



SVEUČILIŠTE U MOSTARU  
FAKULTET STROJARSTVA RAČUNARSTVA I ELEKTROTEHNIKE

## **Programirnje u Javi**

*Projektni zadatak: **Imenik***

**Matej Stojić**  
**Andrej Pejčinović**  
**Filip Oroz**

Akadska godina 2020/2021.

## PROJEKTNI ZADATAK:

### Opis problema:

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

Softver 'Imenik' služi za pohranjivanje svih kontakata korisnika na jedno mjesto.

Pojedini korisnik, osim pohranjivanja kontakata, može označavati kontakte kao favorite i dodavati ih na blok listu. Povrh toga, korisnik ima mogućnost uređivanja pojedinog kontakta. Softver sadrži i mogućnost pohranjivanja vlastitih podataka u sustav.

Administrator aplikacije ima uvid na svim podacima unutar sustava, radi olakšane kontrole nad sustavom.

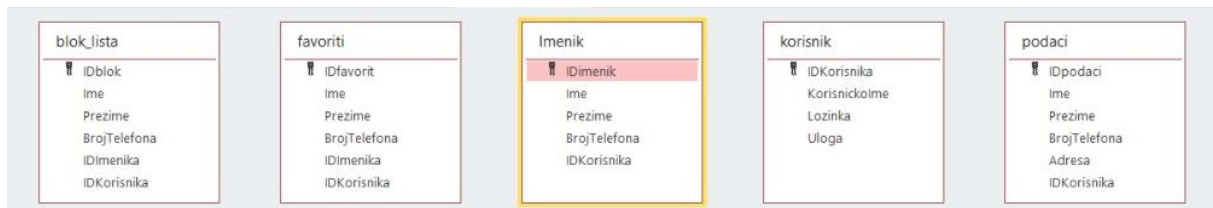
Svrha projekta je olakšano spremanje kontakata u sustav imenika i pojednostavljivanje njihovog uređivanja.

# 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 attribute 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

### Korisnik

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
ID	PK, NOT NULL, Auto Increment	ID korisnika
Korisnickolme	NOT NULL	Korisnicko ime
Lozinka	NOT NULL	Lozinka
Uloga	NOT NULL	Uloga (Administrator/Korisnik)

### Imenik

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
ID	PK, NOT NULL, Auto Increment	ID kontakta
Ime	NOT NULL	Ime kontakta
Prezime	NOT NULL	Prezime kontakta
BrojTelefona	NOT NULL	Broj telefona kontakta
IDKorisnika	FK, NULL	'ID' stupac iz tablice 'korisnik'

### Podaci

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
ID	PK, NOT NULL, Auto Increment	ID podataka
Ime	NOT NULL	Ime korisnika
Prezime	NOT NULL	Prezime korisnika
BrojTelefona	NOT NULL	Broj telefona korisnika

Adresa	NOT NULL	Adresa korisnika
IDKorisnika	FK, NULL	'ID' stupac iz tablice 'korisnik'

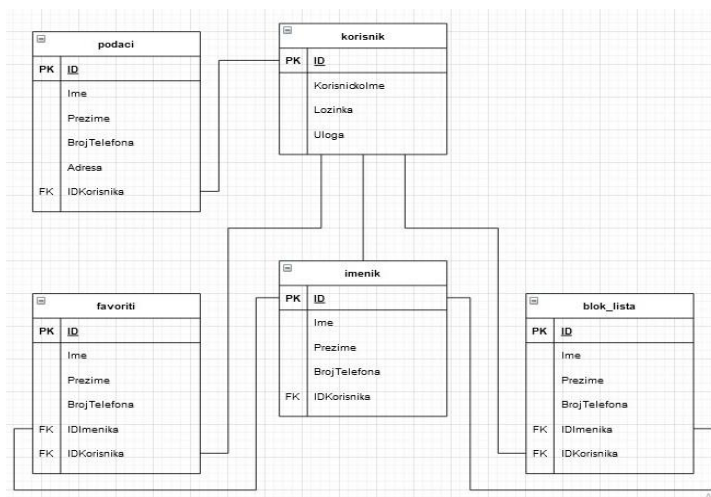
### Blok lista

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
ID	PK, NOT NULL, Auto Increment	ID blokiranog kontakta
Ime	NOT NULL	Ime blokiranog kontakta
Prezime	NOT NULL	Prezime blokiranog kontakta
BrojTelefona	NOT NULL	Broj telefona blokiranog kontakta
IDImenika	FK, NULL	'ID' stupac iz tablice 'imenik'
IDKorisnika	FK, NULL	'ID' stupac iz tablice 'korisnik'

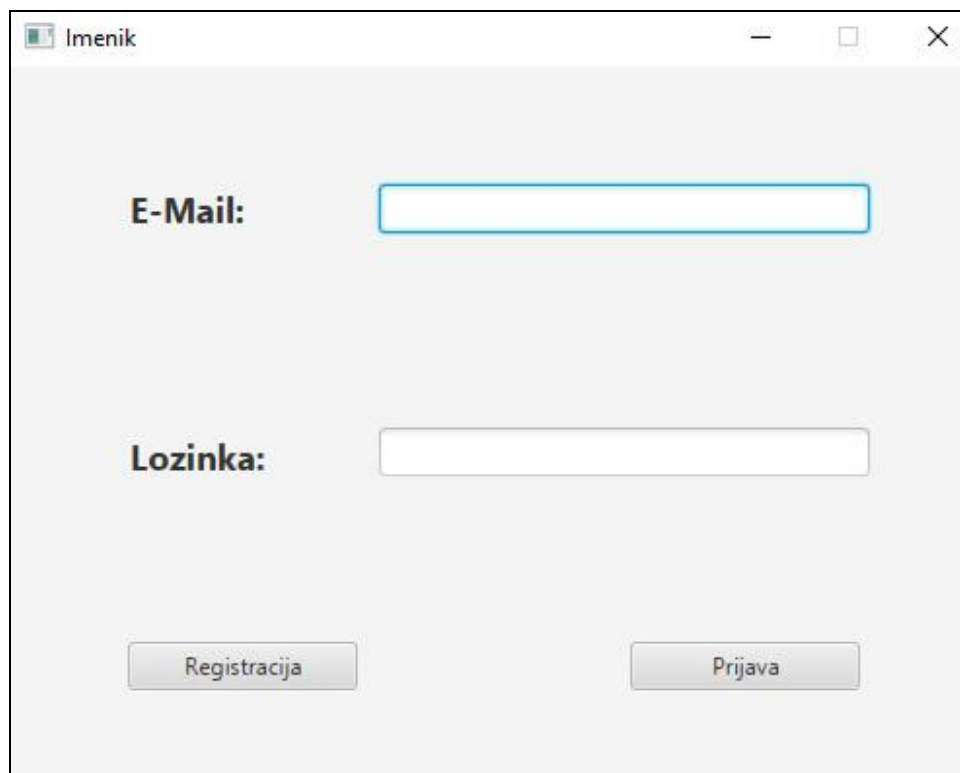
### Favoriti

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
ID	PK, NOT NULL, Auto Increment	ID favorita
Ime	NOT NULL	Ime favorita
Prezime	NOT NULL	Prezime favorita
BrojTelefona	NOT NULL	Broj telefona favorita
IDImenika	FK, NULL	'ID' stupac tablice 'imenik'
IDKorisnika	FK, NULL	'ID' stupac tablice 'korisnik'

