Obraz zawierający tekst

Opis wygenerowany automatycznie

**Information Management**

**Zarządzanie Informacją**

IT Management

**Szymon Wujec**

S20431

**“Data suggestion tool in the digital age using web application “GeoFinderUI” to optimize work time on example of geo-data”**

Engineer Thesis

Thesis supervisor

Mgr inż. Maksymilian Kubica

Warsaw, January, 2023

Spis treści

[1 Introduction 4](#_Toc126794111)

[1.1 Thesis structure description 4](#_Toc126794112)

[1.2 Goals of the thesis 4](#_Toc126794113)

[1.3 Main thesis issues 4](#_Toc126794114)

[1.3.1 Adres URL 4](#_Toc126794115)

[1.3.2 Aplikacja webowa 5](#_Toc126794116)

[1.3.3 Bot 5](#_Toc126794117)

[1.3.4 Web Scraping 5](#_Toc126794118)

[1.3.5 Framework 6](#_Toc126794119)

[1.3.6 Refactoring 6](#_Toc126794120)

[1.3.7 Markup language 8](#_Toc126794121)

[2 Description of application. 11](#_Toc126794122)

[2.1 Description introduction 11](#_Toc126794123)

[2.2 Front-end description 11](#_Toc126794124)

[2.3 Back-end description 12](#_Toc126794125)

[2.4 Tools 13](#_Toc126794126)

[2.4.1 IntelliJ IDEA 13](#_Toc126794127)

[2.4.2 Excel 14](#_Toc126794128)

[2.4.3 Python 15](#_Toc126794129)

[2.4.4 Selenium 16](#_Toc126794130)

[2.4.5 Pandas 17](#_Toc126794131)

[2.4.6 Openpyxl 18](#_Toc126794132)

[2.4.7 HTML 19](#_Toc126794133)

[2.4.8 Google Maps 20](#_Toc126794134)

[2.4.9 Materialize 21](#_Toc126794135)

[3 Functionality process assumptions 23](#_Toc126794136)

[4 Implementation. 26](#_Toc126794137)

[4.1 Front-end code 26](#_Toc126794138)

[4.1.1 Home page 26](#_Toc126794139)

[4.1.2 Summary page 30](#_Toc126794140)

[4.2 Back-end code 31](#_Toc126794141)

[4.2.1 Initialization classes 31](#_Toc126794142)

[4.2.2 Callable classes 37](#_Toc126794143)

[5 Summary 48](#_Toc126794144)

[5.1 Application outcome 48](#_Toc126794145)

[5.2 Podziękowania 48](#_Toc126794146)

[6 Bibliography 49](#_Toc126794147)

[7 Attachment 50](#_Toc126794148)

[7.1 instructions 50](#_Toc126794149)

[7.2 Table of Figures 50](#_Toc126794150)

[7.3 Table of Pictures 51](#_Toc126794151)

[7.4 Table of Code 51](#_Toc126794152)

[7.5 What CD contains 52](#_Toc126794153)

# Introduction

## Thesis structure description

Głównym celem pracy było stworzenie narzedzia ktore pomoze pracownikom pracujacym w korporacjach, lepiej zarzadzac i ograniczac czasowo zbedne zadania, ktore nic nie wnosza do wlasnego rozwoju pracownika. Pomysl na taka aplikacje webowa powstal na podstawie moich osobistych doswiadczen i wyciagnietych wniskow z zadan zlecanych mi w w firmie w ktorej pracowalem. Ta korporacja oferuje szeroko pojete uslugi w kontekscie zarzadzania portfelem klienta w sklad ktorych wchodza duze firmy jak i male start-upy. Na portfel klienta sklada sie jego wlasnosc, a temat pracy opiera sie na powierzchniach geolokalizacyjnych. Konkurencja od ktorej klient przeszedl nie zbierala wymaganych oraz kluczowych dla nas informacji, zadanie ktore otrzymalem bylo przeszukiwanie internetu w oparciu o wyniki wyszukiwania w Google Maps oraz uzpelnianie brakujacych danych.

## Goals of the thesis

Aplikacja “GeoFinderUI” korzystajac z dostepnych baz danych w internecie bedzie uzupelniac danymi sugestyjnymi, wybiorcze dane przeslane do aplikacji za pomoca zestawienia minimalnych informacji na temat konkretnej lokalizacji w arkuszu kalkulacyjnym. Ma to na celu zmniejszyc udzial uzytkownika w przeszukiwaniu wynikow w internecie oraz usprawnic prace zwiazana z zadaniami wymagajacymi zbierania danych geolokalizacyjnych konkretnej lokalizacji.

## Main thesis issues

### Adres URL

Adres URL to Uniform Resource Locator, Najczesciej kojarzony ze stronami World Wide Web (WWW) to nic innego jak ujednolicony format adresowania informacji, aplikacji lub danych na stronach internetowych lub w sieci lokalnej. Uzytkowik po wpisaniu konkretnego, istniejacego adresu URL w swojej przegladarce jest przekierowywany do strony z przypisanym, podanym przez uzytkownika adresem.

(Uniform Resource Locator, 2022)

### Aplikacja webowa

Aplikacja webowa to program uruchomiony na serwerze zewnetrznym lub lokalnym z mozliwoscia polaczenia, komunikowania sie z nia oraz jej uzytkowania przez siec komputerowa z hostem uzytkownika na jego prywatnej sesji. W tym celu wykorzystuje sie urzadzenia kompatybilne posiadajace przegladarke internetowa, takie jak komputer personalny, laptop, smartfon, smartwatch, smartTV i wiele innych tego typu urzadzen.

(Aplikacja internetowa, 2022)

### Bot

Bot to program wykonujacy z gory zalozona czynnosc zamiast czlowieka, a w wielu przypadkach ma za zadanie imiotwac ludzkie zachowanie, ktore nie wykracza poza ramy zdefiniowanej czynnosci. Slowo Bot pochodzi od slowa robot co oznacza zautomatyzowana maszyne. W dzisiejszych czasach zwiekszonego rozwoju technologicznego, na kazdym kroku spotykamy sie z takimi maszynami, w kuchni, w drodze do pracy, szkoly, a nawet w sklepach. Popularnym miejscem wykorzystywania botow sa fora spolecznosciowe, gdzie boty maja za zadanie, pod nieobecnosc personelu nadzorowac emotowana przez uzytkownikow tresc aby byla ona zgodna z regulaminem portalu. Niestety nie wszedzie boty maja przypisane zadanie pilnowania przestrzegania zasad, przykladowo gry FPS lub TPS gdzie boty, znane jako aimboty, wspomagaja graczy w celowaniu i same automatycznie ustawiaja celownik. Jest to niezgodne z zasadami tego typu gier, przez co czesto trzeba sie liczyc z blokada konta nalozona przez administratora za uzywanie wspomagaczy.

(Bot program, 2022)

### Web Scraping

Web scraping to jest proces wykorzystywany w srodowisku internetowym, wykonywany przez boty, ktore zostaly stworzone przez programiste w celu zautomatyzowanego pozyskiwania danych widniejacych w sieci komputerowej, na stronach internetowych World Wide Web. W przeciwienstwie do procesu zwanego screen scraping, ktory tylko kopiuje wyswietlane na ekranie monitora piksele, web scraping odczytuje caly kod zrodlowy HTML, konkretnej strony internetowej. Nastepnie za pomoca wyrazow kluczowych, polozenia na stronie czy typie zamieszczonych informacji, naprzyklad takich jak naglowek albo tytul strony, bot zbiera informacje interesujace dla autora kodu. Niestety wiele stron broni sie przed ulatwionym oraz masowym gromadzeniem danych, przez co wiele witryn internetowych automatycznie wykrywa i blokuje sesje pod pretekstem korzystania z botow.

(Web scraping, 2023)

### Framework

Framework to szkielet do aplikacji webowej. Kazdy framework posiada zestaw komponentow i dostepnych bibliotek przystosowanych do zaprogramowania aplikacji. Istrnieje wiele dostepnych frameworkow, ktore sie od siebie roznia, przez to mozna dobrac odpowiednia platforme programistyczna dla swojej aplikacji w zaleznosci od wymaganej struktury czy ogolnego mechanizmu dzialania programu. Zaletami korzystania z frameworkow podczas programowania sa efektywosc, poprawa jakosci kodu oraz niezawodnosc. Frameworki przez posiadanie zdefiniowanych oraz zapisanych juz narzedzi, wymagaja od programisy wykorzysujacego konkretna platforme programistyczna mniej kodu do napisania, co przeklada sie na efektywnosc. Podobnie sprawa sie tyczy poprawy jakosci kodu, z gory zalozona struktura zapisanego kodu w sposob logiczny i przemyslany tworzona jest z mysla o elastycznosci programu oraz rownoczesnie wplywa na jakosc zapisanej przez programiste tresci. Natomiast niezawodnosc jest uzyskana przez logiczny i przemyslany zapis frameworku, ktory przeszedl wiele zlozonych testow zanim zostal wydany w oficjalnej wersji do uzytku publicznego. Obszary zastosowania frameworkow to: kompilatory różnych języków programowania, aplikacje modelowania finansowego, aplikacje do modelowania w naukach o Ziemi, systemy, wspomagania decyzji, frameworki multimedialne, aplikacje internetowe, middleware, frameworki do aplikacji webowych

(Framework, 2022)

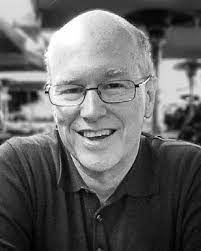
### Refactoring

Refactoring to nazwa techniki polegajacej na restrukturyzowaniu istniejacego body kodu ingerujac w jego strukture, nie zmieniajac przy tym dzialalnosci i funkcjonalnosci zapisanego skryptu. Ten zabieg jest bardzo popularny wsrod programistow aby nie powielac wielokrotnie wystepujacej czesci kodu, lecz zeby przy uzyciu zmiennych wywolywac konkretna jedna metode czyniac ja tym samym uniwersalna. W efekcie, skrypt przybiera postac przejrzystego, czytelnego i klarownego kodu, ktory swoja obietoscia zajmuje znacznie mniej kodu niz przed zabiegiem refactoringu. Pierwszy znany przypadek uzycia terminu refactoring mial miejsce w roku 1990 w publikowanej literaturze. Autorami artykulu byli William Opdyke i Ralph Johnson. Martin Flower jest brytyjskim software developerem. Jego doswiadczenie w tej branzy przekazuje kolejnym pokoleniom w wielu publicznych wypowiedziach, publikacjach i ksiazkach. Jedna z tych ksiazek jest ksiazka pod tytulem “Refactoring: Improving the design of Existing Code” wydanej w roku 2002 w kolaboracji z wspomnianym wczesniej Williamem Opdyke.

(Code Refactoring, 2023)

“Help in understanding the code also helps me spot bugs. I admit I'm not terribly good at finding bugs. Some people can read a lump of code and see bugs, I cannot. However, I find that if I refactor code, I work deeply on understanding what the code does, and I put that new understanding right back into the code. By clarifying the structure of the program, I clarify certain assumptions I've made, to the point at which even I can't avoid spotting the bugs. 49 It reminds me of a statement Kent Beck often makes about himself, "I'm not a great programmer; I'm just a good programmer with great habits." Refactoring helps me be much more effective at writing robust code.”

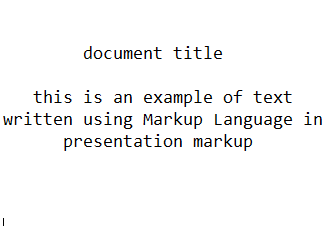
(Flower, Beck, Opdyke, & Brant, 2002)

**Picture 1 Left side, Martin Flower Picture 2 Right side, William Opdyke**

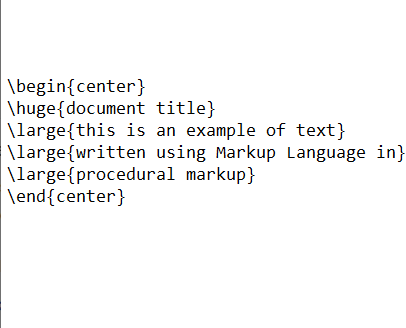
### Markup language

Markup language to nic innego jak forma zapisu kodu zlozona z tekstu i znacznikow reprezentujacych dany element znajdujacy sie w kodzie. Przykladem najpopularniejszego markup language jest jezyk “HTML (HyperTextMarkupLanguage)”, technologia ta jest podstawa stron “World Wide Web (WWW)”. “HTML” na tyle zdominowal rynek podobnych technologii, ze mowiac o kodzie, ktory jest wynikiem polaczenia tekstu i znacznikow, pierwsza mysla odbiorcy prawdopodobnie bedzie “HyperTextMarkupLanguage”, natomiast nie jest to jedyny jezyk z tej kategorii, technologii formatowania tekstu. W rzeczywistosci istnieje wiele takich jezykow, mozemy je spotkac jako stosowane technologie w srodowiskach takich jak business, economy, finance, culture, media, entertainment, science, technology, engineering, and mathematics. Pierwszymi Markup Languages poslugiwano sie, oraz posluguje sie do dzis w przemyslach papierniczych, redaktorskich i wydawniczych. Wiele dostepnych jezykow przystosowanych do roznych srodowisk tematycznych, skutkuje podzialem markup languages na 3 grupy tych jezykow. Znakowanie prezentacyjne, znakowanie proceduralne oraz znakowanie opisowe. Znakowanie prezentacyjne nie posiada w sobie wymogu szczegolnego formatowania dokumentu, jednyna cecha charakterystyczna jest struktura zapisanych informacji. Przykladowo korzystajac ze znakowania prezentacyjnego, w celu zaznaczenia marginesow oraz uzaskaniu efektu centrowania tekstu, stosuje sie zapis kilku znakow rozpoczynajacych nowa linie lub znak spacji, ktore poprzedzaja tekstowa wartosc informacyjna znajdujaca sie w omawianej lini dokumentu.



