

Politechnika Świętokrzyska w Kielcach

Wydział Elektroniki, Automatyki i Informatyki

Projekt: **Technologie obiektowe**

Grupa: 1ID21A	Temat: SilentPackage-Lite– Monitorowanie systemu Windows® 10	Skład grupy: Michał Młodawski
Rok studiów: 4		

SilentPackage-Lite – Monitorowanie systemu Windows 10

Opracowanie i sprawozdanie projektu SilentPackage-
Lite

Kwiecień 2020

1.Opis projektu

Projekt SilentPackage-Lite powstał w celu wykorzystania możliwości nowych technologii w celu stworzenia projektu, którego celem było nieustanne monitorowanie specyfikacji systemu Windows.

2.Wykorzystane technologie

W projekcie wykorzystano najnowocześniejsze technologie dla osiągnięcia jak najlepszych walorów estetycznych aplikacji webowej i szybkości działania całego rozwiązania.

1. Użyte języki programowania

1. Java wraz z framework Spring do warstwy serwerowej.
2. C# w wersji .Net Framework do warstwy agregacji danych.
3. HTML z wykorzystaniem frameworka Bootstrap, JavaScript z frameworkiem jQuery oraz AngularJS.
4. JSON jako format tekstowy dla przejrzystego uporządkowania danych.

2. Wykorzystane oprogramowanie przy projektowaniu i wdrażaniu projektu

1. Adobe XD CC w celu szybkiego prototypowanie interfejsu graficznego.
2. IntelliJ IDEA jako główne IDE do programowanie serwera.
3. JavaDoc do prowadzenia dokumentacji kodu.
4. JetBrains WebStorm jako IDE do interfejsu graficznego.
5. Visual Studio z rozszerzeniem ReSharper

3.Użyte biblioteki w projekcie

1. Biblioteki zewnętrzne

LP.	Nazwa biblioteki	Licencja	Opis
1.	Apache Tomcat	Apache License 2.0	Kontener aplikacji webowych.
2.	Spring Boot	GNU Lesser General Public	Framework Java dodający możliwość tworzenia webowych aplikacji.
3.	Angular.js	MIT	Framework wspomagający tworzenie i rozwój aplikacji internetowych na pojedynczej stronie.
4.	Bootstrap	MIT	Framework CSS dodający przydatne elementy do projektowania stron.
5.	Jquery	MIT	Biblioteka programistyczna dla języka JavaScript, ułatwiająca korzystanie z JavaScriptu.

6.	Font Awesome	Licencja wewnętrzna: https://fontawesome.com/ license	Zestaw piktogramów przedstawionych w formie czcionki.
7.	junit5	Eclipse Public License 2.0	Narzędzie służące do tworzenia powtarzalnych testów jednostkowych.

Tabela 1 Biblioteki zewnętrzne wykorzystane w projekcie.

2. Biblioteki wewnętrzne

1. Open Hardware Monitor – Biblioteka do pobierania informacji o urządzeniu. Fork biblioteki, którego celem było unowocześnienie rozwiązania, port do nowszej wersji .NET Framework i zwracania danych w postaci plików JSON.

4. Funkcjonalność projektu

Projekt został stworzony w oparciu o architekturę klient-serwer. Posiada takie funkcjonalności jak:

1. Pobieranie informacji o procesorze, producent i model oraz temperaturze, obciążeniu i taktowaniu każdego z rdzenia procesora.
2. Pobieranie informacji płyty głównej, nazwa producenta, model, wersja BIOS oraz producent BIOS.
3. Pobieranie informacji o dedykowanej karcie graficznej. Umożliwia pobranie takich danych jak: Producent, model, wersja sterownika, gałąź sterownika, temperaturze rdzenia, ilości całkowitej pamięci vRAM i ilości dostępnej pamięci do alokacji vRAM.
4. Pobranie informacji na temat dysków twardych w tym: Nazwy dysku, formatu, lokalizacji, ilości wolnego oraz zajętego miejsca.
5. Pobieranie informacji o zainstalowanej pamięci RAM w tym producencie, modelu, prędkości, numerze banku i ilości wolnego miejsca.
6. Listy procesów z nazwą, identyfikatorem oraz czasem startu procesu.
7. Graficzny interfejs dostępny z poziomu przeglądarki pracujący w koncepcji „jednej strony” obsługujący zapytania AJAX.
8. Dostęp do API w celu rozszerzenia projektu o dodatkowe punkty wyjścia informacji.

