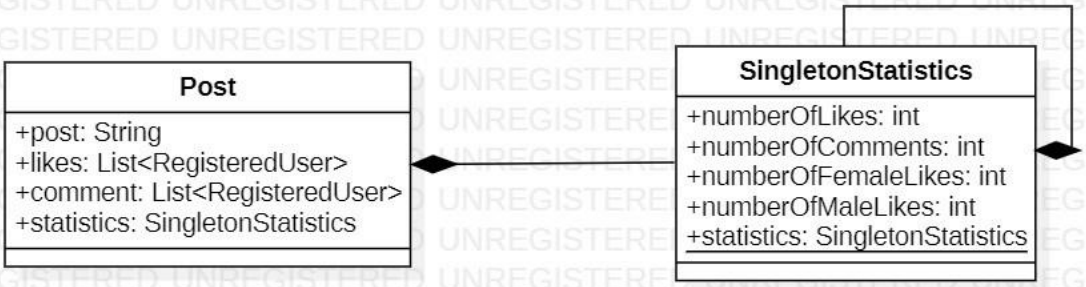
KREACIJSKI PATERNI

Singleton patern

Uloga ovog paterna je da osigura da se klasa samo jednom instancira i da ima globalni pristup prema njenoj instanci.

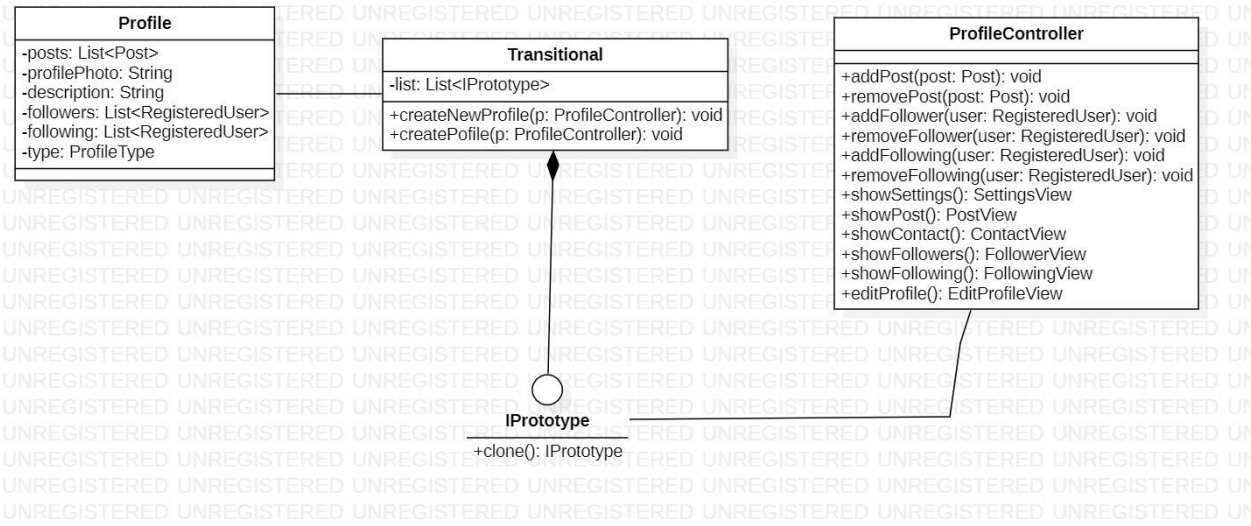
U našem sistemu klasa Statistics će biti implementirana kao singleton klasa.



Prototype patern

Uloga ovog paterna je da klonira objekte koji zauzimaju mnogo resursa.

Ovaj patern možemo primijeniti na način da omogućimo kloniranje instance objekta Profile kao objekta koji zauzima mnogo resursa.



Factory method patern

Uloga ovog paterna je da omogući kreiranje objekata na način da se odluči koja se podklasa instancira.

Ovaj patern bismo u našem sistemu mogli postići ako bismo omogućili da profile pored ljudi mogu imati i životinje. Za realizaciju bismo pri registraciji korisnika trebali uvesti neki atribut kojim bismo omogućili da korisnik napravi profil svom kućnom ljubimcu.

Abstract Factory

Uloga ovog paterna je da omogući kreiranje familije povezanih objekata.

U našem sistemu smo već primijenili ovaj patern tako što smo napravili klasu RegisteredUser koja će posjedovati metode koje će moći koristiti sve naslijeđene klase.

Builder patern

Uloga ovog paterna je odvajanje specifikacije kompleksnih objekata od njihove stvarne konstrukcije. Isti konstrukcijski proces može kreirati različite reprezentacije.

Gledajući dijagram klasa za naš sistem primjećujemo da nema glomaznih i komplikovanih klasa gdje bismo mogli primijeniti builder patern. Ako bismo uveli funkcionalnost da korisnik može pratiti aktivnost na sistemu (gdje i kad je šta lajkao, komentarisao, pregledao) ovaj patern bi nam bio od velike pomoći.