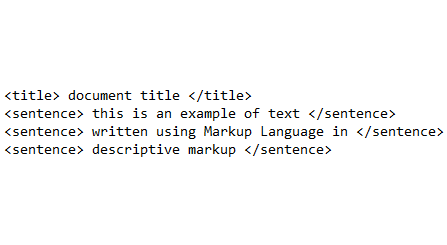
**Figure 1 Presentation markup example**

Znakowanie proceduralne, podobnie jak poprzedni przyklad, odpowiada za prezentacje tekstowej wartosci informacyjnej. Roznica natomiast pomiedzy znakowaniem proceduralnym, a znakowaniem prezentacyjnym jest taka, ze znacznikami nie sa juz znaki spacji czy rozpoczecia nowej lini, tylko pewnego rodzaju komendy skierowane do programu lub urzadzenia odczytujacego dany dokument aby w trakcie wyswietlania zawartego w dokumencie tekstu, uzytkownik koncowy nie widzial uzytych znacznikow przez autora pierwotnego zapisu dokumentu.



**Figure 2 Procedural markup example**

Przedstawiony powyzej przyklad formatowania tekstu w procedural markup, zostal zapisany przy uzyciu jezyka “TeX”. Ostatnim typem formatowania tekstu przy uzyciu markup language jest znakowanie opisowe. Ten typ w odroznieniu od poprzednich dwoch wspomnianych odmian, nie odnosi sie bezposrednio do formatowania ukladu tekstu na stronie dokumentu lecz przy pomocy odpowiedniego oznakowania wydzielonych fraz autor jest w stanie okreslic co oznacza oraz jaka role w calym dokumencie odgrywa wyodrebniony fragment tekstu.



**Figure 3 Descriptive markup example**

Wiele jezykow z kategorii jezykow stworzonych do formatowania tekstu oraz przekazywanych w ten sposob danych, korzysta w swojej strukturze z zapisu danych z polaczenia wymienionych powyzej odmian markup languages. Przykladem takiego jezyka jest “HTML” w ktorym dominuje zapis descriptive markup.

(Markup language, 2023)

# Description of application.

## Description introduction

Aplikacja “GeoFinderUI” to narzedzie hostowane na przyklad korzystajac z lokalnego lub zewnetrznego serwera. Dostep do niej jest za posrednictwem przegladarki uzytkownika. To czyni ja aplikacja webowa. Po wpisaniu wlasciwego adresu URL uzytkownik jest przekierowywany na internetowa strone startowa aplikacji.

## Front-end description

Zaczynajac od gornej czesci strony, znajduje sie na niej przycisk “Download file here”. Jest to przycisk urachamiajacy procedure pobierania pliku. W tym przypadku jest to plik “Template.xlsx”. Jest to przykladowy format wprowadzania danych za posrednictwem arkuszy kalkulacyjnych Excel firmy Microsoft z pakietu Microsoft Office. Przykladowy plik zawiera pozycje takie jak: nazwa miejsca, adres szukanej lokalizacji, miasto oraz kod pocztowy. Za przyklad opisze lokalizacje naszej uczelni, “Polsko-Japonskiej akademi Technik Komputerowych”. Wartosci komorek tworzacych pelny rzad z danymi prezentuje sie nastepujaco Nazwa: PJATK, adres: Koszykowa 86, miasto: Warszawa, kod pocztowy: 02-008. Ponizej znajduje sie przycisk otwierajacy okno eksploratora plikow o tresci “Choose File” w celu wybrania uzupelnionego danymi, pliku “Template.xlsx”. Po odnaleznieniu lokalizacji gdzie ten plik jest zapisany na lokalnym dysku uzytkownika oraz zatwierdzeniu swojego wyboru, w celu uruchomienia glownego procesu aplikacji nalezy kliknac przycisk “Proceed the file”. Na samym dole strony znajduje sie przycisk typu “drop-box”, ktory pozwala zdecydowac uzytkownikowi ile opoznienia w sekundach program powinien sie wstrzymac w momencie automatycznego wyszukiwania fraz na stronach internetowych. Taki zabieg jest konieczny aby uniknac lub zmniejszyc ilosc wystepowania w otrzymanych danych wartosci rownej “N/A”, znaczacej o nie odnalezieniu szukanej przez program informacji. Dzieje sie tak z powodu roznej stabilnosci oraz predkosci polaczenia internetowego. W takim przypadku jezeli uzytkownik jest swiadomy o slabej jakosci swojego polaczenia z internetem, powinien wybrac jedna z liczb z rekomendowanego przedzialu od 1 sekundy do 3 sekund opoznienia, gdzie im wieksza liczba, zwieksza szanse na odnaleznienie szukanej wartosci. Trzeba tez miec na uwadze, ze im wieksza wartosc ustawimy tym bardziej zwiekszy sie czas otrzymania szukanych informacji.

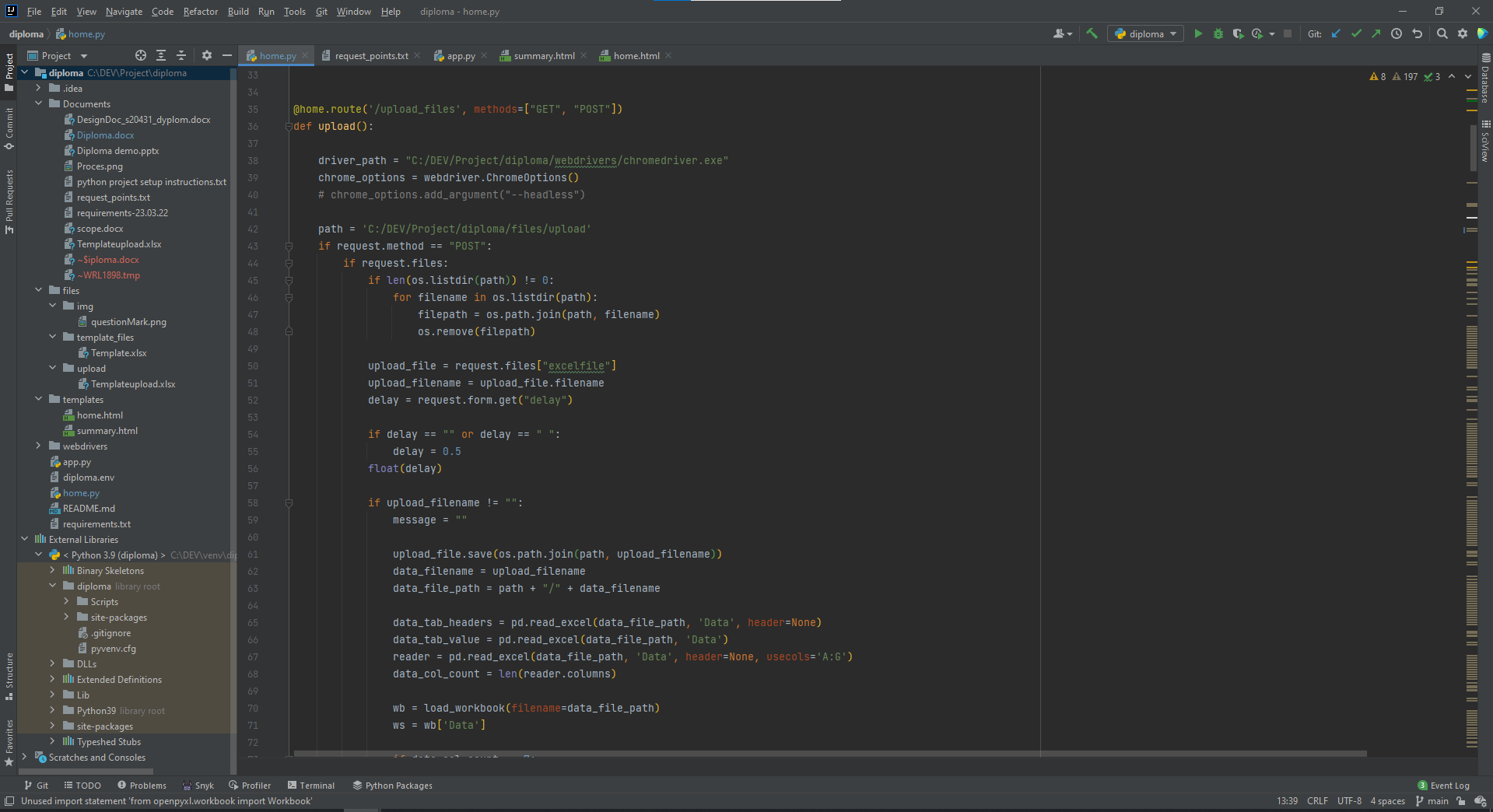
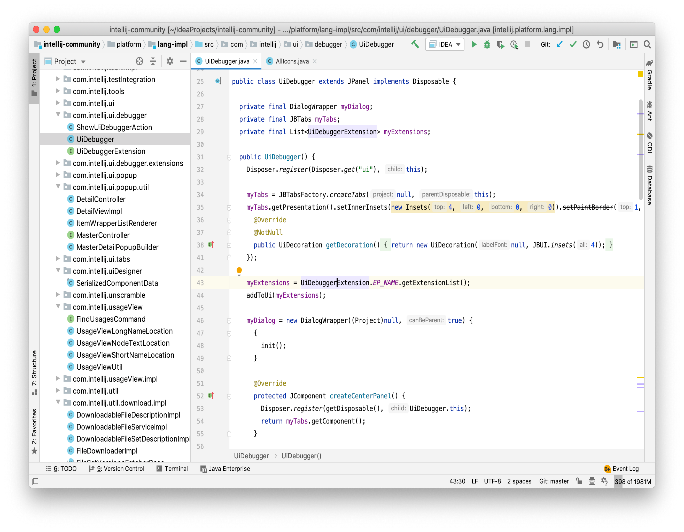
## Back-end description

Uzytkownik po zalaczeniu wszystkich potrzebnych plikow i ustawieniu odpowiednich dla siebie parametrow, zatwierdzeniu ich oraz uruchomieniu glownego procesu aplikacji poprzez przycisk “Proceed the file” zostaje pozostawiony w etapie ladowania strony internetowej. W tym czasie program zapisany w jezyku “Python” o nazwie pliku zawierajacego glowny skrypt “home.py”, pobiera i analizuje wszystkie dostarczone przez uzytkownika informacje aby w efekcie koncowym zwrocic uzytkownikowi plik wypelniony danymi sugestyjnymi przygotowanymi przez program. W tym procesie, pierwszym etapem jest sprawdzenie czy uruchomiona metoda posiada wartosc “POST”. Gdy ten warunek jest spelniony, nastepnym warunkiem jest czy uzytkownik przed uruchomieniem procedury przekazal plik arkusza kalkulacyjnego z danymi.. Nastepnie program przystepuje do sprawdzenia folderu zapisanego w plikach aplikacji “GeoFinderUI”, ktory w kolejnych etapach bedzie przeznaczony do przeslania, zapisania oraz przechowywania pliku “Excel” o podstawowej nazwie “Template.xlsx” zawierajacego dostarczone przez uzytkownika dane. Jezeli program napotka jakiekolwiek pliki tam sie znajdujace, zostana one usuniete. Takie zjawisko ma miejsce w momencie jezeli program nie jest uzywany pierwszy raz i nikt wczesniej nie czyscil tego folderu. Kolejnym krokiem jest zczytanie przeslanych plikow oraz przypisanie ich do zmiennych w celu latwiejszego poslugiwania sie nimi w kolejnych etapach. Jezeli uzytkownik podal wartosc opcjonalna wspomniana w poprzednim rozdziale, dotyczaca opoznienia pozysku informacji, ten parameter jest rowniez pobierany. Nastepnie program otwiera plik arkusza kalkulacyjnego i sprawdza poprawnosc formatu danych. W tym procesie odczytywane zostaja naglowki kolumn, a nastepnie sprawdzane czy w odpowiednich kolumnach znajduja sie odpowiednie dane. Gdy ten warunek jest spelniony, przy uzyciu petli “for” odczytywany jest kolejno kazdy rzad, a pozyskane informacje zostaja przypisane pod zmienne, z ktorych program bedzie korzystal podczas wyszukiwania fraz w internecie. Zostaje uruchomiona zautomatyzowana przegladarka “Google Chrome” w trybie ukrycia oraz caly proces wyszykiwania danych jest przeprowadzany jako proces w tle. Gdy program uzyska poszukiwana informacje, zostaja one wprowadzona do arkusza kalkulacyjnego jako dane sugestyjne. Po zgromadzeniu przez program wszystkich mozliwych do uzyskania informacji, uzytkownik zastaje przeniesiony do strony wyswietlajacej podsumownie. W momencie wykrycia przez program bledu, strona wyswietlajaca podsumowanie bedzie zawierac informacje dla uzytkownika, gdzie wystapil blad. Jezeli proces zostanie wykonany poprawnie, na stronie koncowej uzytkownikowi zostaje wyswietlony guzik inicjujacy pobieranie przygotowanego przez program arkusza kalkulacyjnego o nowej nazwie “S\_DataValues.xlsx”.

