****

**Programowanie obiektowe**

*System Parkingowy*

Prowadzący: Autor:

mgr inż. Ewa Żesławska Wojciech Liszka

w65495

Kierunek: 3 IIZ/2021, grupa GP01

Rzeszów, r.a. 2022/2023

Spis treści

[**1.** **Cele projektu** 3](#_Toc128245297)

[**2.** **Opis techniczny projektu** 3](#_Toc128245298)

[**3.** **Harmonogram realizacji projektu** 4](#_Toc128245299)

[**4.** **Prezentacja warstwy użytkowej projektu** 4](#_Toc128245300)

[**5.** **Repozytorium, system kontroli wersji** 6](#_Toc128245301)

[**6.** **Podsumowanie** 6](#_Toc128245302)

[**7.** **Literatura** 6](#_Toc128245303)

1. **Cele projektu**

Głównym celem projektu jest stworzenie aplikacji umożliwiającej administrację parkingiem i umożliwienie klientom korzystanie z obiektu. Aplikacja wyświetla możliwe do wykonania funkcje w zależności od tego czy zalogowany jest administrator czy też użytkownik.

**Wymagania funkcjonalne**

**Administrator:**

* Dodawanie Klienta (należy podać imię, nazwisko, login, hasło).
* Usuwanie Klienta (należy podać id usuwanego klienta).
* Edycja atrybutów Klienta (należy podać id edytowanego klienta)
* Dodawanie Samochodu (należy podać markę, model, id właściciela,).
* Usuwanie Samochodu (należy podać id samochodu).
* Administratorskie zaparkowanie samochodu (należy podać id parkowanego pojazdu) nie posiada obostrzeń kto może zaparkować samochód.
* Administratorski wyjazd pojazdu (należy podać id pojazdu) nie posiada obostrzeń kto może wyjechać danym pojazdem z parkingu i nie dodaje opłaty za parkowanie do konta właściciela.
* Wyświetlenie wszystkich zaparkowanych pojazdów.
* Wyświetlenie wszystkich klientów.
* Wyświetlenie wszystkich samochodów zapisanych w bazie danych.
* Przyjęcie opłaty za parkowanie (wymaga podania id klienta).
* Wyświetlenie stanu parkingu – które miejsca są zajęte a które wolne.

**Użytkownik:**

* Wyświetlenie wolnych miejsc.
* Zaparkowanie samochodu (należy podać numer rejestracyjny pojazdu) ze sprawdzeniem czy dany pojazd należy do zalogowanego użytkownika.
* Wyjazd pojazdu (należy podać numer rejestracyjny pojazdu) sprawdza czy dany pojazd należy do zalogowanego użytkownika, obciąża konto użytkownika opłatą 3 złotych za każdą godzinę parkowania.
* Wyszukanie miejsca parkingowego które zajmuje pojazd (wymaga podania numeru rejestracyjnego pojazdu) jeżeli należy do zalogowanego użytkownika.
* Sprawdzenie stanu należności zalogowanego użytkownika.

**Wymagania niefunkcjonalne**

* Aplikacja jest prosta w obsłudze.
* Aplikacja stworzona w języku Java.
* Aplikacja nawiązuje połączenie z bazą danych i używa rekordów w niej zapisanych.

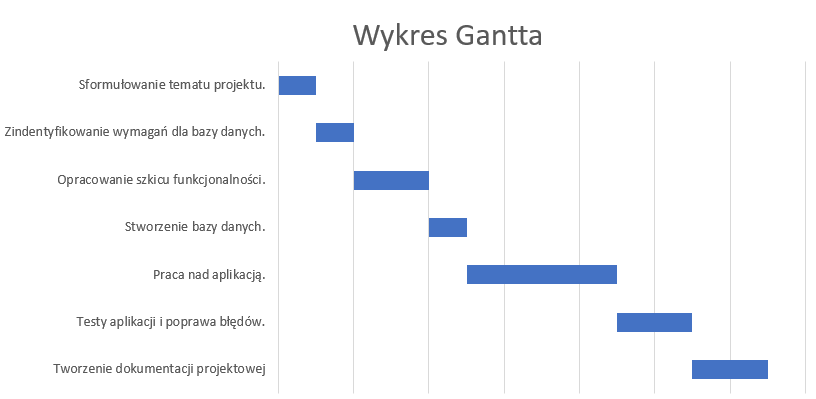
1. **Opis techniczny projektu**

Należy umieścić informacje odnośnie wykorzystywanego języka, narzędzi oraz minimalnych wymagań sprzętowych.

