

## **Suunnitteludokumentti**

Ohjelmistotuotantoprojektin tietojärjestelmä - Otie2

Helsinki 3.11.2004

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

**Kurssi**

581260 Ohjelmistotuotantoprojekti (6 ov)

**Projektiryhmä**

Jani Hanhisalo

Matias Käkelä

Virpi Peuralinna

Niklas Rosenberg

Petteri Salonvaara

Teemu Vilén

**Asiakas**

Turjo Tuohiniemi

**Johtoryhmä**

Juha Taina

**Kotisivu**

<http://www.cs.helsinki.fi/group/otie2>

**Versiohistoria**

Versio	Päiväys	Tehdyt muutokset
0.1	6.10.2004	Ensimmäinen versio - dokumenttipohja, NR
0.2	8.10.2004	Lisätty Käyttöliittymäluokkia, JH
0.3	11.10.2004	Lisätty sovelluslogiikkatason luokkakuvauksia, NR
0.31	13.10.2004	Tarkennettu luvun 5 luokkakuvauksia ja layout-muutoksia, NR
0.32	14.10.2004	Muutoksia ryhmätapaamisen yhteydessä layoutiin NR, MK, VP, JH
0.33	14.10.2004	Muutoksia esimerkkirakenteeseen MK
0.34	15.10.2004	Lisätty käyttäjän perustietojen muokkaus php-sivun kuvaus JH
0.35	15.10.2004	Lisätty aktiivisten projektien haku MK
0.36	17.10.2004	Lisätty osa tietokannan kuvauksesta TV, PS
0.37	16.10.2004	Lisätty projektien haku, MK
0.38	18.10.2004	Lisätty projektien luonti ja muokkaus, NR
0.4	22.10.2004	Lisätty tietokantakaavio ja sekvenssit, PS
0.41	23.10.2004	Lisätty addSerie.php, addWork.php ja LIB- sekä DB-funktioita, NR
0.45	23.10.2004	Lisätty gradeProject.php ja sen LIB,UI ja DB:t, MK
0.46	23.10.2004	Lisätty createPerson.php ja showHelp.php, TV
0.50	23.10.2004	Lisätty search.php ja sen UI ja DB:t, MK
0.51	24.10.2004	Lisätty ProjectMetricSearch.php ja sen UI ja DB:T, VP
0.52	24.10.2004	Lisätty ActiveProjects.php ja UIcreateWorkTable, TV
0.53	24.10.2004	Lisätty authentication.php, modifyProjectInformation.php, NR
0.54	24.10.2004	Lisätty modifyPersonInformation.php, TV
0.55	24.10.2004	Lisätty modifySerieInProject.php, PS
0.56	25.10.2004	Lisätty DBgetMetricsData ja DBgetSerieComment, PS
0.6	25.10.2004	Oikoluettu versio, PS
0.61	25.10.2004	Lisätty index.php, error.php, luku 9, TV
0.62	25.10.2004	addSerieToProject.php, UIcreateSerieForm ja DBaddSerie, NR
0.7	25.10.2004	Rakennetta ja layoutia muokattu oikolukua varten, NR
0.71	27.10.2004	Korjauksia FTR:n jälkeen, NR
0.8	27.10.2004	Tehty arkkitehtuurikuvaus ja 3-tier-mallista kaavio, PS
0.9	31.10.2004	Muokattu ModifySerieInProject ja lisätty 3 uutta funktiota, PS
0.91	1.11.2004	Muokattu p_id, proj_id ja Pers_id -indeksien tilalle tunnukset, PS
0.92	2.11.2004	Uusi versio tietokantakaaviosta ja muokattu arkkitehtuuriosaa, PS
0.93	2.11.2004	Oikoluettu versio, PS
1.0	3.11.2004	Lopullinen versio, NR

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Taustatietoa . . . . .	1
1.2	Termit . . . . .	1
1.3	Dokumentin rakenne . . . . .	2
<b>2</b>	<b>Järjestelmän yleiskuvaus</b>	<b>2</b>
2.1	Toteutusympäristö . . . . .	2
2.2	Toimintaympäristö . . . . .	3
<b>3</b>	<b>Arkkitehtuurikuvaus</b>	<b>3</b>
<b>4</b>	<b>Sovelluslogiikkatason PHP-sivut</b>	<b>4</b>
4.1	Kaikille sivuille yhteiset asiat . . . . .	4
4.2	activeProjects.php . . . . .	4
4.3	addSerie.php . . . . .	5
4.4	addSerieToProject.php . . . . .	6
4.5	addWork.php . . . . .	7
4.6	authentication.php . . . . .	9
4.7	changePreferences.php . . . . .	10
4.8	createPerson.php . . . . .	11
4.9	createProject.php . . . . .	12
4.10	error.php . . . . .	14
4.11	gradeProject.php . . . . .	14
4.12	index.php . . . . .	16
4.13	modifyPersonInformation.php . . . . .	16
4.14	modifyProjectInformation.php . . . . .	17
4.15	modifySerie.php . . . . .	19
4.16	modifySerieInProject.php . . . . .	20
4.17	projectInformation.php . . . . .	21
4.18	projectInformationHours.php . . . . .	21
4.19	projectMetricSearch.php . . . . .	21
4.20	search.php . . . . .	23
4.21	showHelp.php . . . . .	25

<b>5 Käyttöliittymätason funktiot</b>	<b>26</b>
<b>6 Kirjasto- ja työkalufunktiot</b>	<b>28</b>
<b>7 Tietokantatason funktiot</b>	<b>30</b>
7.1 DBMisc . . . . .	30
7.2 DBPerson . . . . .	30
7.3 DBSecurity . . . . .	30
7.4 DBProject . . . . .	31
7.5 DBMetric . . . . .	32
7.6 DBWork . . . . .	34
<b>8 Järjestelmän tietosisältö</b>	<b>35</b>
8.1 Tietokantakaavio . . . . .	35
8.2 Tietokantataulut . . . . .	35
8.2.1 Projects . . . . .	35
8.2.2 Person . . . . .	36
8.2.3 Person_in_project . . . . .	36
8.2.4 Work . . . . .	37
8.2.5 User_rights . . . . .	37
8.2.6 Person_rights . . . . .	38
8.2.7 Content . . . . .	38
8.2.8 Metric_Series . . . . .	39
8.2.9 Content_in_serie . . . . .	39
8.2.10 Serie_in_project . . . . .	40
8.2.11 Metrics_data . . . . .	40
8.2.12 Selection_lists . . . . .	41
8.2.13 List_elements . . . . .	41
8.2.14 List_to_content . . . . .	42
<b>9 Käyttöliittymä</b>	<b>42</b>
9.1 Graafinen ohjeistus . . . . .	42
9.2 Taitto . . . . .	42
9.3 Proto . . . . .	42

# 1 Johdanto

Tämä suunnitteludokumentti kuvaa teknisestä näkökulmasta Otie2 -järjestelmän toteutuksen. Suunnitteludokumentti toimii järjestelmätoteutuksen ohjeena ja siinä kuvataan järjestelmän arkkitehtuuri, luokkarakenne, tietosisältö sekä käyttöliittymä. Suunnitteludokumentin perustana käytetään Otie2 -projektin määrittelydokumenttia.

## 1.1 Taustatietoa

Syksyn 2004 aikana tietojenkäsittelytieteen laitoksella toteutetaan Ohjelmistotuotannon tietojärjestelmä — Otie2 — niminen projektikurssi. Otie2 -projektin tavoitteena on kehittää ohjelmistosovellus tulevista ohjelmistotuotantoprojekteista mitattavien metriikoiden tallennukseen ja katseluun. Metriikalla käsitetään tämän dokumentin yhteydessä mitattavaa tietoa, jolle on määritetty nimi, kuvaus ja tyyppi.

Lisäksi toteutettava ohjelmisto mahdollistaa aiemmin toteutettujen projektiryhmien tietojen syöttämisen järjestelmään. Päättäneiden projektien syöttämisessä järjestelmään hyödynnetään samoja välineitä, kuin uusissa projekteissa. Ohjelmisto on tarkoitettu projekteihin osallistuvien opiskelijoiden, ohjaajien ja vastuuhenkilöiden sekä muiden opiskelijoiden ja tutkijoiden käyttöön.

Toteutettavan ohjelmiston on tarkoitus olla mahdollisimman modulaarinen siten, että sitä on tulevaisuudessa helppo laajentaa tai muokata uusien tarpeiden mukaan.

## 1.2 Termit

**Metriikka:** Mitattava tieto. Metriikalla on nimi, kuvaus ja tyyppi (metriikkamalli). Kuvaus on samalla ohje käyttäjälle. Esimerkiksi valmiista koodista laskettu koodirivien lukumäärä on metriikka.

**Metriikkamalli:** Metriikan tyyppi, joka määrittää miten metriikka tulostetaan ja tallennetaan. Metriikkamalleja ovat totuusarvo, numeerinen arvo, taulukkonumeerinen arvo, valintalista, päivämäärä ja vapaa tekstisyöte.

**Metriikkasarja:** Kokoelma yhteenliittyviä metriikoita. Esimerkkeinä toteutusvaiheessa kerättävät tiedot (ts. metriikat) tai testausvaiheesta kerättävät tiedot.

**PHP:** Hypertext PreProcessor, aiempi merkitys Personal Home Page tools. Yleiskäyttöinen skriptikieli, joka soveltuu erityisesti web-sovellusten toteuttamiseen.

**Projekti:** Helsingin yliopiston tietojenkäsittelytieteen laitoksen kurssin Ohjelmistotuotantoprojekti puitteissa toteutettava projektimuotoinen ryhmätyö, jonka tavoitteena on ohjelmistosovelluksen toteuttaminen.

**TKTL:** Helsingin yliopiston tietojenkäsittelytieteen laitos.

**W3C:** World Wide Web Consortium. W3C:n tavoitteena on kehittää yhteisiä verkkostandardeja ja teknologioita. W3C pyrkii suosituksillaan luomaan yleiskäyttöisiä ohjeituksia mm. yhteensopivuuden takaamiseksi, näin esimerkiksi www-sivut näkyvät eri selaimissa oikein. Ks. <http://www.w3c.org>.

**XML:** eXtensible Markup Language on metakieli rakenteellisen tiedon kuvaamiseen.

### 1.3 Dokumentin rakenne

Suunnitteludokumentin ensimmäinen luku sisältää lyhyen johdannon aiheeseen ja luvussa kaksi esitetään yleiskuva järjestelmästä. Luvussa kolme kuvataan järjestelmän arkkitehtuuri yleisellä tasolla.

Luku neljä sisältää yksityiskohtaiset kuvaukset niistä PHP-sivuista, joiden avulla määritellyt käyttötapaukset suoritetaan. Luvut viisi, kuusi ja seitsemän sisältävät käyttöliittymä-, tietokanta- ja apufunktioiden kuvauksia.

Järjestelmän tietosisältö on esitetty luvussa kahdeksan ja käyttöliittymähahmotelma löytyy luvusta yhdeksän.

## 2 Järjestelmän yleiskuvaus

Otie2 -järjestelmä toteutetaan web-sovelluksena, joka pyörii Apache-palvelimen alaisuudessa. Tietokannanhallintajärjestelmäksi on valittu Oracle. Pääasiallisina toteutuskielinä toimii PHP ja SQL.

Järjestelmän käyttö tapahtuu Internet-selaimen avulla (esim. Internet Explorer tai Mozilla), jolloin asiakaskoneella ei tarvitse olla mitään erityisiä muita sovelluksia tai komponentteja asennettuna.

Järjestelmän sidosryhmät koostuvat opiskelijoista, ohjaajista, vastuuhenkilöistä, tutkijoista ja kirjautumattomista käyttäjistä. Järjestelmän käyttöä kuvaavat käyttötapaukset on määritelty Otie2 -projektin määrittelydokumentissa. Vaikka käyttäjät voidaankin ryhmitellä eri sidosryhmiin, järjestelmän käyttöä rajataan toimintokohtaisilla (ei roolikohtaisilla) oikeuksilla. Kirjautumattomat käyttäjät voivat selata projekteihin liittyviä perustietoja. Tutkijoiden pääasiallinen tiedonhaku perustuu tietokantanäkymien käyttöön.

### 2.1 Toteutusympäristö

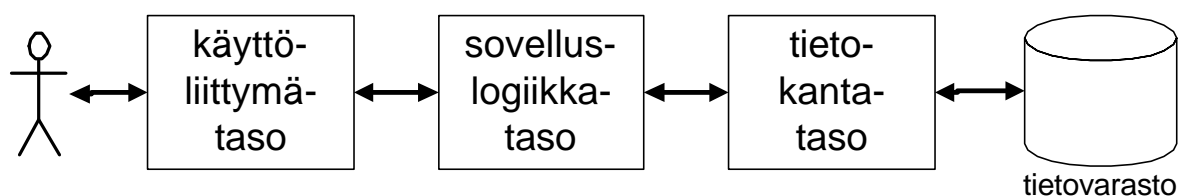
Järjestelmä toteutetaan PHP:n versiolla 4 sekä käyttäen Oraclen SQL-tietokantaa v10g. Ohjelmisto saattaa toimia myös varhaisemmilla PHP:n versioilla tai muilla SQL-palvelimilla, mutta tätä ei tulla erikseen testaamaan. Versionhallintaan käytetään TKTL:n palvelimilta löytyvää CVS:ää. Ensisijaisesti kehitystyö tehdään Linux-alustalle, mutta ohjelmistosta pyritään tekemään käyttöjärjestelmäriippumaton.

## 2.2 Toimintaympäristö

Järjestelmä toimii TKTL:n alkokrunni- sekä bodbacka-palvelimilla (`db.cs.helsinki.fi` ja `bodbacka.cs.helsinki.fi`). Alkokrunniin on sijoitettu osa järjestelmän käyttämistä palvelinohjelmistoista (PHP 4.3.8, Apache). Tietokantapalvelin (Oracle 10g) sijaitsee bodbacka-palvelimella.

## 3 Arkkitehtuurikuvaus

Järjestelmän arkkitehtuuriratkaisuna on 3-tier-malli, joka koostuu kolmesta tasosta: käyttöliittymätaso, sovelluslogiikkataso ja tietokantataso. Tasot kommunikoivat keskenään oheisen kaavion 1 mukaisesti.



Kuva 1: 3-tier-malli

Käyttöliittymätaso vastaa käyttöliittymänä toimivien sivujen tuottamisesta ja toimittamisesta asiakkaalle. Käyttöliittymätaso toteutetaan Web-selainpohjaisena. Käyttöliittymän yleisilme tehdään TKTL:n tyylin mukaiseksi käyttämällä laitoksen CSS-tyylitiedostoja. HTML-dokumenttien dynaamisuus ja interaktiivisuus saavutetaan PHP-skripteillä. Koodissa funktiot kapseloivat tietorakenteensa ja luokan sisältä tietoa välitetään vain funktioiden rajapintojen määritysten mukaisesti. Käyttöliittymätason funktiot nimetään UI-alkuisiksi.

Sovelluslogiikka hoitaa kommunikoinnin käyttöliittymätason ja tietokantatason välillä. Sovelluslogiikkaa suorittaa ajonaikaisesti palvelinpään PHP-tulkki.

