

Ylläpitodokumentti

myva

Helsinki 24.8.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (9 op)

Projektiryhmä

Jaana Diakite

Heikki Hämäläinen

Teemu Hynönen

Lasse Nordgren

Sampsa Somerma

Petri Vuorio

Ryhmän ohjaaja

Jari Suominen

Asiakas

Rasmus Nybergh, Creo Consulting

Johtoryhmä

Kimmo Simola

Kotisivu

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

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
1.0	24.8.2007	Ensimmäinen versio

Sisältö

1 Johdanto	1
2 Sanasto	1
3 Asennusohje	4
3.1 Järjestelmävaatimukset	4
3.2 Tiedostojen asentaminen	4
3.3 URL-uudelleenohjaus	4
3.4 Tietokantaulujen asentaminen	4
3.5 Järjestelmän asetukset	5
3.6 Käyttäjien lisääminen	5
4 Arkkitehtuuri	5
5 Tietomalli	6
6 Tietokanta	17
7 Luokat	31
7.1 Kirjastorakenne	31
7.2 Mallit (Model)	32
7.3 Kontrollerit (Controller)	32
7.4 Näkymät (View)	32
8 Toteutus	32
8.1 Käytetyt ulkoiset komponentit	32
8.1.1 CakePHP	33
8.1.2 Prototype	34
8.1.3 Scriptaculous	34

8.1.4	FCKEditor	34
8.1.5	The DHTML Calendar	34
8.1.6	SimpleTest	35
8.2	Muutokset toteutussuunnitelmaan nähden	35
8.3	Vaatimusten toteutuminen	35
8.3.1	Toiminnalliset vaatimukset	35
8.3.2	Ympäristövaatimukset	42
8.4	Järjestelmässä havaitut puutteet	42
8.5	Jatkokehitys	44
8.5.1	Tietokanta	45
8.5.2	Yksikkötestaus	45

Liitteet

1 Tietokannan taulujen luontilauseet

2 Luokkakaaviot

1 Johdanto

Myyntihenkilöiden valmennuksen hallintajärjestelmä (myva) on Helsingin yliopiston tietojenkäsittelytieteen laitoksen kesän 2007 ohjelmistotuotantoprojektin tuottama järjestelmä. Järjestelmän tilaajana on Creo Consulting. Creo Consulting auttaa yrityksiä myymään paremmin avustamalla uusia yrityksiä myyntistrategian ja tehokkaiden myynnin käytäntöjen luomisessa sekä pidempään toimineita yrityksiä myynnin uudelleensynnyttämisessä. Lisäksi Creo Consulting tarjoaa myyntihenkilöille valmennuksen, jossa opetellaan parempia myynnin käytäntöjä erityisesti uusasiakashankinnassa. Kokemuksen mukaan parhaiten valmennuksen opit menevät perille, mikäli valmennettavat saavat harjoitella oman työnsä ohessa. Myva- järjestelmä toimii valmennusta tukevana ympäristönä siten, että valmennettavat voivat tehdä työtä samanaikaisesti oikeiden asiakkaiden parissa.

Myyntihenkilöiden valmennuksen hallintajärjestelmä sisältää asiakasrekisterin ja rekisterin ylläpidon sekä valmennettavan tietoja. Ohjelmiston tuella valmennettava toteuttaa oppimansa uudet taidot ja toimenpiteet taltioituvat järjestelmään. Valmennettavan työskentelyä ja edistymistä voidaan järjestelmän avulla seurata. Projekti toteutettiin kesän 2007 aikana.

Dokumentti on tarkoitettu selvitykseksi järjestelmän toteutuksesta. Dokumentin toisessa luvussa määritellään dokumentissa käytettävät termit. Kolmannessa luvussa kuvataan järjestelmän arkkitehtuuri, kuten CakePHP-kehysohjelmiston hakemistorakenne. Neljännessä luvussa kuvataan tietomalli, viidennessä luvussa tietomallin riippuvuusmatriisi ja kuudennessä luvussa tietomallista johdettu tietokanta. Seitsemäs luku sisältää järjestelmän luokkien kirjastorakenteen. Luku kahdeksan sisältää toteutuksesta käytetyt ulkoiset komponentit, toteutuneet vaatimukset ja Järjestelmässä havaitut puutteet.

Järjestelmä ei toteuta kaikkia vaatimusmäärittelyn toiminnallisia vaatimuksia. Toteuttamatta jääneet vaatimukset ja keskeneräiset on lueteltu luvussa 8.3. Tietomalli sekä tietokanta on toteutettu niin, että pois jääneet toiminnot voidaan lisätä järjestelmään myöhemmin.

2 Sanasto

Kappaleessa listataan dokumentissa käytetyt termit ja niiden selitteet.

ACID Määritelmä ominaisuuksista, jotka tietokantatransaktioiden tulee täyttää, jotta tietokannan eheys ja luotettavuus voidaan taata. Nämä on kuvattu tarkemmin asiakirjassa ISO/IEC 10026-1:1992 Section 4.

AJAX Asynchronous JavaScript and XML on ohjelmointitekniikka, jonka avulla voidaan luoda interaktiivisia verkkosivuja. Asynkronisuus tarkoittaa sitä, että verkkosivua voidaan päivittää osa kerrallaan vaatimatta koko sivun uudelleenlataamista.

JavaScript viittaa ohjelmointikielen, jolla AJAX-komponentit yleensä toteutetaan. XML on tiedonkuvauskieli, jota käytetään viestiliikenteessä palvelimen ja selaimen välillä.

CakePHP Nopean sovelluskehityksen (RAD) kehysohjelmisto PHP-kielille.

CSS Cascading Styling Sheets on erityisesti verkkosivuille kehitetty tyyliohjeiden kuvaustapa. CSS:n avulla verkkosivun ulkoasu (fontit, värit jne.) ja taitto voidaan erottaa varsinaisesta tekstisisällöstä.

CSV Comma-separated values. Tallennustyyppi taulukkomuotoiselle datalle.

GDD Goal-Derived Design. Helsingin yliopiston tietojenkäsittelytieteen laitoksen sekä Interacta Designin kehittämä käyttöliittymäsuunnitteluprosessi, jossa käyttöliittymä suunnitellaan käyttäjän työtehtävien suorituspolkujen optimaalisuutta ajatellen.

HTML Hypertext Markup Language on verkkosivujen kuvauskieli. HTML-määritelmiä on useita, kuten XHTML, joka on XML-syntaksin mukaista HTML-kieltä.

Inoa Fonectan tarjoama palvelu, jolla saa tarkempaa tietoa yrityksestä, esim. yhteystiedot jne.

JavaScript JavaScript on pääasiassa verkkoympäristössä käytettävä komentosarjakieli. JavaScriptin tärkein sovellus on mahdollisuus lisätä verkkosivuille dynaamista toiminnallisuutta.

Järjestelmä Järjestelmän tilaajan tilaaman tuotteen yleisnimitys.

Järjestelmän tilaaja Creo Consulting.

Kampanja Kokonaisuus, joka sisältää tavoitteet, kampanjaan kuuluvat myyjät ja tavoitteiden kohteena olevat kohdeasiakkaat.

Kampanjan tavoite Tavoite, johon myyjä pyrkii jokaisen kohdeasiakkaan kohdalla, esim. päästä sopimukseen tapaamisesta.

Kampanjan tilaaja Yritys, joka ostaa kampanjan palveluna.

Kampanjan vastuhenkilö Käytetään vaatimusmäärittelydokumentin luvussa 4.2 Tavoitepohjaiset käyttötapaaukset yleiskäsitteenä henkilölle, joka vastaa kampanjoiden hallinnollisista asioista. Muissa luvuissa vastuuhenkilön rooli jakaantuu kampanjan ylläpitäjän ja pääkäyttäjän rooleihin.

Kampanjan ylläpitäjä Myyjä, joka luo ja ylläpitää kampanjoita.

Kohdeasiakas Kampanjan puitteissa yhteydenottojen kohteena oleva yritys.

Käyttäjä Yläkäsite, sisältää sekä kampanjan ylläpitäjän, myyjän ja pääkäyttäjän.

Käyttäjäasiakas Yritys joka käyttää järjestelmää vuokralaisena järjestelmää palvelimiltaan ylläpitävältä taholta.

Luokittelutieto Myyjän kohdeasiakkaalle antama, yhteydenoton perusteella selviävä kategorisointitieto.

Lähiosuma Haun tuloksena palautettava tieto, joka on täsmälleen tai muistuttaa läheisesti hakuparametria.

Majakka Sähköpostiviestiin upotettava viittaus näkymättömään kuvatiedostoon, joka ladataan järjestelmästä viestiä avattaessa. Latauskutsun perusteella järjestelmä saa tiedon, että vastaanottaja on avannut viestin. Monet sähköpostiohjelmat eivät kuitenkaan lataa kuvia ulkoisista lähteistä.

MVC Model-View-Controller. Ohjelmistosuunnittelussa käytetty arkkitehtuurimalli, jossa tietosisältö, sitä esittävä käyttöliittymä sekä edellisten hallinta on erotettu toisistaan.

MySQL Tietokannan hallintajärjestelmä.

Myyjä Työntekijä, joka kampanjassa on yhteydessä kohdeasiakkaisiin.

Myyntiargumentti Väite, jota voi käyttää avuksi keskustelussa kohdeasiakkaan kanssa.

Myyntisuppilo Myyjän työtä helpottamaan tarkoitettu työkalu. Myyntisuppilon suuaukko sisältää tehdyt soittot. Tämän jälkeen se rajautuu niihin, jotka on tavoitettu sekä niihin, joiden kanssa on päästy tavoitteeseen.

PHP Useilla eri alustoilla toimiva ohjelmointikieli, jota käytetään erityisesti Web-palvelinympäristöissä dynaamisten web-sivujen luonnissa.

Profilointitiedot Tietoja myyjien henkilökohtaista ominaisuuksista, esim. luurikammiisuus.

Pääkäyttäjä Ylemmän tason käyttäjä, joka voi luoda myyjiä sekä määrittää näitä kampanjan ylläpitäjiksi. Ei kuitenkaan ole myyjä eikä kampanjan ylläpitäjä.

RAD Rapid Application Development. Ohjelmistojen kehitysmalli, joka tähtää ohjelmiston nopeaan ja helppoon valmistamiseen.

UML Unified Modeling Language. Mm. ohjelmistojen rakenteen kuvaamiseen käytetty graafinen mallinnuskieli.

URL Uniform Resource Locator on merkkijono, jolla osoitetaan www-sivun sijainti verkossa.

Viesti Kohdeasiakkaille lähetävä markkinointisähköpostiviesti.

Y-tunnus Suomen viranomaisten yritykselle myöntämä tunnus, joka yksilöi yrityksen.

Yhteydenotto Myyjä ottaa kohdeasiakkaaseen yhteyttä soittamalla tai lähettämällä viestin.

Yhteyshenkilö Kohdeasiakkaan työntekijä, johon myyjä on suoraan yhteydessä.

XML Extended Markup Language on yleiskäyttöinen kuvauskieli, jossa tietosisällön lisäksi voidaan kuvata tiedon merkitystä. Se on rakenteellinen kuvauskieli, joka auttaa jäsentämään laajoja tietomassoja selkeämmin.

3 Asennusohje

3.1 Järjestelmävaatimukset

Myva-järjestelmän minimivaatimukset ovat seuraavat:

- MySQL versio 5.0.1 tai uudempi
- PHP versio 5.2 tai uudempi.
- Vähintään 16 mt muistia PHP-skriptien käyttöön per ajokerta
- HTTP-palvelinohjelmisto, joka voi suorittaa PHP-skriptejä.

3.2 Tiedostojen asentaminen

Pura myva-1.0.tar.gz -tiedosto hakemistoon, johon haluat järjestelmän asentaa. Muuta tiedostojen oikeudet ympäristön vaatimiksi. Järjestelmä tarvitsee kirjoitusoikeudet hakemistoon /app/tmp. Muuten riittävät normaalit oikeudet.

3.3 URL-uudelleenohjaus

Järjestelmän asennuksessa on otettava huomioon, että URL-uudelleenohjaus toimii ainoastaan siinä tapauksessa, että käytetään CakePHP:n omaa ohjausta. Ohjelmistokehys tukee myös verkkopalvelimen uudelleenkirjoitusmoduuleja, mutta Myva-järjestelmä ei toimi oikein näiden kanssa.

3.4 Tietokantaulujen asentaminen

Pura myva-1.0-database.tar.gz-tiedosto johonkin väliaikaiseen sijaintiin.

Luo tietokanta. Tämän jälkeen on seuraavat sql-tiedostot ajettava kyseiseen tietokantaan:

- myva-create.sql

- myva-initial-data.sql

Esimerkiksi Linux-järjestelmissä tämä tapahtuu komennolla `mysql tietokanta -ukayttajanimi -p < myva-create.sql`. Myva-initial-data.sql-tiedoston ajaminen vaatii sen, että on oltava siinä hakemistossa johon tietokantatiedostot on purettu.

