

# **Yhteenvedodokumentti**

Karstula

Helsinki 3.5.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

**Kurssi**

581260 Ohjelmistotuotantoprojekti (9 + 1 op)

**Projektiryhmä**

Juha-Pekka Juutilainen

Hannu Kukko

Antto Mäkinen

Antti Rajasärkkä

Ari Raunio

Mika Tantarimäki

**Asiakas**

Eija Mäntyharju

**Johtoryhmä**

Sanna Keskkioja

Kimmo Simola

**Kotisivu**

<http://www.cs.helsinki.fi/group/karstula>

**Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
0.1	16.4.2007	Dokumentti luotu
0.2	17.4.2007	Dokumentin päivitystä
0.3	18.4.2007	Dokumentin päivitystä
0.4	22.4.2007	Dokumentin päivitystä
0.5	23.4.2007	Dokumentin päivitystä
0.5	30.4.2007	Dokumentin päivitystä
1.0	2.5.2007	Lopullinen dokumentti

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Yleiskuvaus asiakkaasta ja projektin taustatietoa . . . . .	1
1.2	Yleiskuvaus projektista . . . . .	1
<b>2</b>	<b>Sanasto</b>	<b>1</b>
<b>3</b>	<b>Lopputuotteen esittely</b>	<b>3</b>
3.1	Teoriaosio . . . . .	3
3.2	Harjoitteluosio . . . . .	3
3.3	Testiosio . . . . .	4
3.4	Ylläpito-osio . . . . .	4
<b>4</b>	<b>Dokumenttien tiivistelmät</b>	<b>6</b>
4.1	Projektisuunnitelma (sisäinen) . . . . .	6
4.2	Vaatimusmäärittelydokumentti (ulkoinen) . . . . .	6
4.3	Suunnitteludokumentti (sisäinen) . . . . .	7
4.4	Testaussuunnitelma (sisäinen) . . . . .	8
4.5	Käyttöohje (ulkoinen) . . . . .	8
4.6	Ylläpitedokumentti (ulkoinen) . . . . .	9
<b>5</b>	<b>Projektin päättöanalyysi</b>	<b>9</b>
5.1	Asiakkaan erityispiirteet . . . . .	9
5.2	Käytetty prosessimalli . . . . .	10
5.3	Projektiryhmä ja ryhmän työnjako . . . . .	11
5.4	Ryhmän työskentelytavat . . . . .	11
5.5	Projektin onnistuminen . . . . .	12

# 1 Johdanto

## 1.1 Yleiskuvaus asiakkaasta ja projektin taustatietoa

Karstulan Evankelinen Kansanopisto (KEK) on lähihoitajia kouluttava pieni oppilaitos Keski-Suomessa, aloitusryhmiä on yksi vuosittain. Kyseisen alueen terveyskeskukset (kuusi kappaletta) pyysivät KEK:iltä lääkehoidon koulutuspakettia, joka sisältäisi seuraavat osa-alueet: Lääkeaineoppi, lääkelaskut, pistosharjoittelut. Koulutusta tulisi tarjota jatkossa jatkuvasti kaksi- neljä kertaa vuodessa tapahtuvina kursseina. Uuden säännön mukaan lähihoitajan täytyy uusina lääkelupa viiden vuoden välein, joten koulutettavia tulee jatkossa säännöllisesti vuosittain. Kurssi on helppo toteuttaa muilta osin, mutta lääkelaskut tuottavat usein ongelmia, koska muutama opiskelija jää suorittamaan niitä useampaan kertaan. Nämä muutamat sitovat opettajien voimavaroja kohtalaisen paljon.

KEK tarvitsi siis sellaisen lääkelaskuohjelman, jonka avulla opiskelija voisi suorittaa lääkelaskujen harjoittelun itsenäisesti. Tästä olisi myös se hyöty, että opiskelija voisi suorittaa laskut itselleen sopivana aikana. Useimmat terveyskeskuksen henkilökunnasta ovat kolmivuorotyössä ja yhteisen ajan järjestäminen on usein vaikeaa.

## 1.2 Yleiskuvaus projektista

Projektiryhmä koostui Helsingin yliopiston tietojenkäsittelytieteen opiskelijoista, joiden tarkoituksena on toteuttaa Karstulan evankeliselle kansanopistolle lääkelaskujen harjoitteluohjelma. Ohjelma toteutettiin osana Tietojenkäsittelytieteen Ohjelmistotuotantoprojektin -kurssia keväällä 2007. Kurssin tavoitteena on harjoitella ohjelmiston tuottamista alusta alkaen. Kurssin sisältö voidaan jakaa karkeasti neljään eri osa-alueeseen: määrittelyvaihe, suunnitteluvaihe, toteutusvaihe, testausvaihe.

# 2 Sanasto

Tässä luvussa määritellään ohjelman dokumentaatiossa käytetyt termit.

**Desimaalilasku:** Yksi laskutyypeistä. Desimaalilaskussa joko itse lasku tai sen lopputulos (tai molemmat) sisältävät desimaaliluvun.

