

# **Vaatimusdokumentti**

Kohahdus

Helsinki 14.11.2006

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

## **Kurssi**

581260 Ohjelmistotuotantoprojekti (6 ov)

## **Projektiryhmä**

Taro Morimoto, Projektipäällikkö  
Tuomas Palmanto, Vaatimusmäärittelyvastaava  
Mikko Kinnunen, Suunnitteluvastaava  
Markus Kivilä, Koodivastaava  
Jari Inkinen, Testausvastaava  
Paula Kuosmanen, Dokumenttivastaava

## **Asiakas**

Teemu Kerola

## **Johtoryhmä**

Sanna Keskkioja

## **Kotisivu**

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

## **Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
0.1	15.9.2006	Ensimmäinen versio
0.2	19.9.2006	Lisätty käyttötapauskaavio ja sivusiirtymäkaavio
0.3	20.9.2006	Korjattu asiakkaalta saatujen lisätietojen perusteella
0.4	23.9.2006	Katselmusversio
0.5	25.9.2006	Asiakkaalle kommentoitavaksi toimitettava versio
0.6	29.9.2006	Ensimmäisen iteraation osalta jäädytetty versio
0.7	5.11.2006	Lisätty toisen iteraation vaatimuksia
1.0	14.11.2006	Lopullinen ja jäädytetty versio

# Sisältö

<b>1</b>	<b>Johdanto ja yleiskatsaus</b>	<b>1</b>
1.1	Toteutuksen jako kahteen iteraatioon . . . . .	1
<b>2</b>	<b>Käytettäviä termejä</b>	<b>2</b>
<b>3</b>	<b>Käyttäjryhmät ja järjestelmäarkkitehtuuri</b>	<b>2</b>
3.1	Käyttäjryhmät . . . . .	2
3.2	Sidokset vanhoihin projekteihin . . . . .	2
3.3	Järjestelmäarkkitehtuuri . . . . .	3
<b>4</b>	<b>Käyttjävaatimukset</b>	<b>3</b>
4.1	Toiminnalliset käyttjävaatimukset . . . . .	4
4.2	Opettajan käyttjävaatimukset . . . . .	5
4.3	Opiskelijan käyttjävaatimukset . . . . .	6
4.4	Ei-toiminnalliset käyttjävaatimukset . . . . .	6
4.5	Palautteet . . . . .	7
<b>5</b>	<b>Käyttötapaukset</b>	<b>8</b>
5.1	Opettajan käyttötapaukset . . . . .	8
5.2	Opiskelijan käyttötapaukset . . . . .	9
<b>6</b>	<b>Järjestelmävaatimukset</b>	<b>11</b>
6.1	Toiminnalliset vaatimukset . . . . .	11
6.1.1	Kirjautuminen ja tunnuksen luominen . . . . .	11
6.1.2	Tehtävätyypit . . . . .	12
6.1.3	Uuden tehtävän luonti . . . . .	13
6.1.4	Tehtävän määrittely . . . . .	13
6.1.5	Raportit . . . . .	14
6.1.6	Tehtävään vastaaminen . . . . .	15
6.2	Kriteerien järjestelmävaatimukset . . . . .	15
6.2.1	Hyväksymiskriteerit . . . . .	16
6.2.2	Laadulliset kriteerit . . . . .	17
6.2.3	Kriteerien käyttäminen . . . . .	18

6.2.4	Muita järjestelmävaatimuksia . . . . .	19
6.3	Titokoneen mahdollinen muokkaus . . . . .	20
6.4	Ei-toiminnalliset järjestelmävaatimukset . . . . .	20
6.5	Toimintaympäristövaatimukset . . . . .	21
6.6	Käyttäytyminen virhetilanteissa . . . . .	22

# Sisältö

## 1 Johdanto ja yleiskatsaus

Kohahdus on järjestelmä automaattisesti tarkastettavien TTK-91-konekielen harjoitustehtävien luomiseen ja ratkaisemiseen. Järjestelmä on tarkoitettu käytettäväksi opetuksen tukena, opetettaessa Tietokoneen toiminta -kurssia. Tietojenkäsittelytieteen opettajat voivat tehdä järjestelmään uusia tehtäviä ja määrittellä kuinka ne tarkastetaan automaattisesti. Tietokoneen toiminta -kurssin opiskelijat ja kurssin tehtävistä kiinnostuneet itseopiskelijat voivat ratkaista tehtäviä ja saada palautetta niiden onnistumisesta.

Painopiste projektissa on opettajan ja opiskelijan käyttöliittymillä. Käyttöliittymistä tehdään mahdollisimman selkeät ja helppokäyttöiset. Opettajan käyttöliittymän avulla määritellään tehtävät parametreineen ja kuinka opiskelijan ratkaisun oikeellisuus tarkistetaan, sekä nähdään statistiikkatietoja opiskelijoiden suorituksista. Opiskelijan käyttöliittymän avulla opiskelija valitsee tehtäviä, syöttää niiden ratkaisut ja saa palautetta vastauksensa oikeellisuudesta.

Projekti käyttää valmiina olevaa Tietokone-simulaattoria harjoitustehtävien ratkaisemiseen. Projekti rakennetaan osittain eAssari-kehikseen, joka sopii geneeriseen automaattisesti tarkistettavien tehtävien määrittelyyn ja toteutukseen. EAssari on kuitenkin niin keskeneräinen, että sitä ei voida käyttää tehtävänannon määrittelemässä laajuudessa, vaan pitäydytään lähinnä eAssarin tietokannan käyttämisessä.

Järjestelmä toteutetaan Java-kielellä ja järjestelmän käyttö vaatii, että selain tukee JavaScript-kieltä ja CSS-tyylitiedostoja. Järjestelmä toimii uusimmilla Firefox ja Internet Explorer-selaimilla.

Tässä dokumentissa esitellään käytettävät termit luvussa 2, sidosryhmät ja järjestelmä-arkkitehtuuri luvussa 3, käyttäjävaatimukset luvussa 4, käyttötapaukset luvussa 5 ja järjestelmävaatimukset luvussa 6.

### 1.1 Toteutuksen jako kahteen iteraatioon

Käytämme projektissa iteratiivista prosessimallia, jossa on kaksi iteraatiota eli toistoa.

