

POLITECHNIKA POZNAŃSKA

INFORMATYKA SEMESTR 6

PODSTAWY TELEINFORMATYKI - PROJEKT

InstaScraper

Skład grupy:

Anna ZDROJEWSKA **123641**

anna.zdrojewska@student.put.poznan.pl

Szymon KRÓLIKOWSKI **127055**

szymon.krolikowski@student.put.poznan.pl

Daniel PAWLIKOWSKI **126899**

daniel.pawlikowski@student.put.poznan.pl

Prowadzący:

mgr inż. Przemysław WALKOWIAK

Spis treści

1	Charakterystyka ogólna	1
1.1	Wybór tematu	1
1.2	Opis produktu	1
1.3	Podział prac	2
2	Wymagania funkcjonalne	3
2.1	Ogólna funkcjonalność	3
2.2	Klasyfikacja funkcjonalności	3
3	Wymagania niefunkcjonalne	4
3.1	Wymagania produktowe	4
3.2	Wymagania organizacyjne	4
3.3	Wymagania zewnętrzne	5
3.4	Charakterystyka użytkowników	5
4	Użyte narzędzia i środowiska	6
4.1	Środowisko	6
4.2	Narzędzia	6
5	Model aplikacji	8
5.1	Aktorzy	8
5.2	Diagram aktywności	9
5.3	Diagram sekwencji	10
6	Problemy i ich rozwiązania	11
6.1	Dynamicznie zmieniające się API Instagram	11
6.2	Dynamiczne przewijanie treści profili na Instagramie	13
6.3	Pobieranie danych w jak najkrótszym czasie	15
6.4	Pobieranie danych w pełnej rozdzielczości	17
7	Proces debugowania aplikacji	19
8	Testy jednostkowe	20
9	Projekt interfejsu graficznego	22
10	Instrukcja użytkowania aplikacji	24
10.1	Uruchomienie aplikacji	24
10.2	Obsługa aplikacji	24
10.3	Walidacja	25
11	Perspektywy rozwoju	28
12	Bibliografia	30

1 Charakterystyka ogólna

Niniejsza dokumentacja zawiera wymagania związane ze stworzeniem aplikacji komputerowej InstaScaper. Ten dokument ma za zadanie sprecyzować zarówno wygląd jak i funkcjonalność aplikacji. Całość została podzielona na kilka rozdziałów. W tym rozdziale opisana jest idea projektu.

1.1 Wybór tematu

Web scraping jest techniką wydobywania oraz gromadzenia interesujących nas informacji dostępnych za pośrednictwem stron internetowych. Sposobem pozyskiwania tego typu danych jest parsowanie kodu HTML wybranych stron za pośrednictwem wcześniej stworzonych wyrażeń regularnych. Projekt ma opierać się na pozyskiwaniu informacji zawartych na publicznych profilach w serwisie Instagram takich jak zdjęcia, wideo, tagi oraz wykorzystaniu ich do celów badawczych. Web scraping ma również posłużyć jako sposób obejścia narzuconych przez Instagram ograniczeń, które zawarte są w API.

W dzisiejszych czasach coraz więcej informacji jest upublicznianych w Internecie. Coraz popularniejsze stają się również serwisy takie jak Facebook, Instagram, Twitter, które są kopalniami wiedzy o ludziach, ich stylu życia czy nawet zwyczajach zakupowych. Dzięki temu web scraping zdobywa na popularności. Temat projektu zainteresował naszą grupę, ponieważ uważamy, że warto zaznajomić się z techniką często wykorzystywaną w tak popularnym zagadnieniu jakim jest Big Data.

1.2 Opis produktu

Założeniem niniejszego projektu jest stworzenie web scrapera bazującego na popularnym serwisie zdjęciowym jakim jest Instagram. Aplikacja ma oferować możliwość wybrania danych konkretnego użytkownika, które mają zostać pobrane ze strony oraz miejsce na dysku w jakim ma je zapisać.

Podanie konkretnego użytkownika ma się wiązać z wpisaniem jego nicku w odpowiednim polu aplikacji. Konto użytkownika musi być oznaczone jako profil publiczny w serwisie Instagram.

Wybór danych ma polegać na zaznaczeniu interesującej użytkownika opcji. Możliwości jakie są przewidziane to zdjęcia/wideo oraz zdjęcia/wideo + tagi.

Interfejs ma za zadanie ułatwić użytkownikowi korzystanie z pełnej funkcjonalności aplikacji. Dzięki niej, użytkownik będzie mógł z łatwością pobrać dane w pełnej rozdzielczości, czego nie oferuje strona Instagram.

1.3 Podział prac

1. Anna Zdrojewska

- Wykorzystanie framework'u Selenium w celu automatyzacji ruchów w przeglądarce,
- Interfejs użytkownika z wykorzystaniem JavaFX,
- Implementacja funkcjonalności,
- Praca nad dokumentacją.

2. Szymon Królikowski

- Opracowanie odpowiednich wyrażeń regularnych,
- Dostosowywanie aplikacji do dynamicznej zmiany API Instagram,
- Implementacja funkcjonalności,
- Praca nad dokumentacją.

3. Daniel Pawlikowski

- Prace badawcze związane z mechaniką działania serwisu Instagram,
- Manipulacje linkami na podstawie API Instagram,
- Implementacja funkcjonalności,
- Praca nad dokumentacją.

2 Wymagania funkcjonalne

Ten rozdział opisuje wymagania dotyczące wyniku zachowania systemu, który powinien zostać dostarczony przez funkcje systemu.

2.1 Ogólna funkcjonalność

Aplikacja powinna oferować możliwość pobrania danych z publicznego konta użytkownika serwisu Instagram po podaniu nazwy jego konta w serwisie. Pobieranie danych powinno być jak najszybsze. Dane, które powinny zostać pobrane to zdjęcia, wideo oraz tagi. Aplikacja powinna pobierać dane w jak najwyższej rozdzielczości. Aplikacja powinna być zabezpieczona przed pobieraniem duplikatów zdjęć oraz wideo. Aplikacja powinna posiadać instrukcję obsługi.

2.2 Klasyfikacja funkcjonalności

Tabela 1 zawiera zebrane funkcjonalności wraz z aktorami. Każdy przypadek użycia ma typ priorytetu implementacji. Typ wymagany to taki, który jest niezbędny do poprawnego działania aplikacji, typ pożądaný to zapewniający podstawowe funkcjonalności, a typ opcjonalny to taki, który nie ma wpływu na wynik działania aplikacji.

Tablica 1: Tabela funkcjonalności.

Funkcjonalność	Typ	Aktorzy
Pobieranie zdjęć/wideo	wymagane	Użytkownik
Pobieranie zdjęć/wideo i tagów	wymagane	Użytkownik
Szybkość pobierania	pożądany	Administrator
Ususwanie duplikatów	pożądany	Administrator
Wysoka rozdzielczość	pożądany	Administrator
Instrukcja	opcjonalny	Użytkownik

3 Wymagania niefunkcjonalne

Ten rozdział opisuje czynniki wpływające na implementację aplikacji i wymagania z nią związane. Znajdują się tu również pewne ograniczenia nałożone na projekt.

3.1 Wymagania produktowe

Ta sekcja przedstawia wymagania ograniczające działanie aplikacji. Są one niezbędne do poprawnego korzystania z dostarczonych funkcji.

1. Aplikacja powinna zawierać interfejs w języku polskim,
2. Pobieranie danych z serwisu nie powinno trwać dłużej niż 5 minut,
3. Użytkownik w celu korzystania z pełnej funkcjonalności aplikacji powinien posiadać na dysku wystarczającą ilość miejsca, aby pobrać pliki (przez wystarczającą ilość należy rozumieć około 400KB/zdjęcie i około 600KB/wideo, przy czym należy pamiętać, że pojedyncze dane mogą przekraczać ten rozmiar),
4. Użytkownik w celu korzystania z pełnej funkcjonalności aplikacji powinien posiadać dostęp do Internetu oraz przeglądarkę Google Chrome.

3.2 Wymagania organizacyjne

W tej sekcji przedstawiono ograniczenia dla programisty implementującego aplikację. Wskazują one również, jak powinien przebiegać cały proces prac nad projektem.

1. Do implementacji web scraper'a posłuży język Java SE w wersji 8u162,
2. Do wykonania interfejsu użytkownika posłuży JavaFX,
3. Do parsowania stron posłuży biblioteka.jsoup 1.11.2,
4. Do automatyzacji ruchów na stronie posłuży framework Selenium 3.11 oraz WebDriver,
5. Środowiskiem programistycznym będzie środowisko IntelliJ IDEA 2018.1.4,
6. Narzędziem budowy projektu będzie Gradle 4.6,
7. Systemem kontroli wersji będzie repozytorium Git w serwisie Github.com,
8. Za narzędzie do tworzenia diagramów UML posłuży program Visual Paradigm Community Edition 15.0,
9. Narzędziem do tworzenia dokumentacji będzie LaTeX.

3.3 Wymagania zewnętrzne

Ta kategoria wymagań opiera się na ogólnych założeniach, które są niezbędne przy tworzeniu aplikacji.

1. Korzystanie z aplikacji powinno być łatwe dla użytkownika mającego z nią styczność po raz pierwszy, bez przeprowadzania szkolenia,
2. Aplikacja, w trakcie trwania projektu, powinna być dostosowywana do ewentualnych zmian zachodzących w serwisie Instagram.

3.4 Charakterystyka użytkowników

Aplikacja jest przeznaczona dla osób prywatnych, które chcą pozyskać dane w pełnej rozdzielczości z serwisu Instagram. Dane mogą posłużyć również jako materiał badawczy dla użytkowników.

4 Użyte narzędzia i środowiska

W tej sekcji opisane zostaną poszczególne narzędzia, które posłużą do wykonania projektu.

