

Zadanie 12. Aplikacja wspomagająca pracę dietetyka, której zadaniem jest gromadzenie danych pacjentów oraz wyników pomiarów masy ciała i współczynnika BMI (*Body Mass Index*) wykonanych w kolejnych punktach czasowych. Dla wybranego pacjenta program ma generować wykresy zmienności obu tych parametrów i wyznaczać ich proste statystyki (min, max, średnia). Informacje należy przechowywać w relacyjnej bazie danych.

Specyfikacja aplikacji wspomagającej pracę dietetyka

Projekt z przedmiotu ISMED

Maria Poćwiardowska (297478) i Paweł Drewek (297466)

Wprowadzenie i cele

Aplikacja ma na celu wspomaganie pracy dietetyka poprzez gromadzenie danych pacjentów oraz wyników pomiarów masy ciała i współczynnika BMI (*Body Mass Index*) wykonanych w kolejnych punktach czasowych. Dla wybranego pacjenta program generuje wykresy zmienności obu tych parametrów i wyznacza ich proste statystyki (min, max, średnia). Informacje przechowywane są w relacyjnej bazie danych

Omówienie wymagań

Aplikacja powinna zapewnić następujące funkcje:

- Korzystanie z aplikacji przy pomocy interfejsu graficznego
- Gromadzenie danych pacjentów
- Generowanie wykresów zmienności i wyznaczanie statystyk danych (min, max, średnia)
- Wyświetlanie gromadzonych danych

Projekt aplikacji

Ogólny przegląd projektu aplikacji:

Aplikacja wykorzystuje interfejs graficzny, w celu obsługi aplikacji do:

- Dodawania pacjentów wraz z podstawowymi parametrami
- Usuwania wyznaczonych pacjentów
- Wyświetlania podstawowych danych pacjentów przy wykorzystaniu JTable
- Dodawaniu danych pomiarowych pacjentów
- Generowania wykresów oraz statystyk na podstawie danych

Klasy

Nazwa klasy	Opis	Komentarz
AppView	Interfejs graficzny	Klasa tworząca graficzny interfejs do obsługi aplikacji, opiera się o JTable
AppController	Kontroler	Klasa definiująca zachowanie aplikacji
AppModel	Przechowywanie danych	Klasa przechowująca i obsługująca bazę danych
PatientTableModel	Model tabeli	Klasa tworząca model tabeli wyświetlającej pacjentów
Patient	Pacjent	Klasa odpowiadająca fizycznemu pacjentowi
Measurememt	Pomiar	Klasa odpowiadająca pomiarowi
DialogLibrary	Komunikaty	Klasa zawierająca komunikaty i ostrzeżenia
AddPatientPanel	Dodawanie pacjenta poprzez interfejs graficzny	Klasa tworząca okno dodawania nowego pacjenta
AddDataPanel	Interfejs dodawania danych pomiarowych	Klasa tworząca okno dodawania danych pomiarowych
ChartWindow	Wykresy i dane	Klasa tworząca wykresy i przetwarzająca dane
VirtualDietician	Klasa główna	Tworzy obiekty AppController, AppView i AppModel oraz je łączy
ModelListener	Interfejs listenera modelu	„Nasłuchuje” zmiany w modelu
ViewListener	Interfejs listenera widoku	„Nasłuchuje” zmiany w widoku

Obsługa programu

Po uruchomieniu programu obsługujemy go za pomocą dostępnych przycisków. Należy wybrać przycisk:

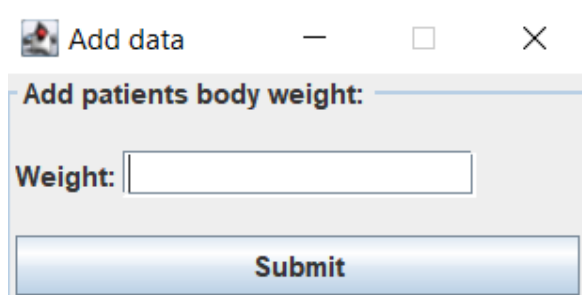
- Add Patient, aby dodać pacjenta – wprowadzenie imienia, numeru pesel oraz wzrostu

- Delete Patient, aby usunąć wybranego pacjenta
- Add Measurement, aby dodać nowy pomiar wagi wybranego pacjenta
- Show Stats, aby wyświetlić statystyki pomiarów wybranego pacjenta

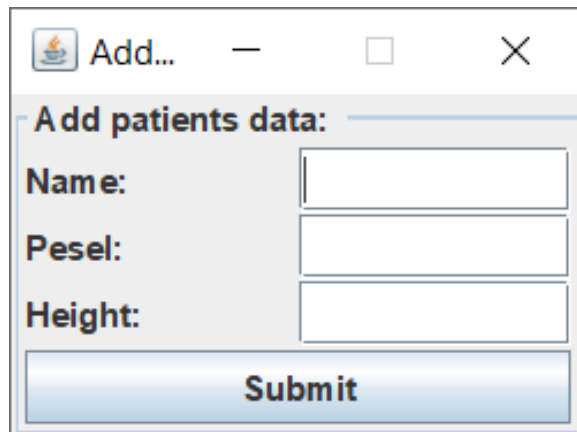
W przypadku przycisków Delete Patient, Add Measurement oraz Show Stats przed wciśnięciem przycisku należy wybrać odpowiedniego pacjenta poprzez kliknięcie jego danych z listy.

