

Projektni zadatak

- maj 2019 -

Za potrebe kompanije MDP potrebno je napraviti jednostavan softver za podršku zaposlenima. Zaposleni rade na različitim lokacijama, a za uspješan rad potrebno je da budu povezani sa centralom. Zbog toga je potrebno napraviti nekoliko aplikacija koje će se koristiti u poslovanju.

Svaki zaposleni na svom računaru treba da koristi *desktop* aplikaciju. Prilikom pokretanja aplikacije prvo se otvara dijalog (*popup*) u kojem korisnik treba da unese korisničko ime i lozinku. Ukoliko su podaci ispravni, otvara se glavni prozor aplikacije u kojem se nalazi meni sa opcijama za zatvaranje aplikacije (odjava sa sistema), pregled korišćenja aplikacije i promjena lozinke, dok se u centralnom dijelu aplikacije nalaze opcije za *chat*. Sa lijeve strane centralnog prostora nalazi se spisak korisnika aplikacije, a u sredini se nalazi prostor za razmjenu poruka. *Chat* može biti između dvije osobe ili između svih zaposlenih u kompaniji. Na desnoj strani glavnog prostora nalazi se forma za slanje fajlova određenom korisniku koji se bira iz *combo-boxa*, i pregled svih fajlova koje je prijavljeni korisnik dobio od drugih korisnika, a koje može preuzeti na svoj računar. Ispod menija, a iznad ostalih dijelova aplikacije, korisnici imaju liniju za dobijanje obavještenja od administratora. Prilikom pregleda korišćenja aplikacije, korisniku se u novom prozoru prikazuje tabela sa podacima o prijavama na sistem (vrijeme prijave, vrijeme odjave i ukupno aktivno vrijeme za tu sesiju kao razlika ova dva vremena).

Centrala kompanije ima svoju *desktop* aplikaciju koju koriste administratori. Ova aplikacija ima listu svih trenutno aktivnih korisnika, pregled istorije korisnika (ista statistika kao kod korisnika za pregled korišćenja aplikacije) koji se aktivira izborom jednog aktivnog korisnika i opciju za gledanje monitora izabranog korisnika. Kada se aktivira opcija za gledanje monitora jednog izabranog korisnika, tada se u novom prozoru prikazuje slika u realnom vremenu. Korisnik ne smije znati da ga administrator gleda, a u jednom trenutku moguće je gledati samo jednog korisnika. Pored toga, aplikacija ima i polje za unos teksta koje se šalje svim korisnicima kao obavještenje koje će se kod njih prikazati ispod menija. Administratori imaju mogućnost kreiranja novih korisničkih naloga, kao i blokade aktivnih naloga nakon čega se korisnici više ne mogu prijaviti na sistem.

Opcije za prijavu na sistem, odjavu, pregled statistike korišćenja i promjenu lozinke dostupne su preko REST servisa. Svi podaci se čuvaju u *in-memory* bazi podataka (npr. Redis). Dozvoljeno je napraviti skriptu ili program koji popunjava početni skup podataka kako bi se olakšalo testiranje sistema. Osim toga, u *in-memory* bazi podataka čuvaju se podaci o korisničkim nalogima. Kreiranje novih korisničkih naloga i blokada postojećih naloga obavlja se pomoću SOAP servisa, koji su zaduženi da u istoj bazi podataka ažuriraju potrebne podatke.

Chat funkcionalnost obavlja se preko *Socket* servera na siguran način. U okviru istog servera potrebno je implementirati opcije za pojedinačnu razmjenu tekst poruka između dva korisnika i za grupnu razmjenu poruka. Nije dozvoljeno uspostavljanje direktnih veza između korisnika.

Mehanizam za nadgledanje monitora zaposlenih potrebno je implementirati pomoću *Socket*-a za šta se može iskoristiti *chat* server. Voditi računa o količini prenesenih podataka u ovom procesu i po potrebi iskoristiti mehanizme komprimovanja podataka.

Čuvanje, prikaz informacija i preuzimanje fajlova potrebno je implementirati kao RMI aplikaciju. Fajlovi svih korisnika čuvaju se na serverskom dijelu aplikacije, tj. u okviru centrale. Potrebno je definisati strukturu i način čuvanja fajlova kako bi sistem bio efikasan. Korisničke aplikacije zaposlenih mogu se posmatrati i kao RMI klijentski dio sistema.

Slanje obavještenja potrebno je implementirati kao *multicast*. Svi korisnici dobijaju i prikazuju poruke. Svako dobijeno obavještenje se serijalizuje na fajl sistemu klijenta. Potrebno je implementirati najmanje 4 različita načina serijalizacije, pri čemu se svaka poruka serijalizuje na drugi način (1. poruka – 1. način, ... 4. poruka – 4. način, 5. poruka – 1. način...).

Sve parametre koji se koriste u GUI aplikacijama i drugim programima potrebno je definisati u odgovarajućim *properties* fajlovima. Za obradu izuzetaka potrebno je koristiti *Logger* klasu.

