Vježba 13. / Zadaća 4. - Opis

Naziv: poslovna (enterprise) aplikacija za rad s bazom podataka putem ORM uz korištenje JPA te korisničku stranu s JSF na bazi predloška i korištenjem Ajax-a.

Enterprise aplikacija služi za rad s bazom podataka putem ORM uz korištenje JPA (bez SQL, koristi se Criteria API), uz korisničku stranu s JSF na bazi predloška, dodavanja i ažuriranje adresa, preuzimanje prognoza, pregled i filtriranje dnevnika, sve opreracije provode se Ajax-om. Enterprise aplikacija pod nazivom **{LDAP\_korisničko\_ime}\_zadaca\_4** sastoji se od tri modula:

1. **{LDAP\_korisničko\_ime}\_zadaca\_4\_1**: EJB modul za podatkovni sloj s JPA
2. **{LDAP\_korisničko\_ime}\_zadaca\_4\_2**: EJB modul za poslovnu logiku i korištenje EJB podatkovnog sloja
3. **{LDAP\_korisničko\_ime}\_zadaca\_4\_3**: Web modul za korisničko sučelje

Svaka akcija korisnika treba biti zapisana u tablicu dnevnik, a temelji se na aplikativnom filtru. Svi prikazi u padajućim izbornicima trebaju imati abecedni popis elemenata.

Postupak počinje unosom novog parkirališta (id, naziv, adresa) i pregledom postojećih parkirališta (u obliku padajućeg izbornika) iz ORM (tablica parkiralista). Dodavanje novog parkirališta (putem gumba Dodaj) Ajaxom pokreće preuzimanje lokacijskih podataka za upisanu adresu. Ako nije upisan ID tada se treba pronaći sljedeći broj u tablici. Slijedi spremanje objekta u ORM (upis u tablicu parkiralista) i ažuriranje pregleda raspoloživih parkirališta. Javi li se pogreška kod dodavanje treba ju ispisati u elementu za poruke ispod dijela za unos (pravokutnik crvene boje). Iz raspoloživih parkirališta može se odabrati jedno ili više parkiralište te (putem gumba Preuzmi) Ajaxom dodati u popis odabranih parkirališta. Sva parkirališta koja su preuzeta više se ne pojavljuje u pregledu raspoloživih parkirališta. Moguće je u popisu raspoloživih parkirališta odabrati jedno parkiralište i pokrenuti njegovo ažuriranje (putem gumba Ažuriraj) u istom pogledu. Podaci tog parkirališta se prikazuju u dijelu za unos podataka. Gumb Upiši do sada je bio sakriven i treba ga prikazati. Nakon ažuriranja podataka (putem gumba Upiši) ažurira se objekt u ORM (tablica parkiralista). Kod ažuriranje mora biti upisan ID inače je pogreška. Javi li se pogreška kod ažuriranja treba ju ispisati u elementu za poruke ispod dijela za unos (pravokutnik crvene boje). Nakon ažuriranja sakriva e gumb Upiši i vraća se na odabir parkirališta. Ako je pokrenuto dodavanje tada treba provesti postupak dodavanja novog parkirališta i sakriti gumb Upiši. Kod popisa odabranih parkirališta putem gumba Vrati može se maknuti jedno ili više parkirališta putem Ajaxa i tada se ono(a) ponovno pojavljuje(u) u pregledu raspoloživih parkirališta i miče se iz pregleda odabranih parkirališta. Nakon pokretanja prognoza (putem gumba Prognoze) putem Ajaxa, preuzimaju se prognoze vremena za sva odabrana parkirališta i 5 dana prognoze. Gumb Prognoze mijenja naziv u Zatvori prognoze. Zatim se prikazuju podaci prognoza (parkiralište, adresa, datum, temp,...) u bloku s tablicom (taj dio je prethodno bio sakriven) i nakon završetka pregleda prognoza (putem gumba Zatvori prognoze) cijeli blok se sakriva. Pregled dnevnika obavlja se u posebnom pogledu. Mogu se unijeti podaci za filtiranje na bazi: ip adrese korisnika, intervala vremena od-do, adrese zahtjeva i trajanja zahtjeva. Može se unijeti od ni jednog podatka pa do svih podataka. Nakon pokretanja pregleda (putem gumba Prikaži) prikazuju se filtirani podaci iz dnevnika. 

Klase i metode trebaju biti komentirane u javadoc formatu. **Projekt se isključivo treba predati u formatu NetBeans projekta.**Prije predavanja projekta potrebno je napraviti Clean na projektu. Zatim cijeli projekt sažeti u .zip (NE .rar) format s nazivom Prije predavanja projekta potrebno je napraviti Clean na projektu. Zatim cijeli projekt sažeti u .zip (NE .rar) format s nazivom **{LDAP\_korisničko\_ime}\_zadaca\_4.zip** i predati u Moodle. Uključiti izvorni kod, primjere datoteka konfiguracijskih podataka i popunjeni obrazac za zadaću pod nazivom **{LDAP\_korisničko\_ime}\_zadaca\_4.doc** (u korijenskom direktoriju projekta).

