



SVEUČILIŠTE U MOSTARU  
FAKULTET STROJARSTVA RAČUNARSTVA I ELEKTROTEHNIKE

## **Programirnje u Javi**

*Projektni zadatak: **Bus Management System***

**Petar Bošnjak**

**Martina Vukanović**

Akadska godina 2021/2022.

## PROJEKTNI ZADATAK:

---

### Opis problema:

Detaljno opisati problem koji je potrebno riješiti izgradnjom programske podrške. Svaki student piše svoju temu minimalno 500 znakova.

Aplikacija "Bus system managment" služi za povezivanje autobusnog poduzeća s njihovim klijentima te olakšava sam postupak osiguravanja prijevoza.

Korisnik nakon pokretanja aplikacije dobije sve informacije vezane za danu relaciju odabirom na nju. Korisničkim sučeljem omogućen je odabir sjedišta u autobusu te kupovina karte za odabranu relaciju. Nakon uspješnog odabira karte sjedište u autobusu postane nedostupno za ostale korisnike.

Administrator aplikacije ima mogućnost pregleda tablica iz baze podataka (Bus, User, Relations). U pregledu također ima izbor dodavanja, brisanja ili uređivanja vrijednosti u tablicama.

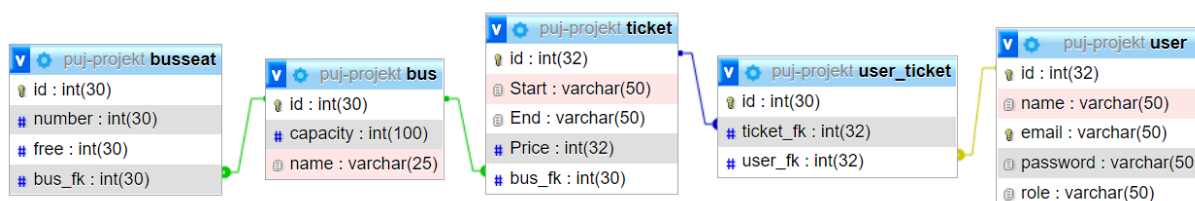
Svrha projekta jeste olakšavanje kupovine karte za neku relaciju. Korisniku je pojednostavljen pristup samoj kupovini time što je dobiven izbor autobusa, svih relacija te rezervacija jedinstvenog sjedišta u odabranom autobusu.

## 1. Modeliranje podataka

Prvi dio zadatka se radi u .

- Poglavlje 1.1. (Relacijski model podataka) - potrebno je staviti sliku relacijskog modela kreiranog u MS Accessu.
- Poglavlje 1.2. (Opis tablica u relacijskom modelu) - potrebno je navesti i opisati sve tablice (i atribute unutar tablica) iz dobivenog modela te na kraju priložiti sliku
- Poglavlje 1.3. (Aplikacija u Javi) - sadrži screenshotove kreiranih formi Javi.

### 1.1. Relacijski model podataka



### 1.2. Opis tablica u relacijskom modelu

#### Busseat

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
id	INT(PK/NOT NULL)	ID sjedišta
number	INT(NOT NULL)	Broj sjedišta u autobusu
free	INT(NOT NULL)	Kontrolna vrijednost za provjeravanje dostupnosti sjedišta
Bus_fk	INT(FK/NOT NULL)	Strani ključ za vrijednost 'id' iz tablice 'bus'

#### Bus

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
id	INT (PK, NOT NULL)	ID autobusa
capacity	INT (NOT NULL)	Kapacitet autobusa
name	VARCHAR (NOT NULL)	Naziv poduzeća

#### Ticket

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
id	INT(PK/NOT NULL)	ID karte
Start	VARCHAR (NOT NULL)	Polazna točka
End	VARCHAR(NOT NULL)	Odredište
Price	INT (NOT NULL)	Cijena karte
Bus_fk	INT(FK/NOT NULL)	Strani ključ za vrijednost 'id' iz tablice 'bus'

## User ticket

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
id	INT(PK/NOT NULL)	ID pomoćne tablice
Ticket_fk	INT(FK/NOT NULL)	Strani ključ za vrijednost 'Id' iz tablice 'ticket'
User_fk	INT(FK/NOT NULL)	Strani ključ za vrijednost 'Id' iz tablice 'user'


## User

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
id	INT(PK/NOT NULL)	ID korisnika
name	INT(NOT NULL)	Korisničko ime
email	INT(NOT NULL)	Email adresa korisnika
password	INT(FK/NOT NULL)	Lozinka
role	VARCHAR (NOT NULL)	Uloga korisnika u sustavu

Table	Action	Rows	Type	Collation	Size
<input type="checkbox"/> bus	Browse  Structure  Search  Insert  Empty  Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB
<input type="checkbox"/> busseat	Browse  Structure  Search  Insert  Empty  Drop	221	InnoDB	utf8mb4_unicode_ci	32.0 KiB
<input type="checkbox"/> ticket	Browse  Structure  Search  Insert  Empty  Drop	4	InnoDB	utf8mb4_unicode_ci	32.0 KiB
<input type="checkbox"/> user	Browse  Structure  Search  Insert  Empty  Drop	39	InnoDB	utf8mb4_unicode_ci	32.0 KiB
<input type="checkbox"/> user_ticket	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB
5 tables	Sum	270	InnoDB	utf8mb4_general_ci	160.0 KiB

### 1.3. Aplikacija u Javi

Register



User name

Email address

Password

Register

Already have an account?

Log in

Email address

Password

Log in

Don't have an account?



