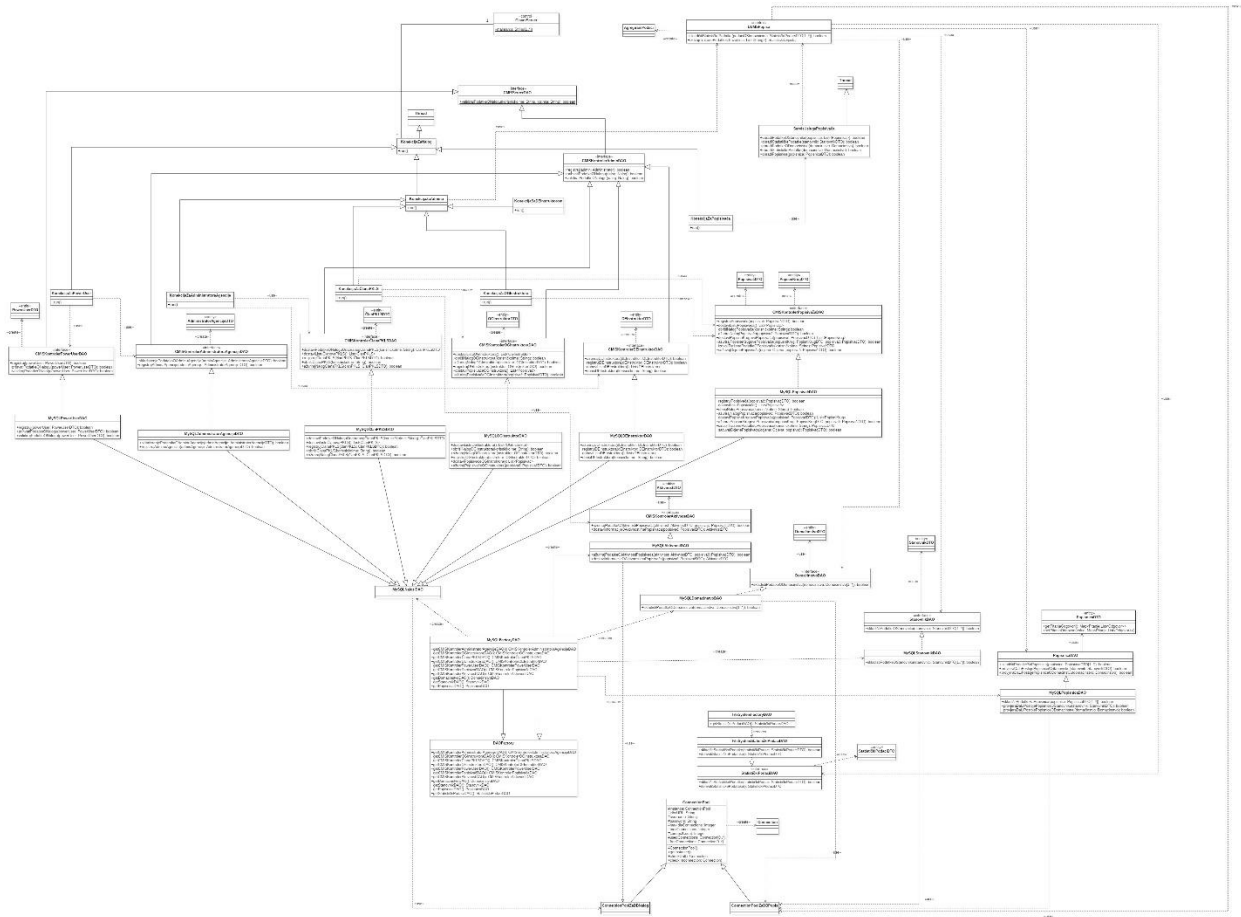


## Organizacija servera



## Opis dijagrama klasa za organizaciju servera

Dijagram klasa Organizacija servera predstavlja klase i veze između klasa kojim se omogućava povezivanje korisnika sa serverom, obradu i validiranje podataka kao i njihovo perzistentno čuvanje u bazi podataka i na fajl sistemu.

