



Univerzitet u Sarajevu  
Elektrotehnički fakultet u Sarajevu  
Odsjek za računarstvo i informatiku



## **Izvještaj**

**Aplikacija za ljekarsku ordinaciju**

**Student: Kanita Dervić**

**Broj indexa: 18176**

**Predmetni nastavnik: Doc.dr Vedran Ljubović**

**Sarajevo, 13.09.2020.**

# Sadržaj

<b>1</b>	<b>Opis aplikacije</b>	<b>1</b>
<b>2</b>	<b>Osnovne ideje implementacije</b>	<b>2</b>
<b>3</b>	<b>Implementacija</b>	<b>3</b>
3.1	Osnovni koncepti	3
3.2	Baza podataka	3
3.3	OOP	3
3.4	GUI	3
3.5	Datoteke	3
3.6	Mrežno programiranje	4
3.7	Izvještaji	4
3.8	Testovi	4
<b>4</b>	<b>Poboljšanje</b>	<b>5</b>
<b>5</b>	<b>Pristupni podaci</b>	<b>6</b>

# 1. Opis aplikacije

Aplikacija predstavljena dalje u izvještaju je napravljena u svrhu demonstracije korištenja aplikacije za neku ljebarsku ordinaciju, te ista predstavlja projekat na predmetu Razvoj programskih rješenja. Sama aplikacija je dosta ograničena, u smislu da ima samo osnovne funkcionalnosti uz dosta prostora za poboljšanje. Trenutno omogućene funkcionalnosti će biti navedene u nastavku.

Korisnik u samom početku ima pravo da se prijavi na svoj račun (ukoliko već ima nalog), ili da napravi novu registraciju. Na prozoru za registraciju se unose osnovni podaci o korisniku (ime, prezime, spol, broj telefona, e-mail adresa i lozinka), a ukoliko je novi korisnik ujedno i član ljebarske ordinacije, ima pravo da označi check box, unese lozinku ordinacije, te na taj način pristupa funkcionalnostima doktora koje će biti opisane kasnije u tekstu. Nakon unosa podataka i validacije, korisnik se vraća na početni ekran odakle se dalje može prijaviti za korištenje aplikacije.

Ukoliko je korisnik definisan kao pacijent, otvara se novi prozor na kojem on vidi podatke o svom korisničkom nalogu, te listu zakazanih termina. Termini su predstavljeni datumom i povodom zakazivanja termina(zdravstveni problem korisnika). Korisnik ima mogućnost da otkaže termin, doda novi, ili modificira postojeći (u smislu mijenjanja datuma ili biranja novog doktora).

Korisnik koji je član ljebarske ordinacije ima drugi pristup aplikaciji. Pri otvaranju prozora, prikazani su svi termini koje doktor ima zakazane, poredane po datumima uz informacije o zdravstvenom problemu koji pacijent ima, te osnovne lične podatke pacijenta. Također, ima mogućnost da vidi detaljnije informacije o pacijentu, otkaže termin, spasi termine u tekstualnu datoteku, ili da napravi izvještaj istih.

## 2. Osnovne ideje implementacije

Početne ideje se ogledaju u pravljenju što jednostavnije aplikacije za svakodnevne korisnike. Kontrole za obične korisnike aplikacije su grupisane i jasne po sebi. Korisnicima (i običnim i članovima ljekarske ordinacije) se daje lagan pregled nadolazećih zakazanih termina. Za članove ljekarske ordinacije je korišten toolbar sa iznad navedenim opcijama.

Forma za registraciju primjenjuje gestalt principe po grupaciji polja, te se koriste neke "logične" validacije podataka (u smislu da kratka imena i prezimena nisu validna, e-mail mora sadržavati znak "@", lozinke moraju imati jedno veliko slovo i broj, brojevi telefona moraju imati oblik "xxx/xxx-xxx" i slično).

Model klase prate JavaBeans specifikacije. Napravljeni su vlastiti tipovi izuzetaka koji bacaju izuzetke u slučaju prosljeđivanja netačnih parametara za datum termina (termin se ne može rezervisati u prošlosti). Dalje, korištena je SQLite baza podataka, te se u projektu nalaze DAO klase koje kontrolišu samu bazu. DAO klase za različite tabele su odvojene.

Aplikacija je dostupna na bosanskom i engleskom jeziku.

Izveštaji su kreirani pomoću Jaspersoft studia.