## Tools

### IntelliJ IDEA

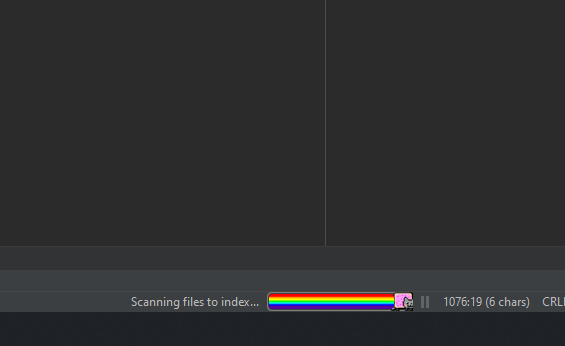
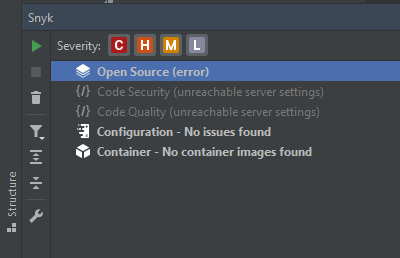
W celu tworzenia zaawansowanych lub tych prostych, rekreacyjnych programow informatycznych do ktorych programista lub organizacja dla ktorej programista pisze program, potrzebny jest edytor kodu. Swietnym edytorem jest IntelliJ IDEA, komercyjne zintegrowane srodowisko programistyczne mogace kompilowac zapisany skrypt. Edytor zostal napisany w jezyku Java przez firme “JetBrains”. Pierwsza publikacja pojawila sie na rynku w pierwszym kwartale roku 2001. Ta wersja dysponowala zbiorem narzedzi wspomagajacych refactoring. Aktualnie oferuje on wiele pomocnych wskazowek oraz skrotow klawiszowych wspomagajacych programiste w jego pracy. Obsluguje 19 jezykow programistycznych i automatyzyjnych takich jak “Java”, “Python”, “Scala” czy jezyki przeznaczone do witryn internetowych jak “HTML”, “CSS”. IntelliJ IDEA zapewnia rowzniez itegralnosc z wieloma innymi srodowiskami lub narzedziami deweloperskimi typu Open Source takimi jak “GIT”, “Apache Maven”, “Apache Ant”, “JUnit”, “CVS” czy “SVN”. Edytor dysponuje tez interesujaca dla wielu fanatycznych informatykow, ciemny motyw ukladu interface.

**Picture 3 Left side, IntelliJ dark theme Picture 4 Right side, IntelliJ default light theme**

Dostepnych jest wiele wtyczek rozszerzajacych funkcjonalnosc surowej wersji IntelliJ IDEA takich jak “Rainbow Brackets” autorstwa “izhangzhihao” poprawiajaca czytelnosc kodu przez kaskadowe kolory nakladanych na siebie nawiasow zaokraglonych, kwadratowych i klamerkowych, lub wtyczka “Snyk Security – Code, Open Source, Container, IaC Configurations”, ktora analizuje napisany przez programiste ko di zwraca raport o bezpieczenstwie stworzonej aplikacji. Raporty sa przedstawiane w sposob bardzo czytelny i nie stanowia problemu w ich zrozumieniu poczatkujacym programistom. “Snyk” przedstawia rowniez potencjalnie mozliwe rozwiazania problemow zwiazanych z lukami bezpieczenstwa. Istnieja rowiez wtyczki rekreacyjne, stworzone do wywolywania usmiechu na twarzy uzytkownika edytora kodu IntelliJ IDEA, jedna z takich wtyczek jest “Nyan Progress Bar” autorstwa “Dimitry Batkovich”, ktora zmienia dotychczasowy pasek ladowania kompilatora w teczowy ogon kultowej w srodowisku programistycznym postaci jaka jest “Nyan Cat”.

(Discover Intellij Idea, 2023)

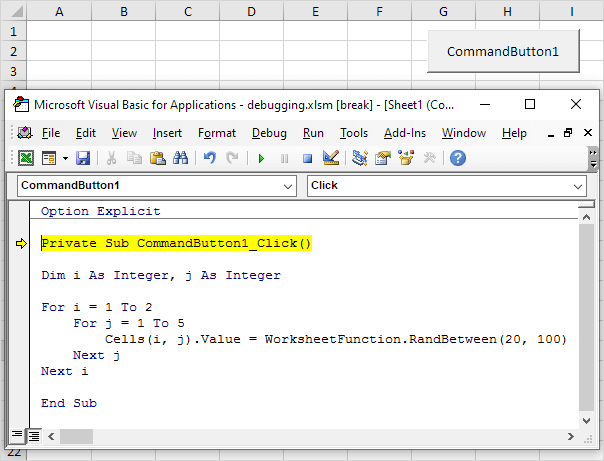
 

**Picture 5 Left side, Nyan Progress Bar Picture 6 Right side, Snyk add-in**

### Excel

Microsoft Excel to arkusz kalkulacyjny stworzony oraz w dalszym ciagu rozwijany do uzytku na systemach operacyjncyh Windows, macOS, Android oraz iOS. Za jego pomoca uzytkownik jest w stanie obliczyc w prosty sposob rozne dzialania matematyczne, przedstawic je na wykresie lub w tabeli przestawnej. Nawet mozna samemu napisac program automatyzujacy process wmagany przez uzytkownika zwanym Macro. Jezykiem niezbednym do zapisania takiego Macro jest Visual Basic For Applications (VBA). Przy kazdej nowo wydanej wersji tego programu otrzymujemy wiele nowosci oraz udoskonalen jak i ulatwien dla uzytkownika. Duzo dodatkowych narzedzi ktore nie sa przewidziane przez oficjalne wydanie developerskie mozna znalezc w postaci add-ins, glownie stanowia one opragromowanie stron trzecich, zapisanych najczesciej przez prywatnych programistow sa to wtyczki rozszerzajace funkcjonowanie programu Microsoft Excel

(Microsoft Excel, 2023)

**Picture 7 Left side, Excel Worksheet Picture 8 Right side, Visual Basic for Applications**

### Python

Jezyk programistyczny posiadajacy wiele frameworkow dedykowanych do konkretnych zadan, aplikacji lub programow oraz z bardzo rozbudowana lista bibliotek standardowych to szeroki wachlarz mozliwosci dla programisty. Idea przewodnia tego jezyka jest klarownosc i czytelnosc kodu zrodlowego. Glowna cecha skladni jest przejrzystosc i zwiezlosc. Python wspiera rozne typy programowania takie jak imperatywny, obiektowy i w znaczaco mniejszym stopniu od poprzednich dwoch, funkcyjny. Charakterystycznym punktem jest dynamiczny system typow oraz zautomatyzowane zarzadzanie pamiecia, dlatego nie spotkamy sie w nim z deklaracjami zmiennych, na przyklad integer, double, String. Python jest bardzo popularnym jezykiem programowania przez co interpretery tego jezyka sa dostepne na wiele systemow operacyjnych. Rozwoj jest przeprowadzany jako Open Source, ktorego zarzadca jest “Python Software Foundation”, wzorowanej na “Apache Software Foundation”. Jest to organizacja non-profit. Implementacja jezyka bedaca rowniez standardowa, jest napisany w jezyku “C”, “CPython”. Jest ona standardowa poniewaz istnieja rowniez inne. Implementacja zapisana w jezyku “Java” jest JPython, a zapisany w “Common Lisp” oraz “IronPython” to “CLPython”, z przeznaczeniem na platforme “.NET”. Poczatki Pythona mialy miejsce we wczesnych latach 90, jego tworca byl Guido van Rossum. Jezyk ten mial byc nastepca jezyka “ABC”, opracowywano go w Centrum Matematyki i Informatyki w Amsterdamie (CWI - Centrum voor Wiskunde en Informatica).

**Picture 9 Left Side, CWI building Picture 10 Right side, Guido van Rossum**

Ciekawostka jest to, ze projektant nie nawiazywal do zwierzecia gdy przystapil to wybierania nazwy jezyka programistycznego. Byl wielkim fanem serialu komediowego, popularnego w latach 70 emitowanego przez telewizje “BBC” pod tytulem “Monty Python’s Flying Circus”. Opierajac sie o informacje pozyskane w internecie, w roku 2020 odnotowano 25 frameworkow dostepych do uzytku publicznego, dedykowanych dla jezyka Python, zapisanych w tym jezyku jak i tych zapisanych w jego oparciu. Aplikacja “GeoFinderUI” zostala zapisana przy uzyciu frameworku “Flask”. Flask to micro web framework, posiada ten tylul poniewaz nie wymaga on poszczegolnych narzedzi czy bibliotek. Jest to swietny wybor przy projektowaniu oraz programowaniu aplikacji webowej.

(Python, 2023)

### Selenium

Selenium to projekt typu open source, najczesciej spotykany pod postacia biblioteki dostepnej dla wielu jezykow programistycznych. Zadaniem tego pakietu jest umozliwienie w prosty sposob automatyzowac dostep do przegladarki. Mozliwym wykorzystaniem funkcjonalnosci tego poteznego narzedzia jest na przyklad wyszukiwanie konkretnej frazy wpisanej automatycznie przez program, a nastepnie przy uzyciu technologii Web Scrapingu, odnalezienie interesujacej uzytkownika informacji zapisanej w kodzie zrodlowym strony internetowej zapisanej w jezyku “HTML” Aby to bylo mozliwe to oprocz poprawnie zapisanej procedury automatyzacyjnej w skrypcie jednego z jezykow wliczajac “C#”, “JavaScript (Node JS)”, “Groovy”, “Ruby”, “Java”, “Perl”, “PHP”, “Scala” i “Python”, potrzebny jest sterownik przegladarki. Na oficjalnej stronie internetowej Selenium dostepnych jest 5 sterownikow do najpopularniejszych przegladarek, “Chrome”, “Firefox”, “Edge”, “Internet Explorer” i “Safari”. Pakiet jest dostepny na 3 systemy operacyjne “Windows”, “Linux”, “macOS”. Autorem Selenium jest Jason Huggins. W roku 2004 stworzyl on narzedzie dla wewnetrznego uzytku w firmie “ThoughtWorks”. Projekt zostal bardzo szybko unzany za wielkie osiagniecie oraz otrzymal wielki rozglos. W roku 2005 Dan Fabulich i Nelson Sproul zaproponowali wydanie serii patchow aby przeksztalcic znany jako “Selenium Remote Control (RC)” w narzedzie Selenium bardzo dobrze znane dzisiaj w srodowisku informatycznym wykorzystujacym glownie technologie Web Scrapingu. Nazwa Selenium pochodzi od zartu zawartego w jedym z maili wyslanych przez Hugginsa. Zart mial na celu przedrzezniac konkurencje ktora byla firma Mercury Interactive Corporation. Mercury to Izraelska firma wykupiona przez swiatowego giganta z dziedziny technologii, HP Software Division. Zartobliwy mail posiadal w swojej tresci zdanie “you can cure mercury poisoning by taking selenium supplements”.

(Selenium, 2022)



**Picture 11 Jason Huggins**

### Pandas

Pandas to biblioteka napisana do uzytku w jezyku programistycznym Python. Oferuje wiele mozliwosci analizy oraz manipulacji danych w glownej mierze danych numerycznych. To opragromowanie jest dystrybuowane w formie darmowego planu subskybcyjnego posiadajacego licencje “BSD” Pierwsza publikacja miala miejsce w roku 2008 przez Wes McKinney, ktory pracowal w tym czasie dla firmy “AQR Capital Management”. Pierwotym zalozeniem konstruowania tego narzedzia byla potrzeba wysokowydajnego sposobu analizy danych finansowych. Nazwa tej biblioteki pochodzi od dwuczlonowego zwrotu „panel data”, gdzie pierwsze litery tworza slowo “panda”. “Panel data” to termin ekonometryczny dla zbiorow danych posiadajacych obserwacje na przestrzeni wielu przedzialow czasu.

(Pandas software, 2023)



**Picture 12 Wes McKinney**

### Openpyxl

W momencie gdy program lub aplikacja, ktora projektuje programista wymaga uzycia arkuszy kalkulacyjnych “Excel” z pakietu “Microsoft Office” to najlepszym rozwiazaniem do zautomatyzowanego poslugiwania sie tym programem jest biblioteka “Openpyxl”. “Openpyxl”to dobrze rozwinieta biblioteka zawierajaca wiele przydatnych narzedzi sterujacych arkuszem kalkulacyjnym. Glownym powodem stworzenia tego pakietu byl brak istnienia podobnych rozwiazan mogacych w prosty, przejrzysty i logiczny sposob otwierac oraz edytowac pliki zapisane w formatach z rozszerzeniami “xls”, “xlsx”, “xlsm”, “xltx i “xltm”. Niestety w swiecie technologii komputerowej istnieje wiele zagrozen w postaci wirusow lub oprogramowan szpiegowskich, ktore czesto mozna spotkac bardzo dobrze ukryte w arkuszach kalkulacyjnych Excel pochodzacych od nieznanego zrodla. Jak podaje oficjalna strona producenta, biblioteka “Openpyxl” nie oferuje w tym zakresie jakiegokolwiek wsparcia. Jedyna mozliwa opcja sugerowana przez wlascicieli opragromowania jest instalacja dodatkowego pakietu jakim jest “defusedxml”. Autorami opragromowania “Openpyxl” sa Eric Gazoni i Charlie Clark. Pierwsza wersja, dostepna do uzytku publicznego byla wersja 1.7.0 wydana pod koniec roku 2013.

