<table>
<tr><td colspan="3">Politechnika Świętokrzyska w Kielcach

Wydział Elektroniki, Automatyki i Informatyki</td></tr>
<tr><td colspan="3">Projekt: Technologie obiektowe</td></tr>
<tr><td>Grupa:

1ID21A</td><td rowspan="2">Temat:

SilentPackage-Lite–Monitorowanie systemu Windows® 10</td><td rowspan="2">Skład grupy:

Michał Młodawski</td></tr>
<tr><td>Rok studiów:

4</td></tr>
</table>

SilentPackage-Lite – Monitorowanie systemu Windows 10

Opracowanie i sprawozdanie projektu SilentPackage-Lite

Kwiecień 2020

Wersja dokumentu 28032020-001PL

1. Opis projektu

Projekt SilentPackage-Lite powstał w celu wykorzystania możliwość nowych technologii w celu stworzenia projektu, którego celem było nieustane 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 formacie 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 zawracania 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 vRMA.
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.

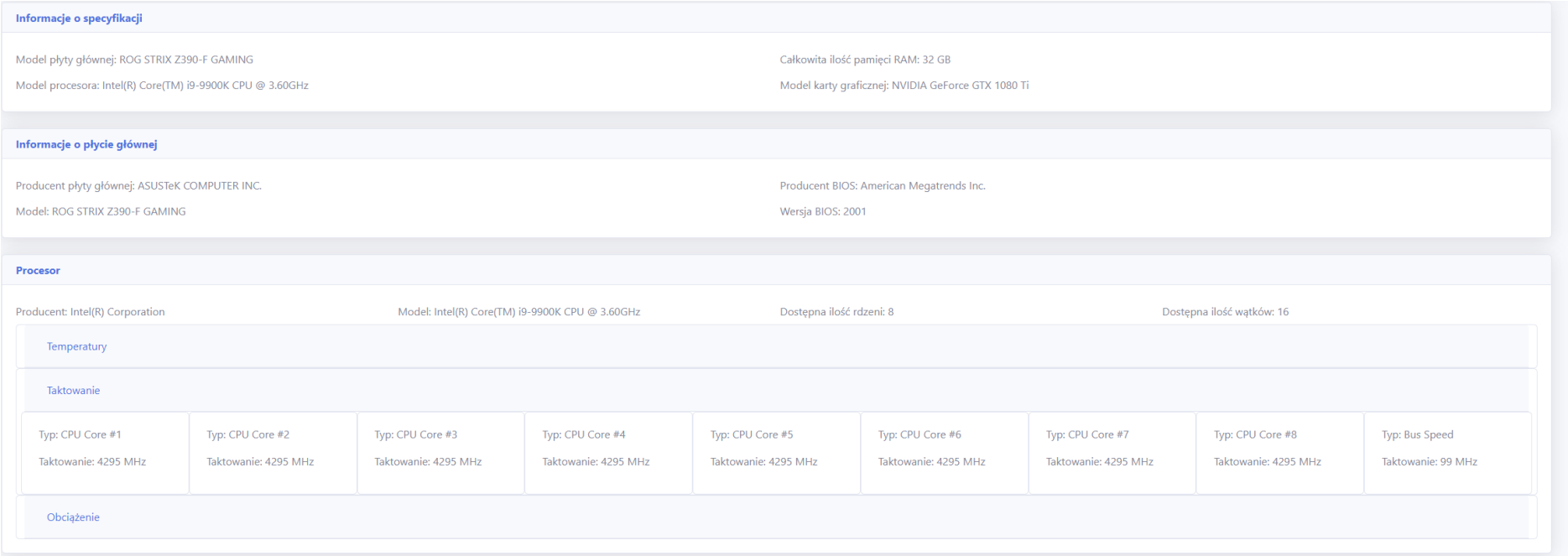
5. Obsługa projektu

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**

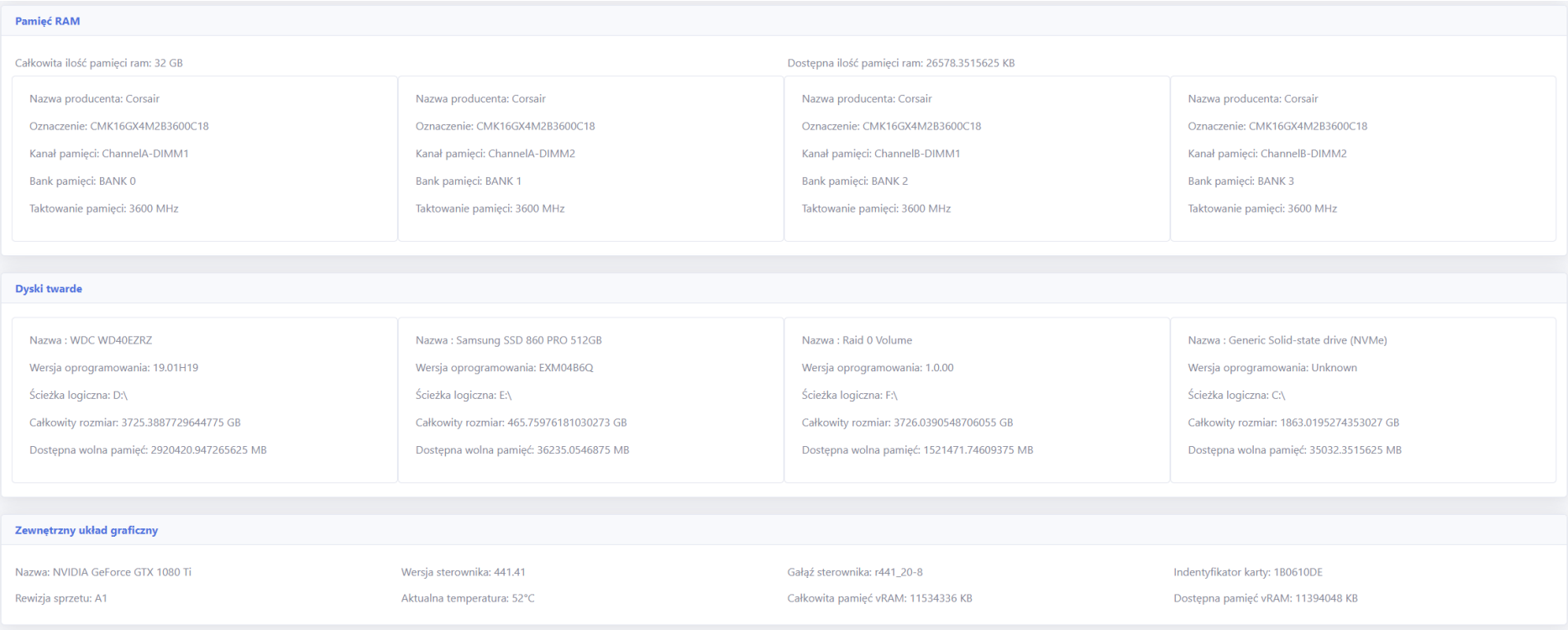
Uwaga! Do poprawnego działania wymaganej jest uruchomienie aplikacji z poziomu użytkownika z prawami administratora.

