Testaussuunnitelma

Oppimistavoitteiden hallintajärjestelmä harri

Helsinki 15.11.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (9 op)

Projektiryhmä

Petri Kinnunen
Lasse Leino
Anne Pääkkö
Minna Ulmala

Asiakas

Harri Laine

Johtoryhmä

Kimmo Simola, vastuuhenkilö
Aleksi Yrttiaho, ohjaaja

Kotisivu

`http://www.cs.helsinki.fi/group/harri`

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
1.1	15.11.2007	Aikataulua muutettu
1.0	7.11.2007	Dokumentti oikoluettu ja käyttötapauksia viimeistelty
0.4	31.10.2007	Tarkastilaisuuden ohjeet ja tarkastuslista
0.3	26.10.2007	Lisää tarkennuksia ja käyttötapaukset lisätty
0.2	20.10.2007	Tarkennettu versio
0.1	7.10.2007	Ensimmäinen LaTeX-versio

Sisältö

1 Johdanto

Opppimistavoitteiden hallintajärjestelmä toteutetaan Helsingin yliopiston tietojenkäsittelytieteen laitokselle kursseille kirjattujen eritasoisten oppimistavoitteiden organisointiin ja hallintaan. Järjestelmä tarjoaa opettajille välineen kirjata kurssien tietoja ja opiskelijoille selkeän näkymän oppimistavoitteista ja esitietovaatimuksista. Teemoja voidaan määritellä kurssille hierarkisesti ja oppimistavoitteisiin voidaan liittää esimerkkejä ja tenttikysymyksiä. Lisäksi järjestelmä tukee esitietovaatimusten ja oppimistavoitteiden vastaavuden hallintaa.

Tässä dokumentissa määritellään toteutettavan järjestelmän testaussuunnitelma: mikä on projektin testausaikataulu, mitä testataan ja miten testaus tehdään. Lisäksi se määrittelee milloin järjestelmää on testattu tarpeeksi.

1.1 Tavoitteet

Testauksen tavoitteena on tarkistaa, että järjestelmästä tulee laadukas ja asiakkaan toivoma ohjelmisto. Testauksella pyritään estämään virheiden synty mahdollisimman aikaisessa vaiheessa sekä havaitsemaan mahdollisia puutteita. Testausuunnitelman tavoitteena on asettaa projektille testauksen suuntaviivat, joiden avulla testausta tehdään koko kehitystyön aikana. Tällöin esimerkiksi ohjelmointikin tehdään testausorientoituneesti.

Testauksen ensisijaiset kohteet ovat toiminnot ja vaatimukset, jotka on määritelty vaatimusdokumentissa prioriteetilla 1 ja on näin valittu toteutettavaksi. Nämä toiminnot täytyy siis olla toteutettu ja testattu, jotta tuote voidaan hyväksyä.

1.2 Sanasto

EUCT eli Extended Use Case Testing
Extended Use Case Testing on järjestelmätestauksen suunnittelumalli, jossa vaatimusdokumentissa kuvatuista käyttötapauksista tehdään ns. laajennettuja käyttötapauksia.

JCoverage
JCoverage on lausekattavuuden laskemiseen tarkoitettu työkalu

JUnit
JUnit on Java-ohjelmointikielelle tarkoitettu testaustyökalu.

Mustalaatikkotestaus eli Black box testing
Mustalaatikkotestauksessa komponenttia testataan tietämättä sen toteutuksen yksityiskohtia. Testaus perustuu syöte- ja tulostetietojen tarkasteluun.

Top-down -menetelmä
Top down -menetelmässä komponentit ja osajärjestelmät integroidaan ylhäältä alas eli saadaan nopeasti näkyviin järjestelmän runko ja käyttöliittymä, vaikka mitään todellista toiminnallisuutta näiden alla ei ole.

Tynkä

Tynkä on testauksessa käytettävä komponentti, joka kuvaa pelkistetysti alemman komponentin toimintaa.

2 Testauksen organisointi

2.1 Testauksen aikataulu ja tarkistuspisteet

Testausprosessissa seurataan testauksen V-mallia, jossa on kolme vaihetta:

- yksikkötestaus, joka on käsitelty kappaleessa 3.
- integrointitestaus, joka on käsitelty kappaleessa 4.
- järjestelmätestaus, joka on käsitelty kappaleessa 5.

Testausprosessi ei ole erillinen vaihe ohjelmiston elinkaaressa vaan sitä tehdään koko toteutusvaiheen ajan. Ryhmän jokainen jäsen on vastuussa järjestelmän testauksesta.