5. Obsługa projektu

Aby móc korzystać z projektu należy posiadać oprogramowanie Java przynajmniej z **wersji 11 oraz system operacyjny Windows® 10 firmy Microsoft z zainstalowanym pakietem .NET framework 4.7.2**

Uwaga! Do poprawnego działania wymaganej jest uruchomienie aplikacji z poziomu użytkownika z prawami administratora.

6. Wygląd interfejsu webowego

1. Strona główna.

Informacje o specyfikacji								
Model płyty głównej: ROG STRIX Z390-F GAMING				Całkowita ilość pamięci RAM: 32 GB				
Model procesora: Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz				Model karty graficznej: NVIDIA GeForce GTX 1080 Ti				
Informacje o płycie głównej								
Producent płyty głównej: ASUSTeK COMPUTER INC.				Producent BIOS: American Megatrends Inc.				
Model: ROG STRIX Z390-F GAMING				Wersja BIOS: 2001				
Procesor								
Producent: Intel(R) Corporation		Model: Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz		Dostępna ilość rdzeni: 8		Dostępna ilość wątków: 16		
Temperatury								
Taktowanie								
Typ: CPU Core #1	Typ: CPU Core #2	Typ: CPU Core #3	Typ: CPU Core #4	Typ: CPU Core #5	Typ: CPU Core #6	Typ: CPU Core #7	Typ: CPU Core #8	Typ: Bus Speed
Taktowanie: 4295 MHz	Taktowanie: 4295 MHz	Taktowanie: 4295 MHz	Taktowanie: 4295 MHz	Taktowanie: 4295 MHz	Taktowanie: 4295 MHz	Taktowanie: 4295 MHz	Taktowanie: 4295 MHz	Taktowanie: 99 MHz
Obciążenie								

Rysunek 1 Częściowy zrzut ekranu z prezentacją funkcjonalności.

Pamięć RAM			
Całkowita ilość pamięci ram: 32 GB		Dostępna ilość pamięci ram: 26578.3515625 KB	
Nazwa producenta: Corsair Oznaczenie: CMK16GX4M2B3600C18 Kanał pamięci: ChannelA-DIMM1 Bank pamięci: BANK 0 Taktowanie pamięci: 3600 MHz	Nazwa producenta: Corsair Oznaczenie: CMK16GX4M2B3600C18 Kanał pamięci: ChannelA-DIMM2 Bank pamięci: BANK 1 Taktowanie pamięci: 3600 MHz	Nazwa producenta: Corsair Oznaczenie: CMK16GX4M2B3600C18 Kanał pamięci: ChannelB-DIMM1 Bank pamięci: BANK 2 Taktowanie pamięci: 3600 MHz	Nazwa producenta: Corsair Oznaczenie: CMK16GX4M2B3600C18 Kanał pamięci: ChannelB-DIMM2 Bank pamięci: BANK 3 Taktowanie pamięci: 3600 MHz
Dyski twarde			
Nazwa : WDC WD40EZRZ Wersja oprogramowania: 19.01H19 Ścieżka logiczna: D\ Całkowity rozmiar: 3725.3887729644775 GB Dostępna wolna pamięć: 2920420.947265625 MB	Nazwa : Samsung SSD 860 PRO 512GB Wersja oprogramowania: EXM0486Q Ścieżka logiczna: E\ Całkowity rozmiar: 465.75976181030273 GB Dostępna wolna pamięć: 36235.0546875 MB	Nazwa : Raid 0 Volume Wersja oprogramowania: 1.0.00 Ścieżka logiczna: F\ Całkowity rozmiar: 3726.0390548706055 GB Dostępna wolna pamięć: 1521471.74609375 MB	Nazwa : Generic Solid-state drive (NVMe) Wersja oprogramowania: Unknown Ścieżka logiczna: C\ Całkowity rozmiar: 1863.0195274353027 GB Dostępna wolna pamięć: 35032.3515625 MB
Zewnętrzny układ graficzny			
Nazwa: NVIDIA GeForce GTX 1080 Ti Rewizja sprzętu: A1	Wersja sterownika: 441.41 Aktualna temperatura: 52°C	Gałąź sterownika: r441_20-8 Całkowita pamięć vRAM: 11534336 KB	Identyfikator karty: 1B0610DE Dostępna pamięć vRAM: 11394048 KB

Rysunek 2 Częściowy zrzut ekranu z prezentacją funkcjonalności.

