SVEUČILIŠTE U MOSTARU

FAKULTET STROJARSTVA RAČUNARSTVA I ELEKTROTEHNIKE

# 

# Programirnje u Javi

# *Projektni zadatak: Hospital managment system*

**Vinko-Tino Zlopaša, Jure Bakula, Andro Raspudić**

Akademska godina 2020/2021.

**PROJEKTNI ZADATAK:**

**Opis problema:**

Očekuje se da ce se troškovi za zdravstvo u svijetu povećati sa sadašnjih 7,7 bilijuna USD na 10 bilijuna USD u 2022. godine. To znači da globalno stanovništvo brzo stari i ima rizik od veće stop kroničnih bolesti. Zbog toga se digitalne tehnologije i drugi pristupi razvijaju. Nažalost činjenica je da su to često skupe inovacije. Sjedinjene Američke Države nemaju univerzalnu zdravstvenu zaštitu te su čak potrošile preko 1,5 bilijuna dolara saveznih sredstava na ovaj sektor 2017 godine. Istraživanja su pokazala da 44 posto Amerikanaca nije htjelo konzultirati svoje ljiečnike kada su bili bolesni, a još je 40 posto dodalo da nisu pratili preporučeno liječenje. Jedno od glavnih rješenja je bilo eliminirati papirnate zapise o informacijama pacijenata i preći na digitalne medije. Prelaskom na digitlani medji smanjili su se troškovi vezani za fizičko skladištenje papira, nabavu papira i slično. Zdravstvena uprava tradicionalno je bila vrlo težak pristup, digitalizacijom se uveo mnogo lakši unos i pregled informacija, praćenje različitih događaja kao i brže provođenje pregleda i manje čekanja u bolnici.

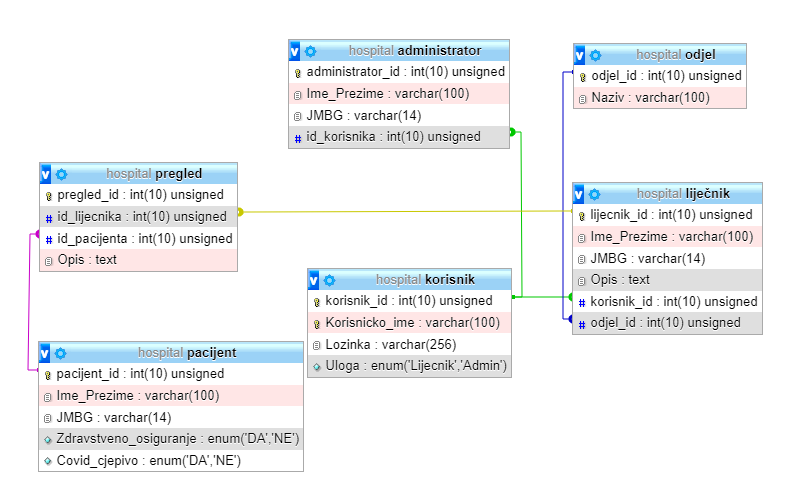
Naš projekt je pokušaj takvog sustava digitalizacije na jednostavnom primjeru sustava upravljanja bolnicama. Aplikacija ima sustav prijave admina ili lijecnika gdje se prijavljuju svojim korisničkim imenom i šifrom te odabirom odgovarajuće uloge (admin ili lijecnik).

Uloga admina ima uvid u sve lijecnike u sistemu te svu informaciju o svakom lijecniku kao što su korisničko ime, lozinku (sifriranu), ime i prezime, odjel te opis. Također ima opcije “dodaj” koja dodaje novog lijecnika, “uredi” koja uređuje postojećeg lijecnika, “ukloni” koja uklanjanja lijecnika i opcije “osviježi” koja osviježi listu lijecnika. Svi stupci liste se mogu zamijeniti i povećati po želji. Odabirom lijecnika iz liste mogu se obaviti sve opcije admina. Opcija “Uredi” otvara novi prozor gdje admin moze promijeniti sve informacije lijecnika, te uz ponovni unos svoje lozinke sprema podatke u bazu pomoću dugma Spremi. Prozor ostaje otvoren sa svim podacima u slučaju neke dodatne promjene. Uz dugme Spremi postoji i dugme Očisti koje čisti sva polja. Opcija “Dodaj” također otvara novi prozor sa svim praznim poljima za unos novog liječnika. Admin ima i opciju “Logout” koja ga odjavljuje i vraca na prozor prijave. Uloga admin također ima opciju “Pronađite doktora” gjde admin može ukucati ime doktora u polje te pronaci željenog doktora. Opcijom “Osviježi” vracamo se na cijlu listu doktora.