**Harjoittelujakso:** Harjoittelujaksolla tarkoitetaan harjoitteluosiossa olevaa yhden laskutyyppin yhtämittaista harjoittelua. Yksi harjoittelujakso kestää niin kauan, kunnes käyttäjä on vastannut oikein viiteen (oletusarvoisesti) laskuun ilman vihjeitä.

**Harjoitteluosio:** Harjoitteluosiolle tarkoitetaan ohjelman osaa, jossa käyttäjä voi harjoitella lääkelaskujen laskemista.

**Harjoitteluosion pääsivu:** Sivun, joka avautuu, kun valitaan ohjelman pääsivulta harjoitteluosio. Tällä sivulla käyttäjä voi valita mitä laskutyyppiä hän haluaa harjoitella.

**Helppo lasku:** Ylläpitäjä valitsee laskulle vaikeustason (helppo tai normaali) syöttäes-

sään uutta harjoitteluosion laskua ylläpitosivun kautta. Helppo lasku tarkoittaa laskua, joka ei välttämättä ole sanallinen (kuten testissä) vaan voi olla esimerkiksi mekaaninen, numeroihin perustuva lasku (esim. Paljonko on 20 % luvusta 50).

**Laskutyypit:** Laskut on jaettu neljään eri kategoriaan: murto-, desimaali-, prosentti- ja verrantolaskut. Näitä kutsutaan yhteisnimellä laskutyypit.

**Läpipääsytodistus:** Ohjelman tulostama paperi, josta ilmenee, että ohjelman käyttäjä on päässyt testiosion läpi.

**Läkelasku:** Läkelaskulla tarkoitetaan kaikkia niitä laskutoimituksia, joita ohjelmalla lasketaan.

**Murtolasku, murtolukulasku:** Yksi laskutyypeistä. Murtolukulaskuissa joko itse lasku tai sen lopputulos (tai molemmat) sisältävät murtoluvun.

**Normaali lasku:** Ylläpitäjä valitsee laskulle vaikeustason (helppo tai normaali) syöttäessään uutta harjoitteluosion laskua ylläpitosivun kautta. Normaalilla laskulla tarkoitetaan laskua, joka on vastaava kuin testiosiossa. Normaalit laskut ovat aina sanallisia.

**Ohjelman osio:** Ohjelma sisältää neljä erillistä osiota: teoriaosio, harjoitteluosio, testiosio ja ylläpito-osio.

**Ohjelman pääsivu:** Sivun, joka avautuu, kun ohjelma käynnistyy. Pääsivulta on pääsy kaikkiin ohjelman osioihin.

**Oppimateriaaliosio:** Ks. teoriaosio.

**Prosenttilasku:** Yksi laskutyypeistä. Prosenttilaskuissa joko itse lasku tai sen lopputulos (tai molemmat) sisältävät prosenttiluvun.

**Teoriaosio:** Teoriaosiossa tarkoitetaan ohjelman osaa, jossa käyttäjä voi tutustua lääkelaskujen teoriaan.

**Teoriaosion pääsivu:** Sivun, joka avautuu, kun käyttäjä valitsee ohjelman pääsivulta teoriaosion. Teoriaosion pääsivulta on pääsy kaikkiin teoriadokumentteihin.

**Testiosio = Koeosio:** Testiosiossa tarkoitetaan ohjelman osaa, jossa käyttäjä voi tehdä lääkelaskutestin.

**Verrantolasku:** Yksi laskutyypeistä. Verrantolaskut muodostuvat suoraan ja kääntäen verrannollisuuksista.

**Ylläpito-näyttö, ylläpidon pääsivu:** Sivun, joka aukeaa, kun ylläpitäjä on valinnut ylläpito-osion ja syöttänyt salasanan oikein. Tältä sivulta ylläpitäjä voi valita haluamansa ylläpito-toiminnon.

**Ylläpito-osio:** Ylläpito-osiolla tarkoitetaan ohjelman osaa, jossa ylläpitäjä voi suorittaa ylläpito-toimia (salasanan vaihto, uusien laskujen lisääminen, vanhojen laskujen muokkaaminen ja poistaminen, läpipääsytodistuksen tulostaminen, yleiset ylläpito-toimet).

## 3 Lopputuotteen esittely

Ohjelma sisältää neljä erillistä osiota: teoriaosio, harjoitteluosio, testiosio ja ylläpito-osio. Ohjelman teoriaosio ja harjoitteluosio ovat toimintoja, joiden avulla käyttäjä voi valmistautua testiosion suorittamiseen. Testiosio suoritetaan valvotuissa oloissa ja sen läpipääsy on edellytys lääkeluvun uusimiselle. Ylläpito-osioista ohjelman pääkäyttäjät pystyvät lisäämään, muokkaamaan ja poistamaan laskutehtäviä sekä suorittamaan muita yleisiä ylläpitotehtäviä.

### 3.1 Teoriaosio

Teoriaosiossa käyttäjä voi tutustua lääkelaskujen taustalla olevaan teoriaan ennen harjoittelun ja testin suorittamista. Teoriaosio on käyttäjien vapaassa käytössä eikä se vaadi erillistä sisäänkirjautumistunnusta. Teoriaosio on jaettu neljään eri osa-alueeseen laskutyypin mukaisesti: prosenttilaskut, desimaalilaskut, murtoluvut ja verrannot. Käyttäjä voi vapaasti valita, minkä laskutyypin teoriaosioon hän haluaa tutustua. Ohjelma avaa kyseisen dokumentin ohjelman yhteyteen. Käyttäjällä on mahdollisuus tutustua teoriaosioon myös kesken harjoitteluosion suoritusta.