* Środowisko programistyczne Javy: Java JDK 19 Oracle version 19.0.1
* Środowisko programistyczne Javy: InteliJ IDEA 2022.2.3 (Community Edition)
* Baza danych PostgreSQL 15.2
* Biblioteka pgJDBC v42.5.3

1. **Harmonogram realizacji projektu**

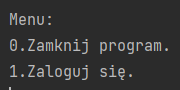
Poniżej zamieszono harmonogram realizacji projektu.



Rysunek 1. Diagram Gantta

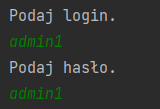
1. **Prezentacja warstwy użytkowej projektu**

Po uruchomieniu programu otrzymamy takie opcje:



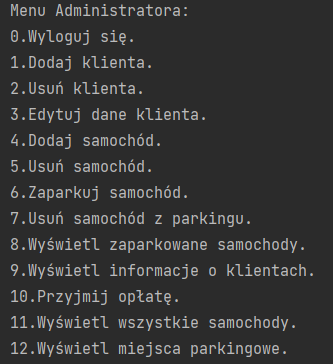
Rysunek 2. Pierwsze menu aplikacji.

Po wybraniu opcji „1”, aplikacja prosi o zalogowanie się (Rys.3).

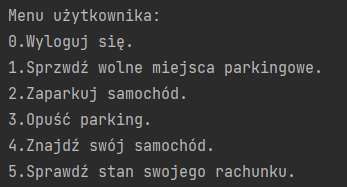


Rysunek 3. Logowanie.

W zależności od tego czy zalogujemy się atrybutami administratora (Rys.4), czy też użytkownika (Rys.5) otrzymamy dwa różne menu funkcji programu.



Rysunek 4. Funkcje dostępne dla administratora.



Rysunek 5. Funkcje dostępne dla administratora.

Opcje dostępne dla administratora oferują szeroki zakres funkcji służących do zarządzania systemem:

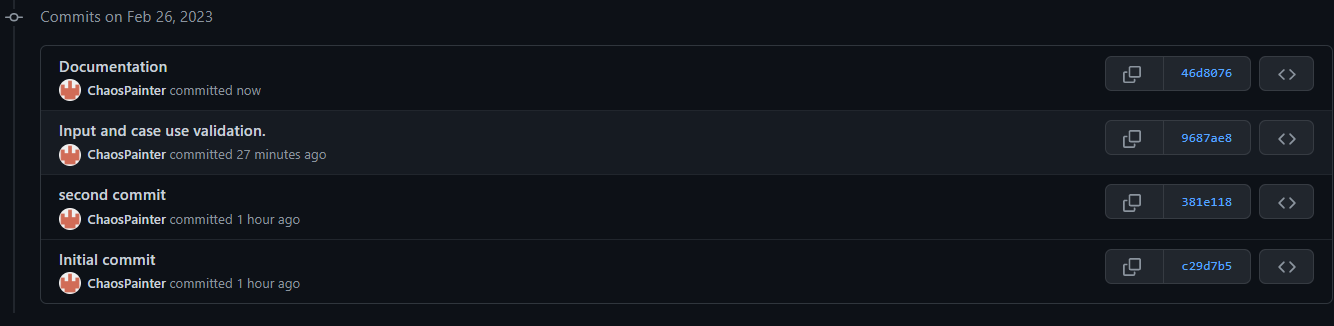
* Wyloguj. – powoduje przejście do poprzedniego menu.
* Dodaj klienta. – umożliwia dodanie do bazy danych nowego klienta po podaniu należących do niego atrybutów.
* Usuń klienta. – usuwa z bazy danych klienta o zadanym id.
* Edytuj dane klienta. – modyfikuje dane klienta o zadanym id, w bazie danych zgodnie z wpisanymi wartościami atrybutów.
* Dodaj samochód. – umożliwia dodanie do bazy danych samochodu zgodnie z podanymi wartościami atrybutów.
* Usuń samochód. – usuwa z bazy danych samochód o podanym id.
* Zaparkuj samochód. – funkcja pozwala na zaparkowanie dowolnego samochodu na dowolnym miejscu parkingu.
* Usuń samochód z parkingu. – funkcja służy do „ręcznego” wyprowadzenia dowolnego zaparkowanego samochodu bez obciążania konta właściciela pojazdu.
* Wyświetl zaparkowane pojazdy. – funkcja pozwala na wyświetlenie wszystkich zaparkowanych samochodów wraz z informacja gdzie się one znajdują.
* Wyświetl informacje o klientach. – wyświetla informacje o klientach znajdujące się w bazie danych.
* Przyjmij opłatę. – modyfikuje stan konta należący do klienta o podanym id, w bazie zgodnie z zadana kwota.
* Wyświetl wszystkie samochody. – wyświetla wszystkie samochody znajdujące się w bazie danych.
* Wyświetl miejsca parkingowe. – wyświetla informacje o stanie miejsca parkingowego oraz o jego położeniu.

