

Spis treści

1.	Wstęp	4
2.	Cel i kontekst projektu	5
2.1	Cel projektu	5
2.2	Kontekst projektu	5
2.2.1	Skala PAD	5
2.2.2	Wizualizacja stanów emocjonalnych	6
3.	Projekt systemu	7
3.1	Źródła wymagań	7
3.2	Koncepcja techniczna: założenia	7
3.2.1	Środowisko eksperymentu	7
3.2.2	Moduły systemu	7
3.2.3	Technologie	7
3.2.4	Inne	7
3.3	Słownik pojęć	8
3.4	Specyfikacja Wymagań Systemowych	9
3.4.1	Priorytety wymagań	9
3.4.2	Wymagania funkcjonalne: dane	9
3.4.3	Wymagania funkcjonalne: serwer	10
3.4.4	Wymagania funkcjonalne: wizualizator	14
3.4.5	Wymagania pozafunkcjonalne	26
3.5	Koncepcja techniczna	28
3.5.1	Komunikacja	28
3.5.2	Interfejs i wizualizacja danych	28
3.6	Projekt bazy danych	29
3.6.1	Diagram relacyjny bazy danych	29
3.6.2	Opis tabel	30
3.6.3	Opis relacji	31
3.7	Projekt interfejsu użytkownika	33

3.7.1	Tryb rzeczywisty jednokanałowy	33
3.7.2	Tryb rzeczywisty, zminimalizowany: etykiety.....	34
3.7.3	Tryb rzeczywisty, zminimalizowany: radar	34
3.7.4	Tryb analizy jednokanałowy	35
3.7.5	Tryb analizy wielokanałowy.....	36
3.7.6	Menu użytkownika	37
4.	Podręcznik użytkownika	38
4.1	Wymagania.....	38
4.2	Struktura folderów aplikacji	38
4.3	Konfiguracja: uwagi	38
4.4	Konfiguracja serwera.....	38
4.5	Konfiguracja wizualizatora	40
4.5.1	Konfiguracja etykiet.....	41
4.6	Konfiguracja bazy danych.....	42
4.7	Uruchomienie aplikacji.....	43
5.	Podręcznik programisty: rozwój aplikacji.....	44
5.1	Wprowadzenie	44
5.2	Notacja stempli czasowych	44
5.3	Tworzenie źródła danych dla serwera.....	44
5.3.1	Format danych.....	44
5.3.2	Komunikacja z serwerem.....	45
5.4	Architektura serwera.....	46
5.4.1	ConnectionHandler.....	46
5.5	Architektura wizualizatora	46
5.5.1	Dane.....	46
5.5.2	Interfejs użytkownika: kontrolki.....	47
5.5.3	Interfejs użytkownika: widoki.....	48
6.	Raport Końcowy	49
6.1	Zespół projektowy	49
6.2	Temat projektu	49
6.3	Kontekst projektu.....	49
6.4	Osiągnięte rezultaty	50

6.5	Proces realizacji projektu	50
6.6	Dokumentacja	50
6.7	Zmiany w trakcie projektu.....	50
7.	Podsumowanie	52
Bibliografia.....		53
Wykaz tabel		53
Wykaz rysunków.....		53

1. Wstęp

Stanem emocjonalnym nazywamy zbiór cech opisujących emocje danej osoby w konkretnym czasie lub jego przedziale. Zbiór cech może być definiowany na wiele sposobów, w zależności od celu i założeń konkretnego badania. Niniejszy projekt skupia się przede wszystkim na afektywnych stanach emocjonalnych, tzn. stymulowanych wpływem bodźców zewnętrznych.

Badanie i analiza stanu emocjonalnego posiada szerokie zastosowanie w badaniach psychologicznych, medycznych i naukowych.

Wykorzystanie badań stanów emocjonalnych obejmuje zastosowania z dziedziny przetwarzania emocjonalnego (ang. Affective Computing), takimi jak systemy inteligentne, edukacja wspomagana komputerowo (ang. Technology-Enhanced Learning), badania ergonomii czy nawet wsparcia w różnych fazach procesów projektowania i marketingu produktów i usług poprzez badanie reakcji potencjalnych klientów.

Odpowiednio skonstruowane narzędzie może dostarczyć wymierne dane o stanach emocjonalnych, dzięki czemu możliwe jest przeprowadzenie badań w szerszym kontekście, łącznie z trudnym technicznie porównywaniem wyników badań czy diagnozy ludzkich emocji i zachowań w określonych sytuacjach.

Ważnym czynnikiem w dziedzinie rozpoznawania emocji jest przetwarzanie i wizualizacja zebranych danych. Celem wizualizacji jest pokazanie aktualnego stanu emocjonalnego i jego zmiany w sposób, który jest dopasowany do rodzaju badania i percepcji osoby je przeprowadzającej.

Celem projektu jest stworzenie narzędzia wspomagającego przeprowadzanie eksperymentów z dziedziny badań stanów emocjonalnych poprzez utrwalanie zebranych danych i wizualizację wyników.

W rozdziale 2. przedstawione zostały cele oraz kontekst projektu. Szczegółowe wymagania dot. systemu oraz jego projekt techniczny znajdują się w rozdziale 3. W rozdziale 4. opisany został sposób konfiguracji oraz użytkowania aplikacji. Szczegółów implementacji pomocne przy dalszym rozwoju aplikacji znajdują się w rozdziale 5. Rozdział 6. zawiera raport końcowy podsumowujący pracę nad projektem, jego wyniki oraz wnioski.

2. Cel i kontekst projektu

2.1 Cel projektu

Celem projektu jest stworzenie aplikacji służącej do wizualizacji oraz analizy stanów emocjonalnych osób do badań naukowych oraz medycznych.

2.2 Kontekst projektu

2.2.1 Skala PAD

Opis matematycznego modelu PAD do określania stanów emocjonalnych osób badanych został zaproponowany w 1977 roku przez Alberta Mehrabiego oraz James'a A. Russela i wspólnie przez nich rozwijany.

Model podaje stan emocjonalny przy pomocy trzech metryk:

- P: Positiveness (ew. Pleasure) – wymiar przyjemności odczuwanej przez osobę badaną wyrażony w skali $\langle -1; 1 \rangle$, gdzie -1 oznacza odczuwanie wysokiej nieprzyjemności, przykrości, 0 jest wartością neutralną, natomiast 1 oznacza wysokie zadowolenie osoby badanej.
- A: Arousal – pobudzenie osoby badanej wyrażone w skali $\langle -1; 1 \rangle$, gdzie -1 oznacza całkowity brak pobudzenia, 0 jest wartością neutralną, natomiast 1 , wysokie pobudzenie osoby badanej.
- D: Dominance – dominacja osoby badanej wyrażona w skali $\langle -1; 1 \rangle$, gdzie -1 oznacza wysokie wycofanie (ucieczkę), 0 jest wartością neutralną, natomiast 1 oznacza wysoki stopień dominacji i pewności siebie (walki).

Algorytmy rozpoznające stan emocjonalny mogą zbierać dane z ograniczoną pewnością co do trafności wyniku, dlatego też każdej z powyższych metryk towarzyszy współczynnik pewności wyrażony w skali $\langle 0; 1 \rangle$, gdzie 0 oznacza zerową, natomiast 1 całkowitą pewność dot. trafności pomiaru.

2.2.2 Wizualizacja stanów emocjonalnych

Potrzeba wizualizacji

Stany emocjonalne reprezentowane przy pomocy modelu PAD są czytelne i łatwe do przetwarzania przez programy komputerowe, jednak dla człowieka, w szczególności słabo obeznanego z pracą na komputerze, wizualizacja punktu w przestrzeni PAD może być nieczytelna.

Na potrzeby wizualizatora należy więc określić szereg metod ułatwiających użytkownikowi aplikacji zrozumienie i analizę zebranych podczas eksperymentu danych.

Etykietowanie

Popularnym sposobem wizualizacji jest użycie słownych etykiet stanów emocjonalnych. Efekt jest osiągany przez przypisanie poszczególnym zakresom wartości skali PAD konkretnych etykiet i wyświetlanie ich dla kolejnych danych napływających do aplikacji.

Wykresy

Wykorzystanie etykiet znacząco poprawia czytelność danych, jednak stanowi metodę o wysokiej abstrakcji i oderwaniu od zebranych danych. Na podstawie samych etykiet trudniej jest śledzić zmieniające się podczas eksperymentu wartości poszczególnych metryk P, A oraz D.

Do analizy dynamiki zmian wartości metryk oraz dla ułatwienia analizy zapisu eksperymentu naturalną opcją jest wizualizacja wartości poszczególnych metryk na wykresach przedstawiających wartości zebranych danych w kolejnych punktach czasu.

Kolory

Wykorzystanie kolorów przy wizualizacji stanów emocjonalnych wpływa pozytywnie na czytelność danych. Wykorzystać można np. kolory zielony i czerwony, popularnie kojarzone z dobrymi i złymi emocjami, szarości do pokazywania stanów neutralnych, natomiast manipulacja jaskrawością koloru (np. od zielonego, przez bładozielony do białego) pozwala na oddanie pewności pomiaru.

3. Projekt systemu

3.1 Źródła wymagań

Podstawowym dokumentem opisującym założenia realizacji projektu jest Raport Techniczny Wizualizatora Stanu Emocjonalnego Dla Eksperymentów Medycznych i Badawczych, autorstwa dr inż. Agnieszki Landowskiej.

Raport stanowi podstawę do specyfikacji wymagań funkcjonalnych jak i pozafunkcjonalnych. Precyzuje także założenia dot. kształtu aplikacji, wybranych metod prezentacji danych oraz interfejsu użytkownika.

3.2 Koncepcja techniczna: założenia

3.2.1 Środowisko eksperymentu

Osoba badana znajduje się przy stanowisku z komputerem, na którym jest zainstalowana aplikacja zbierająca dane dot. jego aktualnego stanu emocjonalnego.

Opiekun eksperymentu znajduje się przy stanowisku komputerowym, na którym zainstalowana jest aplikacja na bieżąco wizualizująca zebrane dane.

Należy także zapewnić późniejszą możliwość analizy danych zebranych podczas eksperymentu.

