

Dodatki

W niniejszym dodatku umieszczono wskazówki dotyczące uruchomienia poszczególnych skryptów projektu, które mogą być pomocne podczas próby odtworzenia procesu budowania aplikacji lub pomocne w przypadku chęci wykonania modyfikacji tego projektu. Co ważne, instrukcja uruchomienia jest przeznaczona dla systemu *Windows*.

Dodatek A Procedura uruchomienia projektu

W pierwszym kroku należy upewnić się, czy zainstalowaliśmy wszystkie wymagane biblioteki.

W celu wykonania automatycznego przeszukania ‘*papierowej bazy danych*’ należy uruchomić skrypt *main.py*. Możemy tego dokonać przy użyciu zintegrowanego środowiska programistycznego (np. *PyCharm*) lub prosto z linii poleceń za pomocą komendy:

- *python main.py*

W pliku *main.py* umieszczone są polecenia, mające na celu wykonanie skryptów w następującej kolejności:

- Utworzenie struktury bazy danych.
- Uzupełnienie utworzonej bazy danych o dane odczytane z plików *.xls* oraz ustalonych wskutek odczytu ścieżek pośrednich do tych plików (z tych danych wynikają takie informacje jak rok akademicki, typ semestru, nazwa wydziału, nazwa prowadzącego itd.).
- Detekcja ‘pustych’ odpowiedzi, usunięcie plików graficznych niewnoszących konstruktywnych informacji oraz wykonanie konwersji plików graficznych do formatu *.pdf*.
- Utworzenie schematu pliku *Ankiety.xlsx*, w którym zamieścimy dane w formie sporządzonego raportu w kontekście przeprowadzonych ankiet.
- Uzupełnienie pliku *Ankiety.xlsx* danymi wyciągniętymi z naszej bazy danych i poddanymi pewnej obróbce (tzn. wykonujemy pewne operacje matematyczne, aby w sposób poprawny uzupełnić pola arkusza w stworzonym szkieletie raportu itp.).

Oczywiście każdy skrypt możemy uruchomić oddzielnie. Należy jednak tutaj pamiętać o podaniu odpowiednich argumentów w wierszu poleceń, ponieważ niektóre skrypty owych wymagają, a w pliku *main.py* zostały one podane.

Po wykonaniu przedstawionej części możemy już przejść do próby uruchomienia aplikacji okienkowej. W tym celu należy uruchomić skrypt *app.py*, znajdujący się w katalogu o nazwie „*frontend*”. W tym celu należy użyć polecenia „*python app.py*”

Po wykonaniu tej komendy możemy korzystać z aplikacji w formie deweloperskiej. W przypadku, gdy chcemy zbudować plik wykonywalny typu *.exe*, musimy wykonać następujące kroki [27]:

1. Upewnij się, czy ścieżka bezpośredniego dostępu do pliku wykonawczego *python.exe* jest ustawiona jako jedna ze zmiennych środowiskowych *Path* (domyślną ścieżką jest:
 - o `%LOCALAPPDATA%\Programs\Python\Python37\Scripts`
jednakże może ona się różnić w zależności od miejsca zainstalowania *Python'a* na naszym dysku).
 2. Otwórz *Eksplorator Windows* i przejdź do folderu *Script*, w który znajduje się plik *python.exe*. Następnie przekopiuj do tego folderu nasz folder „*frontend*”, który zawiera pliki dotyczące naszej aplikacji .
 3. Następnie otwórz Procesor poleceń systemu Windows (*cmd.exe* z menu startu) i ustaw w nim ścieżkę rozkazów na poziomie folderu „*Scripts*” (domyślnie „`%LOCALAPPDATA%\Programs\Python\Python37\Scripts`”).
 4. Zainstaluj *pyinstaller* dzięki użyciu komendy:
 - o `pip install pyinstaller`
 5. Po pozytywnym zainstalowaniu *pyinstaller* wykonaj komendę:
 - o `pyinstaller -w -F --icon=frontend\icon.ico frontend\app.py`
- Po ukończeniu tego polecenia w folderze „*Script*” powinny pojawić się:
- o Folder „*build*”
 - o Folder „*dist*”
 - o Plik *app.spec*

W folderze „*dist*” znajduje się plik *app.exe*, którego uruchomienie skutkuje otwarciem naszej aplikacji. Co ważne, jeżeli w pliku *app.py* nie umieściliśmy wywołania pliku *main.py*, którego funkcje opisano na początku tego dodatku, to po zbudowaniu pliku *app.exe*, należy dodać na jego poziomie wcześniej utworzony plik *my_db.db*.

