****

**WYŻSZA SZKOŁA INFORMATYKI**

**I ZARZĄDZANIA**

**Z SIEDZIBĄ W RZESZOWIE**

**SuplementSelector**

Prowadzący: Wykonawcy:

dr Marek Jaszuk Marcin Kalandyk w55261

Kierunek: 4IID-P

Rzeszów 2017

1. **Założenia projektowe**

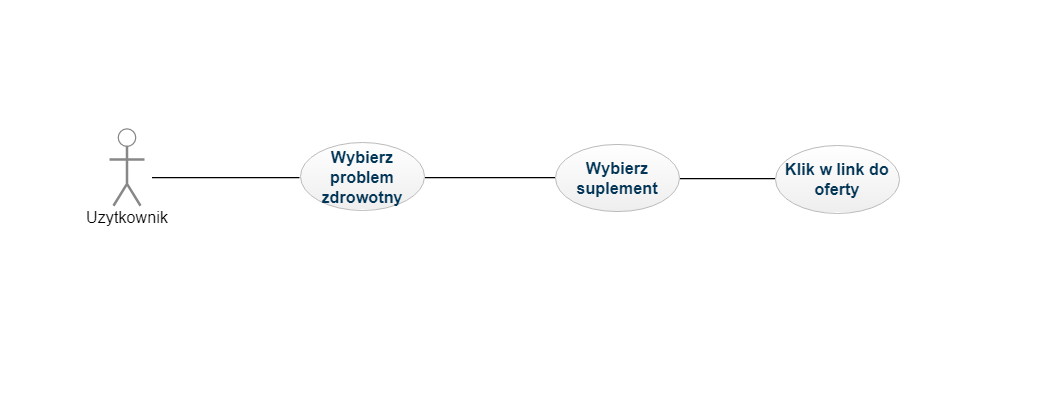
Celem mojego projektu jest utworzenie aplikacji z interfejsem użytkownika wykonanej w technologii WPF w języku C# i serwera w technologii Java. Kod aplikacji pisany był w programie Microsoft Visual Studio (WPF/C#) oraz IntelliJ IDEA CE (Java). Podczas tworzenia pomocne były również takie aplikacje jak Postman (wysyłanie requestów do serwera Javy) czy DB Browser for SQLite (przeglądanie bazy danych). Aplikacja po wyborze problemu zdrowotnego ma za zadanie pomóc nam w doborze suplementu, który ma na celu poprawę naszego stanu zdrowia bądź samopoczucia. Aplikacja dostarcza nam podstawowe informacje o problemie zdrowotnym oraz suplemencie i jego dawkowaniu.

1. **Możliwości i ograniczenia projektu:**

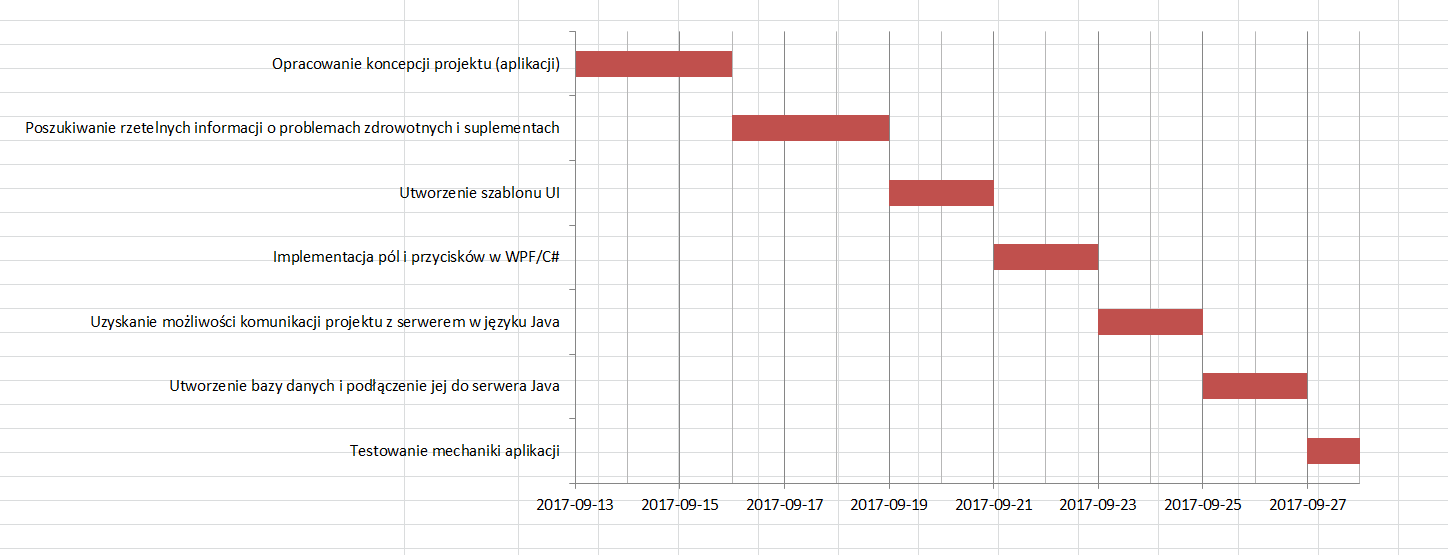
Projekt w aktualnej wersji posiada:

* Wyświetlanie Interfejsu graficznego ( GUI ) który zawiera między innymi:
  + Rozwijaną listę problemów zdrowotnych do wyboru
  + Pole w którym znajdziemy opis wybranego problemu zdrowotnego
  + Pole w którym znajdziemy przyczyny wybranego problemu zdrowotnego
  + Rozwijaną listę suplementów do wyboru których rodzaj będzie zależny od wybranego problemu zdrowotnego
  + Pole w którym wyświetlany będzie opis wybranego suplementu
  + Pole w którym wyświetlane będą zalecenia co do stosowania danego suplementu
  + Pole z linkiem URL z przekierowaniem do strony internetowej sklepu na której będzie można dokonać zakupu danego suplementu
* Serwer w technologii Java.
  + Serwer wykorzystuje technologię SpringBoot
  + Komunikujemy się za pomocą protokołu REST
  + Serwer przechowuje dane w bazie SQLite