4.1 Środowisko

1. IntelliJ IDEA 2018.1.4

Środowisko programistyczne firmy JetBrains dla języka programowania Java. Zostało wybrane przez naszą grupę ze względu na wiele funkcji, dzięki którym ułatwia pisanie kodu. Takimi funkcjami są na przykład podpowiadanie nazw obiektów podczas ich tworzenia, łączenie się z repozytorium Git bez potrzeby wpisywania komend.

4.2 Narzędzia

1. Java SE 8u162

Java jest obiektywnym językiem programowania, który cieszy się bardzo dużą popularnością wśród programistów. Wybraliśmy go, ponieważ znamy i lubimy jego strukturę. Mimo, że na początku projektu, jeszcze w fazie projektowej wyszła już najnowsza 10 wersja tego języka, to zdecydowaliśmy się na użycie starszej 8 wersji. Wynika to z tego, że wybrana przez nas wersja posiada najdłuższe wsparcie techniczne.

2. JavaFX

JavaFX jest domyślną biblioteką Java do tworzenia graficznego interfejsu użytkownika. Pozwala ona na połączenie technologii CSS z plikami FXML dla lepszego wyglądu aplikacji. Łatwość implementacji oraz ładny wygląd skusiły nas do wybrania tego rozwiązania.

3. jSoup 1.11.2

Najpopularniejsza biblioteka dla Javy do parsowania stron www oraz pobierania ich zawartości. Posiada dobrą dokumentację oraz w Internecie jest dużo poradników jak z niej korzystać, co niewątpliwie wpłynęło na nasz wybór tej technologii.

4. Selenium 3.11 + WebDriver

Selenium jest frameworkiem, który potrafi automatyzować ruchy na stronie www. Posiada dobrą dokumentację i jest niezbędny przy projekcie, który opiera się na pobieraniu danych ze strony internetowej. Wybraliśmy go z konieczności zastosowania takiego rozwiązania.

5. Maven

Maven to narzędzie, które pozwala zautomatyzować budowanie projektów. Dzięki niemu można pozyskiwać dostęp do zewnętrznych zasobów takich jak Selenium. Chcąc korzystać z popularnej i dobrze znanej technologii zdecydowaliśmy się właśnie na to narzędzie.

6. repozytorium Git w serwisie Github.com

Bardzo popularny system kontroli wersji. Wszyscy członkowie zespołu posiadają konta w serwisie Github.com, z tego powodu został on wybrany.

7. Visual Paradigm Community Edition 15.0

Ten program służy do tworzenia diagramów UML. UWażamy, że jest jednym z najlepszych narzędzi na rynku do tego służących. Dodatkowo posiada wiele przydatnych opcji oraz jest dostępny w bezpłatnej wersji co ostatecznie przekonało nas do jego zastosowania.

8. LaTeX

System do tworzenia dokumentów. Szeroko stosowany w kręgach akademickich, w tym do tworzenia artykułów naukowych. Wybraliśmy go, ponieważ zapewnia profesjonalny wygląd dokumentacji.

5 Model aplikacji

W tej części dokumentacji zebrano i opisano wybrane elementy systemu, których zrozumienie i implementacja jest szczególnie ważna. Funkcjonalność gotowego produktu zależy w dużej mierze od stosowania się do poniższych wskazówek. Każdy podpunkt tego rozdziału skupia się na innym aspekcie implementacji.

5.1 Aktorzy

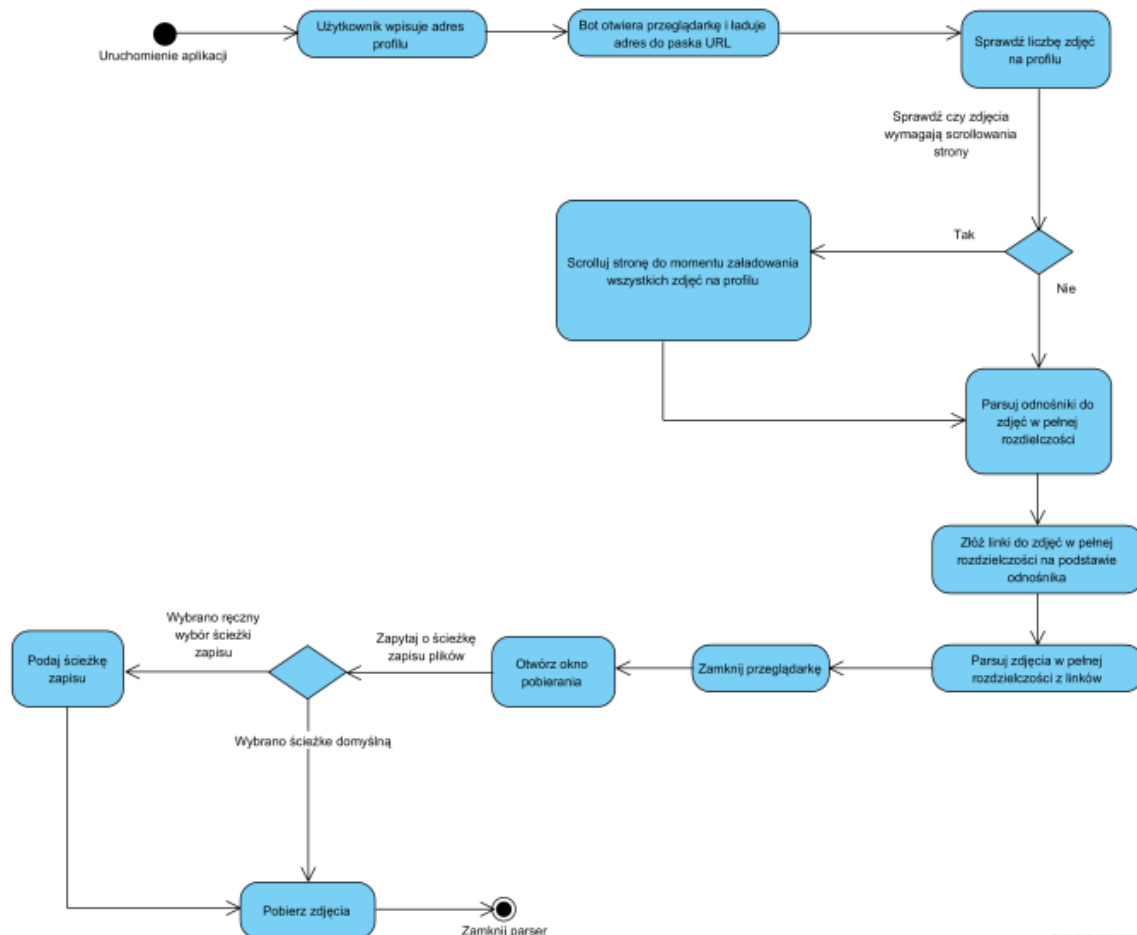
Tabela 2 przedstawia zbiór aktorów występujących w aplikacji wraz z ich opisem. Wyróżnione są dwa typy: Użytkownik oraz Administrator.

Tablica 2: Tabela aktorów.

Aktor	Opis
Użytkownik	Prywatna osoba korzystająca z aplikacji InstaScraper.
Administrator	Programista, aktor zarządzający aplikacją.

5.2 Diagram aktywności

Diagram aktywności (ang. activity diagram) jest diagramem interakcji, który służy do modelowania dynamicznych aspektów systemu. Jego zasadniczą funkcją jest przedstawienie sekwencji kroków, które są wykonywane przez modelowany fragment systemu. Poniższe zdjęcie przedstawia kolejne kroki w procesie parsowania zdjęć przez aplikację:

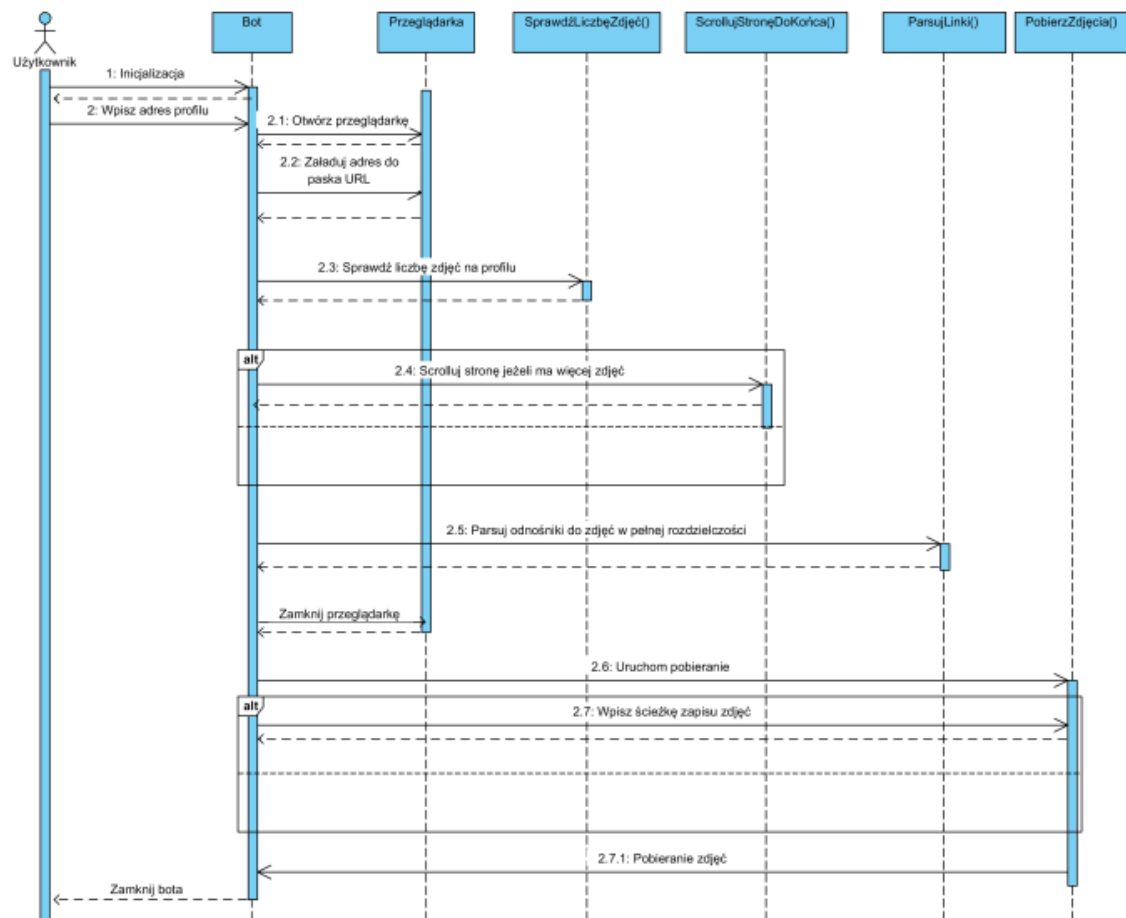


Rysunek 1: Diagram aktywności parsera

5.3 Diagram sekwencji

Diagram sekwencji (ang. sequence diagram) służy do prezentowania interakcji pomiędzy obiektami wraz z uwzględnieniem w czasie komunikatów, jakie są przesyłane pomiędzy nimi. Zasadniczym zastosowaniem diagramów sekwencji jest modelowanie zachowania systemu w kontekście scenariuszy przypadków użycia. Diagramy sekwencji pozwalają uzyskać odpowiedź na pytanie, jak w czasie przebiega komunikacja pomiędzy obiektami.

