

SPAS - loppuraportti

Study Plan And Schedule, kurssihallinta ja lukujärjestyksen teko

Lauri Lavanti

15.1.2013

Tämä tiedosto on loppuraportti Studio 1-kurssin projektina tekemästäni SPAS-ohjelmasta, jonka avulla käyttäjä voi tarkkailla kurssiensa määrää lukuvuosi- ja periodikohtaisesti, sekä rakentaa automaattisesti lukujärjestyksen. Raportissa kuvailen syitä ratkaisuihini sekä tuntemuksiani projektin teon aikana.

Sisällysluettelo

1.	Kuvaus.....	2
1.1	Toiminnallisuus.....	2
1.2	Käyttöliittymä	2
2.	Tekninen toteutus	4
2.1	Toteutusympäristö	4
2.2	Toteutuksen pääpiirteet	5
3.	Kokemukset	6
3.1	Suunnitteluvaiheen ratkaisut	6
3.2	Isommat bugit.....	7
3.3	Työskentelytapa	7
3.4	Kuluneet resurssit.....	7
4.	Yhteenveto	7
4.1	Oma arvio	7
4.2	Tärkeimmät opit	8
5.	Lähteet ja liitteet	8
5.1	Liitteet.....	8
5.2	Lähteet.....	8

1. Kuvaus

SPAS on kaikille Aalto-yliopiston opiskelijoille suunnattu palvelu, joka korvaa tarpeen tehdä jokaisessa periodissa uusi lukujärjestys, tekemällä sen automaattisesti. Käyttäjän tarvitsee vain rekisteröityä palveluun, lisätä haluamansa kurssit kyseiselle periodille/lukuvuodelle, tai halutessaan vaikka viidenkin vuoden päähän, ja luoda kalenteri-tiedosto. Kalenteri-tiedosto itsessään on iCalendar-standardia noudattava ics-tiedosto, jonka voi lisätä sitten linkillä tai lataamalla esimerkiksi Gmailin kalenteriin tai Applen omaan kalenteriohjelmaan.

1.1 Toiminnallisuus

Palvelun toiminnallisuus keskittyy neljään eri osa-alueeseen. Kurssien hakuun, jossa käyttäjä voi hakea kursseja joko hakusanalla ja/tai valitsemalla organisaation ja/tai osaston, josta kursseja hakee. Kurssien hallintaan, jossa käyttäjä näkee miten hänen valitsemansa kurssit jakautuvat eri lukuvuosille ja periodeille sekä kuinka paljon opintopisteitä kertyy per periodi, sieltä käyttäjä myös pääsee vaihtamaan suoritusajoja ja ryhmiä kurssikohtaisesti (se onnistuu myös haun kautta). Käyttäjähallinta, jossa käyttäjä voi vaihtaa sähköpostinsa (jota käytetään, mikäli käyttäjä unohtaa salasanan), salasanan tai poistaa käyttäjänsä. Sekä viimeisenä ja tärkeimpänä lukujärjestyksen tekeminen, johon lisätään vain käyttäjän käynnissä olevalle periodille määrittelemät kurssit ja valittuihin ryhmiin kuuluvat harjoitukset.

1.2 Käyttöliittymä

Palvelu on internetsivusto, jossa yläreunassa on valikko, sekä käyttäjän nimi (josta hän pääsee käyttäjätietojensa hallintaan) ja uloskirjautumis-linkki. Valikon kautta pääsee käsiksi kaikkiin palvelun eri toimintoihin ja näkymiin. Mikäli käyttäjä ei ole kirjautunut sisään, ei hän pääse etusivua edemmäs, josta pääsee rekisteröitymissivulle sekä salasanan palauttamissivulle.