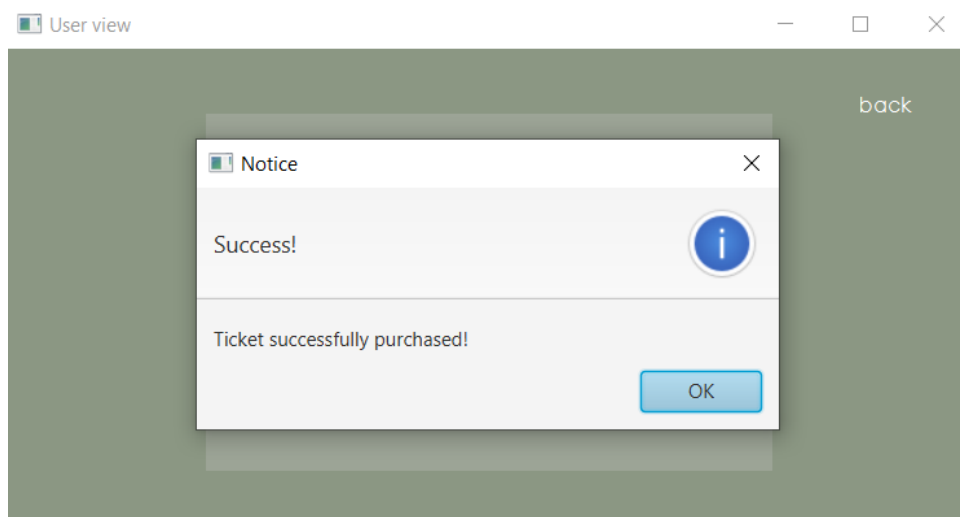
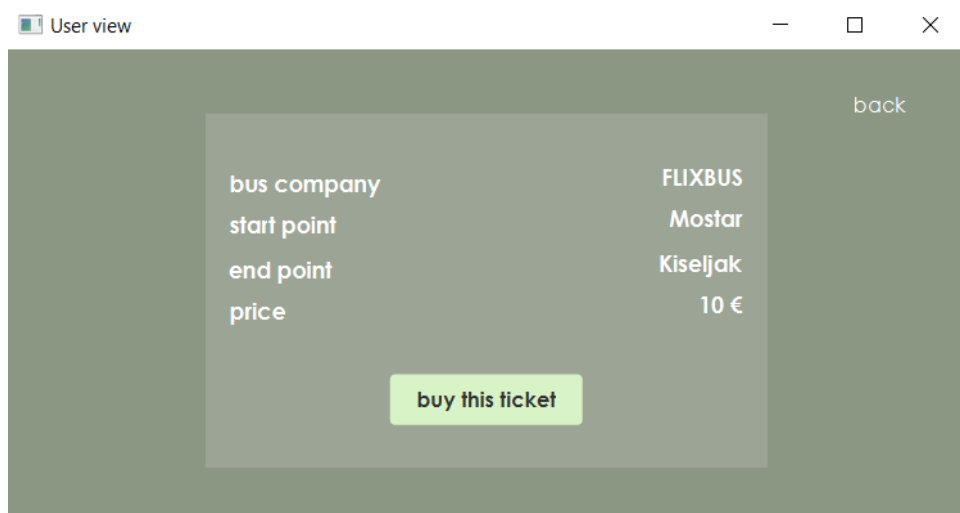
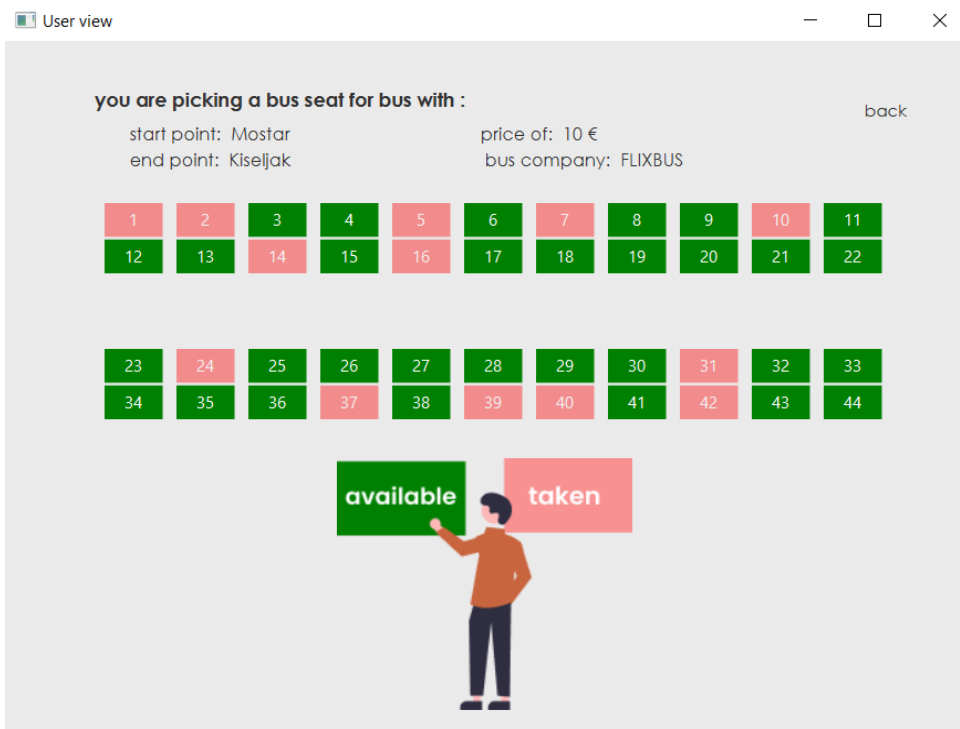
User view