Seuraavaksi on muokattava tiedostoa `app/config/database.php`. Siinä on asetettava oikein tietokantapalvelimen osoite, käyttäjätunnus, salasana ja mahdollisesti portti.

3.5 Järjestelmän asetukset

Ennen käyttöönottoa on muokattava tiedostoa `app/config/myva.php`. Täällä on määritettävä oikein erityisesti ympäristöön viittaavat asetukset.

3.6 Käyttäjien lisääminen

Järjestelmään ei ole toteutettu käyttäjien lisäämistä tai ylläpitoa käyttöliittymän kautta. Ennen kuin järjestelmää voi käyttää, on lisättävä vähintään yksi käyttäjäasiakas. Käyttäjien lisääminen on kuvattu käyttöohjeen kohdassa käyttäjätilien hallinta.

Testataksesi asennusta siirry osoitteeseen, joka osoittaa asennuspolun juureen. Jollei tämä toimi, niin koita siirtyä osoitteeseen `polku/users/login`. Tällöin pitäisi avautua kirjautumisruutu.

4 Arkkitehtuuri

Ohjelmisto on rakennettu käyttäen CakePHP-kehysohjelmistoa versiota 1.1.15.5144. CakePHP on nopean sovelluskehityksen (RAD) kehysohjelmisto, joka noudattaa läheisesti sisällössään Ruby ohjelmointikielen suosittua Ruby on the Rails -kehysohjelmistoa. Kyseessä ei ole suora hiilikopio kyseisestä kehysohjelmistosta, vaan PHP-kielen vahvuudet on pyritty tuomaan mukaan CakePHP-kehysohjelmistoon.

CakePHP:llä tehdyt ohjelmistot rakennetaan MVC-suunnittelumallin mukaisesti. MVC-suunnittelumalli on kuvattu yleisellä tasolla vaatimusmäärittelydokumentaatioissa.

CakePHP olettaa järjestelmän tiedostojen olevan seuraavan hakemistorakenteen alla:

```
app/controllers/  kontrollerit
app/models/       mallit
app/views/        näkymät
```

Muut järjestelmän tiedostot sijoittuvat seuraavasti:

app/config/myva.php	järjestelmän asetukset
app/libs/constants.php	järjestelmän vakiot
app/libs/validators.php	validaattorit
app/tests/	SimpleTestin testiluokat
app/webroot/css	järjestelmän css-tiedostot
app/webroot/js	järjestelmän JavaScript-tiedostot

Kontrollerit (controller) ovat luokkia, jotka tarjoavat järjestelmän http-kutsurajapinnan. Näitä kutsutaan rakenteella:

[polku]/kontrollerin_nimi/metodin_nimi/
parametrit kauttamerkillä (/) erotettuina.

CakePHP olettaa järjestelmässä olevan kontrolleria vastaavan mallin (model). Malli on väylä yhteen tietokantatauluun. Kaikki CakePHP:n mallit on laajennettu AppModel-luokasta, joka tarjoaa useita metodeja tietokannan sisällön käsittelyyn. Mallit sisältävät myös kuvaukset keskinäisistä suhteistaan, esimerkiksi siis käyttäjä-malliin voi liittyä useita asiakas-malleja.

Kontrollerin suorituksen jälkeen näkymät (view) tuottavat html-tulosteen. Näkymillä on julkinen kenttärajapinta, johon kontrollerit syöttävät muuttujien arvot. Kontrollerit hakevat tietosisällön muuttujiin malliluokkien kautta. CakePHP tarjoaa näkymien käyttöön erilaisia avustajaluokkia, joiden avulla voidaan helposti tuottaa esimerkiksi lomakkeiden sisältöjä ja AJAX-komponentteja.

CakePHP tarjoaa kontrollien, mallien ja näkymien käyttöön runsaasti valmiita metodeja. Näiden avulla esimerkiksi tietojen lisääminen ja validoiminen on vaivatonta. Tämän takia esimerkiksi yksinkertaisia lisäämis- ja päivitysoperaatioita varten ei ole tarvetta tehdä erillisiä metodeja malliluokkiin.

Seuraavassa on esimerkinomaisesti kuvattu hakemistorakenne tiedostonimiseen sekä taulurakenne käyttäjäautentikoinnin kirjautumissivuun:

app/controllers/users_controller.php	UserController-luokka
app/models/user.php	User-luokka
app/views/users/index.thtml	UserController-luokan index()-metodin näkymä

Näiden lisäksi CakePHP olettaa olevan olemassa tietokantataulun users.

Kirjautumissivua voisi kutsua osoitteesta <http://www.sovellus.example/user/index> .

5 Tietomalli

Tietomallissa kuvataan järjestelmän tietosisältö. Tietosisältö on tehty tutkimalla vaatimusmäärittelyssä tehtyä käyttöliittymää, eristämällä siitä tietoja, joita järjestelmään voi syöttää, ja tietoja, joita järjestelmä näyttää käyttäjälle. Vaikka vaatimusmäärittelyn toiminnallisten vaatimusten prioriteettiluokan 4 toimintoja ei ole suunniteltu, on näiden asetta-

mat vaatimukset järjestelmälle otettu tietomallissa kuitenkin huomioon. Tietomalli sisältää kaikki toiminnallisten vaatimusten vaatimat luokat. Kuvassa 1 tietomalli on kuvattu UML-luokkakaaviona.

Seuraavana kuvataan eri tietomallin luokat sekä näiden attribuutit. Attribuuttien kohdalla on kuvattu mahdolliset arvoaluerajoitukset.

TM1: Käyttäjäasiakas

Kuvaus: Järjestelmän käyttäjäasiakas.

Nimi String

Kuvaus: Käyttäjän nimi.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen.

TM2: Käyttäjä

Kuvaus: Järjestelmän käyttäjä.

Käyttäjätunnus String

Kuvaus: Käyttäjän käyttäjätunnus, jolla sisäänkirjaututaan järjestelmään.

Sallitut arvot: Vapaamuotoinen teksti. Yksilöllinen.

Pakollisuus: Pakollinen.

Nimi String

Kuvaus: Käyttäjän nimi.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen.

Puhelinnumero String

Kuvaus: Käyttäjän puhelinnumero.

Sallitut arvot: Puhelinnumero.

Pakollisuus: Ei pakollinen.

Sähköpostiosoite String

Kuvaus: Käyttäjän sähköpostiosoite.

Sallitut arvot: Sähköpostiosoite.

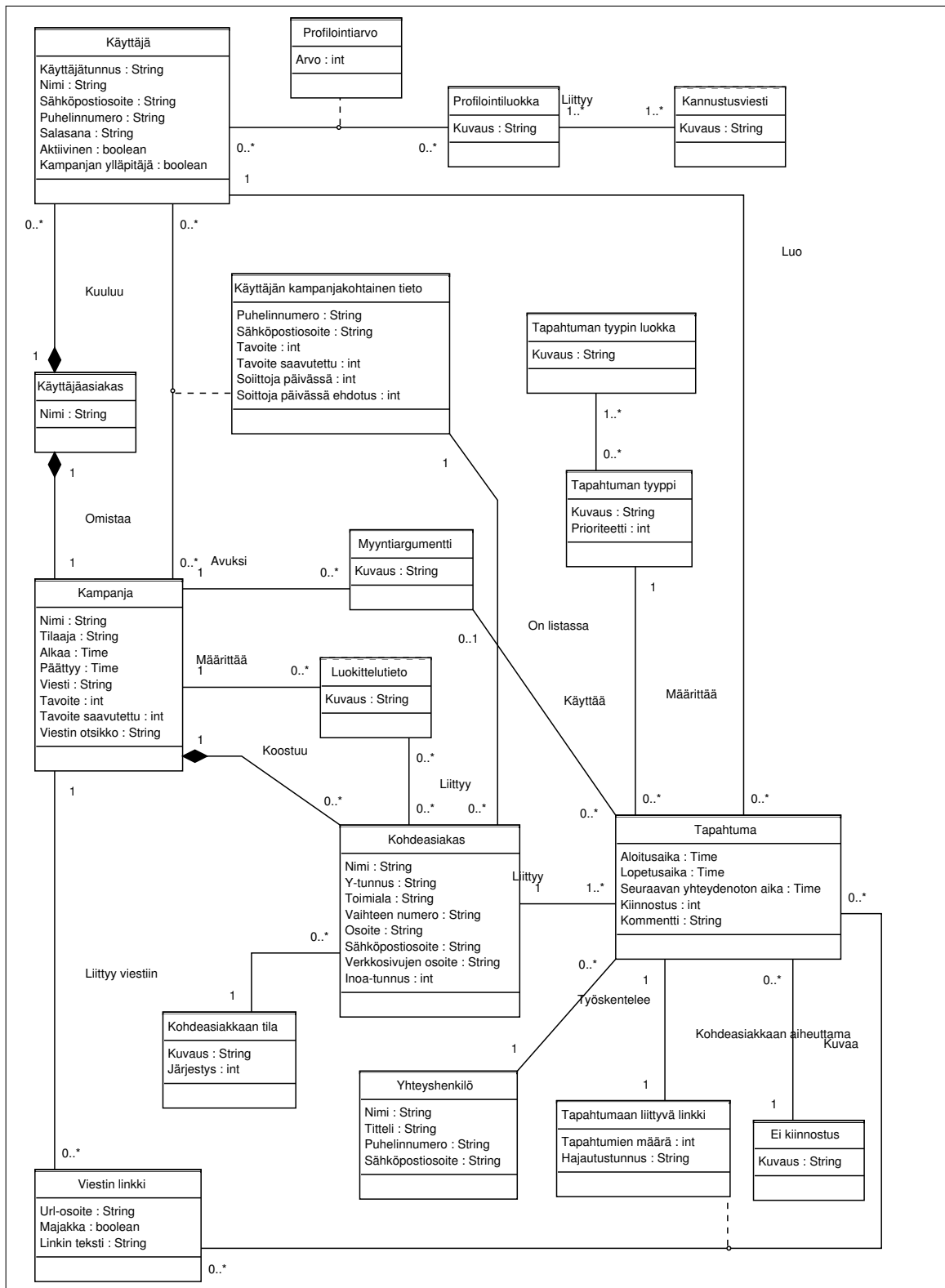
Pakollisuus: Ei pakollinen.

Salasana String

Kuvaus: Käyttäjän salasana.

Sallitut arvot: Salasanasta (tulee täyttää VALID_PASSWORD -muoto vaatimus) muodostettu SHA1-tiiviste heksamerkkijonona.

Pakollisuus: Pakollinen.



Kuva 1: Tietomallikaavio.

Aktiivinen boolean

Kuvaus: Kertoo, onko käyttäjätili aktiivinen, ts. voiko käyttäjä kirjautua järjestelmään.

Sallitut arvot: true=aktiivinen, false=ei aktiivinen

Pakollisuus: Pakollinen.

Kampanjan ylläpitäjä boolean

Kuvaus: Kertoo, onko käyttäjällä oikeus suorittaa kampanjan ylläpitotoimintoja.

Sallitut arvot: true=kampanjan ylläpitäjä, false=ei kampanjan ylläpitäjä

Pakollisuus: Pakollinen.

TM3: Kampanja

Kuvaus: Kampanja.

Nimi String

Kuvaus: Kampanjan nimi.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen.

Tilaaja String

Kuvaus: Kampanjan tilaajan nimi.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen.

Alkaa Time

Kuvaus: Kampanjan alkamispäivä.

Sallitut arvot: Päivämäärä.

Pakollisuus: Pakollinen.

Päätyy Time

Kuvaus: Kampanjan päättymispäivä.

Sallitut arvot: Päivämäärä, >=Alkaa

Pakollisuus: Ei pakollinen.

Viestin otsikko String

Kuvaus: Kampanjaan liittyvä kohdeasiakkaalle lähetettävän viestin otsikko.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen, jos Viesti ei ole tyhjä. Muutoin ei pakollinen.

Viesti String

Kuvaus: Kampanjaan liittyvä kohdeasiakkaalle lähetettävän viestin teksti.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen, jos Viestin otsikko ei ole tyhjä. Muutoin ei pakollinen.

Tavoite int

Kuvaus: Kampanjan soittojen tavoitekappalemäärä päivässä.

Sallitut arvot: ≥ 0 .

Pakollisuus: Ei pakollinen.

Tavoite saavutettu int

Kuvaus: Kampanjan tavoite saavutettu -tilaisten kohdeasiakkaiden tavoitekappalemäärä.

Sallitut arvot: ≥ 0 .

Pakollisuus: Ei pakollinen.

TM4: Viestin linkki

Kuvaus: Kampanjaan liittyvässä viestissä olevien linkkien ym. ulkoisten viittausten tiedot.

URL-osoite String

Kuvaus: Linkin osoittama URL-osoite. Kohdeasiakkaan klikatessa linkkiä selain ohjataan tähän osoitteeseen.

Sallitut arvot: Oikein muotoiltu URL.

Pakollisuus: Pakollinen, jos Majakka=false.

Linkin teksti String

Kuvaus: Linkin käyttäjälle näytettävä teksti.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen, jos Majakka=false.

