**MRKiRM1**

Projektni zadatak:

Decentralizovana čet aplikacija

Student: Božidar Krunić RA160-2019

Mentor: Nemanja Raković

Novi Sad, 2023.

Sadržaj

[Opis zadatka 3](#_Toc124855192)

[Koncept rešenja zadatka 4](#_Toc124855193)

[Rešenje samog zadatka 5](#_Toc124855194)

[SDL dijagrami 6](#_Toc124855195)

[MSC dijagrami 10](#_Toc124855196)

[Testiranje 12](#_Toc124855197)

# Opis zadatka

QT razvojna okruženje – Interaktivna komunikacija na LAN mreži (decentralizovana).

Realizovati chat program tipa >>svaki-sa-svakim<<, bez postojanja centralne tačke razmene poruka.

Klijent po startovanju mora da otkrije sve ostale klijente u odgovarajućoj podmreži. Otkrivanje aktivnih klijenata se može obaviti slanjem inicijalne poruke svakom računaru u podmreži, na zadatom portu. Zatim, ostvaruje konekciju sa svakim otkrivenim klijentom. Svaki klijent mora da poseduje ime, koje se prosleđjuje kao parametar u komandnoj liniji.

Klijent odlazne poruke šalje svakom klijentu sa kojim je povezan. Poruke je potrebno ispisati na ekran, sa podatkom od koga je poruka dobijena. Odgovarajućim ispisom na ekranu potrebno je obavestiti korisnika o događajima kao što su pojava novog klijenta ili isključivanje klijenta.

# Koncept rešenja zadatka

Rešenje zadatka je realizovano u Qt Creator razvojnom okruženju, upotrebom signala I slotova za komunikaciju između objekata I udp soketa. Klijenti se među sobom otkrivaju uz pomoć broadcast paketa koji se svima šalju u datoj podmreži. Prilikom prijema broadcast poruke svaki klijent šalje svoj username originalnom pošiljaocu te broadcast poruke , na osnovu čega se pravi uvid u to koji se sve korisnici nalaze u toj mreži.

Nakon što je pribavljena lista IP adresa i username-ova, svaki klijent void računa o tome ko se novi pojavio u mreži i ko se isključio sa mreže. Svaki klijent šalje poruke svim klijentima u toj mreži i ispisuje username I sadržaj poruke od klijenta koji je poslao poruku.

# Rešenje samog zadatka

Protokol koji se koristi u ovoj mreži je UDP protkol. Njega sam odabrao zbog same jednostavnosti I lake konekcije, jer se ne koriste složene kontrole povezivanja. Traženje korisnika I slanje poruka je realizovano preko broadcast paketa koje dobijaju svi u mreži, a imena korisnika se šalju na IP adresu registratora to jest onog klijenta koji je zahtevao imena korisnika.

Program se prvobitno startuje a onda se unosi korisničko ime, IP adresa datog računara I onda se započinje slanje reg signala koji se periodično šsalje. Nakon svaka dva poslata signala se proverava stanje novih korisnika. Posebno su realizovane niti za aktiviranje signala za registraciju I slanje poruka, pa je postignuto periodično slanje signala. Možda je moglo I drugačijeovo da se odradi, upotrebom nekog tajmera u qt okruženju.

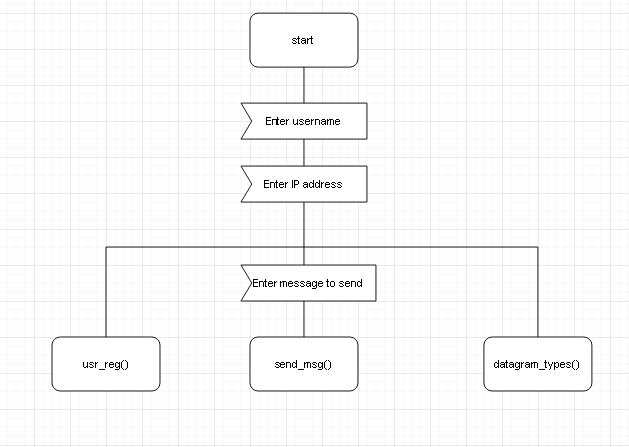
Glavna klasa koja je zadužena za komunikaciju je klijent I ona se sastoji od tri slota za prijem signala. To su slotovi send\_msg(), usr\_reg() I datagram\_types().

Send\_msg() I usr\_reg() se prozivaju iz posebnih niti.

