Projekt zaliczeniowy

Proces ETL

Grupa projektowa: 20

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Imię | Nazwisko | Numer albumu | Grupa dziekańska | Wkład w prace nad projektem[[1]](#footnote-1) | Udział procentowy |
| Monika | Dreliszek | 213739 | WZISN2-1111 | Extract, Transform,  plan rozwiązania, wybór technologii | 30 |
| Marek | Bacewicz | 213834 | WZISN2-1111 | Load, konfiguracja i obsługa bazy danych, komunikacja frontend-backend,  plan rozwiązania, wybór technologii | 50 |
| Łukasz | Fundament | 214040 | WZISN2-1111 | GUI | 20 |

|  |
| --- |
| \_\_/70 pkt |

Spis treści

[1. Dokumentacja techniczna 3](#_Toc532515689)

[1.1. Opis aplikacji 3](#_Toc532515690)

[1.2. Wykorzystane technologie 3](#_Toc532515691)

[1.3. Wymagania systemowe 4](#_Toc532515692)

[Środowisko developerskie 4](#_Toc532515693)

[Minimalne wymagania systemowe 4](#_Toc532515694)

[1.4. Przygotowanie środowiska 4](#_Toc532515695)

[Instalacja środowiska Python 4](#_Toc532515696)

[Instalacja MySQL Server i MySQL Workbench 5](#_Toc532515697)

[Plik konfiguracyjny 5](#_Toc532515698)

[Linki do oprogramowania 5](#_Toc532515699)

[Biblioteki 5](#_Toc532515700)

[1.5. Model danych 6](#_Toc532515701)

[1.6. Zestawienie klas i funkcji 7](#_Toc532515702)

[2. Instrukcja obsługi 11](#_Toc532515703)

[2.1. Uruchomienie aplikacji 11](#_Toc532515704)

[2.2. Opis funkcjonalności aplikacji 11](#_Toc532515705)

[2.3. Zrzuty ekranu wizualizujące wygląd i działanie aplikacji 12](#_Toc532515706)

# Dokumentacja techniczna

Celem niniejszego dokumentu jest przedstawienie technicznego aspektu programu, specyfikacji oraz wymaganej konfiguracji środowiska.

## Opis aplikacji

Aplikacja ma na celu przeprowadzenie procesu ETL, składającego się z następujących etapów:

1. Pozyskanie ofert wynajmu mieszkań z wybranych spośród 10 polskich miast z serwisu internetowego www.gumtree.pl

2. Przekształcenie pozyskanych danych (atrybutów charakteryzujących mieszkania na wynajem, tj. liczba pokoi, cena, metraż) do formatu umożliwiającego zasilenie bazy danych.

3. Zasilenie bazy danych

Kolejne kroki procesu ETL są wywoływane przez użytkownika z poziomu interfejsu graficznego aplikacji. Możliwe jest uruchomienie procesu zarówno w całości jak i etapami. Dodatkowo program umożliwia:

1. Wyczyszczenie bazy danych.
2. Eksport danych z bazy do plików tekstowych (1 rekord to 1 plik).
3. Eksport całości danych do pliku .csv.
4. Możliwość wyboru miasta.

Aplikacja wyświetla informację o obecnie wykonywanym etapie procesu oraz o statusie podsumowującym działanie.

## Wykorzystane technologie

|  |  |  |
| --- | --- | --- |
| Technologia | Wersja | Zastosowanie |
| Python | 3.7 | Język programowania użyty do stworzenia niniejszej aplikacji |
| MySQL Server | 8.0 | System zarządzania bazą danych |
| MySQL Workbench CE | 8.0 | Wizualne środowisko do zarządzania baza danych |
| JetBrains PyCharm CE | 2018.3.1 | Python IDE |

## Wymagania systemowe

### Środowisko developerskie

Zestawienie sprzętowe środowiska developerskiego

|  |  |
| --- | --- |
| Specyfikacja | |
| CPU | Intel Core i7-4800MQ @ 2.7 GHz |
| Rdzenie CPU | 4 |
| RAM | 16 GB |
| Dysk | SDD 240 GB + HDD 500 GB |
| System | Win 10 Pro |

### Minimalne wymagania systemowe

Zestawienie bazujące na minimalnych wymaganiach dostawców oprogramowania.

|  |  |  |
| --- | --- | --- |
| Wymaganie | Minimalne | Zalecane |
| CPU | 3 GHz | 2 x 2 GHz |
| Rdzenie CPU | 1 | 2/4 |
| RAM | 4 GB | 6 GB |
| Dysk | n/a | n/a |
| System | Win 7 lub wyższy | n/a |

