонти завдяки використанню передових телекомунікаційних засобів та Інтернет-ресурсів. Ці додатки можуть ефективно служити для навчання не тільки студентів, але і осіб, які вже працюють, без необхідності відриву від основної діяльності.

Основним плюсом мобільних навчальних додатків, в тому числі для платформи Android, є здатність надавати освітні послуги незалежно від географічного розташування користувача, допоки є доступ до Інтернету. Це надає студентам гнучкість у виборі оптимального для них формату навчання, дозволяючи адаптувати процес освіти до власного ритму життя та специфічних потреб. Додатково, мобільне навчання може виявитися економічно вигідним, оскільки уникнення необхідності фізичної присутності в навчальному закладі видаляє витрати на проживання та харчування. Отже, мобільні навчальні додатки для Android мають ряд переваг, таких як:

- дозволяє користувачам здобувати знання, не виходячи з дому;
- забезпечує доступ до електронних навчальних ресурсів і відео-матеріалів;
- можливість використання вбудованих комунікаційних засобів для взаємодії з іншими користувачами та експертами;
- доступність навчання для осіб з обмеженнями в русі чи інвалідністю;
- можливість зекономити на витратах, таких як проживання та проїзд;

- використання сучасних технологій і методологій для навчання;
- можливість суміщення роботи та навчання.

Мета даної дипломної роботи полягає у розробці мобільного навчального додатку для Android на тему «Сортування вставками».

Об'єктом розробки є мобільний навчальний додаток для Android.

Предметом розробки є програмне забезпечення для Android-додатку та тему «Сортування вставками», розроблене за допомогою Java або Kotlin, та використання Android SDK для створення користувацького інтерфейсу.

Перелік методів включає проектування мобільних навчальних додатків, зокрема методики UX/UI дизайну.

Щодо технічних інструментів, використано: Android Studio для розробки, Git для контролю версій, а також бібліотеки та фреймворки, специфічні для Android розробки.

Пояснювальна записка включає чотири основних розділи: формулювання завдань, аналіз існуючих аналогів, теоретичні аспекти та практична реалізація. Структура роботи розрахована на логічне викладення матеріалу та повне розкриття теми дослідження.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Мета додатку

Сучасний світ технологій вимагає все більш ефективних методів навчання, особливо в області програмування та комп'ютерних наук. Відповідаючи цим вимогам, наша мета полягає у створенні інтерактивного Android-додатку, який стане незамінним помічником для тих, хто бажає вивчити алгоритм сортування вставками.

Ключовою особливістю цього додатку є його підхід до навчання. Він не просто подає суху теоретичну інформацію, але об'єднує теоретичні матеріали з практичними вправами. Це дозволяє користувачам не тільки глибше розуміти концепції алгоритму, але й набувати практичних навичок його застосування.

Таким чином, ми прагнемо надати користувачам зручний інструмент, який не лише розкриє перед ними тонкощі алгоритмічних методів, але й допоможе зміцнити знання, отримані на практиці. Наша мета - робити навчання доступним, ефективним та захоплюючим.

Основні функції додатку

1. Теоретичний матеріал.

Цей розділ Android-додатку служить фундаментом для глибокого розуміння алгоритму сортування вставками. Він включає в себе декілька частин, кожна з яких спрямована на вивчення конкретного аспекту цього алгоритму. Структурованість текстового матеріалу забезпечує послідовне і послідовне засвоєння інформації, дозволяючи користувачеві послідовно рухатися від базових концепцій до більш складних аспектів алгоритму.

Для забезпечення чіткості і наглядності, основні правила та ключові моменти алгоритму супроводжуються конкретними прикладами. Це не тільки

спрощує процес засвоєння матеріалу, але й дозволяє користувачу побачити практичне застосування алгоритму в реальних умовах.

Специфіка роботи на платформі Android вимагає особливого підходу до представлення текстового контенту. Використання форматування HTML в елементах XML дозволяє досягти вищої лаконічності тексту, зберігаючи його читабельність та атрактивність. Такий підхід забезпечує максимальну лаконічність тексту, не перенавантажуючи візуальний інтерфейс додатку, що сприяє комфортному вивченню матеріалу.

2. Практичні вправи

У навчальному процесі важливо не лише засвоювати теоретичний матеріал, але й здатися його застосовувати на практиці. З цією метою наш додаток пропонує ряд практичних вправ у форматі тестів. Ці вправи розроблені таким чином, щоб користувач міг оцінити свій рівень розуміння алгоритму сортування вставками та виявити можливі прогалини у знаннях.

Основна особливість цих тестів полягає у тому, що на кожне питання пропонуються три варіанти відповідей. Користувачу потрібно обрати одну з них, виходячи зі своєї логіки та знань з теми. Такий підхід спонукає до глибшого рефлексування над матеріалом і стимулює критичне мислення.

Таким чином, практичні вправи стають невід'ємною частиною навчального процесу, допомагаючи користувачу не лише консолідувати свої знання, але й активно застосовувати їх у практичній діяльності.

3. Статистика

Для мотивації та кращого розуміння своїх досягнень користувачам важливо мати можливість відстежувати свій прогрес. Саме тому на головній сторінці додатку розміщено розділ статистики, який допомагає користувачу бачити свої результати як за теоретичною частиною, так і за практичними вправами.

Візуально статистика розбита на дві ключові частини:

1. **Основна статистика.** В ній користувач може побачити свої загальні досягнення по теоретичному матеріалу та практичним вправам. Це дає змогу оцінити, наскільки ефективно відбувається вивчення матеріалу та в яких аспектах можливо потрібно приділити більше часу.

2. **Графік правильних відповідей.** Він відображає динаміку правильних відповідей користувача по дням тижня. Такий підхід дозволяє користувачу аналізувати свої найбільш продуктивні дні та, можливо, регулювати графік навчання з урахуванням отриманих даних.

Сумуючи вищевказане, можна стверджувати, що статистика в додатку не лише відображає рівень досягнень користувача, але й служить важливим інструментом самоаналізу та самовдосконалення, допомагаючи користувачу максимально ефективно використовувати свій час і ресурси.

Структура додатку

1. **Головна сторінка.** Доступ до основних розділів додатку та відображення статистики користувача.

2. **Розділ теорії.** Докладний опис алгоритму, приклади код та інші матеріали, які допоможуть користувачеві вивчити алгоритм.

3. **Модуль тестів.** Набір завдань, спрямованих на перевірку розуміння алгоритму користувачем.

4. **Модуль статистики.** Візуалізація даних про активність та успішність користувача в додатку.

З урахуванням зазначених вимог та особливостей, додаток має стати ефективним інструментом для вивчення та поглиблення знань з алгоритму сортування вставками.

РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ

2.1. Навчальний додаток «Quizlet»

«Quizlet» – це інтерактивний сервіс для створення навчальних карток, який допомагає користувачам засвоїти новий матеріал за допомогою віртуальних карток. Сервіс використовується студентами, вчителями та навчальними закладами по всьому світу для підготовки до тестів, контрольних, іспитів або просто для закріплення матеріалу (Рисунок 2.1).

«Quizlet» надає можливість створювати власні набори карток або використовувати готові набори, які створили інші користувачі. Крім того, сервіс пропонує різноманітні режими навчання: тести, ігри, вивчення та інше. Основна мета – зробити процес навчання цікавим та ефективним.



Рисунок 2.1. Лого інтерактивного сервісу «Quizlet»

Основні можливості «Quizlet»:

- створення власних наборів карток;
- можливість додавати зображення та аудіо до карток;
- різні режими навчання: вивчення, написання, тестування, гра та ін.;
- можливість співпрацювати з іншими користувачами, спільно створюючи набори карток;
- додаток доступний для Android та iOS, що дозволяє вивчати матеріал будь-де та будь-коли;

– автоматизоване відстеження прогресу користувача.

«Quizlet» ідеально підходить для студентів, які шукають зручний та інтерактивний спосіб підготовки до тестів або іспитів. Платформа також допомагає вчителям створювати навчальні матеріали для своїх студентів та моніторити їхній прогрес.

Додаткові переваги:

- інтуїтивно зрозумілий інтерфейс;
- велика спільнота користувачів, яка ділиться своїми наборами карток;
- можливість синхронізації прогресу між різними пристроями;
- безкоштовна базова версія з можливістю оновлення до преміум для додаткових функцій.