Seuraavalla sivulla ylin kuva on palvelun etusivusta ilman sisäänkirjautumista (vasemmalla) ja sisäänkirjautumisen jälkeen (oikealla). Seuraava kuva sisältää kurssihaku-näkymän ja kurssien hallintasivun. Viimeinen kuva on kurssin (yksittäisen) hallinnointisivusta ja kalenterin luomis-/päivitys-sivusta.

Ulkoasu on erittäin karkea, ja siihen on syynä se, että minun annettiin ymmärtää, ettei projektissa arvioida kuin Javalla tehdyt osiot, kun taas internetsivujen ulkoasu tehdään HTML:n ja CSS:n avulla, joten en niihin jättänyt aikaa vaan panostin serveri-puolen ohjelman koodiin.

Palvelun etusivu ennen ja jälkeen sisäänkirjautumisen.

lauri, kirjaudu ulos.	
Etusivu Haku Kurssihallinta Kalenteri	Etusivu Haku Kurssihallinta Kalenteri
SPAS - Study Plan And Schedule.	SPAS - Study Plan And Schedule.
Käyttäjänimi: <input type="text"/>	SPAS on palvelu, jolla Aalto-yliopiston opiskelijat voivat suunnitella opintojaan (näyttää kuinka monta opintopistettä on per periodi) ja generoida lukujärjestyksen nykyiselle periodille.
Salasana: <input type="password"/>	Tulossa: Viikkonäkymä. Kielivaihtoehdot (suomi, englanti).
<input type="button" value="Kirjaudu"/>	
<input type="checkbox"/> Pidä minut sisäänkirjautuneena.	
Unohditko salasanasasi? Rekisteröidy.	
SPAS on palvelu, jolla Aalto-yliopiston opiskelijat voivat suunnitella opintojaan (näyttää kuinka monta opintopistettä on per periodi) ja generoida lukujärjestyksen nykyiselle periodille.	
Tulossa: Viikkonäkymä. Kielivaihtoehdot (suomi, englanti).	
© Lauri Lavanti 2013. Ongelmatilanteissa laurilavanti(at)gmail.com	© Lauri Lavanti 2013. Ongelmatilanteissa laurilavanti(at)gmail.com

Kurssien haku ja hallinnointi.

lauri, kirjaudu ulos.	
Etusivu Haku Kurssihallinta Kalenteri	Etusivu Haku Kurssihallinta Kalenteri
Avainsana: <input type="text" value="Japani"/>	Näytä menneet lukuvuodet.
Organisaatio: <input type="text" value="Perustieteiden korkeakoulu"/>	Lukuvuosi 2012-2013
Osasto: <input type="text" value="Kielikeskus"/>	II (28)
<input type="button" value="Hae"/>	III (43)
Japani 1	Mat-1.1110: Matematiikan peruskurssi C1
Japani 2	T-106.1250: Informaatioverkostojen ohjelmointikurssi
Japani 3	T-106.2001: Informaatioverkostot: Studio 1
Japani 4	TU-22.1101: Tuotantotalouden peruskurssi
Japani 5 (o.w)	
Japani 6 (o.w)	
Käytännön japania	Inf-0.1220: Sosiologia
Japanilaista kulttuuria 1	Kie-98.3015: Japani 1
Japanilaista kulttuuria 2	Kie-98.3026: Japani 2
Syventäviä japanin kielen opintoja (o.w)	Mat-1.1120: Matematiikan peruskurssi C2
Tekniikan japania (o.w)	T-110.1100: Johdatus tietoliikenteeseen ja multimediatekniikkaan
Tekniikan japanin seminaari 1 (o.w)	T-111.1100: Digitaalisen median työt
Tekniikan japanin seminaari 2 (o.w)	T-75.2300: Informaatioverkostot: Studio 2
	T71C00800: Viestinnän ja mediatutkimuksen perusteet
	TU-91.2500: Marketing
	Lukuvuosi 2013-2014
© Lauri Lavanti 2013. Ongelmatilanteissa laurilavanti(at)gmail.com	© Lauri Lavanti 2013. Ongelmatilanteissa laurilavanti(at)gmail.com

Yksittäisen kurssin hallinta, sekä kalenterin luonti/päivitys.

lauri, kirjaudu ulos.

lauri, kirjaudu ulos.