3.2.2 Moduły systemu

System użyty do zbierania oraz wizualizacji danych musi składać się z trzech części:

- Aplikacja kliencka – zbiera informacje odnośnie stanów emocjonalnych osoby badanej i reprezentuje zebrane dane w modelu PAD. Wszystkie zebrane informacje wysyła do serwera
- Serwer – zapewnia komunikację pomiędzy aplikacją kliencką i wizualizatorem. Stanowi też warstwę persystencji danych – zapisuje wszystkie informacje o eksperymencie w celu ich odtworzenia dla późniejszych analiz.
- Wizualizator – wizualizuje dane zbierane podczas eksperymentu.

Narzędzia zbierające informacje o stanie emocjonalnych są wytwarzane niezależnie i ich wytworzenie nie jest celem tego projektu. Należy jednak sprecyzować, w jaki sposób i w jakim formacie mają one dostarczać dane do serwera.

3.2.3 Technologie

Aplikacja kliencka jest wytworzona w technologii Java. Aby ograniczyć ew. problemy z brakiem kompatybilności modułów oraz ułatwić integrację systemu, zarówno serwer jak i wizualizator powinny być wykonane w tej samej technologii.

3.2.4 Inne

Wizualizator powinien być aplikacją okienkową działającą w środowisku Windows.

3.3 Słownik pojęć

Klient	Aplikacja dostarczająca dane eksperymentu do serwera.
Wizualizator	Aplikacja wizualizująca stan emocjonalny na podstawie danych z serwera.
Użytkownik	Użytkownik aplikacji.
API	Interfejs o określonej funkcjonalności zaimplementowany w danym module.
Eksperyment	Szereg sesji o wspólnym scenariuszu, różniących się metodą oceny danych, przeprowadzanych na osobie badanej mający na celu zebranie danych o jej stanach emocjonalnych.
Sesja eksperymentu	Pojedyncza sesja eksperymentu, zawierająca dane o stanach emocjonalnych zebranych przy pomocy jednej metody ich oceny.
Wizualizacja jednokanałowa	Wizualizacja danych jednej sesji eksperymentu (zebranych przy pomocy jednej metody).
Wizualizacja wielokanałowa	Wizualizacja danych wielu sesji eksperymentu (zebranych przy pomocy wielu metod).
Tryb rzeczywisty	Wizualizacja w trybie rzeczywistym polega na śledzeniu przez wizualizator stanu emocjonalnego osoby aktualnie badanej, tzn. wyświetlaniu danych aktualnie napływających do serwera.
Tryb analizy	Wizualizacja w trybie analizy polega na załadowaniu i wizualizacji danych zakończonego już eksperymentu.
Kontrolka	Pojedynczy element interfejsu realizujący funkcjonalność wizualizacji danego zestawu informacji.

Tabela 1 Słownik pojęć

3.4 Specyfikacja Wymagań Systemowych

3.4.1 Priorytety wymagań

Wymagania zostały oznaczone następującymi priorytetami:

- wysoki – zrealizowanie wymagania jest niezbędne do osiągnięcia minimalnej funkcjonalności systemu
- średni – wymaganie nie jest konieczne do realizacji systemu, powinny być realizowane po ukończeniu wszystkich funkcji oznaczonych priorytetem wysokim,
- niski – najniższy priorytet oznaczający wymagania, które mogą być zrealizowane w przypadku dostatecznej ilości czasu po ukończeniu wszystkich wymagań oznaczonych priorytetami wysokim oraz średnim.

3.4.2 Wymagania funkcjonalne: dane

ID:	DF1
Nazwa:	Informacje o stanie emocjonalnym
Priorytet:	Wysoki
Opis:	<p>Stan emocjonalny użytkownika przekazywany wewnątrz systemu musi zawierać następujące informacje:</p> <ul style="list-style-type: none">• wartość metryki P (positiveness),• wartość metryki A (arousal),• wartość metryki D (dominance),• stopień pewności pomiaru wartości metryki P,• stopień pewności pomiaru wartości metryki A,• stopień pewności pomiaru wartości metryki D,• znacznik czasowy,• identyfikator metody użytej do pomiaru stanu emocjonalnego.
Powiązane wymagania:	<ul style="list-style-type: none">•

ID:	DF2
Nazwa:	Rozróżnialność eksperymentów i sesji

Priorytet:	Wysoki
Opis:	<p>Eksperyment może składać się z wielu sesji różniących się metodą badania stanu emocjonalnego. Dane muszą być przechowywane w sposób, który umożliwia rozróżnienie poszczególnych eksperymentów oraz sesji tego samego eksperymentu.</p>
Powiązane wymagania:	<ul style="list-style-type: none"> •

ID:	DF3
Nazwa:	Trwałość zebranych danych
Priorytet:	Wysoki
Opis:	<p>System musi zachowywać zebrane dane w celu późniejszego wykorzystania ich w trybie analizy.</p>
Powiązane wymagania:	<ul style="list-style-type: none"> • DF1 • DF2

3.4.3 Wymagania funkcjonalne: serwer

ID:	SF1
Nazwa:	Komunikacja z klientem w trybie rzeczywistym
Priorytet:	Wysoki
Opis:	<p>Serwer musi implementować interfejs pozwalający aplikacji klienta na wysyłanie wyników aktualnie przeprowadzanego eksperymentu w trybie rzeczywistym.</p>
Powiązane wymagania:	<ul style="list-style-type: none"> •

ID:	SF2
Nazwa:	Komunikacja z wizualizatorem w trybie rzeczywistym
Priorytet:	Wysoki
Opis:	<p>Serwer musi implementować interfejs pozwalający aplikacji wizualizatora na odbieranie wyników aktualnie przeprowadzanego eksperymentu w trybie rzeczywistym. Podczas trwania eksperymentu wszystkie dane otrzymane od klienta powinny być natychmiastowo przesyłane do aplikacji wizualizatora</p>
Powiązane wymagania:	<ul style="list-style-type: none"> •

ID:	SF3
Nazwa:	Komunikacja z wizualizatorem w trybie analizy
Priorytet:	Wysoki
Opis:	<p>Serwer musi implementować interfejs pozwalający aplikacji wizualizatora na pobieranie wyników zakończonych eksperymentów wg. identyfikatora eksperymentu oraz metody użytej do zebrania danych.</p>
Powiązane wymagania:	<ul style="list-style-type: none"> •

ID:	SF4
Nazwa:	Zapewnienie trwałości danych
Priorytet:	Wysoki
Opis:	<p>Wszystkie informacje dotyczące stanów emocjonalnych otrzymane od aplikacji klienckiej powinny być zapisywane.</p>

Powiązane wymagania:

- DF1

ID:	SF5
Nazwa:	Konfiguracja serwera
Priorytet:	Wysoki
Opis:	<p>Serwer powinien pozwalać na konfigurację:</p> <ul style="list-style-type: none"> • adresu oraz portu, na którym odbywa się komunikacja z klientem oraz wizualizatorem, • danych niezbędnych do połączenia z bazą danych przechowującą zapis eksperymentów: <ul style="list-style-type: none"> ○ adresu, ○ nazwy użytkownika, ○ hasła, ○ nazwy bazy danych.
Powiązane wymagania:	<ul style="list-style-type: none"> • SF6

ID:	SF6
Nazwa:	Wczytywanie ustawień z pliku XML
Priorytet:	Wysoki
Opis:	<p>Konfiguracja serwera powinna być wczytywana z pliku XML.</p>
Powiązane wymagania:	<ul style="list-style-type: none"> • SF5

3.4.4 Wymagania funkcjonalne: wizualizator

ID:	WF1
Nazwa:	Wizualizacja jednokanałowa w trybie rzeczywistym
Priorytet:	Wysoki
Opis:	
<ol style="list-style-type: none">1. Wizualizator musi umożliwiać wizualizację w trybie rzeczywistym aktualnie przeprowadzanego eksperymentu. Wszystkie elementy interfejsu przedstawiające stan emocjonalny muszą być na bieżąco aktualizowane (tzn. pokazywać najnowsze dostępne dane).2. Wizualizacja musi być możliwa przy pomocy następujących kontrolek:<ul style="list-style-type: none">• etykiety,• wykres metryki P,• wykres metryki A,• wykres metryki D,• wartość metryki P,• wartość metryki A,• wartość metryki D,• radar.	
Powiązane wymagania:	
<ul style="list-style-type: none">• SF2• WF9• WF10• WF13• WF14	

ID:	WF2
Nazwa:	Wizualizacja jednokanałowa w trybie analizy
Priorytet:	Wysoki
<p>Opis:</p> <ol style="list-style-type: none"> 1. Wizualizator musi umożliwiać wizualizację jednokanałową w trybie analizy, tzn. bazując na wczytanych z serwera danych jednej sesji zakończonego eksperymentu. 2. Wizualizacja musi być możliwa przy pomocy następujących kontrolek: <ul style="list-style-type: none"> • etykiety, • wykres metryki P, • wykres metryki A, • wykres metryki D, • wartość metryki P, • wartość metryki A, • wartość metryki D, • radar. 	
<p>Powiązane wymagania:</p> <ul style="list-style-type: none"> • SF3 • WF4 • WF9 • WF10 • WF13 • WF14 	

ID:	WF3
Nazwa:	Wizualizacja wielokanałowa w trybie analizy
Priorytet:	Średni
Opis:	<p>1. Wizualizator musi umożliwiać wizualizację wielokanałową w trybie analizy, tzn. bazując na wczytanych z serwera danych wielu sesji zakończonego eksperymentu.</p> <p>2. Każda sesja powinna mieć przypisany kolor, który będzie użyty do wizualizacji danych do niej przypisanych.</p> <p>3. Wizualizacja musi być możliwa przy pomocy następujących kontrolek:</p> <ul style="list-style-type: none"> • etykiety, • wykres metryki P, • wykres metryki A, • wykres metryki D, • wartość metryki P, • wartość metryki A, • wartość metryki D, • informacja o kolorach kanałów.
Powiązane wymagania:	<ul style="list-style-type: none"> • SF3 • WF4 • WF11 • WF12 • WF15 • WF17

ID:	WF4
Nazwa:	Kontrola wyświetlanych informacji w trybie analizy
Priorytet:	Wysoki
Opis:	<p>W trybie analizy, ilość wczytanych danych może być zbyt duża, by wyświetlić wszystkie na raz z zachowaniem ich czytelności. Użytkownik musi mieć możliwość kontroli, z jakiego przedziału czasowego eksperymentu dane są wyświetlane.</p>

Powiązane wymagania:

- WF2
- WF3

ID:	WF5
Nazwa:	Interfejs: pełny
Priorytet:	Wysoki
Opis:	Domyślny widok wizualizatora zawierający wszystkie kontrolki dla wybranego trybu pracy.
Powiązane wymagania:	<ul style="list-style-type: none">• WF1• WF2• WF3

ID:	WF6
Nazwa:	Interfejs: zminimalizowany do kontrolki etykiet
Priorytet:	Wysoki
Opis:	<p>W trybie rzeczywistym, musi istnieć możliwość zminimalizowania wizualizatora tak, by wyświetlał jedynie kontrolki:</p> <ul style="list-style-type: none">• etykiet,• wartości metryk P, A, D.
Powiązane wymagania:	<ul style="list-style-type: none">• WF10• WF16

ID:	WF7
Nazwa:	Interfejs: zminimalizowany do kontrolki radaru
Priorytet:	Wysoki
Opis:	<p>W trybie rzeczywistym, musi istnieć możliwość zminimalizowania wizualizatora tak, by wyświetlał jedynie kontrolki:</p> <ul style="list-style-type: none"> • radar, • wartości metryk P, A, D.
Powiązane wymagania:	<ul style="list-style-type: none"> • WF13 • WF16

ID:	WF8
Nazwa:	Zawsze na wierzchu
Priorytet:	Wysoki
Opis:	<p>Wizualizator musi oferować użytkownikowi opcję „zawsze na wierzchu”, która po aktywacji zapewnia, że okno aplikacji będzie zawsze widoczne ponad innymi otwartymi w systemie aplikacjami.</p>
Powiązane wymagania:	<ul style="list-style-type: none"> •

ID:	WF9
Nazwa:	Sposób wizualizacji: wykres metryki P, A lub D w trybie jednokanałowym
Priorytet:	Wysoki
Opis:	<p>Kontrolka powinna wyświetlać wykres przebiegu wartości metryki (P, A lub D) w funkcji czasu dla aktualnych danych. Wykres musi zawierać:</p> <ul style="list-style-type: none"> • punkty przedstawiające wartość metryki w danych punktach czasu (im mniejsza wartość, tym punkt narysowany jest niżej), • krzywą łączącą punkty następujące po sobie w czasie, • pole pod krzywą wypełnione przez gradient, którego kolor zależy od wartości metryki w poszczególnych punktach: <ul style="list-style-type: none"> ○ pomiędzy dowolnymi sąsiadującymi punktami A i B gradient przechodzi płynnie z koloru C(A) do koloru C(B), ○ kolor dla danego punktu powinien być wyliczany na podstawie wartości metryki (kolor zielony dla wartości dodatnich, biały dla zerowej oraz czerwony dla ujemnych) oraz stopnia pewności pomiaru: maksymalna pewność pomiaru oznacza brak przezroczystości koloru, natomiast maksymalna niepewność pomiaru kolor całkowicie przezroczysty.
Powiązane wymagania:	<ul style="list-style-type: none"> •

ID:	WF10
Nazwa:	Sposób wizualizacji: etykiety w trybie jednokanałowym
Priorytet:	Wysoki
Opis:	<p>Kontrolka powinna wyświetlać wykres przebiegu etykiet, które opisują kolejne, rozpoznane stany emocjonalne:</p> <ul style="list-style-type: none"> • etykiety powinny być rysowane w jednej linii, • rozmiar etykiety powinien zależeć od wartości pewności pomiaru (im większa pewność, tym większy rozmiar)

Powiązane wymagania:

-

ID:	WF11
Nazwa:	Sposób wizualizacji: wykres pojedynczej metryki P, A lub D w trybie wielokanałowym
Priorytet:	Średni
Opis:	<p>Kontrolka powinna wyświetlać wykresy przebiegu wartości metryki (P, A lub D) w funkcji czasu dla danych zebranych z różnych sesji danego eksperymentu. Wykres musi zawierać:</p> <ul style="list-style-type: none">• punkty przedstawiające wartość metryki w danych punktach czasu (im mniejsza wartość, tym punkt narysowany jest niżej),• krzywą łączącą punkty następujące po sobie w czasie w ramach tej samej sesji,• punkty i krzywe każdej z sesji powinny mieć unikalny kolor
Powiązane wymagania:	<ul style="list-style-type: none">•

ID:	WF12
Nazwa:	Sposób wizualizacji: etykiety w trybie wielokanałowym
Priorytet:	Średni
Opis:	<p>Kontrolka powinna wyświetlać wykres przebiegu etykiet, które opisują kolejne, rozpoznane stany emocjonalne zebrane z różnych sesji danego eksperymentu:</p> <ul style="list-style-type: none">• etykiety dotyczące jednej sesji powinny być rysowane w jednej linii,• rozmiar etykiety powinien zależeć od wartości pewności pomiaru (im większa pewność, tym większy rozmiar),• etykiety każdej z sesji powinny mieć unikalny kolor
Powiązane wymagania:	<ul style="list-style-type: none">•

ID:	WF13
Nazwa:	Sposób wizualizacji: radar
Priorytet:	Wysoki
Opis:	<p>Kontrolka powinna wyświetlać wartość P, A i D ostatniego stanu z aktualnego bufora w formie radaru (kręcącego się koła):</p> <ul style="list-style-type: none"> Wartość metryki P oraz stopień pewności jej pomiaru wpływa na kolor koła: <ul style="list-style-type: none"> kolor zielony dla wartości dodatnich, biały dla zerowej oraz czerwony dla ujemnych, maksymalna pewność pomiaru oznacza brak przezroczystości koloru, natomiast maksymalna niepewność pomiaru kolor całkowicie przezroczysty wartość metryki A wpływa na prędkość obrotu koła: im większa wartość tym szybszy obrót, wartość -1 oznacza stan nieruchomy (obrot koła może być wizualizowany np. przy pomocy wskazówki), wartość metryki D wpływa na grubość linii rysującej obwód koła (im większy stopień dominacji tym grubsza linia)
Powiązane wymagania:	<ul style="list-style-type: none">

ID:	WF14
Nazwa:	Sposób wizualizacji: wartość metryki P, A lub D w trybie jednokanałowym
Priorytet:	Wysoki
Opis:	<p>Kontrolka powinna przedstawiać liczbową reprezentację wartości metryki oraz jej stopnia pewności.</p>
Powiązane wymagania:	<ul style="list-style-type: none">

ID:	WF15		
Nazwa:	Sposób wizualizacji: wartość metryki P, A lub D w trybie wielokanałowym		
Priorytet:	Średni		
Opis:	<p>Kontrolka powinna przedstawiać liczbową reprezentację wartości metryki oraz jej stopnia pewności dla każdego z wizualizowanych kanałów.</p> <p>Kolor czcionki dla danych o każdym kanale powinien być unikalny.</p>		
Powiązane wymagania:	<ul style="list-style-type: none"> • 		

ID:	WF16		
Nazwa:	Sposób wizualizacji: wartość metryk P, A i D w trybie jednokanałowym		
Priorytet:	Wysoki		
Opis:	<p>Kontrolka powinna przedstawiać liczbową reprezentację wartości wszystkich trzech metryk oraz stopni pewności ich pomiarów.</p>		
Powiązane wymagania:	<ul style="list-style-type: none"> • 		

ID:	WF17		
Nazwa:	Sposób wizualizacji: informacja o kolorach kanałów		
Priorytet:	Średni		
Opis:	<p>Kontrolka powinna zawierać nazwy kanałów użytych podczas wizualizacji. Każda z nazw powinna być wyświetlona w kolorze użytym do wizualizacji danego kanału.</p>		

Powiązane wymagania:

-

ID:	WF18
Nazwa:	Wczytywanie etykiet z pliku XML
Priorytet:	Wysoki
Opis: Parametry etykiet wykorzystywane podczas wizualizacji powinny być wczytywane z pliku XML. Specyfikacja każdej z etykiet powinna zawierać: <ul style="list-style-type: none">• przedział wartości dla każdej z metryk (P, A i D),• kolor etykiety W przypadku braku wprowadzonych danych, domyślne przedziały wartości metryk powinny wynosić $\langle -1; 1 \rangle$, natomiast domyślny kolor etykiety powinien być szary.	
Powiązane wymagania: <ul style="list-style-type: none">• WF10• WF12	

ID:	WF19
Nazwa:	Wczytywanie konfiguracji źródła danych oraz trybu pracy z pliku XML
Priorytet:	Wysoki
Opis: Konfiguracja wizualizatora powinna być wczytywana z pliku XML. W szczególności: <ul style="list-style-type: none">• adres oraz port służące do komunikacji z serwerem,• tryb wizualizacji: rzeczywisty lub analizy,• identyfikator eksperymentu, z którego pobrane zostaną dane (tylko w przypadku trybu analizy),• identyfikatory metod, dla których sesje zostaną wczytane (tylko w przypadku trybu analizy)	

Powiązane wymagania:

- SF2
- SF3

ID:	WF20
Nazwa:	Edycja wczytanych etykiet z pliku XML
Priorytet:	Niski
Opis:	Wizualizator powinien umożliwiać edycję przedziałów wartości, nazw oraz kolorów etykiet wczytanych z pliku XML.
Powiązane wymagania:	<ul style="list-style-type: none">• WF18

ID:	WF21
Nazwa:	Eksport etykiet do pliku XML
Priorytet:	Niski
Opis:	Wizualizator powinien umożliwiać na eksport etykiet do pliku XML.
Powiązane wymagania:	<ul style="list-style-type: none">• WF18• WF20

ID:	WF23
Nazwa:	Wybór wyświetlanych kontrolek
Priorytet:	Niski
Opis:	

Wizualizator powinien pozwalać na wybór kontroltek, które mają być wyświetlane w poszczególnych widokach i trybach. Konfiguracja kontroltek powinna być wczytywana z pliku XML.