Ensimmäisessä iteraatiossa toteutamme järjestelmän käyttöliittymän ja eAssaritietokantatoiminnallisuuden. Eli toteutamme käyttötapaukset Kt1, Kt2, Kt3, Kt5, Kt6, Kt8, Kt9, Kt10, Kt11 ja Kt13

Toisessa iteraatiossa toteutamme tehtävien suorittamisen ja tarkastamisen sekä jäljelle jääneet käyttöliittymätoiminnallisuudet, eli käyttötapaukset Kt4, Kt7, Kt12, Kt13 ja Kt14.

Mikäli kaikkia käyttötapauksia ei ehditä ensimmäisessä iteraatiossa toteuttaa, ne siirretään toiseen iteraatioon. Vastaavasti jos aikaa jää yli, mietitään jo alustavasti toiselle iteraatiolle suunniteltuja käyttötapauksia.

## 2 Käytettäviä termejä

TTK91=Auvo Häkkisen kehittämä ohjelmointikieli, joka läheisesti muistuttaa symbolista konekieltä.

KOKSI=Auvo Häkkisen kirjoittama konekielisimulaattori, joka toteuttaa TTK-91-kielen.

Järjestelmä=Projektimme tuotos, Kohahdus

Ohjelma=Opiskelijan kirjoittama TTK91-ohjelma, eli vastaus johonkin tehtävään

eAssari=Tietokantapohjainen ympäristö ohjelmallisesti tarkastettavien harjoitus- ja koe-tehtävien suorittamiseen

Titokone=Koski-nimisen Ohjelmistotuotantoprojektiryhmän vuonna 2004 rakentama järjestelmä konekielisten ohjelmien kääntämiseen ja suorittamiseen.

Koski=Vuoden 2004 Ohjelmistotuotantoprojekti joka rakensi konekielen simulaattorin ja debug-ympäristön, eli Titokoneen

Koskelo=Vuoden 2004 Ohjelmistotuotantoprojekti, joka integroi Titokoneen ja eAssari-kehityksen yhteen. Ratkaisusta ei tullut kuitenkaan käyttökelpoista, eikä sitä ole otettu käyttöön.

Kriteeri=Sääntö jonka mukaan tehtävän oikeellisuus tarkistetaan. Kriteereitä voi olla monta yhdelle tehtävälle.

Aihepiiri=Tehtävälle täytyy määritellä aihepiiri, johon tehtävä kuuluu.

## 3 Käyttäjärühmät ja järjestelmäarkkitehtuuri

### 3.1 Käyttäjärühmät

Konekielisen ohjelmoinnin harjoitteluympäristöä tulevat käyttämään Tietojenkäsittelytieteen laitoksen opettajat ja opiskelijat. Käyttäjärühmiä on näin ollen kaksi. Opettaja voi luoda uusia tehtäviä sekä muokata ja poistaa vanhoja. Lisäksi hän saa nähtäväkseen statistiikkatietoja opiskelijoiden suorituksista. Opiskelija saa nähtäväkseen kaikki tehtävät tai halutessaan vain tietyt valintakriteerit täyttävät tehtävät. Opiskelija voi ratkaista tehtäviä ja saa palautetta yrityksistään.

### 3.2 Sidokset vanhoihin projekteihin

Projektissa tullaan käyttämään Koski-ryhmän Titokonetta tehtävien kääntämiseen, sekä eAssari-kehityksen tietokantaa. Koskelon dokumentaatioon tutustutaan, ja voidaan ottaa oppia esimerkiksi tehtävän oikeellisuuden kriteereistä. Lisäksi Koskelon koodia saatetaan käyttää soveltuvilta osin.



## 4.1 Toiminnalliset käyttäjävaatimukset

### K1. Opettajan käyttöliittymän kieli

**Prioriteetti** : 1

**Kuvaus** Opettajan käyttöliittymä toteutetaan englannin kielellä.

### K2. Opettajan käyttöliittymän kieli

**Prioriteetti** : 3

**Kuvaus** : Opettajan käyttöliittymä toteutetaan myös suomen kielellä.

### K3. Opiskelijan käyttöliittymän kieli

**Prioriteetti** : 1

**Kuvaus** Opiskelijan käyttöliittymä on oltava englanniksi ja suomeksi.

### K4. Ohjelmointitehtävä

**Prioriteetti** : 1

**Kuvaus** Järjestelmä mahdollistaa ohjelmointitehtävän, jossa opiskelijaa pyydetään kirjoittamaan TTK-91-kielinen ohjelma, joka ratkaisee annetun tehtävän.

### K5. Täydennystehtävä

**Prioriteetti** : 1

**Nimi** : Täydennystehtävä

**Kuvaus** Järjestelmä mahdollistaa täydennystehtävän, jossa opiskelijalle näkyy valmiiksi osa ratkaisua josta puuttuu yksi kohta. Opiskelija täydentää puuttuvan kohdan.

### K6. Dynaaminen tehtävä

**Prioriteetti** : 4

**Kuvaus** Kolmatta mahdollista tehtävätyyppiä, dynaamista tehtävää ei toteuteta. Dynaaminen tehtävä on sellainen, joka generoidaan samalla kun opiskelija avaa tehtävän. Näin tehtävässä voi olla joka käynnistyskerralla erilaiset arvot annetuissa rajoissa. Tällaisen tehtäväntyyppin mahdollinen myöhempi lisääminen, esimerkiksi uuden Ohjelmistotuotantoryhmän toimesta, huomioidaan kuitenkin mahdollisuuksien mukaan, tehden tämä lisääminen mahdollisimman helpoksi.

### K7. Testisyötteen

**Prioriteetti** : 1

**Kuvaus** Vastauksien oikeellisuuden tarkastuksessa käytetään sekä näkyviä että salaisia syötteitä. Syötteen ovat aina kokonaislukuja. Salaisten syötteiden käytön tarkoituksena on estää mahdollisten oikopolkujen käyttäminen, esimerkiksi yritys tulostaa pelkkä vastaus ilman pyydettyä ohjelmarakennetta.



## 4.2 Opettajan käyttäjävaatimukset

### K8. Opettajan kirjautuminen

**Prioriteetti** : 1

**Kuvaus** : Opettaja voi kirjautua järjestelmään

### K9. Tehtävälistaus

**Prioriteetti** : 1

**Kuvaus** : Opettaja saa nähtäväkseen kaikki järjestelmässä olevat tehtävät, ja voi lajitella ne aihepiirin mukaan. Haun jälkeen Kohahdus antaa taulukon kyseisistä tehtävistä ja opiskelijoiden suorituksista.