Etusivu Haku Kurssihallinta Kalenteri

Etusivu Haku Kurssihallinta Kalenteri

Mat-1.1120: Matematiikan peruskurssi C2
Opintopisteet 10
Opetusperiodi(t) III-IV

Sisältö
Aina välttämätön aines
Osittaisderivaatta,
tilavuusintegraali, gradientti,
Taylorin sarja, tavalliset
differentiaaliyhtälöt, ryhmän
käsite. Usein tarpeellinen aines
Suppenemistestit, muuttujan
vaihto tilavuusintegraalissa,
Lagrangen kertoimet (optimointi),
Eulerin menetelmä, aliryhmät.
Joskus hyödyllinen aines
Avaruusintegraalit
pallokoordinaateissa, radat.

Alla oleva linkki johtaa kalenteriisi (online-importtaus, esim. gmail).

Lataa tästä.

Suoritusperiodi:

Suoritusvuosi:

© Lauri Lavanti 2013. Ongelmatilanteissa laurilavanti(at)gmail.com

© Lauri Lavanti 2013. Ongelmatilanteissa laurilavanti(at)gmail.com

2. Tekninen toteutus

2.1 Toteutusympäristö

Palvelun ensimmäinen ja ehkä tärkein paketti (*spas.nhandling*) sisältää luokkia tiedon siirtämistä ja lukemista varten Nopan API:sta olio-muotoon. Olennaisimmat ovat Nopan tietojen olio-esitelmät (*NElement* ja sen perivät luokat), joissa käytetään perinteisiä *java.lang* paketin luokkia ja *Calendar*-luokkaa tapahtumien ajankohdan esittämiseen. Lisäksi paketissa on kaksi luokkaa joiden avulla Nopan API:n tarjoamat XML-tiedostot muunnetaan olio-muotoon. Niistä *NReader* sisältää metodit, joilla haetaan oikea tiedosto Nopan API:sta (käyttäen hyväksi Javan sisäänrakennettuja *InputSource*- ja *URL*-luokkia). *XMLParser* sen sijaan sisältää metodit, joilla *NReader*:n antamat *InputSource*:t muunnetaan olio-muotoon. Se käyttää tähän hyväkseen DOM-mallinnusta ja *java.text*- sekä *java.util*-pakettien luokkia, joilla saa muunnettua tekstiä ajanjaksoiksi.

Seuraava paketti ja yhtä oleellinen kuin edellinen (*spas.usercontrol*) puolestaan sisältää luokat käyttäjähallintaan. Luokat *UserHandler* ja *UserLoginHandler* huolehtivat käyttäjätietojen tallennuksesta, muuttamisesta ja tarkistamisesta, toinen turvallisuus-näkökulmasta ja toinen tiedoston muokkauksesta. Ne käyttävät erityisesti samanlaista DOM-mallinnusta kuin *spas.nhandling*-paketin *XMLParser* ja Javan sisäänrakennettua *SecureRandom*-luokkaa. Luokka *UserCourseHandler* puolestaan huolehtii käyttäjän kurssien ja kurssien tietojen tallentamisesta ja muokkaamisesta saman DOM-mallinnuksen avulla. Paketin yksinkertaisin luokka *EmailNewPassword* tekee vain ja ainoastaan sen

mitä sen nimikin lupaa, eli arpoo käyttäjälle uuden salasanan, tallentaa sen ja lähettää sen käyttäjän sähköpostiin J2EE:n valmiiden luokkien avulla. Viimeinen ja koko palvelun tärkein luokka *CalendarCreator* (luokka, jonka ympärille koko palvelu perustuu) puolestaan rakentaa käyttäjän tallentamien kurssien ja Aallon kotisivuilta parsimansa tämän hetkisen periodin avulla iCalendar-tiedoston [iCal4j](#)-paketin avulla.