### 3.2 Harjoitteluosio

Läkelaskujen harjoitteluosio on käyttäjien vapaassa käytössä, eikä se vaadi erillistä sisäänkirjautumistunnusta. Harjoitteluosion avulla käyttäjä voi halutessaan harjoitella laskuja ennen varsinaisen testin suorittamista. Harjoittelusta ei talleteta mitään tietoa käyttäjästä tai tämän vastauksista. Harjoitteluosio on jaettu neljään eri osa-alueeseen: murtoluvut, desimaaliluvut, verrannot ja yksikkömuunnokset. Käyttäjä voi vapaasti valita, mitä osa-aluetta ja missä järjestyksessä hän haluaa harjoitella. Käyttäjä voi myös harjoitella yhtä osa-aluetta useampaan kertaan.

Harjoitteluosiossa ohjelma arpoo sen muistiin talletetuista kysymyksistä käyttäjälle laskutehtäviä. Käyttäjän tehtävänä on laskea lasku esimerkiksi kynää ja paperia apuna käyttäen ja syöttää tämän jälkeen vastaus ohjelmalle tarkistettavaksi. Harjoitteluosio sisältää laskuja, jotka ovat tyypiltään vastaavia kuin testiosiossakin. Osa harjoitteluosion laskuista on myös astetta helpompia laskuja kuin testissä, ja niiden tarkoituksena on lähinnä harjoittaa käyttäjän mekaanista laskutaitoa. Käyttäjän syötettyä vastauksen tiettyyn laskuun ohjelma ilmoittaa, oliko vastaus oikein vai väärin. Mikäli vastaus oli oikea, tulostaa ohjelma näytölle seuraavan laskutoimituksen. Jos vastaus oli väärä, avustaa ohjelma käyttäjää laskun laskemisessa ja tulostaa ruudulle ensimmäisen vihjeen siitä miten lasku tulisi laskea. Mikäli käyttäjä vastaa uudelleen tehtävään väärin, tulostaa ohjelma ruudulle toisen vihjeen. Tämän jälkeen ohjelma ei enää tulosta lisää vihjeitä, mutta kuvaruudulle aktivoituu Ohita kysymys -painike, jota painamalla käyttäjä voi ohittaa laskun. Lisäksi ruudulle aktivoituu Näytä vastaus -painike, joka näyttää käyttäjälle oikean laskukaavan ja vastauksen. Käyttäjä voi myös halutessaan lopettaa harjoitteluosion ennen kaikkiin kysymyksiin vastaamista. Lisäksi ohjelma antaa käyttäjälle mahdollisuuden tutustua teoriaosioon, jos

tämä kokee sen tarpeelliseksi.

Yksi harjoittelujakso kestää niin kauan, kunnes käyttäjä on vastannut oikein viiteen (oletusarvo) kyseisen osa-alueen tehtävään ilman vihjeiden tai teoriaosion hyödyntämistä. Tämän jälkeen ohjelma ilmoittaa käyttäjälle kuinka moneen kysymykseen hän vastasi kokonaisuudessaan sekä antaa suosituksen siitä, että pitäisikö käyttäjän harjoitella kyseisen osion laskuja vielä lisää. Tulos sekä suositus tulostuvat ohjelman pääsivulle kyseisen laskutyypin perään ja ne säilyvät siinä koko ohjelman suorituksen ajan. Mikäli käyttäjä suorittaa saman laskutyypin uudestaan, tulos ja suositus päivittyvät vastaamaan viimeisimmän suorituskerran tulosta.

### 3.3 Testiosio

Testiosio suoritetaan aina sovittuna aikana valvotuissa oloissa koulun tietokoneluokassa. Käyttäjän ei siis tule pystyä suorittamaan testiä muuna aikana omatoimisesti. Testiosioon kirjaututaankin aina tunnuksella, joka on vain valvojien tiedossa. Ennen testiä valvoja avaa testiohjelman sisäänkirjautumistunnuksella.

Testiosio koostuu 10 laskusta, joihin kaikkiin on vastattava. Kaikki kymmenen laskua tulevat kerralla samalle näytölle ja käyttäjä voi suorittaa ne haluamassaan järjestyksessä. Kun käyttäjä on vastannut kaikkiin kymmeneen laskuun, hän lähettää vastauksensa, jolloin ohjelma tarkistaa vastausten oikeellisuuden. Käyttäjä pääsee testin läpi, mikäli hän vastaa kaikkiin laskuihin oikein. Tässä tapauksessa ohjelma antaa käyttäjän tulostaa läpipääsytodistuksen, joka toimitetaan valvojalle. Ennen tulostusta ohjelma tiedustelee käyttäjältä hänen sosiaaliturvatunnuksensa sekä etu- ja sukunimen. Nämä tulostuvat osaksi läpipääsytodistusta. Mikäli käyttäjä ei saa kaikkia laskuja oikein, ilmoittaa ohjelma hänelle missä kohtaa virheet tapahtuivat ja miten lasku olisi pitänyt laskea (laskukaava ja vastaus). Tässä tapauksessa ohjelma ei anna käyttäjän tulostaa läpipääsytodistusta.