### K10. Raporttien tulostaminen

**Prioriteetti** : 2

**Kuvaus** : Opettaja voi pyytää järjestelmältä raporttia opiskelijoiden suorituksista. Opettaja voi myös hakea opiskelijanumeron tai henkilötunnuksen perusteella yhden opiskelijan suorittamat tehtävät.

### K11. Uuden tehtävän lisääminen

**Prioriteetti** : 1

**Kuvaus** : Opettaja voi lisätä uuden tehtävän. Pakollisia tietoja uudelle tehtävälle ovat nimi, kieli ja aihepiiri.

### K12. Tehtävän muokkaaminen

**Prioriteetti** : 1

**Kuvaus** : Opettaja voi muokata tehtävää, ja tämä tehtävä voi olla myös toisen opettajan luoma. Muokkaamisesta tallennetaan muokkaaja ja muokkaus aika.

### K13. Tehtävän poistaminen

**Prioriteetti** : 1

**Kuvaus** : Opettaja voi poistaa tehtävän.

### K14. Tehtävän muokkaaminen uutena

**Prioriteetti** : 2

**Kuvaus** : Opettaja voi muokata uutena vanhaa tehtävää, eli ottaa pohjaksi vanhan tehtävän, tehdä siihen muutoksia, ja sitten tallentaa toisen nimisenä.

### 4.3 Opiskelijan käyttäjävaatimukset

**K15.** Rekisteröityminen

**Prioriteetti** : 1

**Kuvaus** : Opiskelija voi rekisteröityä järjestelmän käyttäjäksi.

**K16.** Kirjautuminen

**Prioriteetti** : 1

**Kuvaus** : Opiskelija voi kirjautua järjestelmään

**K17.** Tehtävien selailu

**Prioriteetti** : 1

**Kuvaus** : Opiskelija voi selaila järjestelmässä olevia tehtäviä, joko kaikkia tai lajiteltuna tehtävän aihepiirin tai sen mukaan onko tehtävä hyväksytysti ratkaistu.

**K18.** Suoritustiedot opiskelijalle

**Prioriteetti** : 1

**Kuvaus** : Opiskelija saa nähtäväkseen omat suoritustietonsa, joissa kerrotaan mitä tehtäviä opiskelija on suorittanut ja mitkä ovat kesken.

**K19.** Tehtävään vastaaminen

**Prioriteetti** : 1

**Kuvaus** : Opiskelija voi vastata valitsemaansa tehtävään ja saada palautetta vastauksestaan.

**K20.** Omien tietojen muuttaminen

**Prioriteetti** : 1

**Kuvaus** : Opiskelija voi muokata omia tietojaan.

### 4.4 Ei-toiminnalliset käyttäjävaatimukset

**K21.** Tehtävien kieli

**Prioriteetti** : 1

**Kuvaus** Tehtävät voivat olla joko suomeksi tai englanniksi

**K22.** Opettajan käyttöliittymän tulee olla selkeä ja helppokäyttöinen

**Prioriteetti** : 1

**Kuvaus** : Käyttöliittymän on oltava ehdottoman selkeä, ja hyvä käytettävyydeltään. Projektissa on käyttöliittymä priorisoitu tärkeimmäksi yksittäiseksi ominaisuudeksi. Käyttöliittymän tulee olla niin havainnollinen ja itsestäänselvä että sen käyttö onnistuu opettajalta vähäisellä käyttöohjeisiin tutustumisella.

**K23.** Opiskelijan käyttöliittymän tulee olla selkeä ja itsensä selittävä

**Prioriteetti** : 1

**Kuvaus** : Opiskelijan käyttöliittymän tulee olla havainnollinen.

**K24.** Palautteet käyttäjälle

**Prioriteetti** : 1

**Kuvaus** : Oletuspalautteiden on oltava tarkkoja ja yksityiskohtaisia. Opettaja voi muokata palautteita tehtäväkohtaisesti.

**K25.** Ei turhia siirtymiä sivujen välillä.

**Prioriteetti** : 1

**Kuvaus** : Sivusiirtymien määrä pidetään minimissä. Eli käytetään mahdollisimman vähän next-tyyppisiä painikkeita lomakkeissa.

**K26.** Opettajan käyttöohje

**Prioriteetti** : 1

**Kuvaus** : Opettajalle luodaan käyttöohje, joka kuvaa kuinka järjestelmää käytetään.

**K27.** Opiskelijan käyttöohje

**Prioriteetti** : 1

**Kuvaus** : Opiskelijalle ei luoda erillistä käyttöohjetta, mutta luodaan lyhyt WWW-sivu, jossa kerrotaan kuinka järjestelmää käytetään. Tälle sivulle luodaan linkki järjestelmään.

## 4.5 Palautteet

Palaute on keskeisessä osassa toteutettavassa ohjelmistossa ja järjestelmän tuleekin pystyä antamaan monipuolista palautetta opiskelijalle.

**P1.** Ohjelma kääntyy Titokoneessa, kaikki tehtävälle määritellyt kriteerit täyttyvät, ja ohjelma toimii oikein sekä normaalilla että salaisella syötteellä

**Palaute** : Annetaan opiskelijalle palaute jossa kerrotaan tehtävän onnistuneen

**P2.** Opiskelijan tekemän ohjelman kääntäminen ei onnistu

**Palaute :** Annetaan opiskelijalle palaute jossa kerrotaan että tehtävä ei onnistunut sekä missä kohtaa virhe tapahtui. Virhe-ilmoituksista annetaan ensimmäinen.

