Bankovni poslovno informacioni sistem

# Milan Rusovac-48/2022

1.Opis realnog Sistema:

- Razvijen je Poslovno Informacioni Sistem za banku koji integriše Microsoft Database Manager za upravljanje SQL bazom podataka. Sistem ima za cilj efikasno praćenje i upravljanje bankarskim operacijama, uključujući kreiranje i održavanje informacija o klijentima, kao i praćenje finansijskih transakcija.

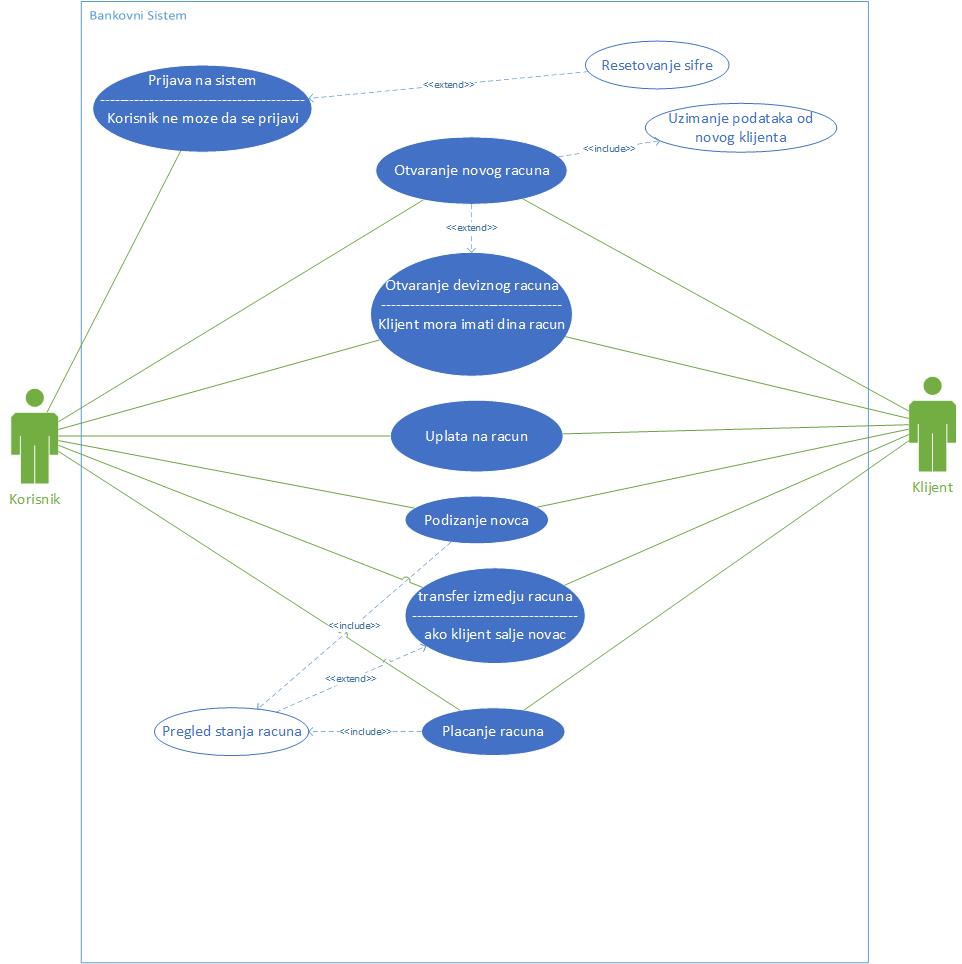
2.Zadatak:

- Glavni zadatak sistema je omogućiti bankarskim radnicima efikasno vođenje evidencije o klijentima, upravljanje njihovim računima i sprovodjenje transakcija. Implementacija Microsoft Database Managera je ključna za stvaranje i održavanje baze podataka koja podržava ove funkcionalnosti. WPF aplikacija treba pružiti intuitivan korisnički interfejs za dodavanje novih klijenata, izvršavanje transakcija i praćenje finansijskih podataka.

3.Opis tehnologija:

- U okviru ovog projekta, korišćene su Microsoft tehnologije, uključujući Microsoft Database Manager za upravljanje SQL bazom podataka. C# je korišćen kao programski jezik za razvoj WPF aplikacije, pružajući snažnu integraciju između korisničkog interfejsa i backend logike. Ova kombinacija tehnologija omogućava efikasno upravljanje podacima klijenata i realizaciju bankarskih funkcionalnosti.