## Przygotowanie środowiska

### Instalacja środowiska Python

1. Po pobraniu środowiska w wersji odpowiedniej do posiadanego systemu (x86 lub x64) z dodatku z linkami należy uruchomić plik instalacyjny i postępować zgodnie z instrukcjami z okna instalacyjnego. Należy dodać w systemie tzw. zmienną środowiskową w polu PATH. W tym celu należy kliknąć prawym przyciskiem myszy 🡪 Komputer 🡪 Właściwości 🡪 Zaawansowane Ustawienia Systemu 🡪 Zaawansowane 🡪 Zmienne środowiskowe
2. W oknie zmienne systemowe należy odnaleźć zmienną ,,PATH", kliknąć Edytuj, W polu należy dopisać ścieżkę prowadzącą do interpretera Pythona oraz skryptów. przykładowa ścieżka: *C:\ Python37\_64; C:\Python37\_64\Scripts*

### Instalacja MySQL Server i MySQL Workbench

1. Sugerowanym sposobem instalacji jest skorzystanie z kreatora instalacji MySQL i wybranie typu instalacji *custom*. W projekcie wymagane są tylko dwa oprogramowania: MySQL Server i MySQL Workbench. Po skończonej instalacji, kreator poprowadzi nas przez kroki konfiguracyjne. Wymagane jest jedynie skonfigurowanie MySQL jako Windows Service. Jako port należy wskazać dowolny wolny.

### Plik konfiguracyjny

1. W folderze projektu wymagany jest plik konfiguracyjny config.py. Zaimportowanie pliku w kolejnych modułach umożliwia korzystanie ze zmiennych globalnych zdefiniowanych w obrębie pliku.
2. W parametrach konfiguracyjnych wyróżniamy:
   * + Ścieżka do folderu Temp
     + Ścieżki do podfolderów, w których przechowywane są pliki będące efektami kolejnych etapów procesu ETL
     + Parametry konfigurujące połączenie z bazą danych
     + Słownik grupujący nazwę miasta z identyfikatorem wykorzystywanym w linku do strony z ogłoszeniami dla danego miasta
     + Limit stron, z których należy pobrać dane

### Linki do oprogramowania

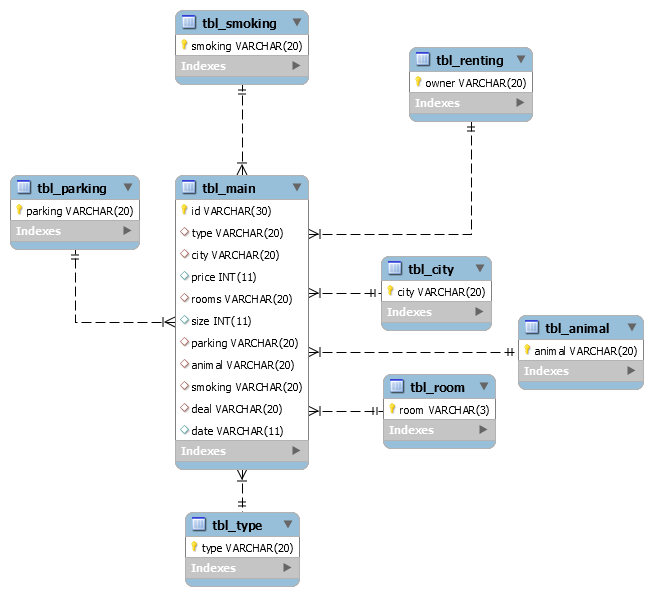
|  |  |
| --- | --- |
| Oprogramowanie | Pobieranie |
| Python interpreter | www.python.org/downloads/release/python-371/ |
| JetBrains Pycharm Community Edition | https://www.jetbrains.com/pycharm/download/download-thanks.html?platform=windows&code=PCC |
| MySQL Installer | https://dev.mysql.com/downloads/mysql/ |

### Biblioteki

|  |  |
| --- | --- |
| Biblioteka | Zastosowanie |
| sys | Dostęp do parametrów i funkcji systemowych |
| time | Funkcje związane z czasem |
| json | Parsowanie i praca z plikami w formacie json |
| os | Praca z plikami i folderami |
| glob | Praca z plikami i folderami |
| csv | Funkcje związane z plikami csv |
| datetime | Funkcje związane z datą i czasem |
| shutil | Praca z plikami i folderami |
| requests | Wysyłanie zapytań http |
| bs4 | Parsowanie html |
| mysql.connector | Łączenie się z bazą MySQL |
| PyQt5 | Tworzenie GUI aplikacji |