Uloga liječnika se prijavljuje na sustav pomocu korisničkog imena i lozinke koju je admin prije dodao u bazu podataka. Liječnik ima dvije stranice “Doktor profil” i “Pacijenti”. Na stranici “Doktor Profil” nalaze se informacije o lijecniku te opcija uređivanja svog profila. Na stranici “Pacijenti” nalazi se lista pacijenata logiranog lijecnika. Tu lijecnik moze viditi sve svoje pacijente te ih dodavati, uređivati i brisati kao što Admin moze sa lijecnicima. Lijencik dodaje pacijenta opcijom “Dodaj” koja otvara novi prozor sa poljima “Ime i prezime”, “JMBG”, “Osiguranje” s odabirom DA ili NE, “Covid Cjepivo” s odabirom DA ili NE, “Pregled” i “Datum” polje koje otvara kalendar sa mogucnosti odabirom datuma pregleda. Opcija “Uredi” otvara novi prozot sa svim prethodnim poljima za uređivanje podataka. Također na stranici “Pacijeniti” imamo opciju “Pronađi Pacijenta” pomoću koje nalazimo unešenog pacijenta.

**1. Modeliranje podataka**

**1.1. Relacijski model podataka**



**1.2. Opis tablica u relacijskom modelu**

**Administrator table**

|  |  |  |
| --- | --- | --- |
| **Naziv atributa** | **Integritet**  ***(PK/FK, NULL / NOT NULL)*** | **Kratki opis atributa** |
| **administrator\_id** | PK | *ID administratora* |
| **Ime\_Prezime** | NOT NULL | *Ime i prezime administratora* |
| **JMBG** | NOT NULL | *JMBG administratora* |
| **id\_korisnika** | FK | *id administratora* |

**Pacijent table**

|  |  |  |
| --- | --- | --- |
| **Naziv atributa** | **Integritet**  ***(PK/FK, NULL / NOT NULL)*** | **Kratki opis atributa** |
| **pacijent\_id** | PK | *ID pacijenta* |
| **Ime\_Prezime** | NOT NULL | *Ime i prezime pacijenta* |
| **JMBG** | UNIQ | *JMBG pacijenta* |
| **Zdravstveno\_osiguranje** | NOT NULL | *Zdravstveno osiguranje pacijenta* |
| **Covid\_cjepivo** | NOT NULL | *Covid cjepivo pacijenta* |

**Korisnik table**

|  |  |  |
| --- | --- | --- |
| **Naziv atributa** | **Integritet**  ***(PK/FK, NULL / NOT NULL)*** | **Kratki opis atributa** |
| **korisnik\_id** | PK | *ID korisnika* |
| **Korisnicko\_ime** | NOT NULL | *Ime korisnika* |
| **Lozinka** | NOT NULL | *Lozinka korisnika* |
| **Uloga** | NOT NULL | *Uloga korisnika* |
|  |  |  |

**Pregled table**

|  |  |  |
| --- | --- | --- |
| **Naziv atributa** | **Integritet**  ***(PK/FK, NULL / NOT NULL)*** | **Kratki opis atributa** |
| **pregled\_id** | PK | *ID pregleda* |
| **id\_lijecnika** | NOT NULL | *ID lijecnika* |
| **id\_pacijenta** | NOT NULL | *ID pacijenta* |
| **Opis** | NULL | *Opis pregleda* |
|  |  |  |