Naziv projekta: **{LDAP\_korisničko\_ime}\_zadaca\_4**

1. Kreiranje direktorija **{LDAP\_korisničko\_ime}\_zadaca\_4**. U nastavku se direktorij za vježbu simbolički označava kao**{vježba}**.
2. startati Java DB
3. kreirati Java DB bazu podataka (Services/Java DB/Create Database...) nwtis\_zadaca\_4, korisničko ime nwtis, lozinka foi2018 (Slika 1)
4. otvoriti vezu prema novoj bazi podataka
5. preuzeti sql za kreiranje tablica iz zadaće 3, kopirati sadržaj i izvršiti (u otvorenoj vezi na bazu podataka odabrati Tables/Execute Command...)
6. preuzeti sql za punjenje tablica iz zadaće 3, kopirati sadržaj i izvršiti (u otvorenoj vezi na bazu podataka odabrati Tables/Execute Command...)
7. preuzeti datoteku dnevnik\_JavaDB.sql, kopirati sadržaj i izvršiti (u otvorenoj vezi na bazu podataka odabrati Tables/Execute Command...)
8. provjeriti sadržaj sheme  (kao na Slici 2.)

**{LDAP\_korisničko\_ime}\_zadaca\_4\_1**: Kreiranje modula za podatkovni sloj s EJB: EntityBean - JPA

1. Kreiranje projekta **{LDAP\_korisničko\_ime}\_zadaca\_4\_1** kao Java EE/EJB Module , server Glassfish, Java EE verzija: Java EE 7
2. kreirati pakete **org.foi.nwtis.{LDAP\_korisnik}.ejb.eb**, **org.foi.nwtis.{LDAP\_korisnik}.ejb.sb**
3. kreirati perzistantnu jedinicu (New/Other/Persistence/Persistence Unit) . Slike 3, 4 i 5 prikazuju aktivnosti koje se provode: određivanje naziva: zadaca\_4\_1PU, kreiranje izvora podataka (Data source) pod nazivom jdbc/NWTiS\_2018 za ranije kreranu vezu prema bazi podataka nwtis\_zadaca\_4. Slika 6 prikazuje podatke kreirane perzistente jedinice. Slike 7 i 8 pokazuju kreirane datoteke opisnika (persistence.xml i glassfish-resources.xml)
4. kreirati shemu baze podataka (New/Other/Persistence/Database Schema). Naziv: NWTiS\_2018 (Slika 7) na **src/conf** i odabrati ranije kreranu vezu prema bazi podataka nwtis\_zadaca\_4 (Slika 8). Slijedi odabir svih tablica (Slika 9). Struktura sheme baze podataka prikazana je na slici 10. Služi nam kao pomoć i nije nužno to obavljati.
5. kreirati klase na entitete (New/Other/Persistence/Entity Classes from Database) na **org.foi.nwtis.{LDAP\_korisnik}.ejb.eb**. Slijedi odabir svih tablica s time da se može odabrati kao osnova izvor podataka jdbc/NWTiS\_2018 (Slika 11) ili shema baze podataka NWTiS\_2018 ako smo ju kreirali u projektu (Slika 12). Preslikavanja tablica baze podataka u klase prikazana je na slici 13. I na kraju se određuju elementi preslikavanja (Slika 14)
6. kreirati session bean-nove (Facade) za klase entiteta (New/Other/Persistence/Session Beans for Entity Classes) na **org.foi.nwtis.{LDAP\_korisnik}.ejb.sb**. Slijedi odabir klasa (Slika 15) za koje će se generirati fasade (Slika 16)

**{LDAP\_korisničko\_ime}\_zadaca\_4\_2**: Kreiranje modula za poslovnu logiku i korištenje EJB: SessionBean (Stateful, Stateless i Singleton)

1. Kreiranje projekta **{LDAP\_korisničko\_ime}\_zadaca\_4\_2** kao Java EE/EJB Module , server Glassfish, Java EE verzija: Java EE 7
2. kreirati paket **org.foi.nwtis.{LDAP\_korisnik}.ejb.sb**
3. kopirati pakete **org.foi.nwtis.{LDAP\_korisnik}.web.podaci**, **org.foi.nwtis.{LDAP\_korisnik}.rest.klijenti**iz**{LDAP\_korisnik}\_zadaca\_3\_1**
4. kreirati Stateless SessionBean **MeteoKlijentZrno**(New/Other/Enterprise JavaBeans/Session Bean/Session Type: Stateless/Create Interface (označiti ni jedan)) u **org.foi.nwtis.{LDAP\_korisnik}.ejb.sb**
5. dodati privatnu varijablu sa svojim podacima (apiKey), koja preuzima vrijednost iz datoteke postavki.
6. kreirati poslovnu operaciju void postaviKorisnickePodatke(String apiKey) (Insert Code/Add Business Method...) koja pridružuje korisničke podatke privatnim varijablama
7. preuzeti klasu MeteoPrognoza.java i prilagoditi paket
8. krerati klasu **OWMKlijentPrognoza**koja nasljeđuje **OWMKlijent** i dodati metodu public MeteoPrognoza[] getWeatherForecast(int id, String latitude, String longitude). Na temelju metode getRealTimeWeather(...) napraviti poziv REST servisa openweathermap.org za prognozu vremena za 5 dana po 3 sata (<http://openweathermap.org/forecast5>) i prilagoditi podatke za traženi tip podatka u odgovoru.
9. u **MeteoKlijentZrno**kreirati poslovnu operaciju public Lokacija dajLokaciju(String adresa) (Insert Code/Add Business Method...)
10. operacija kreira objekt klase GMKlijent, poziva metodu getGeoLocation(adresa) i vraća lokaciju
11. kreirati poslovnu operaciju public MeteoPrognoza[] dajMeteoPrognoze(int id, String adresa) (Insert Code/Add Business Method...) .
12. operacija kreira objekt klase klase OWMKlijentPrognoza, poziva metodu dajLokaciju(String adresa) i s dobivenom lokacijom poziva metodu getWeatherForecast(...)