(Openpyxl, 2022)

**Picture 13 Left side, Eric Gazoni Picture 14 Right side, Charlie Clark**

### HTML

Jezyk “HyperTextMarkupLanguage (HTML)” nalezy do kategorii markup languages. Dominujaca odmiana formatowania tekstu wystepujaca w jezyku “HTML” to descriptive markup. Wyrozniaja te odmiane oznaczenia fragmentow tekstowych lub przekazywanych danych, ktore definuja role pelnienia wyodrebnionej informacji zapisanej w dokumencie. “HTML” zostal stworzony do tworzenia stron internetowych “World Wide Web (WWW)”, pozwala on na opis struktury informacji wyswietlanych wewnatrz strony internetowej. Glownymi dostepnymi narzedziami jest formulowanie hiperlacz, list, tabel, naglowkow, akapitow oraz mozliwosc osadzania w tekscie dokumentu obiektow multimedialnych lub plikowych. Istnieja rowniez elementy nieposiadajace baz danych, ktorymi moga byc naprzyklad interaktywne formularze danych. Poczatkiem jezyka “HTML” mozna oznaczyc rok 1980, w ktorym fizyk Tim Berners-Lee stworzyl prototyp hipertekstowego systemu informacyjnego o nazwie “ENQUIRE”. Celem tego systemu byla organizacja oraz udostepnianie dokumentow zwiazanych z badaniami naukowymi. Uzytkownicy mogli miec wtedy dostep do udostepnionej dokumentacji znajdujacej sie w zupelnie innej, odleglej lokalizacji. W roku 1989 Tim Berners-Lee konkurowal z Robert Cailliau. Obiektem ich konkurencji byla rownolegla propozycja hipertekstowych systemow informacyjnych opartych na sieci Internet. Byl to poczatek przygody prowadzacej do opracowania niesamowitej technologii, ktora dzisiejsi internauci znaja bardzo dobrze, poniewaz w roku 1990 Tim oraz Robert polaczyli swoje sily i przedstawili wspolnie przygotowana propozycje projektu “WorldWideWeb”. Osiagneli sukces gdyz ich propozycja zostala zaakceptowana przez CERN. Pierwsza dostepna publicznie dokumentacja jezyka “HTML” noszaca wtedy nazwe “HTML Tags” zostala wyemitowana w roku 1991. Pierwotna struktura tego jezyka zawierala w sobie 22 znaczniki przystosowane do oznaczania elementow znajdujacych sie w dokumencie. “HTML”, chociaz zostal napisany w oparciu o jezyk “SGML”, nie posiadal w sobie formalnej definicji pierwowzorowego jezyka znacznikowego sluzacego do latwego zapisu dokumentow tesktowych z mozliwoscia prostego przenoszenia, wyswietlnia oraz drukowania w roznych systemach elektronicznego przekazu danych.

(HTML, 2023)

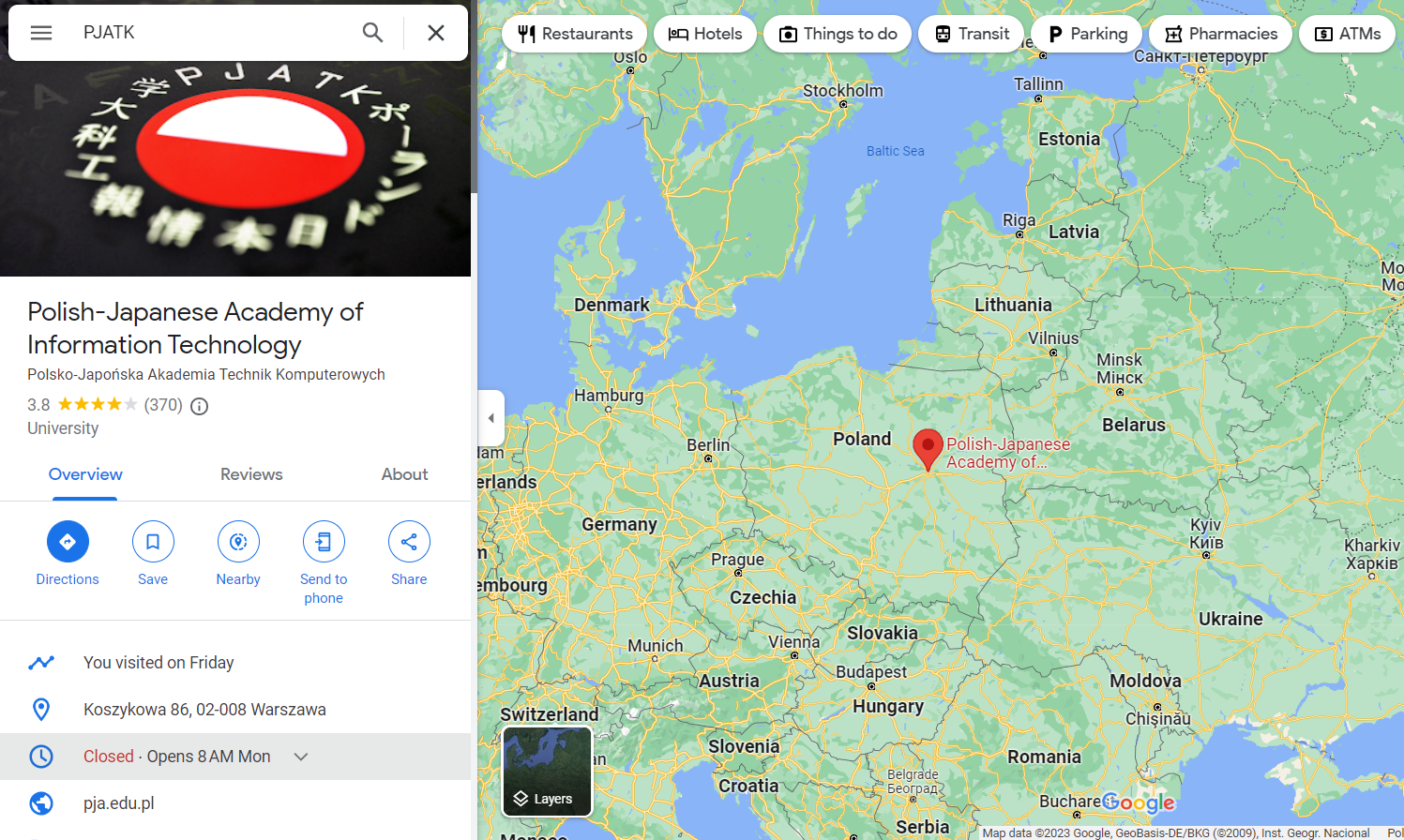
 

**Picture 15 Left side, Timothy Berners-Lee Picture 16 Right side, Robert Cailliau**

### Google Maps

Strona internetowa, niezbedna do przeprowadzania glowniej funkcjonalnosci aplikacji “GeoFinderUI” to “Google Maps”. Firma “Google” ktora jest wlascicielem omawianej strony internetowej posiada ogromne zosoby danych przechowywane w bazach danych, ktorych nowe pozycje sa dodawane w kazdej sekundzie. Aplikacja “Google Maps” wykorzystuje te dane, ktore posiadaja w sobie informacje o lokalizacji znajdujacej sie na kuli ziemskiej, a nastepnie przedstawia je w formie graficznej w formie zaznaczonych punktow na mapie swiata. W celu ulatwienia uzytkownikowi odnalezienia interesujacej go lokalizacji, aplikacja “Google Maps” zostala wyposazona w wyszukiwarke. Po wpisaniu frazy orazj odszukaniu jej, uzytkownikowi zostaje wyswietlona lista najlepiej pasujacych wynikow wyszukiwania. Niestety takie wyszukiwanie nie zawsze jest zgodne z intencjami uzytkownika. Przykladowo, baza danych na ktorej operuje aplikacja, jest na tyle duza, ze istnieje bardzo duze prawdopodobienstwo odnalezienia dwoch lub nawet kilku lokalizacji posiadajacych te sama nazwe. W celu identyfikacji poprawnosci wynikow wyszukiwania, niezbedny jest czlowiek lub program posiadajacy zdolnosci analityczne. Takie scenariusze nie wystepuja kazdorazowo lecz z kazdym dniem pojawia sie wiele nowych lokali uslugowych, firm, parkow zieleni, ulic, obiektow rozrywkowych lub turystycznych, ktorych wlasciciele chca aby ich lokalizacja byla dostepna w bazach danych firmy “Google”. To jest glowny problem, ktory uzytkownik moze napotkac w trakcie korzystania z aplikacji. Na przeciw temu zjawisku, “Google Maps” jest w ciaglym rozwoju oraz udoskonalaniu aby uzytkownik mogl otrzymac najlepszej jakosci wynik swoich wyszukiwan. Rozwoj natomiast nie ogranicza sie tylko i wylacznie do wyszykiwania wprowadzonych przez uzytkownika fraz oraz dostarczeniu wynikow, deweloperzy z firmy “Google” nieustannie tworza nowe funkcjonalnosci, ktore sa udostepniane do uzytku publicznego. Od pierwszej wersji tej aplikacji, dostepnej publicznie, wydanej w roku 2005 zmienilo sie wiele, przykladowo, zostala dodana bardzo uzyteczna funkcja “Street View”, pozwalajaca uzytkownikowi pojawic sie w centrum wyznaczonej, dostepnej dla tej funkcji, lokalizacji oraz przegladac to miejsce w formie wirtualnej rzeczywistosci stworzonej przy pomocy zlozonych ze soba zdjec z kamery mogacej robic tak zwane “zdjecia 360”. “Zdjecia 360” to polaczone ze soba projekcje pozyskane z conajmniej dwoch obiektywow kamery, co w efekcie skutkuje w rozszerzonym kacie widzenia do 360°. Takie zdjecia zazwyczaj sa zapisywane w specjalnych formatach, ktore umozliwiaja odczytanie oraz wyswietlenie pliku w dedykowanych dla tej technologii programach komputerowych oraz daniu uzytkownikowi mozliwosci obserwowania wybranego kadru, a nie kadru narzuconego przez autora fotografii.

(Google Maps, 2022)

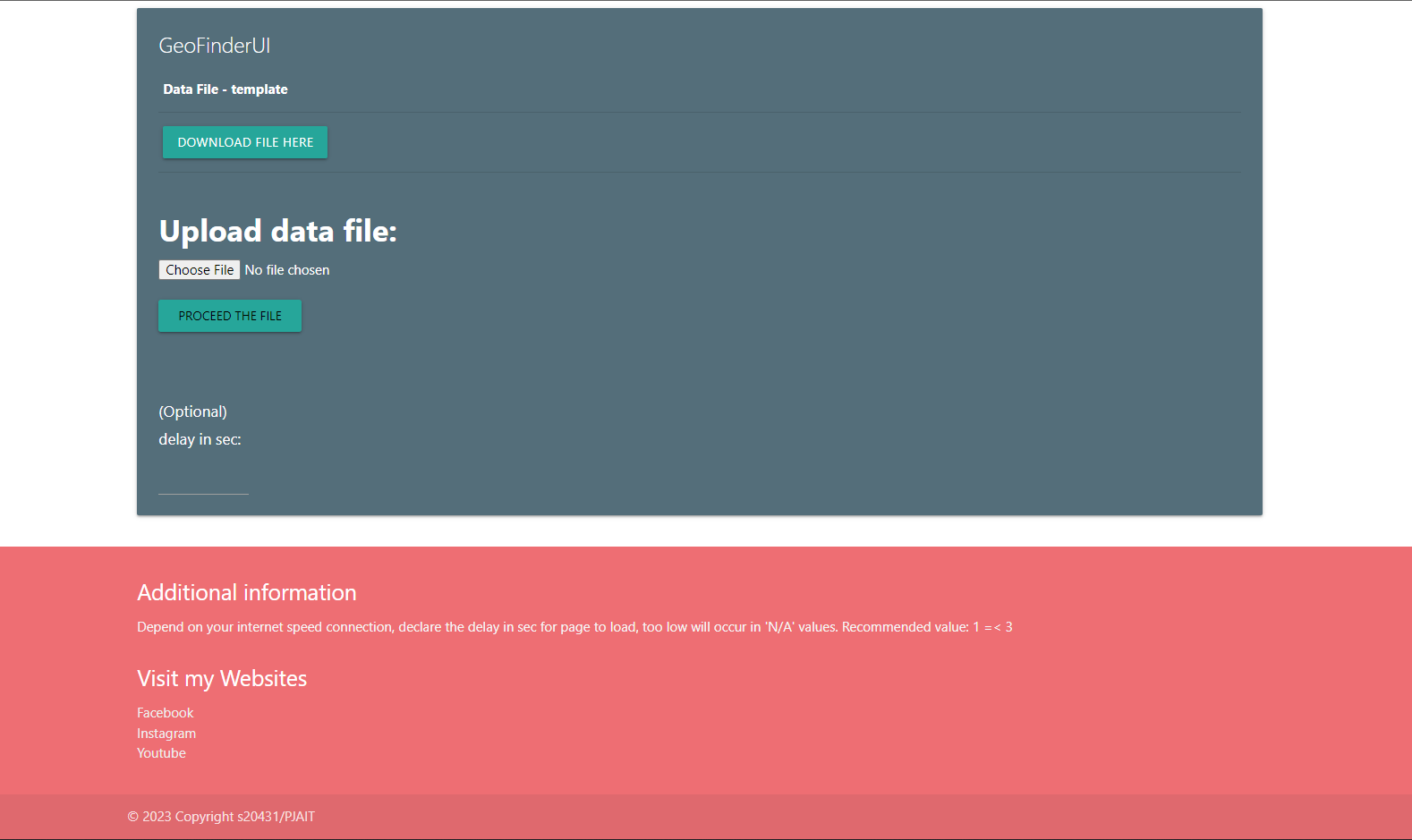


**Figure 4 Google Maps search result**

### Materialize

Aby lepiej zaprezentowac swoja strone oraz zaoszczedzic przy tym ogromna ilosc czasu warto skorzystac z gotowych pakietow stylistycznych “CSS” i “JavaScript” przygotowanych przez portal “materializecss.com”. “Materialize” to strona internetowa ktora oferuje wiele kategorii pakietow stylistycznych do stron “HTML”. Na stronie startowej uzytkownik zostaje poinformowany o krokach wymaganych przed skorzystaniem z korzysci plynacych z budowania wlasnej strony internetowej oraz nadawaniu jej wizualnego wydzwieku przez specjalnie przygotowane znaczniki portalu “Materialize”. Kazdy krok jest przedstawiony w sposob dokladny i logiczny aby kazdy byl w stanie zrozumiec i korzystac z tej strony internetowej. Aby pakiety stylistyczne byly dostepne do uzycia w dokumencie “HTML” nalezy dodac w sekcji “<head> </head>”dwie linijki tekstu zawierajace linki do strony gdzie pakiety stylistyczne sa umieszczone. Pierwszy z nich to link do pakietow “CSS”, a drugi to link do pakietow “JavaScript”. Z lewej strony portalu “Materialize” uzytkownik moze wyszukac interesujacy go pakiet sposrod wielu kategorii. W kazdej kategorii znajduje sie wiele przykladow z roznym wygladem pakietu stylistycznego dla konkretnego obiektu. Po znalezieniu interesujacego pakietu, nalezy skopiowac kod “HTML” widoczny dla uzytkownika pod przykladem wygladu stylizacji obieku “HTML”, nastepnie umiescic go w swoim dokumencie odpowiedzialnym za uklad strony internetowej oraz znajdujacych sie tam tresci, zapisanej w “Hyper Text Markup Language”.

