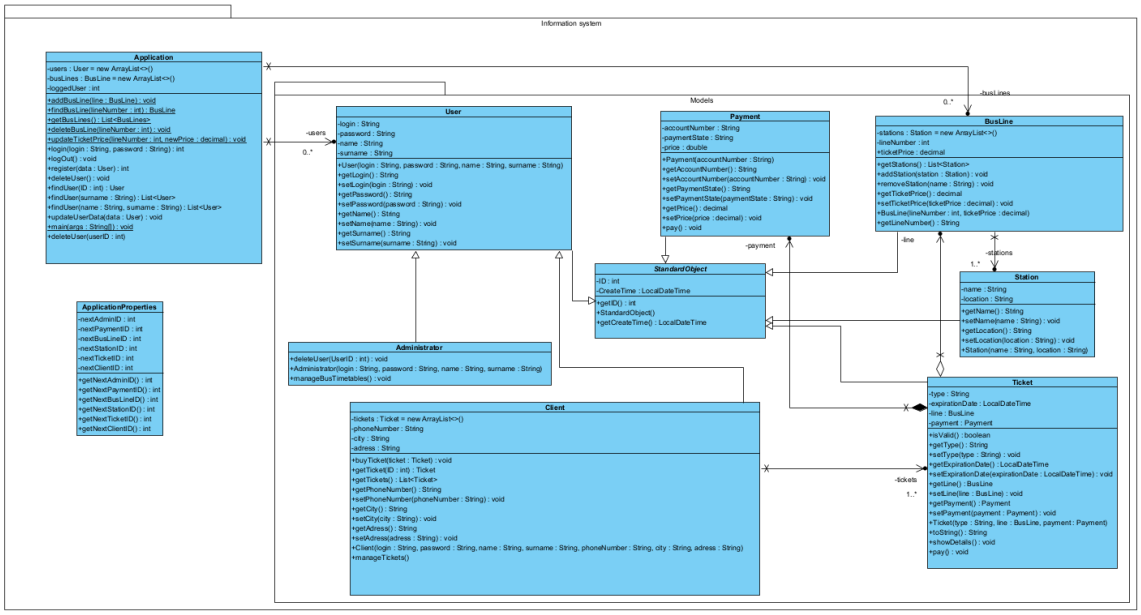
5 Etap - Program obsługujący system informacyjny linii autobusowych

Autorzy: Mateusz Błach (264010), Piotr Pozlewicz (256772), Mateusz Boroch (263899)

