

Kasi

Yhteenvedodokumentti

Helsinki 6.5.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

## **Kurssi**

581260 Ohjelmistotuotantoprojekti (9+1op/6ov)

## **Projektiryhmä**

Lauri Holmas  
Veli-Pekka Kestilä  
Joni Lahtinen  
Tuukka Palomäki  
Markus Penttilä  
Antti-Pekka Sarin  
Ilkka Tikkala

## **Asiakas**

Harri Laine  
Petri Kutvonen

## **Vastuhenkilö**

Kimmo Simola

## **Ohjaaja**

Jaakko Saaristo, Ilari Moilanen

## **Kotisivu**

[www.cs.helsinki.fi/group/kasi](http://www.cs.helsinki.fi/group/kasi)

## **Versiohistoria**

<u>Versio</u>	<u>Päiväys</u>	<u>Tehdyt muutokset</u>
0.1	23.4.2007	ensimmäinen versio
0.2	29.4.2007	lisätty kappaleita
1.0	6.5.2007	lopullinen versio

## Sisällys

1 Johdanto.....	1
2 Sanasto.....	1
3 Lopputuotteen esittely.....	2
3.1 Tietokanta.....	3
3.2 Ostoliittymä.....	3
3.3 Raporttiliittymä.....	5
4 Dokumenttien tiivistelmät.....	7
4.1 Projektisuunnitelma.....	7
4.2 Vaatimusdokumentti.....	7
4.3 Suunnitteludokumentti.....	7
4.4 Testaussuunnitelma.....	8
4.5 Käyttöohje.....	8
4.6 Ylläpitodokumentti.....	8
5 Projektin päättöanalyysi.....	9
5.1 Projektin työryhmä.....	9
5.1.1 Työtunnit.....	10
5.2 Projektin työtavat ja kommunikaatio.....	10
5.3 Projektissa käytetyt työvälineet.....	10
5.4 Projektin onnistuminen.....	11

# 1 Johdanto

Kasi on Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojekti Ohjelmistotuotantoprojekti-kurssin kevään 2007 projektiryhmä. Projektin tehtävänä oli toteuttaa uusi versio kesällä 2006 toteutetusta sormenjälkitunnistukseen perustuvasta kahvikassajärjestelmästä. Kesällä 2006 toteutetussa järjestelmää ei ole otettu käyttöön siinä olevien ongelmien vuoksi. Kevään 2007-projektissa ei jatkokehitetty aikaisemmin kehitettyä järjestelmää, vaan toteutettiin täysin uusi ohjelmisto. Uudessa versiossa otettiin sormenjälkitunnistimen lisäksi käyttöön kosketusnäyttö. Kassatilanteen seuranta toteutettiin selaimen kautta toimivana web-sovelluksena. Ohjelmointi suoritettiin kahdella eri ohjelmointi kielellä, joista Javaa käytettiin ostoliittymän toteutukseen ja raporttiliittymä kirjoitettiin web-ohjelmointiin sopivammalla PHP-ohjelmointikielellä sekä Javascriptillä. Projektissa käytettiin kaksisyklistä projektimallia, jossa ensimmäinen versio käytiin asiakkaan kanssa läpi ja tehtiin toinen suunnittelu ja ohjelmointi kierros.

