

Testaussuunnitelma

HenTyLi

Helsinki 8.12.2004
Ohjelmistotuotantoprojekti
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

1	Johdanto.....	4
2	Testauksen tavoite.....	4
3	Testausstrategia.....	4
4	Testauksen vaiheet.....	4
4.1	Komponenttitestaus.....	4
4.2	Integroititestaus.....	4
4.3	Järjestelmätestaus.....	5
4.4	Hyväksymistestaus.....	5
5	Testaamatta jäävä toiminnallisuus.....	5
6	Testausympäristö.....	5
7	Komponenttitestaus.....	5
7.1	TietokantaYhteys-luokka.....	5
7.1.1	Testien alkuasetukset.....	6
7.1.2	testAvaaYhteys().....	6
7.1.3	testHaeHenkilo().....	6
7.1.4	testHaeHenkilonKoulutus().....	6
7.1.5	testHaeHenkilonMerkit().....	6
7.1.6	testHaeHenkilonPoissaolot().....	6
7.1.7	testHaeHenkilonTapahtumat().....	6
7.1.8	testHaeHenkilonTutkinnot().....	6
7.1.9	testHaeHenkilonTyosuhteet().....	6
7.1.10	testHaeIJKKTyosuhde().....	7
7.1.11	testHaeKaikkiHenkilot().....	7
7.1.12	testHaeKaikkiKoulutus().....	7
7.1.13	testHaeKaikkiPoissaolot().....	7
7.1.14	testHaeKaikkiTapahtumat().....	7
7.1.15	testHaeKaikkiTyosuhteet().....	7
7.1.16	testSuljeYhteys().....	7
7.2	Validoija-luokka.....	8
7.2.1	Testidata.....	8
7.2.2	testNull().....	8
7.2.3	testToimivaIlmanPoissaoloja().....	8
7.2.4	testToimivaJossaPoissaoloja().....	8
7.2.5	testTyhjä().....	8
7.2.6	testVirheellisetPoissaolot().....	9
7.2.7	testVirheellisetTyosuhteet().....	9
7.2.8	testYksinäinenPoissaolo().....	10
7.2.9	testYksinäinenTyosuhde().....	10
8	Hyväksyntätestauksen testitapauksia.....	10
8.1	T1: Henkilön lisääminen järjestelmään, väärä sotu.....	10
8.2	T2: Henkilön lisääminen järjestelmään.....	10
8.3	T3: Henkilön tietojen tarkastelu ja haku.....	10
8.4	T4: Henkilön tietojen muuttaminen.....	11
8.5	T5: Henkilölistaus.....	11
8.6	T6: Henkilöluettelon tulostaminen.....	11
8.7	T7: Ansiomerkin lisääminen.....	11
8.8	T8: Kunniamerkin myöntäminen.....	11
8.9	T9: Edellisen tyosuhteen lisääminen.....	12
8.10	T10: Nykyisen tyosuhteen lisääminen.....	12

8.11	T11: Nimikirjanotteen muodostaminen.....	13
8.12	T12: Nimikirjanotteen tulostaminen	13
8.13	T13: Vuosilomaluettelon muodostaminen.....	13
8.14	T14: Vuosilomaluettelon tulostaminen	13
8.15	T15: Vuosikorotusilmoituksen muodostaminen	13
8.16	T16: Vuosikorotusilmoituksen tulostaminen.....	14
8.17	T17: Vuosikorotusluettelon muodostaminen.....	14
8.18	T18: Vuosikorotusluettelon tulostaminen.....	14

1 Johdanto

Hentyli-projekti kuuluu Helsingin yliopiston tietojenkäsittelytieteen laitoksen Ohjelmistotuotantoprojekti-kurssiin. Projektissa tuotetaan Invalidiliiton Järvenpään koulutuskeskukselle henkilöstöhallintajärjestelmä. Järjestelmään kirjataan ja ylläpidetään talossa työskentelevien henkilöiden henkilö-, työsuhde- ja koulutustiedot. Näiden käyttäjän syöttämien tietojen perusteella järjestelmä laskee kullekin työntekijälle kuuluvat etuisuudet, kuten palkkalisät ja vuosilomaoikeuden.

2 Testauksen tavoite

Testauksen tavoitteena on varmistua Hentyli-projektin tuottaman ohjelmiston laadusta sekä varmistaa että ohjelmisto täyttää asiakkaan sille asettamat vaatimukset.

3 Testausstrategia

Testaus koostuu neljästä osavaiheesta: komponenttitestauksesta, integraatiotestauksesta, järjestelmätestauksesta ja hyväksymistestauksesta. Komponenttitestauksessa testataan yksittäisten luokkien toimintaa erillään muista luokista. Komponenttitestaus suoritetaan pääasiassa toteutuksen yhteydessä moduulin toteuttajan toimesta. Integraatiotestauksessa testataan yksikkötestattujen luokkien rajapintoja ja yhteistoimintaa. Järjestelmätestauksessa testattavana on koko ohjelmisto ja laitteisto kokonaisuutena. Hyväksymistestauksessa tuotettua ohjelmistoa testataan testiaineistolla kehitysryhmän toimesta.

