

PORTTI-PROJEKTI

Juha Erkkilä
Jenni Hytönen
Marko Kivelä
Paula Mali
Lari Väänänen

Sovellussuunnitelma

22.5.2003

Tietoja projektista ja dokumentista

Ryhmän jäsenet:

- Juha Erkkilä (erkkila@cc.jyu.fi),
- Jenni Hytönen (jmhytone@cc.jyu.fi),
- Marko Kivelä (markoki@cc.jyu.fi),
- Paula Mali (pkmali@cc.jyu.fi) ja
- Lari Väänänen (lamava@cc.jyu.fi).

Tilaaajat: Jyväskylän yliopisto

- Antti Auer, koordinaattori, virtuaaliyliopisto,
- Mikko Koljander, ATK-suunnittelija, suunnittelu ja kehittäminen,
- Anu Mustonen, viestintäpäällikkö, viestintä,
- Pekka Olsbo, julkaisukoordinaattori, kirjasto,
- Jussi Talaskivi, ATK-suunnittelija, ATK-keskus,
- Mauno Väisänen, tietohallintopäällikkö, hallintovirasto sekä
- Tarja Vänskä-Kauhanen, tiedottaja, viestintä.

Asettaja: Jyväskylän yliopisto, tietotekniikan laitos

- Markus Inkeroinen, vastaava ohjaaja,
- Jukka-Pekka Santanen, vastaava ohjaaja sekä
- Matti-Pekka Sivosuo, tekninen ohjaaja.

Yhteystiedot: Agora C222.2, 014-260 4963, portti@korppi.it.jyu.fi

Kotisivu: <http://kotka.it.jyu.fi/portti/>

Työn nimi: Portti-projekti, Sovellussuunnitelma

Työ: Sovellussuunnitelma, tietotekniikan Sovellusprojekti

Tiivistelmä: PORTTI-projekti kehittää seuraajan Jyväskylän yliopiston hallintoviraston viestinnän WWW-pohjaiselle julkaisemislomakkeistolle. Sovellussuunnitelmassa esitellään sovelluksen sisäinen rakenne sekä esitellään testaustapoja ja tietoturvaan liittyviä asioita.

Avainsanat: Tietokanta, Zope, Tutka-järjestelmä, henkilöstökoulutus, tiedote, tapahtuma, Sovellusprojekti, lomakkeisto, Sovellussuunnitelma.

Versiohallinta

Taulukossa 1 on esitetty PORTTI-projektin Sovellussuunnitelman versiohistoria.

Versio	Päivämäärä	Tekijät	Muutokset
0.1	13.3.2003	Väänänen	Luotu dokumentti ja muodostettu otsikkojako
0.2	17.3.2003	Mali	Lisää tekstiä.
0.3	20.3.2003	Mali	Lisää tekstiä.
0.4	27.3.2003	Mali	Korjauksia ja täydennystä.
0.5	2.4.2003	Mali	Korjauksia ja täydennystä.
0.6	3.4.2003	Mali	Korjauksia ja täydennystä.
0.7	8.4.2003	Väänänen, Mali	Korjauksia ja täydennystä.
0.8	14.4.2003	Mali	Korjauksia ja täydennystä.
0.9	24.4.2003	Mali	Korjauksia.
1.0	28.4.2003	Mali	Korjauksia.
1.1	22.5.2003	Mali	Korjauksia.

Taulukko 1. Sovellussuunnitelman versiohallintaa.

Hyväksytty:

Pvm.....Allekirjoitus.....

Pvm.....Allekirjoitus.....

Termiluettelo

Aihealueeseen liittyviä termejä ovat seuraavat:

Henkilöstökoulutus	tarkoittaa Jyväskylän yliopiston henkilökunnalle suunnattua koulutusta.
Henkilöstökoulutuksen tapahtuma	on esimerkiksi kurssi, luentosarja tai vierailuluento.
MS Project	on projektin suunnitteluun ja hallintaan liittyvien aikataulujen ja kaavioiden visualisointiin ja suunnitteluun käytettävä työkalu.
Portaali	on WWW-sivusto, jossa on erilaisia osioita erityyppisille käyttäjille. Sivusto voi sisältää mm. linkkejä ja hakupalveluja.
Tapahtuma	on tilaisuus, jolla on aika ja paikka. Tapahtumia ovat esimerkiksi väitöstilaisuus, vierailuluento, seminaari tai teatteri/musiikkiesitys.
Tapahtumakalenteri	on Jyväskylän yliopiston erilaisten tapahtumien tiedotusfoorumi.
Tarkastuspiste	on tilaisuus, jossa hyväksytään valmistuneet dokumentit tai muut tulokset.
Tiedote	käsittää tiedot esimerkiksi väitöksistä, avoimista viroista ja apurahoista. Tiedotteet julkistetaan Ajankohtaista-sivulla.
Tutka-järjestelmä	on Jyväskylän yliopiston hankkeisiin, tutkimukseen, julkaisuihin, vierailuihin ja muuhun tieteelliseen toimintaan liittyvän tiedon keräämiseen tarkoitettu tietojärjestelmä.