4.UML:



**Slučaj upotrebe: Prijava na sistem**

**Kratak opis:** Radnik se prijavljuje na sistem kako bi pristupio svojim bankarskim funkcijama.

**Učesnici:** Radnik (korisnik)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** /

**Opis:** Radnik se prijavljuje na bankarski sistem putem svojih pristupnih podataka (korisničko ime i lozinka).

**Izuzeci:** Radnik je zaboravio lozinku i mora da koristi funkcionalnost za resetovanje lozinke.

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Radnik je uspešno prijavljen na sistem.

**Slučaj upotrebe: Resetovanje lozinke**

**Kratak opis:** Radnik resetuje svoju lozinku ukoliko je zaboravio postojeću.

**Učesnici:** Radnik (korisnik)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Radnik se prijavljuje na sistem.

**Opis:** Radnik zaboravlja lozinku i koristi funkcionalnost za resetovanje lozinke koja mu omogućava da postavi novu lozinku.

**Izuzeci:** Radnik ne može resetovati lozinku bez pravilne identifikacije.

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Radnik je postavio novu lozinku i može je koristiti za prijavu na sistem.

**Slučaj upotrebe: Otvaranje računa**

**Kratak opis:** Klijent otvara bankovni račun, pružajući potrebne informacije.

**Učesnici:** Klijent, Radnik (korisnik)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:**

**Opis:** Klijent se obraća radniku da otvori bankovni račun. Radnik prikuplja potrebne informacije od klijenta, uključujući lične podatke, vrstu računa itd.

**Izuzeci:** Otvaranje računa zahteva određene dokumente i podatke, koje klijent ne može dostaviti.

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Klijent je uspešno otvorio račun.

**Slučaj upotrebe: Otvaranje deviznog računa**

**Kratak opis:** Klijent otvara devizni račun u zavisnosti od postojanja običnog dinarskog računa.

**Učesnici:** Klijent, Radnik (korisnik)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Klijent ima aktivan običan dinarski račun.

**Opis:** Klijent želi otvoriti devizni račun i obaveštava radnika. Radnik proverava da li klijent ima aktivan običan dinarski račun pre nego što otvori devizni račun.

**Izuzeci:** Klijent nema postojeći običan dinarski račun.

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Klijent je uspešno otvorio devizni račun.

**Slučaj upotrebe: Uplata na račun**

**Kratak opis:** Klijent vrši uplatu na svoj bankovni račun.

**Učesnici:** Klijent, Radnik (korisnik)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Klijent je prijavljen na sistem i ima aktivan bankovni račun.

**Opis:** Klijent uplaćuje određeni iznos na svoj bankovni račun, obaveštava radnika o iznosu i transakcija se evidentira.

**Izuzeci:** Klijent nema dovoljno sredstava na računu za uplatu.

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Stanje na računu je ažurirano nakon uplate.

**Slučaj upotrebe: Transfer između računa**

**Kratak opis:** Klijent vrši transfer sredstava sa jednog računa na drugi.

**Učesnici:** Klijent, Radnik (korisnik)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Klijent je prijavljen na sistem i ima aktivan bankovni račun sa dovoljnim sredstvima.

**Opis:** Klijent obaveštava radnika o želji za transferom sredstava sa jednog računa na drugi, pružajući detalje o računima i iznosu za transfer.

**Izuzeci:** Nema dovoljno sredstava na izvoru transfera, ili neispravni računi.

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Transfer je uspešno izvršen, a stanje na oba računa je ažurirano.

