

| | | |
|--|------------------|--|
| Politechnika Świętokrzyska | | |
| Studia niestacjonarne (semestr zimowy) | | |
| Programowanie obiektowe (java) - projekt | 2023/2024 | |
| Imię i Nazwisko: Jakub Paluch Miroslaw Dudek | Grupa: 2IZ11A | Temat projektu: System obsługi klienta |

1. Opis Projektu – Przyjęte Założenia

Projekt zakłada stworzenie systemu obsługi klienta, który umożliwia zarządzanie klientami, wizytami oraz kolejkami. System składa się z aplikacji serwerowej (klasa SerwerApp) oraz aplikacji klienckiej (klasa KlientApp), które komunikują się ze sobą za pomocą sieci. System umożliwia dodawanie, edycję i usuwanie informacji o klientach, zarządzanie terminarzem wizyt oraz obsługę kolejki klientów.

2. Założenia Dodatkowe

Aplikacja serwerowa nasłuchuje na połączenia na wskazanym porcie i obsługuje wielu klientów jednocześnie.

Aplikacja kliencka łączy się z serwerem i umożliwia interakcję użytkownika z systemem.

System wykorzystuje bazę danych do przechowywania informacji o klientach, wizytach i kolejkach.

Interfejs użytkownika jest prosty i intuicyjny, umożliwia łatwą nawigację i wykonanie potrzebnych działań.

3. Konfiguracja i Testowanie Aplikacji

Konfiguracja

Uruchomienie serwera na zdefiniowanym porcie (domyślnie 12345).

Ustawienie połączenia z bazą danych poprzez klasę BazaDanych.

Konfiguracja aplikacji klienckiej do łączenia się z serwerem pod wskazanym adresem IP.

Testowanie

Testowanie komunikacji między klientem a serwerem.

Testowanie operacji CRUD na bazie danych dla klientów, wizyt i kolejek.

Sprawdzenie poprawności działania interfejsu użytkownika i obsługi błędów.

4. Opis Architektury Systemu

System składa się z następujących komponentów:

SerwerApp: Aplikacja serwerowa obsługująca żądania od klientów.

KlientApp: Aplikacja kliencka umożliwiająca interakcję z systemem.

BazaDanych: Odpowiedzialna za połączenie i operacje na bazie danych.

ObsługaKlienta, Wizyta, Zarządzaniekolejka, Autorzy: Klasy odpowiadające za poszczególne funkcjonalności systemu, takie jak zarządzanie klientami, wizytami, kolejką oraz wyświetlanie informacji o autorach.

5. Opis Testowania Systemu

Testowanie systemu obejmowało:

Testy Jednostkowe: Sprawdzenie poprawności działania poszczególnych klas i metod.

Testy Integracyjne: Testowanie współdziałania komponentów systemu, szczególnie komunikacji między aplikacją kliencką a serwerową oraz operacji na bazie danych.

Testy Użytkowe: Przeprowadzenie serii działań symulujących rzeczywiste użycie systemu przez użytkowników, w celu sprawdzenia użyteczności i stabilności aplikacji.

Uwagi:

Niestety zaplanowane testy nie do końca działają, tak jak początkowo tego oczekiwaliśmy z powodu powtarzającego się problemu z dostępem do modułu, gdzie znajdują się wszystkie klasy.

6. Uwagi i Wnioski (kontynuacja)

Projekt ten został zrealizowany przez dwuosobowy zespół, co miało istotny wpływ na jego przebieg i strukturę. Poniżej przedstawiamy podział obowiązków pomiędzy członków zespołu:

Podział Obowiązków

Jakub Paluch:

Opracowanie logiki aplikacji serwerowej (SerwerApp), w tym obsługa połączeń sieciowych i integracja z bazą danych.

Implementacja klasy Serwer, zarządzającej połączeniami.

Tworzenie i zarządzanie bazą danych (BazaDanych), w tym projektowanie schematu bazy i implementacja zapytań SQL.

Testowanie komunikacji sieciowej i operacji bazodanowych.

Mirosław Dudek:

Projektowanie i implementacja interfejsu użytkownika aplikacji klienckiej (KlientApp).

Rozwój klas Obslugaklienta, Wizyta i Zarządzaniekolejka, odpowiadających za poszczególne funkcjonalności systemu.

Przeprowadzanie testów funkcjonalnych, w tym testów użyteczności interfejsu.

Implementacja klasy Autorzy, dostarczającej informacje o twórcach i możliwość przeglądania dokumentacji.

Wspólnie:

Rozwój głównej klasy widokowej "View" i wielokrotna zmiana jej koncepcji przez obu członków zespołu.

Uwagi:

W przypadku niejasności co do danej części kodu wspólnie szukaliśmy rozwiązania danego problemu, więc niełatwo jednoznacznie zamknąć się w sztywnych ramach obowiązku danego członka.

Współpraca i Koordynacja Prac

Praca zespołowa była kluczowym elementem projektu. Oba członki zespołu regularnie konsultowały się ze sobą, dzieląc się postępami i rozwiązując napotkane problemy.

Używane były narzędzia do zarządzania projektami oraz kontroli wersji (np. Git), co pozwalało na efektywną współpracę i śledzenie zmian w kodzie.