Powiązane wymagania:

-

ID:	WF24
Nazwa:	Manipulacja położeniem kontroltek
Priorytet:	Niski
Opis:	Wizualizator powinien umożliwiać manipulację ułożeniem oraz widocznością (widoczna/ukryta) poszczególnych kontroltek z poziomu interfejsu użytkownika.
Powiązane wymagania:	<ul style="list-style-type: none">•

3.4.5 Wymagania pozafunkcjonalne

ID:	WF1
Nazwa:	Wydajność
Priorytet:	Wysoki
Opis:	<p>W trybie rzeczywistym, opóźnienie pomiędzy wysłaniem informacji o stanie emocjonalnym przez aplikację klienta, a odczytaniem jej przez wizualizator nie może przekraczać jednej sekundy.</p>
Powiązane wymagania:	<ul style="list-style-type: none">•

ID:	WF2
Nazwa:	Język dla nazw zmiennych oraz komentarzy
Priorytet:	Wysoki
Opis:	<p>Wszystkie komentarze oraz nazwy zmiennych, klas i metod w kodzie aplikacji muszą być napisane w języku angielskim.</p>
Powiązane wymagania:	<ul style="list-style-type: none">•

ID:	WF3
Nazwa:	Język aplikacji
Priorytet:	Wysoki
Opis:	<p>Wszelkie informacje przekazywane przez aplikację, w tym jej menu i interfejs użytkownika powinny być sformułowane w języku angielskim.</p>

Powiązane wymagania:

-

ID:	WF4		
Nazwa:	Wspierane platformy		
Priorytet:	Wysoki		
Opis:			
Wszystkie moduły systemu muszą być kompatybilne z systemami operacyjnymi z rodziny Microsoft Windows: XP, Vista, 7.			
Powiązane wymagania:			
<ul style="list-style-type: none">			

3.5 Koncepcja techniczna

3.5.1 Komunikacja

Aby spełnić wymagania wydajnościowe dot. opóźnień w transmisji danych, wszelka komunikacja z serwerem powinna odbywać się w sposób połączeniowy (przy pomocy protokołu TCP) i być zaimplementowana na poziomie gniazd (socketów).

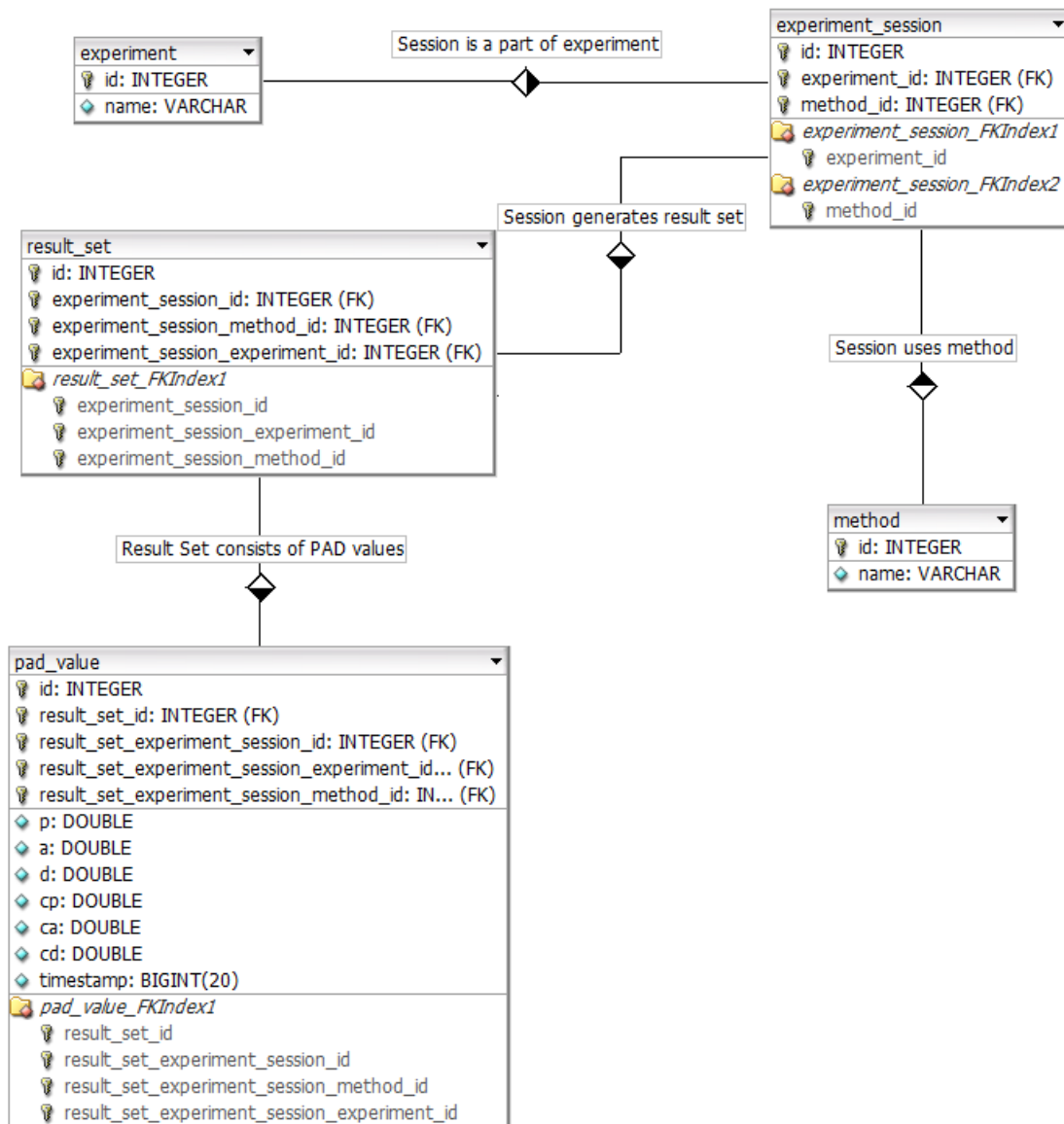
3.5.2 Interfejs i wizualizacja danych

Biorąc pod uwagę kształt aplikacji (aplikacja okienkowa) oraz wizualizowanych danych (wizualizacje dwuwymiarowe) istnieje możliwość wykorzystywania standardowych bibliotek Javy, które zapewnią łatwy rozwój aplikacji oraz kompatybilność z różnymi platformami.

Interfejs użytkownika oraz wszystkie kontrolki wizualizujące dane zostaną wykonane przy pomocy standardowych pakietów Swing oraz AWT.

3.6 Projekt bazy danych

3.6.1 Diagram relacyjny bazy danych



Rysunek 1 Diagram ERD

3.6.2 Opis tabel

experiment

Tabela zawiera podstawowe dane dotyczące przeprowadzanych eksperymentów.

Pola:

- id – identyfikator eksperymentu,
- name – nazwa eksperymentu.

method

Tabela zawiera podstawowe dane identyfikujące metody zbierania danych przez klienta.

Pola:

- id – identyfikator metody,
- name – nazwa metody.

experiment_session

Tabela zawiera dane o poszczególnych sesjach eksperymentów.

Pola:

- id – identyfikator sesji,
- experiment_id – wskazuje na eksperyment, którego częścią jest sesja
- method_id – wskazuje na metodę, użytą do zbierania danych w danej sesji.

result_set

Tabela identyfikuje pojedynczy zbiór wyników dla sesji eksperymentu.

Pola:

- id – identyfikator zbioru wyników,
- experiment_session_id – wskazuje na sesję, do której przypisane są wyniki

pad_value

Tabela zawiera dane o stanach emocjonalnych zebranych podczas eksperymentów.

Pola:

- id – identyfikator stanu,
- result_set_id – wskazuje na zbiór wyników, którego częścią jest dany stan,
- p – wartość metryki P,
- a – wartość metryki A,
- d – wartość metryki D,
- cp – pewność pomiaru metryki P,
- ca – pewność pomiaru metryki A,
- cd – pewność pomiaru metryki D,
- timestamp – znacznik czasowy wyrażony w sekundach

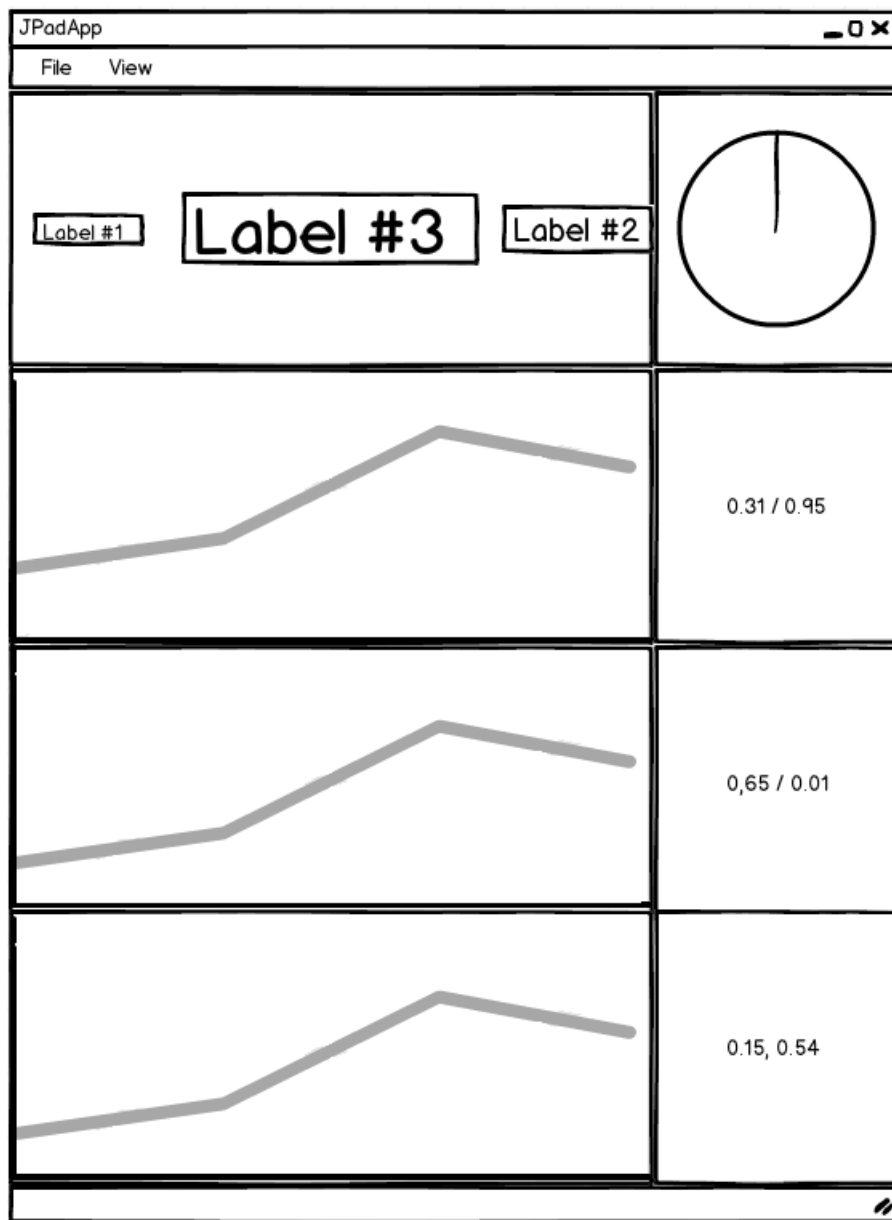
3.6.3 Opis relacji

Nazwa relacji	Liczność	Opis
Session is part of experiment	wiele-do-jednego	Relacja pomiędzy sesją a eksperymentem. Eksperyment może składać się z wielu sesji: każda z nich jest realizowana przy pomocy innej metody oceny danych, generuje więc inne wartości PAD.
Session generates result set	jeden-do-wielu	Relacja pomiędzy sesją a jej zbiorem wyników. Wyniki dla każdej sesji są zebrane w zbiór. Każdy zbiór jest przypisany tylko do jednej sesji. Sesja może posiadać wiele zbiorów wyników (może istnieć konieczność powtórzenia danej sesji).

