

Projekt NCBiR POIR.01.01.01-00-1059/20

Raport nr. 19/2022

Opis oprogramowania zarządzania funkcjonowaniem Urządzenia Pomiarowo-Diagnostycznego do oceny stanu mięśnia.

Etap 1 Faza 2

Autorzy:

Wojciech Maj

Piotr Łach

**Streszczenie:** W raporcie zaprezentowano wyniki prac nad aplikacją dla Komputera Diagnosty obsługującego procedurę pomiarową i diagnostykę badanego mięśnia. . Aplikacja jest narzędziem, które ma za zadanie umożliwić zarządzaniem funkcjonowaniem Urządzenia Pomiarowo-Diagnostycznego a w szczególności : wizualizować sygnały otrzymywane ze Stacji Bazowej oraz wspomagać diagnozowanie stanu mięśnia. Raport zawiera przedstawienie w formie tekstowej a także wizualnej kolejne etapy prac nad stworzeniem aplikacji według przyjętej w projekcie koncepcji pomiarów i diagnostyki.

Sprawdzenie: W. Klembowski (Kierownik B+R)

Akceptacja : D. Gonciarz (Przedstawiciel Zarządu)

Wrocław Styczeń 2022

Materiał stanowi własność Active Life Europe Sp. z o.o. Sp.k.

**Spis Treści.**

1. **Wprowadzenie.**
2. **Opis Aplikacji**
3. **Opis funkcji aplikacji za pomocą UML**
4. **Stos technologiczny**
5. **Przedstawienie wyglądu aplikacji**
6. **Podsumowanie**
7. **Wykaz literatury**

**1. Wprowadzenie**

Przedmiotem projektu POIR.01.01.01-00-1059/20 „Opracowanie nowego zaawansowanego urządzenia pomiarowo – diagnostycznego do akwizycji sygnałów biologicznych podczas różnych rodzajów treningu wysiłkowego oraz analizy i klasyfikacji stanu ćwiczącego na podstawie zebranych informacji” [[[1]](#footnote-2)], jest skonstruowanie prototypu urządzenia diagnostycznego do pomiaru oraz analizy najważniejszych paramentów mięśniowych. Problem badawczy został zdefiniowany w obszarze biomechanicznych i fizjologicznych zależności pomiędzy wybranymi parametrami mięśniowymi, a optymalizacją i bezpieczeństwem procesu treningowego oraz rehabilitacji. Rezultat projektu skierowany będzie m.in. do trenerów, fizjoterapeutów, a także osób, które amatorsko uprawiają sport oraz osób, które przechodzą proces rehabilitacji.

Wynikiem projektu będzie prototyp wielomodułowego (3 moduły: pletysmograf, EMG, MMG, lub inne sygnały biomedyczne) urządzenia pomiarowo - diagnostycznego do rejestracji i analizy parametrów mięśniowych. Pozwoli ono na bezinwazyjny pomiar reakcji mięśni na wysiłek, utrzymanie efektu terapii, śledzenie krzywej oscylacyjnej dotyczącej zmęczenia lub uszkodzenia mięśnia.

Wielomodułowe urządzenie diagnostyczno-pomiarowe umożliwi bezinwazyjny, precyzyjny pomiar najważniejszych parametrów treningowych mięśni :

- sztywność mięśnia,

- napięcie mięśniowe,

- mikrokrążenie w naczyniach obwodowych,

-aktywność elektryczna mięśni.

Rezultat projektu pozwoli na długotrwały zapis oraz analizę ww. parametrów, pod kątem rejestracji i modyfikacji obciążeń treningowych, przewidywania ryzyka urazu, progresu rehabilitacji oraz stanu biomechanicznego mięśniowych jednostek motorycznych.

Minimalna specyfikacja urządzenia zdolnego do korzystania z aplikacji:

System operacyjny: Windows 10/11 64 bit, Linux   
Procesor: Intel Core i3 – 8140 , 2.1 GHz / AMD Ryzen 3100 , 3.6 GHz  
Karta graficzna: Intel UHD Graphics / AMD Radeon™ Graphics   
Pamięć RAM: 8GB  
Miejsce na dysku twardym: 4GB HDD/SSD  
Łączność: WiFi 4/5/LAN 1 Gb/s  
Złącze: RJ-45 LAN (wymagane gdy brak WiFi)  
Inne: klawiatura, mysz/touchpad

W niniejszym raporcie zaprezentowano wizualizację możliwych trybów pomiarowych pod kątem użytkowników oraz zmiennych parametrów, które będą ewoluowano w dalszej fazie badań etapu 1 i 2.

**2. Opis aplikacji**

Jednym z produktów końcowych projektu POIR.01.01.01-00-1059/20 „Opracowanie nowego zaawansowanego urządzenia pomiarowo – diagnostycznego do akwizycji sygnałów biologicznych podczas różnych rodzajów treningu wysiłkowego oraz analizy i klasyfikacji stanu ćwiczącego na podstawie zebranych informacji” [[[2]](#footnote-3)], jest aplikacja dla komputera diagnosty, której głównymi zadaniami są:

1. Zarządzanie kontami użytkowników

* Logowanie
* Rejestracja
* Modyfikacja danych personalnych

1. Prowadzenie pomiarów diagnostycznych

* Tryb Auto
* Tryb Manual

1. Pomoc

* Instrukcja obsługi
* Procedury pomiarowe

Ad 1. Zarządzanie kontami użytkowników

Każdy użytkownik aplikacji będzie mógł się zarejestrować, a następnie logować. Dane w postaci hasła są zaszyfrowane gdzie jednym z priorytetów aplikacji jest bezpieczeństwo. Założenia zakładają dwa rodzaje kont użytkowników podstawowy i wykwalifikowany , różnią się przede wszystkim możliwościami prowadzenia pomiarów. Użytkownik wykwalifikowany może prowadzić badania kilku innym użytkownikom o charakterze podstawowym , a także dostęp do ich historii pomiarowej. Co użytkownik podstawowy będzie miał ograniczenie do prowadzenia badań i dostęp do historii pomiarowej swojej. Każdy użytkownik będzie miał możliwość modyfikacji swoich danych personalnych.

Ad 2. Prowadzenie pomiarów diagnostycznych

W założeniach aplikacji przewidziano dwa tryby prowadzenia badań, gdzie są bezpośrednio powiązane z rodzajem konta użytkownika tzw. podstawowy i wykwalifikowany , tak jak opisano powyżej, tryb Auto jest przede wszystkim przeznaczony dla użytkowników, który chce prowadzić badania wyłącznie dla siebie. A następny tryb pomiarowy dla tzw. trenerów, którzy maja pod sobą wielu użytkowników, którym mogą prowadzić badania a także wgląd do ich historii pomiarowej. Ostatnią różnica pomiędzy tymi trybami jest ilość używanych taśm pomiarowych podczas jednej sesji treningowej. Dla pierwszego trybu to maksymalnie 2 taśmy – mięśni. Następny tryb pomiarowy ma ograniczenie do 20 taśm – mięśni.

Ad 3. Pomoc

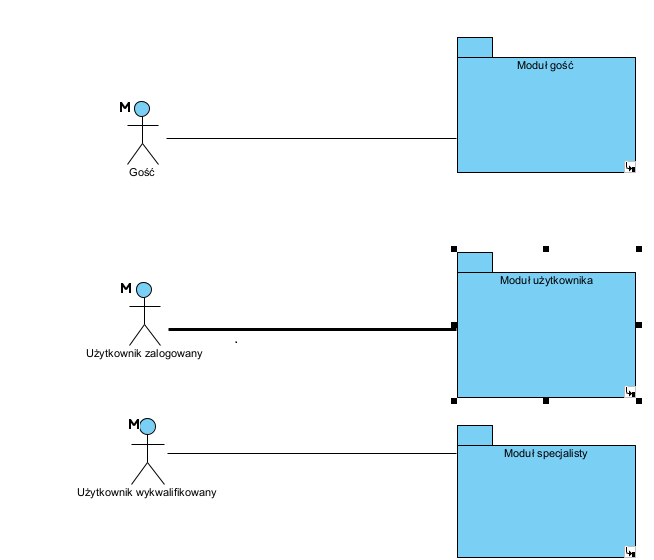
Pomoc jest przede wszystkim skierowana do nowych użytkowników, którzy nie są zapoznani z użytkowaniem aplikacji. Dzięki dostępowi do pomocy pomoże im dużo szybciej się odnaleźć w nowej aplikacji a także zrozumieć jej pewne aspekty.

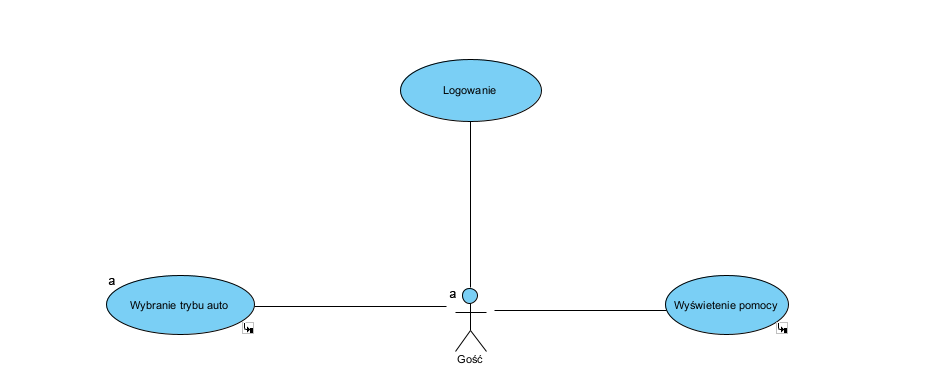
**3. Opis funkcji aplikacji za pomocą UML**

W rama projektowania aplikacji, przygotowano w formie wizualnej funkcjonalność oprogramowania za pomocą diagramu UML. Taka forma jest czytelna dla osób ściśle niezaznajomionych z projektem a także nie biorących udziału w procesie wytwarzania oprogramowania. Warto także zaznaczyć że diagram UML jest swoistym planem implementacji poszczególnych funkcjonalności.