1. **Diagram przypadków użycia:**



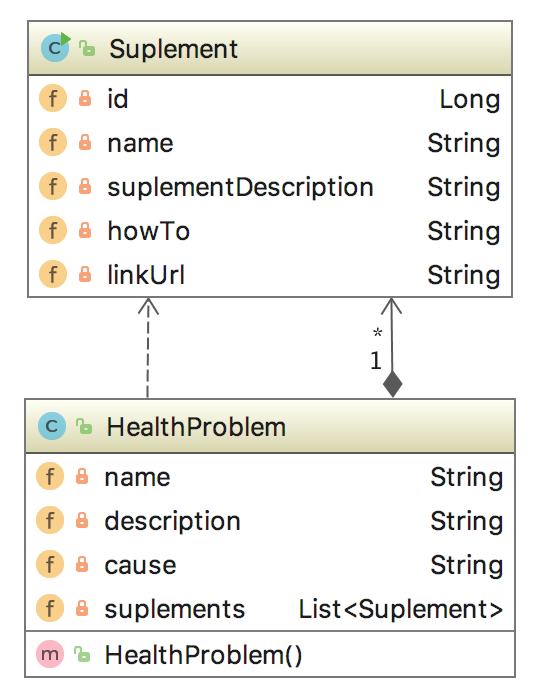
1. **Harmonogram realizacji projektu(wykres Gantta):**

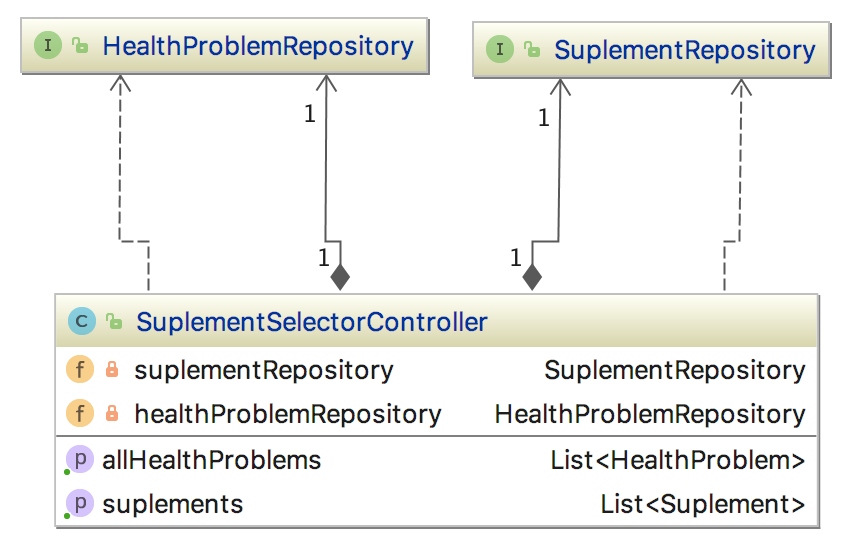


1. **Diagram klas**

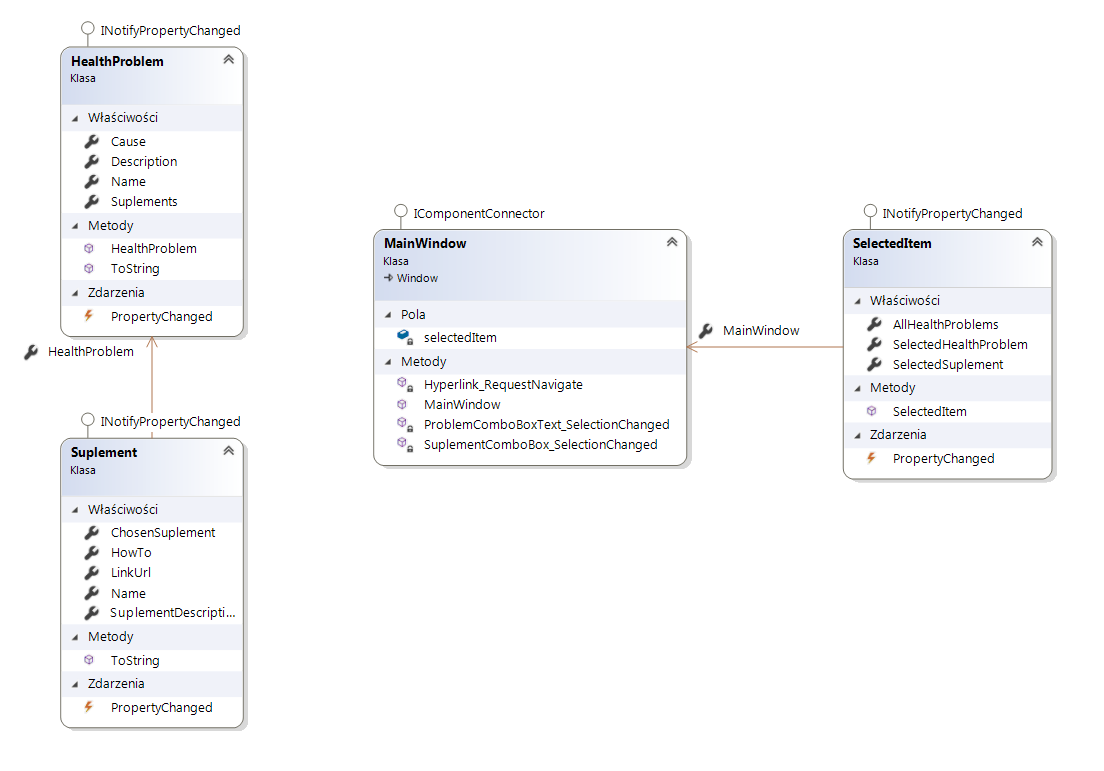
Na rysunku poniżej znajduje się diagram klas, który przedstawia klasy obiektów w programie. Pokazuje uproszczony zarys egzemplarzy (instancji) obiektów i ich zależności istniejących w konkretnym momencie. Diagram klas ukazuje określony fragment struktury działanie programu.