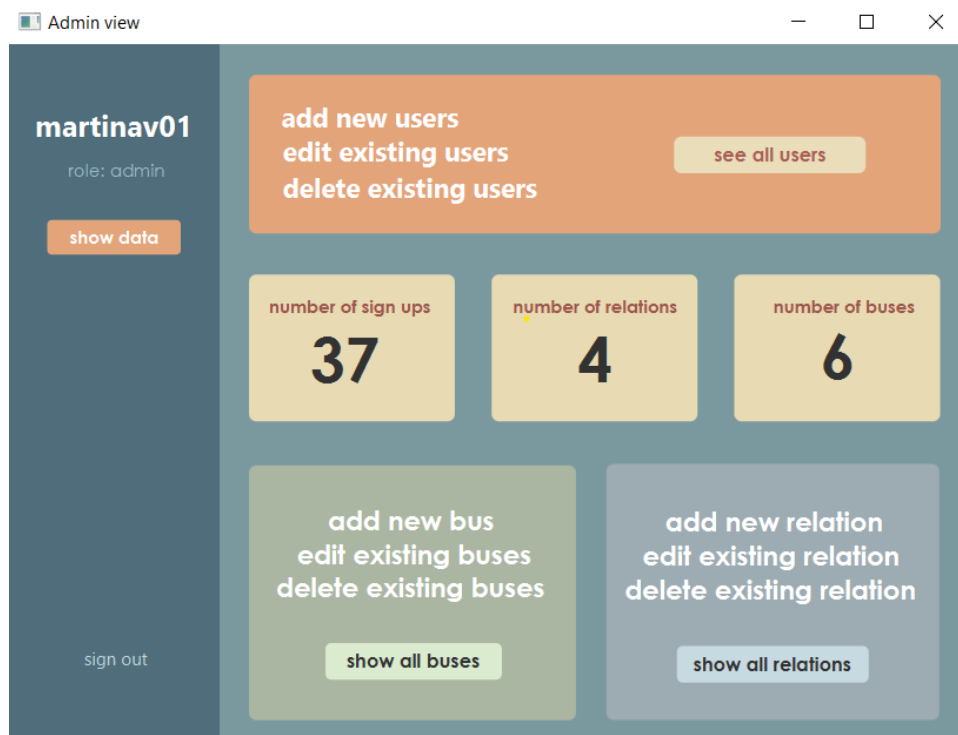
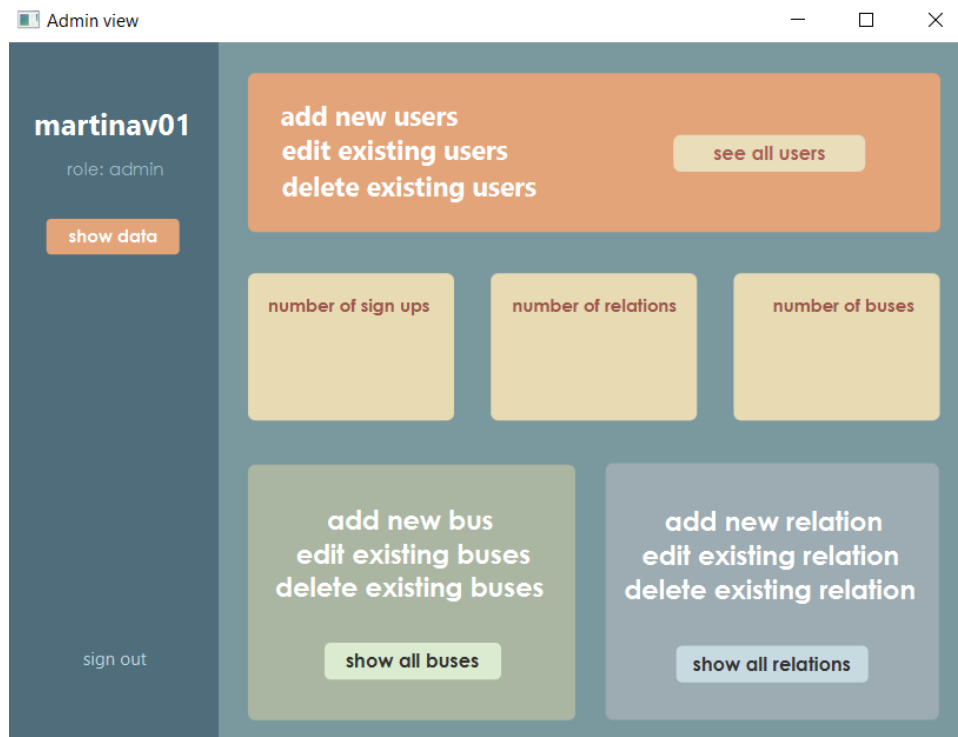
Welcome Sara !

sign out

start	end	price
Mostar	Kiseljak	10
Mostar	Split	30
Mostar	Zagreb	60
Split	Zagreb	40

choose this relation







all users

back

All fields are required!

id	name	email	role
1	Petar	pero@gmail.com	admin
2	Martina	martina@gmail.c...	admin
3	Josip	josip@gmail.com	user
4	Sara	sara@gmail.com	user
5	Norma	nchadbourn4@c...	user
6	Merrily	mgierardi5@disc...	user
7	Christy	csteeples6@yolas...	user

user name

martina

email address

martina@gmail.com

role

admin or user

password

\*\*\*\*\*

add/edit user

delete user

all buses

back

id	name	capacity
1	FLIXBUS	44
2	GLOBTOUR	40
3	Croatia Bus	36
4	Centrotans	28
5	Arriva trans	32

name

flixbus

capacity

44

add/edit bus

delete bus

all relations

back

start	end	price	bus company
Mostar	Kiseljak	10	1
Mostar	Split	30	2
Mostar	Zagreb	60	3
Split	Zagreb	40	4

id	name
1	FLIXBUS
2	GLOBTOUR
3	Croatia Bus
4	Centrotans
5	Arriva trans
6	FLIXBUS

start

Mostar

end

Zagreb

price

30 (e)

bus company

flixbus

add/edit relation

delete relation

## 2. Model programske podrške

Drugi dio zadatka se odnosi na modeliranje programske podrške, detaljno ćete raspisati sve klase i na kraju priložiti dijagram klase koji opisuje izgled programske podrške:

- Napraviti class dijagram
- Opisati metode koje se nalaze unutar svake klase

U navedenim klasama potrebno je opisati :

- 1) Paketi – kratko opisati paket vaše programske podrške
- 2) Klase – kratko opisati klase koje koristite (ukratko za što će se klasa koristiti
  - a. Atribute – kratko pisati attribute koje se nalaze unutar klase;
  - b. Metode – kratko opisati metode koje se koriste za što služe i opišite njihov povratni tip;

### 2.1. Paketi

Projekt “Bus Management System” sastoji se od dva paketa (com.example.pujprojekt i resources).

**com.example.pujprojekt** paket sastoji se od dva podpaketa (controller i model) i klase Main koja pokreće glavni program.

**controller** paket sastoji se od 9 klasa tj. kontrolera. Oni upravljaju protokom podataka u objektu modela i ažuriraju pogled kad god se podaci promijene te osiguravaju samu funkcionalnost aplikacije.

**model** sadrži strukturu baze podataka.

**resources** sadrži 9 FXML datoteka koji su zaslužni za frontend tj. za izgled korisničkog sučelja, te img folder s korištenim slikama u korisničkom sučelju.

## 2.2. Klase - model



### 2.2.1. Klasa Bus

---

Klasa Bus se koristi za pohranjivanje podataka vezanu za atribute autobusa.

#### 2.2.1.1. Atributi

---

**@Entity(type="INTEGER", size=30, primary = true)**

**int id;**

-ID autobusa

**@Entity(type="INTEGER", size=100)**

**int capacity;**

-Kapacitet autobusa

**@Entity(type="VARCHAR", size=25)**

**String name;**

-Ime poduzeća

#### 2.2.1.2. Metode

---

**public int getCapacity();** - metoda koja vraća kapacitet autobusa

**public void setCapacity(int capacity);** - metoda koja postavlja vrijednost kapaciteta

**public void getID();** - metoda koja vraća ID autobusa

**public void setID(int id);** - metoda koja postavlja vrijednost ID

**public string getName();** - vraća vrijednost 'name' (tj. ime) poduzeća

**public string setName(string name);** - postavlja vrijednost 'name' (tj. ime) poduzeća

### 2.2.2. Klasa Ticket

---

Klasa Ticket sadrži podatke o korisniku, relaciji te poduzeću koja će izvršiti putovanje relacije.

#### 2.2.2.1. Atributi

---

**@Entity(type="INTEGER", size=32, primary = true)**

**int id;**

-ID karte

**@Entity(type="VARCHAR", size=50)**

**String Start;**

-Atribut pokazuje polaznu točku relacije

**@Entity(type="VARCHAR", size=50)**  
**String End;**

-Atribut pokazuje odredište realacije

**@Entity(type="INTEGER", size=32)**  
**int Price;**

-Cijena karte

**@Entity(type="INTEGER", size=30)**  
**@ForeignKey(table = "Bus", attribute = "id")**  
**int bus\_fk;**

-Strani ključ za tablicu bus

**@Entity(type="INTEGER", size=32)**  
**@ForeignKey(table = "User", attribute = "id")**  
**int user\_fk;**

-Strani ključ za tablicu user

### ***2.2.2.2. Metode***

---

**public int getPrice()** - vraća vrijednost cijene

**public void setPrice(int price)** - postavlja vrijednost cijene

**public User getUser\_fk()** - vraća vrijednost stranog ključa korisnika

**public void setUser\_fk(int user\_fk)** - postavlja vrijednost stranog ključa korisnika

**public int getId()** - vraća vrijednost ID

**public void setId(int id)** - postavlja vrijednost ID

**public String getStart()** - vraća naziv polazne točke relacije

**public void setStart(String start)** - postavlja vrijednost polazne točke relacije

**public String getEnd()** - vraća naziv odredišta relacije

**public void setEnd(String end)** - postavlja naziv odredišta relacije

**public int getBus\_fk()** - vraća vrijednost stranog ključa tablice bus

**public void setBus\_fk(int bus\_fk)** - postavlja vrijednost stranog ključa tablice bus

### 2.2.3. Klasa User

---

Klasa User sadrži sve podatke vezane za korisnika

#### 2.2.3.1. Atributi

---

**@Entity(type = "INTEGER", size = 32, primary = true)**

**int id;**

-ID korisnika

**@Entity(type="VARCHAR", size=50, isnull = false)**

**String name;**

-Ime korisnika

**@Entity(type="VARCHAR", size=50, isnull = false)**

**String email;**

-email korisnika

**@Entity(type="VARCHAR", size=50, isnull = false)**

**String password;**

-loznika korisnika

**@Entity(type="VARCHAR", size=50, isnull = false)**

**String role;**

-uloga korisnika(admin/user)

#### 2.2.3.2. Metode

---

**public int getId()** - vraća ID korisnika

**public void setId(int id)** - postavlja ID korisnika

**public String getName()** - vraća ime korisnika

**public void setName(String name)** - postavlja ime korisnika

**public String getEmail()** - vraća email korisnika

**public void setEmail(String email)** - postavlja email korisnika

**public String getPassword()** - vraća lozinku korisnika

**public void setPassword(String password)** - postavlja lozinka korisnika

**public String getRole()** - vraća ulogu korisnika

**public void setRole(String role)** - postavlja ulogu korisnika

## 2.2.4. Klasa Table

---

Klasa Table koristi se za CRUD operacije nad bazom podataka.