6. Wygląd interfejsu webowego

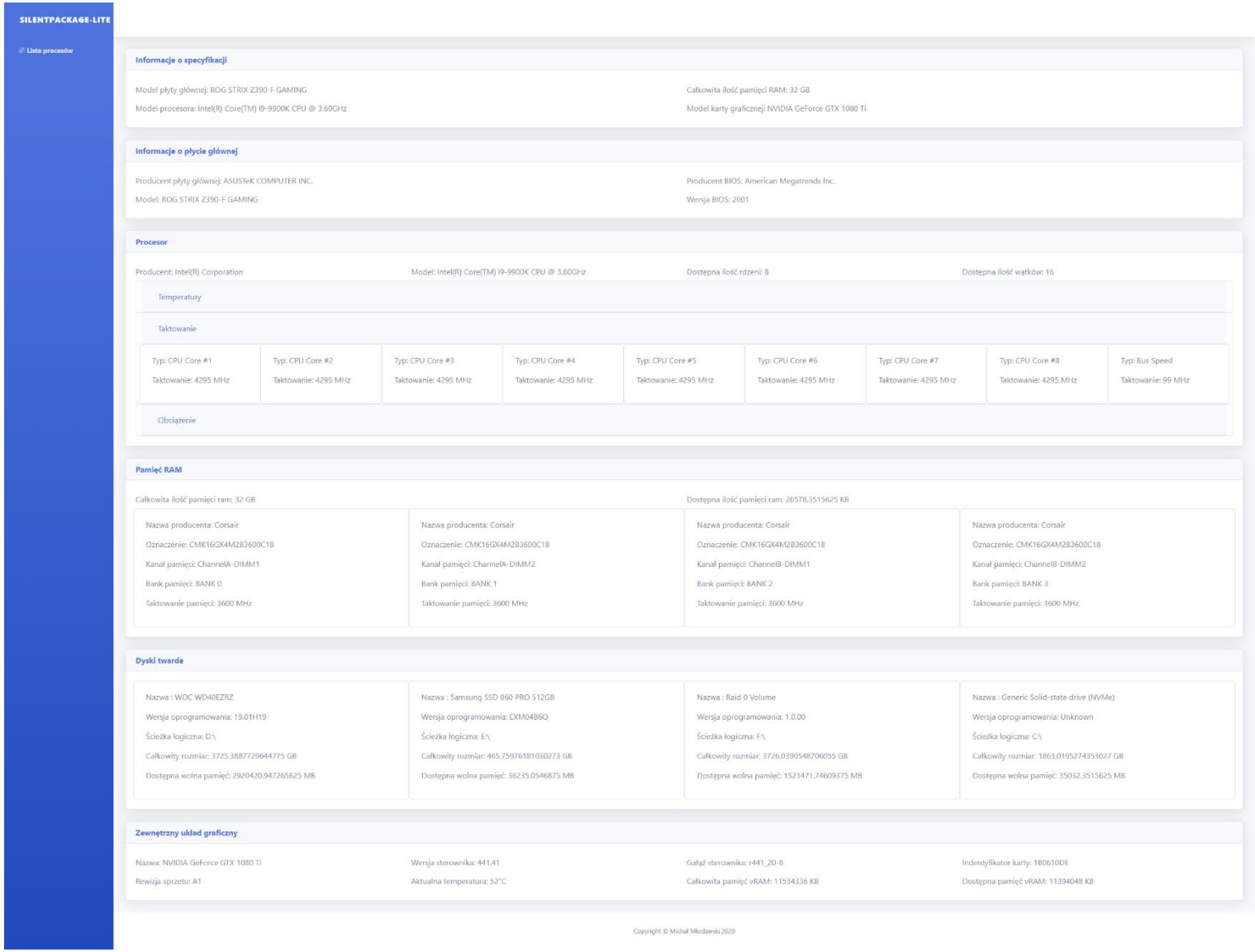
1. Strona główna.



Rysunek 1 Częściowy zrzut ekranu z prezentacją funkcjonalności.

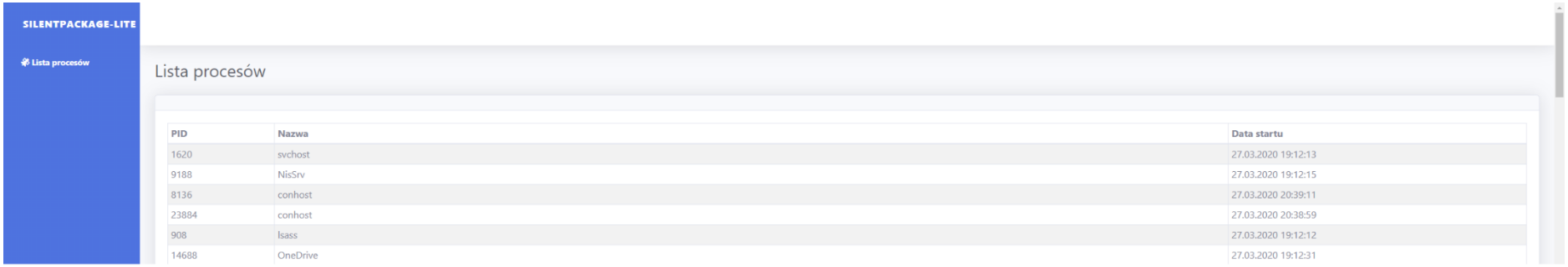


Rysunek 2 Częściowy zrzut ekranu z prezentacją funkcjonalności.