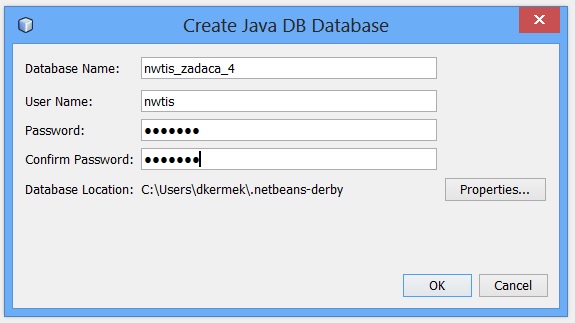
**{LDAP\_korisničko\_ime}\_zadaca\_4**: Kreiranje Enterprise aplikacije

1. Kreiranje projekta **{LDAP\_korisničko\_ime}\_zadaca\_4** kao Enterprise Application, server Glassfish, Java EE verzija: Java EE 7
2. otvoriti projekt
3. na Java EE Modules, Add Java EE Module... Označiti: {LDAP\_korisničko\_ime}\_zadaca\_4\_1, {LDAP\_korisničko\_ime}\_zadaca\_4\_2 i OK
4. kopirati datoteku**glassfish-resources.xml** iz projekta **{**LDAP\_korisničko\_ime}\_zadaca\_4\_1 (Files/{LDAP\_korisničko\_ime}\_zadaca\_4\_1/src/conf/META-INF) u projekt **{**LDAP\_korisničko\_ime}\_zadaca\_4 (Files/{LDAP\_korisničko\_ime}\_zadaca\_4/src/conf/)