2.2. Навчальний додаток «Plant Parent»

«Plant Parent» – це навчальний додаток, який призначений для допомоги користувачам в догляді за рослинами. Завдяки цьому додатку користувачі можуть отримати докладну інформацію про різні види рослин, їх потреби в поливі, світлі, ґрунті та інших аспектах догляду.



Рисунок 2.2. Лого додатку «Plant Parent» (рисунок 2.2).

Особливості «Plant Parent» включають:

- базу даних з описами та рекомендаціями по догляді для сотень видів рослин;

- інструменти для створення розкладу поливу, який адаптується до конкретних умов користувача;
- можливість ведення щоденника своїх рослин, з можливістю додавати фотографії та нагадування;
- форуми для спілкування та обговорення питань між любителями рослин з різних країн світу;
- інтерактивні квізи та завдання для перевірки своїх знань та покращення навичок у догляді за рослинами;
- можливість синхронізації додатка з іншими пристроями та програмами для додаткового комфорту.

В «Plant Parent» також є розділ «Моя бібліотека», де користувачі можуть зберігати свої улюблені статті, відео та інші ресурси, пов'язані з доглядом за рослинами.

Переваги додатка:

- наявність великої кількості корисної інформації про рослини та догляд за ними;
- зручний інтерфейс та простота використання;
- можливість обмінюватися досвідом з іншими користувачами;
- регулярні оновлення та додавання нового контенту;
- адаптація під потреби користувача і його умови утримання рослин.

2.3. Навчальний додаток «SoloLearn»

«SoloLearn» – це навчальний додаток, призначений для вивчення програмування через інтерактивні курси, завдання та виклики. Він забезпечує широкий спектр мов програмування від основ до складних тем для розширеного вивчення. За допомогою «SoloLearn» користувачі можуть не тільки вивчити основи програмування, але й отримати досвід роботи з різними технологіями та інструментами. (Рисунок 2.3).



Рисунок 2.3. Лого навчального додатку «SoloLearn»

Основні можливості «SoloLearn» включають:

- широкий вибір курсів з різних мов програмування, таких як Python, Java, C++, і багато інших;
- інтерактивні завдання та виклики, які допомагають користувачам вдосконалити свої навички;
- спільнота з мільйонами користувачів, де можна отримати відгуки, поради або взяти участь в обговореннях на форумах;
- режим конкуренції, де користувачі можуть змагатися в різних програмувальних викликах;
- офлайн-доступ до матеріалів для зручності навчання без підключення до Інтернету.

Використовуючи «SoloLearn», користувачі можуть слідкувати за своїм прогресом, отримувати нагороди за досягнення та взаємодіяти з іншими користувачами, обмінюючись своїм досвідом та знаннями.

Переваги додатка:

- зручний інтерфейс та структуровані курси;
- можливість вивчення програмування в будь-якому місці та в будь-який час;
- активна спільнота, яка готова допомогти з будь-якими питаннями;
- постійні оновлення курсів, що відповідають актуальним трендам в галузі ІТ;
- безкоштовний доступ до більшості матеріалів з можливістю покупки

преміум-версії для додаткових можливостей.

2.4. Навчальний додаток «Enki»

«Enki» – це навчальний додаток, розроблений для тих, хто бажає вдосконалити свої навички в програмуванні, дизайні, засадах даних, а також в інших сферах ІТ. За допомогою коротких щоденних уроків та викликів, «Enki» допомагає користувачам поглиблювати свої знання та залишатися в курсі найсвіжіших трендів та технологій. (Рисунок 2.4)



Рисунок 2.4. Навчальний додаток «Enki»

Особливості «Enki» включають:

- широкий спектр тем, включаючи мови програмування, засади роботи з даними, інструменти розробки, основи кібербезпеки тощо;
- щоденні уроки та завдання, розроблені професіоналами в галузі, щоб

забезпечити ефективне навчання;

- інтерактивні виклики, що допомагають перевірити та закріпити знання;
- персоналізований підхід до навчання, де користувач може обрати свій власний шлях вивчення;
- можливість слідкувати за своїм прогресом та отримувати рекомендації щодо подальшого навчання.

Завдяки «Епкі», користувачі можуть вивчати нові теми або поглиблювати вже існуючі знання, отримуючи постійні стимули для розвитку та удосконалення своїх професійних навичок.

Переваги додатка:

- адаптивний підхід до навчання, що враховує індивідуальний досвід та потреби користувача;
- короткі, але ефективні уроки, які легко вписуються в щоденний розпорядок;
- акцент на практичних навичках та реальних завданнях;
- постійне оновлення матеріалів, щоб відповідати актуальним трендам в ІТ;
- дружній та інтуїтивно зрозумілий інтерфейс.

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Опис проектних рішень, інструментів та підходів до розробки для андроїд додатку.

Для розробки цього навчального андроїд-додатку було використано наступні технології:

- Програма для розробки концептів дизайну Photoshop;
- Набір для розробки програмного забезпечення Android SDK;
- Реляційна база даних sqlite для збереження інформації локально;
- Мова програмування Kotlin;
- Середовище розробки Android Studio.
- Архітектура MVVM для мобільного додатка

Програма для розробки концептів дизайну Photoshop.

Adobe Photoshop – це лідер серед графічних редакторів, що вже довгі роки залишається вибором номер один серед дизайнерів, фотографів та художників у всьому світі. Відкриваючи Photoshop, користувач стикається з потужним інструментарієм, який дозволяє втілювати найсміливіші креативні ідеї. (Рисунок 3.1).



Рисунок 3.1. Логотип Photoshop.

З часів свого заснування, Photoshop неперервно розвивається, набуваючи все нових функцій. Від базової корекції зображень до складних 3D-моделювань і анімації – можливості програми майже безмежні. Шари, маски, фільтри, кисті – це лише невеликий перелік з того многообрання, яке пропонує Photoshop.

Ще однією важливою особливістю є тісна інтеграція з іншими продуктами від Adobe, такими як Illustrator, Premiere Pro чи After Effects. Це дозволяє професіоналам ефективно комбінувати різні напрямки роботи, отримуючи в результаті гармонійний і завершений продукт.

Завдяки масштабній спільноті користувачів по всьому світу, існує величезна кількість навчальних матеріалів, плагінів та шаблонів, що робить процес освоєння та роботи в Photoshop ще більш комфортним і продуктивним.

Отже, Adobe Photoshop – це не просто програма, це ціла універсум для творчості, де кожен може втілити свої найсміливіші візії. Незалежно від рівня підготовки, Photoshop пропонує інструменти, які задовольняють потреби як новачків, так і справжніх майстрів графічного дизайну [2] .

Набір для розробки програмного забезпечення Android SDK.

Android SDK (Software Development Kit) – це фундаментальний інструментарій для розробників мобільних додатків, який спрямований на створення програм для операційної системи Android. Розроблений і підтримуваний компанією Google, Android SDK включає в себе все необхідне для проектування, розробки та тестування додатків. (Рисунок 3.2).

Android SDK відкриває перед розробниками багатий світ можливостей завдяки своїм API, бібліотекам, емуляторам та іншим корисним інструментам. Однією з основних особливостей є можливість вибору рівня API для певної версії Android, дозволяючи таким чином оптимізувати додатки для різних пристроїв і версій системи.



Рисунок 3.2. Логотип Android SDK.

Android SDK також надає засоби для віртуалізації мобільних пристроїв. За допомогою емулятора розробники можуть тестувати свої додатки в різних умовах і конфігураціях пристроїв, не використовуючи фізичний апарат.

Щодо інтеграції з іншими технологіями, Android SDK дозволяє взаємодіяти з різними сервісами Google, такими як Google Maps, Firebase та інші. Це значно розширює можливості додатків, роблячи їх більш функціональними та інтерактивними.

Спільнота розробників Android є однією з найбільших у світі. Завдяки цьому існує велика кількість ресурсів, плагінів, бібліотек та навчальних матеріалів, які спрощують і прискорюють процес розробки.

Отже, Android SDK – це відкрите вікно в світ мобільної розробки для найпопулярнішої операційної системи в світі. Він надає розробникам комплексний набір інструментів, щоб створювати інноваційні, надійні та ефективні додатки для мільярдів користувачів по всьому світі [6, 7, 13] .

Реляційна база даних SQLite для локального зберігання інформації.

