

# Raport Realizacji Zadań

## 1. Wykaz realizowanych zadań i odniesienie ich do harmonogramu

Celem realizowanego zadania zaliczeniowego było stworzenie aplikacji z Graficznym Interfejsem Użytkownika połączoną z Bazą Danych.

Program został zaimplementowany w języku Python, natomiast sam GUI został zaimplementowany przy pomocy tkinter'a. Aplikacja korzysta z Bazy Danych MySql.

Program posiada następujące funkcjonalności

- 1) Autoryzacja użytkownika
  - a) Logowanie się użytkowników zarejestrowanych
  - b) Rejestracja nowych użytkowników
- 2) Dodawanie oraz modyfikacja uzyskanych wyników z Trójboju Siłowego
- 3) Dodawanie oraz modyfikacja wagi zawodnika
- 4) Statystyka oraz wizualizacja ww. danych

## 2. Napotkane problemy i ich sposób rozwiązania

Największym problemem związanym z implementacją było połączenie aplikacji z Bazą Danych – w tym celu musiałem się dokształcić z zakresu wiedzy połączenie Aplikacja – Baza Danych, by móc przetwarzać dane zapisane w Bazie.

Drugim problemem była walidacja danych podawanych przez użytkownika. W celu rozwiązania tego problemu, trzeba było przemyśleć możliwości jakie może wykonać użytkownik na aplikacji oraz odpowiednio przygotować aplikację na próbę podania nieprawidłowych danych.

### 3. W jakim stopniu realizacja będzie zgodna z projektem

Realizacja posiada wszystkie zaproponowane funkcjonalności, z tym szczególnie, że dane nie mogą być bezpośrednio edytowane, lecz tylko usunięte. Następnie możemy dodać jeszcze raz daną pozycję już poprawnie.

### 4. Opis zrealizowanych funkcji

Program posiada następujące funkcje:

- a) Autoryzacja użytkownika
- b) Dodanie wagi przez zatoryzowanego użytkownika
- c) Dodanie wyników przez zautoryzowanego użytkownika
- d) Wyświetlenie wykresu wagii użytkownika
- e) Wyświetlenie wykresu „Totalu” użytkownika
- f) Wyświetlenie wykresu przysiadu użytkownika
- g) Wyświetlenie wykresu wyciskania użytkownika
- h) Wyświetlenie wykresu martwego ciągu użytkownika
- i) Możliwość usunięcia wagi przez użytkownika
- j) Możliwość usunięcia wyników przez użytkownika
- k) Kilka pomocniczych funkcji jak wyświetlenie pomocy, możliwość wylogowania się czy zamknięcia programu przez użytkownika

## 5. Przewodnik po klasach

### 1) Main\_Window - Lokalizacja: plik Main\_Window.py

Opis: Klasa Main\_Window służy do wyświetlania głównego okna programu – sam układ okna i przycisków znajduje się w dwóch metodach – konstruktorze (`__init__`) oraz `create_widgets()`.

Reszta metod służy do obsługi przycisków.

Metoda `pomoc()` -> Wyświetla informacje o programie

Metoda `wylogujSie()` -> Zamyka okno główne oraz otwiera okno autoryzacji

Metoda `Zamknij()` -> Zamyka aplikację

Metody `dodaj_wage()` oraz `dodaj_wyniki()` -> Zamykają okno główne oraz otwierają okna służące do dodawania wagi lub wyników

Metody `pokazWykresWaga()`, `pokazWykresTotal()` -> Służą do wyświetlania wykresów wagi oraz „totalu” dla danego użytkownika (na podstawie danych dodanych przy pomocy metod `dodaj_wage()` i `dodaj_wyniki()`).

Metody `squatWykres()`, `benchWykres()`, `deadliftWykres()` -> Wyświetlają wykresy dla danego boju.

Metody `deleteWeight()` oraz `deleteResults()` -> Otwierają okna służące do usunięcia wagi lub wyników.

### 2) Register – Lokalizacja: plik Register.py

Opis: Klasa Register służy do wyświetlenia okna rejestracji dla użytkownika a następnie utworzenie (lub nie) nowego konta. Układ okna i przycisków znajdują się w konstruktorze (`__init__`) oraz metodzie `create_widgets()`.

Pozostałe metody:

Metoda `powrot()` -> Cofa użytkownika do okna logowania

Metoda `rejestracja()` -> Metoda służy do wczytania danych przez użytkownika, następnie w przypadku prawidłowych danych zostaje utworzone nowe konto, oraz wyświetla się Okno Główne programu, w

przypadku nieprawidłowych danych użytkownik zostanie o tym poinformowany odpowiednim komunikatem.

