

## Projekt: NWTiS\_2012\_2013 v.29.05.2013.

Sustav se treba sastojati od sljedećih aplikacija:

1. web aplikacija ({korisnicko\_ime}\_aplikacija\_1) u pozadinskom modu (tj. servletu s automatskim startanjem ili putem slušača), pokreće dretvu (konfiguracijom se određuje pravilni vremenski interval (jedinica je sekunda) preuzimanja podataka, npr. 30 sec, 100 sec, 2 min, 10 min, 30 min, 60 min, ...) koja preuzima važeće meteorološke podatke od WeatherBug web servisa za izabrani skup gradova iz SAD (njihovi zip kodovi i tko je zatražio nalaze se u tablici {korisnicko\_ime}\_activezipcodes). Potrebno je spremati minimalno 5 meteoroloških podataka (temp, tlak, vlaga, vjetar, i sl) u bazu podataka. Budući da se za određeni zip kod može vratiti meteorološki podatak u kojem stoji neki drugi zip kod (s drugim nazivom grada i koordinatama), potrebno je upisati u tablicu oba zip koda (zahtijevani i vraćeni). Osnova je uvijek zahtijevani zip kod dok je vraćeni samo dodatna informacija koja je potrebna za neke preglede. Upravljanje pozadinskom dretvom provodi se putem primitivnog poslužitelja koji ima ulogu socket servera na određenom portu (postavkom se određuje). Kada poslužitelj primi zahtjev od klijenta zapisuje podatke u dnevnik rada u bazu podataka. Na svaki zahtjev odgovara se u obliku

OK nn [dodatni dio koji ovisi o vrsti zahtjeva]

nn je status odgovora za primljeni zahtjev. Za administratorski zahtjev prvo treba obaviti autentikaciju korisnika prema bazi podataka te se ako je u redu, šalje email poruka (adresa primatelja i predmet poruke određuju se postavkama) u MIME tipu „text/plain“ s informacijama o komandi (vrijeme izvršavanja, trajanje prethodnog stanja, broj primljenih, neispravnih i izvršenih korisničkih komandi). Provođenje spomenutih operacija ne smije utjecati na sposobnost poslužitelja da primi nove zahtjeve. Zahtjev se temelji na komandama (isključivo u jednom retku), koje mogu biti:

administratorske:

```
USER korisnik; PASSWD lozinka;
```

```
PAUSE; | START; | STOP; | ADD ZIP nnnnn;
```

korisničke:

```
USER korisnik; GET ZIP nnnnn;
```

Objašnjenje komandi:

- **USER korisnik; PASSWD lozinka;** – autentikacija korisnika. Prelazi na obradu komande ako postoji korisnik i odgovara lozinka, odnosno vraća OK 30 ako ne postoji korisnik ili ne odgovara lozinka.

- **PAUSE** – privremeno prekida preuzimanje meteoroloških podataka od sljedećeg ciklusa (i dalje može preuzimati korisničke komande). Vraća OK 10 ako nije bio u pauzi, odnosno OK 40 ako je bio u pauzi.
- **START** – nastavlja s preuzimanjem meteoroloških podataka od sljedećeg ciklusa. Vraća OK 10 ako je bio u pauzi, odnosno OK 41 ako nije bio u pauzi.
- **STOP** – potpuno prekida preuzimanje meteoroloških podataka i preuzimanje korisničkih komandi. Vraća OK 10 ako nije bio u postupku prekida, odnosno OK 42 ako je bio u postupku prekida.
- **ADD ZIP nnnnn** – dodaje zip kod za preuzimanje meteoroloških podataka od sljedećeg ciklusa. Vraća OK 10 ako ne postoji u evidenciji zip kodova za koje se preuzimaju meteorološki podaci, odnosno OK 42 ako postoji i za njega se već preuzimaju podaci.
- **TEST ZIP nnnnn** – vraća status za zip nnnnn tj. da li se za njega preuzimaju podaci. Vraća OK 10 ako postoji u evidenciji zip kodova za koje se preuzimaju meteorološki podaci, odnosno OK 44 ako ne postoji.
- **GET ZIP nnnnn** – vraća meteorološke podatke za zip nnnnn. Vraća OK 10 i

TEMP nn.nn VLAGA nn.nn TLAK nnnn.nn GEOSIR {-  
}nnn.nnnnnn GEODUZ {-}nnn.nnnnnn

ako postoji u evidenciji zip kodova za koje se preuzimaju meteorološki podaci, odnosno, OK 43 ako ne postoji.

