

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 oraz częstości uderzeń serca. Program generuje wykresy zmienności mierzonych parametrów oraz wyznacza podstawowe statystyki. Zbierane pomiary oraz informacje o pacjentach przechowywane są 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 wyświetla listę pacjentów zgromadzoną w bazie danych oraz ich dane takie jak imię, nazwisko, pesel i miejsce urodzenia. Program umożliwia dodanie nowego pacjenta do bazy, aktualizację danych wybranego pacjenta figurującego już w bazie, usunięcie konkretnego pacjenta oraz dodanie wyników badania dla pacjenta wpisanego do bazy. Na podstawie zebranych danych program wylicza statystyki z zapisanych badań takie jak wartość minimalna, maksymalna i średnia każdego z analizowanych parametrów, czyli ciśnienia skurczowego, ciśnienia rozkurczowego oraz tętna. Program obrazuje również zmienność tych parametrów na przestrzeni zgromadzonych badań w postaci wykresów.

Warstwy

Warstwa usługi

Usługa jest realizowana w postaci możliwości modyfikacji listy pacjentów, czyli wprowadzenia nowego pacjenta, modyfikacji danych pacjenta z bazy oraz usunięcia danych konkretnego pacjenta. Dodatkowo w programie możliwe jest wprowadzenie i dodanie wyników badania dla dowolnego pacjenta z bazy.

Warstwa prezentacji

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

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

Wygląd aplikacji

Warstwa prezentacji oraz warstwa usługi jest realizowana poprzez wyświetlanie odpowiednich elementów na ekranie. Podstawowym sposobem na komunikację programu z jego aktualnym użytkownikiem jest stworzony przez nas Graficzny Interfejs Użytkownika (GUI - Graphical User Interface), który został zaprojektowany w oparciu o bibliotekę Swing.

Podstawowymi cechami naszego GUI są:

- intuicyjność wykorzystywanych przycisków i poleceń
- pełna funkcjonalność

W celu zachowania odpowiedniej intuicyjności i czytelności programu dla użytkownika, jego funkcje zostały rozdzielone na dwa okna. Główne okno aplikacji odpowiada za prezentację zgromadzonych danych, czyli listy pacjentów oraz kontakt użytkownika z programem, który jest możliwy w postaci modyfikacji listy pacjentów, wprowadzanie wyników nowych badań oraz wybranie pacjenta, którego badania go interesują. Następnie po wybraniu konkretnego pacjenta z listy, w osobnym oknie prezentowane są statystyki poszczególnych parametrów oraz wykresy obrazujące wszystkie zebrane dane pomiarowe.