Ohjelmointitekniikoihin ja -työkaluihin liittyviä termejä ovat seuraavat:

Apache	on verkkopalvelinohjelmisto, jonka tehtäviin kuuluu HTTP-kyselyihin vastaaminen ja kysytyn sivun lähettäminen asiakkaalle. Apachessa on lisäksi mahdollista tuottaa dynaamisia sivuja erilaisten ohjelmointirajapintojen kautta.
CSS	on WWW-sivujen ulkoasun määrittämiseen käytettävä kieli.
HTML	on WWW-sivujen sisällön, rakenteen ja ulkoasun tekemiseen käytettävä merkkakieli.

Java-pavut	(engl. <i>Java Beans</i>) ovat Java-ohjelmointikielellä luotuja komponentteja, joita voidaan kutsua JSP-sivuilla.
JSP	(Java Server Pages) on skriptaustyylinen ohjelmointikieli, joka mahdollistaa dynaamisten WWW-sivujen toteuttamisen. JSP-tiedosto muodostuu HTML- ja Java-koodista sekä Java-papujen kutsuista.
Lotus Notes	on sovelluskehitysympäristö, joka käsittää mm. työpöydän, kansiot, tietokannan ja käyttöoikeuksien määrittelyn sekä mahdollisuuden käyttöön WWW-sivujen kautta.
PostgreSQL	on vapaan lähdekoodin lisenssin alainen relaatiotietokannan hallintajärjestelmä.
SQL-kyselykieli	sisältää käskyjä tietokannan rakenteen määrittelyyn ja muuttamiseen, tietojen lisäämiseen ja muuttamiseen sekä kyselyjen tekemiseen.
Tietokanta	on kokoelma tiettyä kohdetta kuvaavia tietoja, joita yksi tai useampi tietojärjestelmä käyttää ja päivittää.
Tomcat	on ilmainen sovellusohjelmisto, joka muuntaa JSP-tiedoston selaimella esitettäväksi HTML-tiedostoksi.
Tuote	Zopen oma ”moduuli” tiedon kokoamiseen. Yksi tuote voi olla esimerkiksi täysin toimiva ohjelmisto, jonka voi ottaa käyttöön Zopen add-listasta.
WWW	(World Wide Web) on maailman laajuinen tietoverkko.
XML	(eXtensible Markup Language) on itse luotu merkintäkieli, johon käyttäjä voi luoda rakenteen ja kieliopin omien tarpeidensa mukaan.
ZClass	Zopen vastine perinteisen ohjelmoinnin luokalle.
Zope	(Z Object Publishing Environment) on kehitystyökalu ja sovellusalusta, jolla voidaan kehittää ja ylläpitää WWW-pohjaisia sovelluksia.

Sisällysluettelo

1	JOHDANTO	1
2	VAATIMUSTEN PRIORISOINTI	2
3	SOVELLUKSEN RAKENTEEN KUVAUS	3
3.1	YLIOPISTON PORTAALI.....	3
3.2	TIETOKANTA	4
3.3	OHJELMOINTI.....	4
3.4	LUOKKARAKENNE.....	4
3.5	LUOKKIEN ATTRIBUUTIT	6
3.6	VALMIIT KOMPONENTIT	6
4	KÄYTTÖLIITTYMÄN KUVAUS	7
4.1	MUUTOKSET EDELÄJÄÄN VERRATTUNA	7
4.2	ESIMERKKEJÄ KÄYTTÖLIITTYMÄN SIVUJEN MUUTOKSISTA.....	8
5	TIETOTURVA	10
5.1	KÄYTTÄJIEN OIKEUDET.....	10
5.2	PALVELUN ESTÄMINEN	10
5.3	LOKITIEDOSTO	11
6	LÄHDEKOODIN KOMMENTOINTI JA NIMEÄMISKÄYTÄNNÖT	12
7	SOVELLUKSEN TESTAUSPERIAATTEET	14
7.1	MODUULITESTAUS	14
7.2	INTEGROINTITESTAUS	14
7.3	JÄRJESTELMÄTESTAUS	14
8	YHTEENVETO	15
	LÄHTEET	16

1 Johdanto

PORTTI-projekti suunnittelee ja toteuttaa Jyväskylän yliopiston hallintoviraston viestinnälle WWW-sivujen kautta toimivan julkaisemislomakkeiston, joka aloittaa yliopistossa suunnitteilla olevan portaalihankkeen. Projekti on Jyväskylän yliopiston tietotekniikan Sovellusprojekti.

Projekti kehittää seuraajan nykyiselle, Lotus Notes -pohjaiselle WWW-sivustolle Zope-työkalulla. Sivusto käsittää tiedotteiden, tapahtumailmoitusten sekä henkilöstökoulutuskalenterin tapahtumien julkaisun. Tietojen tallentamiseen käytetään Zopen oliotietokantaa.

Tämä suunnitelma kuvaa sovelluksen sisäistä rakennetta sekä käsittelee testaustapoja ja tietoturvaan liittyviä asioita. Sovellussuunnitelman lisäksi PORTTI-projekti laatii Vaatimusmäärittelyn ja Projektisuunnitelman. Vaatimusmäärittelyssä kuvataan toteutettavan järjestelmän yleinen rakenne, toiminnalliset ja tekniset vaatimukset, käyttäjäryhmät sekä lomakkeiden käyttömahdollisuudet. Projektisuunnitelmassa esitellään PORTTI-projektin taustoja, aikataulua ja muita projektin läpivientiin liittyviä asioita.