Majakka String

Kuvaus: Tieto, onko tähän tietueeseen viitattaessa kyse viestin avaamisesta vai viestissä olleen linkin klikkaamisesta. Ensimmäisessä tapauksessa Linkin teksti ja URL-osoite ovat tyhjiä.

Sallitut arvot: true=majakka, false=linkki

Pakollisuus: Pakollinen.

TM5: Myyntiargumentti

Kuvaus: Myynnin edistämisen tarkoitettu myyntiargumentti.

Kuvaus String

Kuvaus: Myyntiargumentti.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen.

TM6: Luokittelutieto

Kuvaus: Kohdeasiakkaan luokittelu eri ominaisuuksiin.

Kuvaus String

Kuvaus: Luokittelutieto.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen.

TM7: Käyttäjän kampanjakohtainen tieto

Kuvaus: Käyttäjän kampanjaan liittyvät tiedot

Puhelinnumero String

Kuvaus: Käyttäjän puhelinnumero.

Sallitut arvot: Puhelinnumero.

Pakollisuus: Ei pakollinen.

Sähköpostiosoite String

Kuvaus: Käyttäjän sähköpostiosoite.

Sallitut arvot: Sähköpostiosoite.

Pakollisuus: Pakollinen.

Tavoite int

Kuvaus: Kampanjan soittojen tavoitekappalemäärä per päivä.

Sallitut arvot: ≥ 0 .

Pakollisuus: Pakollinen.

Tavoite saavutettu int

Kuvaus: Kampanjan tavoite saavutettu -tilaisten kohdeasiakkaiden kappalemäärä.

Sallitut arvot: ≥ 0 .

Pakollisuus: Pakollinen.

Soittoja päivässä int

Kuvaus: Käyttäjän kampanjan soittojen tavoitekappalemäärä per työpäivä.

Sallitut arvot: ≥ 0 .

Pakollisuus: Pakollinen.

Soittoja päivässä ehdotus int

Kuvaus: Kampanjan ylläpitäjän käyttäjälle ehdottama kampanjan soittojen tavoitekappalemäärä per työpäivä.

Sallitut arvot: ≥ 0 .

Pakollisuus: Pakollinen.

TM8: Kohdeasiakas

Kuvaus: Kampanjan kohteena oleva asiakas.

Nimi String

Kuvaus: Kohdeasiakkaan nimi.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen.

Y-tunnus String

Kuvaus: Kohdeasiakkaan yritystunnus.

Sallitut arvot: Y-tunnus.

Pakollisuus: Ei pakollinen.

Toimiala String

Kuvaus: Kohdeasiakkaan toimiala.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Ei pakollinen.

Vaihteen numero String

Kuvaus: Kohdeasiakkaan vaihteen puhelinnumero.

Sallitut arvot: Puhelinnumero.

Pakollisuus: Ei pakollinen.

Osoite String

Kuvaus: Kohdeasiakkaan katu- ja postiosoite (sekä maa).

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Ei pakollinen.

Sähköpostiosoite String

Kuvaus: Käyttäjän sähköpostiosoite.

Sallitut arvot: Sähköpostiosoite.

Pakollisuus: Ei pakollinen.

Verkkosivujen osoite String

Kuvaus: Kohdeasiakkaan verkkosivujen osoite.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Ei pakollinen.

Inoa-tunnus int

Kuvaus: Kohdeasiakkaan Fonectan Inoa-palvelun tunnus.

Sallitut arvot: ≥ 0 .

Pakollisuus: Ei pakollinen.

TM9: Kohdeasiakkaan tila

Kuvaus: Kohdeasiakkaan valmiusasteen kuvaus.

Kuvaus String

Kuvaus: Kohdeasiakkaan tilan kuvaava teksti.

Sallitut arvot: Ei aloitettu, Kesken, Tavoitetta ei saavutettu, Tavoite saavutettu.

Pakollisuus: Pakollinen.

Järjestys int

Kuvaus: Kohdeasiakkaan tilan järjestysnumero, joka kertoo, missä järjestyksessä tilat tulisi listattaessa esittää.

Pakollisuus: Pakollinen.

TM10: Tapahtuma

Kuvaus: Kohdeasiakkaaseen liittyvä tapahtuma. Tapahtuman syntyyn liittyvät syyt määritellään taulussa TM11. Lähtökohtaisesti aina em. syyn toteutuksessa luodaan uusi tapahtuma. Asiakkaan nykytilaa tarkasteltaessa tarkastellaan viimeisintä (mahdollisesti tietyn tyyppistä) tapahtumaa. Aiemmat tapahtumat muodostavat tapahtumahistorian.

Aloitusaika Time

Kuvaus: Tapahtuman aloitusaika.

Sallitut arvot: Aikaleima

Pakollisuus: Pakollinen.

Lopetusaika Time

Kuvaus: Tapahtuman lopetusaika.

Sallitut arvot: Aikaleima, joissakin tapauksissa sama kuin aloitusaika.
 \geq Aloitusaika

Pakollisuus: Pakollinen.

Seuraavan yhteydenoton aika Time

Kuvaus: Seuraavan yhteydenoton aika.

Sallitut arvot: Päivämäärä ja kellonaika.

Pakollisuus: Ei pakollinen.

Kiinnostus int

Kuvaus: Kiinnostuksen asteen kuvaava luku.

Sallitut arvot: 0–100.

Pakollisuus: Ei pakollinen.

Kommentti String

Kuvaus: Tapahtumaan liittyvä kommentti.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Ei pakollinen.

TM11: Tapahtuman tyyppi

Kuvaus: Tapahtuman tyyppi määrittelee tapahtuman luonteen ja saattaa siten määrittellä sen, kuinka tapahtumaa käsitellään tai kuinka tapahtuma tulkitaan järjestelmässä.

Kuvaus String

Kuvaus: Tapahtuman tyyppin kuvaava teksti.

Sallitut arvot: Luotu, Tietojen muutos, Lähetetty, Avattu, Klikattu, Väärä henkilö, Varattu, Ei vastaa, Ei paikalla, Numero ei toimi, Tavoitettu.

Pakollisuus: Pakollinen.

Prioriteetti int

Kuvaus: Tapahtuman prioriteetti. Kohdeasiakkaisiin viimeisimpänä kohdistuneiden tapahtumien prioriteettia voidaan käyttää hyväksi esim. valittaessa myyjälle seuraavaa kohdeasiakasehdotusta, johon myyjän kannattaisi ottaa yhteyttä.

Sallitut arvot: Kokonaisluku. 0=normaali prioriteetti. ≥ 0 on normaalia korkeampi prioriteetti, ≤ 0 on normaalia pienempi prioriteetti. Usealla eri tyyppillä saa olla myös sama prioriteetti.

Pakollisuus: Pakollinen.

TM12: Tapahtuman tyyppin luokka

Kuvaus: Tapahtuman tyytit luokitellaan luokkiin, joiden perusteella toimintoja voidaan kohdistaa tiettyihin tapahtuman tyyppien joukkoihin.

Kuvaus String

Kuvaus: Tapahtuman tyyppin luokan kuvaava teksti.

Sallitut arvot: Soitto, Viesti, Tietomuutos, Valittavissa, Ulkopuolinen.

Pakollisuus: Pakollinen.

TM13: Tapahtumaan liittyvä linkki

Kuvaus: Tapahtumaan liittyvä linkki. Normaalisti käytännöstä poiketen ainoastaan ensimmäinen kohdeasiakkaan puolelta tullut tapahtuma kirjaan, jotta ulkopuoliset eivät voisi täyttää käytettävää tietovarastoa aiheuttamalla mielivaltaisen määrän tapahtumia. Kohdeasiakkaan aiheuttaessa ensimmäisen linkkiin liittyvän tapahtuman luodaan järjestelmään uusi tapahtuma, jolloin tämä tietue yhdistetään ko. tapahtumaan. Sen jälkeen mahdollisesti tulevien tapahtumien toteutuessa ainoastaan päivitetään tapahtumatietueen sisältämä tapahtuman ajankohta sekä tähän tietueeseen tallennettu tapahtumien yhteislukumäärä. Näin ollen tapahtumatietue sisältää viimeisimmän kohdeasiakkaan aiheuttaman tapahtuman ajankohdan.

Hajautustunnus String

Kuvaus: Viestiä lähetettäessä generoidaan hajautustunnus, josta tulee osa viestiin liitettävää URL-osoitetta. Sen avulla tunnistetaan viestinlähetystapahtuma sekä kohde-URL kohdeasiakkaan avatessa viestin tai klikatessa viestin linkkiä.

Sallitut arvot: 160-bittinen SHA1-hajautusarvo koodattuna 40 merkin merkkijonoksi.

Pakollisuus: Pakollinen.

Tapahtumien määrä int

Kuvaus: Linkkiin kohdistuneiden tapahtumien yhteislukumäärä.

Sallitut arvot: ≥ 0 .

Pakollisuus: Pakollinen.

TM14: Ei kiinnostus

Kuvaus: Kiinnostumattomuuden syy.

Kuvaus String

Kuvaus: Kiinnostumattomuuden syyn kuvaava teksti.

Sallitut arvot: Vapaamuotoista tekstiä, esim. Liian kallis, Juuri hankittu, Tekee itse, Ei budejttia.

Pakollisuus: Pakollinen.

TM15: Yhteyshenkilö

Kuvaus: Kohdeasiakkaan tapahtumaan liittyvä yhteyshenkilö.

Nimi String

Kuvaus: Yhteyshenkilön nimi.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Ei pakollinen.

Titteli String

Kuvaus: Yhteyshenkilön titteli.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Ei pakollinen.

Puhelinnumero String

Kuvaus: Yhteyshenkilön puhelinnumero.

Sallitut arvot: Puhelinnumero.

Pakollisuus: Ei pakollinen.

Sähköpostiosoite String

Kuvaus: Yhteyshenkilön sähköpostiosoite.

Sallitut arvot: Sähköpostiosoite.

Pakollisuus: Ei pakollinen.

TM16: Profilointiluokka

Kuvaus: Käyttäytymiseen liittyviä profileja.

Kuvaus String

Kuvaus: Profiloinnin kuvaava teksti.

Sallitut arvot: Vapaamuotoista tekstiä, esim. Prospektointimotivaatio, Prospektoinnin tavoitetaso, Tavoitehämmennys, Murehtija, Himovalmistautuja, Hyperammattilainen, Esiintymispelko, Roolikieltäjä, Päättämätön, Sosiaalinen epävarmuus, Referenssi-inho, Luurikammo, Kaverinkarttaja, Perhepehmo, Vaistovastustaja.

Pakollisuus: Pakollinen.

TM17: Kannustusviesti

Kuvaus: Profilointitietoon liittyvä kannustava viesti.

Kuvaus String

Kuvaus: Kannustusviestin teksti.

Sallitut arvot: Vapaamuotoista tekstiä.

Pakollisuus: Pakollinen.

TM18: Profilointi-arvo

Kuvaus: Profilointitiedot käyttäjistä.

Arvo int

Kuvaus: Profiloinnin prosenttiarvo.

Sallitut arvot: 0–100.
Pakollisuus: Pakollinen.

