PROJEKTNI ZADATAK  
Upravljanje radom banke

Studenti:

1. Ibrahim Okić

Sadržaj

[1 Tema 1](#_Toc84573880)

[2 Izrada i predavanje projekta 1](#_Toc84573881)

[3 Tehnički zahtjevi 1](#_Toc84573882)

[4 Formalni zahtjevi 2](#_Toc84573883)

[5 Izvještaj o projektu 3](#_Toc84573884)

# Tema

Napisati program za upravljanje radom banke.

# Izrada i predavanje projekta

1. Projekat rade grupe po najviše 4 studenta.
2. Rok za predaju projekta je 20.12.2023. godine.
3. Završen CodeBlocks C++17 projekat i Word dokument arhivirati kao ZIP datoteku i poslati na e-mail adresu predmetnog asistenta sa subject-om “PJIP Projekat”. Obavezno obrisati izvršne/objektne datoteke (.exe i .o) iz projekta prije arhiviranja!
4. Projekat se predaje i ocjenjuje prije izlaska na završni ispit.
5. Ukoliko student odluči ne raditi projekat, može izaći na završni ispit.
6. Ukoliko student pošalje projekat nakon termina završnog ispita, mora ponovo izaći na završni ispit kako bi se ostvarili bodovi na projekat.
7. Ispunjenjem tehničkih i formalnih zahtjeva grupa studenata ostvaruje po 20 bodova.
8. Djelimičnim ispunjenjem tehničkih i formalnih zahtjeva asistent odlučuje o broju bodova koji će se dodjeliti grupi za projekat.

# Tehnički zahtjevi

1. Omogućiti operacije unosa, izmjene, pregleda i brisanja: korisnika, radnika, tekućih i štednih računa, kredita i transakcija.
2. Unos novih stavki, te izmjenu postojećih vršiti sa tastature.
3. Željenu radnju (unos, izmjena, pregled, brisanje) korisnik unosi preko tastature na osnovu ponuđenih opcija u terminalu.
4. Za novi unos definirati potrebne kolone i ograničiti unos ukoliko se ne unesu potrebni podaci.
5. Za povezivanje tabela koristit će se ključevi, koji predstavljaju jedinstveni identifikator određene stavke, kao npr:

* radnik/korisnik: tekstualni zapis JMBG od 13 cifara,
* tekući/štedni račun: tekstualni zapis od 16 cifara;

1. Napraviti dokumentaciju projekta u obliku Word dokumenta. Koristiti ovaj dokument kao šablon, te početi sa 5. poglavljem svoj izvještaj o projektu.