Dijagram obuhvata sljedeće:

- Uspostavljanje konekcije sa korisnicima cjelokupnog sistema
- Registrovanje, izmjena i brisanje naloga
- Obrada i validiranje popisnica na osnovu kojih se kreiraju podaci o stanovnicima i domaćinstvima
- Kreiranje statističkih izvještaja i statističkih podataka
- Evidentiranje aktivnosti popisivača
- Perzistentno čuvanje relevantnih podataka na fajl sistem i u bazama podataka.

Klasa GlavniServer prihvata konekcije od korisnika i na osnovu tipa korisnika kreira odgovarajuću instancu podklasu klase KonekcijaZaNalog. Te podklase su KonekcijaZaPopisivača, KonekcijaZaPowerUser-a i KonekcijaZaAdmin-a.

KonekcijaZaNalog omogućava konekciju odgovarajućeg korisnika sistema sa serverom i prihvata njegove metode sa podacima i izvršava ih.

KonekcijaZaPowerUser predstavlja uspostavljenu konekciju PowerUser korisnika sa serverom i omogućava mu registrovanje drugih PowerUser-a i Administratora agencije uz pomoć klasa CMISKontrolerPowerUserDAO i CMISKontrolerAdministartoraAgencijeDAO.

KonekcijaZaAdmin koja je takođe apstraktna klasa, koju nasljeđuju klase KonekcijaZaAdministratorsAgencije, KonekcijaZaČlanaPKLS, KonekcijaZaOGInstruktora i KonekcijaZaDEInstruktora, omogućava odgovarajućim administratorima uvid u statističke izvještaje.

KonekcijaZaAdministratorsAgencije omogućava administratoru agencije da upravlja sa nalogima članova PKLS i DE instruktora uz pomoć instance klasa MySQLCMISKontrolerČlanPKLSDAO i MySQLCMISKontrolerDEInstruktorDAO koje respektivno implementiraju interfejs CMISKontrolerČlanPKLSDAO i CMISKontrolerDEInstruktorDAO.

KonekcijaZaČlanaPKLS, KonekcijaZaOGInstruktora i KonekcijaZaDEInstruktora omogućava odgovarajuće funkcionalnosti datim korisnicima na sličan način kao i što je to implementirano kod klase KonekcijaZaAdministratorsAgencije.

KonekcijaZaPopisivača omogućava popisivačima da pošalju popisnice na obradu serveru. Poslane popisnice se obrađuju uz pomoć klase ServisUslugaPopisivača i nakon toga se validiraju i smještaju u bazu podataka uz pomoć klase DBMSPopisa.

DAOFactory predstavlja kreacioni obrazac AbstractFactory koja obezbjeđuje interfejs za kreiranje klasa MySQLFactoryDAO i FileSystemDAO za perzistentno čuvanje podataka u bazu i na fajl sistem.

