

Technologie usługowe i mobilne

DOKUMENTACJA



Autorzy:

Agata Kowalewska
Filip Ignasiak

Kierunek:

Kryptologia i Cyberbezpieczeństwo
Bezpieczeństwo Informacji
30.06.2017r

SPIS TREŚCI

1. Wstęp	3
1.1. Cel dokumentu	3
1.2. Zawartość	3
1.3. Szablon opisu wymagań	3
2. Wymagania.....	4
2.1. Wymagania funkcjonalne	4
2.2. Wymagania niefunkcjonalne	4
3. Przypadki Użycia	5
3.1. Szablon opisu przypadków użycia	5
3.2. Opis przypadków użycia	6
3.4. Diagram przypadków użycia.....	8
3.5. Diagram klas	10
3. 5. 1. Aplikacja webowa	10
3. 5. 2. Aplikacja mobilna.....	12
3.6. Diagram wdrożeniowy.....	14
3.7. Diagram komponentów.....	15
4. Prezentacja aplikacji mobilnej	16

1. Wstęp

1.1. Cel dokumentu

Dokument ten powstaje w celu określenia wymagań dotyczących aplikacji mobilnej pokazującej zainstalowane komponenty. Dokument ten będzie źródłem informacji dla zespołu projektowego na temat wytwarzanego systemu, a także podstawą dla zlecającego do weryfikacji funkcjonalności oraz jakości wytworzonego oprogramowania.

1.2. Zawartość

Dokument opisywał będzie szczegółowo cele oraz zakres. W ramach analizy problemu zbadane i opisane zostaną procesy i zdarzenia biznesowe obejmowane przez system, a także granice systemu i jego kontekst. Wyszczególnione zostaną przypadki użycia systemu oraz wymagania funkcjonalne oraz нефункционалне.

1.3. Szablon opisu wymagań

Nr zgłoszenia	Priorytet	Status	<i>Tytuł</i>	Opis	Powiązane wymagania
---------------	-----------	--------	--------------	------	---------------------

Wymagania są opisane według jednolitego szablonu zawierającego następujące informacje:

Nr zgłoszenia

Unikalny numer w ramach całej specyfikacji wymagań.

Priorytet

Waga zgłoszenia ustalająca kolejność jego wykonania. Może przyjmować trzy wartości

- wysoki – wymaganie musi być koniecznie spełnione,
- średni – wymaganie powinno być spełnione, jeśli starczy na to czasu podczas realizacji systemu,
- niski – wymaganie dotyczy funkcji, która może być wprowadzona w następnej wersji systemu, pokazuje prawdopodobny kierunek rozwoju systemu.

Status

Status może przyjmować następujące wartości:

- Początkowy – wymaganie jest w trakcie tworzenia, treść może ulec zmianie,
- W trakcie realizacji – wymaganie jest w trakcie analizy i opracowania,
- Zakończony – wymaganie zweryfikowane i zaakceptowane

Tytuł

Nazwa wymagania

Opis

Treść wymagania.

Powiązane wymagania

Numery innych wymagań, które są w jakiś sposób powiązane z danym wymaganiem.

2. Wymagania

2.1. Wymagania funkcjonalne

Nr	Priorytet	Status	Tytuł	Opis	Powiązane wymagania
1	Wysoki	Początkowy	Szybka obsługa	Możliwość łatwego dotarcia do wskazanego komponentu w aplikacji	6,7,8
2	Wysoki	Początkowy	Obsługa	Zapisywanie danych na serwerze w postaci pliku tekstowego	2,5,6
3	Średni	Początkowy	Łatwość aktualizacji	Możliwość zdalnych aktualizacji systemu oraz dodawanie nowych funkcjonalności	5
4	Niski	Początkowy	Bezpieczeństwo	Użytkownik nie musi przysyłać danych na serwer. Dane nie będą wysyłane bez interakcji użytkownika	2,3,4
5	Wysoki	Początkowy	Obsługa wszystkich rodzajów	Obsługiwane są wszystkie aplikacje, zarówno wbudowane, brandowane, jak i zainstalowane	1
6	Średni	Początkowy	Etykietowanie	Etykietowanie odbywa się poprzez podanie adresu serwera i nazwy użytkownika	1,2,3,7
7	Niski	Początkowy	Język interfejsu	Komunikacja z użytkownikiem odbywa się w jednym języku angielskim	1,3,7

2.2. Wymagania нефunkcjonalne

- System musi zapewniać odtworzenie stanu początkowego po zaistniałej awarii
- Za odbieranie i przechowywanie danych użytkowników odpowiada aplikacja REST wykonana przy użyciu frameworku Spring w języku Java
- Moduły systemu zarządzania muszą być utworzone w tej samej technologii, tzn. napisane w tym samym języku (Java), i korzystać z tego samego serwera
- W przypadku komunikacji z serwerem poprzez sieć Internet, przechowywanie danych po identyfikatorze użytkownika
- Kompatybilność telefonów komórkowych z systemem operacyjnym Android 4.3+. Docelowa wersja systemu to Android Lollipop 5.0 pod API 21

- Wymaganie techniczne telefonów komórkowych:

Telefon komórkowy – Smartphone - minimalna konfiguracja sprzętowa	
Ekran	ekran dotykowy IPS LCD, multitouch 5.0 cali, 720 x 1280 px
Sieć	GSM / CDMA / HSPA / EVDO / LTE
Procesor	Octa-core (4x1.5 GHz Cortex-A53 & 4x1.2 GHz Cortex-A53)
OS	Android 5.0 (Lollipop)
Dodatkowe wymagania	Wi-Fi 802.11 b/g/n, Wi-Fi Direct, hotspot (dla serwera w lokalnej sieci)

3. Przypadki Użycia

3.1. Szablon opisu przypadków użycia

<i>Identyfikator</i>	<i>Tytuł</i>
Aktorzy	
Zdarzenie inicjujące	
Warunki początkowe	
Opis	
Warunki końcowe	
Sytuacje wyjątkowe	
Komentarz	

Aktorzy

Aktorzy związani z danym przypadkiem użycia.