20 grudnia 2023

Spis treści

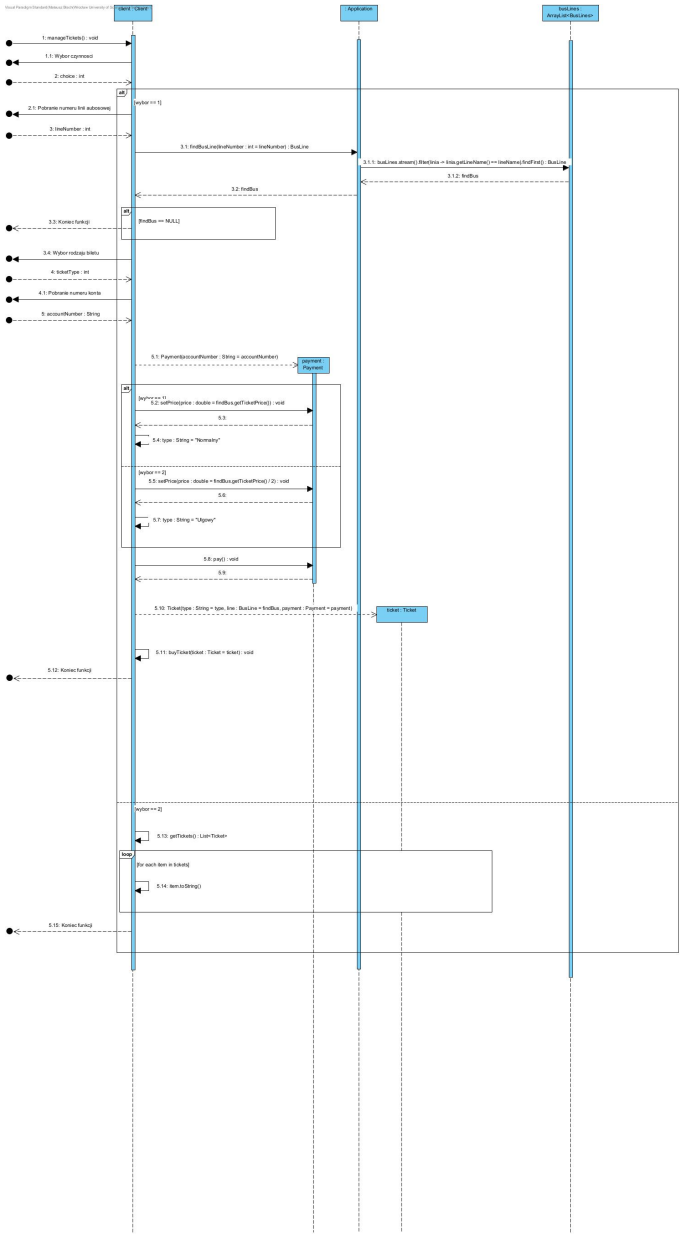
1 Diagram klas



Rysunek 1: Diagram klas

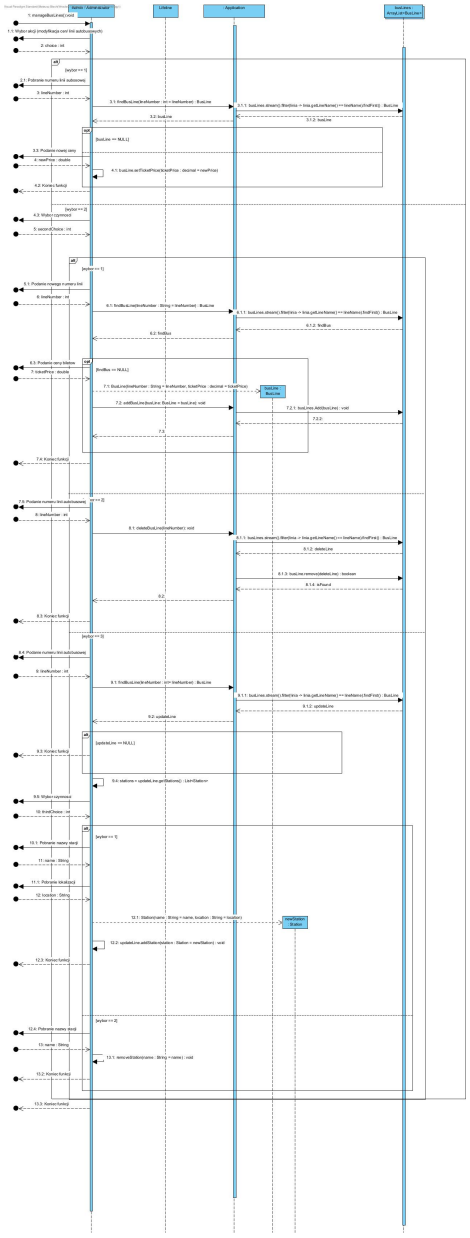
2 Diagramy sekwencji

2.1 Zarządzanie biletami



Rysunek 2: Diagram sekwencji - zarządzanie biletami

2.2 Zarządzanie rozkładami jazdy



Rysunek 3: Diagram sekwencji - zarządzanie rozkładami jazdy

3 Kod programu

3.1 Klasa Application

```
1  package businessLayer;
2
3  import businessLayer.BusLine;
4  import businessLayer.User;
5
6  import java.util.ArrayList;
7  import java.util.List;
8  import java.util.Objects;
9  import java.util.Scanner;
10
11 public class Application {
12     public static List<User> users = new ArrayList<User>();
13     public static List<BusLine> busLines = new ArrayList<>();
14     public int loggedUser = -1;
15     public static void main(String[] args) {
16         addBusLine(new BusLine(1,5.6));
17         addBusLine(new BusLine(2,3.1));
18     }
19
20     public static void addBusLine(BusLine line){
21         busLines.add(line);
22     }
23     public static BusLine findBusLine(int lineNumber) {
24         return busLines.stream()
25                 .filter(busline -> lineNumber == busline.getLineNumber())
26                 .findAny()
27                 .orElse(null);
28     }
29
30     public static List<BusLine> getBusLines(){
31         return busLines;
32     }
33     public static void deleteBusLine(int lineID){
34         busLines.remove(busLines.stream()
35                 .filter(busLine -> lineID == busLine.getID())
36                 .findAny()
37                 .orElse(null)
38         );
39     }
40     public static void updateLine(BusLine line){
41         //TODO
42     }
43     public static void updateTicketPrice(int lineNumber, double newPrice){
44         busLines.stream()
45                 .filter(busline -> lineNumber == busline.getLineNumber())
46                 .findAny()
47                 .orElse(null)
48                 .setTicketPrice(newPrice);
49     }
50
51     public void login(String login, String password){
52         loggedUser = Objects.requireNonNull(users.stream()
53                         .filter(user -> login.equals(user.getLogin()) && password.
   equals(user.getPassword()))
54                         .findAny()
55                         .orElse(null))
56                         .getID();
```

```
57      }
58
59      public void logOut(){
60          loggedUser = -1;
61      }
62
63      public void register(User user){
64          users.add(user);
65      }
66
67      public static void deleteUser(int ID){
68          users.remove(findUser(ID));
69      }
70
71      public static User findUser(int ID){
72          return users.stream()
73                  .filter(user -> ID == user.getID())
74                  .findAny()
75                  .orElse(null);
76      }
77
78      public void updateUserData(){
79          Scanner scanner = new Scanner(System.in);
80          System.out.println("Podaj ID uzytkownika ktorego chcesz zmodyfikowac: ");
81          int ID = Integer.parseInt(scanner.nextLine());
82          User user = findUser(ID);
83      }
84
85
86  }
```

