



**Instytut Informatyki
Kolegium Nauk Przyrodniczych
Uniwersytet Rzeszowski**

Przedmiot:

Programowanie urządzeń mobilnych

Dokumentacja techniczna projektu:

SnapGallery

Wykonał: Szymon Całka, Artur Pas

Prowadzący: mgr inż. Adam Szczur

Rzeszów 2023

1. Temat projektu i nazwa aplikacji

Tematem projektu jest zaprojektowanie aplikacji o nazwie „SnapGallery” która będzie służyć do przeglądania, organizowania i tworzenia zdjęć oraz wideo.

2. Cel projektu

Celem projektu jest opracowanie aplikacji o nazwie „SnapGallery”, kompatybilnej z systemem Android w wersji 10 i nowszymi, która umożliwi użytkownikom intuicyjne i bezpieczne przeglądanie i zarządzanie swoimi zbiorami zdjęć oraz filmów na urządzeniach mobilnych.

2.1. Cele szczegółowe

- **Zarządzanie multimediami:** Aplikacja zapewni narzędzia do przeglądania, organizowania i edycji kolekcji zdjęć i filmów, aby użytkownicy mogli efektywnie zarządzać swoimi multimediami.
- **Robienie zdjęć oraz wideo:** Aplikacja umożliwi dostęp do aparatu, z latarką który będzie pozwalał na robienie zdjęć lub nagrywanie wideo oraz ich zapisanie lub udostępnienie.
- **Zabezpieczenie multimediiów:** Aplikacja będzie oferować funkcję ukrycia multimediiów, gwarantując prywatność użytkowników.
- **Prosty i minimalistyczny interfejs:** Interfejs użytkownika zostanie zaprojektowany w taki sposób, aby był prosty i minimalistyczny, co pozwala użytkownikom skupić się na treści multimedialnej.

3. Funkcjonalności aplikacji

- **Przeglądanie multimediiów:** Dzięki tej funkcji użytkownicy będą mogli przeglądać swoje kolekcje zdjęć i filmów w sposób intuicyjny i przyjazny.
- **Zabezpieczenie multimediiów:** Użytkownicy będą mogli chronić swoje prywatne multimedia, zapewniając im bezpieczeństwo za pomocą hasła lub czytnika linii papilarnych.
- **Obsługa gestów:** Aplikacja umożliwi intuicyjne nawigowanie i interakcję z multimediami za pomocą gestów, co zwiększa wygodę korzystania z aplikacji.

- **Udostępnianie multimediów:** Aplikacja pozwoli na udostępnianie wybranych multimediów na platformy społecznościowe lub innym użytkownikom.
- **Dostęp do aparatu:** Aplikacja będzie zawierać wbudowany aparat do nagrywania filmów oraz robienia zdjęć
- **Korzystanie z latarki:** Aparat aplikacji pozwoli użytkownikom używać latarki w celu doświetlenia sceny podczas robienia zdjęć w warunkach niskiego oświetlenia. Będzie mógł ją aktywować za pomocą przycisku albo potrząśnięcia telefonem.

4. Technologie

- Kotlin 1.9.10
- Android 10 oraz nowsze wersje
- Środowisko programistyczne Android Studio Hedgehog 2023.1.1 Patch 2

