

# EI ENÄÄN MUUTOKSIA TÄHÄN!

Muutokset suoraan Jaakolle

## Peppi - Koulutuksen suunnittelijan ja opettajan palvelut

### Tekninen vaatimusmäärittely

#### [1 Johdanto](#)

#### [2 Termit ja määritelmät](#)

#### [3 Arkkitehtuuriperiaatteet](#)

##### [3.1 Yleiset periaatteet ja vaatimukset](#)

##### [3.2 Noudatettavat standardit](#)

##### [3.3 Autorisointi ja auktorisointi](#)

##### [3.4 Rajapinnat ja liittymät muihin järjestelmiin](#)

###### [3.4.1 Henkilötiedot](#)

#### [4 Toiminnalliset vaatimukset](#)

##### [4.1 Opetussuunnittelu](#)

##### [4.2 Vuosisuunnittelu](#)

###### [4.2.1 Tietojen syöttäminen](#)

###### [4.2.2 Tietojen esittäminen](#)

###### [4.2.3 Raportointi](#)

###### [4.2.4 Puuttuvia palveluja](#)

Metropolia Ammattikorkeakoulu  
Tampereen Ammattikorkeakoulu

Peppi - Koulutuksen suunnittelijan ja  
opettajan palvelut

Versio 0.3

Tekninen vaatimusmäärittely

[4.3 Lukujärjestysuunnittelu](#)

[4.4 Opettajapalvelut](#)

[4.5 Raportointi-, tieto- ja seurantapalvelut](#)

[5 Ei-toiminnalliset vaatimukset](#)

[5.1 Tietojen arkistointi](#)

[5.2 Historiatietojen säilyttäminen](#)

[5.3 Tallennustietojen säilyttäminen](#)

[5.4 Avoimet rajapinnat](#)

[Liitteet](#)

<b>Versio</b>	<b>Päiväys</b>	<b>Tekijä</b>	<b>Selite</b>
0.1	9.12.2010	Jaakko Rannila	Runko
0.2	13.12.2010	Projektiryhmä	1. päivän tuotos
0.3	14.12.2010	Projektiryhmä	2. päivän tuotos

[1 Johdanto](#) (Rami, Ismo, Jaakko, Juhani, Tomi, Osmo, Eija)

[2 Termit ja määritelmät](#) (kaikki)

[3 Arkkitehtuuriperiaatteet](#) (Rami, Tomi ja Simo, Juhani, Jaakko, Ismo, Markku Pekonen)

[3.1 Yleiset periaatteet ja vaatimukset](#) (Rami, Tomi ja Simo, Juhani, Jaakko, Ismo, Markku Pekonen)

[3.2 Noudatettavat standardit](#) Rami, Tomi ja Simo, Juhani, Jaakko, Ismo, Markku Pekonen

[3.3 Rajapinnat ja liittymät muihin järjestelmiin](#) Rami, Tomi ja Simo, Juhani, Jaakko, Ismo, Markku Pekonen

[4 Toiminnalliset vaatimukset](#) (Eija, Virve, Osmo Rami, Tomi ja Simo, Juhani, Jaakko, Ismo, Markku Pekonen)

[5 Ei-toiminnalliset vaatimukset](#) (Eija, Virve, Osmo Rami, Tomi ja Simo, Juhani, Jaakko, Ismo, Markku Pekonen)

[Liitteet](#)

# 1 Johdanto

Tässä dokumentissa kuvaillaan koulutuksen suunnittelijan ja opettajan palvelukokonaisuuden tekniset vaatimukset.

## 2 Termit ja määritelmät

Termi	Selite
Palvelu	Joukko kiinteästi yhteenliittyviä toimintoja, esim. opintosuunnitelman hakupalvelu
Palvelualusta	Tietojärjestelmä joka liittää palvelut yhteen.
Järjestelmä	Palvelualusta ja siihen kiinteästi liittyvät muut tietojärjestelmät

Termi	Selite
ESB	Enterprise Service Bus, palveluväylä

Versio 0.3

Tekninen vaatimusmäärittely

JB1	Java Business integration, JSR 208 ja JSR 312-spesifikaatioissa kuvattu standardi helpottamaan palveluiden ja tietojärjestelmien integrointia
SE	JB1-spesifikaatiossa mainittu Service Engine -komponentti, joka voidaan liittää palveluväylään tietoa prosessoivaksi komponentiksi, esim. BPEL-komponentti, XSLT-komponentti
BC	JB1-spesifikaatiossa mainittu Binding Component -komponentti, jonka avulla palveluväylä liitetään tietyn protokollan avulla rajapintaan
BPEL	Business Process Execution Language, liiketoimintaprosessien kuvaamiseen tarkoitettu XML-pohjainen kieli
SCA	Service Component Architecture, palveluiden toteuttamiseen ja koostamiseen tarkoitettu suhteellisen uusi standardi
OSGi	Standardi modulaarista alustaa varten
WS	Web Service, verkossa oleva palvelun tarjoama rajapinta
SOAP	XML-pohjainen kieli ja protokolla
EJB	Enterprise Service Bean, transaktionaalinen komponenttipohjainen tekniikka palveluiden toteuttamiseen
JSON	
REST/RESTful rajapinta	

### 3 Arkkitehtuuriperiaatteet

Tavoitteena on uudistaa korkeakoulujen sovellusarkkitehtuuria, joka on nykyisin muodostunut joukosta monoliittisia järjestelmiä. Monoliittisille järjestelmille on tyypillistä, että ohjelman toiminnallisuutta voidaan hyödyntää vain järjestelmän oman käyttöliittymän kautta. Jos

Versio 0.3

Tekninen vaatimusmäärittely

monoliittisen järjestelmän tietoa hyödyntää muissa järjestelmissä, edellyttää se haastavia ja kalliita integraatio projekteja, mitkä aiheuttavat usein samojen tietojen päällekkäistä tallentamista useisiin järjestelmiin, mikä aiheuttaa helposti myös tiedon eheydellisiä ongelmia.