# Formalni zahtjevi

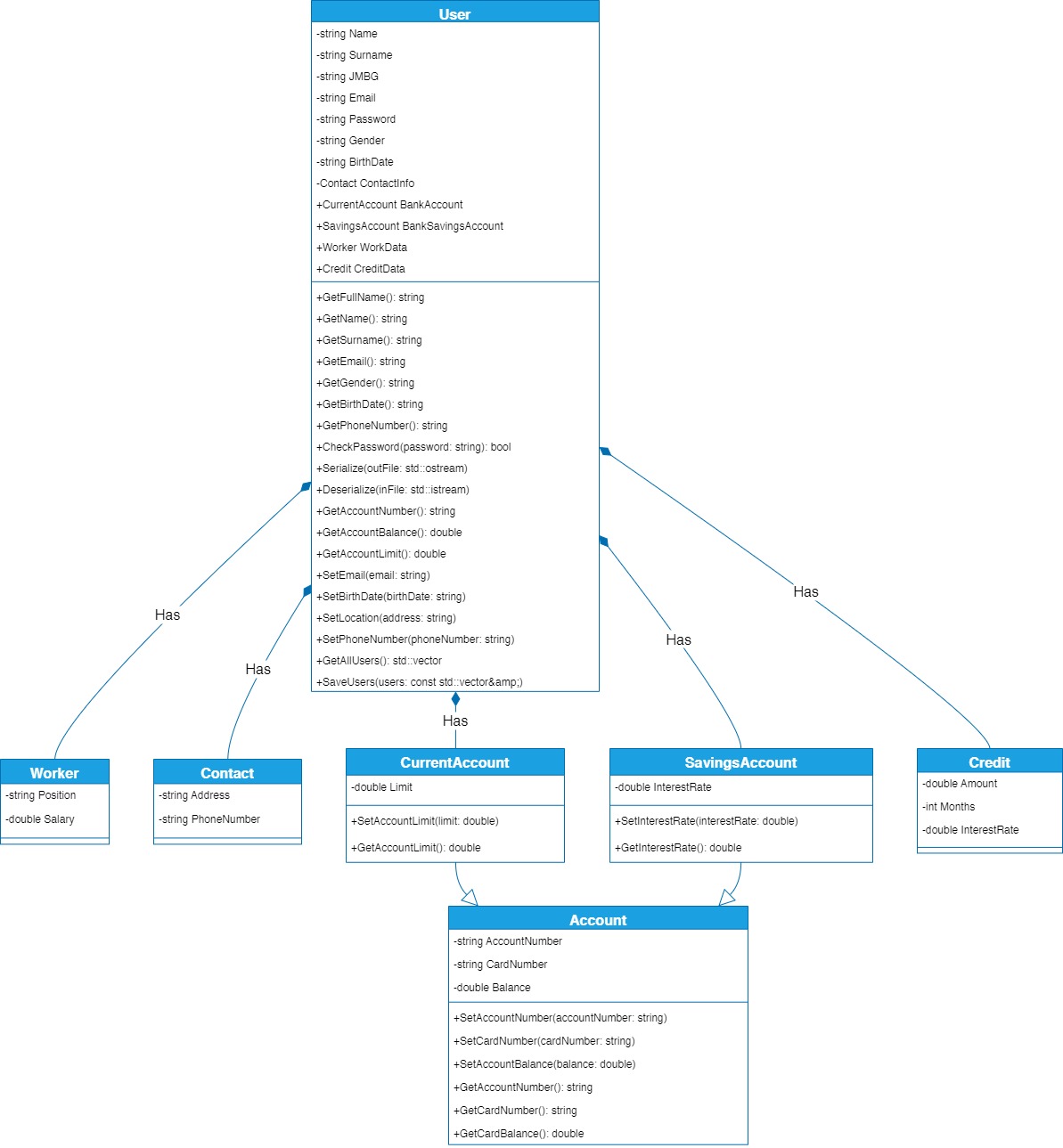
1. Programski kôd (*backend*) pisati u C++ programskom jeziku.
2. Koristiti što više elemenata C++17 standarda.
3. Programsko sučelje (*frontend*) bit će terminal (*Command Prompt* ili UNIX terminal).
4. Strogo se pridržavati principa objektno – orijentiranog programiranja.
5. Jasno definirati potrebne 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. Koristiti STL kontejnere, a gdje to nije moguće koristiti dinamički alociranu memoriju.
11. Smanjiti ponavljanje kôda kroz implementaciju funkcija.
12. Gdje je moguće koristiti metode umjesto funkcija.
13. Deklaracije izdvojiti u .h datoteke, definicije u .cpp datoteke, a napraviti i main.cpp datoteku koja predstavlja glavni program.
14. Paziti na zauzimanje resursa radne memorije.
15. Objekte, klase, članove klase i njene metode imenovati razumljivo.
16. Nastojati na jedinstven način imenovati metode koji obavljaju iste funkcije u više klasa.