**Slučaj upotrebe: Plaćanje**

**Kratak opis:** Klijent vrši plaćanje koristeći svoj bankovni račun.

**Učesnici:** Klijent, Radnik (korisnik), Primalac plaćanja

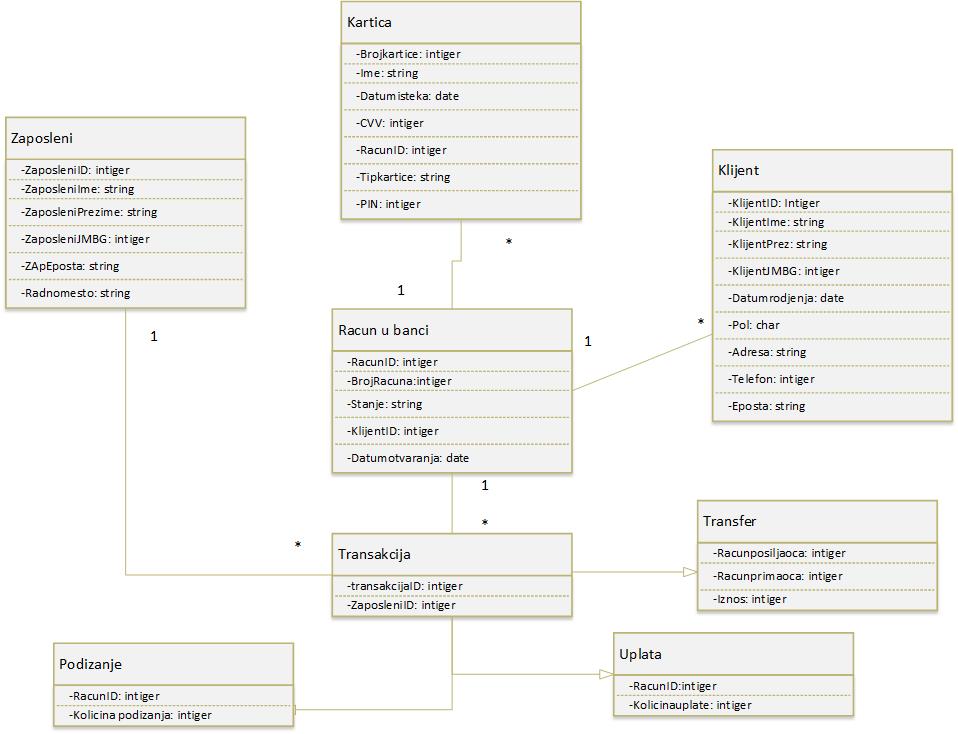
**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Klijent je prijavljen na sistem, ima aktivan bankovni račun, i unosi informacije o plaćanju.

**Opis:** Klijent pruža informacije o plaćanju, uključujući primalca i iznos, a radnik izvršava transakciju prema tim informacijama.

**Izuzeci:** Nema dovoljno sredstava na računu za plaćanje.

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Plaćanje je uspešno izvršeno, a stanje na računu je ažurirano.

Sada su svi vaši use case scenariji opisani. Ako imate dodatna pitanja ili trebate više informacija, slobodno pitajte.



Dijagram klasa prikazuje vezu između klijenta koji može imati više različitih računa u banci, i to kako za jedan račun može postojati više vezanih kartica. Takođe, daje prikaz toga kako radnik u banci može vršiti različite vrste transakcija i promena vezanih za ove račune.

1. Baza podataka

1.1. Tabela "Korisnici":

- Ova tabela sadrži informacije o trenutno zaposlenim korisnicima u banci.

- Polja uključuju ključne informacije o korisnicima poput imena, prezimena, radnog mesta i drugih relevantnih podataka.

1.2. Tabela "Klijenti":

- Ova tabela služi za čuvanje podataka o klijentima banke.

- Svaki red predstavlja jednog klijenta, sa poljima kao što su ime, prezime, adresa, kontakt informacije itd.

1.3. Tabela "Racuni":

- Tabela koja čuva informacije o bankovnim računima.

- Svaki red je povezan sa određenim klijentom i može sadržavati više računa. Polja obuhvataju informacije kao što su broj računa, saldo i datum otvaranja.

1.4. Tabela "Kartice":

- Ova tabela beleži informacije o karticama povezanim sa određenim bankovnim računom.

- Omogućava povezivanje više kartica sa jednim računom.

1.5. Tabela "Services":

- Tabela koja prati korišćenje bankarskih usluga.

- Svaki red predstavlja jednu uslugu, a polja uključuju tip usluge, datum, iznos itd.

1.6. Tabele za specifične usluge:

- Posebne tabele, poput "Transferi", "Depoziti", "PodizanjeNovca", svaka sa detaljnim informacijama o određenoj vrsti usluge.

1.7. Tabela "EvidencijaKorisnikaUsluga":

- Tabela koja prati korisnike koji su izvršili određene usluge.

- Povezuje korisnike sa uslugama i čuva relevantne informacije.

2. SQL Upiti i DML naredbe

2.1. SQL upiti za izlistavanje informacija:

- Primer upita za dobijanje podataka o korisnicima, klijentima, računima, karticama i uslugama.

