

Elektrotehnički fakultet Sarajevo

Izvještaj o projektu

Iz predmeta: Razvoj programskih rješenja

Aplikacija za rezervaciju u restoranu

Profesori: Vedran Ljubović, Senka Krivić

Student: Almina Brulić

Sarajevo, 5.9.2022

SADRŽAJ:

1.	OPIS APLIKACIJE	3
2.	Implementacija.....	4
2.1	Korištenje githuba	4
2.2	Baza podataka	4
2.3	Funkcionalno programiranje	4
2.4	Java kolekcije.....	4
2.5	Objektno zasnovano programiranje	5
2.6	Grafički korisnički interfejs	5
2.7	TestFX i Unit testiranje.....	5
2.8	Datoteke	5
2.9	Izveštavanje	5
2.10	Lokalizacija.....	5
2.11	Maven	6
3.	Prilika za unaprjeđivanje.....	7
4.	Dodatak	7

1. OPIS APLIKACIJE

Aplikacija „Aplikacija za rezervaciju u restoranu“ omogućuje korisnicima i administratorima razne funkcionalnosti, te je prevedena na 3 jezika: Bosanski, Engleski i Njemački jezik. Prije svega, svaki korisnik se treba prijaviti ili registrovati ukoliko nema korisnički račun. Forma za registraciju će imati određene restrikcije, poput dozvoljenog korisničkog imena i lozinke. Nakon registracije, korisnik mora da se prijavi sa prethodno unesenim korisničkim imenom i lozinkom. Nakon toga se otvara glavni prozor za korisnika, odnosno glavni prozor za administratora ukoliko je takva uloga odabrana na nekom od radio button-a sa login forme.

Što se tiče panela za korisnike, korisnici imaju mogućnost rezervacije, otkazivanja rezervacije, pregleda menija restorana, dodavanja raznih jela na svoju „wish“ listu, te prikaz ukupnog iznosa stavki sa wish-liste. Korisnici mogu na tabeli vidjeti stavke menija restorana, kao i odabrati prikaz samo veganskih/vegetarijanskih jela. U dijelu prozora za popunjavanje podataka prilikom rezervacije potrebno je da korisnik unese sve svoje podatke koje je unio prilikom registracije, inače će proces rezervacije biti odbijen, gdje će korisnik biti upozoren o istom. Također, korisnik smije imati maksimalno jednu rezervaciju, koju može da otkaže do 24h prije same rezervacije.

Što se admin panela tiče, jedan admin ima pristup aplikaciji. On može manuelno dodavati nove korisnike, brisati ih i uređivati podatke. Isto tako može dodavati rezervacije, brisati ih i uređivati. Admin klikom na dugme „Printaj današnje rezervacije“ može da dobije izvještaj o svim rezervacijama od tog dana, dok klikom na dume „Printaj meni“ može dobiti izvještaj stavki sa menija, koje također može uređivati.

2. Implementacija

Prilikom implementacije projekta, koristila sam IntelliJ studio aplikaciju, na kojoj je podešena SDK verzija 17. Za izvještaje sam koristila Jaspersoft studio, a za kreiranje fxml file-ova sam koristila Scene Builder. Koristila sam MVC patern.

U daljem tekstu ćete moći pročitati osnovne detalje o implementaciji ovog projekta.

2.1 Korištenje githuba

Za kreiranje repozitorija na githubu sa prihvatila pozivnicu koja se nalazila na c2 u dokumentu „Projekat“, gdje sam nakon toga uradila clone repozitorija i otvorila u IntelliJ okruženju. Nakon određenih promjena radila sam lokalni commit, gdje sam pri kraju implementacije uradila push iz okruženja na github. Pošto mi prvobitno nije radio push, morala sam kroz terminal u okruženju komandom „git push origin main --force“ da manuelno pusham projekat na github profil.

2.2 Baza podataka

Za bazu podataka je korištena SQLite baza, koju sam prethodno uključila među ostale biblioteke projekta. Klasa u kojoj se nalazi konekcija, kao i svi upiti prilikom rada sa bazom se može naći u folderu „Common java classes“ pod nazivom „DatabaseDAO“. Podešeno je da se baza prilikom pokretanja programa kreira u Vašem Users folderu. Za potrebe projekta kreirano je 5 tabela, gdje se detaljniji prikaz može naći na ER dijagramu.

