## Vježba 10. / Zadaća 2.

Naziv: web aplikacija za rad s email porukama s lokaliziranim korisničkim sučeljem i obradom email poruka u pozadinskoj dretvi

Naziv projekta: **{LDAP\_korisničko\_ime}\_zadaca\_2**

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 **{LDAP\_korisničko\_ime}\_zadaca\_2.zip** i predati u Moodle. Uključiti izvorni kod, primjere datoteka konfiguracijskih podataka i popunjeni obrazac za zadaću pod nazivom **{LDAP\_korisničko\_ime}\_zadaca\_2.doc** (u korijenskom direktoriju projekta).

Slušač aplikacije starta pozadinsku dretvu koja provjerava na poslužitelju (adresa i IMAP port određena konfiguracijom, mail.server i mail.imap.port) u pravilnom intervalu (određen konfiguracijom u sek, mail.timeSecThreadCycle) ima li poruka u poštanskom sandučiću korisnika (adresa i lozinka određeni konfiguracijom, mail.usernameThread i mail.passwordThread). Koristi se IMAP protokol. Poruke se učitavaju po grupama od n poruka (određeno konfiguracijom, mail.numMessagesToRead) dok se ne pročitaju sve poruke. Poruke koje sadrže samo jedan privitak (attachment) s točno traženim nazivom (određen konfiguracijom, mail.attachmentFilename) nazivamo NWTiS porukama, obrađuju se tako da se ispituje sadržaj datoteke privitka. Sadržaj datoteke treba biti u "text/json" ili "application/json" formatu i može biti sa sljedećem sintaksom:

* {"id": d{1-4}, "komanda":  "dodaj|azuziraj", ("{atribut {1-30}": "vrijednost d{1-3}|d{1-3}.d{1-2}|{1-30}"){1-5}, "vrijeme": "yyyy.MM.dd hh:mm:ss")}
* npr:
  + {"id": 1, "komanda": "dodaj", "naziv": "Senzor temperature", "vrijeme": "2018.04.08 11:20:45"}
  + {"id": 1, "komanda": "azuriraj", "temp": 20.4, "vrijeme": "2018.04.08 11:26:10"}
  + {"id": 2, "komanda": "dodaj", "naziv": "Senzor RFID", "korisnik": "pero", "vrijeme": "2018.04.09 17:21:56"}
  + {"id": 2, "komanda": "azuriraj", "korisnik": "mato", "vrijeme": "2018.04.09 21:09:01"}
  + {"id": 3, "komanda": "dodaj", "naziv": "Meteo stanica", "temp": 22.0, "vlaga": 77, "vjetar": "NW"  "vrijeme": "2018.04.09 18:19:41"}

Za poruku koja ima privitak ispravne sintakse slijedi provođenja postupka: ako se radi o komandi "dodaj" tada treba dodati zapis u tablicu UREDAJI u bazi podataka, s time da ako već postoji takav id, onda je pogreška. Ako se radi o komandi "azuriraj" tada se provjerava zapis u tablici UREDAJI za zadani id IoT i ako postoji, ažurira se u tablici. Ako ne postoji, onda je pogreška. Svaka primljena poruka upisuje se u tablicu DNEVNIK bez obzira na status prethodne akcije.

Obrađene NWTiS poruke prebacuju se u posebnu mapu (određena konfiguracijom, mail.folderNWTiS). Ako mapa ne postoji, dretva ju treba sama kreirati. Ostale poruke (koje nisu NWTiS poruke) treba ostaviti da imaju isti status (nepročitane ili pročitane) koje su imale prije nego što ih je dretva provjeravala.

Dretva na kraju svakog ciklusa dodaje podatke o radu u datoteku (naziv određen konfiguracijom, mail.threadCycleLogFilename, i sadržaj u UTF-8) u sljedećem obliku:

Obrada poruka broj:  
Obrada započela u: vrijeme\_1 (dd.MM.yyyy hh.mm.ss.zzz)  
Obrada završila u: vrijeme\_2 (dd.MM.yyyy hh.mm.ss.zzz)   
Trajanje obrade u ms: n  
Broj poruka: n - odnosi se na jedan ciklus  
Broj dodanih IOT: n   
Broj ažuriranih IOT: n   
Broj neispravnih poruka: n  
<prazan redak>

Potrebno je napraviti korisnički dio web aplikacije za izbor jezika za lokalizaciju, slanje email poruka, pregled primljenih email poruka i pregled dnevnika.