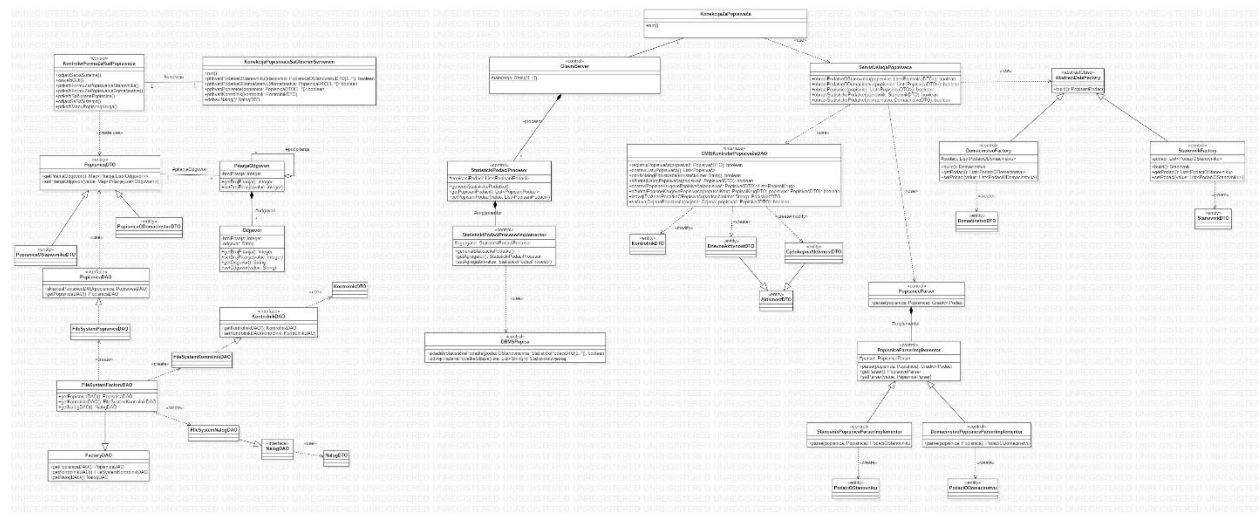
Podaci o stanovništvu, domaćinstvu, popisnicama i aktivnostima popisivača se čuvaju u bazama podataka uz pomoć klasa MySQLStanovnistvoDAO, MySQLDomacinstvoDAO, MySQLPopisnicaDAO i MySQLAktivnostDAO respektivno.

ConnectionPool koja je nadklasa klasama ConnectionPoolZaDBNalog i ConnectionPoolZaDBPopis predstavlja kreacioni projektni obrazac Singleton i omogućava konekciju ka odgovarajućim bazama.

ConnectionPoolZaDBNalog omogućava konekciju ka bazi za čuvanje naloga i aktivnosti popisivača.

ConnectionPoolZaDBPopis omogućava konekciju ka bazi za čuvanje podataka o stanovništvu, domaćinstvu, popisnicama i za generisanje statističkih podataka i izvještaja.

## Popisivanje



Dijagram klasa predstavlja klase i veze između klasa čiji se objekti koriste u procesu neposrednog popisivanja stanovništva i domaćinstava od strane popisivača.

- Kreiranje i upravljanje popisnicama
- Komunikaciju sa glavnim serverom i slanje popisnica
- Kreiranje odgovarajućih objekata za bazu na osnovu popisnica
- Ažuriranje aktivnosti popisivača
- Naknadnu statističku obradu za kreiranje rezultata popisa

KonekcijaPopisivačaSaGlavnimServerom predstavlja konekciju ka glavnom serveru. Pomoću ovih objekata se šalju popisnice ka glavnom serveru. Takođe šalje se i kontrolnik i dobavlja nalog. Metode `prihvatiPopisnice()` i `prihvatiKontrolnik()` se odnose na nakandno slanje popisnica i kontrolnika u slučaju nestanka interneta u procesu popisa. U međuvremenu podaci se čuvaju lokalno na fajl sistemu sto je predstavljeno šemom klasa ispod. Lokalno čuvanje je predstavljeno klasama `FileSystemPopisnicaDAO`, `FileSystemKontrolnikDAO`, `FileSystemNalogDAO`, `FileSystemFactoryDAO` i ostalima u tom dijelu klasnog dijagrama.

Glavni server sa klijentskom aplikacijom komunicira preko KonekcijaZaPopisivača klase koja apstrahuje dio konekcije izmjeđu klijenta i servera na strani servera. Objekti ove klase indirektno još i uslužuju zahtjeve klijenta (popisivača) tako sto delegiraju poziv klasi ServisUslugaPopisivača.

Kreiranje odgovarajućih objekata je predstavljeno AbstractDataFactory i podklasama. One kreiraju objekte klasa Stanovnik i Domaćinstvo.