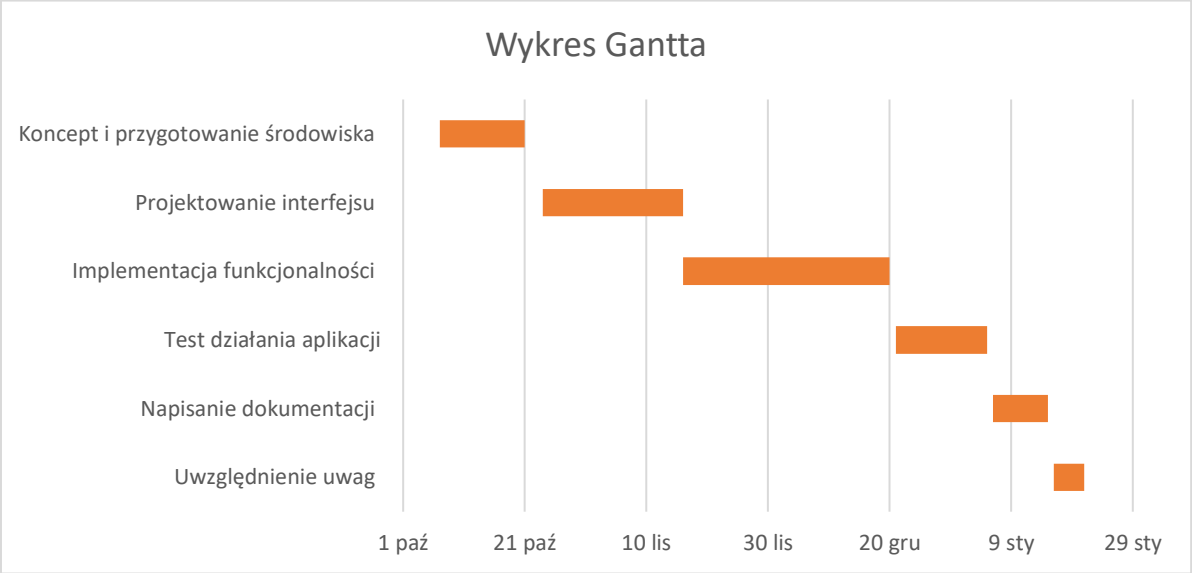
5. Interesariusze aplikacji

Głównym odbiorcą aplikacji będą zwykli użytkownicy smartfonów z systemem android w wersji 10 wzwyż, którzy cenią sobie wielofunkcyjność w jednej aplikacji oraz własną prywatność.

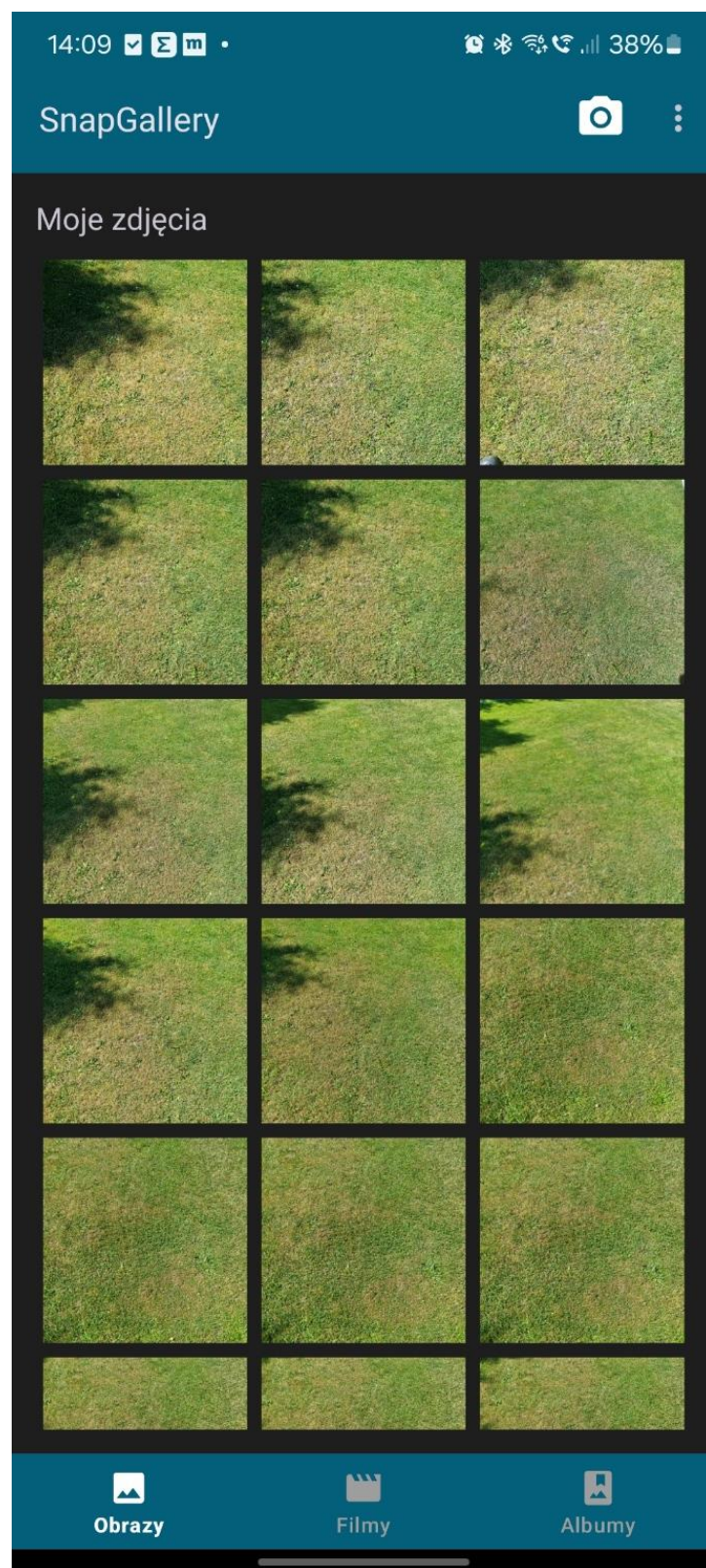
6. Harmonogram realizacji projektu

- Koncept i przygotowanie środowiska (od 07.10.2023 do 21.07.2023)
- Projektowanie interfejsu (od 24.10.2023 do 15.11.2023)
- Implementacja funkcjonalności (od 16.11.2023 do 20.12.2023)
- Test działania aplikacji (od 21.12.2023 do 05.01.2023)
- Napisanie dokumentacji (od 06.01.2024 do 15.01.2023)
- Uwzględnienie uwag (od 16.01.2023 do 21.01.2023)