Važeći meteorološki podaci preuzimaju se putem operacije [GetLiveWeatherByUSZipCode](http://api.wxbug.net/weatherservice.asmx) s WeatherBugWebServices web servisa <http://api.wxbug.net/weatherservice.asmx> čiji je WSDL <http://api.wxbug.net/weatherservice.asmx?WSDL>. Potrebno je obaviti registraciju da bi se mogao koristiti web servis. Adresa <http://weather.weatherbug.com/desktop-weather/api.html>. Drugi zadatak web aplikacije je pružanje web servisa za prikupljene meteorološke podatke kao što su operacije: popis svih zip kodova za koje se prikupljaju meteorološki podaci, trenutni meteorološki podaci za izabrani zip kod, rang lista (prvih n) zip kodova za koje je prikupljeno najviše meteoroloških podataka, posljednjih n meteoroloških podataka za izabrani zip code, meteorološki podaci za zip code u nekom vremenskom intervalu (od Date, do Date) i sl. Potrebno je pripremiti u NetBeans-ima za testiranje vlastitog web servisa. Treći zadatak je vidljivi dio web aplikacije odnosno korisnički dio koji treba sadržavati pregled prikupljenih meteoroloških podataka, pregled zahtjeva za server koji upravlja dretvom i pregled dnevnika korisničkih zahtjeva (kasnije opisano) putem JSP (bez skripteta) i displaytag-a uz filtriranje podataka (elementi za filtriranje: izabrani zip kod, vremenski interval (od-do, u formatu d.d.MM.yyyy hh.mm.ss) u kojem su prikupljeni podaci i sl.), status i straničenje (izbor od od 5, 10, 20, 50, 100, svi). Filtriranje treba omogućiti korištenje od ni jednog pa do svih elemenata u istom pogledu. Korisnički dio web aplikacije treba biti zaštićen putem web/servlet

kontejnera na bazi obrasca/forme za prijavljivanje uz pomoć JDBC pristupa do baze podataka te osiguranja sigurnog kanala (SSL uz vlastiti certifikat s imenom i prezimenom studenta). Četvrti zadatak je vođenje dnevnika korisničkih zahtjeva vidljivog dijela aplikacije na bazi aplikacijskog filtera uz zapis trajanje obrade zahtjeva u bazu podataka. U tablici prikupljenih meteoroloških podataka trebaju biti podaci za minimalno 20 zip kodova/gradova i za svaki od njih minimalno 100 preuzetih podataka u vremenskom intervalu većem od zadnjih 6 sati.

2. enterprise aplikacija ({korisnicko\_ime}\_aplikacija\_2) koja ima EJB i Web module. Aplikacija u pozadinskom modu (tj. servletu s automatskim startanjem ili putem slušača) pokreće dretvu (konfiguracijom se određuje pravilni vremenski interval rada (jedinica je sekunda), npr. 5 sec, 20 sec, 100 sec, ...) koja provjerava u poštanskom sandučiću (adresa poslužitelja, korisničko ime i lozinka definiraju se u konfiguracijskoj datoteci) pristiglu poštu. Od pristiglih email poruka one koje imaju predmet poruke prema postavkama i MIME tip „text/plain“ nazivamo NWTiS porukama. Obradene NWTiS poruke treba prebaciti u mapu/direktorij prema postavkama za NWTiS poruke. Ostale ne-NWTiS poruke treba prebaciti u mapu/direktorij prema postavkama za ne-NWTiS poruke. Na kraju svake iteracije obrade email poruka treba poslati JMS poruku (naziv reda čekanja 1) s podacima o vremenu početka i završetka rada, broju pročitanih poruka, broju NWTiS poruka i sl. Poruka treba biti u obliku ObjectMessage, pri čemu je naziv klase proizvoljan, a njena struktura treba sadržavati potrebne podatke koji su prethodno spomenuti.

