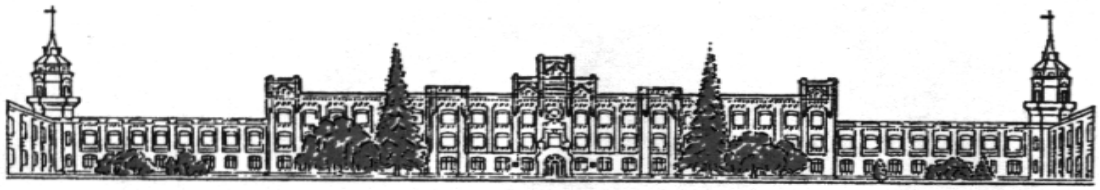


Міністерство освіти і науки, молоді та спорту України
Національний Технічний Університет України
«Київський Політехнічний Інститут»
Навчально-науковий комплекс
«Інститут прикладного системного аналізу»
Кафедра системного проектування

«ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ»

Лабораторна робота № 4 та 5
Розробка тестового мобільний додаток за темою індивідуального завдання.
Модульне тестуваня (Unit-тести) та рефакторинг.

Виконала:
студентка 4 курсу,
Група ДА-61
Фецун Анна
Варіант 24

м. Київ
2019

Мета роботи: розробити тестовий мобільний додаток за темою індивідуального завдання.

Задача: вивчити принципи побудови мобільних додатків на прикладі системи Андроід.

Побудувати інтерфейс користувача та функціонал (частково) інформаційної системи за обраною індивідуальною темою. Частково реалізувати функціонал додатку (CRUD).

Використовувати мобільну нативну платформу.

Завдання

1. Створити інтерфейс користувача мобільного додатку інформаційної системи.

2. Розробити основний функціонал мобільного додатку CRUD.

3. Провести тестування мобільного додатку відповідно до SRS з л.р. 2.

Хід виконання роботи:

1. Описати інтерфейс користувача мобільного додатку.

2. Зробити опис архітектури мобільного додатку.

3. Провести програмування основного функціоналу.

4. Провести тестування мобільного додатку.

5. Оформити протокол виконання лабораторної роботи.

Виконання лабораторної роботи відповідно до плану.

1. Описати інтерфейс користувача мобільного додатку.

На основі макетів з лабораторної роботи, був створенний мобільний інтерфейс користувача на платформу *Android 8.1, Pixel 2 API 27, 1080x1920 : 420 dpi.*

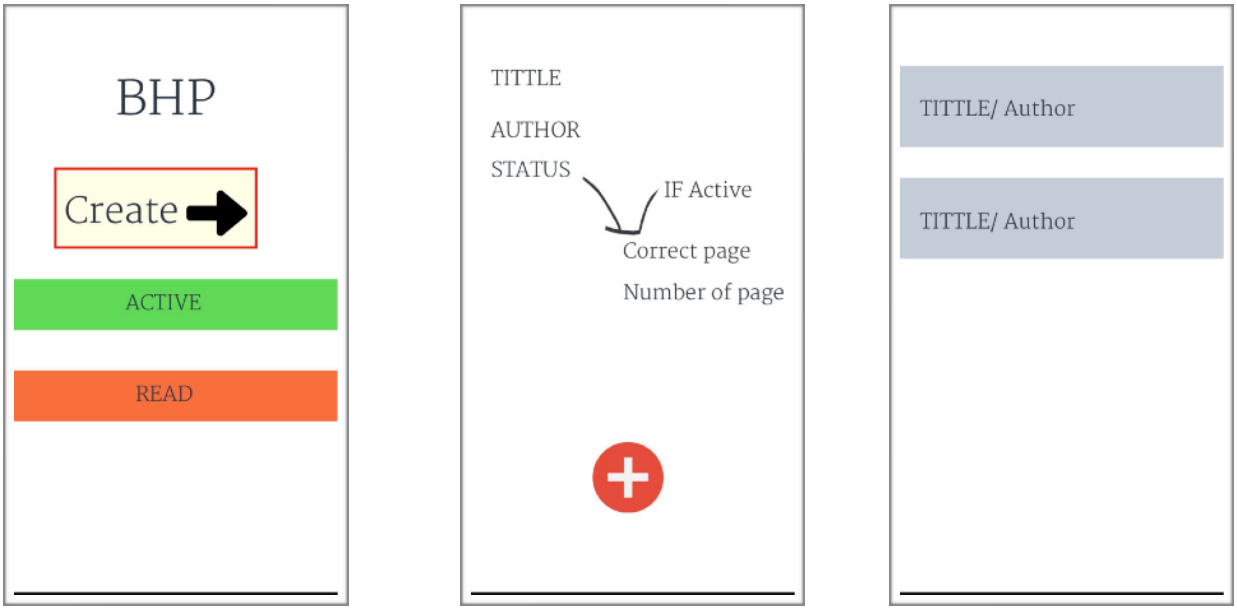


Рис1. Макети

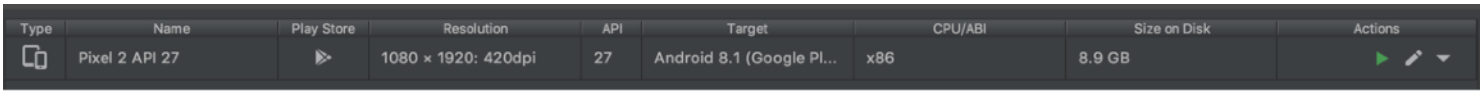


Рис2. Virtual Devices (Android Studio)