### 3.4 Ylläpito-osio

Ylläpito-osioon kirjaututaan aina ohjelman pääsivulta salasanalla, joka on vain testin valvojan ja ohjelman ylläpitäjän tiedossa.

Ylläpito-osioon kuuluu toiminto, jonka avulla voidaan lisätä, muokata ja poistaa laskuja. Syötetyt tiedot tallettavat automaattisesti, kun käyttäjä klikkaa hiirellä jotain toista kysymysriviä, joten käyttäjän ei tarvitse erikseen tallettaa tietoja. Sama pätee myös vanhoja laskuja muokattaessa. Jotta tiedot tallettavat, käyttäjän on kuitenkin muistettava käydä klikkaamassa jotain kysymystä ylläolevasta kysymyslistasta. Haluttaessa tietty kysymys voidaan muuttaa harjoittelukysymyksestä testikysymykseksi (tai päin vastoin), painamalla Siirrä testiin -painiketta (Siirrä harjoituksiin -painiketta). Laskuista talletetaan aina seuraavat tiedot:

1. **Laskutyyppi:** *Prosentti-/desimaali-/murto-/verrantolasku* -> valitaan yksi näistä ja tieto talletetaan, jotta ohjelma tietää mihin laskutyyppiin lisättävä lasku kuuluu (pakollinen tieto)

2. **Laskun vaativuus (Huom. vain harjoitteluosion laskuille):** *Helppo/normaali* -> valitaan, onko lasku helppo (eli ns. lämmittelylasku) vai normaali (testiosion laskuja vastaava). Ohjelma valitsee harjoitteluosiossa aluksi helppoja laskuja, ja kun käyttäjä on vastannut oikein yhteen helppoon tehtävään, niin ohjelma valitsee loput laskut normaaleista laskuista (pakollinen tieto harjoitteluosion laskuille)
3. **Laskukaava:** Ylläpitäjä syöttää tähän kenttään laskun laskukaavan, joka ilmoitetaan testiosion päätteeksi, mikäli käyttäjä ei osaa laskea jotain laskua oikein. Myös harjoitteluosiossa toisen väärän vastauksen jälkeen käyttäjä voi pyytää ohjelmaa näyttämään oikean laskukaavan. Laskukaava sisältää sekä kaavan että vastauksen (esim.  $2+2=4$ ). (pakollinen tieto)
4. **Yksikkö:** Ylläpitäjä syöttää tähän kenttään vastauksen yksikön (esim. litra, desilitra, gramma jne.), joka tulostuu sekä harjoittelu- että testiosiossa vastauskentän perään. Tällä tavalla kerrotaan käyttäjälle, missä yksikössä hänen odotetaan antavan vastauksensa. Syöte on tekstimuotoinen ja kentän maksimipituus on 20 merkkiä. (ei pakollinen)
5. **Vastaus:** Ylläpitäjä syöttää tähän kenttään laskun numeerisen vastauksen, johon ohjelma vertaa käyttäjän vastausta. (pakollinen tieto)
6. **Lasku:** Ylläpitäjä syöttää tähän kenttään laskun sanallisen kuvauksen, joka tulostuu kuvaruudulle. Syöte on tekstimuotoinen. (pakollinen tieto)
7. **Vihje 1:** Ylläpitäjä syöttää tähän kenttään laskuun liittyvän vihjeen, joka tulostuu kuvaruudulle, kun käyttäjä on vastannut harjoitteluosiossa ensimmäisen kerran laskuun väärin, tai käyttäjä on pyytänyt ohjelmaa antamaan ensimmäisen vihjeen. (ei pakollinen)
8. **Vihje 2:** Ylläpitäjä syöttää tähän kenttään laskuun liittyvän vihjeen, joka tulostuu kuvaruudulle, kun käyttäjä on vastannut harjoitteluosiossa toisen kerran laskuun väärin tai, käyttäjä on pyytänyt ohjelmaa antamaan toisen vihjeen. (ei pakollinen)

Lisäksi ylläpito-osioon kuuluu joukko yleisiä toimintoja, jotka on esitelty seuraavaksi.

1. **Tulostusasetukset:** Ohjelma antaa käyttäjän tulostaa läpipääsytodistuksen mikäli hän pääsee testin läpi. Tulostusasetuksista ylläpitäjä voi vaihtaa oletustulostinta, johon läpipääsytodistukset oletusarvoisesti tulostuvat. Lisäksi ylläpitäjä voi asettaa ja muuttaa läpipääsytodisteiden tulosteiden lukumäärää.
2. **Tiedostojen sijainti:** Ohjelma tallettaa harjoittelu- ja testiosion kysymykset ennalta määrättyyn paikkaan tietokoneelle tai tiedostopalvelimelle. Ylläpitäjä voi vaihtaa harjoittelu- ja testiosoiden laskujen ja teoriadokumenttien fyysistä sijaintipaikkaa Karstulan tiedostopalvelimella tai tietyllä koneella.
3. **Harjoitteluosion oikeiden vastausten lukumäärä:** Jotta käyttäjä pääsee harjoitteluosion läpi, häneltä vaaditaan tietty määrä oikeita vastauksia. Ylläpitäjä voi vaihtaa harjoitteluosion läpipääsyyn vaadittavien oikeiden vastausten lukumäärää. Oletusarvoisesti oikeita vastauksia vaaditaan viisi.



