Univerzitet u Sarajevu

Elektrotehnički fakultet Sarajevo

Razvoj programskih rješenja

Izvještaj o projektu

***Brza pošta***

Student Predmetni nastavnik

Beganović Tarik Doc.dr Vedran Ljubović

1. Opis aplikacije

Aplikacija koja se opisuje u nastavku ovog rada predstavlja dosta ograničenu simulaciju aplikacija za brzu poštu. Osnovna ideja je bila razviti aplikaciju sa osnovnim funkctionalnostima.

Korisniku je isporučen administratorski nalog koji ima sve mogućnosti. Korisničko ime i lozinka se nalaze u dodatku na kraju ovog rada. Također, korisniku su isporučena dva kurirska naloga te jedan menadžerski. Adminski nalog, kao što je i rečeno ima sve mogućnosti, dok menadžerski ima skoro iste mogućnosti kao i adminski nalog, osim opcija dodavanja/ažuriranja/brisanja menadžera.

Kurirski nalog nema nikakvih opcija dodavanja/ažuriranja/brisanja, nego ima samo opciju mjesečnog i godišnjeg izvještaja koji se sastoji od pošiljki koje je taj kurir isporučio/dostavio. Kurir samo vidi svoje narudžbe i njihove detalje, dok admin i menadžer vide sve pošiljke koje se nalaze u bazi podataka.

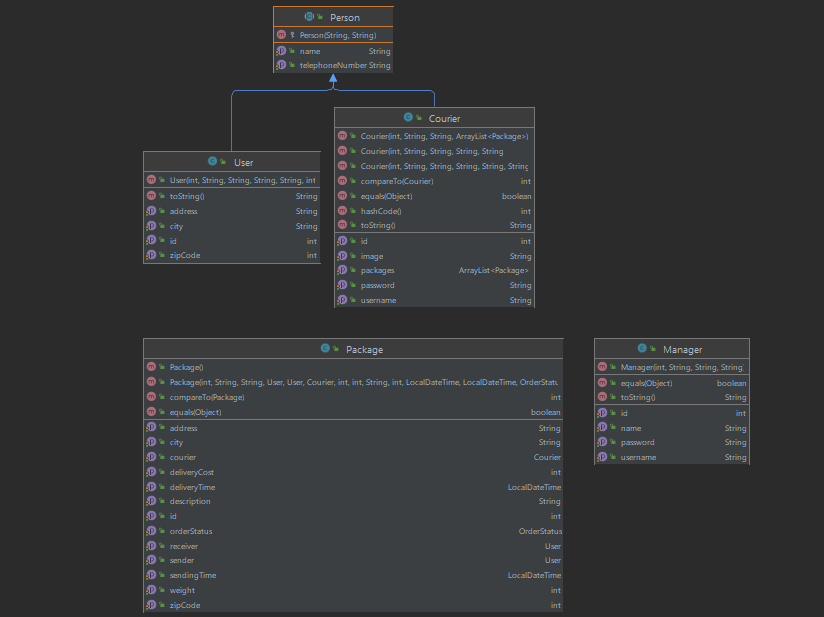
Menadžer je opisan svojim imenom, korisničkim imenom te lozinkom. Prilikom dodavanja/ažuriranja menadžera jedina ograničenja na spomenuta polja su ta da ne smiju biti prazna, te da korisničko ime mora biti unikatno(ne smije biti „admin“ i ne smije se desiti da postoji isto korisničko ime koje je u posjedu drugog menadžera ili kurira).

Kurir je opisan svojim imenom, brojem telefona, korisničkim imenom te lozinkom. Prilikom dodavanja/ažuriranja kurira jedina ograničenja na spomenuta polja su ta da ne smiju biti prazna, te da korisničko ime mora biti unikatno(ne smije biti „admin“ i ne smije se desiti da postoji isto korisničko ime koje je u posjedu drugog menadžera ili kurira). Prilikom brisanja kurira daje se na izbor kurir koji će preuzeti pošiljke obrisanog kurira(ako ih je ikako imao).

Pošiljka je opisana sa malo više atributa: svojim opisom, cijenom dostave, težinom paketa(kg), vremenom slanja i preuzimanja, statusom narudžbe, pošiljaocem i primaocem(oba su opisana sa imenom, brojem telefona, adresom, gradom i poštanskim brojem) i kurirom. Podaci pošiljaoca ne moraju biti poznati ukoliko ne želimo da se paket vrati pošiljaocu u slučaju neke greške. Težina paketa je cjelobrojni tip, te se zaokružuje na veći broj ukoliko stvarna težina paketa nije cjelobrojna.