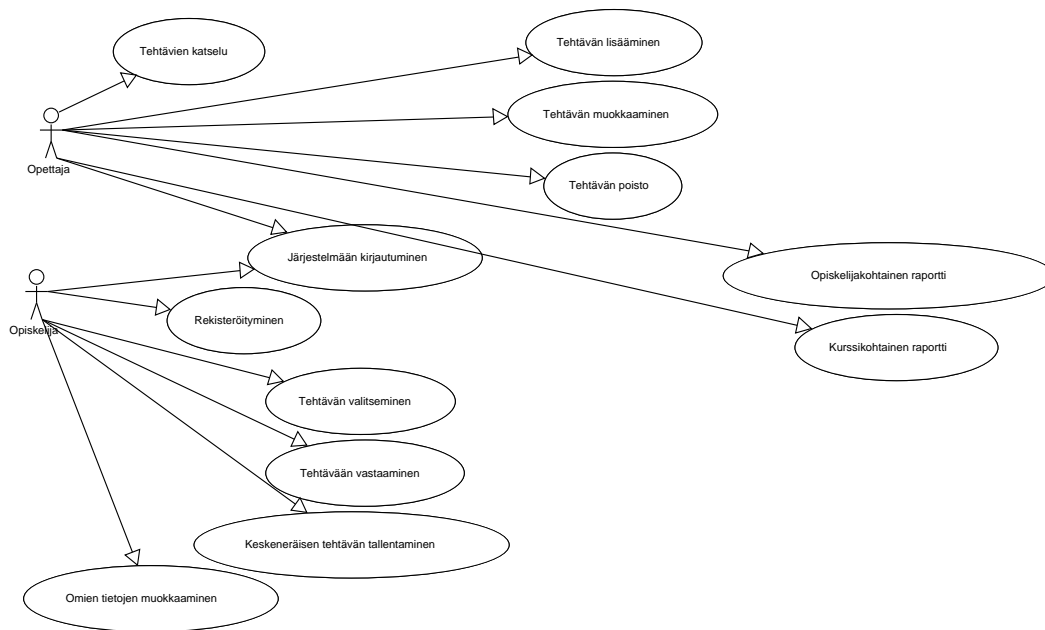
**P3.** Tehtävä ei toimi normaalilla syötteellä

**Palaute :** Ohjelma ei toiminut kun käytettiin tavallista syötettä. Lisäksi kerrotaan mahdollisimman tarkasti missä kohdassa virhe tapahtui.

**P4.** Tehtävä ei toimi salaisella syötteellä

**Palaute :** Ohjelma ei toiminut kun käytettiin salaista syötettä. Lisäksi kerrotaan mahdollisimman tarkasti missä kohdassa virhe tapahtui. Salaista syötettä ei näytetä opiskelijalle.

## 5 Käyttötapaukset



Kuva 2: Käyttötapauskaavio

### 5.1 Opettajan käyttötapaukset

**Kt1.** Järjestelmään kirjautuminen

Opettaja kirjautuu järjestelmään omalla tunnuksella ja salasanallaan (Kuva 3).

**Kt2.** Tehtävien selaaminen

Opettaja selaa järjestelmässä olevia tehtäviä. Tehtäviä voi lajitella ehtojen mukaan.

**Kt3.** Tehtävän lisäys

Opettaja lisää tehtävän järjestelmään määrittelemällä sille nimen, kielen, aihepiirin, tehtävänannon, näkyvät ja salaiset syötteet sekä erilaisia oikeellisuus- ja laatu-kriteerejä.

**Kt4.** Tehtävän muokkaaminen

Opettaja muokkaa mitä tahansa tehtävää, myös jonkin toisen opettajan määrittelemää. Tehtävän kaikki ominaisuudet ovat muokattavissa.

**Kt5.** Tehtävän tulostaminen

Opettaja pyytää luonti-/muokkausnäkyvässä tehtävää helposti tulostettavassa muodossa

**Kt6.** Tehtävän poisto

Opettaja poistaa minkä tahansa tehtävän, myös jonkin toisen opettajan määrittelemän tehtävän.

**Kt7.** Raporttien katselu

Opettaja pyytää raportin opiskelijoiden suorituksista. Raportti on helposti tulostettavassa muodossa. Raportteja on kahdenlaisia: opiskelijakohtaisia ja kurssikohtaisia.

**Kt8.** Uloskirjautuminen

Kun opettaja ei enää käytä järjestelmää, hän kirjautuu ulos.

## 5.2 Opiskelijan käyttötapaukset

**Kt9.** Rekisteröityminen

Opiskelijan pitää rekisteröityä ennen ensimmäistä käyttökertaa.

**Kt10.** Kirjautuminen

Rekisteröityään opiskelija kirjautuu järjestelmään (Kuva 3).

**Kt11.** Tehtävien katselu

Käyttäkseen järjestelmää opiskelijan on kirjauduttava sisään järjestelmään. Kirjaututtuaan opiskelija saa nähtäväkseen järjestelmässä olevat tehtävät, tai hänen valitseman ominaisuudet täyttävät tehtävät. Tehtävistä on jokaista tehtävää kohden tieto onko tehtävä a) tekemättä, b) tehty mutta hylätty, vai c) tehty ja hyväksytty.

**Kt12.** Tehtävien ratkaiseminen

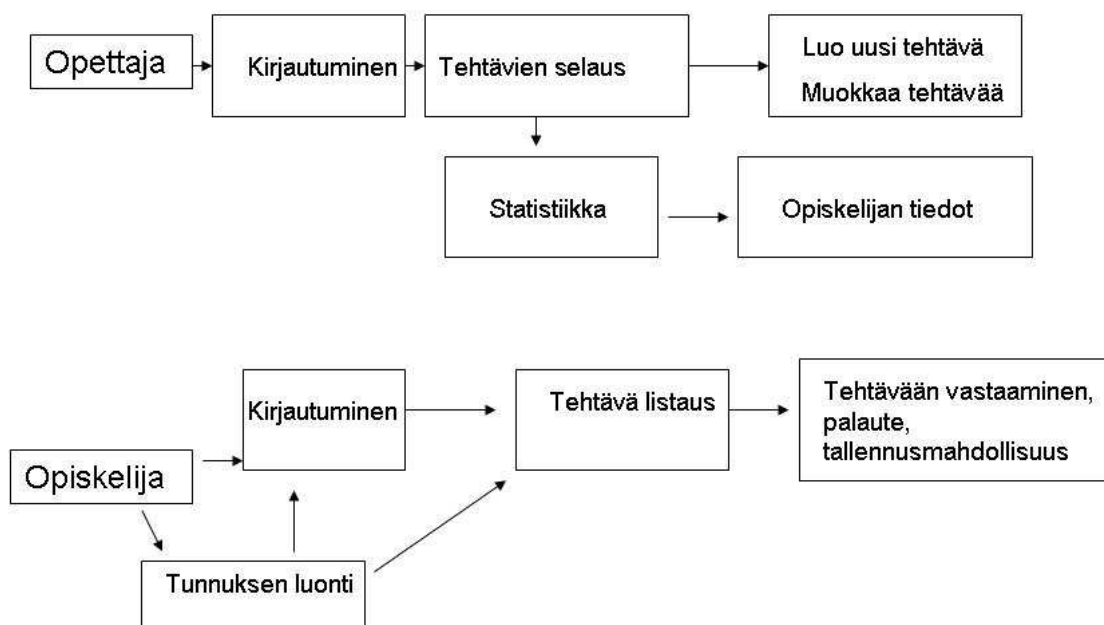
Valittuaan em. tehtävälisauksesta haluamansa tehtävän opiskelija yrittää ratkaista sen syöttämällä kentiin pyydetyn konekielisen ohjelman (ohjelmointitehtävä) tai sen osan (täydennystehtävä) sekä ohjelmalle annettavat syötteet ja lähettämällä sen järjestelmän tarkistettavaksi. Järjestelmä pisteyttää tehtävän ja antaa palautetta tehtävää luodessa/muokatessa annettujen kriteerien mukaan sekä mahdollisesti Titokone-simulaattorin ohjelman suorituksesta antamien tietojen mukaan. Opiskelija voi aina halutessaan palata tehtävälisaukseen ja yrittää jotakin toista tehtävää.