Zdarzenie inicjujące

Zdarzenie, którym użytkownik inicjuje dany przypadek użycia.

Warunki początkowe

Warunki, które muszą być spełnione przez zainicjowanie danego przypadku użycia dla jego poprawnego przebiegu.

Opis

Opis realizacji funkcjonalności reprezentowanej przez dany przypadek użycia, w szczególności opis interakcji między użytkownikiem a systemem.

Warunki końcowe

Warunki, które musi spełniać system po poprawnym zakończeniu interakcji w ramach danego przypadku użycia.

Sytuacje wyjątkowe

Opis interakcji dla przebiegów alternatywnych.

Komentarz

Dodatkowe informacje, np. powiązania z innymi przypadkami użycia.

PU01	Etykietowanie
Aktorzy	Użytkownik aplikacji.
Zdarzenie inicjujące	Użytkownik włącza aplikację.
Warunki początkowe	Użytkownik uruchomił aplikację. Aplikacja zainicjowana prawidłowo.
Opis	1. Użytkownik wprowadza dane do etykietowania: adres serwera i swoją nazwę. 2. System ustawia odpowiednie zmienne.
Warunki końcowe	Użytkownik podał poprawne dane i przeszedł do widoku komponentów.
Sytuacje wyjątkowe	<ul style="list-style-type: none"> Podany adres serwera ma zły format – wymagane ponowne wprowadzenie. Pole adresu i/lub nazwy są puste – wymagane ponowne wprowadzenie.

3.2. Opis przypadków użycia

PU02	Wybór komponentu
Aktorzy	Użytkownik aplikacji.
Zdarzenie inicjujące	Użytkownik poprawnie wpisał dane w panelu logowania.
Warunki początkowe	Użytkownik uruchomił aplikację. Aplikacja zainicjowana prawidłowo.
Opis	1. Użytkownik wybiera z listy dostępnych aplikacji komponent. 2. Wybrany komponent otwiera się w drugiej karcie.
Warunki końcowe	Uruchomienie odpowiedniego komponentu.
Sytuacje wyjątkowe	Komponent nie uruchamia się – prośba o kontakt z twórcami aplikacji.

PU03	Odczytanie informacji o aplikacji
Aktorzy	Użytkownik aplikacji.
Zdarzenie inicjujące	Użytkownik kliknął w prawym górnym rogu na ikonkę trzech kropek.
Warunki początkowe	Użytkownik uruchomił aplikację. Aplikacja zainicjowana prawidłowo. Użytkownik podał poprawne dane w panelu logowania.
Opis	1. Aplikacja wyświetla popup z informacją o twórcach aplikacji
Warunki końcowe	Wyświetlił się popup z tekstem oraz możliwością wyłączenia go po przez przycisk.
Sytuacje wyjątkowe	<ul style="list-style-type: none"> Popup się nie wyświetli – błąd odnotowany w logach Popup się nie zamknie – błąd odnotowany w logach

PU04	Odczytanie danych zapisanych na serwerze
Aktorzy	Administrator serwera.
Zdarzenie inicjujące	Użytkownik aplikacji wysłał poprawnie dane komponentów.
Warunki początkowe	Administrator zalogował się na serwer.
Opis	1. Administrator wchodzi w folder Files na serwerze 2. Administrator wybiera folder z nazwą użytkownika 3. Administrator wybiera plik z odpowiednią datą numeryczną
Warunki końcowe	Plik się poprawnie wyświetlił

Sytuacje wyjątkowe	Plik lub folder mają pustą nazwę (na przykład tylko z podłogą i datą) – użytkownik wysłał dane z aplikacji mając uprawnienia root, bez zmiany nazwy użytkownika
--------------------	---

PU05	Wysłanie danych na serwer
Aktorzy	Użytkownik aplikacji.
Zdarzenie inicjujące	Użytkownik przeciąga palcem w dół dotykając ekranu (swipe down)
Warunki początkowe	Użytkownik uruchomił aplikację. Aplikacja zainicjowana prawidłowo. Użytkownik podał poprawne dane w panelu logowania.
Opis	1. Użytkownik na liście komponentów przewiniętej do 1 pozycji, dotyka ekranu i przesuwając palcem w dół 2. Lista komponentów oraz ich package name wraz z nazwą użytkownika zostają przesłane na serwer 3. Serwer zapisuje dane w formie pliku: nazwa_data numeryczna
Warunki końcowe	Komunikat Success
Sytuacje wyjątkowe	<ul style="list-style-type: none"> • Adres serwera jest błędny – alert Failed • Adres serwera jest poprawny, ale połączenie nie zostało zrealizowane – alert Failed • Adres serwera jest poprawny, ale dane nie zostały poprawnie odebrane – alert Failed i komunikat w serwerze o błędnym formacie danych

3.3. Scenariusze:

Scenariusz 01 – wysłanie listy komponentów:

1. Użytkownik uruchamia aplikację
2. Użytkownik podaje nazwę użytkownika i adres serwera
3. Użytkownik swipuje żeby wysłać listę komponentów na serwer
 - a. W przypadku braku osiągalności usługi odbierającej dane zostanie wyświetlony komunikat o błędzie
 - b. W przypadku błędu danych lub operacji, której nie mogła przetworzyć usługa webowa również pojawi się komunikat o błędzie, zazwyczaj z większą szybkością
 - c. W przypadku powodzenia aplikacja wyświetli wiadomość zwrotną od usługi