Kukin suoritettu testitapaus tuloksineen kirjataan testausdokumenttiin. Yksittäisistä testeistä kerrotaan testaajan nimi, suorituspäivämäärä, testattava asia, kuvaus testistä, odotettu tulos sekä mahdolliset havaitut virheet. Testauksessa löydetty virheet pyritään korjaamaan mahdollisuuksien mukaan. Mikäli jokin vika havaitaan vasta testauksen loppuvaiheessa ja sen korjaaminen olisi mahdotonta käytettävissä olevan ajan puitteissa, virhe jätetään korjaamatta ja dokumentoidaan.

4 Testauksen vaiheet

4.1 Komponenttitestaus

Yksikkötestauksen suorittaa ensisijaisesti moduulin ohjelmoinut projektiryhmän jäsen. Moduulit testataan white-box-testauksena. Testaus suoritetaan joko sopivalla testidatalla, ohjelmoimalla testiajureita tai JUnit-työkalun tukemana automatisoituina yksikkötestauksena. Laskentayksikön luokka Validoiija sekä tietokantayhteysluokka TietokantaYhteys testataan JUnitilla.

4.2 Integrointitestaus

Integrointitestauksessa keskitytään komponenttien välisiin rajapintoihin. Tässä vaiheessa testataan erillisten komponenttien välistä vuorovaikutusta sekä kommunikointia ja toiminnallisuutta niiden välillä. Integroitavien yksiköiden yksikkötestaus täytyy olla hyväksyttävästi suoritettuna ennen tämän vaiheen aloittamista. Integrointitestaus suoritetaan kokoavana testauksena (bottom-up) alhaalta ylöspäin, jolloin alemman tason komponentit kehitetään ja testataan ennen siirtymistä korkeamman tason rakenteisiin. Testaus aloitetaan

integroimalla alimman tason yksiköt yksitellen mukaan ohjelmistoon ja testaamalla nämä. Tämän jälkeen integroidaan seuraavalla tasolla olevat yksiköt alimman tason kanssa. Tätä kiertoa jatketaan kunnes kaikkein ylin taso on integroitu ja testattu. Rajapintoja testataan ajurien avulla. Nämä simuloivat korkeamman tason moduuleita, joihin alemman tason moduulit yhdistyvät. Ajurit kutsuvat rajapintafunktioita testitapausten määrittelemien parametrein ja tarkkailevat funktiokutsujen palautusarvoja. Testauksen tässä vaiheessa pyritään tarkestelemaan erityisesti rajapintafunktiolle annettavat syötteet ja niiden palautusarvot, sekä itse palvelua kutsujan kannalta. Raporttiyksikön luokka Raportti testataan tässä vaiheessa käyttöliittymän kautta.

4.3 Järjestelmätestaus

Järjestelmätestauksessa pyritään testaamaan ohjelmistoa sille määriteltyjen vaatimusten suhteen. Testiaineistona käytetään Hentyli-projektista kerättyä aineistoa. Testaus pyritään suorittamaan niin, että projektiryhmän jäsenet testaavat eri alueita kuin ovat itse toteuttaneet. Koska Hentyli-projektin järjestelmä joudutaan siirtämään projektin lopuksi asiakkaan toimesta varsinaiseen asiakasympäristöön, on tätä varten luotava asennusohjelma. Tämän testaus suoritetaan nykyisessä järjestelmässä poistamalla tietokannan taulut ja ohjelmisto, jonka jälkeen luodaan taulut sekä asennetaan ohjelmisto tällä asennusohjelmalla.

4.4 Hyväksymistestaus

Ohjelmiston hyväksymistestaus suoritetaan käyttöttestaamalla määrittelydokumentin toiminnot käyttötapausten avulla. Kaikki nämä toiminnot ja käyttötapaukset testataan, ja niiden käyttäytyminen ohjelmiston toteutusvaiheen päätyttyä kirjataan. Syötteenä käytetään sekä oikeellista (määrittelydokumentissa esitettyä) että virheellistä (muodoltaan tai kattavuudeltaan vajavaista) dataa. Ohjelmiston katsotaan läpäisseen hyväksymistestaus, mikäli nämä toiminnot toimivat kuten määrittelydokumentti ne kuvaa. Lisäksi järjestelmän on pysyttävänä vakaana myös siinä tapauksessa, että käyttäjäsyötteet poikkeava odotetusta.