**Liječnik table**

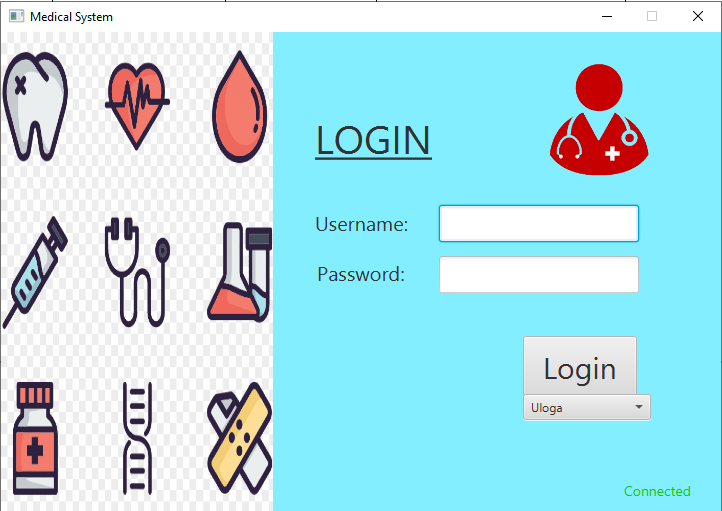
|  |  |  |
| --- | --- | --- |
| **Naziv atributa** | **Integritet**  ***(PK/FK, NULL / NOT NULL)*** | **Kratki opis atributa** |
| **lijecnik\_id** | PK | *ID lijecnika* |
| **Ime\_Prezime** | NOT NULL | *Ime i prezime lijecnika* |
| **JMBG** | NULL | *JMBG lijecnika* |
| **Opis** | NULL | *Opis pregleda* |
| **korisnik\_id** | NULL | *ID korisnika* |
| **odjel\_id** | NULL | *ID odjela* |

**Odjel table**

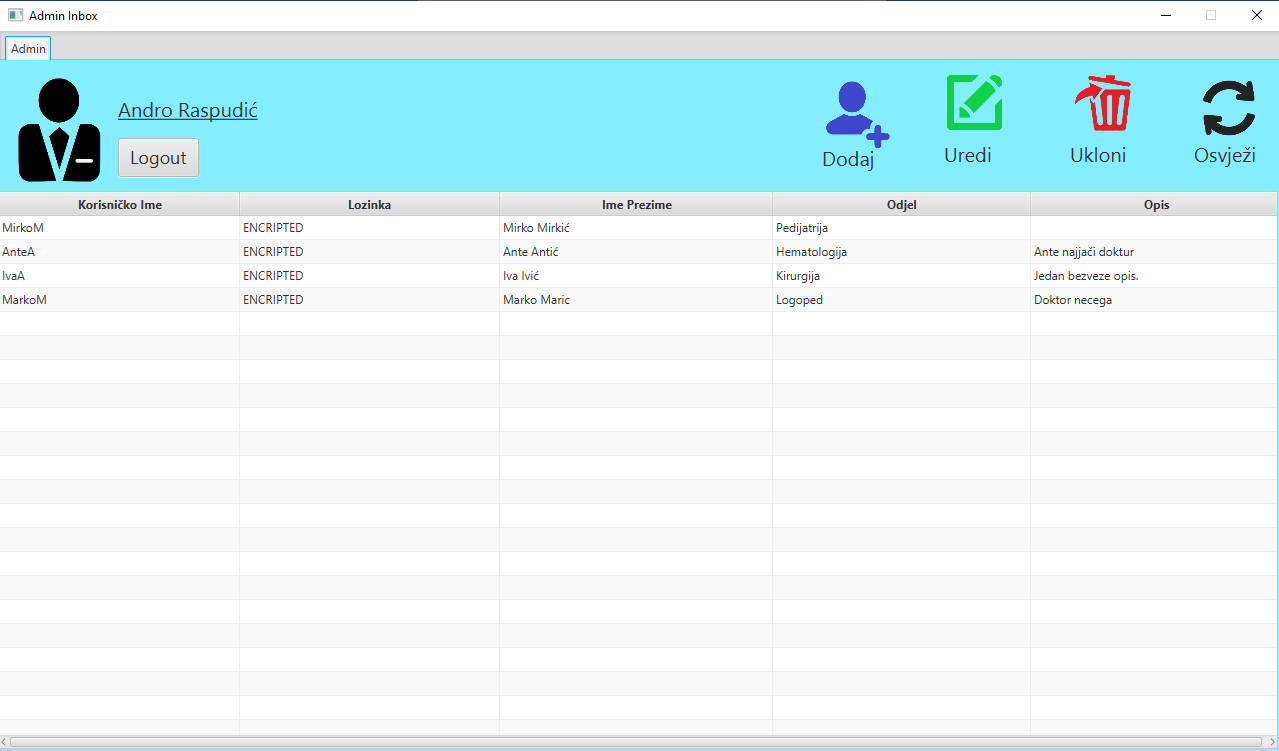
|  |  |  |
| --- | --- | --- |
| **Naziv atributa** | **Integritet**  ***(PK/FK, NULL / NOT NULL)*** | **Kratki opis atributa** |
| **odjel\_id** | PK | *ID odjela* |
| **Naziv** | NOT NULL | *Naziv odjela* |

