

Elektrotehnički fakultet sarajevo

Razvoj programskih rješenja

**Izvještaj o projektu**

**APLIKACIJA ZA UPRAVLJANJE PERSONALNIH TASKOVA**

Student: Amna Trčalo

Predmetni nastavnik: doc.dr Vedran Ljubović

Sarajevo, 25.9.2021.

## OPIS APLIKACIJE

Aplikacija koja se opisuje u nastavku ovog rada je napravljena s ciljem pomoći korisniku u organizaciji njegovog dana i lakšeg razvrstavanja zadataka. Usljed ubrzanog načina života, ova aplikacija može pomoći u zapisivanju podsjetnika, ispunjavanju svih obaveza i napominjanja korisnika o istim.

Neregistrovani korisnik se prije svega treba registrovati, pri čemu je potrebno unijeti lične podatke: ime i prezime, korisničko ime (treba biti jedinstveno), e-mail adresu (potrebna za slanje notifikacija putem e-maila), lozinku, te potvrditi istu. Obzirom da se radi o jednostavnoj aplikaciji, ograničenja na lozinku su minimalna, potrebno je samo da sadrži više od bilo kakva tri karaktera. Ukoliko su svi podaci ispravno uneseni, korisnički račun je kreiran, te se korisniku pojavljuje poruka da sve detaljnije informacije o aplikaciji može pronaći u meniju „Pomoć“.

Prilikom svake sljedeće prijave na aplikaciju, potrebno je samo unijeti korisničko ime i šifru. Ovi podaci se naknadno mogu mijenjati.

Rad u aplikaciji je osmišljen da bude jednostavan, ali i da nudi korisniku fleksibilnost u kreiranju zadataka i njihovom razvrstavanju.

Za razvrstavanje zadataka služe liste.

Postoje četiri defaultne liste koje ne mogu biti obrisane: „Moj dan“, „Planirano“, „Zadaci“ i „Dovršeno“. Pri tome, korisnik može kreirati i brisati vlastite liste, bez ikakvog ograničenja. Prilikom kreiranja zadatka moguće je smjestiti zadatak u jednu od korisnikovih vlastitih listi, u suprotnom zadatak će biti smješten u jednu od defaultnih listi prema sljedećim kriterijima:

- Ukoliko su datum i vrijeme zadatka postavljeni, zadatak će biti spremljen u listu „Planirano“
- Ukoliko datum i vrijeme zadatka nisu postavljeni, zadatak će biti spremljen u listu „Zadaci“

Zadatak je moguće označiti kao dovršen označavanjem checkbox-a pored njegovog imena, pri čemu zadatak neće biti obrisani, ali će biti premješten u listu „Dovršeno“, te korisnik od njega neće primiti nikakva obavještenja. Ovo omogućava korisniku ponovno označavanje dovršenog zadatka kao nedovršenog označavanjem istog checkbox-a pored njegovog imena, pri čemu se on vraća u jednu od defaultnih listi - „Planirano“ ili „Zadaci“. Zadatak se trajno briše ručno, i njegovo brisanje nije moguće poništiti.

Svi zadaci koji počinju danas će se pojaviti u listi Moj dan.

Kreiranje zadatka je vrlo jednostavno i nudi niz pogodnosti: specificiranje datuma i vremena početka ili kraja zadatka, označavanje zadatka kao cjelodnevnog, dodavanje bilješki, dodavanje tekstualnog fajla, te dodavanje podsjetnika. Sve ove pogodnosti su opcionalne, jedino neophodno je specificiranje imena zadatka, pri čemu postoji ograničenje da korisnik ne može imati dva zadatka sa istim imenom.

Ukoliko korisnik odluči da želi postaviti datum i vrijeme zadatka, otvara se novi prozor u kojem upravlja datumom i vremenom, te postavlja podsjetnik.