Viimeinen paketti ohjelmassa (*spas.taglib*) sisältää JSP-sivuilla käytettävien itse tehtyjen *tagien* käytännön toteutuksen. Kaikki paketissa olevat luokat perivät J2EE:n luokan *TagSupport* ja ovat edelleen jaettu kahteen pakettiin (*handlers* ja *printers*) sen mukaan, kuuluvatko ne MVC-mallin View- vai Controller-tasoon. Ne kaikki ovat riippuvaisia muista ohjelman luokista, ja toimivatkin rajapintana JSP-sivujen ja Java-luokkien välillä.

Ainoa luokka, joka ei ole omassa paketissaan on *XMLTools*, joka sisältää apumetodeja XML-tietojen käsittelyyn ja muuntamiseen DOM-muotoon sekä toisinpäin.

Viimeisenä pitää tietenkin muistaa kaikki JSP-sivut, joissa hyötykäytän itse tekemiäni *tageja* sekä J2EE:hen kuuluvan JSTL-kirjaston *core*-pakkauksen *tageja*.

2.2 Toteutuksen pääpiirteet

Alla on kuvattuna ohjelman luokkahierarkia ja – rakenne, sekä lyhyesti jokaisen luokan toimintaa.

- **spas** (Ohjelman pääpaketti.)
 - *XMLTools* (XML:n käsittelyyn tarkoitettuja apumetodeja, joista kaikki ovat staattisia.)
 - **nhandling** (Sisältää Nopan tietojen käsittelyyn ja esittämiseen tehdyt luokat.)
 - *NReader* (Hakee Nopan API:sta XML-tiedostoja.)
 - *XMLParser* (Muuntaa Nopan XML-tiedostot olio-muotoon.)
 - **nelements** (Sisältää Nopan tietojen olio-esitykset.)
 - *NElement* (Kaikkien olio-muotoisten Noppa-tietojen isä, organisaatiot ja osastot.)
 - *NCourse* (Kurssin olio-esitys, periytyy *NElement*-luokasta.)
 - *NEvent* (Kurssitapahtumien olio-esitys, periytyy *NElement*-luokasta.)
 - *NElementType* (Enumeraatio erilaisista Nopasta tulevista tietotyypeistä.)
 - *NElementFactory* (Tehdas-luokka *NElement*:lle ja sen perillisille, vaatii *NElementType*-luokkaa toimiakseen.)
 - **usercontrol** (Sisältää käyttäjäjärjestelmän hallinnointi-luokat.)
 - *UserHandler* (Huolehtii käyttäjäjärjestelmän tietoturvasta, tarvitsee *UserLoginHandler*:a.)
 - *UserLoginHandler* (Huolehtii käyttäjätietojen tallennuksesta ja päivityksestä.)
 - *UserCourseHandler* (Huolehtii käyttäjän kurssien ja kurssitietojen tallennuksesta ja päivityksestä.)
 - *EmailNewPassword* (Vaihtaa käyttäjän salasanan uuteen, ja lähettää sen käyttäjälle, tarvitsee *UserLoginHandler*:a.)
 - *CalendarCreator* (Tekee ja päivittää käyttäjän kalenterin, tarvitsee *UserCourseHandler*:a.)

