

Elektrotehnički fakultet Sarajevo
Razvoj programskih rješenja

Izvještaj o projektu

Upravitelj - upravljanje zgradama

Student: Haris Haskić

Predmetni nastavnik: Doc.dr Vedran Ljubović

Sarajevo, 20.20.2020.

Sadržaj

1. Opis	
aplikacije.....	
3	
2. Osovne ideje pri	
implementaciji.....	6
3. Implementacija.....	
.....7	
3.1 Osnovni	
koncepti.....	7
3.2 Baza podataka	
.....	7
3.3 Grafički interfejs	
.....	7
3.4	
Enum.....	
...8	
3.5 Funkcionalno programiranje	
.....	8

3.6

Izvještaji.....

8

4. Prilike za unaprijeđivanje.....9

5. Dodatak.....10

1. *Opis aplikacije*

Aplikacija koju ću opisati u nastavku svoga rada predstavlja simulaciju aplikacije za upravljanje zgradama, neki dijelovi su izbačeni zbog kompleksnosti teme(finansijske transakcije). Inspiracija za ovaj projekat mi je došla nakon dugog provedenog vremena življenja u zgradi te zanimanja kako taj proces uistinu funkcionise, te koje su njegove mane i vrline. Ideja mi je bila razviti sličnu aplikaciju za real-time upotrebu , tj. razviti aplikaciju koju može upravitelj da koristi. Naravno , htio sam razviti sistem sa osnovim mogućnostima kao što je, registracija , dodavanje zgrada , općina itd. Ovu aplikaciju sam razvio

uz pomoć intervjua sa firmom koja se bavi upraviteljstvom , da bih bolje dobio osjećaj kako to funkcionira. Navest ću vam neke osobine koja moja aplikacija posjeduje.

-Korisnik dobija Administratorski nalog kojim se uvodi u svijet aplikacije za upraviteljstvo. Korisničko ime i password ćete dobiti u nastavku ovoga rada. Želio bih napomenuti da je ovaj program razvijen tako da podržava 3 vrste korisnika. Prvi i glavni je administrator ,user te guest. Koji će biti detaljno objašnjeni u nastavku. Objasnit ću prvi tip korisnika tj. administratora. Prije prijave administratora u aplikaciju, provodi se validacija samih podataka koji su uneseni, naravno pošto je korisnik prijavljen kao administrator imat će sve privilegije koje mu sistem dozvoljava, te ako je sve uspješno validirano, otvara mu se mogućnost da bira između sljedećih funkcionalnosti. Administratoru se daje na uvid 4 buttona od kojih svaki od njih vodi do različitih funkcionalnosti. Ako kliknemo na button „Municipality“ ulazimo u novi prozor gdje vidimo listu svih općina , te koliko zgrada ta općina ima. Također sa desne strane možemo vidjeti da svaku općinu možemo promijeniti tj. izvršiti njen update, jer možda više nije zadužen čovjek koji je ranije bio za tu općinu. Kada kliknemo na button „Add Municipality“ ,možemo dodati općinu te postaviti referenta(usera) za tu istu općinu. Ova aplikacija ne bi bila potpuna bez brisanja općine (ako smo sve zgrade u toj općini izgubili ili predali drugom upravitelju, pa takvu općinu nema smisla držati više u sistemu). Pri mijenjanju tj. update-u općina nam dolazi mogućnost da stavimo više od jednog zaduženog čovjeka – referenta za tu općinu, te također daje nam mogućnost i da ga izbrišemo sa liste zaduženih ljudi za tu općinu. Kod buttona „Buildings“ , sistem zadržava iste mogućnosti kao i za općine, tj. dodavanje, mijenjanje te brisanje. Također pri samom dodavanju, sistem mora verifikovati adresu koja se unosi, u slučaju da adresa postoji u sistemu izbacit će labelu, koja korisnika upozorava da adresa već postoji u bazi podataka i da se treba unijeti nova adresa. Prilikom dodavanja novih zgrada , nalaze se 3 radio-buttona , u

