UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET SARAJEVO
RAZVOJ SOFTVERA

Aplikacija za evidenciju radnog vremena

IZVJEŠTAJ O PROJEKTU

Student:
Haris Kičin

Predmetni nastavnik:
Doc. dr Vedran Ljubović

Sarajevo, septembar 2019.

Sadržaj

1. Opis aplikacije

Aplikacija koja se opisuje u nastavku ovog rada predstavlja aplikaciju koja rješava problem evidencije radnog vremena.
Osnovna ideja je bila razviti platformu sa osnovnim funkcionalnostima, kao što je generisanje izvještaja radnog vremena, evidenciju radnog vremena po broju sati provedenih na poslu, te po broju sati provedenih na projektu.
Razgovor sa korisnikom je pomogao pri specificiranju zahtjeva i razrađivanju funkcionalnosti aplikacije.
Aplikacija koja se prezentira ima sljedeće osobine: Korisniku je isporučen administratorski nalog kojim započinje život aplikacije. Korisničko ime i lozinka se nalaze u dodatku na kraju ovog rada. Svaki radnik u firmi također ima svoje korisničko ime i lozinku.

Prijava administratora

Kada se administrator prijavi sa svojim pristupnim podacima, on ima mogućnost da registruje novog radnika tako što unosi validne podatke o imenu, prezimenu, adresi, poštanskom broju i gradu prebivališta, te poziciji, korisničkom imenu i lozinki. Ukoliko su podaci ispravni, kreiran je račun tog uposlenika, te se ubuduće on može prijavljivati sa tim podacima kako bi evidentirao svoje radno vrijeme i vrijeme provedeno na projektima. Također, administrator pored registracije novog radnika i brisanja postojećeg, može dodavati nove projekte, brisati postojuće te tražiti generisanje izvještaja svih radnika kako bi dobio uvid u radno vrijeme. Također je omogućeno generisanje izvještaja po projektima. Administrator nema pravo mijenjati evidentirano.

Prijava uposlenika

Kada se uposlenik prijavi sa svojim pristupnim podacima, treba izabrati da li evidentira vrijeme dolaska na posao ili odlaska, vrijeme na izabranom projektu. Izvještaj koji je dostupan radniku jeste izvještaj o njegovom radnom vremenu, te on nema pravo mijenjati evidentirano.

Uz projekat je priložen i class dijagram (resources.class-diagram)

2. Osnovna ideja pri implementaciji

Pri implementaciji aplikacije korištena je SQLite baza podataka jer je zadovoljavajuća za projekat ovih razmjera. Ideja projekta je bila da se razviju neke osnovne funkcionalnosti za rad sa bazom podataka.

Izgled aplikacije je jednostavan, lakho razumljiv, s obzirom da je prioritet bio demonstracija funkcionalnosti, ali je konzistentan kroz cijelu aplikaciju. Jezik aplikacije je engleski, po zahtjevu korisnika. U aplikaciji su primijenjeni sljedeći principi dobrog dizajna:
Gestalt principi - objekti koji su međusobno bliski i imaju slične osobine predstavljaju jednu grupu.
Opcije na ekranu su grupisane u skupine 4-7 elemenata. Svi elementi aplikacije su potpuno upotrebljivi koristeći tastaturu (npr Alt + X), a ne samo miš! Forme nisu resizable i dimenzije prozora su minimalne koje mogu prikazati sve kontrole.
Rubovi forme su odmaknuti od formulara. Korišteni su indikatori validnosti pomoću označavanja polja bojom - zelena boja označava ispravan unos, a crvena neispravnu vrijednost.

Kako bi se napravila veza između objekata u bazi i aplikacije kreirane su model klase koje prate JavaBeans specifikaciju.

U čitavom projektu su korištene "najbolje prakse" imenovanja klasa, metoda, atributa itd.
Izvještaji su kreirani koristeći Jaspersoft studio.

3. Implementacija

U nastavku slijede opisi izabranih dijelova implementacije kako bi se objasnila zastupljenost obrađenog gradiva na predmetu Razvoj softvera.

