Objektni dizajn

Smart ZEV



Mrđan Poletanović

Dimitrije Kučuk

Bojan Bulatović

Darko Prelić

Slaviša Stojaković

ETF BANJA LUKA 2020.

Sadržaj

[1. Uvod 3](#_Toc44877381)

[2. Podsistem za rad sa korisnicima 3](#_Toc44877382)

[3. Podsistem za rad sa podacima 4](#_Toc44877383)

[4. Podsistem za aplikativnu i poslovnu logiku 6](#_Toc44877384)

[5. Podsistem za rad sa ZEV-ovima 6](#_Toc44877385)

[6. Podsistem za GUI i forme 6](#_Toc44877386)

[7. Tabelarni prikaz odgovornosti 8](#_Toc44877387)

# Uvod

Sistem Smart ZEV je podijeljen na 5 manjih cjelina (podsistema), i svaki član tima je dobio zadatak da projektuje svoj podsistem.

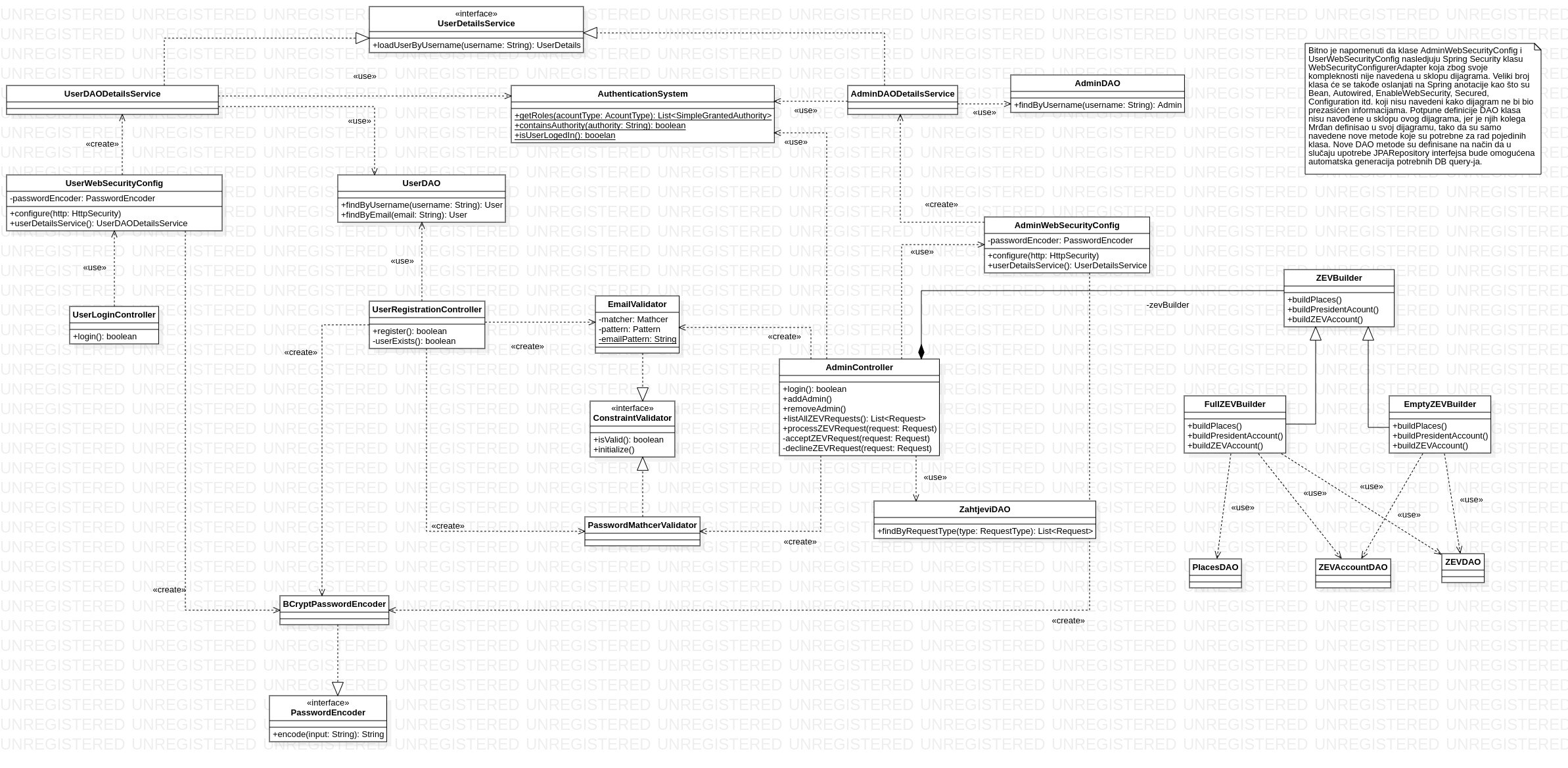
Sistem se sastoji iz sledećih podsistema:

1. Podsistem za rad sa korisnicima
2. Podsistem za rad sa podacima
3. Podsistem za aplikativnu i poslovnu logiku
4. Podsistem za rad sa ZEV-ovima
5. Podsistem za GUI i forme

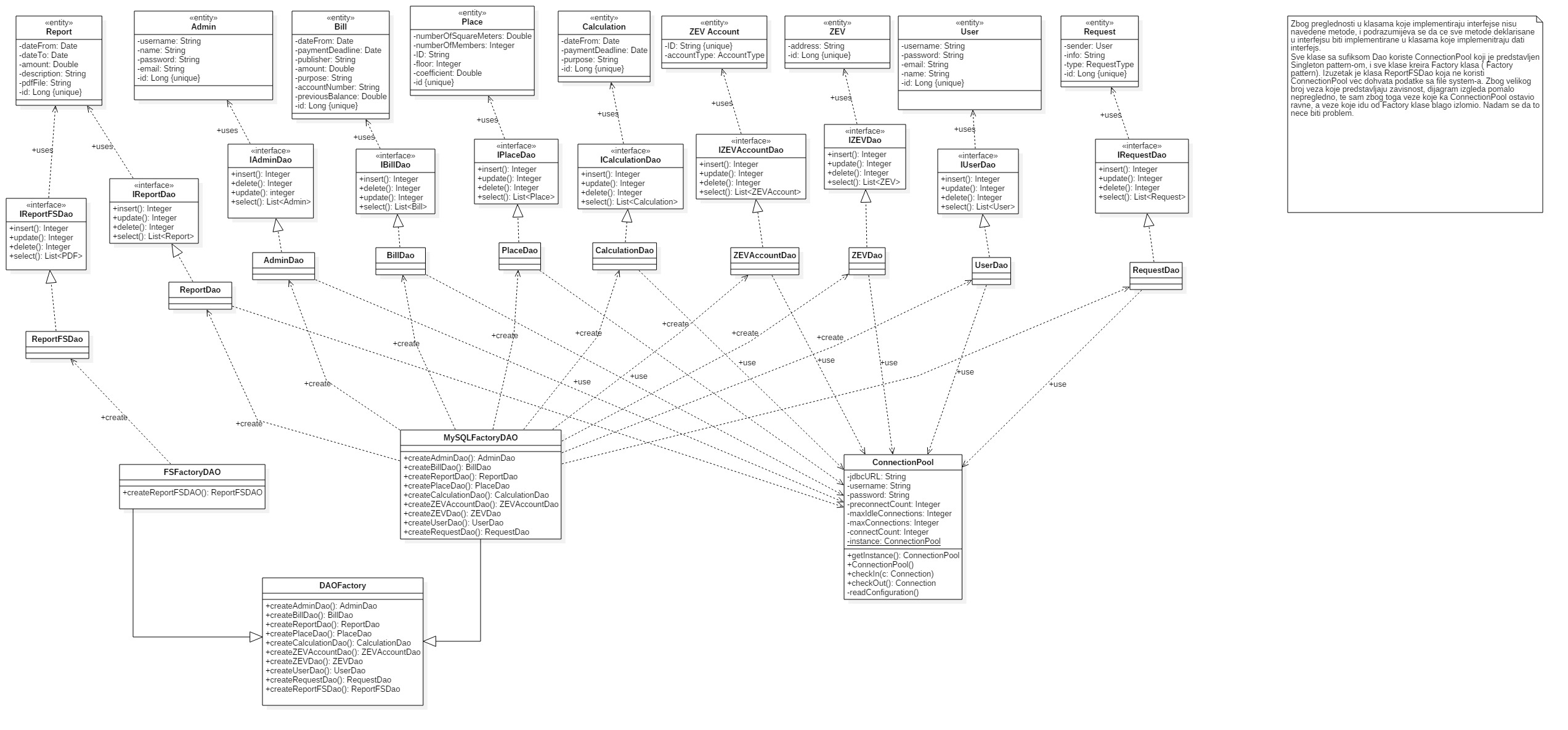
U nastavku teksta biće opisano šta to sve podrazumijevamo pod gore navedenim podsistemima, dijagram klasa za svaki podsistem, dodatna objašnjenja, te naziv autora (člana grupe koji je projektovao dati podsistem). Svi dijagrami biće dostavljeni i u formi slike uz ovaj PDF dokument, kako bi preglednost bila zadovoljena.