Yksikkötestausta tehdään heti kun on jotain testattavaa. Käytännössä tämä tarkoittaa jokaisen luokan testaamista heti sen toteuttamisen jälkeen. Suunnitteludokumentin tarkastustilaisuus pidetään 5.11.2007, jonka jälkeen dokumenttiin tehdään tarvittavat muutokset ja korjaukset. Koodikatselmointi pidetään integrointitestauksen aloituksen jälkeen mutta ennen järjestelmätestausta. Jos projekti on pysynyt aikataulussaan, hyväksymistestaus pidetään 3.12.2007.

Aikataulu:

Tapahtuma	Päivämäärä
Toteutusvaihe alkaa	29.10.2007
Suunnitteludokumentin tarkastus	5.11.2007
Integrointitestaus alkaa	19.11.2007
Koodikatselmointi	26.11.2007
Järjestelmätestaus alkaa	26.11.2007
Hyväksymistestaus	3.12.2007

Katselmoinnit ja tarkastustilaisuudet ovat muodollisia tilaisuuksia, jossa ryhmä yhdessä tarkastaa tehdyn tuotoksen. Niiden tarkoitus on parantaa tuotoksen laatua löytämällä virheet ja puutteet mahdollisimman aikaisessa vaiheessa. Jokainen osallistuja on tutustunut tarkastettavaan tuotokseen etukäteen ja tilaisuudessa dokumentoidaan siitä löydetyt virheet. Apuna käytetään katselmointeja varten tehtyjä tarkistuslistoja. Tilaisuuden puheenjohtajana toimii kyseisen vastuualueen vastaava. Suunnitteluvastaava toimii siis puheenjohtana suunnitteludokumentin tarkastustilaisuudessa ja koodivastaava koodikatselmoinnissa.

2.2 Testitapahtumien ja virheiden raportointi

Testaustapahtumista ja -tuloksista pidetään kirjaa erilliseen testausraporttiin. Raporttiin määritellään, mitä testattiin, millä testitapauksilla ja mitä odotettiin saavan tulokseksi ja mitä lopulta saatiin. Raportista pitää myös ilmetä kuka on testin tekijä ja päivämäärä.

Virhetilanteessa testaaja pyrkii ensin itse katsomaan, mistä virhe on aiheutunut ja mahdollisest korjaamaan sen. Tämän jälkeen testitapaus toistetaan. Tarvittaessa testaaja voi pyytää muilta ryhmän jäseniltä apua ja erityisesti konsultoida testattavan osan tekijää. Virhetilanteet ja uusittu testitapahtuma dokumentoidaan raporttiin. Raporttiin kirjatut virheet, joita ei saada korjatuksi, dokumentoidaan lopulta ylläpitodokumenttiin.

Testitapahtuman raportointiesimerkki:

Testauksen kohde:	Mitä järjestelmän osaa testattiin.
Testisyöte:	Millä testisyötteellä tai -tapauksella kohdetta testattiin.
Odotetut tulokset:	Mitkä ovat odotetut tulokset järjestelmän oikeasta toiminnasta.
Saadut tulokset:	Mitä todella saatiin tulokseksi ja vastaavatko ne odotettuja tuloksia.
Virhetilanteet:	Tapahtuiko virheitä; jos tapahtui niin mitä.
Tekijä ja päivämäärä:	Testaajan nimi ja testauksen päivämäärä.

Järjestelmätestauksessa riittää kirjata onnistuiko testitapaus. Jos testi ei mennyt läpi, kirjataan vielä epäonnistumisen syy. Nämä tiedot kerätään taulukkoon:

Testitapaus	Hyväksytty	Epäonnistumisen syy	Testaaja	Päivämäärä
1	Kyllä/Ei	Miksi testitapaus ei mennyt läpi?	Testaajan nimi	Milloin testattiin

2.3 Testausympäristö- ja työkalut

Jokainen ryhmän jäsen pystyttää testausta varten oman testausympäristön Tietojenkäsittelytieteen laitoksen tietokantapalvelimelle, jossa yksikkö- ja integrointitestaus suoritetaan. Testaustyökaluina käytetään JUnitia ja JCoveragea. Tietokannan toimintaa simuloidaan sopivalla testimateriaalilla. Järjestelmätestauksessa käytetään vain yhtä yhteistä testausympäristöä.

Koska järjestelmä on web-sovellus opiskelijoiden käyttöön, käyttöliittymää testataan ainakin kahdella selaimella, esimerkiksi Internet Explorerilla ja Firefoxilla.