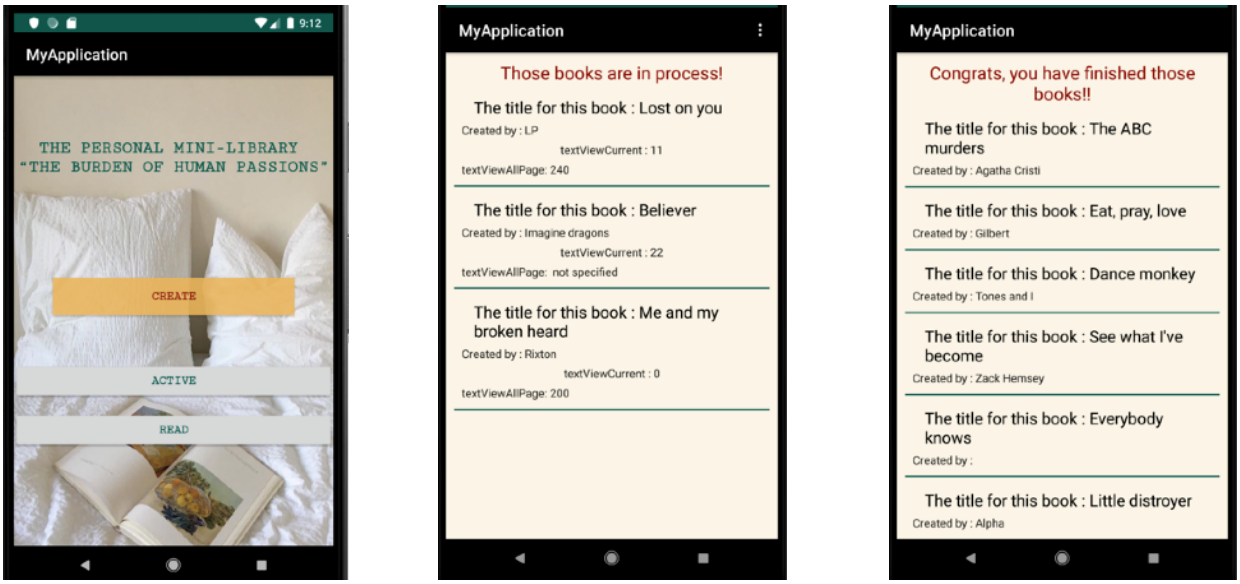


Рис3а. Реалізація інтерфейсів: Main, Active і Read компонентів відповідно.

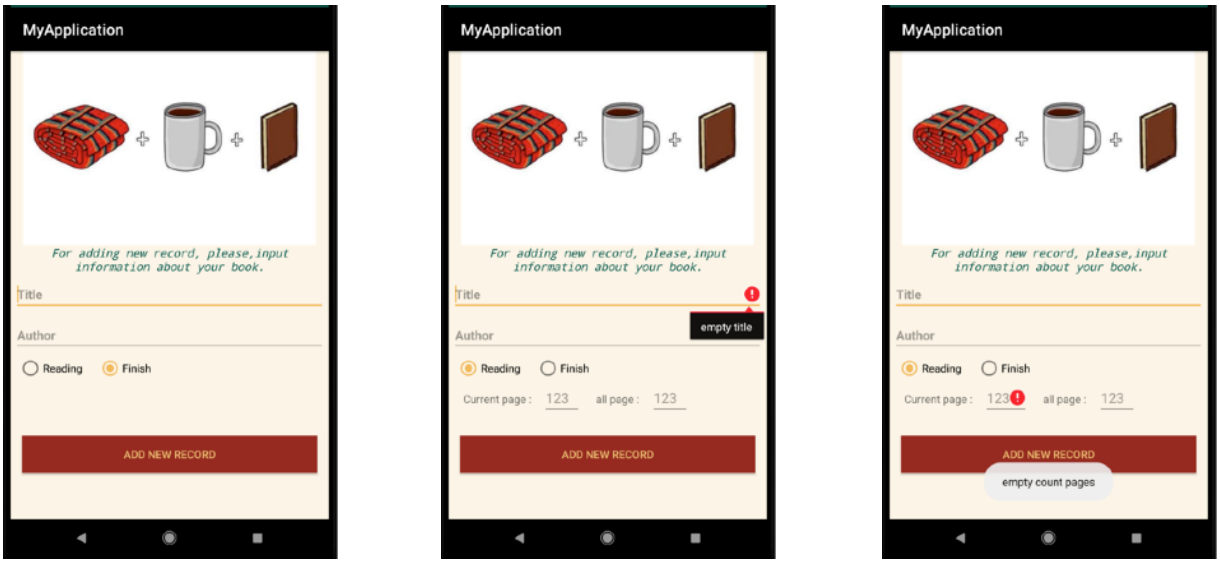


Рис3b. Реалізація Create компонента: обов'язкові поля -Title та current page при reading acntion.

