

## KREACIJSKI PATERNI U PROJEKTU "CHARITY FOUNDATION"

### 1. Singleton Pattern

**Uloga:**

Osigurava da se određena klasa instancira samo jednom i da se ta jedina instanca koristi globalno u sistemu.

**Primjena u projektu:**

- **EmailService** – koristi se za slanje obavještenja donatorima i korisnicima, prilikom verifikacije registracije, potvrde donacije i raspodjele pomoći. Singleton osigurava da postoji samo jedna instanca koja kontroliše pristup SMTP serveru i logiku slanja.
- **Logger** – može se koristiti za centralizovano logovanje grešaka prilikom alokacije donacija, generisanja izvještaja i neuspjelih transakcija.

### 2. Prototype Pattern

**Uloga:**

Omogućava kloniranje postojećih objekata kako bi se kreirali novi, bez direktnog pozivanja konstruktora.

**Primjena u projektu:**

- **Korisnički profili korisnika u potrebi** – prilikom registracije korisnika, možemo koristiti prototip šablona (npr. default obrazac sa poljima za potrebe, kontakt, urgentnost) koji se klonira i popunjava stvarnim podacima korisnika.
- **Volonterske uloge** – moguće je imati osnovni prototip za volontera koji se kopira za nove prijave, uključujući osnovne zadatke i status.

### 3. Factory Method Pattern

#### Uloga:

Omogućava delegiranje odluke o instanciranju objekata podklasama, bez eksplicitnog navođenja konkretne klase.

#### Primjena u projektu:

- **Donacije različitih tipova (novčane, materijalne, usluge)** – koristi se DonationFactory koja na osnovu korisničkog izbora kreira odgovarajući tip donacije (MonetaryDonation, GoodsDonation, ServiceDonation).
- **Upiti korisnika** – sistem može imati MessageFactory koja generiše odgovarajuće objekte upita (TechnicalInquiry, HelpRequest, Complaint) za različite potrebe korisnika.

### 4. Abstract Factory Pattern

#### Uloga:

Omogućava kreiranje grupa povezanih objekata bez otkrivanja konkretnih klasa. Idealno za kreiranje porodica objekata.

#### Primjena u projektu:

- **Korisnički nalozi** – postoji apstraktna fabrika UserFactory koja kreira instance različitih korisnika: Donator, KorisnikUPotrebi, Volonter, Administrator. Svaka od klasa implementira zajednički interfejs i omogućava uniformnu manipulaciju korisnicima.
- **Notifikacijski sistem** – putem NotificationFactory koja generiše EmailNotification, SMSNotification, SystemNotification u zavisnosti od postavki korisnika i vrste događaja.

### 5. Builder Pattern

#### Uloga:

Služi za izgradnju kompleksnih objekata korak po korak, posebno kada su neki atributi opcionog karaktera.

#### Primjena u projektu:

- **Filter za raspodjelu donacija** – koristi se DonationFilterBuilder za kreiranje filtera sa opcijama kao što su: urgentnost, kategorija pomoći, lokacija korisnika, vrijednost pomoći. Builder omogućava jednostavno proširenje bez potrebe za dodatnim konstruktorima.
- **Generisanje izvještaja** – ReportBuilder omogućava administratoru da izgradi izvještaj definisanjem perioda, tipa donacija, korisničke grupe i kriterijuma raspodjele.