

# Programowanie Urządzeń Mobilnych

## Laboratorium

### LISTA 3

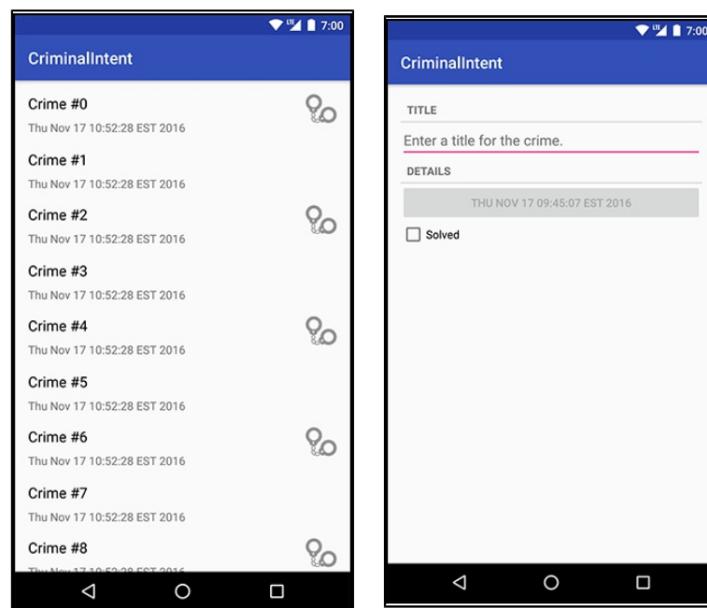
Rafał Lewandków

termin oddania: 03.12.2021

#### Zadanie 1 - 10 pkt

##### StudentCrimeApp (Java/Kotlin)

Aplikacja składa się z dwóch aktywności - aktywność główna zawiera listę wszystkich przewinień, po kliknięciu na element listy następuje przejście na drugą aktywność zawierającą bardziej szczegółowe informacje.



Rysunek 1: Layout aplikacji

- Przygotuj model danych
  - Klasa **Crime** jest modelem danych:

```
public class Crime {
```

```

        private UUID mId;
        private String mTitle;
        private Date mDate;
        private boolean mSolved;
    }

```

- Klasa **CrimeLab** jest źródłem danych - jest to **Singleton** zawierający kolekcję wszystkich przewinień

```

public class CrimeLab {
    private static CrimeLab sCrimeLab;

    private List<Crime> mCrimes;

    public static CrimeLab get(Context context) {
        if (sCrimeLab == null) {
            sCrimeLab = new CrimeLab(context);
        }

        return sCrimeLab;
    }

    private CrimeLab(Context context) {
        mCrimes = new ArrayList<>();
        for (int i = 0; i < 50; i++) {
            Crime crime = new Crime();
            crime.setTitle("Crime #" + i);
            crime.setSolved(i % 2 == 0);
            mCrimes.add(crime);
        }
    }

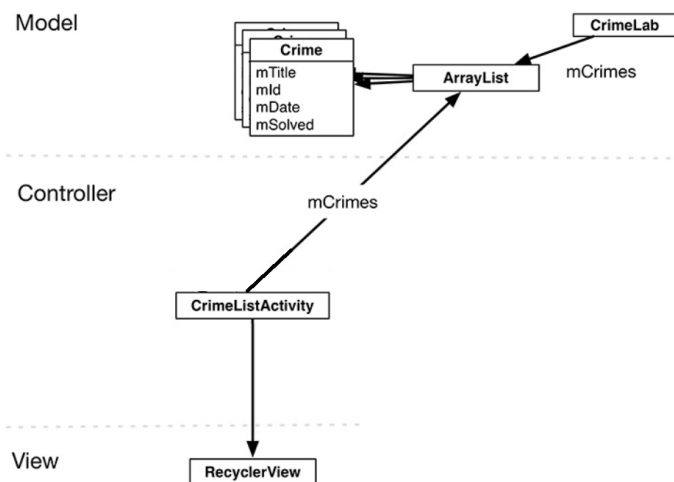
    public List<Crime> getCrimes() {
        return mCrimes;
    }

    public Crime getCrime(UUID id) {
        for (Crime crime : mCrimes) {
            if (crime.getId().equals(id)) {
                return crime;
            }
        }

        return null;
    }
}

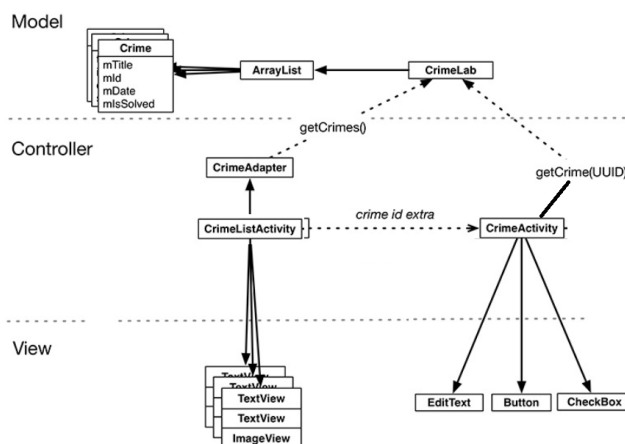
```

- Dodaj do aplikacji **RecyclerView** wyświetlający listę wszystkich elementów kolekcji **List<Crime> mCrimes** – **3 pkt**
- Schemat aplikacji



Rysunek 2: Schemat

- Dodaj obsługę **onClick** elementów listy, który będzie włączał aktywność **CrimeActivity** z widokiem szczegółowym elementu (Rys. 1) – **1 pkt**
- Schemat



Rysunek 3: Schemat

- Przy przejściu do **CrimeActivity** przekazywany jest jeden parametr **crimeId**, z poziomu **CrimeActivity** wywoływana jest metoda **getCrime(UUID)** zwracająca obiekt **Crime** o zadanym identyfikatorze – **1 pkt**
- **CrimeActivity** umożliwia edycję pól, przy powrocie do **CrimeListActivity** lista jest odświeżana i dane aktualizowane – **1 pkt**

- Do layoutu widoku szczegółowego dodaj przycisk usuwający aktualny wpis – **1 pkt**
- Dodaj przycisk umożliwiający dodanie nowego wpisu – **1 pkt**
- Dodaj **DatePickerFragment** do aplikacji umożliwiający wybranie daty przewinienia – **1 pkt**
- Dodaj możliwość wybrania czasu przewinienia – **1 pkt**

## OCENY

Maksymalna liczba punktów: 10

Oceny:

5.0 - 10 pkt

4.5 - 9 pkt

4,0 - 8 pkt

3,5 - 7 pkt

3,0 - 6 pkt