3.2 Klasa ApplicationProperties

```
1   package properties;
2
3   public class ApplicationProperties {
4       static public int nextAdminID = 0;
5       static public int nextPaymentID = 0;
6       static public int nextBusLineID = 0;
7       static public int nextStationID = 0;
8       static public int nextTicketID = 0;
9       static public int nextClientID = 0;
10      static public int getNextAdminID(){
11          int id = nextAdminID;
12          nextAdminID++;
13          return id;
14      }
15      static public int getNextPaymentID(){
16          int id = nextPaymentID;
17          nextPaymentID++;
18          return id;
19      }
20
21      static public int getNextBuslineID(){
22          int id = nextBusLineID;
23          nextBusLineID++;
24          return id;
25      }
26      static public int getNextStationID(){
27          int id = nextStationID;
28          nextStationID++;
```

```
        return id;
    }
    static public int getNextTicketID(){
        int id = nextTicketID;
        nextTicketID++;
        return id;
    }

    static public int getNextClientID(){
        int id = nextClientID;
        nextClientID++;
        return id;
    }

}
```

3.3 Klasa StandardObject

```
package businessLayer;

import java.time.LocalDateTime;

public class StandardObject {


    protected int ID;

    private LocalDateTime timeCreated;

    public StandardObject() {
        this.timeCreated = LocalDateTime.now();
    }

    public int getID() {
        return ID;
    }
    public LocalDateTime getTimeCreated() {
        return timeCreated;
    }
}
```

3.4 Klasa User

```
package businessLayer;


public class User extends StandardObject{
    private String login;
    private String password;
    private String name;
    private String surname;

    public User(String login, String password, String name, String surname) {
        super();
        this.login = login;
        this.password = password;
        this.name = name;
        this.surname = surname;
    }

    public String getLogin() {
```

```
        return login;
    }

    public void setLogin(String login) {
        this.login = login;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getSurname() {
        return surname;
    }

    public void setSurname(String surname) {
        this.surname = surname;
    }
}
```