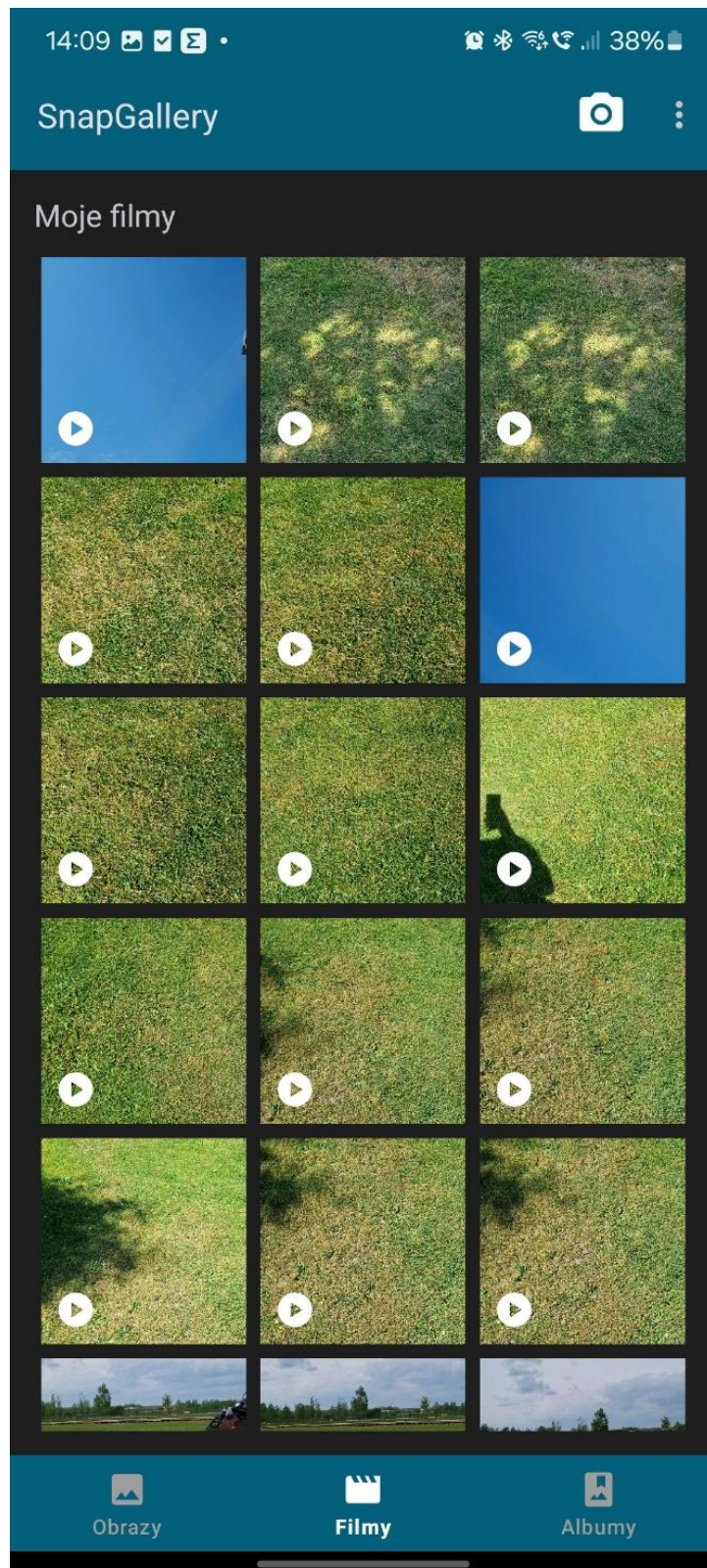
6.1. Wykres Gantta



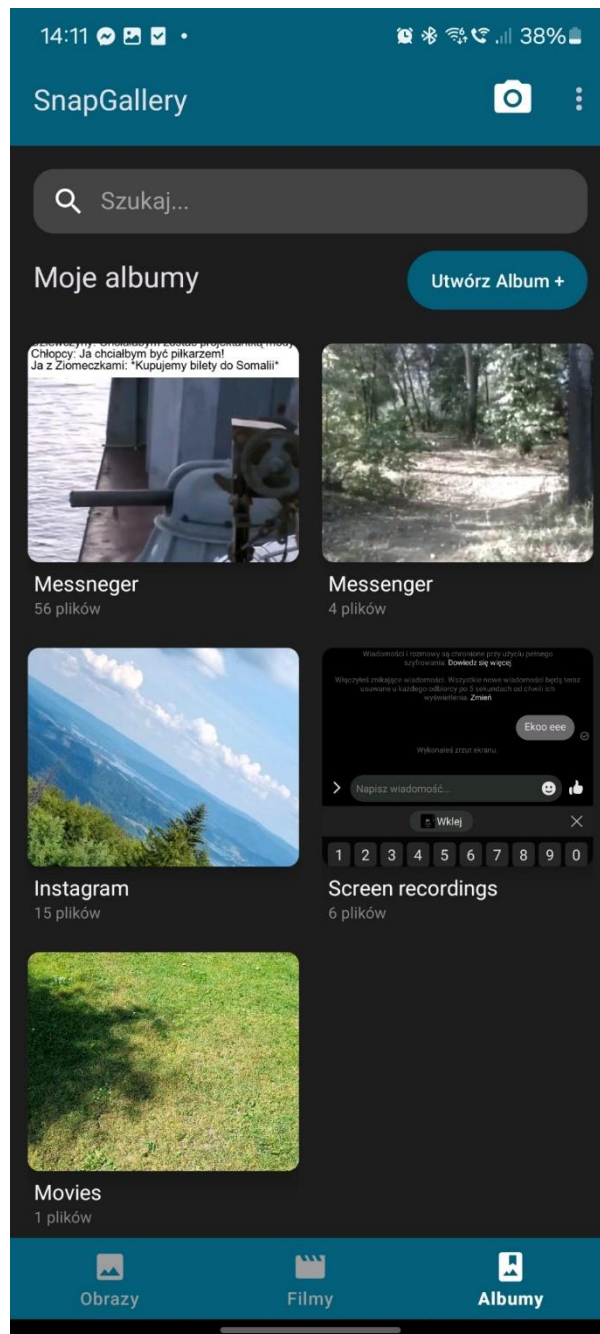