2. Зробити опис архітектури мобільного додатку.

2.1. Модель предметної області

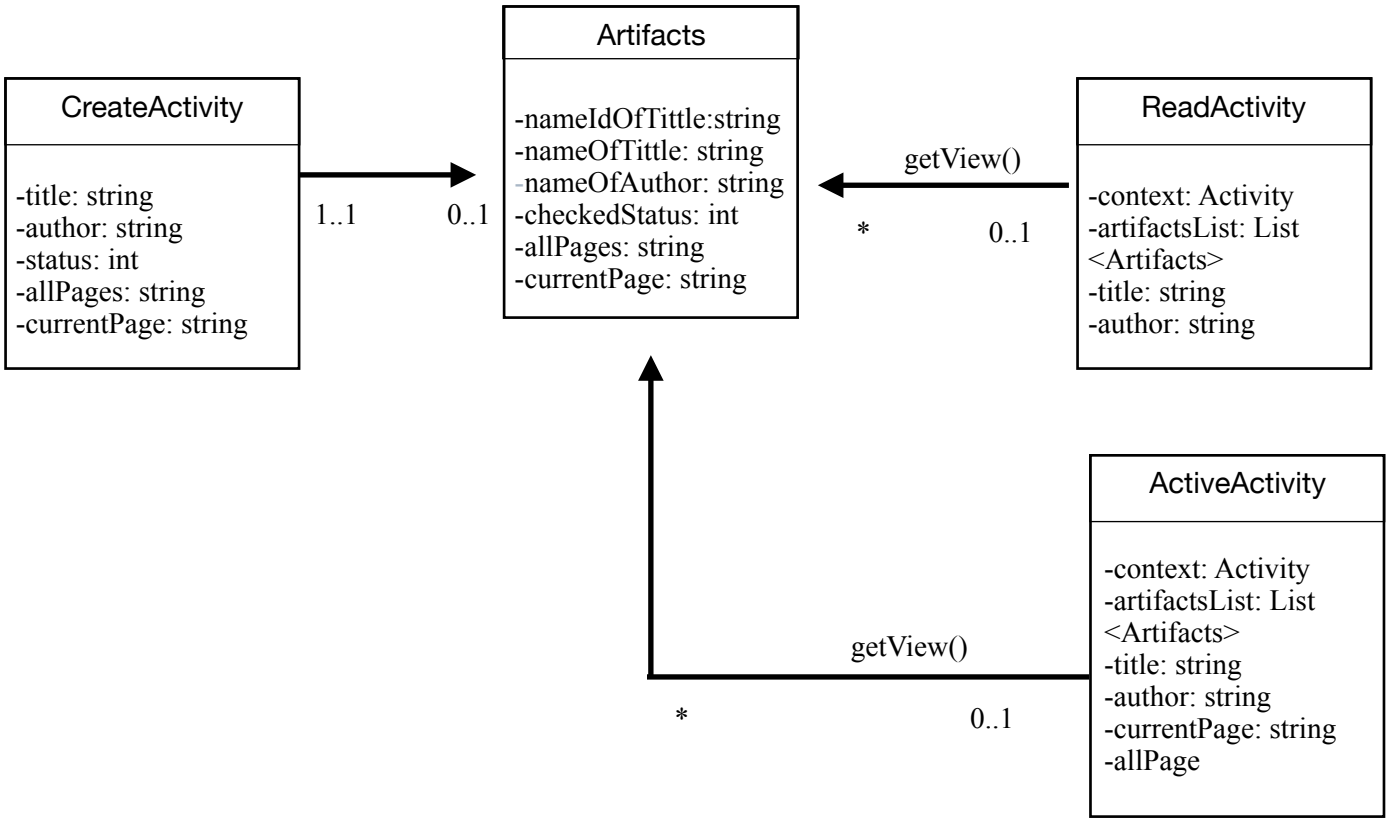
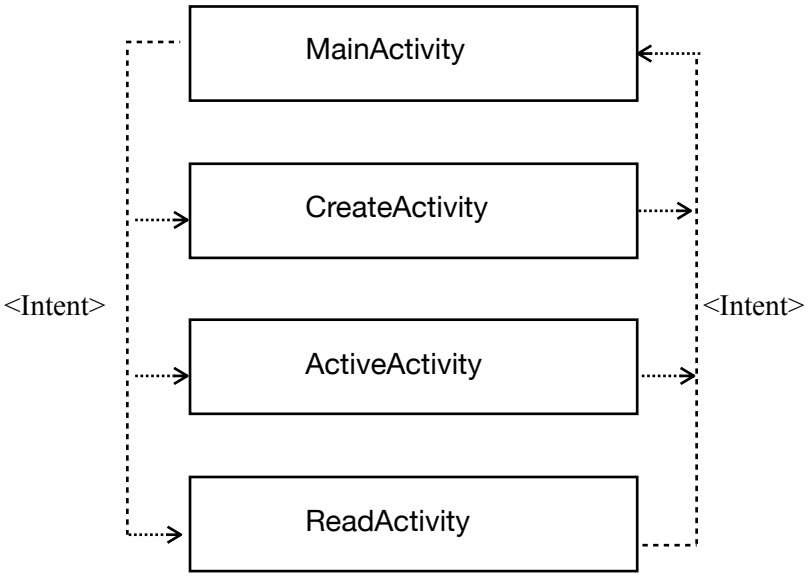