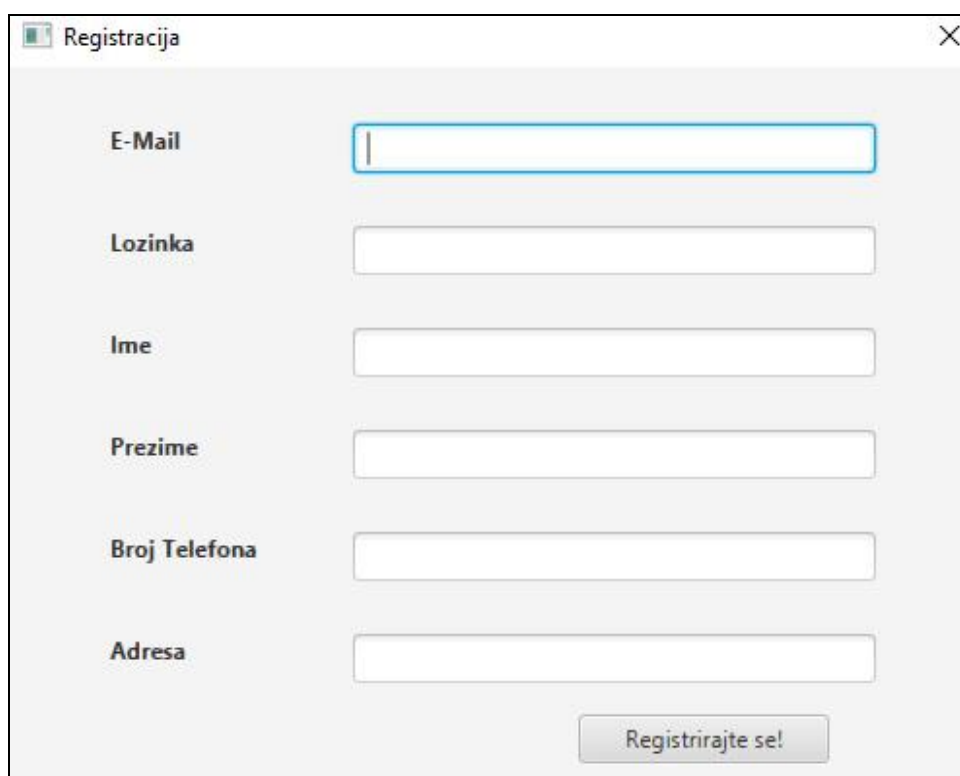
Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> blok_lista	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> favoriti	Browse  Structure  Search  Insert  Empty  Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> imenik	Browse  Structure  Search  Insert  Empty  Drop	6	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> korisnik	Browse  Structure  Search  Insert  Empty  Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> podaci	Browse  Structure  Search  Insert  Empty  Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
5 tables	Sum	13	InnoDB	utf8mb4_general_ci	176.0 KiB	0 B



### 1.3. Aplikacija u Javi



The screenshot shows a Java application window titled "Imenik". It features a light gray background. On the left side, there are two labels: "E-Mail:" and "Lozinka:". To the right of "E-Mail:" is a text input field with a blue border. Below the "E-Mail:" label is the "Lozinka:" label, followed by another text input field. At the bottom of the window, there are two buttons: "Registracija" on the left and "Prijava" on the right. The window has standard Java Swing window controls (minimize, maximize, close) in the top right corner.



The screenshot shows a Java application window titled "Registracija". It features a light gray background. On the left side, there are six labels: "E-Mail", "Lozinka", "Ime", "Prezime", "Broj Telefona", and "Adresa". To the right of each label is a corresponding text input field. The "E-Mail" field has a blue border. At the bottom right of the window, there is a button labeled "Registrirajte se!". The window has standard Java Swing window controls (close button) in the top right corner.

The screenshot shows the 'Imenik' application interface. On the left is a sidebar with the following elements:

- Input fields for 'Ime' (Ivan), 'Prezime' (Ivanić), 'Adresa' (Adresa BB), and 'Broj Tel.' (063/222-222).
- A button 'Uredite svoje podatke' (highlighted with a blue border).
- Buttons: 'Dodajte kontakt', 'Blok Lista', 'Favoriti', and 'Odjava'.

The main area contains a table with columns 'Ime', 'Prezime', and 'Broj Telefona'. The first row contains the data: 'Luka', 'Lukić', and '123456'. To the right of the table are buttons 'Uredi', 'Obriši', 'Blokiraj', and a star icon.

This screenshot shows the 'Imenik' application with the 'Uređivanje podataka' dialog box open. The dialog box contains the following fields and buttons:

- An empty text input field at the top.
- A 'Prezime' label and an empty text input field.
- An 'Adresa' label and an empty text input field.
- A 'Broj Telefona' label and an empty text input field.
- A 'Uredite podatke' button at the bottom.

The background application interface is partially visible behind the dialog box.

This screenshot shows the 'Imenik' application with the 'Dodavanje kontakta' dialog box open. The dialog box contains the following fields and buttons:

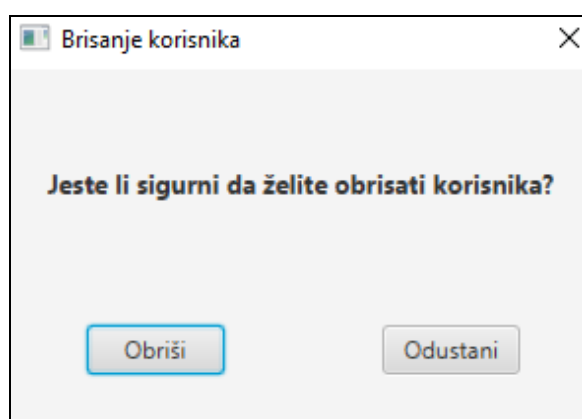
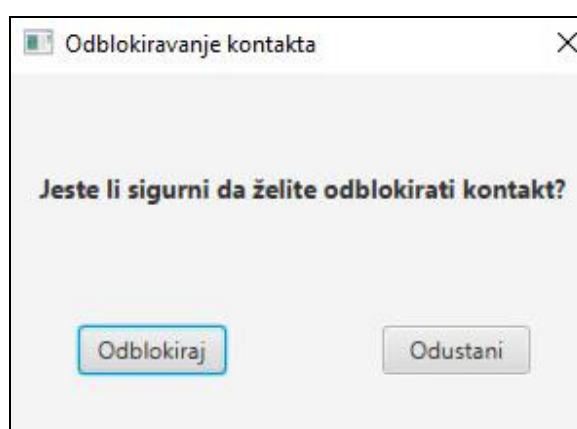
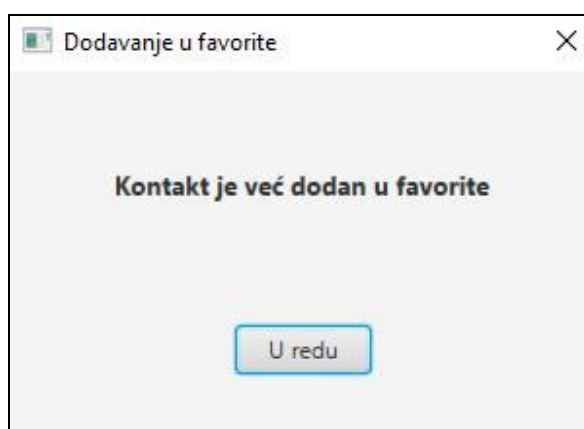
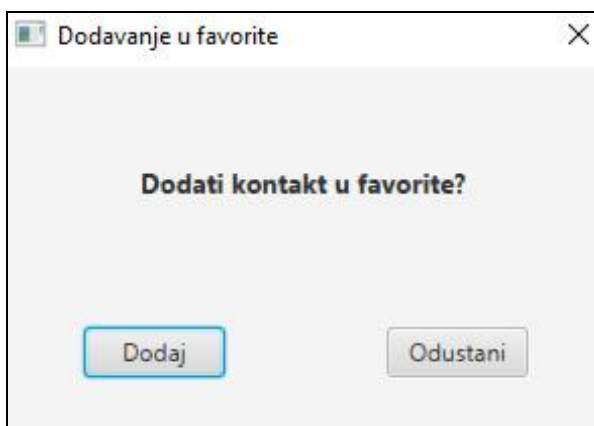
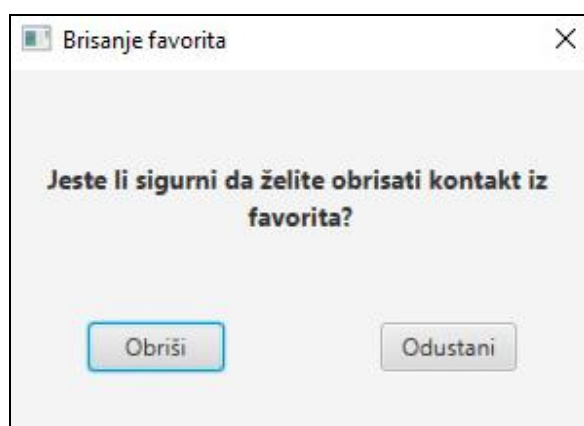
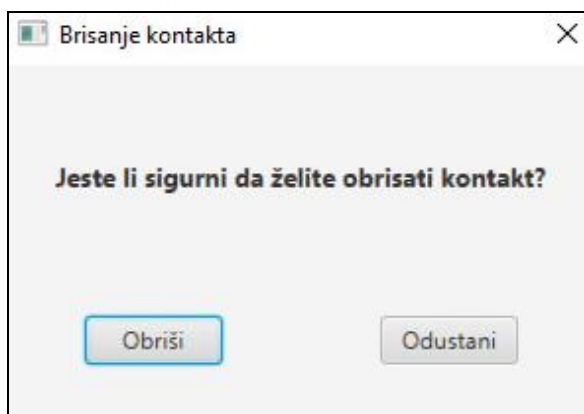
- An empty text input field at the top.
- A 'Prezime' label and an empty text input field.
- A 'Broj Telefona' label and an empty text input field.
- A 'Dodajte kontakt' button at the bottom.