3.5 Klasa Client

```
package businessLayer;

import properties.ApplicationProperties;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Client extends User{
    private List<Ticket> tickets = new ArrayList<>();

    private String phoneNumber;
    private String city;
    private String adress;

    public Client(String login, String password, String name, String surname,
                  String phoneNumber, String city, String adress) {
        super(login, password, name, surname);
        super.ID = ApplicationProperties.getNextClientID();
        this.phoneNumber = phoneNumber;
        this.city = city;
        this.adress = adress;
    }

    public void buyTicket(Ticket ticket){
        this.tickets.add(ticket);
    }
```

```
28
29      public void manageTickets(){
30          System.out.println("""
31                  Wybierz akcje:
32                  1. Zakup biletu
33                  2. Sprawdzenie statusu posiadanych biletow
34                  """);
35          Scanner scanner = new Scanner(System.in);
36          int choice = Integer.parseInt(scanner.nextLine());
37          switch (choice){
38              case 1:
39                  System.out.println("Podaj numer lini autobusowej");
40                  int lineNumber = Integer.parseInt(scanner.nextLine());
41                  BusLine findBus = Application.findBusLine(lineNumber);
42                  if(findBus != null){
43                      System.out.println("""
44                              Wybierz typ biletu
45                              1. Normalny
46                              2. Ulgowy
47                              """);
48                      int ticketType = Integer.parseInt(scanner.nextLine());
49                      System.out.println("Podaj numer konta:");
50                      String accountNumber = scanner.nextLine();
51                      Payment payment = new Payment(accountNumber);
52                      String type = "";
53                      switch (ticketType){
54                          case 1:
55                              payment.setPrice(findBus.getTicketPrice());
56                              type = "normalny";
57                              break;
58                          case 2:
59                              payment.setPrice(findBus.getTicketPrice()/2);
60                              type = "ulgowy";
61                              break;
62                          default:
63                              break;
64                      }
65                      payment.pay();
66                      this.buyTicket(new Ticket(type,findBus,payment));
67                  }
68                  break;
69              case 2:
70                  for(Ticket ticket : tickets){
71                      System.out.println(ticket.toString());
72                  }
73                  break;
74              default:
75                  break;
76          }
77      }
78
79      public List<Ticket> getTickets() {
80          return tickets;
81      }
82
83
84      public String getPhoneNumber() {
85          return phoneNumber;
86      }
87
```

```
88      public void setPhoneNumber(String phoneNumber) {
89          this.phoneNumber = phoneNumber;
90      }
91
92      public String getCity() {
93          return city;
94      }
95
96      public void setCity(String city) {
97          this.city = city;
98      }
99
100     public String getAdress() {
101         return adress;
102     }
103
104     public void setAdress(String adress) {
105         this.adress = adress;
106     }
107
108 }
```

3.6 Klasa Administrator

```
1  package businessLayer;
2
3  import properties.ApplicationProperties;
4
5  import java.util.Scanner;
6
7  public class Administrator extends User {
8      public Administrator(String login, String password, String name, String surname)
       {
9          super(login,password,name,surname);
10         super.ID = ApplicationProperties.getNextAdminID();
11     }
12
13     private void deleteUser(int userID){
14         Application.deleteUser(userID);
15     }
16
17     private void manageBusLinesTimetables(){
18         System.out.println("""
19                 Wybierz akcje:
20                 1. Modyfikacja cen
21                 2. Modyfikacja lini autobusowych
22                 """);
23         Scanner scanner = new Scanner(System.in);
24         int lineNumber;
25         BusLine busLine;
26         double newPrice;
27         int lineID;
28         int choice = Integer.parseInt(scanner.nextLine());
29         switch (choice){
30             case 1:
31                 System.out.println("Podaj numer lini autobusowej");
32                 lineNumber = Integer.parseInt(scanner.nextLine());
33                 busLine = Application.findBusLine(lineNumber);
34                 if(busLine != null){
35                     System.out.println("Podaj nowa cene biletu");
36                     newPrice = Double.parseDouble(scanner.nextLine());
```

```
                    busLine.setTicketPrice(newPrice);
                }
                break;
            case 2:
                System.out.println("""
                        Wybierz akcje:
                        1. Dodanie nowej lini autobusowej
                        2. Usuni Źcie lini autobusowej
                        3. Modyfikacja lini autobusowej
                        """);
                int secondChoice = Integer.parseInt(scanner.nextLine());
                switch (secondChoice){
                    case 1:
                        System.out.println("Podaj numer lini autobusowej");
                        lineNumber = Integer.parseInt(scanner.nextLine());
                        busLine = Application.findBusLine(lineNumber);
                        if(busLine != null){
                            System.out.println("Podaj cene biletu");
                            newPrice = Double.parseDouble(scanner.nextLine());
                            Application.addBusLine(new BusLine(lineNumber,newPrice));
                        }
                        break;
                    case 2:
                        System.out.println("Podaj ID lini autobusowej");
                        lineID = Integer.parseInt(scanner.nextLine());
                        Application.deleteBusLine(lineID);
                        break;
                    case 3:
                        System.out.println("Podaj numer lini autobusowej");
                        lineNumber = Integer.parseInt(scanner.nextLine());
                        BusLine updateLine = Application.findBusLine(lineNumber);
                        if(updateLine == null) break;
                        System.out.println("""
                                Wybierz akcje:
                                1. Dodaj stacje do lini autobusowej
                                2. Usun stacje z lini autobusowej
                                """);
                        int thirdChoice = Integer.parseInt(scanner.nextLine());
                        String name;
                        switch (thirdChoice){
                            case 1:
                                System.out.println("Podaj nazwe stacji");
                                name = scanner.nextLine();
                                System.out.println("Podaj lokalizacje");
                                String location = scanner.nextLine();
                                updateLine.addStation(new Station(name,location));
                                break;
                            case 2:
                                System.out.println("Podaj nazwe stacji");
                                name = scanner.nextLine();
                                updateLine.removeStation(name);
                                break;
                            default:
                                break;
                        }
                        break;
                    default:
                        break;
                }
                break;
```

```
            default:
                break;
        }


    }
}
```