Рис 4. Діаграма класів моделі

Загальний опис контролерів додатку:

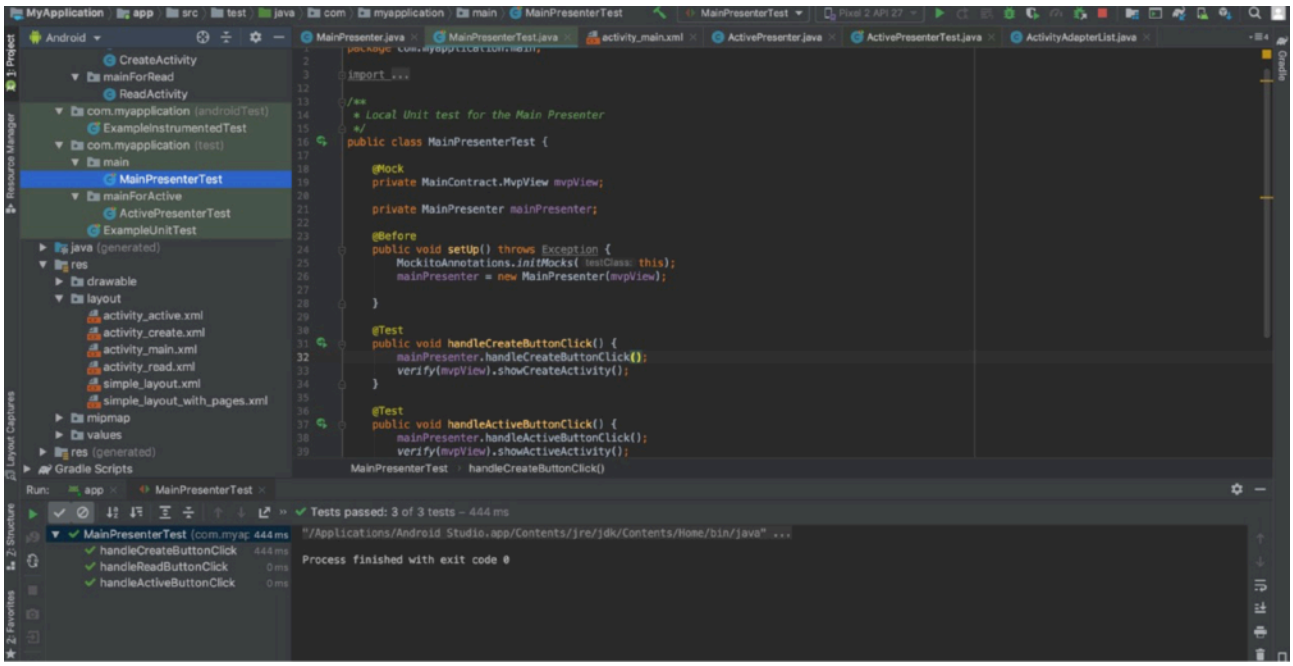
Додаток містить чотири Activity:



1. MainActivity: головний інтерфейс взаємозв’язку.

Реалізовано на основі MVP патерна та методології TDD, а саме:

- MainActivity: Displays the main screen
- MainContract: Represent interface for actions
- MainPresenter: Responsible for handling actions from the View and updating the UI as required.



2. CreateActivity: создание record та заносить в базу даних Firebase Real Time path (“artifacts”).

Record описує клас Artifacts, який містить відповідні get-методи.
3. ReadActivity: відтворює записи з бази даних на основі класу ArtifactsList.
4. ActiveActivity: відтворює записи з бази даних на основі класу ActiveAdapterList. Реалізований аналогічно MainActivity на основі MVP.

3. Провести програмування основного функціоналу.

4. Провести тестування мобільного додатку.

+ поля вводу відповідають заданому типу даних.

+ Помилка створення нового запису в базу даних при відсутності title та current page при reading.

+На основі tdd, розроблено Main та Active Activities.

Рис. Приклад мануального тестування

Лабораторна робота 5.

Модульне тестуваня (Unit-тести) та рефакторинг.

Мета роботи: оволодіти навичками створення програмного забезпечення за метолологією TDD та ознайомитися з процедурами рефакторинга.

Задача:

1. Використовувати методологію Test Driven Development для створення класів архітектурної програмної моделі.

2. Скласти тестові сценарії, які продемонструють функціювання всіх методів проектованої моделі.

3. Виконати юніт-тестування складових частин (внутрішніх класів), що реалізують об’єкт моделювання.

4. Виконати "зовнішнє" юніт-тестування для API.

5. Провести рефакторинг коду програми, для поліпшення реалізації.