7. Projekt GUI



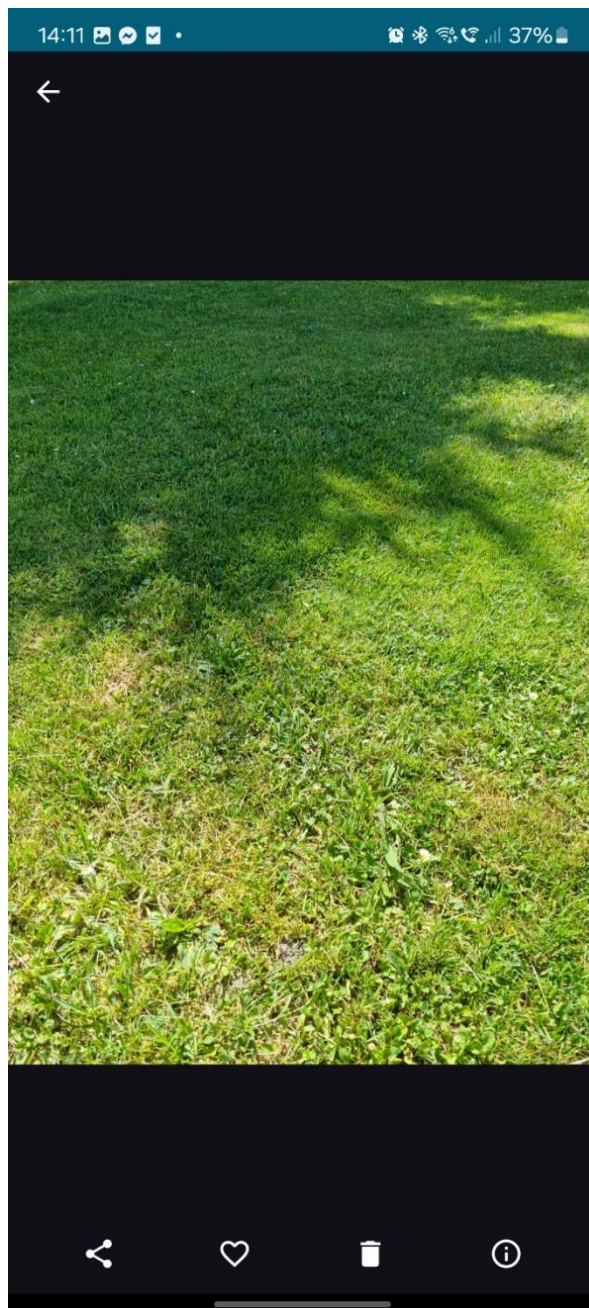
Fragment wyświetlający wszystkie zdjęcia



