

AINEOPINTOJEN HARJOITUSTYÖ: TIETOKANTASOVELLUS
2013/2

Kalenterijärjestelmä

mustached-octo-happiness

Tekijä
SAMULI THOMASSON

7. joulukuuta 2013

Sisältö

1	Johdanto	2
2	Järjestelmä	3
2.1	Yleiskuva järjestelmästä	3
2.2	Käyttötapaukset	3
2.2.1	Autentikointi	3
2.2.2	Kalenterit	4
2.2.3	Kalenterikohteet	5
2.2.4	Vieraskalenterit	7
2.3	Järjestelmän tietosisältö	8
2.3.1	Tietokohde: Kalenteri	8
2.3.2	Kalenterikohteet	8
2.4	Relaatiotietokanta	10
2.5	Järjestelmän yleisrakenne	13
2.6	Järjestelmän komponentit	13
2.7	Käyttöliittymä	13
3	Käyttö	15
3.1	Asennustiedot	15
3.1.1	Esivaatimukset	15
3.1.2	Tietokanta	15
3.1.3	Kehityspalvelin	15
3.1.4	Kehitysympäristö	16
3.1.5	Tuotantoasennus	16
3.2	Käyttöohje	16
4	Testaus, tunnetut bugit ja puutteet & jatkokehitysideat	17

Luku 1

Johdanto

Tämä kalenterijärjestelmä mahdollistaa käyttäjille kalenterien ylläpidon selaimesta. Sovelluksen käyttäjä pystyy katsomaan, lisäämään ja muokkaamaan (kalenteri)merkintöjä omissa kalentereissaan. Kalenteri koostuu joukosta merkintöjä, kuten tapahtumia ja muistutuksia. Järjestelmä mahdollistaa merkintöjen asettamisen julkisesti nähtäville. Myös kokonaisista kalentereista voi tehdä julkisia. Julkisesta kalenterista voi myös tehdä vapaasti muokattavan, jolloin kuka tahansa voi ehdottaa merkintöjä kalenteriin. Vapaasti muokattavassa kalenterissa lopullinen määräysvalta on kalenterin omistajalla – omistajan omia tai hyväksymiä merkintöjä voi poistaa vain omistaja.

Järjestelmän toteutuskieli on Haskell ja kehyksenä toimii *Yesod*. Järjestelmä on yksinkertaisuudessaan ohjelma, joka kuuntelee jossain portissa web-palvelimena. Sitä voi siis ajaa itsenäisenä web-palvelimena suoraan portissa 80. Yleensä samankaltaiset sovellukset kuitenkin ajetaan paikallisessa portissa ja välitetään dedikoidun web-palvelimen (Nginx ja vastaavat) kautta.

Järjestelmän alustajärjestelmältä vaaditaan, että lähdekoodi voidaan kääntää sille. Staattisten kirjastojen ansoista ei käyttöönotto vaadi alustalta kuin käännetyin ohjelman ajamisen, staattiset verkkosivuresurssit (JavaScript ja CSS) sekä tuetun tietokannan. Ohjelma pitää ajaa omassa portissaan, eli ohjelman käyttö vaatii joko ainoan web-palvelimen aseman ulkoisessa portissa, tai (parempi) käänteisproxyä ymmärtävän web-palvelimen alustalta.

Järjestelmä suunnitellaan PostgreSQL-kannalle; vaihdon edellytyksiä on että persistent tukee sitä, ja että tekijä ei ole käyttänyt liian erikoisia SQL-konstruktioita. Itse järjestelmän ominaisuuksien käyttöön asiakasohjelmassa ei tarvitse JavaScriptiä. JavaScriptiä voidaan kuitenkin käyttää käyttökokemuksen parantamiseksi.

Luku 2

Järjestelmä

2.1 Yleiskuva järjestelmästä

Käyttäjät toimivat aina selainkäyttöliittymän kautta. Käyttätapauskaaviossa (2.1) on esitelty ohjelman päätoiminnot. Ne on jaettu kolmeen pääryhmään: käyttäjien autentikointi, operaatiot omassa kalenterissa ja operaatiot jonkun muun kalenterissa.

Järjestelmän käyttäjiä on kahdenlaisia: vierailijoita ja varsinaisia käyttäjiä:

Vierailijoita ovat kaikki jotka eivät ole kirjautuneet sisään, riippumatta siitä onko heillä tunnisteita järjestelmässä vai ei.

Käyttäjiä ovat kaikki sisäänkirjautuneet.

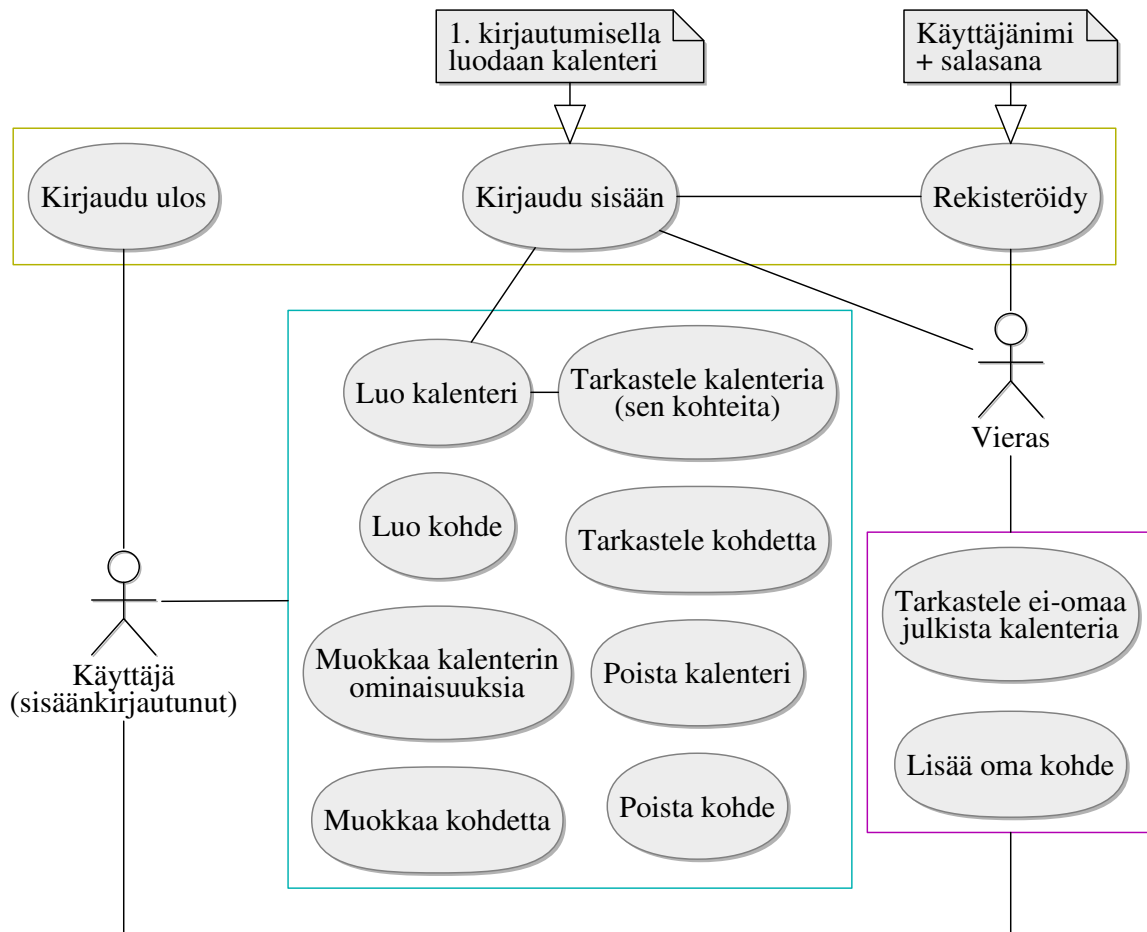
2.2 Käyttötapaukset

Tässä luvussa esitellään sovelluksen eri käyttötapaukset. Ellei toisin mainita, niin esiteltävät käyttötapaukset ovat vain varsinaisille käyttäjille. Käyttötapaukset on jaettu kolmeen joukkoon: autentikointi, kalenterit kohteineen ja vieraskalenterit. Käyttötapauskaavio (2.1) havainnollistaa käyttötapausiatapauksia; eri kategoriat on siinä jaettu visuaalisiin lohkoihin.

2.2.1 Autentikointi

Rekisteröityminen *Vain vieraille.* Ensimmäinen asia jonka uuden tulevan käyttäjän täytyy valitettavasti tehdä, on rekisteröidä jokin tunniste itsestään järjestelmään. Tunniste on esimerkiksi sähköpostiosoite, jolla käyttäjä voi todentaa itsensä.

Rekisteröitymiseen tarjotaan monia vaihtoehtoja. Perinteinen käyttäjänimen (sähköpostiosoitteen) ja salasanan tallennus palvelimelle on vakiovaihtoehto, mutta myös modernempia metodeja tarjotaan: BrowserId, OpenId ja Gmail, mahdollisesti muitakin.



Kuva 2.1: Kalenterijärjestelmän käyttötapauskaavio. Keltaisessa laatikossa on autentikointi, turkoosisssa kalenterien ja kohteiden käyttötapaudet, ja violetissa laatikossa on julkisten kalenterien käyttötapaudet.

Sisäänkirjautuminen *Vain vieraille.* Käyttäjien täytyy pystyä autentikoimaan itsensä aiemmin rekisteröityneeksi käyttäjäksi (mahdollisesti itsekseen), eli järjestelmän kontekstissa korottamaan itsensä vieraasta käyttäjäksi. Sisäänkirjautumiseen käytetään samaa metodia jota käyttäjä käytti rekisteröityessään.

2.2.2 Kalenterit

Käyttäjällä voi olla usea kalenteri järjestelmässä. Käyttäjän luomaa kalenteria kutsutaan *käyttäjän omistamaksi* kalenteriksi. Olemassaolevan kalenterin omistajaa ei voi vaihtaa. Kalenteritoiminnot:

Luonti

Kalenterin luominen on ensimmäinen asia jonka uusi käyttäjä tekee, ja se on helppoa: häneltä kysytään heti ensimmäisen sisäänkirjautumisen jälkeen nimeä kalenterille tekstikentällä. Kalenterille voi halutessaan asettaa jonkin toisen väriattribuutin kuin oletuksen. Lisää attribuutteja voi varsinaisesti säätää myöhemmin, kun kalenteri on luotu (ks. kohta Muokkaus).

Tarkastelu (*kalenterinäkymä*)

Kalenterien tarkastelu on koko järjestelmän tärkein ja laajin käyttötapaus. Sen voisi esittää myös useana pienempänä käyttötapauksena.

Kalentereita ei pääasiassa tarkastella yksitellen, vaan useita päällekkäin. Tarkastelu tapahtuu *kalenterinäkymässä*. Kalenterinäkymässä näytetään kerrallaan yhden viikon tapahtumat kalentereista.¹ Näkyvää viikkoa voi vaihtaa seuraavaan, edelliseen, senhetkiseen viikkoon tai johonkin tiettyyn viikkoon.²

Näkymässä näkyvät kalenterit voi valita.

Näkymässä on esillä keinoja kalenterikohteiden operointiin. Näkyvistä kalenterikohteista näytetään lyhyet yhteenvedot sekä keinot niiden muokkaukseen.

Muokkaus (*kalenterin ominaisuudet*)

Kalenterit ovat vain kohteita kokoavia entiteettejä, mutta niillä on silti muutama muokattava attribuutti jotka vaikuttavat niiden toimintaan. Näitä ovat:

Nimi Kalenterin muista identifioiva tunniste.

Väri Minkä värisinä kalenterin kohteet näkyvät kalenterinäkymässä.

Julkinen Voiko kalenterin kohteita tarkastella kuka vain (käyttäjä tai vieras).

Yhteinen Voiko kuka vain lisätä kalenteriin omia kohteita. Selvästi tämä attribuutti on järkevä vain kun kalenteri on julkinen.

Kaikkia attribuutteja voi muokata muokkausnäkymässä, jossa näkyvät myös nykyiset arvot.

Poisto

Kalenterin voi myös poistaa, ja sekin tapahtuu muokkausnäkymässä. Poistoon kysytään varmistus mikäli kalenteri sisältää kohteita, sillä kohteet poistuvat myös.

2.2.3 Kalenterikohteet

Kalenterit koostuvat erilaisista (kalenteri)kohteista. Kaikille erilaisista kohteista on samantapaiset käyttöliittymät, joissa keskinäiset erot muodostuvat kohdetyyppien erilaisista tietokentistä. Yhteisiä ominaisuuksia, kuten päivämääriä operoidaan samoin kohdetyypistä riippumatta. Lisäksi jokaisesta kohteesta tallennetaan metadattaa, kuten luonti- ja muokauspäivämääriä.

Yleiseen kalenterikohteeseen liittyvät käyttötapaukset:

¹Jos aikaa riittää niin näytettävä sisältö on paremmin säädettävissä (kuukausi, päivä, tai muita näkymiä).

²Tiettyyn viikkoon vain menemällä useasti eteen tai taakse, mitään viikkosyötettä ei (GET-parametria lukuunottamatta voi (kurssin versiossa) antaa.

Luonti

Kalenterikohteita lisätään valitsemalla haluttu kohdetyyppi kalenterinäkymässä. Kalenterinäkymässä voi kohteen lisäykseen edetä joko tietylle ajalle aikanäkymässä (viikkonäkymässä), tai aikanäkymän ulkopuolelta.

Tarkastelu

Kalenterikohteita tarkastellaan kalenterinäkymässä ja kohteita sekä niiden yksityiskohtia valitsemalla kohteen kalenterinäkymässä.

Muokkaus

Kohteen muokkaukseen edetään sen yksityiskohtaisesta näkymästä. Muokattavia attribuutteja on useita ja ne riippuvat kohdetyypistä. Kaikista kohteista muokattavat attribuutit:

Kuvaus Tärkein attribuutti on kuvaus kohteesta.

Kohteiden metadata (luodaan implisiittisesti):³

Luontipäivämäärä Jotta voidaan tarkistaa koska kohde lisättiin kalenteriin.

Viimeinen muokkaus Koska edellinen muokkaus tehtiin?

Kohdetyypeille ominaiset muokattavat attribuutit:

(Nämä muistuttavat (hyvin löyhästi) iCalendar-määrittelyä)

Tapahtuma (Event)

Kalenterien tyypillinen ominaisuus on tapahtumien muistaminen ja niistä muistuttaminen. Tärkeimmät attribuutit:

Alkamisajankohta Koska tapahtuma alkaa?

Kesto Kauanko tapahtuma kestää?

Toistuvuus Kuinka usein ja milloin tapahtuma toistuu (jos toistuu)?

Hälytysajat Kun tapahtumasta halutaan muistutus. Esimerkiksi tietty aika ennen tapahtuman alkamisajankohtaa.

Kommentti Sivuhuomautus tapahtumasta.

Tärkeys Kuinka tärkeä tapahtuma on (jollain asteikolla).

Paikka Tapahtumaan liittyvä maantieteellinen sijainti.

Osallistujat Henkilöt, joita tapahtuma koskee.

Muistiinpano (Note)

Muistiinpanot ovat yleiskäyttöisiä kohteita. Niitä voi käyttää esimerkiksi päiväkirjana tai mikroblogina. Attribuutit:

Sisältö Yleisen kohteen kuvauksesta eroava, rikkaampi kenttä. Sisältö voi olla esimerkiksi HTML:ää tai Markdown:ia.

³Huomaa, että nämä puuttuvat implementaatiosta ainakin toistaiseksi (alhainen prioriteetti).

Tehtävä (Todo)

Tehtävän voi tehdä mistä tahansa tehtävästä. Attribuutit:

Valmis Onko todo valmistunut?

Toistuvuus Toistuuko tehtävä säännöllisesti, milloin?

Hälytysajat Kun tehtävästä halutaan muistutus. Esimerkiksi tietty aika ennen tehtävän deadlinea.

Tärkeys Kuinka tärkeä tehtävä on (jollain asteikolla).

Kopiointi

Kohteista voi luoda kopioita, tai tarkemmin olemassaolevaa kohdetta voi lähteä muokkaamaan ja tallentaa muokattu versio uutena kohteena. Tämäkin tapahtuu yksityiskohtaisesta näkymästä.

Poisto

Kohteen poisto tapahtuu myös sen yksityiskohtaisesta näkymästä. Poistoon kysytään aina varmistus. Toistuvien kohteiden kohdalla kysytään, halutaanko poistaa kaikki toistot vai ainoastaan valittu toisto.

Jako⁴

Yksittäisiä kohteita voi jakaa sellaisenaan muille järjestelmän käyttäjille, ja myös viedä tekstimuotoon esimerkiksi liitettäväksi sähköpostiin. Jakamiseen edetään yksityiskohtaisesta näkymästä.

Toiselle käyttäjälle kohdistuvalle jaolle annetaan käyttäjäjoukko, jotka saavat heille jaetusta kohteesta ilmoituksen. He päättävät haluavatko lisätä kohteen johonkin omaan kalenteriin.

2.2.4 Vieraskalenterit

Vieraskalenteri-käyttötapaukset koskevat käyttäjiä, vieraita ja niitä kalentereita jotka eivät ole heidän omistamia. Vieraskalenteria pystyy tarkastelemaan ja mahdollisesti (omistajan niin asettaessa) tekemään lisäyksiä, mutta *ei* muutoksia kalenteriin tai siinä oleviin muiden omistamiin kalenterikohteisiin. Näiden käyttötapausten oletus on, että kohteena olevan kalenterin omistaja on sallinut julkiset toiminnot. Käyttötapaukset:

Tarkastelu

Kaikille Vieraskalenteria voi tarkastella samalla tavalla kuin omaakin kalenteria (ks. 2.2.2).

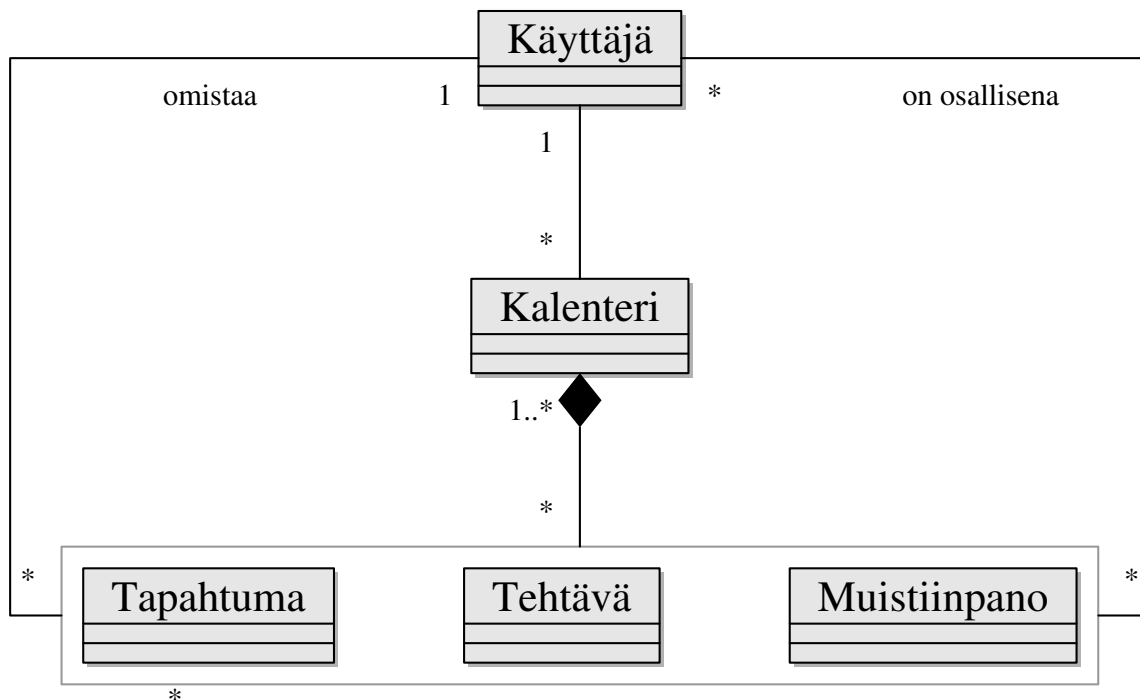
Kohteen luonti

Kaikille. Vieraskalenteriin voi lisätä kohteita jos kalenterin omistaja on sen sallinut. Lisäys tapahtuu samalla tavalla kuin oman kalenterin tapauksessa.

Vieraalta saatetaan vaatia kuva- tai muu varmennus (captcha).

2.3 Järjestelmän tietosisältö

Järjestelmän tietokohteet on esitetty pääpiirteittäin kaaviossa 2.2. Tietosisällöt on kuvattu yksityiskohtaisemmin alaluvuissa. Taulukoissa on esitetty kunkin tietokohteen attribuuttien nimet, arvojoukot ja kuvaukset. Attribuutti on pakollinen ellei arvojoukkoon ole merkattu (V). Listattujen attribuuttien lisäksi myös metatiedot (koska luotu, koska viimeksi muokattu) tallennetaan.



Kuva 2.2: Käsitokaavio järjestelmän tietokohteista. Harmaaseen laatikkoon osoittavat relaatiot liittyvät jokaiseen kohteeseen laatikossa.

2.3.1 Tietokohde: Kalenteri

Attribuutti	Arvojoukko	Kuvailu
Omistaja	Käyttäjä	Kuka loi ja omistaa kalenterin.
Nimi	Merkkijono	Nimi ja samalla tunniste kalenterille.
Kuvaus	Merkkijono (V)	Lisätietoa kalenterista.
Väri	Merkkijono (RGB)	Tunnisteväri kalenterin kohteille.
Onko julkinen	Totuusarvo	Voiko kalenteria katsoa kuka vain?
Julkisesti muokattava	Totuusarvo	Voiko kohteita lisätä kuka vain?

Kalenteriin kuuluu useita kalenterikohteita. Kalenteri kuuluu yhdelle käyttäjälle, omistajalle.

2.3.2 Kalenterikohteet

Kaikille kalenterikohteille tyypilliset attribuutit:

Attribuutti	Arvojoukko	Kuvailu
Omistaja	Käyttäjä	Kohteen luoja ja omistaja.
Nimi	Merkkijono	Nimike kohteelle.

Jokainen kalenterikohde on yhden käyttäjän omistuksessa, mutta voi olla useassa, myös toisen käyttäjän kalenterissa. Kohteet kuuluvat aina vähintään yhteen kalenteriin, eli **kalenterikohde ei voi olla orpo**.

Tärkeys ilmaistaan neliportaisella asteikolla: joutava, normaali, tärkeä tai kriittinen. *Henkilöt* voivat viitata niin järjestelmän käyttäjiin kuin ulkopuolisiin kontakteihin (nimi, sähköposti jne.); käytännössä merkkijonoja.

Tietokohde: Muistiinpano

Attribuutti	Arvojoukko	Kuvailu
(Taulukon 2.3.2 attribuutit)		
Sisältö	<i>Rikas teksti</i>	Muistiinpanon sisältö.

Rikas teksti on joko paljasta tekstiä, tai markdownia.⁵

Tietokohde: Tehtävä

Attribuutti	Arvojoukko	Kuvailu
(Taulukon 2.3.2 attribuutit)		
Valmis	Totuusarvo	Onko Tehtävä jo tehty?
Toistuvuus	<i>Toisto</i>	Mihin aikaan, miten toistuu?
AlkuPvm	Päivämäärä	Mistä alkaen kohde on relevantti?
ViimPvm	Päivämäärä (V)	Mistä alkaen kohde on vanhentunut?
Hälytys	Aikaero	Halutaanko ennen deadlinea hälytys?
Tärkeysaste	<i>Tärkeys</i>	Kuinka kriittinen kohde on.

Toisto kertoo ajankohdan jolloin tehtävän on oltava tehty (deadline), ja mahdollisesti myös kuinka useasti (päivittäin, viikoittain, kuukausittain, ...) kohde toistuu ja mihin asti toisto jatkuu.