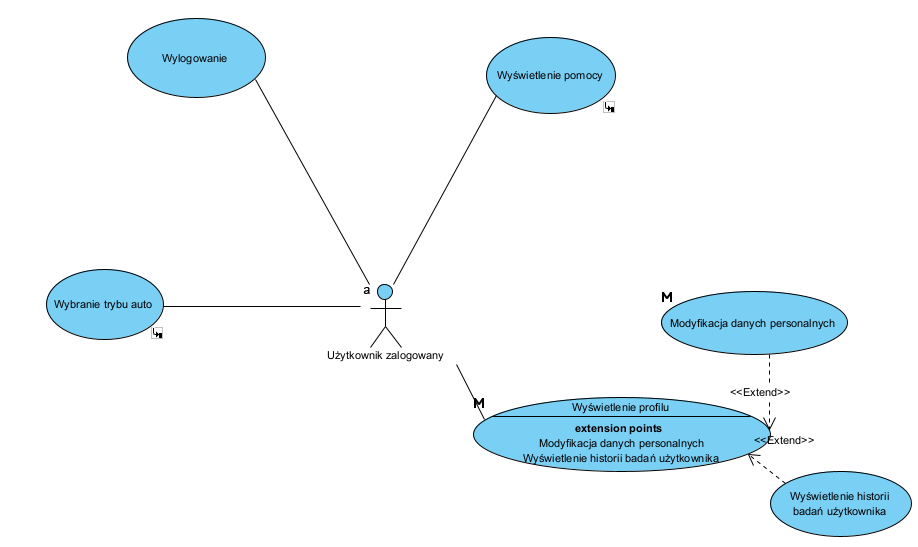
**3.1 Moduł użytkowników**

Założenia aplikacji zakładają trzy typy użytkowników, z poszczególnymi funkcjonalnościami

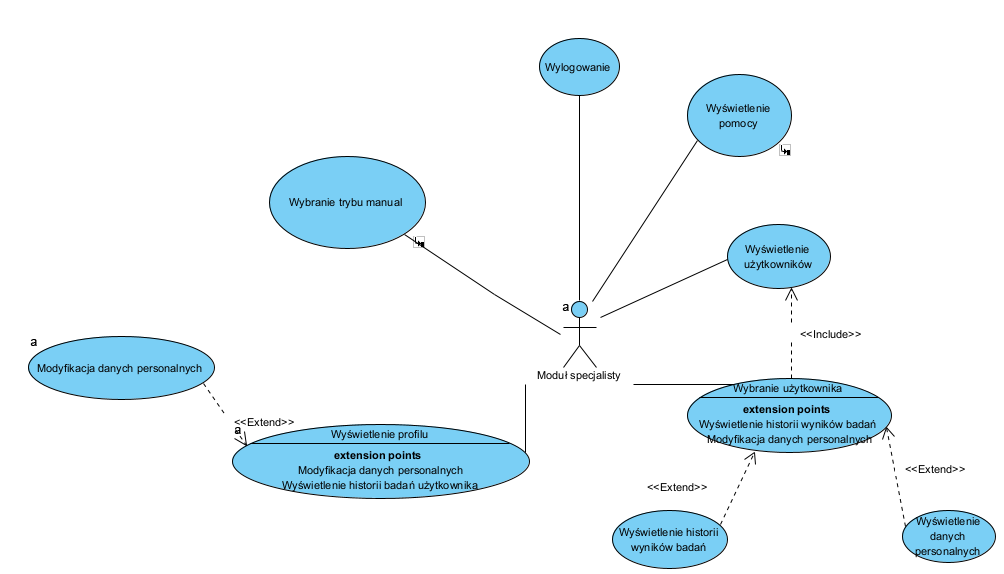
Rysunek 1: Przedstawia wszytskie typy modułów użytkownika dostępne w aplikacji

Rysunek 2: Przedstawienie modułu gościa tzw. możliwości użytkowania aplikacji z tego poziomu

* **Moduł gościa**
* **Moduł Użytkownika (Podstawowego)**

Rysunek 3: Przedstawienie modułu użytkownika podstawowego tzw. możliwości użytkowania aplikacji z tego poziomu

* **Moduł Użytkownika wykwalifikowanego**

Rysunek 4: Przedstawienie modułu użytkownika wykwalifikowanego tzw. możliwości użytkowania aplikacji z tego poziomu

**3.2 Tryby pomiarowe**

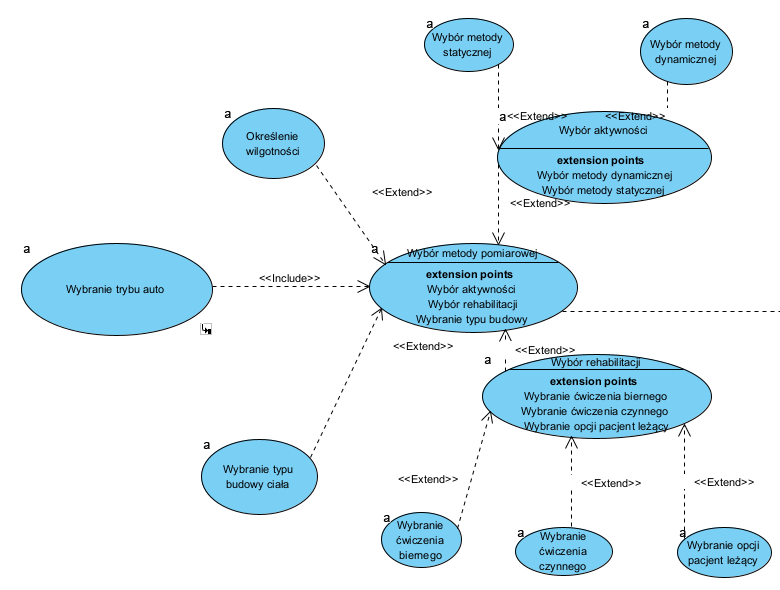
Według postawionych założeń aplikacji dostępne są dwa tryby prowadzenia pomiarów:

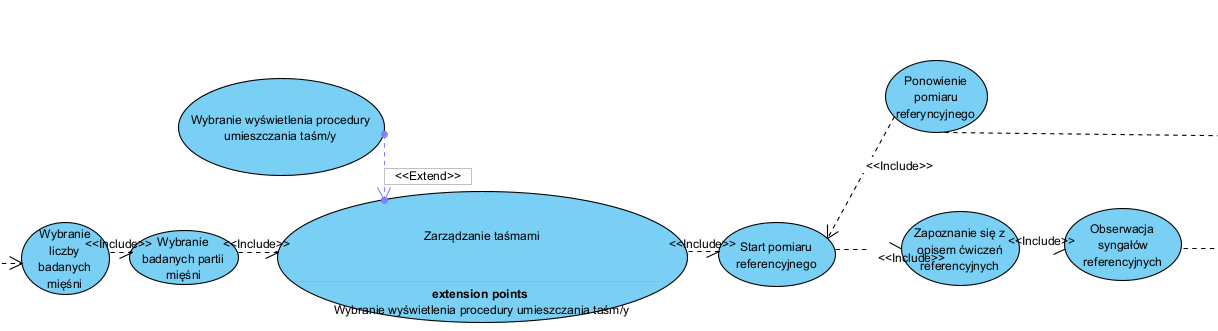
* Auto
* Manual

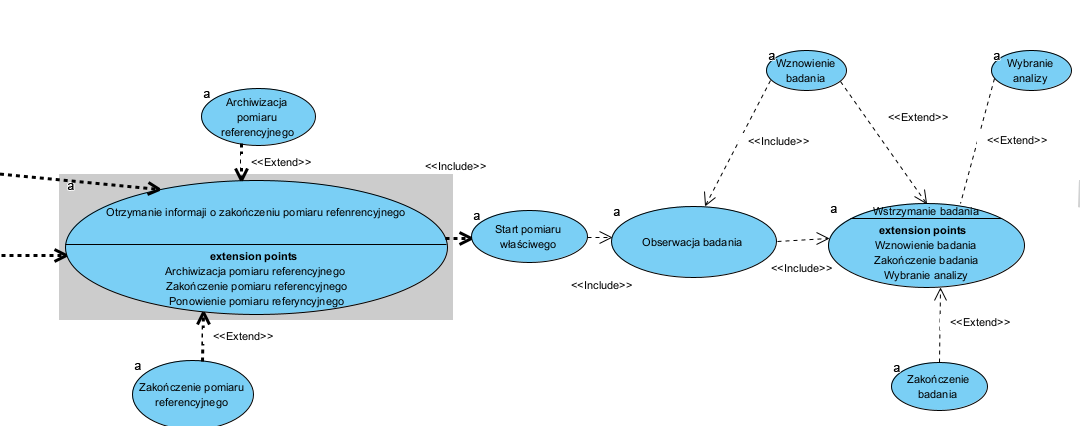
Pierwszy z tych trybów jest dostępny wyłącznie dla modułu gościa i użytkownika podstawowego.

Tryb Manual jest dostępny jest tylko dla modułu użytkownika wykwalifikowanego.

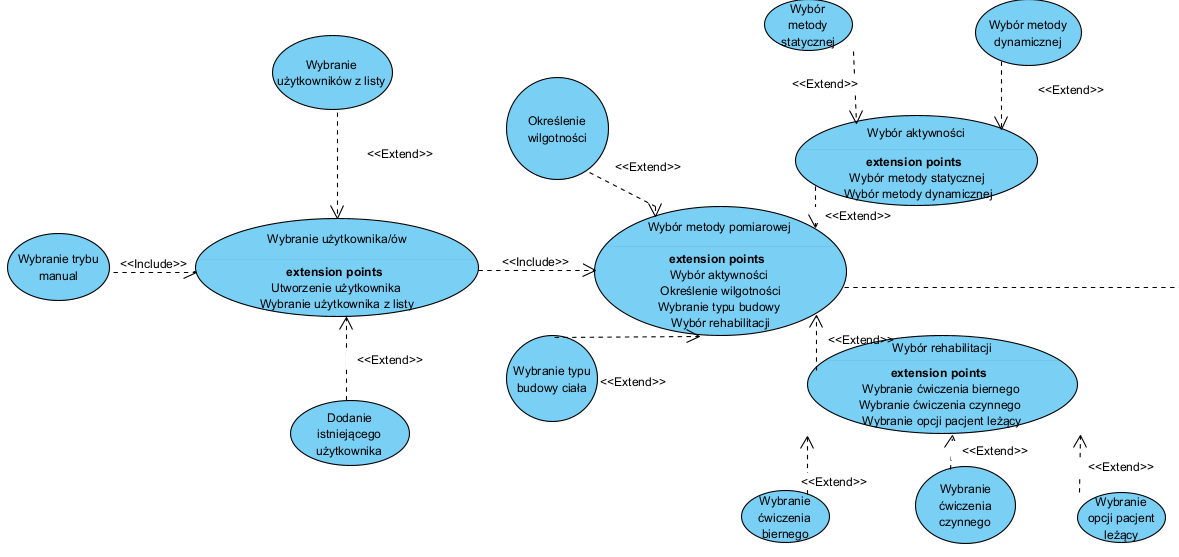
* **Tryb Auto**

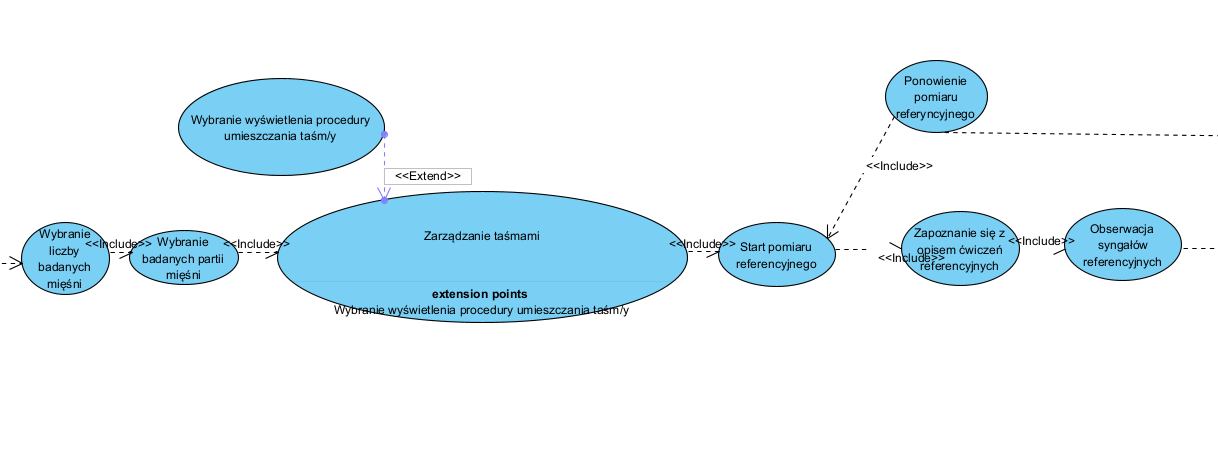
Rysunek 5: Przedstawienie trybu Auto jako przypadki użycia cz.1