# Podsistem za rad sa korisnicima

Komponenta za rad sa korisnicima obuhvata neophodne klase i logiku potrebne za sigurnu registraciju i prijavu korisnika na sistem, kao i rad sa administratorskim nalozima i kreiranjima novh predsjednickih naloga i ZEV-ova. Kako bi se obezbijedila sigurna autentikacija i autorizacija operacija, komponenta se uveliko oslanja na postojeće klase iz Spring Security paketa. Za realizaciju kreiranja novog ZEV-a upotrebljen je Builder kreacioni obrazac.

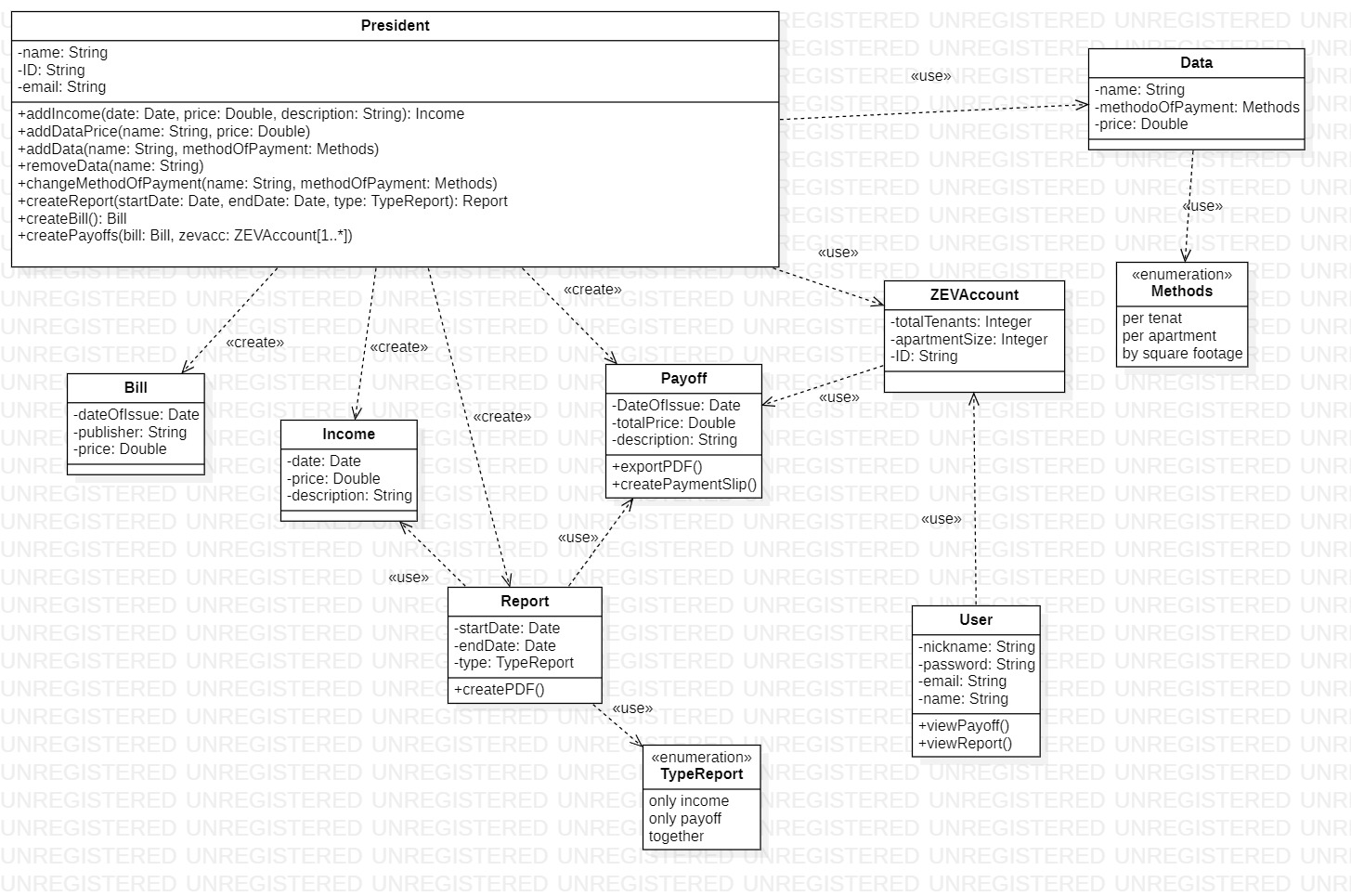


# Podsistem za rad sa podacima

Pod podsistemom za rad sa podacima podrazumijeva se infrastruktura neophodna za komunikaciju sa bazom podataka, za dohvatanje, čuvanje i izmjenu podataka bitnih za sistem Smart ZEV. Svaka klasa koja se koristi u sistemu i čiji podaci se istovremeno nalaze u bazi podataka ima svoj odgovarajući DTO objekat (entitet), DAO interfejs sa metodama koje trebaju biti implementirane, te DAO klasu koja implementira DAO interfejs. Na ConnectionPool klasi je primjenjen kreacioni obrac Singleton. Pored tog kreacionog obrasca, primjenjen je i Factory pattern koji se koristi za kreiranje odgovarajućih DAO objekata. Dodatne napomene se nalaze u sklopu dijagrama klasa koji se nalazi u nastavku ovog dokumenta. 