2. Lista procesów.

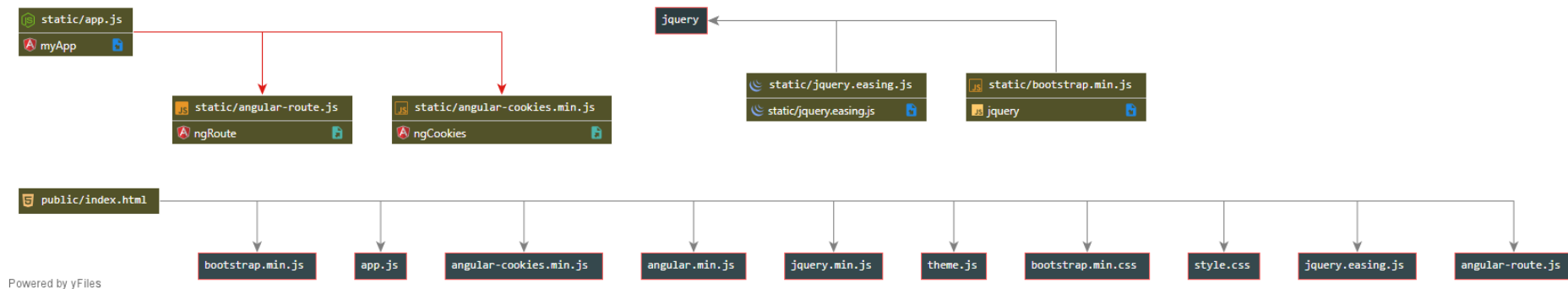


PID	Nazwa	Data startu
1620	svchost	27.03.2020 19:12:13
9188	NisSrv	27.03.2020 19:12:15
8136	conhost	27.03.2020 20:39:11
23884	conhost	27.03.2020 20:38:59
908	lsass	27.03.2020 19:12:12
14688	OneDrive	27.03.2020 19:12:31

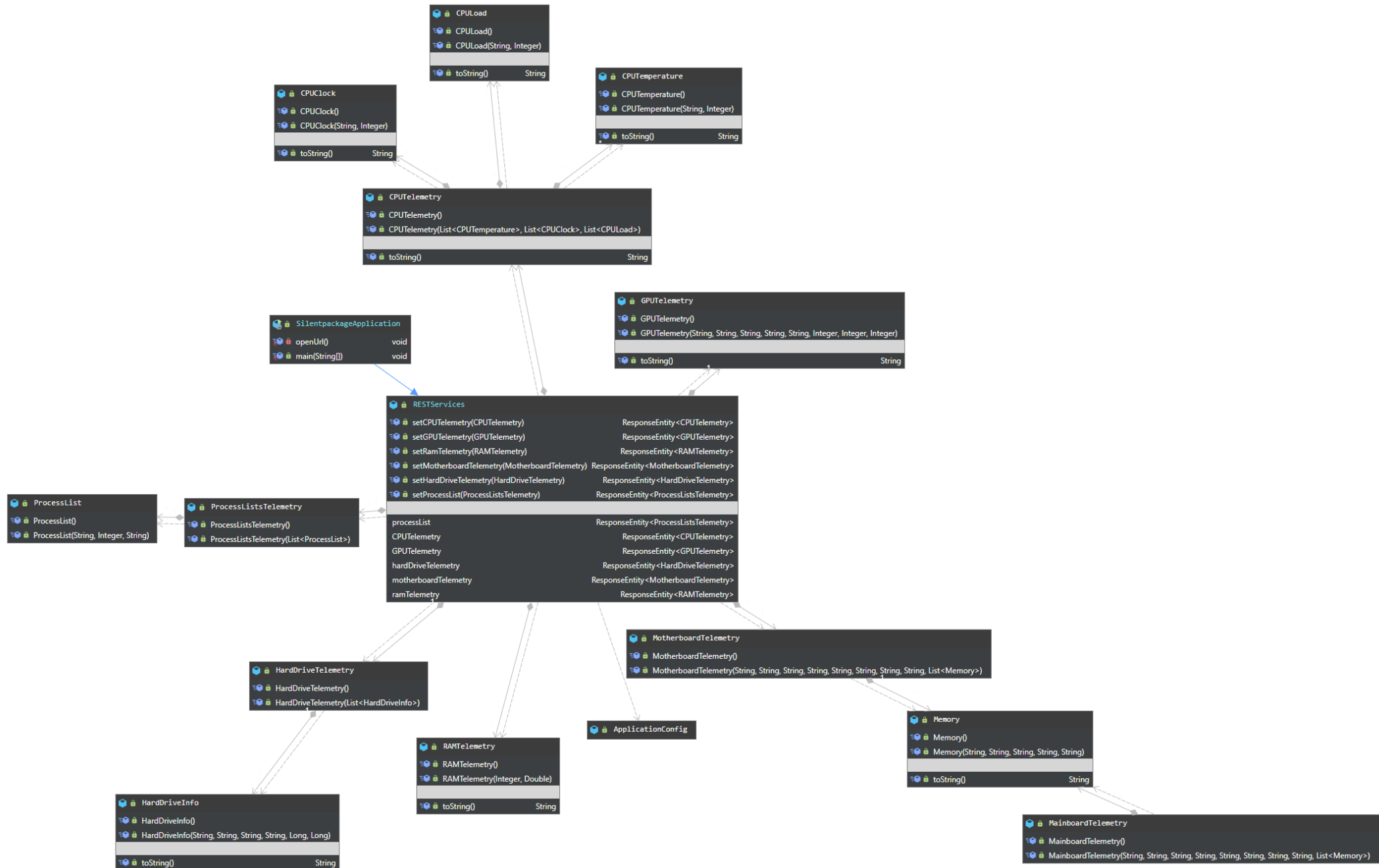
Rysunek 4 Częściowy zrzut ekranu z prezentacją funkcjonalności.