Jeżeli chcemy ponownie utworzyć plik *app.exe*, to przed ponownym wywołaniem wcześniej przedstawionej komendy *pyinstaller* należy usunąć foldery „*build*” i „*dist*” oraz plik *app.spec*

Dodatek B Opis szczegółów implementacji projektu

Projekt zawiera następujące skrypty:

- *main.py*

Nie ma w tym skrypcie relatywnie trudnych rzeczy. Głównym zadaniem tego skryptu jest wywołanie obiektów funkcji poszczególnych instrukcji *def* znajdujących się w pozostałych skryptach (patrz Dodatek A).

Katalog *backend*:

- *baseDb.py*

W tym skrypcie zawarto instrukcje definiujące strukturę większości tabel zawartych w bazie danych oraz umieszczono instrukcje wspierające tworzenie zapytań do bazy danych wraz z obsługą połączenia do tej bazy.

- *fillDb.py*

W tym pliku z użyciem instrukcji pętli gromadzone są informacje zawarte w katalogach ze struktury drzewa odwróconego (takie jak rok, semestr prowadzący itd.) oraz w odpowiednim miejscu wywoływany jest skrypt odpowiedzialny za odczyt danych z plików *.xls* (arkuszy kalkulacyjnych). Tabele bazy danych zostają wypełniane danymi w kontekście informacji o latach, semestrach, wydziałach, prowadzących oraz kursach. W innej instrukcji *def* tego skryptu uzupełniane danymi są tabele „*Comments*” i „*PDF_files*” w kontekście informacji o komentarzach, jakie udzielili poszczególni użytkownicy.

- *excel2db.py*

Za pomocą tego skryptu i jej głównej instrukcji odczytujemy informacje umieszczone w arkuszach kalkulacyjnych. Tworzymy nowe tabele dotyczące wyników, statystyk lub też inne, w zależności od zawartości tych arkuszy. Ta instrukcja wymaga podania kilku parametrów (takich jak *userId*, *departmentId*, *courseId* i inne) dlatego wywołujemy ją w innych skryptach. Ten skrypt zawiera też inne instrukcje służące do mapowania danych.

- *images2pdf.py*

Z użyciem tego skryptu tworzymy odpowiednie pliki *.pdf* dotyczące komentarzy udzielonych przez respondentów i dołączamy do nich odpowiednie pliki graficzne zawierających treści tych komentarzy

- *findFiles.py*

Skrypt pomocniczy ułatwiający eksplorację plików w strukturze drzewa odwróconego.

- *textDetection.py*

Skrypt odpowiedzialny za rozpoznanie, czy na konkretnym pliku graficznym może występować komentarz udzielony przez respondenta. Szczegóły implementacji zostały opisane w tymże pliku projektu. W tym pliku zawarto również utworzenie zbiorczego folderu „komentarze”, w którym to umieszczono wszystkie pliki *.pdf* związane z komentarzami respondentów.

- *surveysSheet.py*

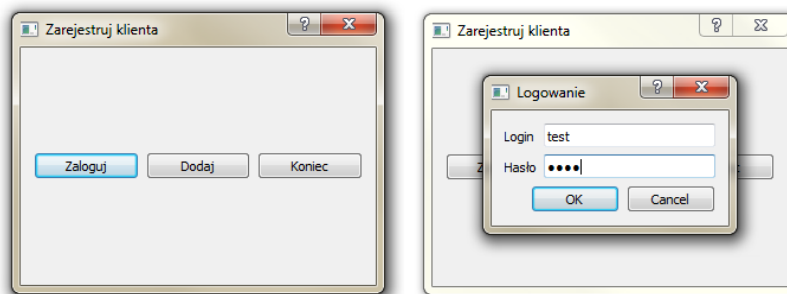
Skrypt służący do utworzenia raportu końcowego z przeprowadzonych ankiet. Zawiera instrukcję do stworzenia pliku i nadaniu jej podstawowej struktury (utworzenie „menu”) oraz instrukcję, której wykonanie skutkuje uzupełnieniem arkusza pozostałymi informacjami.