Виконання.

1. Використовуючи методологію Test Driven Development, було розроблено модуть Main, аналогічно до Main - модуль Active (оскільки створення ідентичне - не описуємо) та додано модуль LoginForm для наглядної демонстрації функціональної розробки за TDD.

Компонент Main, створений на основі MVP-патерну, містить:

- MainContract: представляє інтерфейси для дій;
- MainPresenter: відповідає за обробку дій з View (а саме MvpView) та оновлення інтерфейсу користувача, якщо потрібно.
- MainActivity: вхідна точка додатку.

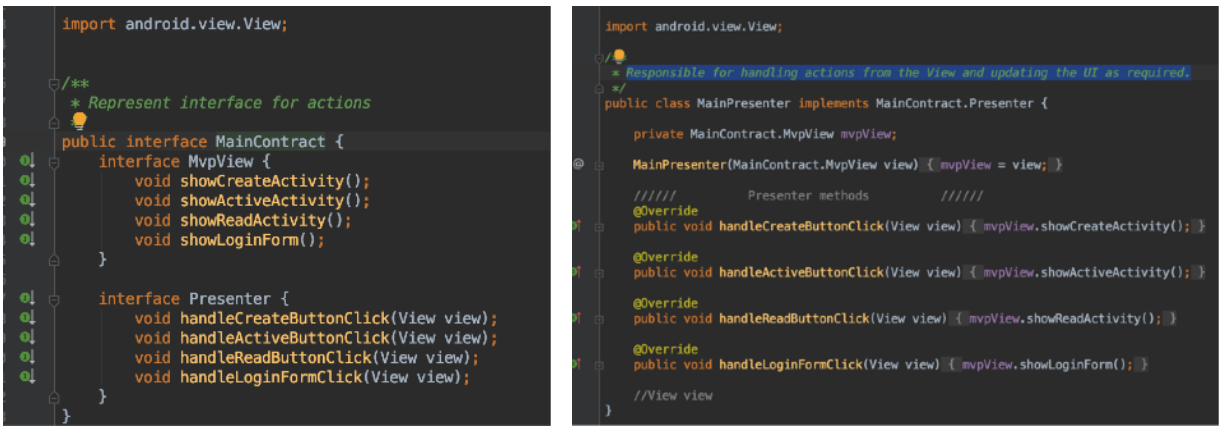
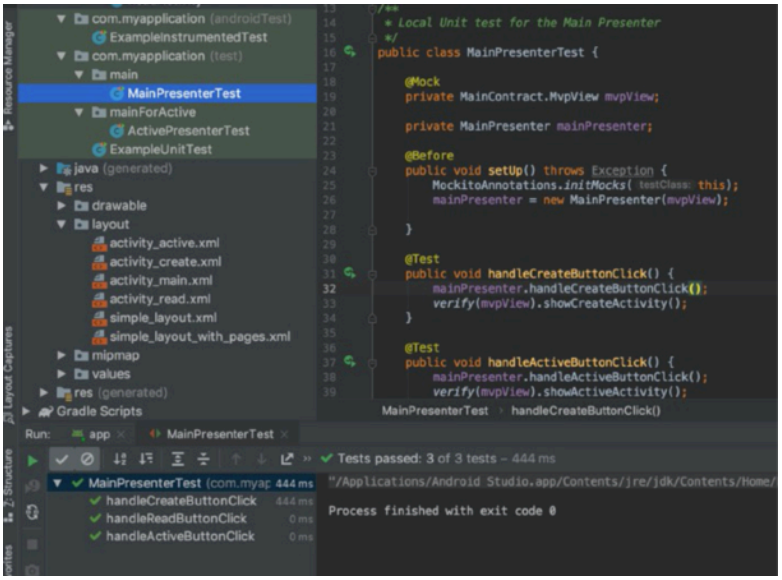
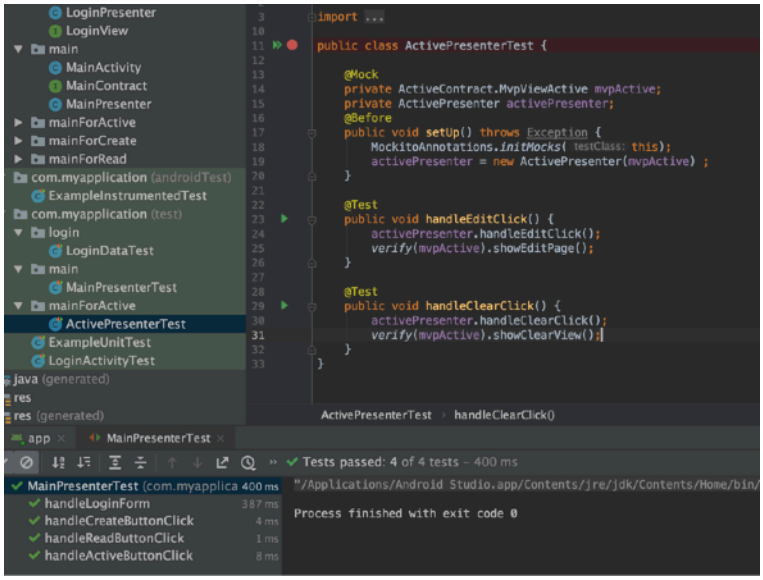


Рис. Вміст відповідних класів: MainContract та MainPresenter.