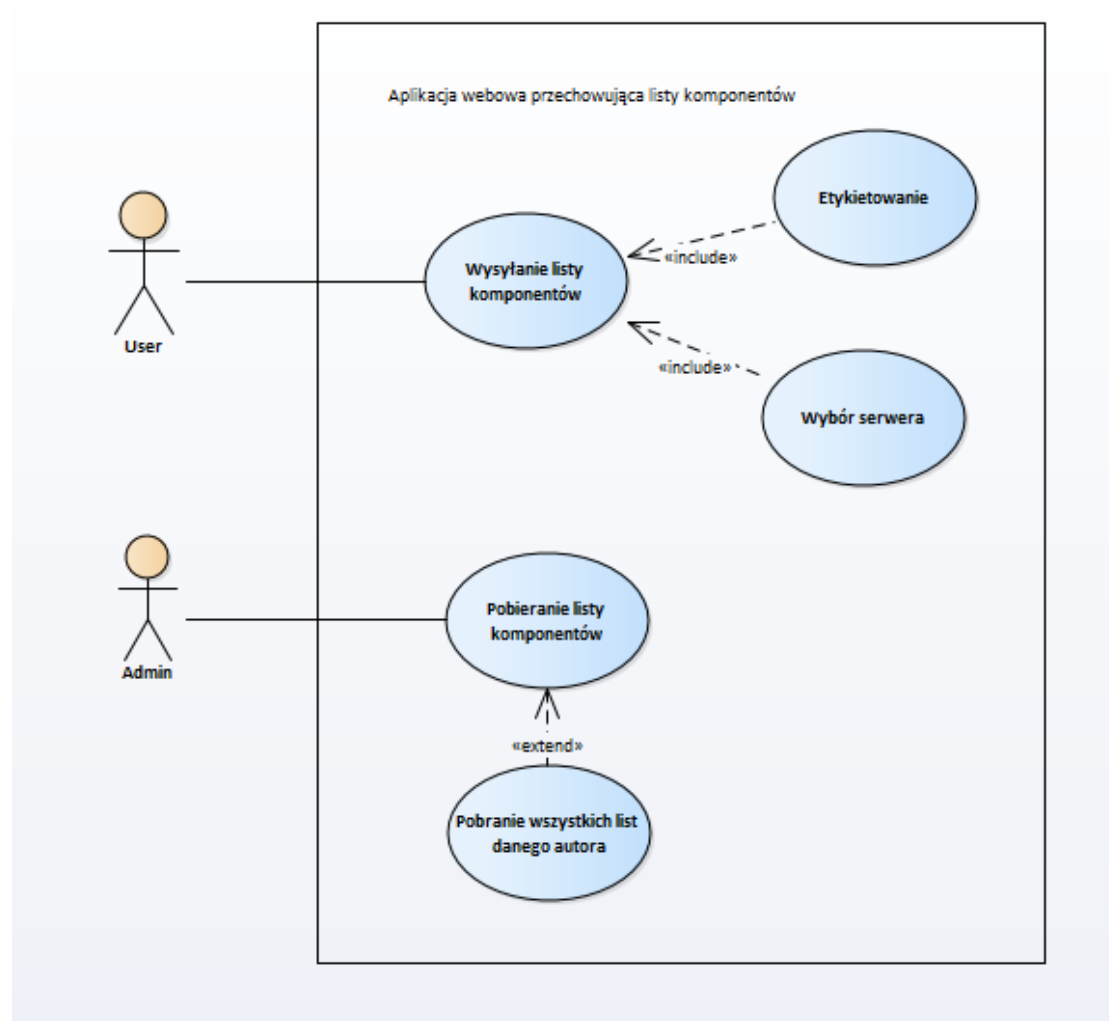
Scenariusz 02 – wybór aplikacji:

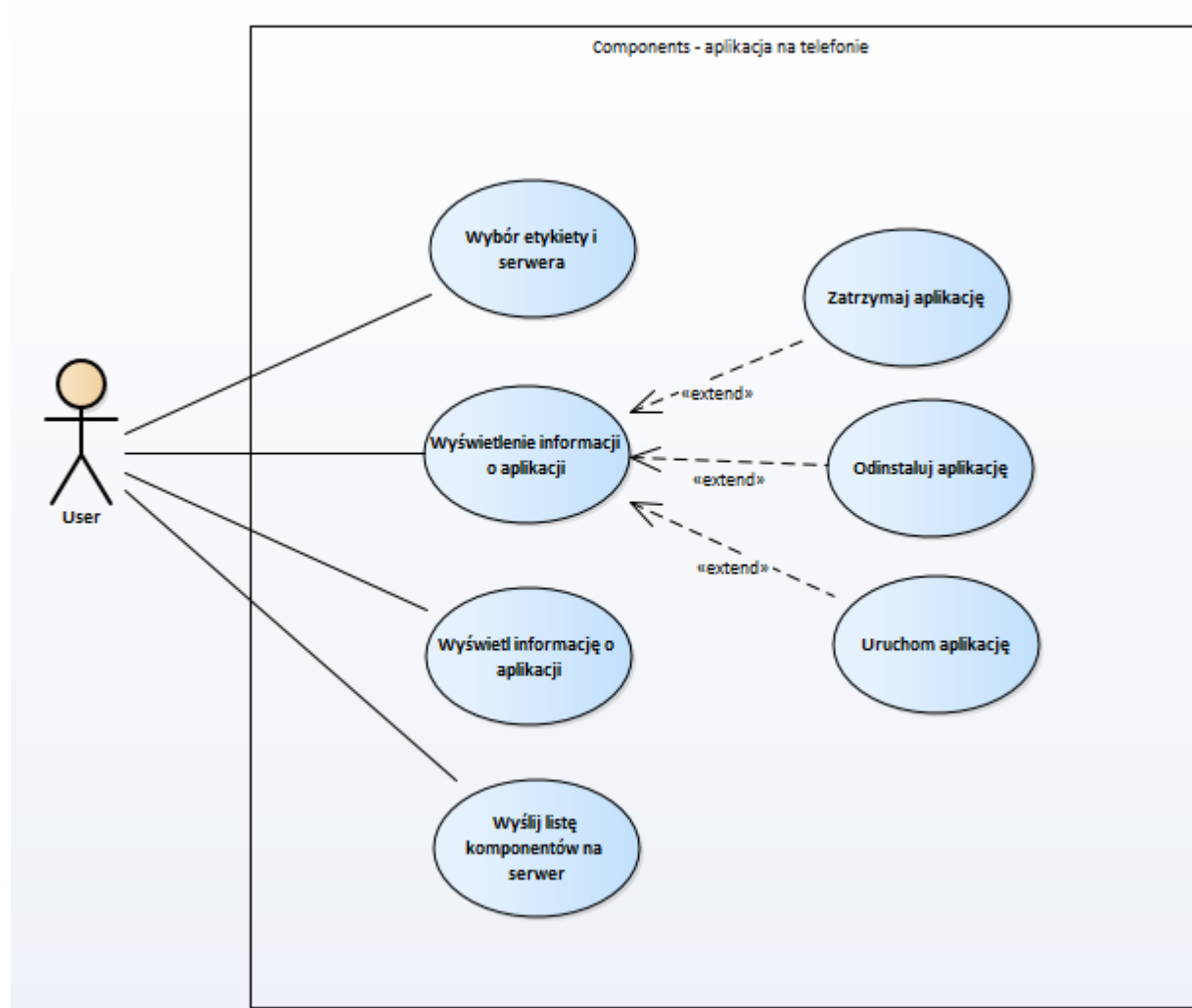
1. Użytkownik uruchamia aplikację
2. Użytkownik podaje nazwę użytkownika i adres serwera
3. Użytkownikowi wyświetlają się informacje o aplikacji
 - a. W przypadku kliknięcia na przycisk **start** aplikacja wystartuje
 - b. W przypadku kliknięcia na przycisk **uninstall** aplikacja się odinstaluje
 - c. W przypadku kliknięcia na przycisk **stop** aplikacja zatrzyma swoje działanie

Scenariusz 03 – wyświetlenie przechowywanych danych:

1. Administrator łączy się w sieci lokalnej z maszyną udostępniającą usługę
 - a. Administrator pobiera wszystkie dane etykietowane przez danego użytkownika
 - b. Administrator pobiera dane konkretnego pliku

