

## Projektni zadatak

- maj 2022 -

Za potrebe države potrebno je napraviti jednostavan sistem za evidenciju i kontrolu putnika prilikom prelaska preko granice. Zbog toga je potrebno napraviti nekoliko aplikacija koje će se ovom prilikom koristiti.

U sklopu države postoji više carinskih terminala, od kojih svaki terminal ima određen broj ulaza i izlaza iz države sa identifikatorom terminala (cjelobrojna vrijednost koja je jedinstvena na nivou cijelog sistema) i naziv. Svaki ulaz i izlaz imaju carinsku i policijsku kontrolu. Terminali se dodaju, ažuriraju, brišu i dohvataju putem administratorske aplikacije. Podaci o svim terminalima čuvaju se binarno serijalizovani u centralnom registru (svaki terminal u odvojenom fajlu pri čemu je ime jedinstveno). Prilikom serijalizacije potrebno je implementirati najmanje četiri načina serijalizacije (1. terminal– 1. način, ... 4. terminal– 4. način, 5. terminal– 1. način...). Sva komunikacija administratorske aplikacije sa centralnim registrom vrši se putem SOAP poruka. Za svaki ulaz ili izlaz na terminalu (policijski i carinski), može da se pokrene odvojena klijentska aplikacija. Prilikom pokretanja klijentske aplikacije izvršava se prijava. Unosi se id ulaza/izlaza, bira da li je policijski ili carinski i naziv carinskog terminala. Ukoliko terminal postoji na centralnom registru (provjera putem SOAP-a) korisnik unosi korisničko ime i lozinku. Korisničko ime i lozinka se čuvaju u *in-memory* bazi podataka i administrišu iz administratorske aplikacije putem REST servisa. Provjera korisničkog imena i lozinke se takođe vrši putem REST servisa. Između klijentskih aplikacija postoji *chat* komunikacija. Klijentska aplikacija ima mogućnost slanja *multicast* poruka svim ulazima/izlazima u sklopu terminala, slanja *broadcast* poruka svim ulazima/izlazima na svim terminalima i slanje poruke određenom ulazu/izlazu određenog tipa na određenom carinskom terminalu. Korisnički interfejs za komunikaciju osmisli samostalno. *Chat* funkcionalnost obavlja se putem *Socket* servera na siguran način. Nakon prijave u klijentsku aplikaciju, u glavom prozoru aplikacije nalazi se meni sa opcijama za zatvaranje aplikacije (lokalne odjave sa sistema), promjenu lozinke i te prikaz evidencije provjerenih putnika (evidencija o provjerenim putnicima se čuva u centralnom registru u tekstualnoj datoteci).

Iz posebne aplikacije simulira se pristup ulazu ili izlazu na određenom carinskom terminalu. Nakon unosa naziva terminala, odabira da li je u pitanju ulaz ili izlaz (sa navođenjem identifikatora) i provjere da li terminal postoji putem SOAP poruka, nastavlja se sa simulacijom. Prvo se pristupa policijskom, a zatim carinskom dijelu terminala. Komunikacija sa policijskim i carinskim dijelom vrši se putem SOAP poruka. Prilikom pristupa policijskom dijelu vrši se provjera identifikatora osobe koja prelazi na način da se putem RMI poruka kontaktira centralni registar potjernica. Ukoliko je osoba na potjernici potrebno je poslati poruku svim ulazima i izlazima da je na snazi privremeno zatvaranje graničnog prelaza dok se osoba ne procesira. Pod procesiranjem se misli da se na tom policijskom ulazu odabere notifikacija da je moguće nastaviti rad na svim drugim prelazima na tom carinskom terminalu. Notifikacija se šalje putem soketa. Ukoliko je osoba prošla policijski dio u mogućnosti je da na carinskom dijelu priloži dokumente koji se čuvaju na fajl serveru za identifikator datog korisnika i koji se iz carinskog dijela aplikacije šalju na fajl server putem RMI poruka. Potrebno je definisati strukturu i način čuvanja fajlova kako bi sistem bio efikasan. Voditi računa o količini prenesenih podataka u procesu slanja dokumenata i po potrebi iskoristiti mehanizme komprimovanja istih. U administratorskoj aplikaciji moguće je preuzeti evidenciju svih detektovanih osoba sa potjernica i carinskih dokumenata, putem REST servisa. Evidencija detektovanih osoba sa potjernica se čuva i ažurira na centralnom registru u XML formatu.

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. Dozvoljeno je napraviti skriptu ili program koji popunjava početni skup podataka kako bi se olakšalo testiranje sistema.