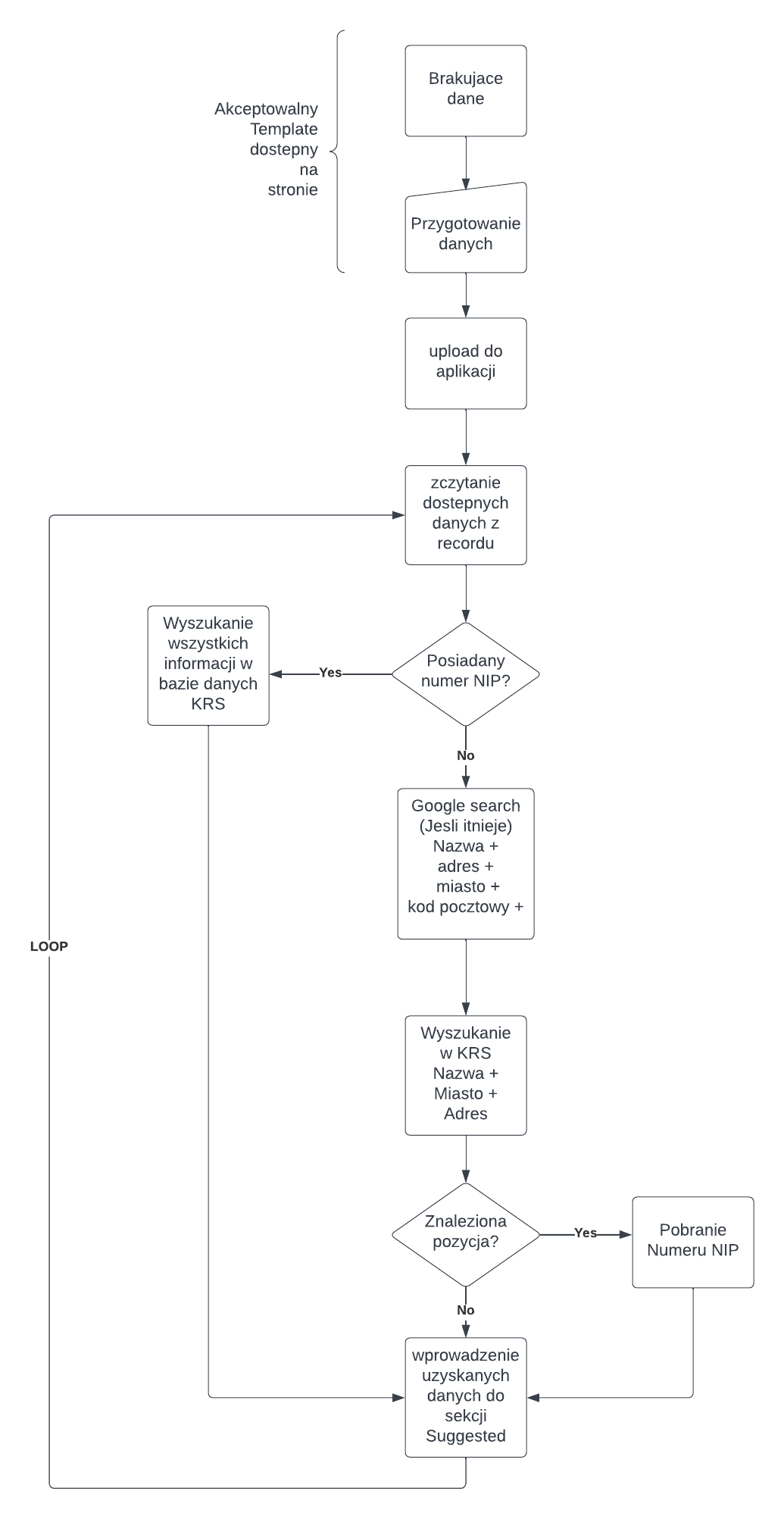
(Materialize, 2023)



**Figure 5 GeoFinderUI application home page with Materialized styling packages**

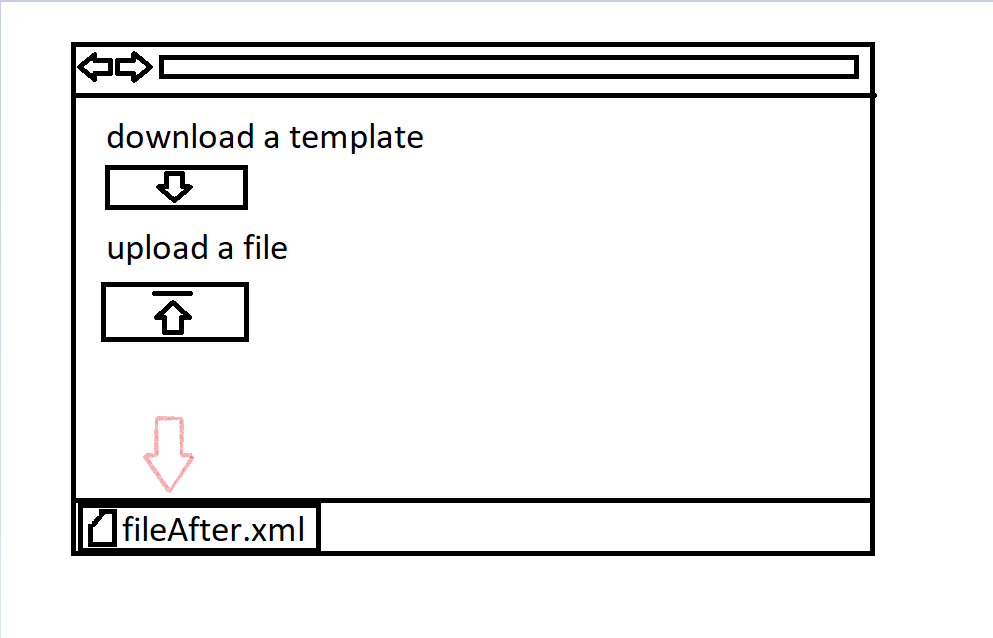
# Functionality process assumptions

Pierwszym etapem tej pracy bylo graficzne sformulowanie procesu aplikacji webowej. Proces ten w glownej mierze opieral sie na wprowadzonym numerze “NIP” lub jego odszukaniu dla konkretnej lokalizaji. Takie informacje sa ogolnodostepne na stronie Krajowego Rejestru Sądowego “(KRS)”. Niestety te zalozenia uwzglednialy rowniez wykorzystanie technologii web scraping. Strona “KRS” bez wyjatku blokuje dostep do danych znajdujacych sie na niej po wykryciu automatyzacyjnego przemieszczania sie po stronie przy uzyciu bota. Technologia web scraping to tworzenie takiego bota potrafiacego odczytywac oraz przeszukiwac kod zrodlowy strony internetowej.

****

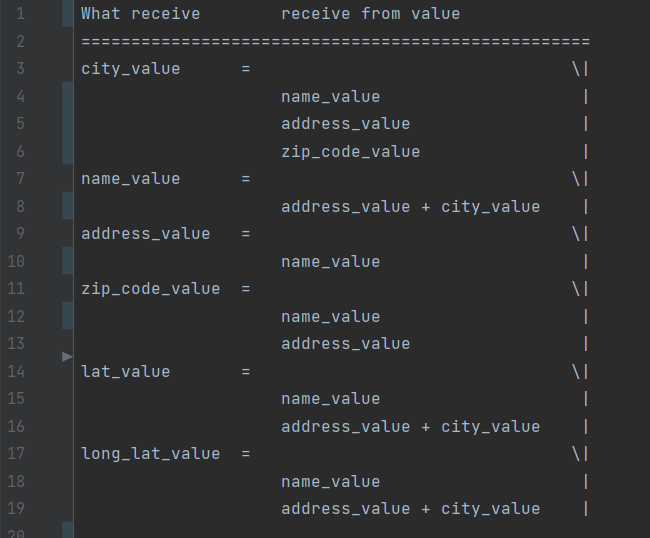
**Figure 6 the original diagrammatic process of the operation of the GeofinderUI application**

Taka przeszkoda wymagala restrukturyzacji funkcjonowania aplikacji. Od tego momentu glownym zrodlem pozyskiwania informacji zostal portal posiadajacy dostep do najwiekszej bazy danych geolokalizacyjnych, ktorym jest “Google Maps”. Przed rozpoczeciem pracy nad programowaniem aplikacji “GeoFinderUI” potrzebne bylo znalezniezie odpowiednich narzedzi, ktore pomoga w uzyskaniu pelniej funkcjonalnosci programu oraz dostarcza najlepszej jakosci wynik koncowy uzytkownikowi. Pierwszymi zalozeniami bylo przechowywanie oraz dostarczanie danych do palikacji w zrozumialy i prosty sposob dla uzytkownika. W tym celu wybor padl na arkusz kalkulacyjny “Excel”. Przez podjeta decyzje o korzystaniu z plikow “.xls” program ktory mial byc zapisany w jezyku programistycznym “Python” wymagal odpowiednich bibliotek bedacych w stanie operowac arkuszami kalkulacyjnymi. Biblioteka “Pandas” oraz biblioteka “Openpyxl” swietnie sie nadaja do takich zadan. Za ich pomoca “GeoFinderUI” odczytuje oraz zapisuje znalezione dane z internetu w arkuszach kalkulacyjnych. Kolejnym etapem bylo dobranie odpowiedniego narzedzia operujacego stronami internetowymi. Narzedzie powinno umozliwic wprowadzanie fraz do okna wyszukiwania w przegladarce lub stronie internetowej, a nastepnie sposrod wynikow wyszukiwania, odnalezc obiekt w kodzie zrodlowym strony internetowej, zawierajacy poszukiwana informacje. Bibioteka “Selenium” spelnia powyzsze wymogi oraz stanowi najwazniejsza czesc funkcjonalnosci programu “GeoFinderUI”. Aplikacja webowa to nie tylko sktypt back-end, w tym celu potrzebna byla wizualizacja ukladu strony. Prosty lecz tresciwy dokument, nakreslajacy cel do ktorego aplikacja swoim wygladem miala dazyc od poczatku zostal przygotowany w programie “Paint” dostepnym w systemie operacyjnym “Windows”. Ten program jest prostym w obsludze oraz mocno ograniczonym w swoich mozliwosciach edytorem graficznym. Byl natomiast wystarczajacy do stworzenia przykladowej wizualizacji wygladu strony.



**Figure 7 sample visualization of the GeoFinderUI start page**

Nastepnym niezbednym krokiem bylo okreslenie mozliwosci otrzymywania danych. Celem tego bylo maksymalne wykorzystanie odstarczonych przez uzytkownika danych aby otrzymac najwiecej dostepnych informacji o danej lokalizacji w wynikach koncowych przygotowanych przez aplikacje “GeoFinderUI”.



**Figure 8 data acquisition points by GeoFinderUI application**

Po zebraniu tak obszernych informacji oraz dokumentow mozna przystapic do programowania aplikacji.

# Implementation.

## Front-end code

Kod front-end to czesc kodu odpowiedzialna za wszystkie elementy widoczne na stronie lub ja modelujace.