**Kt13.** Uloskirjautuminen

Kun opiskelija ei enää käytä järjestelmää, hän kirjautuu ulos.

**Kt14.** Omien tietojen katselu ja muokkaus

Opiskelija katselee tai muokkaa omia tietojaan.



Kuva 3: Kohahduksen tilasiirtymät

## 6 Järjestelmävaatimukset

Järjestelmävaatimukset ovat muotoa

**Prioriteetti :**

- 1 Toteutetaan varmasti
- 2 Toteutetaan mahdollisesti
- 3 Toteutetaan lopuksi jos aikaa on
- 4 Ei toteuteta

**Kuvaus :** Järjestelmävaatimuksen tarkempi kuvaus

### 6.1 Toiminnalliset vaatimukset

#### 6.1.1 Kirjautuminen ja tunnuksen luominen

**V1.** Opettajan tunnuksen luominen

**Prioriteetti :** 4

**Kuvaus :** Asiakkaan kanssa on sovittu että opettajat lisätään järjestelmään erillisillä SQL-lauseilla. Toimintoa ei siis toteuteta tämän projektin puitteissa.

**V2.** Opettajan kirjautuminen järjestelmään

**Prioriteetti :** 1

**Kuvaus :** Opettaja voi kirjautua järjestelmään.

**V3.** Opettajan oppilaiden poistamis-toiminto

**Prioriteetti :** 3

**Kuvaus :** Opettaja voi poistaa opiskelijoiden tunnuksia järjestelmästä

**V4.** Opiskelijan tunnuksen luominen

**Prioriteetti :** 1

**Kuvaus :** Opiskelija voi luoda tunnuksen järjestelmään. Opiskelija antaa tunnuksen luomisen yhteydessä nimensä, käyttäjätunnuksen, salasanan, opiskelijatyypin, sähköpostiosoitteensa ja opiskelijanumeronsa/henkilötunnuksensa. Opiskelijatyypillä tarkoitetaan tässä sitä minkälainen suhde opiskelijalla on järjestelmään, esimerkiksi itseopiskelija tai vuoden 2006 syksyn opiskelija.

**V5.** Opiskelijan kirjautuminen järjestelmään

**Prioriteetti :** 1

**Kuvaus** : Opiskelija voi kirjautua järjestelmään luomallaan tunnuksella.

**V6.** Opiskelijan tietojen muokkaaminen

**Prioriteetti** : 1

**Kuvaus** : Opiskelija voi muokata koko nimeään, salasanaansa, opiskelijatyyppiään ja sähköpostiosoitettaan.

**V7.** Salasanan lähetyk opiskelijalle sähköpostilla

**Prioriteetti** : 3

**Kuvaus** : Jos opiskelija on unohtanut salasanaansa, hänen on mahdollista pyytää järjestelmää lähettämään se sähköpostilla. Opiskelija antaa sähköpostiosoitteensa ja järjestelmä tarkastaa löytyykö sitä tietokannasta, ja jos löytyy järjestelmä lähettää kyseiseen osoitteeseen opiskelijan salasanan.

**V8.** Opettajan salasanan vaihtaminen

**Prioriteetti** : 3

**Kuvaus** : Opettaja voi vaihtaa salasanaansa.

### 6.1.2 Tehtävätyypit

**V9.** Ohjelmointitehtävä

**Prioriteetti** : 1

**Kuvaus** : Ohjelmointitehtävässä opiskelija kirjoittaa TTK-91 -kielisen koodin, jonka pitäisi ratkaista annettu tehtävä.

**V10.** Täydennystehtävä

**Prioriteetti** : 1

**Kuvaus** : Täydennystehtävässä on annettu valmiiksi osa vastauksesta, mutta se sisältää puuttuvan kohdan, johon opiskelijan tulee kirjoittaa oma vastauksensa. Tämä puuttuva kohta voi edeltää tehtävän annettua osuutta, olla sen keskellä tai sen jälkeen.

**V11.** Dynaaminen tehtävä

**Prioriteetti** : 4

**Kuvaus** : Dynaaminen tehtävä on tehtävä, jonka sisältö määräytyy vasta kun tehtävä avataan. Tätä tehtävätyyppiä ei toteuteta projektissa, mutta mahdollisuuksien mukaan huomioidaan, että sen myöhempi lisääminen olisi helppoa esimerkiksi uuden ohjelmistotuotantoryhmän toimesta.



### 6.1.3 Uuden tehtävän luonti

#### V12. Uuden tehtävän lisäys

**Prioriteetti** : 1

**Kuvaus** : Opettaja lisää uuden tehtävän ja määrittelee sille tarkistuskriteerit, sekä palautteet, jotka opiskelija saa eri tilanteissa. Lisäksi tehtävälle pitää määritellä aihepiiri. Tehtävän määrittelyn pitää mahtua yhdelle sivulle. Määrittelyn ei tarvitse mahtua ruutuun kerralla, eli määrittelysivua voi joutua skrollaamaan, mutta tehtävän määrittely ei saa sisältää sivusiirtymiä.

#### V13. Uuden tehtävän luonti vanhan tehtävän pohjalta

**Prioriteetti** : 3

**Kuvaus** : Opettaja voi luoda uuden tehtävän, siten että uuden tehtävän alkuarvot ovat asetettu vanhan tehtävän mukaisesti.