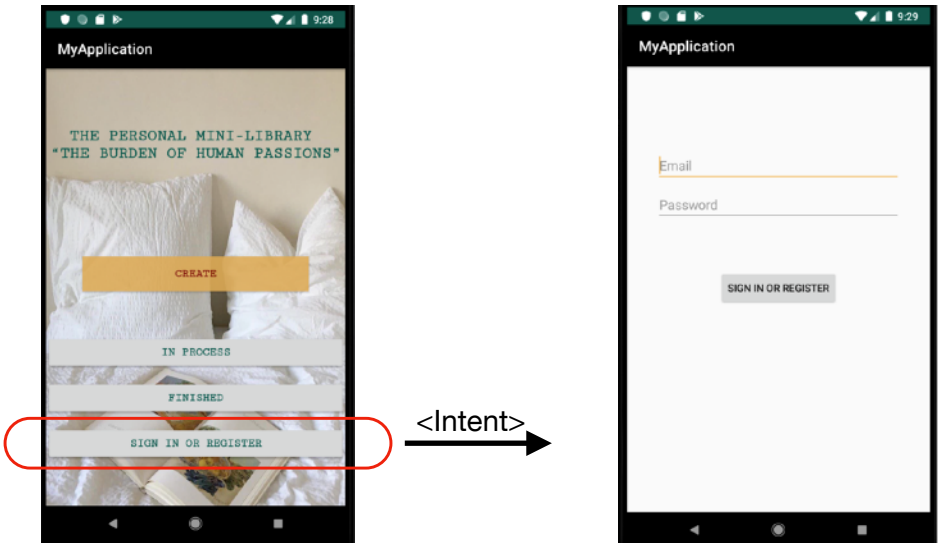
Фінальні тести при розробці з використанням Mockito та Unit.





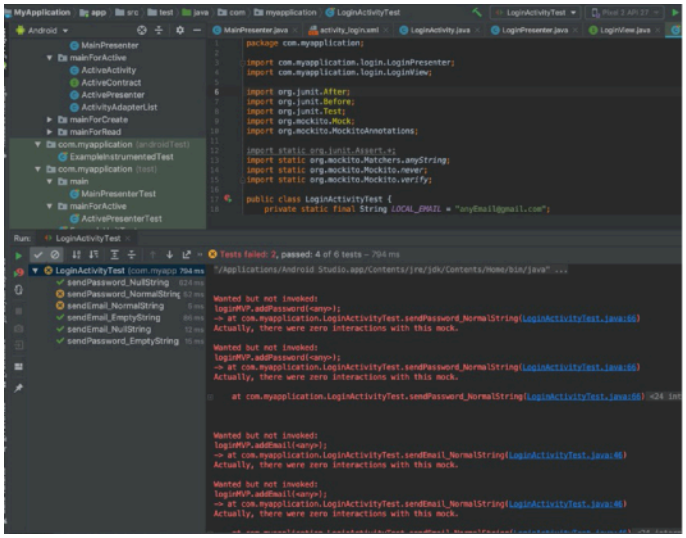
Компонент LoginForm, на основі MVP патенрну, містить:

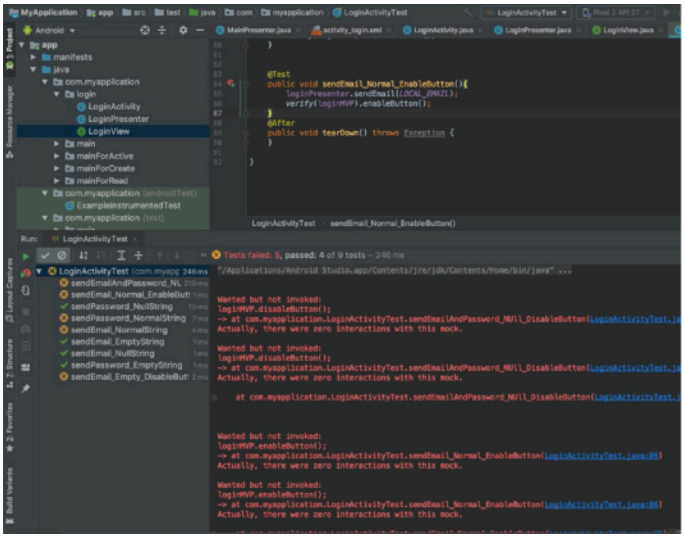
- LoginView: відповідає за інтерфейси
- LoginPresenter: контролер
- LoginData: збереження та обробка даних, без підключення до бази даних.
- LoginActivity: дисплейний вигляд компонента.