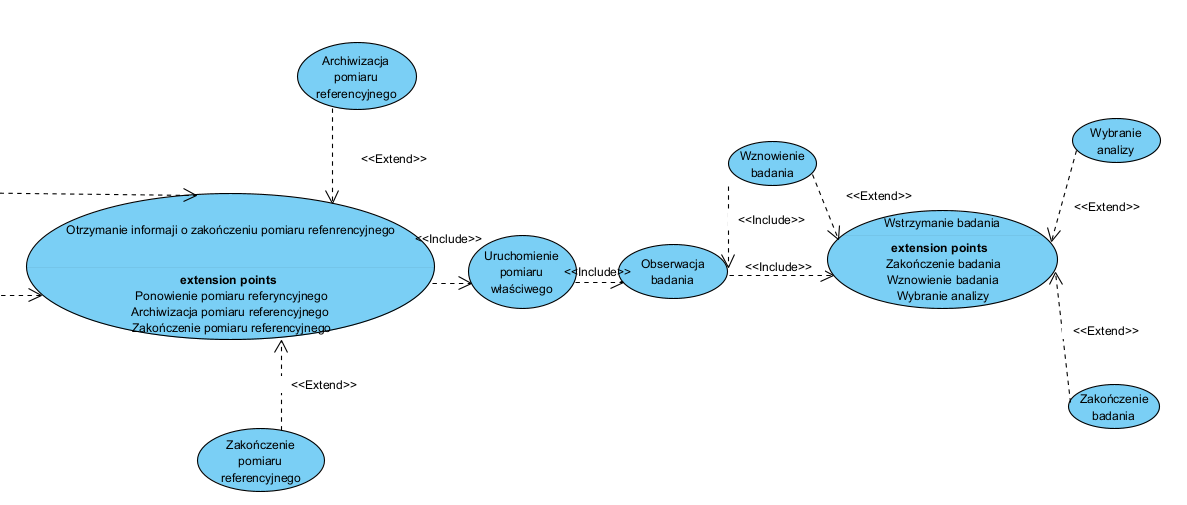
Rysunek 6: Przedstawienie trybu Auto jako przypadki użycia cz.2

Rysunek 7: Przedstawienie trybu Auto jako przypadki użycia cz.3

* **Tryb Manul**

Rysunek 8: Przedstawienie trybu Manual jako przypadki użycia cz.1

Rysunek 9: Przedstawienie trybu Manual jako przypadki użycia cz.2

Rysunek 10: Przedstawienie trybu Manual jako przypadki użycia cz.3

**4. Stos technologiczny**

Podstawowe elementy stosu technologicznego:

* Język programowania Python 3.9
* ProtoBuf
* gRPC
* Kivy

Wykorzystywane moduły python:

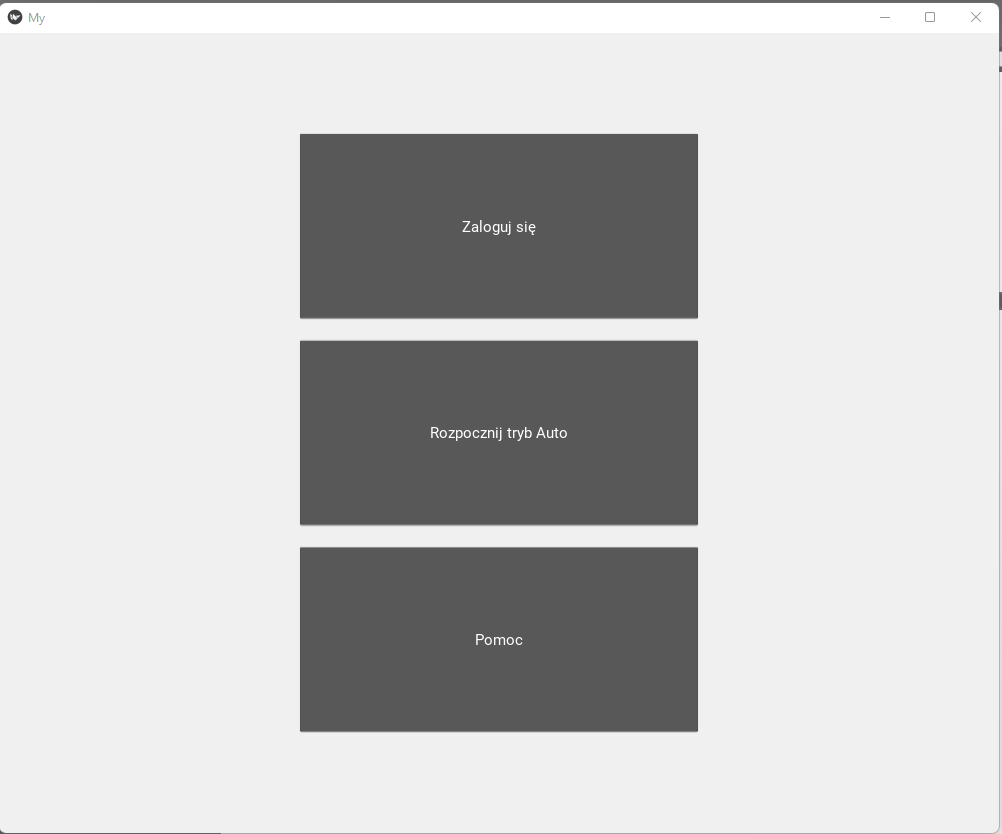
* google-api-python-client
* grpcio
* grpcio-tools
* bcrypt
* Kivy
* kivy-garden.graph
* serial pyserial
* pytest

Domyślnym językiem programowania był Python jednakże w wersji 3.9. Doinstalowane zostały wyżej wymienione moduły. ProtoBuf narzędzie do serializacji danych stworzone przez Googla. W aplikacji jest wykorzystywany w komunikacji z Modułem Akwizycji Danych. gRPC framework służący do komunikacji z wieloma urządzeniami w danej sieci, bazuje na ProtoBufie. Serial – służy do obsługi komunikacji z wykorzystaniem portu UART. Protos – wykorzystywany do tworzenia plików z rozszerzeniem py, służącymi do komunikacji z wykorzystaniem ProtoBufa, narzędzia do komunikacji między Stacją Bazową, a Modułem Akwizycji Danych. gRPCIO oraz gRPCIO-tools – narzędzia do tworzenia odpowiednich plików z rozszerzeniem py, z plików proto, w celu uzyskania narzędzi do komunikacji między Komputerem Diagnosty, a Stacją Bazową.

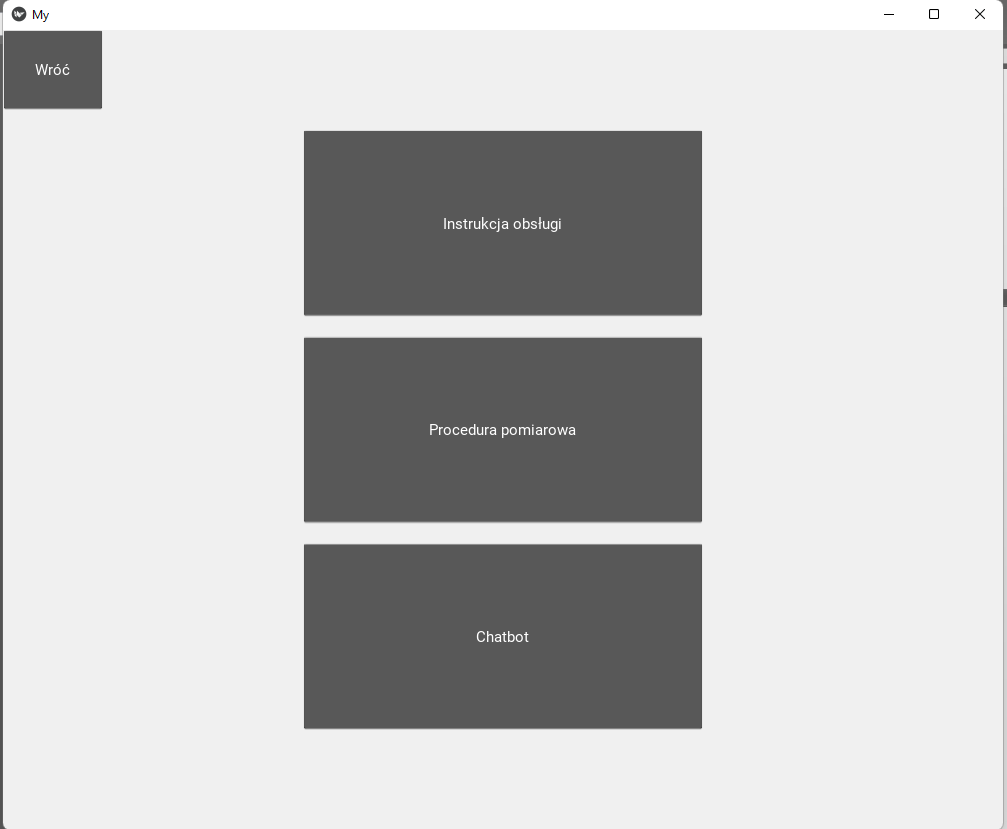
Kivy – narzędzie które jest wykorzystywane do projektowania widoków aplikacji za pomocą plików z rozszerzeniem kv. kivy-garden.graph – jest to dodatkowe narzędzie do frameworka Kivy, wykorzystywany do przedstawiania wykresów w formie graficznej , aby użytkownik miał dużo lepszy wgląd do pozyskiwanych danych.

**5. Przedstawienie wyglądu aplikacji**

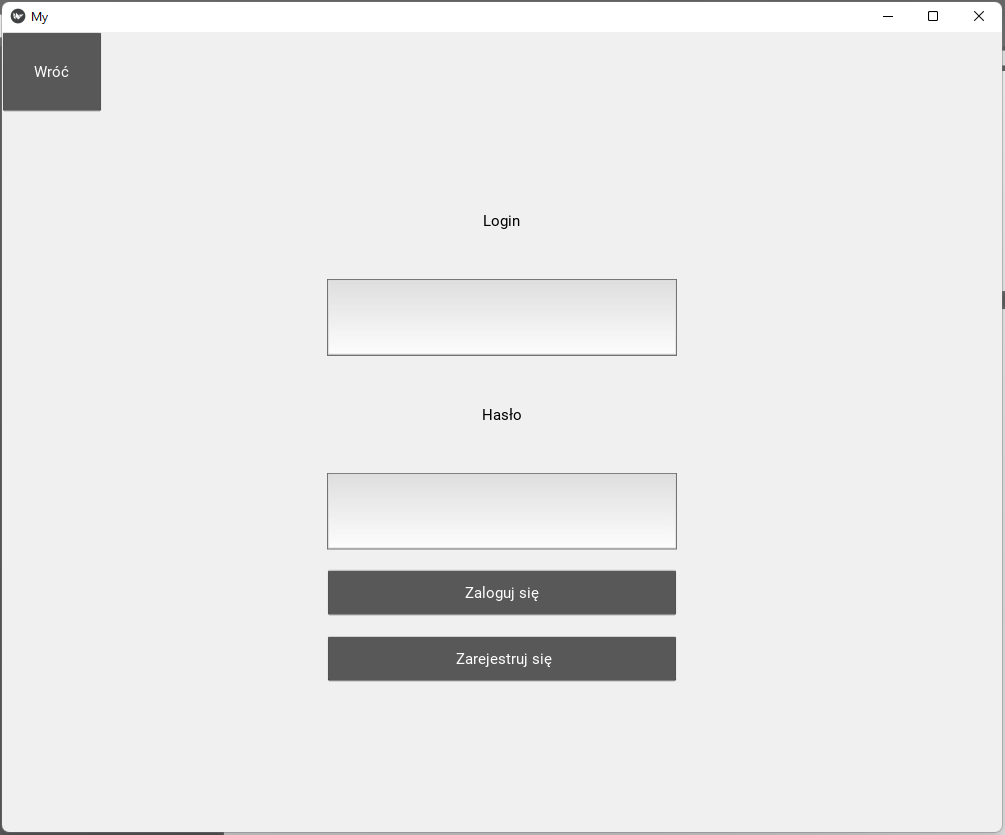
Niniejszy rozdział jest poświęcony przedstawieniem efektu pracy nad aplikacją, poniżej zaprezentowano zrzuty ekranu okna aplikacji.

