**PROJEKTNI ZADATAK**

Student: Edis Kasumović br.Indexa: I-0425/21

Sadržaj

[1 ZADATAK: 3](#_Toc168662861)

[2 Izrada i predavanje projekta 3](#_Toc168662862)

[3 Tehnički zahtjevi 4](#_Toc168662863)

[4 Formalni zahtjevi 4](#_Toc168662864)

[5 SHEMA 5](#_Toc168662865)

[6 Izvjestaj 6](#_Toc168662866)

[6.1 Uvod 7](#_Toc168662867)

[6.2 Database 7](#_Toc168662868)

[6.3 Start 7](#_Toc168662869)

[6.4 Registracija 8](#_Toc168662870)

[6.5 Prijava 8](#_Toc168662871)

[6.6 Admin Panel 9](#_Toc168662872)

[6.7 Naručivanje hrane 9](#_Toc168662873)

[6.8 Kuhinja 10](#_Toc168662874)

**Projekat 1**

# ZADATAK:

Napisati program koji implementira funkcionalnosti za optimizaciju dostave hrane.

# Izrada i predavanje projekta

1. Projekat se radi samostalno.
2. Rok za predaju projekta je 07.01.2024. godine do 23:59h.
3. Package treba da se zove

package com.IMEPREZIMEipia.example;

Npr.  
package com.selenakurticipia.example;

1. Svi projekti moraju biti upakovani (zipovani) i imenovani kao PUJ\_2\_ime\_prezime,  
   (npr. PUJ\_2\_selena\_kurtic) te poslati na selena.kurtic@yahoo.com sa subjectom  
   PUJ\_2\_ime\_prezime do 07.01.2024. do 23:59h.
2. Svi projekati se izlažu od 08.01.2024. u 15:00.
3. Samo studenti koji imaju opravdani razlog (javili su na vrijeme ZASTO ne mogu izlagati) dobit ce termin kada ce moci izlagati svoj projekat.
4. Svi nepotpuni projekti, pogresno imenovani i poslati poslije navedenog vremena nece biti pregledani, uslov za izlazak na zavrsni ispit je predat Projekat 1 i Projekat 2.
5. Projekti MORAJU BITI URADJENI U IntelliJ sa GUI-om (nije dozvoljeno kreiranje GUIa putem code-a)!
6. Projekti koji budu radjeni u drugom programu ili od strane nekoga drugog dobit ce 0 (nulu) koju nece moci popraviti do kraja skolske godine 2023/2024.
7. Ukoliko student odluci ne raditi projekat, ne može izaci na završni ispit.
8. Svi projekti koji budu predani poslije prvog roka za predaju, nece imati mogucnost da osvoje maksimalan broj bodova.
9. Svi projekti koji budu vraceni na popravak, nece moci osvojiti maksimalan broj bodova.

# Tehnički zahtjevi

1. Potrebno je kreirati tri accounta osim admina (Korisnik, Kuhinja i Manadzer, te im ograniciti vidljivost).
2. Omogućiti operacije unosa, izmjene, pregleda i brisanja narudzbe.
3. Omogućiti pravljenje narudzbe.
4. Omogućiti pregled starih narudzbi.
5. Početno punjenje baze izvršiti samostalno vlastitim unosima za proizvoljnu narudzbu.
6. Unos novih stavki, te izmjenu postojećih vršiti preko grafičkog korisničkog sučelja.
7. Napraviti dokumentaciju projekta u obliku Word dokumenta. Koristiti ovaj dokument kao šablon i početi sa 6. poglavljem svoj izvještaj o projektu.

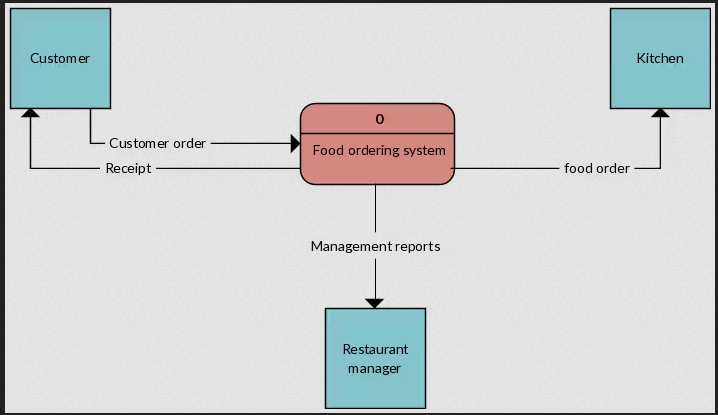
# Formalni zahtjevi

1. Programski kôd (*backend*) pisati u Java programskom jeziku (IntelliJ).
2. Grafičko korisničko sučelje (*frontend*) napraviti korištenjem IntelliJ GUI.
3. Kao bazu podataka koristiti ili obične tekstualne datoteke ili bazu podataka npr. Microsoft SQL Server sa JDBC driverom za konekciju[[1]](#footnote-1) i funkcijama za CRUD operacije.
4. Strogo se pridržavati principa objektno – orijentiranog programiranja.
5. Jasno definirati potrebne klase.
6. Koristiti naslijeđivanje gdje je potrebno.
7. Maksimalno ograničiti vidljivost i pristup članovima klase.
8. Implementirati sve konstruktore i destruktore.
9. Napraviti odgovarajuće verzije konstruktora ovisno o očekivanom unosu.
10. Smanjiti ponavljanje kôda kroz implementaciju metoda.
11. Paziti na zauzimanje resursa radne memorije.
12. Objekte, klase, članove klase i njene metode imenovati razumljivo.