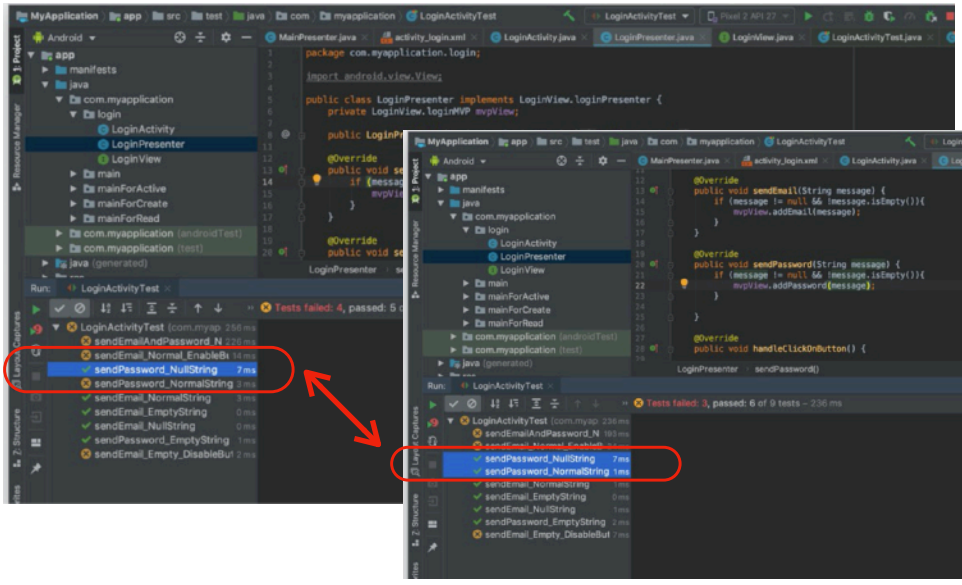


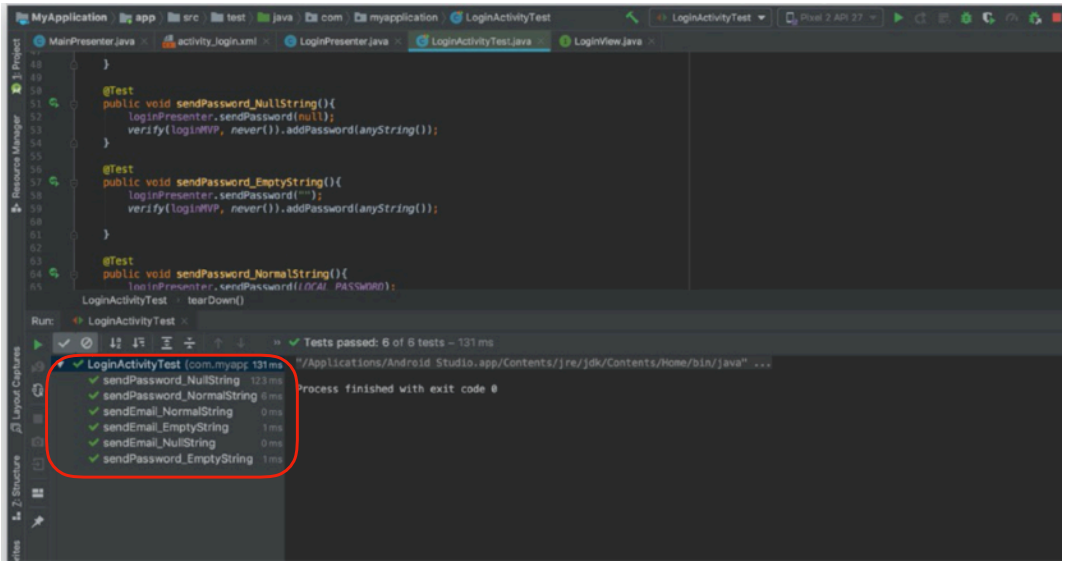
Тести при розробці

Перевірка на коректність даних.