Uudistuksen ydintavoitteena teknologiamielessä on nimenomaan saada aikaiseksi sen tapainen sovellusarkkitehtuuri, jossa ei muodosteta monoliittisia järjestelmiä vaan atomisia palveluita, joita koostamalla saadaan aikaan sähköisiä palveluita. Tällöin palvelut voidaan kohdistaa paremmin oikeille käyttäjille ja tietojen integroimiseen on oma alusta, joka nimenomaan siihen tarkoitukseen kehitetty.

Arkkitehtuuriperiaatteena pidetään seuraavia periaatteita. Kehyksen johon palveluita rakennetaan ja integroidaan tulee olla avoimen lähdekoodin ratkaisu, jolloin tarvittaessa voidaan mennä lähdekooditasolle. Tämä siitä syystä, että tuotettavat palvelut ja integraatiot voivat osoittautua haastaviksi ja tällöin tärkeää, että pystytään tutkimaan kehyksen toimintalogiikkaa. Myös sen takia, että lähdekoodi antaa mahdollisuuden löytää ongelmien todelliset syyt. Tämän lisäksi kehykselle pitää löytyä riittävää tukea tarvittaessa ja kehyksen käytöstä pitää löytyä referenssejä. Lisäksi valittavan kehyksen tulee selviytyä Peppi projektissa toteutetuista Proof of Concept (POC) / Teknisen alustan koestusvaiheesta.

## 3.1 Yleiset periaatteet ja vaatimukset

Peppi toteutetaan palvelupohjaisena järjestelmänä, jonka sovellusintegraatiot ja businesslogiikka toteutetaan selkeästi erotettavassa palvelukerroksessa ja jonka käyttöliittymä logiikka rakennetaan tämän palvelukerroksen päälle. Tarkoitus on myös, että kaiken mitä voidaan tehdä käyttöliittymällä, voidaan tehdä suoraan palvelukerroksesta. Näin käyttöliittymä voidaan täysin vaihtaa menettämättä mitään alkuperäisen käyttöliittymän toiminnallisuutta. Tämä on hyödyllistä kun koostamme uusia käyttöliittymiä monesta eri palvelusta, missä Pepin palvelukerros on vain yksi osa palveluluista.

Palvelukerroksen palvelut tulee toteuttaa sillä tasolla, että käyttöliittymäkerroksessa ei tarvitse enää käyttää transaktioita. Näin saadaan aikaiseksi selkeä työnjako palvelukerroksen ja käyttöliittymän välille. Käyttöliittymä voidaan rakentaa yksinkertaisemmaksi ja sellaisten rajapintojen päälle, mitkä eivät tue transaktionaalisuutta.

Palvelukerroksen tulee tarjota mahdollisuus asynkroniseen viestinvälitykseen siten, että sanoma välitetään määränpähän vähintään kerran.

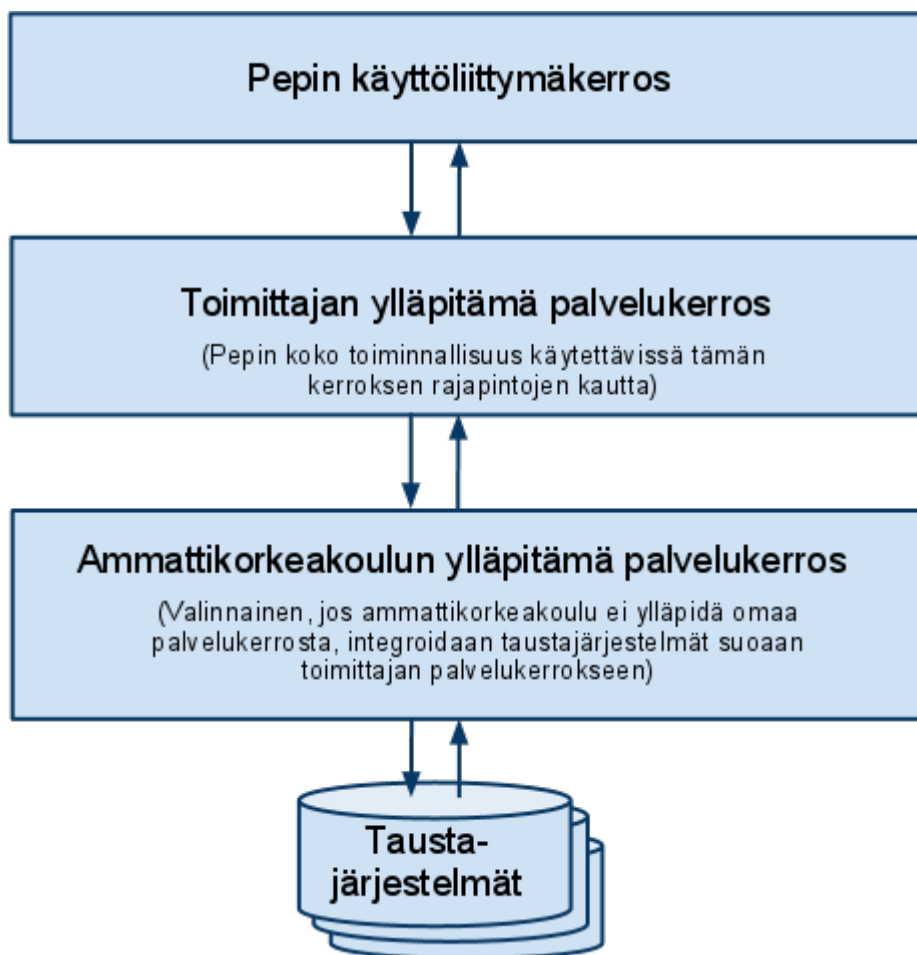
Palvelut tulee tehdä tilattomiksi. Tilan tallennukseen liittyvät asiat hoidetaan käyttöliittymässä, workflowssa tai prosessi yms. moottorissa. Näin palvelut saadaan yksinkertaisemmiksi, vikasietoisemmiksi ja skaalautuvammiksi.