3.1 Osnovni koncepti

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

3.2 Organizacija koda

Projekat je rađen u IntelliJ razvojnom okruženju. Direktorij resources sadrži sve resurse koji su se koristili kroz projekat, što uključuje direktorije css, fxml, img i reports, gdje se respektivno nalazi css kod za validaciju formi, fxml kodovi za odgovarajuće panele, slike koje su se koristile, te kod za izvještaje generisan iz JasperSoft studio okruženja U source direktoriju imamo sve java kodove, grupisane po funkcionalnostima.

Sample.utilities direktorij implementira buissnes logiku projekta, tu se nalazi interfejs WorkTimeTrackerDAO koji sadrži osnovni kod za rad sa bazom, te WorkTimeTrackerSQLLiteDAO koji implementira taj interfejs i sadrži dodatne metode. Kasnije se kroz rad na projektu ukazala potreba za boljom preglednošću koda, te su kreirane klase ProjectDAO, UserDAO, WorkHoursDAO koje implementiraju rad sa bazom vezan za te tabele poput dodavanja i brisanja iz baze. S obzirom da je ProjectWorkHours samo specjalizacija klase WorkHours, ona je nasljeđena iz klase WorkHours te njena logika za rad sa bazom je također u klasi WorkHoursDAO.

Direktorij sample.models sadrži modele klase koje prate JavaBeans specifikaciju, te StopWatch klasu koja je bila potrebna za implementiranje štoperice koja prati radno vrijeme za određeni projekat.

Direktorij sample.controllers sadrži kontrolere za odgovarajuće panele.

Sample.enums i sample.exceptions sadrže enume i korisnički kreirane izuzetke.

3.3 Baza podataka

Kao što je već navedeno, korištena je SQLite baza podataka. Za potrebe projekta kreirano je 6 tabela. Tabele user, position, project su dovoljno jasne da ne zahtijevaju dodatno objašnjavanje, osim da se navede da su to tabele koje čuvaju podatke o registrovanim korisnicima aplikacije (uključujući admina), projektima i pozicijama. Tabela work_hours čuva podatke o radnom vremenu uposlenika. Podaci o radnom vremenu (atribut work_hours u istoimenoj tabeli) su u sekundama. Razlog tome jeste jednostavnost upotrebe. U izvještajima je u obliku 'hh:mm:ss' i za sve dalje potrebe se lahko može doći do sati, minuta i sekundi. Tabela project_work_hours

čuva podatke o vremenu provedenom na određenim projektima (također u sekundama) za sve usere. Tabela employee_has_position predstavlja vezu između tabela user i position.

Rad sa bazom počinje sa interfejsom sample.utilities.WorkTimeTrackerDAO koji propisuje osnovne metode za rad sa bazom. Klasa WorkTimeTrackerSQLiteDao implementira metode iz tog interfejsa pored ostalih za kojima se potreba pokazala u toku rada aplikacije. U ovoj klasi se uspostavlja konekcija sa bazom i vrši se kreiranje tabela, čitanjem iz db.sql file-a.

3.4 Koncepti objektno orjenitranog programiranja

Koncepti OOP (skrivanje informacija, nasljeđivanje, polimorfizam, enkapsulacija) vide se direktno u model klasama. Klasa ProjectWorkHours nasljeđuje klasu WorkHours.

Korisniku je dat samo interface klasa, šta se sa primjercima klase može raditi i koje poruke im se mogu slati. Podaci su objedinjeni sa postupcima koji se vrše nad njima, tj. umjesto "golih" podataka (javnih atributa) nude se metode inspektori i mutatori (geteri i seteri) što ispunjava koncept enkapsulacije, te polimorfizam koji se javlja u metodi getUserByUsername.

3.5 Grafički interfejs

Prilikom implementiranja grafičkog interfejsa korišteni su razni GUI elementi kao što su: AnchorPane, GridPane, Button, Label, TextField, TextArea, PasswordField, ToggleButton, TableView, ProgressIndicator i sl. Funkcije istih nadgledaju kontroleri iz sample.controllers. Kontroleri imaju listenere preko kojih se konstantno validira unos, tamo gdje je potrebno. Uzeti su u obzir koncepti dobrog dizajna korisničkog interfejsa, kao što je navedeno ranije.