### 2.2.4.1. Metode

---

**getTableName(Class)** - vraća naziv tablice

**getAttributeName** - vraća naziv atributa

**create(Class)** - stvara novu tablicu

**save()** - sprema novu vrijednost u tablicu

**update()** - ažurira vrijednosti u tablici

**delete()** - briše vrijednosti iz tablice

**get(Class, int)** - dohvaća vrijednosti iz tablice

**list(Class), listTicket(Class)** - stvara listu sa svim vrijednostima iz tablice

## 2.2.5. Klasa BusSeat

---

Klasa BusSeat sadrži sve podatke vezane za sjedište u autobusu

### 2.2.5.1. Atributi

---

**@Entity(type="INTEGER", size=30, primary = true)**  
int id;

**@Entity(type="INTEGER", size=30)**  
int number;

**@Entity(type="INTEGER", size=30)**  
int free;

**@Entity(type="INTEGER", size=30)**  
**@ForeignKey(table = "Bus", attribute = "id")**  
int bus\_fk;

### 2.2.5.2. Metode

---

**public getNumber()** - vraća broj sjedišta

**public setNumber()** - postavlja broj sjedišta

**public getId()** - vraća ID sjedišta

**public setId()** - postavlja ID sjedišta

**public getFree()** - vraća je li sjedište slobodno ili zauzeto (0 ili 1)

**public setFree()** - postavlja sjedište na slobodno ili zauzeto (0 ili 1)

**public getBus\_fk()** - vraća strani ključ tablice Bus

**public setBus\_fk()** - postavlja strani ključ tablice Bus

### 2.2.6. Klasa User\_Ticket

---

Klasa User\_Ticket stvorena je kao pomoćna tablica i sadrži strane ključeve koji referenciraju na primarne ključeve tablica User i Ticket.

