

Objektni dizajn

Smart ZEV



Mrđan Poletanović

Dimitrije Kučuk

Bojan Bulatović

Darko Prelić

Slaviša Stojaković

ETF BANJA LUKA 2020.

Sadržaj

1. Uvod.....	3
2. Podsystem za rad sa korisnicima	3
3. Podsystem za rad sa podacima	4
4. Podsystem za aplikativnu i poslovnu logiku	6
5. Podsystem za rad sa ZEV-ovima	6
6. Podsystem za GUI i forme	6
7. Tabelarni prikaz odgovornosti	8

1. 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:

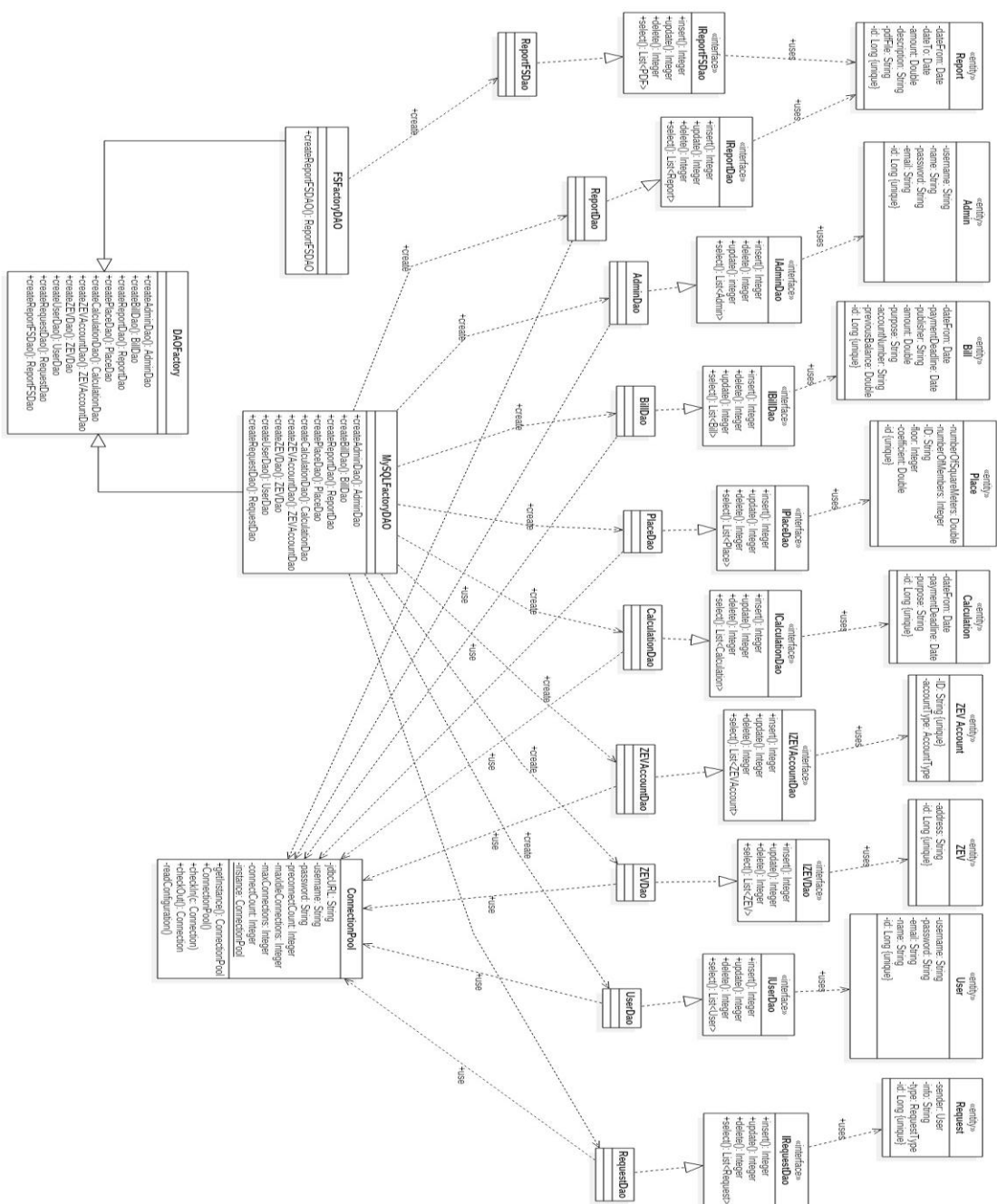
1. Podsistem za rad sa korisnicima
2. Podsistem za rad sa podacima
3. Podsistem za aplikativnu i poslovnu logiku
4. Podsistem za rad sa ZEV-ovima
5. Podsistem za GUI i forme

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.