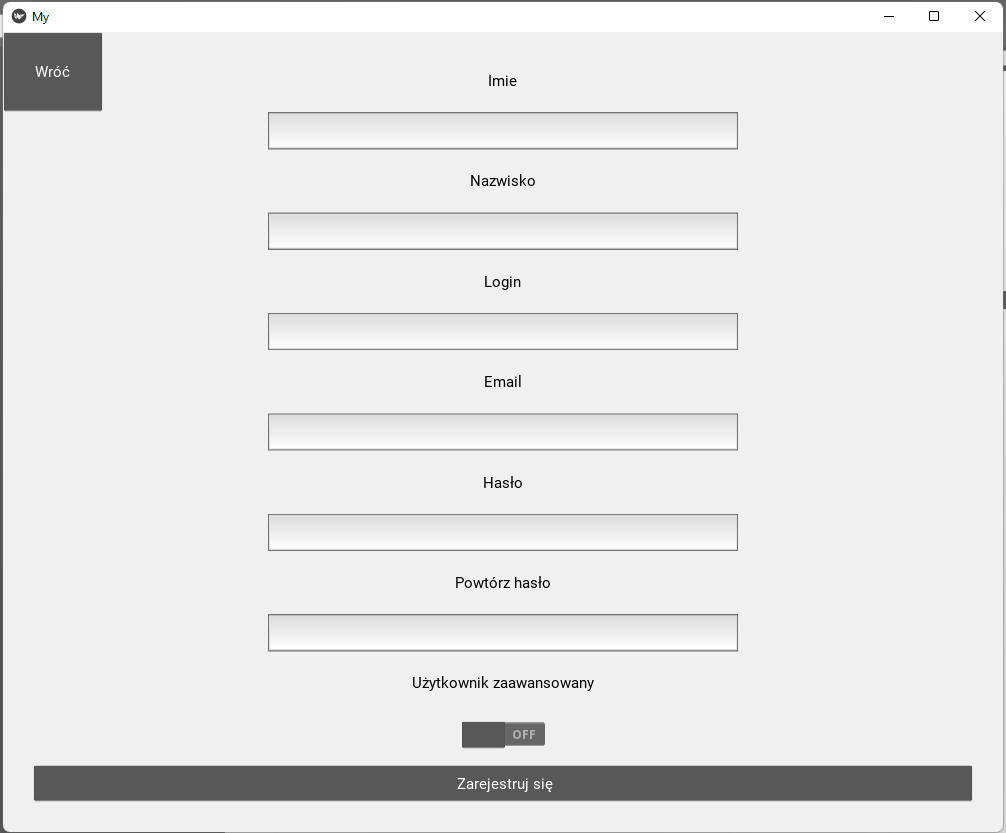


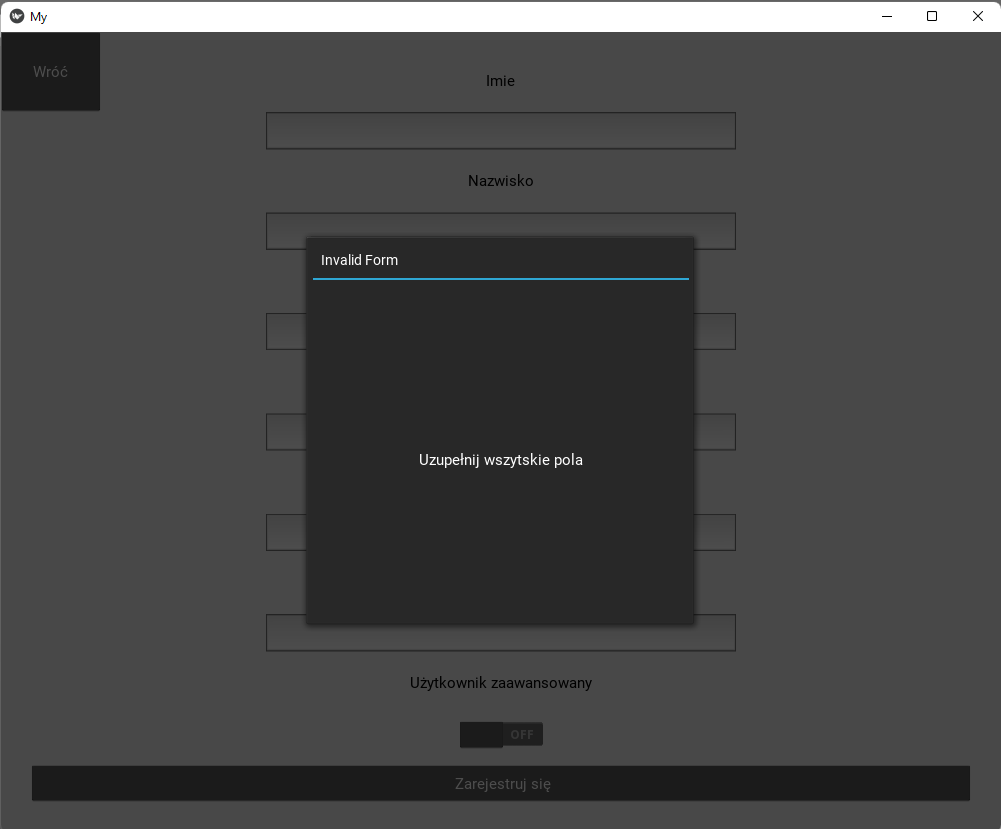
Rysunek 11: Przedstawienie wyglądu aplikacji w trybie gość