- *labels.py*
Plik zawierający tłumaczenia etykiet użytych w całym projekcie.
- *helpres.py*
Skrypt zawierające proste instrukcje pomocnicze, np. konwersja daty na wartość *timestamp*.
- *loginDB.py*
Skrypt wykonujący prostą bazę danych obsługującą rejestrację i logowanie użytkowników aplikacji.

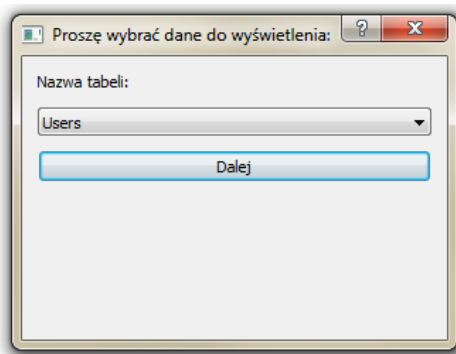
Katalog *frontend*:

- *app.py*
Główny skrypt widoku aplikacji. Definiuje wygląd głównego okna aplikacji oraz zawiera wiele instrukcji obsługujących pewne zdarzenia, które są wywoływane podczas standardowego korzystania z aplikacji przez użytkownika.
- *charts.py*
Skrypt w głównej mierze zawiera instrukcje, których wywołanie skutkuje powstawaniem nowych wykresów pudełkowych czy słupkowych.
- *login.py*
Główny skrypt definiujący wygląd panelu logowania. W nim też nawiązujemy połączenie z bazą danych przeznaczoną do przechowywania danych logowania.
- *loginGui.py*
Skrypt pomocniczy zawierający instrukcje definiujące wygląd mniejszych komponentów w panelu logowania.