5 Testaamatta jäävä toiminnallisuus

Koska erityisiä suoritusvaatimuksia ei ole esitetty ja kannan koska koko tulee pysymään pienenä, jätetään suurimittainen kuormitustestaus tekemättä. Järjestelmää tullaan testaamaan siten että vain yksi käyttäjä on kerrallaan käyttämässä ohjelmaa.

6 Testausympäristö

Järjestelmää testataan asiakkaan Windows 2000 -työasemalla, Sun Java 1.5 -ympäristössä ja MS SQL -tietokannalla.

7 Komponenttitestaus

Ohjelman tietokanta- ja laskentayksikön luokat testataan yksitellen, metodi kerrallaan. Komponenttitestaus suoritetaan automatisoitujen JUnit-testiluokkien avulla TietokantaYhteys- ja Validoija-luokille.

7.1 TietokantaYhteys-luokka

TietokantaYhteys-luokka testataan JUnit-testiluokassa TietokantaYhteysTest. Luokan jokainen julkinen metodi testataan omassa testimetodissaan.

7.1.1 Testien alkuasetukset

Testit vaativat toimiakseen HenTyLi-ohjelman tietokannan, johon on tallennettu vähintään seuraavat henkilöt ja tapahtumat. Henkilön 121277-123W tilaksi on asetettu ”töissä”, henkilölle on tallennettu erilaisia tapahtumia, mm. voimassaoleva työsuhde IJKK:ssa. Henkilön 220338-044R tilaksi on asetettu ”työsuhde päättynyt” ja henkilöllä on oltava tallennettuja tapahtumia. Henkilöllä 120859-013U on oltava ansiomerkki. Henkilöllä 230658-126W on oltava tallennettu poissaolo- ja koulutustapahtuma.

7.1.2 testAvaaYhteys()

Testaa metodia avaaYhteys(). Testaa, että metodi palauttaa arvon true, jolloin yhteys on auki metodin kutsun jälkeen, seuraavissa tilanteissa: Yhteys avataan ensimmäisen kerran, avattu yhteys yritetään avata uudestaan, ja suljettu yhteys avataan.

7.1.3 testHaeHenkilo()

Testaa metodia haeHenkilo(String sotu). Testaa yhden henkilön tietojen hakemista tietokannasta. Testaa ensin, että metodi ei toimi suljetulla yhteydellä. Sen jälkeen testaa, että metodi palauttaa oikealla henkilötunnuksella nullista poikkeavan Henkilo-olion, ja väärällä henkilötunnuksella nullin.

7.1.4 testHaeHenkilonKoulutus()

Testaa metodia haeHenkilonKoulutus(String sotu). Testaa ensin henkilön koulutusten hakemista toimivalla henkilötunnuksella, jolloin metodi palauttaa nullista poikkeavan ArrayListin. Sen jälkeen testaa väärällä henkilötunnuksella, jolloin metodi palauttaa nullin.

7.1.5 testHaeHenkilonMerkit()

Testaa metodia haeHenkilonMerkit(String sotu). Testaa ensin henkilön ansio- ja kunniamerkkien hakemista toimivalla henkilötunnuksella, jolloin metodi palauttaa nullista poikkeavan ArrayListin. Sen jälkeen testaa väärällä henkilötunnuksella, jolloin metodi palauttaa nullin.

7.1.6 testHaeHenkilonPoissaolot()

Testaa metodia haeHenkilonPoissaolot(String sotu). Testaa ensin henkilön poissaolojen hakemista toimivalla henkilötunnuksella, jolloin metodi palauttaa nullista poikkeavan ArrayListin. Sen jälkeen testaa väärällä henkilötunnuksella, jolloin metodi palauttaa nullin.

7.1.7 testHaeHenkilonTapahtumat()

Testaa metodia haeHenkilonTapahtumat(String sotu, boolean kaikki). Testaa ensin tapahtumien hakemista toimivalla henkilötunnuksella, parametrilla kaikki = false. Varmistaa, että palautetussa ArrayListissä olevissa tapahtumissa ei ole KL-, LO-, MVK-, AM- tai VL-kentissä arvoa -1. Lopuksi testaa, että virheellisellä sotulla metodi palauttaa nullin.

7.1.8 testHaeHenkilonTutkinnot()

Testaa metodia haeHenkilonTutkinnot(String sotu). Testaa ensin henkilön tutkintojen hakemista toimivalla henkilötunnuksella, jolloin metodi palauttaa nullista poikkeavan ArrayListin. Sen jälkeen testaa väärällä henkilötunnuksella, jolloin metodi palauttaa nullin.

7.1.9 testHaeHenkilonTyosuhteet()

Testaa metodia haeHenkilonTyosuhteet(String sotu, boolean kaikki). Testaa ensin tapahtumien hakemista toimivalla henkilötunnuksella, parametrilla kaikki = false. Varmistaa, että palautetussa

ArrayListissä olevissa tapahtumissa ei ole KL-, LO-, MVK-, AM- tai VL-kentissä arvoa -1. Lopuksi testaa, että virheellisellä sotulla metodi palauttaa nullin.