W celu zamknięcia programu należy użyć przycisku „Exit” dla poprawnego zamknięcia połączenia do bazy danych.

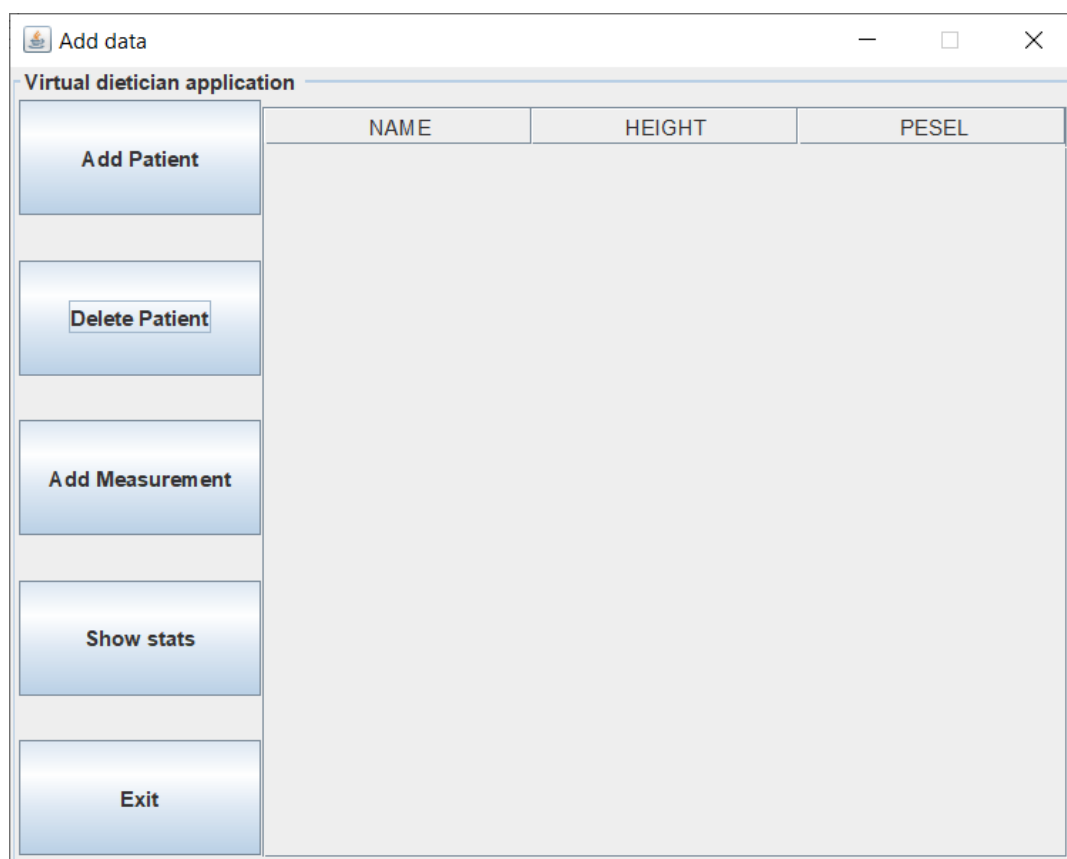
Wygląd interfejsu graficznego



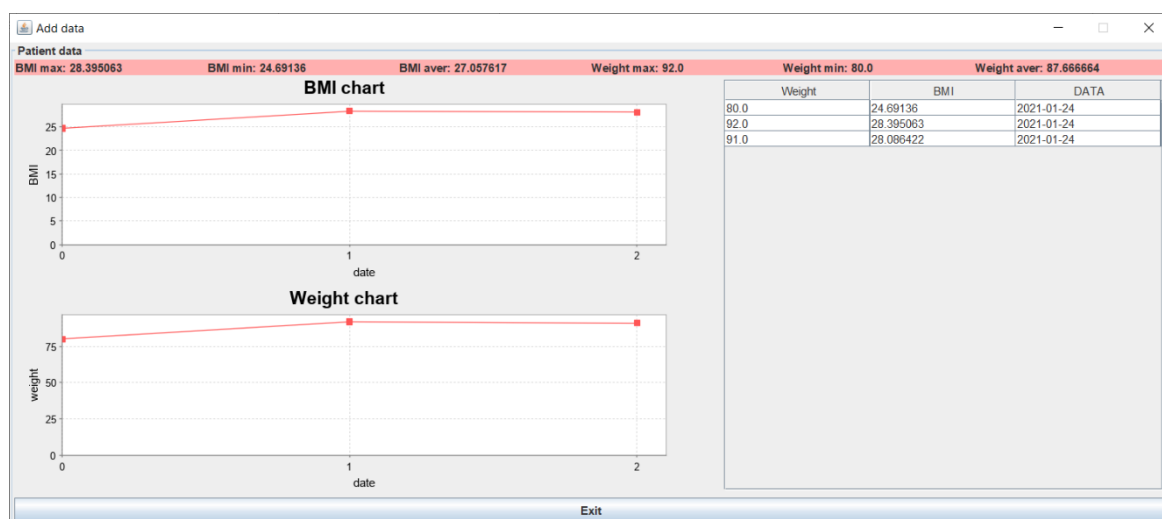
Obrazek 1: Okno do wprowadzania danych medycznych