⁵Riippuen aikataulusta

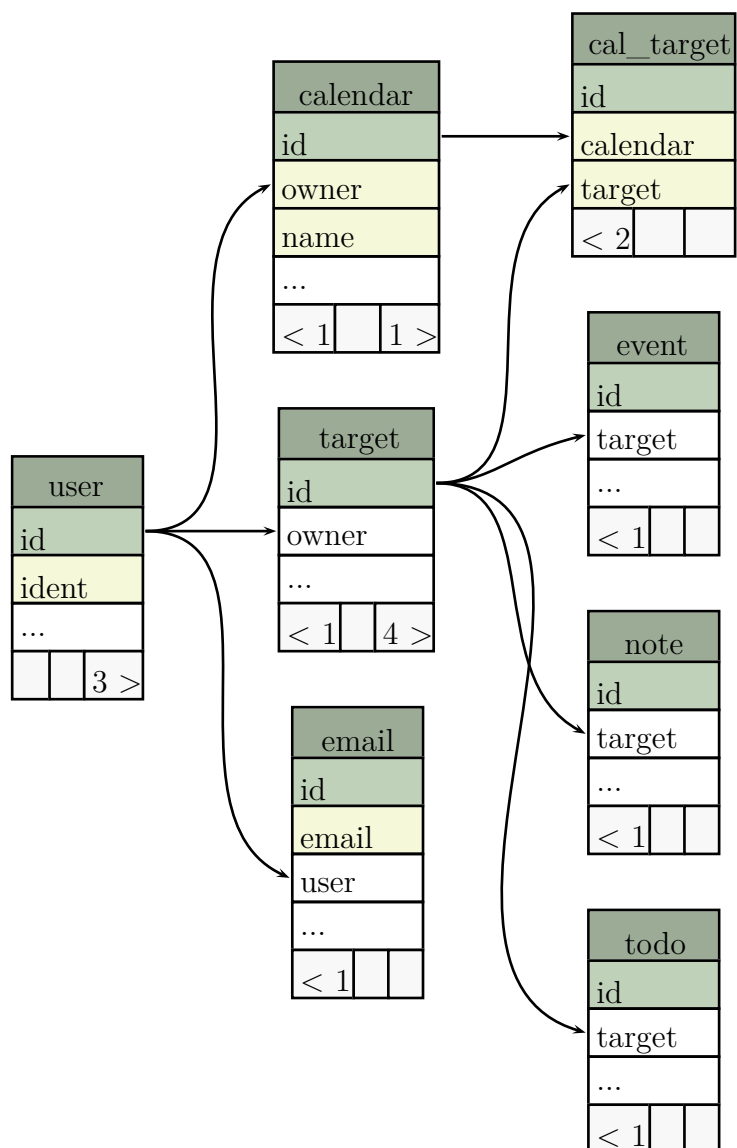
Tietokohde: Tapahtuma

Attribuutti	Arvojoukko	Kuvailu
(Taulukon 2.3.2 attribuutit)		
Toistuvuus	Toisto	Mihin aikaan, miten toistuu?
AlkuPvm	Päivämäärä	Mistä alkaen kohde on relevantti?
ViimPvm	Päivämäärä (V)	Mistä alkaen kohde on vanhentunut?
Paikka	Merkkijono (V)	Kohteeseen liittyvä sijainti.
Tärkeysaste	<i>Tärkeys</i>	Kuinka kriittinen kohde on.
Hälytys	Aikaero (V)	Halutaanko ennen tapahtumaa hälytys?
Osallistujat	<i>Henkilöt</i> (V)	Kohteeseen liittyvät henkilöt.
Kuvaus	Merkkijono (V)	Huomautus kohteesta.

2.4 Relaatiotietokanta

Järjestelmän tietokannan taulut ja niiden pää- ja vierasavaimet on kuvattu kaaviossa 2.3.⁶ Taulukossa 2.4 on lueteltu taulujen sarakkeet ja niiden tyypit kannassa. Todellisuudessa kanta on luotu ja käytetään järjestelmässä *persistent*-kirjaston avulla, ks. tiedosto `config/models`.

⁶Lieköhän pilkulleen tarkka uusimman määrittelyn kanssa?



Kuva 2.3: Järjestelmän relaatiotietokannan taulut ja yhteydet. Huomaa, että nuolet osoittavat vierasavaimiin.

Table	Column	Type	Size	Nulls	Auto
<u>cal_target</u>	calendar	int8	19		
<u>cal_target</u>	id	serial	10		√
<u>cal_target</u>	target	int8	19		
<u>calendar</u>	color	varchar	2147483647		
<u>calendar</u>	desc	varchar	2147483647	√	
<u>calendar</u>	id	serial	10		√
<u>calendar</u>	name	varchar	2147483647		
<u>calendar</u>	owner	int8	19		
<u>calendar</u>	public	bool	1		
<u>calendar</u>	publicedit	bool	1		
<u>email</u>	email	varchar	2147483647		
<u>email</u>	id	serial	10		√
<u>email</u>	user	int8	19	√	
<u>email</u>	verkey	varchar	2147483647	√	
<u>event</u>	alarm	varchar	2147483647		
<u>event</u>	attendees	varchar	2147483647		
<u>event</u>	begin	timestamp	29,6		
<u>event</u>	comment	varchar	2147483647	√	
<u>event</u>	end	timestamp	29,6	√	
<u>event</u>	id	serial	10		√
<u>event</u>	place	varchar	2147483647	√	
<u>event</u>	repeat	varchar	2147483647		
<u>event</u>	target	int8	19		
<u>event</u>	urgency	int8	19		
<u>note</u>	content	varchar	2147483647		
<u>note</u>	id	serial	10		√
<u>note</u>	target	int8	19		
<u>target</u>	id	serial	10		√
<u>target</u>	name	varchar	2147483647		
<u>target</u>	owner	int8	19		
<u>todo</u>	alarm	varchar	2147483647		
<u>todo</u>	begin	timestamp	29,6		
<u>todo</u>	done	bool	1		
<u>todo</u>	end	timestamp	29,6	√	
<u>todo</u>	id	serial	10		√
<u>todo</u>	repeat	varchar	2147483647		
<u>todo</u>	target	int8	19		
<u>todo</u>	urgency	int8	19		
<u>user</u>	id	serial	10		√
<u>user</u>	ident	varchar	2147483647		
<u>user</u>	password	varchar	2147483647	√	

Kuva 2.4: Taulujen kentät ja tyytit kannassa.