3 Yksikkötestaus

Yksikkötestauksessa testataan järjestelmän pienimpiä loogisia yksiköitä. Tämä tarkoittaa kaikkien luokkien sekä tietokannan testausta. Testauksella pyritään samaan aikaan mahdollisimman virheetöntä koodia. Jokainen ryhmän jäsen on vastuussa oman koodinsa testauksesta, raportoinnista ja virheiden korjaamisesta.

3.1 Lähestymistapa ja toiminnallisuuden testaustapa

Yksikkötestauksen lähestymistapana käytetään mustalaatikkotestausta (black box -testing), jossa toteutus on testaajalta piilotettu. Tällöin keskitytään toiminnallisuuteen testaamiseen syötteiden ja tulosteiden avulla.

Jokainen julkinen ei-triviaali metodi testataan erilaisilla syötteillä ja tarkistetaan, että se toimii halutulla tavalla. Ei-triviaalilla käsitetään metodeita, jotka tekevät jotain muutakin kuin palauttavat tai asettavat arvon (ns. getterit ja setterit). Yksityisiä metodeja ei tarvitse erikseen testata, vaan niiden toimivuus tarkistuu julkisten metodien testauksen ohella. Metodien erilaiset parametriyhdistelmät täytyy tarkistaa kuten myös metodit luokan eri tiloissa, jos tällä on merkitystä.

Tietokannan testaus tarkoittaa käytettyjen SQL-lauseiden testaamista. Tällä tarkistetaan, että muokkauslauseet toimivat tietokannassa oikein ja kyselyt palauttavat oikeat tiedot. Jokainen käytettävä SQL-kysely täytyy testata ainakin kerran.

Lähtökohtaisesti toiminnalliset testit tehdään vasta, kun testattava yksikkö on toteutettu. Apuna käytetään JUnitia.

3.2 Testisyötteiden valinta

Mustalaattikkotestauksen syötteet jaetaan pienempiin osajoukkoihin, jotka ovat arvojoukoltaan samantapaisia ja niitä voidaan käsitellä samalla tavalla. Näitä osajoukkoja kutsutaan ekvivalenssiluokiksi ja ne ovat:

- **Epäkelvolliset syötteet,** joiden käyttö johtaa virhetilanteisiin. Tällöin heitetään poikkeus. Epäkelvollisia syötteitä ovat esimerkiksi vääräntyyppiset parametrit.
- **Liian pienet ja suuret syötteet,** jotka ovat hyväksyttyjen syötteiden arvoalueiden ulkopuolella, mutta ovat esimerkiksi tyypiltään oikein.
- **Tyypilliset ja kelvolliset syötteet,** jotka ovat metodien hyväksyttyjä syötteitä. Tällöin metodit toimivat oikein.

3.3 Hyväksymiskriteerit

Yksikkötestauksessa pyritään mahdollisimman hyvään lausekattavuuteen. Lausekattavuus ilmaisee kuinka suurta osaa suoritetut testit ovat testanneet testattavan yksikön rakenteesta. Tämä saadaan laskukaavasta:

```
Lausekattavuus = testatus lauseet / kaikki lauseet * 100%
```

Sen laskemiseen käytetään JCoverage-testaustyökalua. Jotta yksikkötestaus saadaan suoritettua hyväksytysti, lausekattavuuden on oltava vähintään 80% ja kaikki luokan eitriviaalit metodit on testattu. Testauksessa on myös huomioitu poikkeustilanteet ja olion eri tilat. Kaikki SQL-kyselyt on testattu ja todettu toimiviksi.

4 Integrointitestaus

Integrointitestauksessa testataan järjestelmän eri osien yhteistyötä. Kun luokka on läpäissyt yksikkötestauksen, tiedetään, että se toimii yksinään oikein riittävän hyvin. Tämä ei kuitenkaan takaa, että luokka osaa kommunikoida oikein muiden luokkien kanssa ja käyttää niiden rajapintoja ilman virhetilanteita. Integrointitestauksessa testataan siis erityisesti rajapintoja ja niiden toimintaa.

4.1 Lähestymistapa ja rajapintojen testaustapa

Koska käyttöliittymällä on järjestelmässä tärkeä rooli, integrointitestaus tulla tekemään ylhäältä alas eli Top-down -menetelmällä. Yksikkötestatut osat integroidaan yksi kerrallaan suuremmiksi kokonaisuuksiksi. Tämä aloitetaan integroimalla käyttöliittymä toimintalogiikkaan. Koska tässä vaiheessa järjestelmällä ei ole vielä oikeaa toiminnallisuutta, testauksen suorittaminen vaatii tynkiä, jotka muodostavat vähimmäismäärän toiminnallisuutta. Tämän jälkeen integroidaan tietokanta toimintalogiikkaan. Jokaisessa vaiheessa toiminnallisuutta testataan rajapintojen kautta. Tämä tarkoittaa sitä, että ensin selvitetään integroitavien yksikköjen palveluista ne, joita yksiköt tarvitsevat toisiltaan.