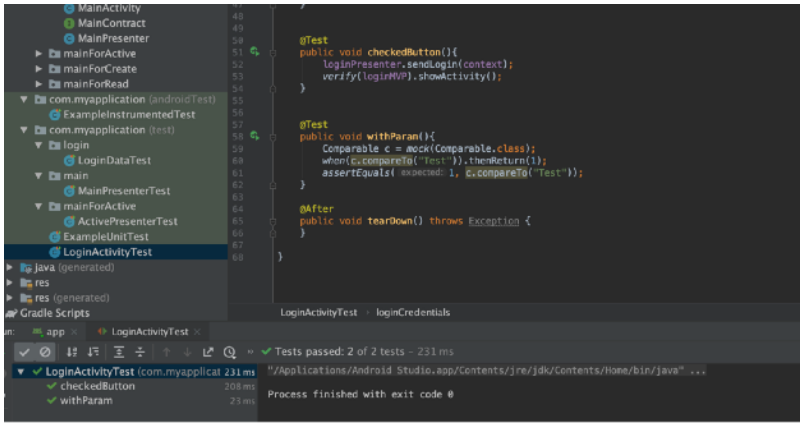




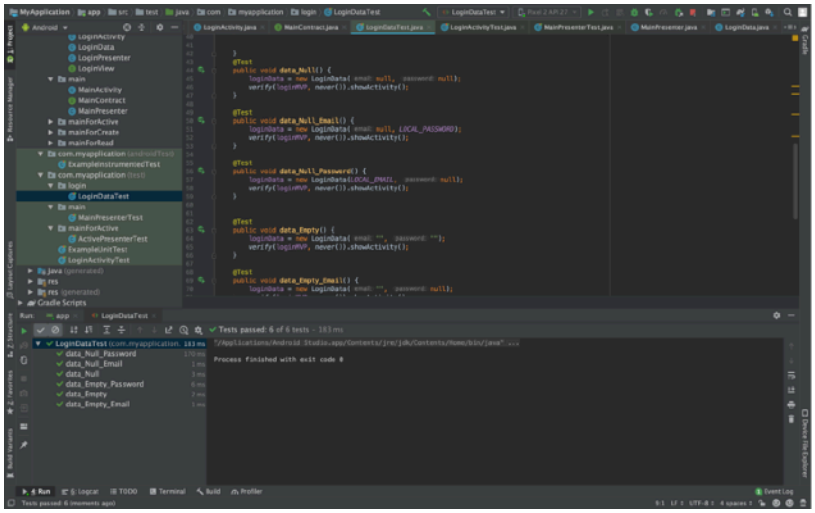
Фінальні тести

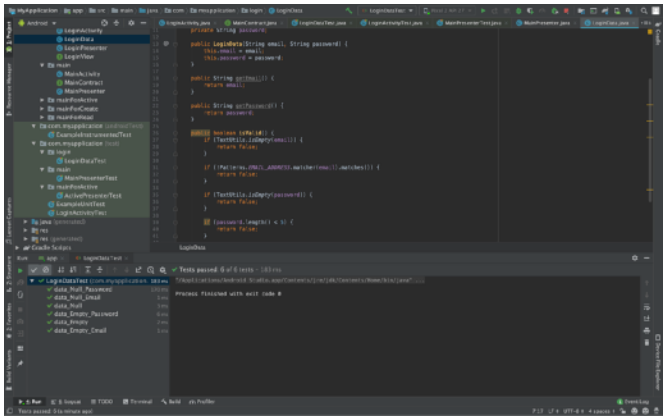
На жаль, для отримання потрібного функціоналу внесено зміни до архітектури додатку.

Тести - реалізація

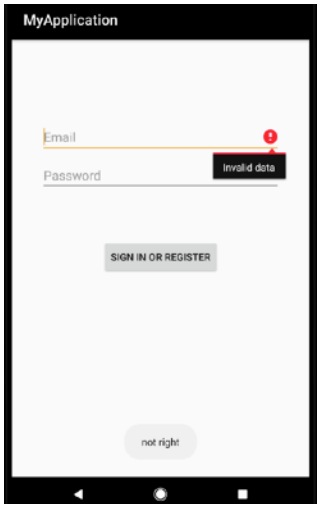


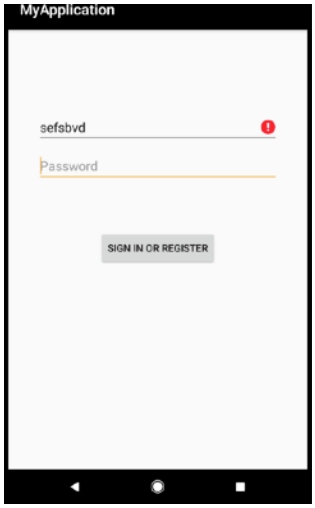
```
public class LoginPresenter implements LoginView.loginPresenter {

    private LoginView.loginMVP mvpView;

    public LoginPresenter(LoginView.loginMVP view) { mvpView = view; }

    @Override
    public void sendLogin(View view) {
        mvpView.showActivity();
    }
```

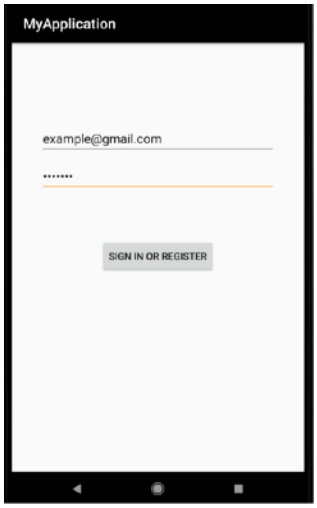




Коректні дані : email - anyEmail@gmail.com, password - 12345 (>5)







Висновки до лабораторних робіт 4 та 5.

За ці дві роботи був створений інтерфейс та частковий функціонал android-додатоку “The personal mini-library “The burden of the human passion””, а саме стоворення нового запису та відображення відповідних книг у своєму модулі. Розробка проводиться на віртуальному пристрої *Android 8.1, Pixel 2 API 27, 1080x1920 : 420 dpi,* мова програмування - Java 8, база даних - Firebase Real Time, тести - Mockito na Unit.

Також була спроба розробки за методологією TDD декількох модулів. Тести та сам принцип розробки - цікава та потрібна частина в життєвому циклі ПО. Це допомагає запобігти найбільш поширених помилок та багів, збільшивши час на розробку.