Większość z powyższych bibliotek to biblioteki standardowe wchodzące w skład dystrybucji Pythona. Pozostałe biblioteki wymienione są w pliku requirements.txt i należy je dodatkowo zainstalować. W tym celu (po dodaniu ścieżki Pythona do zmiennych systemowych) należy uruchomić wiersz poleceń jako administrator i wpisać następującą komendę: *pip install <ścieżka do pliku requirements.txt>.*  W przypadku niepowodzenia należy wpisać komendę dla każdej paczki: pip install <nazwa biblioteki>

## Model danych

Do projektu zostaw wybrany relacyjny model bazy danych. Jako system zarządzania baza danych wykorzystywany jest MySQL. Relacje miedzy tabelami przedstawione są na załączonym diagramie. Główna tabela połączona jest z tabelami pobocznymi zawierającymi wartości pobieranych pól przyjmowanych przez bazę. Dane są ładowane przez program do tabeli głównej pojedynczo. Dane nie zostaną przyjęte, jeżeli zawierają wartości nie znajdujące sie w tabelach pobocznych.

## Zestawienie klas i funkcji

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nazwa | Typ | Opis | Argumenty wejściowe | | Argumenty wyjściowe |
| **extract.py** | | | | | |
| data\_scrape(city) | funkcja | * Pobranie atrybutów ogłoszeń z wybranego miasta * Utworzenie dla każdego ogłoszenia słownika, na który składają się pary: nazwa atrybutu - wartość atrybutu. * Zapis każdego słownika w oddzielnym pliku o unikalnej nazwie będącej identyfikatorem ogłoszenia: *[ID].json*. * Pliki są przechowywane w tymczasowej lokalizacji w folderze *extracted*. | Miasto wybrane z poziomu interfejsu aplikacji | | Status:  -exit\_code  -message |
| **transform.py** | | | | | |
| data\_transform() | funkcja | * Iteracja przez każdy plik .json z folderu *extracted* * Wyodrębnienie, przetworzenie i zapisanie do nowego słownika atrybutów w kolejności i formie zgodnej z wymaganiami bazy danych, tj:   **- ID:** unikalny identyfikator ogłoszenia  **- Rodzaj nieruchomości**: mieszkanie / dom  **- Lokalizacja**: bezpośrednio z parametru determinowanego przez użytkownika  **- Cena:** usunięcie jednostki [zł], w przypadku nienumerycznych wartości zastąpienie pustą wartością  **- Liczba poko**i: Zastąpienie wartości ‘Kawalerka’ wartością ‘1’ oraz wszystkich wartości większych od 5 wartością ‘5<’  -**Wielkość:** usunięcie jednostki [m2]  -**Parking:** ulica / garaż / brak  **-Dozwolone zwierzęta:** tak / nie  **-Dozwolone palenie:** tak / nie  **-Do wynajęcia przez:** właściciel / agencja  **-Data dodania**   * Przypisanie do atrybutu wartości pustej w przypadku niewystąpienia w ogłoszeniu któregoś z powyższych atrybutów lub w przypadku wystąpienia wartości niedozwolonej. * Zapis każdego słownika w oddzielnym pliku o unikalnej nazwie będącej identyfikatorem ogłoszenia: *[ID].json*. * Pliki są przechowywane w tymczasowej lokalizacji w folderze *transformed*. * Obsługa błędów i zwrócenie statusu | n/a | | Status:  -exit\_code  -message |
| **load.py** | | | | | |
| load\_files() | funkcja | * Nawiązanie połączenia z bazą * Iteracja po folderze *transformed* * Sczytanie plików .json * Załadowanie zawartości plików do bazy (kolejność pól musi być zgodna z kolumnami w głównej tabeli) * Przeniesienie przeprocesowanych plików do folderu *processed* | n/a | | Status:  -exit\_code  -message |
| output\_db() | funkcja | * Nawiązanie połączenia z bazą * Pobranie nazw kolumn * Pobranie wszystkich danych z głównej tabeli * Wypisanie zawartości do pliku wyjściowego .w formacie csv | n/a | | Status:  -exit\_code  -message |
| output\_files() | funkcja | * Nawiązanie polaczenia z bazą * Pobranie nazw kolumn * Pobranie wszystkich danych z głównej tabeli * Utworzenie wyjściowego pliku tekstowego dla każdego wiersza tabeli | n/a | | Status:  -exit\_code  -message |
| clear\_db() | funkcja | * Nawiązanie połączenia z bazą * Wywołanie funkcji czyszczącej tabelę główną bazy z obiektu klasy DatabaseConnection | n/a | | Status:  -exit\_code  -message |