zavisnosti od radio-buttona koji se klikne, daje nam mogućnost unosa da li je zgrada novija ili starija te od te informacije nam se otvara polje da li ta zgrada posjeduje garažu. Ta mogućnost nije podržana kod starijih zgrada jer u tom vremenu nije bilo potrebe praviti podzemne garaže ili garaže uz zgrade (konkretno austro-ugarska gradnja). Na kraju dolazi nam padajući meni gdje trebamo da izaberemo gdje ćemo tu zgradu pozicionirati tj. u koju općinu ćemo je smjestiti prilikom dodavanja. Dolazimo do 3 button-a , to je button „Users“. Klikom na njegu dolazimo u novi prozor gdje vidimo spisak svih korisnika te njihove podatke. Sistem zadržava ranije spomenute funkcionalnosti tj. dodavanje ,mijenjanje te brisanje korisnika sa sistema (u slučaju otkaza i slično). Klikom na dodavanje usera ulazimo u novi prozor gdje unosimo podatke u novom korisniku te stavljamo jednu od tri mogućnosti radio buttona , admin , user ili guest. Bitno je napomenuti ako smo prijavljeni kao user ili guest i želimo da update-ujem profil , radio button-i postaju disable-ovani tj. nijedan tip korisnika sem administratora ne mogu mijenjati svoj tip , to pravo ekskluzivno zadržava administrator programa. Dolazimo do buttona koji glasi „Jobs“ . Gdje nam se otvara prozor i sistem nam onda daje mogućnost preko combo boxa da izaberemo zgradu kojoj želimo da dodijelimo posao, naravno referenti mogu vidjeti samo zgrade za koje su zaduženi. Također na uvid nam se daje i mogućnost pregleda poslova za datu ulicu, a sve u istom prozoru. Kada odaberemo ulicu ulazimo u drugi prozor gdje nam se traže detaljne informacije o poslu koji se treba obaviti. Treba posebno obratiti pažnju na datum početka i datum kraja posla da budu napisani u ispravnom formatu. Prilikom dodavanja posla , kada je korisnik tipa guest on vidi samo svoju zgradu i može da otvori nalog da se nešto na zgradi radi, bez kraja datuma , to će u njegovom slučaju biti onemogućeno. Također i na tom mjestu se vrši validacija. Da li je datum ispravno napisan te da li je datum početka posla počinje ranije od datuma kraja posla, te na kraju stavljamo ime firme tj. kooperanta koji će obaviti posao(to vrijedi

samo za administratora i usera). Istakao bih još par stvari , za svaki od button-a sa glavnog menija postoji između ostalog i button „Details“ . Pritiskom na taj button sistem kreira izvještaj. Na svakom od prozora implementirane su button-i „Exit“ i „Back“. Exit terminira program dok pritiskom na dugme back, program se vraća u prijašnji prozor(spriječeno je gomilanje prozora, tako da uvijek postoji jedan aktivni prozor). Kao što sam spomenuo ranije, postoje 3 različita tipa usera koji prisupa programu. Admin ima sve privilegije dok ostala dva tipa nemaju. Pa krenimo od Usera. User kada se log in-a može da vidi zgrade za koju je općinu zadužen i može da doda posao i također može da update-uje svoj profil. Gost može samo da vidi svoje zgrade, jer on predstavlja predstavnika etažnih vlasnika , i da doda nalog za neki posao na svoju zgradu te također da izvrši update profila. Poslije toga user koji je zadužen za taj region tj. tu općinu odobrava posao i stavlja kada je taj posao završen.

2. *Osnovne ideje pri implementaciji*

Pri implementaciji je korišten SQLite baza podataka jer je ona definitivno zadovoljavajuća i dovoljna za ovaj sistem. Ideja je bila da se