Izbor jezika za lokalizaciju (slika 1.) provodi se između raspoloživih jezika u koje nije uključen trenutno važeći jezik za lokalizaciju (za to se koristi atribut rendered). Koriste se gumbi koji na svojim akcijama pozivaju funkciju za postavljanje jezika (lokalizacije) uz argument koji odgovara odabranom jeziku. Svi statički elementi u pogledima trebaju biti pripremljeni da se prikažu u skladu s izabranih jezikom. Sve poruke (validacija i sl) u stalim pogledima trebaju biti prikazane u skladu s izabranih jezikom. U zaglavlju svih pogleda treba biti poveznica na izbor jezika, poveznica na slanje email poruke, poveznica za pregled primljenih email poruka i poveznica za pregled dnevnika. Ne prikazuje se ona poveznica koja se odnosi na važeći pogled. 

Kod odabira slanja email poruka (slika 2.) slijedi prikaz u obliku tablice (ili rešetke-grid) unutar koje je obrazac za unos podataka poruke (primatelj, pošiljatelj, predmet, sadržaj za privitak, brisanje sadržaja za privitak, preuzimanja sadržaja jedne od pripremljenih datoteka u element (kontrolu) sadržaja za privitak. Inicijalno adresa primatelja je iz postavke konfguracije mail.usernameThread. Inicijalno adresa pošiljatelja je iz postavke konfguracije mail.usernameEmailAddress. Inicijalno predmet je iz postavke konfguracije mail.subjectEmail. Potrebno je provjeriti ispravnost (validirati) upisanih podataka za primatelja, pošiljatelja, predmet poruke i sadržaj za privitak. Ispravnost primatelja i pošiljatelja temelji se na korektnost strukture email adrese. Predmet poruke mora sadržavati min 10 znakova. Sadržaj za privitak treba biti ispravan json tekst. U podnožju tablice nalazi se gumb za slanje email poruke. U zaglavlju pogleda treba biti poveznica na izbor jezika, poveznica na pregled primljenih email poruka i poveznica na pregled dnevnika

Kod odabira pregleda primljenih email poruka (slika 3.) prikazuje se padajući izbornik u kojem su elementi mape (direktoriji) iz email korisničkog računa (radi se o objektima klasa javax.mail.Folder i javax.mail.Store) i gump za promjenu mape. Na Od svih mapa samo su od interesa i jedino se prikazuju mapa INBOX i mapa koja je zadana konfiguracijom za NWTiS poruke (određena konfiguracijom, mail.folderNWTiS, a sadrži naziv u obliku NWTiS {LDAP\_korisnik} poruke) ako trenutno postoji. početku je odabrana mapa INBOX. Slijedi tablica s prikazom informacija o n (određen konfiguracijom, mail.numMessagesToShow) najsvježijih poruka (objekti klase javax.mail.Message) iz odabrane mape. Za  svaku email poruku potrebno je prikazati tko je poslao, kada je poslao (hr formatirano dd.mm.gggg hh:mm:ss), predmet poruke, vrsta poruke (NWTiS ili ne NWTiS), sadržaj privitka ako je NWTiS poruka, ako nije NWTiS poruka onda ništa. **NE SMIJU** se čitati sve poruke iz mape nego samo onoliko koliko je potrebno za prikaz prema postavki. Ispod tablice prikazuje se ukupan broj email poruka u izabranoj mapi, gumb za prethodne i sljedeće poruke. Pojedini gumb treba sakriti (koristi se atribut rendered) ako nema sadržaja koji o njemu ovisi. Npr. na početku nema prethodnih poruka. Aktiviranjem pojedinog gumba treba prikazati izabrani skup email poruka. U zaglavlju treba biti poveznica na izbor jezika, poveznica na slanje email poruke i poveznica na pregled dnevnika.