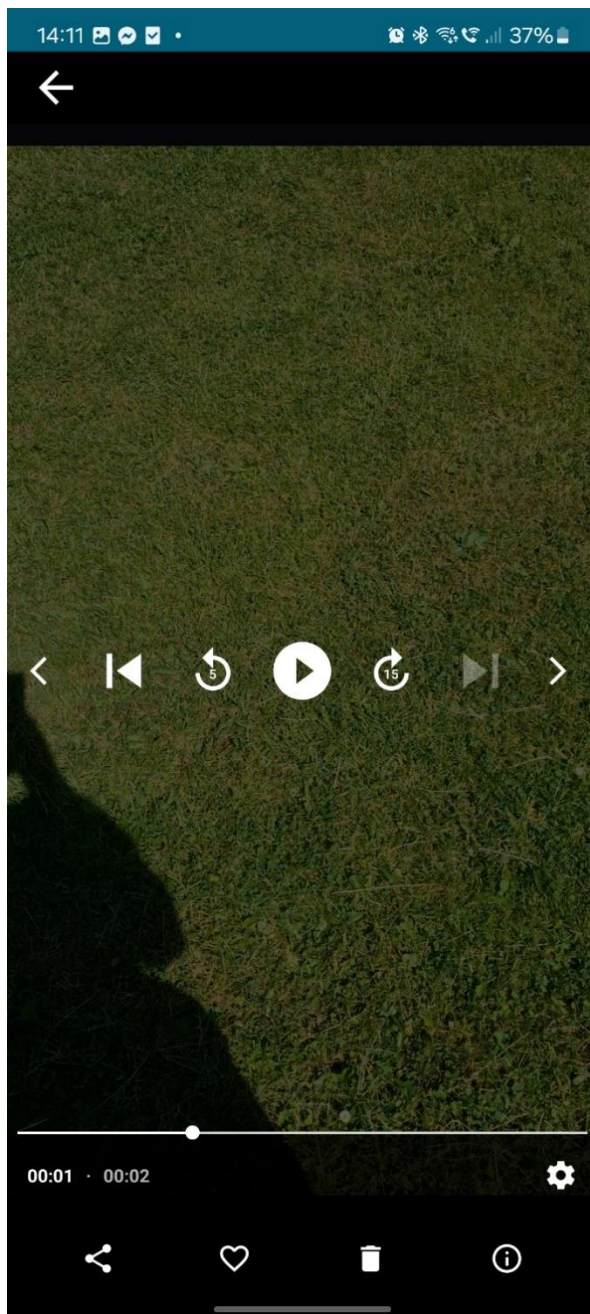
Fragment wyświetlający wszystkie filmy



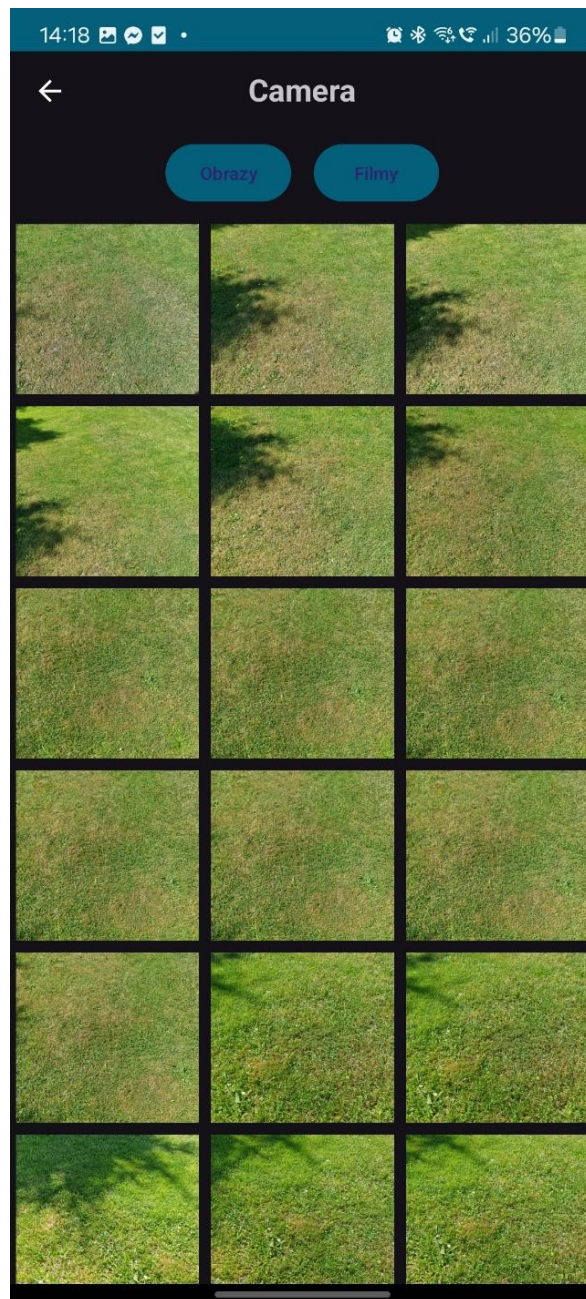
Fragment wyświetlający wszystkie albumy



Wyświetlanie zdjęcia



Odtwarzanie filmu



Wyświetlanie zawartości albumu



Widok aparatu

8. Struktura programu

Struktura programu składa się z kilku kluczowych komponentów.