**1.3. Aplikacija u Javi**

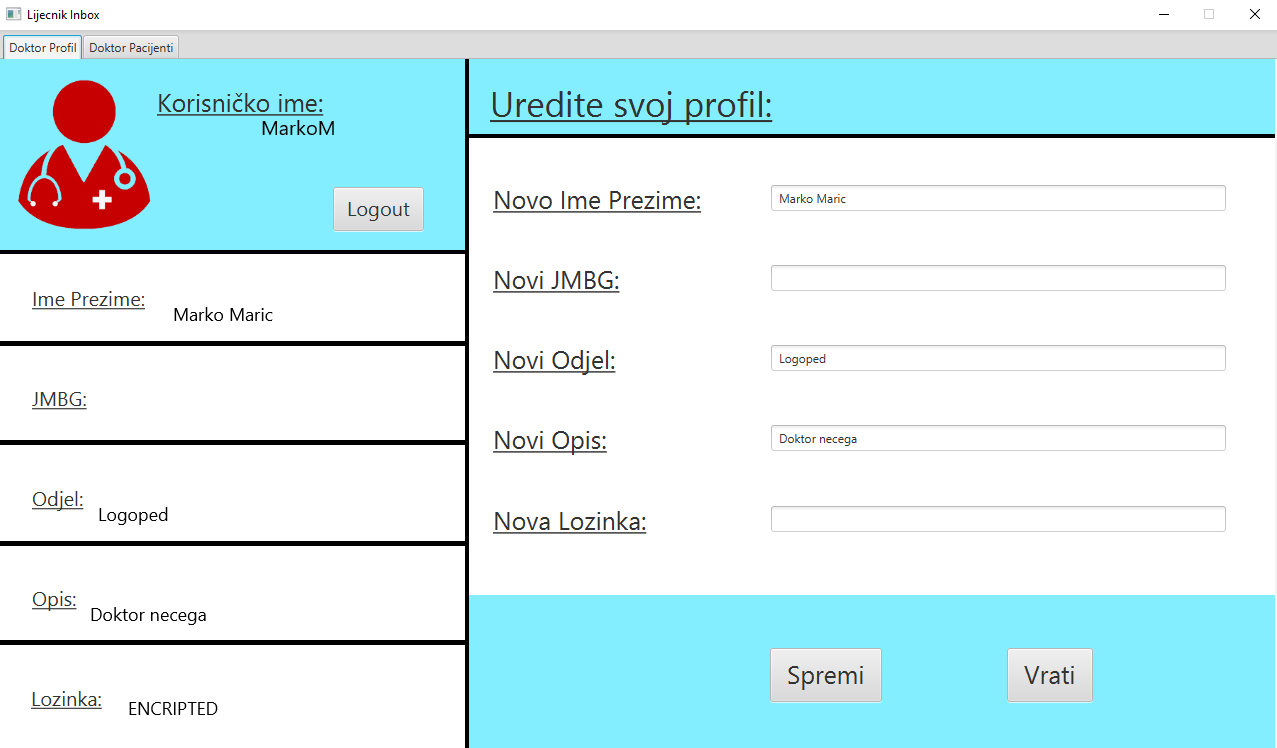
Login:



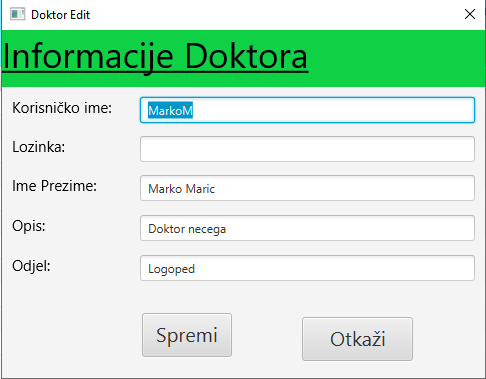
Main Menu (Admin):



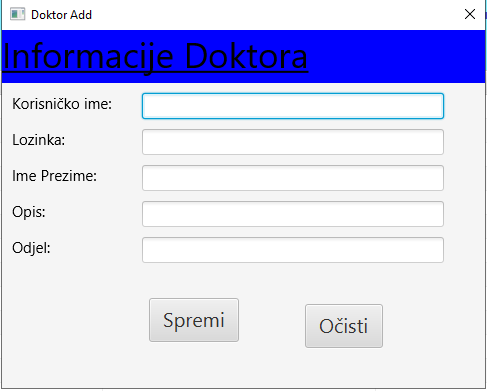
Main menu (Lijecnik):



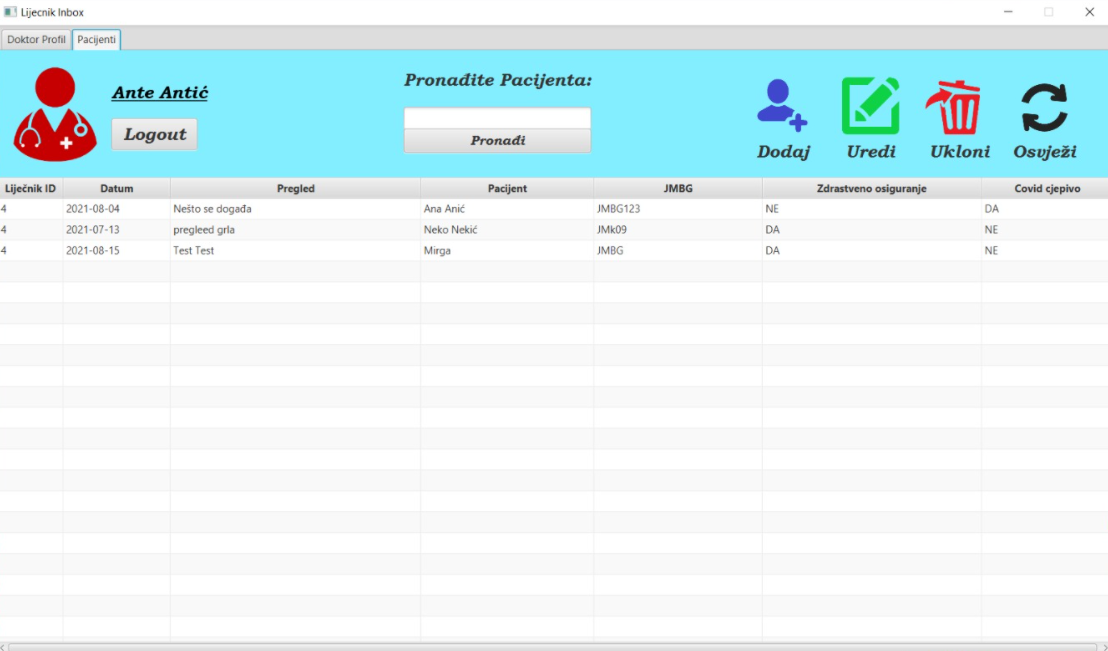
Uredi opcija:



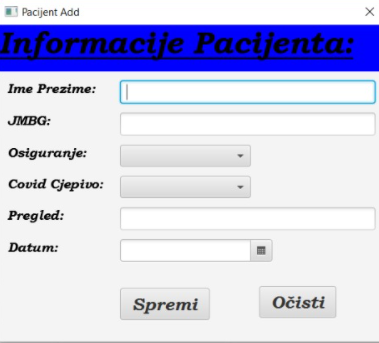
Dodaj opcija:



Pacijenti (Lijecnik):



Dodaj Pacijenta (Lijecnik):



Uredi Pacijenta (Lijecnik):



**2. Model programske podrške**

*Drugi dio zadatka se odnosi na modeliranje programske podrške, detaljno čete 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*
   1. *Atribute – kratko pisati atribute koje se nalaze unutar klase;*
   2. *Metode – kratko opisati metode koje se koriste za što služe i opišite njihov povratni tip;*

**2.1. Paketi**

Unutar MedicalSystem nalaze se paketi Admin, DataBase, Doktor, Login i svaki ima svoje nuzne slike.

**2.2. Klase**

Klase aplikacije su:

* AddDoktorController
* AdminController
* EditDoktorController
* ListOfDoktors
* DbConnection
* AddPacijentController
* Cjepivo
* DoktorController
* EditPacijentController
* LisOfPacijents
* Zdravstveno
* Controller
* LoginModul
* Main
* Option

**2.2.1. Klasa AddDoktorController**

Jedna od CRUD operacija glavnog kontrolera.

***2.2.1.1. Atributi klase AddDokorController***

private TextField addKorisnikIme; - polje za unos korisnickog imena  
private TextField addLozinka; - polje za unos lozinke korisnika  
private ComboBox<Option> addUloga; - kombinirani okvir kombinira padajuci popis i upis za ulogu  
private TextField addOdjel; - polje za unos odjela  
private TextField addOpis; - polje za unos opisa  
private TextField addImePrezime; -polje za unos Ime i prezime

***2.2.1.2. Metode klase AddDokorController***

void clean(MouseEvent event) { }- Metoda cisti polja prozora.

**2.2.2 Klasa AdminController**

Glavni kontroler CRUD operacija za Admin klasu.

***2.2.2.1. Atributi klase AdminController***

private TableView<ListOfDoktors> DoktorTable; -lista doktora  
private TableColumn<?, ?> logDoktor; -stupac doktora  
private TableColumn<?, ?> lozinkaDok; -stupac lozinke doktora  
private TableColumn<?, ?> imeDok; -stupac imena doktora  
private TableColumn<?, ?> odjelDok; -stupac odjela doktora  
private Label LogiraniAdmin; -label logiranog admina  
private Button LogoutBtn; -dugme za logout  
private Label Adminname; -label imena admina  
private TextField TrazeniDoktor; -polje za unos trazenog doktora  
private ObservableList<ListOfDoktors> doktorList; -lista doktora

***2.2.2.2. Metode klase AdminController***

private void addDoktor(MouseEvent mouseEvent) throws IOException {} - metodom odajemo novog Doktora cime se otvara novi prozor za unos podataka.

**2.2.3 Klasa EditDoktorController**

Jedna od CRUD operacija Admin klase.

***2.2.3.1. Atributi klase EditDoktorController***

private TextField editKorisnikIme; -polje za unos novog kor. imena Doktora  
private TextField editLozinka; -polje za unos novu lozinku Doktora  
private TextField editImePrezime; -polje za unos novog imena Doktora  
private TextField editOpis; -polje za unos novog o pisaDoktora  
private TextField editOdjel; -polje za unos novog odjela Doktora  
private Button cancel; -dugme za prekid

***2.2.3.2. Metode klase EditDoktorController***

void cancelEdit(MouseEvent event) {} - ovom metodom prekidamo uređivanje unutar Admina

**2.2.4 Klasa ListOfDoktors**

Klasa koja omogućuje ispis liste doktora.