Ammattikorkeakoulun kannattaa tarjota rajapinnat taustajärjestelmiin palveluina, jotta riippuvuudet taustajärjestelmän rajapinnan yksityiskohdista jää ammattikorkeakoulun oman

Versio 0.3

Tekninen vaatimusmäärittely

sovelluskehityksen vastuulle vähentäen toimittajariippuvuutta. Ammattikorkeakoulu voi toteuttaa palvelut esim. oman palvelukerroksen avulla, jolloin ammattikorkeakoulun palvelukerroskokonaisuus jakautuu kahteen osaan: Siihen mikä on toimittajan ylläpitämä ja siihen mikä on oman ylläpidon piirissä. Valitun tai vallittujen palvelukerroksien olisi tästä syystä tarkoituksenmukaista pystyä integroitumaan löyhästi yhteen esim, jollain viestinvälitykseen liittyvällä protokollalla.



## 3.2 Noudatettavat standardit

Toteutettavan teknisen viitekehityksen pitää tukea yleisesti hyväksytyjä standardeja joihin voidaan tukeutua. (erityisesti: SOAP, WSDL, JAX-WS, Rest. Eduksi katsotaan, jos tukee yleisiä standardeja kuten JBI, OSGi, SCA)

## 3.3 Autorisointi ja auktorisointi

Koska molemmilla ammattikorkeakouluilla on omat taustajärjestelmät identiteetin hallintaan (tunnukset, ryhmät, roolit, organisaatorakenne), autorisointiin ja auktorisointiin tulee tehdä oma

Versio 0.3

Tekninen vaatimusmäärittely

abstrakti palvelukokonaisuus. Tämän palvelukokonaisuuden rajapinnoista tehdään tarvittaessa omat toteutukset molemmille ammattikorkeakouluille. Mikään muu toteutettavan järjestelmän palvelu ei saa sisältää suoraa riippuvuutta ammattikorkeakoulun identiteetin hallinnan taustajärjestelmiin.

Autorisointi täytyy tehdä jo palvelutasolla. Sitä ei voi jättää pelkästään näkymäkerrokselle. Tämä keventää näkymäkerroksen taakkaa sekä vähentää autorisoinnin moninkertaista toteuttamista.

Tarvittavia palveluita ovat ainakin:

- identiteettipalvelu
- ryhmäpalvelu
- roolipalvelu
- oikeuspalvelu

## 3.4 Rajapinnat ja liittymät muihin järjestelmiin

Yleisinä periaatteille integroitaville taustajärjestelmille asetamme, että niiden on tuettava

1) Ohjelmallisia rajapintoja kuten esim. soap, rest, jms, activeMQ ynnä muita vastaavanlaisia rajapintoja. Esimerkkeinä rajapinnoista mitä emme tämän nojalla tue on:

- a) Tekstitiedostoihin perustuvia rajapintoja, koska niillä ei voi muodostaa pyyntö/vastaus-tyyppisiä palveluita eikä toteuttaa virheenkäsittelyä.
- b) Suoraan tietokantatauluihin liittyviä integraatiota, koska ne luovat riippuvuuksia fyysisiin tauluihin. Integraatio on kuitenkin, mikäli muita vaihtoehtoja ei ole, mahdollista tehdä tietokantanäkymien avulla, mikäli rajapinta ei edellytä kantaan kirjoittamista. Jos rajapinta edellyttää tietokantaan kirjoittamista, täytyy rajapinnan toteuttamiseen käyttää tietokantaproseduuria.

2) Hallittua virheenkäsittelyä, mikä edellyttää:

- a) pyytävä osapuoli saa tiedon virheestä siten, että syy voidaan jäljittää tiettyyn tapahtumaan ja/tai korruptoituneeseen tietoon.
- b) Virheviestin käsitteleminen ei vaadi "parsimista" esim. merkkijonon monimutkaista tulkitsemista.
- c) Virheviesti ja kuvaus erotetaan selkeästi siten, että virheviesti on aina sama liittyen samanlaiseen virheeseen, mutta selitteen teksti voi vaihdella.

### 3.4.1 Henkilötiedot

Järjestelmän tulee pystyä hakea tiedot ulkopuolisista järjestelmistä. Henkilötietojen lähdejärjestelmänä käytetään organisaatioiden IAM- tai IDM-järjestelmiä.

Metropoliassa IAM-järjestelmänä on Amme. Ammeen ja Pepin integraatiossa hyödynnetään Tuubi2 projektissa syntynyttä skeemaa, joka tehtiin Tuubin ja Ammeen integraatiota varten.



Versio 0.3

Tekninen vaatimusmäärittely

Tamkin osalta järjestelmä on muutoksen alla, joten vielä ei voida määritellä tapahtuuko haku yhdestä (IDM tai vastaava) tai useammasta järjestelmästä (LDAP ja OHA yms).

Henkilöihin liittyvää tietoa voidaan laajentaa oman profiilin avulla, joka tallentuu Peppi kokonaisuuteen (ks. toiminnallinen määrittely 6.1.3).

### **3.4.2 Organisaatitieto**

Järjestelmäkokonaisuuden tulee tietää mitä ovat organisaation yksiköt ja ketkä ovat yksiköiden päälliköitä/esimiehiä. Kokonaisuuden pitää myös tietää henkilön asema organisaatorakenteessa.