Session uses method	wiele-do-jednego	<p>Relacja pomiędzy sesją a metodą.</p> <p>Sesja jest przeprowadzana przy pomocy danej metody oceny danych. Dana metoda może być użyta w wielu sesjach.</p>
Result set consists of PAD values	jeden-do-wielu	<p>Relacja pomiędzy zbiorem rezultatów a zapisem stanu PAD.</p> <p>Zbiór wyników składa się z zapisu wielu pojedynczych stanów emocjonalnych użytkownika. Każdy stan emocjonalny jest przypisany do konkretnego rezultatu.</p>

3.7 Projekt interfejsu użytkownika

3.7.1 Tryb rzeczywisty jednokanałowy



Rysunek 2 Interfejs dla trybu rzeczywistego, jednokanałowego

Projekt interfejsu użytkownika trybu rzeczywistego, jednokanałowego znajduje się na rysunku 2.

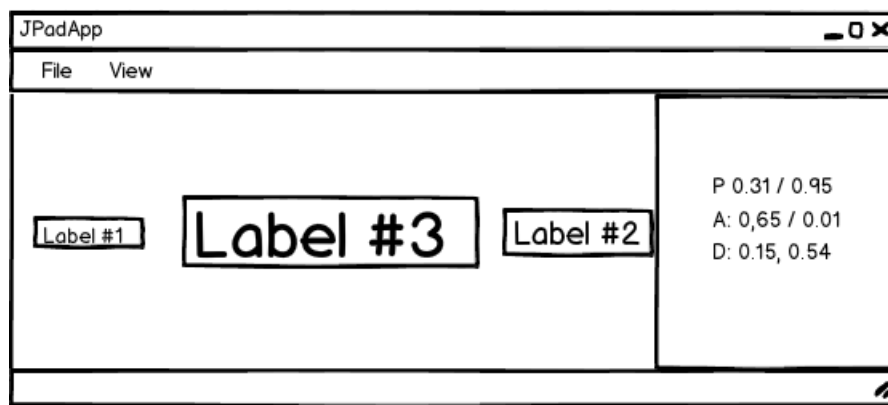
Elementy interfejsu (od lewej do prawej strony):

- Poziom 1:
 - poziome menu (w górnej części)

- Poziom 2:
 - Kontrolka etykiet
 - Kontrolka typu radar
- Poziomy 3, 4 i 5:
 - Wykres wartości P, A lub D na przestrzeni czasu
 - Wartość ostatniej metryki P, A lub D dla aktualnego przedziału czasu

Kontrolki rysujące wykresy wartości oraz etykiet zostały umieszczone w pionie, jedna pod drugą, dla zwiększenia czytelności: mają taką samą szerokość, dzięki czemu etykieta dla danego stanu pojawi się dokładnie nad wartościami poszczególnych metryk (P, A i D) zwizualizowanymi na wykresach.

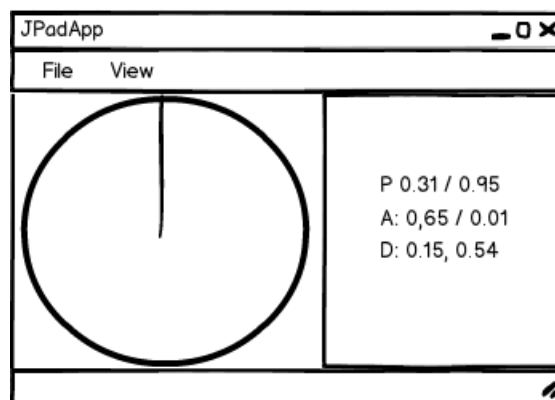
3.7.2 Tryb rzeczywisty, zminimalizowany: etykiety



Rysunek 3 Interfejs dla trybu rzeczywistego, jednokanałowego: zminimalizowany do etykiet.

W trybie rzeczywistym, zminimalizowanym do kontrolki etykiet (rysunek 3.), oprócz menu użytkownika (znajdującego się we wszystkich widokach) znajduje się kontrolka etykiet oraz wartości ostatniego stanu dla aktualnego przedziału czasu.

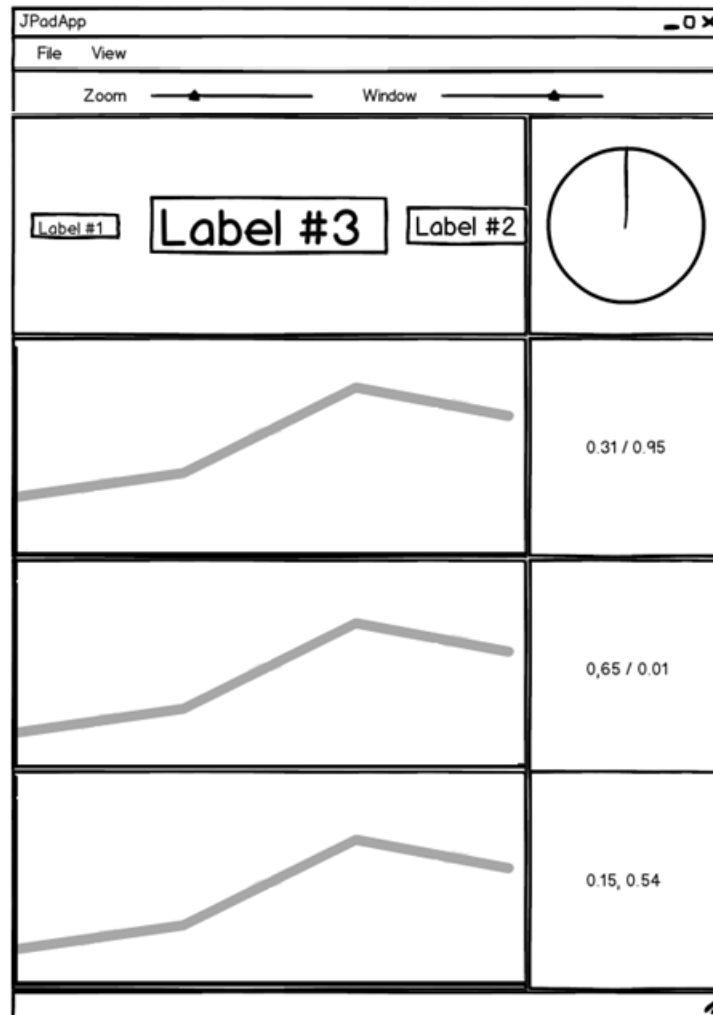
3.7.3 Tryb rzeczywisty, zminimalizowany: radar



Rysunek 4 Interfejs dla trybu rzeczywistego, jednokanałowego: zminimalizowany do radaru.

Tryb rzeczywisty, zminimalizowany (przedstawiony na rysunku 4.), zawiera kontrolkę radaru oraz wartości poszczególnych metryk. Obie kontrolki wizualizującą ostatni stan z aktualnego przedziału czasu.

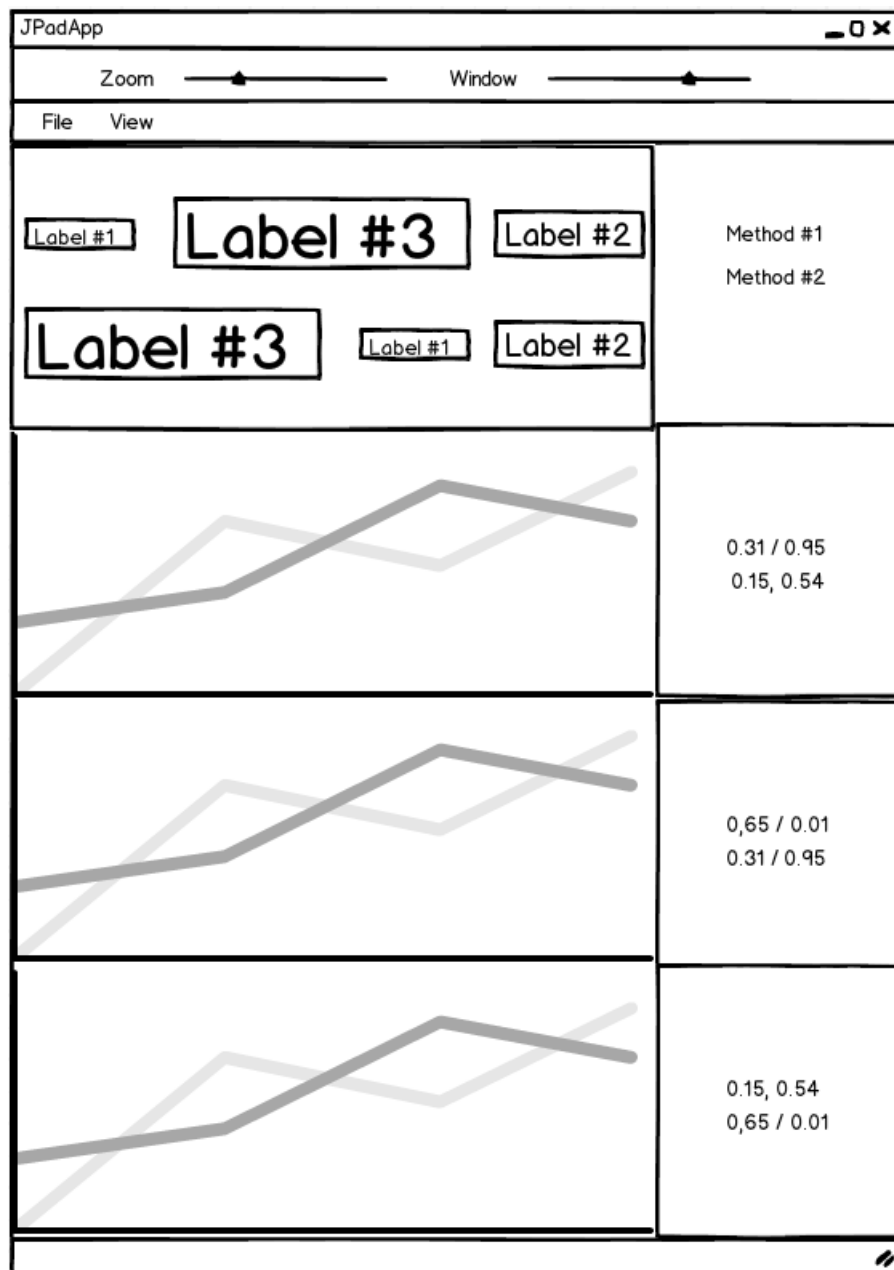
3.7.4 Tryb analizy jednokanałowy



Rysunek 5 Interfejs dla trybu analizy, jednokanałowego.