## 5a. Serwer Java



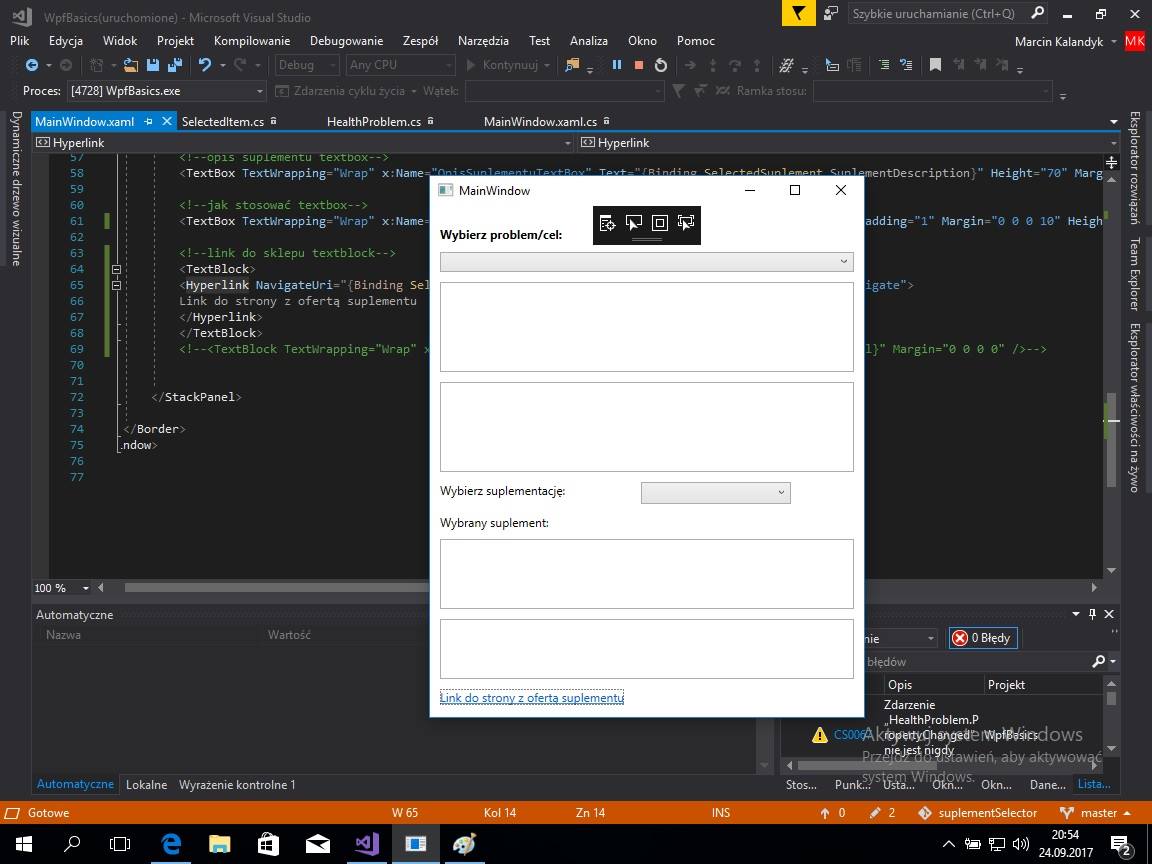


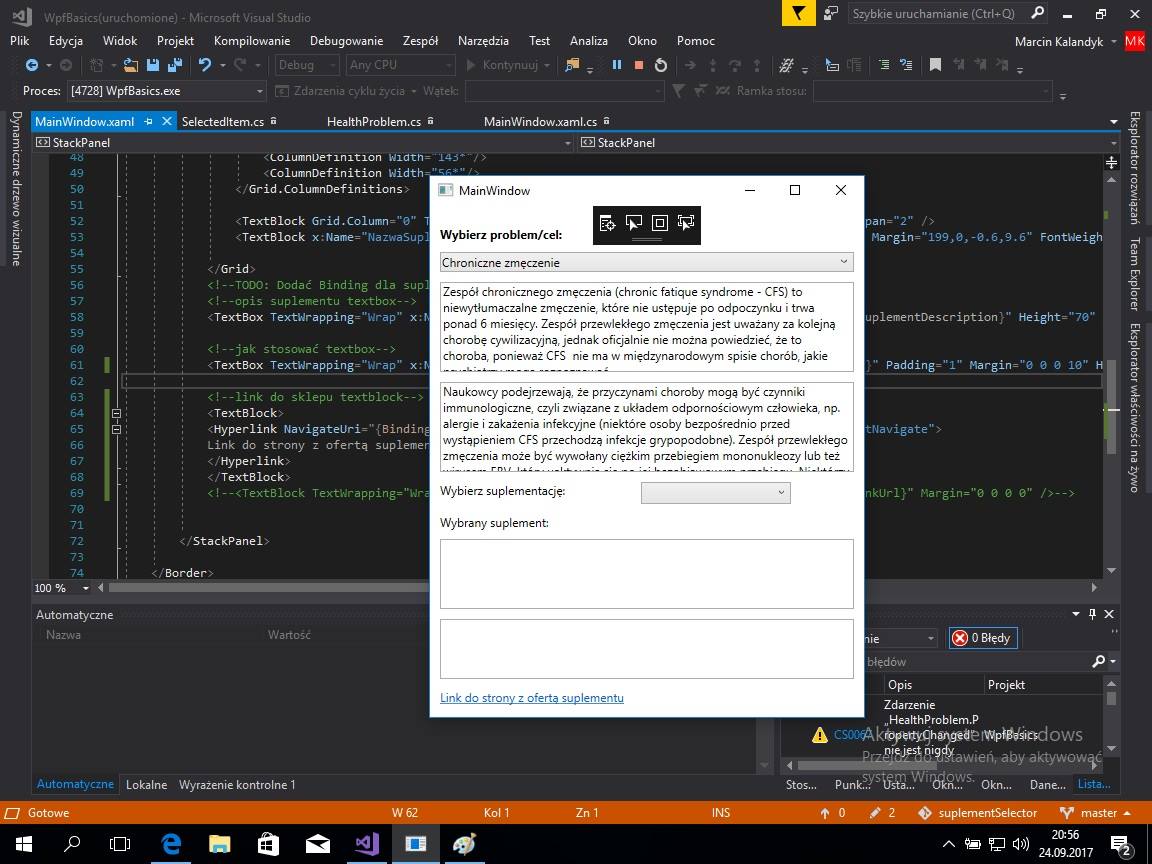
## 5b. WPF/C# (Interfejs użytkownika)