- **taglib** (Sisältää JSP-sivuissa käytettävien *tagien* luokka-toteutukset, paketin kaikki luokat tarvitsevat ohjelman muita luokkia.)
 - **Validation** (Tarkistaa käyttäjän sisäänkirjautumisen oikeellisuuden.)
 - **printers** (Sisältää JSP-sivuille tulostavia *tageja*.)
 - *PrintOrgs* (Tulostaa Aalto-yliopiston organisaatiot *select*-valikkoon.)
 - *PrintDepts* (Tulostaa Aalto-yliopiston osastot *select*-valikkoon.)
 - *SearchCourses* (Tulostaa hakehtoja vastaavat kurssit linkkeinä.)
 - *PrintCourses* (Tulostaa käyttäjän tiedoissa olevat kurssit lukuvuosi- ja periodikohtaisesti.)
 - *PrintCourse* (Tulostaa yksittäisen kurssin tiedot, ja valikot joilla sitä voi muokata, mikäli käyttäjä on lisännyt sen.)
 - **handler** (Sisältää JSP-sivujen lomakkeiden käsitteleviä *tageja*.)
 - *Register* (Rekisteröi käyttäjän tietokantaan.)
 - *NewPword* (Vaihtaa käyttäjän salasanan, mikäli käyttäjä on unohtanut vanhan.)
 - *LogIn* (Kirjaa käyttäjän sisään.)
 - *LogOut* (Kirjaa käyttäjän ulos.)
 - *EditCourse* (Muuttaa kurssin tietoja käyttäjän tiedoissa.)
 - *Calendar* (Luo tai päivittää käyttäjän kalenteri-tiedoston.)
 - *ChangeEmail* (Vaihtaa käyttäjän sähköpostin.)
 - *ChangePword* (Vaihtaa käyttäjän salasanan.)
 - *RemoveUser* (Poistaa käyttäjän.)
- *index.jsp* (Etusivu, jonne kaikki kirjautumattomat käyttäjät ohjataan.)
- *register.jsp* (Rekisteröintisivu.)
- *newpword.jsp* (Sivu, jolla käyttäjä voi pyytää uutta salasanaa sähköpostiinsa.)
- *welcome.jsp* (Etusivu, jonne kaikki kirjautuneet käyttäjät ohjataan.)
- *logout.jsp* (Sivu, jonka kautta kirjaudutaan ulos.)
- *search.jsp* (Sivu, jolla käyttäjä voi hakea kursseja.)
- *coursecontrol.jsp* (Sivu, jolla käyttäjä näkee kurssinsa, ja pääsee hallitsemaan niitä.)
- *course.jsp* (Sivu, jolla hallinnoidaan yksittäisiä kursseja.)
- *calendar.jsp* (Sivu, jolla käyttäjä pääsee päivittämään kalenterinsa.)
- *usercontrol.jsp* (Sivu, jolla käyttäjä voi hallinnoida omia tietojaan.)

3. Kokemukset

3.1 Suunnitteluvaiheen ratkaisut

Suunnitteluvaiheen ratkaisut osuivat oikein lähelle käytännön ratkaisuja. Suurimmat erot olivat suunnittelusta Joda-Time-paketista luopuminen iCal4j:n takia, sillä se vaati Javan omia aikaluokkia toimiakseen, sekä viikkonäkymäisen kalenterin teosta luopuminen, sillä se osoittautui turhan työlääksi ottaen huomioon, että sopivia ohjelmia on lähes jokaisella käytössä jo valmiiksi.

3.2 Isommat bugit

Projektin teon aikana ei enimmäkseen ilmennyt suurempia bugeja, ongelmat liittyivät lähinnä uusien tekniikoiden ja menetelmien opetteluun ja niiden implementointiin projektiin. Lähinnä kaikki ongelmat olisi siis voitu välttää, jos olisin vain tiennyt valmiiksi hieman enemmän tarvittavista tekniikoista.

Mainitsemisen arvoinen oli tosin se, kun yritin siirtyä käyttämään XML:n lukemiseen ja kirjoittamiseen kirjastoa nimeltä [XOM](#). Ainoa ongelma sen kanssa oli kuitenkin se, että se suostuu lukemaan vain "valideja" XML-tiedostoja, eikä kelpuuttanut Nopan tarjoamia.