Tietokantatasolla suoritetaan SQL-komentosarjat ja tietokannan käsittely, Pyrkimys on, ettei sovelluslogiikkatasolla esitetäisi SQL-komentoja, ainakaan dataa muokkaavia komentoja, vaan että pyynnöt välitetään tietokantatasolle funktioiden parametreina ja tietokantatason funktio palauttaa tulokset paluuarvoina. Tietokantatason luokkien funktiot nimetään DB-alkuisiksi.

Erilaisia aputoimintoja varten toteutetaan kirjastoluokka. Näitä toimintoja ovat esimerkiksi muuttujan tyyppin oikeellisuuden, maksimi- ja minimirajojen tarkistus sekä datan jäsentäminen syöttötietovirrasta. Kirjastoluokat luokkien funktiot LIB-alkuisiksi.



## 4 Sovelluslogiikkatason PHP-sivut

Tässä luvussa kuvataan yksityiskohtaisesti järjestelmän sovelluslogiikkataso, joka koostuu PHP-sivuista. Ajatuksena on, että kaikki määritellyt käyttötapaukset voidaan suorittaa tässä luvussa kuvattujen PHP-sivujen avulla.

### 4.1 Kaikille sivuille yhteiset asiat

#### Autentikointi ja sessionkäsitely

Jokaisen sivun alussa tarkastetaan käyttäjän oikeudet session ympäristömuuttujasta \$account ja edellinen sivunlataus \$lasttime. Jokainen PHP-sivu tarkastaa funktiolla DBhasRight(\$account, \$KO), onko käyttäjällä ne oikeudet, joita PHP-sivulla tarvitaan. Virhetilanteessa käyttäjä ohjataan virhesivulle error.php.

Jos muuttujan \$lasttime aikaleiman ja järjestelmän kellon erotus on 30 minuuttia (ei lopullinen arvo), on sessio vanhentunut, se tuhotaan ja käyttäjä ohjataan sivulle error.php. Jokainen sivulataus päivittää session \$lasttime lataushetken aikaleimaksi.

#### Layout

Tarkistusten jälkeen jokainen PHP-sivu kutsuu funktiota UIheader ja sovelluslogiikan suorituksen jälkeen funktiota UIfooter. Nämä funktiot luovat sivun HTML-layoutin ja jättävät jokaisella sivulla sovelluslogiikalle alueen, johon sovelluslogiikan tulosteet sijoittuvat.

Jokaisella sivulla kutsutaan myös funktiota UIcreateNaviMenu(\$userrights, \$projMembership), joka luo navigaatiovalikon oikeuksien ja projektijäsenyyksien perusteella. Parametri \$userrights on taulukko, jonka saa funktiolla \$userrights = DBgetRights(\$lang, \$account). Parametri \$projMembership on taulukko, jonka saa funktiolla \$projMembership = DBgetProjectMembership(\$account).

Käytetty CSS-tiedosto on kopio TKTL:n sivujen käyttämästä CSS-tiedostosta. Kaikki HTML:ää tulostavat funktiot on kategorisoitu UI-funktioiksi.

#### Kielen valinta

Jokainen UI-funktio saa parametrinaan muuttujan \$lang, joka kulkee mukana myös jokaisen sivun parametreissa. DB-funktiot, jotka palauttavat dataa, saavat myös parametrinaan muuttujan \$lang milloin tietokannasta pitää valita minkä kielisenä tietoa pitää hakea.

### 4.2 activeProjects.php

**Kuvaus:** PHP-sivu näyttää näkymänä kaikkien aktiivisten projektien työtunnit. Sivun parametroitavissa näyttämään tunnit projektivaiheittain tai viikoittain.

Kaikki aktiiviset projektit haetaan funktiokutsulla \$projects = DBgetProjects(\$lang, \$active). Täytetään erikseen hashtable \$grouping ['groupname' | '\$project'], missä hashtable \$project on muotoa ['project name' | 'hours'], missä arvo 'hours' vastaa 'groupname':a vastaavaa tuntiryhmitystä:

```
SELECT p.p_name, w.sum(hours), w.phase|w.work_week
FROM projects p, work w
WHERE p.P_name = w.P_name AND p.p_name = $project
```

Tällöin tietorakenne \$projects vastaa taulukon kaikki projekteja, ja \$grouping kaikkia tietokannasta haettuja pääryhmiä (esim. viikot "34, 35, 36, 37"tai vaiheet "PS, VA, SU, TO"). Tietorakenteet luovutetaan eteenpäin funktiolle UIcreateWorkTable(\$lang, \$projects, \$grouping).

#### Sivun vastaanottamat parametrit:

groupby	Tuntien ryhmittely [week tai phase]
lang	käyttäjän kieli (en tai fi)

#### Tarvittavat oikeudet:

Saa nähdä aktiivisten projektien tietoja	KO30
--	------

#### Liittyvät käyttötapaukset:

Aktiivisten projektien näkymä	KT13
-------------------------------	------

## 4.3 addSerie.php

**Kuvaus:** addSerie.php on sivu, jolle päästään navigaatiovalikosta. Sivun näyttää käyttäjälle listan kaikista projektiin liitettävissä olevista metriikkasarjoista.

Metrikkasarjat haetaan tietokannasta funktiolla \$series = DBgetAvailableSeries(\$lang). Parametri \$lang on muotoa "fi"tai "en"ja paluuarvo on hash table (kts. kohta 7.5).

Lista luodaan funktiolla UIcreateSerieList(\$series). Funktion luoman listan metriikkasarjat ovat parametreilla varustettuja linkkejä sivulle addSerieToProject.php.

#### Sivun vastaanottamat parametrit:

proj_id	projektin id (P_name)
lang	käyttäjän kieli (en tai fi)

#### Sekvenssikaavio:

Sivun addSerie.php sekvenssikaavio esitetty kuvassa 2

#### Sivun käyttämät funktiot:

DBgetAvailableSeries(\$lang)  
UIcreateSerieList(\$series)

#### Tarvittavat oikeudet:

Saa lisätä metriikkasarjan mihin tahansa projektiin	KO20
---	------

Saa lisätä metriikkasarjan omaan projektiinsa

KO21

### Liittyvät käyttötapaukset:

Metriikkasarjan kiinnittäminen projektiin

KT17

Mittaustietojen syöttö

KT19

## 4.4 addSerieToProject.php

**Kuvaus:** addSerieToProject.php on sivu, jolle päästään sivulta addSerie.php. Tällöin käyttäjä on jo valinnut mikä metriikkasarja liitetään projektiin, jonka jälkeen kyseisen metriikkasarjan pakolliset metriikat on täytettävä. Sivun saa parametreinaan mm. projektin tunnuksen ja metriikkasarjan id:n.

Jos sivulle päästään suoraan ilman submit-napin painamista, ensin haetaan tietokannasta metriikkasarjan rakenne funktiolla \$structure = DBgetSerieStructure(\$s\_id, \$lang). Tämän jälkeen metriikkasarjan sisältö tulostetaan käyttäjälle funktiolla UIcreateSerieForm(\$P\_name, \$serie\_id, \$url, \$structure, \$serie, \$comment, \$lang). Parametri \$structure on funktion DBgetSerieStructure palauttama taulukko. Parametri \$url kertoo minne lomakkeen action johtaa (tässä tapauksessa samalle sivulle addSerieToProject.php). Parametri \$serie on tässä tapauksessa null.

Kun käyttäjä on täyttänyt metriikkasarjan pakolliset metriikat ja painanut submit-nappia, siirrytään takaisin sivulle addSerieToProject.php. Parametreina saadaan syötetyt metriikat parametreissa metric\_x, jossa x viittaa metriikan id:hen. Ensin tarkistetaan syötettyjen tietojen oikeellisuus funktiolla \$serie = LIBcheckSerieData(\$structure, \$metricsdata). Parametri \$structure on edelleen sama, jonka DBgetSerieStructure palauttaa. Parametri \$metricsdata on taulukko, joka on määritelty funktion LIBcheckSerieData kuvauksessa ja se sisältää käyttäjän metriikkasarjaan täyttämät tiedot.

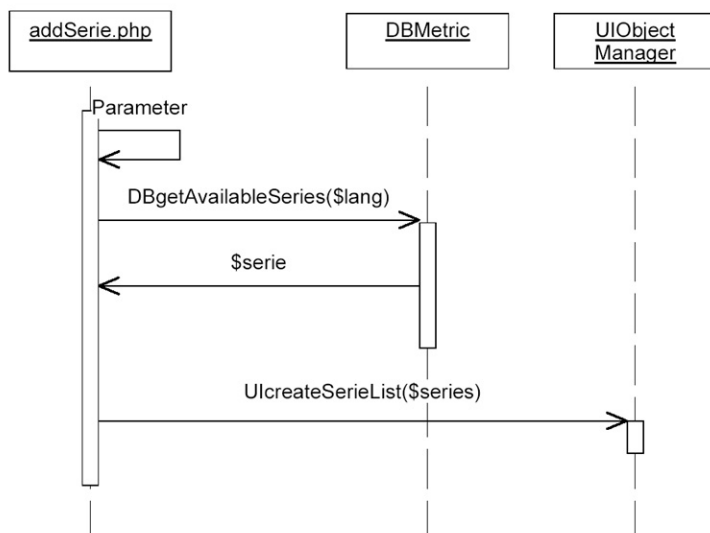
Funktion LIBcheckSerieDatan paluuarvona saatu \$serie sisältää syötetyt tiedot sekä mahdolliset virheet. Parametri on määritelty ko. funktion kuvauksen yhteydessä. Jos tiedoissa oli virheitä, luodaan uudestaan lomake funktiolla UIcreateSerieForm ja tällä kertaa parametri \$serie on sama tietorakenne kuin mitä LIBcheckSerieData palautti. Jos virheitä ei ollut, tiedot tallennetaan tietokantaan funktiolla DBaddSerie(\$serie, \$P\_name, \$s\_id, \$comment). Tässäkin tapauksessa parametri \$serie on sama tietorakenne kuin mitä LIBcheckSerieData palautti.

### Sivun vastaanottamat parametrit:

P_name	Projektin tunnus (P_name)
serie_id	Projektiin lisättävän sarjan id
metric_x	Metriikkaan syötetty tieto, x viittaa metriikan id:hen
lang	Käyttäjän kieli (en tai fi)

### Sekvenssikaavio:

Sivun addSerieToProject.php sekvenssikaavio esitetty kuvassa 3



Kuva 2: Sivun addSerie.php sekvenssikaavio.

#### Sivun käyttämät funktiot:

DBgetSerieStructure(\$s\_id, \$lang)

UIcreateSerieForm(\$P\_name, \$serie\_id, \$url, \$structure, \$serie, \$comment, \$lang)

LIBcheckSerieData(\$structure, \$metricsdata)

DBaddSerie(\$serie, \$P\_name, \$s\_id, \$comment)

#### Tarvittavat oikeudet:

Saa lisätä metriikkasarjan mihin tahansa projektiin

KO20

Saa lisätä metriikkasarjan omaan projektiinsa

KO21

#### Liittyvät käyttötapaukset:

Metriikkasarjan kiinnittäminen projektiin

KT17

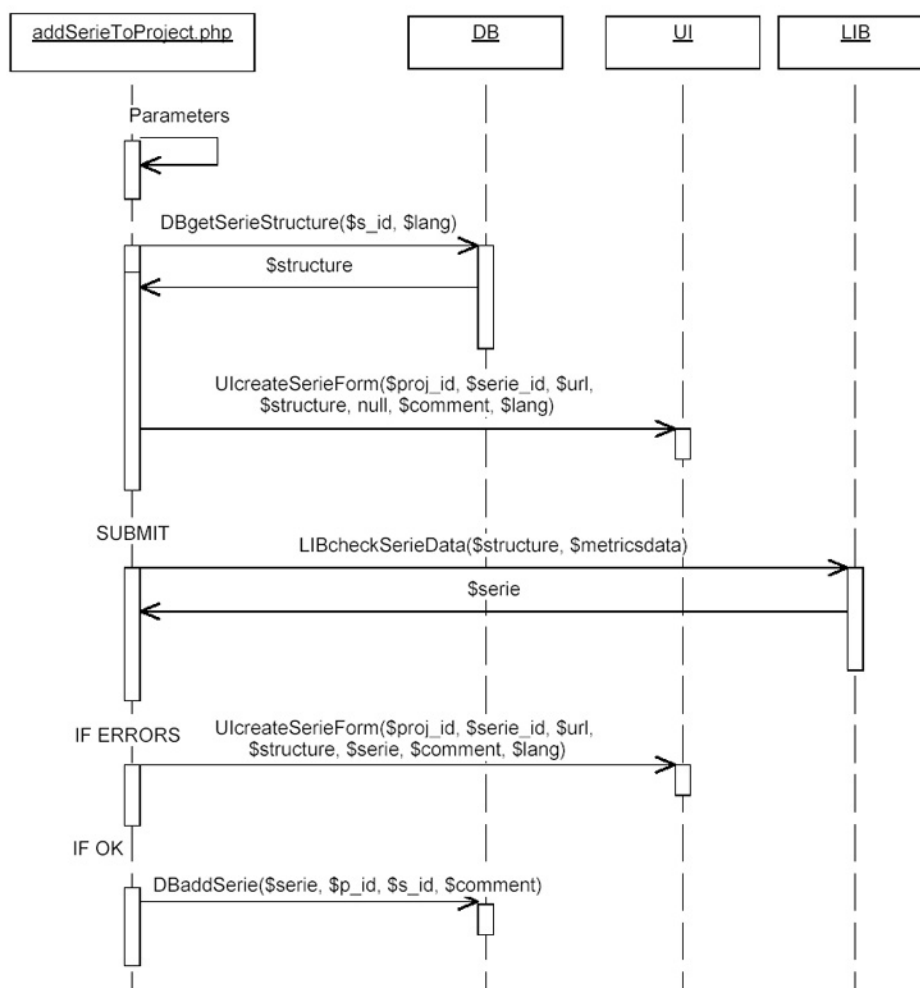
Mittaustietojen syöttö

KT19

## 4.5 addWork.php

**Kuvaus:** addWork.php on sivu, jolle päästään navigaatiovalikosta. Sivulla on lomake käyttäjän tuntikirjanpilotiedoston syöttämiseksi. Lomake sisältää vain yhden kentän, johon voidaan syöttää tiedoston nimi. Tiedosto voidaan hakea myös selaa-toiminnon avulla. Tarvittava HTML-koodi luodaan tällä sivulla.

Submit-nappia painamalla lomakkeen tiedot (tiedoston nimi ja polku) lähetetään samalle sivulle addWork.php, joka vastaanottaa tiedoston. Vastaanotettu tiedosto annetaan funktiolle \$workarray LIBparseWorkFile(\$filehandle), joka palauttaa tarkastetut työtunnit taulukossa tai false jos tiedostossa oli virhe (kts kohta 6, jossa myös kuvattu työtuntitiedoston formaatti).



Kuva 3: Sivun `addSerieToProject.php` sekvenssikaavio.