Drugi zadatak je korisnički dio odnosno web modul, koji treba realizirati putem JSF (facelets) ili OpenFaces uz minimalno dvojezičnu varijantu (hrvatski i engleski jezik). To znači da svi statički tekstovi u pogledima trebaju biti označeni kao „labele“ i dobiti jezične prijevode. Jezik se odabire na početnoj stranici aplikacije. U javnom dijelu aplikacije može se dobiti pregled zip kodova za koje se u {korisnicko\_ime}\_aplikacija\_1 prikupljaju meteorološki podaci, zadnji meteorološki podaci za taj skup zip kodova i sl. Kod pregleda treba postojati straničenje i filtriranje zip kodova na temelju naziv grada. Pri tome se koristi Ajax-om za preuzimanje podataka. Privatni dio treba se štititi putem aplikacijskog filtra koji propušta samo prijavljenog korisnika, a ostale prebacuje na prijavljivanje. Prvi korak za novog korisnika je registracija. Korisnik nakon prijavljivanja mora kreirati svoj meteorološki portfolio ukoliko ga još nema, s time da ih može imati više. Meteorološki portfolio ima svoj identifikator (tj. naziv kao jednu riječ). Korisnik kod definiranja pojedinog portfolia određuje listu zip kodova koji ulaze u portfolio, na bazi filtriranje države, grada i zip kodova. Korisnik započinje rad izborom države (1 ili više), prikazom njenih gradova, izborom grada (1 ili više), prikazom njegovih zip kodova i izborom zip koda (1 ili više). Pri tome se koristi Ajax-om za preuzimanje podataka. Minimalni i maksimalan broj zip kodova za portfolio određuje se postavkama. Nakon što se spremi portfolio slijedi provjera statusa preuzimanja meteoroloških podataka za zip kodove iz portfolia. Provjeravaju se podaci u odnosu na definiranu operaciju web servisa iz {korisnicko\_ime}\_aplikacija\_1. Za zip kod za koji se ne

prikupljaju podaci potrebno je poslati JMS poruku (naziv reda čekanja 2) u kojoj se nalazi zip kod za koji treba početi prikupljati meteorološke podatke. Poruka treba biti u obliku `ObjectMessage`, pri čemu je naziv klase proizvoljan a njena struktura treba sadržavati potrebne podatke koji su prethodno spomenuti. Dodatni bodovi mogu se dobiti ako se za prikaz izabranih zip kodova za portfolio i njihovih podataka (npr. koordinate, trenutna temp i sl.) koristi Google Maps JavaScript API.

Prijavljeni korisnik može pregledavati trenutne, povijesne (od-do) i sl. meteorološke podatke za zip kodove izabranog portfolia, uz stranicenje, na bazi web servisa koje pruža `{korisnicko_ime}_aplikacija_1`. Kod pregleda meteoroloških podataka potrebno je prikazati i zahtijevani i vraćeni zip kod te njihovu zračnu udaljenosti. Ako se razlikuju ti zip kodovi, za zahtijevani zip kod koordinate se preuzimaju iz tablice `ZIP_CODES`.

Administrator sustava može pregledavati poruke u poštanskom sandučiću (adresa poslužitelja, korisničko ime i lozinka definiraju se u konfiguracijskoj datoteci). Pri tome može izabrati mapu/direktorij u kojem pregledava poruke. Nazivi mapa preuzimaju se od mape poštanskog sandučića korisnika email poslužitelja. Izabrane poruke može obrisati. Administrator sustava može pregledavati dnevnik rada uz filtriranje (vremenski interval od-do, **korisnik** i sl) i stranicenje.

Četvrti zadatak je pružanje popisa aktivnih (trenutno prijavljenih) Web korisnika, popisa portfolia izabranog korisnika (na bazi putanje), popis zip kodova iz izabranog portfolia izabranog korisnika (na bazi putanje) putem RESTful web servisa.