SQLite – це легка, компактна та надійна реляційна система управління базами даних, яка ідеально підходить для збереження даних на локальних пристроях. Відмінною особливістю SQLite є те, що вона не вимагає сервера для роботи, а всі її компоненти знаходяться безпосередньо в тілі головного додатка.

(Рисунок 3.3).



Рисунок 3.3. Логотип SQLite.

Однією з переваг SQLite є те, що вона дозволяє зберігати дані у структурованому вигляді без необхідності підтримки окремого сервера або складного налаштування. Це робить її ідеальним вибором для мобільних додатків, настільних програм та додатків IoT, де потрібен швидкий доступ до даних без втрати продуктивності.

SQLite підтримує усі основні операції SQL, що дозволяє розробникам працювати з даними так, як вони звикли, користуючись іншими реляційними СУБД. Хоча SQLite має обмежений обсяг функціональності порівняно з більшими серверними рішеннями, вона все ще надає великий спектр можливостей для роботи з реляційними даними.

Масштабування і безпека також є важливими аспектами SQLite. База даних здатна обробляти великий обсяг інформації, пропонуючи при цьому швидкість і стабільність. Щодо безпеки, SQLite пропонує різні рівні захисту, щоб гарантувати конфіденційність і цілісність даних.

Отже, SQLite є простою, але потужною системою управління базами даних, яка ідеально підходить для локального зберігання інформації. З її допомогою розробники можуть створювати надійні, швидкі та безпечні додатки, що працюють на різних платформах і пристроях [5] .

Мова програмування Kotlin.

Kotlin – це сучасна мова програмування, розроблена компанією JetBrains, відома своєю чистотою, безпекою та інноваційністю. Вона була створена з метою виправлення деяких обмежень та недоліків, які існують у інших мовах, зокрема в Java, і водночас забезпечити повну сумісність з існуючим кодом на Java (Рисунок 3.4).



Рисунок 3.4. Логотип Kotlin.

Для багатьох розробників, перехід на Kotlin став диханням свіжого повітря. Відома своєю концизністю, мова дозволяє писати менше коду, отримуючи ті ж самі результати. Так, наприклад, в Kotlin відсутні такі широко відомі проблеми Java, як null pointer exceptions, завдяки вбудованим механізмам безпеки.

Однією з ключових особливостей Kotlin є підтримка функціонального програмування. Це не лише модний тренд, але й потужний інструмент, який дозволяє розробникам писати більш гнучкий, модульний та перевірений код. Лямбди, розширюючі функції та інші можливості Kotlin надають розробнику широкий спектр інструментів для створення чистого та ефективного коду.

Kotlin також пропонує корутини, які стають альтернативою традиційному підходу до асинхронного програмування. Вони дозволяють розробникам писати лінійний, інтуїтивно зрозумілий код, не втрачаючи при цьому переваг асинхронності.

Мова активно підтримується й розвивається компанією JetBrains, а також має сильну та активну спільноту. Це означає постійні оновлення, нові

інструменти та велика кількість ресурсів для навчання.

Отже, Kotlin представляє собою вдосконалення в світі мов програмування, комбінуючи найкращі практики, безпеку та інноваційність. З його допомогою розробники можуть створювати високоякісні додатки швидше і ефективніше, незалежно від платформи або області застосування [10, 11, 12] .

Середовище розробки Android Studio.

Android Studio – це офіційне інтегроване середовище розробки (IDE) для платформи Android, створене та підтримуване компанією Google. Воно призначене для забезпечення розробників найновітніми інструментами, які спрощують створення додатків для Android. (Рисунок 3.5).



Рисунок 3.5. Логотип Android Studio

Однією з ключових особливостей Android Studio є його здатність до швидкого емулявання додатків на різних пристроях та версіях Android, що дозволяє розробникам легко тестувати та налагоджувати їх продукцію. Інтегровані інструменти для профілювання допомагають виявляти та усувати "вузькі" місця в продуктивності, пам'яті або живленні.

Android Studio також включає в себе різноманітні засоби для дизайну, такі як конструктор макетів, що дозволяє візуально створювати інтерфейси користувача без потреби в ручному кодуванні. Це спрощує процес дизайну та дозволяє розробникам легко переглядати, як їх додаток буде виглядати на різних екранах і роздільних здатностях.

Середовище також має вбудовану підтримку для системи управління версіями Git, що полегшує роботу в команді та ведення версій додатку. Крім того, інтеграція з Google Cloud Platform і Firebase забезпечує можливість легкого розширення можливостей додатку за допомогою хмарних ресурсів.

Отже, Android Studio представляє собою потужний інструмент для розробки додатків на Android, який об'єднує в собі найкращі практики та інноваційні рішення для створення ефективних, високоякісних та конкурентоспроможних продуктів. З його допомогою розробники отримують весь необхідний арсенал інструментів для приведення своїх ідей у життя [8, 9] .

Архітектура MVVM для мобільного додатка.

Впровадження архітектури MVVM у мобільний додаток

Архітектура MVVM (Model-View-ViewModel) (Рисунок 3.6) визначається як сучасний підхід до проектування програмного забезпечення. При виборі архітектурної моделі для мобільного додатка основна увага приділена її перевагам.

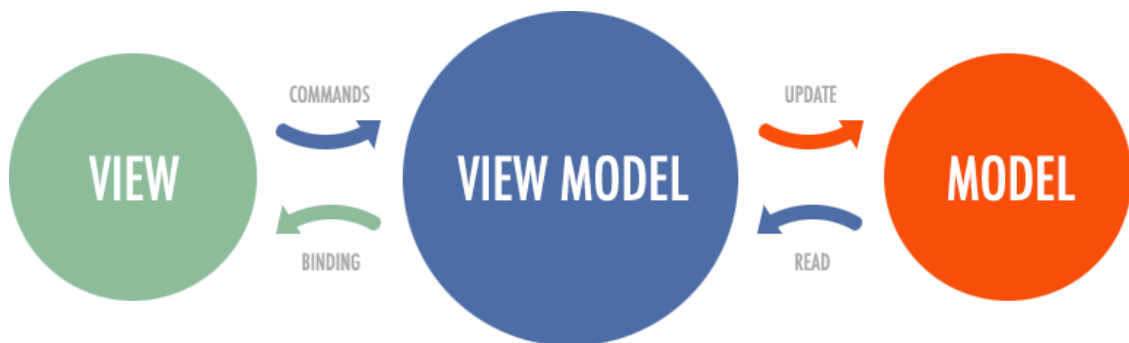


Рисунок 3.6. Архітектура MVVM

MVVM ділиться на три основних компоненти:

- 1) Model - ядро додатка. Містить бізнес-логіку та дані, зберігає інформацію і обробляє запити.
- 2) View - інтерфейс для взаємодії користувача з додатком. Відповідає за відображення інформації.

3) ViewModel - міст між Model та View. Взаємодіє з моделлю, обробляє команди від інтерфейсу і відображає дані в потрібному форматі.

Вагомими перевагами архітектури MVVM є:

- Відокремленість логіки від представлення. Це дозволяє модифікувати інтерфейс без зміни основної логіки додатка.
- Спрощеність тестування. Відокремленість бізнес-логіки від інтерфейсу полегшує процес тестування.
- Підвищена продуктивність. Можливість паралельної роботи над різними частинами додатка, що сприяє швидкості розробки.

Впровадження архітектури MVVM може супроводжуватися певними викликами. Однак гнучкість та масштабованість даної моделі допомагають ефективно їм протистояти, що сприяє створенню якісного продукту [3, 4] .

3.2. Опис методології створення програмного забезпечення

Для розробки мобільного додатку я вибрав методологію водоспаду (каскадну модель). Каскадна модель є послідовною методологією розробки, де кожен наступний етап починається після завершення попереднього. (Рисунок 3.7).