***2.2.4.1. Atributi klase* ListOfDoktors**

private final StringProperty User\_Name; -ime doktora u listi  
private final StringProperty Password; -lozinka doktora  
private final StringProperty Ime\_Prezime; -ime i prezime doktora  
private final StringProperty Opis; -opis doktora  
private final StringProperty Odjel; -odjel doktora

***2.2.4.2. Metode klase* ListOfDoktors**

*public String getOdjel() {} - ovom metodom dobivamo odjel Doktora pomocu return funkcije.*

**2.2.5 Klasa DbConnection**

Klasa koja se pokušava spojiti na MySQL bazu. U slučaju greške vraća error.

***2.2.5.1. Atributi klase DbConnection***

private static final String *DbName - ime baze na koju se spajamo*  
private static final String *User - korisnicko ime za spajanje na bazu*  
private static final String *Password - lozinka baze na koju se spajamo*  
private static final String *ConnectionUrl – url spoja baze*

***2.2.5.2. Metode klase DbConnection***

*public static Connection getConnection() throws SQLException {} - ovom metodom omogucujemo konekciju sa bazom, unutra imamo try i catch u slucaju da bude porblema sa konekcijom vraca se error.*

**2.2.6 Klasa AddPacijentController**

Klasa koja omogućuje dodavanje Pacijenta.

***2.2.6.1. Atributi klase* AddPacijentController**

private TextField PacijentIme; -polje za unos novog imena pacijenta  
private TextField JMBGPacijent; -polje za unos novog JBBG-a pacijenta  
private ComboBox<Zdravstveno> OsiguranjePacijent; -kombinirani okvir osiguranja pacijenta  
private ComboBox<Cjepivo> CovidPacijent; -kombinirani okvir covid pacijenta  
private TextField PregledPacijent; -polje za unos pregleda pacijenta  
private DatePicker DatumPacijent; -odabir datuma

***2.2.6.2. Metode klase* AddPacijentController**

*void saveP(MouseEvent event) { }- ovom metodom spasavamo Doktora u bazu.*

**2.2.7 Klasa Cijepivo**

Klasa Cjepivo ima enumerate vrijednost od DA ili NE.

***2.2.7.1. Atributi klase* Cijepivo**

*DA,*

*NE;*

***2.2.7.2. Metode klase* Cijepivo**

private Cjepivo() {}

**2.2.8 Klasa DoktorController**

DoktorController klasa je glavna klasa za CRUD operacije unutar Doktor paketa.

***2.2.8.1. Atributi klase* DoktorController**

private Text DoktorName; -tekst ime doktora  
private Text ImePrezime; -tekst prezimedoktora  
private Text JMBG; -tekst JMBG doktora  
private Text Odjel; -tekst odjela doktora  
private Text Opis; -tekst opisa doktora  
private Text Lozinka; -tekst lozinke doktora  
private Button DokLogout; -dugme logout doktora  
private TextField editNovoIme; -polje za unos novog imena doktora  
private TextField editNoviJMBG; -polje za unos novog JMBG-a doktora  
private TextField editNoviodjel; -polje za unos novog odjela doktora  
private TextField editNoviOpis; -polje za unos novog opisa doktora  
private TextField editNovaLozinka; -polje za unos nove lozinke doktora  
private Text DokName; -tekst ime doktora  
private TableView<ListOfPacjents> PacijentTable; -lista pacijenata  
private TableColumn<?, ?> LijecnikIDP; -stupac ID-a lijecnika  
private TableColumn<?, ?> DatumP; -stupac datuma  
private TableColumn<?, ?> PregledP; -stupac pregleda  
private TableColumn<?, ?> PacijentP; -stupac pregleda pacijenta  
private TableColumn<?, ?> JMBGP; -stupac JMBG pacijenta  
private TableColumn<?, ?> OsiguranjeP; -stupac ID-a lijecnika  
private TableColumn<?, ?> CovidP; -stupac covid cijepljenja pacijenta  
private Button DokSave; -dugme za spremanje  
private Button DokReturn; -dugme za natrag  
private TextField PacijentTrazilica; -unos polja za trazenje pacijenta  
Connection con = null;  
PreparedStatement prepS = null;  
ResultSet rs = null;  
int DokKorisnikID = 0;  
int DoktorID = 0;  
int OdjelID = 0;  
String lozinka = null;  
String pocetniJMBG = null;  
String TrazenoIme = null;  
private ObservableList<ListOfPacjents> pacijentiLista;

***2.2.8.2. Metode klase* DoktorController**

private void ispisPacijenata() { } - ovom metodom mozemo ispisati listu Pacijenata tj. jedna od CRUD metoda view.