2.5 Järjestelmän yleisrakenne

Järjestelmän pääkomponentit on sijoitettu hakemistoihin seuraavasti.

config/	Järjestelmän asetukset. <code>postgresql.yml</code> on tietokantayhteyksille ja <code>settings.yml</code> sisältää yleiset asetukset. Syntaksina molemmissa YAML.
Handler/	Kontrollerit on sijoitettu Handler -kansioon. Kontrollerien URL:it määrätään <code>config/routes</code> -tiedostossa.
templates/	Näkymät sijaitsevat <code>templates</code> -kansiossa <code>.hamlet</code> ja <code>.lucius</code> -päätteillä.
static/	Staatinen sisältö (kääntövaiheessa).
messages/	Yesodin sisäänrakennetun i18n-tuen sanastot.
*.hs	Kontrollerien käyttämä yleinen logiikka ja tarvittavat resurssit.

2.6 Järjestelmän komponentit

app/main.hs

Itse suoritettavan ohjelman lähdetiedosto (`main`-funktio).

Application.hs, Settings*, Import.hs, Model.hs

Kehityksessä harvemmin koskemista vaativia tiedostoja, joissa toteutetaan (Yesodille) yleistä kontrollilogiikkaa.

CalendarTypes.hs

Kalentereihin ja kohteisiin liittyvät tyypit.

CalendarQueries.hs

Järjestelmän tärkeimmät tietokantakyselyt sekä liittyvä autentikointi.

Foundation.hs

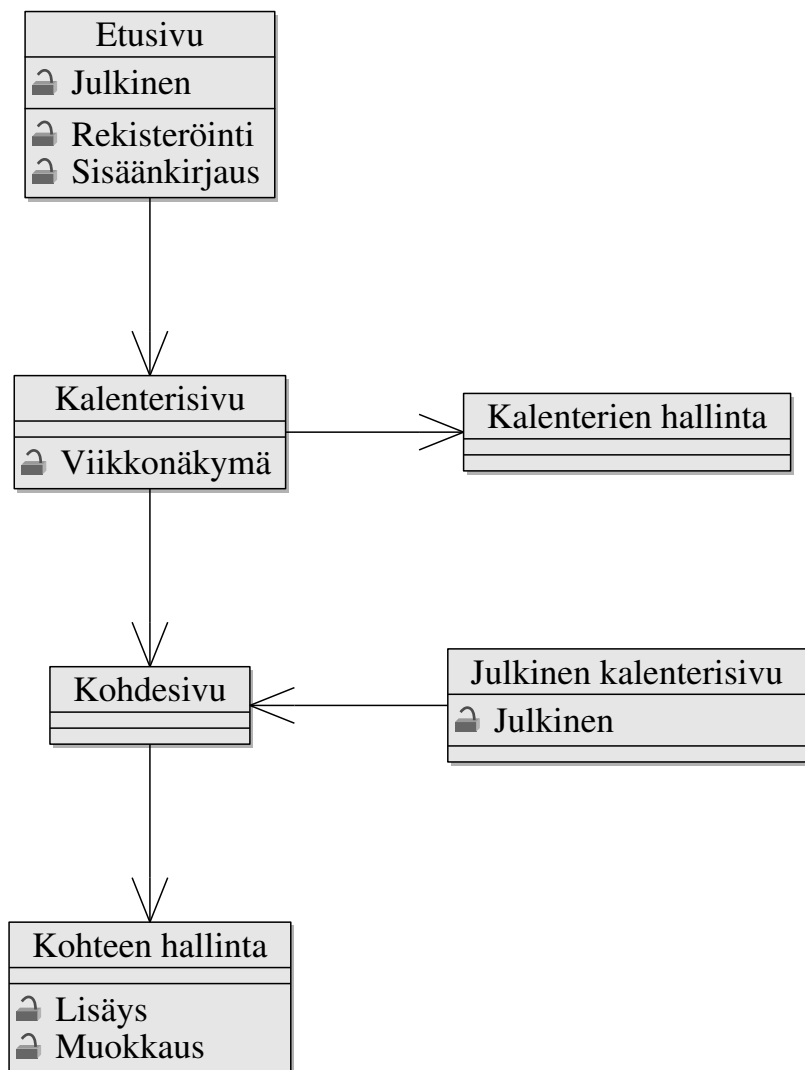
Järjestelmän keskus. Sisältää mm. autentikointinnin ja navigaatiota.

mustached-octo-happiness.cabal

Asennusohjetiedosto *Cabalille*.

2.7 Käyttöliittymä

Käyttöliittymä on kuvattu kaaviossa 2.5. Käyttöliittymässä on yleissiirtymänavigaatio, ellei olla etusivulla, kirjautumissivulla tai julkisella kalenterisivulla.



Kuva 2.5: Käyttöliittymäkaavio. Kaikilta ei-julkisilta sivuilta pääsee siirtymään kalenterinäkömään, kalenteriasetuksiin sekä kirjautumaan ulos navigaatiopalkista.

Luku 3

Käyttö

3.1 Asennustiedot

3.1.1 Esivaatimukset

Järjestelmän asentamiseen vaaditaan toimiva ja ajantasainen Haskell-ympäristö, tai ainakin GHC:n versio 7.4.2 tai uudempi. Järjestelmä on pakattu yhdeksi *cabal*-paketiksi. Sen asentaminen lähdekoodista on siis kätevintä (uudehkolla) *cabal-install*-ohjelmalla.