Organisaatorakenne Peppi kokonaisuuden käyttöön tulee ensisijaisesti pystyä hakemaan taustajärjestelmistä, jos tämän tyyppisiä taustajärjestelmiä on organisaatioilla olemassa. Jos tätä tietoa ei ole taustajärjestelmissä, tulee organisaatorakenne tieto tallentaa johonkin, esimerkiksi Peppi kokonaisuuden alle. Tätä tietoa voi tällöin hyödyntää myös muut palvelut, jotka tarvitsevat yksikkörakenteen ja esimies-alais rakenteen.

### **3.4.3 Kustannustunnisteet (kustannuspaikkahierarkia, toiminnot, projektinumerot)**

Peppi kokonaisuudessa käytettäviä yleisiä tunniste ja nimistö tietoja, kuten kustannuspaikkahierarkia, toiminnot ja projektinumerot, haetaan tietojen lähdejärjestelmistä käyttämällä Peppi kokonaisuuden palveluväylää.

Haettavaa tietoa tulisi käsitellä siten, että Peppi kokonaisuudella on reaaliaikaisesti tunnistetiedot käytettävissä. Tunnistetiedoilla tulee olla voimassaoloaika, joilla voidaan ohjata tunnisteiden käyttöä Peppi kokonaisuudessa.

## **3.5 Kokonaiskuva**

### **3.5.1 Palvelut**

Peppi-projektissa toteutettavat palvelut tarjoavat rajapinnat muiden palveluiden sekä käyttöliittymien käytettäväksi. Rajapinnat suojataan ja versioidaan siten, että moduulin päivitys ei riko rajapintaan kytkeytyviä käyttöliittymiä tai integraatioita. Palvelut voivat käyttää tallennusratkaisuna tietokantaa tai jotain muuta tallennusratkaisua, esimerkiksi materiaalin säilömiseen voidaan käyttää Alfresco-järjestelmää. Tallennusratkaisu on moduulin sisäinen asia eikä näy palvelun ulkopuolelle.

### **3.5.2 Integraatiot ulkoisiin järjestelmiin**

Palvelut keskustelevat ulkopuolisten järjestelmien kanssa alustan tarjoamien komponenttien avulla. Ulkopuolisia järjestelmiä ovat esimerkiksi Winha ja HR-järjestelmä.

Versio 0.3

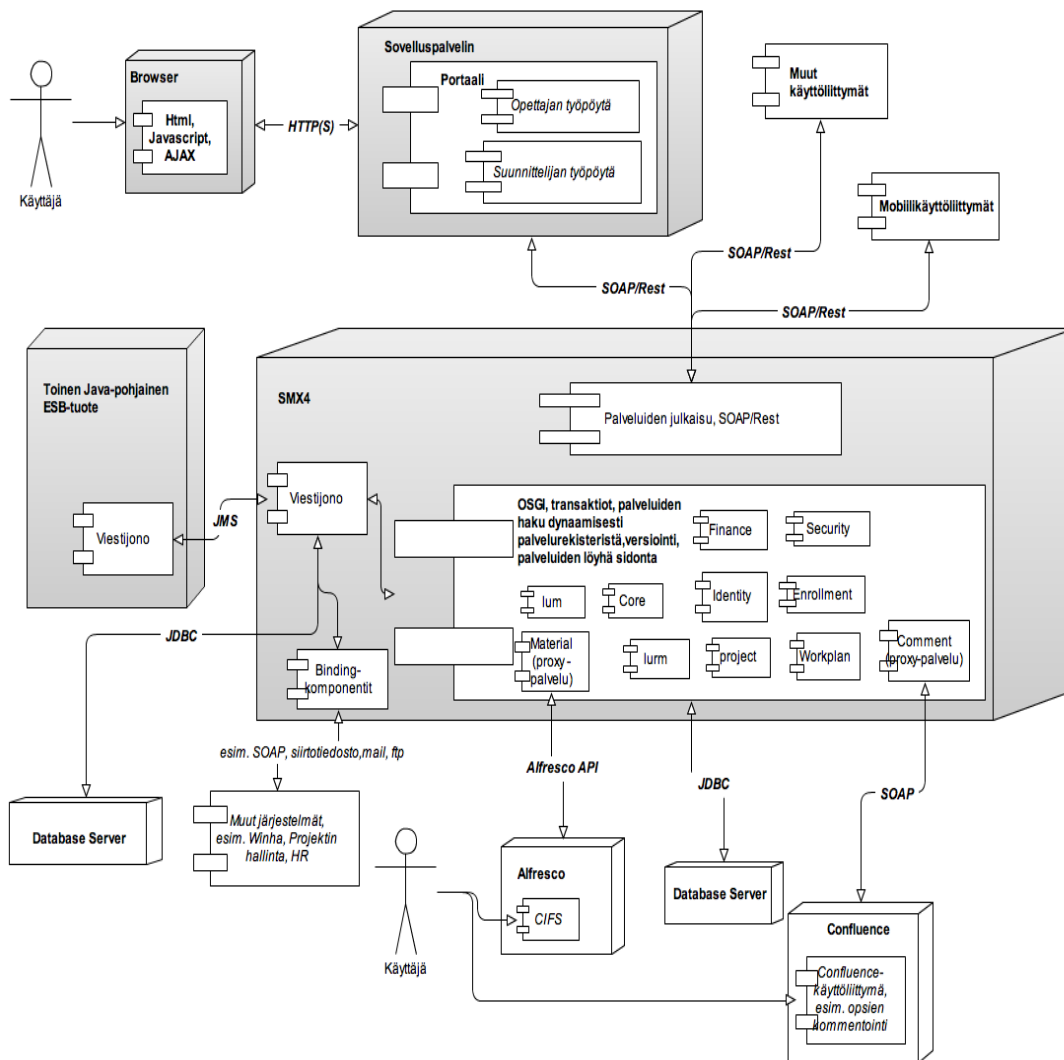
Tekninen vaatimusmäärittely

### 3.5.3 Käyttöliittymät

Peppi-projektissa toteutetaan opettajan ja suunnittelijan työpöytä. Näkymät hakevat tietoa palvelukerrokselta julkaistujen rajapintojen avulla. Näkymät suodattavat toimintoja käyttäjän roolien perusteella. Peppi-palveluiden julkisiin rajapintoihin on mahdollista kytkeä muita asiakasohjelmia, esimerkiksi mobiilikäyttöliittymä herätenäkymään.