Naravno, kako bi sve bilo regularno, postoje određene restrikcije:

- Ukoliko se zadatak označi kao cjelodnevni, korisnik može i mora unijeti samo datum početka zadatka

U nastavku teksta će biti objašnjene restrikcije koje se odnose na zadatke koji nisu cjelodnevni:

- Korisnik može vremenski odrediti samo početak zadatka (bez njegovog kraja), ali pri tome treba postaviti i datum i vrijeme početka
- Korisnik može postaviti datum i vrijeme kraja zadatka, ali pri tome trebaju biti postavljeni i datum i vrijeme njegovog početka
- Restrikcije na prethodne dvije stavke su da postavljeni datum i vrijeme početka zadatka ne mogu biti prije trenutnog datuma i vremena, te da datum i vrijeme kraja zadatka ne mogu biti prije datuma i vremena njegovog početka
- Korisnik može podesiti i vrijeme podsjetnika, pri čemu obavijest može dobiti određeni broj minuta, sati ili dana prije početka zadatka. Naravno, za postavljanje podsjetnika je neophodno da se postave i datum i vrijeme početka zadatka
- Korisnik može odabrati da li će podsjetnik primati notifikacijom ili e-mailom. Moguće je ugasiti sve obavijesti u postavkama.

U bilo kojem trenutku korisnik može urediti zadatak, izbrisati ga ili vidjeti njegove detalje.

Opcijom „Spasi“ (CTRL+S) se svi zadaci spašavaju u .txt file, zajedno sa svim popratnim informacijama.

# IMPLEMENTACIJA

## Softver inženjering

Razvoj ove aplikacije je praćen standardnim aktivnostima u razvoju softvera: specificiranje zahtjeva, analiza, dizajn, implementacija i testiranje.

### Osnovni koncepti

Sva pravila imenovanja promjenjivih, metoda, klasa i konstanti su ispoštovana. Za lakšu organizaciju koda su korišteni paketi.

Zastupljena je upotreba osnovnih Java kolekcija, kao npr. ArrayList() koja je bila dovoljna za potrebe ovog projekta. Osim klase Arrays, zastupljene su i još neke klase obrađene na predavanjima, kao npr. Date i Random.

Korišteni su streamovi i lambda funkcije tamo gdje je bilo moguće. Primjer korištenja je u initialize() metodi klase TaskController ili u stage.setOnHiding() metodi.

Također, korišteni su i enumi, a većina njih je značajno pomogla u prevodu aplikacije.

Testiranje je obavljeno TestFX testovima, pri čemu su testirane najvažnije klase i njihove metode. Klase koje su pokrivene testovima su: DateAndTimeController, LoginController, MyDayController, RegisterController i TaskController.

Kreiran je i vlastiti tip izuzetka FileNotFoundException koji se baca iz klase TaskController ukoliko dodajemo fajl koji sadrži više od 100 karaktera.

### Koncepti objektno orijentisanog programiranja

Ispoštovani su svi koncepti OOP-a (sakrivanje informacija, enkapsulacija, nasljeđivanje i polimorfizam).

Sakrivanje i enkapsulacija su ispoštovani u svakoj od klasa: svi atributi su privatni, a za njihovo čitanje i promjenu se koriste odgovarajući geteri i seteri.

Nasljeđivanje je zastupljeno prilikom kreiranja vlastitog tipa izuzetka (iz klase Exception).

Druge vrste nasljeđivanja u ovoj aplikaciji nisu bile potrebne.

Polimorfizam je zastupljen u metodi lists klase AppDAO.

Što se tiče opštih principa, navest ću neke od njih koji su u ovoj aplikaciji primjenjeni:

KISS (Keep It Simple, Stupid) – u aplikaciji je korišten najmanji mogući broj klasa i prozora.

Neki od prozora i klasa su višestruko upotrebljivani. Primjer je editovanje profila korisnika ili zadatka, pri čemu su iskorištene iste klase/prozori kao za registraciju korisnika i kreiranje zadatka respektivno;

SPT (Single Point of Truth) – zbog ponavljanja određenih podataka i tekstova na više mjesta, kreirane su enum klase u kojima su sadržani svi ti podaci i tekstovi radi lakšeg mijenjanja u budućnosti;

Single Responsibility Principle – kod dizajna klasa je vođeno računa da se drži pravila kohezije, odnosno da klasa opisuje jednu stvar sa svim potrebnim informacijama.

