

Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych

Informatyczne Systemy Medyczne

System przechowywania kart badań kardiograficznych

Magdalena Kryczka 300458  
Joanna Rancew 300465

## Wstęp i cele

Zadaniem aplikacji jest zbieranie danych pacjentów oraz wyników pomiarów ciśnienia krwi i tętna wykonanych w kolejnych punktach czasowych. Program będzie generował wykresy zmienności mierzonych parametrów w czasie oraz będzie wyznaczał ich podstawowe statystyki. Zbierane pomiary oraz informacje o pacjentach będą przechowywane w relacyjnej bazie danych.

## Oczekiwania względem programu

Program powinien:

- Wczytywać dane pacjenta oraz sprawdzać poprawność wprowadzanych danych
- Przechowywać wczytane dane pacjentów
- Pobierać oraz zapisywać pomiary ciśnienia krwi i tętna w danych punktach czasowych
- Generować wykresy opracowane na podstawie zmierzonych wartości badanych parametrów
- Wyznaczać podstawowe statystyki na podstawie pobranych danych (wartość minimalna, wartość maksymalna oraz średnia)
- Przechowywać zebrane wyniki badań w relacyjnej bazie danych
- Obsługiwać wyjątki
- Kontaktować się z użytkownikiem poprzez intuicyjny interfejs

## Koncepcja programu

### Ogólna koncepcja projektu

Program będzie umożliwiał wprowadzenie danych pacjenta, a następnie wgranie dokonanych pomiarów badanych parametrów, czyli ciśnienia krwi i tętna. Na podstawie pobranych danych program będzie rysował przebiegi czasowe zmienności parametrów oraz wyliczał podstawowe statystyki takie jak wartość minimalna, maksymalna i średnia.

## Warstwy

### Warstwa usługi

Usługa będzie realizowana w postaci możliwości modyfikacji listy pacjentów, czyli wprowadzenia nowego pacjenta, modyfikacji danych pacjenta z bazy oraz usunięcia danych konkretnego pacjenta. Dodatkowo w programie będzie możliwość dodawania wyników badań pacjenta.

### Warstwa prezentacji

Warstwa prezentacji będzie służyła do komunikacji użytkownika z programem oraz prezentacji jego zasobów, czyli:

- Listy pacjentów oraz ich danych
- Wykresów narysowanych na podstawie zebranych danych
- Podstawowych statystyk wyliczonych na podstawie danych

## Wygląd aplikacji

Warstwa prezentacji oraz warstwa usługi realizowana będzie poprzez wyświetlanie odpowiednich elementów na ekranie. Podstawowym sposobem na komunikację programu z jego aktualnym użytkownikiem będzie stworzony przez nas GUI, czyli Graphical User Interface, w oparciu o bibliotekę Swing. Dopuszczamy także używanie dodatkowych komponentów dołączanych do projektu jako biblioteki zewnętrzne.

Podstawowymi cechami GUI będą:

- intuicyjność wykorzystywanych przycisków i poleceń
- pełna funkcjonalność

Plan projektu zakłada realizację interfejsu w postaci dwóch okien. Główne okno aplikacji będzie odpowiadało za kontakt użytkownika z programem oraz prezentację zgromadzonych danych, czyli listy pacjentów. Następnie po wybraniu konkretnego pacjenta, w osobnym oknie, będą prezentowane zebrane dane pomiarowe w postaci wykresu oraz wyliczonych statystyk.

Dodaj nowego pacjenta		Dane nowego pacjenta									
Zatwierdź		Imię		Nazwisko		Pesel		Miejsce urodzenia			
Zaktualizuj dane pacjenta		Nowe dane pacjenta									
Zatwierdź		ID		Imię		Nazwisko		Pesel		Miejsce urodzenia	
Usuń dane pacjenta				Dane do usunięcia							
Zatwierdź				ID							
Dodaj nowe pomiary		Dane nowego pomiaru									
Zatwierdź		ID		Ciśnienie		Tętno					
Lista pacjentów											
ID	Imię	Nazwisko	PESEL	Miejsce urodzenia	...						
1											
2											
3											
...											

Rysunek 1 Okno główne programu

Rysunek 2 Okno prezentujące pomiary konkretnego pacjenta

Zielone pola na schemacie będą uzupełniane przez użytkownika. Czerwone pola umożliwiają zatwierdzenie wykonanej czynności. Pozostałe pola będą stałe lub uzupełniane przez program. Naszym celem jest, aby w programie model danych został odseparowany od klas związanych z obsługą widoku oraz kontrolera.

## Ogólna struktura kodu

Kod programu zostanie napisany w języku JAVA. Struktura kodu cechować się będzie czytelnością, podziałem na klasy, odpowiednim formatowaniem oraz nazewnictwem klas, metod i zmiennych. Dodatkowo kod będzie opisany w komentarzach, by był on zrozumiały zarówno dla twórców, jak i osób niezaangażowanych w tworzenie tego programu.