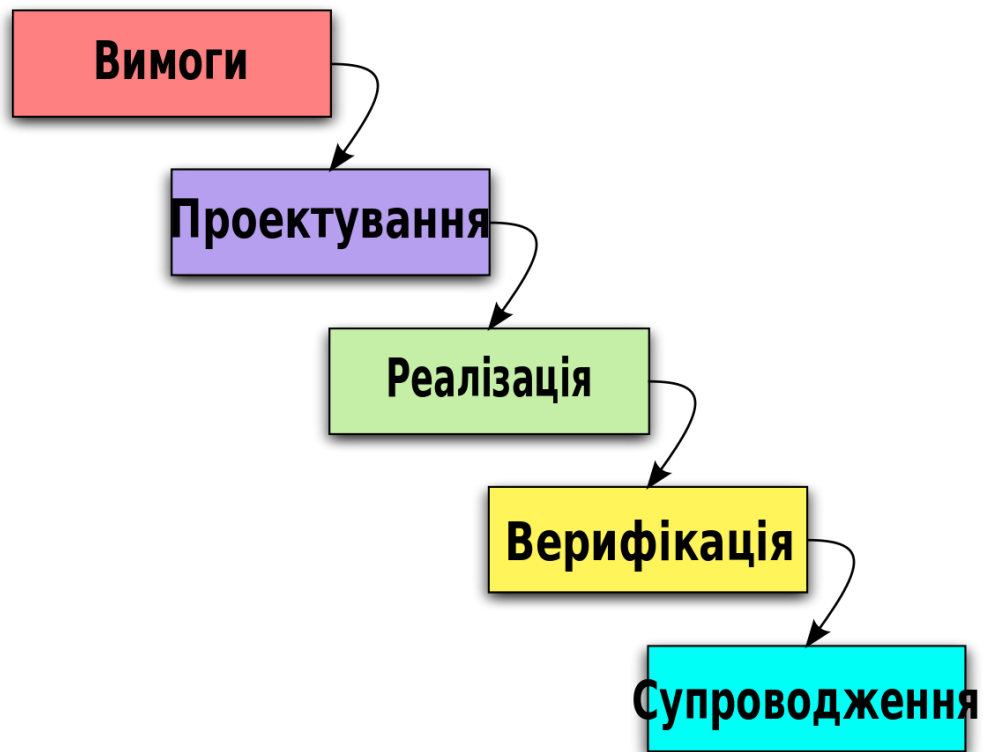


Рисунок 3.7. Модель розробки за каскадною методологією

Каскадна модель ділить процес розробки на послідовні фази:

- 1) **Вимоги.** На цьому етапі визначаються усі вимоги до програмного продукту. Це може включати функціональні вимоги, інтерфейс користувача, платформи та інші аспекти.
- 2) **Проектування.** На цьому етапі створюється архітектура додатку. Визначаються основні модулі, їх взаємодія, дизайн інтерфейсу та інші технічні деталі.
- 3) **Реалізація.** Згідно з проектом розробляється код додатку.
- 4) **Тестування.** Після розробки коду проводиться його тестування. На цьому етапі виявляються та виправляються різноманітні помилки.
- 5) **Впровадження.** Готовий продукт встановлюється та запускається на цільових пристроях.
- 6) **Підтримка.** Після випуску додатку, він може потребувати підтримки, оновлення або додаткового розвитку.

Основною перевагою каскадної моделі є її структурованість і простота. Вона відмінно підходить для невеликих проектів або проектів з чітко визначеними вимогами, які не змінюються протягом часу. Відсутність команди та необхідності швидко реагувати на зміни ринку робить цю модель ідеальною для цього конкретного проекту.

Модель розробки за каскадною методологією (етапи можуть відрізнятися через специфіку проекту, але суть одна у всіх) [14] .

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис та побудова діаграми прецедентів роботи з системою.

У контексті вашого додатку, діаграма Прецедентів (Рисунок 4.1), демонструє ключові можливості та функціональність вашої системи. Прецедент у цьому контексті є відображенням того, що додаток може виконувати, або тих дій, які користувачі можуть здійснювати за допомогою нього. Така діаграма допомагає краще зрозуміти та відобразити взаємодію між користувачами та додатком.

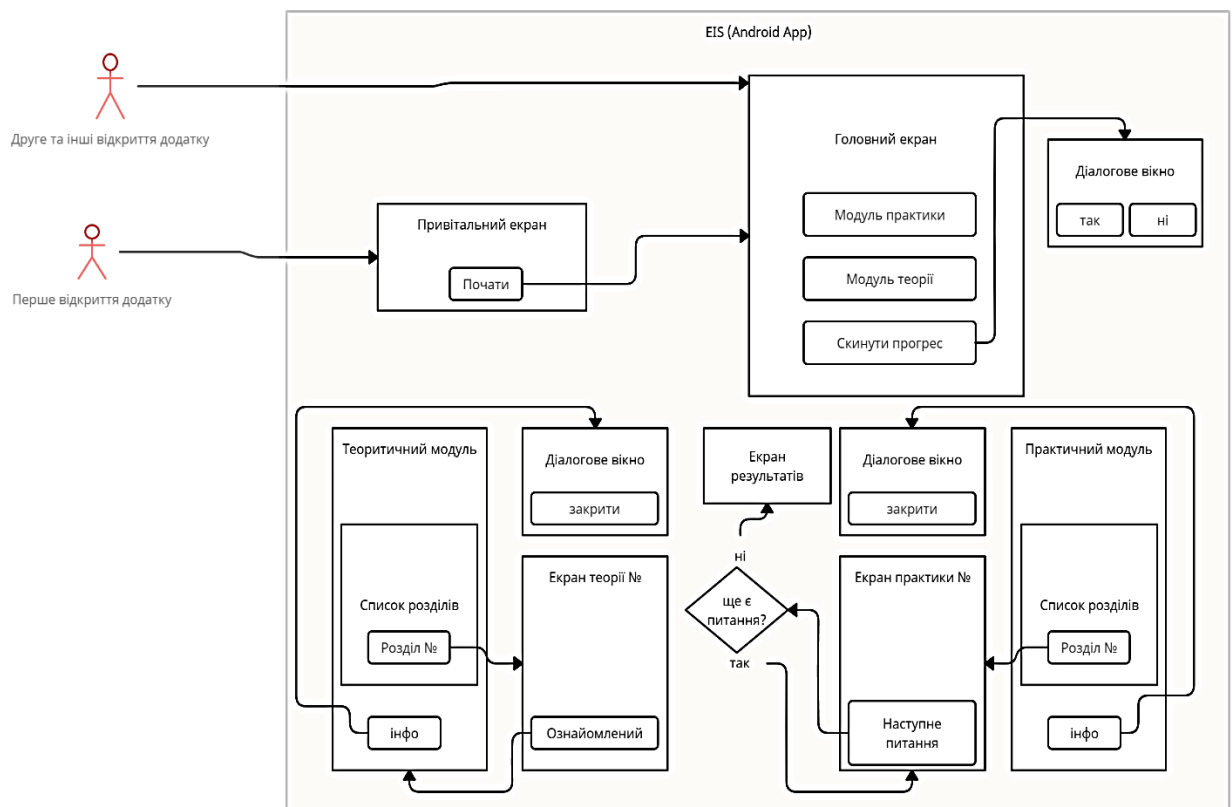


Рисунок 4.1. Діаграма прецедентів

4.2. Інструкція з використання додатка

При відкритті додатка користувача вітає «Splash Screen» - перший екран,

що служить мостом між запуском програми та її основною функціональністю. Цей екран не лише забезпечує візуальне підтвердження запуску додатка, але і є своєрідним "візитною карткою" програми. На схемі додатку він зображений першим

Ось схема всього додатка зображена у навігаційному графіку в Android Studio (Рисунок 4.2)