The background application interface is partially visible behind the dialog box.

[illegible][illegible]

The screenshot displays a contact management application. On the left is a sidebar with navigation links: 'Imenik', 'Prezime', 'Adresa', 'Broj Tel.', 'Uredite svoje podatke', 'Dodajte kontakt', 'Blok Lista', 'Favoriti', and 'Odjava'. The main area features a table with columns 'Ime', 'Prezime', and 'Broj Telefona'. The first row contains the data 'Luka', 'Lukić', and '123456'. To the right of the table are buttons for 'Uredi', 'Obriši', 'Blokiraj', and a star icon. A modal dialog titled 'Uređivanje kontakta' is open in the center, containing three input fields (the first is empty, the second has 'Prezime', the third has 'Broj Telefona') and a 'Uredite kontakt' button at the bottom.





Lista korisnika

Dodajte korisnika

Imenik

Blok Lista

Favoriti

Korisnički podaci

Odjava

ID	Korisničko Ime	Lozinka	Uloga	
1	mstojic@sum.ba	123456	Administrator	<div>Uredi korisnika</div> <div>Obriši korisnika</div>
2	foroz@sum.ba	123456	Administrator	<div>Uredi korisnika</div> <div>Obriši korisnika</div>
3	apejičinović@sum.ba	123456	Administrator	<div>Uredi korisnika</div> <div>Obriši korisnika</div>
15	ivan	123456	Korisnik	<div>Uredi korisnika</div> <div>Obriši korisnika</div>

Lista korisnika

Dodajte korisnika

Imenik

Blok Lista

Favoriti

Korisnički podaci

Odjava

Dodavanje korisnika

Lozinka

Uloga

Dodajte korisnika

ID	Uloga	
1	Administrator	<div>Uredi korisnika</div> <div>Obriši korisnika</div>
2	Administrator	<div>Uredi korisnika</div> <div>Obriši korisnika</div>
3	Administrator	<div>Uredi korisnika</div> <div>Obriši korisnika</div>
15	Korisnik	<div>Uredi korisnika</div> <div>Obriši korisnika</div>

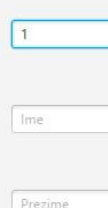
Imenik

ID Korisnika	ID	Ime	Prezime	Broj Telefona	
1	4	Ivan	Ivanić	063/222-333	<div>Uredi kontakt</div> <div>Obriši kontakt</div>
1	17	Luka	Lukić	123456	<div>Uredi kontakt</div> <div>Obriši kontakt</div>
1	19	Marko	Markić	063 226 077	<div>Uredi kontakt</div> <div>Obriši kontakt</div>
2	5	Marko	Markić	063 555 555	<div>Uredi kontakt</div> <div>Obriši kontakt</div>
2	23	Mate	Matić	12356	<div>Uredi kontakt</div> <div>Obriši kontakt</div>
15	41	Luka	Lukić	123456	<div>Uredi kontakt</div> <div>Obriši kontakt</div>

Blok Lista						✕
ID Korisnika	ID	Ime	Prezime	Broj Telefona		
15	11	Mate	Matić	123456	Odblokiraj kontakt	

Favoriti						✕
ID Korisnika	ID Favorita	Ime	Prezime	Broj Telefona	Edit	
15	18	Luka	Lukić	123456	Obriši iz favorita	

Korisnički podaci							✕
ID Korisnika	ID	Ime	Prezime	Broj Telefona	Adresa		
1	5	Matej	Stojić	123456	NekaAdresa	Uredi podatke	
2	10	Filip	Oroz	063 555 555	NekaAdresa	Uredi podatke	
3	7	Andrej	Pejićinović	123 456 789	NekaAdresa	Uredi podatke	
15	13	Ivan	Ivanić	063/222-222	Adresa BB	Uredi podatke	



Uređivanje podataka

1


Ime

Prezime



Adresa

Broj Telefona

Uredite podatke

 Uređivanje kontakta ✕

1



Ime

Prezime

Broj Telefona

Uredite kontakt

The screenshot shows a web application interface for managing users. A modal dialog titled "Uređivanje korisnika" (Editing User) is open, allowing the user to edit the details of a selected user. The dialog contains the following elements:

- A text input field for the user's name (Ime).
- A password input field for the user's password (Lozinka).
- A dropdown menu for selecting the user's role (Uloga).
- A button labeled "Uredite korisnika" (Save User) to confirm the changes.

The background interface includes a sidebar with navigation buttons: "Dodajte korisnika" (Add User), "Imenik" (Directory), "Blok Lista" (Block List), "Favoriti" (Favorites), "Korisnički podaci" (User Data), and "Odjava" (Logout). The main area displays a table of users with columns "ID" and "Uloga".

## 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

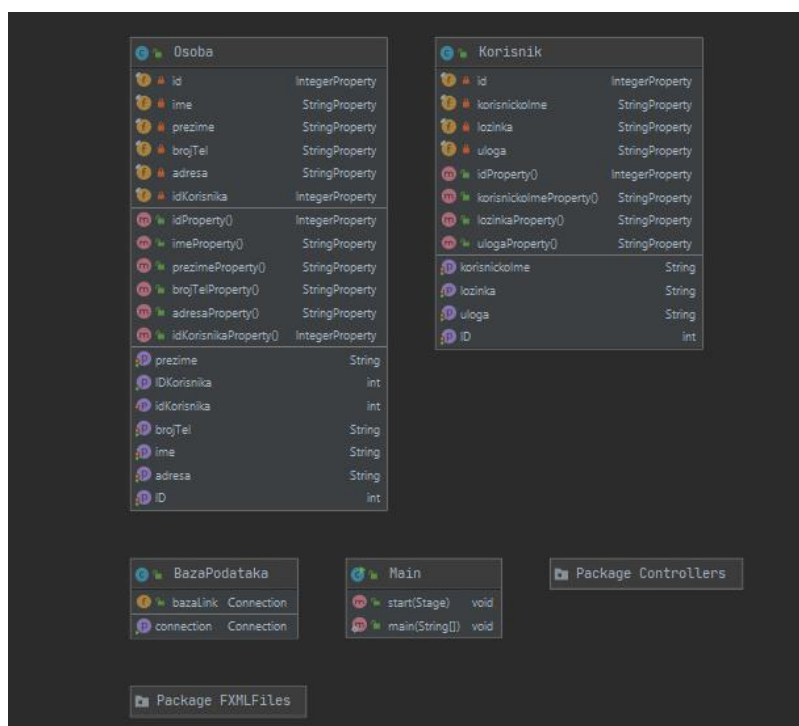
Projekt 'Imenik' sastoji se od jednog paketa(sample) i dva podpaketa(Controllers i FXMLFiles).