Rysunek 3 Zrzut ekranu z prezentacją funkcjonalności

2. Lista procesów.

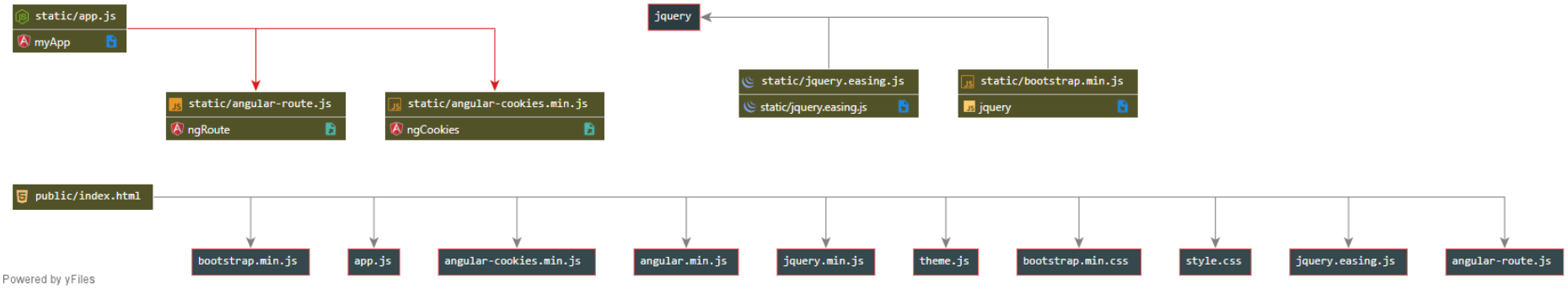


PID	Nazwa	Data startu
1620	svchost	27.03.2020 19:12:13
9188	NisSrv	27.03.2020 19:12:15
8136	conhost	27.03.2020 20:39:11
23884	conhost	27.03.2020 20:38:59
908	lsass	27.03.2020 19:12:12
14688	OneDrive	27.03.2020 19:12:31

