

| OPŠTI PODACI O PROJEKTNOM TIMU | | | | |
|---|--------------|--------|-----------|------------------------|
| Br grupe | Broj indeksa | Ime | Prezime | Email adresa |
| 1 | pr14/2022 | Milos | Susic | milossusic22@gmail.com |
| 1 | pr8/2022 | Bozana | Todorovic | bozanabojan@gmail.com |
| Github link | | | | |
| https://github.com/MilosSusic/ProjekatMreze | | | | |

| OPŠTI PODACI O PROJEKTU | |
|--------------------------------|--|
| Naziv projekta: | Simulacija rada bankovnog sistema |
| TEHNIČKI OPIS PROJEKTA | |
| Sažetak: | <p>Simulacija rada bankovnog sistema predstavlja softverski projekat koji oponaša osnovne funkcionalnosti modernog bankovnog sistema. Cilj projekta je da pruži korisnicima mogućnost registracije i prijave na njihove račune, kao i da omogući osnovne bankarske operacije poput uplate, isplate i transfera sredstava između računa unutar iste banke. Centralni server, kao jezgro sistema, inicijalizuje filijale, upravlja njihovim resursima i koordiniše aktivnosti klijenata koristeći UDP za inicijalizaciju i TCP za komunikaciju između klijenata i filijala. Projekat demonstrira upravljanje resursima u distribuiranim sistemima i implementaciju osnovnih principa bezbedne razmene informacija u mrežnom okruženju.</p> |
| Primenjene metode: | <p>Komunikacioni protokoli:</p> <ul style="list-style-type: none"> • UDP protokol: Koristi se za inicijalizaciju filijala i dodelu početnih resursa, poput maksimalnog budžeta i broja dozvoljenih istovremenih TCP veza po filijali. • TCP protokol: Omogućava pouzdanu dvosmernu komunikaciju između klijenata i filijala za obavljanje registracija, prijave i transakcija. <p>Upravljanje klijentima i resursima:</p> <ul style="list-style-type: none"> • Centralni server dodeljuje resurse filijalama, vodi računa o budžetima i ograničava broj aktivnih veza. • Filijale dinamički prihvataju zahteve od klijenata i upravljaju njihovim sesijama, u skladu sa zadatim ograničenjima. <p>Skladištenje i obrada podataka:</p> <ul style="list-style-type: none"> • Centralizovano čuvanje podataka o svim filijalama i klijentima. • Lokalno vođenje sesija na nivou filijale radi smanjenja opterećenja centralnog servera. <p>Sigurnosne metode:</p> <ul style="list-style-type: none"> • Unos korisničkog imena i lozinke se koristi za autentifikaciju korisnika. • Ograničavanje maksimalnog iznosa transakcije i kontrola budžeta radi minimizacije grešaka i potencijalnih zloupotreba. <p>Model transakcija:</p> <ul style="list-style-type: none"> • Implementacija osnovnih bankarskih operacija: uplata, isplata i transfer sredstava uz proveru raspoloživog stanja i budžetskih limita. |
| Opis projekta: | <p>Projekat se zasniva na razvoju sistema sa centralizovanom kontrolom koju obavlja server. Centralni server predstavlja glavnu tačku za upravljanje informacijama i resursima u okviru bankarskog sistema. Nakon pokretanja simulacije, centralni server inicijalizuje određeni broj filijala, pri čemu svaka filijala dobija:</p> <ul style="list-style-type: none"> • Maksimalni dozvoljeni budžet za transakcije. • Maksimalni broj klijenata koji mogu istovremeno obavljati aktivnosti u filijali (ograničeno brojem istovremenih TCP veza). <p>Svaka filijala koristi dodeljene resurse za upravljanje sesijama klijenata. Klijenti se povezuju sa filijalama putem TCP veze, gde imaju mogućnost:</p> <ol style="list-style-type: none"> 1. Registracije naloga: <ul style="list-style-type: none"> ○ Klijent unosi korisničko ime, lozinku i postavlja ograničenje za |

maksimalnu isplatu sredstava u jednoj transakciji.

- Nakon uspešne registracije, server dodeljuje klijentu jedinstveni broj računa.

2. Prijave na nalog:

- Klijent može da pristupi postojećem nalogu unošenjem korisničkog imena i lozinke.
- Nakon prijave, klijent ima pristup sledećim funkcionalnostima:
 - **Uplata sredstava:** Dodavanje novca na svoj račun.
 - **Isplata sredstava:** Podizanje novca sa računa uz poštovanje limita postavljenog pri registraciji.
 - **Transfer sredstava:** Prebacivanje novca na drugi račun unutar iste banke, unosom broja računa primaoca.