### 6.1.4 Tehtävän määrittely

#### V14. Kriteerien määrittely tehtävälle

**Prioriteetti** : 1

**Kuvaus** : Opettaja saa valita montako kriteeriä määrittelee uudelle tehtävälle, ja mitä nämä kriteerit ovat.

#### V15. Palautteen määrittely tehtävälle

**Prioriteetti** : 1

**Kuvaus** : Opettaja voi määritellä tehtävälle eri tilanteissa annettavan palautteen. Palautekentissä on valmiina oletuspalaute, jota opettaja voi käyttää jos ei halua määritellä tarkempaa palautetta.

#### V16. Tehtävän aihepiiri

**Prioriteetti** : 1

**Kuvaus** : Opettajan pitää määritellä tehtävälle aihepiirin johon tehtävä kuuluu. Esi-merkki: Tehtävällä on aihepiiri Aliohjelmatehtävät.

#### V17. Tarkistusmallin valinta

**Prioriteetti** : 1

**Kuvaus** : Opettaja valitsee tarkistetaanko tehtävä annettujen arvojen perusteella vai malliratkaisun perusteella. Oletuksena on annettujen arvojen perusteella. Jos opettaja valitsee malliratkaisun, web-sivulle ilmestyy kohta johon malliratkaisun voi kirjoittaa.

**V18. Opettajan tehtävälistaus****Prioriteetti** : 1**Kuvaus** : Opettajan tehtävälistauksessa esitetään tehtävien nimi, tyyppi, aihepiiri ja kieli.**V19. Opiskelijan tehtävälistaus****Prioriteetti** : 1**Kuvaus** : Opiskelijan tehtävälistauksessa esitetään tehtävien nimi, tyyppi, aihepiiri ja yrityskerrat, sekä tieto siitä onko hyväksytty, kesken tai ei vielä kertaakaan yritetty.**V20. Vanhan tehtävän muokkaus****Prioriteetti** : 1**Kuvaus** : Opettaja voi muokata vanhaa tehtävää. Kaikki tehtävät ovat kaikkien opettajien muokattavissa.**V21. Tehtävän poisto****Prioriteetti** : 1**Kuvaus** : Opettaja voi poistaa tehtäviä. Nämä tehtävät voivat olla opettajan itsensä tekemiä, tai jonkun toisen opettajan tekemiä.**6.1.5 Raportit****V22. Raportti opiskelijan suorituksista****Prioriteetti** : 1**Kuvaus** : Opettaja voi tulostaa raportin opiskelijan suorituksista.**V23. Raportti kurssikohtaisista suorituksista****Prioriteetti** : 1**Kuvaus** : Opettaja voi tulostaa raportin tietyllä kurssilla olevien opiskelijoiden suorituksista.**V24. Raportti opiskelijalle hänen suorittamistaan tehtävistä****Prioriteetti** : 4**Kuvaus** : Tällaista raporttia ei tehdä, vaan opiskelija voi printata vastaavat tiedot tehtävälistauksesta

### 6.1.6 Tehtävään vastaaminen

#### V25. Tehtävän valitseminen

**Prioriteetti** : 1

**Kuvaus** : Opiskelija selaa tarjolla olevia tehtäviä ja valitsee haluamansa. Tehtävät on luokiteltu aihepiirien mukaan. Tässä listassa erotellaan opiskelijan jo ratkaisemat tehtävät.

#### V26. Tehtävään vastaaminen

**Prioriteetti** : 1

**Kuvaus** : Opiskelija kirjoittaa valmiiksi ratkaisunsa tehtävään ja painaa "suorita-nappia". Järjestelmä antaa palautetta ratkaisun toimivuudesta, ja tallentaa vastauksen.

#### V27. Käännöksen palaute Titokoneesta

**Prioriteetti** : 1

**Kuvaus** : Jos käännös onnistuu, Titokone ei anna erillistä palautetta. Jos käännös epäonnistuu annetaan ensimmäinen käännöksessä löytynyt virheilmoitus. Mahdollisiin seuraaviin virheisiin ei reagoi.

#### V28. Suoritusstatistiikka

**Prioriteetti** : 1

**Kuvaus** : Käyttäjälle näytetään tilastotietoja ohjelman suorituksesta. Tässä tilastossa on suoritettujen käskyjen määrä, muistiviitteiden määrä, dataviitteiden määrä ja koodin koko.

#### V29. Opiskelijan tehtävätilanteen yhteenveto

**Prioriteetti** : 2

**Kuvaus** : Opiskelijan hyväksytysti ratkaistujen ja keskeneräisten tehtävien lukumäärät esitetään opiskelijan käyttöliittymän alareunassa.

## 6.2 Kriteerien järjestelmävaatimukset

#### V30. Tehtävien tarkastuskriteerit

**Prioriteetti** : 1

**Kuvaus** : Kriteerit määrittelevät miten vastauksien oikeellisuus tarkastetaan. Kriteerien mukaisesti päätetään onko vastaus hyväksyttävä, ja valitaan minkälainen palaute opiskelijalle annetaan. Kutakin kriteerityyppiä voi olla useita, ja niille voi jokaiselle antaa omat palautteensa. Esimerkki: Rekisterin R2 arvo ei ollut 4 tai rekisterin R3 arvo ei ollut 1.

### 6.2.1 Hyväksymiskriteerit

#### V31. Hyväksymiskriteerit

**Prioriteetti** : 1

**Kuvaus** : Hyväksymiskriteerit määrittelevät millä ehdoin tehtävän ratkaisu hyväksytään tai hylätään.

#### V32. Opiskelijan ohjelman maksimikoko (32:n bitin sanoina)

**Prioriteetti** : 1

**Kuvaus** : Vastauskoodille voidaan määritellä maksimikoko, jota pidempi vastaus ei saa olla.

#### V33. Stop-kriteeri

**Prioriteetti** : 1

**Kuvaus** : Ohjelmalle määritellään stop-kriteeri, eli suoritettujen käskyjen ehdoton maksimimäärä. Jos tämä määrä ylitetään, ohjelman suoritus lopetetaan ja näin estetään ikuiseen silmukkaan joutuminen. Stop-kriteerinä on oletusarvona suuri luku, joten opettajan stop-kriteerin määrittelemisen unohtumisenkaan ei voi johtaa ikuiseen silmukkaan.

#### V34. Rekisterin arvo

**Prioriteetti** : 1

**Kuvaus** : Kriteerinä voidaan käyttää valittujen rekistereiden arvoja ohjelman suorituksen päätyttyä. Esim. Onko rekisterin R3 arvo ohjelman suorituksen päätyttyä 4?