- Pakiet activity: Tutaj znajdują się wszystkie aktywności potrzebne do wykonywania akcji w programie. Aktywności są odpowiedzialne za interakcję z użytkownikiem i zarządzanie przepływem aplikacji.

- **Pakiet adapters:** Znajdują się tutaj adaptery, które są odpowiedzialne za dostarczanie danych do widoków w interfejsie użytkownika. Adaptery są używane w różnych częściach aplikacji do wyświetlania danych w formie odpowiedniej dla danego widoku.
- **Pakiet fragments:** W tym katalogu znajdują się fragmenty, które są odpowiedzialne za wyświetlanie różnych widoków na ekranie w zależności od tego, który przycisk kliknie użytkownik. Fragmenty są często używane w aplikacjach Android do modularyzacji interfejsu użytkownika i zarządzania jego częściami.
- **Pakiet models:** Tutaj znajdują się modele danych, które definiują strukturę i właściwości danych używanych w aplikacji. Modele są używane do reprezentowania różnych obiektów i informacji w programie.
- **Repozytoria:** Repozytoria są odpowiedzialne za definiowanie interfejsów dostępu do danych w aplikacji. Mogą one obejmować operacje odczytu, zapisu, aktualizacji i usuwania danych z różnych źródeł, takich jak baza danych, usługi sieciowe, pamięć podręczna itp. Repozytoria pomagają oddzielić logikę dostępu do danych od reszty aplikacji i ułatwiają testowanie i utrzymanie kodu.

Równie ważny jest folder `res` zawierający zasoby niezbędne do działania aplikacji.

Najważniejsze z nich to:

- `color`: kolory używane w aplikacji
- `drawable`: ikony i obrazki w aplikacji
- `layout`: opis układów elementów GUI
- `values`: wartości przypisane do kolorów i tekstu używanego w programie

Ta struktura programu pomaga w organizacji kodu i zapewnieniu czytelności oraz łatwości zarządzania projektem. Każdy komponent ma swoje własne zadania i odpowiedzialności, co ułatwia rozwój i utrzymanie aplikacji w przyszłości.

8.1. Dane wykorzystywane przez program

Aplikacja SnapGallery wykorzystuje przede wszystkim dane, takie jak zdjęcia i filmy oraz łączy je w albumy pozwalając na ich łatwe przeglądanie.

8.2. Opis plików zewnętrznych

Aplikacja nie używa żadnych plików zewnętrznych. wszelkie dane, jakimi się posługuje znajdują się wewnątrz aplikacji lub w pamięci telefonu

8.3. Podział na moduły, komunikacja między modułami

Jedynym modułem aplikacji jest automatycznie wygenerowany moduł „app”.

```
dependencies { this: DependencyHandlerScope
    implementation("androidx.biometric:biometric:1.2.0-alpha05")
    implementation("androidx.biometric:biometric-ktx:1.2.0-alpha05")
    implementation("androidx.camera:camera-core:1.3.1")
    implementation("androidx.camera:camera-camera2:1.3.1")
    implementation("androidx.camera:camera-lifecycle:1.3.1")
    implementation("androidx.camera:camera-video:1.3.1")
    implementation("androidx.camera:camera-view:1.3.1")
    implementation("androidx.camera:camera-extensions:1.3.1")
    implementation("androidx.core:core-ktx:1.12.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.11.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("com.squareup.picasso:picasso:2.71828")
    implementation("com.github.bumptech.glide:glide:4.12.0")
    implementation("androidx.activity:activity-ktx:1.8.2")
    implementation("androidx.fragment:fragment-ktx:1.6.2")
    implementation("androidx.cardview:cardview:1.0.0")
    implementation("androidx.viewpager2:viewpager2:1.0.0")
    implementation("androidx.media3:media3-session:1.2.1")
    implementation("androidx.media3:media3-exoplayer:1.2.1")
    implementation("androidx.media3:media3-exoplayer-dash:1.2.1")
    implementation("androidx.media3:media3-ui:1.2.1")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
    annotationProcessor("com.github.bumptech.glide:compiler:4.12.0")
}
```

Komunikacja między klasami odbywa się bezpośrednio np. poprzez utworzenie obiektu jakiejś klasy np. adaptera