3) StartWindow – Lokalizacja: plik StartWindow.py

Opis: Klasa ta służy do otworzenia pierwszego okna aplikacji, gdzie możemy podejrzeć opis programu lub przejść do okna autoryzacji.

Metody tworzące wygląd okna: konstruktor oraz `create_widgets()`.

Pozostałe metody:

Metoda `start_()` -> Metoda zapewnia przejście do okna autoryzacji po kliknięciu przycisku „START”

Metoda `pomoc_()` -> Wyświetla okno z informacjami dotyczącymi aplikacji.

4) User – Lokalizacja: plik User.py

Opis: Klasa służąca do przechowywania informacji na temat zalogowanego użytkownika jak jego id, login, hasło, imię, nazwisko, wzrost.

W klasie znajdziemy podstawowe gettery jak `get_id()` czy `get_login()` oraz dwie metody `save_to_database()` oraz statyczna metoda `get_account()`.

Pierwsza z nich służy do rejestracji użytkownika, gdzie zapisuje dane dotyczące nowego konta do bazy danych.

Druga pozwala na pobranie z bazy danych (o ile istnieje) informacji dotyczących użytkownika o podanym loginie i hasle (podawanym w oknie autoryzacji).

5) Add\_Waga – Lokalizacja: plik Add\_Waga.py

Opis: Klasa służąca do interakcji z użytkownikiem – wczytania wagi z określoną datą. Wygląd okna ponownie znajdują się w konstruktorze oraz metodzie `create_widgets()`.

Klasa ta zawiera dwie metody oprócz ww.

Pierwszą z nich jest metoda `wyslij()` -> Służy do walidacji danych podanych przez użytkownika oraz w przypadku poprawności zapisania informacji o wadze i dacie w Bazie Danych. W przypadku nieprawidłowości użytkownik zostanie poinformowany stosownym oknem.

Drugą metodą jest `cofnij()` -> Służy do powrotu użytkownika do okna głównego.

6) Add\_Wyniki – Lokalizacja: plik Add\_Wyniki.py

Opis: Klasa służąca do interakcji z użytkownikiem – wczytania wyników z określoną datą. Wygląd okna ponownie znajdują się w konstruktorze oraz metodzie create\_widgets().

Klasa ta zawiera dwie metody oprócz ww.

Pierwszą z nich jest metoda wyslij() -> Służy do walidacji danych podanych przez użytkownika oraz w przypadku poprawności zapisania informacji o wynikach i dacie w Bazie Danych. W przypadku nieprawidłowości użytkownik zostanie poinformowany stosownym oknem.

Drugą metodą jest cofnij() -> Służy do powrotu użytkownika do okna głównego.

7) Delete\_Results – Lokalizacja: plik Delete\_Results.py

Opis: Klasa służąca do usunięcia danego wyniku z Bazy Danych, użytkownik może wybrać z listy który rekord ma zostać usunięty.

Wygląd i układ okna znajdują się w konstruktorze i metodzie create\_widgets().

Pozostałe metody:

Metoda complete\_the\_list() -> Metoda pobierające wszystkie wyniki dla danego użytkownika z Bazy Danych a następnie wyświetlenie ich w liście.

Metoda Powrot() -> Służy do powrotu do okna głównego.

Metoda Usun() -> Usuwa z Bazy Danych wybrany rekord

8) Delete\_Weight – Lokalizacja: plik Delete\_Weight.py

Opis: Klasa służąca do usunięcia danej wagi z Bazy Danych, użytkownik może wybrać z listy który rekord ma zostać usunięty.

Wygląd i układ okna znajdują się w konstruktorze i metodzie create\_widgets().

Pozostałe metody:

Metoda complete\_the\_list() -> Metoda pobierające wszystkie rekordy dla danego użytkownika z Bazy Danych a następnie wyświetlenie ich w liście.

Metoda Powrot() -> Służy do powrotu do okna głównego.

Metoda Usun() -> Usuwa z Bazy Danych wybrany rekord

9) Plik database.py

Opis: Jest to plik w którym znajdują się wszystkie operacje na Bazie Danych jak pobranie wszystkich rekordów dotyczących wyników dla danego użytkownika, czy usunięcie danego rekordu.

10) Plik Wykresy.py

Opis: W pliku znajduje się metoda generująca wykres na podstawie przesłanych danych.

11) Plik MAIN.py

Opis: W pliku MAIN znajduje się funkcja main() służąca do włączenia programu.