2.2. SQL DML naredbe za dodavanje, ažuriranje i brisanje podataka:

- Primeri naredbi za dodavanje novih korisnika, klijenata, računa, kartica, usluga, kao i ažuriranje i brisanje postojećih podataka.

2.3. Ostale specifične SQL naredbe:

- Ako postoje specifične naredbe ili procedura vezane za ovu bazu podataka, navedite ih.

6. Kreiranje WPF aplikacije:

6.1. Dodavanje:

- Omogućava korisnicima unos novih podataka, uključujući informacije o klijentima, otvaranje računa i kartica.

- Interfejs treba biti jasan, sa odgovarajućim poljima za unos i validacijom podataka kako bi se osigurala tačnost informacija.

6.2. Izmena:

- Pruža mogućnost korisnicima da vrše izmene na postojećim podacima, uključujući informacije o klijentima, računima i karticama.

- Forma za izmenu treba automatski popuniti trenutne podatke kako bi olakšala proces izmene.

6.3. Brisanje:

- Korisnicima je omogućeno brisanje nepotrebnih informacija, kao što su klijenti, računi ili kartice.

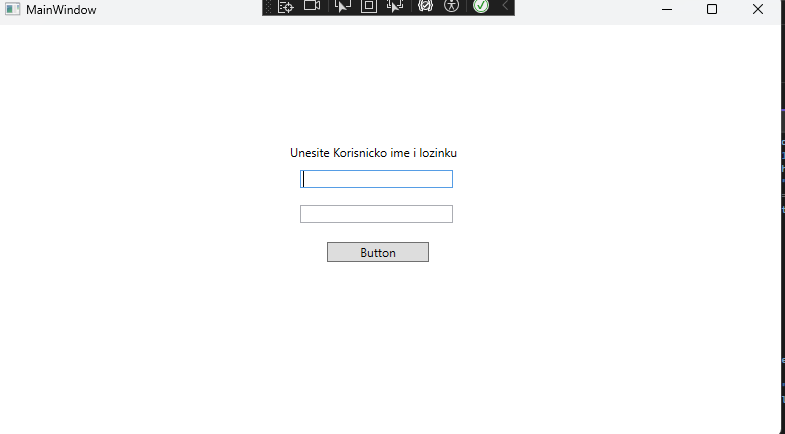
- Pre brisanja, implementirati odgovarajuće provere kako bi se izbeglo slučajno brisanje važnih podataka.

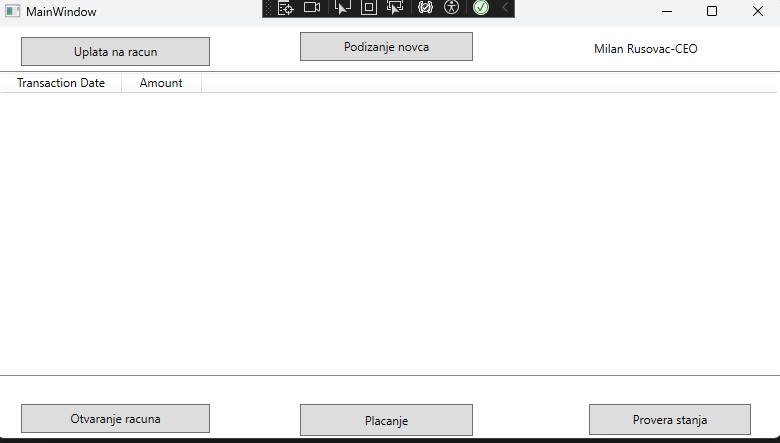
6.4. Izgled aplikacije:

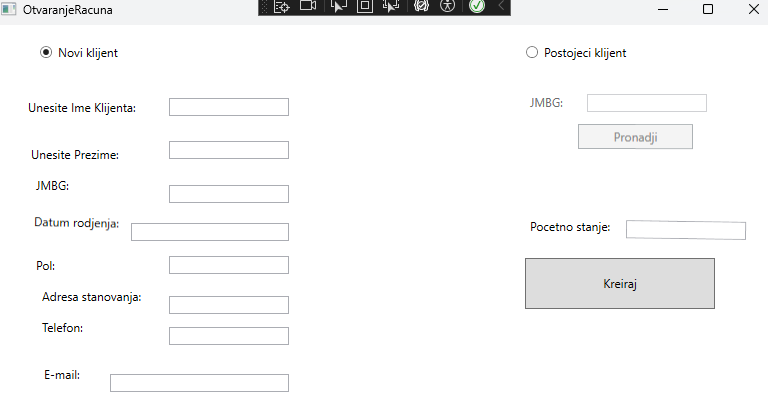
- Definisati estetski i funkcionalno prijatan korisnički interfejs koji odražava potrebe korisnika.