**sample** paket sastoji se od dva paketa, (Controllers i FXMLFiles), i četiri klase. Paket sample funkcionira kao početna točka projekta.

**Controllers** paket se sastoji od 27 klasa koje služe za upravljanje događajima u aplikaciji. Ovaj paket predstavlja 'backend' dio projekta.

**FXMLFiles** paket u sebi sadrži 27 FXML datoteka na osnovu kojih je izgrađen izgled korisničkog sučelja. Ovaj paket predstavlja 'frontend' dio projekta. U sebi ne sadrži klase.

### 2.2. Klase - sample



### 2.2.1. Klasa BazaPodataka

---

Klasa se koristi za ostvarivanje veze s bazom podataka kako joj projekt mogao pristupiti.

#### 2.2.1.1. Atributi

---

**public Connection bazaLink** – atribut koji sadrži vezu s bazom podataka

#### 2.2.1.2. Metode

---

**public Connection getConnection()** – metoda koja prosljeđuje podatke o bazi podataka i vrši povezivanje s istom

### 2.2.2 Klasa Korisnik

---

Klasa se koristi za pohranjivanje podataka o korisniku iz SQL baze.

#### 2.2.2.1. Atributi

---

**private final IntegerProperty id** – ID korisnika

**private final StringProperty korisnickolme** – korisničko ime korisnika

**private final StringProperty lozinka** – lozinka korisnika

**private final StringProperty uloga** – uloga korisnika

#### 2.2.2.2. Metode

---

**public int getID()** – metoda koja vraća ID korisnika

**public void setID(int value)** – metoda koja postavlja vrijednost ID korisnika

**public IntegerProperty idProperty()** – metoda koja vraća svojstvo 'id'

**public String getKorisnickolme()** – metoda koja vraća korisničko ime korisnika

**public void setKorisnickolme(String value)** – metoda koja postavlja vrijednost korisničkog imena korisnika

**public StringProperty korisnickolmeProperty()** – metoda koja vraća svojstvo 'korisnickolme'

**public String getLozinka()** – metoda koja vraća lozinku korisnika

**public void setLozinka(String value)** – metoda koja postavlja vrijednost lozinke korisnika

**public StringProperty lozinkaProperty()** – metoda koja vraća svojstvo 'lozinka'

**public String getUloga()** – metoda koja vraća ulogu korisnika

**public void setUloga(String value)** – metoda koja postavlja vrijednost uloge korisnika

**public StringProperty ulogaProperty()** – metoda koja vraća svojstvo 'uloga'

### 2.2.3. Klasa Osoba

---

Klasa se koristi za pohranjivanje podataka o kontaktu i korisniku iz SQL baze.

## 2.2.3.1. Atributi

private final IntegerProperty id – ID kontakta  
 private final StringProperty ime – ime kontakta  
 private final StringProperty prezime – prezime kontakta  
 private final StringProperty brojTel – broj telefona kontakta  
 private final StringProperty adresa – adresa kontakta  
 private final IntegerProperty idKorisnika – ID korisnika kome kontakt pripada

## 2.2.3.2. Metode

public int getID() – metoda koja vraća ID kontakta/korisnika  
 public void setID(int value) – metoda koja postavlja vrijednost ID kontakta/korisnika  
 public IntegerProperty idProperty() – metoda koja vraća svojstvo 'id'

public String getIme() – metoda koja vraća ime kontakta/korisnika  
 public void setIme(String value) – metoda koja postavlja vrijednost imena kontakta/korisnika  
 public StringProperty imeProperty() – metoda koja vraća svojstvo 'id'

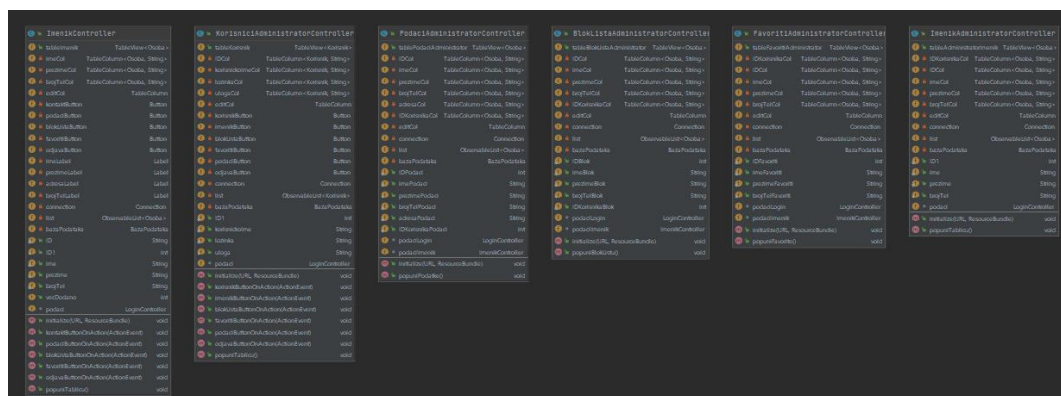
public String getPrezime() – metoda koja vraća prezime kontakta/korisnika  
 public void setPrezime(String value) – metoda koja postavlja vrijednost prezimena kontakta/korisnika  
 public StringProperty prezimeProperty() – metoda koja vraća svojstvo 'prezime'

public String getBrojTel() - metoda koja vraća broj telefona kontakta/korisnika  
 public void setBrojTel(String value) – metoda koja postavlja vrijednost broja telefona kontakta/korisnika  
 public StringProperty brojTelProperty() – metoda koja vraća svojstvo 'brojTel'

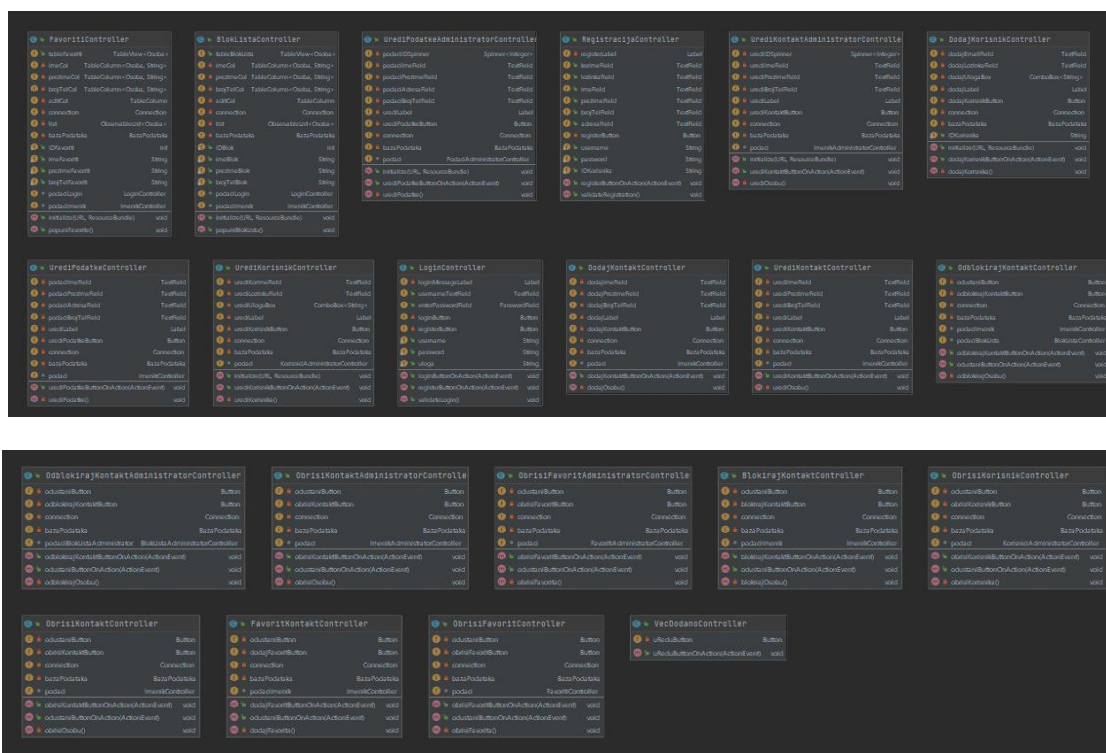
public String getAdresa() – metoda koja vraća adresu korisnika  
 public void setAdresa(String value) – metoda koja postavlja vrijednost adrese korisnika  
 public StringProperty adresaProperty() – metoda koja vraća svojstvo 'adresa'

public int getIDKorisnika() – metoda koja vraća ID korisnika  
 public void setIDKorisnika(int value) – metoda koja postavlja vrijednost ID korisnika  
 public IntegerProperty idKorisnikaProperty() – metoda koja vraća svojstvo 'idKorisnika'

## 2.3. Klase - Controllers







## 2.3.1. Klasa BlokIrjKontaktController

Klasa se koristi za blokiranje kontakta iz imenika.

### 2.3.1.1. Atributi

**private Button odustaniButton** – gumb za odustajanje  
**private Button blokIrjKontaktButton** – gumb za blokiranje kontakta  
**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private sample.BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka  
**ImenikController podacilmenik** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

### 2.3.1.2. Metode

**public void blokIrjKontaktButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba blokIrjKontaktButton

**public void odustaniButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odustaniButton

**private void blokIrjOsobu()** – metoda za blokiranje osobe, osobu koju blokiramo prebacujemo iz imenika u blok listu

## 2.3.2. Klasa BlokListaAdministratorController



Klasa se koristi za popunjavanje blok liste sa strane administratora (administrator može vidjeti blok liste svih korisnika).

### 2.3.2.1. Atributi

---

**public TableView<Osoba> tableBlokListaAdministrator** – tablica blokiranih kontakata  
**private TableColumn<Osoba, String> IDCol** – 'ID' stupac tablice  
**private TableColumn<Osoba, String> imeCol** – 'Ime' stupac tablice  
**private TableColumn<Osoba, String> prezimeCol** – 'Prezime' stupac tablice  
**private TableColumn<Osoba, String> brojTelCol** – 'Broj Telefona' stupac tablice  
**private TableColumn<Osoba, String> IDKorisnikaCol** – 'ID Korisnika' stupac tablice  
**private TableColumn editCol** – stupac za gumb upravljanja kontaktom(odblokiranje)

**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private ObservableList<Osoba> list** – lista u kojoj se spremaju podaci o blokiranom kontaktu

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**public static int IDBlok** – ID odabranog blokiranog kontakta  
**public static String imeBlok** – ime odabranog blokiranog kontakta  
**public static String prezimeBlok** – prezime odabranog blokiranog kontakta  
**public static String brojTelBlok** – broj telefona odabranog blokiranog kontakta  
**public static int IDKorisnikaBlok** – ID korisnika kome pripada odabrani blokirani kontakt

**LoginController podaciLogin** – stvaranje objekta klase LoginController za dohvaćanje korisničkih(login) podataka

**ImenikController podaciImenik** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

### 2.3.2.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za inicijalizaciju i popunjavanje blok liste

**public void popuniBlokListu()** – metoda za popunjavanje tablice blok liste i dodavanje gumba za odblokiranje kontakata (prikazuju se blokirani kontakti svih korisnika u sustavu)

## 2.3.3. Klasa BlokListaController

---

Klasa se koristi za popunjavanje blok liste sa strane korisnika (korisnik može vidjeti samo vlastite blokirane kontakte).

### 2.3.3.1. Atributi

---

**public TableView<Osoba> tableBlokLista** – tablica blokiranih kontakata  
**private TableColumn<Osoba, String> imeCol** – 'Ime' stupac tablice  
**private TableColumn<Osoba, String> prezimeCol** – 'Prezime' stupac tablice  
**private TableColumn<Osoba, String> brojTelCol** – 'Broj Telefona' stupac tablice  
**private TableColumn editCol** – stupac za gumb upravljanja kontaktom(odblokiranje)

**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private ObservableList<Osoba> list** – lista u kojoj se spremaju podaci o blokiranom kontaktu

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**public static int IDBlok** – ID odabranog blokiranog kontakta

**public static String imeBlok** – ime odabranog blokiranog kontakta

**public static String prezimeBlok** – prezime odabranog blokiranog kontakta

**public static String brojTelBlok** – broj telefona odabranog blokiranog kontakta

**LoginController podaciLogin** – stvaranje objekta klase LoginController za dohvaćanje korisničkih(login) podataka

**ImenikController podaciImenik** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

### 2.3.3.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za inicijalizaciju i popunjavanje blok liste

**public void popuniBlokListu()** – metoda za popunjavanje tablice blok liste i dodavanje gumba za odblokiranje kontakata (prikazuju se samo kontakti koji pripadaju korisniku)

## 2.3.4. Klasa DodajKontaktController

---

Klasa se koristi za dodavanje kontakta u imenik.

### 2.3.4.1. Atributi

---

**private TextField dodajImeField** – polje za unos imena

**private TextField dodajPrezimeField** – polje za unos prezimena

**private TextField dodajBrojTelField** – polje za unos broja telefona

**private Label dodajLabel** – dinamična oznaka koja se koristi za razna upozorenja ukoliko korisnik nije zadovoljio neke od predviđenih uvjeta

**private Button dodajKontaktButton** – gumb za dodavanje kontakta u imenik

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**ImenikController podaci** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

### 2.3.4.2. Metode

---

**public void dodajKontaktButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba dodajKontaktButton

**private void dodajOsobu()** – metoda za dodavanje kontakta u imenik

## 2.3.5. Klasa DodajKorisnikController

---

Klasa se koristi za dodavanje korisnika u sustav.

### 2.3.5.1. Atributi

---

**private TextField dodajEmailField** – polje za unos korisničkog imena  
**private TextField dodajLozinkaField** – polje za unos lozinke  
**private ComboBox<String> dodajUlogaBox** – polje za unos uloge (korisnik/administrator)  
**private Label dodajLabel** – dinamična oznaka koja se koristi za razna upozorenja ukoliko administrator nije zadovoljio neke od predviđenih uvjeta  
**private Button dodajKorisnikButton** – gumb za dodavanje korisnika u imenik

**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**public static String IDKorisnika** – ID kreiranog korisnika

### 2.3.5.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za popunjavanje opcija ComboBox-a dodajUlogaBox

**public void dodajKorisnikButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba dodajKorisnikButton

**private void dodajKorisnika()** – metoda za dodavanje korisnika i njegovih podataka u sustav, korisnik se dodaje samo ako već ne postoji

## 2.3.6. Klasa FavoritiAdministratorController

---

Klasa se koristi za popunjavanje liste favorita sa strane administratora (administrator može vidjeti liste favorita svih korisnika).

### 2.3.6.1. Atributi

---

**public TableView<Osoba> tableFavoritiAdministrator** – tablica favorita  
**private TableColumn<Osoba, String> IDKorisnikaCol** – 'ID Korisnika' stupac tablice  
**private TableColumn<Osoba, String> IDCol** – 'ID' stupac tablice  
**private TableColumn<Osoba, String> imeCol** – 'Ime' stupac tablice  
**private TableColumn<Osoba, String> prezimeCol** – 'Prezime' stupac tablice  
**private TableColumn<Osoba, String> brojTelCol** – 'Broj Telefona' stupac tablice  
**private TableColumn editCol** – stupac za gumb upravljanja kontaktom (brisanje iz favorita)

**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private ObservableList<Osoba> list** – lista u kojoj se spremaju podaci o favoritima  
**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**public static int IDFavoriti** – ID odabranog favorita  
**public static String imeFavoriti** – ime odabranog favorita  
**public static String prezimeFavoriti** – prezime odabranog favorita  
**public static String brojTelFavoriti** – broj telefona odabranog favorita

**LoginController podaciLogin** – stvaranje objekta klase LoginController za dohvaćanje korisničkih(login) podataka

**ImenikController podaciImenik** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

### 2.3.6.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za inicijalizaciju i popunjavanje liste favorita

**public void popuniFavorite()** – metoda za popunjavanje tablice favorita i dodavanje gumba za brisanje kontakata iz favorita(prikazuju se favoriti svih korisnika u sustavu)

### 2.3.7. Klasa FavoritiController

---

Klasa se koristi za popunjavanje liste favorita sa strane korisnika (korisnik može vidjeti samo vlastitu listu favorita).