6 Tietokanta

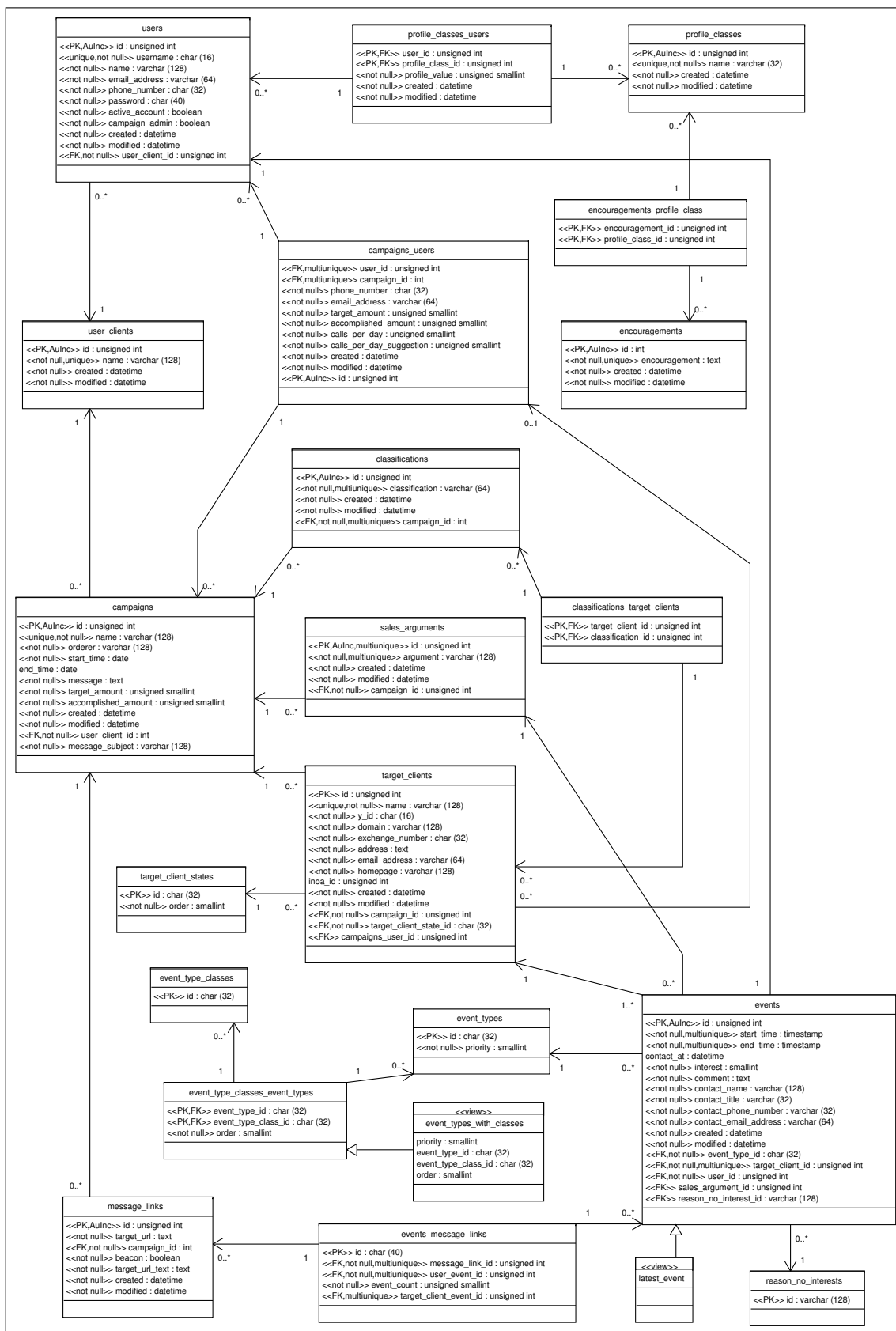
Tässä luvussa kuvataan järjestelmän tietokannan rakenne. Tietokantakaavio on kuvassa 2 ja tietokannan taulujen luontilauseet sekä pohjadatan lisäyslauseet liitteessä 1. Em. luontilauseet on testattu MySQL:n versiolla 5.0.27. Tietokannan suunnittelussa on pyritty muokautumaan järjestelmän toteutuksessa käytettävään CakePHP-kehysohjelmiston oletuksiin, jotta toteutus olisi mahdollisimman suoraviivaista. Tämä on johtanut seuraaviin käytäntöihin:

- Tietokannassa kaikki nimet ovat englanniksi.
- Taulujen nimet ovat monikossa.
- Kaikkien taulujen primääriavaimen nimi on id. Pääsääntöisesti tämä on muusta taulun sisällöstä riippumaton, automaattisesti kasvava kokonaisluku. Tämä siksi, että muun sisällön nimeäminen id:ksi ei monissakaan tapauksissa olisi kuvannut kentän todellista sisältöä. Tähän käytäntöön on tehty muutama poikkeus, jolloin primääriavaimena ei ole sisällöstä riippumaton laskuri. Tällä ratkaisulla on joissakin tilanteissa voitu mahdollistaa tietokantakyselyjen yksinkertaistuminen. Id-kentän sisältöä ei ole alla tietokantakuvauksessa selitetty erikseen sen ollessa edellä kuvatunlainen.
- Moniin tauluihin on lisätty kentät created ja modified, jos on katsottu, että tieto saattaa tulevaisuudessa olla mielenkiintoinen. CakePHP mahdollistaa näiden lähes automaattisen ylläpidon. Kenttä created ilmaisee tietueen luontihetken, modified tietueen viimeisimmän muokkaushetken. Kenttien created ja modified sisältöä ei ole alla tietokantakuvauksessa selitetty erikseen.

Edellisten lisäksi MySQL:n dokumentaatioissa on mainittu, että null-arvoja tulee välttää, jos niitä ei todella tarvita. Tähän on pyritty tietokantasuunnittelussa. Kentissä, joiden arvo ei voi olla null, mutta tietosisältö voi olla tyhjä, ilmaistaan tyhjä arvo merkkijonoissa tyhjänä merkkijonona (pituus nolla) ja kokonaisluvuissa negatiivisena arvona (järjestelmä kirjoittaa -1:n).

Tietokannan kuvauksessa ja kaaviossa käytetyt lyhenteet:

PK	primääriavain
FK	viiteavain
AuInc	Auto Increment



Kuva 2: Tietokantakaavio.

Mikäli taulun tai kentän kuvaus ei poikkea tietomallin vastaavasta, on tietokantakuvaus-
 ssa ainoastaan viite tietomalliin. Vain eroavuudet tietomallista ym. erityspiirteet on ku-
 vattu tässä. Yksittäisen kentän unique-rajoite on mainittu kentän tietotyypissä ja vähintään
 kahden kentän rajoitteet taulukuvauksen alkuosassa.

Sallituista arvoista tietokanta ei tee tarkistuksia, ne on kirjattu tähän osoittamaan, miten
 mallin tulee validoida tietokantaan kirjoitettava tieto ja minkä muotoista tietoa malli voi
 odottaa saavansa tietokannasta. Niille kentille, joille tietotyyppi on riittävä rajaus, eril-
 lisiä sallittuja arvoja ei ole merkitty. Merkkijonojen sallitut arvot on pyritty kuvaamaan
 myva-järjestelmän (ks. liite 3) tai CakePHP:n tiedon validointivakion nimenä, muutoin
 sanallisena kuvauksena. Päivämäärä- ja numeroarvoissa mahdolliset rajoitteet on kuvattu
 lausekkeena. Rajoitteet, jotka on kuvattu yksiselitteisesti tietomallin puolella, ei ole tässä
 erikseen mainittu.

TK1: **user_clients**

Viittaukset: Tietomalli TM1

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

name varchar(128) NOT NULL UNIQUE

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Nimi

created datetime NOT NULL

modified datetime NOT NULL

TK2: **users**

Viittaukset: Tietomalli TM2

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

username char(16) NOT NULL UNIQUE

Sallitut arvot: VALID_USERNAME

Viittaukset: Käyttäjätunnus

name varchar(128) NOT NULL

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Nimi

email_address varchar(64) NOT NULL

Sallitut arvot: VALID_EMAIL tai VALID_EMPTY

Viittaukset: Sähköpostiosoite

phone_number char(32) NOT NULL

Sallitut arvot: VALID_PHONE_NUMBER tai VALID_EMPTY

Viittaukset: Puhelinnumero

password char(40) NOT NULL

Kuvaus: Salasanan 160-bittinen SHA1-tiiviste 40 merkin heksamerkkijonona.

Sallitut arvot: VALID_SHA1

Viittaukset: Salasana

active_account boolean NOT NULL

Viittaukset: Aktiivinen

campaign_admin boolean NOT NULL

Viittaukset: Kampanjan ylläpitäjä

created datetime NOT NULL

modified datetime NOT NULL

user_client_id int unsigned NOT NULL FK

Kuvaus: Käyttäjäasiakas, jolle myyjä kuuluu.

Viittaukset: TK1

TK3: **campaigns**

Viittaukset: Tietomalli TM3

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

name varchar(128) NOT NULL UNIQUE

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Nimi

orderer varchar(128) NOT NULL

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Tilaaaja

start_time date NOT NULL

Viittaukset: Alkaa

end_time date

Kuvaus: Kampanjan päättymisaika. Jos kampanjan päättymisaikaa ei ole määritelty, arvo on null.

Viittaukset: Päättyy

message_subject varchar(128) NOT NULL

Sallitut arvot: VALID_NOT_EMPTY, jos message täyttää ehdon VALID_NOT_EMPTY, muutoin VALID_EMPTY.

Viittaukset: Viesti

message text NOT NULL

Kuvaus: Kampanjaan liittyvä kohdeasiakkaalle lähetettävän viestin teksti HTML-muodossa. Viestissä olevien ulkoisiin kohteisiin kohdistuvien linkkien URL:t ja linkkitekstit tallennetaan myös message_links-tauluun (TK4).

Sallitut arvot: HTML-muotoista tekstiä tai VALID_EMPTY, jos message_subject täyttää ehdon VALID_EMPTY.

Viittaukset: Viesti

target_amount smallint unsigned NOT NULL

Viittaukset: Tavoite

accomplished_amount smallint unsigned NOT NULL

Viittaukset: Tavoite saavutettu

created datetime NOT NULL

modified datetime NOT NULL

user_client_id int unsigned NOT NULL FK

Viittaukset: TK1

TK4: message_links

Kuvaus: Tallennettaessa/päivitetäessä campaigns-taulun (TK3) tietuetta käydään läpi tietueen message-kentän sisältö ja tallennetaan/päivitetään kentästä löytyneet ulkoiset linkit tähän tauluun. Tietueen target_url-kentän arvoa ei saa muuttaa, jos tietueeseen viitataan toisesta taulusta; tarvittaessa luodaan uusi tietue.

Viittaukset: Tietomalli TM4

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

target_url text NOT NULL

Sallitut arvot: VALID_URL, jos beacon=false, muutoin VALID_EMPTY

Viittaukset: URL-osoite

target_url_text text NOT NULL

Sallitut arvot: VALID_NOT_EMPTY, jos beacon=false, muutoin VALID_EMPTY.

Viittaukset: Linkin teksti

beacon boolean NOT NULL

Viittaukset: Majakka

created datetime NOT NULL

modified datetime NOT NULL

campaign_id int unsigned NOT NULL FK

Viittaukset: TK3

TK5: sales_arguments

Viittaukset: Tietomalli TM5

Unique: argument, campaign_id

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

argument varchar(128) NOT NULL

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Kuvaus

created datetime NOT NULL

modified datetime NOT NULL

campaign_id int unsigned NOT NULL FK

Viittaukset: TK3

TK6: classifications

Viittaukset: Tietomalli TM6

Unique: classification, campaign_id

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

classification varchar(64) NOT NULL

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Kuvaus

created datetime NOT NULL

modified datetime NOT NULL

campaign_id int unsigned NOT NULL FK

Viittaukset: TK3

TK7: campaigns_users

Viittaukset: Tietomalli TM7

Unique: user_id, campaign_id

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

user_id int unsigned NOT NULL FK

Kuvaus: Käyttäjä, jonka kampanjasta on kyse.

Viittaukset: TK2

campaign_id int unsigned NOT NULL FK

Kuvaus: Kampanja, johon käyttäjän tiedot liittyvät.

Viittaukset: TK3

phone_number char(32) NOT NULL

Sallitut arvot: VALID_PHONE_NUMBER tai VALID_EMPTY

Viittaukset: Puhelinnumero

email_address varchar(64) NOT NULL

Sallitut arvot: VALID_EMAIL

Viittaukset: Sähköpostiosoite

target_amount smallint unsigned NOT NULL

Viittaukset: Tavoite

accomplished_amount smallint unsigned NOT NULL

Viittaukset: Tavoite saavutettu

calls_per_day smallint unsigned NOT NULL

Viittaukset: Soittoja päivässä

calls_per_day_suggestion smallint unsigned NOT NULL

Viittaukset: Soittoja päivässä ehdotus

created datetime NOT NULL

modified datetime NOT NULL

TK8: target_clients

Viittaukset: Tietomalli TM8

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

name varchar(128) NOT NULL

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Nimi

y_id char(16) NOT NULL

Sallitut arvot: VALID_YID tai VALID_EMPTY

Viittaukset: Y-tunnus

domain varchar(128) NOT NULL

Viittaukset: Toimiala

exchange_number char(32) NOT NULL

Sallitut arvot: VALID_PHONE_NUMBER tai VALID_EMPTY

Viittaukset: Vaihteen numero

address text NOT NULL

Viittaukset: Osoite

email_address varchar(64) NOT NULL

Sallitut arvot: VALID_EMAIL tai VALID_EMPTY

Viittaukset: Sähköpostiosoite

homepage varchar(128) NOT NULL

Sallitut arvot: VALID_URL tai VALID_EMPTY

Viittaukset: Verkkosivujen osoite

inoa_id int unsigned

Kuvaus: Inoa-tunnus. Voi olla null, koska puuttuvaa arvoa ei voida ilmaista negatiivisella luvulla.

Viittaukset: Inoa-tunnus

created datetime NOT NULL

modified datetime NOT NULL

target_client_state_id char(32) NOT NULL FK

Kuvaus: Kohdeasiakkaan tila.

Viittaukset: TK10

campaign_id int unsigned NOT NULL FK

Kuvaus: Kampanja, johon kohdeasiakas kuuluu.

Viittaukset: TK3, TK7

campaigns_user_id int unsigned FK

Kuvaus: Myyjän kampanja, johon kohdeasiakas kuuluu. Kohdeasiakas voi myös olla kuulumatta kenellekään myyjälle.

Sallitut arvot: null=kohdeasiakas ei kuulu kenellekään myyjälle, FK=kohdeasiakas kuuluu viitattuun myyjän kampanjaan.

Viittaukset: TK7

TK9: classifications_target_clients

Kuvaus: Monen suhde moneen -välitaulu, joka mahdollistaa usean luokittelutiedon liittämisen kohdeasiakkaisiin.