## 3. Implementacija

Data su detaljnija objašnjenja implementacije projekta.

### 3.1. Osnovni koncepti

Projekat koristi model klase koje prate JavaBeans specifikaciju. Dalje, korištene su Java kolekcije (ArrayList, ObservableList). Kako je aplikacija predstavljena u jednostavnom obliku, osnovne Java kolekcije su bile dovoljne.

### 3.2. Baza podataka

Korištena je SQLite baza podataka. Projekat ima kreirane tri tabele.

Tabela user predstavlja pohranjivanje samog korisnika u bazu zbog dalje registracije. Ova tabela također čuva informaciju o tome da li je određeni korisnik član ljebarske ordinacije ili obični korisnik (kolona doctor).

Tabela appointment predstavlja samo zakazivanje termina. Napravljena je na način da čuva ID pacijenta, ID doktora, te datum termina u obliku TEXT-a.

Tabela disease čuva informacije o postojećim zdravstvenim problemima koji su uneseni od strane korisnika. Povezane su sa korisnikom na način da čuvamo informaciju o ID-u korisnika.

### 3.3. OOP

Koncepti objektno orijentiranog programiranja se vide u nasljeđivanju klase User od strane klasa Doctor i Patient. Patient se u ovom slučaju razlikuje jer sadrži i informacije o registrovanim zdravstvenim problemima određenog pacijenta.

### 3.4. GUI

Za implementaciju grafičkog interfejsa, korišteni su razni elementi. Registracija korisnika je koristila TextField, PasswordField, CheckBox, DatePicker, Button, TableView, ListView, ChoiceBox, RadioButton, MenuBar. Predstavljanje navedenih elemenata se vršilo preko GridPane-a i AnchorPane-a. Navedeni elementi su kontrolisani od strane controller-a (imenovani su po fxml file-ovima). Postavljani su listeneri preko kojih se provjeravaju unesene informacije.

Prikaz validnosti polja je omogućen preko .css file-a.

Primjenjeni su Gestalt principi pri formi za registraciju, te su elementi na svim prozorima grupisani po funkcijama koje vrše.

### 3.5. Datoteke

Članovi ljebarske ordinacije imaju mogućnost zapisivanja termina u tekstualnu datoteku zbog lakšeg praćenja. Datu funkciju vrši metoda "saveAppointmentsAc-

tion" u Controllers.DoctorController.

### **3.6. Mrežno programiranje**

Mrežno programiranje se koristi prilikom dodavanja novog termina, kada korisnik vrši pretragu određenog zdravstvenog problema. Navedena funkcionalnost se vrši u searchAction metodi u NewAppointmentController. Na URL se šalje vrijednost iz polja za pretragu, te se listaju zadovoljavajući rezultati. U istoj funkciji koristimo thread kako bi naveli rezultate GET metode, uz ažuriranje ikone koja govori da li se učitavaju podaci ili je pretraga završena. GET metodu izvršava funkcija executeGetMethod koja vraća JSON format.

### **3.7. Izvještaji**

Izvještaji su implementirani preko Jaspersoft studio. Članovi ljekarske ordinacije imaju opciju "Print" koja otvara izvještaj svih termina za datog korisnika. Klasa utilities. DoctorReport vrši prikazivanje samog izvještaja.

### **3.8. Testovi**

Uz projekat su priloženi JUnit i JavaFX testovi koji testiraju osnovne funkcionalnosti aplikacije.

## 4. Poboljšanje

Kao što je navedeno na početku, radi se o vrlo običnoj, ali ograničenoj aplikaciji. Predstavljene su osnovne funkcionalnosti koje bi bile potrebne jednoj ljekarskoj ordinaciji.

Za potrebe korisnika, bilo bi potrebno dodati neke detaljnije informacije o zakazivanju termina, mogućnost uvođenja redovnih termina, direktnog kontaktiranja člana ordinacije i slično.

Članovi aplikacije se mogu proširiti u smislu da se doda funkcija sekretara koji će upravljati terminima za ordinaciju, zatim dodati specijalizaciju članova ljekarskog tima (ne rade svi ljekari isti posao) i slično.

## 5. Pristupni podaci

Administratorski nalog:  
E-mail: admin@faks.com  
Lozinka:Test123

Obični nalog:  
E-mail: ssaric@faks.com  
Lozinka:Test123

Prilikom registracije (dodavanje člana ordinacije):  
Lozinka ordinacije: secretAdmin