#### 2.2.6.1. Atributi

---

**@Entity(type="INTEGER", size=30, primary = true)**  
int id;

**@Entity(type="INTEGER", size=32)**  
**@ForeignKey(table = "Ticket", attribute = "id")**  
int ticket\_fk;

**@Entity(type="INTEGER", size=32)**  
**@ForeignKey(table = "User", attribute = "id")**  
int user\_fk;

#### 2.2.6.3. Metode

---

**public getId()** - vraća ID pomoćne tablice

**public setId()** - postavlja ID pomoćne tablice

**public getTicket\_fk()** - vraća vrijednost stranog ključa za tablicu Ticket

**public setTicket\_fk()** - postavlja vrijednost stranog ključa za tablicu Ticket

**public getUser\_fk()** - vraća vrijednost stranog ključa za tablicu User

**public setUser\_fk()** - postavlja vrijednost stranog ključa za tablicu User



## 2.3 Klase - controller

<ul style="list-style-type: none"> <li>AdminPanelController <ul style="list-style-type: none"> <li>display(String): void</li> <li>GoToLogin(): void</li> <li>goToAllUsers(): void</li> <li>goToAllBuses(): void</li> <li>goToAllRelations(): void</li> <li>show(): void</li> </ul> </li> <li>connection: Connection <ul style="list-style-type: none"> <li>getConnection(): Connection</li> <li>labela: Label</li> <li>adminName: Label</li> <li>numberOfSignUps: Label</li> <li>numberOfRelations: Label</li> <li>numberOfBuses: Label</li> <li>button: Button</li> <li>bazaLink: Connection</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>BusLayoutController <ul style="list-style-type: none"> <li>goToUserView(): void</li> <li>make(int): void</li> <li>display(int, String, String, int): void</li> </ul> </li> <li>connection: Connection <ul style="list-style-type: none"> <li>getConnection(): Connection</li> <li>id_BusSeat: Integer</li> <li>numberOfSeat: Integer</li> <li>capacity_Bus: Integer</li> <li>free: Integer</li> <li>username: String</li> <li>labela: Label</li> <li>startPoint: Label</li> <li>endPoint: Label</li> <li>Price: Label</li> <li>busCompany: Label</li> <li>container2: HBox</li> <li>container21: HBox</li> <li>container22: HBox</li> <li>container221: HBox</li> <li>botun: Button</li> <li>labels: List&lt;Button&gt;</li> <li>stage: Stage</li> <li>scene: Scene</li> <li>root: Parent</li> <li>bazaLink: Connection</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>LoginController <ul style="list-style-type: none"> <li>GoToRegister(): void</li> <li>Login(ActionEvent): void</li> <li>validateLogin(): void</li> </ul> </li> <li>connection: Connection <ul style="list-style-type: none"> <li>getConnection(): Connection</li> <li>errorMsg: Label</li> <li>UserEmail: TextField</li> <li>UserPassword: PasswordField</li> <li>loginButton: Button</li> <li>email: String</li> <li>username: String</li> <li>password: String</li> <li>role: String</li> <li>stage: Stage</li> <li>scene: Scene</li> <li>root: Parent</li> <li>bazaLink: Connection</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>RegisterController <ul style="list-style-type: none"> <li>GoToLogin(): void</li> <li>Register(): void</li> <li>errorMsg: Label</li> <li>UserName: TextField</li> <li>UserEmail: TextField</li> <li>UserPassword: PasswordField</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>UserViewController <ul style="list-style-type: none"> <li>goToLogin(): void</li> <li>initialize(URL, ResourceBundle): void</li> <li>fillUsers(): void</li> <li>selectUser(): void</li> <li>goToBusLayout(ActionEvent): void</li> <li>removeSelection(): void</li> <li>display(String): void</li> </ul> </li> <li>connection: Connection <ul style="list-style-type: none"> <li>getConnection(): Connection</li> <li>labela: Label</li> <li>welcome: Label</li> <li>RelationStartTxt: TextField</li> <li>RelationEndTxt: TextField</li> <li>RelationPriceTxt: TextField</li> <li>RelationBusCompanyTxt: TextField</li> <li>UserTbl: TableView</li> <li>RelationID: TableColumn</li> <li>RelationStart: TableColumn</li> <li>RelationEnd: TableColumn</li> <li>RelationPrice: TableColumn</li> <li>RelationBusCompany: TableColumn</li> <li>stage: Stage</li> <li>scene: Scene</li> <li>root: Parent</li> <li>addUserBtn: Button</li> <li>selectedUser: Ticket = null</li> <li>bazaLink: Connection</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>ShowUsersController <ul style="list-style-type: none"> <li>goToAdminPanel(): void</li> <li>addUser(): void</li> <li>initialize(URL, ResourceBundle): void</li> <li>fillUsers(): void</li> <li>deleteUser(): void</li> <li>selectUser(): void</li> <li>removeSelection(): void</li> <li>labela: Label</li> <li>UserNameTxt: TextField</li> <li>UserEmailTxt: TextField</li> <li>UserRoleTxt: TextField</li> <li>UserPasswordTxt: PasswordField</li> <li>UserTbl: TableView</li> <li>UserID: TableColumn</li> <li>UserName: TableColumn</li> <li>UserEmail: TableColumn</li> <li>UserRole: TableColumn</li> <li>addUserBtn: Button</li> <li>selectedUser: User = null</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>ShowRelationsController <ul style="list-style-type: none"> <li>goToAdminPanel(): void</li> <li>addUser(): void</li> <li>initialize(URL, ResourceBundle): void</li> <li>fillUsers(): void</li> <li>fillBuses(): void</li> <li>deleteUser(): void</li> <li>selectUser(): void</li> <li>removeSelection(): void</li> </ul> </li> <li>connection: Connection <ul style="list-style-type: none"> <li>getConnection(): Connection</li> <li>labela: Label</li> <li>RelationStartTxt: TextField</li> <li>RelationEndTxt: TextField</li> <li>RelationPriceTxt: TextField</li> <li>RelationBusCompanyTxt: TextField</li> <li>combo: ComboBox&lt;String&gt;</li> <li>UserTbl: TableView</li> <li>UserTbl1: TableView</li> <li>RelationID: TableColumn</li> <li>BusID: TableColumn</li> <li>BusName: TableColumn</li> <li>RelationStart: TableColumn</li> <li>RelationEnd: TableColumn</li> <li>RelationPrice: TableColumn</li> <li>RelationBusCompany: TableColumn</li> <li>addUserBtn: Button</li> <li>bazaLink: Connection</li> <li>selectedUser: Ticket = null</li> <li>lista: ArrayList&lt;String&gt; = new ArrayList&lt;String&gt;()</li> </ul> </li> </ul>		<ul style="list-style-type: none"> <li>FinalTicketController <ul style="list-style-type: none"> <li>goToUserView(): void</li> <li>funkcija(int): void</li> <li>display(String, int, int): void</li> </ul> </li> <li>connection: Connection <ul style="list-style-type: none"> <li>getConnection(): Connection</li> <li>busName: String</li> <li>price: Integer</li> <li>start: String</li> <li>end: String</li> <li>buy: Button</li> <li>labela: Label</li> <li>labela1: Label</li> <li>labela2: Label</li> <li>labela3: Label</li> <li>stage: Stage</li> <li>bazaLink: Connection</li> </ul> </li> </ul>
		<ul style="list-style-type: none"> <li>ShowBusesController <ul style="list-style-type: none"> <li>goToAdminPanel(): void</li> <li>addUser(): void</li> <li>initialize(URL, ResourceBundle): void</li> <li>fillUsers(): void</li> <li>deleteUser(): void</li> <li>selectUser(): void</li> <li>removeSelection(): void</li> <li>labela: Label</li> <li>BusNameTxt: TextField</li> <li>BusCapacityTxt: TextField</li> <li>UserTbl: TableView</li> <li>BusID: TableColumn</li> <li>BusName: TableColumn</li> <li>BusCapacity: TableColumn</li> <li>addUserBtn: Button</li> <li>selectedUser: Bus = null</li> </ul> </li> </ul>

### 2.3.1. Metode koje se ponavljaju u više klasa

---

**2.3.1.1. public void initialize(URL url, ResourceBundle rb)** - metoda za inicijalizaciju i popunjavanje lista

---

**2.3.1.2. protected void goToLogin()** - metoda za prelazak iz trenutnog prozora u Login prozor

**2.3.1.3. protected void goToAllUsers()** - metoda za prelazak iz trenutnog prozora (AdminPanel) u prozor koji pokazuje sve korisnike

**2.3.1.4. protected void goToAllBuses()** - metoda za prelazak iz trenutnog prozora (AdminPanel) u prozor koji pokazuje sve buseve

**2.3.1.5. protected void goToAllRelations()** - metoda za prelazak iz trenutnog prozora (AdminPanel) u prozor koji pokazuje sve tickete tj. relacije.

**2.3.1.6. protected void goToRegister** - metoda za prelazak iz trenutnog prozora u Register prozor

**2.3.1.7. protected void goToAdminPanel** - metoda za prelazak iz trenutnog prozora (Register, ShowRelations, ShowBuses, ShowUsers) u Admin Panel

**2.3.1.8. protected void goToUserView** - metoda za prelazak iz trenutnog prozora u User view početni prozor

**2.3.1.9. public Connection getConnection()** - metoda koja služi za povezivanje s bazom podataka

### 2.3.2. AdminPanelController

---

Klasa koja nakon prijave u sustav korisnicima s ulogom admina prikazuje prozor s par mogućnosti, ispis podataka o registracijama, relacijama i busevima, te mogućnost prebacivanja na prikaz svih korisnika, relacija i buseva.