**{LDAP\_korisničko\_ime}\_zadaca\_4\_3**: Kreiranje web modula za korisničko sučelje

1. Kreiranje **{LDAP\_korisničko\_ime}\_zadaca\_4\_3** kao Java Web aplikaciju, za Enterprise aplikaciju: **{LDAP\_korisničko\_ime}\_zadaca\_4**, server Glassfish, Java EE verzija: Java EE 7 (Frameworks: JSF), Libraries/Server Library: JSF 2.0, Configuration/JSF Servlet URL pattern: /faces/\* , Prefered Page Language: Facelets
2. kreirati paket **org.foi.nwtis.{LDAP\_korisnik}.web.zrna, org.foi.nwtis.{LDAP\_korisnik}.rest.klijenti**
3. kopirati paket **org.foi.nwtis.{LDAP\_korisnik}.web.podaci**iz**{LDAP\_korisnik}\_zadaca\_4\_2**
4. u paket**org.foi.nwtis.{LDAP\_korisnik}.rest.klijenti**kopirati klase**MeteoPodaci i Lokacija**iz**{LDAP\_korisnik}\_zadaca\_4\_2**
5. kreirati JSF konfiguracijsku datoteke (New/Other/JavaServer Faces/ JSF Faces Configuration)
6. kreirati predložak pod nazivom **predlozak.xhtml** (New/Others/JavaServer Faces/Facelets Template) Layout Style: CSS, izbor izgleda sa zaglavljem, podnožjem i lijevim izbornikom.  U <title>...</title> umetnuti <ui:insert name="naslov">Naslov</ui:insert> tako da se naslov može postaviti u pojedinom pogledu.  Za bottom ime prezime - NWTiS &copy; 2018. U <h:outputStylesheet name="./css..."> obrišite u name ./ tako da bude name="css/..."
7. kreirati JSF stranicu za korištenje predloška pod nazivom **pregled.xhtml** (New/Others/JavaServer Faces/Facelets Template Client), odabrati predložak predlozak.xhtml, Generated Root Tag: <html>. Odabrati samo content tako da se preostalo preuzima iz predloška. Za naslov NWTiS - Pregled parkirališta i prognoza vremena. Sadržaj za top staviti isto kao naslov, za left poveznice <h: ...> na index.xhtml, na pregled.xhtml, na pregledDnevnika.xhtml Pogled služi za pregled i pretraživanje podataka o parkiralištima i prognozama vremena
8. selektirati sadržaj **pregled.xhtml** i kopirati u **index.xhtml** . Promijeniti potrebne elemente (naslov, poveznicu,...)
9. kreirati JSF stranicu za korištenje predloška pod nazivom **pregledDnevnika.xhtml** (New/Others/JavaServer Faces/Facelets Template Client), odabrati predložak predlozak.xhtml, Generated Root Tag: <html> .... Pogled služi za pregled dnevnika.
10. dodati navigaciju u faces-config.xml
11. izgraditi i izvršiti aplikaciju **{LDAP\_korisničko\_ime}\_zadaca\_4**. Provjeriti izgled za sve poglede.
12. kopirati paket **org.foi.nwtis.{LDAP\_korisnik}.web.kontrole**iz**{LDAP\_korisnik}\_zadaca\_2**. Obrisati klasu Poruka
13. kreirati JSF ManageBean **Pregled**(Session scope) koji implementira sučelje Serializable
14. dodati varijable za novo parkiralište, popis raspoloživih parkirališta, lista raspoloživih parkirališta za odabir, popis odabranih parkirališta, lista odabranih parkirališta za vraćanje, ažurirano parkiralište, lista prognoza, prikaz ažuriranjaparkirališta, prikaz pregleda prognoza. Popisi imaju tip podatka List<Izbornik>, a liste imaju tip podatka List<String>
15. dodati settere i gettere za varijable (Može se iskoristiti dio koda iz klase MeteoPrognoza iz projekta vjezba\_12\_3).
16. u **pregled.xhtml** dodati obrazac (slika 19) s elementima za unos novog parkirališta, izbor parkirališta, pregled raspoloživih parkirališta, preged odabranih parkirališta, ažuriranje parkirališta i pregled prognoza.  (Može se iskoristiti dio koda iz pogleda unosParkiralista.xhtml iz projekta vjezba\_12\_3). Predvidjeti element za poruke o pogreškama. Elementi koji trebaju biti sakriveni koriste atribut rendered koji putem UEL ispituje stanje svoje pridružene varijable (true, false). Taj oblik ima svoju manu. Drugi oblik je da se u elementu ili njegovom nadređenom elementu (<h:panelGroup...>) koristi atribut "style" uz vrijednost "display:" nakon koje se pridružuju "none" kada se sakriva odnosno "block" kada se prikazuje. Potrebne su kontrole (gumb) za pojedine akcije. Za svaki gumb potrebno je dodati podršku za ajax (<f:ajax ..> pri čemu su važna dva atributa:
    1. execute - kojim se određuje koje elemente (na bazi id) je potrebno osvježiti, učitati prije nego što se pozove akcija
    2. render - kojim se određuju koje elemente je potrebno osvježiti nakon akcije
17. dodati projekt {LDAP\_korisnicko\_ime}\_zadaca\_4\_3 u Enterprise aplikaciju ukoliko to nije napravljeno do sad
18. u **Pregled**dodati poziv EJB ParkiralistaFacade (Insert Code .../Call Enterprise Bean i odabrati {LDAP\_korisničko\_ime}\_zadaca\_4\_1/ParkiralistaFacade)
19. dodati poziv EJB MeteoKlijentZrno (Insert Code .../Call Enterprise Bean i odabrati {LDAP\_korisničko\_ime}\_zadaca\_4\_2/MeteoKlijentZrno)
20. dodati metode koje će izvoditi akcije dodavanja novog parkirališta, ažuriranja odabranog parkirališta, dodavanje odabranog(ih) parkirališta u odabrana parkirališta, izbacivanje odabranog(ih) parkirališta iz popisa odabranihparkirališta, pregled prognoza za odabrana parkirališta itd
21. dodati kod kojim se preuzimaju podaci o parkiralištu putem ParkiralistaFacade
22. dodati kod kojim se dodaje parkiralište putem ParkiralistaFacade
23. dodati kod kojim se ažurira parkiralište putem ParkiralistaFacade.
24. dodati akcije u pregled.xhtml
25. da bi se vidjele SQL komande koje izvršava EclipseLink može se dodati u persistence.xml (<http://eclipse.org/eclipselink/> <http://wiki.eclipse.org/EclipseLink/Examples/JPA/Logging>)
26. <properties>
27. <property name="eclipselink.logging.level" value="FINE"/>
28. <property name="eclipselink.logging.level.sql" value="FINE"/>
29. <property name="eclipselink.logging.parameters" value="true"/>
30. <property name="eclipselink.logging.logger" value="DefaultLogger"/>
31. </properties>
32. u **pregledDnevnika.xhtml** (slika 20) dodati obrazac s tekstualnim elementima za filtriranje podataka: ipadresa, od, do, adresa zahtjeva, trajanje zahtjeva. Potrebna je kontrola (gumb) za pokretanje pregleda uz filtriranje. Za gumb potrebno je dodati podršku za ajax (<f:ajax ..>). Zatim dodati <h:dataTable... > za prikaz preuzetih podataka iz dnevnika.
33. kreirati JSF ManageBean **PregledDnevnika**(Session scope) i koji implementira sučelje Serializable
34. dodati potrebne varijable za filtriranje podataka i pregled podatke iz dnevnika.
35. dodati gettere i settere
36. dodati poziv EJB DnevnikFacade (Insert Code .../Call Enterprise Bean i odabrati {LDAP\_korisničko\_ime}\_zadaca\_4\_2/DnevnikFacade)
37. dodati metodu koja će izvoditi akciju preuzimanja podataka iz dnevnika na bazi podataka za filtriranje
38. dodati kod kojim se preuzimaju podaci iz dnevnika putem DnevnikFacade
39. Preporučuju se:
    1. za JSF <http://elf.foi.hr/pluginfile.php/8314/mod_resource/content/0/DZone/rc058-010d-jsf2.pdf>
    2. za Ajax <http://www.coreservlets.com/JSF-Tutorial/jsf2/#Ajax>
    3. za Criteria API <http://docs.oracle.com/cd/E19798-01/821-1841/gjitv/index.html>, <http://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Basic_JPA_Development/Querying/Criteria>

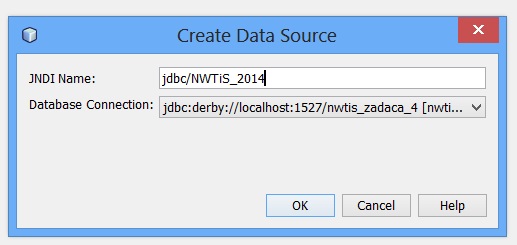
Slika 1.