2. 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 nalogima i kreiranjima novih 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.

dokumenta.



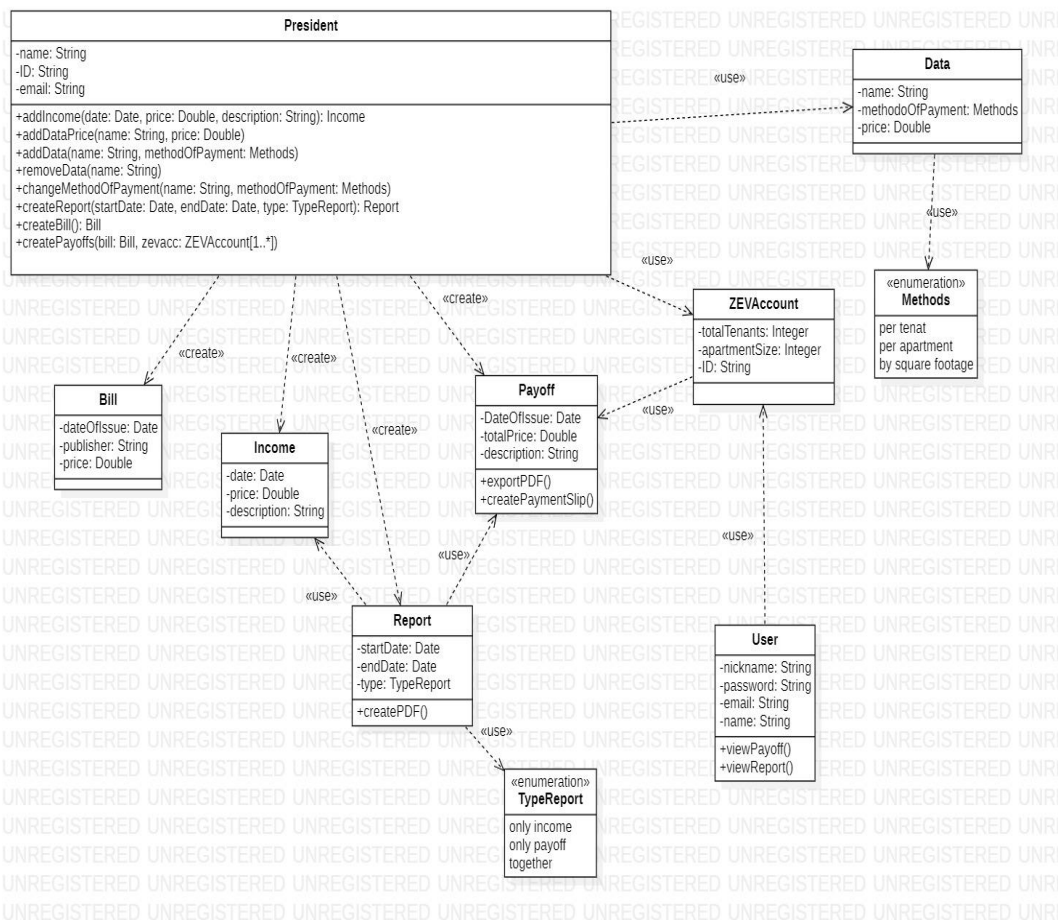
Zbog jednostavnosti u klasama koje implementiraju interfejs nisu navedene metode, podrazumijeva se da su sve metode deklarirane u interfejsu koji implementiraju u klasama koje implementiraju dati interfejs.

Sve klase su sačinom Dao, koriste ConnectionPool koji je predstavljen Singleton pattern-om, sve klase koriste Factory klasu (Factory pattern). Izuzetak je klasa ReportDAO koja ne koristi Zbog velike kompleksnosti i različitosti podataka sa sistemom Zbog velikog broja podataka koji se moraju obraditi i zbog toga što se ne može napraviti Singleton pattern, izostavljen je Singleton pattern za ReportDAO.

ConnectionPool se koristi kao Singleton, ali se ne koristi Singleton pattern, već se koristi Singleton pattern za ConnectionPool. Ovo je zato što ConnectionPool ostaje otvoren tokom cijelog života aplikacije, a ne treba ga zatvarati nakon svake upotrebe.

4. Podsystem za aplikativnu i poslovnu logiku

U podsystemu 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.



5. Podsystem za rad sa ZEV-ovima

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

6. Podsystem 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.

7. 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