Centralni server u realnom vremenu prati aktivnosti filijala i korisnika, uključujući informacije o trenutnim budžetima i stanju računa. Takođe, server obezbeđuje da transakcije budu validne i da sve filijale funkcionišu u skladu sa zadatim ograničenjima.

Zadaci:

| | | | |
|---|--|----------|---------------|
| 1 | Zadatak: Skiciranje projekta | Rok: KT1 | Broj poena: 2 |
| | <p>Opis: Napraviti blok-dijagram i šemu koja prikazuje međusobne relacije između centralnog servera, filijale i klijenta. Dijagram treba da obuhvati osnovne tokove podataka, tipove komunikacije (TCP/UDP), kao i veze između različitih komponenti.</p> <p>Napomena: Uključiti opis klasa:</p> <ul style="list-style-type: none">• Korisnik: ID korisnika, ime, prezime, stanje na računu.• Transakcija: ID transakcije, tip transakcije (uplata, isplata, transfer), iznos, datum.• Enkriptor: Klasa koja implementira osnovni mehanizam šifrovanja i dešifrovanja podataka. | | |
| 2 | Zadatak: Implementacija servera sa osnovnom funkcionalnošću | Rok: KT1 | Broj poena: 3 |
| | <p>Opis: Kreirati server koji pruža sledeće osnovne funkcionalnosti:</p> <ul style="list-style-type: none">• Registrovanje korisnika (prijem i validacija podataka: ime, prezime, lozinka, početno stanje).• Pregled stanja računa (vraćanje trenutnog stanja na zahtev klijenta).• Provera validnosti transakcije (sprečavanje negativnog salda). <p>Napomena: Koristiti klasu Korisnik za čuvanje podataka o registrovanim korisnicima.</p> | | |
| 3 | Zadatak: Implementacija jedne filijale kao klijenta prema serveru | Rok: KT1 | Broj poena: 2 |
| | <p>Opis: Kreirati aplikaciju filijale koja šalje sledeće zahteve serveru:</p> <ul style="list-style-type: none">• Uplata sredstava na račun klijenta.• Isplata sredstava sa računa klijenta. <p>Napomena: Koristiti klasu Transakcija za evidentiranje svih aktivnosti.</p> | | |
| 4 | Zadatak: Implementacija klijenta sa osnovnom funkcionalnošću | Rok: KT1 | Broj poena: 3 |
| | <p>Opis: Kreirati aplikaciju klijenta koja omogućava:</p> <ul style="list-style-type: none">• Pregled stanja na računu.• Prijavu korisnika unosom korisničkog imena i lozinke. | | |

| | | | |
|---|--|----------|---------------|
| | Napomena: Koristiti klasu Korisnik za autentifikaciju i autorizaciju. | | |
| 5 | Zadatak: Proširenje skice projekta za kompleksnu funkcionalnost | Rok: KT2 | Broj poena: 2 |
| | Opis: Ažurirati blok-dijagram i šemu dodavanjem: <ul style="list-style-type: none"> • Više filijala (prikazati paralelne procese). • Više klijenata koji simultano šalju zahteve različitim filijalama. | | |
| 6 | Zadatak: Implementacija servera sa kompleksnom funkcionalnošću | Rok: KT2 | Broj poena: 4 |
| | Opis: Proširiti server tako da podržava: <ul style="list-style-type: none"> • Transfer sredstava između računa unutar iste banke. • Evidentiranje svih transakcija. • Proveru validnosti svih zahteva prema trenutnim resursima filijale. | | |
| 7 | Zadatak: Dodavanje više filijala i klijenata u sistem | Rok: KT2 | Broj poena: 3 |
| | Opis: Omogućiti kreiranje dodatnih instanci filijala i klijenata u sistemu. Svaka filijala treba da se poveže sa serverom i omogući simultanu komunikaciju sa klijentima. | | |
| 8 | Zadatak: Implementacija mehanizma za simultanu komunikaciju koristeći multipleksiranje | Rok: KT2 | Broj poena: 3 |
| | Opis: Realizovati mehanizam kojim server i filijale prate i odgovaraju na zahteve klijenata u realnom vremenu. | | |
| 9 | Zadatak: Dodavanje sistema za enkripciju komunikacije između servera i klijenata | Rok: KT2 | Broj poena: 3 |
| | Opis: Implementirati šifrovanje sadržaja svih poruka koje se razmenjuju između servera, filijala i klijenata koristeći klasu Enkriptor. Enkripcija se zasniva na algoritmu Homofonog šifrovanja sa proširenjem za šifrovanje brojeva. Rečnik se nasumično generiše na serveru i zatim se šalje filijalama koje ga šalju klijentima u toku komunikacije. | | |