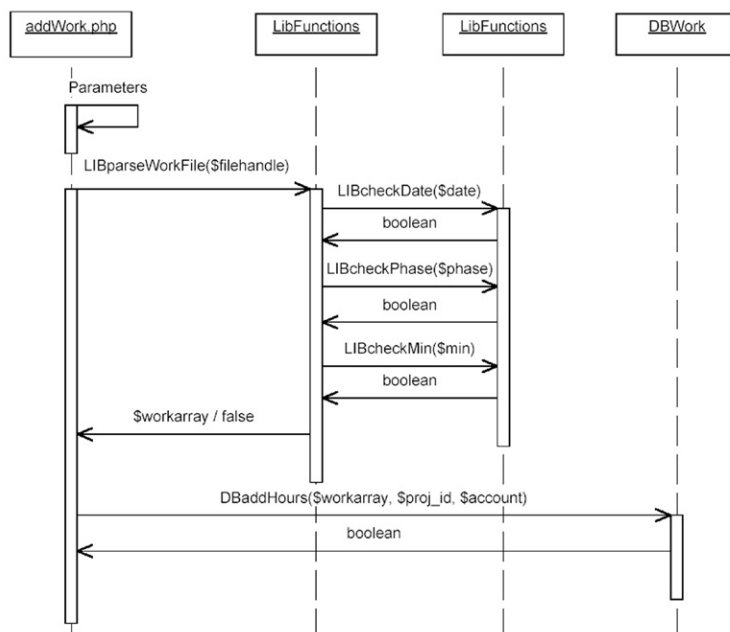
Jos tiedosto läpäisi tarkistukset ja saatiin työtuntitaulukko, ko. taulukko annetaan funktiolle `DBaddHours($workarray, $P_name, $account)`, joka tallentaa työtunnit tietokantaan. Funktio palauttaa `true` tai `false` riippuen siitä onnistuiko operaatio.

#### Sivun vastaanottamat parametrit:

<code>P_name</code>	projektin tunnus ( <code>P_name</code> )
<code>file</code>	tuntikirjanpitotiedoston nimi ja polku

#### Sekvenssikaavio:

Sivun `addWork.php` sekvenssikaavio esitetty kuvassa 4



Kuva 4: Sivun addWork.php sekvenssikaavio.

#### Sivun käyttämät funktiot:

DBaddHours(\$workarray, \$proj\_id, \$account)

LIBparseWorkFile(\$filehandle)

#### Tarvittavat oikeudet:

Saa syöttää tunteja kenelle tahansa

KO5

Saa syöttää omat tunnit omaan projektiin

KO6

#### Liittyvät käyttötapaukset:

Tuntikirjanpidon syöttö

KT16

## 4.6 authentication.php

**Kuvaus:** authentication.php on sivu, joka suorittaa käyttäjän sisäänkirjautumiseen. Sivulle siirrytään silloin kun käyttäjä on syöttänyt käyttäjätunnuksensa ja salasansa navigaatiovalikossa oleviin kenttiin ja painanut submit-nappia.

Jos käyttäjällä on oikeus kirjautua järjestelmään (K02), salasana tarkistetaan funktiolla \$loginok = DBlogin(\$account, \$password) ja jos paluuarvo on true, käyttäjätunnus lisätään sessioon ja palataan sivulle index.php. Tällöin järjestelmä tietää että käyttäjä on kirjautunut järjestelmään. Jos paluuarvo on false tai jos käyttäjällä ei ole oikeutta kirjautua järjestelmään, käyttäjälle annetaan virheilmoitus ja hänet siirretään takaisin pääsivulle index.php.

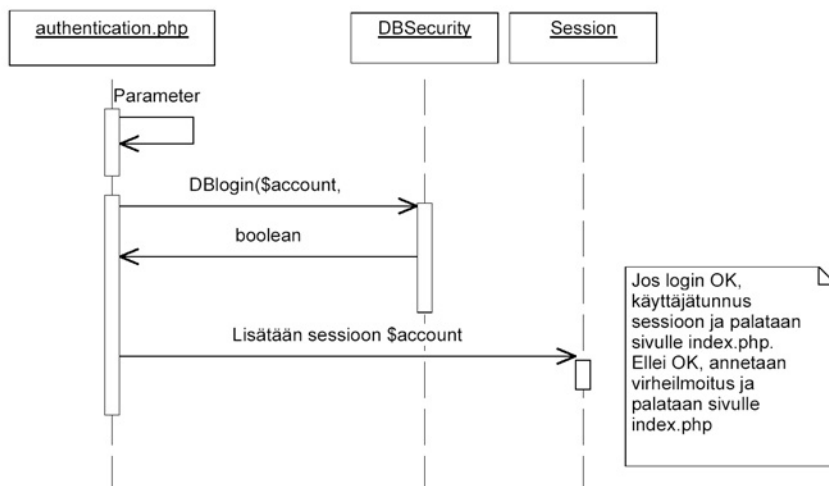
Sivun vastaanottamat parametrit on lähetettävä POST-menetelmällä, jotta salasana ei ole näkyvissä selkokielisenä URL:ssa.

### Sivun vastaanottamat parametrit:

account	Käyttäjätunnus
password	Salasana

### Sekvenssikaavio:

Sivun authentication.php sekvenssikaavio esitetty kuvassa 5



Kuva 5: Sivun authentication.php sekvenssikaavio.

### Sivun käyttämät funktiot:

DBlogin(\$account, \$password)

### Tarvittavat oikeudet:

Saa kirjautua järjestelmään

KO2

### Liittyvät käyttötapaukset:

Sisäänkirjautuminen

KT1

## 4.7 changePreferences.php

Sivu jolla käyttäjä voi muuttaa oman salasansa ja oletuskielen haluamikseen. Oletuskieltä ei varsinaisesti tueta, joten käyttöliittymässä ainoa vaihtoehto on [fi].

## 4.8 createPerson.php

**Kuvaus:** createPerson.php on sivu, jolla syötetään uuden käyttäjän tiedot järjestelmään.

**Käyttöoikeuksien käsittelystä:** Sivulle pääsy vaatii oikeutta KO12, jolla voi muuttaa käyttäjän perustietoja (alla: \$rights). Oikeuksilla KO14 luodaan lomakkeeseen kentät käyttäjän liittämiseksi projekteihin jäsenenä ja ohjaajana. Oikeuksilla KO18 lomakkeeseen tulostetaan projektitason käyttöoikeuskategoriat. Oikeuksilla KO17 lomakkeeseen tulostetaan kaikki käyttöoikeuskategoriat. Nämä tarkastukset on tehtävä paitsi lomaketta luotaessa funktiossa UIcreatePersonForm ja vastaanotettaessa dataa parametreistä. Vajaavaisilla käyttöoikeuksilla parametreistä ei saa tallettaa tietoa tietokantaan vaikka parametreissä olisikin korkeamman tason käyttöoikeuksille tarkoitettua tietoa.

Kun sivua kutsutaan ensimmäisen kerran (!isSubmit), luodaan tyhjä lomake: kutsutaan funktiota \$rights = DBgetRightsList(\$lang) ja antaa sen palauttaman oikeuskategoriataulukon funktiolle UIcreatePersonForm(\$lang, \$rights, \$person, \$url, \$userights). Tässä käyttäjän tiedot sisältävä taulukko \$person (katso alla) on tyhjä. Samoin \$userights on tyhjä.

Funktio UIcreatePersonForm luo lomakkeen, johon käyttäjän tiedot voidaan syöttää. Lomakkeen actionin URL otetaan muuttujasta \$url, ja se osoittaa tähän samaan createPerson.php -sivuun.

Kun lomakkeessa painetaan submit, muodostetaan vastaanotetuista parametreistä taulukkomuuttuja \$person:

<i>name</i>	<i>value</i>	<i>error</i>	<i>length</i>
first_name		[true false]	30
last_name		[true false]	50
account		[true false]	8
password	[!]	[true false]	10
dob	[dd.mm.yyyy]	[true false]	N/A
expiration	[dd.mm.yyyy]	[true false]	N/A
lang	[en fi]	[true false]	N/A

Lisäksi muodostetaan taulukkomuuttuja \$userights, johon talletetaan ne oikeuskategoriat, jotka käyttäjälle lomakkeessa annettiin. Parametreista member ja instructor, joista muodostetaan taulukot \$memberin ja \$instructorin. Taulukoissa on lista projektien nimistä, DBdoQuery-funktiolla tutkitaan onko projekteja olemassa tietokannassa.

Jokainen parametreina saatu arvo tarkastetaan: \$person[0..2]['Error'] = LIBcheckVarchar(\$person[0..2]['value'], \$person[0..2]['length']), \$person[4..5]['Error'] = LIBcheckDate(\$person[4..5]['value'], \$person[3]['Error'] = LIBcheckPassword(\$person[3]['value']) ja \$person[6]['Error'] = LIBcheckLang[\$person[6]['password']].

Lisäksi tarkistetaan erikseen kirjautuneen käyttäjän oikeus myöntää oikeuksia (*täsmentämättä*) ja se, että parametreinä saadut oikeuskategoriat ovat oikeellisia (löytyvät taulukosta \$rights).

Jos arvojen tarkastuksessa syntyi virheitä, luodaan lomake uudestaan jo syötettyjen mut-



ta virheellisten tietojen perusteella: UIcreatePersonForm(\$lang, \$rights, \$person, \$url, \$userrights, \$memberin, \$instructorin).

Jos parametreina saadut tiedot ovat oikeellisia, syötetään tiedot tietokantaan kutsumalla funktiota DBcreatePerson(\$person, \$userrights, \$memberin, \$instructorin).

#### Sivun vastaanottamat parametrit:

first_name	Henkilön etunimi
last_name	Henkilön sukunimi
account	Henkilön käyttäjätunnus
password	Henkilön salasana
dob	Henkilön syntymäaika, dd.mm.yyyy
expiration	Henkilön käyttöoikeuksien vanhenemisaika, dd.mm.yyyy
lang	Henkilön oletuskieli (en fi)
r_[r_id]	Lomakkeessa valitut oikeuskategoriat
member	Projektit joissa jäsenenä (projekti,projekti,..)
instructor	Projektit joissa ohjaajana (projekti,projekti,..)

#### Tarvittavat oikeudet:

Saa luoda uusia käyttäjiä	KO12
Saa liittää käyttäjiä projekteihin	KO14
Saa myöntää ja poistaa kaikkia käyttöoikeuksia	KO17
Saa myöntää projektitason käyttäjäoikeuksia	KO18

#### Liittyvät käyttötapaukset:

Henkilön lisäys projektiin	KT11
Käyttäjän lisäys	KT5

## 4.9 createProject.php

**Kuvaus:** createProject.php on sivu, jolle päästään navigaatiovalikosta, mikäli käyttäjällä on riittävät oikeudet. Sivun avulla voidaan järjestelmään lisätä uusi projekti täyttämällä sivulla olevan lomakkeen kentät.

Sivun kentät täytetään ja lähetetään sivulle itselleen. Lähetetty projektin nimi-parametri tarkastetaan LIBcheckVarchar(P\_name, 8)- funktiolla ja kuvaus tarkastetaan LIBcheckVarchar(description; Sivun tekee sille lähetetyllä projektin nimellä tietokantahaun, jossa tarkistetaan onko jo olemassa ko. niminen projekti. Tietokantakyselyyn käytetään \$result = DBdoQuery(\$query)-funktioita. \$query-parametri sisältää tietokantalauseen.

*\$queryn sisältö*

```
SELECT P_name FROM projects WHERE P_name = $p_name;
```

Jos DBdoQuery-funktio palauttama \$result-muuttuja on tyhjä, ei ole olemassa kyseisen

nimistä projektia ja projekti voidaan lisätä tietokantaan.

Jos \$result-tulosjoukko sisältää ko. tunnuksen, projekti on jo olemassa. createProject.php-sivu ladataan uudelleen ja virheellinen projektin nimi-kenttä korostetaan.

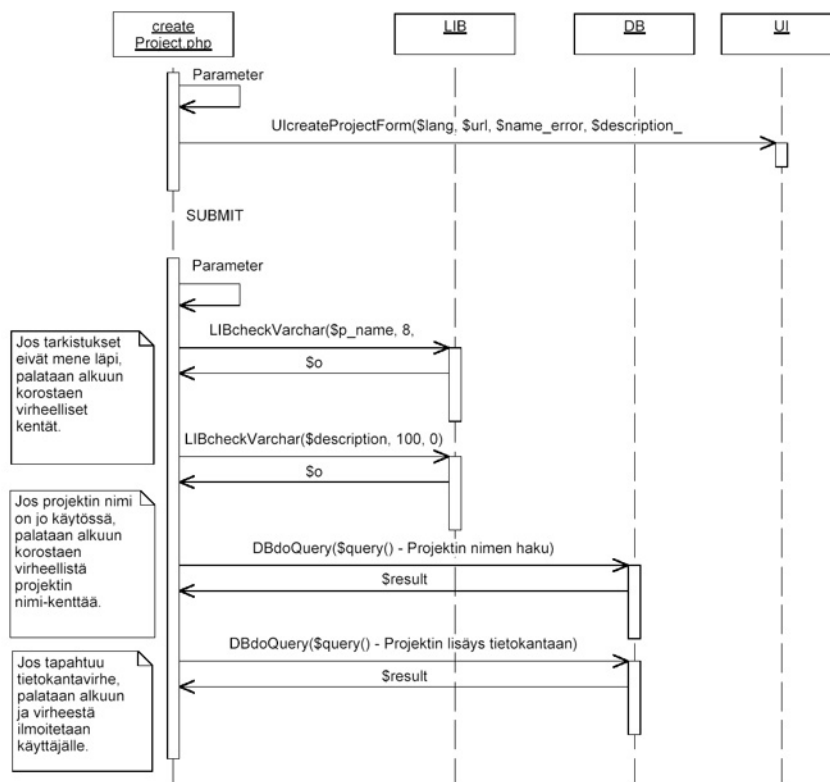
Projekti lisätään tietokantaan käyttämällä \$ok = DBdoQuery(\$query)-funktiota. \$query sisältää INSERT-lauseen, jossa projekti lisätään tietokantaan kuvauksineen. Palautusarvo \$ok sisältää totuusarvoisen tiedon siitä, onnistuiko tietokantalisäys. Jos tallennus onnistuu ladataan sama sivu tyhjänä uudestaan. Jos lisäys ei onnistu, virheestä ilmoitetaan HTTP-virhekoodeilla sekä sivulla virheilmoituksena.

Sivu luodaan käyttämällä UIcreateProjectForm(\$lang, \$url, \$name, \$description)-funktiota. Funktio luo sivulle projektin nimi- sekä kuvauskentät ja tallennuspainikkeen. Tallennuspainike lähettää lomakkeen kentät createProject.php- sivulle.

#### Sivun vastaanottamat parametrit:

p_name	projektin nimi (P_name)
description	projektin kuvaus (Description)
lang	käyttäjän kieli (en tai fi)

**Sekvenssikaavio:** Sivun createProject.php sekvenssikaavio esitetty kuvassa 6



Kuva 6: Sivun createProject.php sekvenssikaavio.