Dla jednokanałowego trybu analizy (rysunek 5.) możliwe jest wykorzystanie interfejsu z trybu rzeczywistego. Jedyna różnica polega na kontrolce sterującej aktualnym przedziałem czasu, z którego dane są pokazywane.

3.7.5 Tryb analizy wielokanałowy

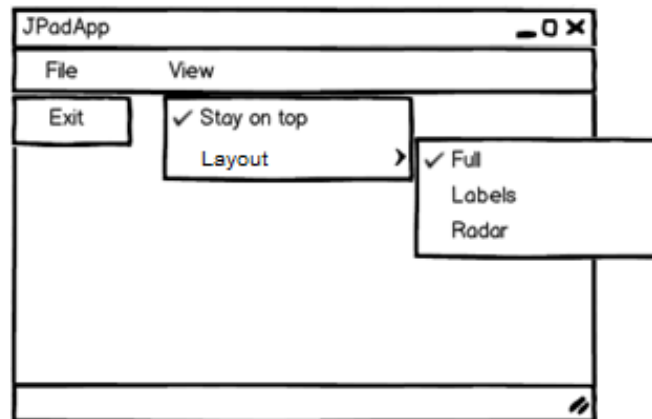


Rysunek 6 Interfejs dla trybu analizy, wielokanałowego.

Układ kontroltek dla wielokanałowego trybu analizy (rysunek 6.) ma identyczny układ do trybu jednokanałowego, jednak wszystkie kontrolki przystosowane są do wyświetlania stanów zebranych przy pomocy wielu metod.

Kontrolka radaru, który jest dedykowana do wizualizacji danych z jednego kanału została zastąpiona przez panel, który informuje użytkownika, jakimi kolorami są narysowane dane z poszczególnych kanałów.

3.7.6 Menu użytkownika



Rysunek 7 Menu aplikacji.

Menu użytkownika (rysunek 7.) jest wyświetlane w każdym widoku i jest odpowiedzialne za sterowanie funkcjami widoku aplikacji.

Opis elementów menu:

- File – Exit:
wyjście z aplikacji
- View – Stay on top:
włączenie lub wyłączenie opcji „zawsze na wierzchu” (wymaganie WF8)
- View – Layout:
wybór opcji widoku aplikacji. Wybór jest możliwy jedynie w przypadku trybu rzeczywistego.

4. Podręcznik użytkownika

4.1 Wymagania

Wszystkie moduły aplikacji (serwer, klient, wizualizator) wymagają zainstalowania Javy w wersji 7.

Dodatkowo, serwer wymaga dostępu do bazy danych MySQL w wersji 5.0 lub wyższej (baza danych nie musi znajdować się na tej samej maszynie).

4.2 Struktura folderów aplikacji

jpadapp

Główny katalog zawierający wszystkie elementy aplikacji.

Zawiera pliki wykonywalne potrzebne do uruchomienia poszczególnych aplikacji:

- *jpad-server.bat* – serwer
- *jpad-sample-client.bat* – przykładowy klient, który generuje losowe stany PAD i przesyła do serwera
- *jpad-visualiser.bat* – wizualizator

jpadapp/app

Folder zawiera kod źródłowy aplikacji, niestandardowe biblioteki użyte do jej stworzenia oraz szablony plików konfiguracyjnych.

jpadapp/config

Katalog zawiera pliki konfiguracyjne używane przez serwer, wizualizator oraz przykładową aplikację kliencką.

jpadapp/docs

Dokumentacja aplikacji oraz jej kodu źródłowego.

4.3 Konfiguracja: uwagi

Konfiguracja działania wszystkich modułów aplikacji jest możliwa przy pomocy plików XML znajdujących się w katalogu *jpadapp/config*. Aby poprawnie uruchomić każdy z modułów, pliki konfiguracyjne muszą zawierać wszystkie węzły oznaczone jako wymagane. Pliki muszą też być poprawnie sformatowanymi dokumentami XML (nie mogą zawierać wpisów w formacie niezgodnych ze standardem XML).

4.4 Konfiguracja serwera

Konfigurację serwera przeprowadza się przez edycję pliku *server-settings.xml*, który znajduje się w katalogu *jpadapp/config*.

Tabela nr 2 zawiera opis poszczególnych węzłów.

Nazwa węzła	Pole wymagane	Opis
host	Tak	<p>Adres, na którym serwer nasłuchuje i komunikuje się z aplikacją klienta lub wizualizatorem.</p> <p>Powinien zawierać adres IP (np. <i>127.0.0.1</i>) lub <i>localhost</i>.</p> <p>Przykładowa postać węzła <i>host</i>:</p> <pre><host>10.237.133.72</host></pre>
port	Tak	<p>Port, na którym serwer nasłuchuje i komunikuje się z aplikacją klienta lub wizualizatorem.</p> <p>Powinien zawierać wartość liczbową (np. <i>8081</i>) . Użytkownik musi także upewnić się, że na danym porcie nie działa żadna inna aplikacja.</p> <p>Przykładowa postać węzła <i>port</i>:</p> <pre><port>8081</port></pre>
db_string	Tak	<p>Łańcuch znaków w formacie Java Database Connectivity (JDBC), zawierający adres oraz nazwę bazy danych, która zostanie użyta przez wizualizator do zapisu oraz odczytu informacji o sesjach przeprowadzanych eksperymentów.</p> <p>Format łańcucha wygląda następująco:</p> <pre>jdbc:<typ bazy>://<adres bazy>/<nazwa bazy></pre> <p>Serwer jest przystosowany do wykorzystania bazy Oracle MySQL, przykładowa wartość <i>db_string</i> dla tej bazy wygląda następująco:</p> <pre>jdbc:mysql://localhost//pad</pre> <p>Przykładowa postać węzła <i>db_string</i>:</p> <pre><db_string>jdbc:mysql://localhost//pad</db_string></pre>
db_user	Tak	<p>Nazwa użytkownika używane przez serwer podczas autoryzacji do bazy danych.</p> <p>Przykładowa postać węzła <i>db_user</i>:</p>

		<code><db_user>wmgaca</db_user></code>
db_password	Tak	<p>Hasło użytkownika używane przez serwer podczas autoryzacji do bazy danych.</p> <p>Przykładowa postać węzła <i>db_password</i>:</p> <p><code><db_password>H3\$abpo#3m</db_password></code></p> <p>Dopuszczalne jest także użycie pustego hasła:</p> <p><code><db_password></db_password></code></p>

Tabela 2. Konfiguracja serwera: plik *server-settings.xml*

4.5 Konfiguracja wizualizatora

Wizualizator korzysta z trzech plików konfiguracyjnych:

- *config/server-settings.xml* – konfiguracja połączenia z serwerem,
- *config/visualiser-settings.xml* – konfiguracja trybu działania wizualizatora,
- *config/labels.xml* – konfiguracja kontrolki etykiet.

Szczegóły konfiguracji pliku *visualiser-settings.xml* zostały umieszczone w tabeli nr 3.

Do konfiguracji połączenia z serwerem wizualizator używa pliku *server-settings.xml*, z którego czytuje wartości węzłów *host* oraz *port*. W przypadku umieszczenia serwera i wizualizatora na tej samej maszynie, wartość węzła *hosta* może być ustawiona na wartość *127.0.0.1* lub *localhost*, jeśli jednak moduły są uruchamiane na oddzielnych maszynach, w konfiguracji wizualizatora niezbędne będzie podanie adresu IP serwera.

Nazwa węzła	Pole wymagane	Opis
mode	Tak	<p>Tryb pracy wizualizatora, wartość węzła może przybrać jedną z dwóch postaci:</p> <ul style="list-style-type: none"> • <i>Load</i> – wizualizator zostanie uruchomiony w trybie <i>analizy</i> (off-line), • <i>auto</i> – wizualizator zostanie uruchomiony w trybie rzeczywistym (on-line) <p>Przykładowa postać węzła <i>mode</i>:</p> <p><code><mode>auto</mode></code></p>
session	Tak	<p>Węzeł <i>session</i> zawiera dwa atrybuty:</p> <ul style="list-style-type: none"> • <i>id</i> – identyfikator eksperymentu (wskazujący na

		<p>kolumnę <i>id</i> tabeli <i>experiment</i>), atrybut określa eksperyment, dla którego dane zostaną wczytane z serwera,</p> <ul style="list-style-type: none"> <i>channels</i> – wymienione po przecinku identyfikatory metod (wskazujące kolumnę <i>id</i> w tabeli <i>method</i>), dla których dane zostaną wczytane. <p>Przykładowa postać węzła <i>session</i>:</p> <pre><session id="5" channels="1, 2, 3, 8" /></pre>
--	--	--

Tabela 3. Konfiguracja wizualizatora: plik *visualiser-settings.xml*

4.5.1 Konfiguracja etykiet

Plik konfiguracyjny etykiet zawiera definicje wszystkich etykiet, które będą wyświetlane przez wizualizator: w punkcie 3.3

- każda etykieta jest zdefiniowana w osobnym węźle *Label* i posiada:
 - atrybut *name*: nazwę do wyświetlenia,
 - atrybut *color*: kolor podany w systemie szesnastkowym,
 - zagnieżdżony węzeł *p*: zakres wartości metryki P: wartość minimalną i maksymalną,
 - zagnieżdżony węzeł *a* zakres wartości metryki A: wartość minimalną i maksymalną,
 - zagnieżdżony węzeł *d* zakres wartości metryki D: wartość minimalną i maksymalną,
- w przypadku braku sprecyzowanej wartości minimalnej lub maksymalnej zakresu, wizualizator użyje wartości domyślnych: odpowiednio *-1* i *1* (także w przypadku całkowitego braku węzła dla danej metryki).
- etykiety są czytane przez wizualizator w kolejności od góry do dołu i pierwsza napotkana etykieta pasująca do danego stanu jest wyświetlona przez wizualizator, a szukanie zostanie przerwane,
- etykiety powinny być definiowane w kolejności od najbardziej szczegółowych i obejmujących najmniejszą powierzchnię w przestrzeni PAD do najbardziej ogólnych.