Dodatek C Wizualizacja aplikacji

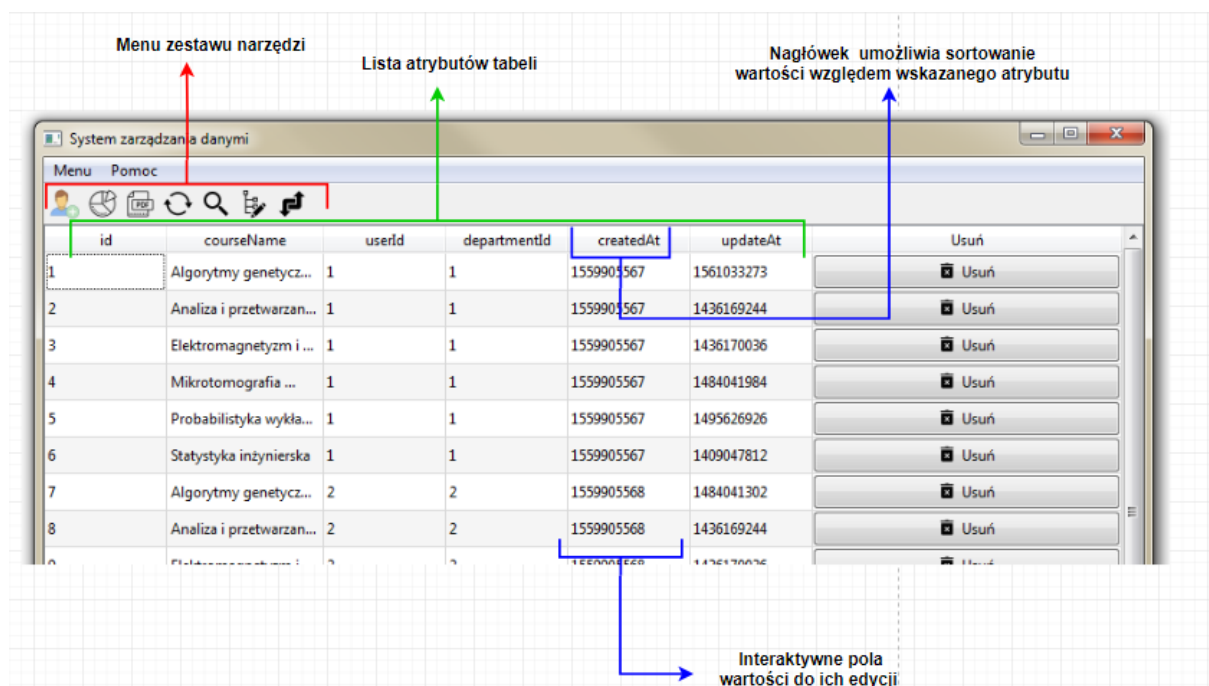


Rys. 1. Panel logowania.



Rys. 2. Okno wyboru tabeli.

Główne okno aplikacji zawiera m.in. menu zestawu narzędzi oraz listę atrybutów wskazanej przez użytkownika tabeli, a także dane, które do nich przynależą. Stworzona aplikacja umożliwia wykonanie sortowania wartości względem dowolnego atrybutu, edycję czy usuwanie danych.



Rys.3. Okno szczegółów wybranej tabeli.

Menu zestawu narzędzi zawiera kilka opcji, jednakże dwoma głównymi funkcjonalnościami aplikacji są:

- Umożliwienie wyświetlania wartości pochodzących z naszej bazy danych w formie wykresów pudełkowych lub słupkowych.
- Umożliwienie podglądu plików *.pdf*, w których to zawarte są komentarze udzielone przez respondentów na poszczególnych ankietach.

The screenshot shows a window titled "Wyświetl dane w formie graficznej". It contains several dropdown menus and buttons. Green arrows point from Polish labels to specific elements in the window:

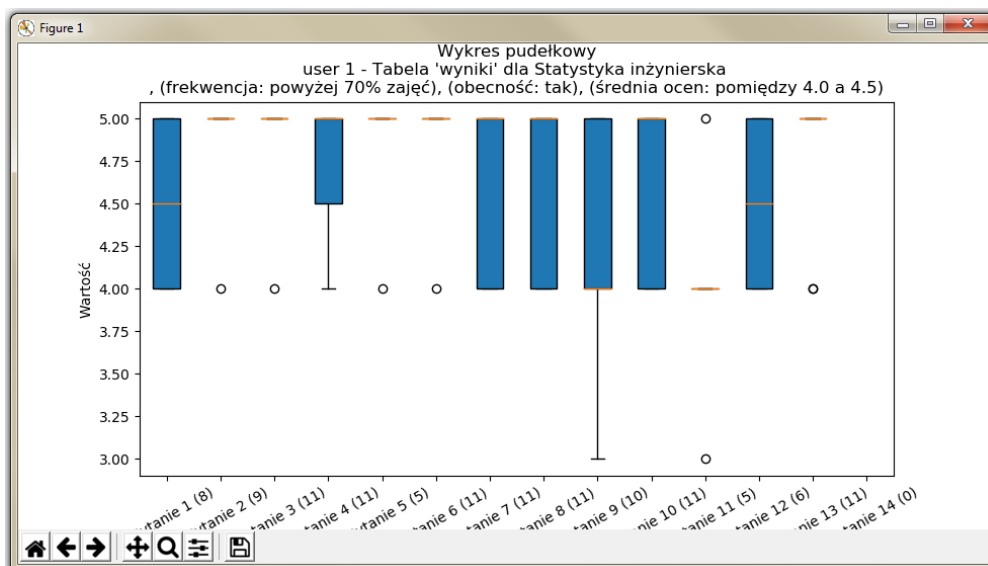
- Pole wyboru użytkownika** points to the "Użytkownik:" dropdown menu.
- Pole wyboru kursu** points to the "Kurs:" dropdown menu.
- Pole wyboru typu danych** points to the "Tabela:" dropdown menu.
- Wybór konkretnych respondentów** points to the "Frekwencja respondentów:" dropdown menu.
- Pole wyboru roku, z którego mają pochodzić dane** points to the "Rok:" dropdown menu.
- Pole wyboru rodzaju wyświetlanych danych** points to the "Rodzaj:" dropdown menu.
- Przyciski akcji** points to the three buttons at the bottom: "Wykres pudełkowy", "Podgląd odpowiedzi P1 - P14", and "Podgląd szczegółów respondentów".