Luvussa 2 priorisoidaan sovellukselle asetettuja vaatimuksia. Luvussa 3 käsitellään sovelluksen rakennetta, ohjelmointia ja tietokantaa. Luvussa 4 esitellään toteutettavaa käyttöliittymää ja muutoksia nykyiseen sivustoon. Luvussa 5 käsitellään tietoturvaan liittyviä asioita. Luku 6 sisältää lähdekoodin kommentointia ja muuttujien nimeämistä. Luvussa 7 esitellään sovelluksen testausperiaatteita.

2 Vaatimusten priorisointi

Sovelluksen sivu- ja sivustorakenteen suunnittelun lähtökohtana on nykyisen sivuston käyttöliittymä. Sivujen sisältö on esitetty Vaatimusmäärittelyn luvussa 5. Toteutettavaa lomakkeistoa tulee pystyä käyttämään ainakin Vaatimusmäärittelyssä mainituilla selaimilla ja niiden versioilla. Käyttöliittymä toteutetaan XHTML 1.0 Strict -standardin mukaisesti ja sen ulkoasun muotoiluun käytetään CSS2-standardin mukaisia tyylilomakkeita.

Portti-projekti suunnittelee ja toteuttaa seuraavat, tilaajan kanssa sovitut ominaisuudet. Kaikkea ei välttämättä ehditä toteuttaa, joten projektissa sovellukselle asetettuja vaatimuksia on priorisoitu.

Portti-projektin puitteissa toteutetaan seuraavat toiminnot:

- 1) lomakkeiston siirtäminen Zopeen,
- 2) lomakkeiston muokkaaminen käytettävämmäksi,
- 3) käyttöoikeustasojen toteuttaminen,
- 4) mahdollisuus tiedotteen kuulumisesta useampaan tyyppiin.

Jos aikaa jää, toteutetaan myös seuraavat tehtävät:

- 5) lomakkeiston käyttö eri organisaatiotasolla,
- 6) ajankohtaista-sivun asetuslomakkeen luonti,
- 7) monikielisyyden tukeminen,
- 8) Sähköjyvä,
- 9) metatiedon lisääminen lomakkeisiin ja
- 10) vanhan aineiston siirto Lotus Notesista Zopeen,
- 11) sähköpostiryhmät.

Projektin puitteissa ei toteuteta rajapintoja Korppi- ja Tutka-järjestelmiin.

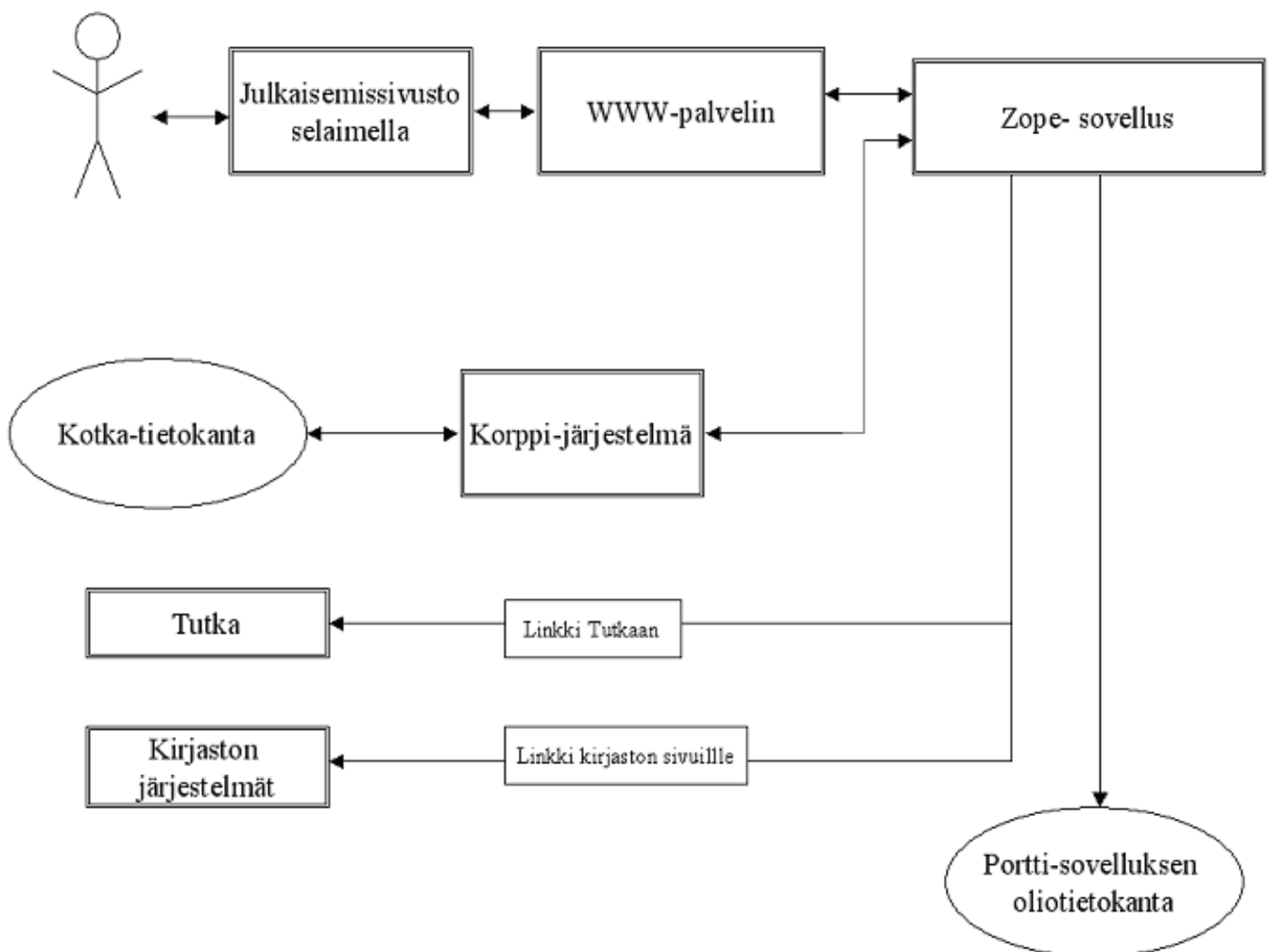
Henkilöstökoulutuksen kurseille täytyy joissakin tapauksissa ilmoittautua. Ilmoittautuminen tapahtuu sovelluksen tarjoaman lomakkeen avulla siten, että täytetyt tiedot joko tallennetaan Zopen tietokantaan tai lähetetään sähköpostitse kurssin luontilomakkeessa annettuun osoitteeseen. Sähköpostitse tai puhelimitse tapahtuneet ilmoittautumiset kirjataan käsin sovellukseen.