Przykładowa konfiguracja etykiet:

```
<Labels>
  <label name="Happy" color="#ff0000">
    <p min="0.3" max="0.5" />
    <a min="-0.3" max="0.4" />
    <d min="0.1" max="0.3" />
  </label>
  <label name="Sad" color="#000000">
    <p max="-0.1" />
  </label>
  <label name="Normal" color="#00ff00">
    <p max="0.5" />
  </label>
</Labels>
```

```

</label>
<label name="Foo" color="#0000ff">
</label>
</labels>

```

Etykiety wygenerowane dla przykładowej konfiguracji przedstawionej powyżej:

- Happy:
 - kolor: #FF0000 (czerwony),
 - P: $\langle 0.3; 0.5 \rangle$,
 - A: $\langle -0.3; 0.4 \rangle$
 - D: $\langle 0.1; 0.3 \rangle$
- Sad:
 - kolor: #000000 (czarny),
 - P: $\langle -1; -0.1 \rangle$,
 - A: $\langle -1; 1 \rangle$
 - D: $\langle -1; 1 \rangle$
 -
- Normal:
 - kolor: #00FF00 (zielony),
 - P: $\langle -1; 0.5 \rangle$,
 - A: $\langle -1; 1 \rangle$
 - D: $\langle -1; 1 \rangle$
 -
- Foo:
 - kolor: #0000FF (niebieski),
 - P: $\langle -1; 1 \rangle$,
 - A: $\langle -1; 1 \rangle$
 - D: $\langle -1; 1 \rangle$

4.6 Konfiguracja bazy danych

Aplikacja serwera wymaga dostępu do bazy danych MySQL w wersji 5.0 lub wyższej.

Konfiguracja bazy danych polega na utworzenie nowej bazy danych na serwerze MySQL. Skrypt tworzący bazę oraz układ tabel znajduje się pod następującą ścieżką: *jpadapp/app/sql/pad.sql*.

Skrypt może być wykonany przy pomocy dowolnego narzędzia ułatwiającego zarządzaniem bazą danych, np.:

- MySQL Workbench,
- SQLyog,
- phpMyAdmin.

Można też skorzystać ze standardowej konsoli MySQL:

```
mysql -u <user> -p < pad.sql
```

Na przykład:

```
mysql -u wmgaca -p < c:\dev\github\jpadapp\app\sql\pad.sql
```

4.7 Uruchomienie aplikacji

Moduły powinny być uruchamiane w następującej kolejności:

- serwer,
- klient (wymagany, jeśli wizualizator ma działać w trybie rzeczywistym),
- wizualizator.

Pliki startowe modułów znajdują się w katalog głównym aplikacji:

- jpadapp-server.bat
- jpadapp-sample-client.bat,
- jpadapp-visualiser.bat.

Przed pierwszym uruchomieniem należy je edytować, zmieniając (jeśli jest taka potrzeba) ścieżkę Javy tak, by wskazywała na jej instalację na danej maszynie:

```
set JAVA_PATH="C:\glassfish3\jdk7\bin\java.exe"
```

Jeśli ścieżka do folderu zawierającego blik java.exe znajduje się w zmiennej środowiskowej *PATH* systemu Windows, podanie pełnej ścieżki jest zbędne:

```
set JAVA_PATH=" java"
```

Wszystkie pliki BAT uruchamiają plik JAR (jpadapp/app/bin/jpadapp.jar) z odpowiednim parametrem precyzującym, która z aplikacji powinna zostać uruchomiona:

- server, -s, --server – uruchomienie serwera
- visualiser, -v, --visualiser – uruchomienie wizualizatora,
- mock, -m, --mock – uruchomienie przykładowego klienta.

Na przykład, w celu uruchomienia wizualizatora można użyć którejkolwiek z poniższych komend:

```
java -jar jpadapp.jar --visualiser
java -jar jpadapp.jar -v
java -jar jpadapp.jar visualiser
```

5. Podręcznik programisty: rozwój aplikacji

5.1 Wprowadzenie

Nazwy klas używane w podręczniku programisty są podawane razem z pełną ścieżką do pakietu, w którym się znajdują, np. `Lib.types.Coords` wskazuje na klasę `Coords` znajdującą się w pakiecie `lib.types`, a `Lib.types.PADDataHandler.getValuesPostCurrentBuffer` metodę `getValuesPostCurrentBuffer` klasy `PADDataHandler` z pakietu `Lib.types`.

Dokumentacja kodu wygenerowana przy pomocy Javadoc znajduje się w folderze `jpadapp/docs/javadoc`

5.2 Notacja stempli czasowych

Czas używany jako stempel czasowy jest podany w milisekundach.

Stempel może oznaczać czas, który minął od rozpoczęcia eksperymentu (tzn. stempel czasowy początku eksperymentu wynosi 0, sekundę później 1000, itd.) lub od danego punktu w przeszłości: np. metoda `System.currentTimeMillis` podaje ilość milisekund, które upłynęły od 1. Stycznia 1970 roku.

Wizualizator podczas rysowania poszczególnych kontrolek i doboru aktualnego okna przedziału czas zawsze bierze pod uwagę czas względny, tzn. od początku eksperymentu. Wobec tego którakolwiek z metod wymienionych powyżej może być użyta do wyliczania stempli czasowych pod warunkiem, że dane dla wszystkich sesji konkretnego eksperymentu będą spójne.

5.3 Tworzenie źródła danych dla serwera

Przykładowa implementacja źródła danych znajduje się w klasie `PadDummyClient`.

5.3.1 Format danych

Podstawowym nośnikiem danych o stanach PAD są obiekty klasy `Lib.types.PADState`, które zawierają wszystkie informacje o danym stanie:

- wartości metryk P, A, D oraz pewności ich pomiaru,
- stempel czasowy pomiaru,
- metodę użytą do zbierania danych.

Para wartość-pewność dla każdej z metryk jest przedstawiona przy pomocy obiektu klasy `Lib.types.PADValue`, która zawiera:

- wartość metryki,
- pewność pomiaru metryki,
- opcjonalnie: stempel czasowy pomiaru.

Źródło danych, dla każdego rozpoznanego stanu emocjonalnego, powinno tworzyć nowy obiekt `Lib.types.PADState`, np.:

```

PADState state = new PADState(
    new PADValue(pVal, pCert), // P
    new PADValue(aVal, aCert), // A
    new PADValue(dVal, dCert), // D
    System.currentTimeMillis(), // Stempel czasowy
    6)                          // Metoda

```

5.3.2 Komunikacja z serwerem

Komunikacja pomiędzy modułami odbywa się przy pomocy połączeń socketowych.

Komunikacja odbywa się poprzez obustronne przysyłanie obiektów reprezentujących poszczególne wiadomości, znajdujące się w pakiecie *Lib.net.packages*.

Do wysyłania i odbierania należy wykorzystać obiekty *java.io.ObjectOutputStream* oraz *java.io.ObjectInputStream*:

```

// Utworzenie połączenia socketowego
Socket socket = new Socket(host, port);

// Pobranie strumieni wyjścia i wejścia
ObjectOutputStream out =
    new ObjectOutputStream(socket.getOutputStream());

ObjectInputStream in =
    new ObjectInputStream(socket.getObjectInputStream());

// Wysłanie pakietu
out.writeObject(PADPackage.getRandom());

// Pobranie pakietu
Package package = (Package)in.readObject();

```

Aby zapewnić poprawną komunikację z serwerem, pierwszym wysłanym pakietem po nawiązaniu połączenia powinien być *Lib.net.packages.HandshakePackage* z typem klienta ustawionym na *Lib.types.Client.DATA_SOURCE*:

```

out.writeObject(new HandshakePackage(Client.DATA_SOURCE));

```

Następnie należy wysłać pakiet informujący serwer o nazwie eksperymentu oraz metody (zostaną utworzone w przypadku ich braku w bazie danych).

```

out.writeObject(new ExperimentInfoPackage("Sample", "Data"));

```

Następnie można rozpocząć transmisję stanów PAD:

```

out.writeObject(new PADPackage(padState));

```

Po zakończeniu eksperymentu klient powinien wysłać pakiet informujący serwer o zakończeniu połączenia, *Lib.net.packages.EndPackage*:

```
out.writeObject(new EndPackage());
```

5.4 Architektura serwera

Główne klasy wchodzące w skład serwera:

- *PadServer* – nasłuchuje na zadanym porcie, nawiązuje połączenia oraz tworzy nowy wątek do obsługi każdego z nich,
- *Lib.net.ConnectionHandler* – zawiera logikę obsługi przychodzących pakietów i komunikacji z klientem oraz wizualizatorem.

5.4.1 ConnectionHandler

Każde połączenie z serwerem jest obsługiwane w odrębnym wątku przez instancję klasy *Lib.net.ConnectionHandler*.

Główna logika obiektu znajduje się w funkcji *run*, której sprawdzany jest typ połączenia (źródło danych lub wizualizator) oraz jego parametry (tryb rzeczywisty, tryb analizy).

Wszystkie obiekty typu *ConnectionHandler* są przechowywane w współdzielonej liście: *ConnectionHandler.connectionHandlers*. Praca serwera oparta na wątkach oraz współdzielenie listy połączeń pozwala na:

- jednoczesne podłączenie więcej niż jednego wizualizatora w trybie rzeczywistym (wszystkie informacje spływające ze źródła danych są przesyłane do wszystkich aktywnych wizualizatorów),
- jednoczesne podłączenie więcej niż jednego źródła danych,
- podłączenie wizualizatora w trybie analizy w dowolnym czasie.