#### Peppi-arkkitehtuurin esimerkkikuva

Deployment Diagram



## 4 Toiminnalliset vaatimukset

### 4.1 Opetussuunnittelu

### 4.2 Vuosisuunnittelu

#### 4.2.1 Tietojen syöttäminen

id	Vaatus	Vaatusen esittäjä	Tärkeys	Perustelu
121-1	Jokaisen tehtävän kohdalla pitää pystyä syöttämään kustannuspaikka	vuosisuunnittelutyö ryhmä	5	On tärkeää, että käyttäjä pystyy syöttämään kaikki kustannuspaikkoihin liittyvät tiedot tehtävälle halutessaan.
121-2	Sama tehtävä pitää olla mahdollista syöttää useammalle henkilölle	vuosisuunnittelutyö ryhmä	3	Helpottaa resursoijaa, kun samaa tehtävää ei tarvitse syöttää erikseen esim. 30 henkilölle.
121-3	Toteutuksen kesto saa olla myös useamman lukuvuoden	vuosisuunnittelutyö ryhmä	3	Totetus voi jatkua useammalle eri lukuvuodelle

Versio 0.3

Tekninen vaatimusmäärittely

121-4	Musiikin yksilöopetuksen tuki	M:HR	2	Nykyinen toteutus on rakennettu "ryhmäpohjaiseen". Malli tekee esim. musiikin koulutusohjelmien yksilöopetuksen hallinnoinnista erittäin työlästä. Yksilöopetuksessa voi ajatella yhden opiskelija-opettaja -parin muodostavan ryhmän: toteutuksia voi näin syntyä satoja.
121-5	Sivutoimisten tuntiopettajien (käytetty myös käsitettä "erikoisopettaja) toteutusten hallinnointi	M:HR	3	Sivutoimisille opettajilta ei ole velvoitettu ylläpitotehtävien hoitoa: mahdollistettava että (a) koulutussuunnittelija voi hoitaa usean toteutuksen perustietojen hallinnoinnin (toteutussuunnittelu) ja/tai (b) että sivutoimiset, joilla ei ole koulutussuunnittelu vaiheessa vielä amk:n käyttäjätunnuksia voi hoitaa toteutuksen perustietoja
121-6	Usean toteutuksen ylläpito	M: OT	4	"Ylläpitotyöliittymät" tietoihin rakennettu ajatelleen että muokataan yhtä tietoyksikköä kerrallaan. Mallilla kadotetaan keskeinen tavoiteltava päämäärä:

Versio 0.3

Tekninen vaatimusmäärittely

				<p>käyttäjä haluaa tehokkaan ja taloudellisen ylläpidon. On usein tilanteita, joissa muokataan useaa ryhmää, useaa toteutusta, useaa opettajaa, useaa opintojaksoa, useaa opetussuunnitelmaa, useaa opintojaksoa</p>
121-7	<p>Toteutuksen ajankohdan määrittely vuosisuunnittelussa - intensiivi viikot</p>	Riikka Vähätalo	3	<p>Vuosisuunnittelija voi syöttää toteutuksen alkamis ja päättymisajankohdan myös kalenteripäivämerkinnöillä, joiden avulla järjestelmä tietää mille periodeille toteutus sijoittuu. Tällöin voidaan vuosisuunnittelussa suoraan kertoa, jos toteutus toteutetaan intensiivi viikkona ryhmälle. Samalla tieto välittyy lukujärjestyksen tekijälle.</p>

#### 4.2.2 Tietojen esittäminen

id	Vaatus	Vaatusen esittäjä	Tärkeys	Perustelu
122-1	<p>Vuosisuunnittelunäky mä tulee esittää käyttäjälle koko OPS:n keston ajan</p>	vuosisuunnitteluty öryhmä	4	<p>Vuosisuunnittelijan tulee nähdä menneet opinnot, käynnissä olevat opinnot ja tulevat opinnot, jotta hän pystyy seuraamaan, että kaikki ryhmän tai koulutusohjelman opinnot tarjotaan järkevästi.</p>

Versio 0.3

Tekninen vaatimusmäärittely

122-2	<p>Suunnittelussa huomioitava että koulutusohjelmasta saattaa tulla eri tutkintoja, ne saattavat olla eri koulutusosalalla</p> <p>Ts. Yhdelle opetussuunnitelmalle pitää pystyä määrittelemään useampi tutkinto. Tällä hetkellä voidaan yhdelle opetussuunnitelmalle määrittellä vain yksi tutkinto.</p>	ek, tamk		<p>Nyt Tamkissa on ohjelma josta valmistuu 7 eri tutkintoon (Hyvinvointiteknologia ylempi amk). Tuleeko jatkossa enemmän vastaavia? Järjestelmien valmiudet tähän?</p> <p>M: yhdestä koulutusohjelmasta 1..n tutkintoa: hoitotyö, musiikki, pop/jazz - musiikki, esittävä taide, useita ylempiä amk tutkintoja sosiaali- ja terveysala.</p>
-------	--	----------	--	---

Versio 0.3

Tekninen vaatimusmäärittely

122-3	Instrumenttiopetuksessa tulee saada selville, mikä on opiskelija seuraavaksi suoritettava opintojakso			
122-4	Sisällön/ datan kieli ja käyttöliittymäkieli Täydellinen kaksikielisyys	jh/met		Käyttöliittymissä pitää pystyä vaihtamaan datan ja sisällön kieli sekä käyttöliittymäkieli. Kaksikielisuuden pitää toimia täydellisesti kaiken OPS tiedon osalta: opintojaksojen nimet, kuvaukset, jne. aina todellisella toteutuskielellä (yksikieliset todistukset sitten erikseen)