3.3 Työskentelytapa

Lähinnä työskentelytavoissa huonoa oli se, että aloitin turhan myöhään työskentelyn. Muuten toimin siten, että työskentelin projektin parissa noin yhdeksästä neljään päivittäin, pitäen aina taukoja silloin kun siltä tuntui ja jatkaen pidempäänkin, jos siltä tuntui.

Ongelmakohta tuli joululoman loppuilla, jolloin aloin päätyä siihen tulokseen, että ohjelma alkaisi olla lähes valmis, ja palkitsin itseni viikon "lomalla" projektista. Se tosin kostautui loppupeleissä, sillä olisin voinut saada koko projektin lopulliseen kuntoonsa jo ennen koulun alkamista.

3.4 Kuluneet resurssit

Projektin parissa työskentelyssä kului kyllä ehdottomasti eniten aikaa, sillä aina kun aloitin tekemään jotain uutta ominaisuutta, piti minun selvittää ensin miten sen pystyy edes tekemään Javalla. Lähinnä harmittaa siis se, ettei näitä Web-puolen Javan ominaisuuksia oltu enemmän käytä läpi kurssilla, mutta hyvin siitä kuitenkin omasta mielestäni selvisin.

4. Yhteenveto

4.1 Oma arvio

Omasta mielestäni projekti onnistui erittäin hyvin ja aikaakin kului kohtuullisesti. Oli myös erittäin mielenkiintoista opetella niin monia uusia asioita, kuten tietoturvallisuutta, XML:n käsittelyä ja JSP-teknologiaa. Lähinnä olisin voinut (ja luultavasti tulenkin myöhemmin, Studio 2:n puitteissa) panostamaan itse käyttöliittymän ulkoasuun ja dynaamisuuteen, mutta kun sitä ei kuitenkaan suoraan tässä arvioida, päätin (kuten edellä mainitsin) keskittyä muihin asioihin.

4.2 Tärkeimmät opit

Opin ehdottomasti sen, että tästä eteenpäin kun teen projekteja, joissa joudun käyttämään uusia resursseja, yritän ottaa huomattavasti enemmän selkoa niiden käytännön toteutuksesta ennen Eclipseen koskemista. Lisäksi opin, että useampia mahdollisia inputteja huomioonottavien ohjelmien teko on erittäin monipuolista, raskasta ja hankalaa.

5. Lähteet ja liitteet

5.1 Liitteet

Ohjelman lähdekoodi dokumentoituna on nähtävissä osoitteessa <https://github.com/SieniMaagi/SPAS>.

Olisin muuten laittanut myös itse ohjelman julkiseen käyttöön jollekin palvelimelle, mutta en tässä ajassa kerinnyt sellaista hankkimaan.

5.2 Lähteet

Lähteinä käytin seuraavia sivustoja:

- <http://docs.oracle.com/javase/7/docs/api/> (Java API)
- <http://docs.oracle.com/javaee/6/api/> (J2EE API)
- <http://tomcat.apache.org/> (Tomcat palvelin)
- http://wiki.modularity.net.au/ical4j/index.php?title=Main_Page (iCal4j kirjasto)
- <http://www.kanzaki.com/docs/ical/> (iCalendar standardit)
- <http://stackoverflow.com/> (Muiden tekemiä kysymyksiä ja niiden vastauksia)
- <http://docs.oracle.com/javase/tutorial/jaxp/index.html> (XML käsittely)
- <http://docs.oracle.com/javaee/6/tutorial/doc/> (JSP yms.)
- http://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm (JSTL)
- <http://www.hscripts.com/tutorials/jsp/jstltags/jstl.php> (JSTL)
- http://www.tutorialspoint.com/java/java_sending_email.htm (Sähköpostin lähettäminen)
- <http://www.mkyong.com/java/how-to-create-xml-file-in-java-dom/> (XML käsittely)
- <http://javarevisited.blogspot.fi/> (Vähän kaikkea)