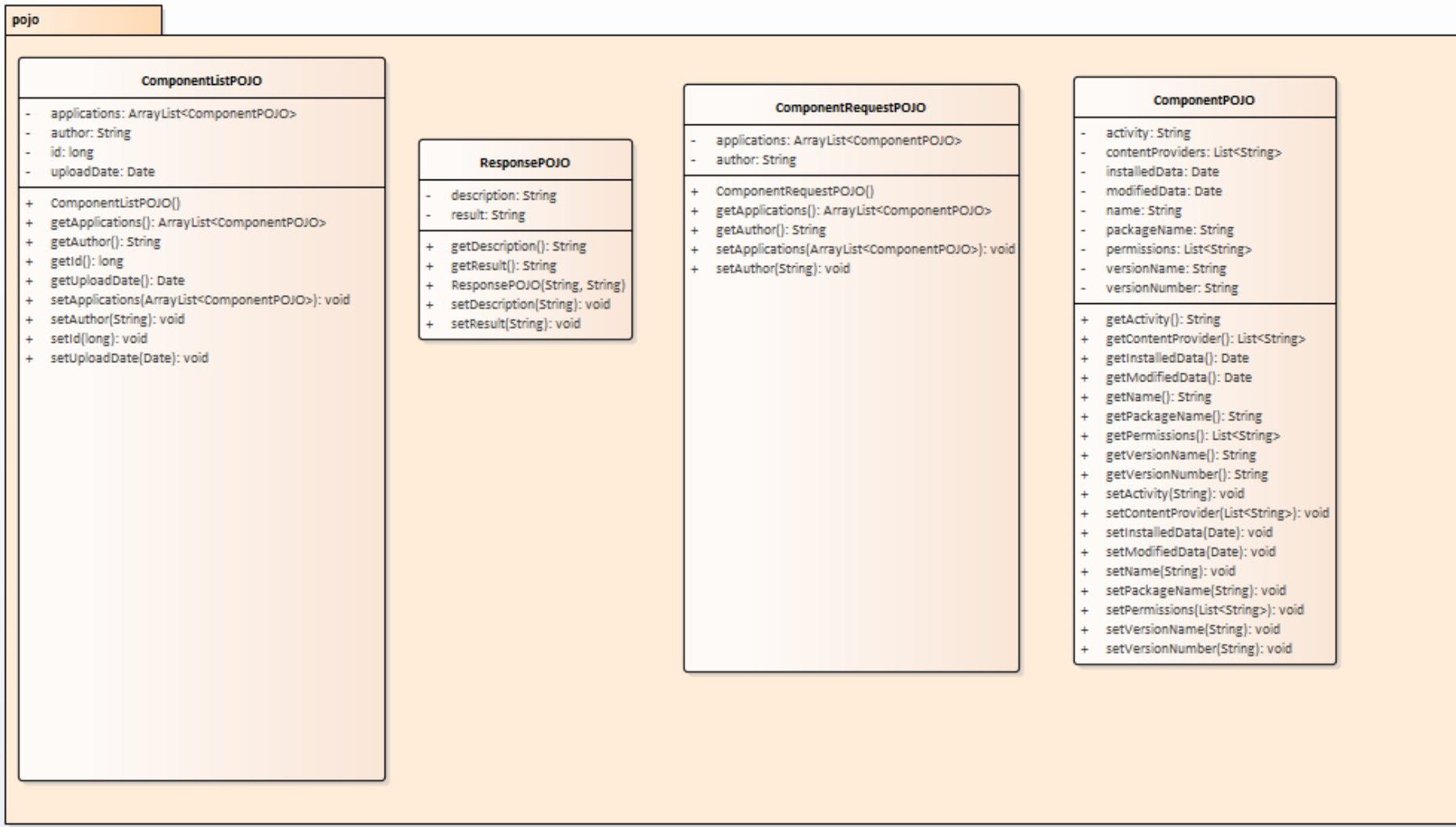
3.4. Diagram przypadków użycia

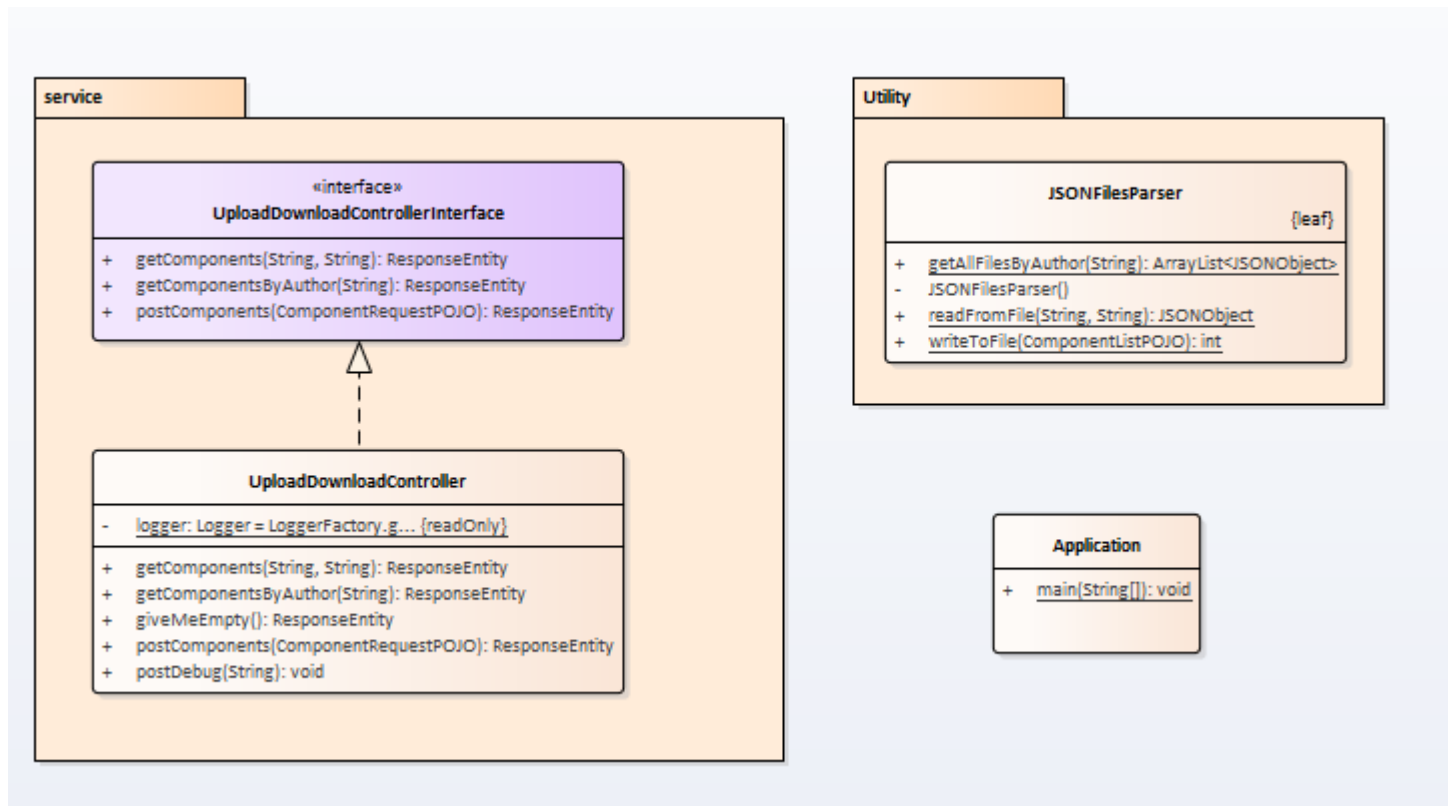




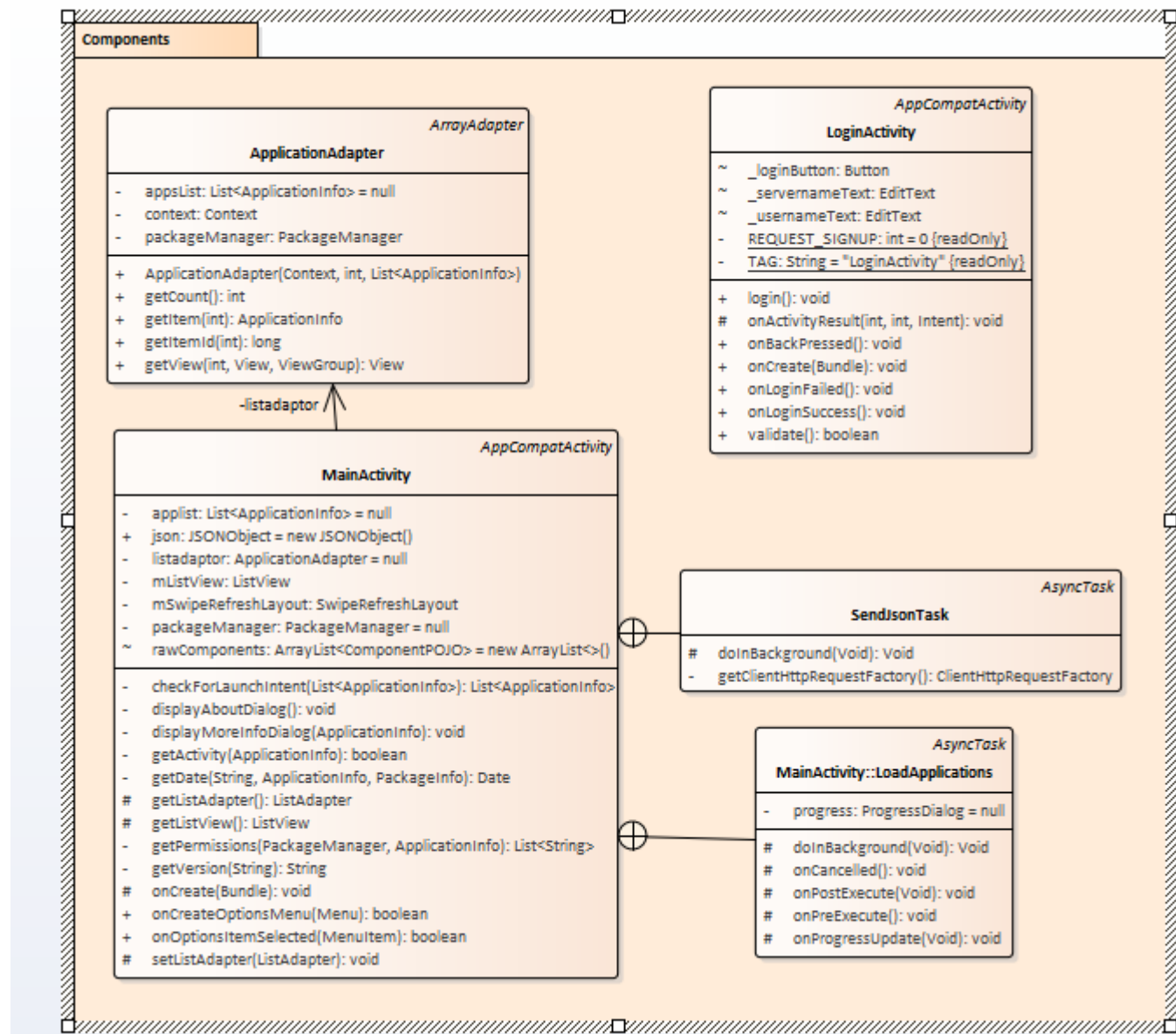
3.5. Diagram klas

3. 5. 1. Aplikacja webowa





3. 5. 2. Aplikacja mobilna



POJO

ComponentPOJO

```

- activity: String
- contentProviders: List<String>
- installedData: Date
- modifiedData: Date
- name: String
- packageName: String
- permissions: List<String>
- versionName: String
- versionNumber: String

+ getActivity(): String
+ getContentProvider(): List<String>
+ getInstalledData(): Date
+ getModifiedData(): Date
+ getName(): String
+ getPackageName(): String
+ getPermissions(): List<String>
+ getVersionName(): String
+ getVersionNumber(): String
+ setActivity(String): void
+ setContentProvider(List<String>): void
+ setInstalledData(Date): void
+ setModifiedData(Date): void
+ setName(String): void
+ setPackageName(String): void
+ setPermissions(List<String>): void
+ setVersionName(String): void
+ setVersionNumber(String): void

```

ResponsePOJO

```

- description: String
- result: String

+ getDescription(): String
+ getResult(): String
+ ResponsePOJO(String, String)
+ ResponsePOJO()
+ setDescription(String): void
+ setResult(String): void