Viittaukset: Tietomallit TM6, TM8

target_client_id int unsigned NOT NULL UNIQUE PK FK

Viittaukset: TK8

classification_id int unsigned NOT NULL UNIQUE PK FK

Viittaukset: TK6

TK10: target_client_states

Viittaukset: Tietomalli TM9

id char(32) NOT NULL UNIQUE PK

Kuvaus: Kohdeasiakkaan tilan kuvaava teksti. Kenttä on taulun pääavain ja siksi nimetty id:ksi.

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Kuvaus

order smallint NOT NULL UNIQUE

Viittaukset: Järjestys

TK11: events

Kuvaus: Tauluun on tehokkuussyistä yhdistetty tietomallista Tapahtuma ja Yhteyshenkilö. Tietomallissa ne on pidetty erillään selkeyden vuoksi.

Viittaukset: Tietomallit TM10, TM15

Unique: target_client_id, start_time

Unique: target_client_id, end_time

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

start_time timestamp NOT NULL

Viittaukset: Aloitusaika

end_time timestamp NOT NULL

Viittaukset: Lopetusaika

contact_at datetime

Kuvaus: Seuraavan yhteydenoton aika. Jos seuraavan yhteydenoton aikaa ei ole määritelty, kentän arvo on null.

Viittaukset: Seuraavan yhteydenoton aika

interest smallint NOT NULL

Sallitut arvot: -1–100, missä -1 ilmaisee, ettei kiinnostusta ole määritelty.

Arvot 0–100 kuvaavat kiinnostuksen määrää.

Viittaukset: Kiinnostus

comment text NOT NULL

Viittaukset: Kommentti

contact_name varchar(128) NOT NULL

Viittaukset: Nimi

contact_title varchar(32) NOT NULL

Viittaukset: Titteli

contact_phone_number varchar(32) NOT NULL

Sallitut arvot: VALID_PHONE_NUMBER tai VALID_EMPTY

Viittaukset: Puhelinnumero

contact_email_address varchar(64) NOT NULL

Sallitut arvot: VALID_EMAIL tai VALID_EMPTY

Viittaukset: Sähköpostiosoite

created datetime NOT NULL

modified datetime NOT NULL

event_type_id char(32) NOT NULL FK

Kuvaus: Tapahtuman tyyppi.

Viittaukset: TK13

target_client_id int unsigned NOT NULL FK

Kuvaus: Kohdeasiakas, johon tapahtuma kohdistuu.

Viittaukset: TK8

user_id int unsigned NOT NULL FK

Kuvaus: Tapahtuman aiheuttanut käyttäjä.

Viittaukset: TK2

sales_argument_id int unsigned FK

Kuvaus: Tapahtuman ollessa yhteydenotto, sen yhteydessä mahdollisesti käytetty myyntiargumentti. Jos tapahtumaan ei liity myyntiargumenttia, arvo on null.

Viittaukset: TK5

reason_no_interest_id varchar(128) FK

Kuvaus: Tapahtuman ollessa yhteydenotto, kohdeasiakkaan mahdollisesti mainitsema syy, miksi ei ole kiinnostunut. Jos tapahtumaan ei liity syytä, arvo on null.

Viittaukset: TK17

TK12: latest_events

Kuvaus: Näkymä events-tilusta, joka sisältää vain kunkin kohdeasiakkaan viimeisimmän tapahtuman, poislukien ne tapahtumat, joiden tapahtumatyyppin luokka on Ulkopuolinen.

Stereotyyppit: view

Viittaukset: TK11

TK13: event_types

Viittaukset: Tietomalli TM11

id char(32) NOT NULL UNIQUE

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Kuvaus

priority smallint NOT NULL

Viittaukset: Prioriteetti

TK14: event_type_classes*Viittaukset:* Tietomalli TM12**id** char(32) NOT NULL UNIQUE PK*Sallitut arvot:* VALID_NOT_EMPTY*Viittaukset:* Kuvaus**TK15: event_type_classes_event_types***Kuvaus:* Monen suhde moneen -välitaulu, joka mahdollistaa usean tapahtuman tyyppin luokan liittämisen tapahtuman tyyppiin.*Viittaukset:* Tietomallit TM11, TM8**event_type_id** char(32) NOT NULL UNIQUE PK FK*Viittaukset:* TK8**event_type_class_id** char(32) NOT NULL UNIQUE PK FK*Viittaukset:* TM12**order** smallint NOT NULL*Kuvaus:* Tapahtuman tyyppin järjestysnumero, joka kertoo, missä järjestyksessä tapahtuman tyypit tulisi listattaessa esittää. Koska järjestys on tässä välitaulussa, järjestys on mahdollista määrittellä erikseen tapahtuman tyyppin luokkakohtaisesti.**TK16: events_message_links***Viittaukset:* Tietomalli TM13*Unique:* message_link_id, user_event_id*Unique:* message_link_id, target_client_event_id**id** char(40) NOT NULL UNIQUE PK*Kuvaus:* Hajautusarvo lasketaan viestin lähetyshetkellä merkkijonosta, joka muodostetaan katenoimalla user_event_id, TK11:start_time ja message_link_id #-merkillä toisistaan erottaen.*Sallitut arvot:* VALID_SHA1*Viittaukset:* Hajautustunnus**event_count** smallint unsigned NOT NULL*Viittaukset:* Tapahtumien määrä**message_link_id** int unsigned NOT NULL FK*Kuvaus:* Tapahtumaan liittyvä linkki.

Viittaukset: TK4

user_event_id int unsigned NOT NULL FK

Kuvaus: Käyttäjän aiheuttama tapahtuma, johon linkki liittyy. Käytännössä viestin lähetystapahtuma.

Viittaukset: TK11

target_client_event_id int unsigned FK

Kuvaus: Kohdeasiakkaan aiheuttama tapahtuma, johon linkki liittyy. Käytännössä viestin avaaminen tai linkin klikkaaminen.

Sallitut arvot: Arvo on null, kunnes syntyy kohdeasiakkaan aiheuttama tapahtuma, jolloin kenttä päivitetään viittaamaan ko. tapahtumaan.

Viittaukset: TK11

TK17: reason_no_interests

Viittaukset: Tietomalli TM14

id varchar(128) NOT NULL UNIQUE

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Kuvaus

TK18: profile_classes

Viittaukset: Tietomalli TM16

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

name varchar(32) NOT NULL UNIQUE

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Kuvaus

TK19: encouragements

Viittaukset: Tietomalli TM17

id int unsigned AUTO_INCREMENT NOT NULL UNIQUE PK

encouragement text NOT NULL

Kuvaus: Kentän tulisi olla UNIQUE, mutta MySQL ei salli kyseistä rajoitetta text-tietotyypille.

Sallitut arvot: VALID_NOT_EMPTY

Viittaukset: Kuvaus

created datetime NOT NULL

modified datetime NOT NULL

TK20: encouragements_profile_classes

Kuvaus: Monen suhde moneen -välitaulu, joka liittää profiointiluokkiin sopivat kannustusviestit toisiinsa.

Viittaukset: Tietomallit TM16, TM18

encouragement_id int unsigned NOT NULL PK FK

Kuvaus: Profiointiluokkaan sopiva kannustusviesti.

Viittaukset: TK19

profile_class_id int unsigned NOT NULL PK FK

Kuvaus: Profiointiluokka, johon kannustusviesti sopii.

Viittaukset: TK18

TK21: profile_classes_users

Kuvaus: Monen suhde moneen -välitaulu, joka liittää käyttäjien profiointiarvot profiointiluokkiin.

Viittaukset: Tietomalli TM18

user_id int unsigned NOT NULL PK FK

Kuvaus: Käyttäjä, jonka profiointiarvon tietue sisältää.

Viittaukset: TK18

profile_class_id int unsigned NOT NULL PK FK

Kuvaus: Profiililuokka, jonka profiointiarvon tietue sisältää.

Viittaukset: TK18

profile_value smallint unsigned NOT NULL

Pakolliset arvot: 0–100

Viittaukset: Arvo

created datetime NOT NULL

modified datetime NOT NULL

TK22: event_types_with_classes

Kuvaus: Näkymä, joka yhdistää tapahtuman tyyppin ja sen luokan.

Stereotyypit: view

Viittaukset: event_types, event_type_classes_event_types, event_type_classes

priority smallint NOT NULL

event_type_id char(32) NOT NULL UNIQUE PK FK

event_type_class_id char(32) NOT NULL UNIQUE PK FK

order smallint NOT NULL

7 Luokat

Toteutuksen rajapintakuvaukset ovat erillisinä HTML-dokumentteina. Kontrollerit ja mallit ovat phpdoc-dokumentaationa, näkymät on dokumentoitu ainoastaan lähdekoodissa. JavaScript:n dokumentaatio on jsdoc-dokumentaationa, joskin osa dokumentaatiosta on luettavissa ainoastaan lähdekoodista (sisällytetty jsdoc:iin). Toteutuksen luokkakaaviokuvat ovat liitteessä 2.

7.1 Kirjastorakenne

Alla on kuvattuna järjestelmän toteutuksen kirjastorakenne.

- app
 - config
 - controllers sisältää kontrollerit
 - libs sisältää vakiot (constants) ja validaattorit(validators)
 - models sisältää mallit
 - plugins
 - tests sisältää testiluokat malleille
 - tmp
 - webroot sisältää tyylitiedostot (css), kuvat (img), javascriptit (js)
 - vendors
 - views sisältää näkymät sekä ? (helpers), ? (layouts)
- cake
- docs
- testi
- vendors

7.2 Mallit (Model)

Mallit (Model) ovat luokkia, jotka tarjoavat palvelut tietokannan käyttöön. Seuraavassa on kuvattuna ne mallit, joilla on metodeja CakePHP:n tarjoamien valmiiden metodien lisäksi. Kaikilla tietokannan tauluilla on joka tapauksessa järjestelmässä niitä vastaava malliluokka. Mallien keskinäiset riippuvuudet johtuvat siitä, että CakePHP:ssa kuvataan jokaisessa mallissa sen riippuvuus muista. Tämän avulla CakePHP:n on mahdollista palauttaa kehysohjelmiston valmiilla metodeilla tietueita, jotka sisältävät tiedot riippuvaisista tauluista. Malliluokkien, joilla ei ole omia metodeja, keskinäiset riippuvuudet selviävät tietokantakaaviosta. Mallit eivät kutsu toistensa metodeita.

7.3 Kontrollerit (Controller)

Kontrollerit ovat järjestelmän komponentteja, jotka ottavat vastaan järjestelmään tulevat http-kutsut. CakePHP:n kutsut ovat muotoa:

```
[polku_ohjelmaan]/kontrollerin_nimi/metodin_nimi/mahdollinen_parametri
```

jne. Seuraavassa alaluvuissa kuvataan kontrolleri-luokat sekä niiden metodit. Stereotyyppi `post` osoittaa ne metodit, jotka saavat parametrinsa CakePHP:n `post`-muuttujien käsittelyn kautta. `Get`-muuttujien parametrit tulevat metodin käyttöön suoraan metodin kutsun kautta paikallisiksi muuttujiksi.

7.4 Näkymät (View)

Näkymät ovat järjestelmän komponentteja, jotka tuottavat html-sivut. Jokaista kontrollerin metodia vastaa näkymätiedosto. Esimerkiksi `UserController`in metodia `index()` vastaa näkymä `user/index.html`. Näkymillä on julkinen kenttärajapinta. Näkymää vastaava kontrolleri tuottaa arvot muuttujiin. Muuttujien tietotyypit sekä `array`-muuttujien tapauksessa indeksivaimet on kuvattu kunkin luokan kohdalla.

8 Toteutus

8.1 Käytetyt ulkoiset komponentit

Seuraavissa aliluvuissa kuvataan järjestelmässä käytetyt ulkoiset komponentit sekä niihin mahdollisesti tehdyt muutokset. Muutokset tulee huomioida, mikäli ulkoisia komponentteja myöhemmin päivitetään.

8.1.1 CakePHP

CakePHP on kuvattu luvussa 4.

Tiedostoon cake/libs/router.php funktio stripEscape() (alkaen riviltä 205) kirjoitettiin rekursiiviseksi, sillä alkuperäinen toteutus tuotti virheilmoituksia sellaisella html-lomakkeella lähetetyllä datalla, jossa name-attribuutissa oli yli kolme kertaa []-merkit. Funktion vanha ja uusi toteutus:

```

/*
 * Does not work - must be recursive!
 */
function stripEscape($param) {
    if(is_string($param) || empty($param)) {
        $return = preg_replace('/^ *-!/', '', $param);
        return $return;
    }
    foreach($param as $key => $value) {
        if(is_string($value)) {
            $return[$key] = preg_replace('/^ *-!/', '', $value);
        } else {
            foreach ($value as $array => $string) {
                $return[$key][$array] = preg_replace('/^ *-!/', '',
                    $string);
            }
        }
    }
    return $return;
}
*/

/*
 * Modified, recursive version:
 */
function stripEscape($param) {
    if(is_string($param) || empty($param)) {
        $return = preg_replace('/^ *-!/', '', $param);
    } else {
        foreach($param as $key => $value) {
            $return[$key] = $this->stripEscape($value);
        }
    }
    return $return;
}

```

8.1.2 Prototype

Prototype (prototypejs.org) on JavaScript-kirjasto. Järjestelmässä käytetään Scriptaculous:n mukana tullutta versiota 1.5.0. Kirjasto sijaitsee hakemistossa `app/webroot/js/lib`.