## Klasy

Sekcja	Klasa	Opis
Sekcja główna programu	System	Klasa główna, zawierająca funkcję main, w której uruchamiany jest program
Sekcja interfejsu	AppView	Klasa odpowiadająca za warstwę prezentacji i wyświetlanie interfejsu
	AppController	Klasa odpowiedzialna za reakcje na zdarzenia pochodzące od GUI

Sekcja zarządzania danymi	PatientData	Klasa przechowująca dane pacjenta
	TestData	Klasa przechowująca zebrane dane pomiarowe z badania
	ModelData	Klasa odpowiada za udostępnianie i pośredniczenie w kontakcie z bazą danych. W podstawowej wersji programu może również przechowywać dane w prostszej strukturze np. liście. Klasa będzie zawierać zestaw metod, które pozwolą np. na pobranie danych pacjenta lub badania
Sekcja bezpośredniego kontaktu z bazą danych	DataBase	Klasa odpowiadająca za dostęp do bazy danych i wykonywanie odpowiednich operacji np. odczyt, zapis, usunięcie, modyfikacja korzystając z języka SQL w metodach napisanych w języku JAVA. Rozważamy utworzenie metod statycznych w tej klasie.
Dodatkowe klasy		Podczas tworzenia programu będziemy na bieżąco analizować nasze założenia i dostosowywać je do tworzenia programu w najkorzystniejszy sposób. Bierzemy pod uwagę stworzenie dodatkowych klas, jeśli będzie potrzeba.

### Biblioteki dodatkowe

Przygotowując się do pracy nad projektem wybrałyśmy poniższe biblioteki, które pozwolą nam na realizację zadania. Bierzemy pod uwagę możliwość rezygnacji z korzystania z wymienionych tu bibliotek lub wybrania nowych bibliotek, jeśli podczas pracy nad projektem uznamy to za bardziej korzystne rozwiązanie.

Wykorzystane będą:

- biblioteki klas standardowych Javy i Swing
- jfreechart, java.awt.Graphics2D lub inna biblioteka do rysowania wykresów
- java.sql, JDBC oraz baza danych ApacheDerby

Pracę nad całym projektem ułatwią biblioteki klas standardowych Javy. W celu zbudowania interfejsu graficznego zostanie użyta biblioteka Swing. Będziemy się starały unikać korzystania z kreatorów graficznych typu „drag nad drop”, jednak na tym etapie nie wykluczamy tej opcji. Do graficznej prezentacji zebranych danych w postaci wykresów zostaną użyte biblioteki takie jak jfreechart oraz java.awt.Graphics2D. Projekt zakłada użycie bazy danych, w tym celu wykorzystany zostanie silnik bazodanowy działający w trybie wbudowanym, jako integralna część aplikacji, a pliki bazy będą przechowywane w katalogu projektu. Wykorzystana zostanie baza danych ApacheDerby. Komunikacja naszej aplikacji z wykorzystywaną bazą danych będzie zrealizowana przy użyciu biblioteki java.sql oraz JDBC, czyli Java DataBase Connectivity. JDBC jest to interfejs umożliwiający aplikacjom napisanym w języku Java komunikację z bazami danych za pomocą języka SQL. Nie wykluczamy, że w trakcie realizacji projektu napotkamy problemy, które będą wymagały użycia dodatkowych bibliotek. Pozwoli nam to jednak zapoznać się z możliwościami środowiska programistycznego i poszerzyć swoje umiejętności. Wszelkie zmiany zostaną uwzględnione w finalnej dokumentacji projektu.

## Podział obowiązków w zespole

W ramach pracy nad projektem zajmujemy się:

	<b>Magdalena</b>	<b>Joanna</b>
Interfejs Graficzny	Kontroler obsługujący zdarzenia	Widok interfejsu graficznego
Wprowadzenie danych pacjenta	Funkcje wprowadzania z kontrolą błędów	Przygotowanie modelu danych pacjenta i badania
Prezentacja wyników działania programu	Wyznaczenie oraz prezentacja statystyk (min., max., średnia)	Prezentacja wyników w postaci wykresów
Baza danych	Poprawna komunikacja programu z relacyjną bazą danych	Stworzenie metod umożliwiających wykonanie działań związanych z danymi w klasie ModelData

Bierzemy pod uwagę wspieranie się w przydzielonych zadaniach lub modyfikację tego podziału, biorąc pod uwagę zdobywane przez nas na bieżąco umiejętności.

Do naszych wspólnych obowiązków należeć będą także:

- indywidualne doszkalanie się z materiałów wykładowych lub innych źródeł
- bieżące udostępnianie postępu pracy na wspólnym repozytorium
- bieżąca komunikacja w zespole oraz wzajemne raportowanie sobie pracy
- reagowanie na zmiany i dostosowywanie założonego planu działania

## Wykorzystanie repozytorium

Całość pracy będzie na bieżąco udostępniana na wspólnym repozytorium. Do udostępniania postępów pracy będziemy używać aplikacji GIT Bash lub aplikacji wbudowanej w środowisko programistyczne, w którym pisać będziemy kod naszego programu. Każde udostępnienie w repozytorium postępu pracy obarczone będzie krótkim komentarzem, który pozwoli zidentyfikować postęp (używanie funkcji „commit” z flagą „-m”). Podczas pracy nad programem będziemy w stałym kontakcie poprzez platformę MS Teams lub inne komunikatory, ale w razie potrzeby, będziemy korzystały także z funkcji komentarzy dostępnych przy aktualizacji repozytorium.

Repozytorium służyć będzie także jako awaryjne miejsce przechowywania aktualnego stanu pracy nad projektem, na wypadek awarii sprzętu członków zespołu projektowego.