

ANALIZA SISTEMA

1. UVOD

Sistem za banku koji omogućava:

- korisnicima banke da urade stvari za koje bi inače morali otići u poslovnicu, kao što je vršenje uplata na drugi račun
- bankarima da vode evidenciju, upisuju nove klijente, potvrđuju zahtjeve za kredite i slično
- administratoru da nadzire čitav spektar radnji vezanih za banku, kao i da dodaje novosti, uređuje objekte na mapi (bankomate i filijale)

2. OSNOVNE KLASSE (MODELI)

1. Korisnik – *apstraktna klasa koja opisuje korisnika koji se može logovati na sistem*

- Atributi:
 - Id (int)
 - Ime (string)
 - Prezime (string)
 - KorisnickoIme (string)
 - Lozinka (string)
- Metode
 - getteri i setteri

2. Adresa – *klasa koja opisuje mjesto na mapi*

- Atributi:
 - Id (int)
 - Latitude (float)
 - Longitude (float)
 - Naziv (String)
- Metode
 - konstruktor
 - getteri i setteri

3. Klijent extends Korisnik – *klasa koja opisuje klijenta banke*

- Atributi:
 - DatumRodjenja (DateTime)
 - Spol (Spol)
 - JMBG (string)
 - BrojTelefona (string)

- BrojLicneKarte (string)
 - Adresa (Adresa)
 - Zanimanje (string)
 - Grad (string)
 - Drzava (string)
 - VrijemeDodavanja (VrijemeDodavanja)
 - Metode
 - konstruktor
 - getteri i setteri
4. Racun – *opisuje bankovni račun*
- Atributi:
 - Id (Int)
 - StanjeRacuna (Float)
 - VrstaRacuna (VrstaRacuna)
 - Klijent (Klijent)
 - Metode
 - konstruktor
 - getteri i setteri
5. Transakcija
- Atributi:
 - Id (int)
 - Vrijeme (DateTime)
 - SaRacuna (Racun)
 - NaRacun (Racun)
 - Iznos (Float)
 - VrstaTransakcije (VrstaTransakcije)
 - NacinTransakcije(NacinTransakcije)
 - Metode
 - konstruktor
 - getteri i setteri
6. KreditBaza – *apstraktna klasa koja opisuje zahtjev za kredit ili kredit koji je u toku ili završen*
- Atributi:
 - Id (int)
 - Racun (Racun)
 - Iznos (float)
 - KamatnaStopa (float)
 - RokOtplate (RokOtplate)
 - Metode
 - getteri i setteri
7. ZahtjevZaKredit extends KreditBaza
- Atributi:

- NamjenaKredita (string)
- MjesecniPrihodi (float)
- ProsjecniTroskoviDomacinstva (float)
- NazivRadnogMjesta (string)
- NazivPoslodavca (string)
- RadniStaz (int)
- BrojNekretnina (int)
- BracnoStanje (BracnoStanje)
- SupruznikIme (string)
- SupruznikPrezime (string)
- SupruznikZanimanje (string)
- ImaNeplacenihDugova (bool)
- BrojNeplacenihDugova (float)
- StatusZahteva (StatusZahtjevaZaKredit)
- Metode
 - konstruktor
 - getteri i setter
 - DajKreditnuSposobnost(): char

8. Kredit extends KreditBaza

- Atributi:
 - IsplaceniIznos (float)
 - PocetakOtplata (DateTime)
 - StatusKredita (StatusKredita)
- Metode
 - konstruktor
 - getteri i setteri
 - UplatiMjesecnuRatu(): void
 - ZavrshiKredit(): void

9. Bankar extends Korisnik

- Atributi:
 - mjestoZaposlenja (Filijala)
- Metode
 - konstruktor
 - getteri i setteri

10. Administrator extends Korisnik

- Metode
 - konstruktor
 - getteri i setteri

11. Filijala extends IMapObjekat

- Atributi:
 - Id (int)
 - Ime (string)

- Adresa (Adresa)
- BrojTelefona (string)
- Metode
 - konstruktori
 - getteri i setteri
 - string DajVrstu()

12. Bankomat extends IMapObjekat

- Atributi:
 - Id (int)
 - Adresa (Adresa)
 - Ime (string)Metode
 - konstruktor
 - getteri i setteri
- Metode
 - konstruktori
 - getteri i setteri
 - string DajVrstu()