7.1.10 testHaeIJKKTyosuhte()

Testaa metodia haeIJKKTyosuhte(String sotu). Testaa ensin henkilön IJKK:ssa voimassaolevan työsuhteen hakemista sellaisella henkilötunnuksella, jolla tiedetään olevan tietokantaan tallennettu voimassaoleva työsuhte. Tarkistetaan että metodi palauttaa nullista poikkeavan Tapahtuma-olion. Sen jälkeen testataan metodia henkilötunnuksella, jonka työsuhte on päättynyt, jolloin metodi palauttaa nullin.

7.1.11 testHaeKaikkiHenkilot()

Testaa metodia haeKaikkiHenkilot(boolean opettajat, boolean muut, boolean eläkeläiset, boolean päättäneet, boolean arkistoidut, int ominaisuus). Metodia testataan ensin niin ominaisuus-arvolla -1 ja boolean arvojen eri yhdistelmillä, lopuksi testataan positiivisella ominaisuus-arvolla.

Eri boolean-arvojen yhdistelmillä testattaessa varmistetaan, että saatu Henkilo-olioista koostuva ArrayList sisältää ainoastaan niitä henkilöryhmiä, joita on pyydetty. Jos opettajat = true, ja muut arvot false, varmistetaan että saadussa listassa on ainoastaan Henkilo-olioita joiden tila = ”töissä” ja henkilöstöryhmä = ”OPE”. Samoin toimitaan testattaessa muita yhdistelmiä.

Lopuksi metodi testaa eri henkilöyhdistelmien hakemista ominaisuus-arvolla 2, jolloin palautuu lista henkilöistä joilla on tietokantaan tallennettu tapahtumia vastaavalla arvolla. Testimetodi testaa myös henkilöiden hakemista virheellisellä ominaisuus-arvolla 324832, jolloin metodi palauttaa null-arvoisen henkilölistan.

7.1.12 testHaeKaikkiKoulutus()

Testaa metodia haeKaikkiKoulutus(). Testaa, että metodi palauttaa nullista-poikkeavan ArrayListin koulutustapahtumia.

7.1.13 testHaeKaikkiPoissaolot()

Testaa metodia haeKaikkiPoissaolot(). Testaa, että metodi palauttaa nullista-poikkeavan ArrayListin poissaolotapahtumia.

7.1.14 testHaeKaikkiTapahtumat()

Testaa metodia haeKaikkiTapahtumat(boolean kaikki). Testaa ensin metodia arvolla kaikki = true, ja varmistaa että metodi palauttaa nullista-poikkeavan ArrayListin tapahtumia. Sen jälkeen testaa metodia arvolla false, ja varmistaa että palautettu ArrayList ei sisällä tapahtumia, joiden KL-, LO-, MVK-, AM- tai VL-kentissä on arvo -1.

7.1.15 testHaeKaikkiTyosuhteet()

Testaa metodia haeKaikkiTyosuhteet(boolean kaikki). Testaa ensin metodia arvolla kaikki = true, ja varmistaa että metodi palauttaa nullista-poikkeavan ArrayListin työsuhteita. Sen jälkeen testaa metodia arvolla false, ja varmistaa että palautettu ArrayList ei sisällä tapahtumia, joiden KL-, LO-, MVK-, AM- tai VL-kentissä on arvo -1.

7.1.16 testSuljeYhteys()

Testaa metodia suljeYhteys(). Testaa, että metodi palauttaa arvon true seuraavissa tilanteissa: uusi yhteysolio suljetaan, avattu yhteys suljetaan, ja kertaalleen suljettu yhteys suljetaan.

7.2 Validoija-luokka

Validoija-luokka testataan JUnit-testiluokassa ValidoijaTest. Kaikissa testiluokan metodeissa testataan Validoija-luokan julkista validoiTapahtumaLista(ArrayList tapahtumalista)-metodia.

7.2.1 Testidata

Testeissä käytetään kolmeatoista työsuhdetapahtumaa (T1-T13), yhtätoista poissaolotapahtumaa (P1-P11) ja yhtä koulutustapahtumaa (K). Tapahtumien kestot ja päivämäärät kuvataan taulukossa, joka näkyy kuvassa 1. Kuvassa näkyvät nuolet kuvaavat tapahtuman alku- ja päättymispäivämäärää. Tapahtumat, joiden nuolen kärki puuttuu, ovat voimassaolevia työsuhteita, joilla ei ole päättymispäivämäärää.