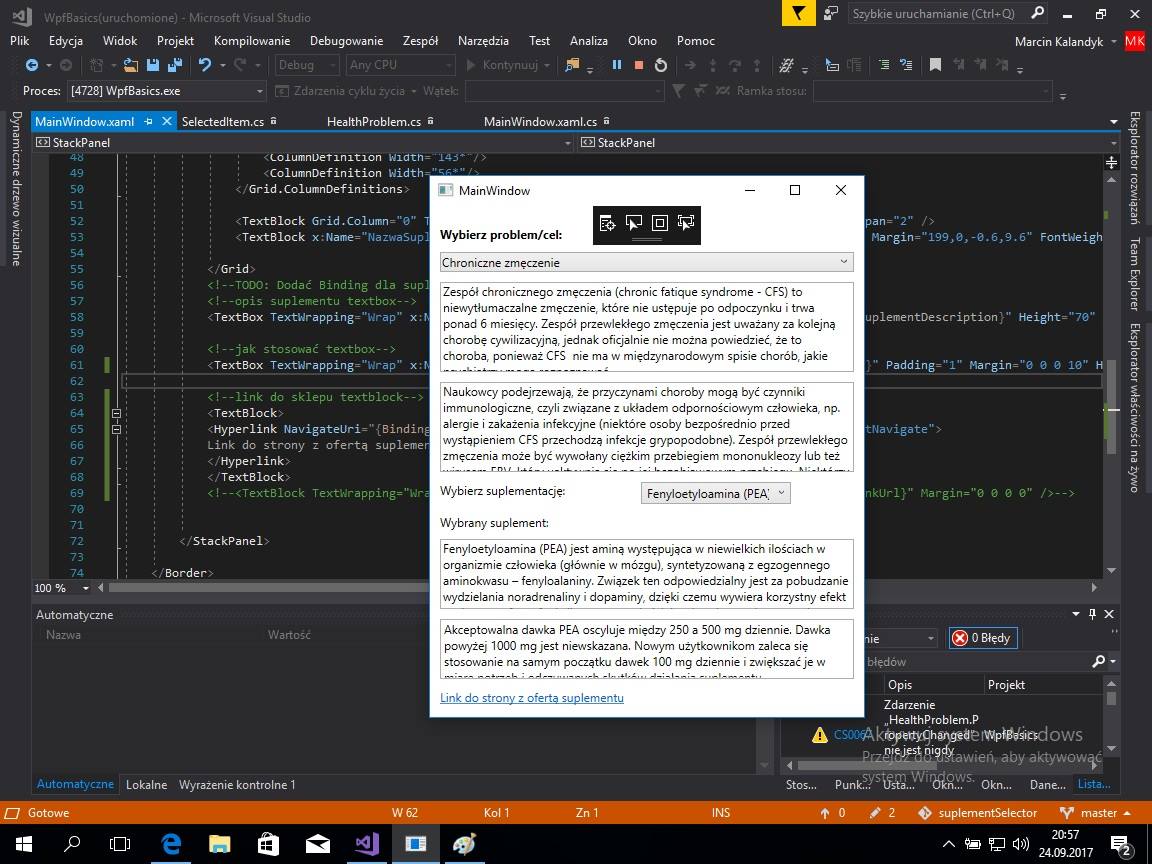
1. **Prezentacja warstwy użytkowej projektu – wygląd interfejsu**

Na pierwszym zrzut ekranu widzimy uruchomioną aplikację, lecz jeszcze bez żadnego wybranego problemu zdrowotnego.