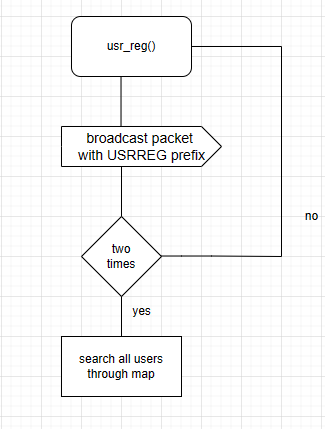
Slot datagram\_types() je srž ovog programa I služi za obradjivanje pristiglih UDP paketa sa mreže. Unutar njega se određuje kojeg tipa je paket, to je postignuto prefiksom koji su dodati na svaki poslati paket. Ako je primljen prefiks sa USRREG na koji se odgovara slanjem vlastitog username-a, ako je primljen USR tada se skladisti username u mapi sa ip adresom odakle je poslat taj username, čuva se u mapi da bi posle moglo lako da se otkrije ko je na mreži, I da bi mogo lako da se otkrije korisnik kome je namenjena ip adresa. Ako je primljen paket u kom se nalazi poruka, ona se zajedno sa imenom korisnika od koga je poslata poruka ispisuje na ekran drugih korisnika. Ovaj soket se aktivira uz pomoć signala kad god se nalaze neki podaci na udp soketu.

Prilikom svaka dva poslata reg signala u istoj funkciji se proverava stanje korisnika, koji je otisao a koji se pridružio. Svaki user koji je u mreži odgovara slanjem svojeg username-a kad god se aktivira reg signal, pa se na osnovu toga ako se salje username ili ne zna da je korisnik tu ili je otišao.