Slika 2.

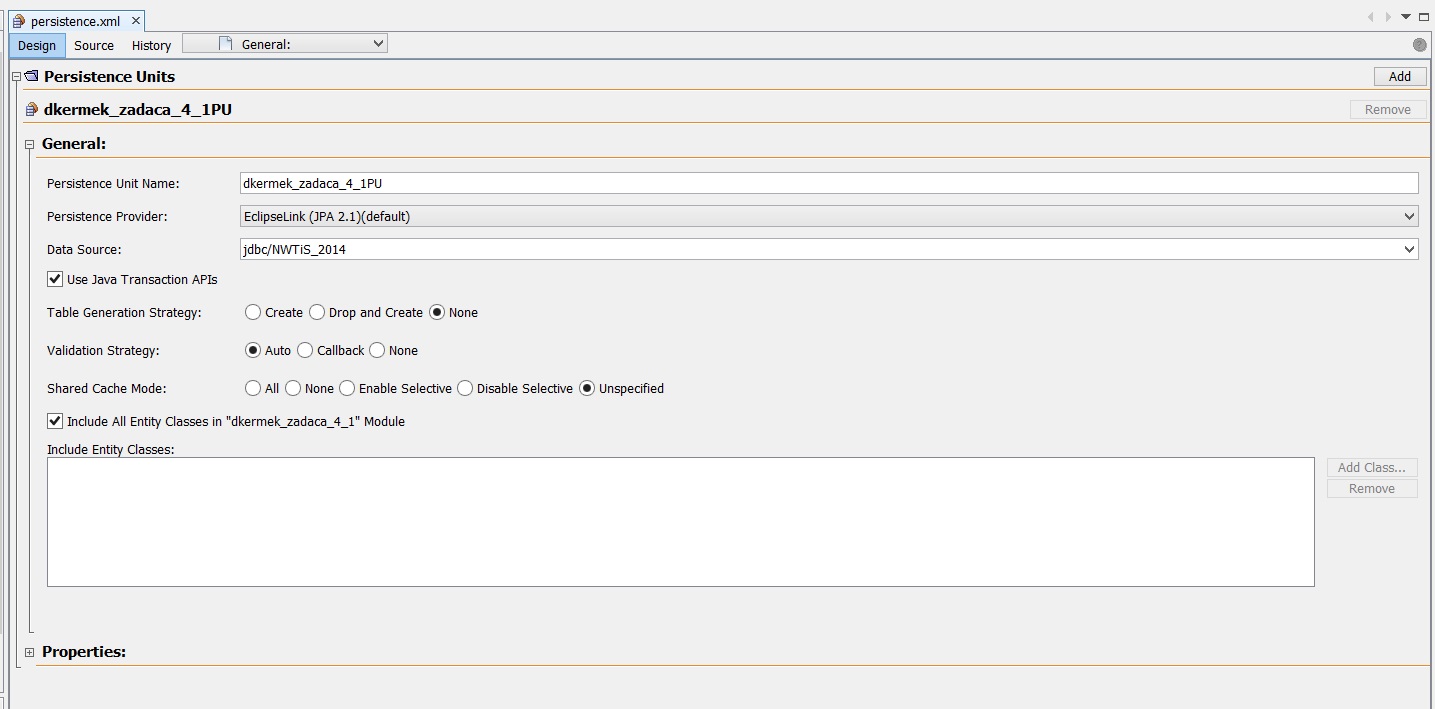
Slika 3.

Slika 4.

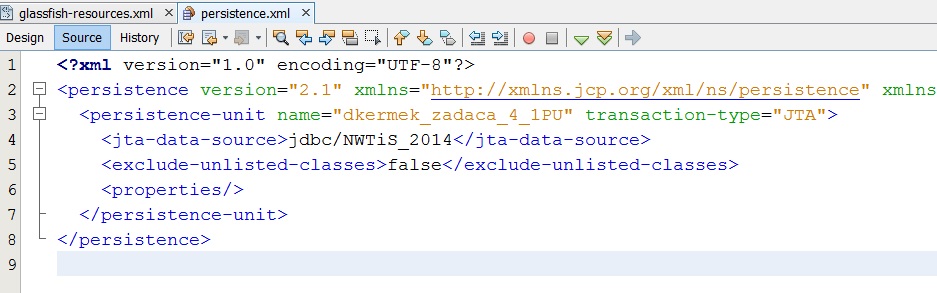


Slika 5.