2.3 Funkcionalno programiranje

Funkcionalni stil programiranja sam realizirala upotrebom anonimnih funkcija na više mjesta radi preglednijeg programskog koda, kao i mogućnosti automatskog generisanja određenih linija. Najviše su korištene u Controller klasama, poput podešavanja validnog izbora datuma u EditReservationController i AddReservationController klasama. Streamove sam iskoristila na nekoliko mjesta za potrebe testiranja nekih funkcija iz DatabaseDAO klase.

2.4 Java kolekcije

U projektu su najviše korištene liste, prilikom čega su upotrijebljene metode iz klasa List i ArrayList poput contains, add, remove i size metoda. Za potrebe iteriranja su najviše korištene rasponske for petlje, dok su u DatabaseDAO klasi na nekim mjestima korišteni iteratori.

2.5 Objektno zasnovano programiranje

Nažalost u kodu prilikom kreiranja korisničkih klasa nisam vidjela potrebu za nasljeđivanjem, stoga taj postulat nije ispunjen u projektu. Možda se mogla napraviti Abstract klasa „Osoba“ pa da nju nasljeđuju klase User i Administrator, međutim odlučila sam da ipak to budu dvije klase. U svim model klasama su atributi privatni, a metode javne. Čitav projekat je smješten u paket ba.unsa.etf.rpr.projekat, dok su metode razvrstane po folderima za administratora, korisnika i za oboje. Korišten je vlastiti tip izuzetka „InvalidUsernameException“, zbog ograničenja da se u bazi ne smiju pojaviti 2 ista korisnička imena, te je taj izuetak obrađen u DatabaseDAO klasi. Za potrebe prepoznavanja potrebnih klasa sam koristila leksičku analizu zahtjeva.

2.6 Grafički korisnički interfejs

Grafički korisnički interfejs sam razvila kreirajući fxml fajlove koji se nalaze u resources folderu u projektu. Izgled samih prozora sam dizajnirala u Scene Builder-u. Korištene su razne kontrole i pane-ovi, poput: GridPane-a, BorderPane-a, VBox-a, HBox-a, ButtonBar-a, TextField-a, PasswordField-a, Label-a, Button-a, TableView-a, ListView-a i drugih. Dizajn prozora sam realizirala koristeći css fajlove.

2.7 TestFX i Unit testiranje

Testirala sam AdministratorController klasu koristeći TestFX. Prethodno sam je uključila među ostale biblioteke projekta koristeći Maven. Za potrebe testiranja su korištene razne metode koje nudi FXRobot klasa, poput clickOn, write i lookUp metoda, kao i assertions-a koje nudi TestFX.

Također je realizirano i Unit testiranje na DatabaseDAO klasi.

2.8 Datoteke

Iskorišteno je čitanje iz datoteke „restaurant.db.sql“ radi kreiranja defaultnog izgleda i popunjavanja baze podataka.

2.9 Izvještavanje

Za potrebe kreiranja izvještaja sam koristila Jaspersoft studio. U projektu administrator može da dobije 2 izvještaja: 1. za dobivanje popisa rezervacija na tadašnji datum i 2. za dobivanje popisa svih stavki menija.

2.10 Lokalizacija

U projektu su dostupni prijevodi na 3 jezika: Bosanski, Engleski i Njemački jezik

2.11 Maven

Korišten je Maven alat za automatsku izgradnju koda, gdje su uvedeni sljedeći dependencies: jasperrsoft, testfx i sqlite-jdbc.

3. Prilika za unaprjeđivanje

Prilikom rada ovog projekta, dosta sam poboljšala svoje znanje i vještine u Javi. Uočila sam da su se neke funkcionalnosti mogle dosta bolje uraditi, nakon toga DatabaseDAO klasu sam mogla da razdvojim u 2 klase, za potrebe admin panela i user panela, te bi mi tako projekat bio dosta bolje organizovan. Također, nisam se najbolje pridržavala SOLID principa, što je opet razlog za pomalo natrpan i ružno organiziran kod u nekim klasama.

4. Dodatak

PRISTUP ADMIN PANELU:

Username=admin

Password=admin