

Ylläpitodokumentti

PLAYOFF

Jari Anttila

Sanna Fröblom

Aarno Sandvik

Tommi Paavilainen

Miikka Kohijoki

Päivi Pääkkö, ohjaaja

Helsinki, 13. joulukuuta 2007

Ohjelmistotuotantoprojekti – yritysviestinnän oppimateriaalin
hallintajärjestelmä

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

sisällys

1. Johdanto.....	1
2. Sanasto.....	1
3. Asennusohje.....	2
3.1 Järjestelmän asennus tiedostojen osalta.....	2
3.2 Järjestelmän tietokannan asennus.....	3
4. Tarkennuksia suunnitelmiin.....	3
4.1 Tietokantakuvaus.....	3
4.1.1 Tietokannan rakenne.....	4
4.1.2 Tietokannan taulut ja luontilauseet.....	4
4.1.2.1 User.....	4
4.1.2.2 Root.....	5
4.1.2.3 User_permissions.....	5
4.1.2.4 NextBranch.....	5
4.1.2.5 Branch.....	5
4.1.2.6 Hyperlinks.....	6
4.1.2.7 Area.....	6
4.1.2.8 Helptext.....	6
4.1.2.9 Helptexts_in_branches.....	7
4.1.2.10 Langpack.....	7
4.1.2.11 Files.....	7
4.1.2.11 FilesInBranch.....	8
4.2 Luokkakuvaus.....	8
4.2.1 addPage.....	8
4.2.2 admin.....	9
4.2.3 admin_functions.....	9
4.2.4 bottom.....	9
4.2.5 browse.....	9
4.2.6 checksAndDbConnection.....	9
4.2.7 deletePage.....	9
4.2.8 download.....	9
4.2.9 editDocumentModel.....	9
4.2.10 editPage.....	9
4.2.11 index.....	9
4.2.12 inteface.....	9
4.2.13 listUsers.....	10
4.2.14 personalArea.....	10
5. Toteutumattomat vaatimukset ja suunnitelman osat.....	10
6. Koodin ylläpitoon liittyvät seikat.....	10

1. Johdanto

Tämä dokumentti on tietojenkäsittelytieteen laitoksen playoff-ryhmän ohjelmistotuotantoprojektin ylläpitodokumentti. Projektin tarkoituksena on luoda materiaalitietopankki Lingua Hankenille, joka korvaa olemassaolevan järjestelmän.

Ylläpitodokumentti on tarkoitettu luettavaksi yhdessä aiempien dokumenttien (vaatimusmäärittely, suunnitteludokumentti) sekä ohjelmakoodin kanssa.

2. Sanasto

PHP: Pääasiallinen ohjelmointikieli, lyhenne sanoista hypertext preprocessor. Php:n prosessointi tapahtuu palvelimella.

JavaScript: Toissijainen ohjelmointikieli. Vaatii toimiakseen javascriptiä tukevan selaimen.

Mysql: Sql-standardia tukeva relaatiotietokanta.

Sivu: Järjestelmässä oleva sisällöntuottajan tekemä aineistokokonaisuus, joka näytetään käyttäjälle web-selaimen kautta.

Dokumenttisivu: Sivun, jolla on sisällöntuottajan tekemä dokumenttimalli sekä siihen liittyvät klikkausalueet ohjeteksteineen. Sisältää myös linkit aineistoon liittyviin dokumenttipohjiin.

Vakiosivu: Perinteisen html-sivun kaltainen sivu, joka voi sisältää tekstiä ja kuvia. Ei sisällä dokumenttimalleja.

Template: Määrittelee sivun rakenteen eli sivulla olevien kenttien paikat.

Kenttä: Kentän avulla määritellään sivun tietyllä paikalla olevan sisällön tyyppi. Vakiosivut voivat otsikon lisäksi sisältää tekstiä ja kuvia, dokumenttisivulla on dokumenttimalli ja mahdolliset linkit dokumenttipohjiin.

Klikkausalue: Sisällöntuottajan dokumenttisivulle dokumenttimallin päälle lisäämä toiminnallinen alue, jota klikkaamalla käyttäjälle näytetään kyseiseen alueeseen liittyvä ohjeteksti.