8.1.3 Scriptaculous

Scriptaculous (script.aculo.us) on JavaScript-kirjasto. Järjestelmässä käytetään versiota 1.7.0. Kirjasto sijaitsee hakemistossa `app/webroot/js/src`.

8.1.4 FCKEditor

FCKEditor (www.fckeditor.net) on JavaScript:llä toteutettu HTML-editori, jota käytetään kampanjan hallintasivulla. Järjestelmässä käytetään editorin versiota 2.4.3. Editori sijaitsee hakemistossa `app/webroot/js/fckeditor`.

Editorin hakemistossa sijaitseva tiedosto `myconfig.js` on järjestelmää varten tehty asetus-tiedosto, eikä osa editorin jakelua.

8.1.5 The DHTML Calendar

The DHTML Calendar (www.dynarch.com/projects/calendar) on JavaScript:llä toteutettu päivämäärän ja kellonajan valintaan tarkoitettu kalenteri, jota käytetään järjestelmässä päivämäärä- ja kellonaikakenttien yhteydessä. Järjestelmässä käytetään kalenterin versiota 1.0. Kalenteri sijaitsee hakemistossa `app/webroot/js/jscalendar`.

Tiedostossa `calendar.js` on korjattu ongelma, joka aiheutti joidenkin elementtien häviämisen kalenterin sulkemisen yhteydessä Internet Explorer 7 -selaimella. Korjaus sijaitsee riveillä 1463-1464:

```
// Alkuperäinen:  
// if (!Calendar.is_ie && !Calendar.is_opera)  
// Korjattu:  
if ((!Calendar.is_ie || (  
    Calendar.is_ie && /msie 7/i.test(navigator.userAgent)  
)) && !Calendar.is_opera)
```

8.1.6 SimpleTest

SimpleTest (<http://simpletest.org>) on PHP-ohjelmien yksikkötestaukseen tarkoitettu ohjelmistokehys. Järjestelmän yksikkötestaus on tehty SimpleTestin tarjoamien toimintojen kautta. Käytössä on versio 1.01.

8.2 Muutokset toteutussuunnitelmaan nähden

Seuraavia muutoksia on tapahtunut toteutussuunnitelman valmistumisen jälkeen:

- Tietokannan välitaulujen nimet on muutettu CakePHP:n vaatimusten mukaisesti siten, että välitaulun yhdistämien taulujen nimet ovat aakkosjärjestyksessä.
- Edellisen seurauksena myös vastaavat mallit, kontrollerit ja näkymät on nimetty uudelleen.
- Tietokantaan tullut kaksi näkymää sekä joitakin uusia tauluja ja kenttiä.
- Näkymien rajapintoja on muutettu osittain siten, että ne ottavat tietoa vastaan `$params['data']` -muuttujan välityksellä erillisen muuttujan sijaan, mikä helpottaa html-lomakkeelta tulleen tiedon kierrättämistä uudelleen saman sivun lomakekenttiin.
- Järjestelmässä on käytetty ajax-kutsuja suunniteltua enemmän.

8.3 Vaatimusten toteutuminen

8.3.1 Toiminnalliset vaatimukset

Seuraavassa taulukossa kuvataan vaatimusmäärittelyssä määritettyjen toiminnallisten vaatimusten toteutuminen. Taulukon jälkeen on listattu vaatimukset sekä tarkempi selvitys mahdollisesta puutteellisuudesta.

Taulukko 1: Vaatimusten toteutuminen.

	Toteutettu	Toteutus puutteellinen	Ei-toteutettu
TV1	X		
TV2	X		
TV3		X	
TV4	X		
TV5	X		
TV6	X		
TV7	X		

Taulukko 1: Vaatimusten toteutuminen.

	Toteutettu	Toteutus puutteellinen	Ei-toteutettu
TV8	X		
TV9	X		
TV10	X		
TV11	X		
TV12	X		
TV13	X		
TV14	X		
TV15	X		
TV16	X		
TV17	X		
TV18	X		
TV19	X		
TV20	X		
TV21		X	
TV22	X		
TV23	X		
TV24	X		
TV25	X		
TV26			X
TV27		X	
TV28			X
TV29		X	
TV30		X	
TV31		X	
TV32	X		
TV33			X
TV34		X	
TV35		X	
TV36			X
TV37	X		
TV38			X
TV39			X
TV40			X
TV41			X
TV42			X
TV43			X
TV44			X
TV45			X

TV1: Kirjautuminen

Toteutettu: Toteutettu vaatimusten mukaisesti.

TV2: Uloskirjautuminen

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV3: Kampanjan tavoitteiden muuttaminen

Järjestelmän toiminta: Kampanjan uudet tavoitteet tallentuvat tietokantaan.

Toteutettu: Toteutettu vaatimusten mukaisesti. Virhekäsittelyä ei toteutettu niin, että ei-numeeriset arvot käsiteltäisiin virheinä. Muuttuvat ohjelmistossa arvoksi 0.

Prioriteetti: 1

TV4: Käyttäjä siirtyy ylläpitosivulle

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV5: Kampanjan luominen

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV6: Kampanjan tietojen täyttäminen

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV7: Kampanjan tietojen muuttaminen

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV8: Myyjän tietojen muuttaminen

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV9: Myyjien liittäminen kampanjaan

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV10: Myyjien poistaminen kampanjasta

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV11: Kohdeasiakkaiden liittäminen myyjälle

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV12: Salasanan vaihto

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV13: Kampanjan valinta listalta

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV14: Soitto listalla olevalle kohdeasiakkaalle

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV15: Kohdeasiakkaan tietojen muuttaminen

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 1

TV16: Lisää yhteydenoton tietoja

Toteutettu: Toteutettu vaatimusten mukaisesti lukuun ottamatta sitä, että kiinnostusta ei aseteta liukusäätimellä.

Prioriteetti: 1

TV17: Yksittäisen kohdeasiakkaan lisääminen

Toteutettu: Toteutettu vaatimusten mukaisesti

Prioriteetti: 1

TV18: Viestin lähetys valituille kohdeasiakkaille

Toteutettu: Toteutettu vaatimusten mukaisesti.
Prioriteetti: 1

TV19: Viestin lähetys ilman valintaa

Toteutettu: Toteutettu vaatimusten mukaan.
Prioriteetti: 1

TV20: Myyjän tavoitteiden muuttaminen

Toteutettu: Toteutettu vaatimusten mukaan.
Prioriteetti: 2

TV21: Kohdeasiakkaiden lisääminen tiedostosta

Toteutettu: Toteutettu puutteellisesti. Lisääminen mahdollista ainoastaan kampanjaan kuuluvan myyjän kautta.
Prioriteetti: 2

TV22: Pikasoitto

Toteutettu: Toteutettu vaatimusten mukaan.
Prioriteetti: 2

TV23: Myyjän raporttien katselu

Toteutettu: Toteutettu vaatimusten mukaan.
Prioriteetti: 2

TV24: Kaikkien raporttien tarkastelu

Toteutettu: Toteutettu vaatimusten mukaan.
Prioriteetti: 2

TV25: Myyjän raporttien tarkastelu

Toteutettu: Toteutettu vaatimusten mukaan. Pieniä puutteita tiedoissa.
Prioriteetti: 2

TV26: Raporttien aikarajaus

Toteutettu: Toteutus ei toimi.

Prioriteetti: 2

TV27: Soittohistorian katselu

Toteutettu: Toteutettu puutteellisesti sekä virheellisiä tietoja.

Prioriteetti: 2

TV28: Myyntisuppilon käyttäminen

Toteutettu: Ei toteutettu

Prioriteetti: 2

TV29: Miten menee näkymä

Toteutettu: Toteutettu puutteellisesti. Työaika puuttuu näkymästä.

Prioriteetti: 2

TV30: Viestin esikatselu

Toteutettu: Toteutettu puutteellisesti. Viestissä ei ole allekirjoitusta.

Prioriteetti: 2

TV31: Päivätavoitteen asettaminen

Toteutettu: Toteutettu puutteellisesti. Virhekäsittely puuttuu.

Prioriteetti: 2

TV32: Kampanjan valinta haun avulla

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 3

TV33: Viestin lähetys soittoikkunassa

Toteutettu: Ei toteutettu.

Prioriteetti: 3

TV34: Avaa Inoa tai kohdeasiakkaan verkkisivut

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 3

TV35: Vie raportteja taulukkomuodossa

Toteutettu: Toteutettu puutteellisesti. Raportteja ei voi viedä taulukkolaskentaohjelmaan, vaan data on kopioitava selaimesta.

Prioriteetti: 3

TV36: Aikavälin rajausta myyjän raporteissa

Toteutettu: Toteutus ei toimi oikein.

Prioriteetti: 3

TV37: Pitkään kestäneen soiton hyväksyminen

Toteutettu: Toteutettu vaatimusten mukaisesti.

Prioriteetti: 3

TV38: Kannustusviesti

Toteutettu: Ei toteutettu.

Prioriteetti: 3

TV39: Lisää uusi käyttäjäasiakas

Toteutettu: Ei toteutettu.

Prioriteetti: 4

TV40: Katsele käyttäjäasiakkuuksia

Toteutettu: Ei toteutettu.

Prioriteetti: 4

TV41: Lisää uusi käyttäjätunnus

Toteutettu: Ei toteutettu.

Prioriteetti: 4

TV42: Käyttäjätunnusten tietojen katselu

Toteutettu: Ei toteutettu.

Prioriteetti: 4

TV43: Käyttäjätunnusten tietojen muuttaminen

Toteutettu: Ei toteutettu.

Prioriteetti: 4

TV44: Käyttäjän liittäminen asiakkuuteen

Toteutettu: Ei toteutettu.

Prioriteetti: 4

TV45: Muokkaa käyttäjäasiakaan nimeä

Toteutettu: Ei toteutettu.

Prioriteetti: 4

8.3.2 Ympäristövaatimukset

- Tietokannan on oltava MySQL versio 5.0.1 tai uudempi. Kasvanut vaatimus johtuu näkymien (view) käytöstä tietokannassa.
- WWW-palvelimen vähimmäismuistivaatimus on 16 megatavua. Erityisesti CakePHP:n suosimat suuret taulukkorakenteet vaativat odottamattoman paljon muistia.

8.4 Järjestelmässä havaitut puutteet

1. Jos tietokantaan ei saada yhteyttä, seuraa tästä omituisia uudelleenohjauksia, erityisesti silloin, jos järjestelmää ei ole asennettu palvelimen juureen.
2. Kampanjalla on oltava ainakin yksi sales_argument. Muuten soittosivulla tulee virheilmoituksia.
3. Soittosivulla soittamatta olevien lukumäärä voi mennä negatiiviseksi.
4. Soittosivulla soiton kesto lasketaan väärin. Tietokantatasolla soiton kesto on oikein.
5. Jos soittosivun tiedoissa on tallennettaessa virheitä, yhteyshenkilön nimikenttä tyhjenee itsestään.
6. Soittosivulla viimeksi käytetty myyntiargumentti tulisi olla esivalittu.
7. Soittosivulla uuden yhteydenottoajan kalenterin pitäisi hyväksyä päivämäärä kaksoisklikkauksella. Nyt valittuaan päivämäärän käyttäjä joutuu avaamaan kalenterin uudelleen valitakseen kellonajan.

8. Raportit-sivun aikarajaukset eivät toimi. Raportit-sivun tarvitaan asiakkaita -1, tarkoittaa ääretöntä lukumäärää. Tämä on loogista siinä mielessä, että jos yhtään tavoite saavutettua ei ole saatu, ei voida tilastollisesti määrittää tarvemäärää kyseisestä aineistosta.
9. Jos käyttäjän kirjautuminen vanhenee, ns. ajax-ikkunat (miten menee, haku ja salasanan vaihto) palauttavat kirjautumisikkunan. Workaround: avaa uusi ikkuna, jossa kirjautut järjestelmään, ja koita uudelleen samalta sivulta.
10. Ylläpitäjän raportit:
 - Soittojen lukumäärää ei näytetä.
 - Raporttien aikarajaus ei toimi.
 - Käyttäjien tavoitteiden asettamisessa ovat sarakeotsikot väärässä järjestyksessä.
 - Yhteydenottoraporttien Kesto ei toimi.
11. Kampanja-sivun JavaScript:t toimivat hitaasti vähänkään suuremmilla tietomäärillä.
12. Käytettäessä automaattista soitonvalintaa Peruuta-nappi ei palauta sivulle, jossa automaattista valintaa kutsuttiin.
13. Vaihdettaessa kampanjaa soittoruudun ollessa auki järjestelmä yrittää avata soittoruudussa olevan yrityksen toisessa kampanjassa ja aiheuttaa virheen.
14. Automaattinen soitonvalinta tarjoaa vain ensimmäistä vaihtoehtoa. Ei vaihda yritystä kun toiminto valitaan uudelleen.
15. Automaattinen soitonvalinta saattaa epäonnistua eikä tarjoa yhtään asiakasta.
16. Soittoruutu ei hyväksi tyhjää syötettä Yhteyshenkilö-kenttään.
17. Soittoruudun Inoa-linkki ja Avaa (www-linkki) avaavat tyhjä sivut, jos osoitteet puuttuvat.
18. Miten menee?-ikkuna ei huomauta jos asiakkaita vähemmän kuin tavoitteita.
19. Valittaessa yritys Käsittelemättömät- tai Käsitellytlistoilta navigointipalkki ei näytä kummassa kategoriassa ollaan.
20. Kampanjavalikossa kampanjoiden järjestystä/esitystapaa tulisi selkeyttää.
21. Kampanjan tallentaminen tekee ylimääräisen sivusiirtymän. Korjattavissa siten, että valikkopalkkien luonti suoritetaan vasta ApplicationController:n afterFilter():ssä, jolloin tieto onnistuneesta tallentamisesta voidaan asettaa tietoriville setInfoBar()-funktiolla.