```

ComponentRequestPOJO

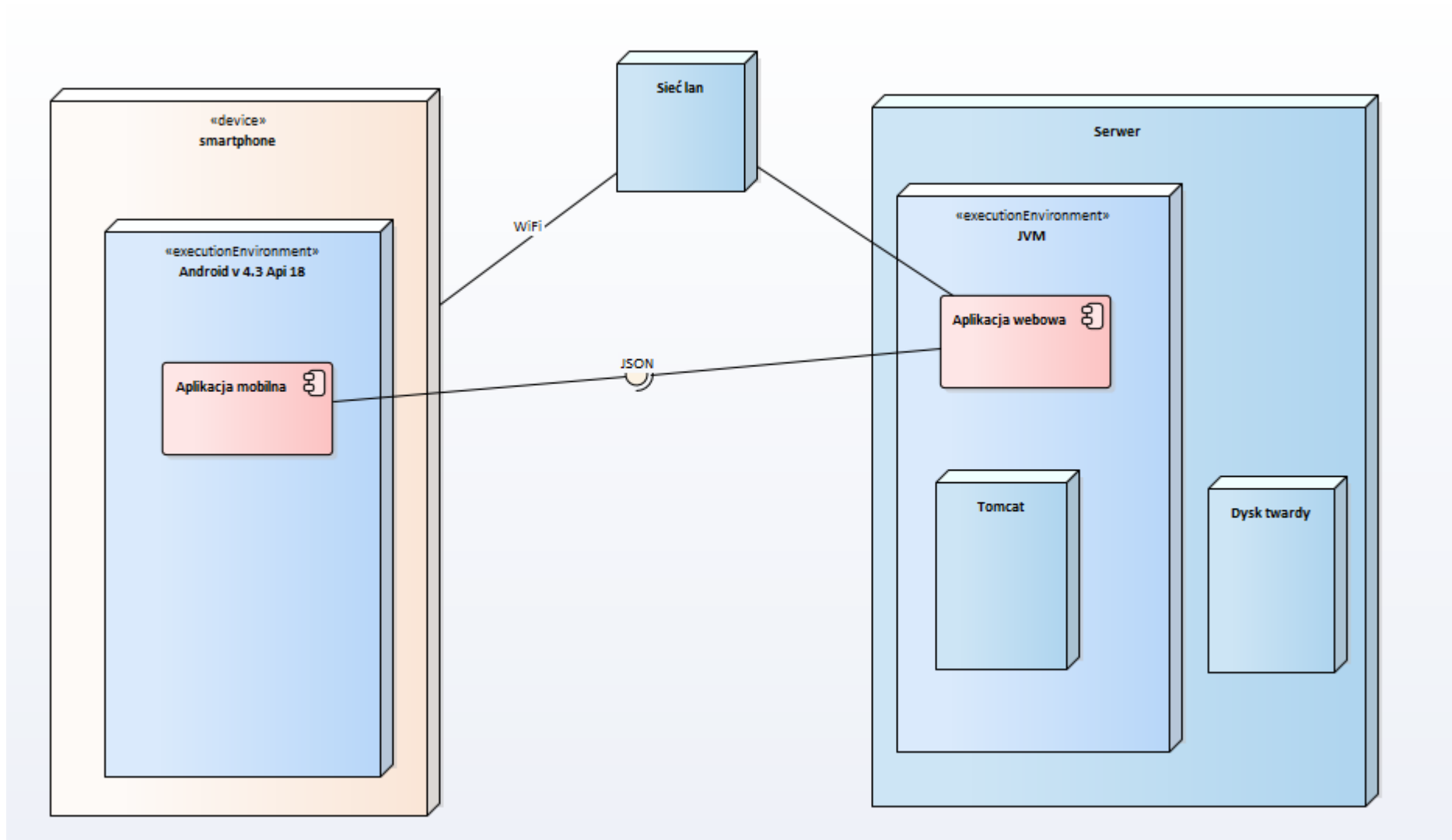
```

- applications: ArrayList<ComponentPOJO>
- author: String

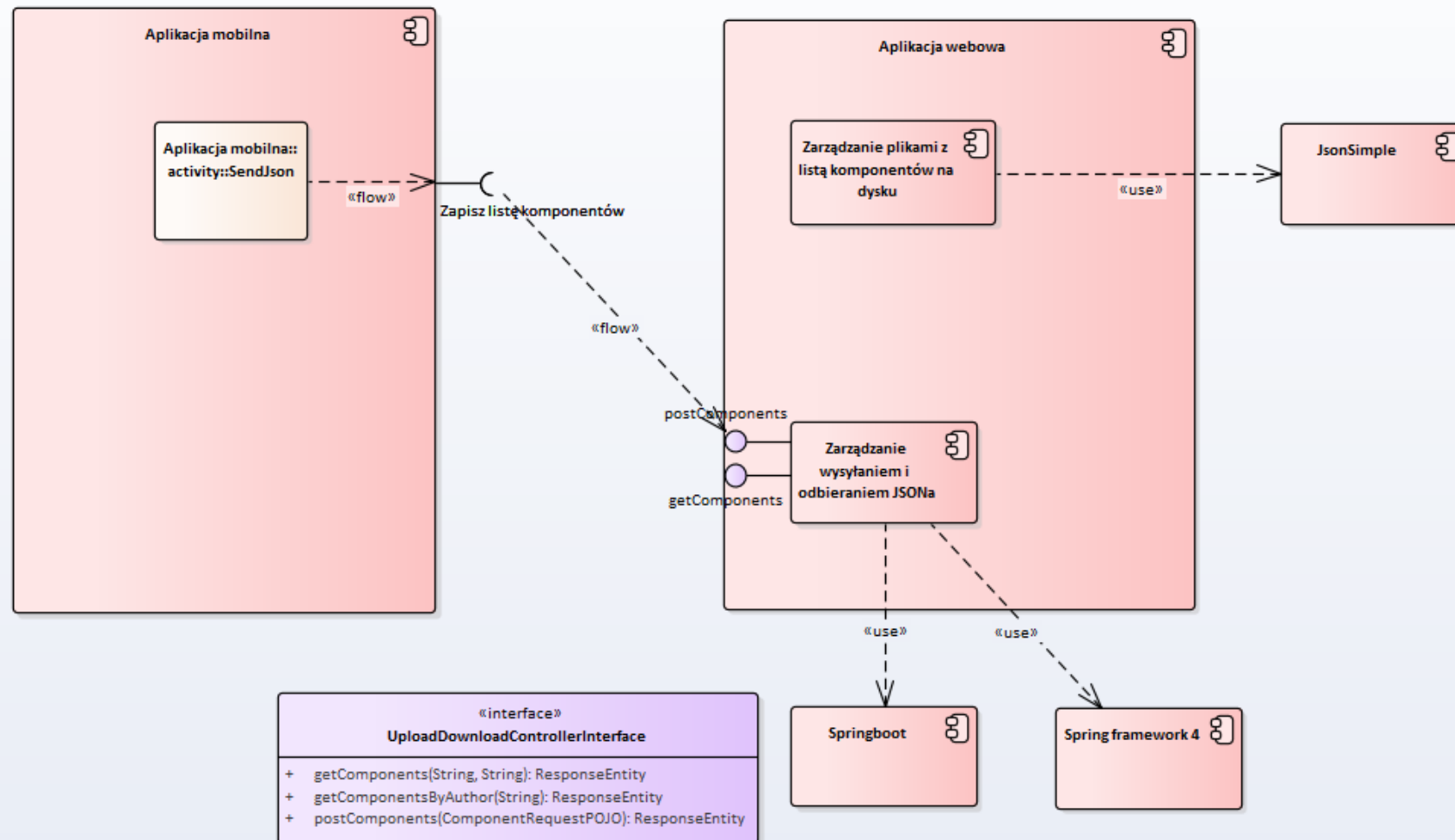
+ getApplications(): ArrayList<ComponentPOJO>
+ getAuthor(): String
+ setApplications(ArrayList<ComponentPOJO>): void
+ setAuthor(String): void