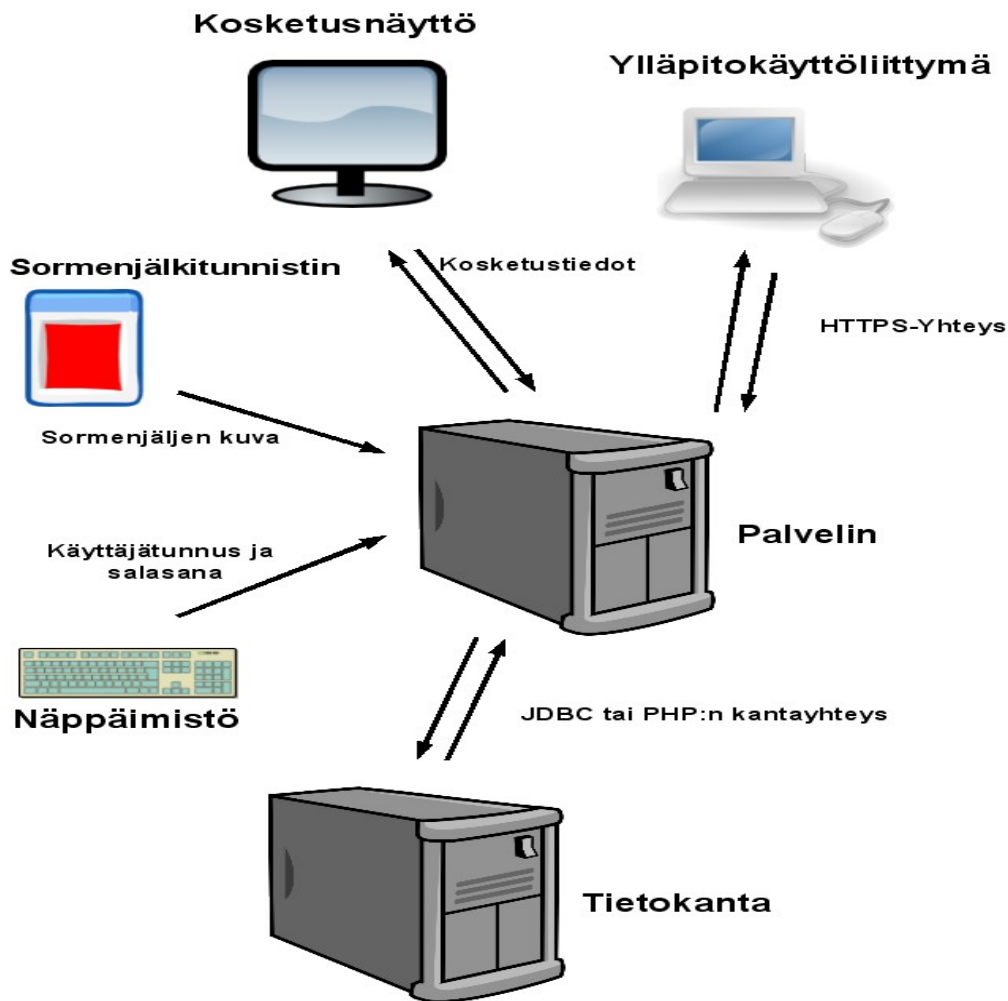
Tämä dokumentti esittelee projektin tuotoksena syntyneen CoffeeTouch-ohjelmiston sekä muut projektin tuottamat dokumentit. Dokumentti sisältää myös projektin päätösanalyysin. Dokumentin tavoitteena on antaa yleiskuva projektin tuotoksista, ennen kuin lukija tutustuu muihin dokumentteihin, joista löytyvät tarkemmat yksityiskohdat.

## 2 Sanasto

Kuvaus	Selitys
<b>CoffeeTouch</b>	Kahvikassa-ohjelmisto
<b>EUCT</b>	<i>Extended Use Case Test</i> Laajennettu käyttötapaus
<b>Ostoliittymä</b>	Osajärjestelmä, jonka kautta käyttäjät pystyvät ostamaan ja tuomaan tuotteita kahvihuoneessa
<b>Raporttiliittymä</b>	Osajärjestelmä, joka on tarkoitettu käyttäjien ja tuotteiden hallinnointiin, sekä raporttien tuottamiseen
<b>Vakioannos</b>	Käyttäjän itselleen asettama oletustuote
<b>Web-sovellus</b>	Selaimen välityksellä käytettävä palvelinsovellus

### 3 Lopputuotteen esittely

CoffeeTouch-ohjelma on sormenjälkitunnistukseen ja kosketusnäyttöön perustuva työpaikan kahvikassajärjestelmä. Ohjelman avulla voidaan kirjata käyttötapahtumat, tuotteiden tuonti, käytön seuranta ja tuotteiden hallinta helposti. CoffeeTouch-ohjelma toimii sekä Linux- ja Windows XP-alustalla. Järjestelmä jakautuu kolmeen osajärjestelmään: tietokantaan, ostoliittymään ja raporttiliittymään. Kuvassa 1. esitellään ohjelman toimintaympäristö ja sen osat.



Kuva 1: Yleiskuva järjestelmästä

## 3.1 Tietokanta

Tietokanta on toteutettu siten, että se ei ole sidoksissa mihinkään määrättyyn tietokantajärjestelmään. Tietokannan yksinkertaistamiseksi kuitenkin on käytetty joitakin tietokantojen ominaisuuksia, jotka on toteutettu kantariippuvasti. Toteutusvaiheessa kantariippuvuudesta johtuen on laadittu erilliset kannanluontilauseet kohdetietokannoiksi tarkoitetuille PostgreSQL:lle ja Oraclelle. Normaalit haut ja lisäykset on toteutettu siten, että niihin ei tarvitse tehdä muutoksia, jos tietokantaa vaihdetaan. Jos lauseita kuitenkin joudutaan muuttamaan, ne on tallennettu ohjelmiston asetustiedostoihin, joissa niiden editointi on helppoa, ja ne ovat heti käytettävissä ohjelman uudelleenkäynnistyksen jälkeen.