## **Grafički korisnički interfejs**

Korisnički interfejs je kreiran kroz FXML datoteke poštujući Gestalt princip.

Stiliziranje je obavljeno u CSS fajlovima.

Sve statičke datoteke (i fxml i css fajlovi i slike) su smješteni u Resources folderu.

Prilikom implementiranja grafičkog interfejsa korišteni su razni GUI elementi: AnchorPane, StackPane, GridPane, BorderPane, VBox, ListView, CheckListView, ImageView, Button, Label, TextField, TextArea, PasswordField, ChoiceBox, RadioButton, DatePicker, Spinner, CheckBox, Separator, ButtonBar, MenuBar...

Upravljanje svim elementima se vrši u kontroler klasama iz ba.unsa.etf.rpr.project.controller.

Izgled aplikacije je jednostavan i konzistentan. Alerti su napravljeni po uzoru na standardne alerte iz klase Alert, meni također sadrži standardne opcije koje su dovoljne za ovakav tip aplikacije, ikonice, te kratice (CTRL+S, CTRL+A, CTRL+E, CTRL+H).

Cijelu aplikaciju je moguće koristiti bez upotrebe miša korištenjem shortcut-a, te kretanjem kroz polja koristeći Tab tipku. Također, svaki defaultButton je označen kao takav, a još jedna pogodnost za korisnika je olakšano kreiranje zadatka pritiskom na tipku enter, te prikazivanje detalja o zadatku pritiskom tipke enter nakon selektovanja određenog zadatka.

Aplikacija je responsive - prozori sa formularima su fiksne dimenzije, dok se glavni prozor može rezizirati (pri tome je vođeno računa o minimalnim dimenzijama prozora).

Pomoć korisniku je prisutna na svakom prozoru u vidu tooltips-a. Također, na glavnom prozoru se nalazi Meni za pomoć kojem se može pristupiti shortcut-om CTRL+H.

Validacija unesenih podataka postoji gdje god je bila potrebna – prilikom registrovanja i prijave korisnika, te kreiranja zadatka.

## **Baza podataka**

Pri implementaciji korištena je SQLite baza podataka. Smještena je pod nazivom todoDatabase.db u folderu .todoApp u home folderu korisnika.

Za potrebe ove aplikacije kreirane su 4 tabele: tabela users koja čuva podatke o korisnicima, tabele tasks i lists koje čuvaju podatke o zadacima i listama za svakog korisnika respektivno, te klasa quotes u kojoj su smješteni citati koji se pojavljuju na dnu ekrana pri korištenju aplikacije.

Klasa koja se bavi učitavanjem podataka iz baze i uspostavljanjem konekcije je AppDAO klasa, čime se odvajaju baza podataka i business logika.

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

## **Datoteke**

U aplikaciji je korištena klasa FileChooser u dva slučaja – prilikom spašavanja zadataka (Save) i dodavanja bilješke. U prvom slučaju se koristi metoda showSaveDialog (primjer upotrebe u MyDayController-u), a u drugom showOpenDialog (primjer upotrebe u TaskController-u).

## **Internacionalizacija**

Jezik aplikacije je bosanski i engleski i moguće ga je mijenjati u postavkama. Nakon promjene neophodna je ponovna prijava na aplikaciju. Prevedene su sve labele, poruke, tekstovi i sl. Također je prilagođen i format datuma i vremena:

- bosanski jezik  
sati: 14:10  
datum: 25. sep 2021.
- engleski jezik  
sati: 2:10  
datum: Sep 25, 2021

## **Mrežno programiranje**

Primjer mrežnog programiranja je slanje mail-a korisniku. Za tu svrhu je napravljena posebna klasa MailClass.

(NAPOMENA: u klasi MailClass je potrebno navesti e-mail adresu i password sa koje se šalje mail korisniku. Ta dva unosa su trenutno označena sa xxxx).

## **Maven**

Korišten je Maven alat za automatsku izgradnju koda, čime je generisan izvršni .jar file u target folderu kojeg je moguće pokrenuti iz komadne linije.