Kod odabira pregleda dnevnika (slika 4.) prikazuje se odabir intervala Od - Do i gump za pregled. Upisanim elementima Od i Do treba provjeriti ispravnost (validirati) u formatu datumskog tipa podatka (dd.mm.gggg hh:mm:ss). Slijedi tablica s prikazom informacija o n (određen konfiguracijom, mail.numLogItemsToShow) najsvježijih zaspisa u dnevniku za traženi interval. Za svaki zapis iz dnevnika potrebno je prikazati kada je spremljen i njegov sadržaj. **NE SMIJU** se čitati u memoriju svi zapisi iz dnevnika nego samo onoliko koliko je potrebno prema postavki. Ispod tablice prikazuje se ukupan broj zapisa u dnevniku koji zadovoljavaju interval Od Do, gumb za prethodne i sljedeće zapise dnevnika. Pojedini gumb treba sakriti (koristi se atribut rendered) ako nema sadržaja koji o njemu ovisi. Npr. na početku nema prethodnih zapisa iz dnevnika. Aktiviranjem pojedinog gumba treba prikazati izabrani skup zapisa iz dnevnika. U zaglavlju treba biti poveznica na izbor jezika, poveznica na slanje email poruke i poveznica na pregled poruka.

Pregledi trebaju biti vizualno sređeni pomoću css-a.

Metode u klasama NE smiju imati više od 25 linija programskog koda, u što se ne broji definiranje metode, njenih argumenata i lokalnih varijabli. U jednoj liniji može biti jedna instrukcija. Linija ne može ima više od 132 znaka.

JavaMail API nalazi se na adresi: <https://javamail.java.net/nonav/docs/api/>

JavaMail FAQ nalaze se na adresi <http://www.oracle.com/technetwork/java/faq-135477.html>

Preporučeni koraci:

1. Kreiranje projekta **{LDAP\_korisnicko\_ime}\_zadaca\_2** kao Java Web aplikaciju, server **Glassfish**, Java EE verzija: **Java EE 7**, kontekst **{LDAP\_korisnicko\_ime}\_zadaca\_2** , (Frameworks: JSF), Libraries/Server Library: JSF 2.\*, Configuration/JSF Servlet URL pattern: /faces/\* , Prefered Page Language: Facelets
2. kreirati pakete **org.foi.nwtis.{LDAP\_korisnik}.web.slusaci**,**org.foi.nwtis.{LDAP\_korisnik}.web.filteri**, **org.foi.nwtis.{LDAP\_korisnik}.web.kontrole**, **org.foi.nwtis.{LDAP\_korisnik}.web.zrna, , org.foi.nwtis.{LDAP\_korisnik}.web.dretve**
3. Kreirati direktorij lib u projektu
4. kreirati direktorij **resources** u Web direktorij, unutar njega kreirati direktorij **css**
5. kreirati datoteku **osnovna.css** (New/Other/Web/Cascading Style Sheet). Pripremiti selektore klase (eksplicitni css) za stilsko uređenje pogleda.
6. Kopiranje biblioteke dist\vjezba\_03\_2.jar iz projekta **vjezba\_03\_2** u direktorij lib
7. Kopiranje biblioteke dist\vjezba\_06\_1.jar iz projekta **vjezba\_06\_1** u direktorij lib
8. Kopiranje biblioteke gson.jar iz projekta **vjezba\_03\_2** u direktorij lib
9. Dodavanje biblioteke (jar/folder) lib\vjezba\_03\_2.jar u projekt
10. Dodavanje biblioteke (jar/folder) lib\vjezba\_06\_1.jar u projek
11. Dodavanje biblioteke (jar/folder) lib\gson.jar u projekt
12. Napomena: biblioteka za JavaMail sastavni je dio Glassfish biblioteke te ju nije potrebno posebno dodavati
13. Preuzeti primjere datoteke konfiguracija te kopirati u direktorij WEB-INF
14. upisati u **web.xml**inicijalni parametar konteksta **konfiguracija** i dodati naziv datoteke konfiguracije
15. kreirati slušaca konteksta **SlusacAplikacije**(anotirani bez uključivanja u web.xml) i kopirati kod funkcije contextInitialized(...) iz vjezba\_08\_1
16. preuzeti priloženu datoteku u poveznici "SQL za kreiranje tablica" i izvršiti u MySQL i JavaDB
17. preuzeti priloženu datoteku u poveznici "SQL za punjenje tablica" i izvršiti u MySQL i JavaDB
18. kreirati klasu **ObradaPoruka** koja je dretva i starta se prilikom inicijaliziranja konteksta. Trajanje ciklusa rada određeno je konfiguracijom.
19. za čitanje email poruka može se preuzeti programski kod iz primjera s predavanja koji je priložen pod nazivom WebMail.java
20. u metodi contextDestroyed(...) klase **SlusacAplikacije**dodati da se dretva zaustavlja.
21. izgraditi, isporučiti, izvršiti i testirati aplikaciju
22. kreirati properties datoteku **prijevod** (New/Other/Properties) u **org.foi.nwtis.{LDAP\_korisnik}**. Za elemente koji se dodaju vrijedi pravilo da prvo dolazi naziv pogleda npr. index, zatim . pa naziv elementa. Npr. index.naslov=Početna stranica, index.izborJezika=Izaberi jezik
23. dodati lokalizacije za hr, en i de.
24. dodati faces-config (New/JSF(JSF Faces Configuration))
25. dodati u JSF konfiguraciju za prijevod (bundle, varijabla **p**). Primjer na predavanjima prikazuje potrebe elemente ili pogledati <http://jdevelopment.nl/internationalization-jsf-utf8-encoded-properties-files/>, <http://www.tutorialspoint.com/jsf/jsf_internationalization.htm>
26. kreirati JSF ManageBean **Lokalizacija**(New/JSF(JSF Manage Bean) (bez uključivanja u web.xml), scope: session
27. kreirati varijablu za odabrani jezik (String)
28. kreirati setter i getter za odabrani jezik. Važeća lokalizacija dobije se iz FacesContext.getCurrentInstance().getViewRoot().getLocale(); Nova lokalizacija postavlja se putem FacesContext.getCurrentInstance().getViewRoot().setLocale(Locale);
29. kreirati funkciju koja će biti akcija za preuzimanje odabranog jezika public Object odaberiJezik(String jezik). Može se koristiti ideja iz <http://jdevelopment.nl/internationalization-jsf-utf8-encoded-properties-files/> ili  iz  <http://courses.coreservlets.com/Course-Materials/pdf/jsf/jsf2/JSF2-Event-Handling.pdf> .
30. dodati funkciju public String saljiPoruku() koja vraća "saljiPoruku", funkciju public String pregledPoruka() koja vraća "pregledPoruka" i  funkciju public String pregledDnevnika() koja vraća "pregledDnevnika"
31. u **index.xhtml** (slika 1.) dododati osnovna.css putem <h:outputStylesheet library="css" name="osnovna.css"/>