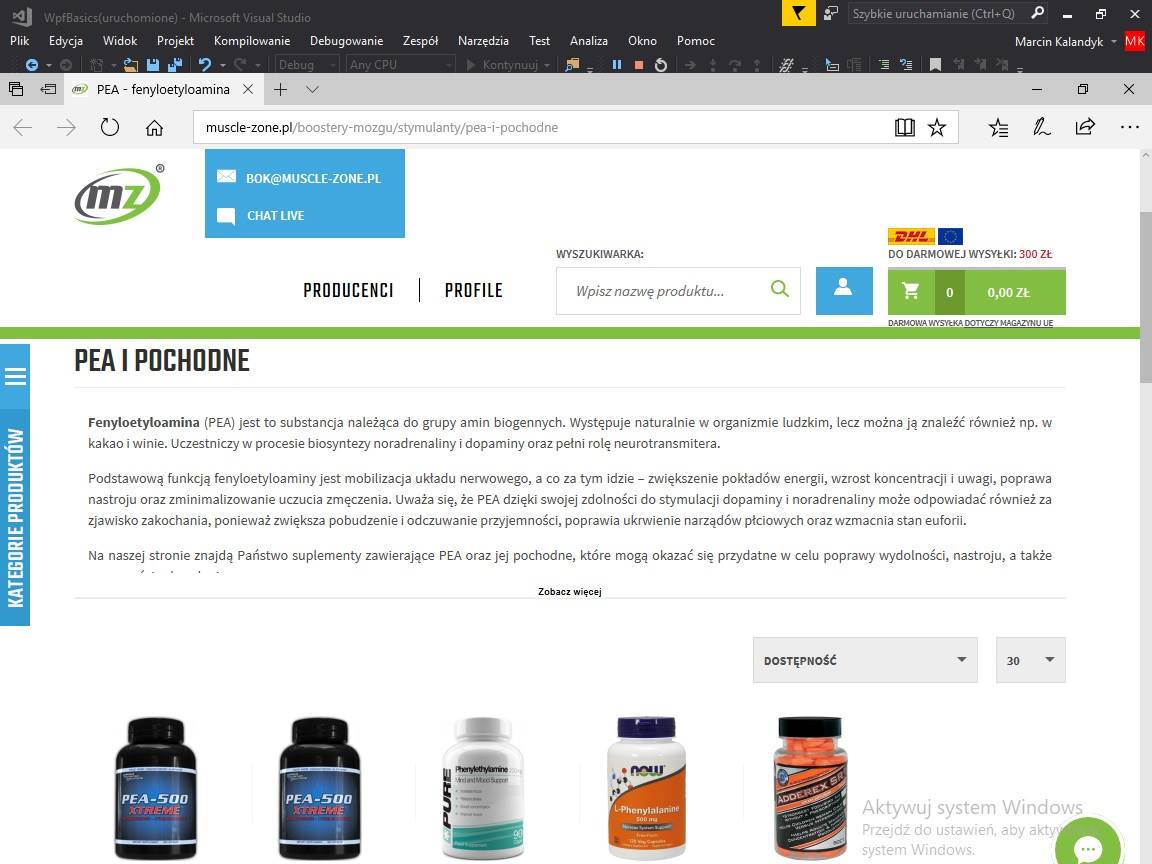


Drugi zrzut ekranu ilustruje aplikację po wyborze problemu zdrowotnego. 

Natomiast trzeci zrzut ilustruje aplikację po wyborze problemu zdrowotnego i wybraniu odpowiedniego dla danego problemu suplementu.



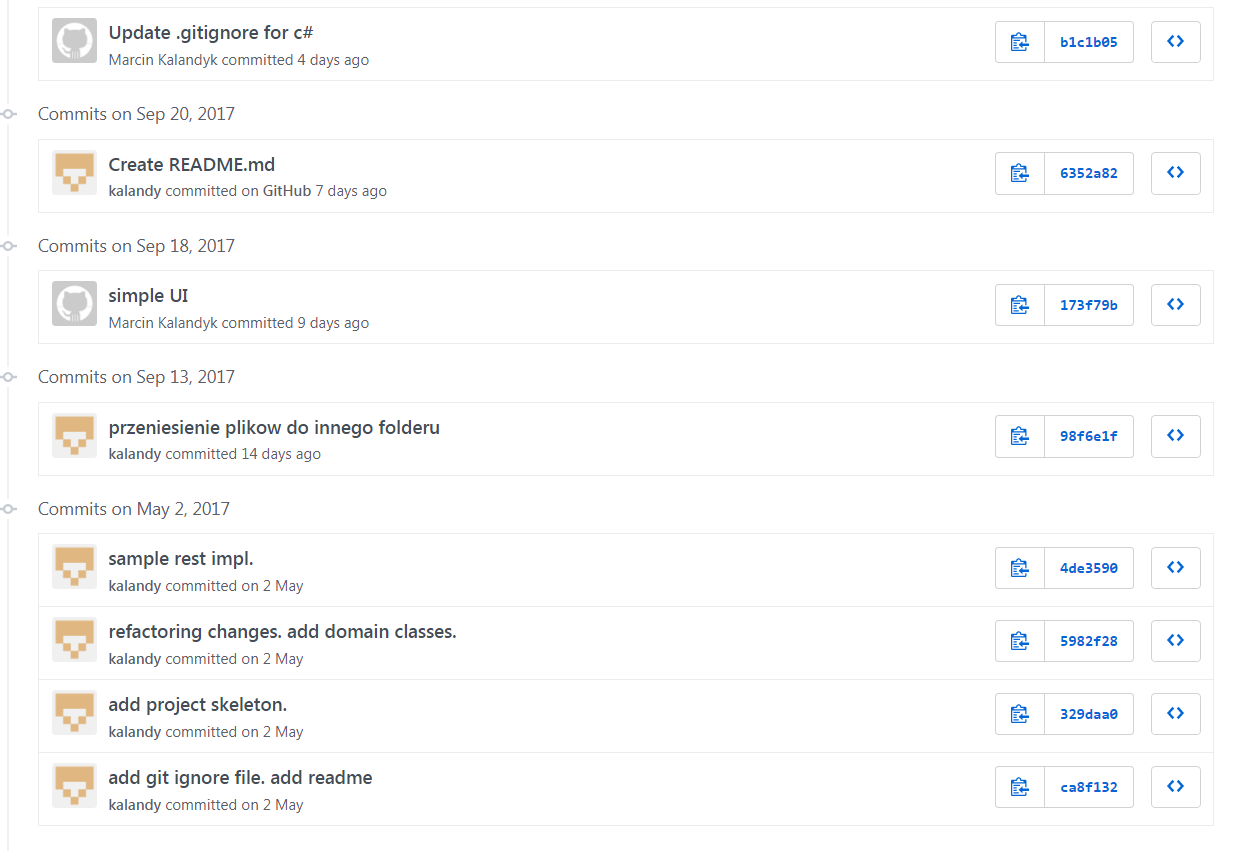
Czwarty zrzut wyświetla stronę internetową po kliknięciu w „link do strony z ofertą suplementu”