## 3.2 Ostoliittymä

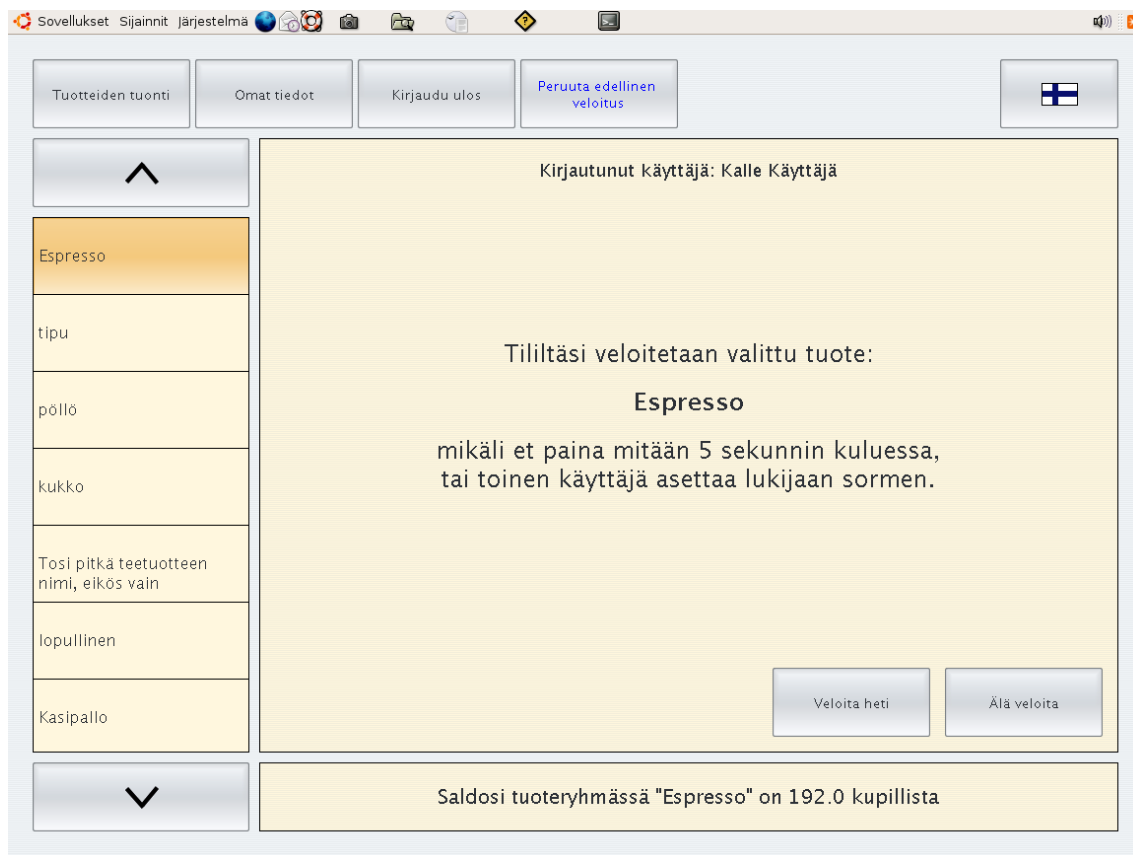
Ostoliittymä toimii kahvihuoneen näkymänä järjestelmään. Kuvassa 2. esitellään ostoliittymän aloitusnäkyä. Ostoliittymän avulla lisätään uusia käyttäjiä sekä rekisteröidään ja muokataan heidän sormenjälkiään. Rekisteröitymisvaiheessa voidaan määrittellä käyttäjälle vakioannos, jota voidaan muuttaa myöhemmin. Järjestelmään tuodaan tuotteita sekä kirjataan käyttötapahtumat sormenjälkitunnistimen ja kosketusnäytön avulla. Ostoliittymä keskustelee tietokannan lisäksi myös kosketusnäytön (joka korvaa hiiren osoitinlaitteena), sormenjälkilukijan ja näppäimistön kanssa.