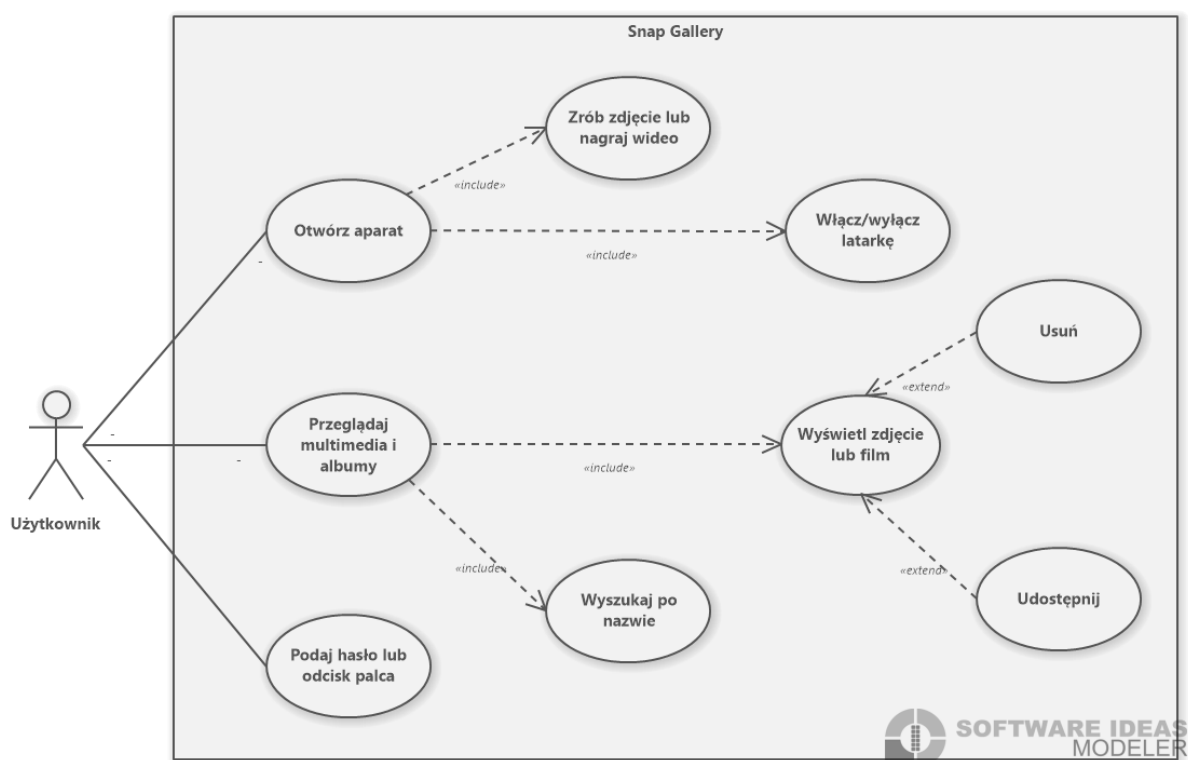
```
val adapter = FullScreenImageAdapter(context: this, ImageRepository.images)
```


Lub poprzez Intent w przypadku aktywności

```
val intent = Intent( packageContext: this, CameraActivity::class.java)
this.startActivity(intent)
```

9. Diagramy UML

9.1. Diagram przypadków użycia



Definicja aktorów

AKTOR: Użytkownik

OPIS: Użytkownik jest osobą mającą dostęp do aplikacji po podaniu hasła lub odcisku palca, może on przeglądać zdjęcia, filmy i albumy oraz otworzyć aparat w celu utworzenia nowego zdjęcia lub wideo

PRZYPADKI UŻYCIA:

- PU Otwórz aparat powiązane:
 - przez <<include>> z PU Włącz/wyłącz latarkę
 - przez <<include>> z PU Zrób zdjęcie lub nagraj wideo

- PU Przeglądaj multimedia i albumy powiązane:
 - przez <<include>> z PU Wyświetl zdjęcie lub film powiązane:
 - przez <<extend>> z PU Udostępnij
 - przez <<extend>> z PU Usuń
 - przez <<extend>> z PU Sortuj
 - przez <<extend>> z PU Usuń
- PU Podaj hasło lub odcisk palca

Definicje scenariuszy przypadków użycia (PU — przypadek użycia, WS - warunki wstępne, WK - warunki końcowe)