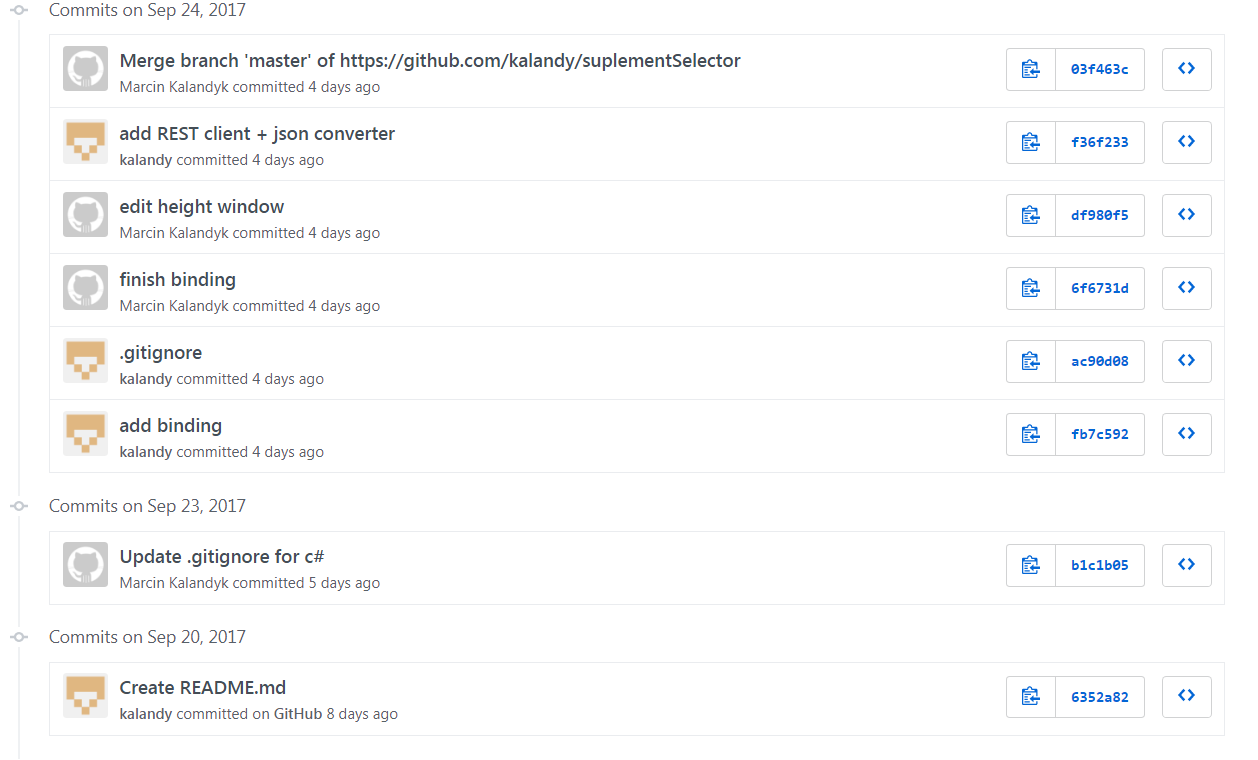


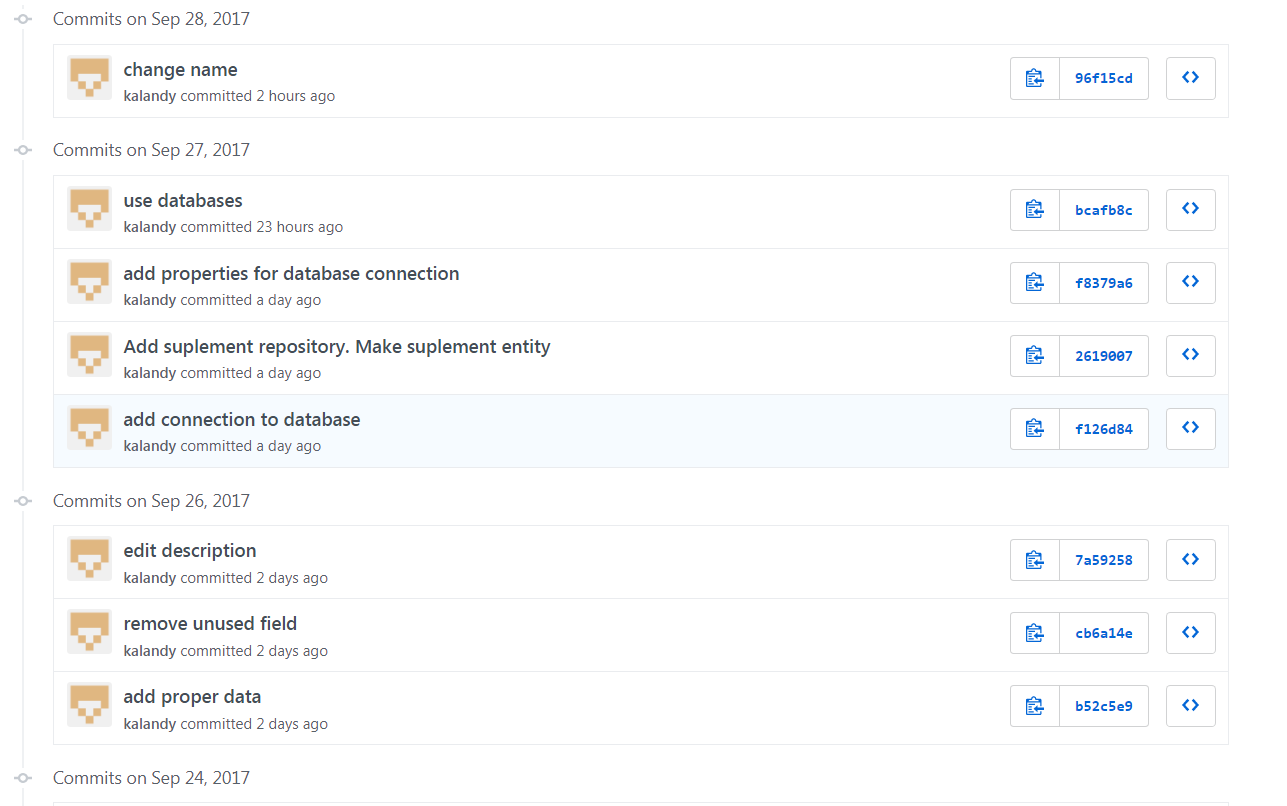
1. **Zrzut ekranu pokazujący historię wersjonowania projektu w systemie kontroli wersji.**