4.2 Hyväksymiskriteerit

Integrointitestaus voidaan katsoa hyväksytyksi, kun kaikkien yksiköiden välinen yhteistyö on tarkistettu rajapintojen kautta ja myös poikkeustapaukset on testattu.

5 Järjestelmätestaus

Integrointitestauksen jälkeen aloitetaan järjestelmätestaus, jossa testataan järjestelmää kokonaisuutena. Tällöin tarkistetaan, että järjestelmä toimii asiakkaan haluamalla tavalla eli

vastaa vaatimusmäärittelyä. Järjestelmätestauksessa testataan myös ei-toiminnallisia vaatimuksia.

Järjestelmätestauksen kaikki testit käydään läpi samalla kertaa yhdessä testi-istunnossa ja tähän osallistuu koko ryhmä.

5.1 Lähestymistapa

Järjestelmätestauksessa ei enää huomioida järjestelmän toteutustapaa, vaan järjestelmää testataan käyttöliittymän kautta. Jokainen virhe kirjataan ylös testausraporttiin.

5.2 Testitapaukset

Testitapaukset saadaan vaatimusdokumenttiin kirjatuista toiminnosta ja käyttötapauksista. Käyttötapaukset muutetaan testausta varten laajennetuiksi käyttötapauksiksi (Extended Use Case Test), joita on helpompi testata. Laajennetuista käyttötapauksissa määritellään erilaisia skenaarioita ja niiden toiminta- ja tulostiedot. Testitapauksissa käyttötapaus viittaa vaatimusdokumentissa nimettyihin käyttötapauksiin. Skenaario a) määrittelee käyttäjän tavallisen työnkulun ja seuraavat skenaariot määrittelevät, miten ne poikkeavat tästä.

5.2.1 Laajennetut käyttötapaukset

1. **Käyttötapaus:** KT1
   Kuvaus: Julkaistujen kurssitietojen (oppimistavoitteet, teemat, esitietovaatimukset) tutkiminen.
   Skenaariot:

   (a) Toiminta: Käyttäjä valitsee kurssilistasta halutun kurssin.
   Tulos: Järjestelmä näyttää kurssin päätiedot (nimen ja kurssikoodin) sekä oppimistavoitteet.

   (b) Toiminta: Käyttäjä valitsee kurssilistasta halutun kurssin sekä kurssinäkymästä esitietovaatimukset-linkin.
   Tulos: Järjestelmä näyttää kurssin päätiedot (nimen ja kurssikoodin) sekä esitietovaatimukset.

2. **Käyttötapaus:** KT2
   Kuvaus: Julkaistujen kurssitietojen (tavoitteet, teemat, vaatimukset) hakeminen kurssin nimellä tai sen osalla.
   Skenaariot:

   (a) Toiminta: Käyttäjä kirjoittaa osan kurssin nimestä.
   Tulos: Järjestelmä näyttää kurssilistauksessa kaikki hakuehtoja vastaavat kurssit.

(b) Toiminta: Käyttäjä kirjoittaa kurssin koko nimen.
Tulos: Järjestelmä näyttää kurssilistauksessa vain etsityn kurssin.

3. **Käyttötapaus:** KT3, KT4
Kuvaus: Käyttäjä tutkii esitietovaatimuksia ja mihin kursseihin ne on liitetty.
Skenaariot:

(a) Toiminta: Käyttäjä valitsee kurssin, johon esitietovaatimus on kytketty.
Tulos: Järjestelmä näyttää valitun kurssin päätiedot (nimen ja kurssikoodin) sekä oppimistavoitteet.

4. **Käyttötapaus:** KT41
Kuvaus: Kirjautuminen.
Skenaariot:

(a) Toiminta: Käyttäjä syöttää käyttäjätunnuksen ja salasanan oikein.
Tulos: Kirjautuminen onnistuu ja järjestelmä näyttää käyttäjälle aloitusnäkymä.

(b) Toiminta: Käyttäjä syöttää käyttäjätunnuksen tai salasanan väärin.
Tulos: Ilmoitetaan käyttäjälle virheestä.