- Postaviti odgovarajuće kontrole za unos podataka, prikazivanje informacija i navigaciju između različitih funkcionalnosti.

- Prilagoditi izgled prema principima dobre prakse u dizajnu korisničkog interfejsa.







7. Testiranje aplikacije:

7.1. Planiranje testova:

- Definisano je niz testova koji pokrivaju sve ključne funkcionalnosti aplikacije, uključujući dodavanje, izmenu, brisanje, unos podataka i druge operacije.

7.2. Sprovođenje testova dodavanja:

- Testiranje dodavanja novih klijenata, otvaranja računa i kartica kako bi se proverilo da li se podaci uspešno čuvaju u bazi.

- Simulirane su različite scenarije dodavanja, uključujući regularne i ekstremne situacije.

7.3. Sprovođenje testova izmene:

- Izmene su testirane kako bi se proverilo ispravno ažuriranje postojećih podataka, uključujući informacije o klijentima, računima i karticama.

- Testirane su i validacije podataka kako bi se osigurala bezbednost od pogrešnih unosa.

7.4. Sprovođenje testova izvlačenja podataka:

- Pregledano je izvlačenje podataka o klijentima, računima i transakcijama kako bi se proverilo tačno prikazivanje informacija korisnicima.

- Različite scenarije su simulirane kako bi se testirala tačnost i kompletnost prikaza.

7.5. Provera sigurnosti:

- Testirane su sigurnosne funkcionalnosti aplikacije, uključujući autentifikaciju i autorizaciju, kako bi se osiguralo da samo ovlašćeni korisnici mogu pristupiti određenim funkcionalnostima.

- Pokušani su i pogrešni unosi i neautorizovani pristupi kako bi se proverila bezbednost sistema.

7.6. Testiranje performansi:

- Provereno je kako aplikacija radi pod različitim uslovima opterećenja kako bi se osiguralo efikasno i pouzdano funkcionisanje.

7.7. Dokumentacija rezultata testiranja:

- Detaljno su dokumentovani rezultati testiranja, uključujući pronađene greške, ispravke, i zaključke o ispravnosti aplikacije.

8. Zaključak:

Projekat je usmeren na razvoj Poslovnog Informacionog Sistema za banku, sa fokusom na efikasno praćenje bankarskih operacija i klijentskih interakcija. U toku realizacije projekta, ostvareni su sledeći ključni rezultati:

8.1. Postignuti ciljevi:

- Implementiran je funkcionalan sistem koji omogućava dodavanje, izmenu, brisanje podataka, kao i praćenje bankarskih operacija.

- Razvijena WPF aplikacija pruža intuitivan korisnički interfejs za korisnike banke.

8.2. Tehnički izazovi:

- Tokom implementacije, susretali smo se sa određenim tehničkim izazovima, poput integracije sa Microsoft Database Managerom i osiguranja sigurnosti aplikacije.

8.3. Testiranje i bezbednost:

- Sprovedeni su obuhvatni testovi, uključujući testiranje dodavanja, izmene, brisanja podataka i provere bezbednosti.

- Identifikovane greške su korigovane, a bezbednosni aspekti su proaktivno osigurani.

8.4. Zaključak o uspehu projekta:

- Projekat se smatra uspešnim s obzirom na postizanje postavljenih ciljeva i efikasno rešavanje izazova.

- Aplikacija predstavlja korisno sredstvo za praćenje bankarskih operacija i unapređenje korisničkog iskustva.

8.5. Preporuke za buduće unapređenje:

- Iako postignut uspeh, preporučuje se nastavak praćenja i ažuriranja sistema kako bi se osigurala njegova dugotrajna funkcionalnost.

- Razmatranje dodatnih funkcionalnosti ili optimizacija kako bi se poboljšalo korisničko iskustvo.

Kroz implementaciju ovog Poslovnog Informacionog Sistema, stvorena je solidna osnova za dalji razvoj i unapređenje bankarskih procesa.