Nastojati na jedinstven način imenovati metode koji obavljaju iste funkcije u više klasa.

# SHEMA

Imati na umu da je ovo samo idejni mockup te nije krajnje definiran projekat.



# Izvjestaj

## Uvod

Ovaj projekat predstavlja sistem za naručivanje hrane koji omogućava korisnicima da se prijave, pregledaju dostupnu hranu, dodaju je u korpu, obave narudžbinu, kao i administratorima da administriraju sistemom dodavanjem, brisanjem i ažuriranjem stavki hrane. Sistem je implementiran koristeći Java programski jezik, uz upotrebu Java Swing za grafički interfejs i MySQL bazu podataka za skladištenje podataka o korisnicima, hrani i narudžbinama. Projekat se sastoji od nekoliko ključnih komponenti:

1. Korisnički interfejs: Grafički interfejs omogućava korisnicima lako korišćenje sistema putem intuitivnog korisničkog interfejsa.
2. Baza podataka: MySQL baza podataka koristi se za skladištenje podataka o korisnicima, hrani i narudžbinama.
3. Korisničke funkcionalnosti: Korisnici mogu da se prijave, pregledaju dostupnu hranu, dodaju je u korpu, obave narudžbinu i pregledaju istoriju narudžbina.
4. Administratorske funkcionalnosti: Administratori mogu da dodaju, brišu i ažuriraju stavke hrane, kao i da pregledaju sve narudžbine.

Ovaj projekat omogućava korisnicima da lako naruče hranu putem intuitivnog korisničkog interfejsa, dok administratorima pruža alate za efikasno upravljanje sistemom.

## Database

Ovaj Java kod definiše klasu DataBase koja se koristi za upravljanje vezom sa MySQL bazom podataka i izvršavanje raznih operacija kao što su dodavanje korisnika, autentifikacija korisnika, dobavljanje svih namirnica, dobavljanje svih narudžbi, dodavanje, brisanje i ažuriranje namirnica, kao i čuvanje narudžbi. Metode klase koriste PreparedStatement za izvršavanje SQL upita i upravljanje podacima u bazi. Veza sa bazom se inicijalizuje metodom connect(), koja se poziva na početku svake metode, a zatvara se metodom close() na kraju. Metoda addUserToDatabase dodaje novog korisnika u bazu, getAuthenticatedUser vraća korisnika ako su podaci za prijavu tačni, getAllFood i getAllOrders vraćaju liste svih namirnica i narudžbi, dok addFood, deleteFood i updateFood dodaju, brišu i ažuriraju namirnice u bazi podataka. Metoda saveOrder čuva podatke o narudžbini u bazi. Korištenje ovih metoda omogućava interakciju sa bazom podataka za razne funkcionalnosti aplikacije.

## Start

Ovaj Java Swing kod definiše klasu Start koja predstavlja glavni prozor aplikacije sa dva dugmeta: "Login" i "Signup". Kada korisnik klikne na "Login", trenutni prozor se skriva, otvara se novi prozor za prijavu (instanca klase LogIn), i trenutni prozor se uništava. Slično, kada se klikne na "Signup", otvara se novi prozor za registraciju (instanca klase SignUp). Glavni prozor se postavlja sa panelom mainPanel, koji sadrži sve ostale komponente, poput labele za dobrodošlicu i dva dugmeta. Prozor se automatski prilagođava svom sadržaju, podešen je da se zatvori pri izlasku iz aplikacije, i prikazuje se korisniku na ekranu.

## Registracija

Ovaj Java Swing kod definiše klasu SignUp koja predstavlja dijalog prozor za registraciju korisnika. Prozor sadrži tekstualna polja za unos korisničkog imena, adrese, emaila i lozinke, kao i dva dugmeta: "Sign Up" i "Close". Kada se klikne na dugme "Sign Up", poziva se metoda registerUser koja validira unos korisničkih podataka. Ako su svi podaci validni, korisnik se registruje putem metode addUserToDatabase iz klase DataBase, koja vraća objekat korisnika ako je registracija uspešna. U slučaju uspešne registracije, dijalog se zatvara i otvara se novi prozor klase foodOrder. Ako unos nije validan ili registracija ne uspe, prikazuje se odgovarajuća poruka greške. Klik na dugme "Close" zatvara aplikaciju. Dijalog se prikazuje kao modalni prozor, što znači da blokira sve druge prozore dok se ne zatvori.