5. **Käyttötapaus:** KT5
Kuvaus: Pääteeman lisääminen.
Skenaariot:

(a) Toiminta: Käyttäjä syöttää pääteeman nimen/kuvauksen molemmilla kielillä ja tallentaa sen.
Tulos: Pääteema tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(b) Toiminta: Käyttäjä syöttää pääteeman nimen/kuvauksen vain suomeksi.
Tulos: Pääteema tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(c) Toiminta: Käyttäjä syöttää pääteeman nimen/kuvauksen vain englanniksi.
Tulos: Pääteema tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(d) Toiminta: Käyttäjä syöttää tyhjän pääteeman nimen/kuvauksen sekä suomeksi että englanniksi.
Tulos: Pääteemaa ei tallenneta. Järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

6. **Käyttötapaus:** KT6
Kuvaus: Alateeman lisääminen.
Skenaariot:

(a) Toiminta: Käyttäjä lisää uuden alateeman oikean pääteeman alle, syöttää alateeman nimen/kuvauksen molemmilla kielillä ja tallentaa sen.

Tulos: Alateema tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(b) Toiminta: Käyttäjä syöttää alateeman nimen/kuvauksen vain suomeksi.
Tulos: Alateema tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(c) Toiminta: Käyttäjä syöttää alateeman nimen/kuvauksen vain englanniksi.
Tulos: Alateema tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(d) Toiminta: Käyttäjä syöttää tyhjän alateeman nimen/kuvauksen sekä suomeksi että englanniksi.
Tulos: Alateemaa ei tallenneta. Järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

7. **Käyttötapaus:** KT7
Kuvaus: Oppimistavoitteen lisääminen.
Skenaariot:

(a) Toiminta: Käyttäjä lisää uuden oppimistavoitteen oikean teeman alle, syöttää oppimistavoitetekstin molemmilla kielillä, valitsee halutun syvyyden ja tallentaa uuden oppimistavoitteen.
Tulos: Oppimistavoite tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(b) Toiminta: Käyttäjä syöttää oppimistavoitetekstin vain suomeksi.
Tulos: Oppimistavoite tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(c) Toiminta: Käyttäjä syöttää oppimistavoitetekstin vain englanniksi.
Tulos: Oppimistavoite tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(d) Toiminta: Käyttäjä syöttää tyhjän oppimistavoitetekstin sekä suomeksi että englanniksi.
Tulos: Oppimistavoitetta ei tallenneta. Järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(e) Toiminta: Käyttäjä ei valitse syvyyttä.
Tulos: Oppimistavoite tallennetaan syvyyden oletusarvolla ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

8. **Käyttötapaus:** KT8
Kuvaus: Teeman muokkaaminen
Skenaariot:

(a) Toiminta: Käyttäjä muokkaa teeman sekä suomenkielistä että englanninkielistä nimeä/kuvausta ja tallentaa muutokset.
Tulos: Muokattu teema tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(b) Toiminta: Käyttäjä muokkaa vain suomenkielistä teeman nimeä/kuvausta.
Tulos: Muokattu teema tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(c) Toiminta: Käyttäjä muokkaa vain suomenkielistä teeman nimeä/kuvausta.
Tulos: Muokattu teema tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(d) Toiminta: Käyttäjä ei muuta mitään mutta tallentaa teeman.
Tulos: Järjestelmä siirtyy kurssi-näkymään.

9. **Käyttötapaus:** KT9
Kuvaus: Oppimistavoitteen muokkaaminen
Skenaariot:

(a) Toiminta: Käyttäjä muokkaa oppimistavoitteen tekstiä molemmilla kielillä sekä vaihtaa syvyyden.
Tulos: Muokattu oppimistavoite tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(b) Toiminta: Käyttäjä muokkaa ainoastaan oppimistavoitteen suomenkielistä tekstiä.
Tulos: Muokattu oppimistavoite tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(c) Toiminta: Käyttäjä muokkaa ainoastaan oppimistavoitteen englanninkielistä tekstiä.
Tulos: Muokattu oppimistavoite tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(d) Toiminta: Käyttäjä muuttaa vain oppimistavoitteen syvyyttä.
Tulos: Muokattu oppimistavoite tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(e) Toiminta: Käyttäjä ei muuta mitään mutta tallentaa oppimistavoitteen.
Tulos: Järjestelmä siirtyy kurssi-näkymään.