**Sivun käyttämät funktiot:**

DBdoQuery(\$query)

UIcreateProjectForm(\$lang, \$url, \$name, \$description)

LIBcheckVarchar(\$p\_name, \$minlen, \$maxlen)

**Tarvittavat oikeudet:**

Saa kirjautua järjestelmään	KO2
Saa luoda uusia projekteja	KO10

**Liittyvät käyttötapaukset:**

Projektin luonti	KT8
------------------	-----

**4.10 error.php**

**Kuvaus:** Yleinen virhesivu joihinkin virhetapauksiin.

**4.11 gradeProject.php**

**Kuvaus:** gradeProject.php on sivu, jolle päästään navigaatiovalikosta, jos käyttäjällä on oikeudet antaa arvosanoja käyttäjille ja projekteille. Sivun avulla käyttäjä voi antaa arvosanat projektille ja sen jäsenille.

Sivu luodaan käyttämällä UIcreateGradeForm(\$project, \$members, \$lang, \$url, \$error). Funktio luo sivulle arvosanantallennuslomakkeen. Lomakkeella on kunkin projektin opiskelijajäsenen kohdalla tekstikenttä, johon arvosana voidaan kirjoittaa. Arvosanat voivat olla esitäytettyjä jos ne ovat syötetty tietokantaan aiemmin. Sivulle generoituu myös tallenna-painike, joka lähettää kentät \$url -parametrin sisältämälle gradeProject.php -sivulle. Parametri \$allok kertoo, onko sivulle syötetty tieto virheellistä. Siten \$allok -parametrin pitää olla tyhjä kun tyhjä sivu ladataan. UIcreateGradeForm-funktio korostaa ne kentät, joihin syötetyt arvosanat \$project ja \$members-muuttujissa ovat virheellisiä.

Sivulla on kentät projektin arvosanalle sekä kaikkien jäsenten arvosanoille. Projektin ja sen opiskelijajäsenten tiedot haetaan tietokannasta DBgetProjectMembers(\$P\_name) ja DBgetProjectInformation(\$P\_name)-funktioilla. Kaikki funktiolla saadut arvosanat esitäytetään lomakkeeseen. Kentät, joiden arvosanat puuttuvat tietokannasta, jätetään tyhjiksi. Täytetyt kentät lähetetään sivulle itselleen.

Parametreina saatujen kenttien oikeellisuus tarkistetaan LIBcheckGrade(\$val)-funktioilla, joka tarkistaa sisältääkö \$val oikeellisen arvosanan. Jos sivulle syötetään parametreina virheellisiä arvosanoja, sivu ladataan uudelleen lähettäen sille parametreina virheiden mukaan muokattu \$error-parametri. Uudelleenlatauksen yhteydessä virheelliset kentät korostetaan. Sivuhakee projektien jäsenet DBGetProjectMembers(\$p\_name)-funktioilla ja projektin tiedot DBgetProjectInformation(\$p\_name)-funktioilla. Parametreina saatuja arvo-

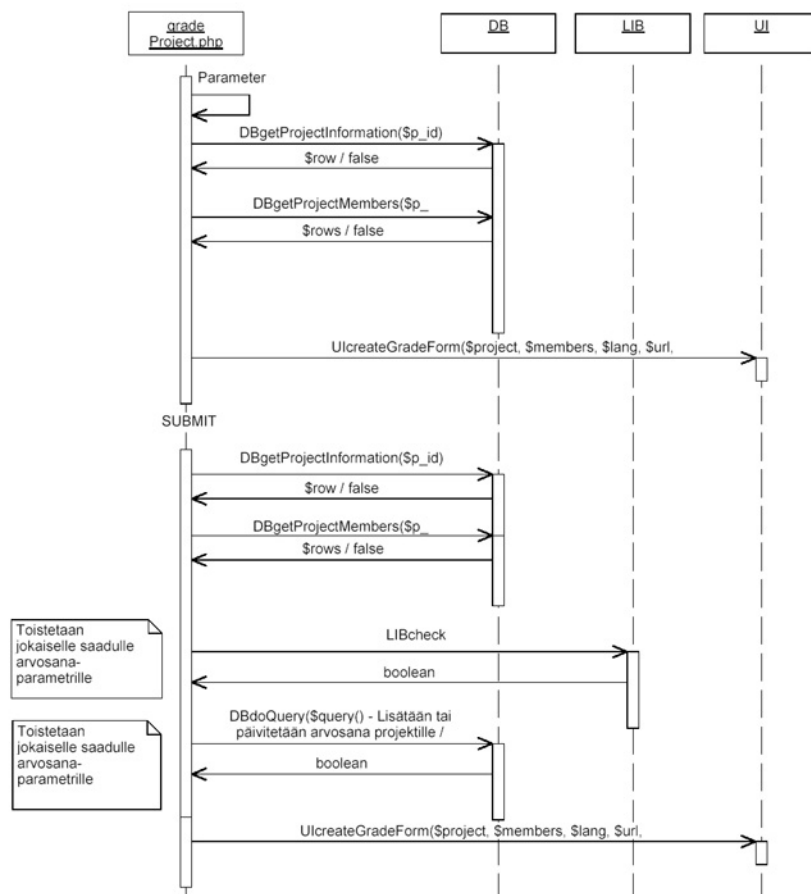
sanoja verrataan funktioiden palauttamiin arvosanatietoihin. \$members -taulukko sisältää mm. käyttäjien arvosanat ja \$project projektin arvosanan. Tietokantaan tallennetaan vain ne arvot, jotka poikkeavat tietokannan aiemmasta arvosanasisällöstä. Projektin ohjaajille ei tallenneta arvosanoja.

Muuttuneet arvot lisätään tai päivitetään tietokantaan DBdoQuery(\$query)-funktiolla. Päivitettäviin tietoihin käytetään UPDATE-kyselyä ja lisättäviin INSERT-kyselyä. INSERT-kyselyä tulee käyttää silloin, kun tietokantakyselyn tuloksena saatujen \$project tai \$members-taulukoiden grade-sarake on tyhjä. DBdoQuery(\$query)-funktiota käytetään erikseen jokaiselle muuttuneelle arvosanalle. Funktio palauttaa \$ok-muuttujassa totuusarvon jokaisen kyselyn onnistumisesta. Jos kysely ei onnistu, virheestä ilmoitetaan HTTP -virhekoodeilla sekä sivulla virheilmoituksena.

### Sivun vastaanottamat parametrit:

p_name	projektin yksikäsitteinen tunniste (P_name)
lang	käyttäjän kieli (en tai fi)

**Sekvenssikaavio:** Sivun gradeProject.php sekvenssikaavio esitetty kuvassa 7



Kuva 7: Sivun gradeProject.php sekvenssikaavio.

**Sivun käyttämät funktiot:**

DBdoQuery(\$query)

DBgetProjectInformation(\$P\_name)

DBgetProjectMembers(\$P\_name)

UIcreateGradeForm(\$project, \$members, \$lang, \$url, \$error)

LIBcheckGrade(\$val)

**Tarvittavat oikeudet:**

Saa syöttää arvosanoja mille tahansa projektille tai käyttäjälle    KO23  
(+selaus)

Saa syöttää arvosanoja omille projekteille ja näiden jäsenille    KO24  
(+selaus)

**Liittyvät käyttötapaukset:**

Arvosanojen syöttö    KT12

**4.12 index.php**

**Kuvaus:** index.php on järjestelmän pää- ja aloitussivu. Sivulla ei käytännössä tehdä muuta kuin kutsutaan funktioita UIheader(\$account), UIfooter(\$account) ja UIcreateNaviMenu(). Onnistuneen (tai epäonnistuneen) sisäänkirjautumisen jälkeen käyttäjä siirretään tälle sivulle. Sama pätee uloskirjautumiseen.

**4.13 modifyPersonInformation.php**

**Kuvaus:** Luo lomakkeen, jolle haetaan käyttäjän tiedot muutettavaksi, vastaanottaa lomakkeen, tarkistaa tiedot ja tallettaa tiedot tietokantaan. Tällä sivulla muutetaan nimenomaan käyttäjän tietoja ja käyttöoikeuksia niillä käyttöoikeuksilla, joilla voi luoda uusia käyttäjiä ja joko myöntää projektitason tai laajemman tason käyttöoikeuksia.

**Käyttöoikeuksien käsittelystä:** Sivulle pääsy vaatii oikeutta KO12, jolla voi muuttaa käyttäjän perustietoja (alla: \$rights). Oikeuksilla KO14 luodaan lomakkeeseen kentät käyttäjän liittämiseksi projekteihin jäsenenä ja ohjaajana. Oikeuksilla KO18 lomakkeeseen tulostetaan projektitason käyttöoikeuskategoriat. Oikeuksilla KO17 lomakkeeseen tulostetaan kaikki käyttöoikeuskategoriat. Nämä tarkastukset on tehtävä paitsi lomaketta luotaessa funktiossa UImodifyPersonForm ja vastaanotettaessa dataa parametreistä. Vajaavaisilla käyttöoikeuksilla parametreistä ei saa tallettaa tietoa tietokantaan vaikka parametreissä olisikin korkeamman tason käyttöoikeuksille tarkoitettua tietoa.

Tultaessa sivulle ensimmäistä kertaa (ei submit) muodostetaan createPerson.php:n \$rights- ja \$person-hashtableja vastaavat taulut funktioilla DBgetRights (\$lang, \$account) ja DBgetPersonInformation(\$account). \$memberin- ja \$instructorin-hashtablet muodostetaan DBdoQuery-funttiolla.

Lomake muodostetaan funktiolla `UImodifyPersonForm($lang, $rights, $person, $url, $user-rights, $memberin, $instructorin)`. Erotuksena `createPerson.php`:n lomakkeeseen henkilöä ei voi irrottaa projektista, mutta käyttäjän voi lisätä uusiin projekteihin.

Submit: parametrit luetaan ja tarkistetaan kuten `createPerson.php`:ssa. Jos tiedoissa on virheitä, muodostetaan lomake uudestaan funktiolla `UImodifyPersonForm`.

Jos tiedot ovat oikein suoritetaan toimenpiteet: `$person` sisältö päivitetään (UPDATE), User-rights-taulusta poistetaan vanhat oikeudet ja lisätään uudet hashtablesta `$rights`. Käyttäjä lisää `$memberin` ja `$instructorin` -taulujen mukaisesti uusiin projekteihin.

Huom: henkilöiden poistamista projektista ei voi tehdä lomakkeen kautta, kyseessä on poikkeustoimi.

#### Sivun vastaanottamat parametrit:

Kts. `createPerson.php`

#### Tarvittavat oikeudet:

Saa luoda uusia käyttäjiä	KO12
Saa liittää käyttäjiä projekteihin	KO14
Saa myöntää ja poistaa kaikkia käyttöoikeuksia	KO17
Saa myöntää projektitason käyttäjäoikeuksia	KO18

#### Liittyvät käyttötapaukset:

Henkilön lisäys projektiin	KT11
Käyttäjätietojen muokkaus	KT6

## 4.14 modifyProjectInformation.php

**Kuvaus:** `modifyProjectInformation.php` on sivu, jolle päästään navigaatiovalikosta kun klikataan toimintoa *Projektin perustiedot* tietyn projektin alla. Sivulla tarkistetaan onko käyttäjällä oikeus muuttaa projektin perustietoja funktiolla `$hasright = DBhasRight($account, $KO)`.

Jos sivulle päästään ilman submit-napin painallusta, haetaan projektin nykyiset perustiedot funktiolla `$project = DBgetProjectInformation($P_name)`. Tämän jälkeen luodaan lomake projektin perustietojen muokkausta varten funktiolla `UcreateModifyProjectForm($lang, $project, $url, $error)`. Lomakkeen submit johtaa samalle sivulle `modifyProjectInformation.php` ja parametreina annetaan uudet perustiedot.

Jos sivulle päästään submit-napin painalluksella, projektin perustiedot tallennetaan tietokantaan. Tiedot saadaan parametreina. Tietojen oikeellisuus tarkistetaan funktioilla `LIBcheckVarchar($p_name, $minlen, $maxlen)` ja `LIBcheckDate($date)`. Jos tiedoissa oli virheitä, lomake piirretään uudestaan funktiolla `UcreateModifyProjectForm()` ja taulukko-muotoisessa parametrissa `$error` kerrotaan missä tiedoissa oli virhe. Katso ko. funktion määrittelystä `$error` taulukkoparametrin muoto.

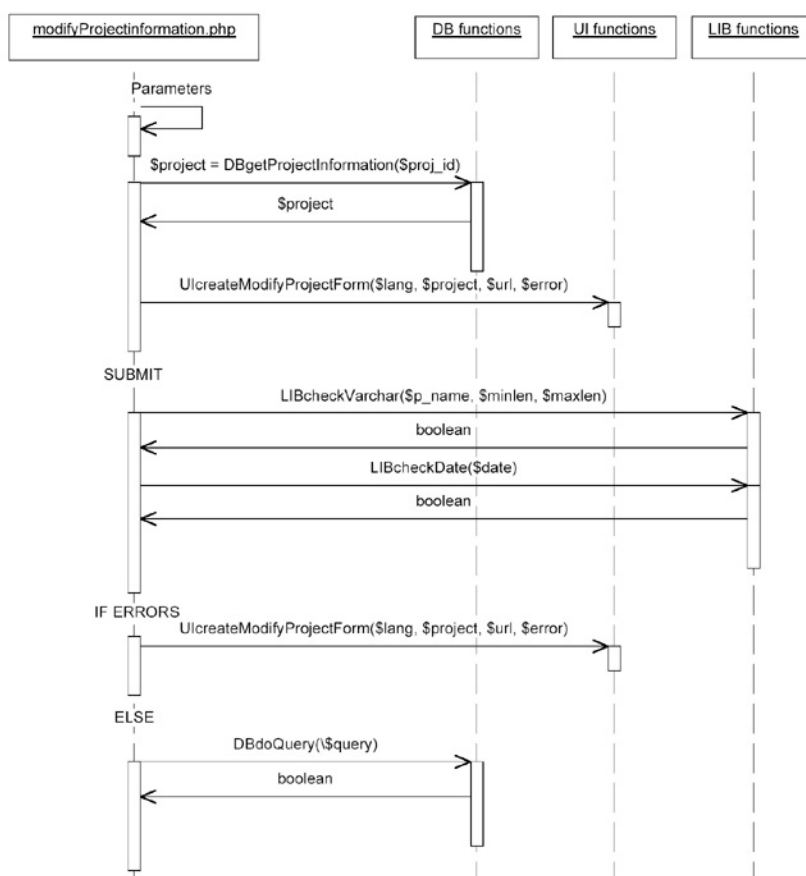
Jos tiedot oli oikein itse tallennus tapahtuu funktiolla DBsaveProjectInformation(\$project, \$P\_name). Parametri \$project on samanmuotoinen taulukko kuin mitä aiemmin saatiin funktiolla DBgetProjectInformation.

### Sivun vastaanottamat parametrit:

P_name	Projektin nimi
Description	Projektin kuvaus
Begin_date	Projektin aloituspvm
End_date	Projektin päättymispvm
Homepage	Projektin kotisivu
Process_model	Projektissa käytetty prosessimalli
Project type	Projektin tyyppi, esim. informaatiojärjestelmät

### Sekvenssikaavio:

Sivun modifyProjectInformation.php sekvenssikaavio esitetty kuvassa 8



Kuva 8: Sivun modifyProjectInformation.php sekvenssikaavio.