3 Sovelluksen rakenteen kuvaus

Luvussa käsitellään sovelluksen rakennetta luokkakaavion avulla ja esitellään ohjelmointiin ja tietokantoihin liittyviä ratkaisuja.

3.1 Yliopiston portaali

PORTTI-projektissa toteutettava tiedotuksen julkaisemislomakkeisto aloittaa yliopistossa suunnitteilla olevan portaalihankkeen toteuttamisen. Portaali tulee käsittämään useita WWW-sovelluksia, kuten Tutka- ja Korppi-järjestelmät, yliopiston kirjaston sekä virtuaaliyliopiston sovellukset sekä Portti-projektissa toteutettavan julkaisemislomakkeiston. Kaikki nämä sovellukset tulevat tarvittaessa olemaan kaikkien yliopiston organisaatioiden käytävissä. Kuvassa 1 on esitetty portaalien rakennetta ja sovelluksen yhteyksiä muihin järjestelmiin.



Kuva 1. Portaalien arkkitehtuuri.

3.2 Tietokanta

Henkilöstökoulutuksen ja tapahtumakalenterin tapahtumien sekä ajankohtaisten tiedotteiden tallennukseen suunnitellaan ja toteutetaan Zopella oliotietokanta. Oliokannan tauluihin tallennetaan kukin tapahtuma ja tiedote omaksi oliokseen, johon liittyy attribuutteja. Tallennettavat tiedot on esitetty Vaatimusmäärittelyn luvun 5 taulukoissa.

Zopen oliokanta toimii automaattisesti, eikä käyttäjän tarvitse paneutua sen rakenteen muokkaamiseen. Oliokanta ei periaatteessa ole aito oliokanta, vaan järjestelmän oliot tallentavat itsensä automaattisesti palvelimen kovalevylle tietokannan kaltaiseen muotoon.

3.3 Ohjelmointi

Portti-sovelluksen ohjelmointiin käytetään DTML- ja python-kieliä. Lomakkeisto luodaan XHTML 1.0 Strict -standardin mukaisesti ja ulkoasun muotoiluun käytetään CSS2-standardin mukaisia tyyliomakkeita.

Zopen katalogeja käytetään hakujen tekemiseen tietokannasta. Haut toteutetaan sana-, kategoria- ja/tai aikavälihakuina.

Python-skriptejä tullaan käyttämään mahdollisuuksien mukaan. Kaikki merkkijonojen käsittely ja monimutkaisemmat operaatiot tehdään pythonilla. Pythonissa on myös mahdollisuus tehdä monikäyttöisiä funktioita, joita voi käyttää skriptin sisällä sekä kutsua skriptin ulkopuolelta.