System do badań

Dodaj nowego pacjenta

Zatwierdź Imię Nazwisko Pesel Miejsce urodzenia

Aktualizuj dane pacjenta

Zatwierdź ID Imię Nazwisko Pesel Miejsce urodzenia

Usuń dane pacjenta

Zatwierdź ID

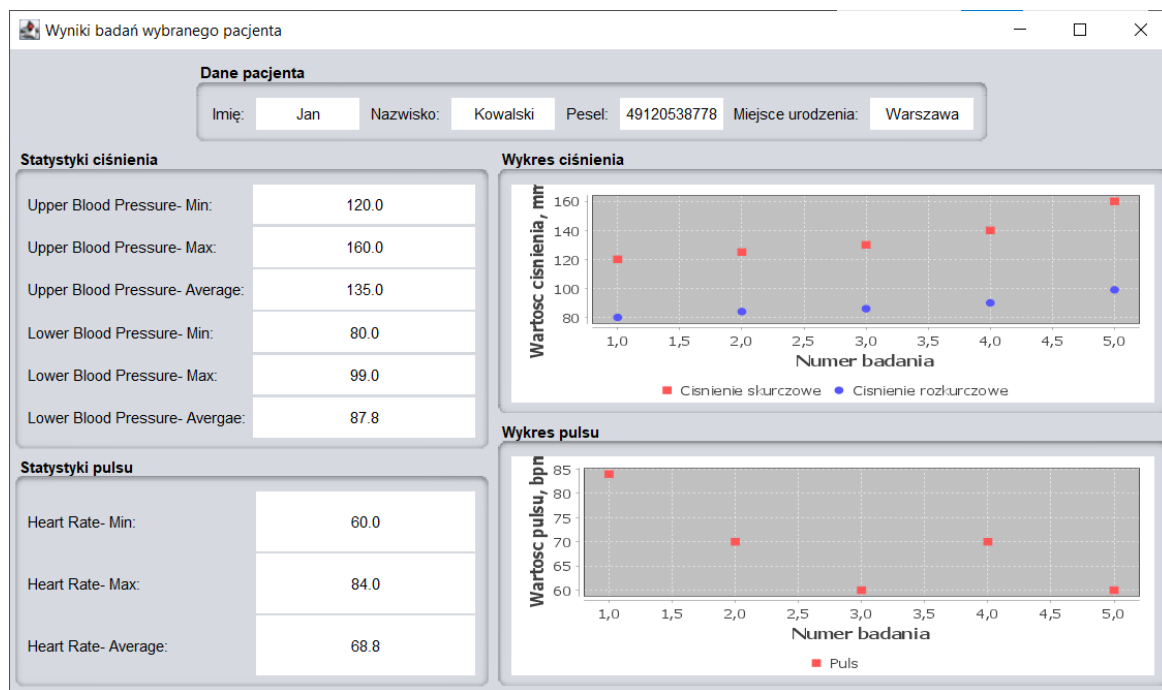
Dodaj nowy pomiar

Zatwierdź ID pacjenta Ciśnienie Tętno

Lista pacjentów do wyboru

ID	Imię	Nazwisko	PESEL	Miejsce urodzenia
1	Jan	Kowalski	49120538778	Warszawa
2	Adam	Nowak	59110534728	Warszawa
3	Karolina	Dąbek	68040634558	Gdynia

Rysunek 1 Okno główne programu



Rysunek 2 Okno prezentujące pomiary konkretnego pacjenta

Obsługa programu

Bezpośrednio po uruchomieniu programu na ekranie użytkownika wyświetlane jest okno główne naszej aplikacji przedstawione na Rysunku 1, które umożliwia użytkownikowi wykonanie następujących operacji:

- Dodanie nowego pacjenta do bazy

Aby poprawnie wykonać tą operację, najpierw użytkownik powinien uzupełnić wszystkie pola tekstowe w tej sekcji, wpisując kolejno imię pacjenta, jego nazwisko, pesel w postaci ciągu 9 cyfr oraz nazwę miejscowości, w której urodził się pacjent. Następnie użytkownik powinien zatwierdzić wprowadzone przez siebie dane, naciskając przycisk „Zatwierdź” w sekcji „Dodaj nowego pacjenta”.

- Modyfikacja/aktualizacja danych pacjenta z bazy

W celu zmiany i aktualizacji danych pacjenta będącego już w bazie, którego dane wyświetlane są w tabeli, użytkownik proszony jest o podanie identyfikatora pacjenta, którego dane mają być zmodyfikowane, a następnie wprowadzenie nowych danych w kolejne pola tekstowe, tak aby podać wszystkie potrzebne informacje. Ważne jest uzupełnienie wszystkich atrybutów potrzebnych do aktualizacji danych pacjenta w bazie. Po uzupełnieniu wszystkich pól tekstowych użytkownik powinien potwierdzić swoją wolę modyfikacji danych pacjenta o wskazanych identyfikatorze poprzez naciśnięcie przycisku „Zatwierdź” w sekcji „Aktualizuj dane pacjenta”

- Usunięcie pacjenta oraz jego danych z bazy

Operacja usunięcia pacjenta z bazy wymaga od użytkownika podania tylko jednej informacji, czyli identyfikatora wybranego pacjenta. Oczywiście warunkiem poprawnego wykonania tej operacji jest podanie ID pacjenta, które istnieje i był wcześniej zapisany w bazie. Użytkownik podaje ID w polu tekstowym, a następnie potwierdza tą informację za pomocą przycisku „Zatwierdź” w sekcji „Usuń dane pacjenta”.

- Dodanie nowego badania

Dodanie nowego badania odbywa się poprzez wprowadzenie ID pacjenta w odpowiednie pole tekstowe oraz wpisanie wyników pomiarów badania, czyli ciśnienia

krwi oraz pulsu. Ciśnienie krwi powinno być podane w postaci dwóch liczb, rozdzielonych ukośnikiem („/”). Jako pierwszą wartość należy podać ciśnienie skurczowe (wyższe), a jako drugą ciśnienie rozkurczowe (niższe). Puls powinien być wprowadzony tradycyjnie jako jedna liczba. Po wprowadzeniu danych użytkownik powinien potwierdzić wprowadzone dane poprzez naciśnięcie przycisku „Zatwierdź” w sekcji „Dodaj nowy pomiar”.