### Sivun käyttämät funktiot:

DBhasRight(\$account, \$KO)

DBgetProjectInformation(\$P\_name)

DBsaveProjectInformation(\$project, \$P\_name)  
 LIBcheckVarchar(\$p\_name, \$minlen, \$maxlen)  
 LIBcheckDate(\$date)  
 UIcreateModifyProjectForm(\$lang, \$project, \$url, \$error)

**Tarvittavat oikeudet:**

Saa muuttaa minkä tahansa projektin perustietoja	KO3
Saa muuttaa oman projektinsa perustietoja	KO4

**Liittyvät käyttötapaukset:**

Projektin perustietojen muokkaus	KT9
----------------------------------	-----

## 4.15 modifySerie.php

**Kuvaus:** modifySerie.php on sivu, jolle päästään navigaatiovalikosta. Sivun näyttää käyttäjälle listan projektiin liitettyistä metriikkasarjoista, joista valitaan metriikkasarja, jonka dataa halutaan muokata.

Metriikkasarjat haetaan tietokannasta funktiolla \$series = DBgetSeriesOfProject(\$P\_name, \$lang). Parametri \$lang on muotoa "fi"tai "en"ja paluuarvo on hash table (kts. kohta 7.5).

Lista luodaan funktiolla UIcreateSerieList(\$series). Funktion luoman listan metriikkasarjat ovat parametreilla varustettuja linkkejä sivulle modifySerieInProject.php.

**Sivun vastaanottamat parametrit:**

P_name	projektin nimi
lang	käyttäjän kieli (en tai fi)

**Sivun käyttämät funktiot:**

DBgetSeriesOfProject(\$P\_name, \$lang)  
 UIcreateSerieList(\$series)

**Tarvittavat oikeudet:**

Saa lisätä metriikkasarjan mihin tahansa projektiin	KO20
Saa lisätä metriikkasarjan omaan projektiinsa	KO21

**Liittyvät käyttötapaukset:**

Mittaustietojen syöttö	KT19
------------------------	------



## 4.16 modifySerieInProject.php

**Kuvaus:** modifySerieInProject.php on sivu, jolle päästään sivulta modifySerie.php. Tällöin käyttäjä on jo valinnut projektista minkä metriikkasarjan dataa muokataan, jonka jälkeen kyseisen metriikkasarjan pakolliset metriikat on täytettävä. Sivun saa parametreina mm. projektin nimen ja metriikkasarjan id:n.

Ensin haetaan tietokannasta metriikkasarjan rakenne funktiolla `$structure = DBgetSerieStructure($s_id, $lang)`. Tämän jälkeen metriikkasarjan sisältö tulostetaan käyttäjälle funktiolla `UcreateSerieForm($P_name, $serie_id, $url, $structure, $serie, $comment, $lang)`. Parametri `$structure` on funktion `DBgetSerieStructure` palauttama taulukko.

Nyt käyttäjällä on mahdollisuus muokata metriikkasarjan metriikkojen dataa. Kun käyttäjä painaa submit-nappia, siirrytään takaisin sivulle `addSerieToProject.php`. Parametreina saadaan syötetyt metriikat parametreissa `metric_x`, jossa `x` viittaa metriikan id:hen. Ensin tarkistetaan syötettyjen tietojen oikeellisuus funktiolla `$serie = LIBcheckSerieData($structure, $metricsdata)`. Parametri `$structure` on edelleen sama, jonka `DBgetSerieStructure` palauttaa. Parametri `$metricsdata` on taulukko, joka on määritelty funktion `LIBcheckSerieData` kuvauksessa ja se sisältää käyttäjän metriikkasarjaan täyttämät tiedot.

Funktion `LIBcheckSerieData` paluuarvona saatu `$serie` sisältää syötetyt tiedot sekä mahdolliset virheet. Parametri on määritelty ko. funktion kuvauksen yhteydessä. Jos tiedoissa oli virheitä, luodaan uudestaan lomake funktiolla `UcreateSerieForm` ja tällä kertaa parametri `$serie` on sama tietorakenne kuin mitä `LIBcheckSerieData` palautti. Jos virheitä ei ollut, tiedot tallennetaan tietokantaan funktiolla `DBupdateSerie($serie, $P_name, $s_id, $comment)`. Tässäkin tapauksessa parametri `$serie` on sama tietorakenne kuin mitä `LIBcheckSerieData` palautti.

### Sivun vastaanottamat parametrit:

<code>P_name</code>	projektin nimi
<code>serie_id</code>	Projektiin lisättävän sarjan id
<code>MP_id</code>	projekt-serie-yhdisteen uniikki id ( <code>MP_id</code> )
<code>metric_x</code>	Metriikkaan syötetty tieto, <code>x</code> viittaa metriikan id:hen
<code>lang</code>	Käyttäjän kieli (en tai fi)

### Sivun käyttämät funktiot:

`DBgetSerieStructure($s_id, $lang)`  
`UcreateSerieForm($P_name, $serie_id, $url, $structure, $serie, $comment, $lang)`  
`LIBcheckSerieData($structure, $metricsdata)`  
`DBupdateSerie($serie, $P_name, $s_id, $comment)`

### Tarvittavat oikeudet:

Saa lisätä metriikkasarjan mihin tahansa projektiin ja muuttaa syötettyjä mittaustietoja	KO20
Saa lisätä metriikkasarjan omaan projektiinsa	KO21

ja muuttaa syötettyjä mittaustietoja

#### **Liittyvät käyttötapaukset:**

Mittaustietojen syöttö KT18

### **4.17 projectInformation.php**

**Kuvaus:** Projektin perustiedot, projektin jäsenet, kiinnitetyt metriikkasarjat, linkki sivulle projectInformationHours.php.

Jos projekti on aktiivinen ts. käynnissä, sen tietoja ei näytetä ilman vaadittuja oikeuksia (KO30) ellei aktiivinen käyttäjä ole itse projektin jäsen.

#### **Tarvittavat oikeudet:**

Saa nähdä aktiivisten projektien tietoja KO30

#### **Liittyvät käyttötapaukset:**

Aktiivisten projektien projektikohtaiset näkymät KT14, KT15

### **4.18 projectInformationHours.php**

**Kuvaus:** Projektiin syötetyt tunnit jokaisella projektin jäsenellä. Parametroitavissa näyttämään työtunnit vaiheittain ja viikottain.

Jos projekti on aktiivinen ts. käynnissä, sen tietoja ei näytetä ilman vaadittuja oikeuksia (KO30) ellei aktiivinen käyttäjä ole itse projektin jäsen.

#### **Tarvittavat oikeudet:**

Saa nähdä aktiivisten projektien tietoja KO30

#### **Liittyvät käyttötapaukset:**

Aktiivisten projektien projektikohtaiset näkymät KT14

### **4.19 projectMetricSearch.php**

**Kuvaus:** projectMetricSearch.php on sivu, jolla etsitään projektien täyttämiä metriikkatietoja. Metriikoiden tiedot näytetään isossa taulukossa projektikohtaisesti. Kun sivua kutsutaan ensimmäisen kerran (lisSubmit), luodaan tyhjä lomake: kutsutaan funktiota \$metricseries = DBgetAvailableSeries(\$lang) ja käy sen palauttaman metriikkasarjataulukon läpi rivi riviltä foreach \$line in \$metricseries

`$metricseries[i][structure] = DBgetSerieStructure($metricseries[i][s_id],$lang)`  
 Nyt lähetetään saatu taulukko funktiolle `UIcreateMetricSearchForm($lang, $metricSeries, $url)`.

Funktio `UIcreateMetricSearchForm` luo lomakkeen, jossa valitaan ne metriikat tai metriikkasarjat joiden tietoja halutaan tarkastella. Lomakkeen actionin URL otetaan muuttujasta `$url`, ja se osoittaa tähän samaan `projectMetricSearch.php` -sivuun.

Kun lomakkeessa painetaan submit, saadaan `$_post` muuttujassa kaksi taulukkoa. Toisessa on lueteltu kaikki valittujen metriikoiden `m_id` ja toisessa kaikkien valittujen metriikkasarjojen `s_id` eli `metric[m_id]` ja `serie[s_id]`.

Ensin haetaan sivulle kaikki projektit, joiden tietoja halutaan tarkastella. `$active` parametri kertoo haetaanko aktiivisten vai vanhojen projektien tietoja. Tämä taas riippuu siitä onko hakuun tultu aktiivisten projektien linkistä vai vanhojen projektien linkistä. `$projects = DBgetProjects($lang,$active)`.

Seuraavaksi käydään läpi lomakkeelta saatu taulukko valituista metriikkasarjoista `$serie[s_id]` ja pyydetään tietokanta hakemaan kaikki sarjaan liittyvät metriikat ja lisätään ne `$metric[m_id]` tauluun.

```
foreach $line in $serie
$pastetMetric=DBgetMetricsFromSerie($line)
foreach $line in $pastetMetric
$metric[i]=$line
```

Nyt meillä on taulu kaikista haettavista metriikoista ja kaikista projekteista. Muodostetaan taulukko jonka ensimmäinen sarake on projektin nimi ja liitetään projektit tähän tauluun. Seuraavien sarakkeiden nimet muodostetaan tietokantahaulla `$metric[]` taulussa olevien metriikoiden id numeroiden avulla ja samalla haetaan taulukkoon jokaisen projektin tähän metriikkaan kuuluva data. `foreach$line in $metric`  
`$name = DBgetMetricName($line, $lang)` `foreach $line2 in $projects $result[i][$name] = DBgetMetricData($line, $line2)`

Saatu `$result` taulukko on muotoa

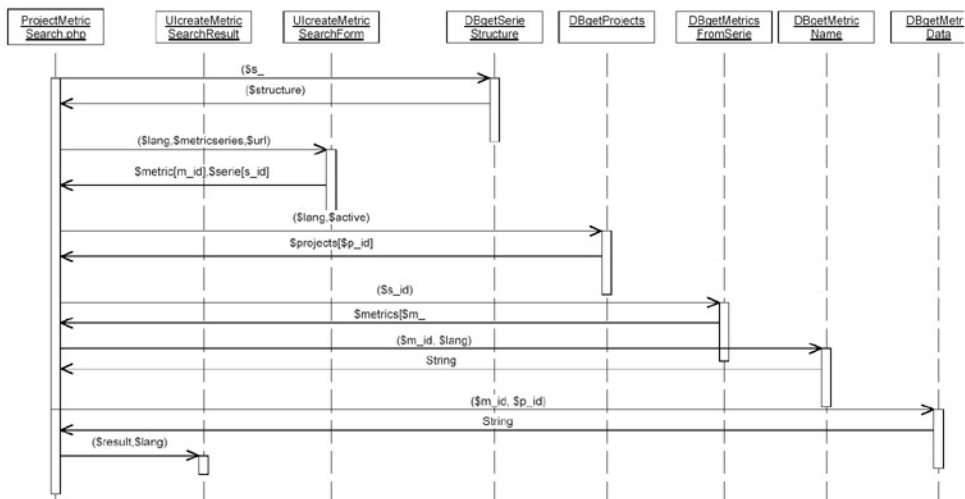
<i>name</i>	<i>value</i>
<code>project_name</code>	String
<code>metric_name</code>	String
<code>metric_name</code>	String
..	

Tämän jälkeen lähetetään saatu taulukko käyttöliittymäfunktiolle, joka muodostaa haun tuloksista taulukon. `UIcreateMetricSearchResult($result, $lang)`

**Sekvenssikaavio:** Sivun `projectMetricSearch.php` sekvenssikaavio esitetty kuvassa 9

**Sivun käyttämät funktiot:**

`DBgetAvailableSeries($lang)`



Kuva 9: Sivun projectMetricSearch.php sekvenssikaavio.

```

DBgetSerieStructure($s_id,$lang)
UIcreateMetricSearchForm($lang,$metricSeries,$url)
DBgetProjects($lang,$active)
DBgetMetricsFromSerie($m_id)
DBgetMetricName($m_id, $lang)
DBgetMetricData($m_id, $P_name)
UIcreateMetricSearchResult($result, $lang)

```

#### Tarvittavat oikeudet:

Saa nähdä aktiivisten projektien tietoja

KO30

#### Liittyvät käyttötapaukset:

Projektien selaus ja haku järjestelmästä

KT15

## 4.20 search.php

**Kuvaus:** search.php on sivu, jonne päästään navigaatiovalikosta. Sivun avulla voi järjestelmästä hakea projekteja osan perustietojen perusteella. Sivua voi käyttää kuka vain, mutta aktiiviset projektien tulokset sisältyvät hakuun vain, jos käyttäjällä on siihen tarvittavat oikeudet.

Sivun luo UIcreateSearchForm(\$lang, \$procmodels, \$projtypes, \$url, \$text)-funktio. Sivulla on nimikenttä, valintalaatikat prosessimalleille sekä projektityypeille, tekstikentät alkamis- ja päättymispäiville ja Haku-painike. Painiketta painamalla kenttien sisältö lähetetään search.php-sivulle. Valintalaatikoiden sisältö haetaan tietokannasta tietokantafunktiolla. DBgetListContent(\$l\_name, \$lang) hakee listan nimen perusteella kaikki listaan liittyvät alkio. Funktiota käytetään sekä prosessimallien että projektityyppien valintalaatikolistojen luomiseen.

Parametreina saatujen arvojen oikeellisuus tarkistetaan. Kenttien ei tarvitse olla täytetty. Täytetyt kentät tulkitaan hakuehdoiksi, tyhjät kentät eivät taas rajaa hakua millään lailla. \$name-parametri tarkistetaan käyttämällä LIBcheckVarchar(\$name,0,8)- funktiota. Alkamis- ja päättymispäivämäärät tarkistetaan LIBcheckDate(\$date)-funktiolla. Jos joku tarkistusfunktioista palauttaa falsen totuusarvonaan, virhemuuttujiin merkitään virhetilanne ja sivu ladataan uudestaan.

Parametreista pt\_[Elem\_id] ja pm\_[Elem\_id] muodostetaan taulukot, \$p\_types ja \$p\_models, jotka sisältävät valitut projektityypit ja projektimallit.

Haku suoritetaan käyttämällä DBsearchProjects(\$name, \$p\_models, \$p\_types, \$startdate, \$enddate)-funktiota. Funktiolle välitetään parametreina hakua rajaavat tiedot. Funktio palauttaa taulukon, joka sisältää kaikki hakuehtoihin täsmänneiden projektien tiedot. Jos käyttäjällä ei ole oikeutta tutkia aktiivisten projektien tietoja (KO30), taulukosta poistetaan aktiiviset projektit. Päätyneet projektit tunnistetaan siitä, että projektin tietojen End\_date-tietue sisältää arvon. Aktiivisilla projekteilla ko. tietue on tyhjä.

Hakutulos esitetään taulukossa. Taulukon sarakkeiden nimet muodostetaan \$lang -muuttujan perusteella.