Kuva 2: Ostoliittymän aloitusnäky

Kosketusnäyttöä käytetään pääasiallisena syöttö ja osoitinlaitteena. Sormenjälkilukijan avulla käyttäjä tunnistetaan järjestelmässä. Näppäimistön avulla käyttäjä syöttää käyttäjätunnuksensa ja salasansansa rekisteröimisvaiheessa, jotta käyttäjä saadaan yhdistettyä Tietojenkäsittelytieteen laitoksen käyttäjätietoihin tunnistustarpeita varten. Ostoliittymä käyttää jo rekisteröityjen käyttäjien tunnistamiseen sormenjälkitunnistusta. Sormenjäljen/-jäljet käyttäjä syöttää käyttäessään järjestelmää ensimmäisen kerran. Järjestelmään syötettyjä sormenjälkiä voi muokata myöhemmin.

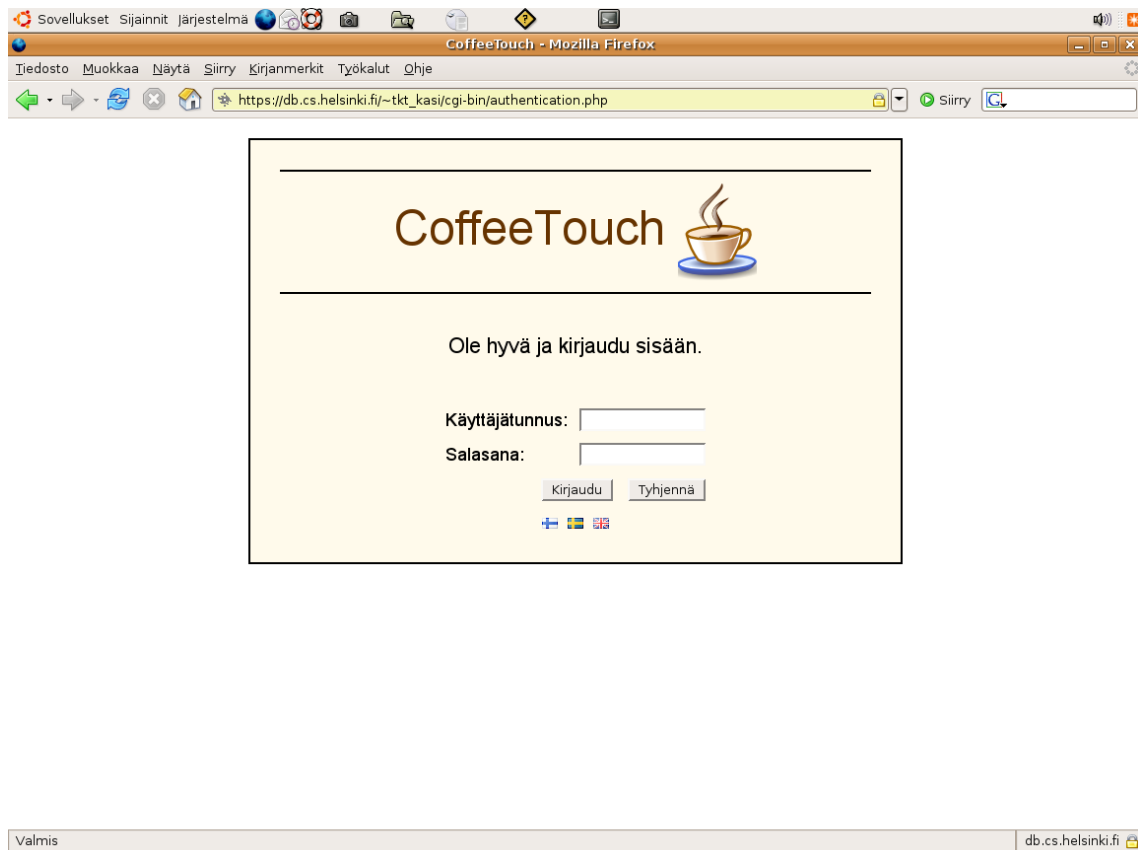
Rekisteröitynyt käyttäjä pystyy suorittamaan vakioannoksen oston vain asettamalla sormensa sormenjälkilukijaan. Järjestelmään voi myös kirjautua syöttämällä käyttäjätunnuksen ja salasanan. Kuvassa 3. näkymä ostojärjestelmästä tunnistetun käyttäjän suorittaessa vakioannoksen oston.