- Wyświetlenie historii i statystyk badań wybranego pacjenta

W celu objerzenia historii zapisanych badań pacjenta oraz statystyk wyliczonych na podstawie zgromadzonych pomiarów użytkownik powinien wybrać interesującego do pacjenta z tabeli wyświetlanej w dolnej części okno głównego poprzez naciśnięcie na odpowiedni wiersz tabeli. Po wykonaniu tej akcji na ekranie wyświetli się drugie okno, którego przykład pokazany jest na Rysunku 2. W nim prezentowane są zgromadzone dane oraz ich analiza. W górnej części widnieją dane wybranego pacjenta, czyli imię, nazwisko, pesel i miejsce urodzenia, a poniżej przedstawione są wyniki badań. Po lewej stronie użytkownik widzi wyliczone statystyki ciśnienia i pulsu. Statystyki obejmują ciśnienie skurczowe, zwane „górnym”, ciśnienie rozkurczowe, zwane „dolnym” oraz częstość skurczów serca, inaczej puls lub tętno. Podawane są trzy wartości każdego z tych parametrów, na które składają się wartość minimalna, wartość maksymalna i wartość średnia, liczona jako średnia arytmetyczna ze wszystkich zgromadzonych pomiarów. Po prawej stronie zaprezentowane są wykresy, narysowane na podstawie zebranych danych. Na górnym wykresie, obrazującym zmienność ciśnienia, widoczne są dwie charakterystyki, które odpowiadają analogicznym składowym ciśnienia: charakterystyka górna- ciśnienie skurczowe, charakterystyka dolna- ciśnienie rozkurczowe. Na dolnym wykresie obrazowana jest zmienność tętna na przestrzeni zgromadzonych badań. Okno to służy jedynie do prezentacji danych, a użytkownik nie może wykonać w nim żadnych operacji. Okno zamykane jest za pomocą przycisku „X” znajdującego się w jego prawym górnym rogu.

Przyciski służące do zatwierdzania wprowadzonych danych, które potwierdzają wolę użytkownika do wykonania konkretnej operacji znajdują się po lewej stronie każdej z sekcji. Przechodzenie między kolejnymi polami tekstowymi (w ramach jednej sekcji) jest możliwe poprzez użycie klawisza tabulacji 'Tab' lub tradycyjnie poprzez ustawienie kursora w polu tekstowym za pomocą myszy.

Ogólna struktura kodu

Kod programu zostanie napisany w języku JAVA. Staraliśmy się, aby struktura kodu cechowała się czytelnością, rozumiałym podziałem na pakiety i klasy, odpowiednim formatowaniem oraz nazewnictwem klas, metod i zmiennych. Dodatkowo kod opisany jest w komentarzach, tak aby był on zrozumiały zarówno dla twórców, jak i osób niezaangażowanych w tworzenie tego programu.

W całym kodzie staraliśmy się także przyjąć stałe nazewnictwo: `NazwaKlasy`, `nazwaMetody` oraz `nazwaZmiennej`. Stosowaliśmy także skróty oraz „_”, co ułatwiało często zrozumienie kodu.

Uruchomienie i kompilacja aplikacji

Niezbędnym elementem procesu uruchomienia aplikacji są:

- zamknięcie dostępu do bazy danych w trybie embedded, w tym także poprzedniej sesji naszej aplikacji
- dołączenie odpowiednich bibliotek do projektu

Klasy

Sekcja	Klasa	Opis
Sekcja główna programu	AppSystem	Klasa główna, zawierająca funkcję main, w której uruchamiany jest program
Sekcja kontrolera	AppController	Klasa odpowiedzialna za reakcje na zdarzenia pochodzące od GUI
Sekcja interfejsu	AppView	Klasa odpowiadająca za warstwę prezentacji i wyświetlanie interfejsu
	JTableModel	Klasa odpowiedzialna za model wyświetlanej tabeli pacjentów
Sekcja modelu danych	AppModel	Klasa przechowująca model aplikacji
	PatientData	Klasa przechowująca dane pacjenta
	TestData	Klasa przechowująca dane pomiarowe z badania
Sekcja bezpośredniego kontaktu z bazą danych	DataBase	Klasa odpowiadająca za komunikację z bazą danych, dostęp do bazy danych i wykonywanie odpowiednich operacji np. odczyt, zapis, usunięcie, modyfikacja. Klasa korzysta z języka SQL w metodach napisanych w języku JAVA.

Biblioteki dodatkowe

Wykorzystywane są:

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

Pracę nad całym projektem ułatwiły nam biblioteki klas standardowych Javy. W celu zbudowania interfejsu graficznego wykorzystana została biblioteka Swing. Do graficznej prezentacji zebranych danych w postaci wykresów użyto biblioteki jfreechart. W celu połączenia programu z wykorzystywaną bazą danych użyliśmy silnika bazodanowego działającego w trybie wbudowanym „embedded”, czyli jako integralna część aplikacji. Pliki bazy przechowywane są bezpośrednio w katalogu projektu. Wykorzystana została baza danych ApacheDerby. Komunikacja naszej aplikacji z wykorzystywaną bazą danych została zrealizowana przy użyciu biblioteki java.sql oraz JDBC, czyli Java DataBase Connectivity.