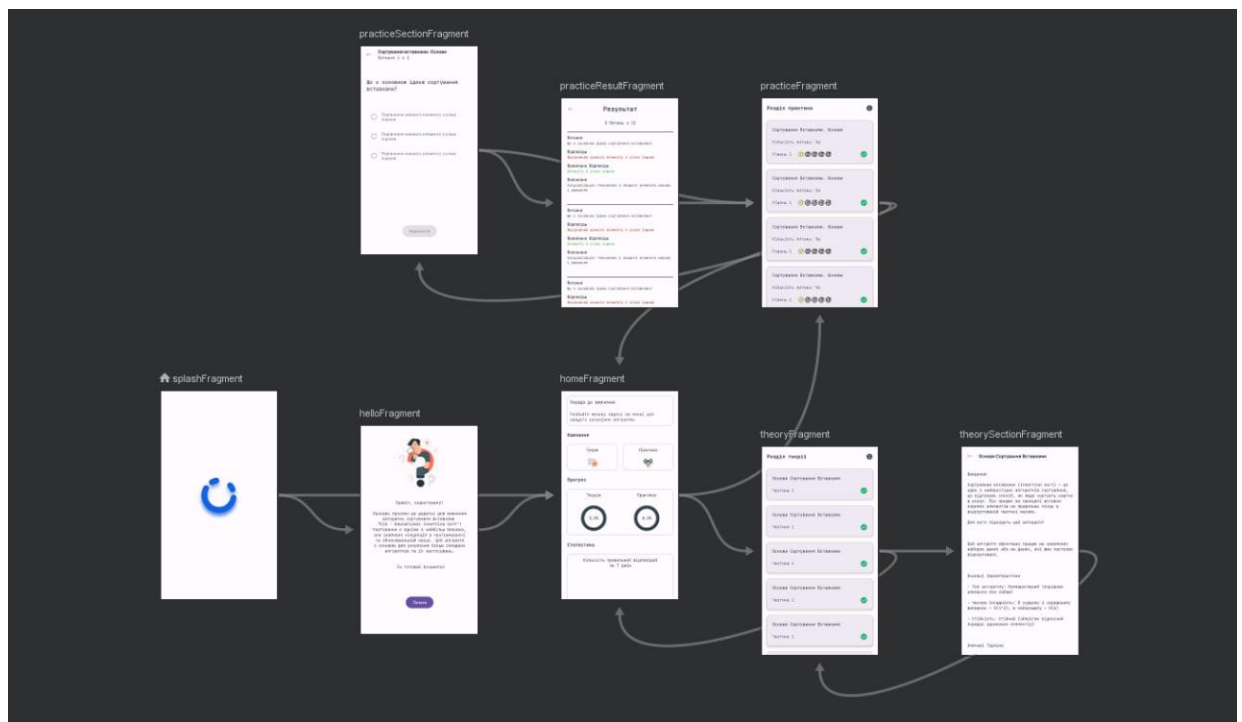


Рисунок 4.2. Схема додатку

За першого запуску, після «Splash Screen», користувач відразу зустрічається з привітальним екраном (Рисунок 4.3). Для переходу до основного меню додатка достатньо натиснути «Почати». Проте цей привітальний екран з'являється лише один раз: при наступних запусках додатка користувач потрапляє відразу на головний екран.

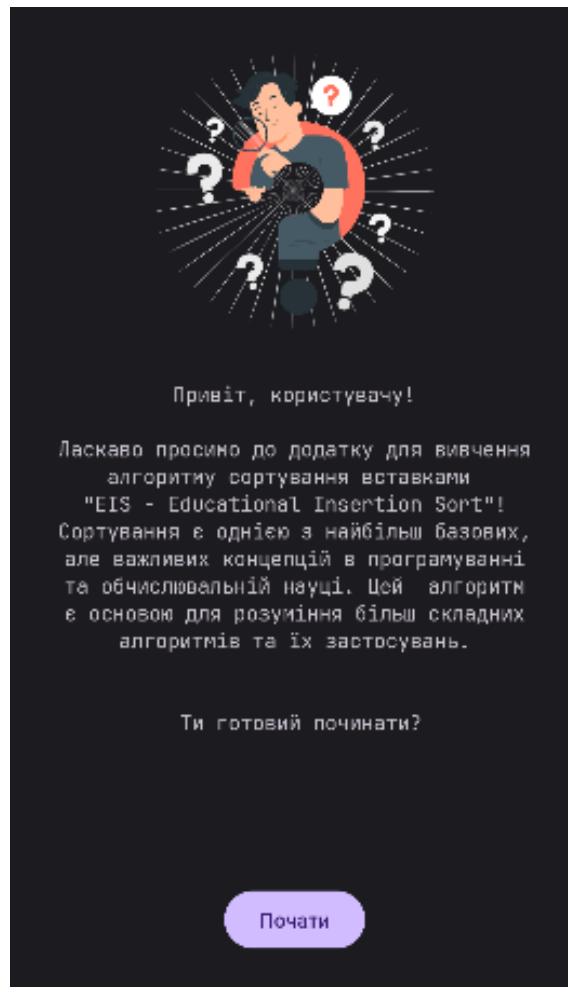


Рисунок 4.3. Привітальна (стартова) сторінка

Головний екран включає такі елементи:

Порада до вивчення (Рисунок 4.4) - це корисний текстовий напам'ят, який генерується при кожному запуску додатка, і надає поради для ефективного вивчення алгоритмів.

Розділ "Теорія та практика" (Рисунок 4.4) - центральний компонент, що дозволяє користувачеві обрати тематичний модуль для вивчення або практики.

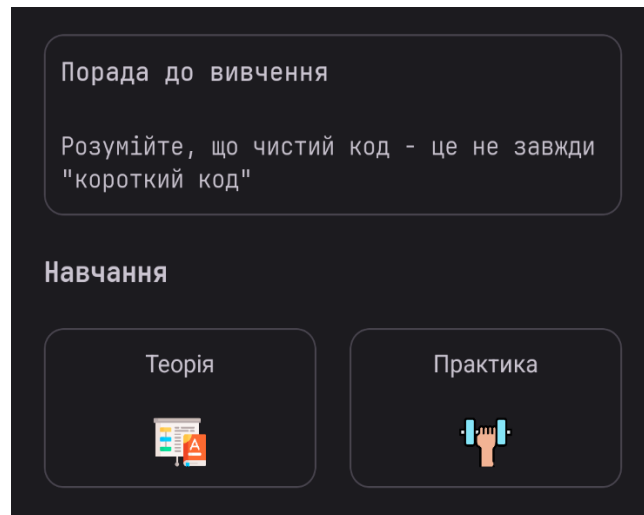


Рисунок 4.4. Поради до вивчення та розділи додатку

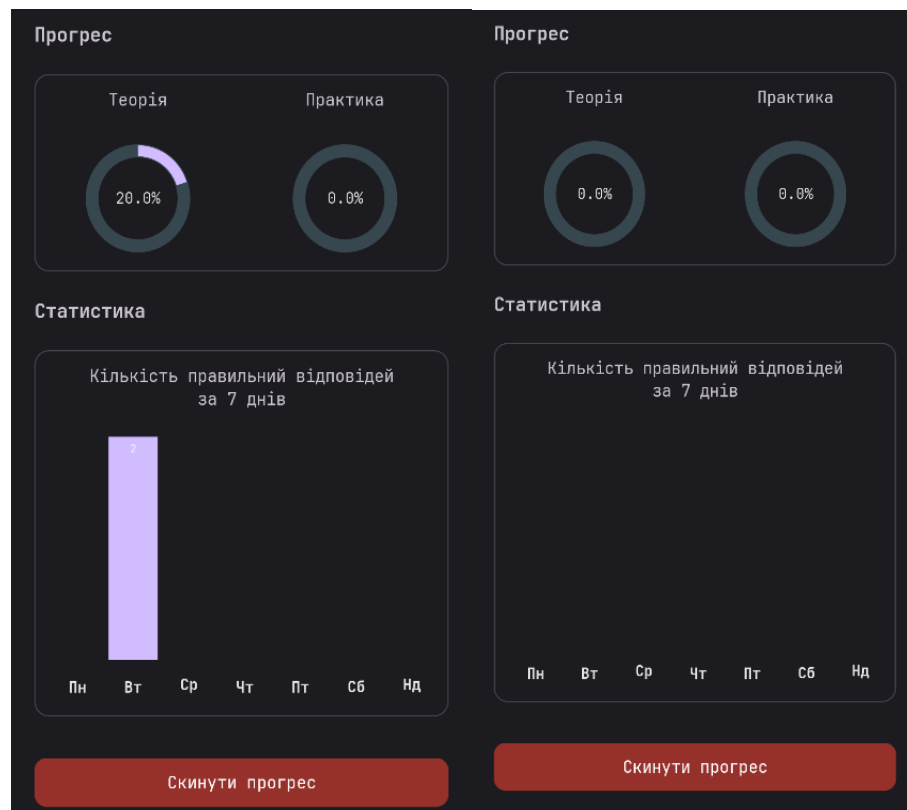


Рисунок 4.5.1. Статистика

Рисунок 4.5.2. Прогрес користувача

Статистика та Прогрес (Рисунок 4.5.1), (Рисунок 4.5.2) - візуалізація досягнень користувача. Тут відображено прогрес вивчення, а також графік правильних відповідей на тести, організований по днях тижня. Користувач має можливість скинути всю статистику за допомогою кнопки "Стерти прогрес".