Rys. 4. Okno określania obostrzeń dla wyświetlanych danych liczbowych.

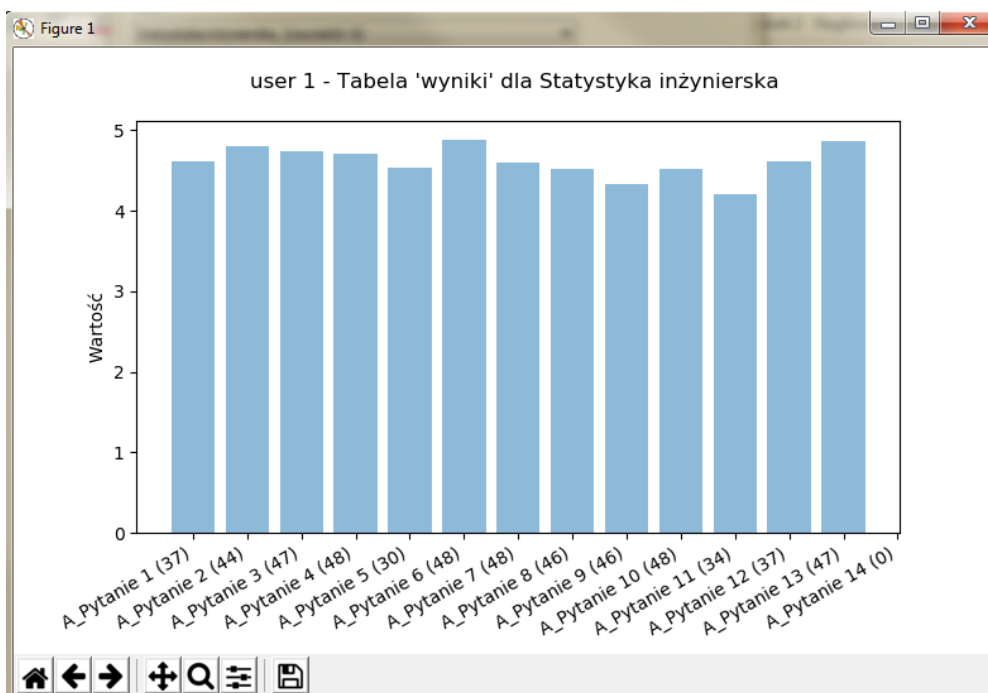
The screenshot shows a window titled "Wyświetl komentarze jako pdf". It contains several dropdown menus and a button. Green arrows point from Polish labels to specific elements in the window:

- Pole wyboru użytkownika** points to the "Użytkownik:" dropdown menu.
- Pole wyboru kursu** points to the "Kurs:" dropdown menu.
- Pole wyboru typu komentarz** points to the "Typ komentarzy:" dropdown menu.
- Przyciski akcji** points to the "Podgląd PDF" button.