10. **Käyttötapaus:** KT10
Kuvaus: Teeman poisto.
Skenaariot:

(a) Toiminta: Käyttäjä poistaa teeman, johon ei liitty oppimistavoitteita tai alateemoja.
Tulos: Teema poistetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

(b) Toiminta: Käyttäjä poistaa teeman, johon liittyy oppimistavoitteita ja/tai alateemoja.
Tulos: Järjestelmää ilmoittaa, että samalla poistetaan myös mahdolliset oppimistavoitteet ja alateemat ja pyytää vahvistamaan poiston.

11. **Käyttötapaus:** KT11
Kuvaus: Oppimistavoitteen poisto.
Skenaariot:

    (a) Toiminta: Käyttäjä poistaa oppimistavoitteen, johon ei liitty esimerkkejä tai tenttikysymyksiä.
    Tulos: Oppimistavoite poistetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään teemat ja oppimistavoitteet.

12. **Käyttötapaus:** KT22, KT23, KT24
Kuvaus: Esitietovaatimuksen lisääminen.
Skenaariot:

    (a) Toiminta: Käyttäjä syöttää esitietovaatimustekstin molemmilla kielillä, valitsee syvyyden, liittää esitietovaatimuksen haluttuun kurssiin sekä teemaan ja tallentaa uuden esitietovaatimuksen.
    Tulos: Esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

    (b) Toiminta: Käyttäjä ei liitä esitietovaatimusta mihinkään.
    Tulos: Esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

    (c) Toiminta: Käyttäjä liittää esitietovaatimuksen useampaan kuin yhteen kurssiin/teemaan.
    Tulos: Esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

    (d) Toiminta: Käyttäjä syöttää esitietovaatimustekstin vain suomeksi.
    Tulos: Esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

    (e) Toiminta: Käyttäjä syöttää esitietovaatimustekstin vain englanniksi.
    Tulos: Esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

    (f) Toiminta: Käyttäjä syöttää tyhjän tekstin molemmilla kielillä.
    Tulos: Esitietovaatimus ei tallenneta. Järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

    (g) Toiminta: Käyttäjä ei valitse esitietovaatimuksen syvyyttä.
    Tulos: Esitietovaatimus tallennetaan syvyyden oletusarvolla ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

13. **Käyttötapaus:** KT26
Kuvaus: Esitietovaatimuksen muokkaaminen
Skenaariot:

    (a) Toiminta: Käyttäjä muokkaa esitietovaatimuksen tekstiä molemmilla kielillä, vaihtaa syvyyden sekä muokkaa kursseja/teemoja, johon esitietovaatimus on liitetty.

Tulos: Muokattu esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

(b) Toiminta: Käyttäjä muokkaa ainoastaan esitietovaatimuksen tekstiä.
Tulos: Muokattu esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

(c) Toiminta: Käyttäjä muuttaa vain esitietovaatimuksen syvyyttä.
Tulos: Muokattu esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

(d) Toiminta: Käyttäjä poistaa kurssin/teeman, johon esitietovaatimus on liitetty.
Tulos: Muokattu esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

(e) Toiminta: Käyttäjä lisää uuden kurssin/teeman, johon esitietovaatimus liitetään.
Tulos: Muokattu esitietovaatimus tallennetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

(f) Toiminta: Käyttäjä ei muuta mitään mutta tallentaa esitietovaatimuksen.
Tulos: Järjestelmä siirtyy kurssi-näkymään.

14. **Käyttötapaus:** KT27
Kuvaus: Esitietovaatimuksen poisto.
Skenaariot:

(a) Toiminta: Käyttäjä poistaa valitun esitietovaatimuksen
Tulos: Esitietovaatimus poistetaan ja järjestelmä siirtyy kurssi-näkymään, jossa näytetään esitietovaatimukset.

15. **Käyttötapaus:** KT28, KT29
Kuvaus: Muokkaajan luominen.
Skenaariot:

(a) Toiminta: Käyttäjä merkitsee henkilön muokkaajaksi ja tallentaa valinnan.
Tulos: Henkilö tallennetaan muokkaajaksi ja järjestelmä siirtyy kurssi-näkymää.

16. **Käyttötapaus:** KT30, KT29
Kuvaus: Muokkaajan poistaminen.
Skenaariot:

(a) Toiminta: Käyttäjä poistaa muokkaaja-merkinnän ja tallentaa.
Tulos: Henkilöltä poistetaan muokkaaja-asema ja järjestelmä siirtyy kurssi-näkymään.