Після її натиснення з'являється діалогове вікно (Рисунок 4.6), що попереджає про втрату всіх зібраних даних.

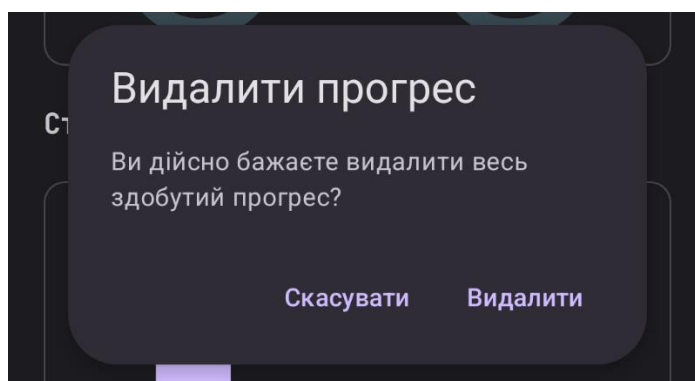


Рисунок 4.6. Попередження про видалення даних

У розділі "Теорія" розміщена кнопка FAQ, натиснувши на яку, користувачу відкривається діалогове вікно (Рисунок 4.8) із загальною інформацією про додаток. Також тут представлений список розділів теорії (Рисунок 4.7).

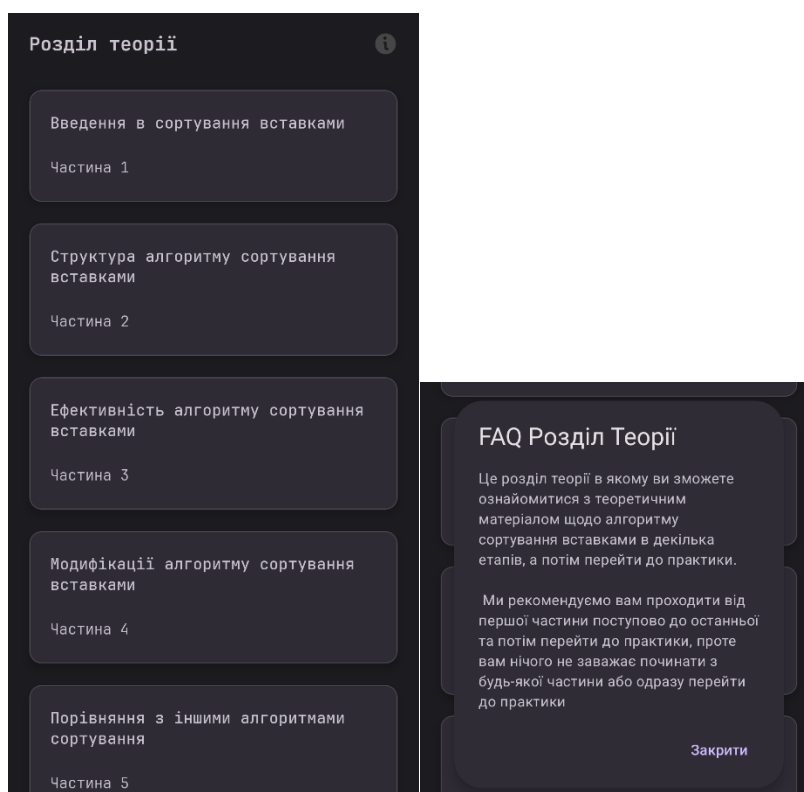


Рисунок 4.7. Розділ Теорія Рисунок 4.8. Інформація про додаток

Кожен конкретний розділ теорії включає в себе докладний опис теми, а також кнопку для відстежування особистого прогресу користувача в опануванні матеріалу (Рисунок 4.9).

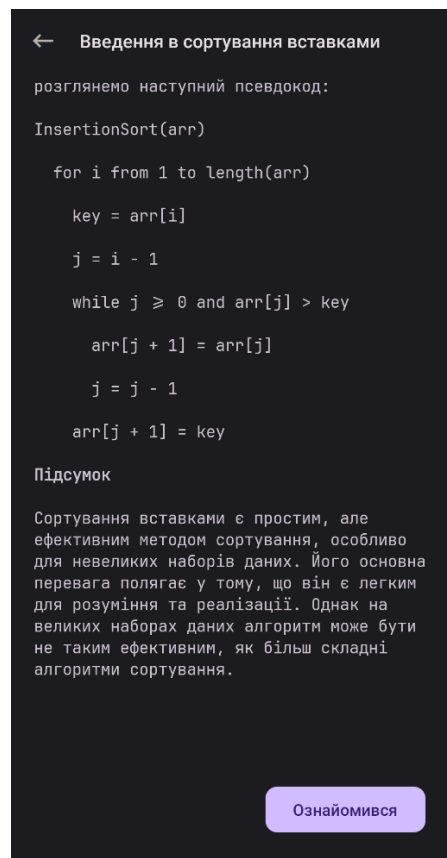


Рисунок 4.9. Частина процесу тестування із поясненням

У розділі "Практика" також знаходиться кнопка FAQ, відкриваючи яку користувач знову зустрічається з діалоговим вікном (Рисунок 4.11) та отримує інформацію про додаток. Окрім того, тут представлено список практичних тестів, що відповідають темам теорії (Рисунок 4.10).

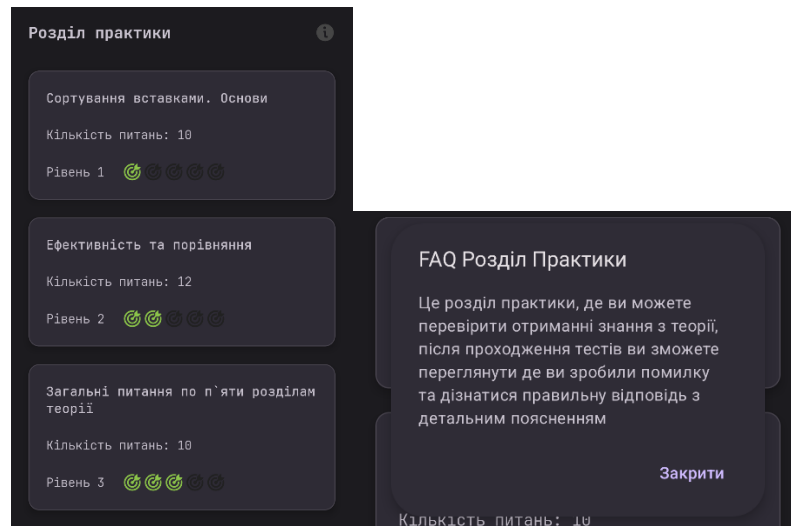


Рисунок 4.10. Практичні тести Рисунок 4.11. Інформація про додаток

Кожне завдання в тестах містить питання, три варіанти відповіді та кнопку "Далі". Користувач має можливість пропустити питання, якщо бажає (Рисунок 4.12).

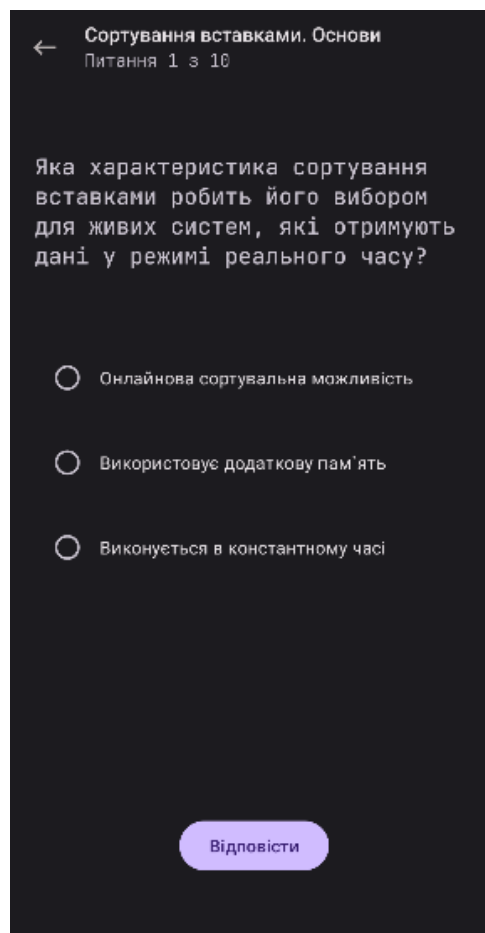


Рисунок 4.12. Частина процесу тестування

Після завершення тестування користувач отримує звіт із вказівкою на правильні та неправильні відповіді. Додатково, до кожної помилкової відповіді надається пояснення з правильним аргументуванням (Рисунок 4.13).