4. **Testin kysymysten jakauman ja lukumäärän muuttaminen:** Ohjelman ylläpitäjällä on mahdollisuus säätää testissä olevien eri laskutyyppeiden suhdetta. Ylläpitäjä voi esimerkiksi päättää, että testissä ei ole yhtään verrantolaskua ja ne korvataan murtolaskuilla. Testin laskujen yhteismäärä määräytyy näiden summana, eikä sitä ole rajoitettu. On kuitenkin tärkeätä aina varmistaa, että tietokantaan on talletettuna tarvittava määrä kysymyksiä. Mikäli näin ei ole, ohjelma ilmoittaa tästä ylläpitäjälle.

## 4 Dokumenttien tiivistelmät

### 4.1 Projektisuunnitelma (sisäinen)

Projektisuunnitelma määrittelee reunaehdot sekä ajan että henkilöressurssien käytölle. Dokumenttia pidettiin yllä koko projektin ajan. Sen avulla varmistettiin, että projekti pysyy aikataulussa. Projektisuunnitelmassa määritetään toteutettava ohjelmisto yleisellä tasolla, missä ajassa ohjelmisto toteutetaan sekä projektiin varatut resurssit. Henkilöressusseista on määritetty lisäksi vastuualueet. Lisäksi dokumentissa on esiteltyä projektille määritetyt riskit sekä niiden todennäköisyys ja haitallisuus. Kokoarviot on laskettu toimintopisteiden avulla ja työmääräarvio on tehty COCOMO II -mallin avulla.

Projektisuunnitelman sisältö on seuraavanlainen:

1. **Johdanto**
2. **Sanasto**
3. **Projektiorganisaatio**
4. **Riskianalyysi**
5. **Projektin laitteisto- ja ohjelmistovaatimukset**
6. **Koko- ja kustannusarviot**
7. **Työn ositus**
8. **Projektin aikataulu**
9. **Seuranta- ja raportointimenetelmät**

### 4.2 Vaatimusmäärittelydokumentti (ulkoinen)

Vaatimusdokumentti on asiakkaan ja projektiryhmän välinen sopimus siitä, mitä projekti tulee tuottamaan. Lisäksi se toimii lähtökohtana suunnittelulle.

Vaatimusmäärittelydokumentissa on esitelty ohjelman toteutettavat ominaisuudet ja toiminnallisuus. Vaatimukset on esitetty käyttäjävaatimuksien ja järjestelmävaatimuksien

avulla. Käyttäjän vaatimuksissa toiminnallisuus on esitetty yleisinä vaatimuksina ja järjestelmävaatimuksissa tarkkoina toiminnallisten ja ei-toiminnallisten vaatimusten kuvauksina. Dokumentissa on myös käyty läpi ohjelman asettamat vaatimukset käyttöympäristölle sekä esitelty mahdolliset virhetilanteet ja ohjelman toiminta virhetilanteiden syntyessä. Ohjelman toimintaa on selkeytetty käyttötapauskaavioiden, käyttötapauksen ja vuokaavioiden avulla. Dokumentin liitteenä on käyttöliittymän prototyyppi.

Vaatimusmäärittelydokumentin sisältö on seuraavanlainen:

1. **Esipuhe**
2. **Johdanto**
3. **Käyttäjän vaatimukset**
4. **Järjestelmävaatimukset**
5. **Ympäristövaatimukset**
6. **Järjestelmäarkkitehtuuri**
7. **Järjestelmämallit**
8. **Järjestelmän kehitys**
9. **Liitteet: Käyttöliittymäprotokuvat**

### **4.3 Suunnitteludokumentti (sisäinen)**

Suunnitteludokumentti kuvaa toteutettavan järjestelmän sellaisella tasolla, että toteutus on tehtävissä suoraan sen pohjalta. Dokumentissa on käyty läpi ohjelman arkkitehtuuri yleisellä tasolla sekä esitelty käytetyt suunnittelumallit. Myös poikkeusten käsittely on määriteltä dokumentissa. Suunnitteludokumentissa ohjelman tekninen kuvaus on toteutettu tarkalla tasolla. Suunnitteludokumentista löytyy ohjelman luokkakaavio, sekvenssikaavio, luokkien rajapintakuvaukset sekä metodien tarkat kuvaukset ja määritelmät.

Suunnitteludokumentin sisältö on seuraavanlainen.