Parser klasa predstavlja pripremu podataka za AbstractDataFactory klasu. Objekti ove klase parsiraju popisnice i popunjavaju ih dodatnim sadržajem. Parser klasa je predstavljena bridge strukturnim obrascem. Metodu build() nad DataFactory klasama poziva KonekcijaZaPopisivača na strani glavnog servera.

Ažuriranje podataka na CMIS serveru se predstavlja interfejsom CMISKontrolerPopisivačaDAO (objektom neke njegove potklase) koji kreira DnevnaAktivnostDTO, kreira ili ažurira CjelokupnaAktivnostDTO i ažurira KontrolnikDTO na serversoj strani.

Procesor klasa služi naknadnoj statističkoj obradi nakon završetka popisa. Inicira se od strane glavnog servera. Predstavljen je bridge strukturnom obrascem zbog potencijalnog proširenja i izmjena po pitanju koji agregatni podaci se trebaju generisati. Procesor klasa samo specifikuje upite da se izvoje podaci iz baze i prosljeđuje ih DBMS-u popisa na izvršavanje.

[illegible]

Administrator kao i svaki drugi korisnik sistema može da bira jezik i pismo svoje aplikacije.

Statistički izvjestaji predstavljaju apstrakciju svih konkretnih tipova statističkih izvjestaja. Prvo se mogu podjeliti po kategorijama, pa na konkretne što je prikazano hijerarhijom nasljeđivanja.

Napomena: Da bi obezbjedili gorenavedeni kriterijum „istih dnevnih aktivnosti“ potrebno je

Bilo koji statistički izvještaj sadrži jedan ili više statističkih podataka. Statistički podaci mogu biti agregatni i popisani podaci. Agregatni podaci predstavljaju neke izračunate vrijednosti kao što je procenat, broj itd, dok popisani podaci su podaci dobijeni direktno na osnovu popisnica. Agregatni podaci sadrže mapu koja sadrži parove ključ-vrijednost, gdje ključ predstavlja ime podatka i gdje vrijednost predstavlja brojčanu vrijednost podatka.

Statistički podaci mogu biti stanovnik i domaćinstvo.