Na poniższym zdjęciu zilustrowany został przebieg kolejnych kroków parsera, wraz z uwzględnieniem aktorów wykonujących poszczególne czynności.



Rysunek 2: Diagram sekwencji parsera

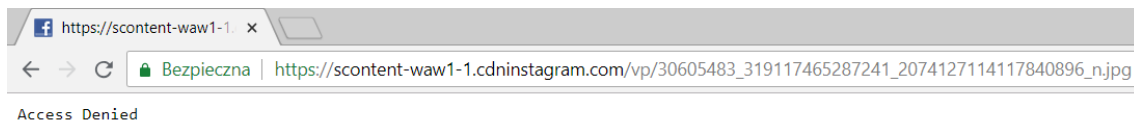
6 Problemy i ich rozwiązania

W tym rozdziale opisane są problemy z jakimi zetknął się zespół w trakcie tworzenia aplikacji oraz ich rozwiązania.

6.1 Dynamicznie zmieniające się API Instagram

- Problem

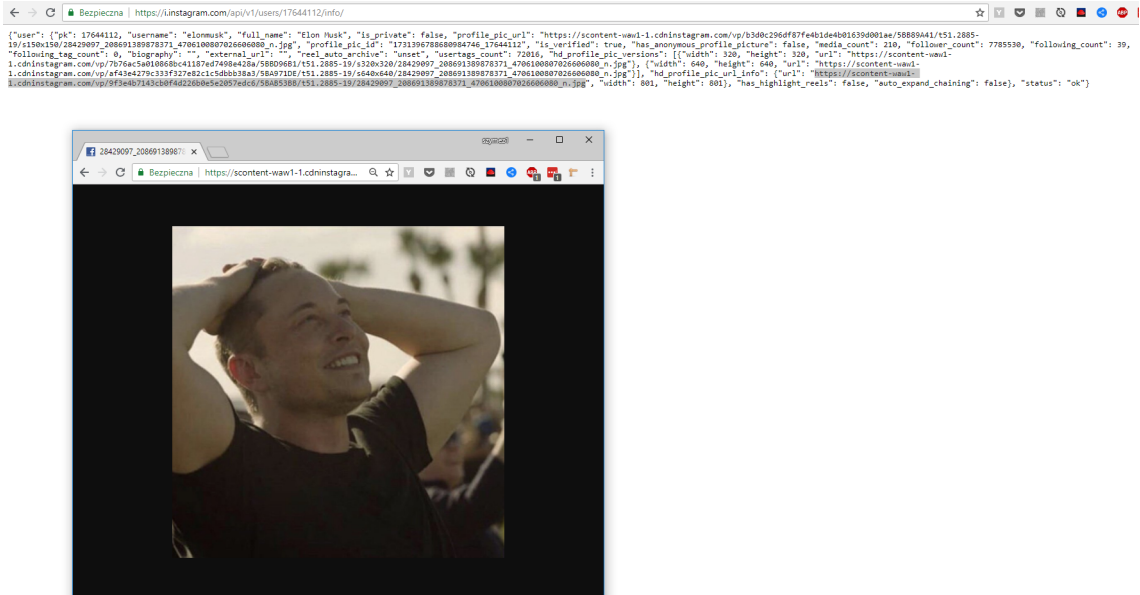
Wydarzeniami związane z ujawnieniem wykorzystywania danych do manipulacji przedwyborczych przez firmę Columbia Analytics na podstawie serwisu społecznościowego Facebook miały zdecydowany wpływ na rozwój projektu. Instagram jako serwis należący do firmy Facebook zaostrzył po tych wydarzeniach dostęp do danych użytkowników. Manipulacja linkami, której używaliśmy, z dnia na dzień stała się bezużyteczna.



Rysunek 3: Komunikat po próbie dostępu do zdjęcia profilowego.

- Rozwiązanie

Społeczność programistów z całego świata szybko poradziła sobie z częścią problemu. W ciągu dwóch tygodni od zablokowania dostępu powstało wiele różnych rozwiązań umożliwiających pobieranie niektórych danych. My wykorzystaliśmy kopiowanie odnośników do zdjęć z profilu użytkownika.

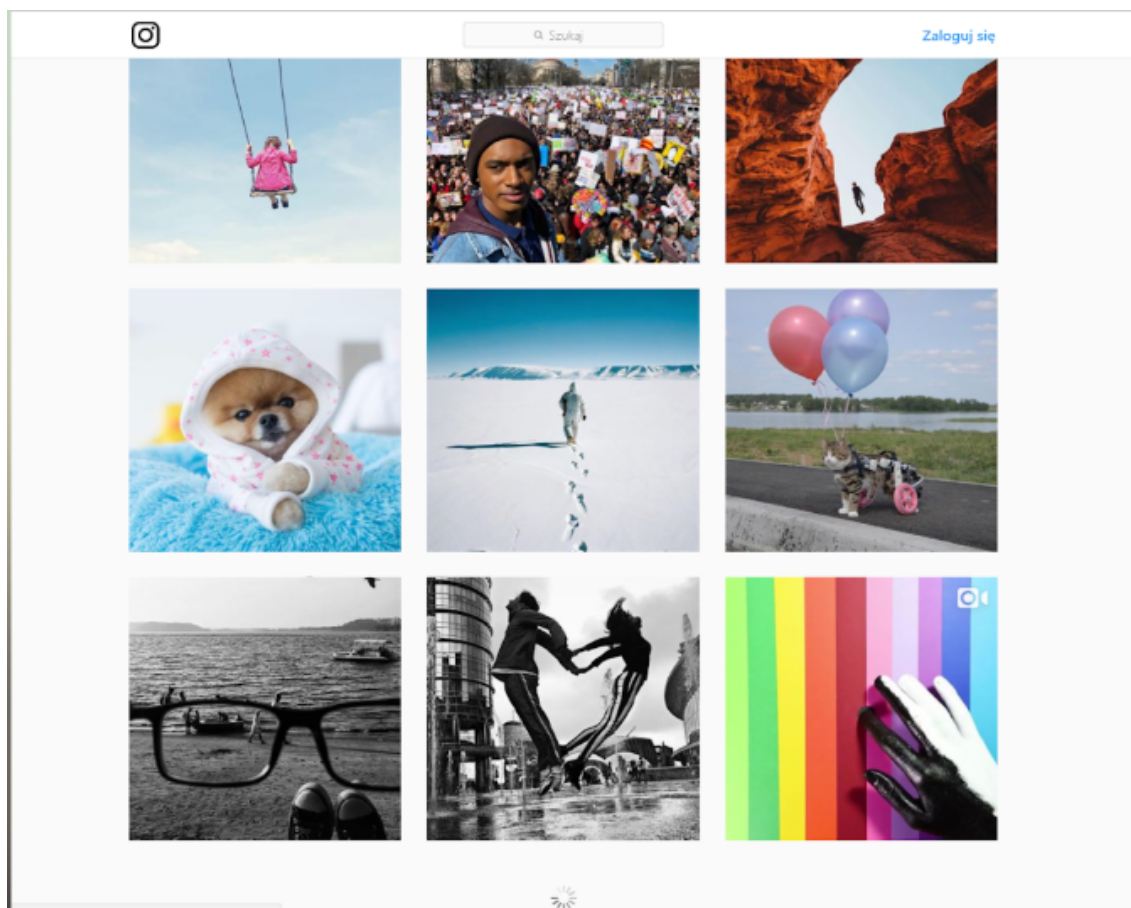


Rysunek 4: Sposób dostępu do zdjęcia profilowego w pełnej jakości (manipulacja linkami).

6.2 Dynamiczne przewijanie treści profili na Instagramie

- Problem

Przewijając publiczny profil użytkownika serwisu Instagram można zauważyć, że zdjęcia są buforowane na bieżąco. Przy tym pojawia się pewien znaczący problem, buforowane są zdjęcia tylko aktualnie widoczne w rozdzielczości przeglądarki. Zdjęcia buforowane wcześniej znikają, a przewijając stronę na początek ponownie trzeba oczekiwać na buforowanie zdjęć.



Rysunek 5: Zdjęcia doładowywują się na bieżąco wraz z przewijaniem strony w dół.