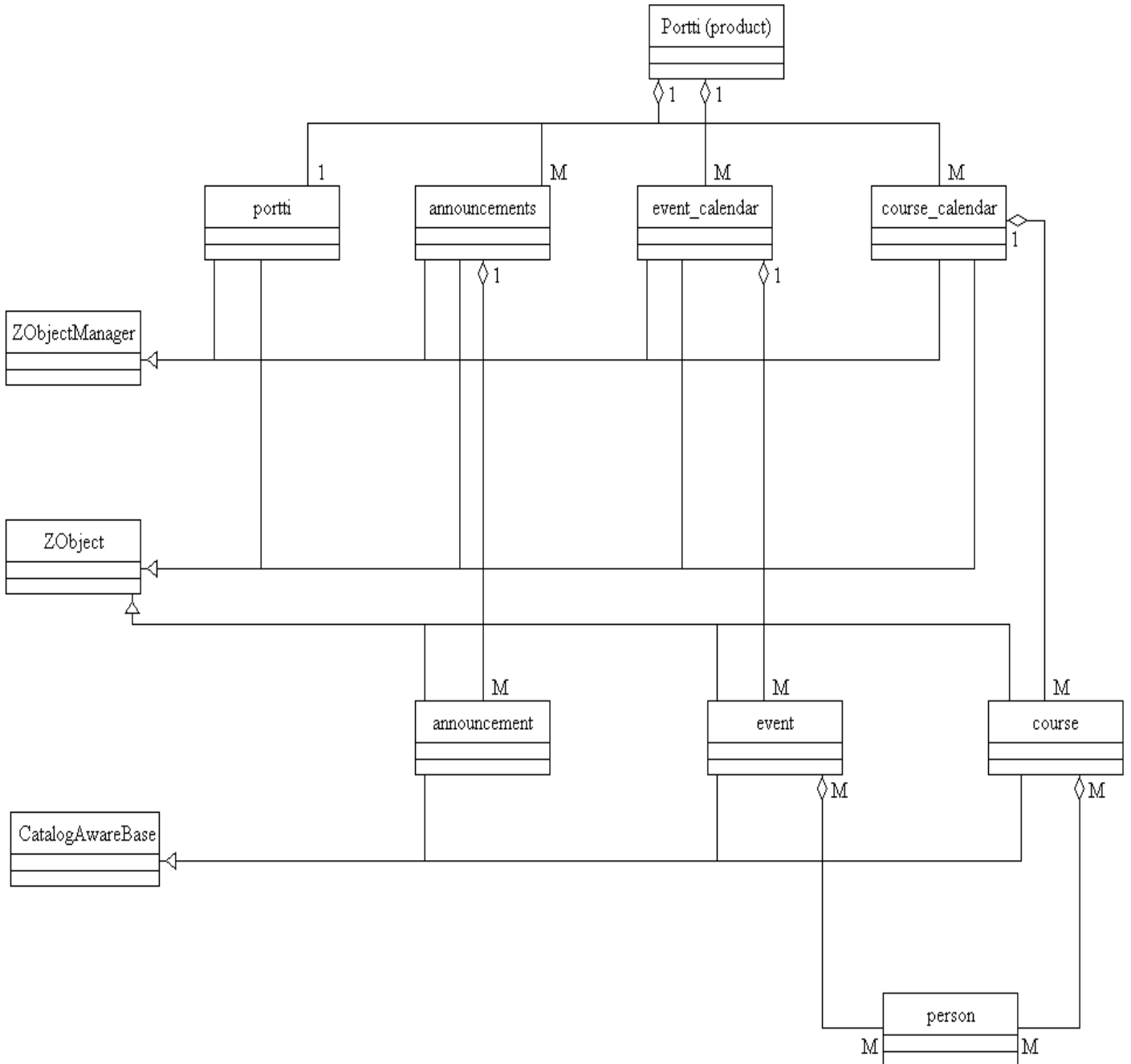
Itse ohjelmointi tullaan toteuttamaan osittain code-and-fix -tyylillä (ohjelmointia ilman tarkkaa suunnittelua) suunnitellun luokkarakenteen päälle. Metodit tullaan esittämään sovellusraportissa, koska tässä vaiheessa ei ole mahdollista tarkasti määrätä tulevia operaatioita luokkien sisällä. Tämä johtuu siitä, että toteutettava sovellus muuttuu ja elää koko ajan uusien ominaisuuksien lisäämisen vuoksi.

3.4 Luokkarakenne

Zopessa moduulia vastaa tuote, joka sisältää luokkia. Projektissa tuotteita luodaan ainakin yksi, mutta mahdollisesti useampikin. Tuotteen sisältämiä luokkia ovat ainakin Zopen omat luokat, kuten ZobjectManager, Zobject ja CatalogAwareBase sekä ryhmän luomat luokat portti, event_calendar ja event, course_calendar, course, announcements ja announcement sekä person.

event_calendar, course_calendar ja announcements luokat käsittävät kukin yhden olion, kun taas event, course, person ja announcement luokat käsittävät useita event-, course-, person- ja announcement-olioita. portti-luokka sisältää mm. tiedotusosion etusivun, sivuston tyylitiedoston ja kuvat sekä sivuston ylä- ja alatunnisteet. course ja event luokkiin liittyvään person-luokkaan tallennetaan henkilöstökoulutuksen kursseille ja tapahtumakalenterin tapahtumiin ilmoittautuneet.

Kuvassa 2 on esitetty UML-kaavio Portti-sovelluksen luokkarakenteesta. Zopessa on mahdollisuus periä luokkia toisista luokista. Toteutettavan sovelluksen tapauksessa ominaisuuksia peritään vain Zopen valmiista luokista. Muissa luokissa perintää ei hyödynnetä, koska ryhmällä ei ole kokemusta perinnästä Zopessa ja ominaisuuteen tutustuminen veisi aikaa.



Kuva 2. Sovelluksen luokkarakenne.

Sovellus on jaettu yhden tuotteen sisällä useaan osaan ZClasseilla. Jokainen luokka on oma osa-alueensa. Tällä tavoin saadaan aikaan järjestelmä, jossa tarvittaessa voidaan luoda monta rinnakkaista tapahtumakalenteria, kurssikalenteria tai mediatiedoteosiota. Luokkien attribuutit ja metodit tullaan esittelemään sovellusraportissa yksityiskohtaisesti.

Zopessa luokkarakenne on polkumainen. Esimerkiksi tapahtumakalenterin osalta se olisi seuraavanlainen: `Control_Panel/products/portti/event_calendar/property sheets/methods/event/property sheets/methods`. Oikeus päästä suoraan käsiksi polulla oleviin tietoihin tapahtuu ylhäältä alaspäin. Tämä tarkoittaa sitä, että esimerkiksi `event_calendar`-luokkaan tehtyä metodia voidaan kutsua `happening`-luokasta. Tällä tavoin polun yhteisiä operaatioita kannattaa sijoittaa ylemmälle polun tasolle.