# Podsistem za aplikativnu i poslovnu logiku

U podsistemu za aplikativnu i poslovnu logiku opisani su načini izdavanja računa, obračuna, izvještaja za korisnike aplikacije u pojedinim ZEV-ovima. Na osnovu konfigurisanih podataka predsjednik ZEV-a će kreirati račune iz kojih će kreirati obračune za svaki ZEV nalog. Na osnovu proteklih prihoda i obračuna predsjednik može i da kreira izvještaje za neki vremenski period. Svaki korisnik sa svoga ZEV naloga može da pregleda obračune vezane za njega.

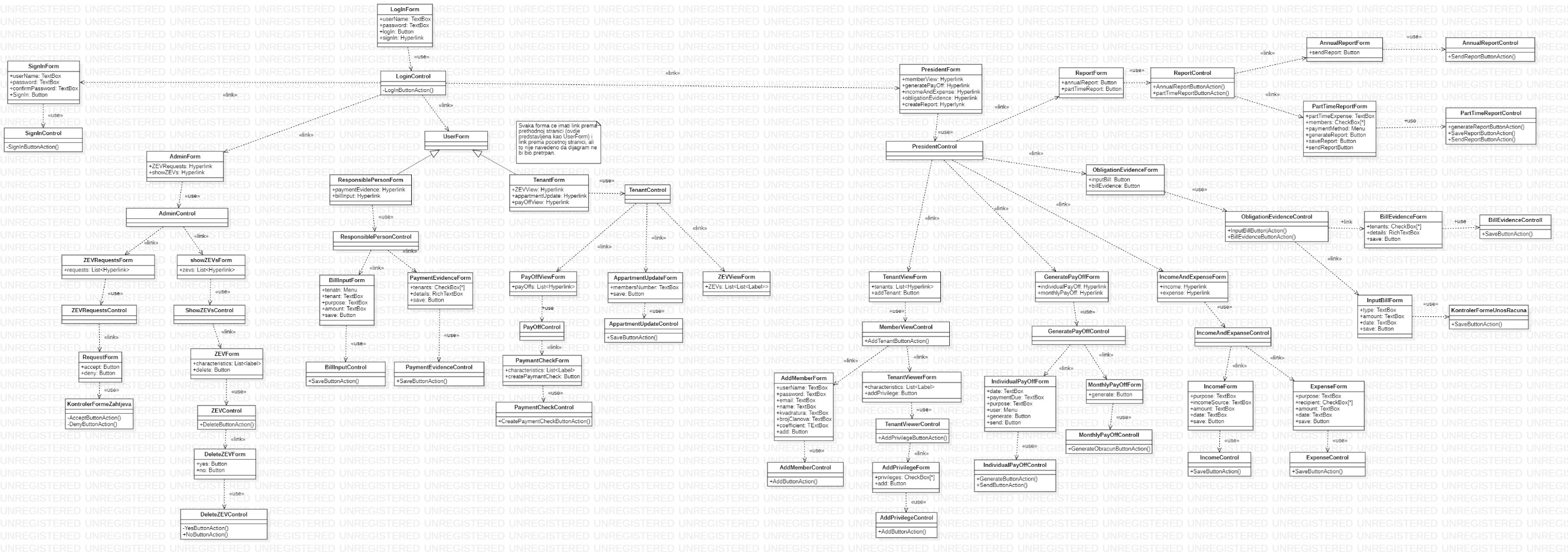
.

# Podsistem za rad sa ZEV-ovima

Kolega zadužen za ovaj podsistem iz privatnih razloga nije stigao odraditi svoj dio posla do navedenog roka.

# Podsistem za GUI i forme

Na sledećem dijagramu su prikazane forme i kontroleri potrebni za korisnički grafički interfejs. Takođe, prikazani su samo osnovni atributi sa kojima korisnik interaguje i na osnovu kojih se vrše obračuni i dobijaju rezultati.



# Tabelarni prikaz odgovornosti

|  |  |  |
| --- | --- | --- |
| Redni broj | Autor | Tema |
| 1. | Mrđan Poletanović | Podsistem za rad sa podacima |
| 2. | Bojan Bulatović | Podsistem za rad sa korisnicima |
| 3. | Dimitrije Kučuk | Podsistem za aplikativnu i poslovnu logiku |
| 4. | Darko Prelić | Podsistem za rad sa ZEV-ovima |
| 5. | Slaviša Stojaković | Podsistem za GUI i forme |