#### V35. Suoritettujen muistiviittausten lukumäärä

**Prioriteetti** : 2

**Kuvaus** : Kriteerinä voidaan käyttää suoritettujen muistiviittausten lukumäärä. Muistiviittausten tarkoitetaan koodi- ja dataviitteiden summaa.

#### V36. Suoritettujen dataviittausten määrä

**Prioriteetti** : 3

**Kuvaus** : Kriteerinä voidaan käyttää suoritettujen dataviittausten määrää.

#### V37. Suoritettujen käskyjen lukumäärä

**Prioriteetti** : 1

**Kuvaus** : Kriteerinä voi käyttää suoritettujen käskyjen lukumäärää.

#### V38. Data-alueen koko

**Prioriteetti** : 2

**Kuvaus** : Data-alue tarkoittaa globaalisti määriteltyjä tietorakenteita, ja kriteerillä voidaan rajoittaa sen kokoa.

**V39.** Pinon koko

**Prioriteetti** : 2

**Kuvaus** : Kriteerinä voidaan käyttää pinon maksimikokoa

**V40.** Joidenkin tiettyjen muuttujien (esim. X) arvot

**Prioriteetti** : 1

**Kuvaus** : Kriteerinä voi olla joidenkin tiettyjen muuttujien arvo ohjelman suorituksen jälkeen. Muuttujien nimissä ei huomioida isoja ja pieniä kirjaimia. Esimerkki: Käsitellään samalla tavalla muuttujia x ja X.

**V41.** Tulostettujen lukujen arvo

**Prioriteetti** : 1

**Kuvaus** : Kriteereinä voidaan käyttää sitä mitä opiskelijan ohjelma on tulostanut Titokoneen näytölle (CRT), esimerkiksi luku 4.

**V42.** Pakolliset käskyt

**Prioriteetti** : 1

**Kuvaus** : Ohjelmalle voidaan määritellä pakollisia käskyjä, joiden käyttöä vaaditaan hyväksytyyn suoritukseen.

**V43.** Kielletyt käskyt

**Prioriteetti** : 1

**Kuvaus** : Ohjelmalle voidaan määritellä käskyjä joita ei saa käyttää. Näiden käskyjen käyttö johtaa vastauksen hylkäämiseen. Esimerkki: MUL.

**V44.** Tiedostoon tulostettujen lukujen arvot

**Prioriteetti** : 4

**Kuvaus** : Ei toteuteta

## 6.2.2 Laadulliset kriteerit

**V45.** Laadulliset kriteerit

**Prioriteetti** : 1

**Kuvaus** : Laadullisten kriteerien perusteella voidaan antaa negatiivista tai positiivista palautetta, mutta ne eivät vaikuta tehtävän hyväksymiseen. Jos laadullinen kriteeri täyttyy, annetaan sille määritelty positiivinen palaute.

**V46.** Opiskelijan ohjelman ihannekoko (32:n bitin sanoina)

**Prioriteetti** : 1

**Kuvaus** : Ohjelmakoodille voidaan määritellä ihannekoko. Jos opiskelijan ohjelma on kooltaan pienempi tai yhtä suuri kuin ihannekoko, annetaan opiskelijalle kiitokset lyhyestä ratkaisusta.

**V47.** Data-alueen ihannekoko

**Prioriteetti** : 2

**Kuvaus** : Data-alue tarkoittaa globaalisti määriteltyjä tietorakenteita, ja data-alueen kokoa voidaan käyttää laadullisena kriteerinä.

**V48.** Pinon ihannekoko

**Prioriteetti** : 2

**Kuvaus** : Pinon ihannemaksimikokoa voidaan käyttää laadullisena kriteerinä.

**V49.** Suoritettujen käskyjen ihannelukumäärää

**Prioriteetti** : 1

**Kuvaus** : Suoritettujen käskyjen lukumäärää voidaan käyttää laadullisena kriteerinä.

**V50.** Suoritettujen muistiviitteiden ihannelukumäärä

**Prioriteetti** : 1

**Kuvaus** : Suoritettujen muistiviitteiden määrää voidaan käyttää laadullisena kriteerinä.

### **6.2.3 Kriteerien käyttäminen**

**V51.** Kriteerin ja vastauksen vertailu

**Prioriteetti** : 2

**Kuvaus** : Opiskelijan vastausta voidaan vertailla kriteeriin nähden operaatioilla =, !=, <, >, <= ja >=. Joidenkin kriteerien kohdalla on käytettävissä ainoastaan operaatio <=. Opettaja määrittelee vertailuoperaatiot tehtävää luodessaan.

**V52.** Kriteerien yhdistäminen tehtävän oikeellisuuden määrittämisessä

**Prioriteetti** : 1

**Kuvaus** : Kriteerit yhdistetään aina AND-tyyppisesti, eli kaikkien hyväksymiskriteerien pitää täytyä, jotta vastaus hyväksytään.

## 6.2.4 Muita järjestelmävaatimuksia

**V53.** Tehtävien aihepiirien luokittelu

**Prioriteetti** : 1

**Kuvaus** : Tehtävät voi luokitella pieniksi kokonaisuuksi. Esimerkki: Opettaja määrittelee aihepiiriin "alkeistehtävät", joka sisältää helppoja tehtäviä.

**V54.** Tehtäväaihepiirin lisäys

**Prioriteetti** : 1

**Kuvaus** : Opettaja voi lisätä uuden tehtäväaihepiiriin.

**V55.** Aihepiirin muokkaus

**Prioriteetti** : 1

**Kuvaus** : Opettaja voi muokata aihepiirien nimiä.

**V56.** Aihepiirin poisto

**Prioriteetti** : 1

**Kuvaus** : Opettaja voi poistaa aihepiiriin. Aihepiiriä ei voi poistaa jos on määriteltä tehtäviä, jotka kuuluvat aihepiiriin. Tällöin annetaan kehoitus poistaa kyseiset tehtävät ensin, tai siirtää kuuluvaksi toiseen aihepiiriin.

**V57.** Opiskelijan käyttöliittymän kieli

**Prioriteetti** : 1

**Kuvaus** : Opiskelijan käyttöliittymä on sekä suomeksi että englanniksi. Ensimmäisessä iteraatiossa käyttöliittymä toteutetaan englanniksi.