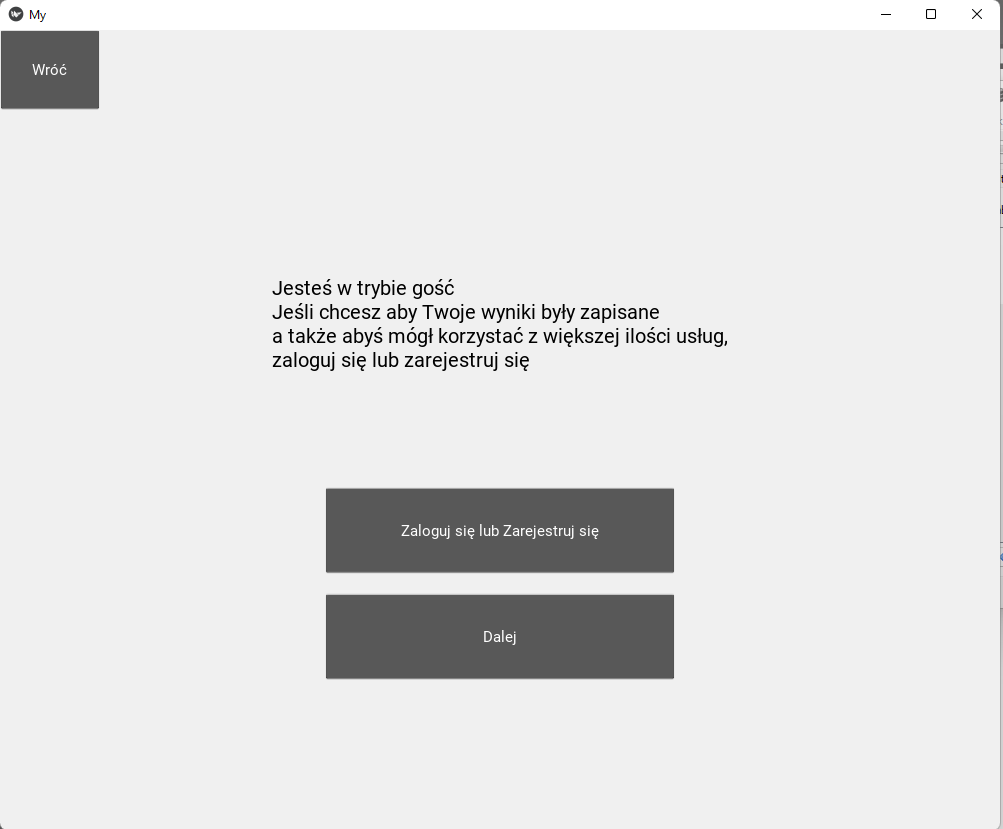


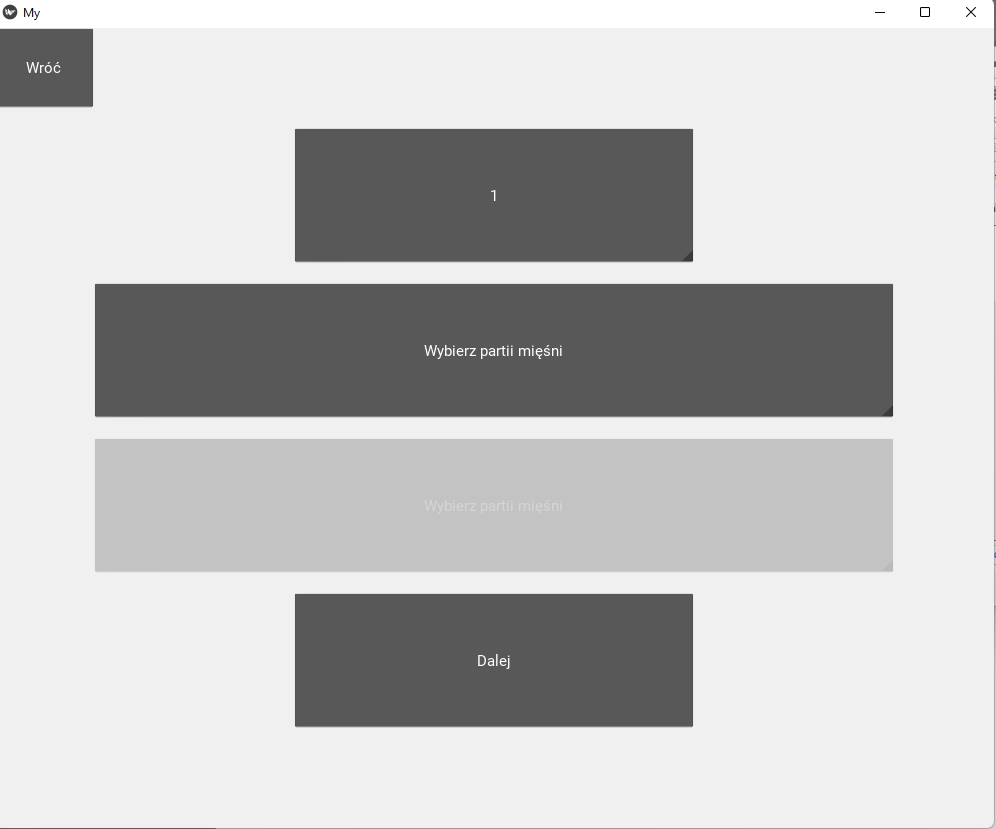
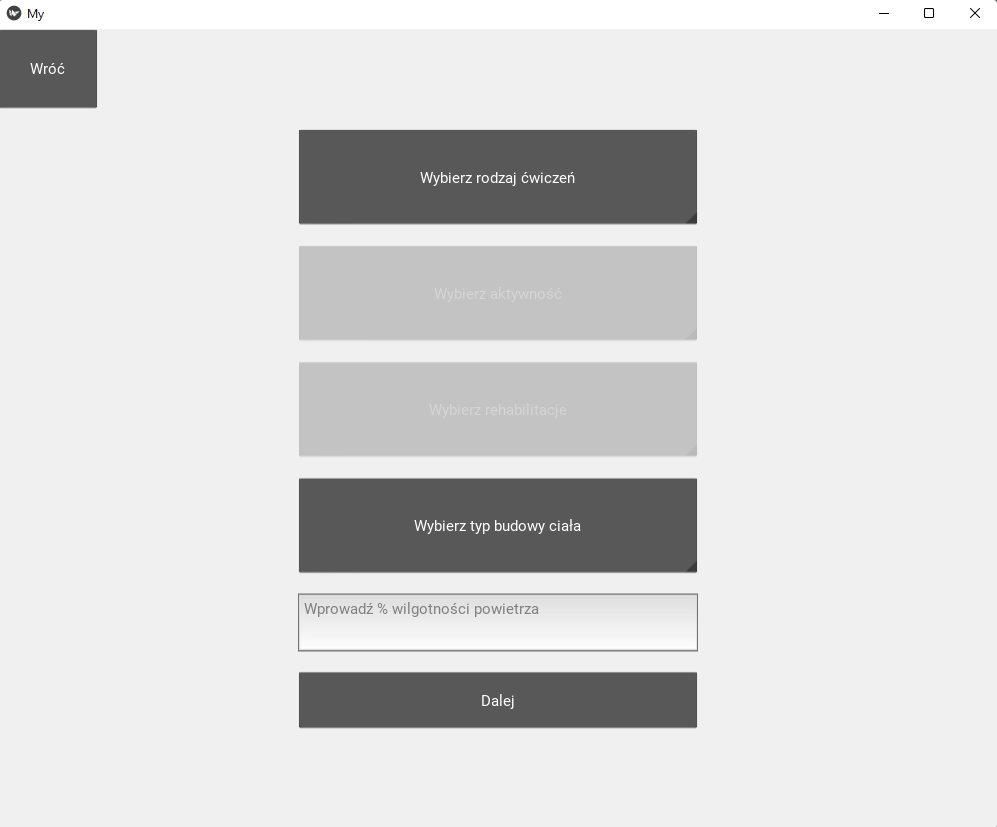
Rysunek 12: Przedstawienie modułu pomocy w aplikacji

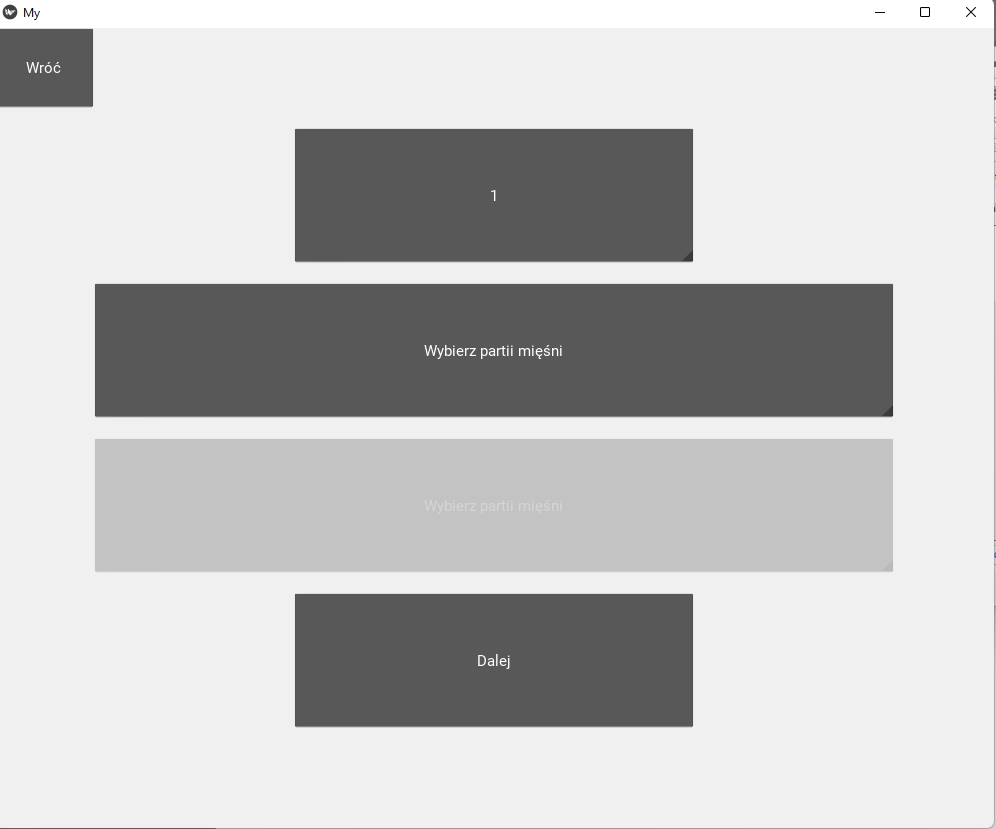
Rysunek 13: Przedstawienie widoku logowania w aplikacji

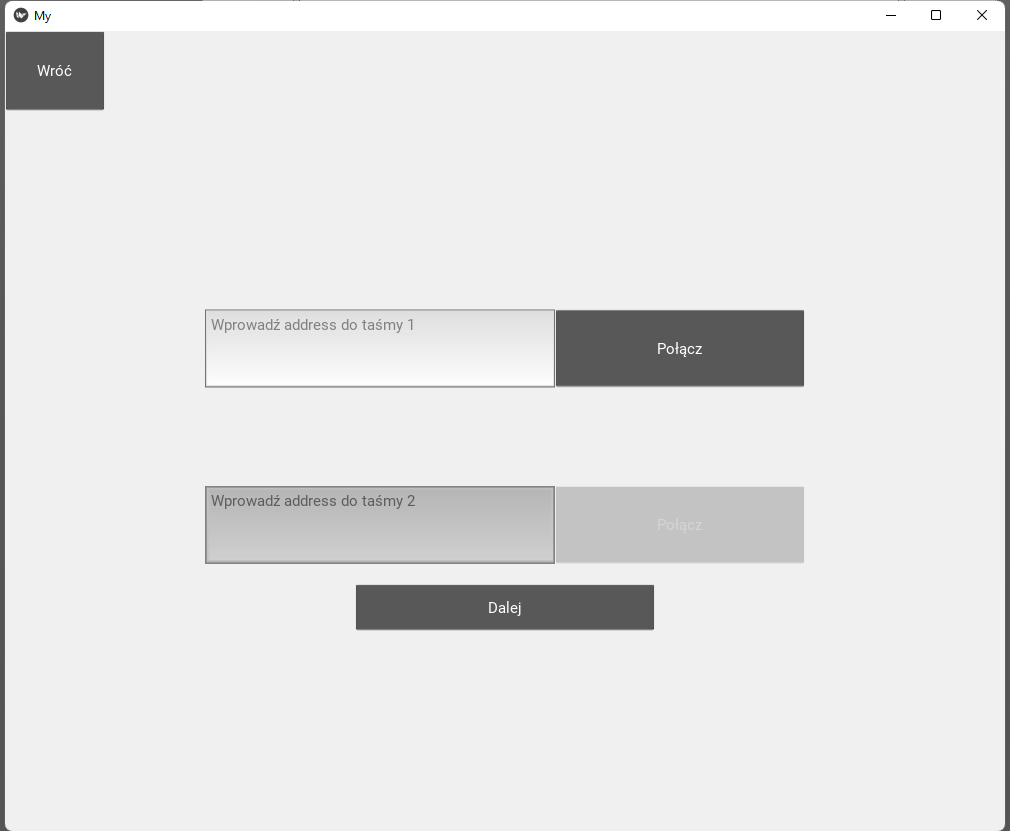
Rysunek 14: Przedstawienie okna rejestracji nowych użytkowników w aplikacji

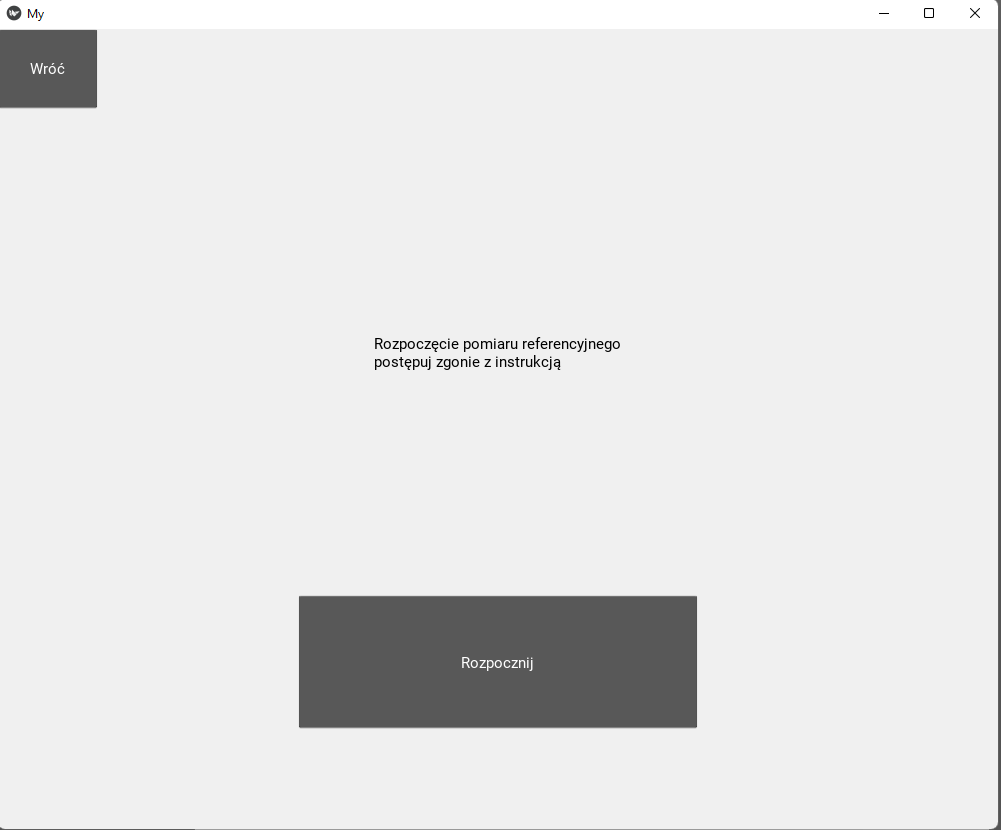
Rysunek 15: Przedstawienie okna powiadamiającego użytkownika o błędnym uzupełnieniu pul w formularzu