Kuva 3: Vakioannoksen ostonäkymä

### 3.3 Raporttiliittymä

Raporttiliittymä antaa verkkoselaimella käytettävän, HTML-pohjaisen näkymän ohjelman tietovarastoon. Kuvassa 4. raporttiliittymän sisäänkirjautumisnäky. Raporttiliittymään liitytään normaalin www-selaimen avulla käyttäen HTTPS-protokollaa. Sen kautta käyttäjät pystyvät seuraamaan omaa kahvinkulutustaan ja koko järjestelmän toimintaa. Käyttäjät pääsevät näkemään järjestelmän kaikki käyttäjät, mutta voivat vain muokata omia nimitietojaan.



Kuva 4: Raporttiliittymän kirjautumisnäky

Pääkäyttäjät voivat raporttiliittymän kautta muuttaa tarjolla olevaa tuotevalikoimaa ja poistaa käyttäjiä. Raporttien avulla voidaan seurata ostotapahtumia ja tuotteiden tuontia käyttäjäkohtaisesti. Järjestelmässä olevien ryhmien kokonaissaldoa voidaan seurata raporttiliittymän kautta luotavan raportin avulla (kuva 5).



Sovellukset Sijainnit järjestelmä

CoffeeTouch - Mozilla Firefox

Tiedosto Muokkaa Näytä Siirry Kirjanmerkit Työkalut Ohje

https://db.cs.helsinki.fi/~tk\_kasi/cgi-bin/reportsGroups.php?dds=04&mms=04&yys=2007&dde=04

# CoffeeTouch

## Ryhmä- ja tuotekohtainen raportti

Valitse vasemmalta ensin tuoteryhmä ja halutessasi sen alta jokin tietty tuote. Raportissa näytetään tuoteryhmän tai tuotteen käyttö aikavälillä ja keskimäärin päivää kohden.

**Raportit**

[Käyttäjäkohtainen tuonti](#)

[Käyttäjäkohtainen vienti](#)

[Tuotteiden arvojen muutokset](#)

Ryhmä- ja tuotekohtainen raportti

[Kaikkien käyttäjien saldot](#)

---

**Tuotteet**

---

**Käyttäjät**

---

**About CoffeeTouch**

---

**Kirjautu ulos**

Login: [ijikkal](#)



**Valitse raportin aikaväli**

4	4	2007	
4	5	2007	

**Tuoteryhmät**

**Espresso**

**Tuotavat tuotteet**

[Espresso kahvi](#)

[Maito](#)

[Sokeri](#)

**Vietävät tuotteet**

[Espresso](#)

[Espresso maidolla](#)

[Tuupla espresso](#)

[Kaffepullia](#)

[Kasi](#)

[Puu](#)

[Tappio](#)

[Tavallinen kahvi](#)

[Tee](#)

[Tyo](#)

[Uusi ryhmä](#)

[A-ryhmä](#)

**Espresso**

Aikaväli: 31 päivää

Suosituin tuontituote: Espresso kahvi

Suosituin vientituote: Espresso

Tuonteja yhteensä: 9, arvoitaan 300 kuppia

Vientejä yhteensä: 3, arvoitaan 3 kuppia

Arvon erotus (tuonti - vienti): 297 kuppia

Keskimäärin per päivä:

0.3 tuontia, arvo 9.7 kuppia

0.1 vientiä, arvo 0.1 kuppia

Arvon erotus (tuonti - vienti): 9.6 kuppia

Valmis db.cs.helsinki.fi

Kuva 5: Raporttisivu

## 4 Dokumenttien tiivistelmät

Tämä luku sisältää tiivistelmät projektin aikana syntyneistä dokumenteista. Dokumentteja on tuotettu projektin prosessin aikana runsaasti.

### 4.1 Projektisuunnitelma

Projektisuunnitelma on ensimmäinen projektin aikana syntynyt dokumentti. Se on tarkoitettu projektiryhmän sisäiseen käyttöön ohjaamaan projektin etenemistä alusta päättymiseen asti. Suunnitelmassa kuvataan projektiryhmä ja ryhmän jäsenten vastuualueet ja vastuualueen varahenkilö.

Projektisuunnitelmassa on tehty riskianalyysi. Riskianalyysissä käsitellään projektin aikana mahdollisesti syntyviä riskejä, jotka muodostavat uhan projektin etenemiselle. Riskien todennäköisyys on arvioitu, vakavuus selvitetty, kuvattu varautuminen kyseiseen riskiin ja miten selvittää tilanteesta, jossa riski toteutuu.

Vaatimusmäärittelyn jälkeen projektisuunnitelmaan on laadittu ohjelmiston kokoarvio. Kartoitettujen ominaisuuksien arvioitiin noin 9500 koodiriviä, kun ohjelmointikieliä ovat Java ja PHP.