3.7 Klasa Station

```
package businessLayer;

import properties.ApplicationProperties;

public class Station extends StandardObject{
    private String name;
    private String location;
    public Station(String name, String location){
        super();
        this.name = name;
        this.location = location;
        super.ID = ApplicationProperties.getNextStationID();
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getLocation() {
        return location;
    }

    public void setLocation(String location) {
        this.location = location;
    }
}
```

3.8 Klasa BusLine

```
package businessLayer;

import properties.ApplicationProperties;

import java.util.ArrayList;
import java.util.List;

public class BusLine extends StandardObject{
    private List<Station> stations;
    private final int lineNumber;
    private double ticketPrice;

    public BusLine(int lineNumber,double ticketPrice){
        super();
        this.lineNumber = lineNumber;
        this.ticketPrice = ticketPrice;
        this.stations = new ArrayList<>();
        super.ID = ApplicationProperties.getNextBuslineID();
```

```
    }
    public List<Station> getStations() {
        return stations;
    }

    public double getTicketPrice() {
        return ticketPrice;
    }

    public void addStation(Station station){
        this.stations.add(station);
    }

    public void removeStation(String name){
        for(Station station :stations){
            if(station.getName().equals(name)){
                stations.remove(station);
                break;
            }
        }
    }

    public void setTicketPrice(double ticketPrice) {
        this.ticketPrice = ticketPrice;
    }
    public int getLineNumber() {
        return lineNumber;
    }
}
```

3.9 Klasa Ticket

```
package businessLayer;

import properties.ApplicationProperties;

import java.time.LocalDateTime;

public class Ticket extends StandardObject{
    private String status;
    private String type;
    private LocalDateTime expirationDate;
    private BusLine line;
    private Payment payment;

    public Ticket(String type, BusLine line, Payment payment){
        super();
        super.ID = ApplicationProperties.getNextTicketID();
        this.type = type;
        this.line = line;
        this.payment = payment;
        this.expirationDate = LocalDateTime.now().plusHours(1);
    }

    boolean isValid(){
        return LocalDateTime.now().isAfter(super.getTimeCreated());
    }

    public String getStatus() {
        return status;
    }
```

```
30
31      public void setStatus(String status) {
32          this.status = status;
33      }
34
35      public String getType() {
36          return type;
37      }
38
39      public void setType(String type) {
40          this.type = type;
41      }
42
43      public BusLine getLine() {
44          return line;
45      }
46
47      public void setLine(BusLine line) {
48          this.line = line;
49      }
50
51      public Payment getPayment() {
52          return payment;
53      }
54
55      public void setPayment(Payment payment) {
56          this.payment = payment;
57      }
58
59      @Override
60      public String toString() {
61          return "Bilet{" +
62                  "status='" + status + '\'' +
63                  ", typ='" + type + '\'' +
64                  ", data wygasniecia=" + expirationDate +
65                  ", linia=" + line.getLineNumber() +
66                  ", status platnosci=" + payment.getPaymentState() +
67                  ", ID=" + ID +
68                  '}';
69      }
70  }
```

3.10 Klasa Payment

```
1   package businessLayer;
2
3   import properties.ApplicationProperties;
4
5   public class Payment extends StandardObject{
6       private String accountNumber;
7       private String paymentState;
8
9
10
11      private double price;
12      public Payment(String accountNumber){
13          super();
14          this.accountNumber = accountNumber;
15          super.ID = ApplicationProperties.getNextPaymentID();
16          this.paymentState = "unpaid";
17      }
```

```
18
19      public String getAccountNumber() {
20          return accountNumber;
21      }
22
23      public void setAccountNumber(String accountNumber) {
24          this.accountNumber = accountNumber;
25      }
26
27      public String getPaymentState() {
28          return paymentState;
29      }
30
31      public void setPaymentState(String paymentState) {
32          this.paymentState = paymentState;
33      }
34
35      public double getPrice() {
36          return price;
37      }
38
39      public void setPrice(double price) {
40          this.price = price;
41      }
42
43      public void pay(){
44          this.paymentState = "paid";
45      }
46  }
```