7. Diagramy prezentujące projekt.

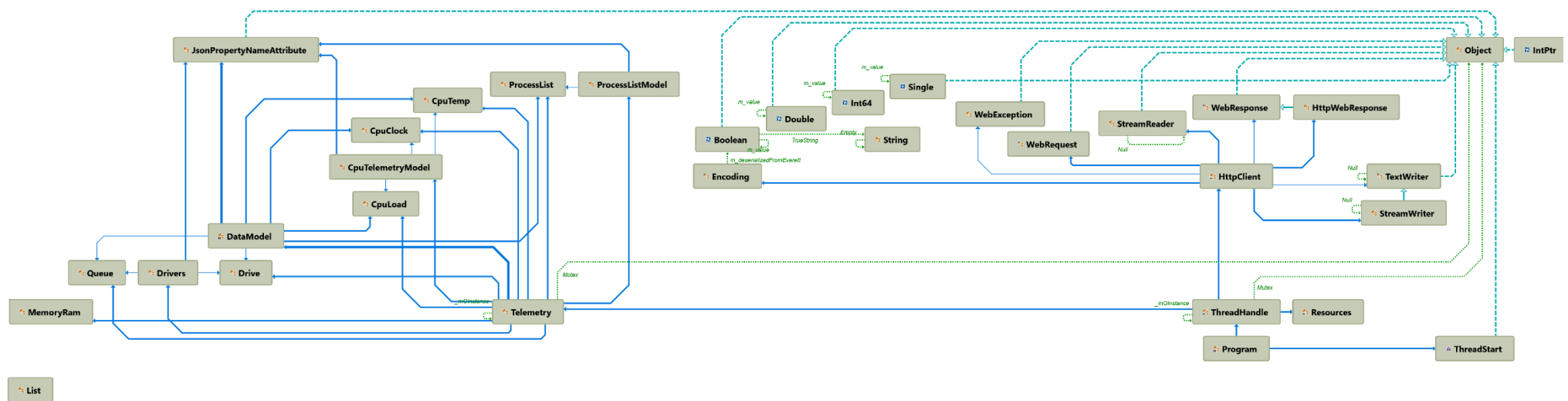
1. Diagram klas interfejsu webowego.