#### 2.3.7.1. Atributi

---

**public TableView<Osoba> tableFavoritiAdministrator** – tablica favorita

**private TableColumn<Osoba, String> imeCol** – 'Ime' stupac tablice

**private TableColumn<Osoba, String> prezimeCol** – 'Prezime' stupac tablice

**private TableColumn<Osoba, String> brojTelCol** – 'Broj Telefona' stupac tablice

**private TableColumn editCol** – stupac za gumb upravljanja kontaktom(brisanje iz favorita)

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private ObservableList<Osoba> list** – lista u kojoj se spremaju podaci o favoritima

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**public static int IDFavoriti** – ID odabranog favorita

**public static String imeFavoriti** – ime odabranog favorita

**public static String prezimeFavoriti** – prezime odabranog favorita

**public static String brojTelFavoriti** – broj telefona odabranog favorita

**LoginController podaciLogin** – stvaranje objekta klase LoginController za dohvaćanje korisničkih(login) podataka

**ImenikController podaciImenik** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

#### 2.3.7.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za inicijalizaciju i popunjavanje liste favorita

**public void popuniFavorite()** – metoda za popunjavanje tablice favorita i dodavanje gumba za brisanje kontakata iz favorita(prikazuju se samo favoriti koji pripadaju korisniku)

### 2.3.8. Klasa FavoritKontaktController

---

Klasa se koristi za dodavanje kontakta u listu favorita.

#### 2.3.8.1. Atributi

---

**private Button odustaniButton** – gumb za odustajanje  
**private Button dodajFavoritButton** – gumb za dodavanje favorita  
**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private sample.BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka  
**ImenikController podaciImenik** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

#### 2.3.8.2. Metode

---

**public void dodajFavoritButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba dodajFavoritButton

**public void odustaniButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odustaniButton

**private void dodajFavorita()** – metoda za dodavanje osobe u favorit listu

### 2.3.9. Klasa ImenikAdministratorController

---

Klasa se koristi za popunjavanje imenika sa strane administratora (administrator može vidjeti kontakte svih korisnika).

#### 2.3.9.1. Atributi

---

**public TableView<Osoba> tableAdministratorImenik** – tablica kontakata  
**private TableColumn<Osoba, String> IDKorisnikaCol** – 'ID Korisnika' stupac tablice  
**private TableColumn<Osoba, String> IDCol** – 'ID' stupac tablice  
**private TableColumn<Osoba, String> imeCol** – 'Ime' stupac tablice  
**private TableColumn<Osoba, String> prezimeCol** – 'Prezime' stupac tablice  
**private TableColumn<Osoba, String> brojTelCol** – 'Broj Telefona' stupac tablice  
**private TableColumn editCol** – stupac za gumbe upravljanja kontaktom (brisanje, uređivanje)

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private ObservableList<Osoba> list** – lista u kojoj se spremaju podaci o kontaktu

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**public static int ID1** – ID odabranog kontakta iz tablice

**public static String ime** – ime odabranog kontakta

**public static String prezime** – prezime odabranog kontakta

**public static String brojTel** – broj telefona odabranog kontakta

**LoginController podaciLogin** – stvaranje objekta klase LoginController za dohvaćanje korisničkih (login) podataka

### 2.3.9.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za inicijalizaciju i popunjavanje imenika

**public void popuniTablicu()** – metoda za popunjavanje tablice imenika i dodavanje gumba za brisanje i uređivanje kontakata (prikazuju se kontakti svih korisnika u sustavu)

### 2.3.10. Klasa ImenikController

---

Klasa se koristi za popunjavanje imenika sa strane korisnika (korisnik može vidjeti samo vlastite kontakte).

#### 2.3.10.1. Atributi

---

**public TableView<Osoba> tableImenik** – tablica kontakata

**private TableColumn<Osoba, String> imeCol** – 'Ime' stupac tablice

**private TableColumn<Osoba, String> prezimeCol** – 'Prezime' stupac tablice

**private TableColumn<Osoba, String> brojTelCol** – 'Broj Telefona' stupac tablice

**private TableColumn editCol** – stupac za gumb upravljanja kontaktom (brisanje, uređivanje, dodavanje u favorite)

**private Button kontaktButton** – gumb za dodavanje kontakta

**private Button podaciButton** – gumb za uređivanje podataka

**private Button blokListaButton** – gumb za otvaranje blok liste

**private Button favoritiButton** – gumb za otvaranje liste favorita

**private Button odjavaButton** – gumb za odjavu iz sustava

**private Label imeLabel** – dinamična oznaka koja prikazuje ime korisnika

**private Label prezimeLabel** – dinamična oznaka koja prikazuje prezime korisnika

**private Label adresaLabel** – dinamična oznaka koja prikazuje adresu korisnika

**private Label brojTelLabel** – dinamična oznaka koja prikazuje broj telefona korisnika

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private ObservableList<Osoba> list** – lista u kojoj se spremaju podaci o kontaktu

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**public static String ID** – SQL upit koji daje ID ćeliju tablice 'korisnik'

**public static int ID1** – ID odabranog kontakta iz tablice

**public static String ime** – ime odabranog kontakta

**public static String prezime** – prezime odabranog kontakta

**public static String brojTel** – broj telefona odabranog kontakta

**LoginController podaciLogin** – stvaranje objekta klase LoginController za dohvaćanje korisničkih (login) podataka

#### 2.3.10.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za inicijalizaciju i popunjavanje imenika

**public void kontaktButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba kontaktButton

**public void podaciButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba podaciButton

**public void blokListaButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba blokListaButton

**public void favoritiButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba favoritiButton

**public void odjavaButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odjavaButton

**public void popuniTablicu()** – metoda za popunjavanje tablice imenika, podataka o korisniku i dodavanje gumba za brisanje, blokiranje i uređivanje kontakata, te za dodavanje kontakata u listu favorita (prikazuju se samo kontakti koji pripadaju korisniku)

### 2.3.11. Klasa KorisniciAdministratorController

---

Klasa se koristi za popunjavanje liste korisnika.

#### 2.3.11.1. Atributi

---

**public TableView<Korisnik> tableKorisnik** – tablica kontakata  
**private TableColumn<Korisnik, String> IDCol** – 'ID' stupac tablice  
**private TableColumn<Korisnik, String> korisnickolmeCol** – 'Korisničko ime' stupac tablice

**private TableColumn<Korisnik, String> lozinkaCol** – 'Lozinka' stupac tablice  
**private TableColumn<Korisnik, String> ulogaCol** – 'Uloga' stupac tablice  
**private TableColumn editCol** – stupac za gumbe upravljanja korisnikom (brisanje, uređivanje)

**private Button korisnikButton** – gumb za dodavanje korisnika  
**private Button imenikButton** – gumb za otvaranje imenika  
**private Button blokListaButton** – gumb za otvaranje blok liste  
**private Button favoritiButton** – gumb za otvaranje liste favorita  
**private Button podaciButton** – gumb za otvaranje liste podataka korisnika  
**private Button odjavaButton** – gumb za odjavu iz sustava

**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private ObservableList<Korisnik> list** – lista u kojoj se spremaju podaci o korisniku  
**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**public static int ID1** – ID odabranog korisnika iz tablice  
**public static String korisnickolme** – korisničko ime odabranog korisnika  
**public static String lozinka** – lozinka odabranog korisnika  
**public static String uloga** – uloga odabranog korisnika

**LoginController podaciLogin** – stvaranje objekta klase LoginController za dohvaćanje korisničkih(login) podataka

### 2.3.11.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za inicijalizaciju i popunjavanje imenika

**public void korisnikButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba korisnikButton

**public void imenikButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba imenikButton

**public void blokListaButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba blokListaButton

**public void favoritiButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba favoritiButton

**public void podaciButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba podaciButton

**public void odjavaButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odjavaButton

**public void popuniTablicu()** – metoda za popunjavanje tablice korisnika i dodavanje gumba za brisanje i uređivanje korisnika

### 2.3.12. Klasa LoginController

---

Klasa se koristi za prijavljivanje(logiranje) korisnika u sustav.