1. **Johdanto**
2. **Toteutusympäristö**
3. **Arkkitehtuuri**
4. **Poikkeukset**
5. **Käyttöliittymä**
6. **Logiikkajärjestelmä**
7. **Tietokanta**

## 4.4 Testaussuunnitelma (sisäinen)

Testaussuunnitelma sisältää tiedot siitä, miten toteutettavan järjestelmän testaus tehdään. Suunnitelmassa on käyty läpi projektissa käytetyt työkalut sekä menetelmät. Testaussuunnitelma on tehty erikseen yksikkötestausta, integrointitestausta ja järjestelmätestausta varten. Lopuksi dokumentissa on esitelty menetelmät ja kriteerit vaatimusten validointia ja testien hyväksymistä varten. Myös testausaikataulu on esitettyä suunnitelmassa.

Testaussuunnitelman sisältö on seuraavanlainen:

1. **Johdanto**
2. **Sanasto**
3. **Resurssit**
4. **Yksikkötestaus**
5. **Integrointitestausta**
6. **Järjestelmätestaus**
7. **Muu testaus**
8. **Testausaikataulu**

## 4.5 Käyttöohje (ulkoinen)

Käyttöohjeessa ohjelman toiminta on esitettyä siten, että ohjelman loppukäyttäjä voi sen perusteella oppia käyttämään järjestelmää täydellisesti ilman ulkopuolista apua. Käyttöohjeessa neuvotaan kuinka ohjelma voidaan asentaa ja kuinka se käynnistetään tämän jälkeen. Lisäksi ohjelman käyttöä on havainnollistettu kuvaamalla ohjelman tarjoamia toimintoja ja havainnollistamalla niitä kuvien avulla.

Käyttöohjeen sisältö on seuraavanlainen:

1. **Johdanto**
2. **Sanasto**
3. **Ohjelman käyttäjäryhmät**
4. **Ohjelman asennus- ja käynnistysohje**
5. **Toiminnot**
6. **Kysymyksiin vastaaminen ja käyttäjän syötteiden muoto**

## 4.6 Ylläpitodokumentti (ulkoinen)

Ylläpitodokumentti käsittelee sellaisia asioita, joita mahdolliset jatkoryhmät tai asiakkaan edustajat tarvitsevat muokataksaan ohjelmaa. Dokumenttia ei ole tarkoitettu luettavaksi yksinään, vaan yhdessä suunnitteludokumentin ja ohjelmakoodin kanssa. Näin ollen dokumentista on viitteitä muihin dokumentteihin. Ylläpitodokumentissa on käyttöohjeen tavoin esitettyä ohjelman asennusohje. Lisäksi dokumentissa on esitettyinä ne vaatimukset ja suunnitelman osat, jotka muuttuivat projektin aikana. Myös toteutumattomat ohjelman osat on esiteltyinä. Lopuksi dokumentissa esitellään erinäisiä ylläpitoon liittyviä seikkoja.

Ylläpitodokumentin sisältö on seuraavanlainen:

1. **Johdanto**
2. **Sanasto**
3. **Asennusohje**
4. **Jäädytyksen jälkeen muuttuneet vaatimukset**
5. **Jäädytyksen jälkeen muuttuneet suunnitelman osat**
6. **Toteutumattomat vaatimukset ja suunnitelman osat**
7. **Ylläpitoon liittyvät seikat**

## 5 Projektin päättöanalyysi

### 5.1 Asiakkaan erityispiirteet

Muista kevään 2007 ohjelmistotuotantoprojekteista poiketen asiakkaamme oli Tietojenkäsittelytieteen laitoksen ulkopuolelta. Tämä toi osaltaan projektiimme omia haasteita, sillä pitkästi välimatkasta johtuen (Karstula-Helsinki) asiakkaan kasvottainen tapaaminen ei ollut mahdollista montaa kertaa. Lisäksi yhteyshenkilönämme toimi Karstulan Evanke-lisen Kansanopiston opettaja, jolla ei ollut tietoteknistä erityisosaamista.

Heti projektin alussa ryhmämme oli sähköposti ja puhelin yhteydessä asiakkaaseen ja kar-toitti ohjelman käyttötarkoitusta ja yleisiä vaatimuksia. Näiden vaatimusten pohjalta pro-jektiryhmämme pystyi suunnittelemaan ja ideoimaan ohjelman toimintoja sekä tekemään käyttöliittymästä hahmotelman. Sitä muka kun ryhmällemme tuli kysymyksiä ohjelmaan liittyen, niin olimme yhteydessä asiakkaaseen ja sovimme kuinka tietyt toiminnot olisi hyvä hoitaa.

Virallinen asiakastapaaminen järjestettiin 29.1.2007, jolloin asiakas oli varannut vaati-musmäärittelyä varten koko päivän. Ennen tapaamista olimme laatineet käyttöliittymästä prototyypin PowerPointilla, joten pystyimme tämän avulla käymään läpi ohjelman toi-minnallisuutta sekä vaatimuksia. Tämä lähestymistapa osoittautui hyvin käyttökelpoisek-si, sillä näin asiakkaan oli helpompi hahmottaa ohjelman suunniteltua toimintalogiikkaa.

Toisaalta ryhmämme oli helpompi esittää suunnitelmia konkreettisen esimerkin avulla. Päivän aikana saimme hyvin käytyä läpi ohjelman toteutettavat toiminnot ja molemmille osapuolille jäi selkeä kuva toteutettavasta ohjelmasta.

Myöhemmin projektin aikana emme joutuneet enää tekemään vaatimuksiin suurempia muutoksia. Muutokset koskivat lähinnä ohjelman yksityiskohtia kuten syötteiden muotoa. Asiakkaan puolelta ohjelmaan ei tullut enää projektin kuluessa lisävaatimuksia ja kaikki toteutettavat lisäominaisuudet otettiin mukaan suunnitelmiin projektiryhmän toimesta.

Toinen epävirallinen asiakastapaaminen järjestettiin 30.3.2007, jolloin määrittelyvastuumme kävi tapaamassa asiakasta Helsingin messukeskuksessa. Tapaamisessa asiakkaalle esiteltiin lopullinen käyttöliittymän prototyyppi ja käytiin vielä läpi ohjelman toiminnallisuus sen perusteella.

Etätyöskentelystä huolimatta yhteydenpito asiakkaaseen oli vaivatonta ja ongelmattonta koko projektin ajan. Kysymyksiä tuli puolin ja toisin projektin kuluessa, mutta ne pystyttiin hyvin ratkomaan sähköpostin ja puhelimen avulla. Myöskään asiakkaan tietoteknisen erityisosaamisen puute ei millään tavalla haitannut projektin etenemistä. Päinvastoin voidaan ajatella, että ryhmämme sai täysin vapaat kädet itse toteutuksen kannalta ja asiakas esitti ainoastaan ne ominaisuudet, mitä ohjelmalta vaaditaan. Ainoana negatiivisina puolena asiakkaaseen yhteydenpidossa oli, että ryhmämme ei koskaan saanut selville olisiko Karstulan tietotekninen ympäristö ollut soveltuva myös www-pohjaisen ohjelmiston toteuttamiseen. Yhteyshenkilömme ei osannut vastata tähän ja hänen antamansa henkilö ei koskaan vastannut tiedusteluihimme. Odottelimme vastausta parin viikon ajan ja päätimme tämän jälkeen toteuttaa ohjelman yksittäisasennuksena jokaiselle KEK:n koneelle. Lisäksi joitain projektiin liittyviä dokumentteja saimme odotella asiakkaan päästä, sillä he eivät olleet osanneet varautua niiden tekemiseen. Tämä ei kuitenkaan merkittävästi vaikuttanut projektin etenemiseen.

## 5.2 Käytetty prosessimalli

Päätimme ohjelmistuotantoprojektimme aluksi käytettävästä prosessimallista ja päädyimme yksimielisesti käyttämään vesiputousmallia. Vesiputousmalli oli kaikille tuttu ja se tuntui mielekkäimmältä valinnalta myös tähän projektiin. Tämän prosessimallin riskinä kuitenkin nähtiin, että se ei välttämättä toimi halutulla tavalla, jos asiakas muuttaa vaatimuksia koko ajan projektin aikana. Tämän riskin ehkäisemiseksi asiakkaalle pyrittiin korostamaan, että olisikin hyvin tärkeää saada vaatimusmäärittely tehtyä heti projektin alussa, niin että niihin ei tulisi suuria muutoksia projektin kuluessa. Projektiryhmämme onnistuikin hyvin toteuttamaan vaatimusmäärittelylle varatun asiakastapaamisen ja kartoittamaan näin ohjelman toiminnallisuuden ja vaatimukset. Projektin aikana vaatimuksia muutettiin vain hieman ja kyse oli enemmänkin hienosäädöstä kuin radikaaleista muutoksista. Nämäkin muutokset tehtiin lähes kokonaan projektiryhmän aloitteesta.

Myös prosessin muiden vaiheiden osalta mallia pystyttiin noudattamaan tarkasti. Vaatimusten selkiytyttyä aloitettiin suunnittelu. Kun suunnitteludokumentti oli saatu valmiiksi, suoritettiin katselmointi, jonka perusteella suunnitelmaa muutettiin ryhmän palautteen perusteella. Itse toteutus aloitettiin vasta, kun suunnittelu oli saatu ryhmän mielestä val-

miiksi ja toimivaksi. Toteutuksen aikana suunnitelmaa muutettiin havaittujen puutteiden tai ongelmien mukaan. Suuria muutoksia ei jouduttu tekemään projektin kuluessa. Testaus hoidettiin osittain päällekkäin toteutuksen kanssa siten, että valmistuneita luokkia yksikkötestattiin niiden valmistuttua. Integraatiotestaus tehtiin koko ohjelmalle sen valmistuttua. Lopuksi ohjelma ja tarvittava dokumentaatio luovutettiin asiakkaalle CD:llä. Projektiryhmä ei osallistunut loppuasennukseen asiakkaan tiloissa.

### **5.3 Projektiryhmä ja ryhmän työnjako**

Projektiryhmämme koostui kuudesta tietojenkäsittelytieteen opiskelijasta. Kaikki ryhmäläisistämme olivat suorittaneet perus- ja aineopinnot jo lähes kokonaan. Osa ryhmäläisistämme oli lisäksi suorittanut jo syventäviäkin opintoja ja olivat opintojen loppuvaiheessa. Myös kokemus ohjelmistotuotannosta vaihteli ryhmämme sisällä. Kokeneimmilla jäsenillä oli usean vuoden työkokemus ohjelmistotuotannosta, kun taas kokemattomimmilla taustana oli ainoastaan koulun peruskurssit. Yleisesti ottaen ryhmästä löytyi kuitenkin vahvaa kokemusta ja osaamista erityisesti suunnitteluun ja toteutukseen liittyen.

Kullekin ryhmän jäsenelle määrättiin vastuualue projektin alussa. Vastuualueet olivat projektipäällikkö, vaatimusmäärittelyvastaava, suunnitteluvastaava, koodausvastaava, testi-vastaava ja dokumentointivastaava. Työnjako suoritettiin tiedustelemalla kunkin jäsenen halukkuutta ottaa vastuu tietyistä osa-alueista. Suurin kiinnostus oli koodausvastaavan rooliin. Sen sijaan projektipäällikön, määrittelyvastaavan ja testausvastaavan roolit saatiin jaettua suoraan, sillä niihin oli vain yksi halukas kuhunkin. Suunnittelu, koodaus ja dokumentointi arvottiin lopulta kolmen jäsenen kesken.

Projektin aikana tehtäviä jaettiin osaamiseen ja halukkuuteen perustuen. Ne henkilöt, jotka olivat halukkaita koodaajia saivat keskittyä ohjelmointiin, kun taas vähemmän koodausintoiset pystyivät osallistumaan enemmän esimerkiksi dokumentointiin.

### **5.4 Ryhmän työskentelytavat**

Projektiryhmämme kokoontui koko kurssin ajan säännöllisesti kaksi kertaa viikossa. Viikoittaiset tapaamisajat sovittiin yhdessä kurssin alussa siten, että ne kävivät kaikille ryhmän jäsenille. Yksi tapaaminen oli kestoaltaan aina noin 1,5 tuntia. Projektipäällikkö lähetti muille ryhmäläisille aina ennen kokousta asialistan, jonka avulla käsiteltävät asiat olivat kaikille jo etukäteen selvillä. Kokouksissa käsiteltiin asioita myös asialistan ulkopuolelta aina tarpeen tullen. Yleisimmin kokouksissa käytiin läpi toteutuksen tilaa ja kunkin ryhmäläisen työtehtäviä. Myös vastaan tulleita ongelmia pyrittiin yhdessä ratkaisemaan kokousten yhteydessä. Lisäksi viikoittain seurattiin projektin alussa määriteltyjä riskejä ja niiden tilaa.

Peruskokousten lisäksi kurssin aikana pidettiin muutamia ns. erityistapaamisia kuten esimerkiksi vaatimusmäärittely- ja suunnitteludokumentin katselmointitilaisuudet. Joistakin asioista pidettiin yhteisiä ryhmäläisten järjestämiä koulutustilaisuuksia. Näissä koulutuksissa joku ryhmän jäsenistä oli tutustunut tiettyyn asiaan ja esitteli sen muille. Tällaisia

tilaisuuksia pidettiin esimerkiksi CVS:n käytöstä ja testauksesta.

Pääosin ryhmämme jäsenet työskentelivät itsenäisesti ja kommunikointi hoitui sähköpostin välityksellä. Lisäksi kokouksissa käsiteltiin joitain ongelmakohtia, joiden ratkaiseminen olisi ollut sähköisesti työlästä. Tämä toimintapa onnistui pääosin hyvin, sillä jokaiselle oli määritetty selkeät tehtävät kokouksissa, joten lähtökohtaisesti kaikille oli selvillä mitä tehtäviä seuraavaan kokoukseen mennessä pitäisi tehdä ja mitä pitäisi olla valmiina. Viikoittaisten kokouksien lisäksi kokoonnuimme vain kahtena erillisenä kertana yhdessä. Molemmat kokoontumiskerrat olivat suunnitteluvaiheessa, sillä silloin katsoimme, että suunnittelu kannattaa tehdä yhteistyönä.

Valmiit tuotokset laitettiin ryhmän käytössä olevaan hakemistoon, johon kaikilla oli käyttöoikeudet. Versionhallinnassa hyödynnettiin CVS:ää. Lisäksi jotain tuotoksia säilytettiin ryhmäläisten henkilökohtaisissa kansioissa, johon myös muulla ryhmällä oli pääsy. Tällä tavoin pyrittiin varmistamaan, että mikäli joku ryhmäläinen lopettaa projektin, niin hänen siihenastiset tuotokset ovat muiden saatavilla.

## **5.5 Projektin onnistuminen**

Yleisesti ottaen projektimme onnistui hyvin. Ohjelma saatiin toteutettua vaatimusmäärittelydokumentin mukaisesti ennalta määrätyn aikataulun puitteissa. Projektin aikana olimme joissakin kohdissa aikataulusta jäljessä, mutta saimme kurottua eron lopussa kiinni. Missään vaiheessa aikatauluero ei kuitenkaan ollut vakava. Myös ryhmätyö onnistui hyvin, vaikka kommunikaatio rajoittuikin pääosin viikottaisiin kokouksiin ja sähköpostiin. Jokainen ryhmämme jäsen osallistui kutakuinkin yhtä suurella työpanoksella projektiin ja jokainen hoiti oman työosuutensa mallikkaasti.