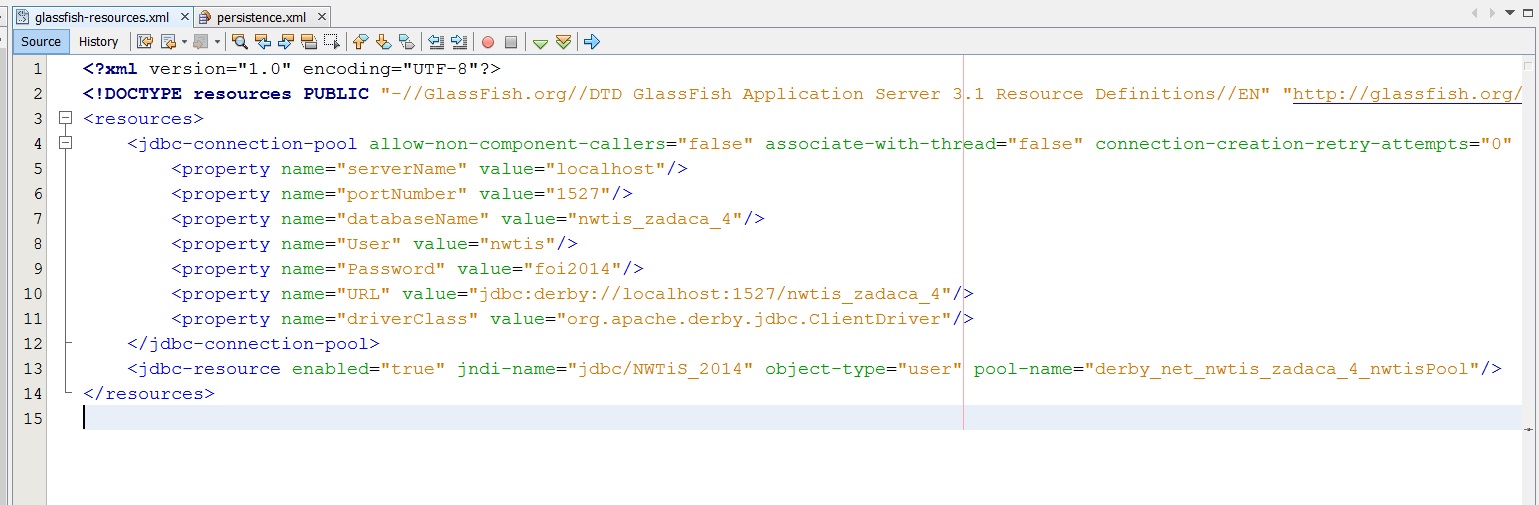
Slika 6.



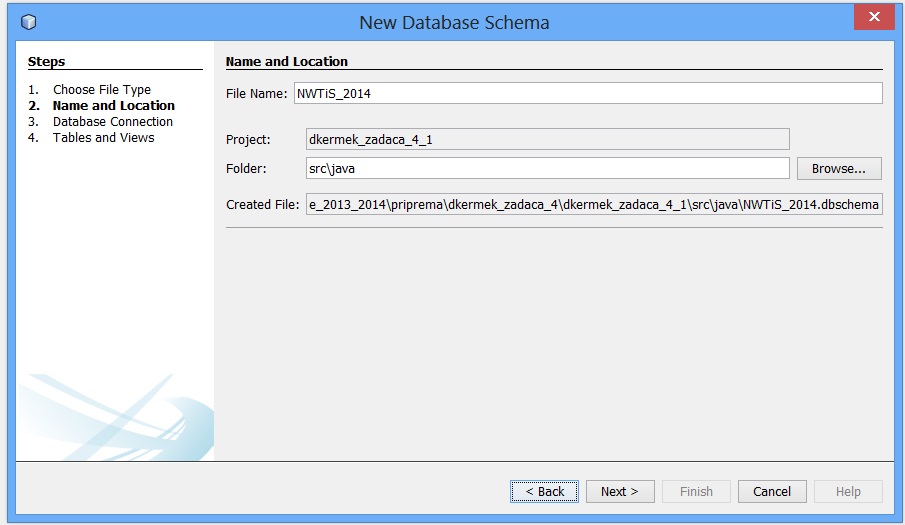
Slika 7.



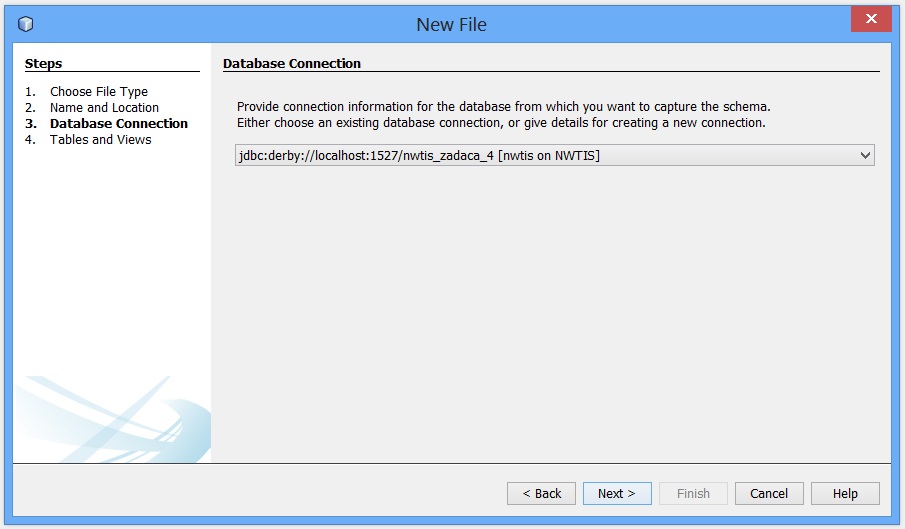
Slika 8.