Dokumenttimalli: Sisällöntuottajan tekemä yrityskirjeen malli, johon liittyy klikkausalueiden avulla selattavia ohjetekstejä.

Oma alue: Loppukäyttäjän palvelintila. Loppukäyttäjillä on oman alueen hallinnointia varten sovellukseen sisältyvä käyttöliittymä.

Oppimateriaalialue: Sisällöntuottajat säilyttävät tuottamaansa ja tarvitsemaansa materiaalia oppimateriaalialueella.

Käyttöliittymäkieli: Kieli, jolla kaikille sivuille yhteiset komponentit näytetään.

Materiaali: Puun ylimmältä tasolta alkava materiaalikokonaisuus.

Julkinen alue: Julkiseksi määritelty osa käyttäjän omasta alueesta, johon haaran sisällöntuottajalla on luku- ja kirjoitusoikeus.

Status: Määrittää onko käyttäjä admin, sisällöntuottaja vai loppukäyttäjä.

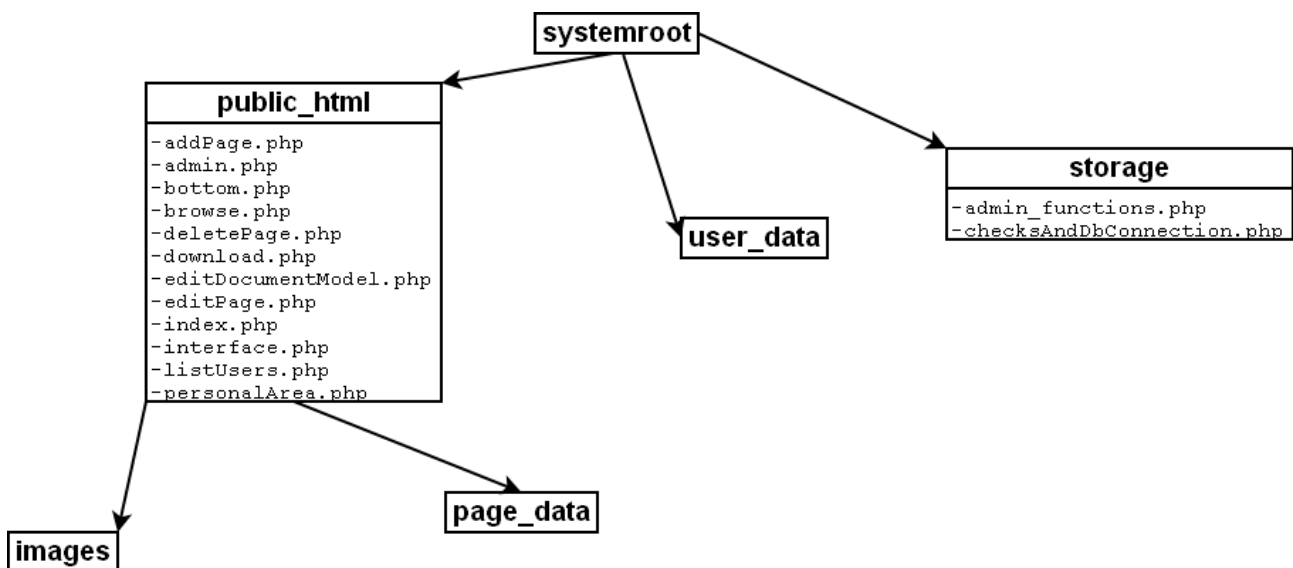
3. Asennusohje

Järjestelmä vaatii toimiakseen palvelimen, jolle on asennettu Mysql, PHP ja Ghostscript. Järjestelmän käyttäjät tarvitsevat vain javascriptiä tukevan internet-selaimen. Olemme testanneet järjestelmän toimivuuden kolmen alla mainitun selaimen kanssa.