32. dodati u zaglavlju poveznice <h:commandLink ...> za slanje poruke, za pregled poruka i za pregled dnevnika
33. dodati obrazac <h:form ...> u kojem će biti gumbi <h:commandButton ...> za svaki od ponuđenih jezika. Ako se radi o gumbu za jezik npr. hr tada se njegov prikaz ne ostvaruje pomoću atributa
    * rendered="#{lokalizator.odabraniJezik ne 'hr'}"
34. izgraditi, isporučiti, izvršiti i testirati aplikaciju
35. dodati navigacijska pravila za prethodni pogled tako da svaki izlaz bude usmjeren na traženi pogled
36. izgraditi, isporučiti, izvršiti i testirati aplikaciju. Ako se tijekom izvršavanja javi pogreška zbog Session (No active contexts for scope type javax.enterprise.context.SessionScoped), preporučuje se obrisati označenu postavku za GlassFish (slika 3). Drugo rješenje je odati obradu iznimke koja prebacuje na početni pogled index.xhtml
37. kreirati JSF MenagesBean **SlanjePoruka**, scope: request
38. kreirati String varijablu za adresu email poslužitelja u koju se upisuju podaci iz konfiguracije
39. kreirati String varijable za primatelja, pošiljatelja, predmet u koje se inicijalno upisuju podaci iz konfiguracije
40. kreirati String varijablu za sadržaj privitka poruke, String varijablu za odabranu datoteku, List<String> za popis datoteka u koji treba preuzeti nazive datoteka iz direktorija WEB-INF s ekstenzijom .json
41. dodati gettere i settere za varijable
42. kreirati funkciju public String preuzmiSadrzaj() za preuzimanje sadržaja odabrane datoteke. Funkcija čita sadržaj odabrane datoteke i postavlja kao sadržaj za privitak
43. kreirati funkciju public String saljiPoruku() za slanje upisane poruke. Može se iskoristiti primjer s predavanja koji je priložen pod nazivom SendMail.java. Dodati varijablu za poruku. Ako je u redu, ispisuje se poruka da je poruka uspješno poslana, inače tekst progreške.
44. dodati funkciju public String promjenaJezika() koja vraća "promjenaJezika", funkciju public String pregledPoruka() koja vraća "pregledPoruka" i funkciju public String pregledDnevnika() koja vraća "pregledDnevnika"
45. kreirati JSF **slanjePoruka.xhtml** (slika 2.)
46. kreirati obrazac za unos potrebnih podataka za slanje email poruke (prima, šalje, predmet) <h:inputText ...> , sadržaj privitka <h:inputTextArea ..>, padajući izbornik s nazivima datoteka <h:selectOneListBox...> , gumb za brisanje sadržaja <h:commandButton ...>, gumb za preuzimanje sadržaja datoteke <h:commandButton ...> i gumb za slanje poruke <h:commandButton ...>. Labele se označavaju kao <h:commandLabel ...>
47. padajući izbornik ima 5 prikazanih elemenata za nazive datoteka a elementi izbornika
    * <f:selectItems value="#{zrno.nizDatoteka}" var="c" itemLabel="#{c.labela}" itemValue="#{c.vrijednost}" />
48. dodati u zaglavlju poveznice <h:commandLink ...> za promjenu jezika, za pregled poruka i za pregled dnevnika
49. dodati navigacijska pravila za prethodni pogledi
50. izgraditi, isporučiti, izvršiti i testirati aplikaciju
51. preuzeti klasu **Izbornik**u čije objekte će se spremati podaci o elementima izbornika
52. preuzeti klasu **Poruka** u čije objekte će se spremati podaci o email porukama
53. kreirati JSF ManegedBean **PregledPoruka**, scope: request
54. kreirati varijable za adresu poslužitelja, korisničko ime i lozinku (određeni konfiguracijom)
55. kreirati varijablu za preuzete mape tipa List<Izbornik> i varijablu za izabranu mapu tipa String
56. kreirati varijablu za preuzete poruke tipa List<Poruka**>**
57. kreirati varijable za ukupan broj poruka u izabranoj mapi, broj poruka koje se mogu prikazati, pozicije od i do prikazanih poruka, sve tipa int
58. kreirati funkciju void preuzmiMape() koja preuzima mape iz email korisničkog računa. Za vježbe se kreiraju 2 mapa pomoću linije
    * new Izbornik("INBOX","INBOX");
59. kasnije treba promijeniti kod koji će preuzeti stvarne mape iz korisničkog računa.
60. kreirati funkciju void preuzmiPoruke() koje preuzima dozvoljeni broj email poruka iz izabrane mape na bazi trenutne pozicije u mapi. Za vježbe se kreira 10 poruka pomoću linije
    * new Poruka(Integer.toString(i++), new Date(), new Date(), "pero@localhost", "P " + Integer.toString(i), "Poruka " + Integer.toString(i), NWTiS\_Poruka)