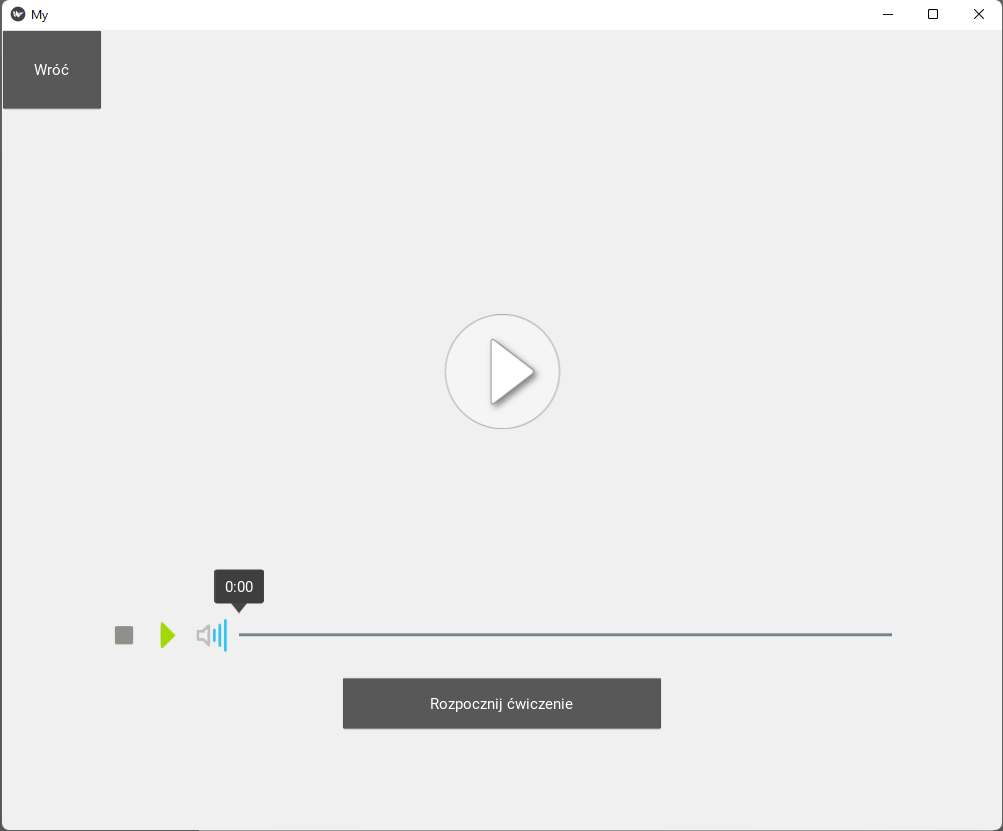
Rysunek 16: Przedstawienie okna rozpoczynającego tryb pomiarowy Auto w aplikacji

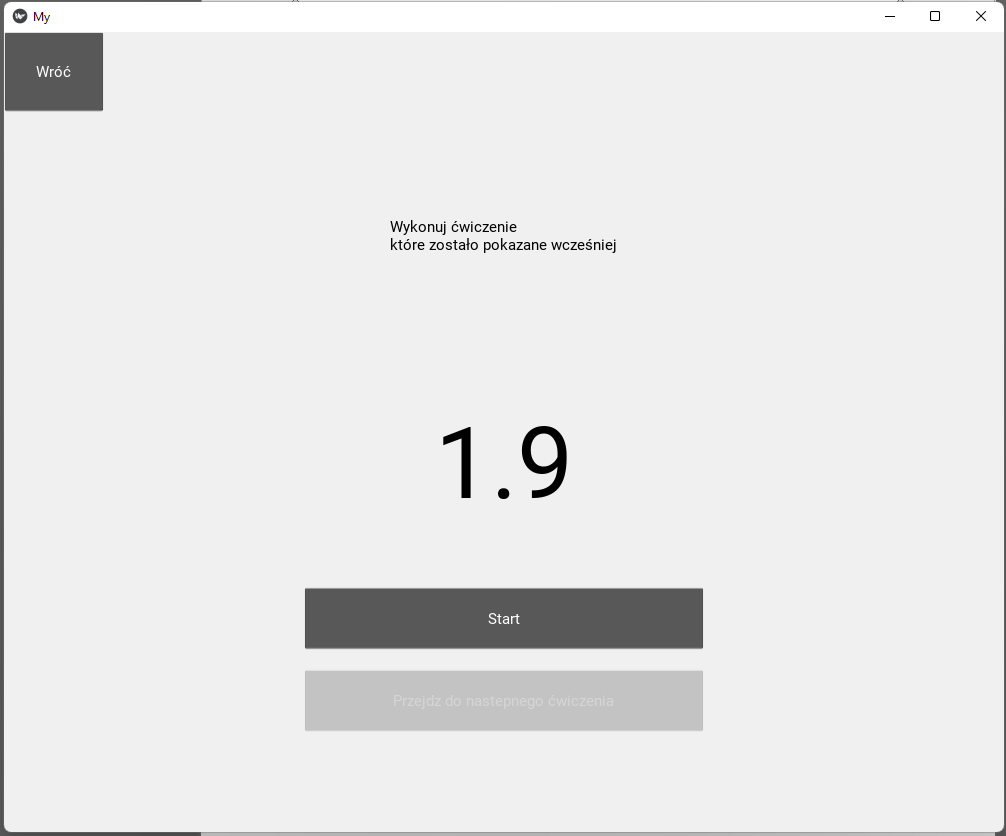


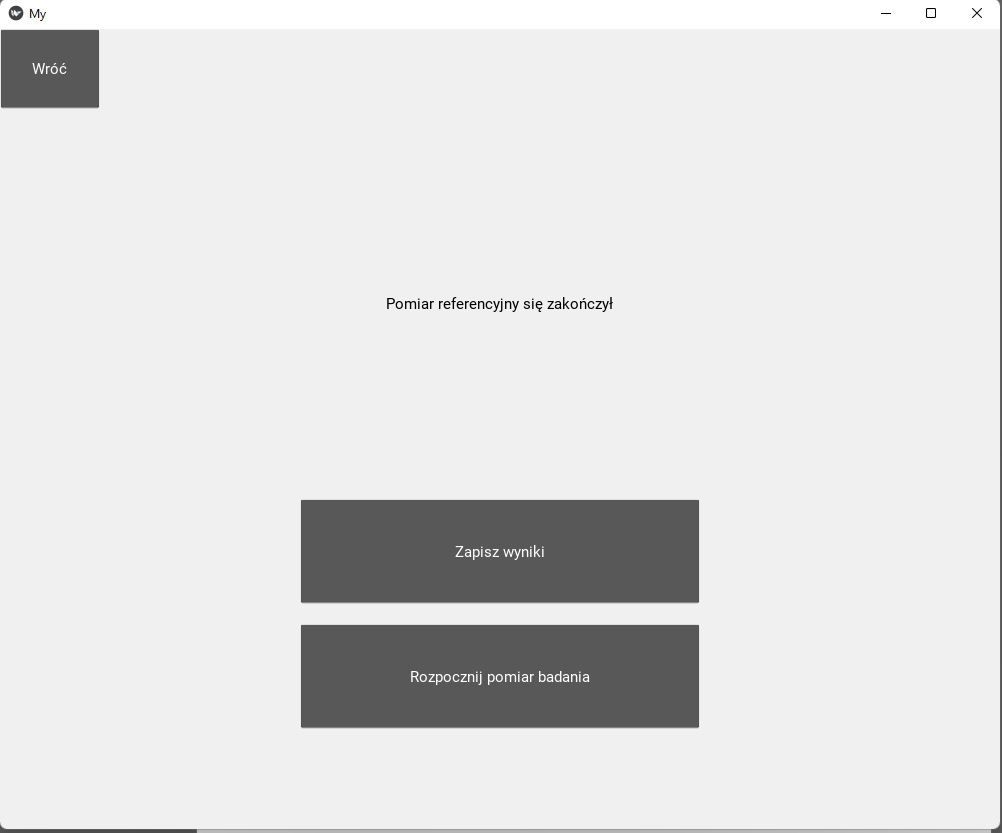
Rysunek 17: Przedstawienie wyboru liczby taśm, tryb Auto ma dostęp wyłącznie do maksymalnie dwóch taśm

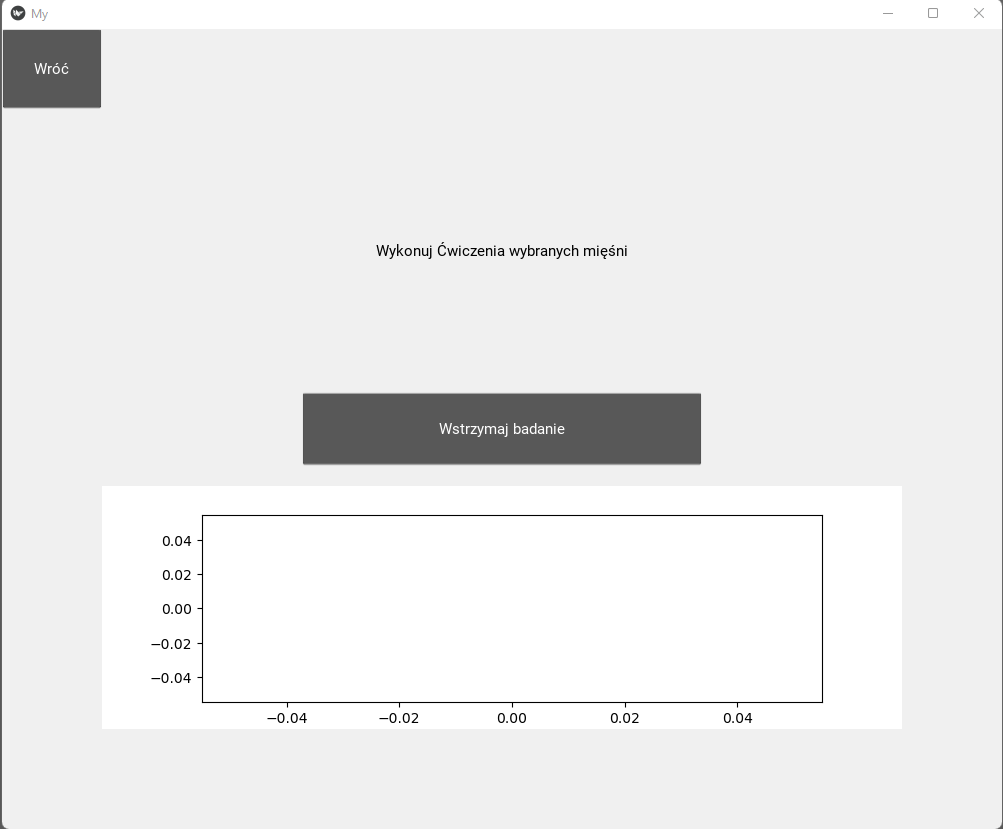
Rysunek 18: Przedstawienie połączenia się z taśmami