17. **Käyttötapaus:** KT31
Kuvaus: Vastuuhenkilön liittäminen kurssiin.
Skenaariot:

   (a) Toiminta: Käyttäjä merkitsee henkilön vastuuhenkilöksi ja tallentaa valinnan.
   Tulos: Henkilö tallennetaan vastuuhenkilöksi ja järjestelmä siirtyy kurssi-näkymään.

   (b) Toiminta: Käyttäjä yrittää valita kurssille toisen vastuuhenkilön.
   Tulos: Järjestelmä sallii kurssille vain yhden vastuuhenkilön.

18. **Käyttötapaus:** KT32
   Kuvaus: Vastuuhenkilön poistaminen kurssista.
   Skenaariot:

   (a) Toiminta: Käyttäjä poistaa vastuuhenkilö-merkinnän ja tallentaa.
   Tulos: Henkilöltä poistetaan vastuuhenkilö-asema ja järjestelmä siirtyy kurssi-näkymään.

19. **Käyttötapaus:** KT33
   Kuvaus: Tarkastajan luominen.
   Skenaariot:

   (a) Toiminta: Käyttäjä merkitsee henkilön tarkastajaksi ja tallentaa valinnan.
   Tulos: Henkilö tallennetaan tarkastajaksi ja järjestelmä siirtyy kurssi-näkymään.

20. **Käyttötapaus:** KT34
   Kuvaus: Tarkastajan poistaminen.
   Skenaariot:

   (a) Toiminta: Käyttäjä poistaa tarkastaja-merkinnän ja tallentaa.
   Tulos: Henkilöltä poistetaan tarkastaja-asema ja järjestelmä siirtyy kurssi-näkymään.

21. **Käyttötapaus:** KT35
   Kuvaus: Kurssin tilan muuttaminen muokattavaksi.
   Skenaariot:

   (a) Toiminta: Kurssin tilaksi valitaan muokattava ja tallennetaan valinta.
   Tulos: Kurssi tila on nyt muokattava ja järjestelmä siirtyy kurssi-näkymään.

22. **Käyttötapaus:** KT36
   Kuvaus: Kurssin tilan muuttaminen muokattavasta suojatuksi.
   Skenaariot:

   (a) Toiminta: Kurssin tilaksi valitaan suojattu ja tallennetaan valinta.
   Tulos: Kurssi tila on nyt suojattu ja järjestelmä siirtyy kurssi-näkymään.

23. **Käyttötapaus:** KT37
   Kuvaus: Kurssin tilan muuttaminen tarkastettavaksi.
   Skenaariot:

   (a) Toiminta: Kurssin tilaksi valitaan tarkastettava ja tallennetaan valinta.
   Tulos: Kurssi tila on nyt tarkastettava eikä sitä voi muokata. Järjestelmä siirtyy kurssi-näkymään

24. **Käyttötapaus:** KT38
    Kuvaus: Kurssin julkaisu.
    Skenaariot:

    (a) Toiminta: Kurssin tilaksi valitaan julkinen ja tallennetaan valinta.
    Tulos: Kurssin tiedot näkyvät internetissä kaikille ja lisäksi kurssista muodostuu työversio suojattuun tilaan. Järjestelmä siirtyy kurssi-näkymään.

25. **Käyttötapaus:** KT39
    Kuvaus: Kurssin palauttaminen tarkastettavasta suojatuksi.
    Skenaariot:

    (a) Toiminta: Kurssin tilaksi valitaan hylätty ja tallennetaan valinta.
    Tulos: Kurssi tila on nyt suojattu ja sitä voidaan muokata. Järjesteläm siirtyy kurssi-näkymään.

5.3 Hyväksymiskriteerit

Järjestelmätestaus on suoritettu hyväksytysti, kun laajennetut käyttötapaukset on testattu ja järjestelmä ei kaadu käytön aikana. Jotta testitapaus on hyväksytty, pitää kaikkien skenaarioiden mennä läpi.

6 Hyväksymistestaus

Kun järjestelmätestaus on tehty, suoritetaan hyväksymistestaus asiakkaan kanssa. Tilaisuutta ei kokonaisuudessaan dokumentoida testausraporttiin, mutta mahdolliset virhetilanteet kirjataan ylläpitodokumenttiin. Hyväksymistestaus pidetään 3.12.2007.

Liite 1. Suunnitteludokumentin tarkastustilaisuus

Osallistujat

Anne Pääkkö, puheenjohtaja
Petri Kinnunen
Minna Ulmala

Aleksi Yrttiaho, ohjaaja, sihteeri

Roolit

Puheenjohtaja huolehtii, että tilaisuus sujuu jouhevasti ja se pysyy aikataulussa.