- Rozwiązanie

Do automatyzacji procesu przeglądania został użyty framework Selenium. Do pobierania danych została użyta klasa Javy Set, która nie może zawierać w sobie duplikatów. Ma to na celu uniknięcie powtarzania się linków do danych podczas buforowania zdjęć.

```
public void renderAllPhotos() throws InterruptedException, IOException {
    PageScroller scroller = new PageScroller(driver);
    int defaultStartScrollingPixels = 1290;
    int pixelsToScroll = 980;
    Integer loadedImgRows = 12;
    Integer previousNumberOfLoadedRows = 0;
    Integer imgRowsToLoad = HelperClass.numberCeiling(imageCounter()/3);

    scroller.scrollSomePixelsDown( scrollingStartInPixels: 0,pixelsToScroll);
    while(fullImgUrlSet.size() < imageCounter()){

        Thread.sleep( millis: 560);
        scroller.scrollSomePixelsDown( scrollingStartInPixels: 0,defaultStartScrollingPixels);
        defaultStartScrollingPixels+=1290;
        Thread.sleep( millis: 460);

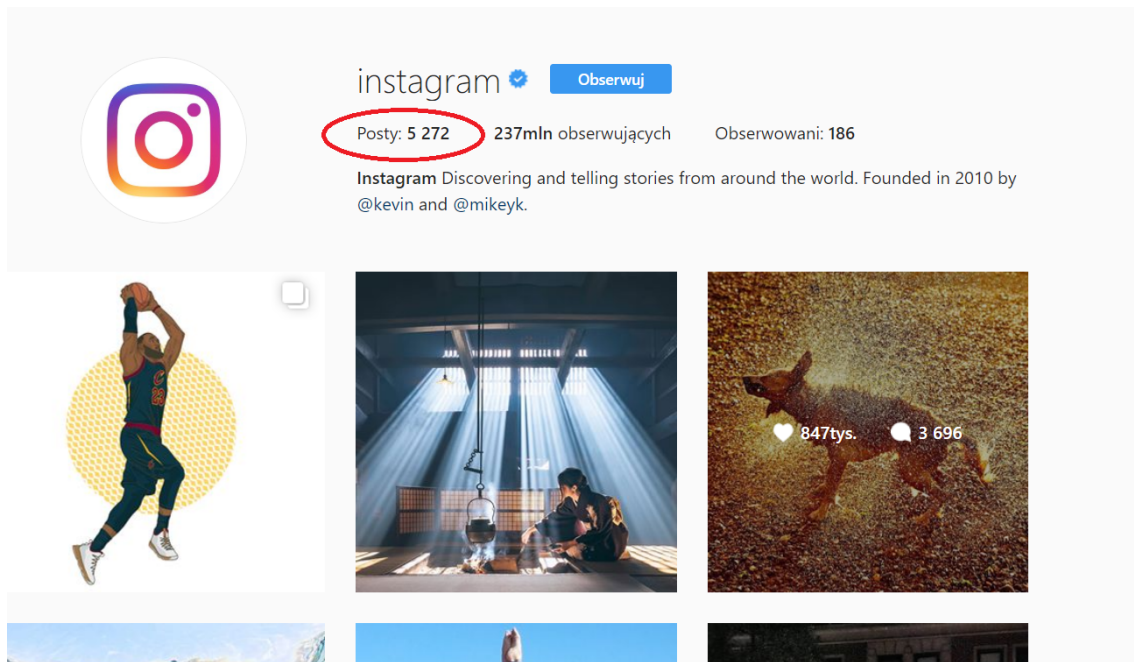
        createImageSet(Jsoup.parse(driver.getPageSource()));
    }
    driver.quit();
    createFullResImgSet();
}
```

Rysunek 6: Wykorzystanie kolekcji typu Set w celu usunięcia duplikatów.

6.3 Pobieranie danych w jak najkrótszym czasie

- Problem

Popularni użytkownicy serwisu Instagram mają udostępnione tysiące zdjęć i filmów. Pobieranie tak dużej ilości danych poprzez aplikację wiąże się z wydłużeniem czasu całej operacji. W dzisiejszych czasach czekanie 20 minut na pobranie nawet dużej zawartości jest wręcz niedopuszczalne.



Rysunek 7: Informacja dotycząca ilości zdjęć na profilu Instagram

- Rozwiązanie

Pobieranie początkowo odbywało się na jednym wątku przez co trwało to bardzo długo. Czas pobierania został skrócony wykorzystując wielowątkowość. Zostaje utworzone tyle wątków pobierania ile rdzeni ma procesor na komputerze, co znacznie przyspiesza pobieranie zamykając ramy czasowe w 5 minutach przy dużej liczbie zdjęć.

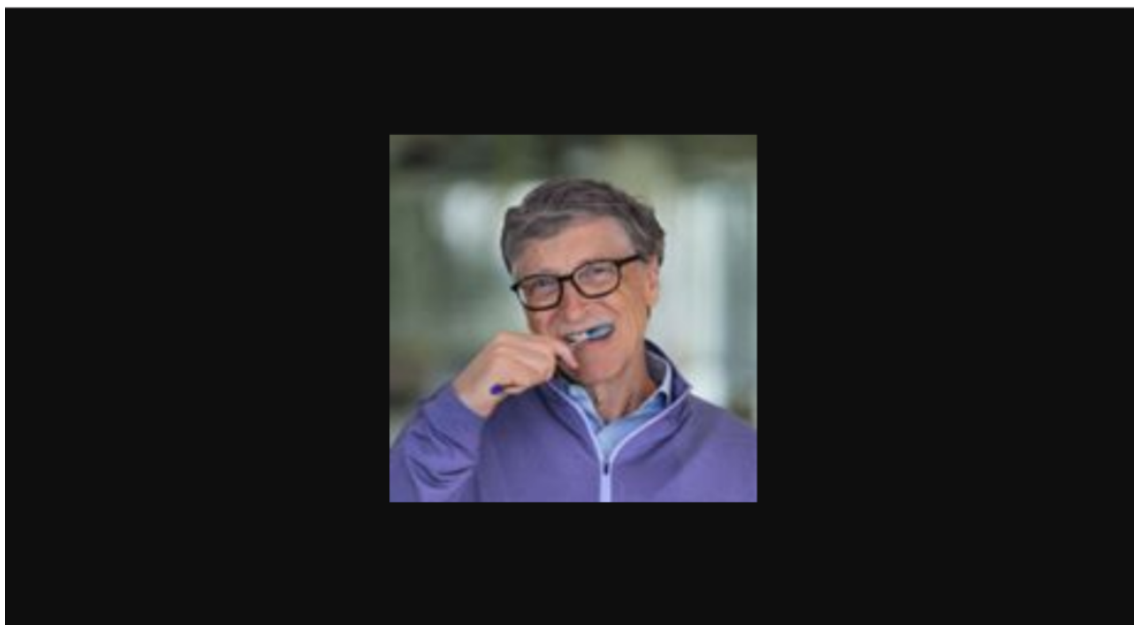
```
ThreadLocal<WebDriver> driverT;  
driverT = (ThreadLocal) initialValue() → { return new ChromeDriver(options); };  
  
final AtomicInteger counter = new AtomicInteger( initialValue: 1);  
  
long startTimer = System.currentTimeMillis();  
  
fullImgUrlSet.parallelStream().forEach((url) -> {  
    driverT.get().get(url);  
  
    Document doc = Jsoup.parse(driverT.get().getPageSource());  
  
    Elements body = doc.select( cssQuery: "main.SCxLW article.QBXjJ");  
    Elements fullImg = body.select("img.FFVAD");  
    if (!fullImg.isEmpty()) {  
        String fullResImgUrl = fullImg.attr( attributeKey: "src");  
        mySet.add(fullResImgUrl);  
    }  
    System.out.printf("Loading photos... %d/%d loaded\n", counter.get(), fullImgUrlSet.size());  
    counter.getAndAdd( delta: 1);  
});  
  
driverT.get().quit();
```

Rysunek 8: Wielowątkowe rozwiązanie parsowania zdjęć.

6.4 Pobieranie danych w pełnej rozdzielczości

- Problem

Zdjęcia na profilach w serwisie Instagram nie są wyświetlane w pełnej rozdzielczości, czyli takiej jaką posiada autor zdjęcia.



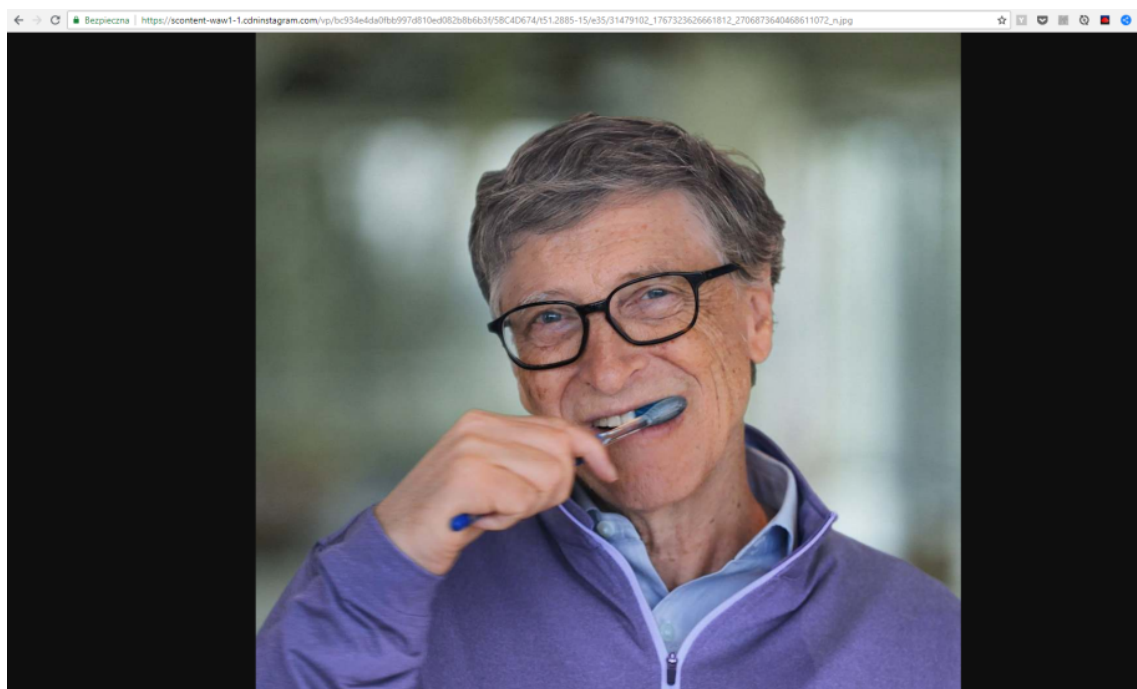
Rysunek 9: Zdjęcie w małej rozdzielczości dostępne dla zwykłego użytkownika.