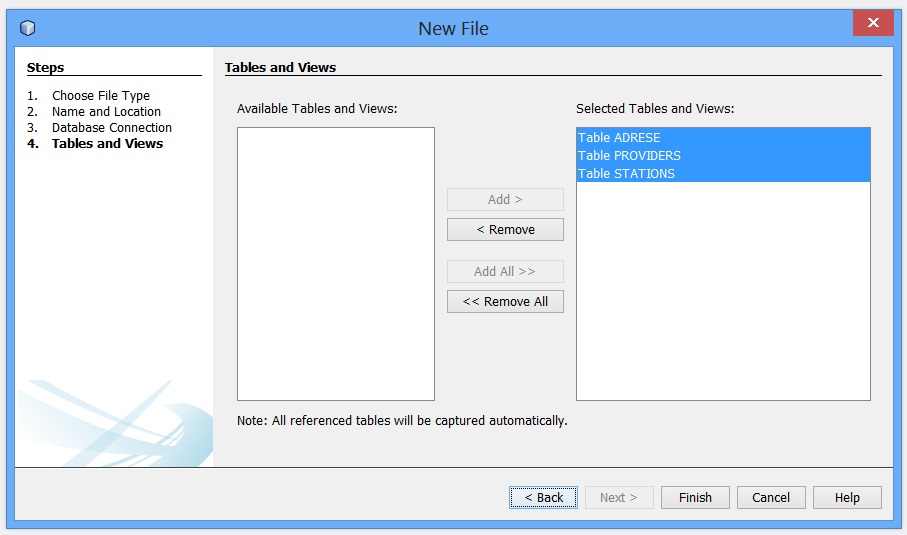
Slika 9.



Slika 10.

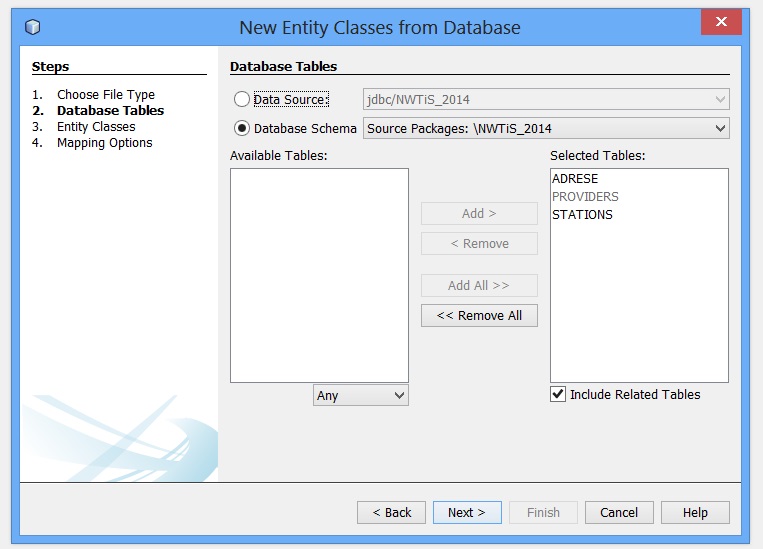


Slika 11.

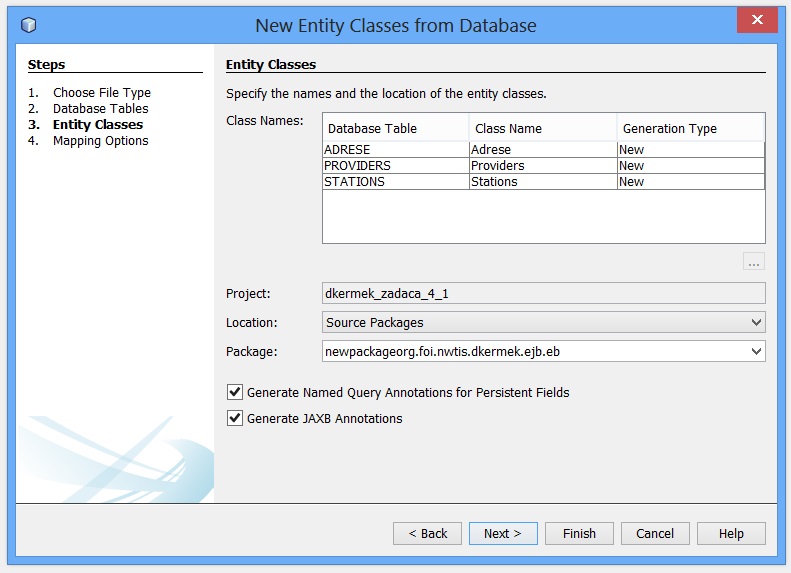


Slika 12.

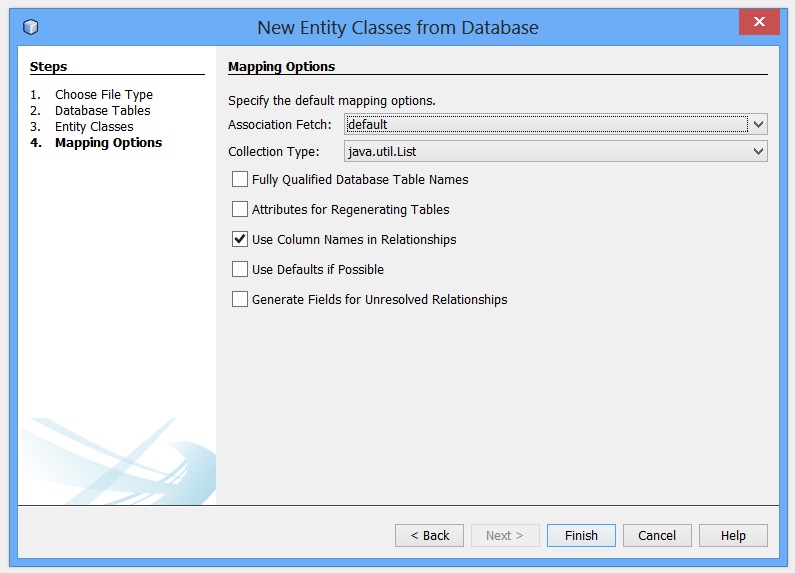
Slika 13.