#### **Sivun vastaanottamat parametrit:**

name	projektin nimi
pt_[Elem_id]	lomakkeessa valitut projektityypit
pm_[Elem_id]	lomakkeessa valitut projektimallit
startdate	projektin alkamispäivämäärä (dd.mm.yyyy)
enddate	projektin päättymispäivämäärä (dd.mm.yyyy)
lang	käyttäjän kieli (en tai fi)

**Sekvenssikaavio:** Sivun search.php sekvenssikaavio esitetty kuvassa 10

#### **Sivun käyttämät funktiot:**

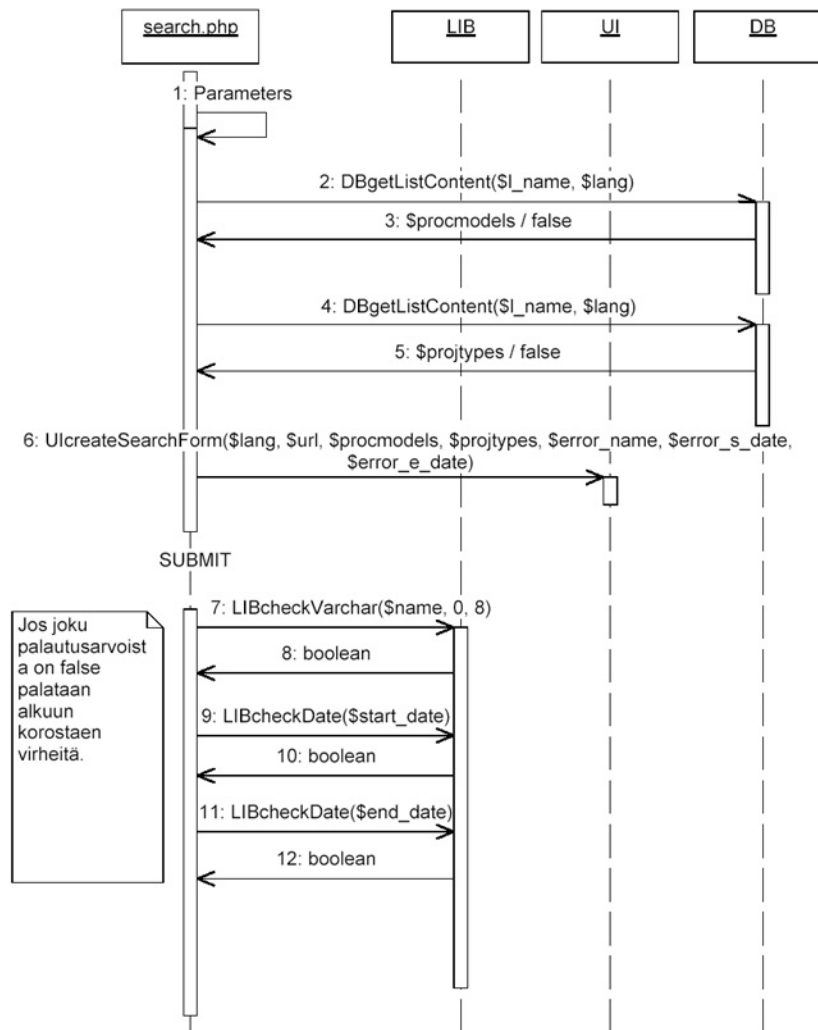
DBsearchProjects(\$name, \$p\_models, \$p\_types, \$startdate, \$enddate)  
 DBgetListContent(\$l\_name, \$lang)  
 UIcreateSearchForm(\$lang, \$procmodels, \$projtypes, \$url, \$text)  
 LIBcheckVarchar(\$p\_name, \$minlen, \$maxlen)  
 LIBcheckDate(\$date)

#### **Tarvittavat oikeudet:**

Saa nähdä aktiivisten projektien tietoja KO30

#### **Liittyvät käyttötapaukset:**

Projektien selaus ja haku järjestelmästä KT15



Kuva 10: Sivun search.php sekvenssikaavio.

## 4.21 showHelp.php

**Kuvaus:** showHelp.php on sivu, joka tulostaa käyttäjälle sivuun tai metriikkasarjaan liittyvän suoraohjeen. showHelp.php saa parametrina joko sitä kutsuvan PHP-sivun nimen tai metriikkasarjan id-numeron. Esim. help\_id = addSerie tai help\_id = 35.

Liittyvä ohje haetaan funktiokutsulla \$helptext = LIBgetHelpText(\$help\_id, \$lang). Saat suoraohjeen sisältö annetaan funktiolle UIshowHelp(\$helptext, \$lang), joka tulostaa ohjeen HTML-sivuksi.

### Sivun vastaanottamat parametrit:

help_id	suoraohjeen id (help_id)
lang	käyttäjän kieli (en tai fi)

### Tarvittavat oikeudet:

N/A

## 5 Käyttöliittymätason funktiot

Tässä luvussa kuvataan yksityiskohtaisesti järjestelmän käyttöliittymäfunktiot. Funktioille yhteistä on, että ne tulostavat HTML:ää ja sisältävät vain siihen liittyvää sovelluslogiikkaa.

### **function UIcreateSerieList(\$series)**

Luo ja tulostaa listan projektiin liitettävissä olevista metriikkasarjoista. Parametri \$series on taulukko metriikkasarjoista, jonka funktio DBgetAvailableSeries palauttaa (kts. kohta 7.5). Listassa esiintyvä metriikkasarja on linkki sivulle addSerieToProject.php tarvittavilla parametreilla.

ECHO

```
<A HREF="addSerieToProject?proj_id=p_id&
      serie_id=s_id&
      lang=fi/en">
$serie_name</A>
$description
```

### **function UIcreateSerieForm(\$P\_name, \$serie\_id, \$url, \$structure, \$serie, \$comment, \$lang)**

Luo ja tulostaa lomakkeen, jolla käyttäjä voi täyttää pakolliset tiedot ennen kuin sarja liitetään projektiin. Parametri \$url kertoo mihin lomakkeen action johtaa. Parametri \$structure sisältää metriikkasarjan rakenteen ja se on funktion DBgetSerieStructure palauttama taulukko. Parametri \$comment viittaa metriikkasarjan kuvaukseen (struct\_value). Parametri \$serie (joka voi olla myös null) sisältää mahdollisesti käyttäjän jo aiemmin syöttämät tiedot sekä mahdolliset virheet. Parametri \$serie on määritelty funktion LIBcheckSerieData kuvauksen yhteydessä.

### **function UIcreatePersonForm((\$lang, \$rights, \$person, \$url, \$userrights, \$memberin, \$instructorin))**

Tulostaa HTML-lomakkeen uuden käyttäjän luomiseksi saamiensa parametrien mukaan joko tyhjänä tai esitäytettynä, mikäli tiedot ovat virheellisiä.

### **function UImodifyPersonForm((\$lang, \$rights, \$person, \$url, \$userrights, \$memberin, \$instructorin))**

Tulostaa HTML-lomakkeen käyttäjän tietojen muuttamiseen. Muoto sama kuin UIcreatePersonForm -funktiossa.

### **function UIshowHelp(\$helpstring, \$lang)**

Tulostaa suoraohjesivun ja sille parametrinaan saamansa \$helpstring -muuttujan sisällön.

### **function UIcreateProjectForm(\$lang, \$url, \$name, \$description)**

Tulostaa projektin luontisivun lomakkeineen. Lomake sisältää projektin nimi- sekä kuvauskentät ja tallennuspainikkeen. Parametri \$lang sisältää sivulla käytettävän kielen, \$url sisältää tallennuspainikkeen URL:n, Kaksialkioinen taulukko \$name sisältää sekä halutun projektin nimen että tiedon virheellisestä projektin nimikentästä. Annettu nimi sijaitsee

\$name[0]-alkiossa. Jos nimi on oikeellinen \$name[1] sisältää true-totuusarvon, muutoin falsen. \$description-parametri sisältää tiedon kuvauskenttään syötetyt tiedot taulukossa. Sen rakenne on samanlainen, kuin \$name-parametrilla. Virheelliset kentät korostetaan.

Projektin nimi syötetään tekstikenttään, jonka enimmäispituus on kahdeksan merkkiä. Kuvauskenttä on tekstikenttä, jonka enimmäispituus on 100 merkkiä.

#### **UIcreateModifyProjectForm(\$lang, \$project, \$url, \$error)**

Tulostaa lomakkeen projektin perustietojen muokkausta varten. Taulukkomuotoinen parametri \$project sisältää projektin nykyiset perustiedot ja sen muoto on sama kuin mitä funktio DBgetProjectProjectInformation palauttaa. Parametri \$url kertoo minne lomakkeen action johtaa. Taulukkomuotoinen parametri \$error kertoo missä tiedoissa oli mahdollisesti muotovirhe. Parametrin tietoja käytetään lomakkeen uudelleenpiirrosta, eli käyttäjälle kerrotaan mitkä tiedot on korjattava ennen kuin muutokset voidaan tallentaa tietokantaan. Taulukko \$error on yksiulotteinen ja jokaisessa alkiossa voidaan sanallisesti ilmaista virhe. Alkioiden järjestys on ['Name', 'Description', 'Begin date', 'End date', 'Homepage', 'Process model', 'Project type'].

#### **function UIcreateGradeForm(\$project, \$members, \$lang, \$url, \$error)**

Funktio tulostaa arvostelulomakkeen. Lomake sisältää projektin nimen, jokaisen käyttäjän etunimen, sukunimen sekä tekstikenttiä, joihin projektin ja henkilöiden arvosanat voidaan syöttää. Arvosanat esitätetään, mikäli \$members tai \$project-tilat sisältävät ne. Lomakkeeseen ei tulosteta \$members-tilan käyttäjiä, joiden rooli on ohjaaja. Ne tekstikentät korostetaan, joiden \$members tai \$project-tilan arvosanatiedoissa on virheitä. \$error-tila ilmaisee virheellisten kenttien olemassaolon.

#### **function UIcreateSearchForm(\$lang, \$procmodels, \$projtypes, \$url, \$text)**

Funktio tulostaa hakulomakkeen. Lomake sisältää nimikentän, alkamis- ja päättymispäivämääräkentät sekä prosessimalli- ja prosessityyppivalintalistat. Valintalistat prosessimalleihin ja projektityyppeihin saadaan parametreista \$procmodels ja \$projtypes. Lomake sisältää myös Haku-painikkeen, joka lähettää tiedot \$url-parametrin osoittamalle sivulle. \$text -parametri on kaksiulotteinen taulukko ja se sisältää name-, s\_date- ja e\_date-kentät sekä tiedot niihin syötetyistä virheellisistä arvoista. Tekstikenttään syötetty projektin nimi sijaitsee \$text[0][0]-alkiossa. Jos nimi on oikeellinen \$text[0][1] sisältää true-totuusarvon, muutoin falsen. s\_date ja e\_date-kentät ovat käyttäytyvät samoin ja ovat vastaavasti \$text[1][0] ja \$text[2][0] -alkioissa. Virheelliset kentät korostetaan.

#### **function UIcreateMetricSearchForm(\$lang, \$metricseries, \$url)**

Funktio tulostaa hakulomakkeen. Lomake koostuu kaikista mahdollisista metriikkasarjojen nimistä ja metriikkasarjan nimen alla luetelluista siihen sarjaan kuuluvista metriikoista. Jokaisen nimen vieressä on valintaruutu. <input type="checkbox" value="s\_id/m\_id" name="metric[]/serie[]>contentname/serienname. Action toimintoon liitetään url parametrilla \$url.

#### **function UIcreateMetricSearchResult(\$result, \$lang)**

Funktio muodostaa saamastaan tietotaulukosta taulukon sivulle. Parametrina saatu taulukko on jo oikean muotoinen, joten se tarvitsee vain tulostaa näkyviin.

#### **function UIcreateWorkTable(\$lang, \$projects, \$grouping)**



Tulostaa aktiivisten projektien työtunnit:

```
foreach $project in $projects
foreach $group in $grouping
tulosta projektin nimi ja tunnit vaiheittain;
```

### **function UIcreateNaviMenu(\$userrights, \$projMembership)**

Funktio luo jokaisella sivulla esiintyvän navigaatiovalikon. Jos käyttäjä ei ole kirjautunut järjestelmään, eli sessiossa ei ole käyttäjätunnusta, funktio luo vakiovalikon.

Jos käyttäjätunnus on tallennettu sessioon, käyttäjä on kirjautunut järjestelmään. Funktio saa parametrina käyttäjän oikeudet taulukossa \$userrights. Tämä taulukko on samaa muotoa kuin mitä funktio DBgetRights palauttaa. Funktio saa parametrina myös käyttäjän projektijäsenyydet taulukossa \$projMembership. Tämä taulukko on samaa muotoa kuin mitä funktio DBgetProjectMembership palauttaa.

Tämän jälkeen luodaan navigaatiovalikko projektijäsenyyksien ja oikeuksien perusteella. Projektijäsenyydet määräävät mitkä projektit ovat näkyvissä navigaatiovalikossa. Varsinaiset linkit toimintoihin määräytyvät käyttäjän oikeuksien mukaan. Luettelo oikeuksista ja niihin liittyvistä toiminnoista löytyvät määrittelydokumentin liitteistä.

### **function UIheader(\$account)**

Funktio generoi sivulle headerin include-komennolla:

```
<!--include virtual="/include/header-fi.html-->
```

### **function UIfooter(\$account)**

Funktio generoi sivulle laitoksen footerin include-komennolla:

```
<!--include virtual="/include/trailer-fi.html-->
```

## **6 Kirjasto- ja työkalufunktiot**

Tässä luvussa kuvataan yksityiskohtaisesti järjestelmän työkalu-, kirjasto- ja apufunktiot.

### **function LIBgetHelp(\$help\_id, \$lang), return \$helpstring)**

Lukee suoraohjeet sisältävästä tiedostosta parametrinaan saamansa ohjeen-id mukaisen ohjeen ja palauttaa sen.

### **function LIBcheckPassword(\$password), return true|false**

Tarkistaa parametrinaan saamansa salasanan (mm. pituus > 0) ja palauttaa totuusarvon sen mukaan onko salasana kelvollinen.

### **function LIBparseWorkFile(\$filehandle), return \$workarray / false**

Saa parametrina tiedostoviitteen, jonka avulla tiedosto avataan. Tämän jälkeen tiedostosta luetaan jokainen rivi. Jokaisella rivillä on tabulaattorilla eroteltuja tietoja seuraavassa muodossa:

<i>PVM</i>	<i>LUOKKA</i>	<i>TUNNIT</i>	<i>TEHTÄVÄ</i>
12.3.2004	VA	2	dokumentin kirjoittamista
14.3.2004	SU	3	ohjelman rakenteen suunnittelua

Rivin tiedot tarkistetaan funktioilla `LIBcheckDate($date)`, `LIBcheckPhase($phase)` ja `LIBcheckMin($min)`. Kaikki tarkistusfunktiot palauttavat joko `true` tai `false`. Jos tarkistuksessa ei ilmennyt virheitä, luodaan ja palautetaan taulukko, jossa tiedot ovat samassa järjestyksessä kuin tiedostossakin. Jos tapahtui virhe, palautetaan taulukon sijasta `false`.

**function LIBcheckDate(\$date), return boolean**

Tarkistaa parametrina annetun päivämäärän oikeellisuuden. Oikea muoto on DD.MM.YYYY. Palauttaa `true` tai `false`.

**function LIBcheckPhase(\$phase), return boolean**

Tarkistaa parametrina annetun työvaiheen oikeellisuuden. Työvaiheita ovat mm "PS", "VA", "SU", "TO", "TE". Palauttaa `true` tai `false`.

**function LIBcheckMin(\$min), return boolean**

Tarkistaa parametrina annetun minuuttimäärän oikeellisuuden. Parametrin on oltava kokonaisluku. Palauttaa `true` tai `false`.