W projekcie głównie posługiwałem się systemem, który pozwala kontrolować wersje programu. W tym celu użyłem Github. Dzięki temu narzędziu posiadałem kopię projektu w bezpiecznym miejscu oraz możliwość ułatwionego sprawdzania zmian i postępu w tworzonym projekcie.

https://github.com/kalandy/suplementSelector

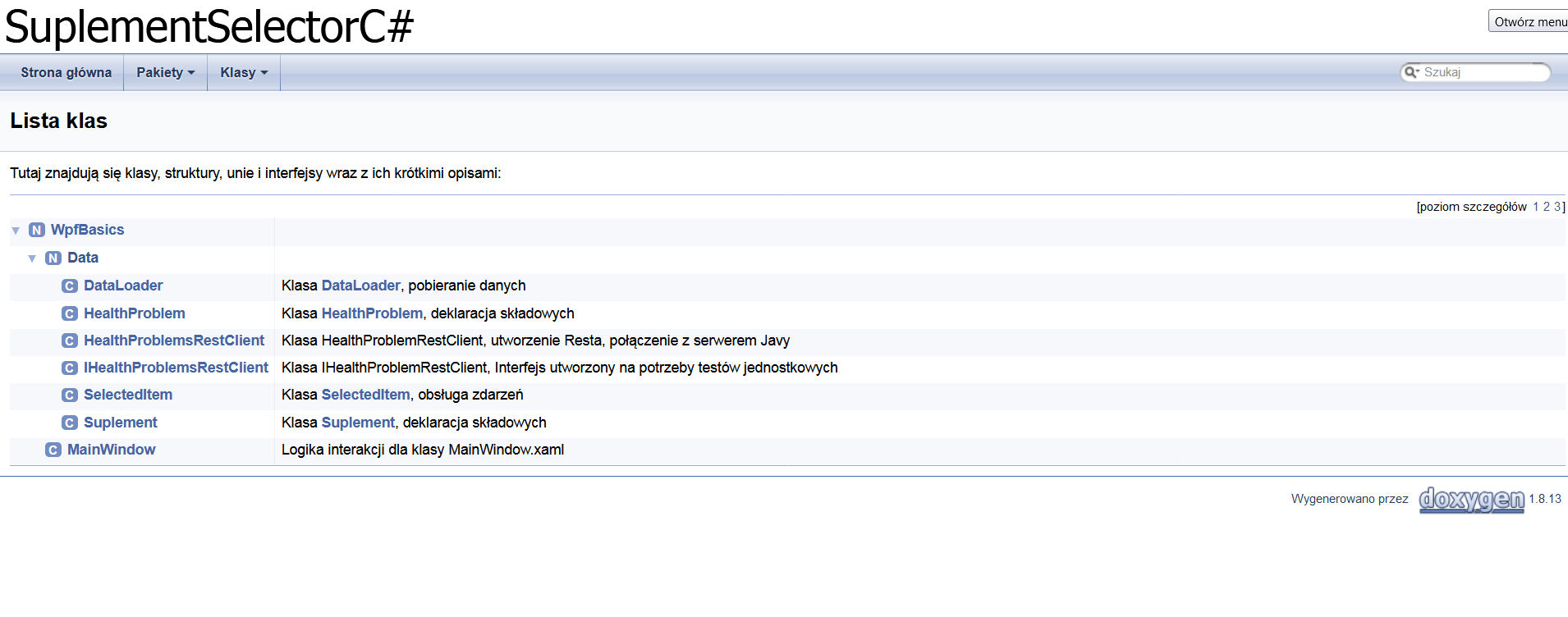


****

****

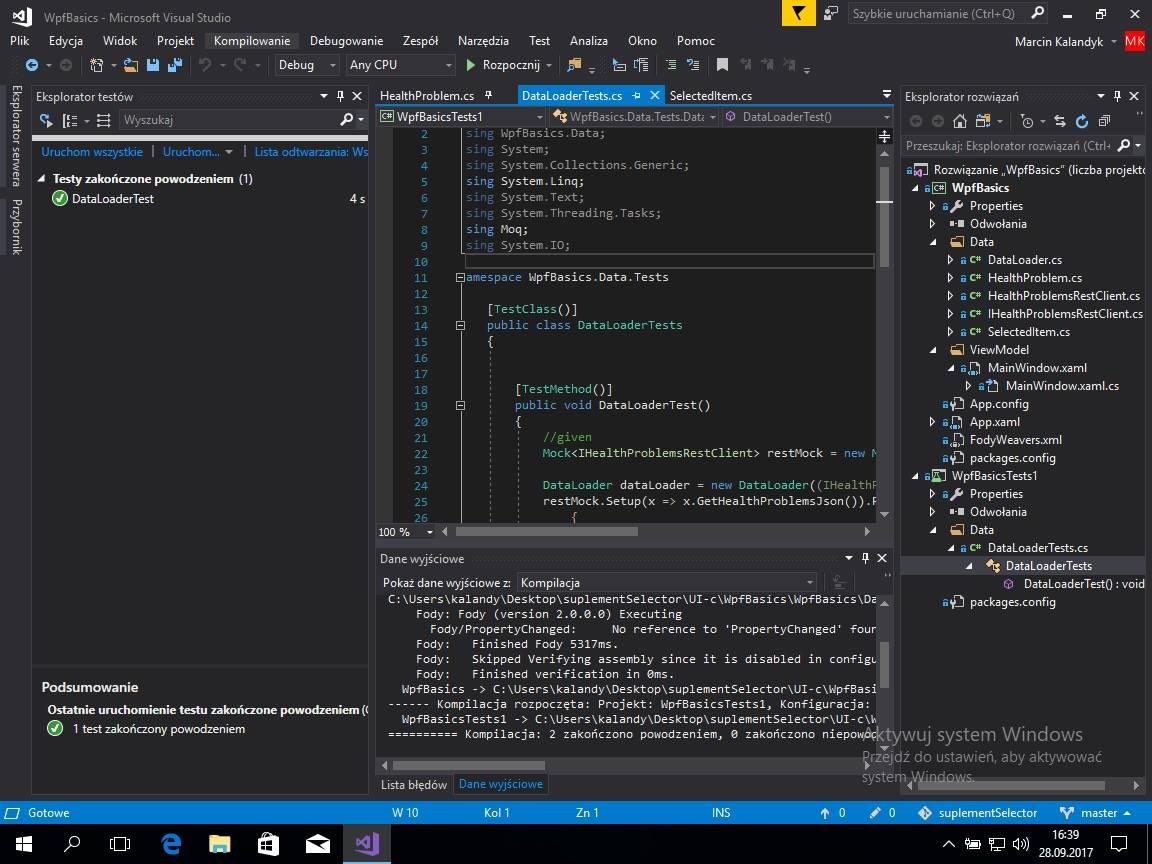
1. **Doxygen – zrzut ekranu pokazujący** klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

## 8a) C#

****

## 8b) JavaE:\Pobrane\22139830_1694930507207884_1933749315_o.png

1. **Testy jednostkowe – zrzut ekranu po wykonaniu testu jednostkowego**



1. **Materiały źródłowe**

**Poradnik do WPF UI**<https://www.youtube.com/watch?v=Vjldip84CXQ&list=PLrW43fNmjaQVYF4zgsD0oL9Iv6u23PI6M>

**Java**[**Kathy Sierra**](https://helion.pl/autorzy/kathy-sierra)**,**[**Bert Bates**](https://helion.pl/autorzy/bert-bates) „[Java Rusz głową!](https://www.ceneo.pl/Click/Offer/?e=aw38ouYWG0gXf3Li4we_wRuwlrNvB0SSv4J4o4VJKNu0EJ-fgZwtBLqgwPDbEKTWc3pB7F5KQgFJfnEAdoLY-r1tA4okfVa-5SXBNmUGVbC1JL3X6cw9sU_c1YhT36ArNNS5jaCnwLgg4NGxH4JMkCVYwhRBDeQhAQBit-IaLUOlUEzCWQUN3RO1WMBkqVTnLo-4921PY7ulUEzCWQUN3TKqJd_AWN7atYKmxbgYFlhmNODfFLZbE0vLmzQeHvKecDpautMdi_TMC4LmeZOmrA%3d%3d)”

**GitHub Tutorial**

<https://www.youtube.com/watch?v=0fKg7e37bQE>