22. Siirryttäessä kampanjaan, jossa vain ylläpito mahdollisuus muilta kuin ylläpitosivuilta tapahtuu virheitä.
23. Raportit ei tunnista avattuja sähköpostiviestejä.
24. Suorituskykyongelmat:
 - Kampanjan ylläpito on huonosti käytettävä isolla kohdeasiakasmäärällä. Tiedetyt toiminnot kestävät niin kauan, että tästä seuraa selaimen aikakatkaisuvaikutus. Luultavasti käytettävyyksensä on jossain 500 asiakasta per kampanja.
 - Järjestelmä hidastuu merkittävästi isoilla events-aulun määrillä. Tämä johtuu siitä, että järjestelmässä on tietokantakyselyitä, joissa käytetään in-rakennetta tehottomalla tavalla. In-rakenteiden korvaaminen join-rakenteella korjaisi tämän ongelman. Noin 100.000 eventin tapauksessa raporttisivut hidastuvat nykyversiossa käyttökelvottomiksi. Erityisesti näkymä latest_events tulisi korjata.
25. Kohdeasiakkaiden lisäys: Lisäys tapahtuu aina nykyiselle käyttäjälle myös Kampanja-sivulta. Näin ollen lisäävän ylläpitäjän on kuuluttava lisäyshetkellä kampanjaan, siirrettävä lisätyt asiakkaat pois itseltä, ennen kuin ne voi jakaa muille myyjille.
26. (X)html-sivuja eikä css-koodeja ei olla validoitu.
27. Asiakkaiden lisäyksen jälkeen mahdollisesti ohjaa väärälle sivulle.
28. Koodissa voi potentiaalisesti olla bugeja, joissa on käytetty users/id-tietoa campaigns_users/id sijasta. Tällöin järjestelmä hakee väärää campaigns_id-aulun tietoa.
29. Raportit-sivulla tavoitteiden tallentaminen ei aina onnistu.
30. Paikoin numeron puuttuminen (esim. nolla-arvolla) outoa.
31. Virhetilanteissa joutuu joskus tekemään ylimääräistä navigointia.
32. Käyttöohjeeseen tulisi lisätä käyttöä havainnollistavia kuvia.

8.5 Jatkokehitys

Järjestelmän jatkokehityksen kannalta kriittisintä olisi ensimmäisenä keskittyä edellisessä alaluvussa mainittujen bugien ja puutteiden korjaamiseen. Tämän jälkeen prioriteeteista seuraavana olisi suositeltavaa tehdä käyttäjähallinta. Vaatimusmäärittelydokumentissa on hahmotelmia käyttöliittymästä. Puuttuvalle kannustusjärjestelmälle on olemassa tarvittava tietokantarakenne. Puuttuville 3. prioriteetin toiminnoille on myös olemassa rajapinta suunnitteludokumentissa.

8.5.1 Tietokanta

Osa järjestelmän tietokantakyselyistä on erittäin raskaita nykyisellä toteutuksella. Kaikki in-rakenteet pitäisi korvata join-rakenteilla. Samalla tulisi myös miettiä mahdollisuutta siirtää vanhoja kampanjatietoja esimerkiksi historiatauluihin. Toinen vaihtoehto olisi salia kampanjoiden poistaminen järjestelmästä.

8.5.2 Yksikkötestaus

Järjestelmää jatkokehittäessä on suositeltavaa hyödyntää jo olemassaolevia yksikkötestejä. Nämä ovat käytettävissä kontrollerin tests/test kautta. Testien ajaminen vaatii kaksi asiaa:

- app/config/core.php-tiedostossa on asetettava debug-taso korkeammaksi kuin nolla
- Tietokannassa pitää olla myva-1.0-database.tar.gz-tiedostossa olevat pohjadata. Tietokannassa ei saa olla pohjadata lisäksi mitään muuta.

Olisi hyvin suositeltavaa muokata vanhat yksikkötestit sellaisiksi, että ne eivät vaadi erillisen datan ajamista. Osa vanhoista yksikkötesteistä on tehty näin. Testauksessa pitäisi miettiä mahdollisuuksia testata automaattisesti myös muita kuin model-luokkia. CakePHP valitettavasti ei anna helppoa mahdollisuutta tämän tekemiseen.

Liite 1. Tietokannan taulujen luontilauseet

Alla on tietokannan taulujen luontilauseet sekä pohjadataan lisäyslauseet.

Tietokannan luontilauseet:

```

CREATE TABLE `user_clients` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `name` varchar(128) NOT NULL UNIQUE,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

CREATE TABLE `users` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `username` char(16) NOT NULL UNIQUE,
  `name` varchar(128) NOT NULL,
  `email_address` varchar(64) NOT NULL,
  `phone_number` char(32) NOT NULL,
  `password` char(40) NOT NULL,
  `active_account` boolean NOT NULL,
  `campaign_admin` boolean NOT NULL,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  `user_client_id` int unsigned NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`user_client_id`) REFERENCES `user_clients`(`id`)
ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE `campaigns` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `name` varchar(128) NOT NULL UNIQUE,
  `orderer` varchar(128) NOT NULL,
  `start_time` date NOT NULL,
  `end_time` date,
  `message_subject` varchar(128) NOT NULL,
  `message` text NOT NULL,
  `target_amount` smallint unsigned NOT NULL,
  `accomplished_amount` smallint unsigned NOT NULL,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  `user_client_id` int unsigned NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`user_client_id`) REFERENCES `user_clients`(`id`)
ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE `campaigns_users` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `user_id` int unsigned NOT NULL,
  `campaign_id` int unsigned NOT NULL,
  `phone_number` char(32) NOT NULL,
  `email_address` varchar(64) NOT NULL,
  `target_amount` smallint unsigned NOT NULL,
  `accomplished_amount` smallint unsigned NOT NULL,
  `calls_per_day` smallint unsigned NOT NULL,
  `calls_per_day_suggestion` smallint unsigned NOT NULL,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`user_id`) REFERENCES `users`(`id`) ON DELETE CASCADE,
  FOREIGN KEY (`campaign_id`) REFERENCES `campaigns`(`id`) ON DELETE CASCADE,
  CONSTRAINT UNIQUE (`user_id`, `campaign_id`)

```

```

) ENGINE=InnoDB;

CREATE TABLE `message_links` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `target_url` text NOT NULL,
  `target_url_text` text NOT NULL,
  `beacon` boolean NOT NULL,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  `campaign_id` int unsigned NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`campaign_id`) REFERENCES `campaigns`(`id`)
  ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE `sales_arguments` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `argument` varchar(128) NOT NULL,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  `campaign_id` int unsigned NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`campaign_id`) REFERENCES `campaigns`(`id`)
  ON DELETE CASCADE,
  CONSTRAINT UNIQUE (`argument`, `campaign_id`)
) ENGINE=InnoDB;

CREATE TABLE `target_client_states` (
  `id` char(32) NOT NULL UNIQUE,
  `order` smallint NOT NULL UNIQUE,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

CREATE TABLE `target_clients` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `name` varchar(128) NOT NULL,
  `y_id` char(16) NOT NULL,
  `domain` varchar(128) NOT NULL,
  `exchange_number` char(32) NOT NULL,
  `address` text NOT NULL,
  `email_address` varchar(64) NOT NULL,
  `homepage` varchar(128) NOT NULL,
  `inoa_id` int unsigned,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  `target_client_state_id` char(32) NOT NULL,
  `campaign_id` int unsigned NOT NULL,
  `campaigns_user_id` int unsigned,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`campaign_id`) REFERENCES `campaigns`(`id`)
  ON DELETE CASCADE,
  FOREIGN KEY (`target_client_state_id`) REFERENCES `target_client_states`(`id`)
  ON DELETE CASCADE,
  FOREIGN KEY (`campaigns_user_id`) REFERENCES `campaigns_users`(`id`)
  ON DELETE SET NULL,
  CONSTRAINT UNIQUE (`name`, `campaign_id`)
) ENGINE=InnoDB;

CREATE TABLE `classifications` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `classification` varchar(64) NOT NULL,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  `campaign_id` int unsigned NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`campaign_id`) REFERENCES `campaigns`(`id`)
  ON DELETE CASCADE,
  CONSTRAINT UNIQUE (`classification`, `campaign_id`)

```

```

) ENGINE=InnoDB;

CREATE TABLE `classifications_target_clients` (
  `target_client_id` int unsigned NOT NULL,
  `classification_id` int unsigned NOT NULL,
  PRIMARY KEY (`target_client_id`, `classification_id`),
  FOREIGN KEY (`target_client_id`) REFERENCES `target_clients`(`id`)
    ON DELETE CASCADE,
  FOREIGN KEY (`classification_id`) REFERENCES `classifications`(`id`)
    ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE `event_types` (
  `id` char(32) NOT NULL UNIQUE,
  `priority` smallint NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

CREATE TABLE `event_type_classes` (
  `id` char(32) NOT NULL UNIQUE,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;
INSERT INTO `event_type_classes` (`id`) VALUES ('Soitto');
INSERT INTO `event_type_classes` (`id`) VALUES ('Viesti');
INSERT INTO `event_type_classes` (`id`) VALUES ('Tietomuutos');
INSERT INTO `event_type_classes` (`id`) VALUES ('Valittavissa');
INSERT INTO `event_type_classes` (`id`) VALUES ('Ulkopuolinen');

CREATE TABLE `event_type_classes_event_types` (
  `event_type_id` char(32) NOT NULL,
  `event_type_class_id` char(32) NOT NULL,
  `order` smallint NOT NULL,
  PRIMARY KEY (`event_type_id`, `event_type_class_id`),
  FOREIGN KEY (`event_type_id`) REFERENCES `event_types`(`id`)
    ON DELETE CASCADE,
  FOREIGN KEY (`event_type_class_id`) REFERENCES `event_type_classes`(`id`)
    ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE `reason_no_interests` (
  `id` varchar(128) NOT NULL UNIQUE,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

CREATE TABLE `events` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `start_time` datetime NOT NULL,
  `end_time` datetime NOT NULL,
  `contact_at` datetime,
  `interest` smallint NOT NULL,
  `comment` text NOT NULL,
  `contact_name` varchar(128) NOT NULL,
  `contact_title` varchar(32) NOT NULL,
  `contact_phone_number` varchar(32) NOT NULL,
  `contact_email_address` varchar(64) NOT NULL,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  `event_type_id` char(32) NOT NULL,
  `target_client_id` int unsigned NOT NULL,
  `user_id` int unsigned NOT NULL,
  `sales_argument_id` int unsigned,
  `reason_no_interest_id` varchar(128),
  PRIMARY KEY (`id`),
  FOREIGN KEY (`event_type_id`) REFERENCES `event_types`(`id`),
  FOREIGN KEY (`target_client_id`) REFERENCES `target_clients`(`id`),
  FOREIGN KEY (`user_id`) REFERENCES `users`(`id`)
    ON DELETE CASCADE,
  FOREIGN KEY (`sales_argument_id`) REFERENCES `sales_arguments`(`id`),

```

```

    FOREIGN KEY (`reason_no_interest_id`) REFERENCES `reason_no_interests`(`id`)
) ENGINE=InnoDB;

```

```

CREATE VIEW `latest_events` AS (
  SELECT *
    FROM `events`
   WHERE `end_time` IN (
      SELECT max(`end_time`)
        FROM `events` AS `latest`
       WHERE `latest`.`target_client_id` = `events`.`target_client_id`
         AND `latest`.`event_type_id` NOT IN (
            SELECT `event_type_id`
              FROM `event_type_classes_event_types`
             WHERE `event_type_class_id` = 'Ulkopuolinen'
          )
    )
)
);