PU Włącz/wyłącz latarkę

OPIS

CEL: Włączenie lub wyłączenie latarki

WS: Użytkownik musi mieć otwarty aparat

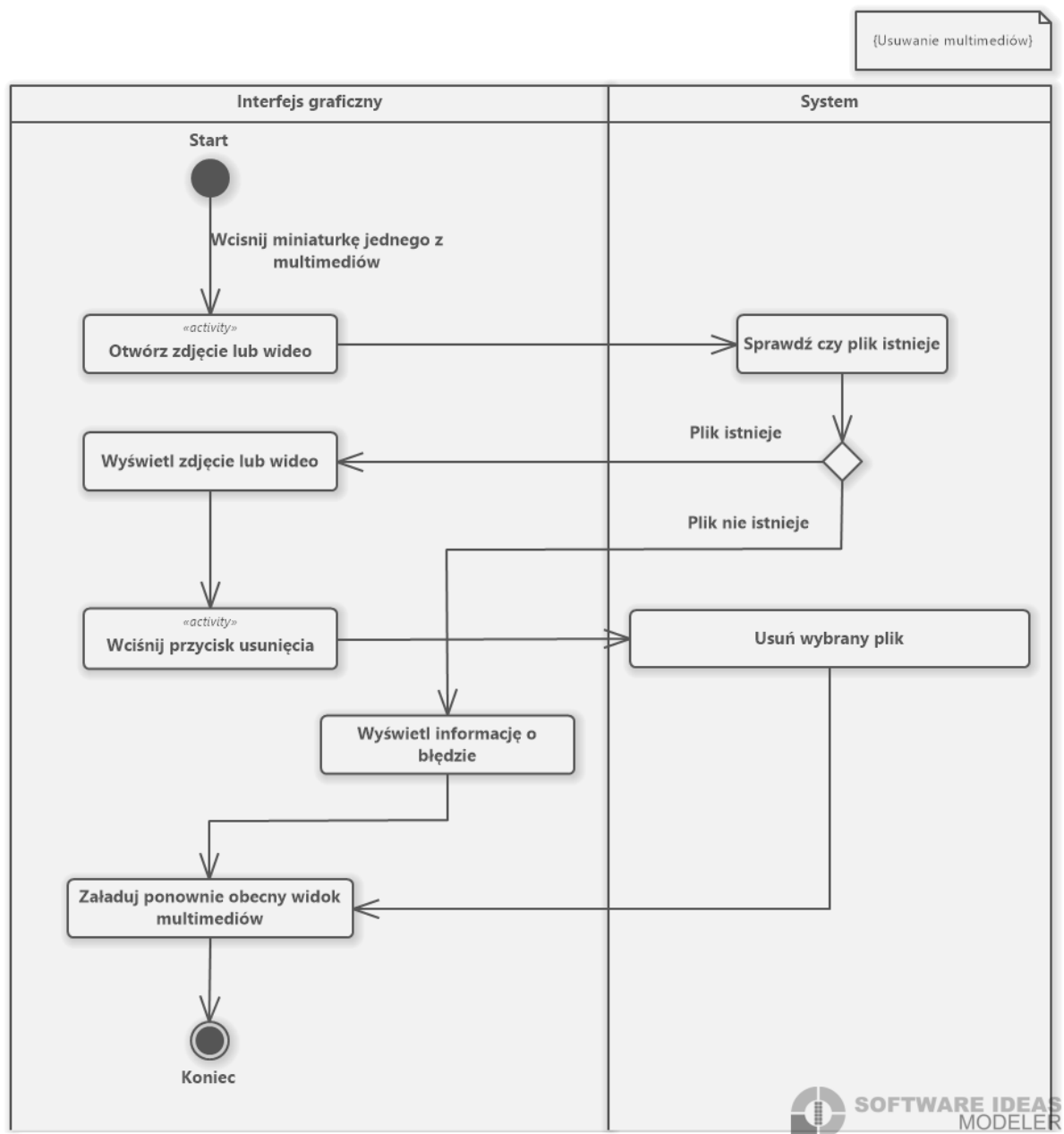
WK: Wciśnięcie przycisku z ikoną latarki.

PRZEBIEG:

Włączenie latarki jest dostępne poprzez włączenie latarki i naciśnięcie przycisku z ikoną błyskawicy w celu doświetlenia sceny. Jeśli latarka jest włączona wciśnięcie przycisku wyłączy latarkę.

9.2. Diagram czynności / aktywności

Poniżej przedstawiony jest diagram aktywności „Usuwanie multimediiów”. Aby usunąć plik użytkownik musi nacisnąć miniaturkę jednego z multimediiów w otwartym oknie co spowoduje sprawdzenie czy plik istnieje i wyświetli się podgląd wciśniętego zdjęcia lub filmu albo błąd i otwarte okno zostanie odświeżone. Po otwarciu podglądu filmu lub obrazu, użytkownik może wcisnąć przycisk usuń aby wykasować plik z pamięci telefonu. Po usunięciu plików otwarty widok multimediiów jest odświeżany.



10. Literatura

Introduction to Android – <http://developer.android.com/guide/index.html>

Shane Conder, Lauren Darcey, Android. Programowanie aplikacji na urządzenia przenośne, Helion

Jeff Friesen, Java. Przygotowanie do programowania na platformę Android, Helion 2011

Gonera P.: Android w akcji. Wydanie II, Helion (2011)

Wildermuth S.: Podstawy Windows Phone 7.5. Projektowanie aplikacji przy użyciu Silverlight, APN Promise (2012)

Gail Rahn Frederick, Rajesh Lal: Projektowanie witryn internetowych dla urządzeń mobilnych, Helion 2012

Jacek Matulewski, Bartosz Turowski: Projektowanie aplikacji dla urządzeń mobilnych z systemem Windows Mobile, Helion 2010