3.5 Luokkien attribuutit

Luokkien attribuutit on määritellään `property sheet`-lomakkeilla, joihin kootaan luokkaan kuuluvat ominaisuudet. Yhteen `property sheet`tiin voi sisällyttää mielensä mukaan attribuutteja. Mahdollisia attribuuttien tyyppejä ovat `boolean`, `date`, `string`, `int`, `float`, `lines`, `long`, `text`, `tokens`, `selection` ja `multiple selection`. Attribuutteihin pääsee käsiksi `property sheet`in omistavan olion kautta.

3.6 Valmiit komponentit

Sovelluksessa tullaan käyttämään valmiita tuotteita mahdollisuuksien mukaan. Zopen omista ratkaisuista ainakin `ZCatalogit` tulevat olemaan osa haku-osiota. Projektin edetessä katsotaan, onko käyttötarkoitusta vastaavia ohjelmaosioita saatavilla. Jos täysin käyttötarkoitukseen vastaavaa komponenttia ei löydy, joudutaan miettimään, onko järkevämpää tehdä se alusta asti uudelleen, vai muokata valmis komponentti omaan käyttöön soveltuvaksi.

4 Käyttöliittymän kuvaus

Sovelluksen sivu- ja sivustorakenteen suunnittelun lähtökohtana on nykyisen sivuston käyttöliittymä. Sivuston ulkoasu tulee pysymään nykyisellään. Käyttöliittymä toteutetaan XHTML 1.0 Strict -standardin mukaisesti ja sen ulkoasun muotoiluun käytetään CSS2-standardin mukaisia tyyliomakkeita.

4.1 Muutokset edeltäjään verrattuna

Käyttöliittymään tehtävät muutokset liittyvät linkkeihin, arkiston selaamiseen, hakulomakkeeseen sekä tapahtumakalenterin tapahtumien ja ajankohtaisten tiedotteiden luontilomakkeisiin.

Linkkeihin tehtävät muutokset ovat seuraavat:

- *Login*-teksti vaihdetaan *Sisäänkirjautumiseksi*,
- uloskirjautuminen lisätään,
- linkkien ryhmittelyä muutetaan,
- *Sulje*-linkki poistuu kokonaan sekä
- ylä- ja alapalkeissa oleva *Pääsivu* muuttuu *Yliopiston pääsivuksi*.

Lomakkeisiin tehtävät muutokset ovat seuraavat:

- lomakekenttien sijoittelua muutetaan,
- *Lähetä*-painike lisätään lomakkeen alalaitaan ja
- pakolliset kentät merkitään asteriskilla (*).

Arkiston selaamiseen tehtävä muutos on seuraava:

- linkeistä tehdään suurempia muuttamalla linkiksi vuosiluku tai kuukausi entisen pienen kolmion sijaan.

Hakulomakkeeseen tehtävä muutos on seuraava:

- yksi hakusivu useilla hakukentillä monen eri hakusivun sijaan.

4.2 Esimerkkejä käyttöliittymän sivujen muutoksista

Kuvassa 3 on esitetty *Mediatiedotteet*-sivuston etusivu. Linkkeihin tehtävät muutokset näkyvät ala- ja yläpalkeissa (*Yliopiston pääsivu*) sekä vasemman reunan linkkilistassa (*Sisäänkirjautuminen* ja *Uloskirjautuminen*).



Kuva 3. *Mediatiedotteet*-etusivu.

Kuvassa 4 on esitetty ajankohtaisiin tiedotteisiin tulevan tapahtuman luontilomake. Lomakekenttien sijoittelua on muokattu nykyiseen lomakkeeseen verrattuna. Pakolliset kentät on merkattu yleisesti käytetyllä asteriskilla (*).



Tiedotteet
Arkisto
Haku

Näyttelyt
Väitökset

Käyttöohjeet

Uusi tiedote

Muodosta sähköjyvä
Julkaisemattomat
Hyväksyttävät
Roskakori
Uloskirjautuminen

Käyttäjä: tiedottaja

Tiedotteen julkaiseminen

Tällä lomakkeella voit luoda tiedotteen tai julkaista tekstidokumenteja. Muista täyttää ainakin tähdellä merkityt kentät. (* = kenttä on pakollinen) Yhteyshenkilön sähköpostiosoite on pakollinen, koska siihen osoitteeseen lähetetään tieto mahdollisesta hylkäyksestä.

Voimassaolo * - (pp.kk.vvvv)

Kellonaika (hh:mm)

Paikka

Julkaisupvm (jos eri kuin luontipvm)

Julkaisun kellonaika (hh:mm)

Tiedotetyyppi *

Otsikko *

Julkaistaan sähköjyvässä

Jos haluat, että tiedote julkaistaan Sähköjyvässä, voit luoda tekstistä tiivistelmän (noin 600 merkkiä).

Tiivistelmä

Tiedote/
tekstidokumentti *

Yhteyshenkilö

Sähköposti*

Organisaatio

Lisätiedot

Liitä kuva

Liitä tiedosto

Kuva 4. Tapahtumakalenterin tapahtuman luontilomake.