U vrhu aplikacije imamo dva meni-a: „Fajl“ i „Pomoć“. Klikom na prvi dobijemo na izbor mogućnost prevođenja aplikacije na željeni jezik(engleski ili bosanski), možemo sačuvati trenutačno stanje baze podataka u datoteku koju želimo te možemo zatvoriti program. Klikom na drugi možemo doći do informacionog prozora, koji nam daje podatke o tehničkoj podršci, GitHub link itd.

U prilogu se nalazi class diagaram napravljen od strane IntellJ okruženja.



1. Implementacija

3.1) Osnovni koncepti

U projektu se koriste model klase koje prate JavaBeans specifkaciju. Zastupljena je i upotreba osnovnih Java kolekcija, kolekcija ArrayList te TreeSet koje su bile dovoljne za potrebe ovog projekta.

3.2)Koncepti OOP-a

Koncepte OOP-a možemo vidjeti u klasama Person, User, Courier itd. User i Courier su izvedeni iz klase Person. Imamo enkapsulaciju polja u svim klasama kako bi se odvojio javni od privatnog dijela.

3.3) Maven

Korišten je Maven alat za automatsku izgradnju koda. Uveli smo dependencies za bazu podataka, jasper i JavaFx testove. Ovime smo postigli i da imamo izvršni .jar fie u target folderu (untitled-1.0- SNAPSHOT-jar-with-dependencies.jar) koji se može pokrenuti iz komandne linije. Iz nekog razloga izvještaje traži u untitled-1.0- SNAPSHOT-jar-with-dependencies.jar! datoteci koja ne postoji, međutim kad se otvori kao .zip vidi se da je sve tu.

3.4)Izvještaj

Korišten je Jaspersoft Studio alata za kreiranje izvještaja. U aplikaciji imamo 3 vrste izvještaja: „mainReport“ te „courierReport“ koji ima dva oblika: godišnji i mjesečni. „Glavni“ report mogu dobiti samo admin i menadžeri dok kurirski report svaki kurir može dobiti sam za sebe.

3.5)Funkcionalno programiranje

U slučaju mrežnog programiranja, korištene su lambda funkcije u slučaju listenera u kontrolerima i na drugim mjestima.

3.6)Mrežno programiranje

Mrežno programiranje je iskorišteno tako što na „about.fxml“ formi imamo GitHub link za profil autora aplikacije

3.7)Niti

Niti su korištene prilikom pretrage slika za kurira na računaru korisnika.

3.8)Enum

U projektu se koristi jedan enum koji se naziva OrderStatus, kojem smo preklopili toString() metodu za lakše zapisivanje u bazu podataka

3.9) Baza podataka

Za bazu podataka smo koristili SQLite. Napravljen je dump, koji se koristi u testovima, da bi svaki put vratio bazu na izvorno stanje. Baza ima 5 tabela: User, Courier, Package, Manager te Register. Svaka od ovih baza čuva podatke o istoimenoj klasi, osim Register koja služi za praćenje veze između pošiljki i kurira.

3.10)Internacionalizacija

U programu smo dali korisniku na izbor između 2 jezika: bosanski i engleski. Standardni jezik pri pokretanju je engleski.

3.11)Grafički interfejs

Uzeti su u obzir koncepti dobrog dizajna korisničkog interfejsa (Geštalt principi) gdje su svi logički vezani elementi grupisani i na ekranu. Za prikaz validnih/nevalidnih polja pozivamo se na appearance.css file i u kontroleru se ovisno o validnosti postavlja odgovarajuća boja TextField-a. Prilikom implementiranja grafičkog interfejsa korišteni su razni GUI elementi kao što su : Button, TextField, Label, RadioButton, MeniBar, ButtonBar, BorderPane, GridPane, HBox, VBox itd. Funkcije istih se nalaze u kontrolerima.

3.12)Datoteke

Moguće je spremiti trenutačno stanje baze u datoteku po želji.

4)Dodatak

Administratorski nalog

username:admin

password:admin

Kuriri:

Courier 1

username:courier1

password:courier1

Courier 2

username:courier2

password:courier2

Menadžeri:

Managaer 1

username:manager1

password:manager1

Manager 2

username:manager2

password:manager2