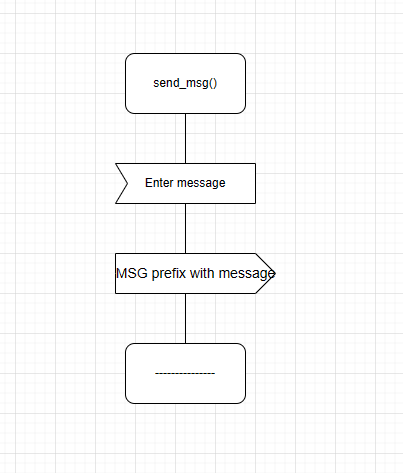
# SDL dijagrami



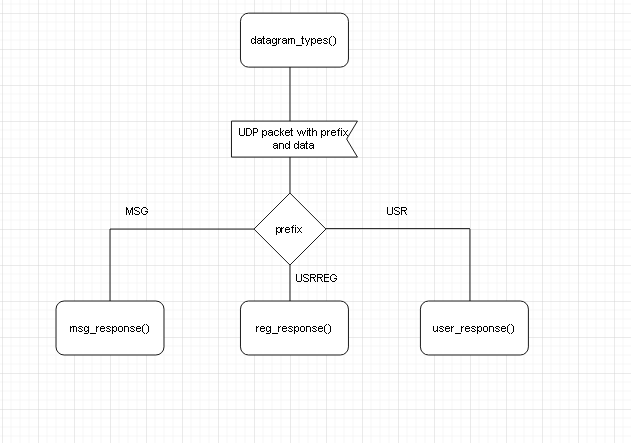
Dijagram startovanja programa



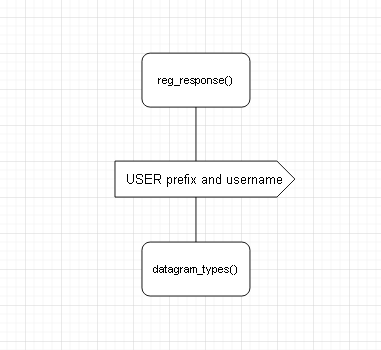
Dijagram usr\_reg



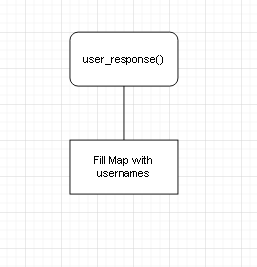
Dijagram send\_msg



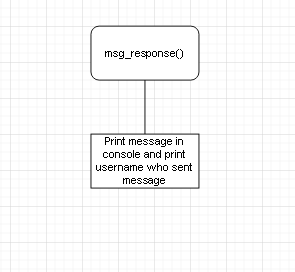
Dijagram datagram\_types



Dijagram reg\_response

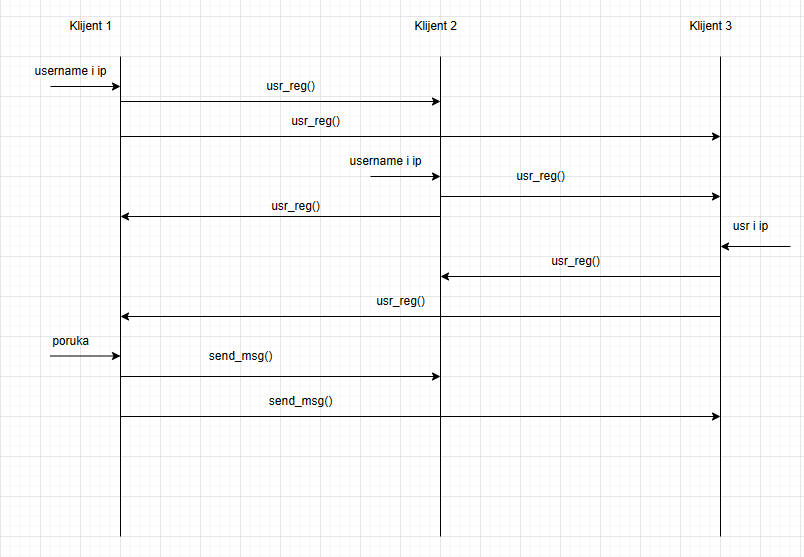


Dijagram user\_response

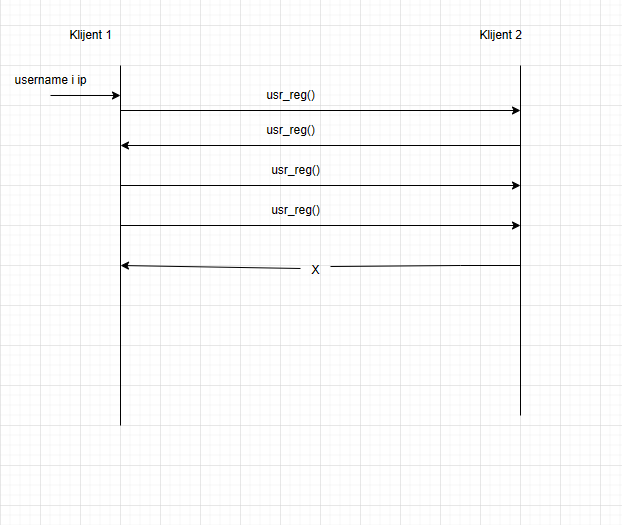


Dijagram msg\_reponse

# MSC dijagrami



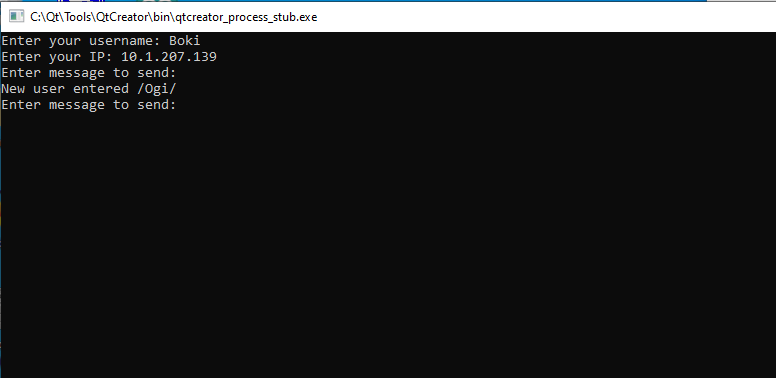
MSC dijagram uspesne komunikacije



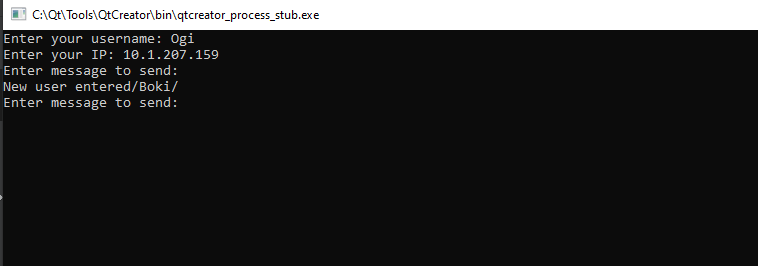
MSC dijagram neuspele komunikacije

# Testiranje

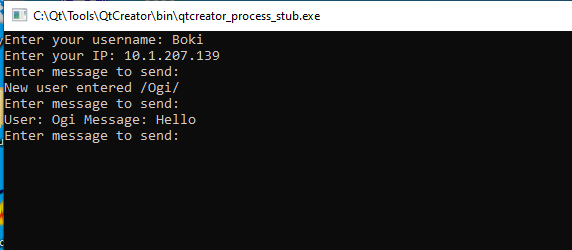
Program se pokreće na dva ili više računara u nekoj mreži, u ovom slučaju je testiranje izvršeno na računarima u laboratoriji. Kada se program pokrene klijent 1unosi svoj username I ip adresu, I čeka pojavu nekog drugog klijenta. Klijent dva se pridružuje mreži I unosi svoj username I IP adresu. Oba korisnika dobijaju poruku da je drugi klijent na mreži. Klijent dva šalje broadcast poruku I klijent 1 to dobija. Nakon nekog vremena klijent 2 izlazi iz mreže, I na broadcast se šalje poruka da je klijent 2 napustio mrežu.



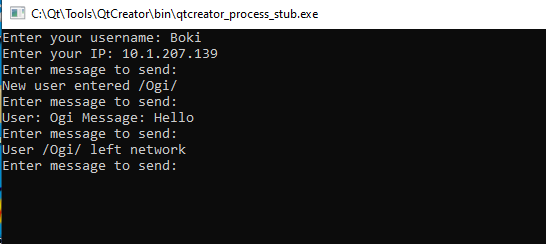
Klijent Boki dobija obaveštenje da je Ogi novi user u mreži



Klijent Ogi dobija obaveštenje da je Boki novi user u mreži



User Ogi je poslao broadcast poruku Hello, Boki dobija tu poruku



Obaveštenje da je Ogi napustio mrežu