5 Tietoturva

Tietoturvalla tarkoitetaan toimenpiteitä, joilla varmistetaan tiedon luottamuksellisuus, eheys ja käytettävyys. Toteutettavaan sovellukseen tuotetaan uutta aineistoa sivuilla, joissa valmiita tapahtumia voidaan selaila, hakea ja päivittää. Koska näiden monien eri toimintojen toteuttamiseen tarvitaan eri tasoisia käyttäjiä, on tietoturva taattava. Zope mahdollistaa käyttäjien autentikoinnin ja sisältää kattavan oikeuksien hallinnan.

5.1 Käyttäjien oikeudet

Julkaisemislomakkeistolle määritetään eritasoisia käyttäjäryhmiä ja käyttöoikeuksia, jotka ovat organisaatiokohtaisia. Käyttäjäryhmät ovat peruskäyttäjä, tiedon syöttäjä, tiedottaja ja ylläpitäjä.

Organisaatiotasoja voivat olla mm. yliopisto, tiedekunta, laitos ja linja. Jokaisella tasolla on omat tiedon syöttäjänsä ja tiedottajansa, jotka laativat ja hyväksyvät omaan organisaatioonsa liittyvät tiedotteet ja tapahtumat. Julkaisemislomakkeiston ylläpitäjä toimii yliopistotasolla.

Tiedotetyyppikohtaisia oikeuksia ei tulla tarvitsemaan. Jokainen organisaatiossa tiedottajan oikeudet omaava saa muokata, poistaa tai hyväksyä kaikkia kyseisen organisaation ja sen aliorganisaatioiden tiedotetyyppejä.

Peruskäyttäjällä ei ole käyttäjätunnusta, joten hän lähinnä selailee tapahtumia ja tiedotteita. Peruskäyttäjä voi kuitenkin luoda tapahtumaehdotuksen tapahtumakalenteria varten. **Tiedon syöttäjä** laatii tiedotteita ja tapahtumia sekä tapahtumakalenteriin että henkilöstökoulutuksen tapahtumakalenteriin. **Tiedottajan** tulee kuitenkin hyväksyä ne ennen julkaisua. Hyväksymisen ja hylkäämisen lisäksi tiedottaja voi muokata ja poistaa jo julkaistuja tapahtumia ja tiedotteita. **Ylläpitäjä** on järjestelmän pääkäyttäjä. Ylläpitäjä voi muokata järjestelmän asetuksia sekä käyttäjätunnuksia.

5.2 Palvelun estäminen

Palvelun estäminen (engl. *denial of service*) tarkoittaa tietojärjestelmän kuormittamista siten, että tarjottavaa palvelua ei voi käyttää ainakaan tehokkaasti.

PORTTI-sovelluksen tapauksessa joku ilkeämielinen voi esim. asentaa tietokoneeseen ohjelman, joka lähettää WWW-palvelimelle jatkuvasti pyyntöjä avata *Ajankohtaista*-pääsivu. Tällöin palvelimen pyyntöjenkäsittelykapasiteetti on todennäköisesti riittämätön, jolloin *Ajankohtaista*-sivun tietoja ei voida lähettää todellisille asiakkaille. Toinen mahdollinen tapa estää palvelu on ohjata palvelunkäyttäjät esim. omalle kotisivulleen muuttamalla yliopiston WWW-palvelimen asetuksia.

Jos havaitaan suurta WWW-palvelimen kuormitusta ja palvelun estoyritystä jostain tietystä osoitteesta, osoite voidaan rajata Portti-sovelluksessa palveluiden ulkopuolelle Apachen avulla.

5.3 Lokitiedosto

Ylläpitäjän tulee pystyä seuraamaan järjestelmässä tehtyjä toimintoja. Toimintojen kirjaamiseen käytetään lokitiedostoa, johon tallentuu tehty toiminto sekä sen tekijä ja päivämäärä. Näin lokitiedoston avulla voidaan jäljittää mahdollisia virhetilanteita ja tietoturvariskejä. Lokitiedostoon tallentuvat myös tapahtuneet järjestelmävirheet.

6 Lähdekoodin kommentointi ja nimeämiskäytännöt

Ohjelmakoodi kommentoidaan sovitun käytännön mukaisesti englanniksi. Jokaisen lähdekooditiedoston alkuun kirjoitetaan projektin nimi, lisenssimerkintä, tekijän nimi, päivämäärä, tiedoston nimi ja kuvaus lähdekoodin tarkoituksesta. Lisäksi jokaisen metodin ja aliohjelman toiminta kommentoidaan erikseen.

Luokkien, olioiden, metodien ja attribuuttien nimet kirjoitetaan englanniksi aloittaen pienellä kirjaimella. Jos nimi koostuu kahdesta sanasta, sanojen väliin lisätään alaviiva. Myös jälkimmäinen sana aloitetaan pienellä kirjaimella.