3.6 Datoteke

Prilikom kreiranja projekta, informacije o istom se zapisuju u .txt datoteku kako bi se arhivirali podaci o svim ikada aktivnim projektima. To radi metoda writeIntoTextFIle klase sample.model.Project.

3.7 Enumi i interfejs

U projektu se koristi jedan interfejs WorkTimeTrackerDAO čija uloga je objašnjena ranije.

Enumi se koriste za odabir help ili about prozora (sample.enums.ContentsType) te odabira između četiri vrste izvještaja (sample.enums.ReportType). Na osnovu enuma ReportType klasa PrintReport zna koju putanju da koristi da bi prikazala izvještaj.

3.8 Mrežno programiranje i trendovi

Mrežno programiranje se koristi prilikom validacije poštanskog broja osobe. Ovo se radi u metodi okAction u klasi sample.controllers.AddEmployeeController koja koristi url da validira poštanski broj. U novom thread-u se izvrši validacija poštanskog broja, a zatim izvršava funkcija nad TextField objektom u ovisnosti od validnosti koja prikazuje boje za obradu, validno ili nevalidno.

3.9 Funkcionalno i generičko programiranje

U slučaju funkcionalnog programiranja, korištene su lambda funkcije kao i u slučaju listenera u kontrolerima za validaciju polja. Generičko programiranje nije iskorišteno iz razloga jer nije bilo logičke niti funkcionalne potrebe da neka funkcija prima bilo koji tip ili da klasa bude generička.

3.10 Vlastiti tipovi izuzetaka

Sample.exceptions sadrži vlastite tipove izuzetka: InvalidCredentialException i PersonDoesNotExistException, što uveliko olakšava da se otkrije i prikaže korisniku koja se greška dogodila prilikom logina (izuzeci se hvataju u LoginController klasi).

3.11 Izvještaji

Izvještaji su implementirani korištenjem Jaspersoft Studio alata za kreiranje izvještaja. U aplikaciji se prilikom pritiska na dugme Work hours report u admin panelu otvara izvještaj svih radnika sa njihovim radnim vremenima po datumima, a pritiskom na dugme Project work hours report se otvara izvještaj o satima provedenim na aktivnim projektima. Analogno se otvara i u employee panelu samo što radnik može dobiti uvid u svoje radne sate, ne od drugih uposlenika. Niko od navedenih nema pravo mijenjati izvještaj.

Klasa PrintReport na osnovu datog enuma bira između četiri putanje i prikazuje izvještaje (iz resources.reports direktorija).

3.12 Testiranje

Test resources direktorij sadrži testove koji vrše provjere da li određene cjeline rade ono što bi trebale. Unit test poziva cjelinu (kreira objekat, poziva metodu...) a zatim provjerava da li izlaz odgovara očekivanom. Također je korišten TestFX za testiranje JavaFX-a. Testiranje je vršeno u skladu sa klasičnim pogledom na razvoj softvera. Testovi su pisani nakon implementacije.

Dva testa su zakomentarisana iz razloga što u razvojnom okruženju u kojem je projekat rađen, prilikom otvaranja admin panela ili employee panela, samo prvi put prilikom istog pokretanja se ne prikaže sadržaj na prozoru, već drugi put, što se nije dešavalo kada sam testirao u drugom razvojnom okruženju (Eclipse).

4. Zaključak

Prilikom rada na ovom projektu otvorile su se mnoge mogućnosti za napredak aplikacije i potreba za više znanja i iskustva. Aplikacija je odličan kostur za dalji razvoj i ima odlične funkcionalnosti koje su potrebne korisniku.

Bilo je različitih poteškoća prilikom izrade projekta, što je naravno samo stepenica za dodatni napredak i širenje znanja.

5. Dodatak - pristupni podaci

Administratorski nalog:

Username: admin
Password: password

Pristupni podaci za jednog korisnika:

Username: akicin1
Password: pass123