### 4.2.3 Raportointi

id	Vaatus	Vaatuskese n esittäjä	Tärkeys	Perustelu
123-1	Suunnittelutyön aikana nähtävä koko klusterin suunniteltu opetustarjonta eli toteutusten joukko <ul style="list-style-type: none"> <li>• substanssin mukainen jaoteltu</li> <li>• opettajittain</li> <li>• ryhmittäin</li> </ul>	liiketalous (sk)		Listaus toteutuksista eri hakuehdoin eri tarkoituksiin

Versio 0.3

Tekninen vaatimusmäärittely

	<ul style="list-style-type: none"> <li>• äin</li> <li>• kustannuspai</li> <li>• koittain</li> <li>• period</li> <li>• eittan</li> <li>• lukuka</li> <li>• usittai</li> <li>• koulut</li> <li>• usohje</li> <li>• lmittai</li> <li>• n</li> <li>• virtuaa</li> <li>• lihenkil</li> <li>• öittäin</li> <li>• kaikki</li> </ul> <p>virtuaalilihenkilöillä olevat toteutukset</p>			
123-2	<p>Opettajien opetustuntien (kalenteriin merkittyjen) seuranta tulisi olla helpompaa</p>	M:HR		<p>Tällä hetkellä musiikin koulutusohjelmissa XLS:n kerätään tiedot yksilöopetuksesta ja tietoihin lisätään mahdolliset ryhmäopetukset</p> <p>Lukujärjestyssuunnittelusta pitää pystyä luomaan raportti tietyn opettajan varauksista tietyillä toteutuksilla.</p>
123-3	<p>Opiskelija(t) ja ryhmä(t) nähtävät suunnittelun aikana</p>	M:OT		<p>Käyttötapaus kuvauksista löytyy sanoja "...ensimmäisen vuosikurssin opiskelijoille...", "...toisena vuonna...", "...läsnaoleva...", "...ryhmä...", "...pienryhmä...". Opiskelijoita on siis voitava tarkastella opiskelijan opintojen tilanteen mukaisesti ("ensimmäinen", "toinen"), opiskelijan tilan ("läsna", "läsna+1", "yliaikainen") ja ryhmien</p>



Versio 0.3

Tekninen vaatimusmäärittely

				("saapumisryhmä", "pienryhmä", "opetusryhmä") toteutusten suunnittelun aikana.
--	--	--	--	--

#### 4.2.4 Erilliset palvelut

id	Vaatus	Vaatumuksen esittäjä	Tärkeys	Perustelu
124-1	Musiikkipedagogien harjoitteluoppilaiden hallinta on saatava mukaan sovellukseen	M:HR		<p>Pedagogiopiskelijalla on kaksi harjoitusoppilasta kahden vuoden aikana. Harjoitusoppilaita hallitaan XLS:ssä eikä XLS:stä ole mitään liittymiä opetuksen hallintoon tai opintoihin.</p> <p>Jos harjoitusoppilaat on syötetty Winhaan ja Winhalle on määritelty harjoitusoppilaat ryhmä, voidaan tämä hallinnollinen ryhmä liittää toteutukselle.</p>
124-2	Muutosten seuranta ja hallinta	vuosisuunnittelu työryhmä	1	Nähtävä kenttätasolla : kuka muokannut, milloin ja miten muokattu. Muutoksia on voitava tarkastella kertyneenä historiatietona
124-3	Toteutusten osajoukkojen ehdotus tai julkaisu	M:OT		Koulutussuunnittelija suunnittelee toteutusten joukkoa. Toteutusten joukosta osa toteutuksia on ehdotettava opettajille ennen kuin toinen tai toiset osajoukot valmistuvat. Esim. sosiaali- ja terveysalan harjoittelun prosessit vaativat (?) että suunnittelu aloitetaan ensin harjoittelun

Versio 0.3

Tekninen vaatimusmäärittely

				suunnittelusta. Harjoittelun toteutusten suunnittelun on oltava valmis ja julkaistava ennen muita toteutuksia.
124-4	Suunnittelun näkymä voitava vaihtaa "kriittisen resurssi" mukaisesti	M:OT	2	Koulutuksen suunnittelu alkaa kulttuurialalla "kalenterista" tai "tilan kalenterista". Jos toimipisteessä on yksi tila, jossa opetus tapahtuu ja tilaa käyttää usea ryhmä, koulutusohjelma, organisaatioyksikkö, niin toteutusten suunnittelun alkaa sijoittamalla tilan kalenteriin opetustapahtumia. Tilan kalenteri rajoittaa toteutusten määrää.

### 4.3 Lukujärjestysuunnittelu

id	Vaatus	Vaatusen esittäjä	Tärkeys	Perustelu
131-1	Varauksille eri näkyvyystasot esim. 1=täysin alustava ja 5=julkaistu ja vahvistettu	Lukujärjestysuunnittelu	4	Suunnittelun alussa valtavasti muutoksia ja näkyvyys aluksi vain suunnittelijoille, siitä pikkuhiljaa opettajille ja lopuksi julkaisu
131-2	Useita näkymiä pitää pystyä pitämään auki samanaikaisesti	Lukujärjestysuunnittelu	4	Pitää pystyä tarkastelmaan nopeasti sijoittelua eri näkökulmista eli tilan, ryhmän, opettajan,