61. kasnije treba promijeniti kod koji će preuzeti stvarne poruke sa poslužitelja
62. kreirati funkciju public String promjenaMape() koja će biti akcija za promjenu mape u padajućem izborniku. Vraća "PromjenaMape".
63. kreirati funkciju public String prethodnePoruke() koja će biti akcija za prethodne (starije) email poruke u mapi. Vraća "PrethodnePoruke". Gumb se ne prikazuje ako nema prethodnih poruka za što se koristi atribut rendered.
64. kreirati funkciju public String sljedecePoruke() koja će biti akcija za sljedeće (novije) email poruke u mapi. Vraća "SljedecePoruke". Gumb se ne prikazuje ako nema sljedećih poruka za što se koristi atribut rendered.
65. dodati funkciju public String promjenaJezika() koja vraća "promjenaJezika", funkciju public String saljiPoruku() koja vraća "saljiPoruku" i funkciju public String pregledDnevnika() koja vraća "pregledDnevnika"
66. kreirati JSF **pregledPoruka.xhtml**(izgled na slici 3)
67. obrazac u kojem se nalazi padajući izbornik za mape <h:selectOneMenu ...>  a elementi izbornika
    * <f:selectItems value="#{zrno.nizMapa}" var="c" itemLabel="#{c.labela}" itemValue="#{c.vrijednost}" />
68. dodati gumb za slanje <h:commandButton ...> uz akciju na ciljanu funkciju.
69. dodati <h:dataTable ...> za prikaz podataka o email porukama. Za pojedinu poruku koristi se objekt klase **Poruka**.  Za stupce se koristi <h:column...> za naslov stupca se koristi <f:facet name="header"...> a za vrijednost <h:outputText ...>
70. dodati converter <f:convertDateTime pattern="..."/> za vrijeme email poruke
71. dodati za ukupno poruka <h:inputText ...>
72. dodati gumbe za slanje <h:commandButton ...> uz akciju na ciljane funkcije.
73. dodati u zaglavlju poveznice <h:commandLink ...> za promjenu jezika, za slanje poruke i za pregled dnevnika
74. dodati navigacijska pravila za prethodni pogled
75. izgraditi, isporučiti, izvršiti i testirati aplikaciju
76. kreirati JSF ManegedBean **PregledDnevnika**, scope: request
77. kreirati varijablu za preuzete zapise dnevnika tipa List<Dnevnik**>**
78. kreirati varijable za ukupan broj zapisa u dnevniku u izabranom intervalu,  broj zapisa koji se mogu prikazati, pozicije od i do prikazanih zapisa, sve tipa int
79. kreirati funkciju void preuzmiZapise() koja preuzima zapise iz tablice DNEVNIK u bazi podataka na temelju intervala Od Do. Za vježbe se kreira 5 zapisa pomoću linije koju kasnije treba zamijeniti kodom koji će preuzeti stvarne zapise
    * new Dnevnik(Integer.toString(i++), "{'id': 1, 'komanda': 'dodaj', 'naziv': 'Senzor temperature', 'vrijeme': '2018.04.08 11:20:45'}");
80. kreirati funkciju public String promjenaIntervala() koja će biti akcija za promjenu intervala Od Do. Vraća "PromjenaIntervala".
81. kreirati funkciju public String prethodniZapisi() koja će biti akcija za prethodne (starije) zapise u dnevniku. Vraća "PrethodniZapisi". Gumb se ne prikazuje ako nema prethodnih zapisa.
82. kreirati funkciju public String sljedeciZapisi() koja će biti akcija za sljedeće (novije) zapise u dnevniku. Vraća "SljedeciZapisi".  Gumb se ne prikazuje ako nema sljedećih zapisa.
83. dodati funkciju public String promjenaJezika() koja vraća "promjenaJezika", funkciju public String saljiPoruku() koja vraća "saljiPoruku" i funkciju public String pregledPoruka() koja vraća "pregledPoruka"
84. kreirati JSF **pregledDnenvika.xhtml**(izgled na slici 3)
85. obrazac u kojem se nalazi unos početka intervala <h:inputText ...> i unos kraja intervala <h:inputText ...>
86. dodati gumb za slanje <h:commandButton ...> uz akciju na ciljanu funkciju.
87. dodati <h:dataTable ...> za prikaz podataka o zapisima dnevnika. Za pojedini zapis koristi se objekt klase **Dnevnik**.  Za stupce se koristi <h:column...> za naslov stupca se koristi <f:facet name="header"...> a za vrijednost <h:outputText ...>. Vrijeme treba
88. dodati converter <f:convertDateTime pattern="..."/> za vrijeme zapisa u dnevnik
89. dodati za ukupno zapisa u dnevniku <h:inputText ...>
90. dodati u zaglavlju poveznice <h:commandLink ...> za promjenu jezika, za slanje poruke i za pregled poruka
91. dodati navigacijska pravila za prethodni pogled
92. izgraditi, isporučiti, izvršiti i testirati aplikaciju

