

**Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie**

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki



DESIGN PATTERNS

**ANNA GNOIŃSKA, JULIA IGNACYK,
ANTON BOHATIUK, MICHAŁ PIENIĄDZ**

**IMPLEMENTACJA APLIKACJI MOBLINEJ DO
NAGRYWANIA I UDOSTĘPNIANIA VIDEO ORAZ
IMPLEMENTACJA CZĘŚCI SERWEROWEJ
REALIZUJĄCEJ PRZETWARZANIE CELEM
OPTIMALIZACJI JAKOŚCIOWEJ ORAZ
POZWALAJĄCEJ NA UDOSTĘPNIENIE
OKREŚLONEGO ZASOBU VIDEO PUBLICZNIE.**

Kraków 2020/2021

Spis treści

1	Wprowadzenie	3
1.1	Opis projektu	3
2	Założenia projektowe	4
2.1	Funkcjonalności i założenia	4
3	Organizacja projektu	5
3.1	High overview architecture	5
3.2	Schemat bazy danych	6
3.3	Diagram przepływu w aplikacji	6
3.4	Diagram klas aplikacji	8
3.5	Diagram klas części serwerowej	9
4	Wzorce projektowe	10
4.1	Fasada	10
4.2	Mapper	11
4.3	Builder	11
5	Specyfikacja dynamiki systemu	12
5.1	Diagram przypadków użycia aplikacji przez użytkownika	12
5.2	Logowanie się do aplikacji	12
5.3	Dodawanie filmu	13
5.4	Usuwanie filmu	14
5.5	Przegląd wszystkich filmów Story	14
5.6	Przegląd wszystkich filmów użytkownika	15
5.7	Wyświetlanie filmu	16

1. Wprowadzenie

1.1. Opis projektu

Celem projektu jest utworzenie z użyciem wzorców projektowych aplikacji mobilnej służącej do przeglądania oraz tworzenia filmów video. Aplikacja będzie umożliwiała przesłanie filmu na serwer, utworzenie swoich oraz publicznych filmów innych użytkowników.

2. Założenia projektowe

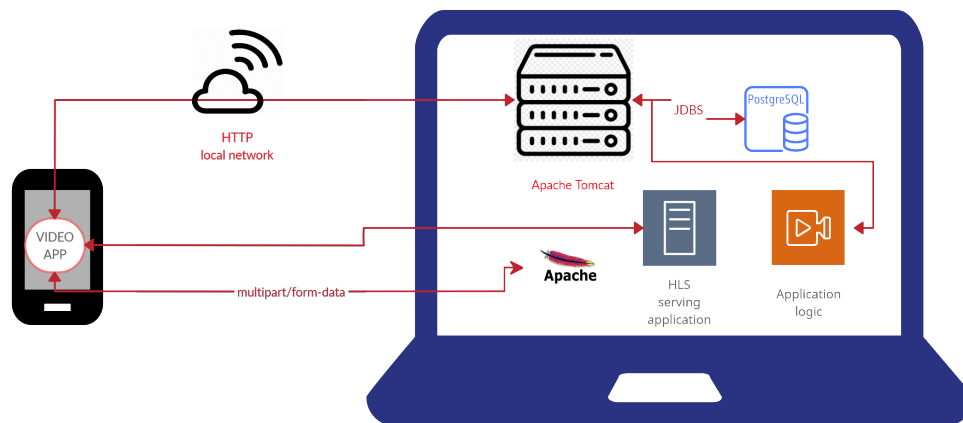
Aplikacja powinna działać podobnie do znanych portali social media umożliwiających nagranie filmiku (typu Instagram, Snapchat). Użytkownik po zalogowaniu ma możliwość nagrania video, a następnie po uprzedniej akceptacji zgody przesłania go do części serwerowej celem kompresji jakościowej. Implementacja części serwerowej realizująca wspomniane przetwarzanie pozwala również na udostępnienie określonego zasobu video publicznie. Użytkownik może wycofać zgodę lub usunąć materiał video.

2.1. Funkcjonalności i założenia

- Jako użytkownik mogę nagrać filmik oraz dodać go na serwer
- Jako użytkownik mogę udostępnić do publicznego odtwarzania za moją akceptacją filmik
- Jako użytkownik mogę oglądać swoje filmiki i innych użytkowników
- Jako użytkownik mogę usunąć materiał lub wycofać zgodę publicznego odtwarzania mojego filmiku
- Tylko zalogowani użytkownicy mogą korzystać z powyżej określonych funkcjonalności