7.2.2 testNull()

Metodissa testataan tapahtumalistoja, jotka sisältävät null-arvoja. Metodi testaa, että validoija heittää poikkeuksen kaikille listoille, jotka sisältävät nullin. Metodissa testataan seuraavat tapaukset:

- Lista sisältää pelkän nullin
- listan lopussa on null
- listan alussa on null
- listassa on koulutustapahtumia, ja niiden jälkeen null,

7.2.3 testToimivallmanPoissaoloja()

Metodissa testataan validoijaa tapahtumalistoilta, jotka eivät sisällä poissaoloja. Kaikki testattavat tapahtumalistat ovat valideja, joten testataan että validoija ei heitä poikkeusta. Seuraavat tapaukset testataan:

- peräkkäiset tapahtumat T1, T3 ja T5
- peräkkäiset tapahtumat T1, T4, T3 ja T5
- tapahtumat T4 ja T3, ja niiden seassa koulutustapahtumia

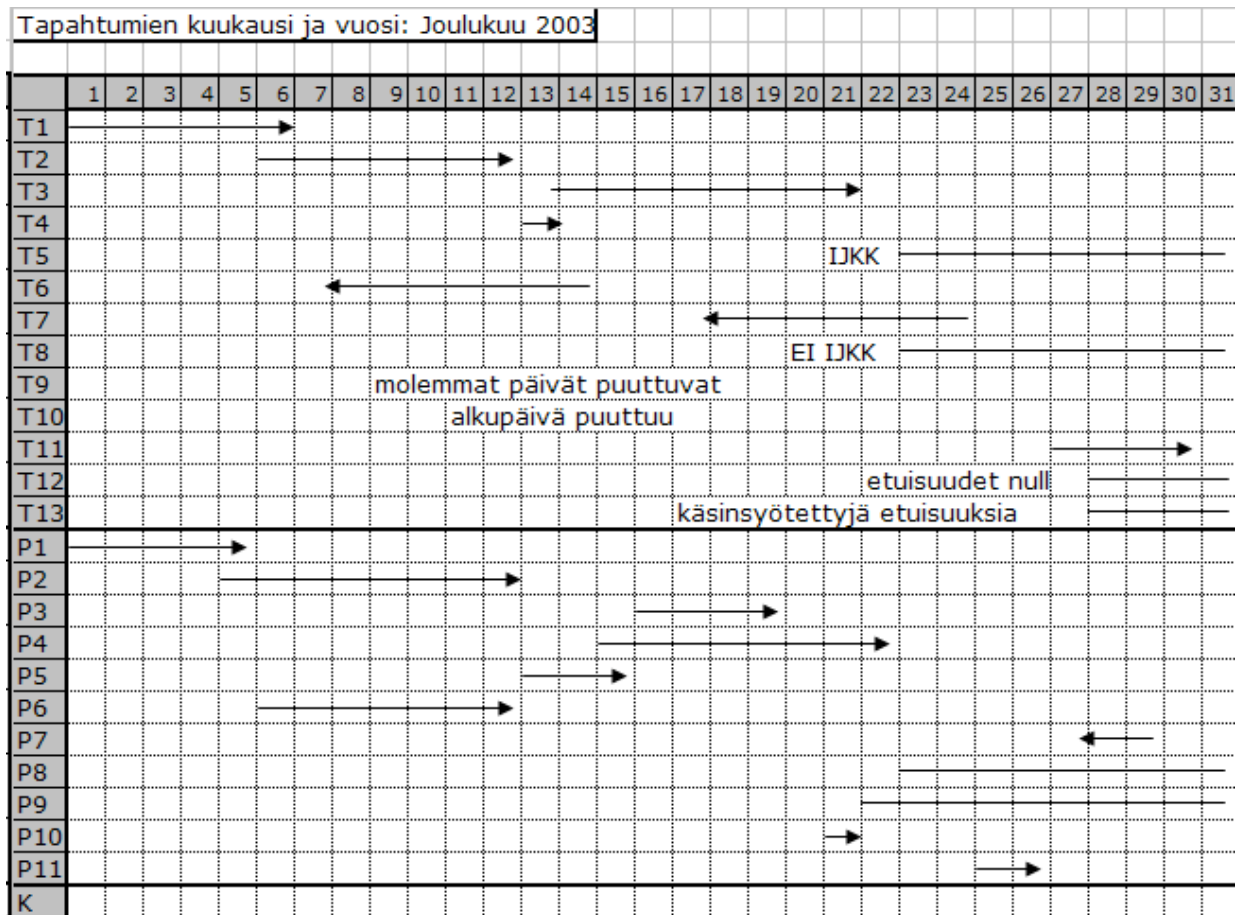
7.2.4 testToimivaJossaPoissaoloja()

Metodissa testataan validoijaa tapahtumalistoilta, jotka työsuhteiden lisäksi sisältävät poissaoloja. Kaikki metodissa testattavat tapahtumalistat ovat valideja, joten testataan että validoija ei heitä poikkeusta yhdessäkään tapauksessa. Metodissa testataan seuraavat tapaukset:

- peräkkäiset tapahtumat T1, P1, T3, P3, T5
- poissaolo voimassaolevassa työsuhteessa: T1, P1, T4, T5, P8. Poissaolon P8 päättymispäivämäärä on tulevaisuudessa.
- poissaolo on saman pituinen kuin työsuhde, seassa koulutustapahtumia: T2, P6, T4, K, T3, K
- saman työsuhteen sisällä kaksi perättäistä poissaoloa. T1, T3, P3, P10

7.2.5 testTyhjä()

Metodissa testataan validoijaa tyhjällä tapahtumalistalla, ja listalla joka sisältää pelkkiä koulutustapahtumia. Testataan, että molemmissa tapauksissa validointimetodi pääsee loppuun asti, eikä heitä poikkeusta.



Kuva 1 Validoidjassa käytettävät testitapahtumat

7.2.6 testVirheellisetPoissaolot()

Metodissa testataan validoijaa tapahtumalistalla jossa on virheellisiä poissaolotapahtumia. Kaikilla virheellisillä tapahtumalistoilta testataan, että validointimetodi heittää poikkeuksen. Metodissa testataan seuraavat tapaukset:

- poissaolo päättyy päivää myöhemmin kuin työsuhte: T3, P4
- poissaolo ennen työsuhteita: P1, T3, T5
- poissaolo limittäin kahden perättäisen työsuhteen päällä: T4, P5, T3, K
- poissaolo alkaa ennen voimassaolevaa työsuhdetta: P1, T3, P3, T5, P9
- ensin hyväksytyt poissaolo, sen jälkeen saman työsuhteen sisällä virheellinen: T3, P3, P11

7.2.7 testVirheellisetTyösuhteet()

Metodissa testataan validoijaa tapahtumalistoilta, joissa on virheellisiä työsuhteita. Metodissa testataan, että validoija huomaa virheet, ja heittää kaikissa tapauksissa poikkeuksen. Metodissa testataan seuraavat tapaukset:

- työsuhteessa päivämäärät väärässä järjestyksessä: T1, T6
- työsuhteesta puuttuu molemmat päivämäärät: T9
- työsuhteen alkupäivämäärä puuttuu: T10
- päällekkäisiä työsuhteita: T1, T2, T5
- voimassaoleva työsuhte, joka ei ole IJKK:ssa: T1, T8
- kaksi voimassaolevaa työsuhdetta: T5, T12
- voimassaolevassa työsuhteessa käsin syötettyjä etuisuuksia: T2, T3, K, T13
- voimassaolevan työsuhteen kanssa päällekkäinen työsuhte: T2, T5, T11

7.2.8 testYksinäinenPoissaolo()

Metodissa testataan validoijaa tapahtumalistoilla, jotka sisältävät yksittäisen poissaolon, ilman yhtäkään työsuhdetta. Testataan että validoija heittää poikkeuksen tilanteessa jossa tapahtumalistassa on pelkkä poissaolo P1, ja P1:n lisäksi koulutustapahtumia.

7.2.9 testYksinäinenTyösuhde()

Metodissa testataan validoijaa tapahtumalistoilla, joissa on pelkkä yksinäinen työsuhdetapahtuma. Testaa, että validoija hyväksyy listan, jossa on pelkkä päättynyt työsuhde T1, voimassaoleva työsuhde T5, tai voimassaoleva työsuhde T5, ja sen lisäksi koulutustapahtumia.

8 Hyväksyntätestauksen testitapauksia

8.1 T1: Henkilön lisääminen järjestelmään, väärä sotu

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyssä.

Kuvaus:

1. Paina ”Insert new row” (+).
2. Kirjoita sotu-kenttään: Sotu: ’090862-064B’
3. Siirry Sukunimi-kenttään.

Lopputila: Näytölle ilmestyy virheilmoitus Tarkasta sotu. Kuittaa painamalla OK.

8.2 T2: Henkilön lisääminen järjestelmään

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyssä.

Kuvaus:

1. Paina ”Insert new row” (+). Kirjoita seuraavat tiedot ao. kenttiin:
2. Sotu: ’090862-064A’,
3. Sukunimi: ’Lehtonen’,
4. Etunimet: ’Anna Leena’,
5. Kutsumanimi: ’Anna’,
6. Katuosoite: ’Radiotie 5’,
7. Postinumero: ’04220’,
8. Postitoimipaikka: ’Järvenpää’,
9. Henkilöstöryhmä: valitse ”OPE”,
10. Nimike: ’Opettaja’,
11. Koti puh: ’019433169’,
12. Työ puh: ’0194221713’,
13. GSM: ’0507725073’,
14. Email: ’anna.lehtonen@ijkk.fi’,
15. Tila: valitse ”töissä”.
16. Tämän jälkeen paina ”Save changes”.

Lopputila: Henkilö lisättiin näytöllä näkyvään listaan..