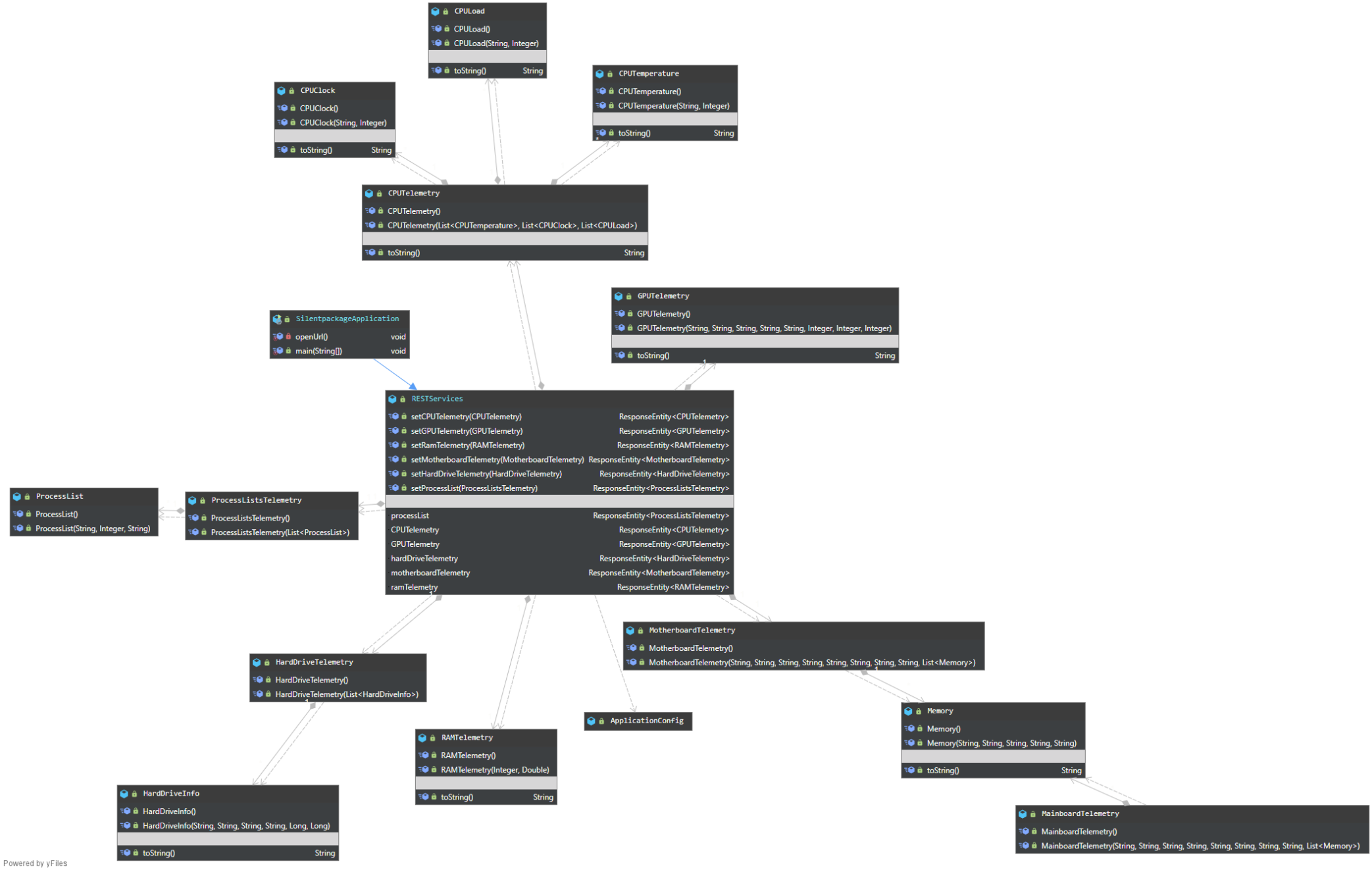
Rysunek 4 Częściowy zrzut ekranu z prezentacją funkcjonalności.