## Vježba 10. / Zadaća 2.

Naziv: web aplikacija za rad s email porukama s lokaliziranim korisničkim sučeljem i obradom email poruka u pozadinskoj dretvi

Naziv projekta: **{LDAP\_korisničko\_ime}\_zadaca\_2**

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 **{LDAP\_korisničko\_ime}\_zadaca\_2.zip** i predati u Moodle. Uključiti izvorni kod, primjere datoteka konfiguracijskih podataka i popunjeni obrazac za zadaću pod nazivom **{LDAP\_korisničko\_ime}\_zadaca\_2.doc** (u korijenskom direktoriju projekta).

Slušač aplikacije starta pozadinsku dretvu koja provjerava na poslužitelju (adresa i IMAP port određena konfiguracijom, mail.server i mail.imap.port) u pravilnom intervalu (određen konfiguracijom u sek, mail.timeSecThreadCycle) ima li poruka u poštanskom sandučiću korisnika (adresa i lozinka određeni konfiguracijom, mail.usernameThread i mail.passwordThread). Koristi se IMAP protokol. Poruke se učitavaju po grupama od n poruka (određeno konfiguracijom, mail.numMessagesToRead) dok se ne pročitaju sve poruke. Poruke koje sadrže samo jedan privitak (attachment) s točno traženim nazivom (određen konfiguracijom, mail.attachmentFilename) nazivamo NWTiS porukama, obrađuju se tako da se ispituje sadržaj datoteke privitka. Sadržaj datoteke treba biti u "text/json" ili "application/json" formatu i može biti sa sljedećem sintaksom:

* {"id": d{1-4}, "komanda":  "dodaj|azuziraj", ("{atribut {1-30}": "vrijednost d{1-3}|d{1-3}.d{1-2}|{1-30}"){1-5}, "vrijeme": "yyyy.MM.dd hh:mm:ss")}
* npr:
  + {"id": 1, "komanda": "dodaj", "naziv": "Senzor temperature", "vrijeme": "2018.04.08 11:20:45"}
  + {"id": 1, "komanda": "azuriraj", "temp": 20.4, "vrijeme": "2018.04.08 11:26:10"}
  + {"id": 2, "komanda": "dodaj", "naziv": "Senzor RFID", "korisnik": "pero", "vrijeme": "2018.04.09 17:21:56"}
  + {"id": 2, "komanda": "azuriraj", "korisnik": "mato", "vrijeme": "2018.04.09 21:09:01"}
  + {"id": 3, "komanda": "dodaj", "naziv": "Meteo stanica", "temp": 22.0, "vlaga": 77, "vjetar": "NW"  "vrijeme": "2018.04.09 18:19:41"}

Za poruku koja ima privitak ispravne sintakse slijedi provođenja postupka: ako se radi o komandi "dodaj" tada treba dodati zapis u tablicu UREDAJI u bazi podataka, s time da ako već postoji takav id, onda je pogreška. Ako se radi o komandi "azuriraj" tada se provjerava zapis u tablici UREDAJI za zadani id IoT i ako postoji, ažurira se u tablici. Ako ne postoji, onda je pogreška. Svaka primljena poruka upisuje se u tablicu DNEVNIK bez obzira na status prethodne akcije.

Obrađene NWTiS poruke prebacuju se u posebnu mapu (određena konfiguracijom, mail.folderNWTiS). Ako mapa ne postoji, dretva ju treba sama kreirati. Ostale poruke (koje nisu NWTiS poruke) treba ostaviti da imaju isti status (nepročitane ili pročitane) koje su imale prije nego što ih je dretva provjeravala.

Dretva na kraju svakog ciklusa dodaje podatke o radu u datoteku (naziv određen konfiguracijom, mail.threadCycleLogFilename, i sadržaj u UTF-8) u sljedećem obliku:

Obrada poruka broj:  
Obrada započela u: vrijeme\_1 (dd.MM.yyyy hh.mm.ss.zzz)  
Obrada završila u: vrijeme\_2 (dd.MM.yyyy hh.mm.ss.zzz)   
Trajanje obrade u ms: n  
Broj poruka: n - odnosi se na jedan ciklus  
Broj dodanih IOT: n   
Broj ažuriranih IOT: n   
Broj neispravnih poruka: n  
<prazan redak>

Potrebno je napraviti korisnički dio web aplikacije za izbor jezika za lokalizaciju, slanje email poruka, pregled primljenih email poruka i pregled dnevnika.

Izbor jezika za lokalizaciju (slika 1.) provodi se između raspoloživih jezika u koje nije uključen trenutno važeći jezik za lokalizaciju (za to se koristi atribut rendered). Koriste se gumbi koji na svojim akcijama pozivaju funkciju za postavljanje jezika (lokalizacije) uz argument koji odgovara odabranom jeziku. Svi statički elementi u pogledima trebaju biti pripremljeni da se prikažu u skladu s izabranih jezikom. Sve poruke (validacija i sl) u stalim pogledima trebaju biti prikazane u skladu s izabranih jezikom. U zaglavlju svih pogleda treba biti poveznica na izbor jezika, poveznica na slanje email poruke, poveznica za pregled primljenih email poruka i poveznica za pregled dnevnika. Ne prikazuje se ona poveznica koja se odnosi na važeći pogled. 