8.3 T3: Henkilön tietojen tarkastelu ja haku

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyssä.

Kuvaus:

1. Valitse etsittäväksi kentäksi ”Sukunimi”.
2. Kirjoita Etsittävä arvo-kenttään ’Lehtonen’.

Lopputila: Henkilön tiedot ilmestyvät näytölle.

8.4 T4: Henkilön tietojen muuttaminen

Alkutila: Käyttäjä on etsinyt käsiteltävän henkilön henkilöluettelosta etsimistoiminnon kautta.

Kuvaus:

1. Syötä Työ puh-kenttään uusi puhelinnumero '0194221777'.
2. Tämän jälkeen valitse "Save changes".

Lopputila: Henkilön uusi puhelinnumero tallentui.

8.5 T5: Henkilölistaus

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyvässä.

Kuvaus:

1. Siirry Raportit-välilehdelle.
2. Valitse Raporttivalinnat "Luettelot",
3. Luettelon tyyppi "Henkilöluettelo (suppea)",
4. Raportissa näytettävät henkilöstöryhmät "Opettajat" ja "Muu henkilöstö"
5. Paina "Luo Raportti".

Lopputila: Näytölle ilmestyy henkilölistaus valituista henkilöistä.

8.6 T6: Henkilöluettelon tulostaminen

Alkutila: Käyttäjä on valinnut ruudulle näkyvään henkilöluetteloan halutut henkilöt (T5).

Kuvaus:

1. Paina "Tulosta".

Lopputila: Ruudulla näkyvä luettelo tulostuu paperille.

8.7 T7: Ansiomerkin lisääminen

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyvässä.

Kuvaus:

1. Siirry Tapahtumat-välilehdelle.
2. Tapahtumatyyppin lisäys: Paina "Insert new row" (+),
3. valitse tyyppi "ansiomerkki"
4. kirjoita tapahtuman nimi-kenttään 'ansiomerkki 10 v'.
5. Paina "Save changes".
6. Ansiomerkin lisäys: Paina "Insert new row" (+),
7. Valitse tapahtuman tyyppi "ansiomerkki 10v"
8. pisteeksi "Järvenpään koulutuskeskus".
9. Valitse henkilö "Lehtonen Anna".
10. Paikka 'IJKK'
11. Kirjoita Alku pvm kenttään '3.12.2004'.
12. Lisätietoja-kenttään kirjoita 'Muistoksi Aarikan kynttilänjalka'.
13. Paina "Save changes".

Lopputila: Tapahtuma lisättiin näytöllä näkyvään listaan..

8.8 T8: Kunniamerkin myöntäminen

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyvässä.

Kuvaus:

1. Siirry Tapahtumat-välilehdelle.
2. Tapahtumatyyppin lisäys: Paina "Insert new row" (+),
3. valitse tyyppi "kunniamerkki"
4. kirjoita tapahtuman nimi-kenttään 'kunniamerkki'.
5. Paina "Save changes".
6. Kunniamerkin lisäys: Paina "Insert new row" (+),
7. Valitse tapahtuman tyyppi "kunniamerkki"
8. pisteeksi "Järvenpään koulutuskeskus".
9. Valitse henkilö "Lehtonen Anna".
10. Tapahtumakuvaus kenttään 'Rauhanajan maanpuolustusmitali'
11. Paikka 'IJKK'
12. Kirjoita Alku pvm kenttään '3.12.2004'.
13. Lisätietoja-kenttään kirjoita 'Presidentti Halonen myönsi ansiokkaasta avustustyöstä, muistoksi seinäkello kaiverruksin'.
14. Paina "Save changes".

Lopputila: Tapahtuma lisättiin näytöllä näkyvään listaan..

8.9 T9: Edellisen työsuhteen lisääminen

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyvässä.

Kuvaus:

1. Siirry Tapahtumat-välilehdelle.
2. Tapahtumatyyppin lisäys: Paina "Insert new row" (+),
3. valitse tyyppi "työkokemus"
4. kirjoita tapahtuman nimi-kenttään 'työkokemus opetustyö muualla'.
5. Valitse automaattinen MVK, VL, KL ja LO.
6. Paina "Save changes".
7. Työkokemuksen lisäys: Paina "Insert new row" (+),
8. valitse tapahtuman tyyppi "työkokemus opetustyö muualla",
9. pisteeksi "muu"
10. henkilöksi "Lehtonen Anna".
11. Kirjoita seuraavat tiedot ao. kenttiin: Paikka: 'Helsingin lyseo',
12. alku PVM: '01.08.1983',
13. loppu PVM: '31.07.1988',
14. Tapahtuma kuvaus: 'Lukion äidinkielen sijainen',
15. Paikkakunta: 'Helsinki'.
16. Tarkasta näytölle ilmaantuvat kertymät (MVK, VL, KL ja LO: 60).
17. Paina "Save changes".