- Rozwiązanie

Poprzez manipulację linkami, można uzyskać dostęp do oryginalnych zdjęć z serwisu Instagram, które są dostępne w kilku rodzajach rozdzielczości, w tym jakości HD. Jest to wykonywane przez funkcję przedstawioną poniżej:

```
public void createImageSet(Document doc) {  
  
    String staticInstaLink = "https://www.instagram.com";  
  
    Elements body = doc.select(cssQuery: "main.SCxLW div.v9tJq");  
    for (Element el : body.select("article div div div.Nnq7C")) {  
        for (Element image : el.select(cssQuery: "div.v1Nh3 a")) {  
            String fullImgSiteURL = staticInstaLink + image.attr(attributeKey: "href");  
            fullImgUrlSet.add(fullImgSiteURL);  
        }  
    }  
}
```

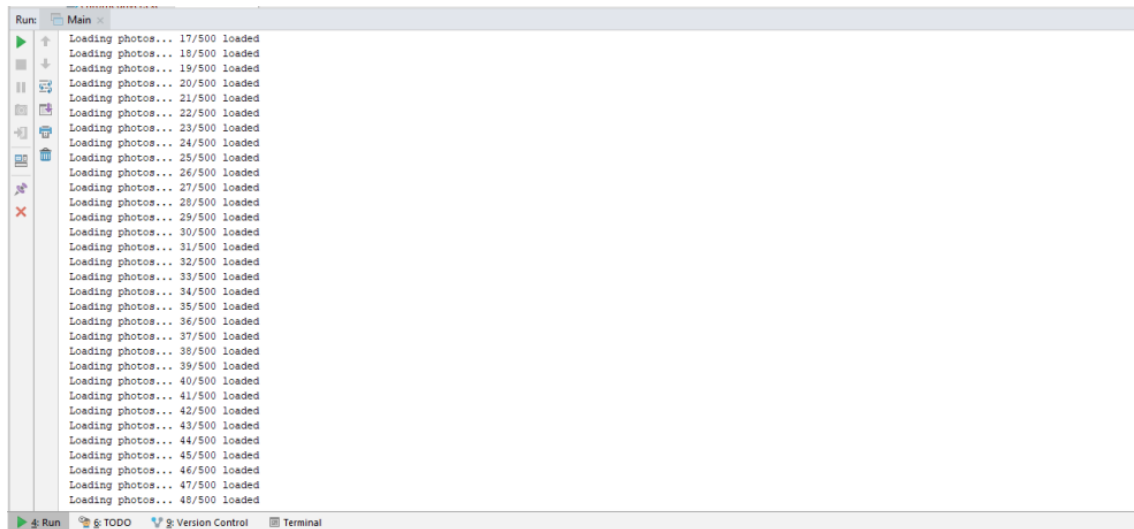
Rysunek 10: Rozwiązanie wykorzystujące manipulacje linkami.



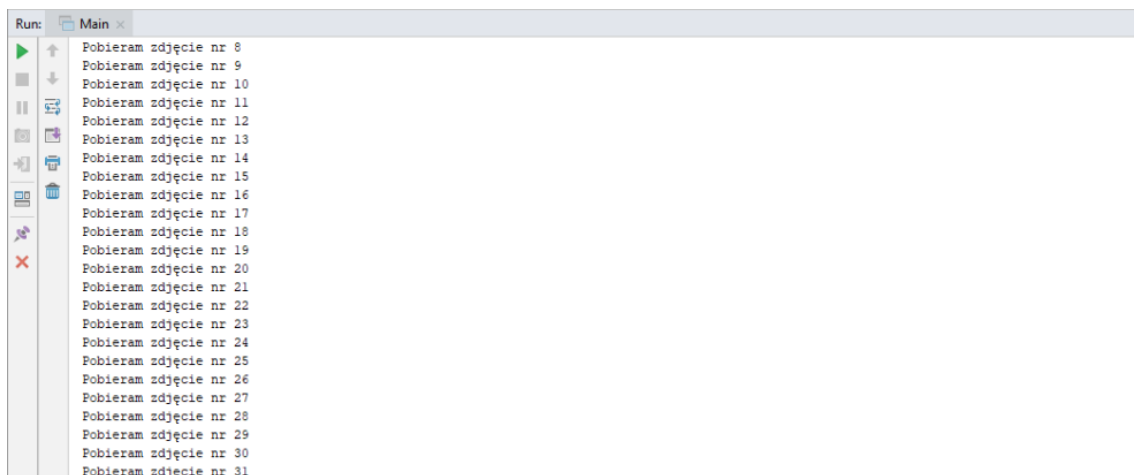
Rysunek 11: Zdjęcie w pełnej rozdzielczości uzyskane sposobem opisanym powyżej.

7 Proces debugowania aplikacji

Podczas wykonywania przez aplikację operacji w konsoli możemy śledzić informacje o aktualnym działaniu. Znajdują się tam m.in. komunikaty o ilości załadowanych obiektów do bufora, czy ilość aktualnie pobranych zdjęć. Pozwala to śledzić postęp realizowanych czynności.



Rysunek 12: Konsola z informacjami o ładowaniu obiektów do bufora.



Rysunek 13: Konsola z informacjami o pobieraniu zdjęć z serwisu.

8 Testy jednostkowe

Testy jednostkowe to metoda testowania wytwarzanego oprogramowania, polegająca na pisaniu metod testujących określone, małe fragmenty naszego programu (jednostki). Jednostkami mogą być np. metody lub klasy.

```
public class PageScraperTest {

    private WebDriver setDriver() {
        System.setProperty("webdriver.chrome.driver", "Path to driver.exe");

        WebDriver driver = new ChromeDriver();
        driver.get("https://www.instagram.com/instagram");
        return driver;
    }

    @Test
    public void testImageCounter() {

        WebDriver testDriver = setDriver();
        ImageScraper testScraper = new ImageScraper(testDriver);
        double expectImageNumber = 5273;

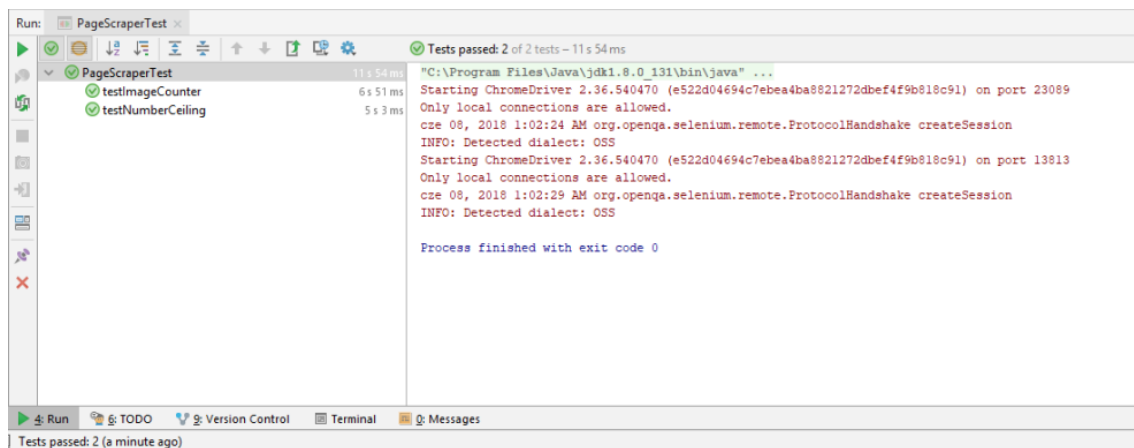
        Assert.assertEquals(expectImageNumber, testScraper.imageCounter(), delta: 0.1);
        testDriver.quit();
    }

    @Test
    public void testNumberCeiling() {
        WebDriver testDriver = setDriver();

        ImageScraper testScraper = new ImageScraper(testDriver);
        int expectedNumber = 1757;

        Assert.assertEquals(expectedNumber, numberCeiling(testScraper.imageCounter()/3));
        testDriver.quit();
    }
}
```

Rysunek 14: Testy jednostkowe napisane do aplikacji.