Kod odabira slanja email poruka (slika 2.) slijedi prikaz u obliku tablice (ili rešetke-grid) unutar koje je obrazac za unos podataka poruke (primatelj, pošiljatelj, predmet, sadržaj za privitak, brisanje sadržaja za privitak, preuzimanja sadržaja jedne od pripremljenih datoteka u element (kontrolu) sadržaja za privitak. Inicijalno adresa primatelja je iz postavke konfguracije mail.usernameThread. Inicijalno adresa pošiljatelja je iz postavke konfguracije mail.usernameEmailAddress. Inicijalno predmet je iz postavke konfguracije mail.subjectEmail. Potrebno je provjeriti ispravnost (validirati) upisanih podataka za primatelja, pošiljatelja, predmet poruke i sadržaj za privitak. Ispravnost primatelja i pošiljatelja temelji se na korektnost strukture email adrese. Predmet poruke mora sadržavati min 10 znakova. Sadržaj za privitak treba biti ispravan json tekst. U podnožju tablice nalazi se gumb za slanje email poruke. U zaglavlju pogleda treba biti poveznica na izbor jezika, poveznica na pregled primljenih email poruka i poveznica na pregled dnevnika

Kod odabira pregleda primljenih email poruka (slika 3.) prikazuje se padajući izbornik u kojem su elementi mape (direktoriji) iz email korisničkog računa (radi se o objektima klasa javax.mail.Folder i javax.mail.Store) i gump za promjenu mape. Od svih mapa samo su od interesa i jedino se prikazuju mapa INBOX i mapa koja je zadana konfiguracijom za NWTiS poruke (određena konfiguracijom, mail.folderNWTiS, a sadrži naziv u obliku NWTiS {LDAP\_korisnik} poruke) ako trenutno postoji. Na početku je odabrana mapa INBOX. Slijedi tablica s prikazom informacija o n (određen konfiguracijom, mail.numMessagesToShow) najsvježijih poruka (objekti klase javax.mail.Message) iz odabrane mape. Za  svaku email poruku potrebno je prikazati tko je poslao, kada je poslao (hr formatirano dd.mm.gggg hh:mm:ss), predmet poruke, vrsta poruke (NWTiS ili ne NWTiS), sadržaj privitka ako je NWTiS poruka, ako nije NWTiS poruka onda ništa. **NE SMIJU** se čitati sve poruke iz mape nego samo onoliko koliko je potrebno za prikaz prema postavki. Ispod tablice prikazuje se ukupan broj email poruka u izabranoj mapi, gumb za prethodne i sljedeće poruke. Pojedini gumb treba sakriti (koristi se atribut rendered) ako nema sadržaja koji o njemu ovisi. Npr. na početku nema prethodnih poruka. Aktiviranjem pojedinog gumba treba prikazati izabrani skup email poruka. U zaglavlju treba biti poveznica na izbor jezika, poveznica na slanje email poruke i poveznica na pregled dnevnika.

Kod odabira pregleda dnevnika (slika 4.) prikazuje se odabir intervala Od - Do i gump za pregled. Upisanim elementima Od i Do treba provjeriti ispravnost (validirati) u formatu datumskog tipa podatka (dd.mm.gggg hh:mm:ss). Slijedi tablica s prikazom informacija o n (određen konfiguracijom, mail.numLogItemsToShow) najsvježijih zaspisa u dnevniku za traženi interval. Za svaki zapis iz dnevnika potrebno je prikazati kada je spremljen i njegov sadržaj. **NE SMIJU** se čitati u memoriju svi zapisi iz dnevnika nego samo onoliko koliko je potrebno prema postavki. Ispod tablice prikazuje se ukupan broj zapisa u dnevniku koji zadovoljavaju interval Od Do, gumb za prethodne i sljedeće zapise dnevnika. Pojedini gumb treba sakriti (koristi se atribut rendered) ako nema sadržaja koji o njemu ovisi. Npr. na početku nema prethodnih zapisa iz dnevnika. Aktiviranjem pojedinog gumba treba prikazati izabrani skup zapisa iz dnevnika. U zaglavlju treba biti poveznica na izbor jezika, poveznica na slanje email poruke i poveznica na pregled poruka.

Pregledi trebaju biti vizualno sređeni pomoću css-a.

Metode u klasama NE smiju imati više od 25 linija programskog koda, u što se ne broji definiranje metode, njenih argumenata i lokalnih varijabli. U jednoj liniji može biti jedna instrukcija. Linija ne može ima više od 132 znaka.

JavaMail API nalazi se na adresi: <https://javamail.java.net/nonav/docs/api/>

JavaMail FAQ nalaze se na adresi <http://www.oracle.com/technetwork/java/faq-135477.html>