Lopputila: Työsuhte lisättiin näytöllä näkyvään listaan.

8.10 T10: Nykyisen työsuhteen lisääminen

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyvässä.

Kuvaus:

1. Siirry Tapahtumat-välilehdelle.
2. Tapahtumatyyppin lisäys: Paina "Insert new row" (+),
3. valitse tyyppi "työkokemus"
4. kirjoita tapahtuman nimi-kenttään 'työkokemus opetustyö meillä'.
5. Valitse automaattinen MVK, VL, KL, LO ja AM.

6. Paina ”Save changes”.
7. Työkokemuksen lisäys: Paina ”Insert new row” (+),
8. valitse tapahtuman tyyppi ”työkokemus opetustyö meillä”,
9. pisteeksi ”Järvenpään koulutuskeskus”
10. henkilöksi ”Lehtonen Anna”.
11. Kirjoita seuraavat tiedot ao. kenttiin: Paikka: ’Järvenpään koulutuskeskus’,
12. alku PVM: ’01.08.1988,
13. loppu PVM: jätetään tyhjäksi,
14. Tapahtuma kuvaus: ’Erityisopettaja’,
15. Paikkakunta: ’Järvenpää’.
16. Tarkasta näytölle ilmaantuvat kertymät (MVK, VL, KL, LO: 196 ja AM: 16).
17. Paina ”Save changes”.

Lopputila: Työsuhte lisättiin näytöllä näkyvään listaan.

8.11 T11: Nimikirjanotteen muodostaminen

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyvässä.

1. Siirry Raportit-välilehdelle.
2. Valitse Raporttivalinnat ”Tapahtumalista”,
3. Valitse henkilö ”Lehtonen Anna”,
4. Paina ”Luo Raportti”.
5. Tarkasta raportin tiedot.

Lopputila: Henkilön nimikirjanote ilmestyy näytölle ja tiedot ovat oikein.

8.12 T12: Nimikirjanotteen tulostaminen

Alkutila: Käyttäjä on avannut käsiteltävän henkilön nimikirjanotteen näytölle (T11).

Kuvaus:

1. Paina ”Tulosta”.

Lopputila: Ruudulla näkyvä luettelo tulostuu paperille.

8.13 T13: Vuosilomaluettelon muodostaminen

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyvässä.

1. Siirry Raportit-välilehdelle.
2. Valitse Raporttivalinnat ”Luettelot”,
3. Luettelon tyyppi ”Vuosilomaluettelo”,
4. Raportissa näytettävät henkilöstöryhmät ”Opettajat”
5. Paina ”Luo Raportti”.
6. Tarkasta raportin tiedot.

Lopputila: Henkilön nimikirjanote ilmestyy näytölle ja tiedot ovat oikein.

8.14 T14: Vuosilomaluettelon tulostaminen

Alkutila: Käyttäjä on avannut vuosilomaluettelon näytölle (T14).

Kuvaus:

1. Paina ”Tulosta”.

Lopputila: Ruudulla näkyvä luettelo tulostuu paperille.

8.15 T15: Vuosikorotusilmoituksen muodostaminen

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyvässä.

1. Siirry Raportit-välilehdelle.
2. Valitse Raporttivalinnat ”Vuosikorotusilmoitus”,
3. Valitse henkilö ”Lehtonen Anna”,
4. Paina ”Luo Raportti”.
5. Tarkasta raportin tiedot.

Lopputila: Henkilön vuosikorotusilmoitus ilmestyy näytölle ja tiedot ovat oikein.

8.16 T16: Vuosikorotusilmoituksen tulostaminen

Alkutila: Käyttäjä on avannut vuosikorotusilmoituksen näytölle (T16).

Kuvaus:

1. Paina ”Tulosta”.

Lopputila: Ruudulla näkyvä vuosikorotusilmoitus tulostuu paperille.

8.17 T17: Vuosikorotusluettelon muodostaminen

Alkutila: Käyttäjä on sisäänkirjautunut järjestelmään ja ohjelma on avausnäkyssä.

1. Siirry Raportit-välilehdelle.
2. Valitse Raporttivalinnat ”Luettelot”,
3. Luettelon tyyppi ”Vuosikorotusluettelo”,
4. Raportissa näytettävät henkilöstöryhmät ”Opettajat”
5. Paina ”Luo Raportti”.
6. Tarkasta raportin tiedot.

Lopputila: Henkilöiden vuosikorotusluettelo ilmestyy näytölle ja tiedot ovat oikein.

8.18 T18: Vuosikorotusluettelon tulostaminen

Alkutila: Käyttäjä on avannut vuosikorotusilmoituksen näytölle (T18).

Kuvaus:

1. Paina ”Tulosta”.

Lopputila: Ruudulla näkyvä vuosikorotusluettelo tulostuu paperille.