Obrazek 2: Okno dodawania pacjenta

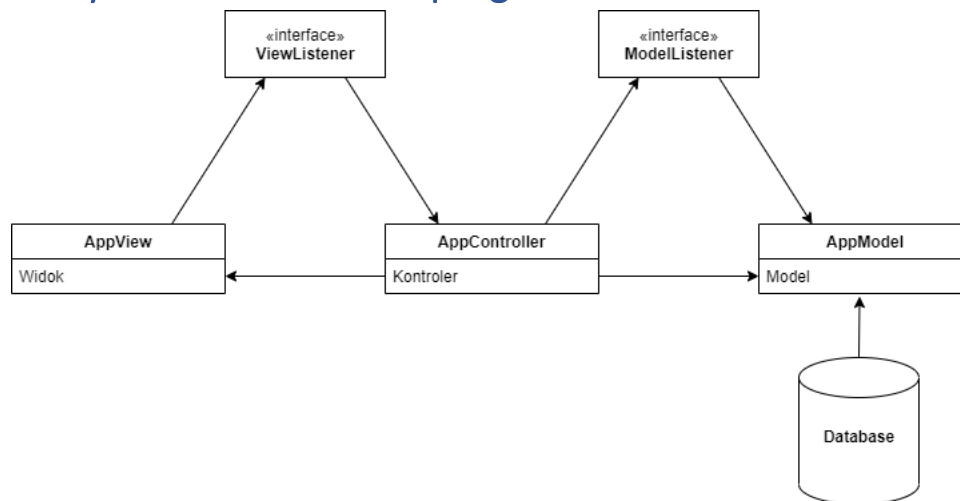


Obrazek 3: Główne okno aplikacji i jej elementy

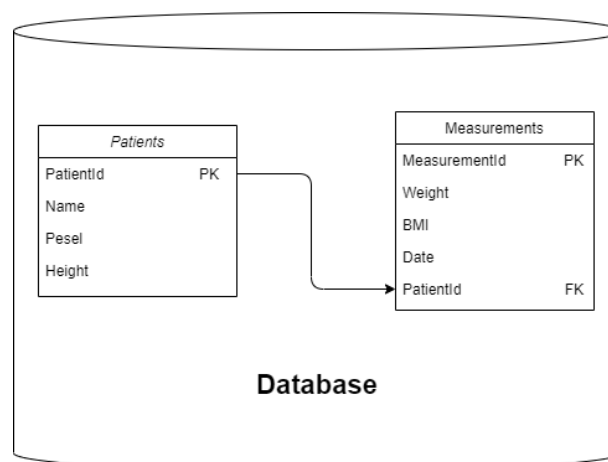


Obrazek 4: Okno wraz wykresami i danymi pomiarowymi

Uproszczony schemat działania programu



Uproszczony schemat bazy danych



Podział zadań

Maria – interfejs graficzny i częściowa obsługa poprzez kontroler. Stworzenie klasy **AddPatientPanel**, **AddDataPanel** oraz **ChartWindow**

Paweł – obsługa bazy danych i obsługa kontrolera. Stworzenie **DialogLibrary** oraz **Patient** i **Measurement**.

Do współpracy nad projektem zostało wykorzystane repozytorium GIT, niestety w przypadku problemów u Marii, commity były dodawane ręcznie przez WebGUI GitLaba.

Biblioteki

Do realizacji programu zostały wykorzystane biblioteki AWT oraz biblioteki Swing. Do obsługi bazy danych wykorzystujemy biblioteki **derby.jar**, **debryshared.jar**,

derbytools.jar. Do tworzenia wykresów wykorzystujemy biblioteki jcommon-1.0.0.jar oraz jfreechart1.0.0.jar.

Zależności

Aplikacja używa zależności zewnętrznych w postaci silnika bazodanowego działającego w trybie wbudowanym i bazy danych przechowywanej w katalogu projektu. Silnikiem tym jest Derby Apache w trybie embedded. Dodatkowo wykorzystywana jest biblioteka JFreeChart do tworzenia wykresów wagi oraz BMI. Powinna opierać się również na podstawowych funkcjach oferowanych przez platformę JAVA.

Wykonanie

Aplikacja wykonuje się w dowolnym katalogu komputera użytkownika, niezależnie od poziomu dostępu bieżącego użytkownika, o ile użytkownik ma uprawnienia do uruchamiania aplikacji Java i ma dostęp do odpowiednich plików wykonywalnych Java potrzebnych do uruchomienia aplikacji.