```

```

CREATE TABLE `events_message_links` (
  `id` char(40) NOT NULL UNIQUE,
  `event_count` smallint unsigned NOT NULL,
  `message_link_id` int unsigned NOT NULL,
  `user_event_id` int unsigned NOT NULL,
  `target_client_event_id` int unsigned,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`message_link_id`) REFERENCES `message_links`(`id`)
    ON DELETE CASCADE,
  FOREIGN KEY (`user_event_id`) REFERENCES `events`(`id`)
    ON DELETE CASCADE,
  FOREIGN KEY (`target_client_event_id`) REFERENCES `events`(`id`)
    ON DELETE CASCADE,
  CONSTRAINT UNIQUE (`message_link_id`, `user_event_id`),
  CONSTRAINT UNIQUE (`message_link_id`, `target_client_event_id`)
) ENGINE=InnoDB;

```

```

CREATE TABLE `profile_classes` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  `name` varchar(32) NOT NULL UNIQUE,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

```

```

CREATE TABLE `profile_classes_users` (
  `user_id` int unsigned NOT NULL,
  `profile_class_id` int unsigned NOT NULL,
  `profile_value` smallint unsigned NOT NULL,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  PRIMARY KEY (`user_id`, `profile_class_id`),
  FOREIGN KEY (`user_id`) REFERENCES `users`(`id`)
    ON DELETE CASCADE,
  FOREIGN KEY (`profile_class_id`) REFERENCES `profile_classes`(`id`)
    ON DELETE CASCADE
) ENGINE=InnoDB;

```

```

CREATE TABLE `encouragements` (
  `id` int unsigned AUTO_INCREMENT NOT NULL UNIQUE,
  -- encouragement tulisi olla UNIQUE, mutta tyyppin vuoksi ei ole
  `encouragement` text NOT NULL,
  `created` datetime NOT NULL,
  `modified` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

```

```

CREATE TABLE `encouragements_profile_classes` (
  `encouragement_id` int unsigned NOT NULL,
  `profile_class_id` int unsigned NOT NULL,
  PRIMARY KEY (`encouragement_id`, `profile_class_id`),
  FOREIGN KEY (`encouragement_id`) REFERENCES `encouragements`(`id`)
)

```

```

        ON DELETE CASCADE,
    FOREIGN KEY (`profile_class_id`) REFERENCES `profile_classes`(`id`)
        ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE VIEW `event_types_with_classes` AS (
    SELECT `priority`, `event_type_id`, `event_type_class_id`, `order`
    FROM `event_types`
        INNER JOIN `event_type_classes_event_types`
            ON `event_type_id` = `event_types`.`id`
        INNER JOIN `event_type_classes`
            ON `event_type_classes`.`id` = `event_type_classes_event_types`.`event_type_class_id`
);

```

Tietokannan pohjadataan lisäyslauseet:

```

LOAD DATA LOCAL INFILE `target_client_states.txt`
    INTO TABLE `target_client_states`
    -- CHARACTER SET `iso-8859-1`
    FIELDS TERMINATED BY `\\t` ENCLOSED BY `"` ESCAPED BY `\\`
    LINES TERMINATED BY `\\n` STARTING BY `"`
    (`id`, `order`)
;

LOAD DATA LOCAL INFILE `event_types.txt`
    INTO TABLE `event_types`
    -- CHARACTER SET `iso-8859-1`
    FIELDS TERMINATED BY `\\t` ENCLOSED BY `"` ESCAPED BY `\\`
    LINES TERMINATED BY `\\n` STARTING BY `"`
    (`id`, `priority`)
;

LOAD DATA LOCAL INFILE `event_type_classes_event_types.txt`
    INTO TABLE `event_type_classes_event_types`
    -- CHARACTER SET `iso-8859-1`
    FIELDS TERMINATED BY `\\t` ENCLOSED BY `"` ESCAPED BY `\\`
    LINES TERMINATED BY `\\n` STARTING BY `"`
    (`event_type_id`, `event_type_class_id`, `order`)
;

LOAD DATA LOCAL INFILE `profile_classes.txt`
    INTO TABLE `profile_classes`
    -- CHARACTER SET `iso-8859-1`
    FIELDS TERMINATED BY `\\t` ENCLOSED BY `"` ESCAPED BY `\\`
    LINES TERMINATED BY `\\n` STARTING BY `"`
    (`name`)
;

LOAD DATA LOCAL INFILE `reason_no_interests.txt`
    INTO TABLE `reason_no_interests`
    -- CHARACTER SET `iso-8859-1`
    FIELDS TERMINATED BY `\\t` ENCLOSED BY `"` ESCAPED BY `\\`
    LINES TERMINATED BY `\\n` STARTING BY `"`
    (`id`)
;

```

Tiedosto target_client_states.txt

```

Ei aloitettu 10
Kesken 20
Tavoitetta ei saavutettu 30
Tavoite saavutettu 40

```

Tiedosto event_types.txt

Luotu 0
 Tietojen muutos 0
 Lähetetty 10
 Avattu 20
 Klikattu 30
 Väärä henkilö 40
 Varattu 60
 Ei vastaa 50
 Ei paikalla 40
 Numero ei toimi 0
 Tavoitettu 0

Tiedosto event_type_classes_event_types.txt

Luotu Tietomuutos 10
 Tietojen muutos Tietomuutos 20
 Tietojen muutos Valittavissa 10
 Lähetetty Viesti 20
 Avattu Viesti 10
 Avattu Ulkopuolinen 10
 Klikattu Viesti 30
 Klikattu Ulkopuolinen 20
 Väärä henkilö Soitto 50
 Väärä henkilö Valittavissa 50
 Varattu Soitto 40
 Varattu Valittavissa 40
 Ei vastaa Soitto 20
 Ei vastaa Valittavissa 20
 Ei paikalla Soitto 10
 Ei paikalla Valittavissa 10
 Numero ei toimi Soitto 30
 Numero ei toimi Valittavissa 30
 Tavoitettu Soitto 60
 Tavoitettu Valittavissa 60

Tiedosto profile_classes.txt

Prospektointimotivaatio
 Prospektoinnin tavoitetaso
 Tavoitehämmennys
 Murehtija
 Himovalmistautuja
 Hyperammattilainen
 Esiintymispelko
 Roolikieltäjä
 Päättämätön
 Sosiaalinen epävarmuus
 Referenssi-inho
 Luurikammo
 Kaverinkarttaja
 Perhepehmo
 Vaistovastustaja

Tiedosto reason_no_interests.txt

Juuri hankkinut
 Tekee itse
 Ajankohtaista myöhemmin
 Ei budjettia
 Budjetointi kesken/suorittamatta

Liite 2. Luokkakaaviot

Seuraavilla sivuilla on luokkakaaviot LK-kuva1–LK-kuva3. Luokkakaaviokuvat vastaavat järjestelmän rajapintoja vain viitteellisesti. Erityisesti metodit ja muuttujarajapinnat on toteutettu toisin. Paikkansapitävä dokumentaatio löytyy lähdekoodista ja PHPDOC-dokumentaatiosta. Merkittävimmät yleiset muutokset on kuvattu luvussa 8.2.

LK-kuva1: Model

Model

Event
+getLatestEvent(\$target_client_id: array)
+getTargetClientEvents(\$target_client_id:int, \$start_time:DateString=null, \$send_time:DateString=null): array
+getDifferentEmailCounts(\$user_id:int,\$campaign_id:int): int
+getTargetClientCountByEventType(\$user_id:int, \$campaign_id:int, \$events:array, \$include:boolean=true, \$start_time:DateString=null, \$send_time:DateString=null): int
+getClientEventTypesReport(\$user_id:int, \$campaign_id:int, \$event_type_classes:array, \$start_time:DateString, \$send_time:DateString): array
+chooseNextCall(\$user_id:int,\$campaign_id:int): int
+getEmailLinksReport(\$user_id:int,\$campaign_id:int, \$start_time:DateString=null, \$send_time:DateString=null): array
+getBestContactTimes(\$user_id:int,\$campaign_id:int, \$time_frame:int): string
+getBestAccomplishedTimes(\$user_id:int,\$campaign_id:int, \$time_frame:int): string
+getBestCallsAmountPerDay(\$user_id:int,\$campaign_id:int): int
+getCampaignEvents(\$user_id:int,\$campaign_id:int, \$start_time:DateString=-1, \$send_time:DateString=-1): array

Campaign
+checkNearHits(\$path_to_file:string): array
+searchContacts(\$search:array): array

TargetClient
+getTargetClientStatesReport(\$user_id:int, \$campaign_id:int, \$start_time:DateString, \$send_time:DateString): array
+getTargetClientsNeeded(\$user_id:int,\$campaign_id:int, \$type:string): int

UserProfileValues
+getEncouragement(\$user_id:int): string

TargetClientsClassification
+getTargetClientsClassificationReport(\$user_id:int, \$campaign_id:int, \$start_time:string=null, \$send_time:string=null): array

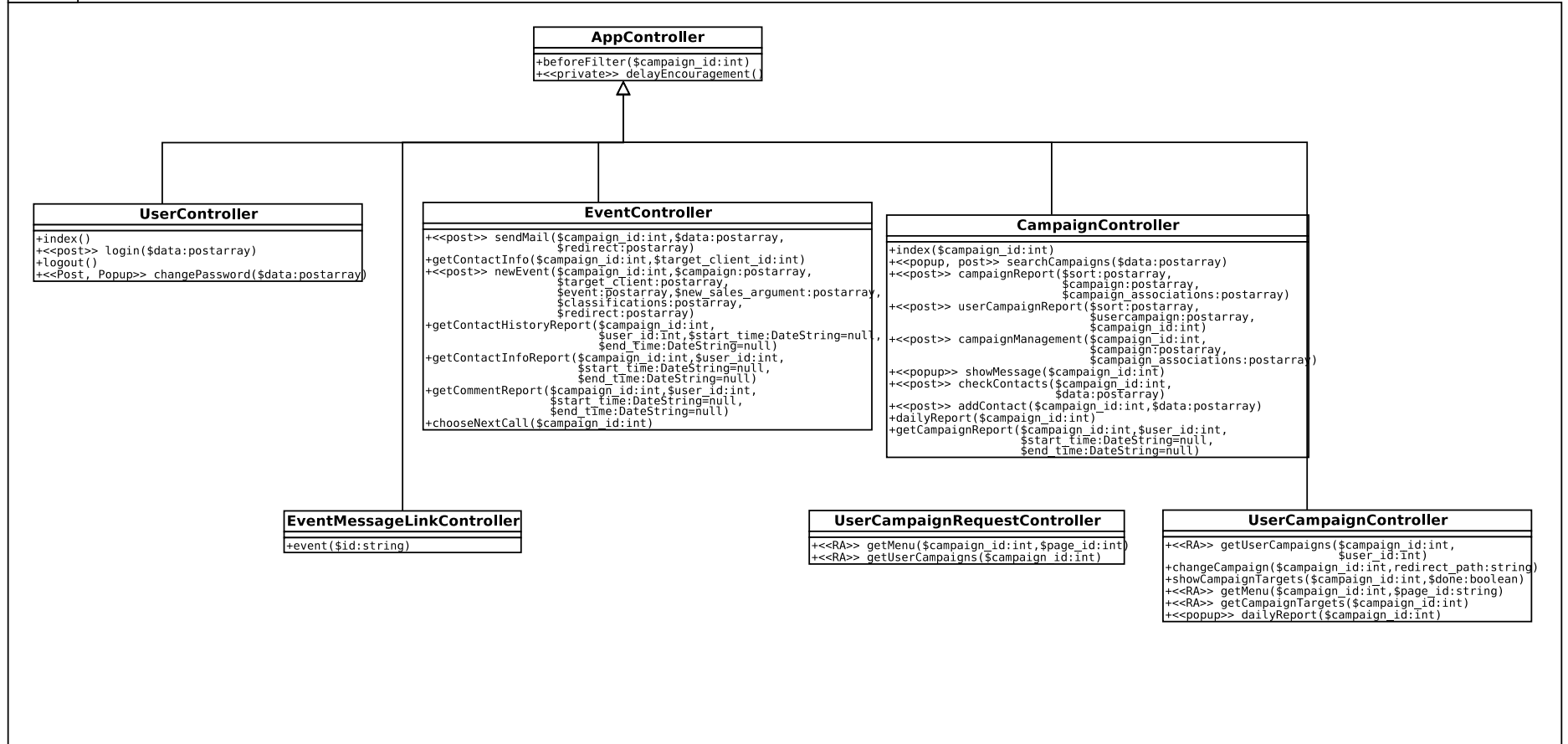
EventMessageLink
+messageEvent(\$id:string): boolean

UserCampaign
+getTargetClientsCount(\$user_id:int,\$campaign_id:int, \$limit:int): int
+getCallsPerDay(\$user_id:int,\$campaign_id:int): int
+getCallsDoneToday(\$user_id:int,\$campaign_id:int): int
+getUserCampaigns(\$user_id): array
+getCampaignList(\$user_id:int,\$campaign_id:int, \$done:boolean): array
+getUsersTargets(\$campaign_id:int): array
+getWorkHoursNeededToday(\$user_id:int,\$campaign_id:int): DateString

MessageLink
+parseMessage(\$message:string): boolean

LK-kuva2: Controller

Controller



View