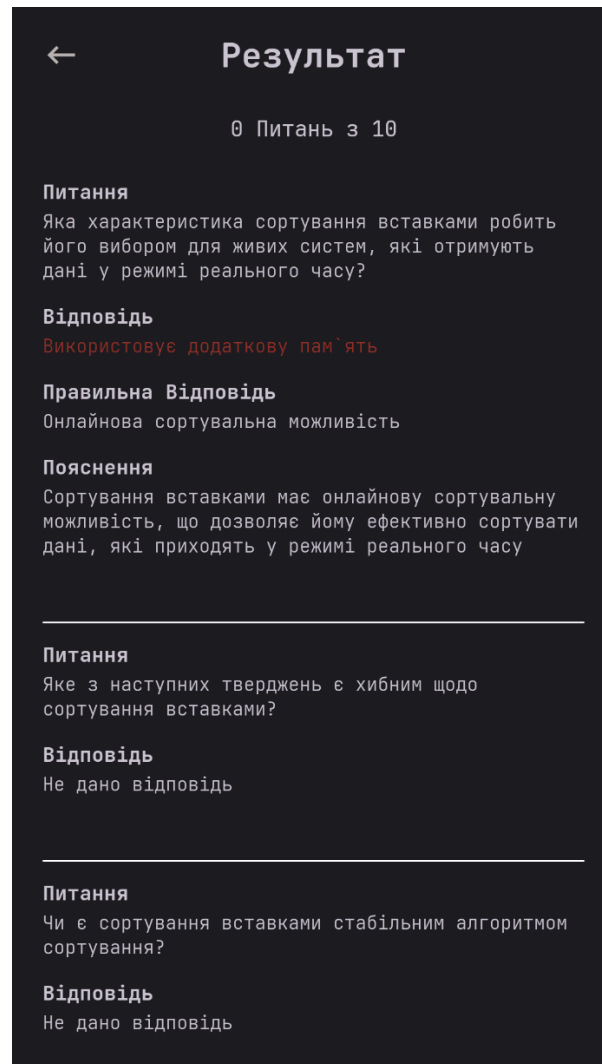


Рисунок 4.13. Результати тестування із поясненням

ВИСНОВКИ

В сучасному світі навчання через смартфон стає все більш популярним завдяки зручності та мобільності. Додаток для вивчення алгоритмів сортування вставками демонструє, як новітні технічні досягнення можуть оптимізувати освітній процес. В ході цього проєкту були розширені навички створення мобільних додатків для навчання. Основне завдання полягало в розробці програмного забезпечення для навчання через мобільний інтерфейс.

Отже, у результаті виконання дипломного проєкту нами було повністю виконано поставлені завдання дипломної роботи, а саме:

- 1) Визначено ключові вимоги до мобільного додатку для вивчення алгоритмів сортування;
- 2) Проведено аналіз переваг та обмежень аналогічних додатків для навчання;
- 3) Проаналізовано та описано підходи та інструменти, використані при створенні додатку;
- 4) Розроблено інтуїтивний мобільний інтерфейс для взаємодії користувача з матеріалом;
- 5) Створено докладну інструкцію користувача;
- 6) Представлено діаграму прецедентів, що ілюструє взаємодію користувача з додатком
- 7) Розроблено та апробовано навчальний андроїд застосунок з теми «Сортування вставками» для дистанційного навчального курсу «Алгоритми і структури даних».

Розроблений застосунок можна використовувати для будь-яких професійних дисциплін під час вивчення алгоритмів сортування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Черненко О. О. Курсовий проєкт із фаху : методичні рекомендації щодо оформлення пояснювальних записок до курсового проєкту для студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра, магістра /О. О. Черненко. – Полтава : ПУЕТ, 2022. – 58 с. Режим доступу: URL: <http://dspace.puet.edu.ua/handle/123456789/12852> - Назва з екрану
2. Посібник користувача Photoshop. Режим доступу: URL: <https://helpx.adobe.com/ua/photoshop/user-guide.html> - Назва з екрану
3. MVVM Clean Architecture Pattern in Android with Use Cases. Режим доступу: URL: <https://medium.com/@ami0275/mvvm-clean-architecture-pattern-in-android-with-use-cases-eff7edc2ef76> - Назва з екрану
4. Model-View-ViewModel (MVVM) . Режим доступу: URL: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm> - Назва з екрану
5. SQLite documentation. Режим доступу: URL: <https://www.sqlite.org/docs.html> - Назва з екрану
6. What Does Android SDK Mean? . Режим доступу: URL: <https://www.techopedia.com/definition/4220/android-sdk> - Назва з екрану
7. What Is the Android SDK and How to Start Using It. Режим доступу: URL: <https://code.tutsplus.com/the-android-sdk-tutorial--cms-34623t> - Назва з екрану
8. Meet Android Studio. Режим доступу: URL: <https://developer.android.com/studio/intro> - Назва з екрану
9. Developer workflow basics. Режим доступу: URL: <https://developer.android.com/studio/workflow> - Назва з екрану
10. Kotlin home page. Режим доступу: URL: <https://kotlinlang.org/> - Назва з екрану
11. Kotlin for Android. Режим доступу: URL: <https://kotlinlang.org/docs/android-overview.html> - Назва з екрану

12. Welcome to our tour of Kotlin! . Режим доступу: URL:
<https://kotlinlang.org/docs/kotlin-tour-welcome.html> - Назва з екрану
13. Android SDK and it's Components. Режим доступу: URL:
<https://www.geeksforgeeks.org/android-sdk-and-its-components/> - Назва з екрану
14. Каскадна модель (waterfall model) . Режим доступу: URL:
<https://qalight.ua/baza-znaniy/kaskadna-model-waterfall-model/> - Назва з екрану

ДОДАТОК А

Код програми

HOME FRAGMENT FROM SOURCE CODE (HomeFragment.kt) :

```
package com.educational.insertsort.presentation.fragments.home_screen

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.activity.addCallback
import androidx.fragment.app.Fragment
import androidx.lifecycle.lifecycleScope
import androidx.navigation.fragment.findNavController
import com.educational.insertsort.R
import com.educational.insertsort.data.storage.StorageImpl.Companion.currentDate
import com.educational.insertsort.data.storage.StorageImpl.Companion.isSameWeek
import com.educational.insertsort.data.storage.StorageImpl.Companion.lastSaveCalendar
import com.educational.insertsort.databinding.FragmentHomeBinding
import com.educational.insertsort.domain.ViewModelPractice
import com.educational.insertsort.domain.ViewModelStatistic
import com.educational.insertsort.domain.ViewModelTheory
import com.educational.insertsort.presentation.MainActivity.Companion.logs
import com.educational.insertsort.presentation.fragments.home_screen.sub.ListOfTipsHome
import com.educational.insertsort.presentation.fragments.home_screen.sub.ListOfTipsToLearn
import com.google.android.material.dialog.MaterialAlertDialogBuilder
import com.google.android.material.snackbar.Snackbar
import kotlinx.coroutines.CoroutineExceptionHandler
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.flow.collectLatest
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext
import org.koin.androidx.viewmodel.ext.android.viewModel
import java.util.Calendar

class HomeFragment : Fragment() {

    private lateinit var binding: FragmentHomeBinding

    private val exceptionHandler = CoroutineExceptionHandler { _, throwable ->
```

```

        logs(throwable)
    }

    private val snackbarToExit by lazy {
        Snackbar.make(
            binding.root,
            resources.getString(R.string.exit_snackbar),
            Snackbar.LENGTH_SHORT
        )
    }

    private val viewModelTheory: ViewModelTheory by viewModel()
    private val viewModelPractice: ViewModelPractice by viewModel()
    private val viewModelStatistic: ViewModelStatistic by viewModel()

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        binding = FragmentHomeBinding.inflate(inflater)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        lifecycleScope.launch(exceptionHandler) {
            viewModelTheory.flowCompletedTheoryData.collectLatest {
                var value = String.format("%.2f", (100f / 5) * it.size)
                    .replace(",", ".").toFloat()
                if (value > 100f) value = 100f
                binding.circularProgressBarTheory.progress = value
            }
        }

        lifecycleScope.launch(exceptionHandler) {
            viewModelPractice.flowPracticeCompleteData.collectLatest {
                var value = String.format("%.2f", (100f / 3) * it.size)
                    .replace(",", ".").toFloat()
                if (value > 100f) value = 100f
                binding.circularProgressBarPractice.progress = value
            }
        }
    }