Projektisuunnitelma sisältää myös projektille laaditun aikataulun ja työn osituksen. Projektin aikataulua on päivitetty työn edetessä.

### 4.2 Vaatimusdokumentti

Vaatimusdokumentin tarkoituksena on tarkentaa toteutettavalle järjestelmälle ohjelmisto- ja järjestelmävaatimukset. Vaatimusdokumentti toimii runkona ohjelmiston arkkitehtisuunnittelulle, tarkemmalle suunnittelulle ja järjestelmätestaussuunnittelulle sekä toimii integrointitestauksen perustana. Se on laadittu asiakkaan kanssa pidettyjen kokousten perusteella. Dokumentti määrittelee toteutettavan ohjelmiston ja toimii sopimuksena asiakkaan ja projektiryhmän välillä. Projektiryhmä katselmoi vaatimusdokumentin projektin ensimmäisessä syklissä.

Projektin toisen syklin alussa esiteltiin ensimmäisen syklin aikana kehitettyä ohjelmistoa asiakkaalle. Tämän perusteella ei toiseen sykliin tullut muutoksia vaatimusmäärittelyyn, vaan tuotteen kehittämistä jatkettiin alkuperäisen vaatimusmäärittelyn mukaan.

### 4.3 Suunnitteludokumentti

Suunnitteludokumentissa käydään läpi vaatimusmäärittelyn pohjalta projektin ohjelmistoarkkitehtuuri tarvittavalla tarkkuudella toteutuksen mahdollistamiseksi. Ensin arkkitehtuuri kuvataan yleisellä tasolla ja sen jälkeen esitellään ohjelmiston tietokannan rakenne. Ohjelmiston tarkempi arkkitehtuuri on kuvattu omissa luvuissaan. Näissä kuvataan niiden komponentit ja komponenttien väliset rajapinnat. Lisäksi kuvataan rajapinto-

jen käyttämät tieto-oliot. Olio- ja komponenttikuvauksien tekemiseen on käytetty UML-kieltä, ja toteutukseen ja luokkasuunnitteluun tarvittavat tiedot on kirjoitettu auki.

Tarkemmat API-kuvaukset on kuvattu Javadoc, PHPdoc ja Jsdoc dokumenteissa, jotka on luotu järjestelmän toteutuksen aikana. Suunnitteludokumentin liitteenä ovat luokka-kaaviot. Tietokannan luontilauseet ovat erillisessä dokumentissa.

## 4.4 Testaussuunnitelma

Testausdokumentti sisältää testaussuunnitelman ja testausraportin. Testaussuunnitelmas-  
sa on ohjeet järjestelmän testauksen toteuttamiseen. Periaatteena on, että luokan toteut-  
taja on yksikkötestauksen suorittaja. Testaussuunnitelma antaa ohjeet testauksen  
toteuttamiseen ja antaa kattavuuskriteerin testaukseen. Valittu lausekattavuuskriteeri  
näiden testien osalta on 90 %.

Projektisuunnitelman mukaan integrointitestausta ei suoriteta omana erillisenä vaihee-  
naan, vaan se sisällytetään laajennettujen käyttötapauksen kautta tapahtuvaan järjestel-  
mätestaamiseen.

Järjestelmätestauksen tavoitteena on osoittaa, että ohjelmisto täyttää sille vaatimusmää-  
rittelydokumentissa asetetut vaatimukset, eli asiakkaan verifioimat käyttötapaukset voi-  
daan suorittaa ohjelmistolla. Järjestelmätestauksessa lähestymistavaksi on valittu laajen-  
nettujen käyttötapauksen EUCT (Extended Use Case Test) käyttö. Laajennetut käyttöta-  
paukset käydään läpi skenaario-skenaariolta ja katsotaan, että jokainen skenaario omalla  
syötteellään tuottaa halutun lopputuloksen. Mikäli toisen syklin järjestelmätestauksessa  
ilmenee sellaisia ohjelmiston toimintaan liittyviä virheitä, joita ei aikataulun kiireelli-  
syyden vuoksi voida korjata, nämä kirjataan ylläpidodokumenttiin.

## 4.5 Käyttöohje

Käyttöohjeessa annetaan ohjeet ohjelmiston käynnistämiseen ja käyttämiseen. Varsinai-  
set asennusohjeet ylläpidodokumentin yhteydessä. Käyttöohjeessa kuvataan ohjelmiston  
kaikki toiminnot. Toiminnot kuvataan siten, että loppukäyttäjä voi oppia käyttämään oh-  
jelmistoa sen avulla.