**function LIBcheckVarchar(\$p\_name, \$minlen, \$maxlen), return boolean**

Funktio tarkistaa, että sille syötetty `$p_name`-parametri sisältää vain kirjaimia tai numeroita. Funktio tarkastaa myös sille syötetyn parametrin pituuden oikeellisuuden `$minlen` ja `$maxlen`-parametrien perusteella. `$minlen` sisältää tarkistettavan vähimmäispituuden ja `$maxlen` sisältää enimmäispituuden. Jos `$parametri` on kaikilta osiltaan oikeellinen palautetaan `true`, muutoin `false`.

**function LIBcheckInteger(\$integer, \$length), return boolean**

Funktio tarkistaa, että sille annettu arvo on tyyppiä `integer`.

**function LIBcheckDecimal(\$decimal, \$length), return boolean**

Funktio tarkistaa, että sille annettu arvo on tyyppiä `decimal`.

**function LIBcheckGrade(\$val), return boolean**

Funktio tarkistaa, että sille syötetty `$val`-parametri sisältää oikeanlaisen arvosanan. Sallitut arvosanat pienimmästä alkaen ovat 1, 1+, 2-, 2, 2+, 3-, 3. Jos `$val` on oikean muotoinen palautetaan `true`, muuten `false`.

**LIBcheckSerieData(\$structure, \$metricsdata), return \$serie**

Funktiolla tarkistetaan metriikkasarjaan syötettyjen tietojen oikeellisuus. Funktio käyttää apunaan funktioita `LIBcheckDate($date)`, `LIBcheckDecimal($decimal, $length)`, `LIBcheckInteger($integer, $length)` ja `LIBcheckVarchar($p_name, $minlen, $maxlen)`.

Parametri `$structure` on taulukko, jonka muoto on sama kuin mitä `DBgetSerieStructure` palauttaa. Se kuvaa tietyn metriikkasarjan rakenteen. Parametri `$metricsdata` sisältää käyttäjän (sarjaan) syöttämät tiedot ja se on kaksiulotteinen taulukko, jossa aina viitataan tiettyyn contentiin [`'content_id'`, `'value'`].

Tämä funktio palauttaa aina hashtablein `$serie`, jonka muoto on [`'content_id'`, `'value'`, `'error_description'`]. Jos tiedoissa ei ollut virheitä, sarake `'error_description'` on tyhjä. Jos virheitä esiintyi, niistä mainitaan sarakkeessa `'error_description'`.

## 7 Tietokantatason funktiot

Tässä luvussa kuvataan tietokantatason palvelut, eli funktiot. Funktiot on kategorisoitu toiminnallisuuden mukaan eri include-tiedostoihin.

### 7.1 DBMisc

Include-tiedosto *DBMisc* sisältää yleisiä tietokantafunktioita.

**function DBgetListContent(\$l\_name, \$lang), return int**

Palauttaa tietokannasta listoina olevat tietotyypit taulukossa.

**function DBdoQuery(\$query) return hashtable / boolean**

Funktiolle syötetään parametrina sql-kysely, joka suoritetaan tietokannassa. Tietokantakyselyn perusteella palautetaan joko tulosjoukko tai totuusarvo kyselyn onnistumisesta.

### 7.2 DBPerson

Include-tiedosto *DBPerson* sisältää käyttäjiin liittyviä tietokantafunktioita.

**function DBcreatePerson(\$person, \$userights, \$memberin, \$instructorin, return boolean**

Lisää uuden käyttäjän tietokantaan, antaa tälle käyttöoikeudet ja liittää tämän projekteihin jäseneksi tai ohjaajaksi.

**function DBgetProjectMembership(\$account), return hashtable**

Palauttaa taulukon niistä projekteista, johon käyttäjä kuuluu. Taulukko sisältää projektitunnukset ja projektinimet.

*SQL-lauseet:*

```
SELECT a.P_name, b.P_name FROM Person_in_project AS a
      JOIN Projects AS b
      ON a.P_name = b.P_name WHERE a.account = $account
```

**function DBgetPersonInformation(\$account), return \$person**

Hakee Person-tilusta käyttäjän tiedot ja palauttaa ne hashtablessa \$person.

### 7.3 DBSecurity

Include-tiedosto *DBSecurity* sisältää käyttöoikeuksiin liittyviä tietokantafunktioita.

**function DBlogin(\$account, \$password), return boolean**

Funktio tarkistaa jos annettu käyttäjätunnus ja salasana ovat oikein. Salasana muutetaan ensin PHP:n kirjastofunktiolla MD5-muotoon (32 merkkiä).

*SQL-lauseet:*

```
SELECT * FROM Person
        WHERE account = $account AND Password = $MD5Pwd
```

**function DBgetRightsList(\$lang), return \$rights**

Palauttaa tietokannasta kaikki käyttöoikeuskategoriat taulukossa \$rights, jossa: ['name', 'r\_id'].

**function DBhasRight(\$account, \$KO), return boolean**

Funktion avulla voidaan kysyä jos tietyllä käyttäjällä on tietty oikeus.

*SQL-lauseet:*

```
SELECT * FROM Person_rights
        WHERE account = $account AND R_id = $KO
```

**function DBgetRights(\$lang, \$account), return \$userrights**

Hakee User\_rights-taulusta \$account-parametrin mukaiset käyttöoikeuskategoriat ja palauttaa ne hashtablessa \$userrights.

## 7.4 DBProject

Include-tiedosto *DBProject* sisältää projekteihin liittyviä tietokantafunktioita.

**function DBgetProjectMembers(\$P\_name), return hashtable**

Funktiolla haetaan kaikki tiettyyn projektiin liittyvät henkilöt. \$P\_name on haettavan projektin tunnus. Funktio palauttaa arvonaan tietokantahaun tuloksena syntyneen hashtablen, jossa on kaikki haettavaan projektiin liittyvät henkilöt.

**function DBgetProjectInformation(\$P\_name), return hashtable**

Funktiolla haetaan tietyn projektin tiedot. \$P\_name on haettavan projektin tunnus. Funktio palauttaa arvonaan tietokantahaun tuloksena syntyneen hashtablen, jossa on haettavan projektin tiedot.

*SQL-lauseet:*

```
SELECT P_name, Description, Begin_date,
        End_date, Project_type, Process_model,
        Grade, Homepage
FROM Projects
```

**function DBsaveProjectInformation(\$project, \$P\_name), return boolean**

Funktio tallentaa olemassaolevan projektin muuttuneet tiedot. Parametri \$project on taulukko, jonka muoto on sama kuin mitä DBgetProjectInformation palauttaa. Ennen varsinaista UPDATE-lausetta funktiossa pitää tarkistaa, että annettu projekti (\$P\_name:n perusteella) on olemassa. Jos operaatio onnistui, palautetaan true. Jos esim. projektia ei ole olemassa tai UPDATE-lause epäonnistuu, palautetaan false.

*SQL-lauseet:*

```
UPDATE Projects SET
  P_name = $projName, Description = $description
  Begin_date = $beginDate, End_date = $endDate,
  Homepage = $homepage, Process_model = $processModel,
  Project_type = $projectType, M_date = GETDATE()
WHERE P_name = projid
```

### **function DBgetProjects(\$lang,\$active), return hashtable**

Funktio hakee kaikki projektit, jotka ovat päättyneet tai aktiivisia, riippuen \$active parametrissa. Paluuarvona palautuu taulukko joka on seuraavaa muotoa:

<i>Sarake</i>	<i>Tyyppi</i>	<i>Selite</i>
p_name	varchar(50)	Projektin nimi.

*SQL-lauseet:*

```
if $active=="active"
SELECT p_name_ FROM Projects
      WHERE End_date IS NULL;
else
SELECT p_name_ FROM Projects
      WHERE End_date IS NOT NULL;
```

## **7.5 DBMetric**

Include-tiedosto *DBMetric* sisältää metriikkasarjoihin liittyviä tietokantafunktioita.

### **function DBgetAvailableSeries(\$lang), return hashtable**

Funktiolla haetaan kaikki projektiin liitettävissä olevat metriikkasarjat. Parametri \$lang kertoo käyttäjän UI-kielen ja on muotoa "fi"tai "en". Paluuarvo on tietokantahaun tuloksena syntynyt hashtable, jonka rakenne on:

<i>Sarake</i>	<i>Tyyppi</i>	<i>Selite</i>
s_id	number(11)	Metriikkasarjan uniikki id
serie_name_fi	varchar(50)	Metriikkasarjan nimi. Huom! Jos käyttäjän kieli on englanti, sarakkeen nimi on serie_name_en
description_fi	varchar(250)	Metriikkasarjan kuvaus. Huom! Jos käyttäjän kieli on englanti, sarakkeen nimi on description_en

*SQL-lauseet:*

```
SELECT s_id, serie_name_$lang AS serie_name,
      description_$lang AS description
FROM Metric_series WHERE visible = 1;
```

### **function DBgetMetricsFromSerie(\$s\_id), return hashtable**

Funktio hakee kaikki sarjaan liittyvien metriikoiden id:t ja palauttaa ne taulukkona.

*SQL-lauseet:*

```
SELECT Content_in_Serie.c_id FROM Content_in_Serie
WHERE Content_in_Serie.s_id = $s_id AND
      Content.Content_type = "metr" AND
      Content_in_Serie.c_id = Content.c_id;
```

**function DBgetMetricName(\$m\_id, \$lang), return String**

Funktio hakee metriikan nimen tietokannasta ja jos metriikka on taulukkomuotoisen metriikan lapsi, muodostetaan nimi siten, että nimi alkaa äitimetriikan nimellä ja jatkuu alaviivan jälkeen lapsimetriikan nimellä.

**function DBgetMetricData(\$c\_id, \$P\_name), return String**

Funktio hakee sisällön id:n avulla taulusta Content sisällön datatyyppin. Tämän jälkeen haetaan oikeantyyppinen tieto Metrics\_data taulusta. Jos tietoja löytyi useampia, yhdistetään ne yhdeksi Stringiksi pilkkujen avulla.

*SQL-lauseet:*

```
$type=SELECT Data_type FROM Content
WHERE C_id = $c_id
```

```
$result = SELECT $type from Metrics_data
WHERE P_name = $P_name and C_id = $c_id
```

**function DBgetMetricsData(\$MP\_id), return table**

Funktio hakee Metrics\_data taulusta datasisällön.

*SQL-lause:*

```
\$metrics=SELECT c_id, number_data, varchar_data, date_data,
decimal_data FROM Metrics_data
WHERE MP_id = $MP_id
```

**function DBgetSerieComment (\$MP\_id), return string**

Funktio hakee Serie\_in\_project-taulusta metriikkasarjan kommenttikentän.

Tämä kommentti on lähinnä metriikkasarjan projektiin liittäneen henkilön huomautus,

*SQL-lause:*

```
\$metrics=SELECT Serie_comment FROM Serie_in_project
WHERE MP_id = $MP_id
```

**function DBgetSerieStructure(\$s\_id, \$lang), return hashtable**

Funktio hakee tietokannasta tietyn metriikkasarjan (parametri \$s\_id) rakenteen ja palauttaa sen taulukkona.

*SQL-lause:*

```
SELECT c.C_id, cs.Struct_value_$lang AS Struct_value,
       c.Content_type, c.Content_name_$lang AS Content_name,
       c.Data_type, c.Desc_$lang AS Description, c.Required
FROM Content AS c JOIN Content_in_Serie AS cs
ON cs.C_Id = c.C_Id
WHERE cs.S_Id = $s_id AND c.Visible = 1
```

Huom! Lisäksi on haettava taulusta Parent\_content tieto siitä, onko jokin content jonkun toisen contentin lapsi. Myöskin, jos jonkin contentin data\_type on "LIST", pitää hakea kyseisen listan tiedot List\_elements taulusta. Nämä tiedot pitäisi palauttaa tässä samassa \$structure hashtableissa.

**function DBaddSerie(\$serie, \$P\_name, \$s\_id, \$comment), return boolean**

Funktio liittää metriikkasarjan projektiin ja tallentaa sarjaan käyttäjän syöttämät pakolliset tiedot. *SQL-lause:*

```
BEGIN TRANSACTION
INSERT INTO Serie_in_project VALUES
      ($s_id, $P_name, $comment)
INSERT INTO Metrics_data VALUES
      (values from $serie)
-- IF OK
COMMIT TRANSACTION
-- ELSE
ROLLBACK TRANSACTION
```

**function DBgetSeriesOfProject(\$P\_name, \$lang) return hashtable**

Funktio palauttaa taulukon projektin metriikkasarjojen ID:stä, eli lista mitkä metriikkasarjat on kiinnitetty projektiin.

**function DBupdateSerie(\$serie, \$P\_name, \$s\_id, \$comment) return boolean**

Funktio päivittää tietokantaan muokatun metriikan datat.

## 7.6 DBWork

Include-tiedosto *DBWork* sisältää työtuntikirjanpitoon liittyviä tietokantafunktioita.

**function DBaddHours(\$workarray, \$P\_name, \$account), return boolean**

Funktio saa parametrina taulukon työtunneista. Taulukon muoto on kuvattu kohdassa 6. Koska uusi työtuntitiedosto korvaa käyttäjälle mahdollisesti aiemmin tallennetut työtunnit, poistetaan ensin aiemmat tunnit. Funktio palauttaa true tai false riippuen siitä onnistuiko operaatio.

*SQL-lauseet:*

```
DELETE FROM Work WHERE account = $account
BEGIN TRANSACTION
```

```

-- for each row in $workarray
INSERT INTO Work VALUES
    ($account, $P_name, hours, phase, Work_date)
-- if OK
COMMIT TRANSACTION
-- else
ROLLBACK TRANSACTION

```

## 8 Järjestelmän tietosisältö

Tässä luvussa kuvataan järjestelmän tietosisältö.

### 8.1 Tietokantakaavio

Tietokantakaavio on esitetty kuvassa 11.

### 8.2 Tietokantataulut

#### 8.2.1 Projects

##### Kuvaus

Taulu *Projects* sisältää järjestelmän projektien tiedot.

##### Kardinaliteetti

Projects : Person\_in\_project N:N, Serie\_in\_project 1:N, Metrics\_data 1:N

##### Attribuutit

P_name	varchar(8)	Projektin nimi ja uniikki tunniste
Description	varchar(100)	Projektin kuvaus
C_date	timestamp	Projektitietueen luontiajankohta
M_date	timestamp	Projektitietueen viimeinen muokkausajankohta
Begin_date	date	Projektin aloituspäivämäärä
End_date	date	Projektin päättymispäivämäärä
Project_type	varchar(20)	Projektin tyyppi
Process_model	varchar(20)	Projektin malli
Grade	varchar(2)	Projektin arvosana (1-, ... , 3)
Homepage	varchar(200)	Projektin kotisivun URL-osoite

##### Pääavain

P\_name varchar(8)

##### Viiteavaimet

N/A



### 8.2.2 Person

#### Kuvaus

Taulu *Person* sisältää järjestelmän käyttäjien tiedot.