Versio 0.3

Tekninen vaatimusmäärittely

				toteutuksen ym. näkökulmista.
131-3	Tiloille ja välineille pitää pystyä syöttämään tuntihinta	Ohjausryhmä	3	Tietyissä tilanteissa halutaan arvioida kustannuksia. Tällöin pitää olla mahdollista arvioida myös tilan käytöstä aiheutuvia kustannuksia
131-4	Opetustapahtuman tilalle tai välineelle pitää pystyä syöttämään tilikartan mukaisten tunnisteiden mukaiset kategoriat (esim. kustannuspaikka, toiminto, projektinumero)	Ohjausryhmä	3	Jos kustannuksia halutaan kohdistaa, tulee yksittäisille varauksille pystyä syöttämään kustannuspaikka
131-5	Opetustapahtuman ja sille asetettujen resurssien pitää periä se kustannuspaikkajako, joka toteutukselle on merkittynä (esim. 1 tunnin mittainen opetustapahtuma, jossa on käytössä tila P301 ja väline dokumenttikamera)	Jaakko Rannila	3	Tämä sen vuoksi, että varauksen tekijän ei tarvitse erikseen tallentaa kustannuspaikkajakoa, jos se pääsääntöisesti periytyy toteutukselta.

## 4.4 Opettajapalvelut

id	Vaatusmus	Vaatimuksen esittäjä	Tärkeys	Perustelu
----	-----------	----------------------	---------	-----------

Versio 0.3

Tekninen vaatimusmäärittely

44-1	Työpöydän pitää olla muokattavissa	Opettajapalvelut	1	

## 4.5 Raportointi-, tieto- ja seurantapalvelut

id	Vaatus	Vaatimuksen esittäjä	Tärkeys	Perustelu

Päivämäärien hallinta (mm. periodit)

Opintojaksojen osat käytössä vuoteen 2016 asti hoitoalalla.

Kustannuspaikkatietojen ylläpito (poisto, lisäys, muutos),

Budjetti - hallittava vuosittain/lukuvuosittain

Projektit - vähintään nimi (ja kustannuspaikka ja projektinumero) ja sen ajan hallinta

Vuosisuunnittelun **opettajaroolin hallinta**=lukuvuoden mittainen

KRIITTINEN ASIA: Roolin muutoksen hallinta esim sivutoiminen->päätoiminen

### Järjestelmän logikirjaukset

Järjestelmän on pystyttävä kirjaamaan ylläpitäjän määrittämän tason mukaisia tapahtumia tapahtumalogiin erilaisten virhetilanteiden selvittämisen helpottamiseksi.

Lisäksi järjestelmän on kirjattava milloin käyttäjät ovat kirjautuneet sisään ja ulos. Tai vaihtoehtoisesti ylläpitäjä voi tarvittaessa katsoa jollakin tavalla, onko järjestelmässä käyttäjiä kirjautuneena (esim ennen järjestelmän uudelleen käynnittämistä).

## 5 Ei-toiminnalliset vaatimukset

### 5.1 Tietojen arkistointi

Versio 0.3

Tekninen vaatimusmäärittely

id	Vaatusmus	Vaatusmuksen esittäjä	Tärkeys	Perustelu
21-1	<p>Kun tietoa ei saa enään tietyn päivämäärän jälkeen käyttää sen käyttö tulee estää käyttäjältä. Esimerkiksi kustannuspaikkojen vanheneminen tulee toteuttaa siten, että kustannuspaikkalistaus esitetään sen mukaisesti mikä on tehtävän alku ja loppupäivämäärä. Jos kustannuspaikan käyttö loppuu 31.12.2010 ja tehtävä päättyy 1.1.2011 niin kyseistä kustannuspaikkaa ei voida valita. Tehtävä tulee muuttaa päättymään 31.12.2010, jotta kustannuspaikan voi valita.</p>	<p>vuosisuunnitt elutyöryhmä</p>	5	<p>Tämä on tärkeää, koska mm. kustannuspaikkatiedot ja projektinumerotiedot ovat aikaan sidottuja. Tiettyjä asioita ei saa enään käyttää tietyn päivän jälkeen.</p>
21-2	<p>Eri tiedoille kuten kustannuspaikoille, projekteille, toimipisteille, rooleille tulee olla voimassaoloaika. Voimassaolon päättymisaika voi olla myös toistaiseksi, jolloin päättymisaikaa ei ole määritelty.</p>	<p>vuosisuunnitt elutyöryhmä</p>	5	<p>Tiedot vanhentuvat ja niillä on voin "historiallista" merkitystä. Kun suunnittelua tehdään ei erilaisissa listauksissa tarvitse esittää tietoja, joita ei voi enään käyttää</p>

Versio 0.3

Tekninen vaatimusmäärittely

--	--	--	--	--

## 5.2 Historiatietojen säilyttäminen

id	Vaatusmus	Vaatumuksen esittäjä	pvm	Tärkeys	Perustelu
22-1	Vanhojen työaikasuunnitelmien tulee säilyä myöhempää tarkastelua varten	vuosisuunnitteluryhmä	24.8.2010	4	Opettajan ja esimiehen tulee tietää miten työaikasuunnitelma on muuttunut esimerkiksi hyväksytystä ja allekirjoitetusta versiosta. Tällöin em. käyttäjien pitää nähdä miten, kuka ja missä vaiheessa työaikasuunnitelmaa on muutettu
22-1	Kun käyttäjä "poistaa" jonkun asian käyttöliittymän kautta tulee mahdollisimman useissa tapauksissa muuttaa poistetun tiedon tila siten, että	Jaakko Rannila	1.12.2010	4	Tällöin on helpompi palauttaa tiedot takaisin, jos käyttäjällä käy virhe. Samalla pystytään selvittämään tilanteita, joissa "tiedot

Versio 0.3

Tekninen vaatimusmäärittely

	<p>tieto ei kuitenkaan poistu tietokannasta vaan sen tila muuttuu "poistettu"-tilaan. Poistosta tulee jäädä tietokantaan merkintä siitä, kuka on poistanut ja koska.</p>				<p>katoavat itsestään".</p>