### Home page

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>GeoFinderUI</title>  
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>

<style>  
 html {  
 height: 100%;  
 }  
 body {  
 min-height: 100%;  
 display: flex;  
 flex-direction: column;  
 }  
 .content {  
 flex: 1;  
 }  
 footer {  
 margin-top: auto;  
 display: block;  
 bottom: 0;  
 width:100%;  
 height: auto;  
 }  
</style>

</head>

**Code 1 The <Head> section with the style part needed for positioning the content of the GeoFinderUI start page**

Cala widoczna dla uzytkownika czesc aplikacji zostala zapisana w jezyku “HTML”. Duza role w jej wygladzie odgrywaja pakiety stylizacyjne dostarczone przez portal “Materialize”. Przy ich pomocy strona nabrala profesjonalny oraz przyjazny dla uzytkownika wyglad. Byloby to niemozliwe bez dodania 2 linijek kodu stanowiacych odnosniki do lokalizacji zapisanych pakietow stylistycznych “CSS” oraz “JS”. Odnosniki zostaly umieszczone w gornej czesci kodu zrodlowego strony internetowej w sekcji “<head> </head>”. W tej sekcji mozna tez dostrzec podsekcje “<style> </style>”. Tak przygotowana sekcja stylistyczna pozwala rozmiescic elementy znajdujace sie na stronie internetowej w sposob uniwersalny w odniesieniu do rozmiaru okna w ktorym strona jest wyswietlana. W momencie gdy rozmiar ekranu wyswietlajacego strone jest na tyle maly, ze elementy strony nachodza na siebie, zostaje wyswietlony scroll bar po prawej stronie okna, umozliwiajacy odczytanie kazdej znajdujacej sie na stronie informacji. Podsekcja “<footer> </footer>” otrzymala atrybuty stylistyczne, wymuszajace wyswietlanie tej podsekcji w najnizszym mozliwym polozeniu na stronie, lecz nie dalej niz wymaga tego przestrzen przeznaczona na prawidlowe wyswietlenie wszystkich elementow strony internetowej. Taki sam zestaw ustawien znajduje sie na kazdej z istniejacych podstron aplikacji “GeoFinderUI”.

<table class="table">  
 <thead>  
 <tr>  
 <th>Data File - template</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr>  
 <td>  
 <a href="/download\_template"><button class="waves-effect waves-light btn black-text">Download file here</button></a>  
 </td>  
 </tr>  
 </tbody>  
</table>

**Code 2 Part of the code that allows the user to download the official version of the data template acceptable by the application**

W celu najlepszego swiadczenia uslug aplikacji, zostal umieszczony na stronie odnosnik pozwalajacy pobrac oficjalny plik template obowiazujacy podczs dostarczania danych do aplikacji przez uzytkownika. Odnosnik inicjuje uruchomienie dedykowanego skryptu do pobierania plikow znajdujacych sie na stronie, zapisanego w jezyku programistycznym “Python”.

<form enctype="multipart/form-data" action="/upload\_files" method="post">  
 <div>  
 <h4> <b>Upload data file: </b></h4>  
 <p><input type="file" name="excelfile" accept=".xlsx, .xls"/></p>  
 </div><br>  
 <div>  
 <input type="submit" id="proceed-xlsx-file" class="btn waves-effect waves-light" value="Proceed the file">  
 </div><br>  
 <div>  
 <h6>(Optional)</h6>  
 <h6>delay in sec:</h6>  
 <p><input class="col s1" type="number" step="0.5" min="1" name="delay"/></p>  
 </div>  
</form>

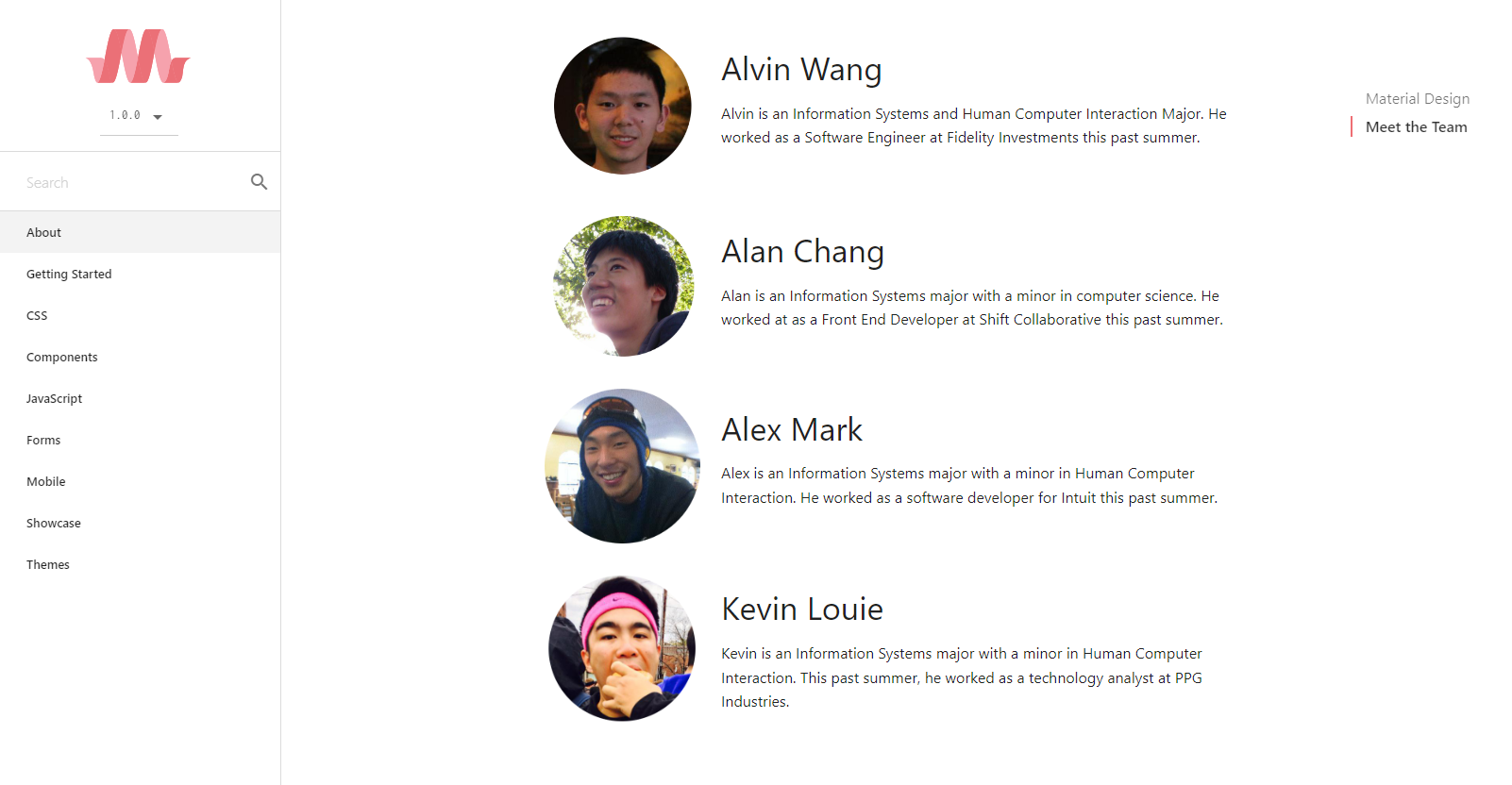
**Code 3 The form section needed for receiving and sending files and information to the back-end code**

Aby w latwy sposob odbierac i przetwarzac pliki lub opcjonalnie dostarczone informacje o dodatkowym czasie potrzebnym do zaladownia strony, a nastepnie przesylac je do kodu back-end, aplikacja wykorzystuje sekcje “<form> </form>”. Za pomoca tej sekcji, strony internetowe bez problemu radza sobie z dostarczaniem niezbednych informacji wykorzystywanych w jezykach programistycznych jako zmienne. Sekcja “form”zawiera w sobie 3 pola “<input>” z czego pierwszy sluzy do przesylania wybranego pliku przez uzytkownika, drugi zatwierdza wybory oraz uruchamia glowna funkcjonalnosc aplikacji, a ostatni jest niewymaganym polem informujacym program o opoznieniu pozyskiwania danych w celu wczesniejszego zaladowania pelnej strony.

<footer class="page-footer">  
 <div class="container">  
 <div class="row">  
 <div class="col s12 m12">  
 <h5 class="white-text">Additional information</h5>  
 <p class="grey-text text-lighten-4">Depend on your internet speed connection, &#013;declare the delay in sec for page to load, &#013;too low will occur in 'N/A' values. Recommended value: 1 =< 3</p>  
 </div>  
 <div class="col s12">  
 <h5 class="white-text">Visit my Websites</h5>  
 <ul>  
 <li><a class="grey-text text-lighten-3" href="https://www.facebook.com/m3mo.wujec/">Facebook</a></li>  
 <li><a class="grey-text text-lighten-3" href="https://www.instagram.com/m3mo\_w/">Instagram</a></li>  
 <li><a class="grey-text text-lighten-3" href="https://www.youtube.com/@m3mo\_w">Youtube</a></li>  
 </ul>  
 </div>  
 </div>  
 </div>  
 <div class="footer-copyright">  
 <div class="container">  
 © 2023 Copyright s20431/PJAIT  
 </div>  
 </div>  
 </footer>  
</body>  
</html>

**Code 4 The part of the code responsible for the footer visible on each page of the "GeoFinderUI" web application**

Kazda strona wyswietlana w ramach aplikacji “GeoFinderUI” posiada w sobie czesc kodu odpowiedzialna za generowanie stopki strony. Stopka znajdujaca sie na stronie glownej posiada dodatkowa informacje ktorej nie znajdziemy na zadnej innej stronie. Informacja ta ma na celu wyjasnienie uzytkownikowi do czego sluzy pole “delay in sec:”. Wiele stron dostepnych w internecie posiada informacje o autorze danej strony lub aplikacji webowej. Aplikacja “GeoFinderUI” takze posiada informacje o autorze w formie linkow zewnetrznych do portali spolecznosciowych autora w ktorych publikowane sa informacje o hobby, zainteresowaniach a takze informacje o doswiadczeniu zawodowym autora.



**Figure 9 Information about the authors of materializecss.com**

### Summary page

<div class="container">  
 <div class="row">  
 <div class="col s12 m12">  
 <div class="card blue-grey darken-1">  
 <div class="card-content white-text">  
 {% if message %}  
 <span class="card-title">{{ message }}</span>  
 {% endif %}  
 <p>As GeoFinderUI developers, we are extremely happy that you are using our tool.</p>  
 </div>  
 <div class="card-action">  
 {% if check %}  
 <a href="/download\_file"><button class="btn btn-default">Download file here</button></a>  
 {% endif %}  
 <a href="/home"><button class="btn btn-default">return to home page</button></a>  
 </div>  
 </div>  
 </div>  
 </div>  
</div>

**Code 5 The part of the code that contains the information and its layout on the summary page**

Strona poswiecona podsumowaniu dokonanej procedury przez aplikacje “GeoFinderUI”, posiada w sobie elementy ktore moga wyswietlac inne informacje w zaleznosci od wyniku przeprowadzonej procedury. Wartosci zmiennych sa przekazywane do wyswietlenia za posrednictwem skryptu back-end jako parametry dodane do funkcji “render\_template()” z pakietu narzedzi stanowiacych biblioteke “flask”. Strona potrafi zmienic swoj uklad w momencie pozytywnego wyniku generowania pliku z danymi lub w momencie wyniku negatywnego. Gdy wynik jest negatywny, zostaje wyswietlona informacja o bedzie oraz nie jest dostepny dla uzytkownika guzik pozwalajacy pobrac plik wygenerowany przez skrypt zapisany w jezyku programistycznym “Python”.

## Back-end code

### Initialization classes

import os  
import time  
  
from flask import Flask, Blueprint, render\_template, *request*, send\_file  
from selenium import webdriver  
from selenium.webdriver.common.keys import Keys  
from selenium.webdriver.common.by import By  
from selenium.common.exceptions import NoSuchElementException  
from selenium.common.exceptions import TimeoutException  
from selenium.webdriver.support.ui import WebDriverWait  
from selenium.webdriver.support import expected\_conditions as *EC*import pandas as *pd*from openpyxl.workbook import Workbook  
from openpyxl import load\_workbook

**Code 6 Part of the code containing the import of the necessary tools used in the "GeoFinderUI" application**

Plik “home.py” rozpoczyna sie od czesci kodu poswieconej importowaniu bibliotek oraz poszczegolnych narzedzi potrzebnych do funkcjonowania aplikacji.

@*home*.route('/')  
def index():  
 return render\_template('home.html')  
  
  
def download(*filepath*):  
 print(*filepath*)  
 return send\_file(*filepath*, as\_attachment=True)  
  
  
@*home*.route('/download\_template')  
def download\_template():  
 *path* = os.path.realpath('files/template\_files/Template.xlsx')  
 return download(*path*)  
  
  
@*home*.route('/download\_file')  
def download\_file():  
 *path* = os.path.realpath('files/download/S\_DataValues.xlsx')  
 return download(*path*)

**Code 7 Side methods in "home.py"**

Pierwsze metody znajdujace sie w pliku “home.py” to metody poboczne, sluzace do inicjalizacji strony startowej po wpisaniu odpowiedniego adresu “URL” oraz metody umozliwiajace pobranie plikow takich jak “Template.xlsx” i plik arkusza kalkulacyjnego posiadajacy wyniki koncowe glownego procesu aplikacji “GeoFinderUI”.