13. Novost – *klasa koja predstavlja jednu vijest na oglasnoj ploči*

- Atributi:
 - Id (int)
 - VrijemeDodavanja (DateTime)
 - Naslov (string)
 - Sadržaj (string)
 - Prikazana (bool)
- Metode
 - konstruktor
 - getteri i setteri

14. Konverzija

- Atributi:
 - Iznos (float)
 - KonvertovaniIznos (float)
 - IzValute (Valute)
 - UValutu (Valuta)
- Metode
 - konstruktor
 - getteri i setteri

15. StatusKredita (enum)

- Aktivan
- Završen

16. VrstaTransakcije (enum)

- UobicajenoPlacanje
- IndividualnoPlacanje
- Kupnja
- IndividualniDohodak
- UobicajeniDohodak

17. VrstaRacuna (enum)

- Tekuci
- Ziro
- Devizni
- Stedni

17. BracnoStanje (enum)

- Ubraku
- Razveden
- Samac

18. NacinTransakcije (enum)

- Interna
- NaRacunDrugeBanke
- SaRacunaDrugeBane

19. RokOtplate (enum)

- Trajanje_1_godina
- Trajanje_5_godina
- Trajanje_10_godina
- Trajanje_15_godina
- Trajanje_20_godina

20. Spol (enum)

- Muško
- Žensko

21. StatusZahtjevaZaKredit (enum)

- Neobradjen
- Odobren
- Odbijen

3. INTERFACE

1. IOglasnaPloca

- Metode

- DodajNovost(novost: Novost): void
- UrediNovost(novost: Novost): void
- UkloniNovost(novost: Novost): void
- DajSveNovosti() : List<Novost>
- DajSvePrikazaneNovosti() : List<Novost>
- DajNovost(id: int) : Novost
- DaLiPostojiNovost(id: int) : bool

2. IBankari

- Metode

- DodajBankara(bankar: Bankar)
- UrediBankara(bankar: Bankar)
- UkloniBankara(id: int): void
- DajSveBankare(): List<Bankar>
- DajBankara(id: int): Bankar
- DaLiPostojiBankar(id: int): bool
- DajBankara(korisnickoIme: string): Bankar

3. IAdministratori

- Metode

- DajAdministratora(korisnickoIme: string): Administrator
- DajAdministratora (id: int)

4. IFilijaleBankomati

- Metode

- DodajBankomat(bankomat: bankomat)
- UrediBankomat(bankomat: bankomat)
- DodajFilijalu(filijala: filijala): void
- UrediFilijalu(filijala: filijala): void
- UkloniBankomat(id: int): void
- UkloniFilijalu(id: int): void
- DajSveMapObjekte(): List<ImapObjekat>
- DajSveFilijale(): List<Filijala>
- DajSveBankomate(): List<Bankomat>
- DajFilijalu(id: int): Filijala
- DajBankomat(id: int): Bankomat
- DaLiPostojiFilijala(id: int): bool
- DaLiPostojiBankomat(id: int): bool

5. IKrediti

- Metode

- DajKredit(id: int): Kredit

- DaLiPostojiKredit(id: int): bool
- DajSveKredite(): List<Kredit>
- DajSveKrediteKlijenta(id: int): List<Kredit>

6. IRacuni

- Metode
 - OtvoriRacun(racun: Racun): void
 - ZatvoriRacun(id: int): void
 - DajRacun(id: int): Racun
 - DajSveRacune(): List<Racun>
 - DaLiPostojiRacun(id: int): bool
 - DajSveRacuneKlijenta(id: int): List<Racun>
 - UrediStanjeRacuna(racun: Racun): void
 - DajRacune(id: int): List<Racun>

7. IKlijenti

- Metode
 - DodajKlijenta(klijent: Klijent)
 - UrediKlijenta(klijent: Klijent)
 - UkloniKlijenta(klijent: Klijent)
 - DajSveKlijente(): List<Klijent>
 - DajKlijenta(id: int): Klijent
 - DajKlijentaLK(brojLicneKarte: string): Klijent
 - DaLiPostojiKlijent(id: int): bool
 - DajKlijenta(korisnickoIme: string): Klijent