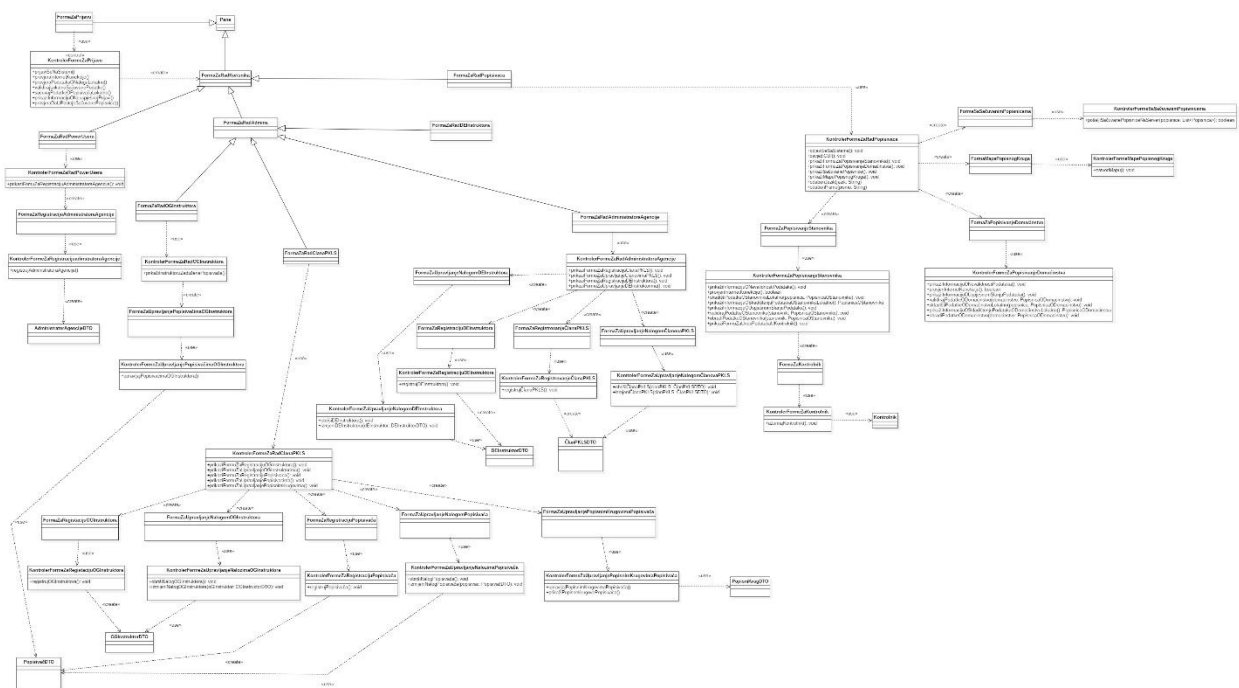
Domaćinstvo sadrži osnov korištenja stana realizovan enumeracijom. Sastoji se iz stana o kojem ćemo malo kasnije. Domaćinstvo može biti porodično i neporodično. Porodično sadrži jednu ili više porodica. Porodice imaju svoj tip i predstavljaju listu pojedinih članova domaćinstva. Član domaćinstva može biti porodica ili pojedinac. Neporodično domaćinstvo može biti samačko (gdje postoji jedan pojedinac) i višečlano (gdje postoji 2 ili više pojedinaca). Domaćinstvo sadrži apstraktnu metodu `getClanovi()`, koja će vratiti člana domaćinstva. U podklasi porodično domaćinstvo `getClanovi()` će vratiti listu svih porodica koje će biti referencirane klasom član domaćinstva, dok kod samačkog i višečlanog domaćinstva će vratiti listu pojedinaca

referencirane isto clanom domacinstva, čime dobijamo polimorfno ponašanje svih vrsta domaćinstava.

Stan predstavlja klasu koja je u sklopu domaćinstva i ima dosta svojih atributa koji su realizovani enumeracijama i jednostavnim klasama. Izuzetak je atribut zgrada koja sadrži više atributa, ali to ne mjenja činjenicu da je i to jednostavna klasa.

Svi atributi sadrže setere i getere za vanjski pristup istim.

## Forme i kontroleri za korisnike



Dijagram prikazuje hijerarhiju formi i kontrolera sa kojima interaguju korisnici sistema. Sve forme su generalizovane klasom FormaZaKorisnika. Svaka konkretna forma ima svoj kontroler. Kontroleri kreiraju nove forme ukoliko je to potrebno. Dijelovi dijagrama su kreirani po istom principu, pa je dovoljno objasniti jedan dio. Npr. FormaZaRadOGInstruktora koristi svoj kontroler kako bi kreirala novu formu FormaZaUpravljanjePopisivacimaOGInstruktora koja opet koristi svoj kontroler. Organizacija korisnika je prikazana na sljedecem dijagramu radi preglednosti.

[illegible]

Svi korisnici su generalizovani klasom KorisnikSistema. OGINstruktor mora da ima informaciju o opštini ili gradu za koji je zaduzen, pri cemu se opština ili grad posmatraju na isti nacin, pa je zbog toga iskoristen Composite projektni obrazac. Ista situacija je kod DEInstruktora za drzavu ili entitet. Na dijagramu su date i informacije o klasama koje su u vezi sa popisnim krugovima za koje su zaduzeni popisivaci.



**Zaduzenja:**

1. Popisivanje – David Baralic
2. Pracenje popisa i statisticki podaci – Kristijan Stepanov i Risto Maric
3. Organizacija servera – Sasa Sukalo i Sinisa Sukalo
4. Organizacija korisnika i forme i kontroleri za korisnike – Marijana Zeljkovic