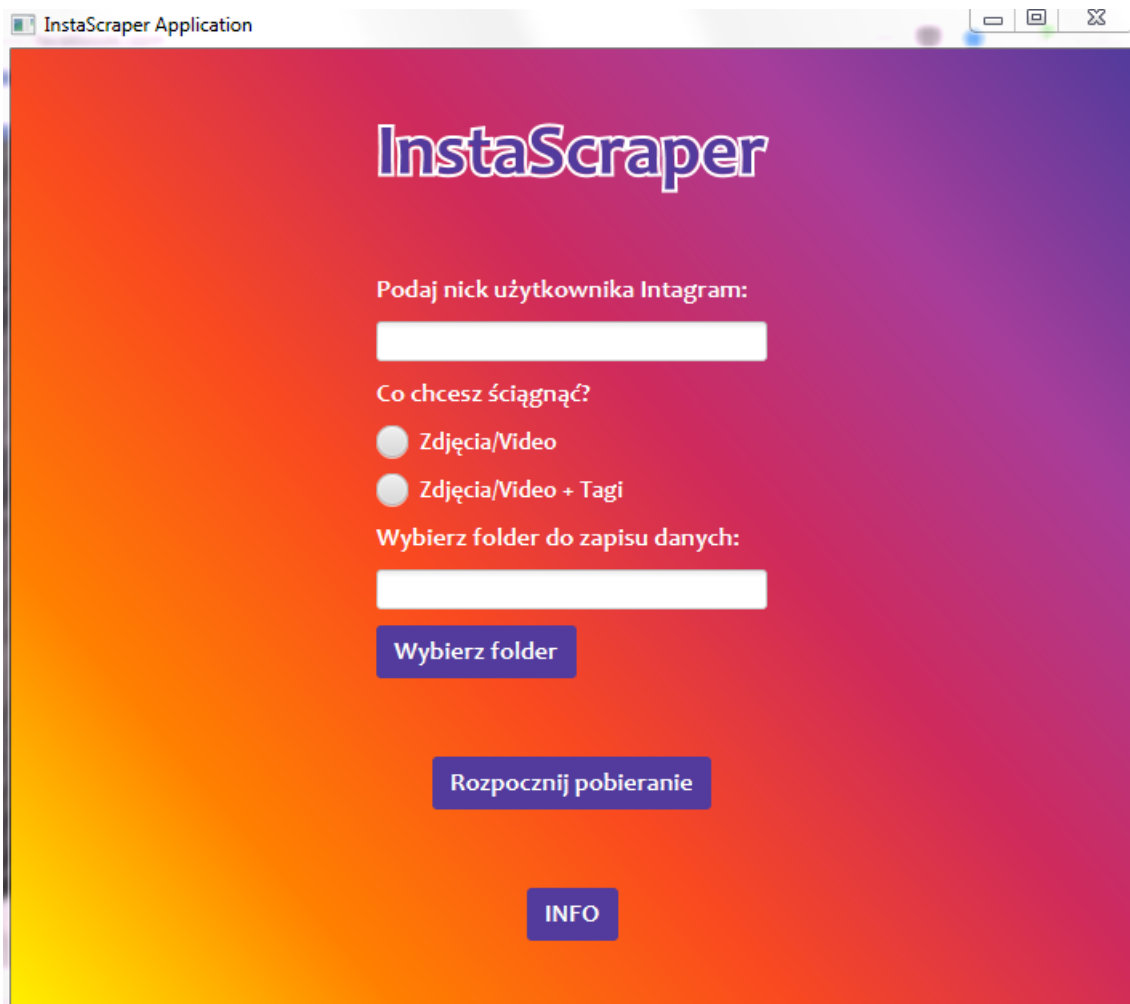
Rysunek 15: Wyniki testów jednostkowych.

9 Projekt interfejsu graficznego

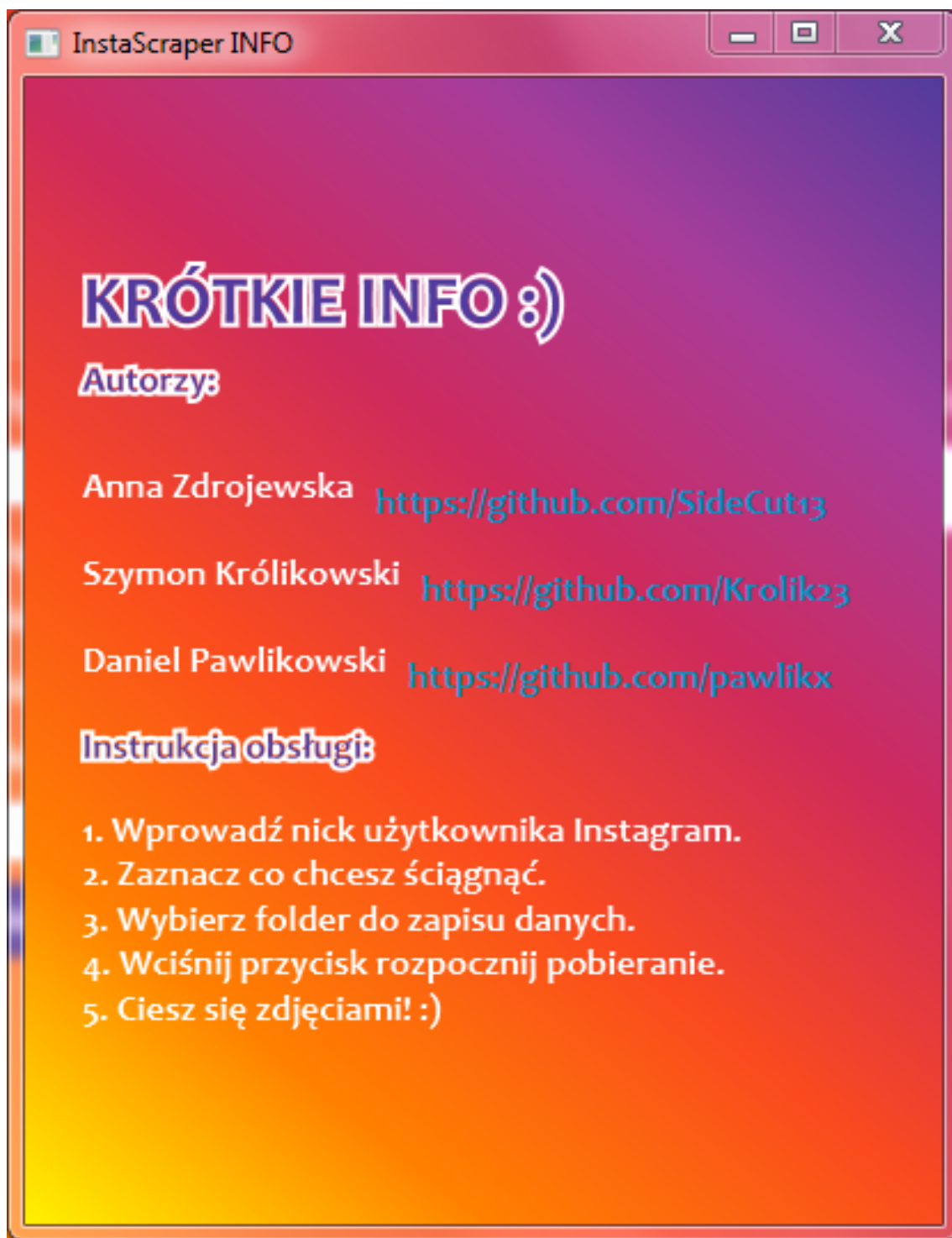
Interfejs graficzny przygotowany dla użytkownika zawiera wszystkie niezbędne elementy do wykorzystania całej funkcjonalności. W aktualnej wersji aplikacji są to:

1. Pole do wprowadzenia nazwy konta osoby z Instragram,
2. RadioButton z wyborem kontentu do pobrania,
3. Wybór folderu na dysku do przechowywania pobranych danych,
4. Przycisk rozpoczęcia pobierania,
5. Przycisk INFO, po wciśnięciu którego wyświetlają się informacje o aplikacji.

Poniżej znajduje się zrzut ekranu obrazujący wygląd interfejsu. Kolory zostały dobrane aby odzwierciedlić logo serwisu Instagram.



Rysunek 16: Interfejs graficzny strony domowej aplikacji.



Rysunek 17: Interfejs graficzny informacji o aplikacji(INFO).

10 Instrukcja użytkowania aplikacji

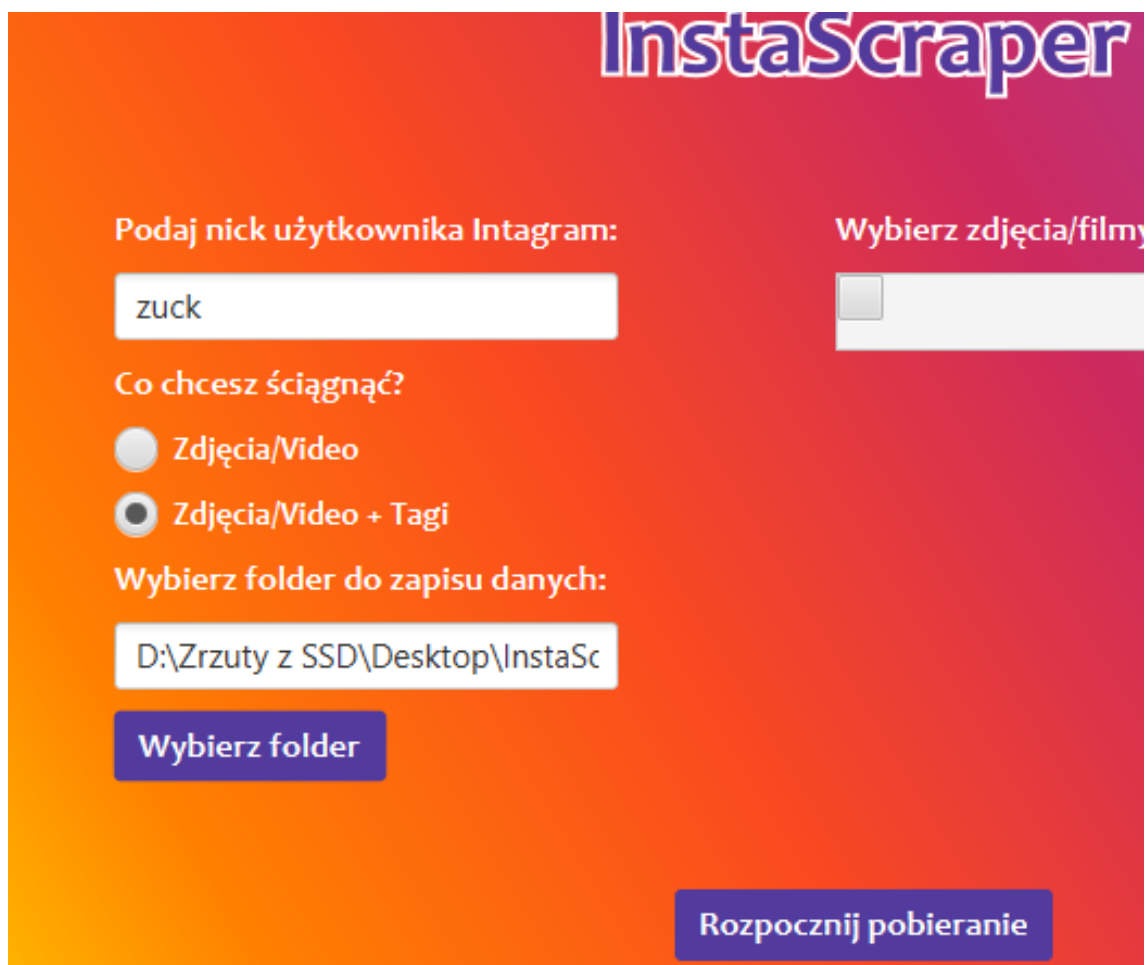
W tej części dokumentacji zawarta jest pełna instrukcja jak krok po kroku korzystać z aplikacji InstaScraper.

10.1 Uruchomienie aplikacji

Do uruchomienia aplikacji niezbędny jest dostęp do Internetu oraz przeglądarka Google Chrome. Należy mieć również zainstalowaną JRE Javy. Samo uruchomienie następuje poprzez dwukrotne kliknięcie lewym przyciskiem myszy na plik wykonywalny aplikacji.

10.2 Obsługa aplikacji

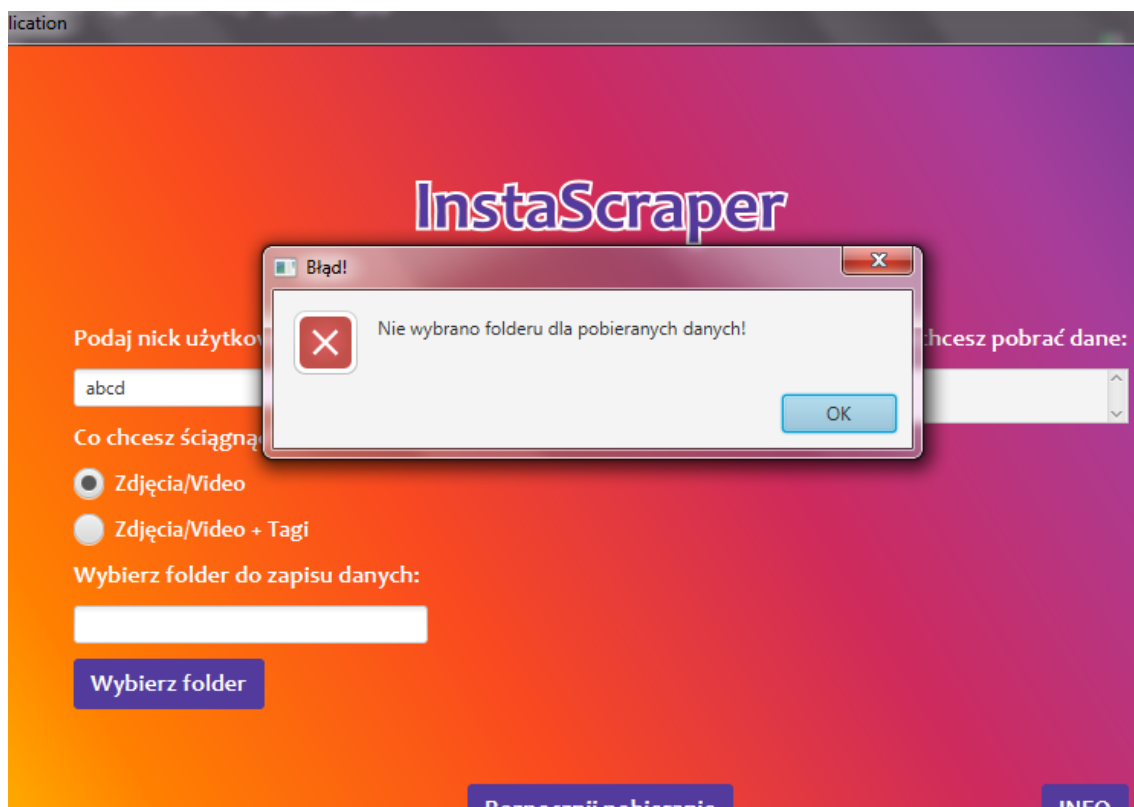
Po uruchomieniu należy wpisać nazwę konta użytkownika Instagram w polu tekstowym. Następnie należy wybrać jakie dane chcemy pobrać. Do wyboru jest: zdjęcia/wideo oraz zdjęcia/wideo + tagi. Następnie trzeba wyspecyfikować folder, w którym chcemy zapisać pobierane dane. Po wprowadzeniu wszystkich danych należy kliknąć przycisk "Rozpocznij pobieranie".



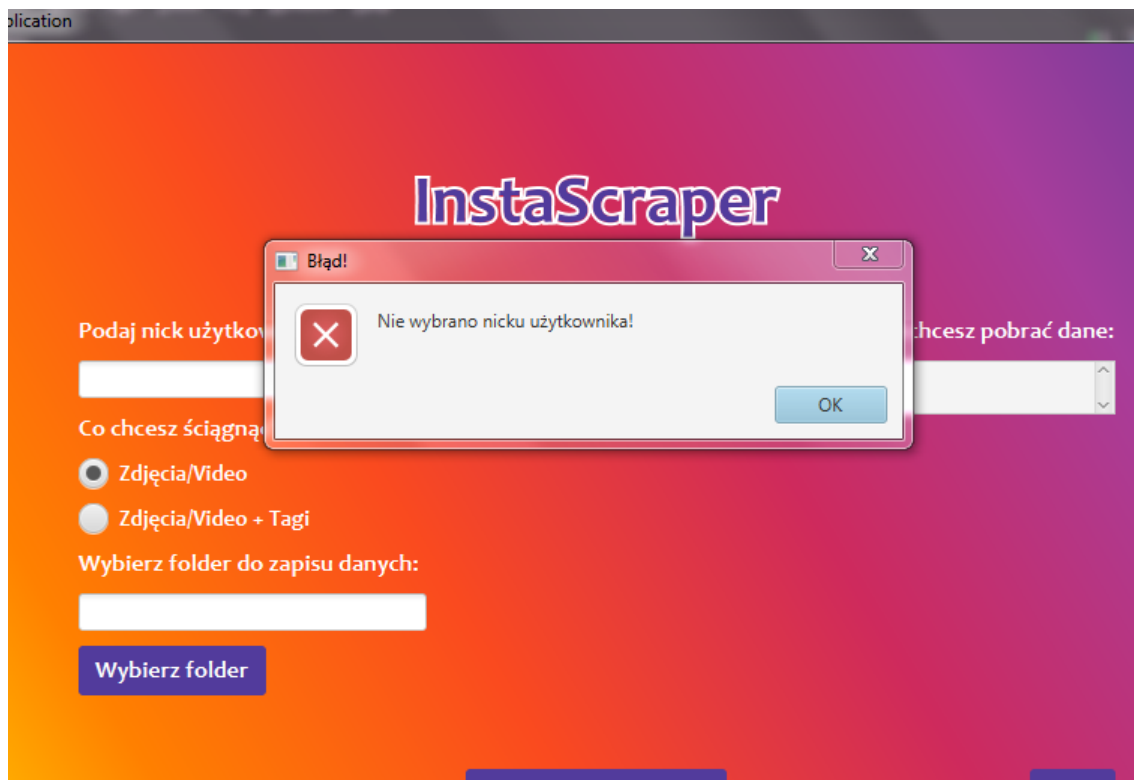
Rysunek 18: Wprowadzanie danych przed rozpoczęciem pobierania.

10.3 Walidacja

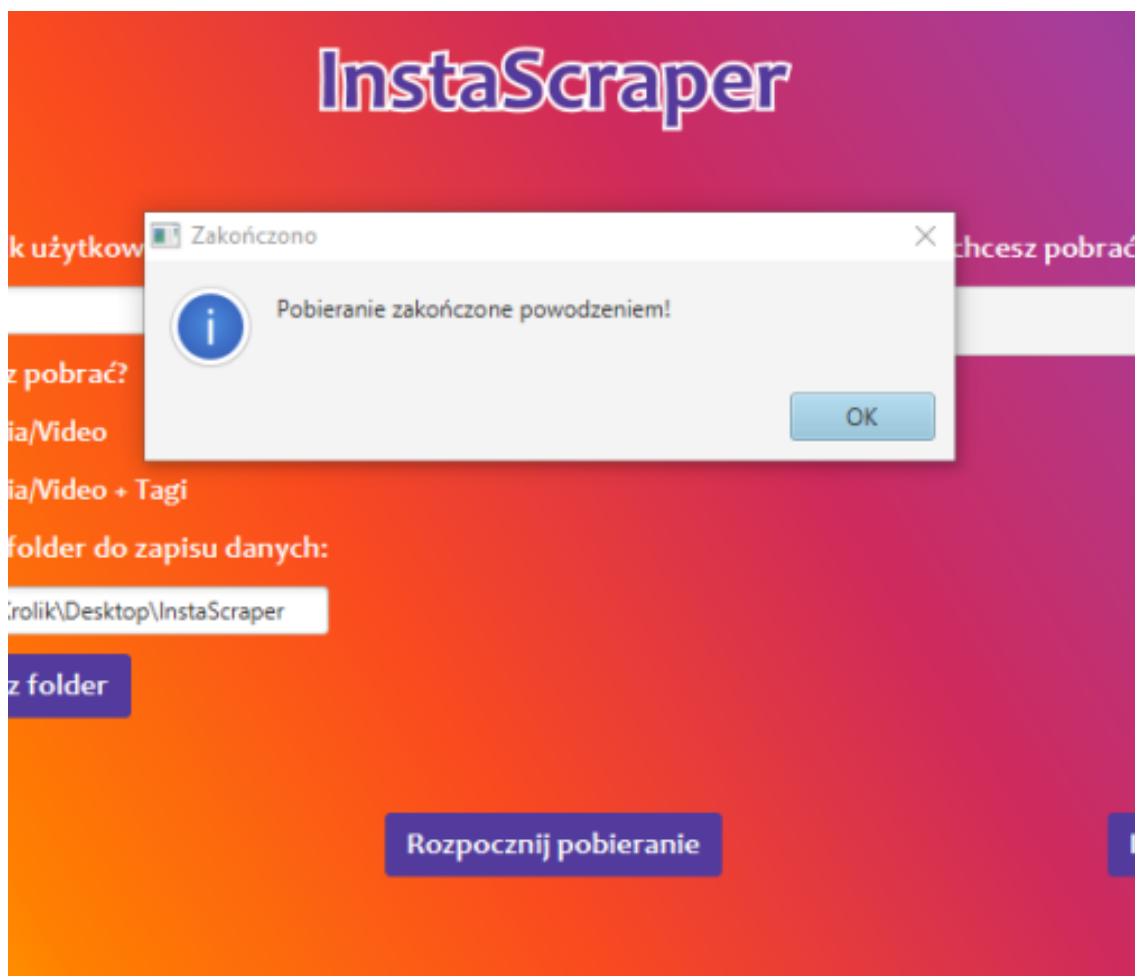
Aplikacja została częściowo zabezpieczona przed błędami użytkownika. Aby rozpocząć pobieranie musi zostać wpisana nazwa użytkownika, w przeciwnym wypadku skutkuje to pojawieniem się komunikatu. Dodatkowo folder do danych musi być wyspecyfikowany. Jeżeli nie jest to pojawia się komunikat z prośbą o jego wybranie. Po zakończeniu pobierania również pojawia się komunikat o poprawnym pobraniu danych.