3. Organizacja projektu

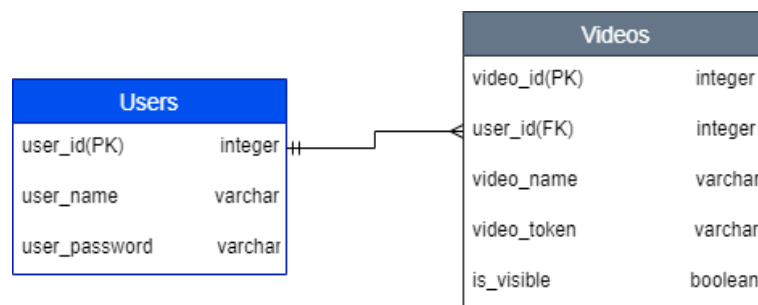
3.1. High overview architecture



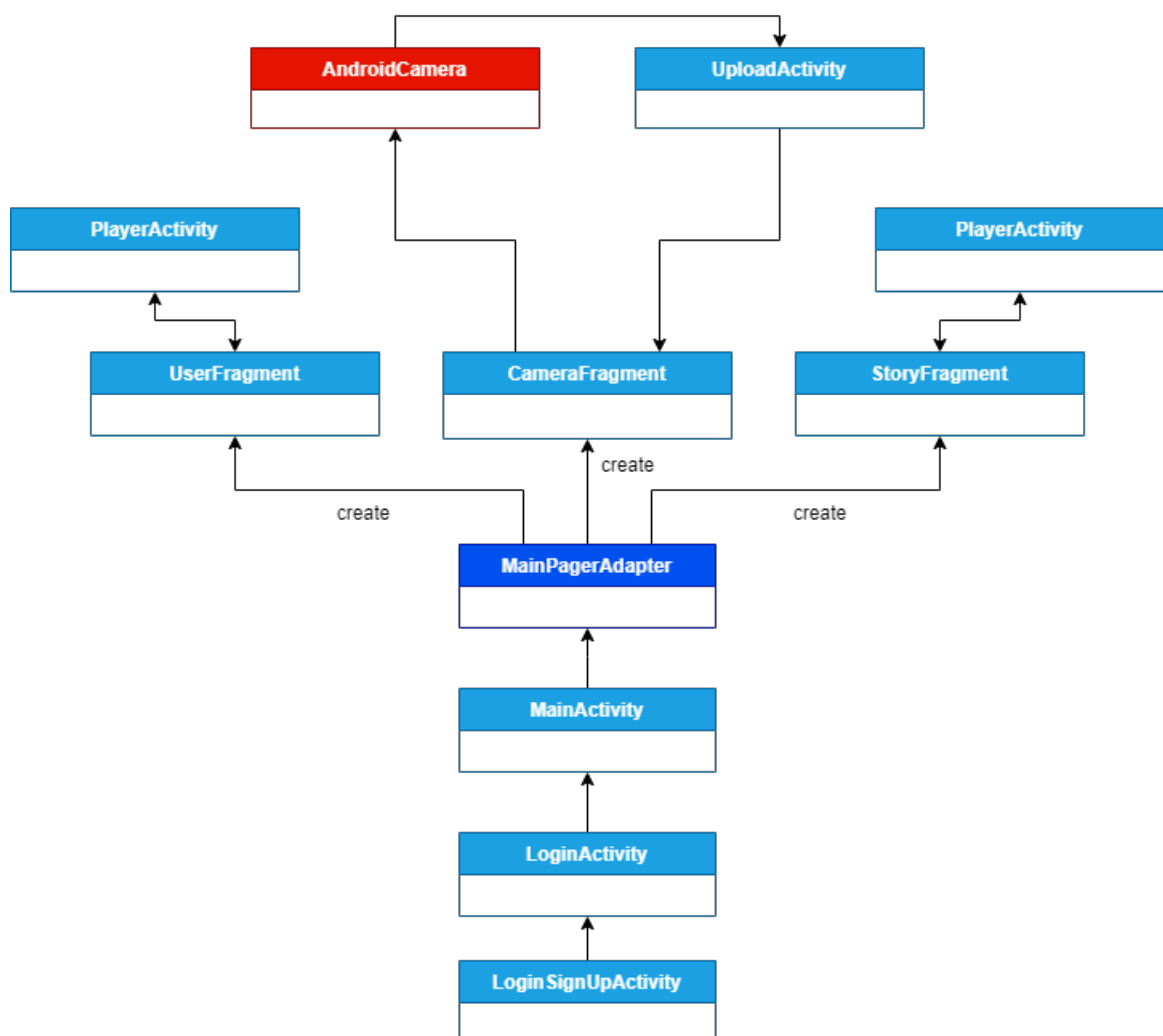
Aplikacja mobilna zostanie zaimplementowana w języku Java. Do procesu tworzenia wykorzystujemy IDE Android Studio, a projekt zostanie oparty na Gradle. Film z urządzenia mobilnego na serwer będzie dostarczony poprzez wysłanie plików w typie multipart/form-data na Apache. Filmy otwarzane z serwera będą za pomocą playera (na podstawie Exo-Player). Transmisja danych do playera będzie odbywać się w standardzie HLS.