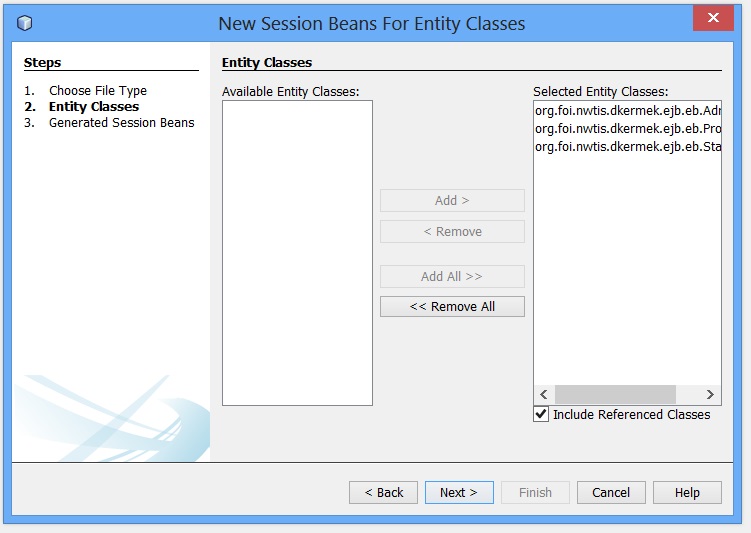
Slika 14.



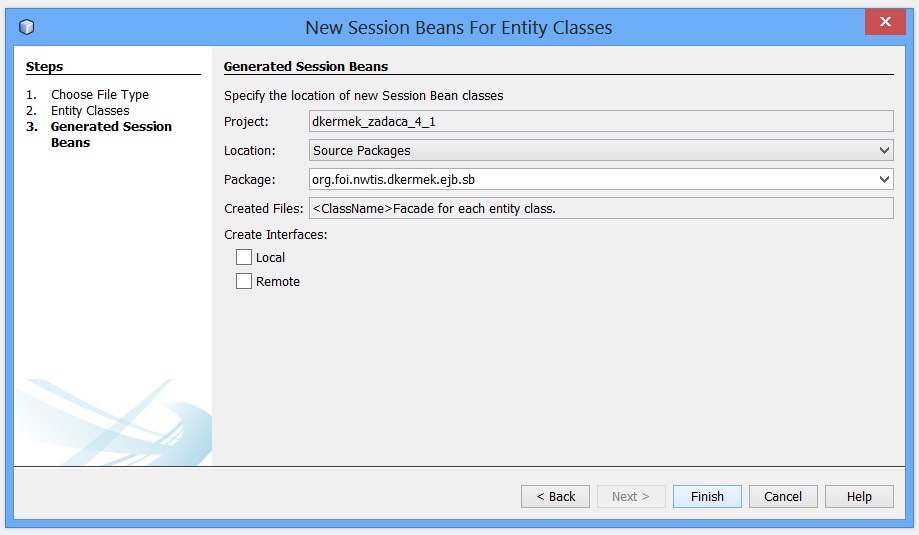
Slika 15.



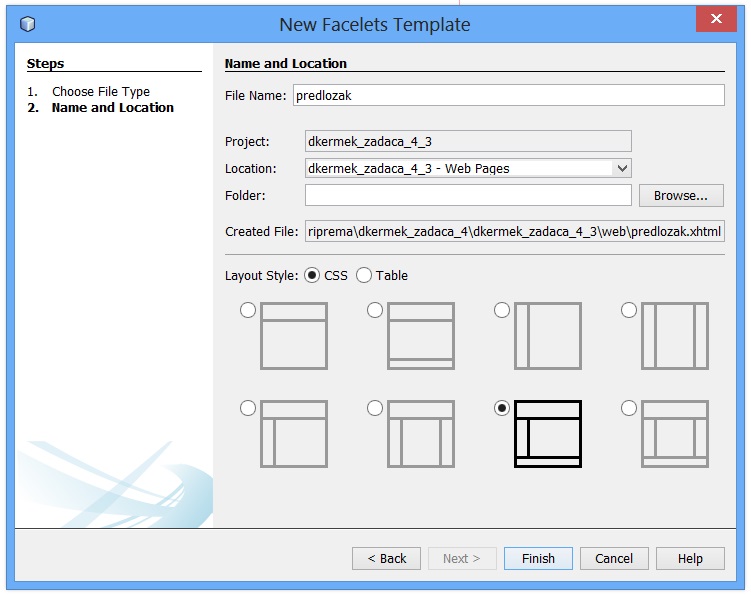
Slika 16.



Slika 17.



Slika 18.



Slika 19. (elementi u prozirnom pravokutniku se trebaju sakrivati i prikazivati na temelju akcija korisnika)

Slika 20. Pregled dnevnika