Nimi	Versio
PHP	5.0
Mysql	5.0
Ghostscript	8.15.2
Internet Explorer	7.0
Firefox	2.0
Opera	9.0

3.1 Järjestelmän asennus tiedostojen osalta

Ohjelmiston on toimitettu asiakkaalle cd:llä jolta löytyvä playoff.zip-niminen paketti on purettava siihen hakemistoon johon ohjelma halutaan asentaa. Tällä tarkoitetaan kuvassa 1 näkyvää systemroot hakemistoa, paketin purun yhteydessä luodaan muut hakemistot ja siirretään tiedostot niihin. Purun jälkeen on varmistuttava, että tiedostot todella löytyvät oikeista kansioista.



Kuva 1 - Hakemistorakenne

3.2 Järjestelmän tietokannan asennus

Toimitetulta cd:ltä löytyy playoffluonti.txt-niminen tiedosto joka sisältää luontilauseet ohjelman vaatimille tietokantatauluille. Ensimmäiseksi on luotava itse tietokanta. Tietoturvasyistä suositellaan käyttämään itse nimettyä kantaa, käyttäjää ja salasanaa. Nämä muutokset on merkittävä storage-kansiossa olevan checksAndDbConnection.php-tiedostoon funktion db_connect parametreihin. Rivillä 7 kaksi viimeistä suluissa olevaa arvoa määrittävät käyttäjän ja salasanan. Rivillä 11 valitaan käytettävä tietokanta. Näiden on vastattava luotua kantaa.

Cd:ltä löytyy myös eri käyttöliittymäkielel mahdollistavat insert lauseet jotka on ajettava kantaan sen luomisen jälkeen. Nämä löytyvät erillisestä langpack.db-nimisestä tiedostosta.

4. Tarkennuksia suunnitelmiin

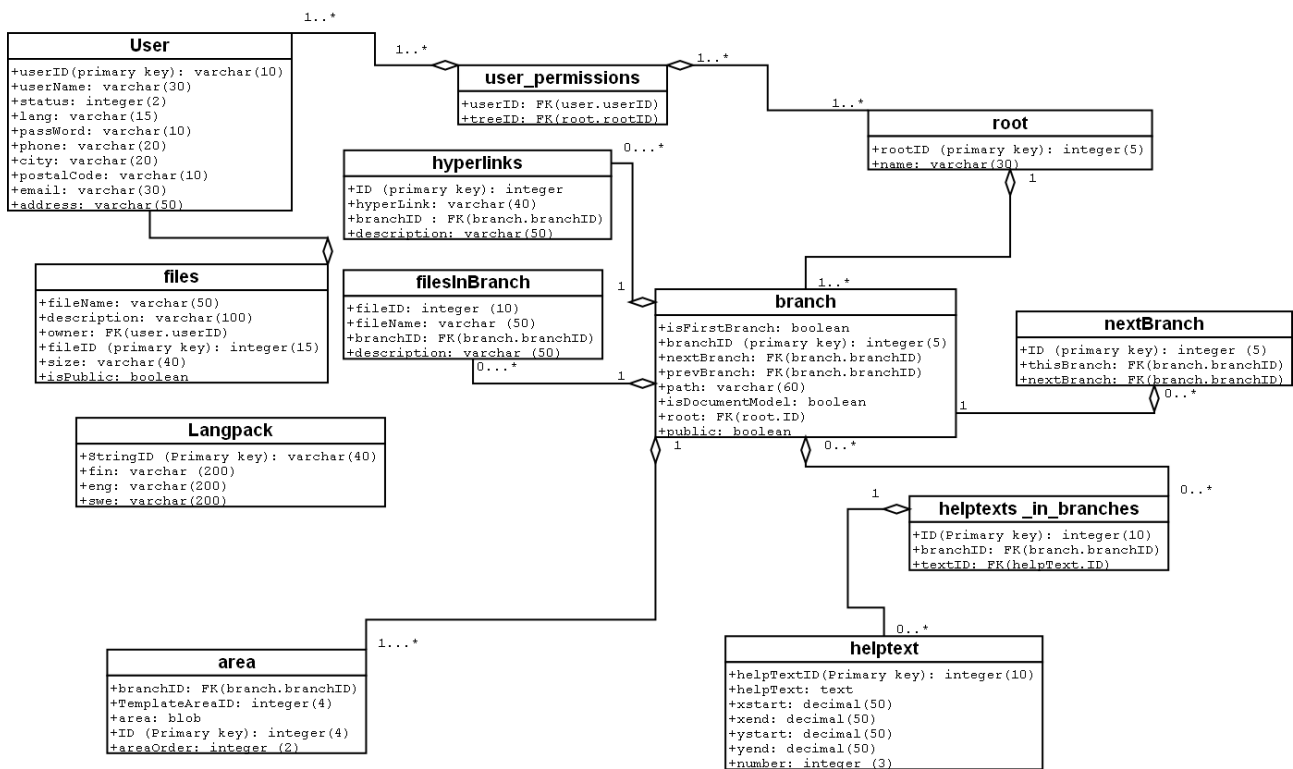
Toteutuksen aikana sekä tietokannan rakenteeseen että luokkakaavioon tuli muutoksia.