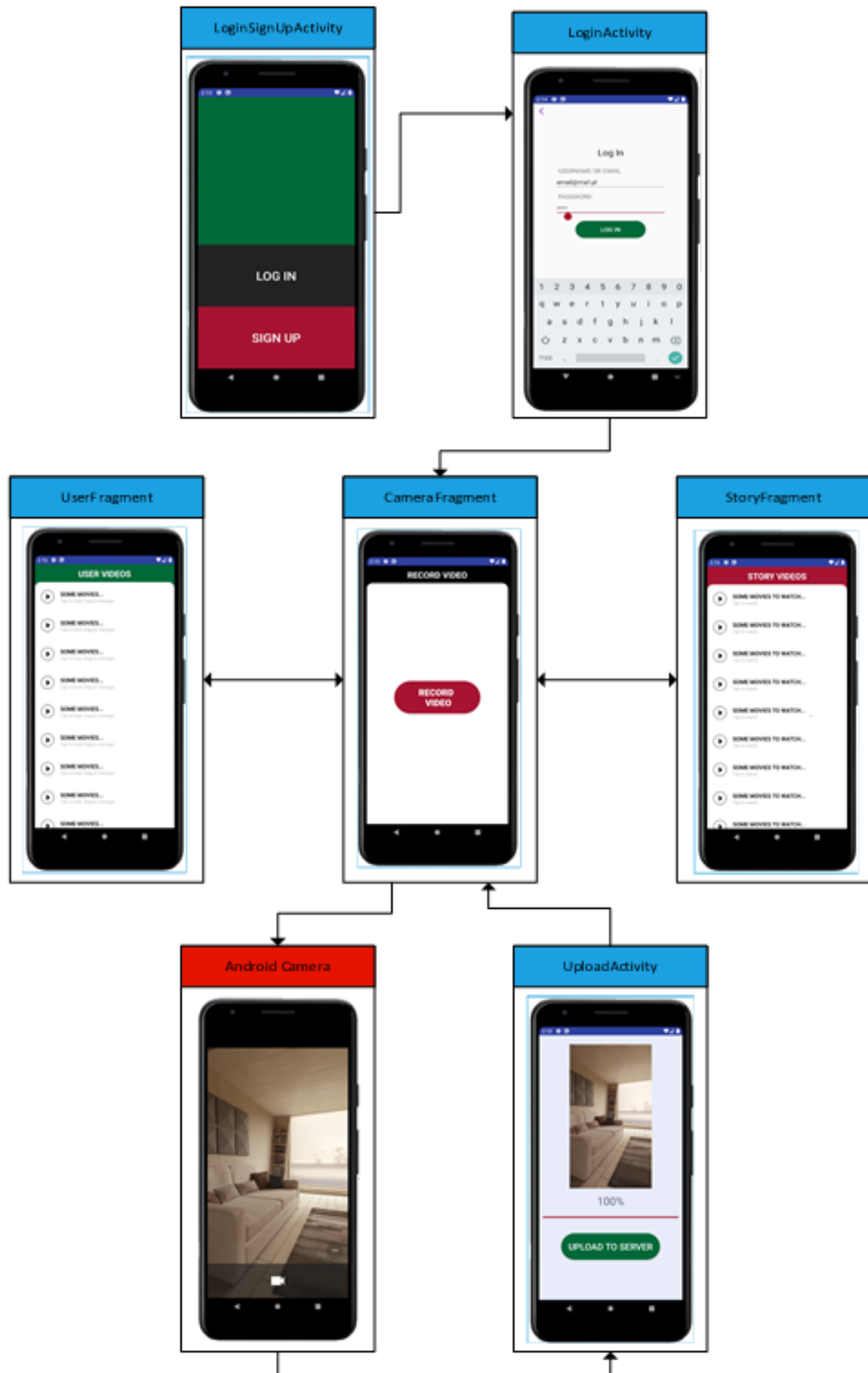
Serwer będzie postawiony na komputerze. Serwerem aplikacyjnym (webowym) będzie Apache Tomcat. Dodatkowo na serwerze będziemy kompresować nadsyłane video za pomocą biblioteki, a także łączyć się z bazą danych PostgreSQL za pomocą JDBC, posiadającą wiedzę na temat użytkowników i filmów. Pliki nadesłane od użytkowników będą przechowywane na partycji komputera, na której pracuje serwer. Za serwowanie plików odpowiada serwis napisany w języku GO (HLS serving application).