## Prijava

Ovaj Java Swing kod definiše klasu LogIn, koja predstavlja dijalog prozor za prijavu korisnika. Prozor sadrži tekstualno polje za unos emaila i lozinke, kao i dva dugmeta: "Login" i "Close". Kada korisnik klikne na dugme "Login", preuzima se unoseni email i lozinka, i poziva se metoda getAuthenticatedUser iz klase DataBase kako bi se proverili uneti podaci. Ako su email i lozinka "admin", otvara se admin panel (adminPanel). Ako su email i lozinka "kuhinja", otvara se kuhinjski panel (kitchenPanel). Ako uneti podaci odgovaraju nekom korisniku u bazi podataka, otvara se panel za naručivanje hrane (foodOrder). Ako podaci nisu validni, prikazuje se poruka o grešci. Dugme "Close" zatvara dijalog. Dijalog se prikazuje kao modalni prozor, što znači da blokira interakciju sa drugim prozorima dok se ne zatvori. Takođe, dijalog je postavljen da se zatvori kada korisnik izabere opciju za zatvaranje.

## Admin Panel

Ovaj Java Swing kod definiše klasu adminPanel, koja predstavlja dijalog prozor za administraciju hrane u bazi podataka. Prozor sadrži tekstualna polja za unos imena hrane i cene, tabelu za prikaz hrane i tri dugmeta: "Add Food", "Edit Food" i "Delete Food". Kada se klikne na dugme "Add Food", uneta hrana se dodaje u bazu podataka putem metode addFoodToDatabase, i osvežava se prikaz tabele. Dugme "Delete Food" briše selektovani red iz tabele i baze podataka putem metode deleteFoodFromDatabase. Klik na red u tabeli popunjava tekstualna polja sa podacima o selektovanoj hrani. Dugme "Edit Food" ažurira podatke o selektovanoj hrani u bazi podataka putem metode updateFoodInDatabase i osvežava tabelu. Metoda displayTable učitava sve podatke o hrani iz baze podataka i prikazuje ih u tabeli. Dijalog je modalni prozor, što znači da blokira interakciju sa drugim prozorima dok se ne zatvori. Takođe, dijalog je postavljen da se zatvori kada korisnik izabere opciju za zatvaranje.

## Naručivanje hrane

Ovaj Java Swing kod definiše klasu foodOrder, koja predstavlja dijalog prozor za naručivanje hrane. Prozor sadrži dve tabele: jednu za prikaz dostupne hrane (allFoodTable) i drugu za prikaz hrane u korpi (foodBasketTable), kao i dugmad za dodavanje hrane u korpu, uklanjanje hrane iz korpe i slanje narudžbine. Takođe, postoji labela za prikaz ukupne cene. Konstruktor klase inicijalizuje sve komponente, postavlja naslov prozora, veličinu, modalnost i operaciju zatvaranja. Metoda displayTable učitava sve dostupne hrane iz baze podataka i prikazuje ih u tabeli. Dugme "Add to Basket" dodaje selektovanu hranu iz tabele sa svim hranama u korpu i ažurira ukupnu cenu, dok dugme "Delete from Basket" uklanja selektovane stavke iz korpe i takođe ažurira ukupnu cenu. Dugme "Submit Order" proverava da li je korisnik autentifikovan, a zatim šalje narudžbinu tako što čuva podatke u bazi podataka za svakog korisnika, prikazujući poruku o uspešnoj narudžbini i resetujući tabelu korpe. Ukupna cena se izračunava i prikazuje u labeli svaki put kada se korpa ažurira.

## Kuhinja

Ovaj Java Swing kod definiše klasu kitchenPanel, koja predstavlja dijalog prozor za pregled narudžbina u kuhinji. Prozor sadrži tabelu (ordersTable) za prikaz narudžbina i dugme (updateOrdersTable) za osvežavanje tabele sa narudžbinama. Konstruktor klase inicijalizuje komponente, postavlja naslov prozora, veličinu, modalnost i operaciju zatvaranja, i povezuje dugme sa metodom updateOrdersTable koja se poziva kada korisnik klikne na dugme. Metoda updateOrdersTable preuzima sve narudžbine iz baze podataka putem metode getAllOrders iz klase DataBase, a zatim popunjava tabelu sa ovim podacima koristeći DefaultTableModel. Svaki red u tabeli predstavlja jednu narudžbinu sa kolonama za ime korisnika, naziv hrane, cenu hrane i vreme narudžbine. Na ovaj način, kuhinjsko osoblje može da osveži i vidi trenutne narudžbine.

1. <https://docs.microsoft.com/en-us/sql/connect/jdbc/step-3-proof-of-concept-connecting-to-sql-using-java?view=sql-server-2017> [↑](#footnote-ref-1)