4.1 Tietokantakuvaus

Tietokanta koostuu 12 eri taulusta. Näistä yksitoista on yhteydessä toisiinsa ja kahdestoista on erillinen, eri kielitekstit sisältävä taulu.

4.1.1 Tietokannan rakenne

Kuva 2 esittelee tietokannan rakennetta ja eri taulujen riippuvuuksia.



Kuva 2: Tietokannan rakenne

4.1.2 Tietokannan taulut ja luontilauseet

Alla on kuvaus tietokannan jokaisesta taulusta.

4.1.2.1 User

Taulu user pitää sisällään tiedot yksittäisestä käyttäjästä. Taulun avaimena on userID.

```

CREATE TABLE user (
  userID varchar(10) ,
  userName varchar(30),
  status integer(2),
  lang varchar(15) default NULL,
  passWord varchar(10),
  phone varchar(20),
  address varchar(50) default NULL,
  city varchar(20),
  postalCode varchar(10) default NULL,
  email varchar(30) default NULL,
  fileSize varchar(100),
  active boolean,
  PRIMARY KEY (userID)) TYPE=INNODB;
    
```

4.1.2.2 Root

Taulu root pitää sisällään tiedot materiaalikokonaisuuden eri haarojen juurista.

```
CREATE TABLE root (  
rootID integer(5) not null auto_increment,  
name varchar(50),  
description varchar(100),  
PRIMARY KEY (rootID)) TYPE=INNODB;
```

4.1.2.3 User_permissions

Taulu yhdistää käyttäjät eri puihin.

```
CREATE TABLE user_permissions (  
ID integer(5) not null auto_increment,  
userID varchar(10),  
treeID integer(5),  
PRIMARY KEY (ID),  
FOREIGN KEY (userID) REFERENCES user(userID)  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (treeID) REFERENCES root(rootID)  
ON DELETE CASCADE ON UPDATE CASCADE ) TYPE=INNODB;
```

4.1.2.4 NextBranch

Taulu sisältää linkin seuraavaan haaraan.

```
CREATE TABLE nextBranch(  
ID integer(5) not null auto_increment,  
thisBranch integer(5),  
nextBranch integer(5),  
PRIMARY KEY (ID),  
FOREIGN KEY (thisBranch) REFERENCES branch(branchID)  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (nextBranch) REFERENCES branch(branchID)  
ON DELETE CASCADE ON UPDATE CASCADE) TYPE=INNODB;
```

4.1.2.5 Branch

Taulua käytetään puun oksana. Pitää sisällään vakio- ja dokumenttimallisivuja.

```
CREATE TABLE branch(  
branchID integer(5) not null auto_increment,  
isFirstBranch boolean,  
prevBranch integer(5),  
path varchar(60),
```



```
isDocumentModel boolean,  
root integer(5),  
public boolean,  
name varchar(50),  
description varchar(100),  
documentFileName varchar(100),  
PRIMARY KEY (branchID),  
FOREIGN KEY (root) REFERENCES root(rootID)  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (prevBranch) REFERENCES branch(branchID)  
ON DELETE CASCADE ON UPDATE CASCADE) TYPE=INNODB;
```