3.2. Schemat bazy danych

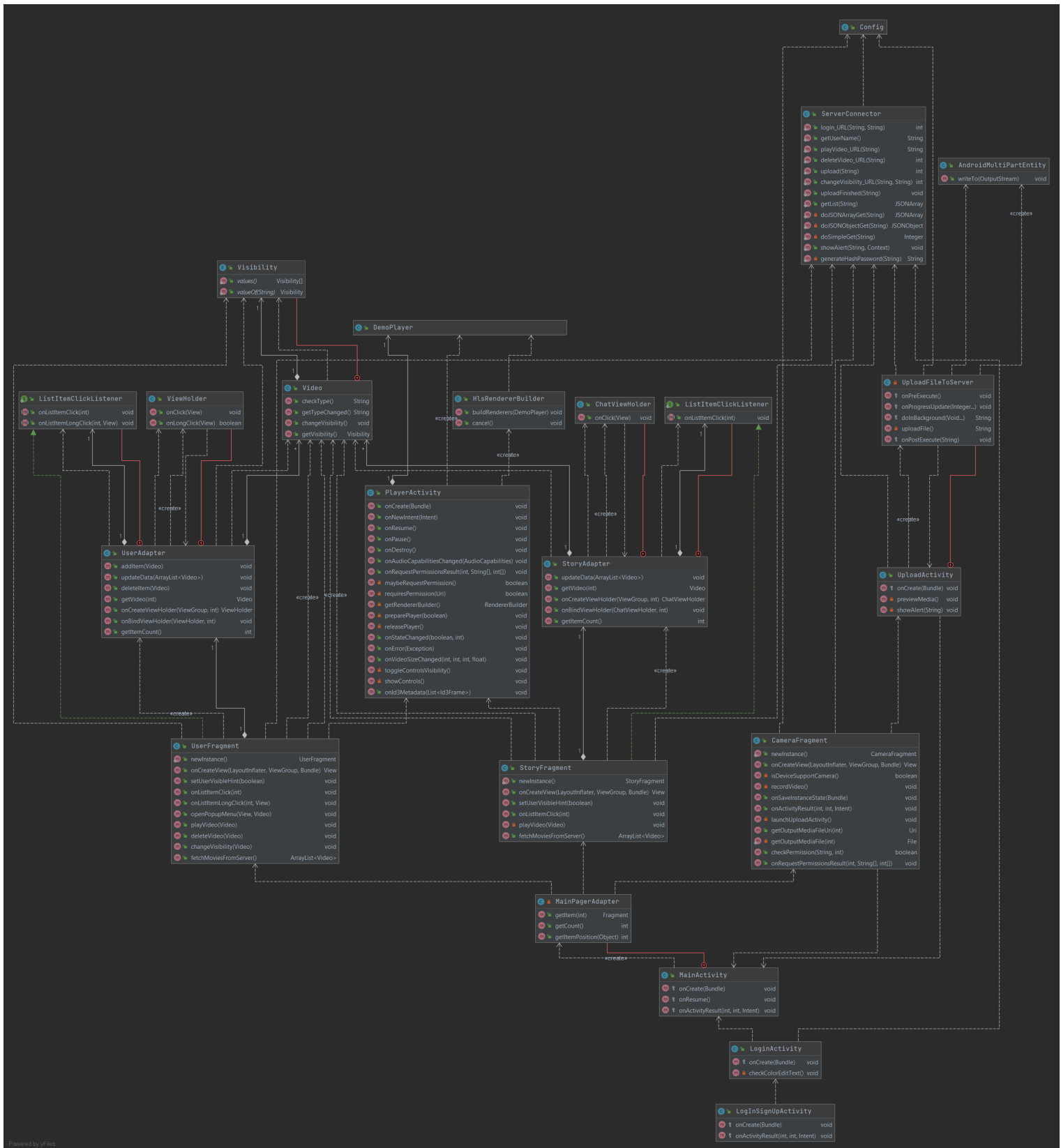


3.3. Diagram przepływu w aplikacji

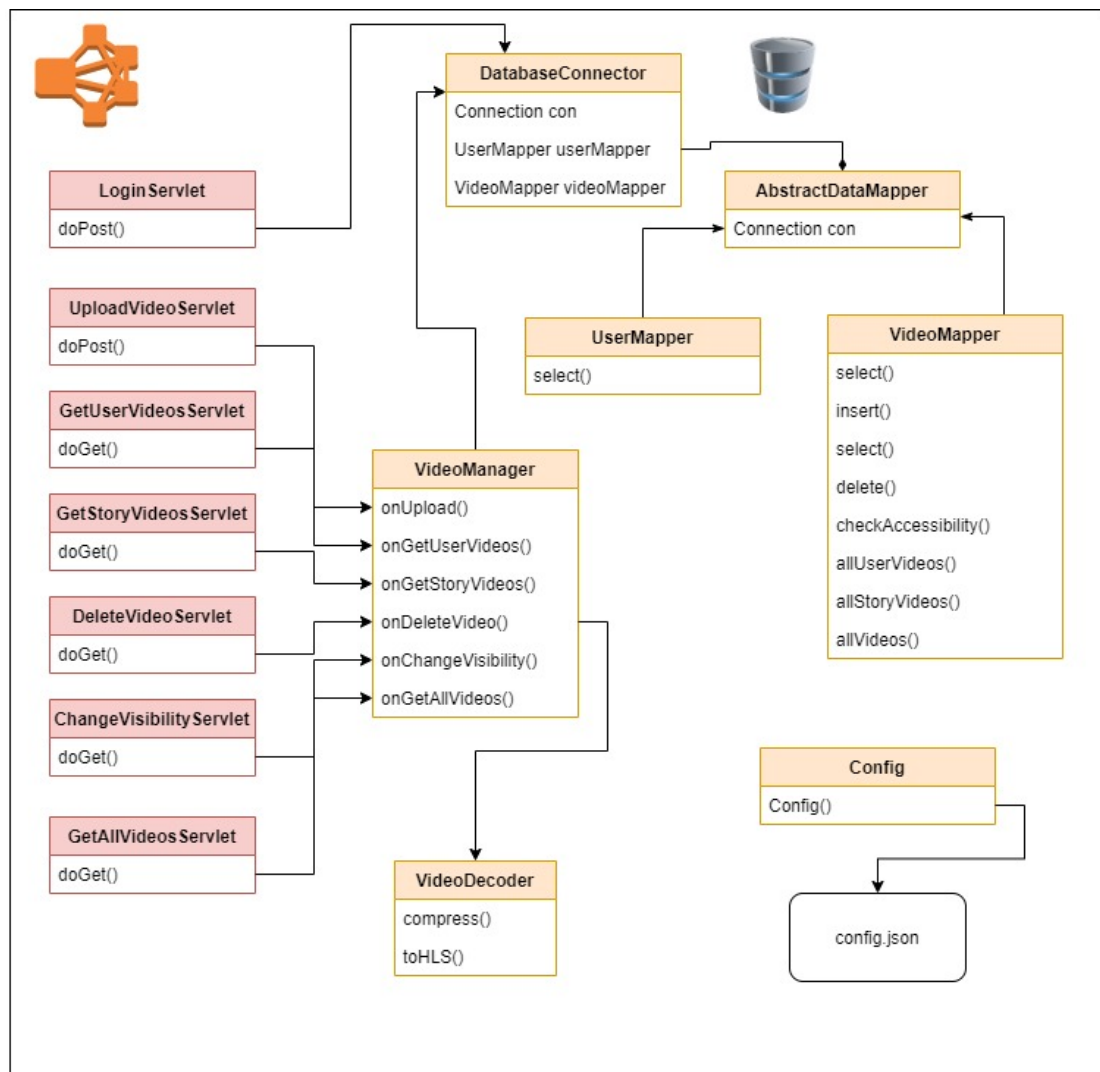




3.4. Diagram klas aplikacji



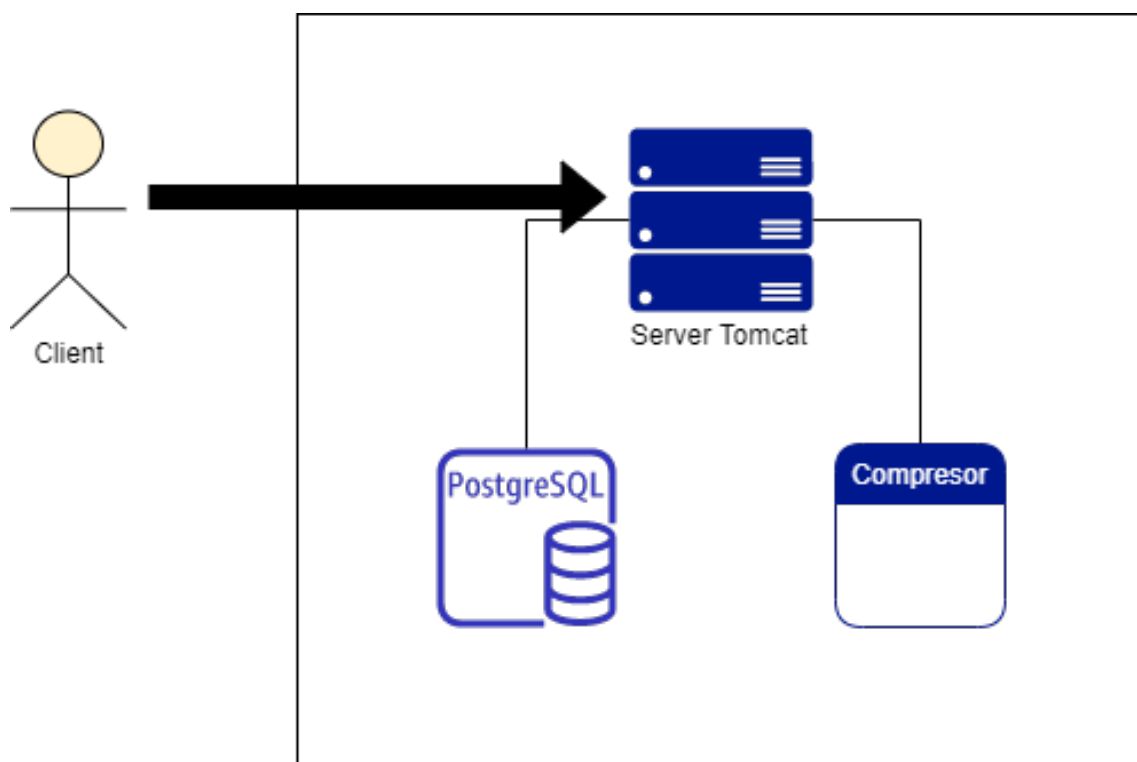
3.5. Diagram klas części serwerowej



4. Wzorce projektowe

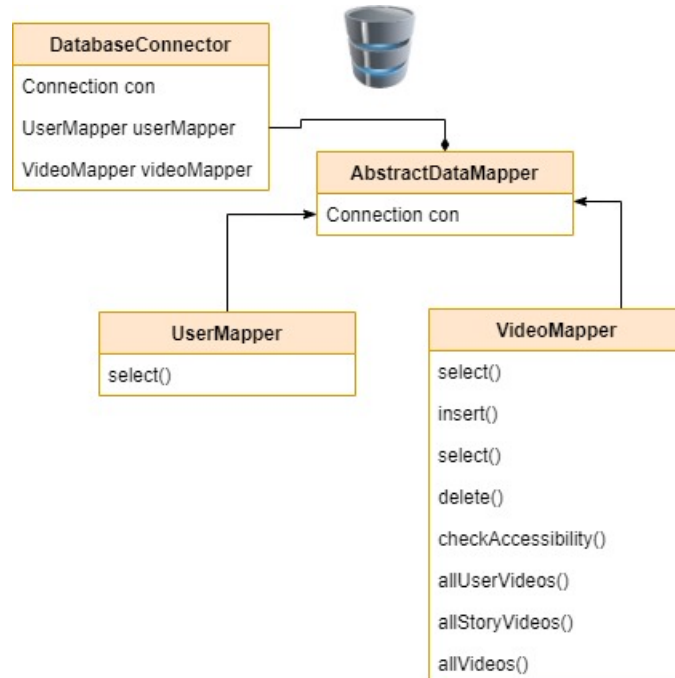
4.1. Fasada