Käyttöohjeessa käsitellään ensin ostoliittymän käyttö ja sen jälkeen käsiteellään raport-  
tiliittymän käyttö.

## 4.6 Ylläpidodokumentti

Ylläpidodokumentissa kuvataan kummankin osajärjestelmän, Java-kielellä toteutetun,  
kahvihuoneeseen tarkoitetun asiakasliittymän, ja PHP-kielellä toteutetun verkkopalveli-  
men kautta käytettävän raportointiliittymän asennus ja asetusten säätö.

Lisäksi dokumentissa on otettu huomioon järjestelmän jatkokehittäjien mahdolliset tar-  
peet. Dokumentissa kuvataan miten järjestelmä on tuotettu, mitä työkaluja on käytetty ja  
millaisia ratkaisuja on projektissa tehty.

Dokumentissa pyritään myös antamaan jatkokehittäjille kehitysideoita ja selvittämään niistä mahdollisimman tarkkaan taustat, jotta idean toteuttaminen olisi mahdollisimman selkeä operaatio.

Ohjelmistossa olevat selkeät virheet ja puutteet on myös dokumentoitu mahdollisimman laajasti ja tarkasti.

## 5 Projektin päättöanalyysi

Projekti alkoi tammikuussa uhmakkaasti. Seitsenhenkisen projektitiimin tavoitteena oli tehdä laadukas ja asiakkaan tarpeet täyttävä ohjelmisto, joka valmistuisi nopeasti. Toteutustavaksi valittiin kaksi syklinen projektimalli ja alettiin töihin. Kaikki ei kuitenkaan mennyt projektin edetessä ihan suunnitelmien mukaan ja projektissa oli muutamia kriittisiä vaiheita.

Tällaisia kriittisiä vaiheita oli suunnittelun loppuvaiheissa tullut väsymys sen jatkuvaan hiomiseen. Tämän ongelman kanssa päällekkäin tuli vielä projektin ohjaajan vaihdos, joka uusine vaatimuksineen söi myös projektin moraalialueita. Seuraava ongelma olikin sitten, että emme olleet pysyneet aluksi laaditussa aikataulussa. Tämä aikataulun pitämättömyys ja muun projektiryhmän tyytymättömyys projektipäällikön vastuulla olleeseen raporttiliittymään aiheuttivat sitten projektipäällikön vaihtamisen.

Tämän jälkeen aika on kulunut varsin liukkaasti raporttiliittymän korjaamisessa, sekä ostoliittymän bugien korjaamisessa ja dokumentaatiota viimeistellessä. Projekti on siihen käytettävissä olevaan aikaan nähden kohtalaisen hyvässä kunnossa.

### 5.1 Projektin työryhmä

Projektin työryhmä koostui seitsemästä henkilöstä. Projektin aluksi kaikille määriteltiin vastuualue ja varavastuualue. Projektin edetessä monet jäsenistä auttoivat myös muiden alueiden töissä. Viikolla 14 kuitenkin jouduimme tilanteeseen, jossa ohjaaja ryhmän kanssa keskusteltuaan vaihtoi projektipäällikköä.

Alkuperäinen työnjako oli seuraava:

<b>Nimi</b>	<b>Tehtävä</b>	<b>Varalla</b>
Holmas Lauri	Määrittelyvastaava	Tikkala
Kestilä Veli-Pekka	Suunnitteluvastaava	Penttilä
Lahtinen Joni	Koodausvastaava	Kestilä
Palomäki Tuukka	Projektipäällikkö	Holmas
Penttilä Markus	Käyttöliittymävastaava	Lahtinen
Sarin Antti-Pekka	Testausvastaava	Palomäki
Tikkala Ilkka	Dokumentointivastaava	Sarin

Projektipäällikön vaihdon yhteydessä määriteltiin työnjakoa uusiksi siten, että Veli-Pekka Kestilästä tuli uusi projektipäällikkö ja Tuukka Palomäki siirtyi suunnitteluvastavaksi. Kuitenkin positiivista on, että tämän tai muiden projektin kohtaamien ongelmien johdosta kukaan projektiryhmästä ei keskeyttänyt.

Lisäksi on mainittavaa, että ryhmän jäsenet pääasiassa osallistuivat kaikkiin projektin vaiheisiin paikaten ja auttaen muita näiden osa-alueilla kykyjensä ja käytettävissä olevan aikansa mukaisesti.