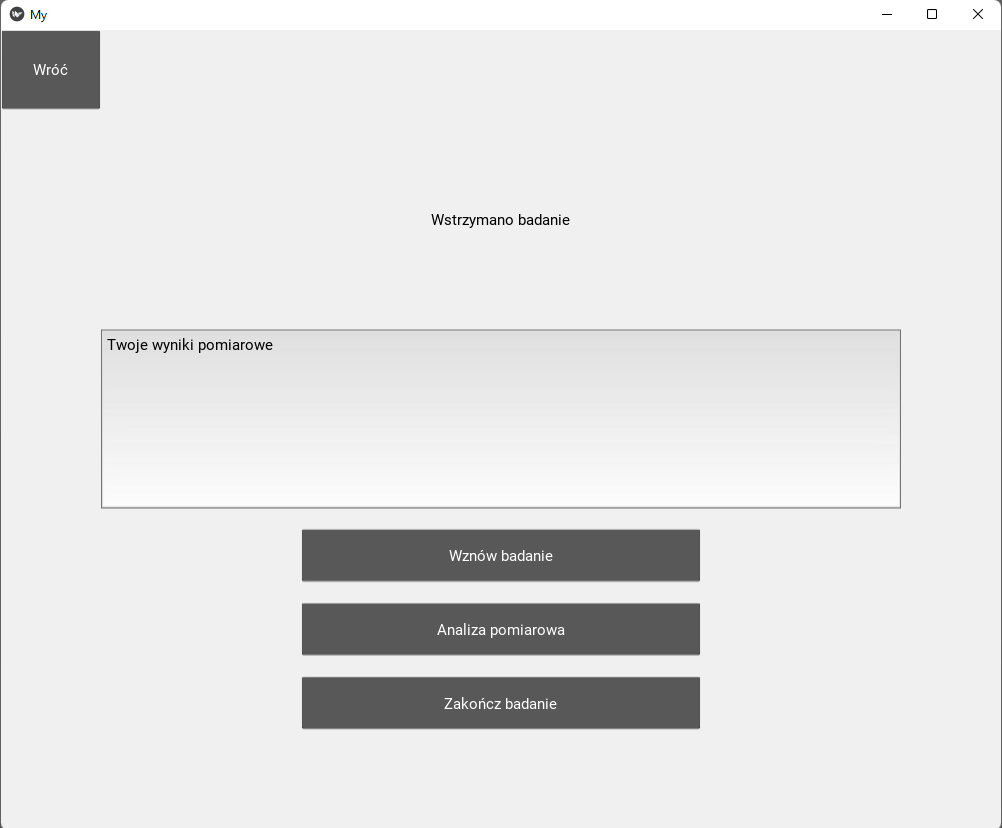
Rysunek 19: Przedstawienie okna widoku aplikacji, który odpowiada za rozpoczęcie pomiaru referencyjnego

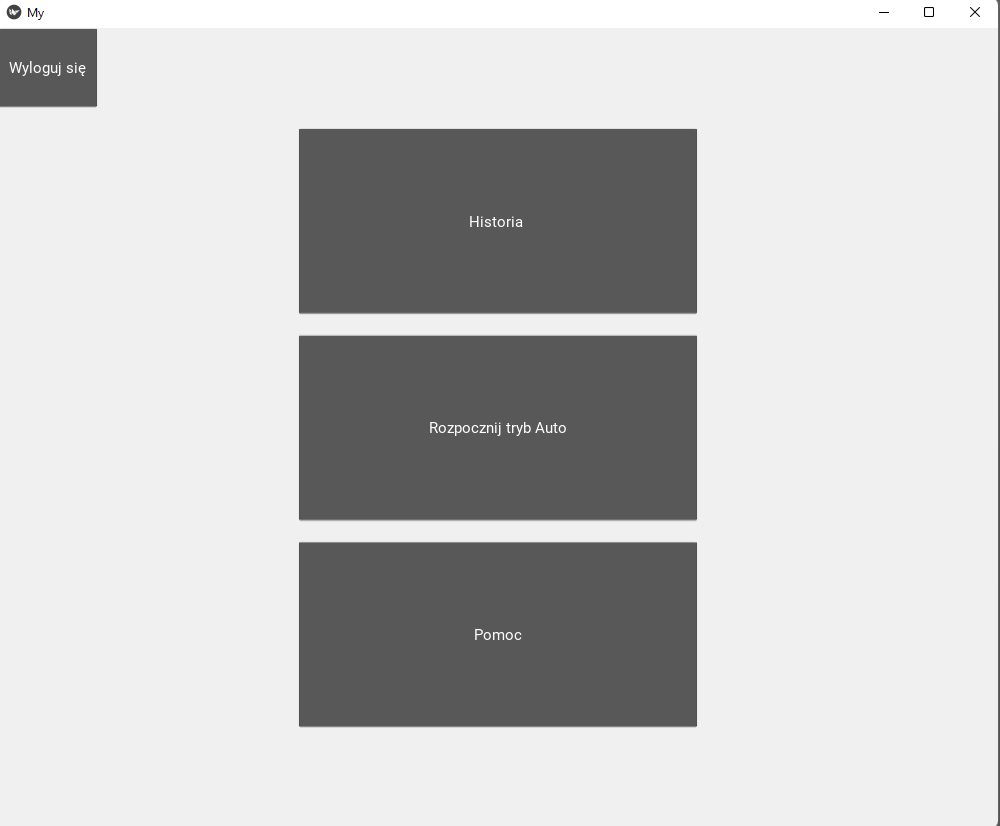
Rysunek 20: Przedstawienie okna aplikacji odpowiedzialnego za wizualizację ćwiczenia referencyjnego

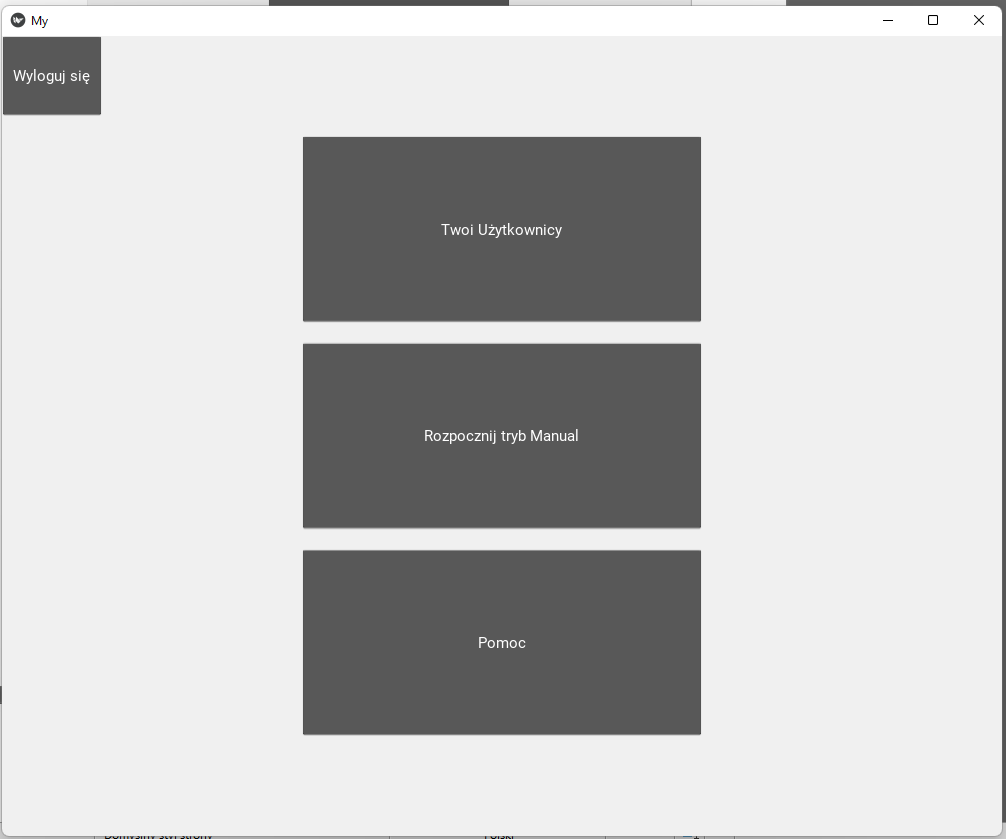
Rysunek 21: Przedstawienie okna aplikacji, który pokazuje przez jaki czas użytkownik musi wykonywać ćwiczenie pokazane w poprzednim oknie

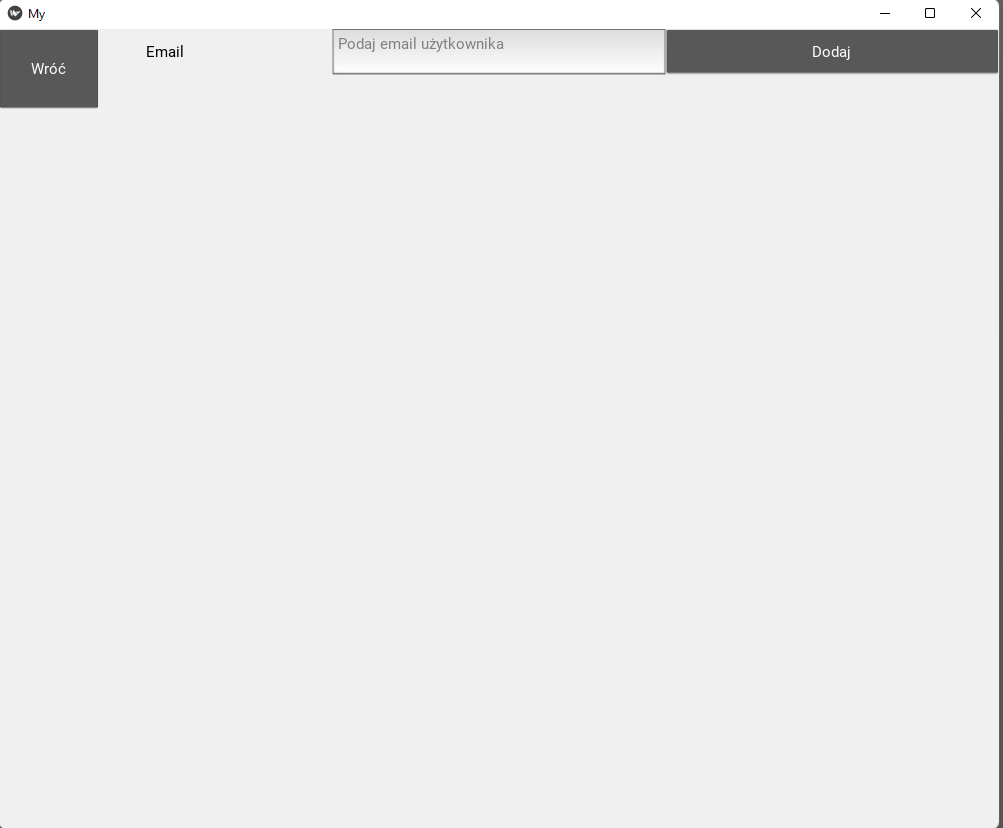
Rysunek 22: Przedstawienie widoku aplikacji informującego o zakończonym pomiarze referencyjnym , gdzie istnieje możliwość zapisania wyników