Fasadą w naszym przypadku będzie zbiór webserwisów, które dla klienta będą wystawiać interfejs do całej aplikacji uruchomionej na tomcat'cie.



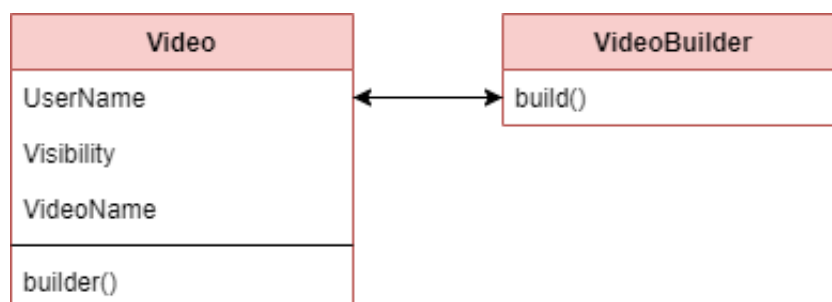
4.2. DataMapper

Wykorzystujemy mechanizm oddzielenia obiektów aplikacji od źródła danych (izolowanie warstw). W naszym przypadku jest to klasa DatabaseConnector.



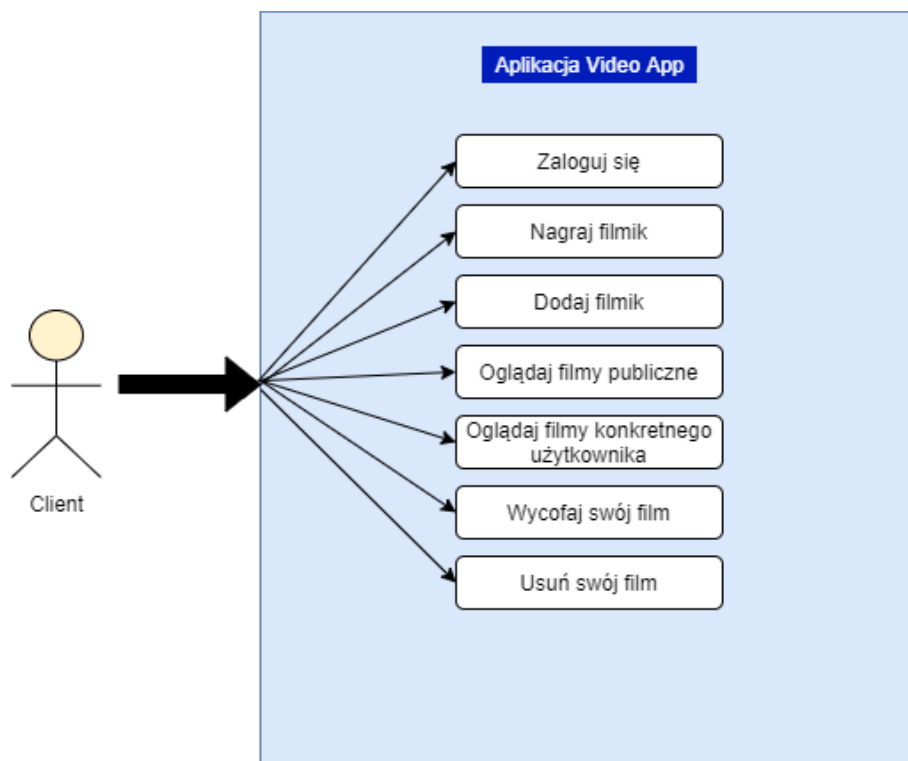
4.3. Builder

W naszym projekcie jest wykorzystywany do tworzenia obiektów klasy Video, która ma w sobie 3 pola (VideoName, UserName, Visibility), z myślą o tym, aby później w przypadku innych atrybutów video konstrukcja była czytelna i łatwo zmienialna, a także by nie wymagała tworzenia innych konstruktów.