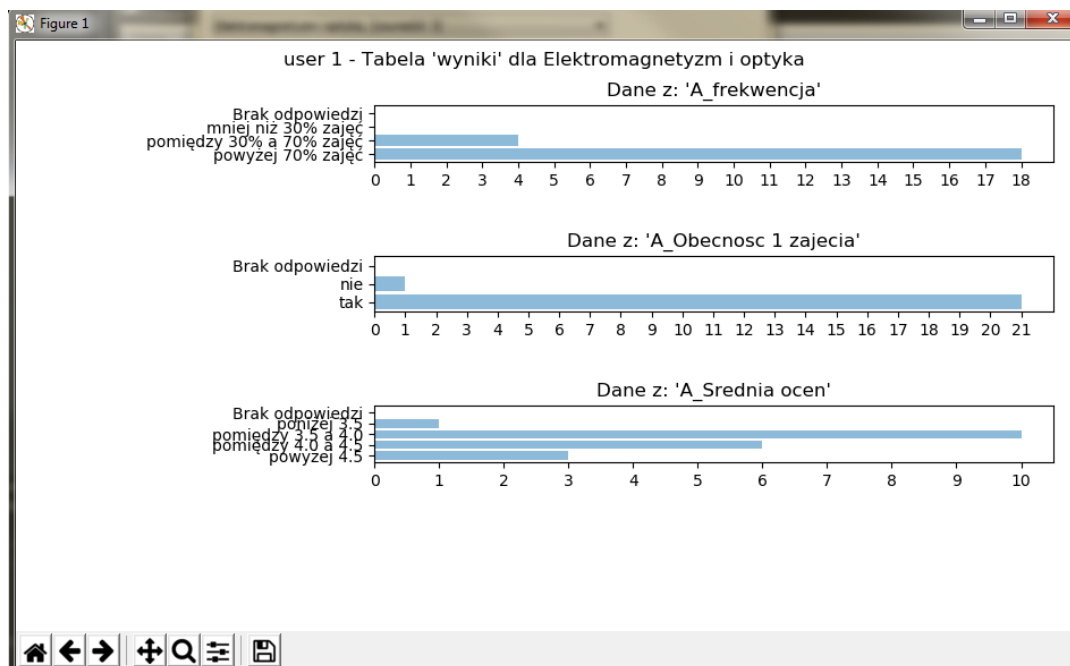
Rys. 5. Okno określania obostrzeń dla wyświetlanych plików .pdf.



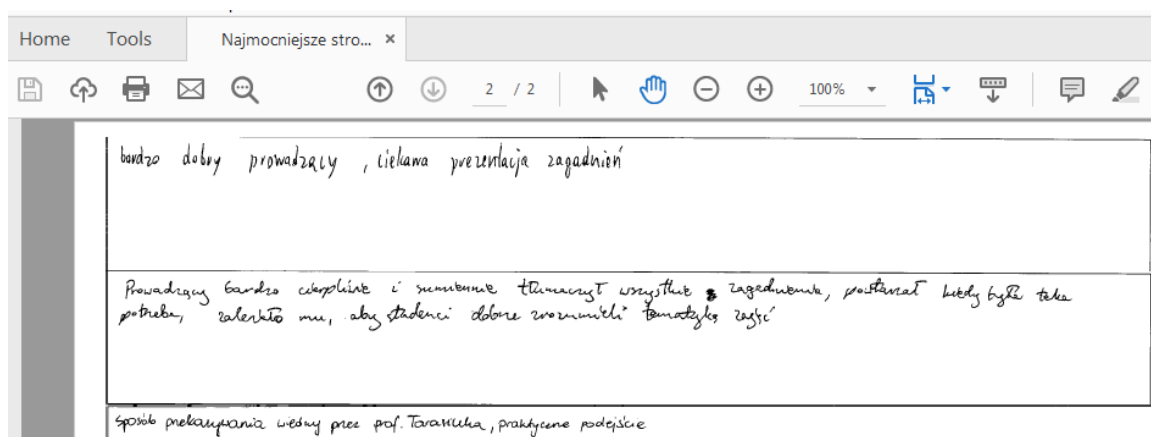
Rys. 6. Wykres pudełkowy przedstawiający wartości pochodzące z bazy danych.



Rys. 7. Wykres słupkowy przedstawiający wartości pochodzące z bazy danych (szczegóły ocen).



Rys. 8. Wykres słupkowy przedstawiający wartości pochodzące z bazy danych (szczegóły respondentów).



Rys. 9. Prezentacja widoku scalonych komentarzy respondentów w utworzonych plikach .pdf.