4.1.2.6 Hyperlinks

Taulussa säilytetään vakiosivuilla pidettäviä hyperlinkkejä.

```
CREATE TABLE hyperlinks(  
ID integer(5) not null auto_increment,  
hyperLink varchar(40),  
description varchar(50),  
branchID integer(5),  
PRIMARY KEY (ID),  
FOREIGN KEY (branchID) REFERENCES branch(branchID)  
ON DELETE CASCADE ON UPDATE CASCADE  
) TYPE=INNODB;
```

4.1.2.7 Area

Taulua käytetään tallentamaan vakiosivujen alueita.

```
CREATE TABLE area (  
ID integer(5) not null auto_increment,  
templateAreaID varchar(25),  
branchID integer(5),  
area longblob,  
areaOrder integer(2),  
PRIMARY KEY (ID),  
FOREIGN KEY (branchID) REFERENCES branch(branchID)  
ON DELETE CASCADE ON UPDATE CASCADE  
) TYPE=INNODB;
```

4.1.2.8 Helptext

Kertoo klikkausalueen paikan ja ohjetekstin sisällön.

```
CREATE TABLE helptext (  
helpTextID integer(10) not null auto_increment,  
number int(3),
```

```
helpText text,  
xstart decimal(50),  
xend decimal (50),  
ystart decimal(50),  
yend decimal(50),  
PRIMARY KEY (helpTextID)) TYPE=INNODB;
```

4.1.2.9 Helptexts_in_branches

Taulua käytetään yhdistämään ohjetekstejä dokumenttimalleihin.

```
CREATE TABLE helptexts_in_branches (  
ID integer(10) not null auto_increment,  
textID integer(10),  
branchID integer(5),  
PRIMARY KEY (ID),  
FOREIGN KEY (branchID) REFERENCES branch(branchID)  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (textID) REFERENCES helptext(helpTextID)  
ON DELETE CASCADE ON UPDATE CASCADE  
) TYPE=INNODB;
```

4.1.2.10 Langpack

Käytetään eri käyttöliittymäkielten säilytykseen.

```
CREATE TABLE langpack (  
stringID varchar(40),  
fin varchar(200),  
eng varchar(200),  
swe varchar(200),  
PRIMARY KEY (StringID)) TYPE=INNODB;
```

4.1.2.11 Files

Käytetään käyttäjien tiedostojen tietojen selvittämiseen.

```
CREATE TABLE files (  
fileName varchar(50),  
owner varchar(10),  
isPublic boolean,  
description varchar(100),  
size varchar(40),  
fileID integer(15) not null auto_increment,  
PRIMARY KEY(fileID),  
FOREIGN KEY (owner) REFERENCES user(userID)  
ON DELETE CASCADE ON UPDATE CASCADE) TYPE=INNODB;
```

4.1.2.11 FilesInBranch

Käytetään eri materiaalikokonaisuuteen tallennettujen tiedostojen yhdistämiseen haaroihinsa.

```
CREATE TABLE filesInBranch (  
fileID integer(10) not null auto_increment,  
fileName varchar(50),  
branchID integer(5),  
description varchar(100),  
PRIMARY KEY(fileID),  
FOREIGN KEY (branchID) REFERENCES branch(branchID)  
ON DELETE CASCADE ON UPDATE CASCADE) TYPE=INNODB;
```

4.2 Luokkakuvaus

Luokkakaavio ja luokkien sisältö kokivat suurimmat muutokset suunnitteludokumentin jäädyttämisen jälkeen. Luokkia on 14 aiemman 16 sijaan. Seuraavien suunnitteludokumentissa mainittujen luokkien suunniteltu toiminnallisuus on yhdistetty muihin luokkiin, eikä niitä ole valmiissa toteutuksessa:

- copyPage (yhdistetty addPage-luokkaan)
- userPublicArea (yhdistetty listUsers-luokkaan)
- search (yhdistetty admin-luokkaan)
- editUserData (yhdistetty admin-luokkaan)
- addUser (yhdistetty admin-luokkaan)
- managerUser (poistettu)

Luokka editDocumentPage tunnetaan nykyään nimellä editDocumentModel. Seuraavat uudet luokat löytyvät toteutuksesta: admin_functions, bottom, checksAndDbConnection sekä download.