#### 2.3.12.1. Atributi

---

**private Label loginMessageLabel** – dinamična oznaka za prikaz raznih upozorenja ukoliko korisnik nije zadovoljio određene uvjete

**public TextField usernameTextField** – polje za unos korisničkog imena

**public PasswordField enterPasswordField** – polje za unos lozinke

**private Button loginButton** – gumb za prijavu u sustav

**private Button registerButton** – gumb za otvaranje prozora za registraciju

**public static String username** – pohranjeno unešeno korisničko ime

**public static String password** – pohranjena unešena lozinka

**public static String uloga** – pohranjena uloga korisnika

#### 2.3.12.2. Metode

---

**public void loginButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba loginButton



**public void registerButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba registerButton

**public void validateLogin()** – metoda za provjeru točnosti login forme

### 2.3.13. Klasa ObrisiFavoritAdministratorController

---

Klasa se koristi za brisanje favorita iz liste favorita sa strane administratora.

#### 2.3.13.1. Atributi

---

**private Button odustaniButton** – gumb za odustajanje

**private Button obrisiFavoritButton** – gumb za brisanje favorita

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**FavoritiAdministratorController podaciFavoriti** – stvaranje objekta klase

FavoritiAdministratorController za dohvaćanje podataka iz liste favorita

#### 2.3.13.2. Metode

---

**public void obrisiFavoritButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba obrisiFavoritButton

**public void odustaniButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odustaniButton

**private void obrisiFavorita()** – metoda za brisanje favorita iz liste favorita

### 2.3.14. Klasa ObrisiFavoritController

---

Klasa se koristi za brisanje favorita iz liste favorita sa strane korisnika.

#### 2.3.14.1. Atributi

---

**private Button odustaniButton** – gumb za odustajanje

**private Button obrisiFavoritButton** – gumb za brisanje favorita

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**FavoritiController podaciFavoriti** – stvaranje objekta klase FavoritiController za dohvaćanje podataka iz liste favorita

#### 2.3.14.2. Metode

---

**public void obrisiFavoritButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba obrisiFavoritButton

**public void odustaniButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odustaniButton

**private void obrisiFavorita()** – metoda za brisanje favorita iz liste favorita

### **2.3.15. Klasa ObrisiKontaktAdministratorController**

---

Klasa se koristi za brisanje kontakta iz imenika sa strane administratora.

#### **2.3.15.1. Atributi**

---

**private Button odustaniButton** – gumb za odustajanje

**private Button obrisiKontaktButton** – gumb za brisanje kontakta iz imenika

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**ImenikAdministratorController podaci** – stvaranje objekta klase

ImenikAdministratorController za dohvaćanje podataka iz imenika

#### **2.3.15.2. Metode**

---

**public void obrisiKontaktButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba obrisiKontaktButton

**public void odustaniButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odustaniButton

**private void obrisiOsobu()** – metoda za brisanje kontakta iz imenika

### **2.3.16. Klasa ObrisiKontaktController**

---

Klasa se koristi za brisanje kontakta iz imenika sa strane korisnika.

#### **2.3.16.1. Atributi**

---

**private Button odustaniButton** – gumb za odustajanje

**private Button obrisiKontaktButton** – gumb za brisanje kontakta iz imenika

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**ImenikAdministratorController podaci** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

#### **2.3.16.2. Metode**

---

**public void obrisiKontaktButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba obrisiKontaktButton

**public void odustaniButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odustaniButton

**private void obrisiOsobu()** – metoda za brisanje kontakta iz imenika

### **2.3.17. Klasa ObrisiKorisnikController**

---

Klasa se koristi za brisanje korisnika iz sustava.

### **2.3.17.1. Atributi**

---

**private Button odustaniButton** – gumb za odustajanje  
**private Button obrisiKorisnikButton** – gumb za brisanje korisnika iz sustava  
**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka  
**KorisniciAdministratorController podaci** – stvaranje objekta klase  
KorisniciAdministratorController za dohvaćanje podataka iz liste korisnika

### **2.3.17.2. Metode**

---

**public void obrisiKorisnikButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba obrisiKorisnikButton

**public void odustaniButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odustaniButton

**private void obrisiKorisnika()** – metoda za brisanje korisnika iz sustava

## **2.3.18. Klasa OdblokirajKontaktAdministratorController**

---

Klasa se koristi za odblokiranje kontakta iz blok liste sa strane administratora.

### **2.3.18.1. Atributi**

---

**private Button odustaniButton** – gumb za odustajanje  
**private Button odblokirajKontaktButton** – gumb za odblokiranje kontakta u blok listi  
**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private sample.BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka  
**BlokListaAdministratorController podaciBlokListaAdministrator** – stvaranje objekta klase BlokListaAdministratorController za dohvaćanje podataka iz blok liste

### **2.3.18.2. Metode**

---

**public void odblokirajKontaktButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odblokirajKontaktButton

**public void odustaniButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odustaniButton

**private void odblokirajOsobu()** – metoda za odblokiraje kontakta u blok listi

## **2.3.19. Klasa OdblokirajKontaktController**

---

Klasa se koristi za odblokiranje kontakta iz blok liste sa strane korisnika.

### **2.3.19.1. Atributi**

---

**private Button odustaniButton** – gumb za odustajanje  
**private Button odblokirajKontaktButton** – gumb za odblokiranje kontakta u blok listi  
**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private sample.BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka  
**ImenikController podaciImenik** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika  
**BlokListaController podaciBlokLista** – stvaranje objekta klase BlokListaController za dohvaćanje podataka iz blok liste

### 2.3.19.2. Metode

---

**public void odblokirajKontaktButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odblokirajKontaktButton

**public void odustaniButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba odustaniButton

**private void odblokirajOsobu()** – metoda za odblokiranje kontakta u blok listi

### 2.3.20. Klasa PodaciAdministratorController

---

Klasa se koristi za popunjavanje tablice podataka o korisnicima u sustavu.

#### 2.3.20.1. Atributi

---

**public TableView<Osoba> tablePodaciAdministrator** – tablica podataka o svim korisnicima u sustavu

**private TableColumn<Osoba, String> IDCol** – 'ID' stupac tablice

**private TableColumn<Osoba, String> imeCol** – 'Ime' stupac tablice

**private TableColumn<Osoba, String> prezimeCol** – 'Prezime' stupac tablice

**private TableColumn<Osoba, String> brojTelCol** – 'Broj Telefona' stupac tablice

**private TableColumn<Osoba, String> adresaCol** – 'Adresa' stupac tablice

**private TableColumn<Osoba, String> IDKorisnikaCol** – 'ID Korisnika' stupac tablice

**private TableColumn editCol** – stupac za gumb upravljanja podacima(uređivanje)

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private ObservableList<Osoba> list** – lista u kojoj se spremaju podaci korisnika

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**public static int IDPodaci** – ID odabranog korisnika

**public static String imePodaci** – ime odabranog korisnika

**public static String prezimePodaci** – prezime odabranog korisnika

**public static String brojTelPodaci** – broj telefona odabranog korisnika

**public static String adresaPodaci** – adresa odabranog korisnika

**public static int IDKorisnikaPodaci** – ID korisnika za kojega se vežu podaci

**LoginController podaciLogin** – stvaranje objekta klase LoginController za dohvaćanje korisničkih(login) podataka

**ImenikController podaciImenik** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

### 2.3.20.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za inicijalizaciju i popunjavanje liste podataka

**public void popuniPodatke()** – metoda za popunjavanje tablice podataka i dodavanje gumba za uređivanje podataka(prikazuju se podaci svih korisnika u sustavu)

### 2.3.21. Klasa RegistracijaController

---

Klasa se koristi za registriranje korisnika u sustav.