#### 2.3.2.1. Atributi

---

**Label adminName** – ispisuje username (korisničko ime) admina

**Label numberOfSignUps** – ispisuje ukupan broj svih registriranih korisnika

**Label numberOfRelations** – ispisuje ukupan broj relacija

**Label numberOfBuses** – ispisuje ukupan broj autobusa

**Button button** – pokreće metodu show()

### 2.3.2.2. Metode

---

**public void display(String username)** - ispisuje username prijavljenog admina

**public void show()** - ispisuje broj korisnika, relacija i autobusa

### 2.3.3. RegisterController

---

Klasa se koristi za registraciju novih korisnika u sustav.

#### 2.3.3.1. Atributi

---

**private Label errorMsg** - atribut ispisuje error poruku ukoliko podaci nisu pravilno upisani

**private TextField UserName** - atribut pohranjuje korisničko ime koje je upisano

**private TextField UserEmail** - atribut pohranjuje email koji je upisan

**private PasswordField UserPassword** - atribut pohranjuje lozinku koja je upisana

#### 2.3.3.2. Metode

---

**public void Register()** - ubacuje novog korisnika u bazu podataka, ako je registracija uspješna prebacuje prozor na User view prozor, ako ne ispisuje error

### 2.3.4. LoginController

---

Klasa se koristi za prijavljivanje već postojećih korisnika u sustav.

#### 2.3.4.1. Atributi

---

**public static String email** – email adresa korisnika

**public static String username** – korisničko ime

**public static String password** – lozinka korisnika

**public static String role** -uloga

**private Label errorMsg** – atribut ispisuje error poruku ako podaci nisu pravilno upisani

**private TextField UserEmail** – atribut pohranjuje email adresu korisnika

**private PasswordField UserPassword** – atribut pohranjuje lozinku korisnika

**private Button loginButton** – pokreće metodu Login() iz LoginController-a

#### 2.3.4.2. Metode

---

**protected void Login()** - vrijednosti (email i password) prosljeđuje u metodu validateLogin(), ako je validateLogin() uspješno prošao prebacju prozor na odgovarajući prozor (ako je role prijavljenog korisnika admin – Admin Panel, user – User view)

**public void validateLogin()** - provjerava postoji li korisnik u bazi podataka i podudaraju li se unesene vrijednosti

### 2.3.5. FinalTickerViewController

---

Klasa koja prikazuje detalje karte koju korisnik izabere, te mogućnost kupovine karte.

#### 2.3.5.1. Atributi

---

**public static String busName** – ime autobusa  
**public static Integer price** – cijena relacije  
**public static String start** – početna točka relacije  
**public static String end** - odredište

**Button buy** – ažuriranje sjedišta na zauzeto i pokretanje alert prozora

**Label labela** – ispisuje ime autobusa  
**Label labela1** – ispisuje početna točka relacije  
**Label labela2** - odredište  
**Label labela3** – cijena relacije

#### 2.3.5.2. Metode

---

**public void funkcija(int bus\_fk)** - ispisuje početnu točku, odredište te cijenu i naziv autobus poduzeća (gdje je primarni ključ proslijeđeni bus\_fk)

**public void display(String id, int numberOfSeat, int bus\_fk)** - ažurira sjedište u izabranom autobusu i relaciji kao zauzeto te otvara alert prozor kao prikaz da je kupovina uspješno završena

### 2.3.6. BusLayoutController

---

Klasa se koristi za prikaz sjedišta izabranog autobusa te njihovu popunjenost. Korisnik bira sjedište koje mu odgovara te prelazi na završnu fazu kupovine karte.

#### 2.3.6.1. Atributi

---

**public static Integer id\_BusSeat** – ID odabranog sjedišta  
**public static Integer numberOfSeat** - redni broj odabranog sjedišta  
**public static Integer capacity\_Bus** - kapacitet busa  
**public static Integer free** – vrijednost sjedišta (slobodno ili zauzeto, 0 ili 1)  
**public static String username** – korisničko ime

**Label startPoint** – ispisuje početnu točku odabrane relacije  
**Label endpoint** - odredište  
**Label Price** – cijena odabrane relacije  
**Label busCompany** – naziv poduzeća autobusa

**private HBox container2, container21, container22, container221** – horizontalni spremnici u koje se spremaju sjedišta  
**private List<Button> labels** – lista gumba koji predstavljaju sjedišta

### 2.3.6.1. Metode

---

**public void make(int bus\_fk)** - stvara layout sjedišta autobusa kojem odgovara proslijeđeni primarni ključ bus\_fk.

**public void display(int bus\_fk, String start, String end, int price)** - ispisuje početnu točku, odredište, cijenu i autobus poduzeće izabranog autobusa

### 2.3.7. UserController

---

UserController klasa prikazuje početni prozor koji korisnik vidi kada se uspješno prijavi u aplikaciju. U početnom prozoru ima mogućnost izabrati jednu od ponuđenih relacija nakon čega se prelazi na izbor sjedišta.