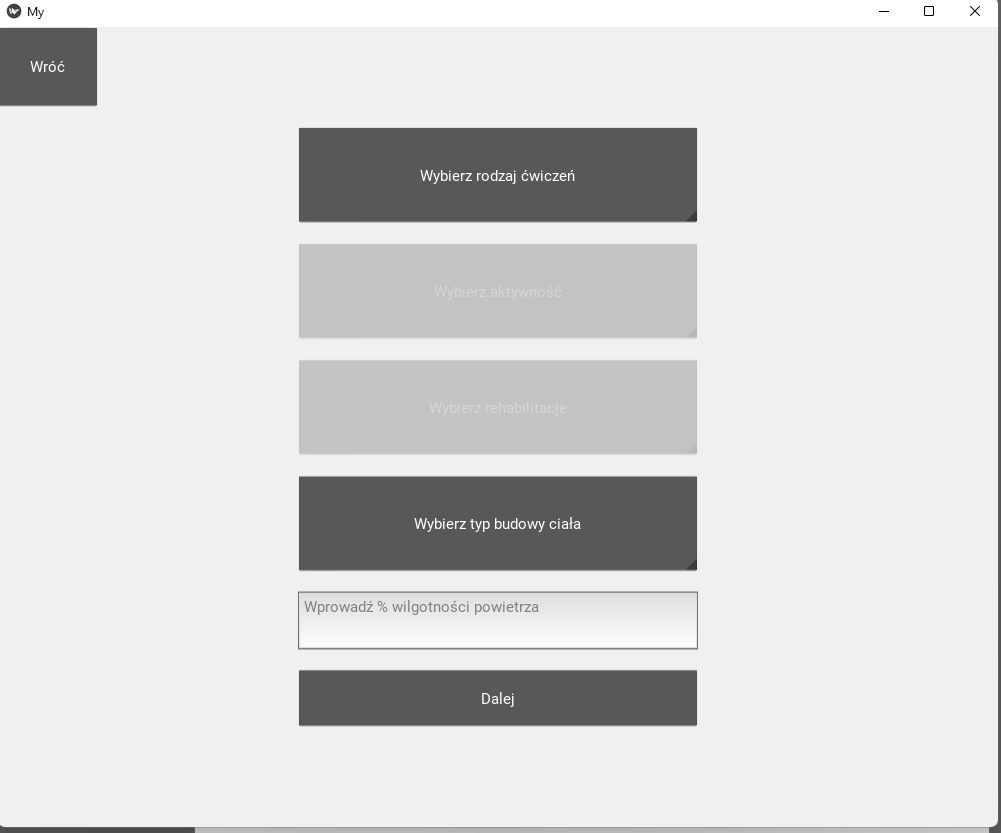
Rysunek 23: Przedstawienie okna aplikacji wyświetlanego użytkownikowi podczas prowadzenia pomiaru właściwego

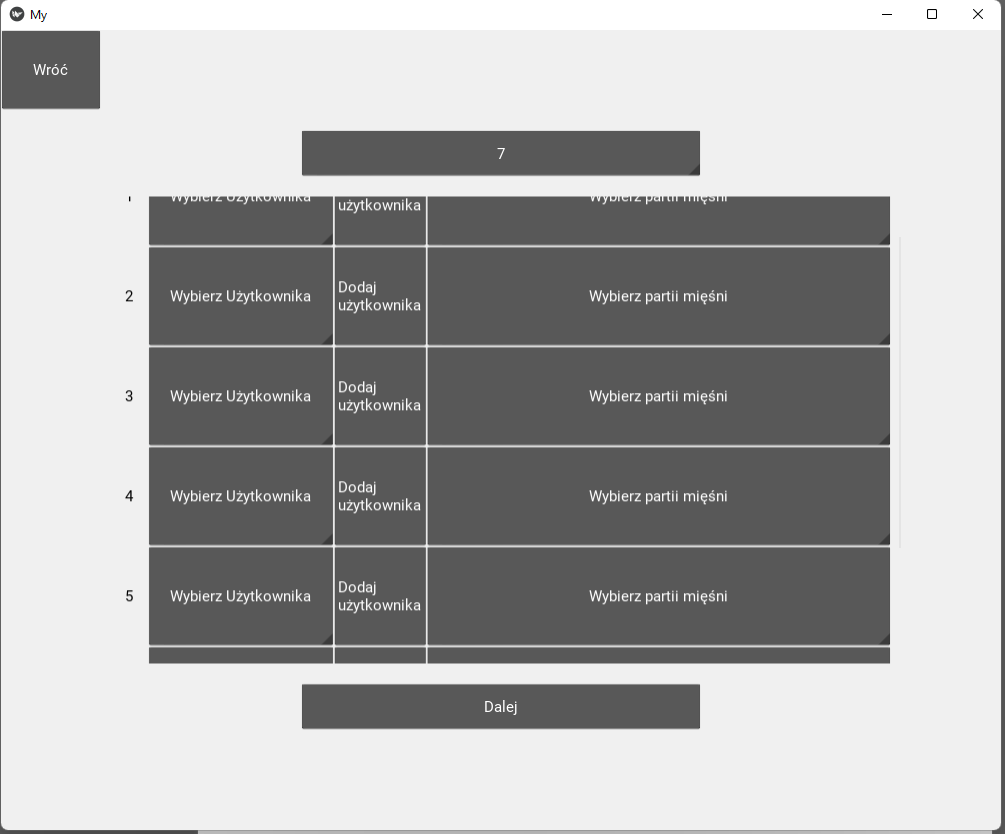
Rysunek 24: Przedstawienie okna aplikacji wyświetlającego się podczas zatrzymania pomiaru właściwego

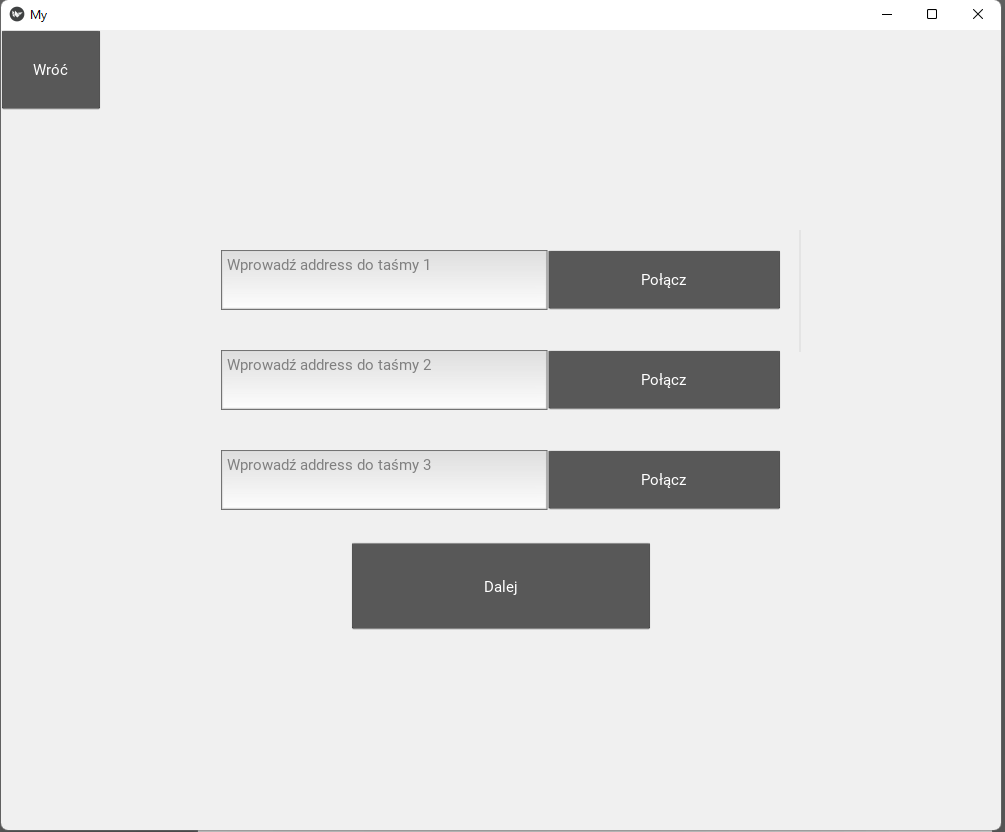
Rysunek 25: Przedstawienie okna aplikacji wyświetlane po zalogowaniu się na konto użytkownika podstawowego

Rysunek 26: Widok aplikacji podczas zalogowania się na konto użytkownika wykwalifikowanego

Rysunek 27: Przedstawienie widoku aplikacji gdzie wyświetlani są podlegający użytkownicy podstawowi

Rysunek 28: Przedstawienie okna aplikacji wyświetlanego jako rozpoczęcie trybu manual

Rysunek 29: Przedstawianie okna aplikacji odpowiadającego wyborowi ilości taśm a także użytkowników i badanych mięśni

Rysunek 30: Przedstawienie okna aplikacji odpowiadającego za połączenie sie z taśmami

**6. Podsumowanie**

Aplikacja na moment zdania raportu zgodnie z założeniami projektowymi i diagramem UML   
nie oferuje wyłącznie wizualizacji danych otrzymywanych z Stacji bazowej w trybie Manual. Reszta przebiegu badawczego jest zgodna z schematem UML a implementacja nie owocowała pojawieniem się znacznych problemów.

Jednym z największych kamieni milowych tworzenia oprogramowania powstało na poziomie komunikacji aplikacji ze stacją bazową co wiązało się z dobraniem odpowiednich frameworków   
i narzędzi , które nie sprawiałyby dodatkowych problemów z implementacją. Na rozwiązanie powyższego problemu, zostało wykorzystane narzędzie gRPC , które poprzez protokół HTTP znacznie uskutecznia obustronną komunikacje.

Wdrażanie poszczególnych elementów schematu UML nie sprawiło problemów, które znacznie wpłynęłyby na opóźnienie pracy implementacjyjnej aplikacji.

Oprogramowanie działa stabilnie gdzie użytkowanie nie powoduje wywoływania błędów, które na fazie prototypu są bardzo częstym zjawiskiem. Testując manualnie aplikacje można dojść do wniosku zamiany narzędzia jakim jest Matplotlib służące do wizualizacji danych pomiarowych na bardziej zaawansowaną bibliotekę, ponieważ framework dotychczas używany jest ograniczony do wizualizacji danych w czasie rzeczywistym , gdzie odświeżanie wykresu nie jest satysfakcjonujące.

Kolejną kwestią poprawy wydajności elementu wizualizacji danych pomiarowych, niestety wiąże się ze zmianą wykorzystywanego narzędzia. Po szczegółowym przeglądzie dostępnych narzędzi przeznaczonych w szczególności do tego typu zadań, została wytypowana biblioteka PyQtGraph, której ograniczenia przede wszystkim związane z szybkością wizualizacji danych jest znacznie mniej ograniczone co do przypadku w dotychczasowego wykorzystywanego narzędzia Matplotlib.

**7 Wykaz literatury.**

[1] Wniosek NCBIR POIR.01.01.01-00-1059/20 „Opracowanie nowego zaawansowanego urządzenia pomiarowo – diagnostycznego do akwizycji sygnałów biologicznych podczas różnych rodzajów treningu wysiłkowego oraz analizy i klasyfikacji stanu ćwiczącego na podstawie zebranych informacji”

Projekt AL TP.

[2] Raport 9/2021. Wizualizacja schematów użytkowania urządzenia w różnych trybach pomiarowych. Etap 1 Faza 2. Marta Kopaczyńska, Aleksandra Kisielewicz, Elżbieta Pietrzak-Szul.

[3] Raport 16/2022. Opis modelu użytkowego Urządzenia Pomiarowo-Diagnostycznego. (W. Klembowski, Konrad Bruliński).

[4] Dokumentacja do wykorzystywanego narzędzia Kivy <https://kivy.org/doc/stable/>

[5] Dokumentacja do wykorzystywanego narzędzia gRPC https://grpc.io/docs/

[6] Dokumentacja do wykorzystywanego narzędzia Matplotlib

<https://matplotlib.org/stable/index.html>

[7] Dokumentacja do wykorzystywanego narzędzia ProtoBuf

https://developers.google.com/protocol-buffers/docs/overview

[8] Dokumentacja do wykorzystywanego narzędzia SQLite https://www.sqlite.org/docs.html

[9] Dokumentacja do wykorzystywanego narzędzia PyTest

<https://docs.pytest.org/en/6.2.x/contents.html>

1. [↑](#footnote-ref-2)
2. [↑](#footnote-ref-3)