### **5.1.1 Työtunnit**

Ryhmän jäsenet pitivät kirjaa työtunneistaan Tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojekteille tarkoitetussa työtuntien seurantajärjestelmässä. Työtunteja tuli kaikenkaikkiaan raporttijärjestelmän mukaan 1900 tuntia ja luultavasti se on hieman alakanttiin projektiin oikeasti käytettyjen tuntien kanssa. Oikeampi työmäärä on luultavasti lähempänä 2000 tuntia tai hieman sen yli.

Muuten työtunneissa oli kohtalaisesti hajontaa. Vähiten tehneellä on hieman yli 200-tuntia kasassa ja eniten töitä tehneellä hieman yli 300-tuntia. Projektin keskiarvon ollessa 270-tunnin paikkeilla.

## **5.2 Projektin työtavat ja kommunikaatio.**

Projektia tehtiin pääasiassa hajautettuna ryhmätyönä, eli projektin etenemisen kannalta tärkein väline oli projektin IRC-kanava. Sen avulla ryhmän toimintaa koordinoitiin ja jäsenet saivat nopeasti palautetta tekemisistään. Projekti piti myös kaksi noin kaksituntista palaveria viikoittain, joissa käytiin tehtyjä asioita läpi ja suunniteltiin projektin etenemistä yleisellä tasolla. Lisäksi näissä palavereissa projektin ohjaaja pääsi tutustumaan projektin tilaan.

Projektilla oli käytössä myös WiKi, jota käytettiin ahkerasti dokumentteja editoidessa. Muuten projektin sähköpostilista oli varsin vähäisessä käytössä. Eli suurin osa kommunikaatiosta hoidettiin IRC:ssä. Lisäksi projektin www-sivut toimivat lähinnä valmistuneen dokumentaation tallettamiseen tarkoitettuna paikkana.

## **5.3 Projektissa käytetyt työvälineet.**

Projektin kuluessa ryhmä evaluoi ja otti sen perusteella käyttöön työvälineitä eri työvaiheiden toteuttamisen tueksi. Dokumentaatio hoidettiin OpenOffice-tekstinkäsittelyohjelmalla. Projektin aikataulusuunnitelma tehtiin aluksi OO:n Calc-taulukkolaskennalla ja loppuvaiheissa MS-Projektilla.

Ohjelman suunnittelussa käytettiin Poseidon for UML. uml-editoria, johon osa ryhmästä osti lisenssin ja loppuryhmä käytti ilmaista evaluaatioversiota. Sinällään ryhmän mielestä olisi ollut hyvä, että näin kriittiseen työvaiheeseen olisi laitoksen puolesta ollut tarjota jonkinlaisia työkaluja.

Ohjelmoinnissa käytettiin apuna Eclipseä ja siihen saatavia lisämoduuleita. Näiden avulla saatiin aikaiseksi toimiva ympäristö projektin Java, PHP ja JavaScript-kehitykseen. Lisäksi kaikki ohjelmakoodi pidettiin CVS-versionhallinnassa, joka sulautuu varsin hyvin Eclipsen kanssa käytettäväksi.

## 5.4 Projektin onnistuminen

Ryhmän mielestä projekti onnistui vaikeuksista huolimatta varsin hyvin. Kaikista ongelmista pääasiassa päästiin yli tai ne kierrettiin tavalla tai toisella. Ryhmän pääsi ehkä yllettämään projektin laajuus ja ongelmat pysyä aikataulussa, kun projektin työntekijöiden työnteke on ainoastaan heidän omantuntonsa varassa. Tähän tietysti olisi voinut auttaa, että projektissa olisi ollut alusta lähtien ankarampi projektipäällikkö, joka olisi säälimättä potkinut muita tekemään töitä.

Projektin ajankäytön suunnittelu olisi ehkä kannattanut hoitaa edelleen kaksisyklisellä mallilla, mutta kuitenkin siten, että ensimmäisessä syklissä olisi tehty projektin ostoliittymä ja toisessa syklissä projektin raporttiliittymä. Tällöin koko projektiryhmä olisi osallistunut molempien osien tekemiseen. Ja näin olisi eräiltä projektissa esiintulleilta ongelmilta vältytty.

Sinällään jos projektin laajuus olisi ollut hieman pienempi olisi voitu laatuun kiinnittää nykyistä enemmän huomiota. Lisäksi projektin loppudokumentaation tekemiseen olisi jäänyt hieman enempi aikaa, mikä olisi varmasti näkynyt myös sen laadun paranemisenä.