```

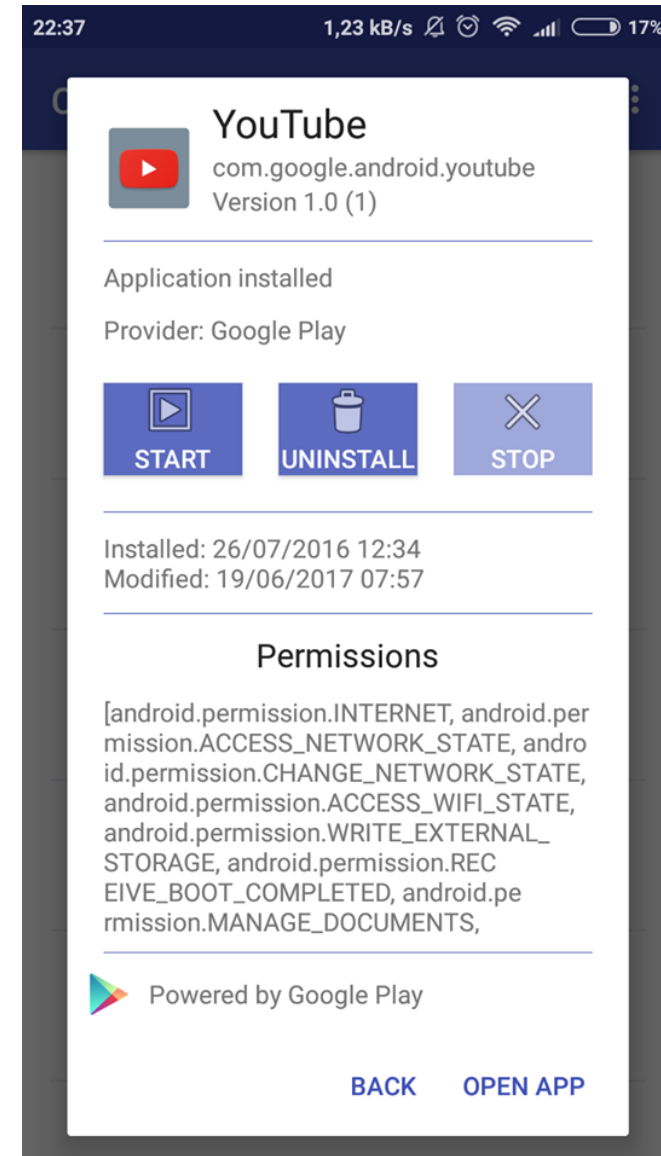
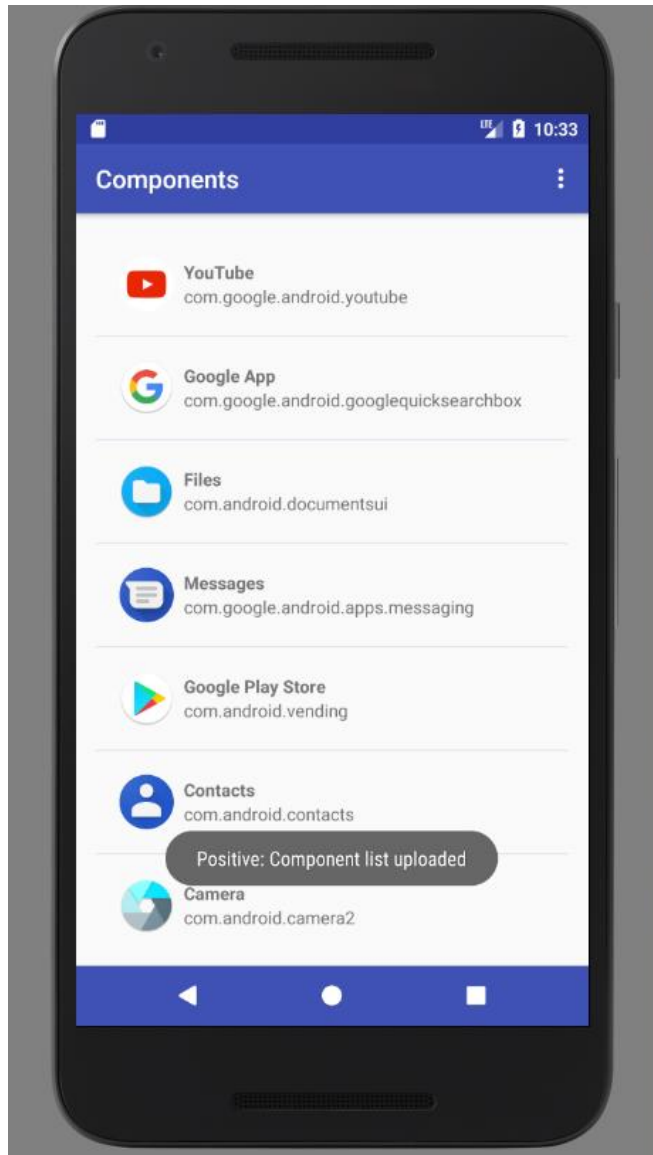
3.6. Diagram wdrożeniowy




3.7. Diagram komponentów



4. Prezentacja aplikacji mobilnej



0:23 ... 0,11 kB/s 8%







Username
Agata

Server address
http://192.168.99.1:8090

START

0:23 ... 1,08 kB/s 8%

Components

-  **YouTube**
com.google.android.youtube
-  **Google**
com.google.android.googlequicksearch
-  **Backup Your Mobile**
com.backupyourmobile
-  **Konto Xiaomi**
com.xiaomi.account

Credentials

Kowalewska Agata
Filip Ignasiak

2017. Military University of
Technology in Warsaw.

GREAT!