Pristup do podataka u bazi podataka treba biti realiziran putem ORM-a tj. putem session, entity bean-ova i criteria API.

3. enterprise aplikacija (`{korisnicko_ime}_aplikacija_3`) koja ima EJB i Web module. Aplikacija preuzima dvije vrste JMS poruka (kao `MessageDriven Bean`) za stanje obrade email poruka i zip kod za koji treba početi preuzimati meteorološke podatke o pregled podataka meteorološke prognoze. Kod primitka JMS poruke za zip kod potrebno je provjeriti status prikupljanja meteoroloških podataka za traženi zip kod putem slanja zahtjeva socket serveru iz `{korisnicko_ime}_aplikacija_1` (postavkom se određuje adresa i port). Ukoliko se još ne prikupljaju podaci, šalje se zahtjev da se počne s preuzimanje meteoroloških podataka za traženi zip kod. Primitljene poruke se spremaju u memoriji aplikacije. Ako aplikacija prestaje s radom, potrebno je poruke serializirati na vanjski spremnik (naziv datoteke u postavkama, smještena u `WEB-INF` direktoriju). Kada se aplikacija podiže potrebno je učitati serializirane poruke (ako postoji datoteka) u memoriju aplikacije. Web modul aplikacije koristi JSF (facelets) ili OpenFaces za pregled spremljenih JMS poruka. Korisnik može brisati izabrane poruke ili sve poruke. Korisnik može pozivom RESTful web servisa dohvatiti i ispisati podatke za aktivne korisnike `{korisnicko_ime}_aplikacija_2`, za neko korisničko ime ispisati nazive njegovih portfolia ili ispisati zip kodove za izabrani portfolio. Za jedan od prethodno

ispisanih zip kodova može tražiti trenutne meteorološke podatke putem slanja zahtjeva socket serveru iz {korisnicko\_ime}\_aplikacija\_1 te ih ispisati.

Komunikacija između aplikacija:

- putem web servisa, email poruka, JMS poruka i vlastitog jezika na bazi socket servera (NE putem baza podataka).

Sustav ima sljedeću instalacijsku i programsku arhitekturu:

#### **{korisnicko\_ime}\_aplikacija\_1:**

- Web poslužitelj: Tomcat
- **EE osobine: EE6 Web, uz obavezno korištenje web.xml i bez korištena EE6 osobina za filtere, slušače, servlete i sl.)**
- korisničko sučelje: JSP
- baza podataka: MySQL - naziv nwtis\_{korisnickoime}\_bp\_1, treba sadržavati tablice: {korisnicko\_ime}\_activezipcodes i ostale koje su potrebne za rad
- rad s bazom podataka: JDBC, SQL
- **JMS red poruka: NWTiS\_{korisnicko\_ime}\_1**
- vlastiti socker server

#### **{korisnicko\_ime}\_aplikacija\_2:**

- Web poslužitelj: Glassfish
- EE osobine: EE6 (anotacije za filtere, slušače, servlete i sl.)
- korisničko sučelje: JSF (facelets) ili OpenFaces
- baza podataka: JavaDB – naziv nwtis\_{korisnickoime}\_bp\_2, treba sadržavati podatke i tablice (STATES, CITIES, COUNTIES, ZIP\_CODES) iz baze podataka NWTiS\_2012 (iz vježbe 13/zadaća 4) i ostale koje su potrebne za rad.
- rad s bazom podataka: ORM (EclipseLink), Criteria API
- **JMS red poruka: NWTiS\_{korisnicko\_ime}\_2**

#### **{korisnicko\_ime}\_aplikacija\_3:**

- Web poslužitelj: Glassfish
- EE osobine: EE6 (anotacije za filtere, slušače, servlete i sl.)
- korisničko sučelje: JSF (facelets) ili OpenFaces
- baza podataka: ne koristi bazu podataka.

Projektna dokumentacija:

dokumentacija.html koja je dostupna iz svih web aplikacija, sadrži opis projektnog zadatka i glavne odrednice projektnog rješenja. Potrebna je Java doc dokumentacija. Nije potrebna korisnička dokumentacija