@*home*.route('/upload\_files', methods=["GET", "POST"])  
def upload():  
  
 *driver\_path* = os.path.realpath("webdrivers/chromedriver.exe")  
 *chrome\_options* = webdriver.ChromeOptions()  
 *chrome\_options*.add\_argument("headless")  
 *chrome\_options*.add\_argument("disable-gpu")  
  
 *path* = os.path.realpath('files/upload')  
 *path\_download* = os.path.realpath('files/download')

**Code 8 The beginning of the main method responsible for the functionality of the application**

Poczatek glownej metody odpowiedzialnej za funkcjonalnosc aplikacji to definiowanie zmiennych niezbednych do uzytkownia skryptu. Znajduja sie tam funkcje zwracajace sciezke absolutna danej lokalizacji na podstawie jej fragmentu oraz zmienna inicjalizujaca sterownik do przegladarki “Chrome” od firmy “Google”. Do tej zmiennej zostaly rowniez dodane argumenty umozliwiajce sterowaniem przegladarki w tle aby na ekranie uzytkownika nie pojawial sie zautomatyzowany proces pozysku danych z przegladarki.

if *request*.files:  
 if len(os.listdir(*path*)) != 0:  
 for *filename* in os.listdir(*path*):  
 *filepath* = os.path.join(*path*, *filename*)  
 os.remove(*filepath*)  
 if len(os.listdir(*path\_download*)) != 0:  
 for *filename* in os.listdir(*path\_download*):  
 *filepath* = os.path.join(*path\_download*, *filename*)  
 os.remove(*filepath*)  
  
 *upload\_file* = *request*.files["excelfile"]  
 *upload\_filename* = *upload\_file*.filename  
 *delay* = *request*.form.get("delay")  
  
 if *delay* == "" or *delay* == " ":  
 *delay* = 0.5  
 float(*delay*)

**Code 9 Part of the code responsible for preparing and cleaning the location for storing data files**

Przed rozpoczeciem procesu przeszukiwania stron internetowych w celu zebrania danych tworzacych wynik koncowy, program analizuje przestrzen docelowa do przechowywania plikow z danymi oraz w momencie znalezienia jakiegokolwiek pliku zostaje on usuniety, taki zabieg minimalizuje ryzyko wystapienia bledu, rowniez umiejscowienie takiej procedury na poczatku kodu ma duzy wplyw na zniwelowanie potencjalnego powstania bledu, poniewaz gdyby poprzedni uzytkownik nie doprowadzilby procesu pozyskiwania danych do konca, a fragment kodu czyszczacy lokalizacje znajdowalby sie na koncu procedury, to aktualny uzytkownik moglby otrzymac wyniki na podstawie dostarczonych danych przez swojego poprzednika.

*data\_tab\_value* = pd.read\_excel(*data\_file\_path*, 'Data')  
*reader* = pd.read\_excel(*data\_file\_path*, 'Data', header=None, usecols="A:G")  
  
*wb* = load\_workbook(filename=*data\_file\_path*)  
*ws* = *wb*['Data']

*ws*['F1'] = "S\_Name"  
*ws*['G1'] = "S\_Address"  
*ws*['H1'] = "S\_City"  
*ws*['I1'] = "S\_Zip-code"  
  
*driver\_chrome* = webdriver.Chrome(executable\_path=*driver\_path*, options=*chrome\_options*)  
*driver\_chrome*.get("https://www.google.com/maps/")

**Code 10 Part of the code responsible for preparing the data file and launching the automated browser**

Aby uzytkownik byl dobrze poinformowany o wprowadzanych wartosciach do arkusza kalkulacyjnego, program przypisuje nazwy naglowkow bedace polaczeniem “S” odpowiadajacym slowu “Suggested” oraz typie przekazywanych danych, takich jak address, city lub zip-code.

for *record* in range(len(*data\_tab\_value*.index)):  
 *name\_value* = (*data\_tab\_value*.iat[*record*, 0])  
 *address\_value* = (*data\_tab\_value*.iat[*record*, 1])  
 *city\_value* = (*data\_tab\_value*.iat[*record*, 2])  
 *zip\_code\_value* = (*data\_tab\_value*.iat[*record*, 3])

**Code 11 A loop procedure that goes through all the records entered in the data file**

Cala aplikacja opiera sie na petli “for”, program w tym momencie zbiera dane dostarczone przez uzytkownika, kazda iteracja to kolejny wiersz w arkuszu kalkulacyjnym.

if pd.isna(*name\_value*):  
 if *address\_value* and *city\_value*:  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *city\_value*)  
 if pd.isna(*zip\_code\_value*):  
 if *s\_name* != "N/A":  
 *s\_zip\_code* = zipcode\_f\_name(*driver\_chrome*, *search*, *delay*, *s\_name*)  
 else:  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
  
 elif *address\_value* and *zip\_code*:  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *zip\_code\_value*)  
 if *s\_city* != "N/A":  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *s\_city*)  
  
if pd.isna(*address\_value*):  
 if *name\_value* and (*s\_address* == '' or *s\_address* == "N/A"):  
 *s\_address* = address\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*city\_value*) and *s\_address* != "N/A" and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_address(*driver\_chrome*, *search*, *delay*, *s\_address*)  
 elif pd.isna(*city\_value*) and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*zip\_code\_value*) and *s\_address* != "N/A" and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *s\_address*)  
 elif pd.isna(*zip\_code\_value*) and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
  
if pd.isna(*city\_value*):  
  
 if *name\_value* and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*address\_value*) and (*s\_address* == '' or *s\_address* == "N/A"):  
 *s\_address* = address\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*zip\_code\_value*) and *s\_address* != "N/A" and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *s\_address*)  
  
 elif *address\_value* and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
 if pd.isna(*name\_value*) and *s\_city* != "N/A" and (*s\_name* == '' or *s\_name* == "N/A"):  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *s\_city*)  
 elif *zip\_code\_value* and *s\_city* == "N/A" and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *zip\_code\_value*)  
 if *s\_name* == '' or *s\_name* == "N/A":  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *s\_city*)  
 if pd.isna(*zip\_code\_value*) and *s\_address* != "N/A" and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
  
 elif *zip\_code\_value* and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *zip\_code\_value*)  
  
if pd.isna(*zip\_code\_value*):  
 if *name\_value* and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*city\_value*) and *s\_zip\_code* != "N/A" and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *s\_zip\_code*)  
 elif pd.isna(*city\_value*) and pd.isna(*address\_value*) is not True and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
 if pd.isna(*address\_value*) and (*s\_address* == '' or *s\_address* == "N/A"):  
 *s\_address* = address\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
  
 elif *address\_value* and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
 if pd.isna(*city\_value*) and *s\_zip\_code* != "N/A" and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *s\_zip\_code*)  
 elif pd.isna(*city\_value*) and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
 if pd.isna(*name\_value*) and *s\_city* != "N/A" and (*s\_name* == '' or *s\_name* == "N/A"):  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *s\_city*)

**Code 12 Part of code with nested "if" statements that returns final data values**

Program posiada wielokrotnie zagniezdzone procedury warunkowe aby w najbardziej efektywny sposob otrzymac poszukiwane dane. Kazda sciezka pozyskiwania danych jest precyzyjnie przygotowana pod wszystkie mozliwe do wystapienia scenariusze ukladu dostarczonych danych. Program zastal zaprojektowany i napisany w taki sposob, ze jest dostepnych 9 sposobow uzyskania kolejnej informacji o danej lokalizacji z kazdej sposrod maksymalnie 4 informacji dostarczonych przez uzytkownika.

data\_to\_file(*name\_value*, *s\_name*, *address\_value*, *s\_address*, *city\_value*, *s\_city*, *zip\_code\_value*, *s\_zip\_code*, *ws*, *record*)  
  
 *message* = "Success! - file is ready"  
 *check* = 'pass'  
  
 *driver\_chrome*.quit()  
 *wb*.save(*download\_file\_path*)  
 else:  
 *message* = "Please use the official version of the template file available on main page"  
 else:  
 *message* = "Please use the official version of the template file available on main page"  
 else:  
 *message* = "please reload page and make sure to choose the data file"  
 else:  
 *message* = "please reload page and make sure to choose the data file"  
else:  
 *message* = "upload filed!"  
return render\_template('summary.html', message=*message*, check=*check*)

**Code 13 Last part of main class code**

Ostatni fragment glownego modulu aplikacji, wprowadza zebrane przez program dane do pliku arkusza kalulacyjnego oraz go zapisuje pod nazwa “S\_DataValues.xlsx”. Skrypt przypisuje do zmiennej “message” informacje koncowa o przeprowadzonym procesie. Zmienna “check” jest jest znacznikiem wykorzystywanym do generowania strony zawierajacej podsumowanie. Jezeli ta zmienna posiada wartosc “pass”, uzytkownik posiada dostep do guzika inicjujacego pobranie pliku posiadajacego dane koncowe zapisane w arkuszu kalkulacyjnym o nazwie “S\_DataValues.xlsx”.

### Callable classes

def cookies(*driver\_chrome*):  
 *delay\_cookie* = 10  
 try:  
 myElem = WebDriverWait(*driver\_chrome*, *delay\_cookie*).until(EC.visibility\_of\_element\_located((By.XPATH, '/html/body/c-wiz/div/div/div/div[2]/div[1]/div[3]/div[1]/div[1]/form[2]/div/div/button')))  
 except TimeoutException:  
 print("Loading took too much time!")  
 try:  
 *driver\_chrome*.find\_element(by=By.XPATH, value="/html/body/div[2]/div[2]/div[3]/span/div/div/div/div[3]/button[2]/div").click()  
 except Exception:  
 pass  
 try:  
 *driver\_chrome*.find\_element(by=By.XPATH, value="/html/body/c-wiz/div/div/div/div[2]/div[1]/div[3]/div[1]/div[1]/form[2]/div/div/button").click()  
 except Exception:  
 pass

**Code 14 Class dedicated to accept cookies on website**

Klasa dzieki ktorej przegladarka moze zatwierdzic wyskakujace okno wymagajace zaakceptowania przechowywania plikow cookie emitowanych przez odwiedzana przez program strone internetowa. Przy pomocy web scrapingu aplikacja lokalizuje przysick zatwierdzajacy oraz nastepnie zostaje on zdalnie klikniety.

def address\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*name\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_address* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[7]/div[1]/button/div[1]/div[2]').text  
 except NoSuchElementException:  
 *s\_address*="N/A"  
  
 if *s\_address*=="N/A" or *s\_address*=="":  
 try:  
 *s\_address* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_address*="N/A"  
  
 if *s\_address*=="N/A" or *s\_address*=="":  
 try:  
 *s\_address* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_address*="N/A"

return *s\_address*

**Code 15 A class that returns address value from name value**

Pozyskiwanie adresu z podanej przez uzytkownika nazwy miejsca rozpoczyna sie od wprowadzenia frazy w pole wyszukiwarki, a nastepnie jej zatwierdzeniu. W momencie nieodnalezienia poprawnej wartosci, program szuka odpowiedniej informacji w innych polozeniach na stronie zawierajacej wyniki wyszukiwania. Nastepnie zwraca wartosc przypisana do zmiennej “s\_address”.

def city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *zip\_code\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*zip\_code\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/div[1]/h1/span[1]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 try:  
 if *s\_city* != "N/A":  
 *temp\_city* = *s\_city  
 s\_city* = *temp\_city*.split()[1]  
 except Exception:  
 print("too fast")  
  
 if *s\_city*=="":  
 *s\_city*="N/A"  
 return *s\_city*

**Code 16 A class that returns city value from zip-code value**

Pozyskiwanie miasta z kodu pocztowego jest bardzo dokladnym posobem na ustalenie miasta w jakiej znajduje sie dana lokalizacja. Wartosc otrzymana z wyszukiwania sklada sie z polaczenia kodu pocztowego oraz miasta, w tam celu program musi rozdzielic pobrany tekst oraz przypisac pod zmienna “s\_city” druga czesc tekstu, posiadajacego indeks o wartosci 1.

def name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *city\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*address\_value* + " " + *city\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.CLASS\_NAME, value="bfdHYd Ppzolf")  
 *s\_name* = *s\_name*.get\_attribute("aria-label")  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/div[1]/h1').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[15]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div/span').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[17]/div/div[2]/div[2]/div[1]/div/div/div/div[1]').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[17]/div/div[2]/div[2]/div[1]/div/div/div/div[1]').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[19]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[15]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[27]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[18]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="":  
 *s\_name*="N/A"  
 return *s\_name*

**Code 17 A class that returns name value from address value and city value**

Klasa zwracajaca nazwe lokalizacji jest jedna z najbardziej rozbudowanych klas wykorzystywanych w aplikacji “GeoFinderUI”. Jej obszernosc wynika z kilku mozliwych pozycji szukanej informacji na stronie poddanej technice przeszukiwania kodu zrodlowego, web scraping. Kod zrodlowy stron “Google Maps” nie posiada ujednoliconych znacznikow elementow znajdujacych sie na stronie z wynikami wyszukiwania. Jest to problematyczne, dlatego program musi sie odwolywac do polozenia na stronie wykorzystujac “X-Path”, ktory jest sciezka kolejno wystepujacych znacznikow naprzyklad takich jak “<div>” lub “<span>” liczonych od pocztku kodu zrodlowego strony.

def city\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
 *exception\_check* = 0  
  
 *search*.send\_keys(*name\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/h2[2]/span').text  
 *exception\_check* = 1  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 if *s\_city*=="N/A" or *s\_city*=="":  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 if *s\_city*=="N/A" or *s\_city*=="":  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 try:  
 if *s\_city*!="N/A" or *s\_city*!="":  
 *temp\_city* = *s\_city* if *exception\_check* == 0:  
 *temp\_city* = *temp\_city*.split(sep=",")[1]  
 *s\_city* = *temp\_city*.split()[1]  
 except Exception:  
 *s\_city*="N/A"  
  
 if *s\_city*=="":  
 *s\_city*="N/A"  
 return *s\_city*

**Code 18 A class that returns city value from name value**

Klasa przypisujaca oraz zwracajaca wartosc zmiennej “s\_city” uzyskanej przez wprowadzenie w pole wyszukiwania nazwy lokalizacji w glownej mierze opiera sie na odnalezieniu pelnego adresu tej lokalizacji. W tym przypadku odnalezienie tej informacji na stronie jest prostrze, poniewaz przy kazdym wyniku wyszukiwania dla pola posiadajacego wartosc adresowa, dolaczony jest skrot dla uzytkownikow pozwalajacy skopiowac z pozniomu strony pelny adres do schowka w pamieci podrecznej urzadzenia. Ze wzgledu na mozliwe wystapienie kodu zrodlowego oraz elementow w nim zawartych w jezyku polskim lub angielskim, kryteria odnajdowania odpowiedniej pozycji zna stronie sa powtorzone, lecz posiadajace wartosc slowna “Copy address” w dwoch jezykach. Nastepnym etapem jest rozdzielenie otrzymanej wartosci pelnego adresu oraz wyciagniecie z niej informacji o miescie w ktorym znajduje sie dana lokalizacja.