7. Diagramy prezentujące projekt.

1. Diagram klas interfejsu webowego.

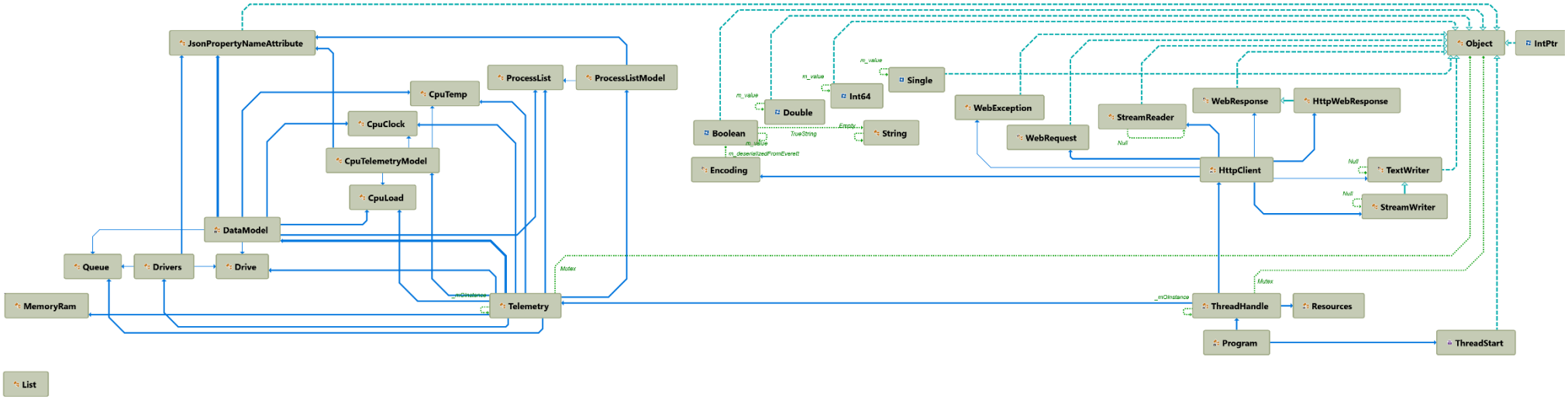


2. Diagram klas części serwerowej.



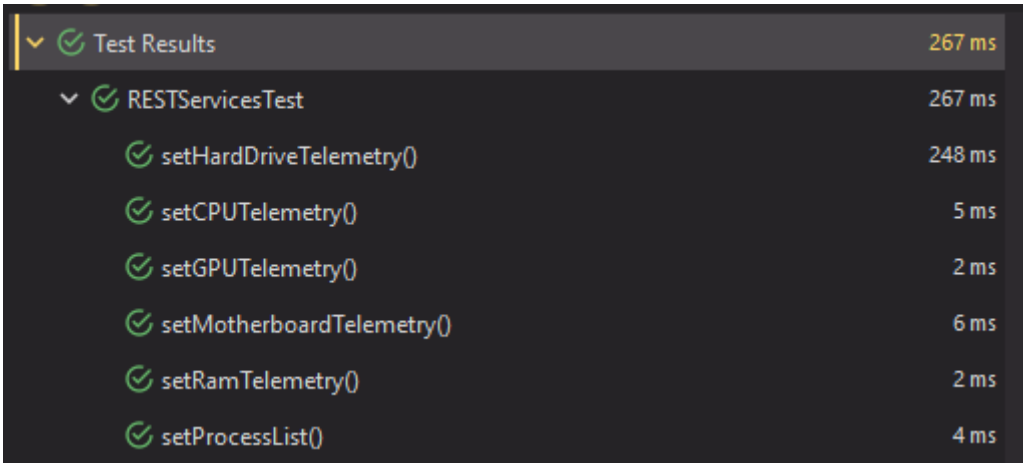
Powered by yFiles

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

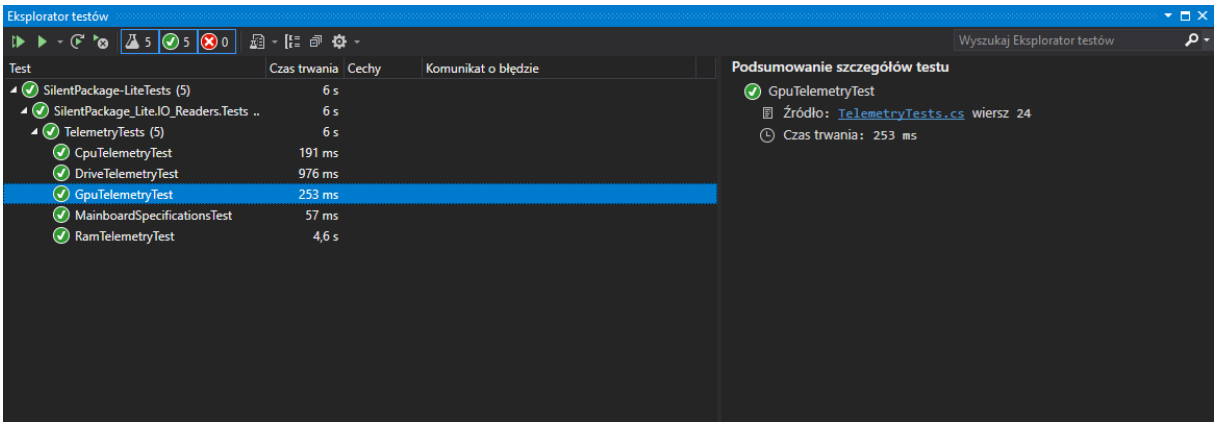


8. Testy

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



Rysunek 5 Wynik testów jednostkowych dla serwera



Rysunek 6 Wynik testów jednostkowych dla klienta.

9. Podsumowanie projektu

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