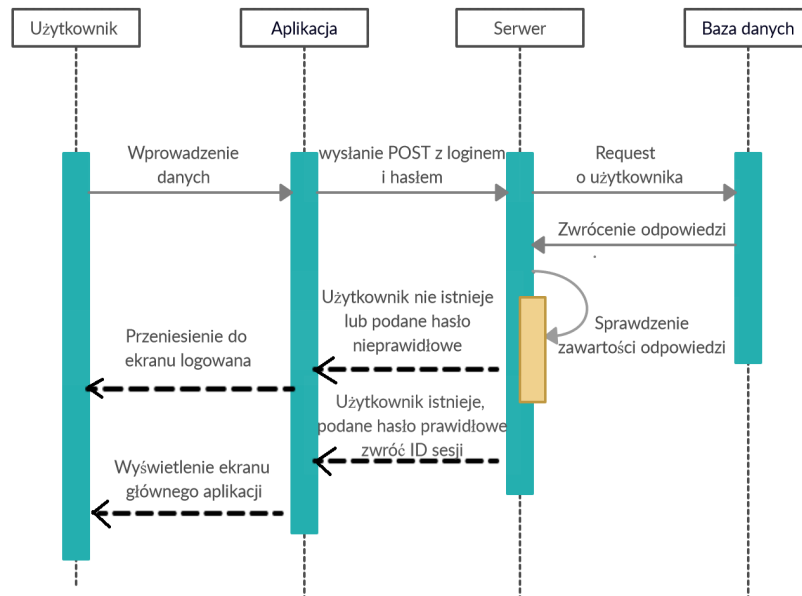
5. Specyfikacja dynamiki systemu

5.1. Diagram przypadków użycia aplikacji przez użytkownika



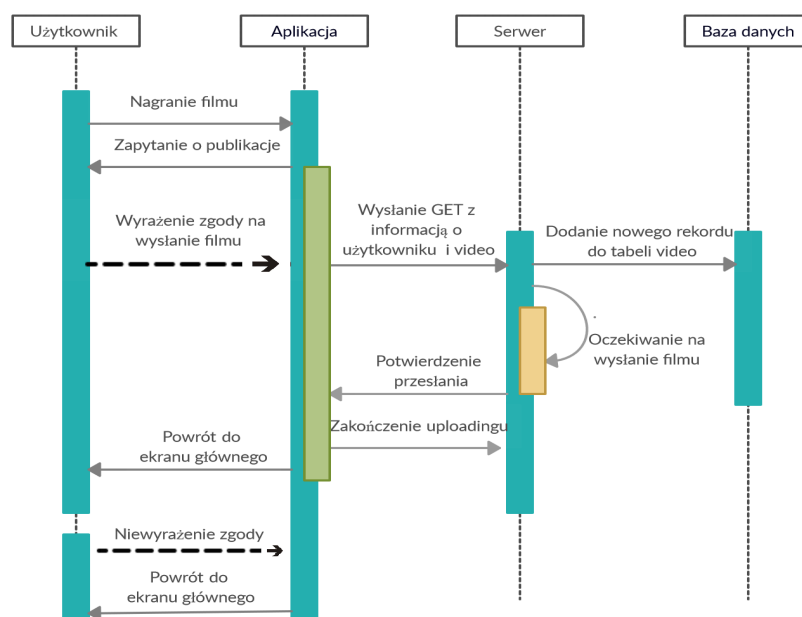
5.2. Logowanie się do aplikacji

Podczas logowania użytkownik podaje email i hasło, które zostaje przesłane na backend. Backend weryfikuje czy dany email jest widoczny w bazie danych oraz czy podane hasło jest prawidłowe. W zależności od otrzymanego komunikatu aplikacja wyświetla informację o błędzie lub ekran główny.



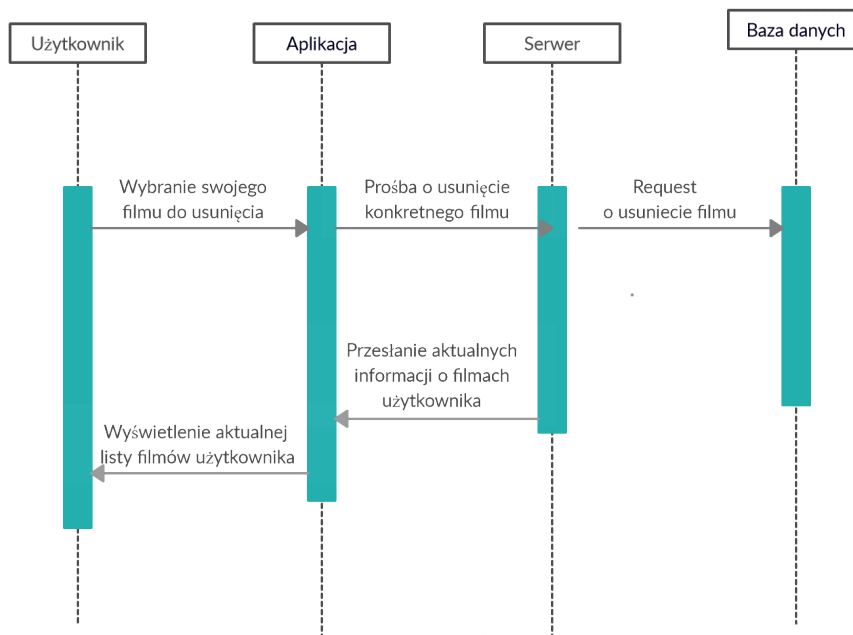
5.3. Dodawanie filmu

Użytkownik w aplikacji ma możliwość nagrania filmiku. Po wykonaniu tej czynności aplikacja pyta o chęć przesłania video. Jeśli klient wyrazi zgodę wysłany jest GET z informacją o użytkowniku oraz metadane filmu. Zostaje dodany nowy rekord w bazie danych (tabeli Video). Serwer potwierdza przesłanie, a aplikacja zaczyna uploading. W momencie niewyrażenia zgody na przesłanie nagrany film przepada.