def zipcode\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
 *exception\_check* = 0  
  
 *search*.send\_keys(*name\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/h2[2]/span').text  
 *exception\_check* = 1  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="N/A" or *s\_zip\_code*=="":  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="N/A" or *s\_zip\_code*=="":  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"

try:  
 if *s\_zip\_code*!="N/A" or *s\_zip\_code*!="":  
 *temp\_zip\_code* = *s\_zip\_code* if *exception\_check* == 0:  
 *temp\_zip\_code* = *temp\_zip\_code*.split(sep=",")[1]  
 *s\_zip\_code1* = *temp\_zip\_code*.split()[0]  
 *s\_zip\_code* = *s\_zip\_code1* except Exception:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="":  
 *s\_zip\_code*="N/A"  
 return *s\_zip\_code*

**Code 19 A class that returns zip-code value from name value**

Podobnie jak klasa uzyskujaca miasto w ktorej polozona jest dana lokalizacja z nazwy tej lokalizacji, wykorzystuje pelen adres lokalizacji aby nastepnie go rozdzielic i przypisac pod zmienna “s\_zip\_code” wartosc odpowiadajaca kodowi pocztowemu lokalizacji zawartemu w pelnym adresie wyszukiwanej lokalizacji.

def city\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*address\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/div/div[2]/div[2]/div[1]/div/div/div/div[4]/div').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 if *s\_city*=="N/A" or *s\_city*=="" or *s\_city*.upper() == *address\_value*.upper():  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 if *s\_city*=="N/A" or *s\_city*=="" or *s\_city*.upper() == *address\_value*.upper():  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"

try:  
 if *s\_city*!="N/A" or *s\_city*!="":  
 *temp\_city* = *s\_city  
 s\_city* = *temp\_city*.split(",")[1]  
 *temp\_city* = *s\_city  
 s\_city* = *temp\_city*.split()[1]  
 except Exception:  
 *s\_city*="N/A"  
  
 if *s\_city*=="":  
 *s\_city*="N/A"  
  
 return *s\_city*

**Code 20 A class that returns city value from address value**

Klasa uzyskujaca nazwe miasta w kotrym dana lokalizacja sie znajduje na podstawie jej adresu, wykorzystuje ten sam sposob uzyskania informacji jak w poprzednich klasach, poprzez pelny adres lokalizacji, a nastepnie rozdzelenie wartosci tekstowej i wyciagnieciu z niej szukanej wartosci, ktora w tym przypadku jest nazwa miasta w kotej ta lokalizacja sie znajduje.

def zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*address\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/div/div[2]/div[2]/div[1]/div/div/div/div[4]/div').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="N/A" or *s\_zip\_code*=="" or *s\_zip\_code*.upper() == *address\_value*.upper():  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="N/A" or *s\_zip\_code*=="" or *s\_zip\_code*.upper() == *address\_value*.upper():  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 try:  
 if *s\_zip\_code*!="N/A" or *s\_zip\_code*!="" or *s\_zip\_code*.upper() != *address\_value*.upper():  
 *temp\_zip\_code* = *s\_zip\_code  
 s\_zip\_code* = *temp\_zip\_code*.split(",")[1]  
 *temp\_zip\_code* = *s\_zip\_code  
 s\_zip\_code* = *temp\_zip\_code*.split()[0]  
 except Exception:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code* == "" or *s\_zip\_code*.upper() == *address\_value*.upper():  
 *s\_zip\_code*="N/A"  
  
 return *s\_zip\_code*

**Code 21 A class that returns zip-code value from address value**

Aby uzyskac kod pocztowy danej lokalizacji wykorzystujac podany adres tej lokalizacji, program wykorzystuje taki sam sposob jak poprzednia opisana klasa. Sposob polega na otrzymaniu pelnego adresu, a nastepnie rozdzieleniu go i wyciagnieciu fragmentu tekstu odpowiadajacego za kod pocztowy danej lokalizacji.

# Summary

## Application outcome

Aplikacja “GeofinderUI” to program zwracajacy uzytkownikowi dane sugestyjne dotyczace danej lokalizacji, ktore zostaly uzyskane na podstawie minimalnych danych wejsciowych dostarczonych przez uzytkownika. Wszystkie wyniki wyszukiwania sa objete ryzykiem niepoprawnosci danych, wynika to z odnajdywania przez program pierwszego, najbardziej pasujacego wyniku wyszukiwania do wprowadzonych danych wejsciowych. Zalozeniem tej aplikacji jest zredukowanie w najwiekszym mozliwym stopniu czasu potrzebnego do pozyskiwania danych geolokalizacyjnych wymaganych w zdefiniowanym srodowisku biznesowym, wymagajacym uzyskiwania konkretnych danych wyjsciowych. Aplikacje udalo sie sformulowac oraz zapisac, osiagajac w ten sposob w pelni funkcjonalny program. Dzieki temu projektowi osiagnalem znacznie szersze spojrzenie na tematy informatyczne, glownie zwiazane z programowaniem. Moja wiedza zaostala wzbogacona o wiele nowych informacji, umiejetnosci oraz doswiadczen.

## Podziękowania

Najbardziej interesującym oraz satysfakcjonującym elementem procesu tworzenia aplikacji było osiaganie zalozonych celow. Podziękowania należą się **mgr inż Maksymilianowi Kubicy**, za fachowe nadzorowanie projektu oraz za dostarczenie wielu cennych wskazowek w trakcie procesu pisania pracy inzynierskiej. Wiedza oraz doświadczenie jakie zostały przez niego przekazane są bezcenne, a kierownictwo nad projektem pozwoliło na jego prawidłowy rozwój. Za udostępnienie miejsca do pracy, pomoc w trakcie burzy mózgów oraz wsparcie podziękowania należą się wszystkim bliskim - rodzicom i przyjaciołom.

# Bibliography

* *Aplikacja internetowa*. (2022, December). From Wikipedia: https://pl.wikipedia.org/wiki/Aplikacja\_internetowa
* *Bot program*. (2022, May). From Wikipedia: https://pl.wikipedia.org/wiki/Bot\_(program)
* *Code Refactoring*. (2023, January). From Refactoring: https://refactoring.com/
* *Discover Intellij Idea*. (2023, January). From jetbrains: https://www.jetbrains.com/help/idea/discover-intellij-idea.html
* Flower, M., Beck, K., Opdyke, W., & Brant, J. (2002). Refactoring Helps You Find Bugs. In M. Flower, *Refactoring: Improving the design of Existing Code* (pp. 48-49). Addison-Wesley.
* *Framework*. (2022, October). From Wikipedia: https://pl.wikipedia.org/wiki/Framework
* *Google Maps*. (2022, May). From https://www.google.com/maps/about/#!/
* *HTML*. (2023, January). From https://html.spec.whatwg.org/
* *Markup language*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Markup\_language
* *Materialize*. (2023, January). From https://materializecss.com/
* *Microsoft Excel*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Microsoft\_Excel
* *Openpyxl*. (2022, May). From https://openpyxl.readthedocs.io/en/stable/
* *Pandas software*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Pandas\_(software)
* *Python*. (2023, January). From https://docs.python.org/pl/3/
* *Selenium*. (2022, December). From https://www.selenium.dev/
* *Uniform Resource Locator*. (2022, December). From Wikipedia: https://pl.wikipedia.org/wiki/Uniform\_Resource\_Locator
* *Web scraping*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Web\_scraping

# Attachment

## instructions

Instalacja gry odbywa się poprzez włączenie aplikacji „PandaMunch.apk” na urządzeniu z zainstalowanym systemem Android oraz z włączoną możliwością instalacji aplikacji z nieznanych źródeł.

Wybór opcji w grze odbywa się poprzez dotknięcie korespondujących przycisków.

Aby rozpocząć rozgrywkę, należy dotknąć przycisku „Start”. Aby kontynuować rozgrywkę po restarcie gry, należy dotknąć przycisk „Continue”.

Na każdym z poziomów gracz ma możliwość zebrania trzech bambusów. Aby przejść do dalszych poziomów, trzeba zebrać odpowiednią ich ilość.

Zadaniem gracza jest omijanie przeszkód, jakie znajdują się na jego drodze. Aby przeskoczyć nad przepaścią, rzeką, bądź wężem gracz musi wykonać skok. Aby ominąć ekologa, musi wykonać unik w prawo lub lewo. W przypadku napotkania półki skalnej, zadaniem gracza jest wykonanie akcji turlania.

Sterowanie w grze:

Swipe w górę – skok

Swipe w dół – turlanie

Swipe w prawo – unik w prawo

Swipe w lewo – unik w lewo

## Table of Figures

[**Figure 1 Presentation markup example** 8](#_Toc126792744)

[**Figure 2 Procedural markup example** 9](#_Toc126792745)

[**Figure 3 Descriptive markup example** 10](#_Toc126792746)

[**Figure 4 Google Maps search result** 21](#_Toc126792747)

[**Figure 5 GeoFinderUI application home page with Materialized styling packages**.......................................................................................................................22](#_Toc126792748)

[**Figure 6 the original diagrammatic process of the operation of the GeofinderUI application** 24](#_Toc126792749)

[**Figure 7 sample visualization of the GeoFinderUI start page** 25](#_Toc126792750)

[**Figure 8 data acquisition points by GeoFinderUI application** 25](#_Toc126792751)

[**Figure 9 Information about the authors of materializecss.com** 30](#_Toc126792752)

## Table of Pictures

[**Picture 1 Left side, Martin Flower Picture 2 Right side, William Opdyke** 7](#_Toc126793112)

[**Picture 3 Left side, IntelliJ dark theme Picture 4 Right side, IntelliJ default light theme** 13](#_Toc126793113)

[**Picture 5 Left side, Nyan Progress Bar Picture 6 Right side, Snyk add-in** 14](#_Toc126793114)

[**Picture 7 Left side, Excel Worksheet Picture 8 Right side, Visual Basic for Applications** 15](#_Toc126793115)

[**Picture 9 Left Side, CWI building Picture 10 Right side, Guido van Rossum** 16](#_Toc126793116)

[**Picture 11 Jason Huggins** 17](#_Toc126793117)

[**Picture 12 Wes McKinney** 18](#_Toc126793118)

[**Picture 13 Left side, Eric Gazoni Picture 14 Right side, Charlie Clark** 19](#_Toc126793119)

[**Picture 15 Left side, Timothy Berners-Lee Picture 16 Right side, Robert Cailliau**.........................................................................................................................20](#_Toc126793120)

## Table of Code

[**Code 1 The <Head> section with the style part needed for positioning the content of the GeoFinderUI start page** 26](#_Toc126793275)

[**Code 2 Part of the code that allows the user to download the official version of the data template acceptable by the application** 27](#_Toc126793276)

[**Code 3 The form section needed for receiving and sending files and information to the back-end code** 28](#_Toc126793277)

[**Code 4 The part of the code responsible for the footer visible on each page of the "GeoFinderUI" web application** 29](#_Toc126793278)

[**Code 5 The part of the code that contains the information and its layout on the summary page** 30](#_Toc126793279)

[**Code 6 Part of the code containing the import of the necessary tools used in the "GeoFinderUI" application** 31](#_Toc126793280)

[**Code 7 Side methods in "home.py"** 32](#_Toc126793281)

[**Code 8 The beginning of the main method responsible for the functionality of the application** 32](#_Toc126793282)

[**Code 9 Part of the code responsible for preparing and cleaning the location for storing data files** 33](#_Toc126793283)

[**Code 10 Part of the code responsible for preparing the data file and launching the automated browser** 34](#_Toc126793284)

[**Code 11 A loop procedure that goes through all the records entered in the data file** 34](#_Toc126793285)

[**Code 12 Part of code with nested "if" statements that returns final data values** 36](#_Toc126793286)

[**Code 13 Last part of main class code** 37](#_Toc126793287)

[**Code 14 Class dedicated to accept cookies on website** 38](#_Toc126793288)

[**Code 15 A class that returns address value from name value** 39](#_Toc126793289)

[**Code 16 A class that returns city value from zip-code value** 39](#_Toc126793290)

[**Code 17 A class that returns name value from address value and city value** 41](#_Toc126793291)

[**Code 18 A class that returns city value from name value** 43](#_Toc126793292)

[**Code 19 A class that returns zip-code value from name value** 44](#_Toc126793293)

[**Code 20 A class that returns city value from address value** 45](#_Toc126793294)

[**Code 21 A class that returns zip-code value from address value** 47](#_Toc126793295)

## What CD contains

* **\Maksymilian\_Kubica\_s12605\_PracaDyplomowa\_EPG**
  + Elektroniczna wersja pracy inżynierskiej
* **\SOURCE**
  + Projekt źródłowy gry mobilnej
* \**BUILD**
  + Skompilowana wersja gry mobilnej