# Izvještaj o projektu

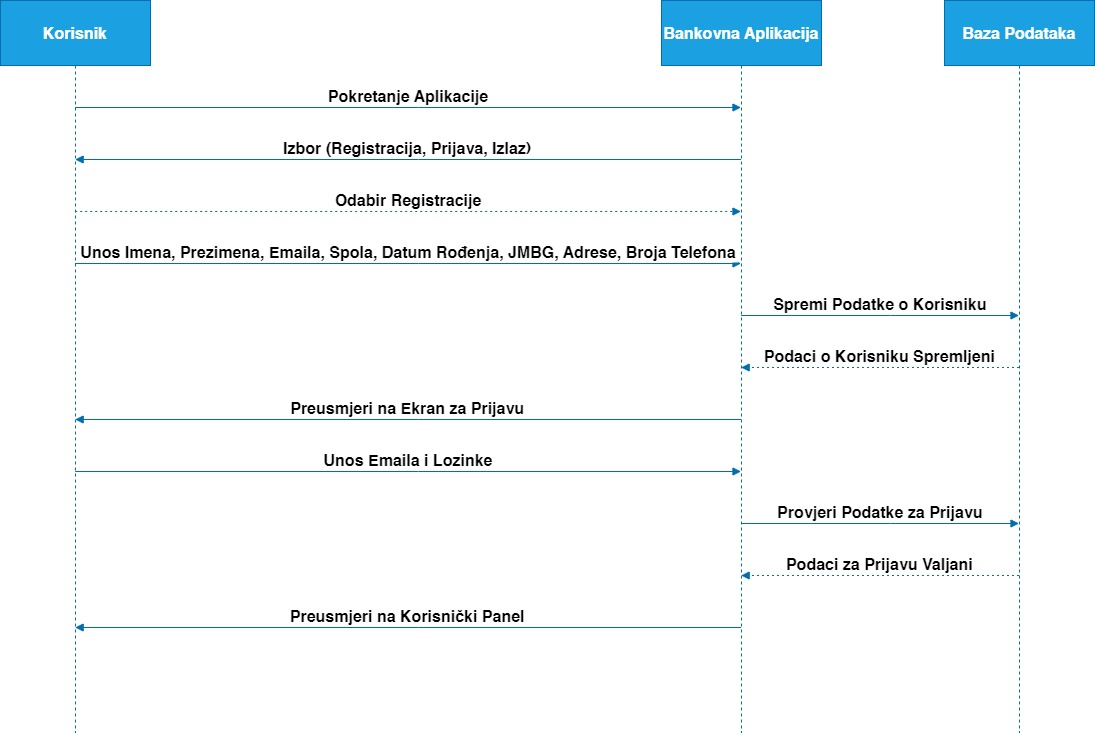
Sama izrada projekta **‘Upravljanje radom banke’** trajala je oko 7 dana gdje sam svaki dan par sati radio na izradi i dizajniranju samih funkcionalnosti ove aplikacije.

Prije nego što sam krenuo pisati sam kod htio sam grafički prikazati kako bi izgledao cijeli rad aplikacije pomoću dijagrama. Za izradu dijagrama koristio sam **draw.io** web stranicu i koristio sam se Klasnim dijagramima i Sekvencijalnim dijagramima.

* Klasni dijagram u koji sadrži sve klase korištene u aplikaciji:



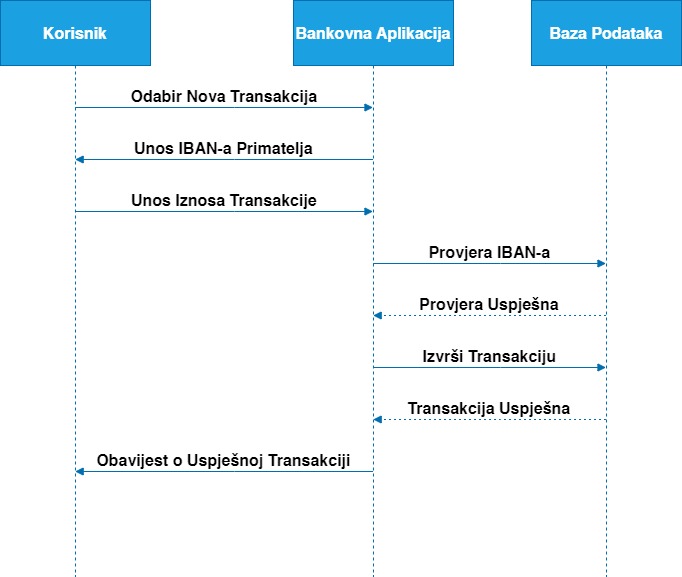
* Proces prijave i registracije na aplikaciju izgleda ovako:



* Opcije koje se nalaze na glavnom korisničkom sučelju koje se pojavljuje nakon prijave:



* Proces transakcije:



* Administrativni dio za zaposlene osobe u banci:

