

# Lohtu-projekti

## Suunnitteludokumentti

Versiohistoria:

1.0	13.2.2003	Käyttöliittymäsuunnitelma	Virve
1.1	14.2.2003	Tietokantasuunnitelma	Andreas
1.2	18.2.2003	17.2.2003 katselmoinnissa tulleet muutokset	Andreas
1.3	18.2.2003	17.2.2003 katselmoinnissa tulleet muutokset	Virve
1.4	19.2.2003	Tietokantatauluja luotaessa huomattavat virheet korjattu	Andreas
1.5	24.2.2003	Sovelluslogiikka	Andreas
2.0	27.2.2003	26.2.2003 katselmoinnissa tulleet muutokset	Andreas
2.1	9.3.2003	Toisen syklin 1. toteutusviikon ominaisuuksia	Kimmo
2.2	14.3.2003	10.3.2003 katselmoinnissa tulleet muutokset	Kimmo
2.3	17.3.2003	17.3.2003 katselmoinnissa tulleet muutokset	Kimmo

Helsinki 19. helmikuuta 2003

Kimmo Airamaa, Andreas Asuja, Mari Muuronen, Seppo Pastila, Virve Taivaljärvi

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Laudatur-projektityö, kevät 2003

# Sisältö

<b>1. KÄYTTÖLIITTYMÄSUUNNITELMA.....</b>	<b>1</b>
1.1. YLEISIÄ PERIAATTEITA .....	1
1.2. KÄYTTÖTAPAUKSET.....	1
1.2.1. Yleistä.....	1
1.2.2. Käyttötapaus 1: Projektien ja opiskelijoiden lisäys.....	1
1.2.3. Käyttötapaus 2: Projektin suunnittelu .....	2
1.3. JÄRJESTELMÄN SISÄÄNKÄYNTI .....	3
1.4. JÄRJESTELMÄN SIVUSTON VALIKKORAKENNE.....	5
1.5. NÄYTTÖKUVAUKSET .....	5
1.5.1. Projektin perustiedot.....	5
1.5.2. Syklin suunnittelu.....	6
1.5.3. Projekti- ja opiskelijatietojen haku Kurssijärjestelmästä.....	8
<b>2. TIETOKANNAN KUVAUS .....</b>	<b>9</b>
2.1. TAULU SOFTWARETYPES .....	9
2.2. TAULU PROCESSMODELS .....	9
2.3. TAULU DATATYPES .....	9
2.4. TAULU MANDATORYQUANTITIES .....	12
2.5. TAULU PROJECTQUANTITIES .....	12
2.6. TAULU QUANTITYSTAGES .....	12
2.7. TAULU ROLES .....	13
2.8. TAULU PROJECTS .....	13
2.9. TAULU USERS .....	13
2.10. TAULU COMPONENTS.....	14
2.11. TAULU PHASES .....	14
2.12. TAULU TASKS .....	14
2.13. TAULU USERSTASKS.....	15
2.14. TAULU TASKQUANTITIES .....	15
2.15. TAULU MEASURES .....	16
<b>3. SOVELLUSLOGIIKKA.....</b>	<b>18</b>
3.1. YLEINEN ARKKITEHTUURI.....	18

3.2.	PAKETTIJAKO.....	19
3.3.	PAKETTI FI.HELSENKI.CS.LOHTU.UTIL.....	20
3.3.1.	<i>fi.helsinki.cs.util.LohtuInit</i> .....	20
3.3.2.	<i>fi.helsinki.cs.util.LohtuResources</i> .....	20
3.4.	PAKETTI FI.HELSENKI.CS.LOHTU.MODEL.....	21
3.4.1.	<i>fi.helsinki.cs.lohtu.model.DAOFactory</i> .....	21
3.4.2.	<i>fi.helsinki.cs.lohtu.model.ProcessModelVO</i> .....	21
3.4.3.	<i>fi.helsinki.cs.lohtu.model.ProcessModelDAO</i> .....	21
3.4.4.	<i>fi.helsinki.cs.lohtu.model.SoftwareTypeVO</i> .....	21
3.4.5.	<i>fi.helsinki.cs.lohtu.model.SoftwareTypeDAO</i> .....	21
3.4.6.	<i>fi.helsinki.cs.lohtu.model.ComponentVO</i> .....	23
3.4.7.	<i>fi.helsinki.cs.lohtu.model.ComponentDAO</i> .....	23
3.4.8.	<i>fi.helsinki.cs.lohtu.model.ProjectVO</i> .....	23
3.4.9.	<i>fi.helsinki.cs.lohtu.model.ProjectDAO</i> .....	23
3.4.10.	<i>fi.helsinki.cs.lohtu.model.UsersVO</i> .....	23
3.4.11.	<i>fi.helsinki.cs.lohtu.model.UsersDAO</i> .....	23
3.4.12.	<i>fi.helsinki.cs.lohtu.model.TaskDAO</i> .....	23
3.4.13.	<i>fi.helsinki.cs.lohtu.model.TaskVO</i> .....	24
3.4.14.	<i>fi.helsinki.cs.lohtu.model.PhaseDAO</i> .....	24
3.4.15.	<i>fi.helsinki.cs.lohtu.model.PhaseVO</i> .....	24
3.4.16.	<i>fi.helsinki.cs.lohtu.model.UsersTaskDAO</i> .....	24
3.4.17.	<i>fi.helsinki.cs.lohtu.model.UsersTaskVO</i> .....	24
3.5.	PAKETTI FI.HELSENKI.CS.LOHTU.VIEW .....	27
3.6.	PAKETTI FI.HELSENKI.CS.LOHTU.UTIL.CONTROL.....	27
3.6.1.	<i>fi.helsinki.cs.lohtu.control.ProjectController</i> .....	28
3.6.2.	<i>fi.helsinki.cs.lohtu.control.AdminController</i> .....	28
3.6.3.	<i>fi.helsinki.cs.lohtu.control.StatsController</i> .....	28
3.6.4.	<i>fi.helsinki.cs.lohtu.control.CycleController</i> .....	28
3.7.	JSP –SIVUT .....	29
3.7.1.	<i>Project</i> .....	29
3.7.2.	<i>Admin</i> .....	29
3.7.3.	<i>Stats</i> .....	29
3.8.	POIKKEUKSET .....	30
3.8.1.	<i>java.lang.IllegalArgumentException</i> .....	30

3.8.2.	<i>java.util.NoSuchElementException</i> .....	30
3.8.3.	<i>fi.helsinki.cs.lohtu.LohdutonException</i> .....	30
3.9.	SEKVENSSIKAAVIOT.....	30

# 1. Käyttöliittymäsuunnitelma

## 1.1. Yleisiä periaatteita

Käyttöliittymäsuunnittelussa on pyritty tuottamaan hyvien käyttöliittymäsuunnitteluperiaatteiden mukainen käyttöliittymä. Käyttöliittymäratkaisuisissa joudutaan kuitenkin tekemään kompromisseja. Ensimmäinen rajoittava tekijä on se, että käyttöliittymä toteutetaan internet-selaimelle ja sen tulisi toimia yleisimmissä selaimissa (Windows Internet Explorer ja Mozilla). Selainten välillä on kuitenkin merkittäviä eroja erityisesti siinä, miten hyvin ne ymmärtävät JavaScript-kieltä, jonka avulla käytettävyyttä voitaisiin monin eri ratkaisuin parantaa. Ja ottaen huomioon myös toisen rajoittavan tekijän eli Lohtu-projektin rajallisen aikataulun ei tässä projektissa ole mahdollista perehtyä kovin tarkasti selainten eroihin tai tehdä käyttöliittymästä esimerkiksi hienoa graafista Java Applet-toteutusta.

Näistä syistä, käyttöliittymän suunnittelun teknisenä perustana pidetään staattisia html-lomakkeita. Parhaamme mukaan rakennamme tälle pohjalle projektien suunnittelua ja seuranta selkeyttävän, yksinkertaisen ja helposti käytettävän käyttöliittymän.

## 1.2. Käyttötapaukset

### 1.2.1. Yleistä

Tähän lukuun kerätään lyhyitä kuvauksia käyttötapauksista, joihin toteutettavat ominaisuudet perustuvat. Näitä kuvauksia täydennetään ja lisätään Lohtu-projektin edetessä kunkin syklin suunnitteluvaiheessa. Varsinaisesti kaikkia käyttötapauksissa esille tulevia ominaisuuksia ei välttämättä toteuteta Lohtu-projektin puitteissa, mutta ne kirjataan kuitenkin tähän, jotta käyttötapauksesta saataisiin oikea kokonaiskäsitys.

### 1.2.2. Käyttötapaus 1: Projektien ja opiskelijoiden lisäys

Lähtötilanne: Lukukausi on alkamassa. Opiskelijat ovat ilmoittautuneet kurssille ja ennakkotietovaatimukset täyttävät on valittu mukaan ja jaettu projekteihin. Tiedot on tallennettu Kurssijärjestelmään ja näistä tiedoista on luotu ja tallennettu XML-dokumentti.

Käyttötilanne: Projekteja ja opiskelijoita lisäävät vain OhTu-vastaavat. He voivat tehdä sen kahdella tavalla, joko siirtämällä tiedot suoraan Kurssijärjestelmästä tai tallettamalla tiedot käsin ylläpito näyttöjen kautta. Jälkimmäistä tapaa käyttävät ne projektivastaavat, jotka eivät käytä Kurssijärjestelmää. Lisäksi mahdollisten ongelmatilanteiden tai muutostarpeiden (esim. alkuperäinen opiskelija ei osallistukaan projektiin vaan hänen tilalleen tulee toinen opiskelija) vuoksi erilliset ylläpitosivut ovat välttämättömät. Varsinainen tietojen siirtäminen Kurssijärjestelmästä on käyttöliittymältään hyvin yksinkertainen, ja se on kuvattu luvussa 1.5.3.

Lopputilanne: Alkavan lukukauden OhTu-projektien ja projektiryhmäläisten tiedot on tallennettu projektinhallinnan työkaluun. Projektiryhmäläiset voivat aloittaa projektinsa suunnittelun. Projekti näkyy työkalussa myös projektien seuranta- ja vertailutoiminnoissa.

### **1.2.3. Käyttötapaus 2: Projektin suunnittelu**

Lähtötilanne: Projekti on avattu ja se sisältää tiedot projektiin kuuluvista opiskelijoista. OhTu-vastaava on myös määritellyt muut hallinnolliset tiedot eli prosessimallit ja ohjelmistotyypit sekä prosessimalleissa seurattavat suureet.

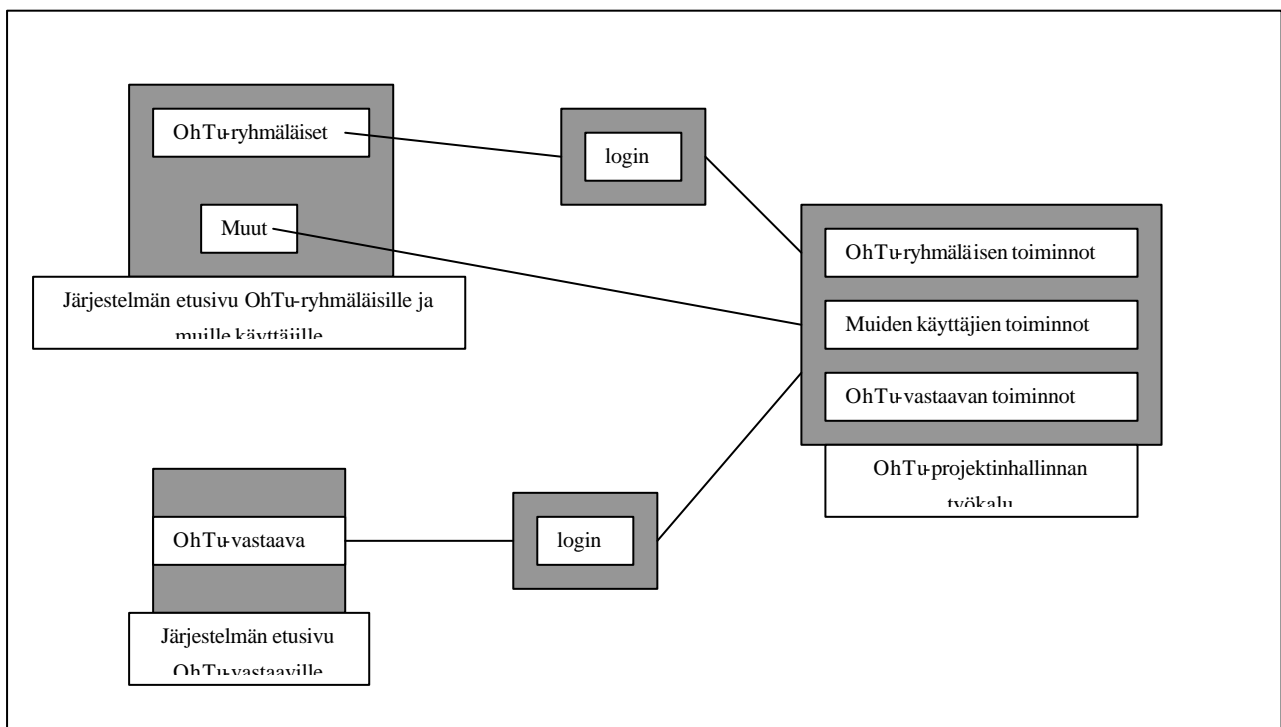
Käyttötilanne: Projektiryhmä aloittaa työkalun käytön projektinsa perustietojen syöttämisellä. Nämä tiedot syötetään pääsääntöisesti vain kerran. Todennäköisesti kuitenkin erityisesti ohjelman ositusta palataan myöhemmin tarkentamaan. Myös muita tietoja voidaan muuttaa. Perustiedoista projektiryhmäläiset siirtyvät suunnittelemaan projektin syklejä. Suunnittelu samoin kuin suunnitelmien jäädyttäminen tehdään pääsääntöisesti sykli kerrallaan. Kuitenkin myös tulevien syklien tietoja halutaan varmasti tallettaa ainakin osittain jo ennen kuin näihin sykleihin varsinaisesti siirrytään. Erityisesti vaihejako voidaan tehdä koko projektin ajalle jo heti aluksi. Tehtäviä suunnitellaan tarkemmin vasta juuri ennen syklin aloittamista. Tietojen syöttämisen järjestys ei ole vakio vaan perustietoihin voidaan palata syklien suunnittelun jälkeenkin ja syklejä voidaan suunnitella eri järjestyksessä. Oleellista on kuitenkin, että syklit on jäädytettävä oikeassa järjestyksessä (ensimmäinen, toinen jne) eivätkä projektiryhmäläiset pääse tallettamaan sykliin toteutumatietoja ennen kuin sykli on jäädytetty. Projektin suunnitteluun liittyviä käyttöliittymän näyttöjä on kuvattu luvuissa 1.5.1 ja 1.5.2.

Lopputilanne: Projekti on suunniteltu ainakin alkavan syklin osalta ja suunnitelmat on jäädytetty. Projektin jäsenet voivat aloittaa toteutumätietojen tallettamisen työkaluun ja seurata suunnitelmien toteutumista.

### 1.3. Järjestelmän sisäänkäynti

Yleiskuva järjestelmän sisäänkäynnistä on esitetty kuvassa 1. Varsinainen järjestelmän etusivu on tarkoitettu OhTu-ryhmäläisille sekä muille mahdollisille käyttäjille. Etusivulle kirjoitetaan yleistä esittelytietoa järjestelmästä sekä ohjeistusta. OhTu-ryhmäläiset tunnistetaan sisäänkirjautumisen avulla, jotta heille voidaan tarjota mahdollisuus ylläpitää oman projektinsa tietoja. Muita käyttäjiä ei tunnisteta vaan he pääsevät etusivulta suoraan toimintoihinsa rajattuun järjestelmään. Etusivun hahmotelma on esitetty kuvassa 2 ja sisäänkirjautumissivu kuvassa 3.

OhTu-vastaaville on täysin oma etusivunsa järjestelmään. Heidän sisäänkäyntinsä piilotetaan omalle sivulleen, jotta muilla käyttäjillä ei synny turhaan kiusausta koettaa onneaan päästä käsiksi hallinnollisiin toimiin ja jotta pienennetään riskiä, että joku joskus onnistuisi. OhTu-vastaavat myös tunnistetaan kirjautumisen avulla.



Kuva 1. Käyttäjien saapuminen järjestelmään.

**OhTu-projektinhallinnan työkalu**

Tämän työkalun tavoitteena on toimia Tietojenkäsittelytieteen laitoksen OhTu-ryhmien apuna sekä projektin suunnittelussa että seurannassa.

OhTu-ryhmäläinen, kirjaudu sisään, niin pääset suoraan käsiksi oman projektisi tietoihin.

[OhTu-ryhmäläiset](#)

Muut käyttäjät pääsevät katselmaan OhTu-projektien seurantatietoja.

[Muut käyttäjät](#)

Kuva 2. Järjestelmän varsinainen etusivu.

**Käyttäjän tunnistus**

**Tunnus (sähköpostiosoite)**

**Salasana**

**Unohtuiko salasana?**

Voit [pyytää](#) järjestelmää lähettämään sen sähköpostisi.

Kuva 3. Sisäänkirjautumissivu.



## **1.4. Järjestelmän sivuston valikkorakenne**

Tähän valikkorakennekuvaukseen lisätään tietoja ja toimintoja sitä mukaa kuin Lohtu-projektin aikana niitä suunnitellaan ja toteutetaan.

Järjestelmän valikkorakenteeseen vaikuttaa suoraan käyttäjän rooli. Mikäli käyttäjä tunnustetaan OhTu-vastaavaksi, näkyvät hänen valikossaan linkit hallinnollisiin tehtäviin. Lisäksi hän voi suorittaa projektien seurantaan ja vertailuun liittyviä toimenpiteitä.

- Hallinnolliset toimet
  - Tietojen haku Kurssijärjestelmästä

Jos käyttäjä on jonkin aktiivisen projektin jäsen, näytetään hänen toimintovalikossaan pääsy oman projektinsa suunnittelu- ja toteumasivuille. Lisäksi hän pääsee projektien seurantaan ja vertailuun.

- Projektin suunnittelu
  - Projektin perustiedot
  - 1. syklin suunnittelu
  - Lisää sykli

OhTu-projektin valikossa näkyy niin monta syklin suunnittelu –toimintoa kuin projektissa on syklejä. Lisää sykli –linkistä lisätään uusi sykli projektiin, jolloin näytölle avataan sivu uuden syklin tietojen suunnittelulle sekä lisätään valikkoon uuden syklin linkki (yllä olevassa esimerkissä valikkoon lisättäisiin siis ”2. syklin suunnittelu” -linkki).

Jos käyttäjä ei kuulu kumpaankaan tunnustettavista käyttäjäryhmistä tarjotaan hänelle ainoastaan projektien seuranta- ja vertailutoimintoja.

## **1.5. Näyttökuvaukset**

### **1.5.1. Projektin perustiedot**

Tällä sivulla tallennetaan projektin perustiedot, mikä periaatteessa tehdään vain kerran. Näistä tiedoista muut ovat yleistä informaatiota, mutta ohjelman ositus vaikuttaa esimerkiksi syklin suunnitteluun siten, että jos ositus on tehty, voidaan tehtävä kohdistaa johonkin tiettyy osaan. Ohjelman ositus on ainoa vapaaehtoisesti täytettävä tieto näytöllä. Kaikki muut tiedot ovat pakollisia.

Kun käyttäjä painaa Tallenna-painiketta ja tallennus onnistuu, antaa järjestelmä ilmoituksen tietojen onnistumisesta ja vie käyttäjän ensimmäisen syklin suunnittelusivulle. Jos tiedoissa on virheitä, palataan perustiedot-sivulle, jossa virheellisen kentän otsikko on punainen ja sen yhteydessä on toimintaohje (Esim Valitse prosessimalli). Tallennuksen muuten epäonnistuessa näytetään erillinen virheilmoitussivu toimintaohjeineen.

**Projektin perustiedot**

Projektin perustiedot tallennetaan periaatteessa yhden kerran projektin alussa. Tiedoista kaikki muut ovat yleistä informaatiota projektista paitsi ohjelman ositus, jonka avulla projektissa mitattavien suureiden kohdennusta voidaan tarkentaa.

**Nimi**  **Projektin nimeksi** kirjoitetaan projektiryhmän keksimä nimi, joka yksilöi projektin muista.

**Proessimalli**  **Proessimallilla** tarkoitetaan projektin suorituksen mallia, jossa projekti jaetaan vaiheisiin eri mallien mukaisesti. Esimerkiksi lisääville malleille on tyypillistä syklinen toiminta...

**Ohjelmistotyyppi**  **Ohjelmistotyyppillä** viitataan lähinnä tietojenkäsittelyn suuntautumislinjaan, jonka alainen projekti ensisijaisesti on, esimerkiksi Hajautetut järjestelmät tai Algoritmien linja.

**Ohjelman ositus** **Ohjelman ositus** on vapaaehtoinen toiminto, jossa ohjelma jaetaan selkeisiin osiin, esimerkiksi käyttöliittymään, sovelluslogiikkaan ja tietokantaan tai vielä tarkemmalla tasolla valmiisiin luokkiin. Ohjelman osituksen tarkoituksena on tarjota mahdollisuus seurata suureita (esim. työaikaa) ohjelman osiin kohdistettuina. Tällöin projektin seurantajärjestelmään tallettuu tieto esimerkiksi siitä, paljonko työtunteja jonkin tietyn ohjelman osan ohjelmointiin on käytetty.

**Ohjelman osat**

	Poista osa
	Poista osa
	Poista osa
Lisää osa	

Kuva 4. Projektin perustiedot –sivu.

### 1.5.2. Syklin suunnittelu

Sivulla suunnitellaan yhden syklin sisältämät vaiheet sekä näihin sisältyvät työtehtävät. Työtehtävä voidaan määrittellä välietapiksi (milestone), jolloin se näkyy syklin vaiheiden mukana projektin Gantt-kaaviossa.

Vaiheen suure-kentät ovat readonly-kenttiä, joihin lasketaan tiedot vaiheen työtehtävistä. Näitä kenttiä ei toteuteta vielä toisen syklin aikana, vaikka ne ovatkin nähtävissä kuvassa 5. Jokaiseen tehtävään liitetään kaikki ne suureet, jotka on projektissa määritelty seurattaviksi. Lisäksi jokainen tehtävä voidaan kohdistaa johonkin ohjelmiston osaan, jos ohjelman ositus on tehty. Jos ositusta ei ole toteutettu, ei tehtävillä näy ollenkaan osakenttiä. Pakollisia kenttiä ovat ainoastaan vaiheen nimi ja tehtävän nimi. Muut kentät ovat vapaaehtoisesti täytettäviä.

## 1. syklin vaiheet

Tallenna Poista sykli

Sykli jaetaan erillisiin työvaiheisiin, joita esimerkiksi perinteisen vesiputousmallin mukaisesti ovat 1. projektin suunnittelu, 2. määrittely, 3. suunnittelu, 4. toteutus ja 5. testaus. Vaiheille arvioidaan aloitus- ja lopetuspäivämäärät. Lisäksi vaihe voidaan edelleen jakaa työtehtäviin, joiden avulla myös kokonaisvaiheen suureiden arviointi helpottuu. Tehtäville määritellään myös tekijät sekä päivämäärä, jolloin tehtävän tulee olla suoritettuna. Tehtävä voidaan lisäksi määrittellä ns. välietapiksi (milestone), eli tarkistuspisteeksi, jolloin tietyt keskeiset tehtävät pitäisi olla tehtynä ennen kuin voidaan edetä projektissa. Projektista luotavassa Gantt-kaaviossa esitetään graafisesti kustakin projektin syklistä vaiheet sekä näiden välietapit.

---

Vaiheen nimi	Alkupvm (pp.kk.vvvv)	Loppupvm (pp.kk.vvvv)		
Projektin suunnittelu	15.1.2003	28.1.2003	<span>Poista vaihe</span>	

Tehtävä	Määräaika	Tekijä(t)	Osa	Arvioidut suuret		
Projektin suunnittelu	28.1.2003	Andreas Asuja Kimmo Airamaa Mari Muuronen Seppo Pastila Virve Taivaljärvi		Työaika	10	<span>Poista tehtävä</span>
	<b>Välietappi</b> <input type="checkbox"/>			LOC	0	
				Dokumentit	0	
				Virheet	1	
Projektisuunnitelman kirjoitus	28.1.2003	Andreas Asuja Kimmo Airamaa Mari Muuronen Seppo Pastila Virve Taivaljärvi		Työaika	10	<span>Poista tehtävä</span>
	<b>Välietappi</b> <input type="checkbox"/>			LOC	0	
				Dokumentit	1	
				Virheet	1	
Projektisuunnitelman katselmont	28.1.2003	Andreas Asuja Kimmo Airamaa Mari Muuronen Seppo Pastila Virve Taivaljärvi		Työaika	2	<span>Poista tehtävä</span>
	<b>Välietappi</b> <input type="checkbox"/>			LOC	0	
				Dokumentit	0	
				Virheet	1	
		Andreas Asuja Kimmo Airamaa Mari Muuronen Seppo Pastila Virve Taivaljärvi		Työaika	0	<span>Poista tehtävä</span>
	<b>Välietappi</b> <input type="checkbox"/>			LOC	0	
				Dokumentit	0	
				Virheet	0	

Lisää tehtävä

<b>Yhteensä:</b>	Työaika	0
	LOC	0
	Dokumentit	0
	Virheet	0

Lisää vaihe  
Tallenna

Kuva 5. Syklin suunnittelusivu.

Kun sykli on tyhjä (eli sivulle tullaan ensimmäistä kertaa projektissa), sivulla on valmiina näkyvissä yksi vaihe ja sillä neljä tehtävää. Tässä vaiheessa kaikki kentät ovat luonnollisesti tyhjiä. Käyttäjä

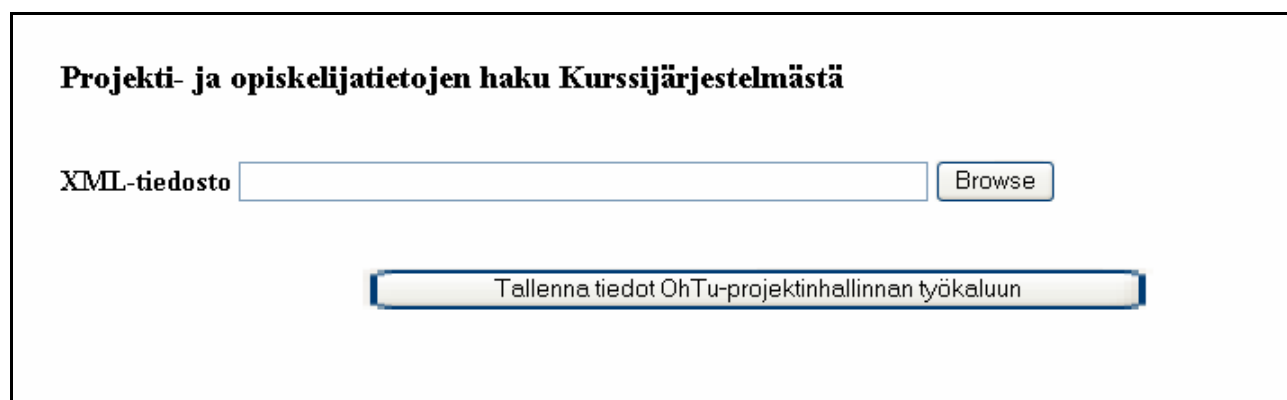
voi lisätä vaiheelle tehtävän painamalla Lisää tehtävä –painiketta. Samoin uusi vaihe lisätään Lisää vaihe –painikkeella. Uudessa lisätyssä vaiheessa on jälleen neljä tyhjää tehtävää valmiina. Syklin sisältämät vaiheet esitetään sivulla aloituspäivämäärän mukaisessa järjestyksessä. Jos kahdella vaiheella on sama aloituspäivä, järjestetään toissijaisesti lopetuspäivämäärän mukaan. Samoin tehtävät järjestetään määräaikansa mukaan.

Sivulla käyttäjä voi lisäksi poistaa yksittäisiä tehtäviä ja vaiheita sekä koko syklin. Vaiheen ja syklin poistaminen varmistetaan vielä käyttäjältä ennen lopullisen poiston suorittamista.

Vaikka suureiden arviointi on mukana tämän sivun kuvauksessa, kyseistä ominaisuutta ei toteuteta Lohtu-projektin kahdessa ensimmäisessä syklissä. Toteutettavat ominaisuudet on rajattu Määrittelydokumentissa.

### 1.5.3. Projekt- ja opiskelijatietojen haku Kurssijärjestelmästä

OhTu-vastaava voi hakea projekt- ja opiskelijatiedot suoraan omasta Kurssijärjestelmästä. Tämä tapahtuu syöttämällä käyttöliittymän kenttään tiedot sisältävän XML-tiedoston polku. Tietoja tallennettaessa OhTu-projektinhallinnan työkalu lukee tiedoston sisältämän XML-muotoisen tiedon ja luo sen perusteella järjestelmään uudet projektit sekä näille projekteille osanottajat. Tietojen tallennuksen onnistumisesta ja mahdollisista virhetilanteista annetaan käyttäjälle ilmoitus. Tallennuksen onnistuessa järjestelmä luo jokaiselle lisätylle opiskelijalle salasanan, jonka avulla käyttäjä voi kirjautua järjestelmään. Nämä salasanat listataan OhTu-vastaavan nähtäväksi ja opiskelijoille välitettäväksi.



**Projekt- ja opiskelijatietojen haku Kurssijärjestelmästä**

XML-tiedosto

Kuva 6. Tietojen haku Kurssijärjestelmästä

## 2. Tietokannan kuvaus

Tietokantana käytetään PostGres:iä ja siten sen sallimia tietotyyppejä. Erityisesti tulee huomioida käytetty tietotyyppi Serial, joka toimii lähes kaikkien taulujen avaimena. Se on integer muotoinen ja sallii arvot välillä 0 - 2147483647. Serial –tietotyyppi huolehtii siitä, että sarakkeeseen generoidaan uusi uniikki avain.

Tietokanta on kuvattu kuvassa 7 sekä seuraavissa kappaleissa. Lisäksi kuvassa 8 on esimerkki tietokannan sisällöstä. Tässä syklissä ei oteta kantaa tietokannan indeksointiin.

### 2.1. Taulu SoftwareTypes

Sisältää tiedon eri ohjelmistotuotetyypeistä.

swtId	Serial	PK, NOT NULL
softwareType	varchar(50)	UNIQUE, NOT NULL

### 2.2. Taulu ProcessModels

Sisältää eri prosessimallit.

pmId	Serial	PK, NOT NULL
processModel	varchar(50)	UNIQUE, NOT NULL

### 2.3. Taulu Datatypes

Sisältää suureiden tietotyyppit. Tulee sisältämään arvot positiivisille kokonaisluvuille ja positiivisille reaalityyppien luvuille.

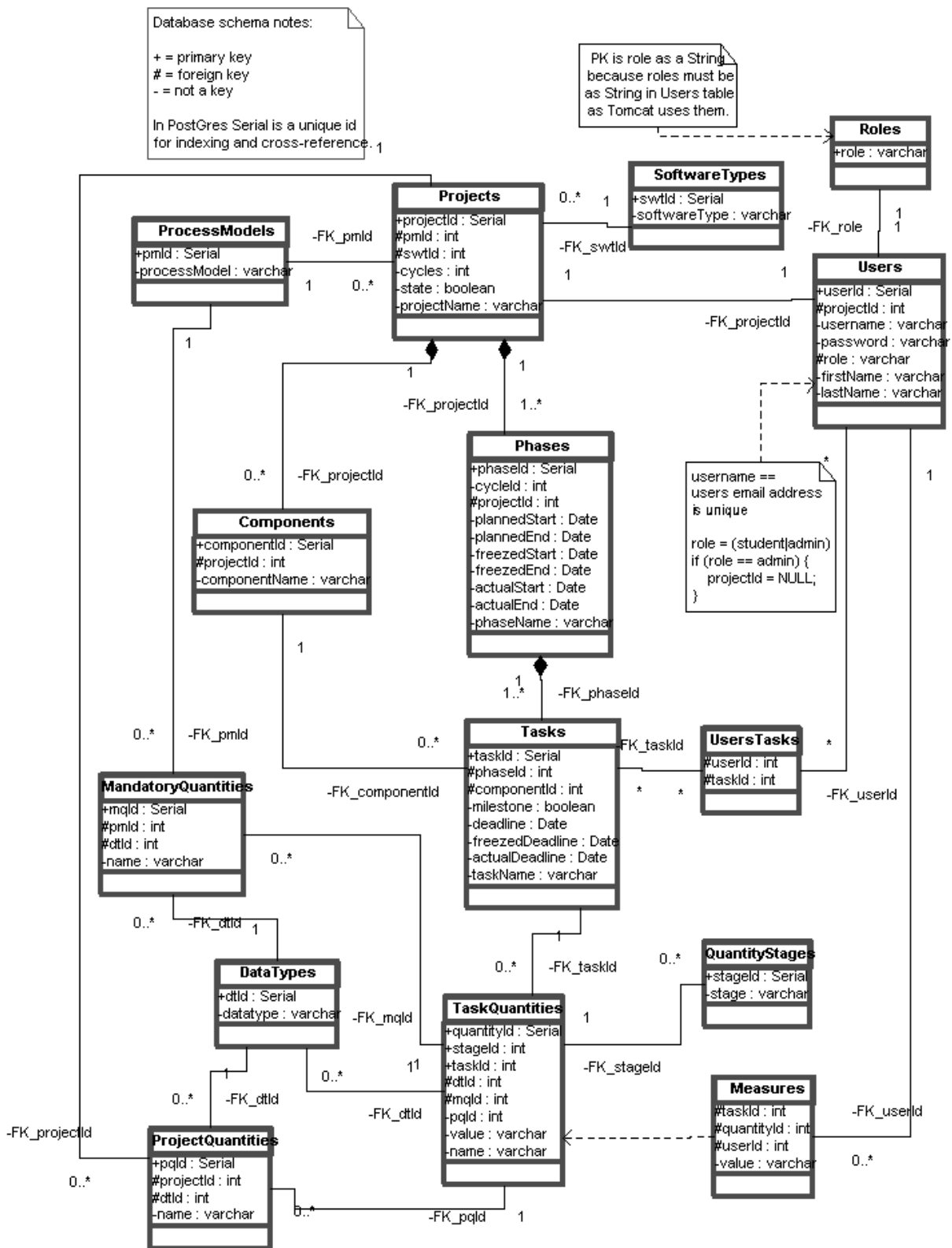
DtId	Serial	PK, NOT NULL
datatype	varchar(50)	UNIQUE, NOT NULL

Esimerkkisisältö:

<b>dtId</b>	<b>datatype</b>
1	positiveInteger
2	positiveDouble



Kuva 7. Tietokantakaavio



## 2.4. Taulu MandatoryQuantities

Sisältää jokaisessa prosessimallissa kerättävät pakolliset suureet.

MqId	Serial	PK, NOT NULL
PmId	int	FK References ProcessModels(pmId), NOT NULL
DtId	int	FK References DataTypes(dtId), NOT NULL
Name	varchar(50)	UNIQUE, NOT NULL

## 2.5. Taulu ProjectQuantities

Sisältää projektissa kerättävät vapaavalintaiset suureet.

PqId	Serial	PK, NOT NULL
Projected	int	FK References Projects(projectId), NOT NULL
DtId	int	FK References DataTypes(dtId), NOT NULL
Name	varchar(50)	NOT NULL

## 2.6. Taulu QuantityStages

Suureen tila. Arvoja esim. 'plan', 'freezed', 'completed'

StageId	Serial	PK, NOT NULL
Stage	varchar(15)	UNIQUE, NOT NULL

Esimerkkisisältö:

<b>stageId</b>	<b>stage</b>
1	plan
2	freezed
3	completed



## 2.7. Taulu Roles

Tällä taululla ei ole Serial –tyyppistä avainta, koska halutaan, että Users –taulussa on käyttäjän rooli varchar –tyyppisenä. Taulun tehtävä on siis vain pitää huolta, että taulussa Users ei ole kuin tässä taulussa määritellyjä rooleja.

Role	varchar(10)	PK, NOT NULL
------	-------------	--------------

## 2.8. Taulu Projects

Sisältää perustiedot projekteista.

Projected	Serial	PK, NOT NULL
PmId	int	FK References ProcessModels(pmId)
SwId	int	FK References SoftwareTypes(swId)
Cycles	int	DEFAULT 1
State	boolean	DEFAULT true
ProjectName	varchar(50)	

## 2.9. Taulu Users

Sisältää käyttäjien perustiedot.

UserId	Serial	PK, NOT NULL
Projected	int	FK References Projects(projectId) NOT NULL
Username	varchar(150)	UNIQUE, NOT NULL
Password	varchar(10)	NOT NULL
Role	varchar(10)	FK References Roles(role) NOT NULL
FirstName	varchar(20)	NOT NULL
LastName	varchar(50)	NOT NULL

## 2.10. Taulu Components

Sisältää projektin teknisen jaon eri osasiin, kuten käyttöliittymä, tietokanta, serveri.

ComponentId	Serial	PK, NOT NULL
Projected	int	FK References Projects(projectId) NOT NULL
componentName	varchar(50)	NOT NULL

## 2.11. Taulu Phases

Sisältää projektin vaiheet.

PhaseId	Serial	PK, NOT NULL
CycleId	int	NOT NULL
Projected	int	FK References Projects(projectId) NOT NULL
PlannedStart	date	
PlannedEnd	date	
FreezedStart	date	
FreezedEnd	date	
ActualStart	date	
ActualEnd	date	
PhaseName	varchar(50)	NOT NULL

## 2.12. Taulu Tasks

Sisältää vaiheiden tehtävät.

Tasked	Serial	PK, NOT NULL
PhaseId	int	FK References Phases(phaseId) NOT NULL
ComponentId	int	FK References Components(componentId)
Milestone	boolean	NOT NULL

Deadline	date	
freezedDeadline	date	
actualDeadline	date	
TaskName	varchar(50)	NOT NULL

### 2.13. Taulu UsersTasks

Taulu, jolla käyttäjät ja heille annetut tehtävät liitetään toisiinsa.

Tasked	int	FK References Tasks(taskId) NOT NULL
UserId	int	FK References Users(userId) NOT NULL

### 2.14. Taulu TaskQuantities

Sisältää jokaiseen tehtävään liitetyt suureet.

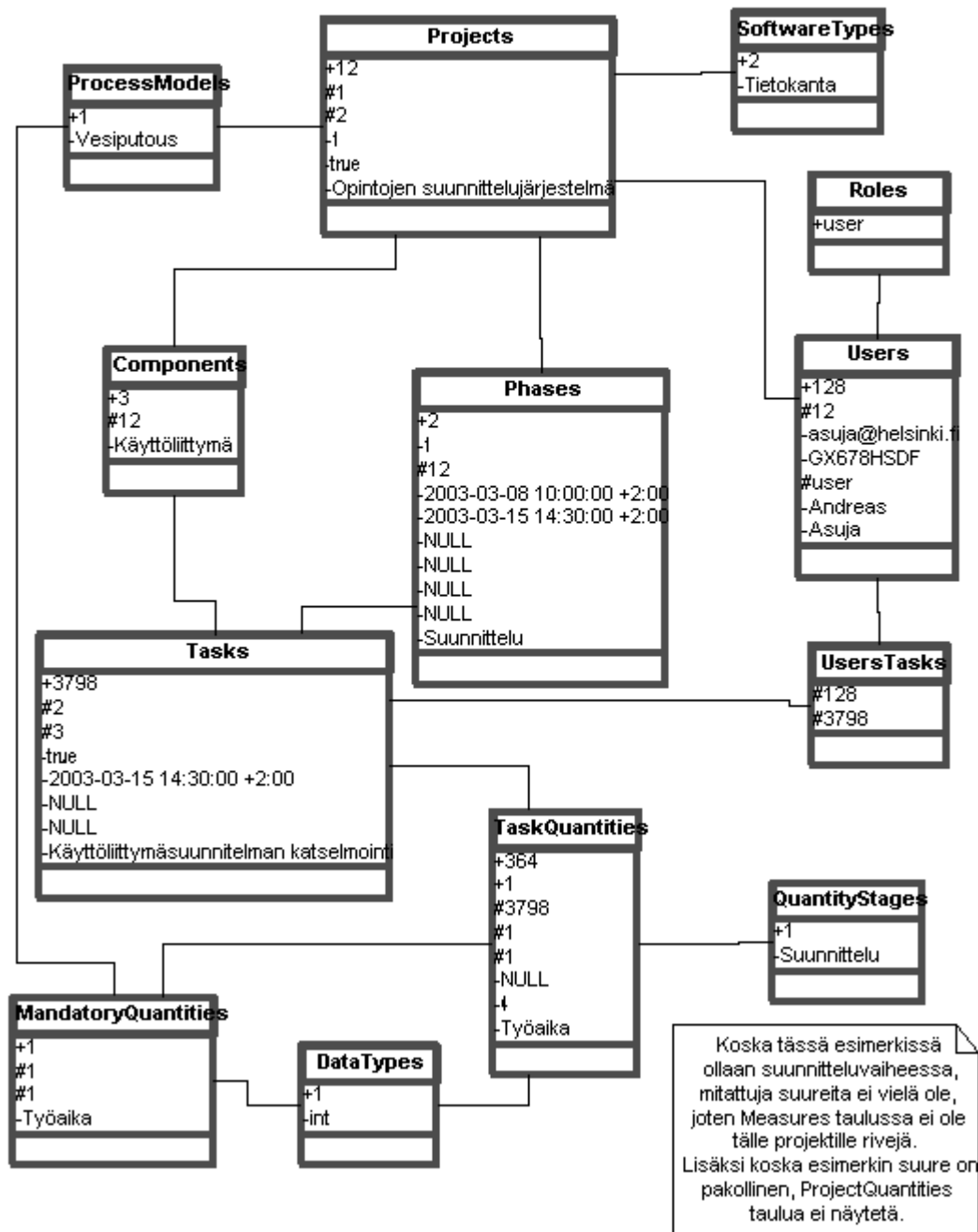
QuantityId	int	NOT NULL	PK (quantityId, stageId, taskId)
StageId	int	FK References QuantityStages(stageId) NOT NULL	
Tasked	int	FK References Tasks(taskId) NOT NULL	
DtId	int	FK References DataTypes(dtId), NOT NULL	
MqId	int	FK References MandatoryQuantities(mqId)	
PqId	int	FK References ProjectQuantities(pqId)	
Value	varchar(100)		
Name	varchar(50)	NOT NULL	

## 2.15. Taulu Measures

Sisältää suureisiin liittyvät tuloksien kirjaukset.

Tasked	int	NOT NULL	Nämä kaksi kenttää viittaavat implisiittisesti taulun TaskQuantities sarakkeisiin quantityId ja stageId.
QuantityId	int	NOT NULL	
UserId	int	FK References Users(userId) NOT NULL	
Value	varchar(100)	NOT NULL	

Kuva 8. Esimerkki tietokannan sisällöstä

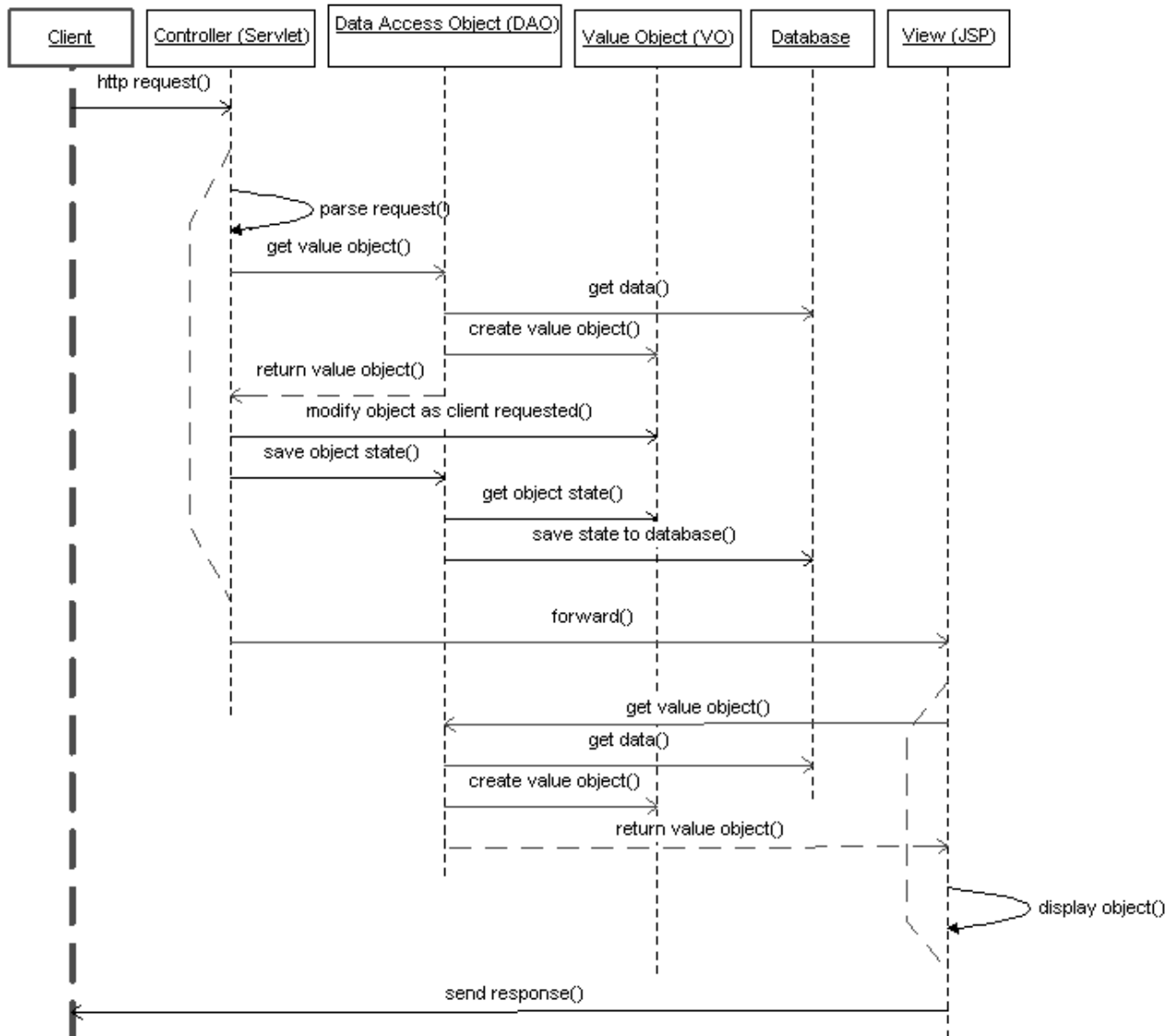


### 3. Sovelluslogiikka

Syklissä 1 toteutetaan vain sisäänkirjautuminen ja projektin perustietojen syöttö. Tässä dokumentissa on kuitenkin kuvattu sovellusta hiukan laajemmin jatkosyklejä ajatellen.

#### 3.1. Yleinen arkkitehtuuri

Sovelluksen arkkitehtuurimallina on Model-View-Controller. Servlet Container –ympäristössä tämä tarkoittaa yleensä sitä, että Controllerin osassa on servlet, joka vastaanottaa käyttäjän pyynnöt ja tekee muutokset tarvittaviin olioihin (Model), joiden tila talletetaan tietokantaan. Sitten käytetään JSP –sivuja tulostettaessa käyttäjälle vastaus (View).



Kuva 9. Model-View-Controller arkkitehtuuri

Tiedot mallinnetaan luokkina ja niiden tallentamiseen tietokantaan käytetään hyväksi Data Access Object (DAO) J2EE –suunnittelumallia. Jokaista tietorakenneluokkaa vastaa yksi DAO –luokka. Tietorakenneluokat toteutetaan käyttäen Value Object (VO) J2EE –suunnittelumallia. Sekä DAO-että VO- luokat käyttävät lisäksi Iterator –suunnittelumallia.

Sovelluksen ohjauksesta vastaavat kolme servlettiä, jotka toteuttavat kolme sovelluksen päätoimintoa. ProjectController vastaa karkeasti projektiin suunnittelusta eli OhTu –projektiin osallistuvien toiminnoista. AdminController hoitaa projektien hallinnoinnin eli OhTu –vastaavan toiminnot. StatsController hoitaa kaikille avoimien OhTu –projektien tuloksien seuranta-toiminnot. Servletit käyttävät Front Controller –J2EE suunnittelumallia. Lisäksi on olemassa LohtuInit –servlet, jonka tehtävänä on asettaa LohtuResources –luokalle tietokanta- ja lokitusparametrit.

Käyttäjälle tulostettavat sivut tehdään JSP –tekniikalla. Sivuille ei ole suoraa pääsyä, vaan käyttäjät ohjataan ensin Controllereille, jotka valitsevat parametrien mukaan oikeat sivut. Controllerit myös alustavat tiettyjä olioita, joita JSP –sivut tarvitsevat. Controllerit pitävät huolen myös siitä, että käyttäjien HTML-lomakkeiden tuplalähetykset eivät onnistu eli siis saman lomakkeen lähettäminen kahteen kertaan esim. lähetyksnappia tuplaklikkaamalla ei aiheuta ongelmia. Tähän käytetään Synchronized Token –käytäntöä. Controllerit asettavat jokaiseen response olioon Cache-Control: no-cache –http-headerin, jotta käyttäjä ei voi selaimen takaisin –toiminnolla mennä edelliselle sivulle ja näin käyttää vanhaa tokenia.

### **3.2. Pakettijako**

Sovellus jaetaan paketteihin seuraavasti:

- fi.helsinki.cs.lohtu.util  
Sisältää kaikkien pakettien käyttämiä yleisiä resursseja
- fi.helsinki.cs.lohtu.model  
Sisältää tietorakenneluokat ja niiden DAO:t. Tietorakenneluokkien nimet päättyvät kirjaimiin VO ja DAO:iden nimet DAO.
- fi.helsinki.cs.lohtu.view  
JSP –sivujen apuluokat. Näitä ei ole syklissä 1.
- fi.helsinki.cs.lohtu.control  
Controller –luokat ja niiden apuluokat.

Paketit on kuvattu tarkemmin kappaleissa 3.3 - 3.6.

### 3.3. Paketti fi.helsinki.cs.lohtu.util

Paketin luokkakaavio on kuvassa 10. Paketin luokat tarjoavat erilaisia palveluja muiden pakettien luokille.

#### 3.3.1. fi.helsinki.cs.util.LohtuInit

Tarkoitus: Asettaa LohtuResources –luokan parametrit. Käynnistetään heti, kun lohtu sovellus ladataan Tomcatiin. Asettaa LohtuResources –luokalle käytetyn aikaleimojen muodon.

JUnit-testattava: Ei

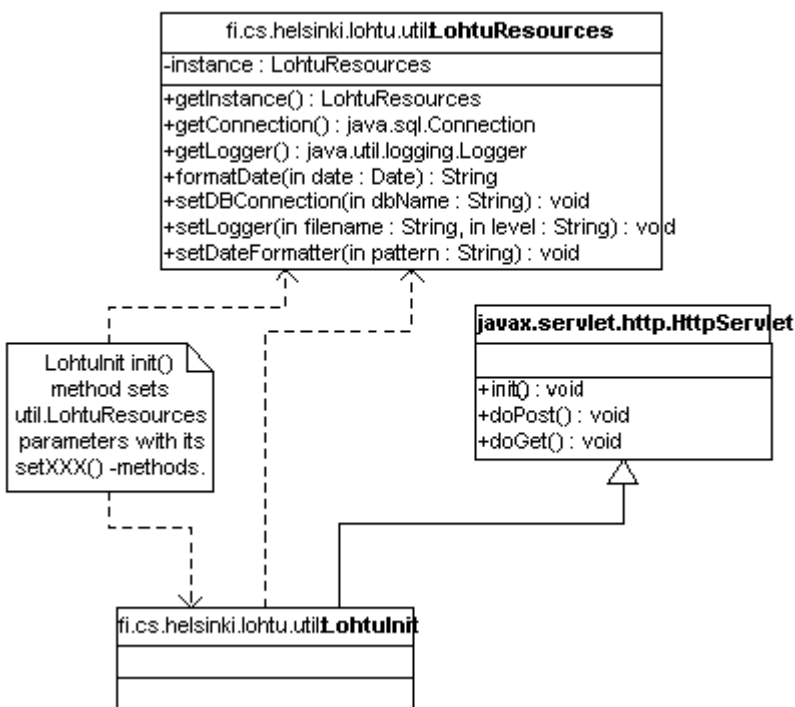
Toteutusykyli: 1

#### 3.3.2. fi.helsinki.cs.util.LohtuResources

Tarkoitus: Singleton –suunnittelumallin toteuttava luokka, joka pitää viitteet tietokantayhteys pooliin ja lokitiedostoon. Tämän luokan metodein voidaan asettaa päivämäärän muoto ja muuntaa haluttu päivämäärä järjestelmässä käytettävään muotoon.

JUnit-testattava: Ei

Toteutusykyli: 1



Kuva 10. Paketti fi.helsinki.cs.lohtu.util



### 3.4. Paketti **fi.helsinki.cs.lohtu.model**

Paketin luokkakaaviot ovat kuvissa 11 ja 12. Sisältää tietorakenneluokat (Value Objects, VO) ja niiden tietokantaan talletusluokat (Data Access Objects, DAO) sekä luokan DAOFactory, jolla voidaan luoda haluttuja DAO –luokkia. Kaikkien VO –luokkien setXXX –metodit ja *julkiset* konstruktorit asettavat ko. luokkien isDirty –muuttujan arvoksi tosi (true).

#### 3.4.1. **fi.helsinki.cs.lohtu.model.DAOFactory**

Tarkoitus: DAOFactory –luokalla luodaan haluttuja DAO –luokkien instansseja. Se hakee luotaessa itselleen yhden tietokantayhteyden (ja DAO:ille) ja pitää yllä viitteitä luomiinsa DAO –instansseihin. Näin saavutetaan suorituskykyetua, kun käytetään samaa tietokantayhteyttä kaikissa DAO:issa ja DAO:iden SQL –lauseet voidaan tallettaa PreparedStatement:teihin, joita on nopea ajaa, kun ne on kerran luotu. Controller servletit luovat pyynnön saadessaan DAOFactory –olion ja tallentavan sen HttpServletRequest –olion attribuutteihin nimellä ”DAOFactory”.

jUnit-testattava: Ei

Toteutus sykli: 1

#### 3.4.2. **fi.helsinki.cs.lohtu.model.ProcessModelVO**

Tarkoitus: Tietorakenneluokka joka mallintaa prosessimallit.

jUnit-testattava: Kyllä

Toteutus sykli: 1

#### 3.4.3. **fi.helsinki.cs.lohtu.model.ProcessModelDAO**

Tarkoitus: Hoitaa prosessimalliolioiden haun ja talletuksen tietokantaan.

jUnit-testattava: Ei

Toteutus sykli: 1

#### 3.4.4. **fi.helsinki.cs.lohtu.model.SoftwareTypeVO**

Tarkoitus: Tietorakenneluokka, joka mallintaa ohjelmistotyypit.

jUnit-testattava: Kyllä

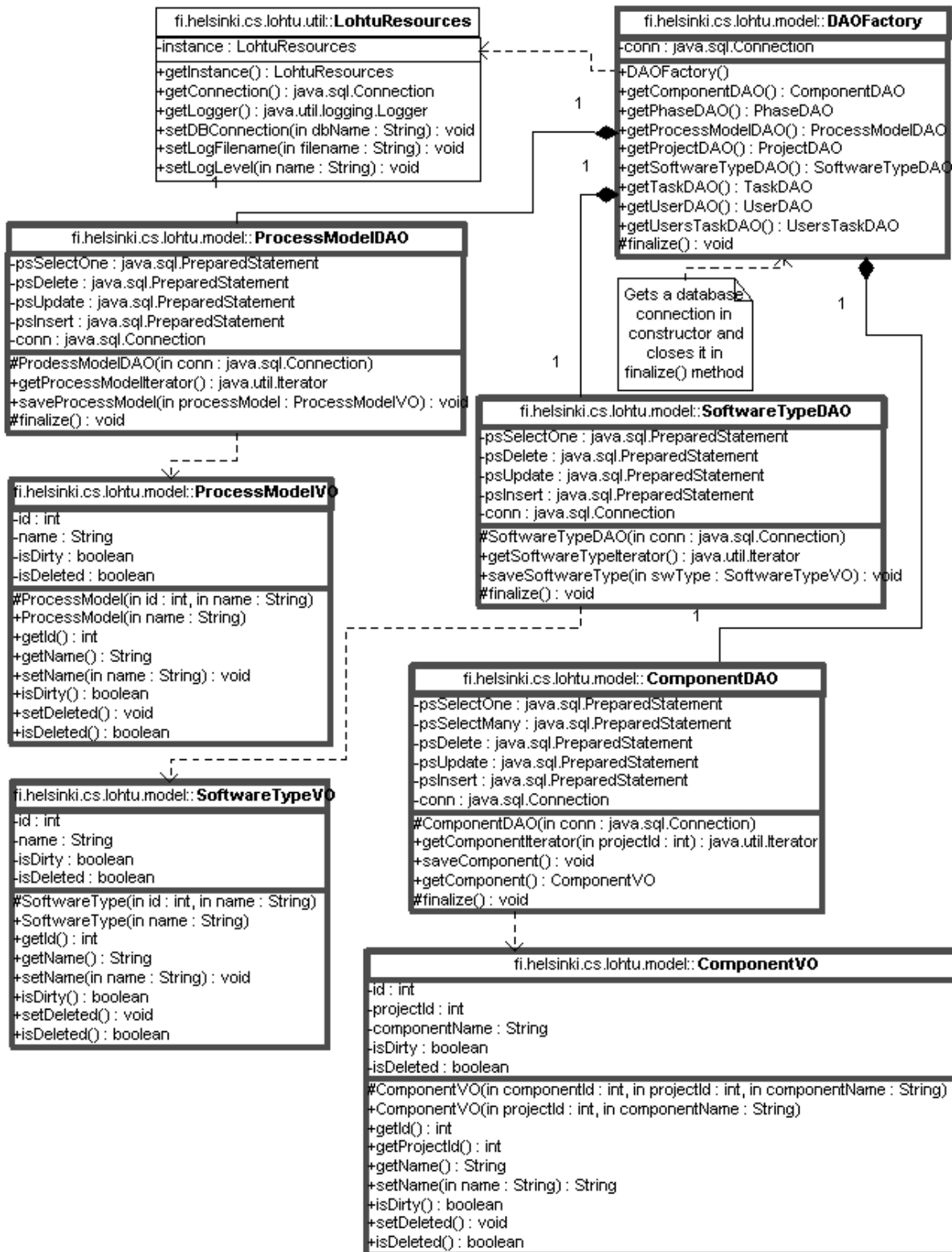
Toteutus sykli: 1

#### 3.4.5. **fi.helsinki.cs.lohtu.model.SoftwareTypeDAO**

Tarkoitus: Hoitaa ohjelmistotyypilioiden haun ja talletuksen tietokantaan.

jUnit-testattava: Ei

Toteutus sykli: 1

Kuva 11. Paketin `fi.helsinki.cs.lohtu.model` osa 1.

**3.4.6. fi.helsinki.cs.lohtu.model.ComponentVO**

Tarkoitus: Tietorakenneluokka, joka mallintaa projektin ohjelmistokomponentin.

jUnit-testattava: Kyllä

Toteutus sykli: 1

**3.4.7. fi.helsinki.cs.lohtu.model.ComponentDAO**

Tarkoitus: Hoitaa ohjelmistokomponenttiolioiden haun ja talletuksen tietokantaan.

jUnit-testattava: Ei

Toteutus sykli: 1

**3.4.8. fi.helsinki.cs.lohtu.model.ProjectVO**

Tarkoitus: Tietorakenneluokka, joka mallintaa projektin tiedot.

jUnit-testattava: Kyllä

Toteutus sykli: 1

**3.4.9. fi.helsinki.cs.lohtu.model.ProjectDAO**

Tarkoitus: Hoitaa projektiolioiden haun ja talletuksen tietokantaan.

jUnit-testattava: Ei

Toteutus sykli: 1

**3.4.10. fi.helsinki.cs.lohtu.model.UsersVO**

Tarkoitus: Tietorakenneluokka, joka mallintaa käyttäjän tiedot.

jUnit-testattava: Kyllä

Toteutus sykli: 2

**3.4.11. fi.helsinki.cs.lohtu.model.UsersDAO**

Tarkoitus: Hoitaa käyttäjäolioiden haun ja talletuksen tietokantaan.

jUnit-testattava: Ei

Toteutus sykli: 2

**3.4.12. fi.helsinki.cs.lohtu.model.TaskDAO**

Tarkoitus: Hoitaa tehtäväolioiden haun ja talletuksen kantaan. Luokkayhteyksiä kuvattu kuvassa 19.

jUnit-testattava: Ei

Toteutus sykli: 2

#### **3.4.13. fi.helsinki.cs.lohtu.model.TaskVO**

Tarkoitus: mallintaa tehtävät tietorakenneluokkana.

jUnit –testattava: on

Toteutus sykli: 2

#### **3.4.14. fi.helsinki.cs.lohtu.model.PhaseDAO**

Tarkoitus: hoitaa vaiheolioiden haun ja tallennuksen tietokantaan.

jUnit –testattava: ei

Toteutus sykli: 2

#### **3.4.15. fi.helsinki.cs.lohtu.model.PhaseVO**

Tarkoitus: mallintaa vaiheet tietorakenneluokkina.

jUnit –testattava: on

Toteutus sykli: 2

#### **3.4.16. fi.helsinki.cs.lohtu.model.UsersTaskDAO**

Tarkoitus: hoitaa käyttäjien ja tehtävien toisiinsa yhdistävien luokkien haun ja tallennuksen tietokantaan.

jUnit –testattava: ei

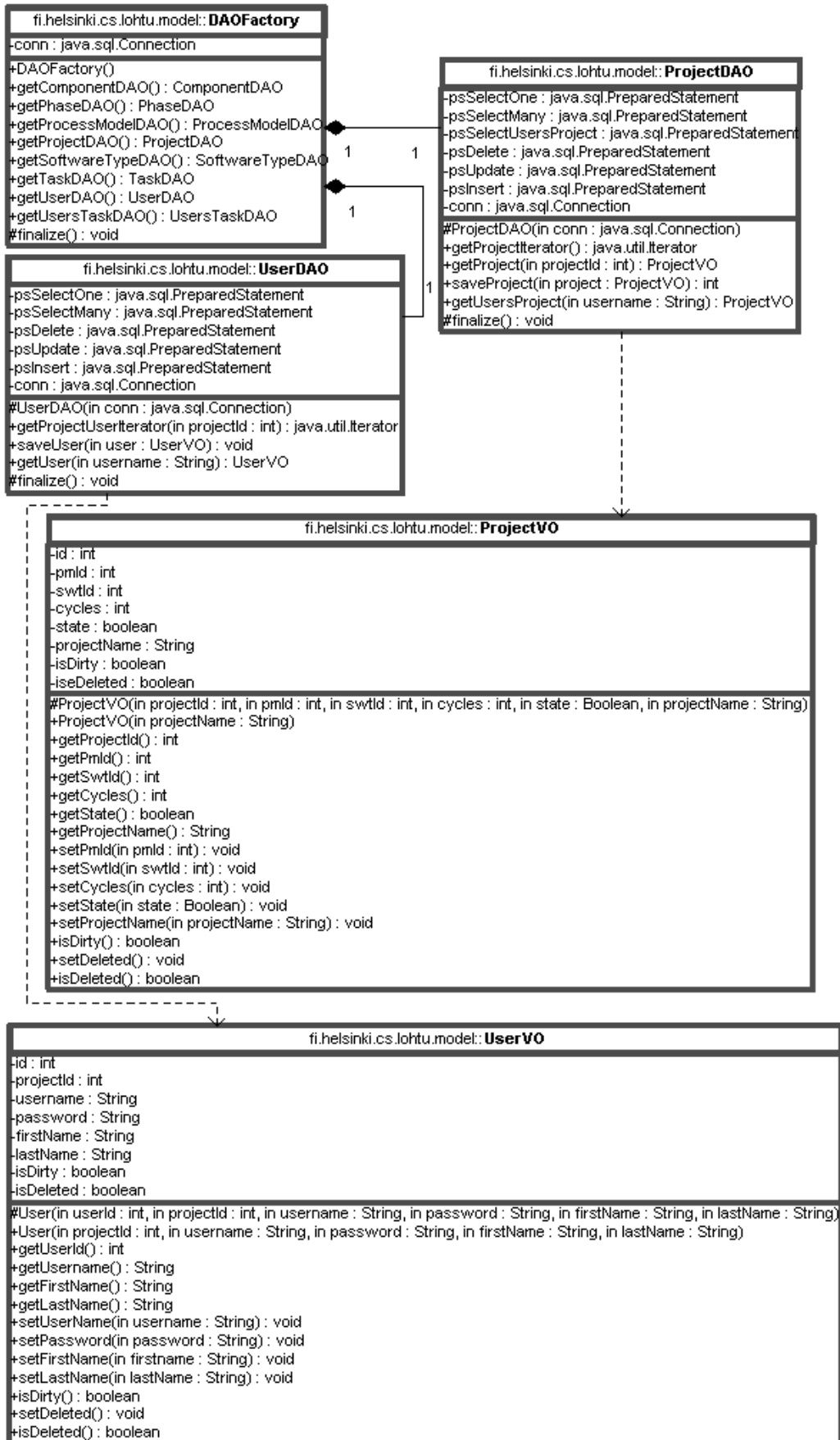
Toteutus sykli: 2

#### **3.4.17. fi.helsinki.cs.lohtu.model.UsersTaskVO**

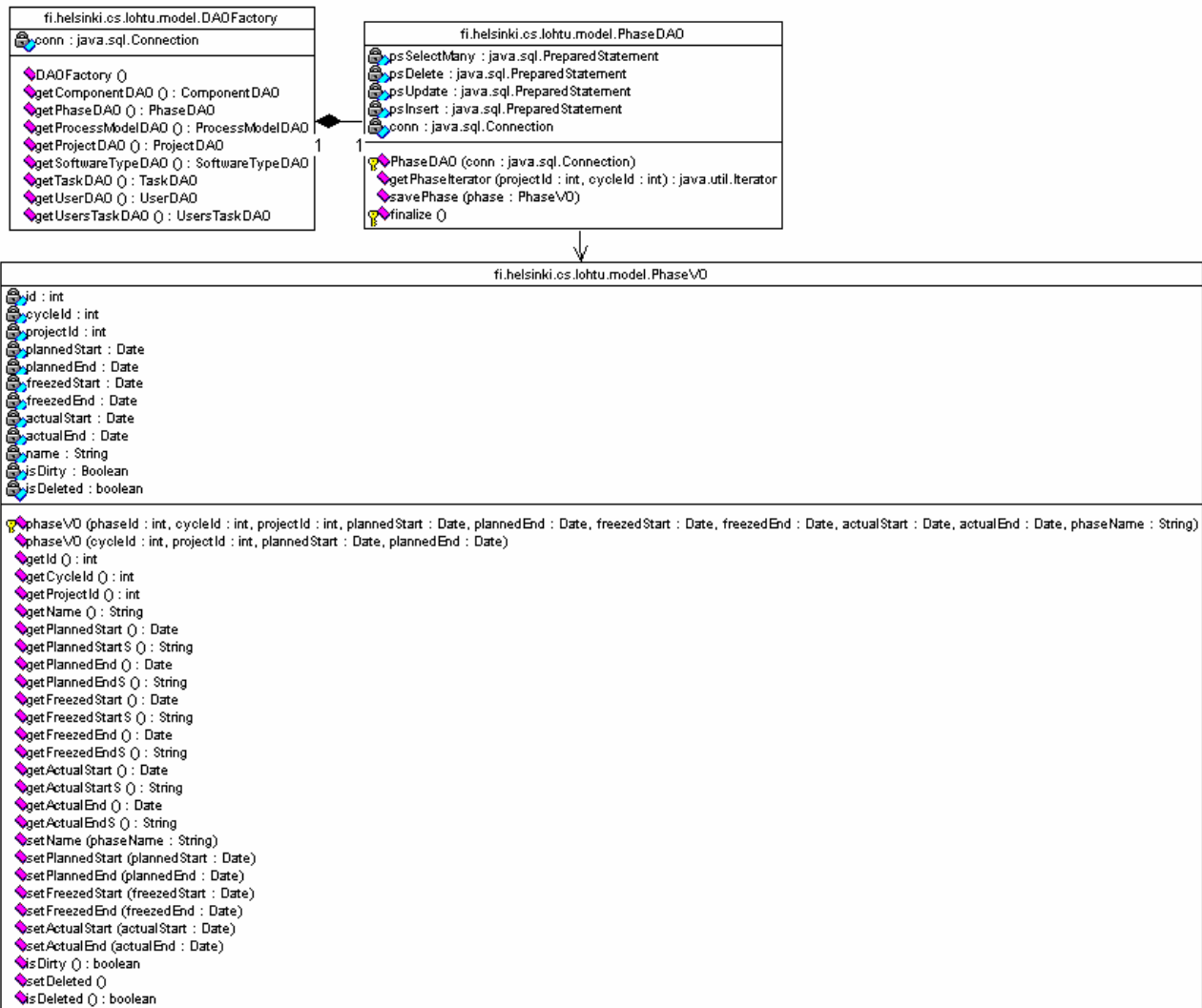
Tarkoitus: mallintaa työvaiheiden yhdistämisen käyttäjiin tietorakenneluokkina.

jUnit –testattava: on

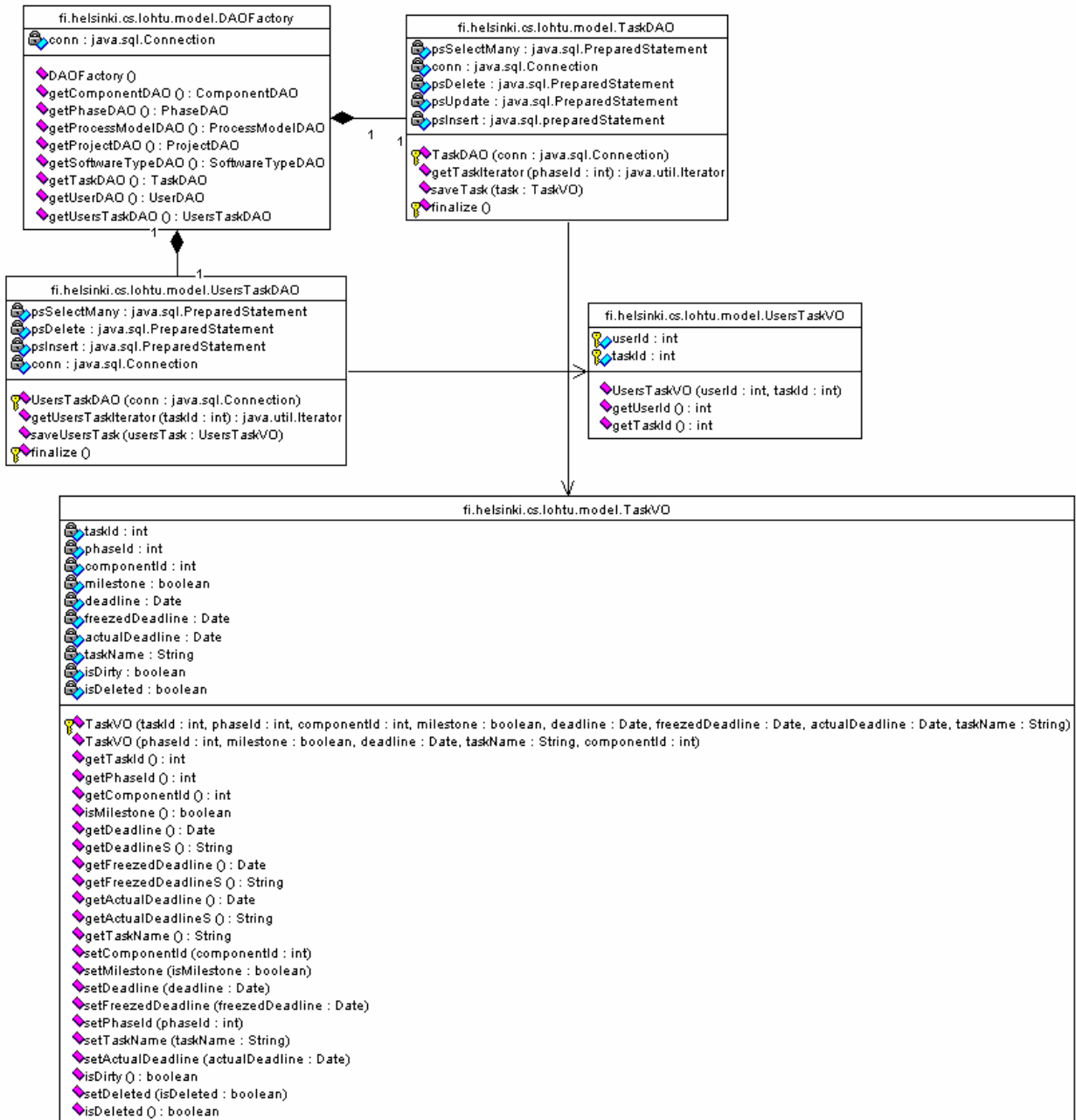
Toteutus sykli: 2



Kuva 12. Paketin fi.helsinki.cs.lohtu.model osa 2.



Kuva 13. Paketin `fi.helsinki.cs.lohtu.model` työvaiheisiin liittyvät luokat.



Kuva 14. Paketin `fi.helsinki.cs.lohtu.model` työtehtäviin liittyvät luokat.

### 3.5. Paketti `fi.helsinki.cs.lohtu.view`

Paketti on tyhjä syklissä 1.

### 3.6. Paketti `fi.helsinki.cs.lohtu.util.control`

Luokkakaavio on kuvassa 13. Sisältää servletit, jotka ohjaavat sovelluksen toimintaa.

Kaikki \*Controller servletit luovat alussa DAOFactory –olion ja tallettavat sen HttpServletRequest –olion attribuuttiin nimeltä ”DAOFactory”. Näin se on käytettävissä myös JSP –sivuilla.

\*Controller servletit tallettavat kaikki HTML –lomakkeiden parsinnassa tapahtuneet virheet java.util.HashMap –olioon, jonka ne tallettavat HttpServletRequest –olion attribuuttiin nimeltä ”Errors”. JSP –sivuilla voidaan sitten tuon HashMap:in avulla tulostaa HTML –lomakkeelle virheilmoitukset oikeisiin kohtiin. HashMap:in avaimina käytetään lomakkeen name –kenttiä. Eli esim. projektin perustiedot –sivulla projektin nimi –kentän name on project.name ja jos tätä parametria käsiteltäessä tulee virhe, talletetaan se HashMap:iin avaimella projet.name ja arvoksi virheilmoitusteksti, esim. ”Nimi oli liian pitkä, maksimipituus on 50 merkkiä”.

### **3.6.1. fi.helsinki.cs.lohtu.control.ProjectController**

Tarkoitus: Ohjaa OhTu –projektilaisten kurssin perustietojen syöttämistä ja muuttamista. Käsittelee HTML –lomakkeilta tulevat parametrit ja tallettaa ne VO– ja DAO –luokkien avulla. Ohjaa sitten käsittelyn eteenpäin JSP –sivulle.

jUnit-testattava: Ei

Toteutus sykli: 1

### **3.6.2. fi.helsinki.cs.lohtu.control.AdminController**

Tarkoitus: Ohjaa OhTu vastaavan toimintoja.

jUnit-testattava: Ei

Toteutus sykli: 3

### **3.6.3. fi.helsinki.cs.lohtu.control.StatsController**

Tarkoitus: Ohjaa kaikkien käyttäjien toimintoja.

jUnit-testattava: Ei

Toteutus sykli: 3

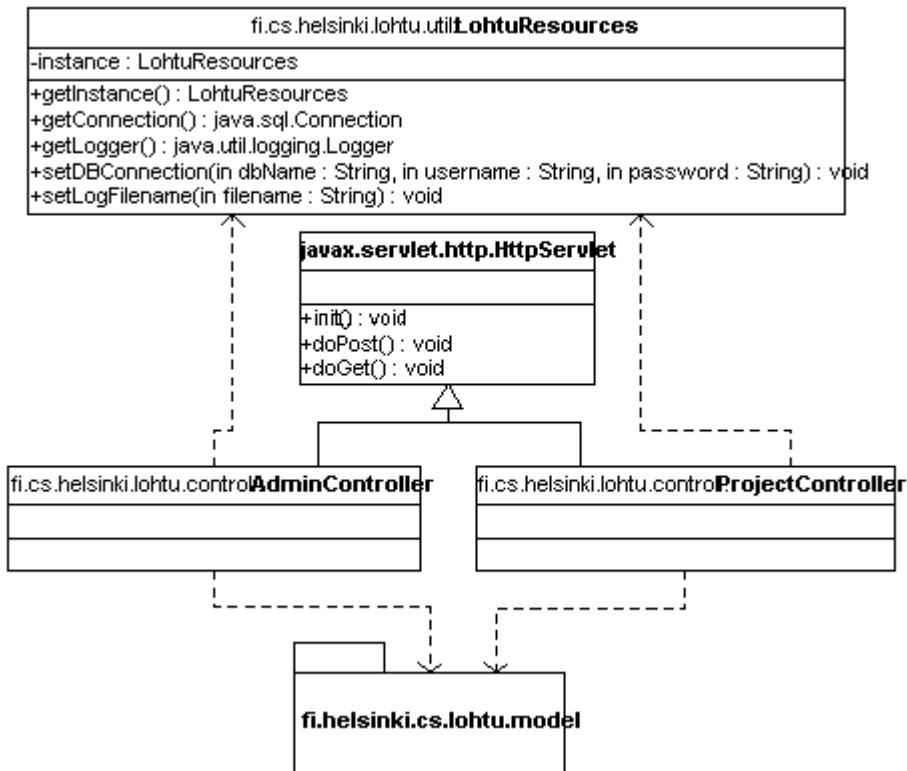
### **3.6.4. fi.helsinki.cs.lohtu.control.CycleController**

Tarkoitus: Ohjaa projektiryhmän jäsenten toimintoja heidän suunnitellessaan syklejä ja niiden vaiheita.

jUnit –testattava: Ei

Toteutus sykli: 2





Kuva 15. Paketti fi.helsinki.cs.lohtu.control.

### 3.7. JSP –sivut

JSP –sivut talletetaan hakemistorakenteeseen sen mukaan, mihin toimintoon ne liittyvät.

- Project  
Kaikki OhTu –projektiin osallistuvien toiminnot raporteja lukuunottamatta.
- Admin  
OhTu –vastaavan toiminnot.
- Stats  
Raporttien muodostussivut.

#### 3.7.1. Project

Syklissä 1 sisälsi ProjectBasics.jsp –sivun ja toisessa syklissä tehdään vielä CyclePlanning.jsp ja vasempaan laitetaan tuleva navigointisivu, josta voidaan mm. lisätä uusia syklejä.

#### 3.7.2. Admin

Suunnitellaan syklissä 2.

#### 3.7.3. Stats

Suunnitellaan syklissä 3.

## 3.8. Poikkeukset

Poikkeuksia käytetään sovelluksessa hyväksi aina, kun metodi tai konstruktori ei suoriudu tehtävästään oletetulla tavalla. Poikkeuksia heitettäessä tekstiosaan laitetaan selkeä ilmoitus, jonka voi tulostaa myös sovelluksen käyttäjälle. Lohtu –sovelluksessa käytetään kolmea eri poikkeusta.

### 3.8.1. `java.lang.IllegalArgumentException`

Tämä poikkeus heitetään kun:

- metodi tai konstruktori joka tarvitsee argumentteja toteaa jonkun argumentin vialliseksi.
- esimerkki: Jos `ProjectDAO` –luokan `getProject(int id)` –metodilla yritetään hakea projektia, jonka `id` on negatiivinen.

### 3.8.2. `java.util.NoSuchElementException`

Tämä poikkeus heitetään kun:

- metodi tai konstruktori ei pysty luomaan haluttua oliota tai palauttamaan primitiivi tyyppin arvoa, koska niitä ei ole olemassa
- esimerkki: Jos `ProjectDAO` –luokan `getProject(int id)` –metodilla yritetään hakea projektia, jota ei ole olemassa.

### 3.8.3. `fi.helsinki.cs.lohtu.LohdutonException`

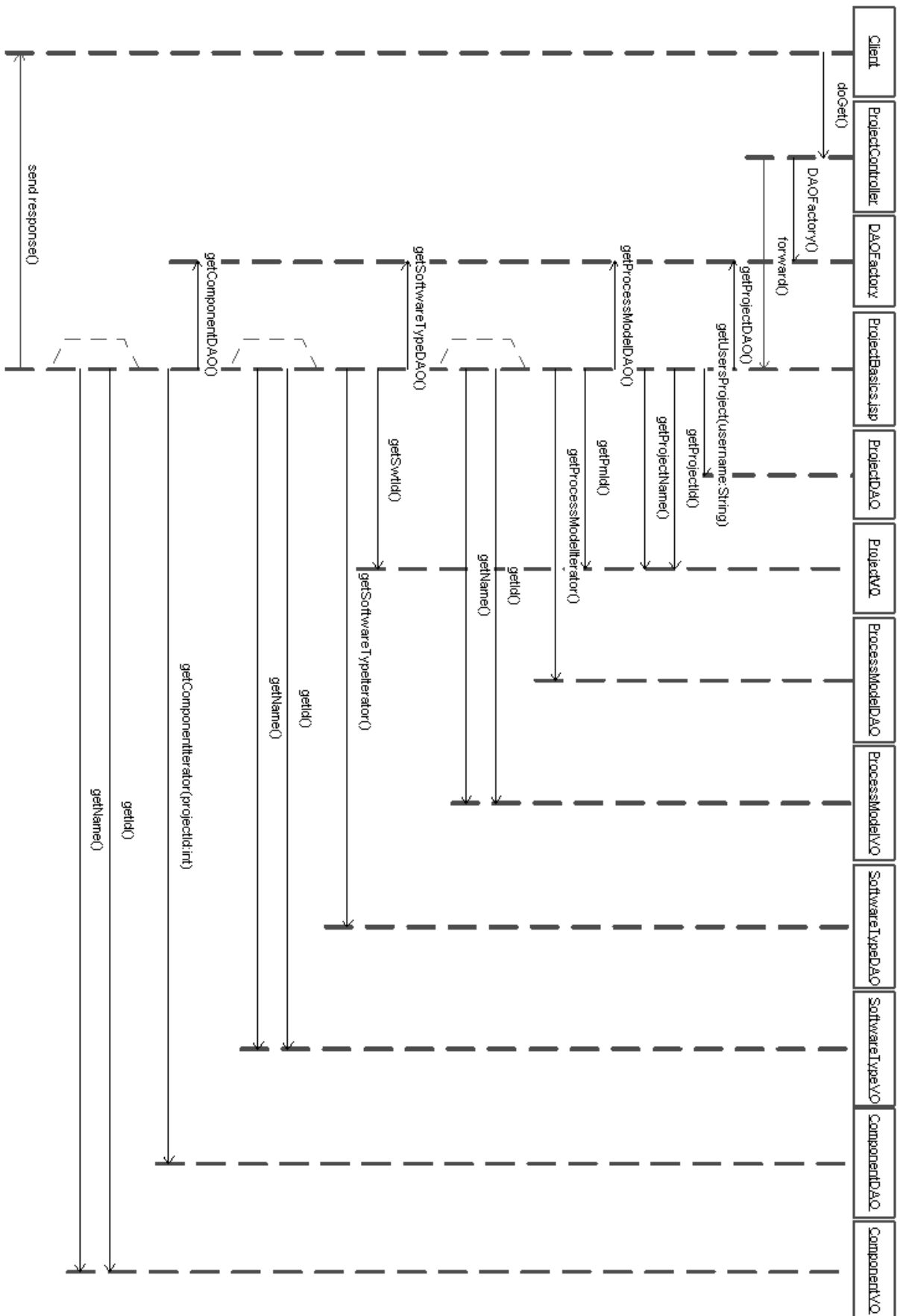
Tämä poikkeus heitetään kun:

- metodi tai konstruktori ei suoriudu tehtävästään, koska joku resurssi ei ole saatavilla tai ei toimi oletetusti. Käytetään myös kaikissa muissa virhetilanteissa.
- esimerkki: Jos metodi ei pysty käyttämään tietokantayhteyttä.

## 3.9. Sekvenssikaaviot

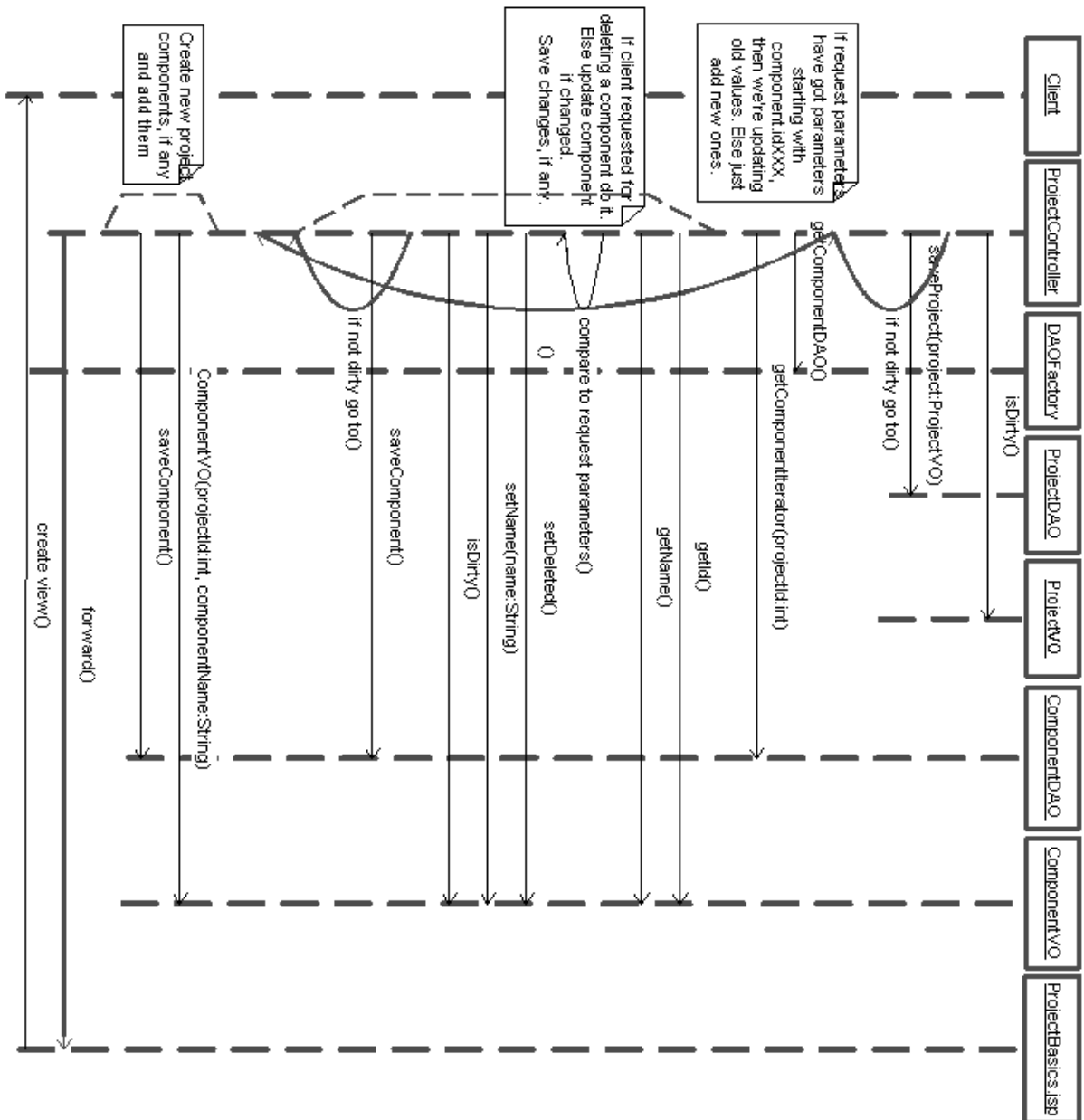
Seuraavissa kaavioissa on esitetty tyypillisiä toiminnankulkuja sovelluksen luokissa.

`CycleController`in kontrollinkulku on melko samanlaista lisäysten, poistojen ja muokkausten yhteydessä, joten toistuvat samanlaiset operaatiot on korvattu kommenteilla osassa sekvenssikaavioita.

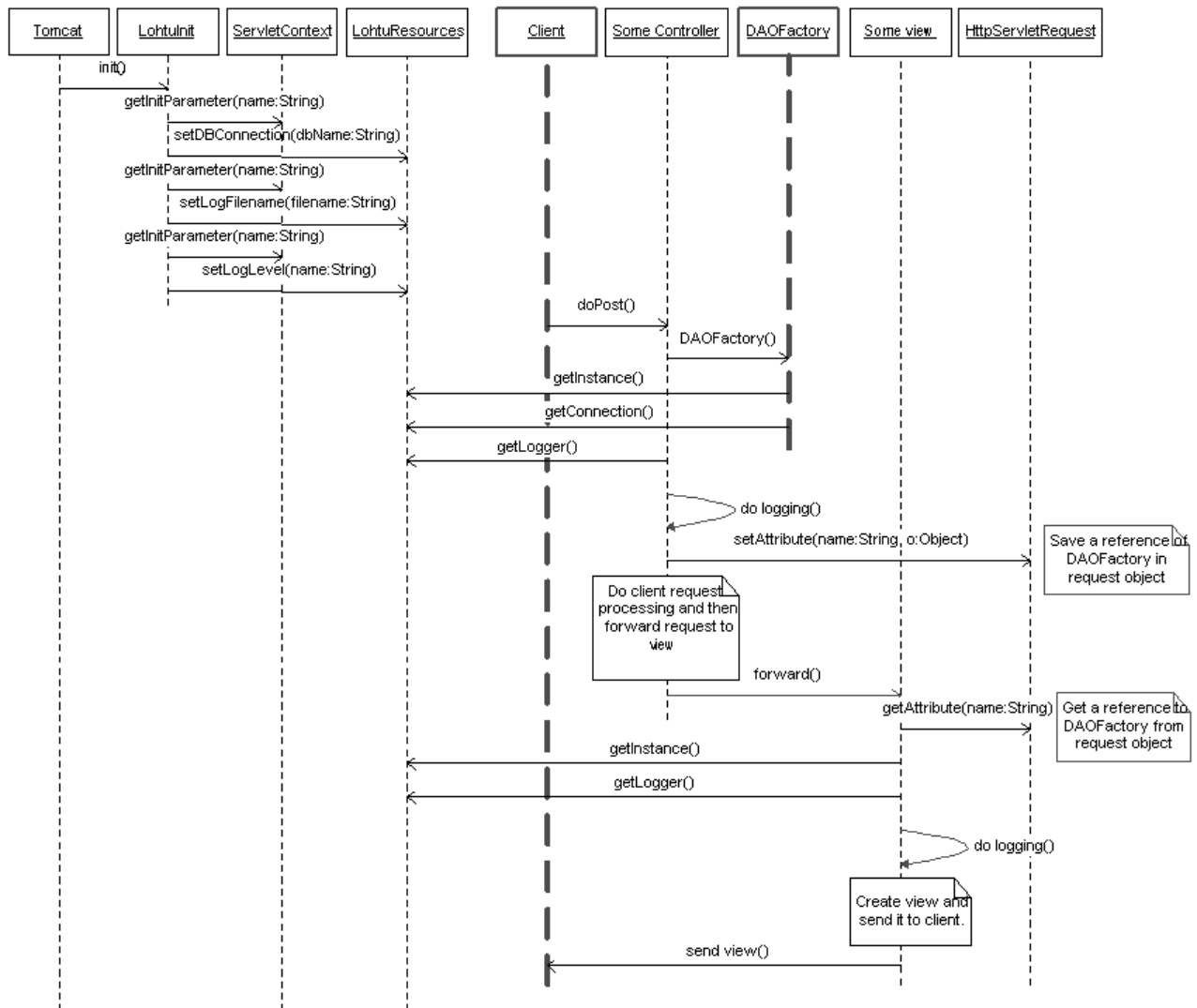


Kuva 16. Projektin perustiedot sivun näyttö.

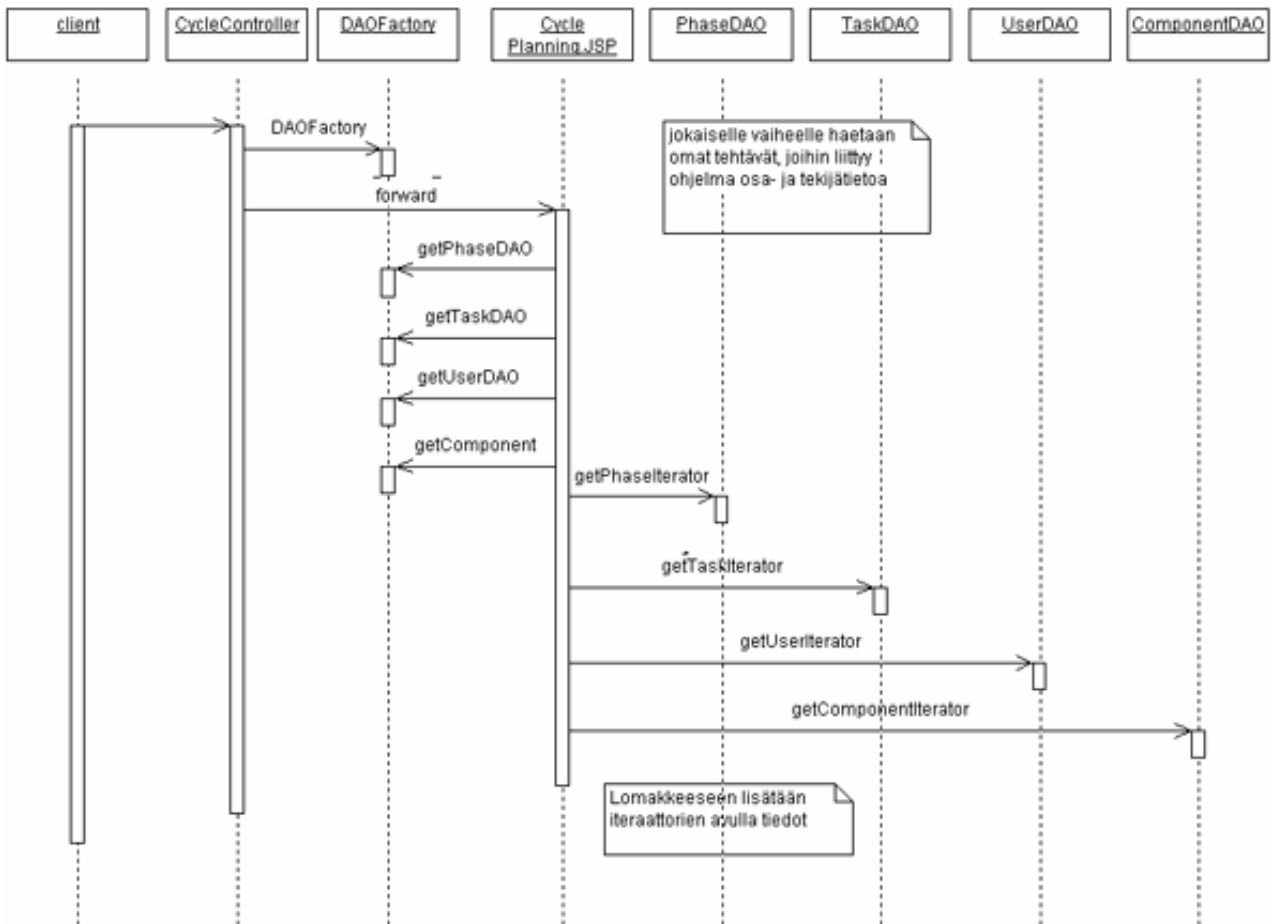




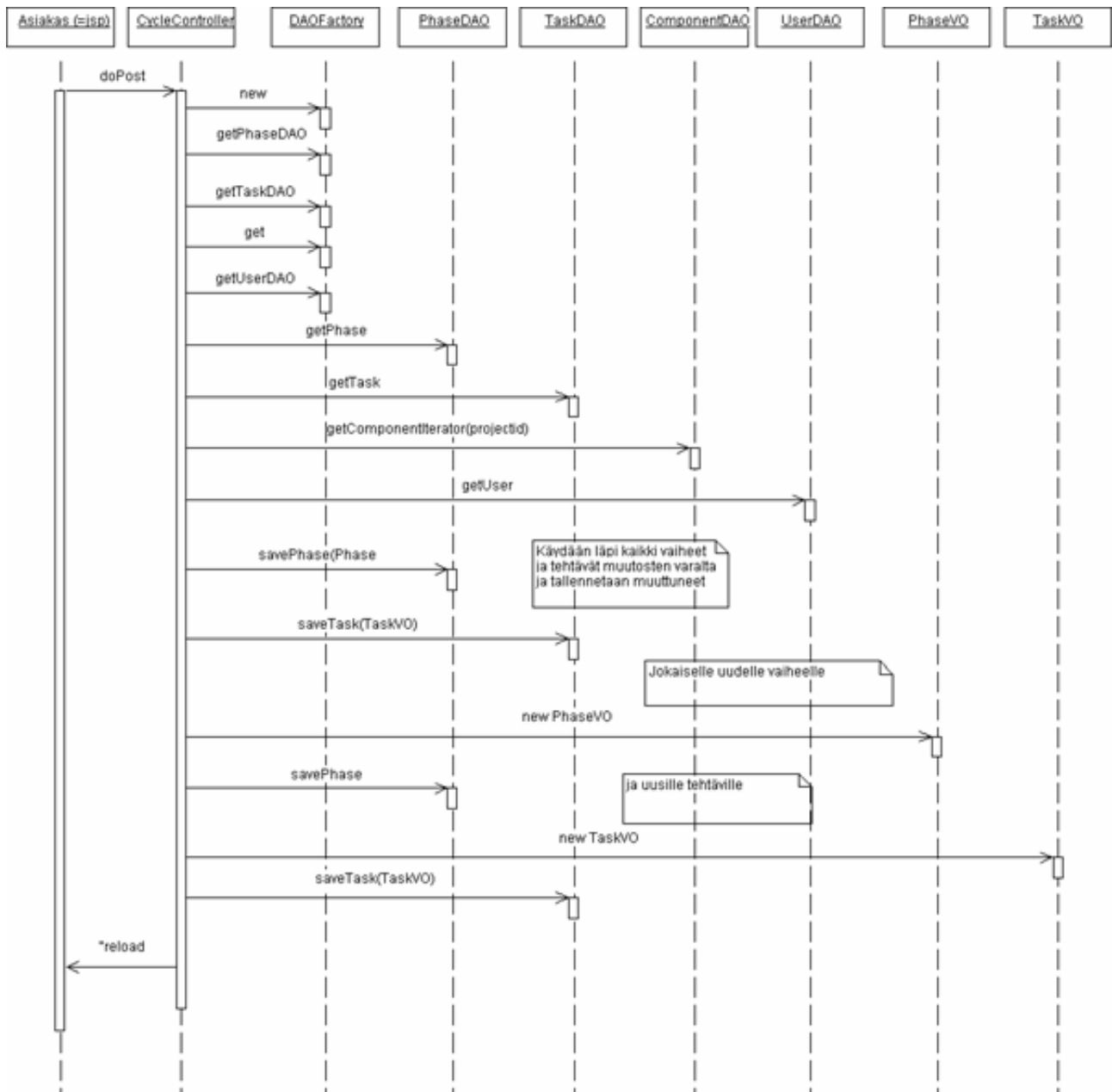
Kuva 17b. Projektin perustiedot sivun lomakedatan talletus.



Kuva 18. Lohtu- sovelluksen käynnistys ja DAOFactoryn käyttöesimerkki.

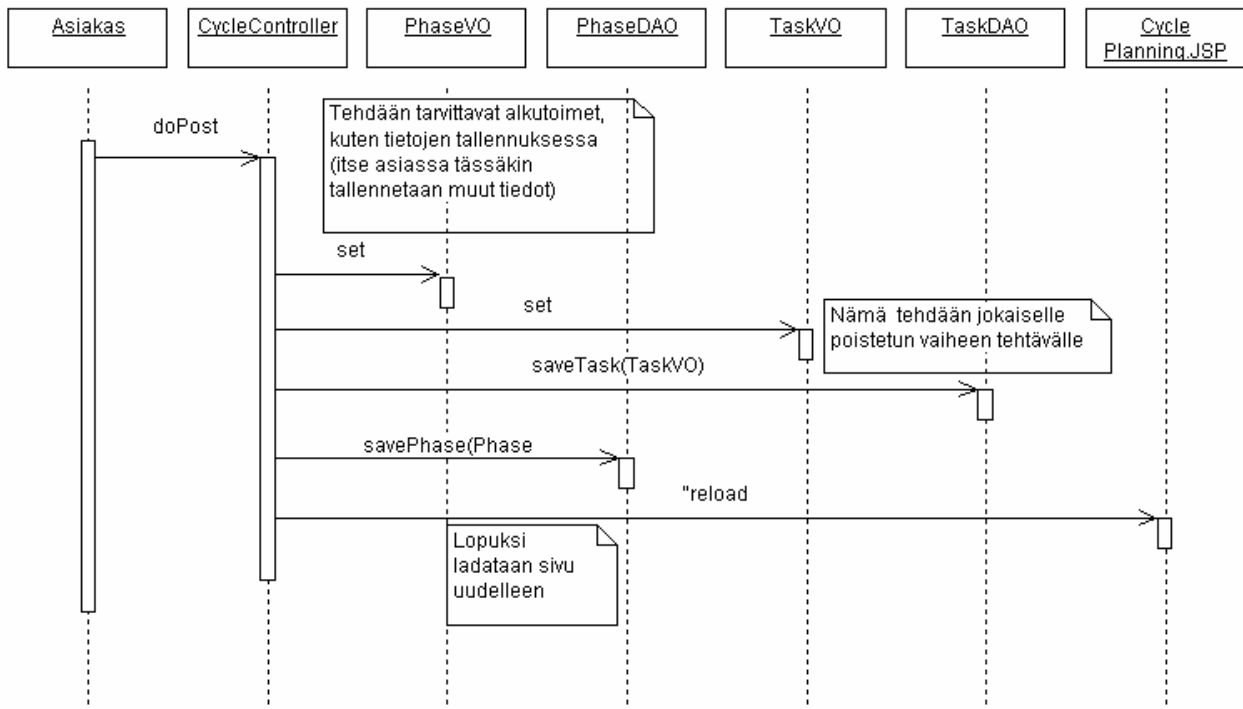


Kuva 19. CyclePlanning –sivun lataaminen. Tiedot lomakkeen kenttiin ja 'valikoihin' lisätään iteraattorien avulla.



Kuva 20. CyclePlanning –sivun tietojen päivittäminen.





Kuva 21. Projektin vaiheen poistaminen CyclePlanning –sivun kautta. Tehtävien poisto tapahtuu vastaavalla tavalla kuin vaiheiden toisto. Syklin poistoa ei ole toistaiseksi suunniteltu. Poistoja tehdessä suoritetaan myös kuvassa 20 kuvatut toimenpiteet.