Opcje dostępne dla użytkownika mają na celu umożliwienie wykonania prostych czynności związanych z użytkowaniem parkingu przez klienta:

* Wyloguj się. – powoduje powrót do poprzedniego menu.
* Sprawdź wolne miejsca parkingowe. – wyświetla listę wolnych miejsc parkingowych.
* Zaparkuj samochód. – po podaniu numeru rejestracyjnego pojazdu wyznacza właścicielowi miejsce parkingowe oraz zapisuje w bazie czas przyjazdu.
* Opuść parking. – po podaniu numeru rejestracyjnego pojazdu pozwala właścicielowi na opuszczenie parkingu, zapisuje w bazie czas wyjazdu i na podstawie długości pobytu obciąża konto właściciela.
* Znajdź swój samochód. – wyświetla właścicielowi miejsce na którym zaparkowany jest samochód o podanym numerze rejestracyjnym.
* Sprawdź stan rachunku. – wyświetla stan konta dla zalogowanego użytkownika.

1. **Repozytorium, system kontroli wersji**

Projekt został zrealizowany z wykorzystaniem systemu kontroli wersji GitHUB. Do projektu stworzono dokumentację. Dokumentacja oraz projekt został umieszczony w repozytorium dostępnym pod adresem: <https://github.com/ChaosPainter/System-Parkingowy-Projekt.git>



Rysunek 6. Historia komitów.

1. **Podsumowanie.**

Projekt systemu parkingowego skierowany jest w głównej mierze do firm. Pozwala on na zarządzanie informacjami dotyczącymi użytkowników oraz samego parkingu. Aplikacja spełnia wymogi funkcjonalne i niefunkcjonalne opisane w wymaganiach projektowych. Projekt można bez problemu rozwijać o kolejne funkcje administratorskie jak i opcje dostępne dla użytkowników. Podczas dalszego rozwoju aplikacji należy wprowadzić „Input Sanitization” pod katem potencjalnych kwerend SQL. Należy również rozszerzyć walidację informacji przyjmowanych z konsoli dla przypadków wpisania danych w innym formacie niż wymagany.

1. **Literatura**

Materiały źródłowe – wskazanie literatury i materiałów źródłowych wykorzystanych przy realizacji projektu.

1. Cay S. Horstmann, Gary Cornell, Java. Podstawy. Wydanie VIII,2008, Wyd. Helion.
2. Anthony Molinaro, SQL Cookbook, 2005, Wyd. O'Reilly Media, Inc.
3. <https://www.youtube.com/watch?v=o9dcSS_82gw> [dostęp 25.02.2023]
4. <https://www.youtube.com/watch?v=WFT5MaZN6g4> [dostęp 25.02.2023]