| DatabaseConnection | klasa | * Nawiązanie i przerwanie połączenia z bazą danych * Wykonywanie podstawowych zapytań bazodanowych: select, insert, clear * Przy inicjowaniu obiektu następuje połączenie z bazą danych przy pomocy mysql connetor’a, natomiast przy destrukcji połączenie jest zrywane. | n/a | | n/a |
| select() | metoda | * Wysłanie zapytania typu select do bazy i zwrócenie jego wartości. | query – zapytanie select | | query\_results –lista rekordów |
| input\_one() | metoda | * Wysłanie zapytania typu insert do bazy. | query – zapytanie insert | | n/a |
| clear\_database() | metoda | * Czyszczenie zawartości głównej tabeli w bazie danych | n/a | | n/a |
| **main\_app.py** | | | | | |
| ETL | klasa | * Klasa służąca do wywoływania funkcji z innych modułów. | n/a | n/a | |
| GuiWebScraper | klasa | * Klasa odpowiedzialna za inicjowanie GUI aplikacji i skoordynowanie ze sobą wszystkich funkcji z pozostałych modułów. | n/a | n/a | |
| initUI() | metoda | * Inicjuje graficzny interfejs użytkownika * Tworzy kolekcje elementów, używana później do blokowania przycisków w trakcie egzekucji procesów | n/a | n/a | |
| etl() | metoda | * Przeprowadza cały proces ETL | n/a | n/a | |
| handle\_f\_output() | metoda | * Obsługuje argumenty wyjścia z funkcji wywoływanych z innych modułów * W przypadku sukcesu uaktualnia pasek statusu aplikacji * W przypadku błędu bąbelkuje błąd obsługiwany wyżej w aplikacji | n/a | n/a | |
| refresh\_buttons() | metoda | * Uaktualnia status przycisków GUI na aktywny/nieaktywny. * Obsługuje dwa scenariusze:   ustawienie wszystkich przycisków na nieaktywny  oraz ustawienie statusu przycisków zgodnie ze słownikiem | flag\_disable - zmienna typu boolean, decydująca, który scenariusz powinien zostać wybrany | n/a | |
| extract() | metoda | * Funkcjonalność przypisana do przycisku extract. * Uruchamia odpowiednia metodę z klasy ETL. * Przekazuje wybrane miasto z listy rozwijalnej * Po zakończeniu odblokowuje możliwość przeprowadzenia procesu transformacji | n/a | n/a | |
| transform() | metoda | * Funkcjonalność przypisana do przycisku transform * Uruchamia odpowiednią metodę z klasy ETL * Po zakończeniu usuwa pliki z ekstrakcji i odblokowuje możliwość ładowania | n/a | n/a | |
| load() | metoda | * Funkcjonalność przypisana do przycisku load * Uruchamia odpowiednią metodę z klasy ETL * Po zakończeniu usuwa pliki z folderu *transform* | n/a | n/a | |
| print() | metoda | * Funkcjonalność przypisana do przycisku print * Uruchamia odpowiednią metodę z klasy ETL * Odpowiada za wyeksportowanie głównej tabeli bazy do pliku .csv | n/a | n/a | |
| print\_files() | metoda | * Funkcjonalność przypisana do przycisku Print files * Uruchamia odpowiednia metodę z klasy ETL * Odpowiada za eksport z bazy rekordów do indywidualnych plików | n/a | n/a | |
| clear() | metoda | * Funkcjonalność przypisana do przycisku Clear * Uruchamia odpowiednia metodę z klasy ETL * Odpowiada za wyczyszczenie głównej tabeli bazy danych | n/a | n/a | |
| **helpers.py** | | | | | |
| get\_from\_dict() | funkcja | * Pobranie wartości odpowiadającej danemu kluczowi z danego słownika | dict\_key  dict\_name | dict\_vale | |
| add\_to\_dict() | funkcja | * Dodanie zadanej wartości do zadanego klucza w zadanym słowniku | dict\_key,  dict\_value  dict\_name | n/a | |
| read\_json() | funkcja | * Sczytuje plik w formacie json do słownika | path | body – zmienna typu słownik | |
| write\_json() | funkcja | * Zapisuje słownik do pliku w formacie json | path,  data – zmienna typu słownik | n/a | |
| write\_pretty\_json() | funkcja | * Zapisuje słownik do pliku w formacie json, zachowując czytelne formatowanie – oddzielne linie dla każdej pary klucz - wartość | path,  data – zmienna typu słownik | n/a | |
| dict\_to\_string() | funkcja | * Konwertuje słownik do zmiennej typu string | my\_dict – zmienna typu słownik | out\_str | |
| clear\_directory() | funkcja | * Usuwa wszystkie pliki z folderu | path | n/a | |
| clear\_temp\_directories() | funkcja | * Usuwa wszystkie pliki z wszystkich tymczasowych lokalizacji | n/a | n/a | |
| get\_filename() | funkcja | * Zwraca nazwę pliku z podanej ścieżki | path | file\_name | |