Alustaja aloittaa tuotoksen kohdan tarkastuksen, ja ryhmä käy alustajan ohjaamana läpi tarkastettavan tuotteen etsien siitä mahdollisia puutteita ja virheitä

Asiantuntija selittää mahdolliset tekniset yksityiskohdat, joista muu tarkastusryhmä ei saa selvää.

Sihteeri dokumentoi löydetyt virheet ja puutteet.

Tarkastaja etsii puutteita ja virheitä.

Osio	Alustaja	Asiantuntija
Johdanto	Minna	Anne
Arkkitehtuurisuunnittelu	Anne	Minna
Käyttöliittymäsuunnittelu	Petri	Anne
Tietokantasuunnittelu	Anne	Petri
Integroinnin suunnittelu	Petri	Minna
Luokkasuunnittelu	Minna	Petri

Lisäksi jokainen ryhmän jäsen toimii tarkastajana.

Ennen tilaisuutta

Suunnitteludokumentti on tarkastettu niin ettei siinä enää ole ilmeisiä virheitä. Jokainen osallistuja on valmistautunut tilaisuuteen tutustumalla huolellisesti suunnitteludokumenttiin sekä tarkastustilaisuutta varten tehtyyn tarkastuslistaan.

Tarkastustilaisuus

Tarkastustilaisuus kestää 90-120 minuuttia ja sen tarkoitus on löytää dokumentin puutteita ja virheitä:

- Puute = vajaa määritys tai puuttuva toiminta
- Virhe = väärin tehty määritys tai ei-toivottu toiminta

Tarkastustilaisuudessa ei ratkota löydettyjä ongelmia tai keskustella niistä vaan ne kirja-

taan ylös. Jokaisesta puutteesta ja virheestä dokumentoidaan sivunumero ja kappale, jossa ko. virhe tai puute esiintyy, lyhyt kuvaus sekä tyyppi:

- V = selvä vakava virhe (asia ilmaistu väärin)
- T = tulkinnanvarainen virhe (epäselvää tekstiä)
- P = puute (jotain puuttuu)

Tarkastustilaisuus etenee jokaisen tarkastettavan osan kohdalla seuraavasti:

1. Alustaja esittelee tarkastettavan kohdan.
2. Puheenjohtaja aloittaa keskustelun.
3. Osallistujat tuovat esille puutteita ja virheitä.
4. Sihteeri kirjaa nämä ylös.
5. Puheenjohtaja ohjaa ryhmän seuraavaan kohtaan.

Tarkastustilaisuuden päätös

Ryhmä äänestää tuotoksen hyväksymisestä; joko dokumentti hyväksytään sellaisenaan tai muutoksin tai sitten se hylätään ja korjausten jälkeen pidetään uusi tarkastustilaisuus.

Liite 2. Suunnitteludokumentin tarkastuslista

Dokumentti

1. Onko dokumentti mahdollisimman lyhyt kattaen kuitenkin kaiken toteutuksen ja ylläpidon kannalta oleellisen materiaalin?
2. Ovatko kaikki tärkeät suunnitteluratkaisut dokumentoitu?
3. Ovatko kaikki osat dokumentoitu selkeästi ja ristiriidattomasti?

Arkkitehtuuri

4. Onko koko arkkitehtuuri kuvattu kuvaustekniikoin (esim. UML:lla)?
5. Voidaanko arkkitehtuurilla toteuttaa kaikki vaatimukset?
6. Voidaanko arkkitehtuurilla toteuttaa kaikki käyttötapaukset?
7. Onko arkkitehtuuri kuvattu riittävän tarkasti?

Osajärjestelmät

8. Toimiiko osajärjestelmien viestinvälitys oikein?
9. Onko virheellisten viestien käsittely kunnossa?

Tietokanta

10. Onko tietokanta normalisoitu?
11. Voidaanko tietokannalla toteuttaa tarvittava tietojen säilytys?

Luokat

12. Onko luokat suunniteltu järkevästi loogisiksi kokonaisuuksiksi?
13. Onko luokkia tarpeeksi / liian paljon?
14. Onko jokaisella luokalla kirjattuna tehtävä ja testaussuunnitelma?
15. Voidaanko luokkien avulla toteuttaa arkkitehtuuri?

Metodit

16. Onko kaikki tarvittavat metodit kuvattu?
17. Ovatko metodien parametrit selkeästi määritelty?
18. Ovatko metodien palautearvot selkeästi määritelty?
19. Voidaanko luokan metodien avulla toteuttaa luokan toiminnallisuus?
20. Onko ylimääräisiä metodeja?