### 5.3 Tallennustietojen säilyttäminen

id	Vaatus	Vaatusen esittäjä	pvm	Tärkeys	Perustelu
23-1	<p>Toteutusten ja tehtävien tallentaja ja ajankohta pitää tallentua</p>	<p>vuosisuunnitt elutyöryhmä</p>	<p>24.8.2010</p>	<p>4</p>	<p>Koska tietoja käsittelee useat eri henkilöt on tärkeää nähdä, kuka on tiedot tallentanut ja minä ajankohtana, jolloin epäselvissä tilanteissa voidaan ottaa yhteyttä oikeaan henkilöön.</p>

Versio 0.3

Tekninen vaatimusmäärittely


## 5.4 Avoimet rajapinnat

id	Vaatus	Vaatimukset esittäjä	pvm	Tärkeys	Perustelu
24-1	Opetussuunnitelman tietojen julkaisu ja esittäminen REST/RESTful rajapinnan avulla	Jaakko Rannila	29.10.2010	4	Opetussuunnitelmien sisältöä halutaan julkaista eri paikoissa ja tilanteissa. Ohjausryhmä on linjannut, että opetussuunnitelman sisältöä pitää pystyä julkaisemaan myös sosiaalisissa medioissa. Tämän takia opetussuunnitelma pitää olla saatavilla ja haettavilla rajapinnan avulla eri sisältöalustoilla.
24-2	Toteutus tietojen julkaisu ja esittäminen	Jaakko Rannila	29.10.2010	4	Toteutukseen liittyvää tietoa halutaan julkaista eri



Versio 0.3

Tekninen vaatimusmäärittely

	REST/RESTful rajapinnan avulla				paikoissa ja tilanteissa. Tiedot pitäisi olla haettavissa rajapinnan avulla
24-3	Työaikasuunnitelman tietojen julkaisu ja esittäminen REST/RESTful rajapinnan avulla	Jaakko Rannila	29.10.2010	4	Työaikasuunnitelmiin liittyvää tietoa halutaan julkaista eri paikoissa ja tilanteissa. Tiedot pitäisi olla haettavissa rajapinnan avulla.
24-4	Työjärjestysten, kuten luokkatilojen varausten julkaisu ja esittäminen REST/RESTful rajapinnan avulla	Jaakko Rannila	29.10.2010	4	Työjärjestyksiä halutaan julkaista eri paikoissa, laitteissa ja tilanteissa. Tiedot pitäisi olla haettavissa rajapinnan avulla.

Kun tietoa ei saa enään tietyn päivämäärän jälkeen käyttää sen käyttö tulee estää käyttäjältä. Esimerkiksi kustannuspaikkojen vanheneminen tulee toteuttaa siten, että kustannuspaikkalistaus esitetään sen mukaisesti mikä on tehtävän alku ja loppupäivämäärä. Jos kustannuspaikan käyttö loppuu 31.12.2010 ja tehtävä päättyy 1.1.2011 niin kyseistä kustannuspaikkaa ei voida valita. Tehtävä tulee muuttaa päättymään 31.12.2010, jotta kustannuspaikan voi valita.

Eri tiedoille kuten kustannuspaikoille, projekteille, toimipisteille, rooleille tulee olla voimassaoloaika. Voimassaolon päättymisaika voi olla myös toistaiseksi, jolloin päättymisaikaa ei ole määritetty.

## 5.5 Virheen käsittely

Id	Vaatus	Vaatimuksen esittäjä	pvm	Tärkeys	Perustelu
E01	Mikäli käyttäjä onnistuu tekemään pyynnön käyttöliittymässä, johon hänellä ei ole oikeuksia, tulee palvelukerros tehdä tästä lokimerkintä ja heittää poikkeus asiakasohjelmalle	Eero Manninen	31.1.2011	1	Palvelukerrosen vastuulla on myös auktorisointi
E05	Palvelukerroselta tulevat Runtime-tyyppiset poikkeukset täytyy käsitellä yhdenmukaisesti tavalla opettajan ja suunnittelijan työpöydän näkymissä	Eero Manninen	31.1.2011	1	odottamattomat virhetilanteet käsitellään hallitusti
E10	Palvelukerroselta tulevat poikkeukset, jotka liittyvät virheelliseen dataan,	Eero Manninen	31.1.2011	2	Loppukäyttäjä on helppoa nähdä puuttuva/virheellinen tieto

Versio 0.3

Tekninen vaatimusmäärittely

	täytyy käsitellä opettajan ja suunnittelijan näkymissä siten, että käyttäjä saa tiedon mihin virheelliseen dataan poikkeus liittyy				ja korjata viesti
E15	Runtime poikkeuksille tulee generoida tunniste ja välittää se asiakasohjelmalle	Eero Manninen	31.1.2011	3	Voidaan yksilöidä tapahtumalokitiedostossa
E20	Poikkeukset tulee kirjoittaa lokiin vain kerran	Eero Manninen	31.1.2011	2	Lokitiedot on selkeämmin nähtävissä
E25	Häiriön tapahtuessa lokiin täytyy tehdä merkintä, jolla häiriön syy voidaan jäljittää tiettyyn kohtaan lähdekoodissa	Eero Manninen	31.1.2011	1	Häiriön aiheuttaja voidaan korjata nopeasti

## Liitteet

Teknisen alustan koestusvaiheen raportit / Proof of Concept vaiheen raportit

- OpenESB PoC -raportti

Metropolia Ammattikorkeakoulu  
Tampereen Ammattikorkeakoulu

Peppi - Koulutuksen suunnittelijan ja  
opettajan palvelut

Versio 0.3

- Kuali PoC -raportti
- Kuali Days 2010 -raportti

Tekninen vaatimusmäärittely

Metropolia Ammattikorkeakoulu  
Tampereen Ammattikorkeakoulu

Versio 0.3

Peppi - Koulutuksen suunnittelijan ja  
opettajan palvelut

Tekninen vaatimusmäärittely