**2.2.9 Klasa EditPacijentController**

EditPacijentController klasa omogućuje uređivanje Pacijenta. Jedna je od CRUD operacija.

***2.2.9.1. Atributi klase* EditPacijentController**

private TextField ImePacijentaEdit; -polje za unos novog imena pacijenta  
private TextField JMBGPacijentEdit; -polje za unos novog JMBG-a pacijenta  
private ComboBox<Zdravstveno> OsiguranjePacijentaEdit; -kombinirani okvir osiguranja pacijenta  
private ComboBox<Cjepivo> CovidPacijentaEdit; -kombinirani okvir covid cijepljenja pacijenta  
private TextField PregledPacijentaEdit; -polje za unos novog pregleda pacijenta  
private DatePicker DatumPacijentaEdit; -polje za unos novog datuma pregleda  
private Button cancelEditPacijent; -dugme za prekid uređivanja  
Connection con = null;  
PreparedStatement prepS = null;  
ResultSet rs = null;  
String bivsiJMBG = null;  
String query = null;  
String query2 = null;  
int PacijentId = 0;  
int jelSveOK = 0;  
int IDkorisnika;

***2.2.9.2. Metode klase* EditPacijentController**

void cancelEditPacijent(MouseEvent event) {} - metoda prekida uređivanja Pacijenta.

**2.2.11 Klasa ListOfPacijents**

Omogućuje ispis svih Pacijenata.

***2.2.11.1. Atributi klase* ListOfPacijents**

private final IntegerProperty DoktorID;   
private final StringProperty Datum; -string datuma pacijenta  
private final StringProperty Pregled; -string pregleda pacijenta  
private final StringProperty Pacijent; -string pacijenta  
private final StringProperty JMBG; -string JMBG-a pacijenta  
private final StringProperty Osiguranje; -string osiguranja pacijenta  
private final StringProperty Cjepivo; -string covid cijepiva pacijenta

***2.2.11.2. Metode klase* ListOfPacijents**

public int getDoktorID() {} - ovom metodom dobivamo ID doktora tako sto funkcija vraca this.DoktorID.get().

**2.2.12 Klasa Zdravstveno**

Klasa Zdravstveno oznacava da li pacijent ima Zdravstveno osiguranje ili ne.

***2.2.12.1. Atributi klase* Zdravstveno**

*DA,*

*NE;*

***2.2.12.2. Metode klase* Zdravstveno**

private Zdravstveno() {  
}

**2.2.13 Klasa Controller**

Glavni kontroler za CRUD operacije unutar paketa Login.

***2.2.13.1. Atributi klase* Controller**

LoginModul loginModul = new LoginModul(); - novi login modul  
private Label chechkCon; -provjera konekcije  
private TextField Username; -unos polja korisnickog imena  
private ComboBox<Option> ComboBox; - kombinirani okvir  
private Button LoginButton; -dugme za prijavu  
private PasswordField Password; -polje za lozinku

***2.2.13.2. Metode klase* Controller**

public void AdminLogin() { } - metoda provjere logiranja admina.

**2.2.14 Klasa LoginModul**

Prikazuje te omogućuje prijavu korisnika na sustav. Provjerava ispravnost unesenih podataka

***2.2.14.1. Atributi klase* LoginModul**

***2.2.14.2. Metode klase* LoginModul**

public boolean isConnected() {} - metoda provjerava da li je iko logiran.

**2.2.15 Klasa Main**

Glavna klasa alpikacije Main u paketu Login.

***2.2.15.1. Atributi klase* Main**

***2.2.15.2. Metode klase* Main**

public void start(Stage primaryStage) throws Exception {} - glavna metoda aplikacije.