Dokumentacja kodu

Kod jest rzetelnie skomentowany z uwagami, które pozwoliły na stworzenie przejrzystego JavaDocs. Komentarze uzupełniałyśmy na bieżąco oraz na koniec, po podsumowaniu pracy.

JavaDocs jest dostępne na repozytorium: `./doc/index.html`

Działanie programu

Działania związane ze zdarzeniami obsługiwane są w modelu MVC, który pozwala na przejrzysty sposób działania. Dodatkowo klasa AppModel pośredniczy w komunikacji z bazą danych.

Baza danych jest w pierwszej kolejności czyszczona (usuwane są tabele Patients oraz Tests), ze względu na utrudnienia w poprawnym zamykaniu i problem z generowanymi indeksami rekordów po ponownym uruchomieniu. Zidentyfikowałyśmy przyczynę tego problemu – niepoprawną obsługę operacji zamykania programu. Niestety nie udało nam się zapobiec wystąpieniu tych błędów.

Program zawiera skuteczną ochronę przed wprowadzaniem błędnych danych oraz błędami programu. Komunikaty, które powinien zobaczyć użytkownik wysyłane są jako MessageDialog, natomiast pozostałe błędy wysyłane są na konsolę. Program nie chroni przed problemem używania bazy danych przez inny program lub poprzednią sesję, dlatego niezbędne jest zamknięcie działających programów z użyciem bazy danych, przed uruchomieniem aplikacji.

Szczegółowy sposób działania programu skomentowany jest w kodzie oraz w dokumentacji.

Podział obowiązków w zespole

W ramach pracy nad projektem podzieliliśmy się początkowo:

	Magdalena	Joanna
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

Do naszych wspólnych obowiązków należało 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

Rzeczywisty podział pracy, odbiegał nieco od założonego planu. Zdecydowałyśmy się zmienić go ze względu na spójność pracy (np. tworzenie kontrolera po stworzeniu interfejsu, wymagało edycji interfejsu). Dodatkowo nakład pracy, który był na poszczególnych etapach, a także nasz harmonogram pracy spowodował zmiany w podziale zadań.

	Magdalena	Joanna
Interfejs Graficzny	Kontroler obsługujący zdarzenia	Stworzenie głównego interfejsu graficznego
	Poprawa głównego interfejsu graficznego oraz stworzenie widoku prezentacji badań	
Wprowadzenie danych pacjenta	Dopracowanie modelu danych pacjenta i badania	Przygotowanie modelu danych pacjenta i badania
		Funkcje wprowadzania z kontrolą błędów
Prezentacja wyników działania programu	Wyznaczenie oraz prezentacja statystyk (min., max., średnia)	Prezentacja wyników w postaci wykresów
Baza danych	Stworzenie metod umożliwiających wykonanie działań związanych z danymi w klasie DataBase	Poprawna komunikacja programu z relacyjną bazą danych

Wykorzystanie repozytorium

Podczas tworzenia projektu nasza praca była na bieżąco udostępniana na wspólnym repozytorium. Do udostępniania postępów pracy używana była aplikacja GIT Bash. Staraliśmy się, aby każde udostępnienie postępu pracy w repozytorium obarczone było krótkim komentarzem, który pozwoli zidentyfikować postęp. Podczas pracy nad programem byliśmy w stałym kontakcie poprzez platformę MS Teams oraz inne komunikatory, aby możliwie szybko i skutecznie rozwiązywać napotykaną trudności.

Zastosowaliśmy też branch, jednakże ze względu na spójną i równoległą pracę nad podobnymi elementami projektu, pracowaliśmy przede wszystkim na przemian.

Podsumowanie

Praca nad projektem umożliwiła nam zdobycie nowych umiejętności, dokładniejsze poznanie języka oraz jego pracy z bazą danych w trybie „embedded”. Ponadto poznaliśmy lepiej charakter pracy nad projektem w zespole, trudności z tym związane, a także metody stosowane we wspólnej pracy z wykorzystaniem repozytorium. Staraliśmy się także pracować regularnie, by w ostatnich dniach dokończyć jedynie te najważniejsze aspekty.

Końcowa wersja aplikacji spełnia prawie wszystkie założenia postawione we wstępnej dokumentacji.