**Rest**[**http://restsharp.org/**](http://restsharp.org/)

**JSON**

**[https://www.newtonsoft.com/json/help/html/SerializingJSON.htm  
  
Diagram Klas](https://www.newtonsoft.com/json/help/html/SerializingJSON.htmDiagram Klashttps://msdn.microsoft.com/pl-pl/library/hyxd8c85.aspxDiagram Klas – Intellij Ideahttps://stackoverflow.com/questions/8942751/use-intellij-to-generate-class-diagramSpringBoothttps://spring.io/guides/gs/spring-boot/)**

**[https://msdn.microsoft.com/pl-pl/library/hyxd8c85.aspx  
  
Diagram Klas – Intellij Idea  
https://stackoverflow.com/questions/8942751/use-intellij-to-generate-class-diagram](https://www.newtonsoft.com/json/help/html/SerializingJSON.htmDiagram Klashttps://msdn.microsoft.com/pl-pl/library/hyxd8c85.aspxDiagram Klas – Intellij Ideahttps://stackoverflow.com/questions/8942751/use-intellij-to-generate-class-diagramSpringBoothttps://spring.io/guides/gs/spring-boot/)**

**[SpringBoot](https://www.newtonsoft.com/json/help/html/SerializingJSON.htmDiagram Klashttps://msdn.microsoft.com/pl-pl/library/hyxd8c85.aspxDiagram Klas – Intellij Ideahttps://stackoverflow.com/questions/8942751/use-intellij-to-generate-class-diagramSpringBoothttps://spring.io/guides/gs/spring-boot/)** [https://spring.io/guides/gs/spring-boot/](https://www.newtonsoft.com/json/help/html/SerializingJSON.htmDiagram Klashttps://msdn.microsoft.com/pl-pl/library/hyxd8c85.aspxDiagram Klas – Intellij Ideahttps://stackoverflow.com/questions/8942751/use-intellij-to-generate-class-diagramSpringBoothttps://spring.io/guides/gs/spring-boot/)**SpringData**  
https://docs.spring.io/spring-data/jpa/docs/current/reference/html/