Seuraavassa on esitetty esimerkki DTML-kielen mukaisesta kommentointitavasta:

```
<dtml-comment>
*****
*Project name: Porttti *
*Licence: AFL, version 1.2 *
*Metod name: show_as_link *
*Created: 2.4.2003 *
*Modified: *
* - Lari Väänänen 2.4.2003, creation *
* - Marko Kivelä 2.4.2003, repaired linking *
* *
*Description: This metod creates and shows a link from *
* the happening. All neccessary properties *
* are included. *
* *
* *
*COPYRIGHT (C) 2003 PORTTI-PROJEKTIRYHMÄ *
*****
</dtml-comment>

</dtml-comment>
* Get happenings and show then as links
</dtml-comment>

<dtml-with
expr="propertysheets.happening_properties.propertyValues
()">

  <A class= uutiset href="&dtml-
absolute_url;/show_happening"><dtml-var date_start>-
<dtml-var date_end> <dtml-var header>:</a>

</dtml-with>
```

Seuraavassa on esitetty python-esimerkki toteutetusta skriptistä.

```
#Project name: Portti
#Licence: AFL 1.2
#Method name: compare_date
#Created: 4.4.2003
#Modified:
  - Lari Väänänen 2.4.2003, creation
  - Marko Kivelä 2.4.2003, repaired linking
#Description: return true if date1 is same than date2

temp=str(m_date1)
date1=temp.split('/')
temp=str(m_date2)
date2=temp.split('/')

if ((date1[0] == date2[0]) and (date1[1] == date2[1]
    and (date1[2] ==date2[2]))) :
    return 0

return 1
```

Python-skripteissä ja DTML-metodeissa pyritään käyttämään yhtenäistä kommentointikäytäntöä.

7 Sovelluksen testausperiaatteet

Sovelluksen testausta varten laaditaan testaussuunnitelma, kun toteutusvaihe on aloitettu. Varsinainen testaus aloitetaan tehdyn suunnitelman mukaisesti moduulitestauksella, jonka jälkeen tehdään integrointi- ja järjestelmätestaus. Luvussa kuvataan näitä testausmuotoja.

7.1 Moduulitestaus

Sovellus toteutetaan pienempinä osakokonaisuuksina, joita kutsutaan moduuleiksi. Moduuleja testataan ohjelmointivaiheessa kutakin erikseen koko prosessin ajan, jolloin sovellusta on helpompi hallita sekä virheitä helpompi löytää ja korjata. Projektiryhmä suorittaa moduulitestauksen.

7.2 Integrointitestaus

Integrointitestaus aloitetaan rinnakkain moduulitestauksen kanssa. Kun kaksi tai useampia moduuleja on saatu ohjelmoitua, niitä aletaan linkittää toisiinsa. Linkitysvaiheen integrointitestauksessa pyritään etsimään mahdolliset virheet osien välisessä viestien välityksessä. Lisäksi testataan, tekeekö moduulijoukko tarkoitettuja toimintoja oikein. Projektiryhmä suorittaa integrointitestauksen.

7.3 Järjestelmätestaus

Integrointitestauksen valmistuttua järjestelmä testataan kokonaisuudessaan käyttäjän näkökulmasta. Järjestelmätestauksen avulla pystytään tarkistamaan, että järjestelmä toimii määrittelynsä mukaisesti. Järjestelmätestausta varten laaditaan erillinen testaussuunnitelma. Testaus suoritetaan Vaatimusmäärittelyssä mainituilla selaimilla ja niiden versioilla, joita ovat seuraavat:

- Mozilla 1.x,
- Internet Explorer 5.x,
- Opera 6.x,
- Lynx 2.8.4,
- Netscape Communicator 4.x sekä
- Konqueror 3.x.

Järjestelmätestaus suoritetaan kehittämisympäristöstä erillisessä testausympäristössä. Testauksen suorittaa projektiryhmä, ainakin osa tilaajan edustajista ja ohjaajat sekä mahdollisuuksien mukaan lisäksi joku ulkopuolinen.

8 Yhteenveto

PORTTI-projekti suunnittelee ja toteuttaa Jyväskylän yliopiston hallintoviraston viestinnälle WWW-pohjaisen julkaisemislomakkeiston. Tehtävä käsittää lomakkeiston muokkaamisen sekä sen siirtämisen uuteen kantaan.

Tämä suunnitelma kuvaa sovelluksen sisäistä rakennetta ja käyttöliittymää sekä käsittelee testaustapoja ja tietoturvaan liittyviä asioita. Tarvittaessa sovellukselle asetettujen vaatimusten toteutusta priorisoidaan luvussa 2 esitetyllä tavalla.

Lähteet

Erkkilä Juha, Hytönen Jenni, Kivelä Marko, Mali Paula ja Väänänen Lari, ”PORTTI-projekti, projektisuunnitelma”, saatavana pdf-muodossa osoitteessa http://kotka.it.jyu.fi/portti/suunnitelmat/ps26_3.pdf, Jyväskylän yliopisto, tietotekniikan laitos, 26.3.2003.

Erkkilä Juha, Hytönen Jenni, Kivelä Marko, Mali Paula ja Väänänen Lari, ”PORTTI-projekti, vaatimusmäärittely”, saatavana pdf-muodossa osoitteessa http://kotka.it.jyu.fi/portti/vaatimusmaarittelyt/vm19_3.pdf, Jyväskylän yliopisto, tietotekniikan laitos, 19.3.2003.

Heinonen Petri, ”Tekstinkäsittely”, saatavana HTML-muodossa osoitteessa <http://appro.mit.jyu.fi/doc/tekstinkasittely/>, Jyväskylän yliopisto, tietotekniikan laitos, 5.2.2002.

Heinonen Petri ja Poimala Sami, pro gradu -tutkielman Word-pohja, saatavana HTML-muodossa osoitteessa <http://www.mit.jyu.fi/progradut/tyylipohjat/gradupohja.dot>, Jyväskylän yliopisto, tietotekniikan laitos, 15.1.2003.