Doświadczenia i Wnioski

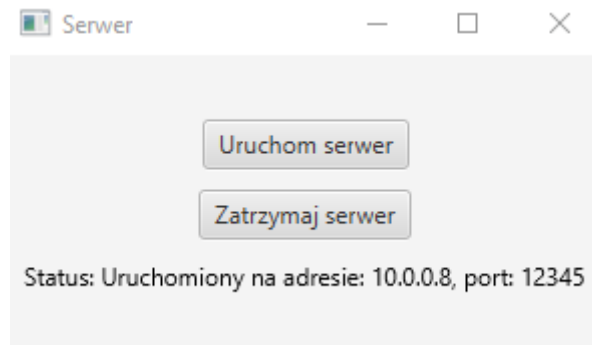
Nasz cały zespół zyskał cenne doświadczenie w pracy nad skomplikowanym projektem informatycznym, gdzie komunikacja i dobra organizacja okazały się tak samo ważne, jak umiejętności techniczne.

Współpraca wymagała elastyczności i umiejętności szybkiego adaptowania się do zmieniających się wymagań oraz rozwiązywania nieoczekiwanych problemów.

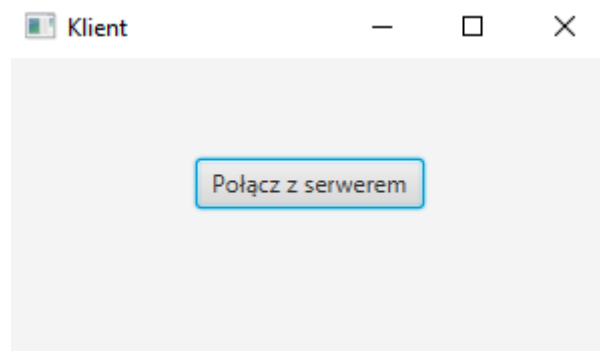
Projekt ukazał znaczenie podziału obowiązków i specjalizacji w pracy zespołowej, co pozwoliło na efektywniejsze wykorzystanie umiejętności i doświadczenia każdego członka zespołu.

Nasza aplikacja:

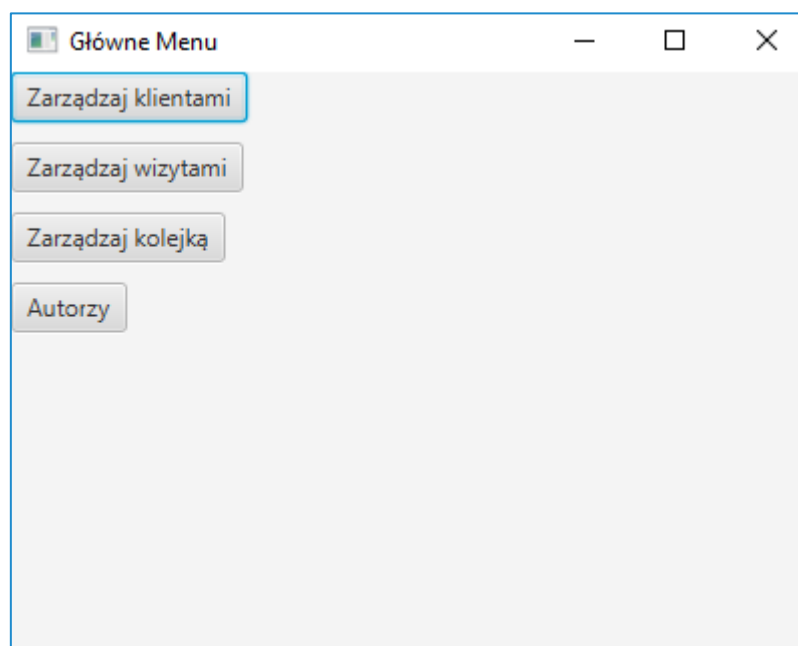
'Widok serwera'



'Widok klienta'



'Menu główne'



‘Widok umożliwiający zarządzanie bazą klientów’

Główne Menu

Dodaj klienta

Usuń klienta

Podaj imię i nazwisko klienta do Dodania lub usunięcia, np. Jan Kowalski

Edytuj klienta

Aktualne imię i nazwisko klienta

Nowe imię klienta

Nowe nazwisko klienta

Powrót

| Imię | Nazwisko | |
|-------|----------|--|
| Imię | nazwisko | |
| piotr | Wilk2 | |
| Piotr | Wilk | |
| Jakub | Paluch | |
| | | |
| | | |

‘Widok umożliwiający zarządzanie wizytami’

Główne Menu

01.02.2024

Wpisz Godzinę. Użyj formatu HH:mm

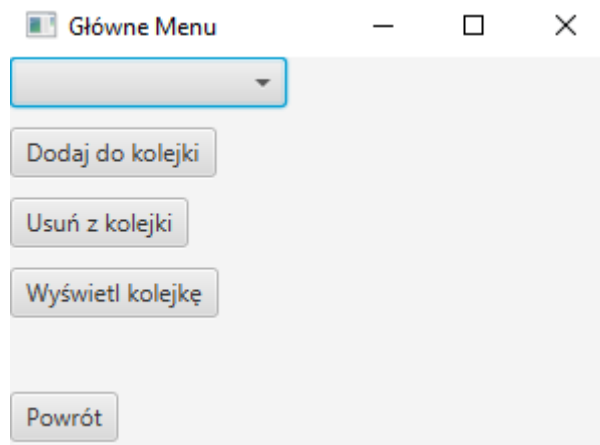
Dodaj wizytę

Wybierz ID wizyty do usunięcia:

Usuń wizytę

Powrót

‘Widok umożliwiający zarządzanie kolejką’



‘Widok wyświetlający autorów – ustawiony na sztywno, dodatkowo zawierający przycisk umożliwiający wyświetlanie sprawozdania’