#### Kardinaliteetti

Person : Person\_in\_project N:N, Person\_Rights N:N

#### Attribuutit

Account	varchar(8)	Käyttäjän käyttäjätunnus ja tunniste
First_name	varchar(30)	Käyttäjän etunimi
Last_name	varchar(50)	Käyttäjän sukunimi
Date_of_birth	datetime	Käyttäjän syntymäaika
C_date	timestamp	Päivämäärä, jolloin käyttäjä on luotu
M_date	timestamp	Päivämäärä, jolloin käyttäjän tietoja on muutettu
Default_lang	varchar(2)	Käyttäjän kieli (FI, EN)
Expiration_date	date	Päivämäärä, jolloin käyttäjätunnus vanhenee
Grade	varchar(2)	Käyttäjän arvosana
Password	varchar(32)	Käyttäjän salasana (MD5)

#### Pääavain

Account varchar(8)

#### Viiteavaimet

N/A

### 8.2.3 Person\_in\_project

#### Kuvaus

Taulu *Person\_in\_project* kiinnittää käyttäjät projekteihin.

#### Kardinaliteetti

Person\_in\_project: Person 1:1, Projects 1:1

#### Attribuutit

Account	number(11)	Projektin jäsenen tunnus
P_name	number(11)	Projektin tunnus
Role	varchar(4)	Henkilön rooli projektissa: PRM - Project member INST - Project instructor

#### Pääavain

(Account, P\_name)

#### Viiteavaimet

(P\_name) viittaa Projects-tauluun (Account) viittaa Person-tauluun

### 8.2.4 Work

#### Kuvaus

Taulu *Work* sisältää projektien jäsenten työtunnit projektikohtaisesti per henkilö.

#### Kardinaliteetti

Work: Person\_in\_project 1:1

#### Attribuutit

H_id	number(11)	Tietueen uniikki tunniste
Account	varchar(8)	Projektin jäsenen tunnus
P_name	varchar(8)	Projektin tunnus
Hours	number(4,1)	Tuntien lukumäärä puolen tunnin tarkkuudella
Phase	varchar(2)	Projektin vaiheen tunnuskirjaimet
Work_date	date	Päiväys.
Work_week	number(2)	Viikon numero

#### Pääavain

(H\_id)

#### Viiteavaimet

(Account, P\_name) viittaa Person\_in\_project-tauluun

#### Sekvenssit

Work\_seq, jonka parametrit: START WITH 1 INCREMENT BY 1 NOMAXVALUE NOCACHE NOCYCLE

### 8.2.5 User\_rights

#### Kuvaus

Taulu *User\_rights* sisältää eri käyttöoikeustasot.

#### Kardinaliteetti

User\_rights: Person\_rights N:N

#### Attribuutit

R_id	number(11)	Tietueen uniikki tunniste
Name_fi	varchar(100)	Käyttöoikeuden nimi suomeksi
Name_en	varchar(100)	Käyttöoikeuden nimi englanniksi

#### Pääavain

(R\_id)

#### Viiteavaimet

N/A

## 8.2.6 Person\_rights

### Kuvaus

Taulu *Person\_rights* liittää yhteen käyttöoikeudet ja henkilöt.

### Kardinaliteetti

Person\_rights: Person 1:1, User\_rights 1:1

### Attribuutit

R_id	number(11)	Käyttöoikeuden tunnus
Account	number(11)	Käyttäjätunnus

### Pääavain

(R\_id, Account)

### Viiteavaimet

(Account) viittaa Person-tauluun.

(R\_id) viittaa User\_rights-tauluun.

## 8.2.7 Content

### Kuvaus

Taulu *Content* sisältää kaikki metriikkasarjojen elementit; metriikat (yksittäiset metriikat, taulukot, listat) ja sarjojen taittoelementit (otsikot, selitykset).

Sisältöelementit erotellaan tyypeittäin taulun *Content\_type* -kentällä.

### Kardinaliteetti

### Attribuutit

C_id	number(11)	Sisältöelementin uniikki tunniste
Content_type	varchar2(4)	Elementin tyyppi (ots1..3, sel1..3, metr, taul, list)
Content_name_fi	varchar2(50)	Elementin nimi suomeksi
Content_name_en	varchar2(50)	Elementin nimi englanniksi
Data_type	varchar2(10)	Metriikan tietotyyppi (varchar, number, decimal, date)
C_date	date	Päivämäärä, jolloin metriikka on luotu
M_date	date	Päivämäärä, jolloin metriikan tietoja on muutettu
Desc_fi	varchar2(50)	Elementin kuvaus suomeksi
Desc_en	varchar2(50)	Elementin kuvaus englanniksi
Visible	number(1)	Näytetäänkö elementti metriikkasarjoissa (1 = true)
Required	number(1)	Onko metriikan arvon syöttäminen pakollista

### Pääavain

C\_id number(11)

**Viiteavaimet**

N/A

**8.2.8 Metric\_Series****Kuvaus**

Taulu *Metric\_series* sisältää metriikkasarjojen perustiedot.

**Kardinaliteetti****Attribuutit**

S_id	number(11)	Metriikkasarjan uniikki tunniste
Serie_name_fi	varchar2(50)	Sarjan suomeksi
Serie_name_en	varchar2(50)	Sarjan nimi englanniksi
C_date	date	Päivämäärä, jolloin sarja on luotu
M_date	date	Päivämäärä, jolloin sarjan tietoja on muutettu
Description_fi	varchar2(250)	Sarjan kuvaus suomeksi
Description_fi	varchar2(250)	Sarjan kuvaus englanniksi
Visible	number(1)	Näytetäänkö sarja käyttöliittymässä

**Pääavain**

S\_id number(11)

**Viiteavaimet**

N/A

**8.2.9 Content\_in\_serie****Kuvaus**

Taulu *Content\_in\_serie* sisältää tiedon siitä, mitä elementtejä Content-tilusta metriikkasarja sisältää ja missä järjestyksessä ne on tarkoitus listata HTML-sivuilla.

Lisäksi, rakenne-elementeillä (otsikot, selitteet) Struct value fi/en -sarakeessa on rakenne-elementin HTML-sivuilla näkyvä sisältö. Esimerkiksi sarjaan liitetyn otsikkotyypin teksti, Content\_type "OTS1-> Struct\_value\_fi "Projektisuunnitelma".

**Kardinaliteetti****Attribuutit**

CS_id	number(11)	Content-Serie-yhdisteen uniikki tunniste
C_id	number(11)	Liitetyn Content-elementin tunniste
S_id	number(11)	Sarja, johon Content-elementti liittyy
COrder	number(11)	Vapaa järjestysnumero taittoa varten
Struct_value_fi	varchar2(500)	Rakenne-elementin sarjakohtainen sisältö
struct_value_en	varchar2(500)	Rakenne-elementin sarjakohtainen sisältö

**Pääavain**

CS\_id number(11)

**Viiteavaimet**C\_id number(11) viittaa *Content*-tauluun S\_id number(11) viittaa *Metric\_series*-tauluun**8.2.10 Serie\_in\_project****Kuvaus**Taulu *Serie\_in\_project* kuvaa liitosta, jossa metrikkasarja kiinnitetään projektiin.**Kardinaliteetti**

Metric series : Projects N:N

**Attribuutit**

MP_id	number(11)	Project-Serie-yhdisteen uniikki tunniste
P_name	varchar(118)	Projekti, johon sarja liittyy
S_id	number(11)	Sarja, joka on liitetty projektiin
Serie_comment	varchar2(100)	Liitetyn sarjan kommentti

**Pääavain**

MP\_id number(11)

**Viiteavaimet**P\_name varchar(8) viittaa *Projects*-tauluunS\_id number(11) viittaa *Metric\_series*-tauluun**8.2.11 Metrics\_data****Kuvaus**Taulu *Metrics\_data* sisältää projektiin liitetyn metrikkasarjan metriikoiden mittaustiedot, eli metriikoille annetun datan. Syötetty tieto talletetaan tyyppin (*Content* -taulun *Content\_type* -sarake) mukaan tyyppikohtaiseen sarakkeeseen.**Kardinaliteetti****Attribuutit**

MP_id	number(11)	Project-Serie -yhdisteen uniikki tunniste
C_id	number(11)	Metriikka, jolle mittaustieto on syötetty
S_id	number(11)	Sarja, johon mittaustieto liittyy
P_name	varchar(8)	Projekti, johon mittaustieto liittyy
Number_data	number(20)	Integer-muotoinen data
Varchar_data	varchar2(250)	Varchar-muotoinen data
Date_data	date	Date-muotoinen data
Decimal_data	number(10,1)	Reaalilukumuotoinen data
Elem_id	number(11)	Listaelementtityyppinen data

**Pääavain**

MP\_id number(11) viittaa *Serie\_in\_project* -tauluun ja C\_id number(11) *Content* -tauluun.

**Viiteavaimet**

P\_name number(11) viittaa *Projects*-tauluun

S\_id number(11) viittaa *Metric series*-tauluun

Elem\_id varchar(8) viittaa *List\_elements* tauluun

**8.2.12 Selection\_lists****Kuvaus**

Taulu *Selection\_Lists* sisältää erilaisten listojen (metriikat, perustiedot) perustiedot.

**Kardinaliteetti****Attribuutit**

L_id	number(11)	Listan uniikki tunniste
L_name_fi	varchar2(50)	Listan nimi suomeksi
L_name_en	carchar2(50)	Listan nimi englanniksi

**Pääavain**

L\_id number(11)

**Viiteavaimet**

N/A

**8.2.13 List\_elements****Kuvaus**

Taulu *List\_elements* sisältää listojen listaelementit, ts. sisällön.

**Kardinaliteetti**

List\_elements : Selection\_lists N:1

**Attribuutit**

Elem_id	number(11)	Listaelementin uniikki tunniste
Elem_data_fi	varchar2(50)	Listaelementin varchar-muotoinen data (fi)
Elem_data_en	varchar2(50)	Listaelementin varchar-muotoinen data (en)
L_id	number(11)	Viittaus Selection_lists -tauluun

**Pääavain**

Elem\_id number(11)

**Viiteavaimet**

L\_id number(11) viittaa *Selection\_lists* -tauluun.

### 8.2.14 List\_to\_content

#### Kuvaus

Taulu *List\_to\_content* sisältää tiedon siitä, mikä lista liittyy *Content* -taulun LIST-tyyppiseen metriikkaan.

#### Kardinaliteetti

List\_to\_content : Content 1:N

#### Attribuutit

C_id	number(11)	Listaelementin uniikki tunniste
L_id	number(11)	Viittaus Selection_lists -tauluun

#### Pääavain

N/A

#### Viiteavaimet

L\_id number(11) viittaa *Selection\_lists* -tauluun ja C\_id number(11) *Content* -tauluun.

## 9 Käyttöliittymä

### 9.1 Graafinen ohjeistus

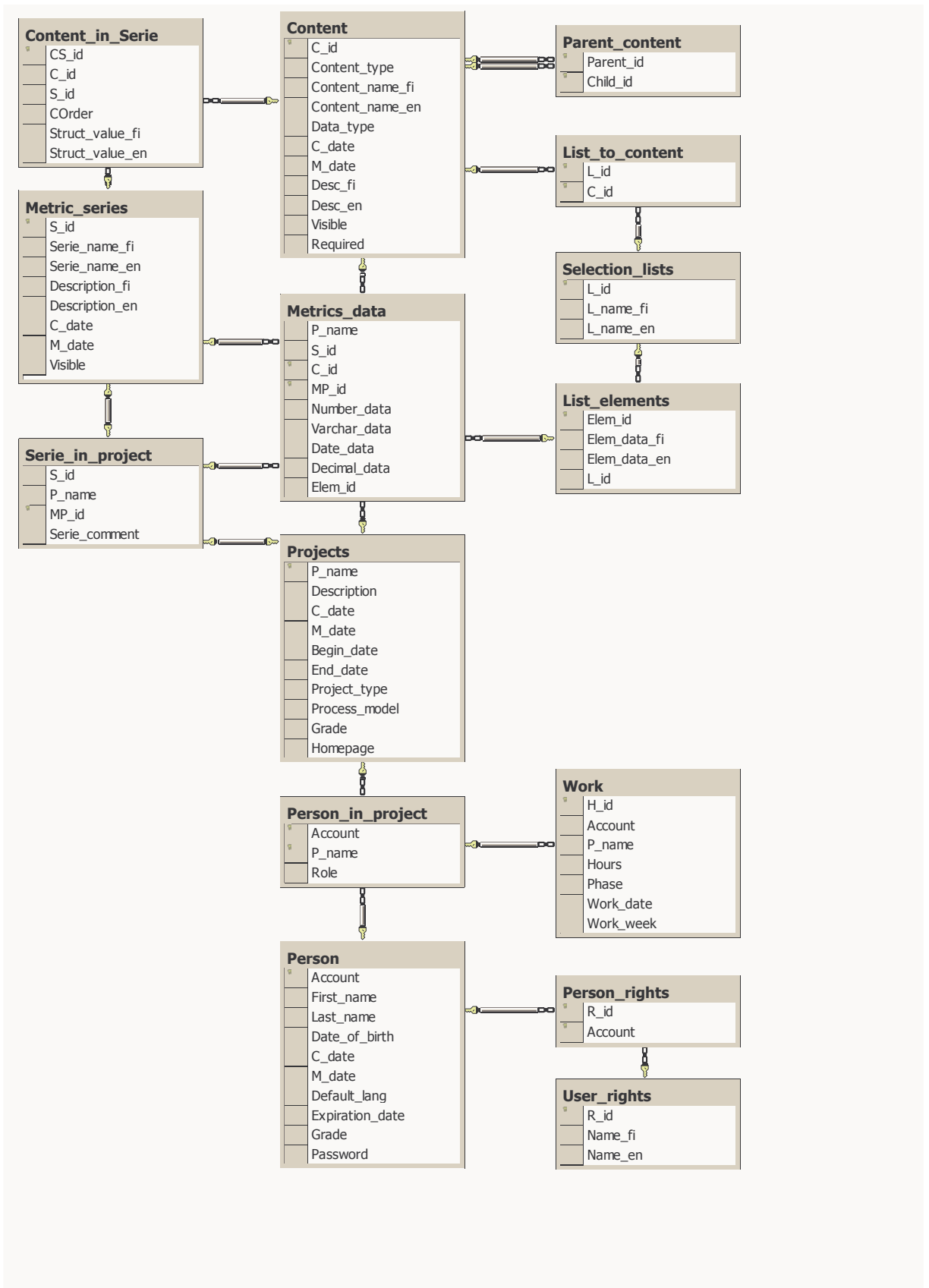
Ohjelmiston tuottamilla sivuilla käytetään kopiota TKTL:n sivuilla käytössä olevasta CSS-tiedostosta. Ohjelmiston tuottama HTML sisältää vain perustageja (H1, P, TABLE, TD, TH,..), jotka määritellään CSS-tiedostossa.

### 9.2 Taitto

Jokainen PHP-sivu kutsuu funktioita UIheader(\$account) ja UIfooter(\$account), jotka tuottavat HTML-sivujen layoutin. Näiden väliin tulostetaan kunkin PHP-sivun tuottama sisältö.

### 9.3 Proto

Käyttöliittymän dokumentti on HTML-proton muodossa osoitteessa: <http://www.cs.helsinki.fi/group/otie2/kali/>



Kuva 11: Järjestelmän tietokantakaavio.