8. ITransakcije

- Metode
 - Uplati(transakcija: Transakcija)
 - DajSveTransakcije(): List<Transakcija>
 - DajTransakciju(id: int): Klijent
 - DajTransakcije(id: int): List<Transakcija>
 - DaLiPostojiTransakcija(id: int): bool

9. IZahtjeviZaKredit

- Metode
 - PodnesiZahtjevZaKredit(zahtjevZaKredit: ZahtjevzaKredit): void
 - RijesiZahtjev(id: int, bool: prihvacen): void
 - DajSveZahtjeve(): List<ZahtjevZaKredit>
 - DajZahtjev(id: int): ZahtjevZaKredit
 - DaLiPostojizahtjev(id: int): bool

4. REPOSITORY KLASSE

1. OglasnaPloca

- Metode

- DodajNovost(novost: Novost): void
- UrediNovost(novost: Novost): void
- UkloniNovost(novost: Novost): void
- DajSveNovosti() : List<Novost>
- DajSvePrikazaneNovosti() : List<Novost>
- DajNovost(id: int) : Novost
- DaLiPostojiNovost(id: int) : bool

2. Bankari

- Metode

- DodajBankara(bankar: Bankar)
- UrediBankara(bankar: Bankar)
- UkloniBankara(id: int): void
- DajSveBankare(): List<Bankar>
- DajBankara(id: int): Bankar
- DaLiPostojiBankar(id: int): bool
- DajBankara(korisnickoIme: string): Bankar

3. Administratori

- Metode

- DajAdministratora(korisnickoIme: string): Administrator
- DajAdministratora (id: int)

4. FilijaleBankomati

- Metode

- DodajBankomat(bankomat: bankomat)
- UrediBankomat(bankomat: bankomat)
- DodajFilijalu(filijala: filijala): void
- UrediFilijalu(filijala: filijala): void
- UkloniBankomat(id: int): void
- UkloniFilijalu(id: int): void
- DajSveMapObjekte(): List<ImapObjekat>
- DajSveFilijale(): List<Filijala>
- DajSveBankomate(): List<Bankomat>
- DajFilijalu(id: int): Filijala
- DajBankomat(id: int): Bankomat
- DaLiPostojiFilijala(id: int): bool
- DaLiPostojiBankomat(id: int): bool

5. Krediti

- Metode

- DajKredit(id: int): Kredit

- DaLiPostojiKredit(id: int): bool
- DajSveKredite(): List<Kredit>
- DajSveKrediteKlijenta(id: int): List<Kredit>

6. Racuni

- Metode
 - OtvoriRacun(racun: Racun): void
 - ZatvoriRacun(id: int): void
 - DajRacun(id: int): Racun
 - DajSveRacune(): List<Racun>
 - DaLiPostojiRacun(id: int): bool
 - DajSveRacuneKlijenta(id: int): List<Racun>
 - UrediStanjeRacuna(racun: Racun): void
 - DajRacune(id: int): List<Racun>

7. Klijenti

- Metode
 - DodajKlijenta(klijent: Klijent)
 - UrediKlijenta(klijent: Klijent)
 - UkloniKlijenta(klijent: Klijent)
 - DajSveKlijente(): List<Klijent>
 - DajKlijenta(id: int): Klijent
 - DajKlijentaLK(brojLicneKarte: string): Klijent
 - DaLiPostojiKlijent(id: int): bool
 - DajKlijenta(korisnickoIme: string): Klijent

8. Transakcije

- Metode
 - Uplati(transakcija: Transakcija)
 - DajSveTransakcije(): List<Transakcija>
 - DajTransakciju(id: int): Klijent
 - DajTransakcije(id: int): List<Transakcija>
 - DaLiPostojiTransakcija(id: int): bool

9. ZahtjeviZaKredit

- Metode
 - PodnesiZahtjevZaKredit(zahtjevZaKredit: ZahtjevzaKredit): void
 - RijesiZahtjev(id: int, bool: prihvacen): void
 - DajSveZahtjeve(): List<ZahtjevZaKredit>
 - DajZahtjev(id: int): ZahtjevZaKredit
 - DaLiPostojizahtjev(id: int): bool

5. BAZA PODATAKA

Biti će neophodna baza podataka koja će čuvati sve tabele slične klasama iznad.

6. API

API će se koristiti za dobijanje informacija o trenutnom stanju deviza pri konvertovanju istih.