#### 2.3.21.1. Atributi

---

**private Label registerLabel** – dinamična oznaka za prikaz raznih upozorenja ukoliko korisnik nije zadovoljio određene uvjete

**public TextField korImeField** – polje za unos korisničkog imena

**public TextField lozinkaField** – polje za unos lozinke

**public TextField imeField** – polje za unos imena

**public TextField prezimeField** – polje za prezimena

**public TextField brojTelField** – polje za unos broja telefona

**public TextField adresaField** – polje za unos adrese

**private Button registerButton** – gumb za registraciju korisnika

**public static String username** – pohranjeno unešeno korisničko ime

**public static String password** – pohranjena unešena lozinka

**public static String IDKorisnika** – pohranjeni ID kreiranog korisnika

#### 2.3.21.2. Metode

---

**public void registerButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba registerButton

**public void validateRegistration()** – metoda za provjeru točnosti register forme i provjeru je li korisnik već postoji u sustavu, nakon provjere obavlja se ubacivanje korisnika i njegovih podataka u sustav

### 2.3.22. Klasa UrediKontaktAdministratorController

---

Klasa se koristi za uređivanje kontakta u imeniku sa strane administratora.

#### 2.3.22.1. Atributi

---

**private Spinner<Integer> urediIDSpinner** – spinner za odabir ID-a kontakta

**private TextField uredilmeField** – polje za unos imena

**private TextField urediPrezimeField** – polje za unos prezimena

**private TextField urediBrojTelField** – polje za unos broja telefona

**private Label urediLabel** – dinamična oznaka koja se koristi za razna upozorenja ukoliko korisnik nije zadovoljio neke od predviđenih uvjeta

**private Button urediKontaktButton** – gumb za uređivanje kontakta u imeniku

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**ImenikAdministratorController podaci** – stvaranje objekta klase

ImenikAdministratorController za dohvaćanje podataka iz imenika

### **2.3.22.2. Metode**

---

**public void initialize(URL url, ResourceBundle rb)** – u ovom slučaju metoda se koristi za kontrolu vrijednosti koje se mogu unijeti u spinner (moguće je unijeti samo brojeve >1)

**public void urediKontaktButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba urediKontaktButton

**private void urediOsobu()** – metoda za uređivanje kontakta u imeniku

## **2.3.23. Klasa UrediKontaktController**

---

Klasa se koristi za uređivanje kontakta u imeniku sa strane korisnika.

### **2.3.23.1. Atributi**

---

**private TextField uredilmeField** – polje za unos imena

**private TextField urediPrezimeField** – polje za unos prezimena

**private TextField urediBrojTelField** – polje za unos broja telefona

**private Label urediLabel** – dinamična oznaka koja se koristi za razna upozorenja ukoliko korisnik nije zadovoljio neke od predviđenih uvjeta

**private Button urediKontaktButton** – gumb za uređivanje kontakta u imeniku

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**ImenikController podaci** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

### **2.3.23.2. Metode**

---

**public void urediKontaktButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba urediKontaktButton

**private void urediOsobu()** – metoda za uređivanje kontakta u imeniku

## **2.3.24. Klasa UrediKorisnikController**

---

Klasa se koristi za uređivanje korisnika u sustavu.

### **2.3.24.1. Atributi**

---

**private TextField urediKorImeField** – polje za unos korisničkog imena  
**private TextField urediLozinkuField** – polje za unos lozinke  
**private ComboBox<String> urediUloguBox** – polje za unos uloge (korisnik/administrator)  
**private Label urediLabel** – dinamična oznaka koja se koristi za razna upozorenja ukoliko korisnik nije zadovoljio neke od predviđenih uvjeta  
**private Button urediKorisnikButton** – gumb za uređivanje korisnika u sustavu

**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**KorisniciAdministratorController podaci** – stvaranje objekta klase  
KorisniciAdministratorController za dohvaćanje podataka iz tablice korisnika

### 2.3.24.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – metoda za popunjavanje opcija  
ComboBox-a urediUloguBox

**public void urediKorisnikButtonOnAction(ActionEvent event)** – metoda za kontroliranje  
akcije gumba urediKorisnikButton

**private void urediKorisnika()** – metoda za uređivanje korisnika u sustavu

### 2.3.25. Klasa UrediPodatkeAdministratorController

---

Klasa se koristi za uređivanje podataka u tablici podataka svih korisnika sa strane administratora.

#### 2.3.25.1. Atributi

---

**private Spinner<Integer> podaciIDSpinner** – spinner za odabir ID-a retka podataka  
**private TextField podaciImeField** – polje za unos imena  
**private TextField podaciPrezimeField** – polje za unos prezimena  
**private TextField podaciAdresaField** – polje za unos adrese  
**private TextField podaciBrojTelField** – polje za unos broja telefona  
**private Label urediLabel** – dinamična oznaka koja se koristi za razna upozorenja ukoliko korisnik nije zadovoljio neke od predviđenih uvjeta  
**private Button urediPodatkeButton** – gumb za uređivanje podataka u tablici podataka

**private Connection connection** – atribut za spremanje veze s bazom podataka  
**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**PodaciAdministratorController podaci** – stvaranje objekta klase  
PodaciAdministratorController za dohvaćanje podataka iz tablice podataka

#### 2.3.25.2. Metode

---

**public void initialize(URL url, ResourceBundle rb)** – u ovom slučaju metoda se koristi za kontrolu vrijednosti koje se mogu unijeti u spinner (moguće je unijeti samo brojeve >1)

**public void urediPodatkeButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba urediPodatkeButton

**private void urediPodatke()** – metoda za uređivanje podataka u tablici podataka

### 2.3.26. Klasa UrediPodatkeController

---

Klasa se koristi za uređivanje vlastitih podataka od strane korisnika.

#### 2.3.26.1. Atributi

---

**private TextField podaciImeField** – polje za unos imena

**private TextField podaciPrezimeField** – polje za unos prezimena

**private TextField podaciAdresaField** – polje za unos adrese

**private TextField podaciBrojTelField** – polje za unos broja telefona

**private Label urediLabel** – dinamična oznaka koja se koristi za razna upozorenja ukoliko korisnik nije zadovoljio neke od predviđenih uvjeta

**private Button urediPodatkeButton** – gumb za uređivanje podataka u tablici podataka

**private Connection connection** – atribut za spremanje veze s bazom podataka

**private BazaPodataka bazaPodataka** – stvaranje objekta klase BazaPodataka

**ImenikController podaci** – stvaranje objekta klase ImenikController za dohvaćanje podataka iz imenika

#### 2.3.26.2. Metode

---

**public void urediPodatkeButtonOnAction(ActionEvent event)** – metoda za kontroliranje akcije gumba urediPodatkeButton

**private void urediPodatke()** – metoda za uređivanje podataka korisnika

### 2.3.27. Klasa VecDodanoController

---

Klasa se koristi za upozorenje korisniku da je kontakt već dodan na listu favorita

#### 2.3.27.1. Atributi

---

**private Button uReduButton** – gumb za potvrdu i izlazak iz prozora

#### 2.3.27.2. Metode

---

**public void uReduButtonOnAction(ActionEvent event)** – metoda za kontrolu akcije gumba uReduButton