razvije i stvori funkcionalnost za rad sa bazom podataka. U sistemu se koristi klasa „BuildingManagmentDAO“ , koja komunicira sa bazom. U ovoj klasi su obrađene razne komplicirane metode kako bi podaci koji dolaze iz baze bili obrađeni te ih ustupaju aplikaciji na daljnje korištenje. Šta je ustvari postignuto time. Time je postignuto potpuno odvajanje baze podataka od ostatka programa tj. bussines logike, što nam daje mogućnost za nadogradnju ili potpunu promjenu baze podataka bez da se ostatak koda ili sistema mijenja u potpunosti. Htio sam također postići da izgled aplikacije bude sto jednostavniji i intuitivniji za korištenje, te sam više dao prioritet funkcionalnosti nego samom izgledu. Jezik aplikacije je, kao što je zahtijevano, engleski. Kako bih izbjegao silno dupliciranje koda , u više prilika su bile iskorištene iste klase i prozori, te sam time htio postići da aplikacija bude manje kompleksnija. Tako se npr. „registracija korisnika“, koja se može uraditi pri pokretanju aplikacije , nudi mogućnost registrovanja novih korisnika, koja se razlikuje po tipu. Da li je to admin , user ili guest, a razlike u njima su specificirane korištenjem enuma. Te sam istu mogućnost „registrovanja korisnika “ iskoristio pri samom update-u korisnika, gdje pritiskom na dugme update korisnika se otvara isti prozor, gdje se može mijenjati njegov sadržaj. Kako bi se napravila veza između samih objekata koji se nalaze u bazi i onih objekata koji se nalaze u aplikaciji, kreirane su klase koje u potpunosti prate JavaBeans specifikaciju. Želio bih da napomenem da su kreirani i izvještaji koji koriste JasperSoft studio. Također uz ovaj rad će biti priložena slika koja predstavlja class dijagram projekata.

3. Implementacija

U nastavku će biti opisani izabrani dijelovi implementacije kako bi se objasnila zastupljenost obrađenog gradiva na predmetu Razvoj programskih rješenja.

3.1 Osnovni koncepti

U projektu se koriste model klase koje prate u potpunosti JavaBeans specifikaciju. Zastupljena je i upotreba osnovnih Java kolekcija te kolekcije ArrayList() koje su bile dovoljne za izgradnju ovog projekta.

3.2 Baza podataka

U ovom projektu je korištena SQLite baza podataka. Za potrebe ovog projekta kreirano je 7 tabela. Uz ovaj rad također ćete dobiti i ER dijagram koji će biti generisan za bolju vizualizaciju.

3.3 Grafički interfejs

Prilikom implementiranja grafičkog interfejsa koristio sam razne GUI elemente kao što su : GridPane, Button, Label, TextField, TextArea, PasswordField, RadioButton, ListView, ComboBox i sl. Funkcije GUI elemenata nadgledaju i pare rad , kontrolori(tj. kontroleri). U kontolerima su postavljeni liseneri preko kojih se konstantno validira raznorazni unos podataka.Uzeti su u obzir koncpetu dobrog dizajna korisničkog interfejsa (Gestalt principi) gdje su svi logički vezani elementi grupisani i na ekranu. Za prikaz validnih/nevalidnih polja pozivamo se na .css file i u kontroleru , u ovisnosti o validnosti, postavljamo odgovarajuću boju.

3.4 Enum

Enumi se koriste za odabir između admina, usera i guesta.

3.5 Funkcionalno programiranje

Korištene su lambda funkcije u slučaju listenera i u kontrolerila .

3.6 Izvještaji

Izvještaji su implementirani korištenjem JasperSoft Studio alata. U aplikaciji se prilikom pritiska na dugme „Details“ otvara izvještaj u ovisnosti u kojem je prozoru klinuto dugme „Details“. Recimo ako se nalazimo u meniju za Usera , ako kliknemo na dugme „Details“ izbacuje nam izvještaj baš za Usera. Također implementirani su i drugi izvještaji koje prate glavne buttone.

4. Prilike za unaprjeđivanje

Uočeno je mnogo mnogo mogućnosti za poboljšanja. Pošto je poznato da program evoluirao kroz svoj život, istaknuo bih neke stvari. Ovaj sistem , da bi bio upotrebljiv , potrebno je bude velikih razmjera tj. da ima već upisanu ogromnu bazu ljudi, zgrada itd. Također za te potrebe ne bi trebao biti lokaliziran samo na jednom računar. Pored toga, pošto se nalaze podaci koje su osjetljive prirode, treba uraditi totalnu zaštitu tih podataka. Kako izlazimo iz opsega kursa, bitno je ipak napomenuti i ovu mogućnost. Uočena je još jedna prilika za unaprjeđivanje. Potrebno bi bilo uraditi i finansijske tokove. Tipa plaćanje faktura po završetku posla, skidanje novaca sa ulaza za realizaciju posla itd. Samim tim bi poboljšali upotrebljivost sistema te

bi ga uveliko kompletirali u formu finalnog proizvoda koji bi se mogao koristiti kao real-time proizvod.

5. Dodatak

Administratorski nalog

Korisničko ime: h1

Lozinka: h1