# Instrukcja obsługi

## Uruchomienie aplikacji

W obecnej postaci w celu zapewnienia pełnej funkcjonalności aplikacja musi być uruchomiona z poziomu IDE. Odbywa się to przez uruchomienie głównego modułu programu: main\_app.py

## Opis funkcjonalności aplikacji

Poniżej przedstawione zostały funkcjonalności aplikacji. Funkcjonalności aplikacji uruchamiane są przez interakcje z elementami GUI:

* Lista rozwijana z nazwami miast - parametryzuje miasto, dla którego ma zostać wykonany web scraping.
* ETL - przycisk uruchamiający cały proces ETL
* Extract, Transform, Load - przyciski uruchamiające poszczególne kroki procesu. Uruchomienie następnego jest możliwe jedynie po wykonanie poprzedniego.
* Print – przycisk eksportujący dane z tabeli głównej bazy danych do pliku wyjściowego w formacie csv
* Print files – przycisk eksportujący rekordy z głównej tabeli bazy danych do plików .json zawierających pojedyncze rekordy
* Clear - przycisk czyszczący zawartość głównej tabeli

Podczas pracy aplikacji usuwane są pliki tymczasowe:

* Przy uruchamianiu programu oraz procesu Extract czyszczone są wszystkie tymczasowe lokalizacje.
* Po wykonaniu procesu Transform usuwane są pliku ekstrakcji.
* Po wykonaniu procesu Load usuwane są pozostałości po procesie Transform.

Podczas pracy programu nie jest możliwa interakcja z GUI aplikacji

## Zrzuty ekranu wizualizujące wygląd i działanie aplikacji

|  |  |
| --- | --- |
| Opis | Zrzut ekranu |
| Widok aplikacji po uruchomieniu | https://lh3.googleusercontent.com/SNQJctfNnBQ-6GAMwPd9AtSSbFhEJhgbhKcFdqoQlMVlJs0oQ8AJfcBvyA-4gBKknw3OunXyLcG7UC3sW7SHtjWbTgQdY7FBI8g3N0W8V2SjQ1LWqzsYrQFop9Ek-4-QQQ |
| Aplikacja podczas pracy | https://lh4.googleusercontent.com/4xMcMaJm_10zpswN0q80zW2ZL02Hp9Y6wgfsibe0rLp3_eIIumIbi3aR8nPcNZjMulaby683HS_HyuEl6Msu0tWjaNSiqW6-foDLvDCB6cbFFRibxiDgW1sc3dqrhc70hg |
| Widok po zakończeniu jednego z procesów | https://lh3.googleusercontent.com/KzgHSWSmaWAG18kKF0QPFNnpr1sdVdtGMrkhzvW9fl1j7LEbypFmtwWAUpRMNGUfIkMP1fWSQFiWsp_as1K_xVx9ZefSbS62cCiTJQjs-WRRHYCsLSi-kSJEXxfltDqYEQ |
| Widok prezentujący wybieranie miasta, dla którego ma odbyć sie ekstrakcja danych | https://lh4.googleusercontent.com/mvxRUiwkflBiWSNe9s8peESfTec-igKA5y15DDVrWDd7KICDFSqCliOS9xSLWGaY9fbEN9wNlTsmEEBCUXG4tGolhbhAgqurV8oM9AM4XtAog2rfywHYVbqcyiBgfbswQQ |
| Widok prezentujący stan aplikacji po wykonaniu procesu transform. Przycisk Transform jest wyszarzony, ponieważ pliki tymczasowe pochodzące z ekstrakcji zostały juz usunięte. | https://lh4.googleusercontent.com/dA19QYH9ad76ATYrTV5RQLHkHpda7-57kLtRxIumdQOQf9hrTvooNpAWK4DQV6lCBqyYj7jejrghBtQPOm0EfvlEqlIKYnZyEab8VK686wRl2zfxh3GfGEEePldL-vThWg |

1. proszę wymienić konkretne zadania [↑](#footnote-ref-1)