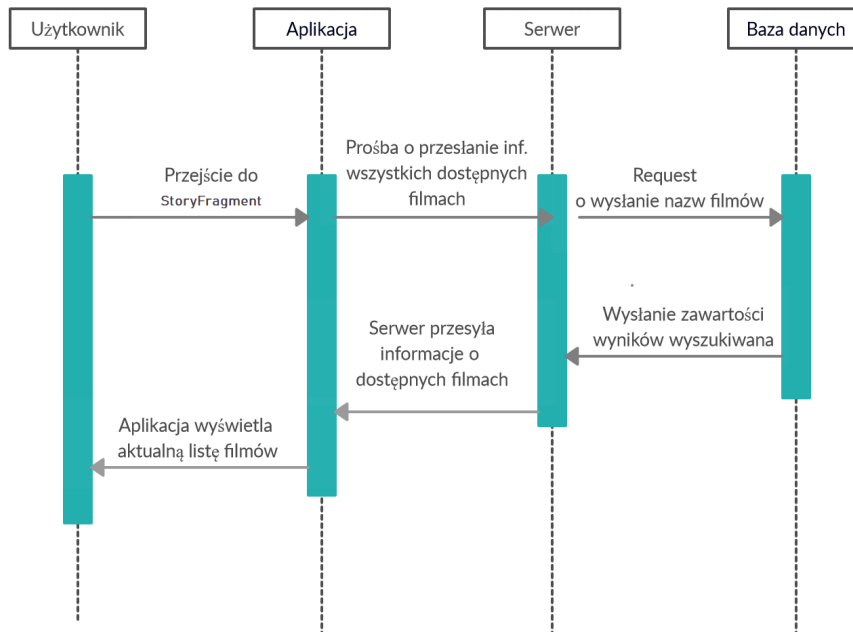
5.4. Usuwanie filmu

Użytkownik wybiera film, który chce usunąć. Z aplikacji zostaje wysłana prośba do serwera o usunięcie konkretnego video. Do bazy danych trafia request o usunięciu filmu. Do aplikacji zostają przesłane bieżące informacje o dostępnych filmach, a one są wyświetlane użytkownikowi.



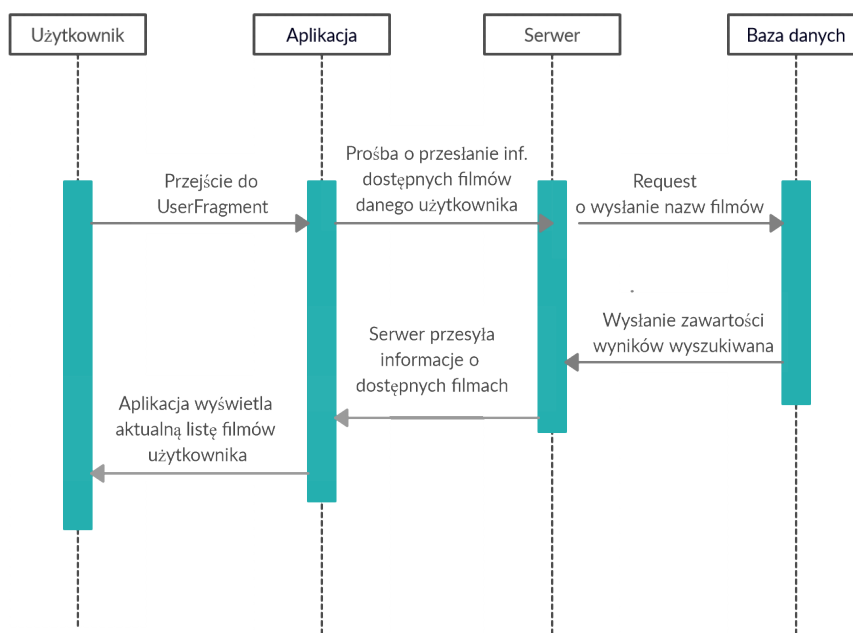
5.5. Przegląd wszystkich filmów Story

Użytkownik po przejściu do StoryFragment wywołuje wysłanie prośby przez aplikację do serwera o dostarczenie informacji na temat dostępnych do odtworzenia filmów video. Serwer wysyła request do bazy danych o wysłanie nazw filmów możliwych do przeglądnięcia. Baza danych odsyła wyniki wyszukiwania, a serwer podaje dalej informacje o dostępnych filmach. Aplikacja wyświetla użytkownikowi listę filmów, które może oglądać.



5.6. Przegląd wszystkich filmów użytkownika

Sytuacja działa na prawie takiej samej zasadzie jak opisana powyżej, tylko z tą różnicą, że tu wyświetlamy filmy konkretnego użytkownika.



5.7. Wyświetlanie filmu

Klient wybiera film, który chce oglądać. Aplikacja przesyła prośbę do Serwera o streaming wybranego video. Serwer natomiast prosi bazę danych o ścieżkę do filmu. Po otrzymaniu jej rozpoczyna się streaming wybranego video z wykorzystaniem HLS. Użytkownik rozpoczyna oglądanie video.