Alla on listaus valmiin toteutuksen luokista. Luokkien funktioita ei ole kuvattu tässä, koska ne on kommentoitu kattavasti ohjelmakoodissa.

4.2.1 addPage

Uusi sivu lisätään addPage-luokan kautta. Sivulle valitaan template sekä annetaan nimi ja kuvaus. 'Jatka'-nappia painamalla luodaan uusi sivu annettujen tietojen pohjalta ja ohjataan sisällöntuottaja tälle juuri luodulle sivulle muokkaustilaan (editPage tai editDocumetModel).

4.2.2 admin

Admin-luokka käsittelee sivulle tulleet adminin pyynnöt, ja kutsuu tarvittavia funktioita

admin_functions-luokasta.

4.2.3 admin_functions

Tämä luokka hoitaa kaikki adminin toiminnot, kuten käyttäjien lisäyksen, poiston ja haun, sekä järjestelmän ylläpitotehtävät.

4.2.4 bottom

Tämä luokka sulkee html-tagit, ja se pitää sisällyttää php-tiedostojen loppuun.

4.2.5 browse

Toteuttaa materiaalin selauksen.

4.2.6 checksAndDbConnection

Luokka hoitaa tietokantayhteyden, käyttöliittymäkielen valinnan sekä tarkistaa käyttäjän statuksen.

4.2.7 deletePage

Poistaa sivun sekä kaikki sen alisivut.

4.2.8 download

Hoitaa latauksen järjestelmästä.

4.2.9 editDocumentModel

Dokumenttimallin lisäys ja muokkaus.

4.2.10 editPage

Tämän luokan avulla sisällöntuottajat voivat editoida vakiosivuja. Luokasta löytyvät työkalut kuvien ja linkkien lisäämiseen, sekä tekstin ja sivun tietojen editoimiseen.

4.2.11 index

Aloitussivu jonka kautta järjestelmään kirjaudutaan.

4.2.12 interface

Käyttöliittymä, muuttuu eri käyttäjäryhmien mukaan.

4.2.13 listUsers

Sivun avulla sisällöntuottaja voi listata käyttäjät, joilla on oikeudet tiettyyn

materiaalikonaisuuteen. Listan käyttäjätunnusta klikkaamalla sisällöntuottaja pääsee käyttäjän julkisella alueelle. Käyttäjistä listataan myös heidän nimensä, jos sellainen on tallennettu kantaan. Käyttäjän julkisella alueella sisällöntuottaja voi ladata kyseisen käyttäjän tiedostoja sekä lisätä uusia tiedostoja.

4.2.14 personalArea

Käyttäjän oma alue, jolla voi käsitellä omia tiedostojaan.

5.Toteutumattomat vaatimukset ja suunnitelman osat

Vaatimuksia ei jäänyt toteuttamatta. Muutamiin vaatimukseen tuli tarkennuksia, mutta kaikki vaatimusdokumentissa määritelty toiminnallisuus löytyy järjestelmästä.

Muutoksia:

- Uusia käyttäjiä lisätessä salasana ei ole oletusarvoisesti käyttäjän syntymäaika, ellei admin itse päättä näin.
- Vakiosivun kenttien vaihto keskenään tapahtuu leikkaamalla ja liittämällä kenttien sisältö, mitään erillistä "Vaihda"-nappia ei ole.
- Käyttötapausta ST10 ei toteutettu.

Uusia vaatimuksia:

- Käyttäjätunnuksen on oltava vähintään 3 merkkiä tiedostosta lisättäessä.
- Salasanan on oltava vähintään 2 merkkiä tiedostosta lisättäessä.

6.Koodin ylläpitoon liittyvät seikat

Ohjelmakoodi on kirjoitettu englanniksi, mutta kommentoitu suomeksi. Kaikki funktiot on kommentoitu mahdollisimman selkeästi jatkokehitysmahdollisuuksia silmälläpitäen.