Podstawowe funkcje klasy *ConnectionHandler*

- *handle* – stworzenie handlera dla nowego połączenia,
- *broadcast* – przesłanie pakietu do wszystkich podłączonych wizualizatorów,
- *registerHandler*, *removeHandler* – dodanie lub usunięcie handlera z listy,
- *send()* – przesłanie pakietu do odbiorcy,
- *readPackage()* – odczytanie otrzymanego pakietu.

5.5 Architektura wizualizatora

5.5.1 Dane

Wszystkie dane dotyczące stanów emocjonalnych przechowywane są w postaci obiektów *Lib.types.PADState*.

Z uwagi na konieczność rozróżnienia stanów z różnych sesji, optymalizacji zużycia pamięci oraz łatwości rozwoju aplikacji, wszystkie dane przechowywane w obiekcie *lib.types.PADDateHandlerContainer*, który został zbudowany wg. wzorca projektowego singleton. *PADDataHandlerContainer* jest kontenerem, który zawiera obiekty *lib.types.PADDataHandler* bezpośrednio przechowujące stany emocjonalne z poszczególnych sesji.

W trybie jednokanałowym dostęp do kontenerów obiektów PADState otrzymujemy w następujący sposób:

```
PADDataHandlerContainer.getInstance().get();
```

W trybie wielokanałowym:

```
// zwraca wszystkie handlersy jako obiekt mapujący  
// identyfikator kanału na obiekt handlera  
PADDataHandlerContainer.getInstance().getAll();  
  
// zwraca listę wszystkich handlerów  
PADDataHandlerContainer.getInstance().getList();
```

5.5.2 Interfejs użytkownika: kontrolki

Wszystkie kontrolki używane do wizualizacji stanów emocjonalnych dziedziczą po klasie

lib.ui.panels.base.Panel.

Dostęp do danych stanów emocjonalnych odbywa się przez atrybut *data* wskazujący na obiekt *PADDataHandler* lub *PADDataHandlerContainer* (w zależności od trybu działania aplikacji: jedno lub wielokanałowego).

Logika rysowania jest implementowana w metodzie *customPaintComponent*, którą każdy nowy panel powinien nadpisywać. Razem z konstruktorem stanowi to minimum metod, które należy zaimplementować tworząc nową kontrolkę, np.:

```
class CustomWidgetPanel extends Panel {  
    public CustomWidgetPanel(int width, int height) {  
        super(PAD.Type.PAD, width, height);  
    }  
  
    @Override  
    public void customPaintComponent(Graphics2D g2d) {  
        PADDataHandler handler =  
            PADDataHandlerContainer.getInstance().get();  
  
        if (handler.isEmpty) continue;  
        g2d.drawString(handler.getLast().getP().getValue().toString(),  
            getCenterX(), getCenterY( ))
```

```
}  
}
```

5.5.3 Interfejs użytkownika: widoki

Wszystkie widoki używane przez aplikację wizualizatora (on-line jednokanałowy, off-line jedno i wielokanałowy) dziedziczą po klasie *Lib.ui.frames.base.Frame*.

Budowanie widoku odbywa się w metodach *setFullLayout* oraz, dla trybu on-line jednokanałowego, *setMinimalLabelLayout* oraz *setMinimalRadarLayout*.

6. Raport Końcowy

6.1 Zespół projektowy

Opiekun projektu

dr inż. Agnieszka Landowska

Klient projektu

Wydział Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej
Katedra Inżynierii Oprogramowania

Uczestnicy projektu

Imię i nazwisko	Nr indeksu	Katedra
Wojciech Gaca	106238	Katedra Inżynierii Oprogramowania

6.2 Temat projektu

Tematem pracy jest stworzenie wizualizatora stanu emocjonalnego osoby badanej do zastosowań w projektach naukowych oraz medycznych.

Podstawowe zagadnienia do opracowania obejmowały:

- zapoznanie się z wybranymi zagadnieniami z dziedziny affective computing,
- projekt wizualizatora,
- implementacja wizualizatora,
- testy rozwiązania,
- opis rozwiązania.

6.3 Kontekst projektu

Wizualizator stanu emocjonalnego osób badanych ma na celu wizualne pokazanie rozpoznanych stanów emocjonalnych.

U podstaw koncepcji wizualizatora leżą następujące założenia:

- reprezentacja stanów emocjonalnych oparta o ciągły model PAD,
- trzy tryby pracy, w tym jeden tryb rzeczywisty, wizualizujący na bieżąco przeprowadzany eksperyment,
- możliwość pełnej redefinicji etykiet stanów emocjonalnych, w tym również dla sesji już zamkniętych
- różnorodność metod wizualizacji stanów emocjonalnych,
- możliwość wyboru formy wizualizacji stanu emocjonalnego,
- elastyczność i konfigurowalność rozwiązania.

6.4 Osiągnięte rezultaty

W wyniku projektu osiągnięto następujące rezultaty:

- stworzony został serwer, wizualizator stanu emocjonalnego oraz przykładowy klient generujący dane wraz z dokumentacją projektową i podręcznikiem użytkownika,
- zaimplementowane funkcje pokrywają wszystkie wymagania niezbędne dla uzyskania podstawowej funkcjonalności systemu.

6.5 Proces realizacji projektu

Projekt rozpoczął się od zapoznania z tematem przez zespół projektowy, sprecyzowania wymagań przez przedstawiciela klienta oraz wspólnego ustalenia koncepcji rozwiązania. Następnie, w tygodniowych iteracjach, implementowano kolejne funkcjonalności wizualizatora.

Wykorzystane narzędzia:

- IntelliJ Idea – środowisko programistyczne,
- SQLyog – narzędzie ułatwiające pracę z bazą danych MySQL,
- DBDesigner – projektowanie bazy danych,
- Balsamiq Mockups – projektowanie interfejsów użytkownika,
- Git, Github – przechowywanie oraz kontrola wersji dokumentacji oraz kodu aplikacji.

6.6 Dokumentacja

W ramach projektu stworzono następujące dokumenty:

- wizję systemu,
- specyfikację wymagań systemowych,
- koncepcję rozwiązania,
- projekt bazy danych,
- projekt interfejsu użytkownika,
- raport końcowy,
- podręcznik użytkownika,
- podręcznik programisty (w celu dalszego rozwoju systemu).

6.7 Zmiany w trakcie projektu

Rzeczywiste nakłady pracy okazały się dużo wyższe od początkowo zakładanych, dlatego też nie udało się zrealizować wymagań o niższych priorytetach. Aby osiągnąć cel jakim jest stworzenie funkcjonalnej aplikacji w określonym terminie, już podczas prac projektowych zdecydowano o realizacji tylko funkcjonalności o najwyższym priorytecie.

Trudności i rozbieżności pomiędzy początkowymi założeniami, a wynikami wynikły głównie z:

- zbyt krótkiego czasu na realizację projektu przez zespół jednoosobowy: rozpoczęcie prac nad projektem miało miejsce w połowie listopada,
- niewielkiego doświadczenia w narzuconej technologii: wymagania dot. docelowej technologii rozwiązania, tzn. języka i środowiska Java były narzucone z góry i nie pokrywały się z mapą kompetencji wykonawcy projektu,
- niespodziewanej przerwy w realizacji projektu, przez co prace projektowe zostały wstrzymane na dwa tygodnie,
- błędów w szacowaniu czasu: m.in. nie wzięcia pod uwagę czasu wytworzenia serwera, który początkowo nie był częścią wymagań dot. tworzonego systemu.

Powyższe czynniki sprawiły, że ryzyko związane z niedotrzymaniem terminu było wysokie.

7. Podsumowanie

W ramach projektu udało się osiągnąć postawiony cel, to znaczy stworzyć aplikację wizualizującą stany emocjonalne na potrzeby eksperymentów naukowych oraz badawczych.

Stworzony został serwer będący centralną częścią systemu, przykładowa implementacja źródła danych oraz aplikacja wizualizatora. Do wymienionych części systemu powstała także dokumentacja projektowa i użytkowa.

Wszystkie założenia i wymagania oznaczone jako niezbędne do osiągnięcia minimalnej funkcjonalności systemu zostały zrealizowane.

Nie wszystkie funkcje wizualizatora o niższych priorytetach udało się zaimplementować, do najważniejszych należą:

- edycja i eksport konfiguracji etykiet z poziomu narzędzia,
- elastyczny interfejs użytkownika pozwalający na manipulację kontrolkami podczas działania aplikacji.

Dalsze prace przy rozwoju narzędzia powinny uwzględnić powyższe funkcje. Szkielet aplikacji jest kompletny i łatwo rozszerzalny, kolejnymi krokami powinno być dodanie – na miarę potrzeb – możliwości wizualizacji stanów emocjonalnych przy pomocy nowych kontroltek, np. wizualizację w trójwymiarowej przestrzeni PAD czy z użyciem zestawów symboli odwzorowujących emocje.

Zrealizowane rozwiązanie zostanie zaimplementowane w ramach projektów grupowych lub prac dyplomowych prowadzonych na Wydziale ETI Politechniki Gdańskiej.

Bibliografia

Landowska, Agnieszka: Raport Techniczny - Wizualizator Stanu Emocjonalnego Dla Eksperymentów Medycznych i Badawczych, WETI PG, 2012.

Fowler, Ben: UML w Kropelce, LTP, Warszawa, 2005

Mehrabian, Albert: Basic dimensions for a general psychological theory, 1980.

Fry, Ben: Visualising Data, O'Reilly, 2007

Wykaz tabel

Tabela 1 Słownik pojęć	8
Tabela 2. Konfiguracja serwera: plik server-settings.xml	40
Tabela 3. Konfiguracja wizualizatora: plik visualiser-settings.xml	41

Wykaz rysunków

Rysunek 1 Diagram ERD	29
Rysunek 2 Interfejs dla trybu rzeczywistego, jednokanałowego	33
Rysunek 3 Interfejs dla trybu rzeczywistego, jednokanałowego: zminimalizowany do etykiet.	34
Rysunek 4 Interfejs dla trybu rzeczywistego, jednokanałowego: zminimalizowany do radaru.....	34
Rysunek 5 Interfejs dla trybu analizy, jednokanałowego.....	35
Rysunek 6 Interfejs dla trybu analizy, wielokanałowego.	36
Rysunek 7 Menu aplikacji.....	37