2. Diagram klas części serwerowej.

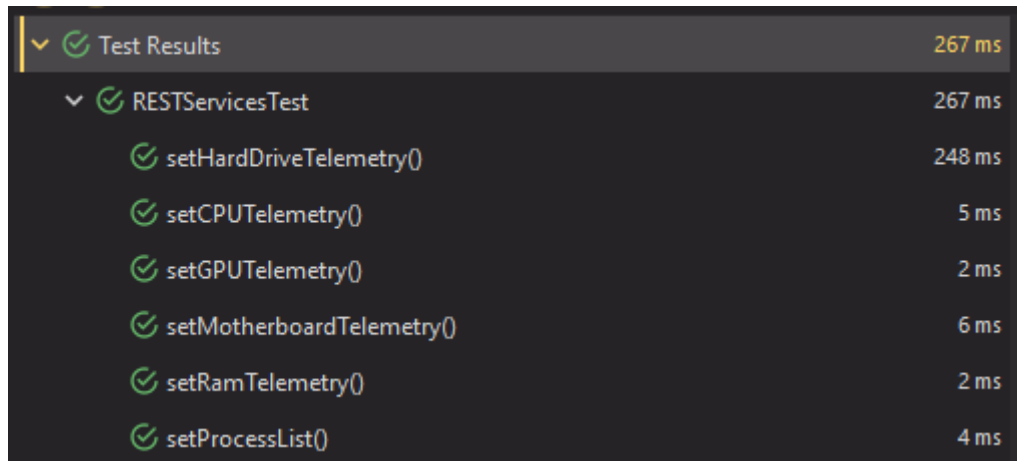


3. Diagram klas i zależności części projektu odpowiedzialnej za generowanie telemetrii (klient).



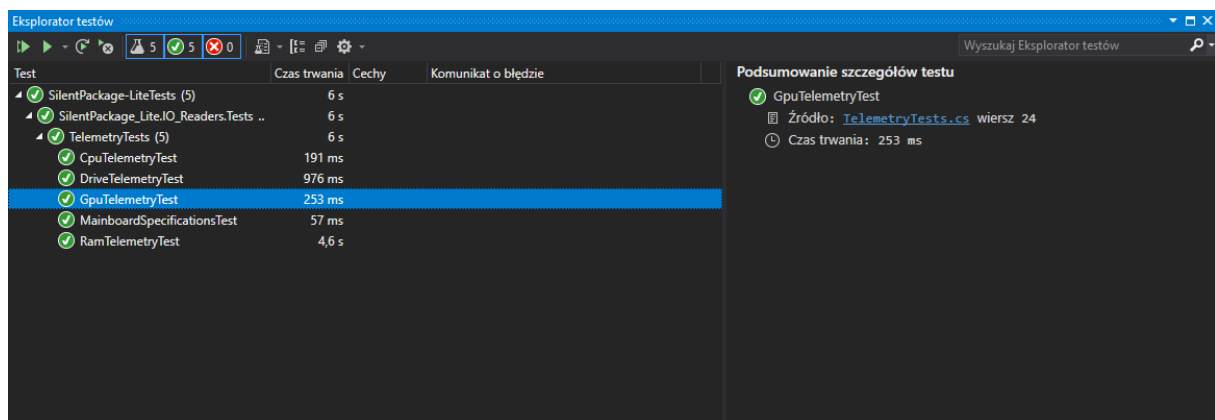
8. Testy

Podczas realizacji projektu przeprowadzono testy jednostkowe oraz funkcjonalne w ramach czarnej skrzynki w celu wyeliminowania potencjalnych problemów.



Test Results	267 ms
RESTServicesTest	267 ms
setHardDriveTelemetry()	248 ms
setCPUTElemetry()	5 ms
setGPUTElemetry()	2 ms
setMotherboardTelemetry()	6 ms
setRamTelemetry()	2 ms
setProcessList()	4 ms

Rysunek 5 Wynik testów jednostkowych dla serwera



Test	Czas trwania	Cechy	Komunikat o błędzie
SilentPackage-LiteTests (5)	6 s		
SilentPackage_LiteIO_Readers.Tests ..	6 s		
TelemetryTests (5)	6 s		
CpuTelemetryTest	191 ms		
DriveTelemetryTest	976 ms		
GpuTelemetryTest	253 ms		
MainboardSpecificationsTest	57 ms		
RamTelemetryTest	4,6 s		

Podsumowanie szczegółów testu

- GpuTelemetryTest
- Źródło: [TelemetryTests.cs](#) wiersz 24
- Czas trwania: 253 ms

Rysunek 6 Wynik testów jednostkowych dla klienta.

9. Podsumowanie projektu

Podczas realizacji projektu nie napotkano większych. Dzięki wykorzystaniu nowoczesnych technologii, wzorów projektowych takich jak Singleton oraz Fasada stworzono projekt kompaktowy i łatwy do rozwinięcia w przyszłości. Chciałbym podziękować społeczności Stack Overflow za niezliczoną pomoc.