#### 2.3.7.1. Atributi

---

**Label labela** – ispisuje error poruku ako nije izabran nijedana relacija

**Label welcome** – ispisuje personaliziranu poruku za user korisnika

**TableView UserTbl** -naziv tablice koja sadrži informacije o relacijama

**TableColumn RelationID** – stupac tablice koji pokazuje ID relacije

**TableColumn RelationStart** – stupac tablice koji pokazuje početnu točku relacije

**TableColumn RelationEnd** – stupac tablice koji pokazuje odredište

**TableColumn RelationPrice** – stupac tablice koji pokazuje cijenu relacije

**TableColumn RelationBusCompany** – stupac tablice koji pokazuje poduzeće autobusa

#### 2.3.7.2. Metode

---

**public void display(String username)** - ispisuje početnu personaliziranu poruku za prijavljenog korisnika (uloga korisnika je user)

**private void fillUsers()** - metoda koja popunjava tablicu tako što stvara listu svih dostupnih relacija tj. ticketa.

**protected void selectUser** - metoda koja sprema vrijednosti odabrane relacije

**protected void removeSelection()** - poništava spremljene vrijednosti iz metode selectUser

**protected void goToBusLayout** - prelazak iz trenutnog prozora u Bus layout prozor (prozor koji prikazuje dostupna sjedišta) uz slanje vrijednosti preko display metode opisane u BusLayoutController-u

### 2.3.8. ShowBusesController

---

Klasa se koristi za prikaz, stvaranje, uređivanje i brisanje autobusa (vrijednosti iz tablice Bus), pristupiti joj mogu korisnici s ulogom admina.

#### 2.3.8.1. Atributi

---

**Label labela** – ispisuje error poruku ako vrijednosti nisu pravilno unesene

**TextField BusNameTxt** – pohranjuje naziv autobusa

**TextField BusCapacityTxt** – pohranjuje kapacitet autobusa

**TableView UserTbl** – naziv tablice koja sadrži informacije o autobusima

**TableColumn BusID** – stupac tablice koji pokazuje ID autobusa

**TableColumn BusName** – stupac tablice koji pokazuje naziv autobusa

**TableColumn BusCapacity** – stupac tablice koji pokazuje kapacitet autobusa

#### 2.3.8.2. Metode

---

**public void addUser()** - dodaje novi autobus u bazu podataka

**public void deleteUser()** - briše odabrani autobus iz baze podataka

**public void selectUser()** - sprema vrijednosti odabranog autobusa u za to namijenjene varijable

**public void removeSelection()** - poništava spremljene vrijednosti iz metode selectUser

**public void fillUsers()** - popunjava tablicu stvaranjem liste svih dostupnih autobusa

### 2.3.9. ShowRelationsController

---

Klasa se koristi za prikaz, stvaranje, uređivanje i brisanje relacija (vrijednosti iz tablice Ticket), pristupiti joj mogu korisnici s ulogom admina.

#### 2.3.9.1. Atributi

---

**Label labela;** - atribut izbacuje error ukoliko sva dana polja nisu ispunjena

**TextField RelationStartTxt;** - atribut pohranjuje polaznu točku koja je upisana

**TextField RelationEndTxt;** - atribut pohranjuje odredište koje je upisano

**TextField RelationPriceTxt;** - atribut pohranjuje cijenu koja je upisana

**TextField RelationBusCompanyTxt;** - atribut pohranjuje autobusnu tvrtku koja je upisana

**TableView UserTbl;** - naziv atributa koji sadrži informacije vezane za tablicu Ticket  
**TableColumn RelationID;** - stupac tablice koji sadrži id relacije  
**TableColumn RelationStart;** - stupac tablice koji sadrži polaznu točku  
**TableColumn RelationEnd;** - stupac tablice koji sadrži odredište  
**TableColumn RelationPrice;** - stupac tablice koji sadrži cijenu relacije  
**TableColumn RelationBusCompany;** - stupac tablice koji sadrži naziv autobusne tvrtke

**Button addUserBtn;** - atribut izvršava addUser() metodu

### 2.3.9.2. Metode

---

**public void addUser()** - dodaje novu relaciju u bazu podataka

**public void deleteUser()** - briše odabranu relaciju iz baze podataka

**public void selectUser()** - sprema vrijednosti odabrane relacije u za to namijenjene varijable

**public void removeSelection()** - poništava spremljene vrijednosti iz metode selectUser

**public void fillUsers()** - popunjava tablicu s relacijama stvaranjem liste svih dostupnih relacija

**public void fillBuses()** - popunjava tablicu s busevima stvaranjem liste svih dostupnih buseva (id, name)

### 2.3.10. ShowUsersController

---

Klasa se koristi za prikaz, stvaranje, uređivanje i brisanje korisnika (vrijednosti iz tablice User), pristupiti joj mogu korisnici s ulogom admina.

#### 2.3.10.1. Atributi

---

**Label labela;** - atribut izbacuje error ukoliko sva dana polja nisu ispunjena

**TextField UserNameTxt;** - atribut pohranjuje korisnicko ime koje je upisano

**TextField UserEmailTxt;** - atribut pohranjuje email koji je upisan

**TextField UserRoleTxt;** - atribut pohranjuje ulogu korisnika koja je upisana

**PasswordField UserPasswordTxt;** - atribut pohranjuje lozinku koja je upisana

**TableView UserTbl;** - naziv tablice koji sadrži podatke vezane za korisnika

**TableColumn UserID;** - stupac tablice koji sadrži ID korisnika

**TableColumn UserName;** - stupac tablice koji sadrži korisnicko ime

**TableColumn UserEmail;** - stupac tablice koji sadrži email korisnika

**TableColumn UserRole;** - stupac tablice koji sadrži ulogu korisnika (admin/user)

**Button addUserBtn;** - atribut izvršava addUser() metodu

### **2.3.10.2. Metode**

---

**public void addUser()** - dodaje novog korisnika u bazu podataka

**public void deleteUser()** - briše odabranog korisnika iz baze podataka

**public void fillUsers()** - popunjava tablicu korisnika spremanjem liste svih korisnika iz baze podataka

**public void selectUser()** - sprema vrijednosti odabranog korisnika u za to namijenjene varijable

**public void removeSelection()** - poništava spremljene vrijednosti iz metode selectUser