Rysunek 19: Komunikat o braku wybrania folderu.



Rysunek 20: Komunikat o pustym polu z nazwą użytkownika.



Rysunek 21: Komunikat o pomyślnym zakończeniu pobierania.

11 Perspektywy rozwoju

Aplikacja, której poświęcona jest niniejsza dokumentacja może być dalej rozwijana. Pojawiło się kilka pomysłów na dodatkowe funkcjonalności, które mogłyby zostać zaimplementowane w przyszłości. Poniżej przedstawiono propozycje rozszerzenia aplikacji:

1. "Subskrybcja" użytkownika

Sprawdzanie przy włączaniu aplikacji, czy osoby z listy obserwowanych nie dodały nowego kontentu. Jeśli tak, to aplikacja pobierałaby owe elementy automatycznie.

2. Pobieranie lokalizacji

Dodanie do aplikacji opcji pobrania lokalizacji użytkownika ze zdjęć/wideo. Taka funkcjonalność pozwalałaby, na przykład zweryfikować, czy nowe zdjęcie wieży Eiffla faktycznie pochodzi z Paryża a nie jest to tylko skserowana pocztówka.

3. Pobieranie komentarzy

Dodanie opcji pobierania komentarzy pod zdjęciami, w celu ich analizy.

4. Pobieranie zawartości profili prywatnych

Ta wersja aplikacji zawiera tylko dostęp do konktentu na publicznych profilach użytkowników Instagram. Można byłoby rozszerzyć dostęp również o konta oznaczone jako prywatne. Jest to zadanie trudne do wykonania, lecz możliwe z teoretycznego punktu widzenia. Dostęp do tego typu strzeżonej zawartości mógłby zostać uzyskany poprzez odpowiednie odwoływanie się do konkretnego serwera aplikacji.

5. Analiza pobranych danych

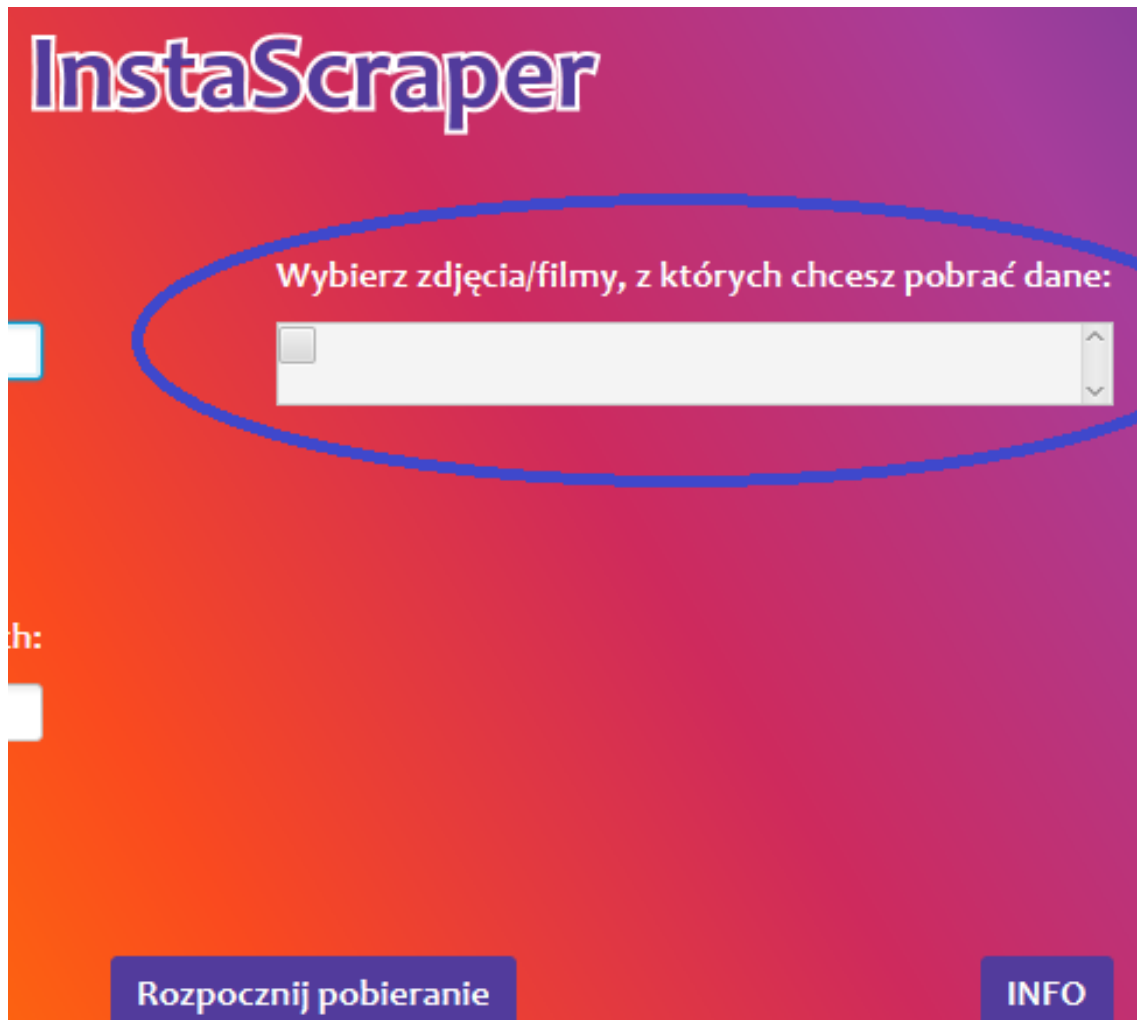
Stworzenie podsystemu, który analizowałby pobrane tagi, komentarze, lokalizację i wnioskował na ich podstawie.

6. Pasek stanu ładowania zdjęć profilu wraz z przewidywanym czasem

Funkcjonalnością jaka zostanie dodana na drodze dalszego rozwoju aplikacji, będzie pasek stanu ładowania zdjęć wybranego profilu w serwisie instagram. Wspomniane udogodnienie będzie szczególnie przydatne w przypadku parsowania kont o większej objętości". Profile tego typu potrafią zawierać nawet do 7000 zdjęć, pasek stanu da znać użytkownikowi, o tym że wszystko przebiega zgodnie z założoną architekturą.

7. Pobieranie jedynie wyselekcjonowanych przez użytkownika zdjęć.

Jest to funkcjonalność, której nie udało się zaimplementować ze względów czasowych. Użytkownik zaznaczałby zdjęcia z konkretnego profilu, a następnie aplikacja zamiast pobierać wszystkie zdjęcia, pobrała by tylko zaznaczone.



Rysunek 22: Panel wyboru zdjęć do pobrania.

12 Bibliografia

Źródła jakie zostały wykorzystane przy tworzeniu aplikacji InstaScraper.

Literatura

- [1] Forum internetowe dla programistów
<https://stackoverflow.com/> (dostęp 7.06.2018),
- [2] Strona internetowa API Instagram
<https://www.instagram.com/developer/> (dostęp 7.05.2018),
- [3] Strona internetowa dokumentacji biblioteki.jsoup
<https://jsoup.org/apidocs/overview-summary.html> (dostęp 7.06.2018),
- [4] Strona internetowa dokumentacji framework'a Selenium
<https://www.seleniumhq.org/docs/> (dostęp 7.06.2018),
- [5] Strona internetowa dokumentacji języka programowania Java 8
<https://docs.oracle.com/javase/8/docs/api/> (dostęp 7.06.2018),
- [6] Strona internetowa do tworzenia dokumentacji za pomocą narzędzia LaTeX
<https://www.overleaf.com/> (dostęp 7.06.2018),
- [7] Strona internetowa narzędzia do budowy projektów Maven
<https://maven.apache.org/> (dostęp 7.06.2018),
- [8] Strona internetowa narzędzia do tworzenia diagramów UML Visual Paradigm
<https://www.visual-paradigm.com/> (dostęp 7.06.2018),
- [9] Strona internetowa narzędzia do tworzenia dokumentacji
<https://www.overleaf.com/> (dostęp 7.06.2018).