

1. Pogledajte Sumejjin post o arhitekturi i kako importovati projekt. Vaš root folder je zubarska-ordinacija. Ovdje ćete vidjeti gdje koje klase stavljati.
2. Instalirajte MySQL ( Tokom dana će te dobiti MySQLDump file kojim ćete postaviti bazu)
3. Instalirajte Hibernate tools imate objašnjenje u Uputstvu. (možda nije nužno)
4. Objašnjenje arhitekture (pogledati dijagram paketa): Postoje 3 sloja :

a. DAL – Data Acces layer – Najniži sloj, prvi iznad baze.

i. Domain Modeli – ovo su POJO(SAMO privatni atributi i geteri i seteri, kao POCO u C#) klase koje ustvari predstavljaju tabele u bazi i služe da se preko Hibernate manipuluje podacima, ove klase služe samo za prijenos podataka, tj. da se podaci smjeste u njih ili u Listu njih i pošalju.

ii. U samom Paketu DAL se nalazi klasa HibernateUtil koju koristimo da pozovemo SessionFactory() i dobijemo Session koji nam je potreban

b. BLL – Business Layer Logic – Sloj iznad DAL-a

Ovaj sloj koristi DAL sloj, ali NE KORISTI prezentacijski UI sloj. U njemu se nalaze sve funkcije(funkcionalnosti) koje naš sistem nudi. Npr. dodajNovogPacijenta, nadjiSveTerminePoPacijentu, otkaziTermin, itd -> Ove klase su **Manager klase** i nalaze se u **BLL.Servisi**. Cilj je da ovaj Softver ne zavisi od Prezentacije, tj. da prezentacijski sloj može biti bilo što... GUI(Swing), Webstranica, Android aplikacija,..., koju samo „nakačimo“ na BLL iz kojeg dobivamo potrebne podatke koje prezentiramo.

Mi želimo da za svaku funkcijalnost pozovemo (samo jednu)metodu i da dobijemo informacije koje nam trebaju, koje ćemo prezentirati. Npr. ako želimo sve termine po pacijentu pozovemo nadjiSveTerminePoPacijentu(neki\_id) i hoćemo da nam ona vrati **podatke za prezentaciju**, ti podaci su ustvari naši **ViewModeli**. ViewModeli su opet POJO(Poco) klase koji služe kao i DomainModeli samo za prijenos podataka.

**Zaključak:** u Managerima(servisima) napunimo ViewModele i vratimo ih onom ko ih je pozvao(najčešće prezentacijskom sloju). Kako ih punim? Koristeći DAL sloj, tj. Hibernatom dohvatamo iz baze podatke, pravimo Query-e i punimo podatke.

c. (G)UI – Prezentacijski sloj – Najviši sloj - iznad BLLa

Kao što mu ime kaže služi (SAMO) za prezentaciju. Ovdje **ne smije** biti nikakve logike, osim one koja služi za prezentaciju (da se na osnovu unesenih podataka i klikova, pozove neka metoda i da se prikaže rezultat). U ovom sloju na Evente:

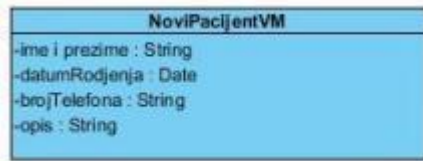
- i. uzimaju potrebni podaci
- ii. otvara konekcije pomoću HibernateUtil
- iii. Kreira Manager klasu u čiji konstruktor šaljemo konekciju
- iv. zatvaramo konekciju
- v. prikazujemo podatke korisniku

Zašto ovdje otvaramo konekciju koja je iz DAL sloja? Jer se je to najlakši način da najbolje iskoristimo uspostavu konekcije s bazom (da je otvaramo što manji broj puta, ako je moguće jednom, a da što kraće konekcija ostane otvorena)

**POGLEDATI DIJAGrame SEKVENCE (BIT ĆE JASNIJE, zanemarite malo transakcije da vas ne buni)**

## Kreiranje modela. (POJO klasa)

1. Iz UML dizajna iz Class dijagrama uzmete neki ViewModel. Npr.



2. Obavijestite raju da ste ga uzeli.
3. dodate privatne attribute

```
package ba.unsa.etf.si.tim12.bll.viewmodel;

import java.util.Date;

public class NoviPacijentVM {
    private String imeIPrezime;
    private Date datumRodjenja;
    private String brojTelefona;
    private String opis;
}
```

4. Zatimm da napravite **seter** i **getere**. Pustite Eclipse da to sam uradic: Source->Generate Getters & Setters...->Select All->OK
5. Čestitamo imate svoj prvi ViewModel