**V58.** Opettajan käyttöliittymä englanniksi

**Prioriteetti** : 1

**Kuvaus** : Opettajan käyttöliittymä tehdään englannin kielellä

**V59.** Opettajan käyttöliittymä suomeksi

**Prioriteetti** : 3

**Kuvaus** : Opettajan käyttöliittymä tehdään jos ehditään myös suomenkielisenä

### 6.3 Titokoneen mahdollinen muokkaus

**V60.** Symbolitaulu Titokoneesta

**Prioriteetti** : 1

**Kuvaus** : Titokoneen muokkaus siten että symbolitaulu saadaan tulostettua

**V61.** Makefile Titokoneelle

**Prioriteetti** : 2

**Kuvaus** : Luodaan Titokoneelle makefile. Makefile kääntää Titokoneen, luo Titokoneesta jar- ja zip-muotoiset levityspaketit sekä luo javadocin.

**V62.** Titokoneen päivitysohje

**Prioriteetti** : 3

**Kuvaus** : Päivitetään olemassaolevaa Titokoneen-päivitysohjetta.

### 6.4 Ei-toiminnalliset järjestelmävaatimukset

**V63.** Kuormitus

**Kuvaus** : Se kuinka monta käyttäjää järjestelmä pystyy käsittelemään kerralla, ja montako käyttäjää vuositason periytyy eAssarista ja Titokoneesta. Tarkat tiedot asiasta eivät ole tämän projektin ulottuvissa.

**V64.** Opettajan käyttöliittymän selkeys

**Kuvaus** : Käyttöliittymän tulee olla mahdollisimman selkeä ja helppokäyttöinen.

**V65.** Opiskelijan käyttöliittymän selkeys

**Kuvaus** : Käyttöliittymän tulee olla mahdollisimman selkeä ja helppokäyttöinen. Hyvä käyttöliittymä on sellainen joka on niin havainnollinen, että sen käyttöohjeisiin tarvitsee tutustua tuskin ollenkaan. Opiskelijan käyttöliittymästä tehdään sellainen, että sen käyttäminen ei vaadi ollenkaan erillistä käyttöohjetta.

**V66.** Tehtävän pisteytys

**Kuvaus** : Tehtävästä voi saada pisteitä joko 0 tai 100. 0 tarkoittaa että tehtävä ei ole onnistunut ja 100 että tehtävä on ratkaistu.

**V67.** Selkeä käyttöohje opettajalle

**Kuvaus** : Käyttöohjeen tulee olla kattava ja selkeä.

**V68.** Käyttöohje opiskelijalle



**Kuvaus** : Opiskelijalle ei luoda erillistä käyttöohjetta, mutta luodaan lyhyt WWW-sivu, jossa kerrotaan kuinka järjestelmää käytetään. Tälle sivulle luodaan linkki järjestelmään.

**V69.** Opettaja vastaa kriteerien mielekkyydestä

**Kuvaus** : Opettaja on vastuussa siitä, että tehtävänanto ja kriteerit vastaavat toisiinsa ja ovat muutenkin järkeviä. Järjestelmä ei tee tarkistuksia tehtävien mielekkyydestä.

## 6.5 Toimintaympäristövaatimukset

**V70.** eAssari-yhteensopivuus

**Kuvaus** : Järjestelmän tulee toimia yhteistyössä eAssari-kehiksen tietokannan kanssa

**V71.** Titokone-yhteensopivuus

**Kuvaus** : Järjestelmän tulee toimia yhteistyössä Titokoneen kanssa

**V72.** Sijainti

**Kuvaus** : Järjestelmä sijaitsee Helsingin Yliopiston palvelimella, ja se toimii Tomcat-alustalla.

**V73.** Tuetut selaimet

**Kuvaus** : Kohahduksen taataan toimivan uudemmilla Firefox (1.5.x) ja Internet Explorer (6 ja 7) -selaimilla ja järjestelmän käyttö ei vaadi erillisten laajennustan asentamista. Hyvin todennäköisesti järjestelmä toimii myös ko. selainten vanhemmilla versioilla ja muilla selaimilla, mutta ainoastaan uusimmat Firefox ja Internet Explorer taataan toimivaksi.

**V74.** JavaScript

**Kuvaus** : Järjestelmä vaatii JavaScriptiä toimiakseen. Jos JavaScript-tuki on selaimesta pois päältä, tästä huomautetaan kirjautumissivulle tultaessa. Itse kirjautuminen ei onnistu ilman JavaScriptiä, vaan kirjautumisyrittäyksestä ilman JavaScriptiä annetaan kehoitus laittaa JavaScript päälle.

**V75.** Evästeet

**Kuvaus** : Järjestelmä käyttää evästeitä (cookie) hyväkseen. Jos evästeet ovat pois käytöstä, tästä huomautetaan kirjautumissivulla. Järjestelmää on kuitenkin mahdollista käyttää siten että evästeet ovat poissa päältä.

## 6.6 Käyttäytyminen virhetilanteissa

Järjestelmän käyttö vaatii sisäänkirjautumista. Sekä opettajan että opiskelijan on kirjaututtava sisään järjestelmään pystyäkseen käyttämään sitä.

Mahdolliset virhetilanteet ja niihin reagoiminen:

**E1.** Väärä tunnus kirjautumisen yhteydessä

Annetaan virheilmoitus "Väärä tunnus tai salasana, yritä uudestaan"

**E2.** Väärä salasana kirjautumisen yhteydessä

Annetaan virheilmoitus "Väärä tunnus tai salasana, yritä uudestaan"

**E3.** Väärän tyyppinen syöte tehtävän luonnissa

Jos opettaja antaa vääräntyyppisen syötteen tehtävän luonnissa, tästä annetaan virheilmoitus. Esimerkki: Opettaja yrittää antaa rekisterille R1 arvon "k".

**E4.** Väärän tyyppinen syöte opiskelijan vastauksessa

Jos opiskelija antaa vääräntyyppisen syötteen kenttään, tästä annetaan virheilmoitus. Esimerkki: opiskelija kirjoittaa syöte-kenttään kirjaimia, vaikka kyseinen kenttä saa sisältää vain kokonaislukuja (kaikkien Titokoneen syötteiden tulee olla kokonaislukuja).

**E5.** Verkkoyhteys ei toimi

Annetaan virheilmoitus "Verkkoyhteys ei toimi"

**E6.** Tietokantavirhe

Jos tietokantayhteys pätkii annetaan virheilmoitus "Ongelma tietokantayhteydessä".