```

```

lifecycleScope.launch(exceptionHandler) {
    viewModelStatistic.flowTestStatisticData.collectLatest { list ->
        binding.weekBarChartView.dayValues = list
    }
}

binding.resetProgressButton.setOnClickListener {
    AlertDialog.Builder(requireContext())
        .setTitle("Видалити прогрес")
        .setMessage("Ви дійсно бажаєте видалити весь здобутий прогрес?")
        .setPositiveButton("Видалити") { dialog, _ ->
            dialog.dismiss()
            viewModelTheory.resetTheoryData()
            viewModelPractice.resetPracticeCompleteData()
            viewModelStatistic.clearTestStatisticData()
        }
        .setNegativeButton("Скасувати") { dialog, _ ->
            dialog.cancel()
        }.create().show()
}

binding.theoryPart.setOnClickListener {

findNavController().navigate(R.id.action_homeFragment_to_theoryFragment)
}

binding.practicePart.setOnClickListener {

findNavController().navigate(R.id.action_homeFragment_to_practiceFragment)
}

}

override fun onResume() {
    super.onResume()

    lifecycleScope.launch(exceptionHandler) {
        val lastSavedDate = withContext(Dispatchers.IO) {
            viewModelStatistic.getLastSavedDate()
        }
    }
}

```

```

logs(
    "RESULT: \n" +
        "is same week ${
            isSameWeek(
                currentDate(),
                lastSavedDate.lastSaveCalendar()
            )
        }\n" +
        "currentDate = ${currentDate().get(Calendar.WEEK_OF_YEAR)}\n"
+
        "lastSavedDate
${lastSavedDate.lastSaveCalendar().get(Calendar.WEEK_OF_YEAR)}"
    )

    if (!isSameWeek(currentDate(), lastSavedDate.lastSaveCalendar())) {
        viewModelStatistic.clearTestStatisticData()
    }
}

// exit from app if user dabble press back in 2 seconds
var backTimeExit = 0L

requireActivity().onBackPressedDispatcher.addCallback {
    if (backTimeExit + 2000L >= System.currentTimeMillis()) {
        requireActivity().finish()
    } else {
        snackbarToExit.show()
        backTimeExit = System.currentTimeMillis()
    }
}
// set tips to learn
binding.tipsTextField.text = resources.getString(listOfTipsToLearn.random())
}
}

```

HOME FRAGMENT XML CODE (fragment_home.xml) :

```

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

```

```

android:layout_height="match_parent"
tools:context=".presentation.fragments.home_screen.HomeFragment">

```

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```

```

<com.google.android.material.card.MaterialCardView
    android:id="@+id/materialCardView"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/medium_margin"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/easy_margin"
    android:fontFamily="@font/pitagon_medium"
    android:text="@string/tips_card"
    android:textSize="15sp" />

```

```

<TextView
    android:id="@+id/tipsTextField"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/easy_margin"
    android:fontFamily="@font/pitagon_light"
    tools:text="@string/tipsToLearn1" />

```

```

</LinearLayout>

```

```

</com.google.android.material.card.MaterialCardView>

```

```

<TextView
    android:id="@+id/textView2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/medium_margin"

```

```

        android:fontFamily="@font/pitagon_bold"
        android:text="@string/education_text_label"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/materialCardView" />

<com.google.android.material.card.MaterialCardView
    android:id="@+id/theoryPart"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="@dimen/medium_margin"
    android:layout_marginTop="@dimen/medium_margin"
    android:layout_marginEnd="@dimen/easy_margin"
    android:clickable="true"
    android:focusable="true"
    app:layout_constraintEnd_toStartOf="@+id/practicePart"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintHorizontal_chainStyle="packed"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="@dimen/easy_margin"
            android:gravity="center_horizontal"
            android:text="@string/theoretical_part" />

        <ImageView
            android:layout_width="32dp"
            android:layout_height="32dp"
            android:layout_gravity="center_horizontal"
            android:layout_margin="@dimen/easy_margin"
            android:importantForAccessibility="no"
            android:src="@drawable/theory" />
    </LinearLayout>
</com.google.android.material.card.MaterialCardView>

<com.google.android.material.card.MaterialCardView

```

```

android:id="@+id/practicePart"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginStart="@dimen/easy_margin"
android:layout_marginTop="@dimen/medium_margin"
android:layout_marginEnd="@dimen/medium_margin"
android:clickable="true"
android:focusable="true"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.5"
app:layout_constraintStart_toEndOf="@+id/theoryPart"
app:layout_constraintTop_toBottomOf="@+id/textView2">

```

```
<LinearLayout
```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

```

```
<TextView
```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/easy_margin"
    android:gravity="center_horizontal"
    android:text="@string/practice_part" />

```

```
<ImageView
```

```

    android:layout_width="32dp"
    android:layout_height="32dp"
    android:layout_gravity="center_horizontal"
    android:layout_margin="@dimen/easy_margin"
    android:importantForAccessibility="no"
    android:src="@drawable/practice" />

```

```
</LinearLayout>
```

```
</com.google.android.material.card.MaterialCardView>
```

```
<TextView
```

```

    android:id="@+id/textView3"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/medium_margin"
    android:fontFamily="@font/pitagon_bold"
    android:text="@string/progress_text_label"
    android:textSize="16sp"

```

```

app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/theoryPart" />

```

```

<com.google.android.material.card.MaterialCardView
    android:id="@+id/materialCardView2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/medium_margin"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:orientation="vertical">

```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/easy_margin"
    android:fontFamily="@font/pitagon_regular"
    android:gravity="center_horizontal"
    android:text="@string/theoretical_part" />

```

```

<com.educational.insertsort.presentation.fragments.home_screen.sub.CircularProgress
sBar

```

```

    android:id="@+id/circularProgressBarTheory"
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_gravity="center"
    android:layout_margin="@dimen/easy_margin" />

```

```

</LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"

```

```
android:orientation="vertical">
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/easy_margin"
    android:fontFamily="@font/pitagon_regular"
    android:gravity="center_horizontal"
    android:text="@string/practice_part" />
```

```
<com.educational.insertsort.presentation.fragments.home_screen.sub.CircularProgress
sBar
```

```
    android:id="@+id/circularProgressBarPractice"
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_gravity="center"
    android:layout_margin="@dimen/easy_margin" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
</com.google.android.material.card.MaterialCardView>
```

```
<TextView
    android:id="@+id/textView4"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/medium_margin"
    android:fontFamily="@font/pitagon_bold"
    android:text="@string/statistic_text_label"
    android:textSize="16sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/materialCardView2" />
```

```
<com.google.android.material.card.MaterialCardView
    android:id="@+id/materialCardView3"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/medium_margin"
    android:layout_marginTop="@dimen/medium_margin"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4">
```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/easy_margin"
        android:fontFamily="@font/pitagon_regular"
        android:gravity="center_horizontal"
        android:text="@string/statistic_test_7d" />

```

```

<com.educational.insertsort.presentation.fragments.home_screen.sub.WeekBarChart
View

```

```

        android:id="@+id/weekBarChartView"
        android:layout_width="match_parent"
        android:layout_height="220dp"
        android:layout_gravity="center"
        android:layout_margin="@dimen/easy_margin" />
</LinearLayout>

```

```

</com.google.android.material.card.MaterialCardView>

```

```

<com.google.android.material.button.MaterialButton
    android:layout_width="0dp"
    android:id="@+id/resetProgressButton"
    android:layout_height="wrap_content"
    android:layout_marginStart="@dimen/medium_margin"
    android:layout_marginTop="@dimen/big_margin"
    android:layout_marginEnd="@dimen/medium_margin"
    android:layout_marginBottom="@dimen/medium_margin"
    android:backgroundTint="@color/red"
    android:textColor="@color/white"
    android:fontFamily="@font/pitagon_regular"
    android:text="@string/reset_progress_text"
    app:cornerRadius="10dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/materialCardView3" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.core.widget.NestedScrollView>

```