Hae lähdekoodi:

```
git clone http://github.com/SimSaladin/mustached-octo-happiness
cd mustached-octo-happiness
```

Huom. Varmista, että `~/.cabal/bin` (tai `.cabal-sandbox/bin`, jos työskentelet hiekkalaatikossa) on `PATH`:issa.

3.1.2 Tietokanta

1. Asenna ja konfiguroi PostgreSQL. Järjestelmä luo taulut itse, mutta ei tietokantaa.
2. Luo järjestelmälle käyttäjä ja tietokanta.
3. Aseta tietokantaparametrit tiedostoon `config/postgresql.yml`.

3.1.3 Kehityspalvelin

Ensimmäinen asennus kestää pitkään useiden riippuvuuksien vuoksi (nopealla koneella 15min, hitaammalla vielä kauemmin):

```
cabal install
```


Tietyt binäärit pitää asentaa käsin ennen ylemmää asennettaessa puhtaalla systeemillä (alex?, happy?) ja ikääntyneemmällä cabalilla. Nämä tulee kuitenkin selkeästi esille riippuvuuksia asennettaessa ylemmällä.

3.1.4 Kehitysympäristö

Kehityspalvelimen käynnistys:

```
yesod devel
```

Nyt järjestelmä kuuntelee osoitteessa `http://localhost:3000`, jos kaikki meni oikein.

Huom. Nopeaa katsausta järjestelmän koodikannasta kaipaaville haddock-dokkarit löytyvät osoitteesta `http://animu.paivola.fi/dump/mustached-octo-happiness`.

Huom. tiedostossa `config/settings.yml` voi asettaa applikaation yleisiä asetuksia, esimerkiksi vaihtaa kuuntelevaa porttia.

3.1.5 Tuotantoasennus

Tuotantoasennusta varten voi tutustua skriptiin `script/deploy.sh`. Lyhyesti, käänetään cabal-paketti production-lipalla ja kopioidaan binääri ja staattiset tiedostot tuotantopalvelimelle. Suosittu tapa ajaa tällaisia applikaatioita tuotannossa on käänteisvälityspalvelimen (reverse proxy) avulla esimerkiksi Nginx-instanssilta.

3.2 Käyttöohje

Allekirjoittaneen julkinen instanssi järjestelmästä löytyy toistaiseksi osoitteesta `http://mustached-octo-happiness.functor.paivola.fi`. Sitä voi käyttää vapaasti; rekisteröityminen tapahtuu suoraan käyttäjätunnus/salasana-parilla, tai jollain sivulla näkyvillä muilla metodeilla.

Rekisteröitymisen ja ensikirjautumisen yhteydessä sinua pyydetään luomaan ensimmäinen kalenteri. Ensimmäisen kalenterin luomisen jälkeen pääset yleisnäkymään, josta näet ja voit operoida kalentereitasi.

Luku 4

Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Autentikoinnista ja autorisoinnista

Autentikaoidessa BrowserId, Gmail jne. failaavat, jos hashdb:llä joku on rekisteröinyt saman postin nimimerkiksi. Myöskään kyseisiä vaihtokirjautumisia ei ole testattu; ne on konfiguroitu aikaisemman kokemuksen ja näppituntuman avulla.

Tietokantakyselyistä

Kaikissa tietokantaoperaatioissa ei taideta tarkastaa kyselijän oikeutta operaatioon. Jotkin operaatiot kalentereihin ja kohteisiin saattavat olla mahdollista käyttäjille, jotka eivät tarkasti ottaen olisi niihin oikeutettuja. Esimerkiksi joitain poisto-operaatioita ei merkata, joten jos käyttäjä arvaa jonkin kalenterin tai kohteen tunnisteiden, voi hän poistaa sen triviaalilla POST-kyselyllä oikeaan URL:iin.

Kosmetiikka

Viikkonäkymässä ensimmäinen päivä korostetaan taka-ajatuksena merkata tämänhetkistä päivää. Ensimmäinen päivä viikkonäkymässä ei kuitenkaan ole senhetkinen päivä, jos katsotaan tulevia tai menneitä tapahtumia.

Kalenterilistauksessa on listataan kaikkien kohteiden lukumäärä kalentereista. Pitäisikö menneet kohteet vain unohtaa tästä?

Yleistä testauksesta

Applikaatiota ei ole ollut aikaa testata ohjelmallisesti. Kuitenkin, syötteet, tietokantakyselyt ja -mallit, reititys jne. on toteutettu laajasti testatuilla ja katastrofaalisen

toiminnan estävillä työkaluilla, joten applikaation voi teorioiltaan mieltää suurimmilta osin oikein toimivaksi. Toisaalta applikaatiota on testattu paitsi kehittäjän puolesta, myös keskivertoa uutta käyttäjää simuloiva on jakanut mielipiteitä ja huomioita. Testattavaa ja korjattavaa pitäisi lisätä ainakin seuraaviin:

Lomakkeiden arvojen järjellisyys Esimerkiksi tekstikentät voivat olla lähes äärettömän monta merkkiä pitkiä.

CSS ei liene yhteensopiva kovinkaan monen selaimen kanssa. Lisäksi käyttöliittymän toiminnassa on ei-toivottoja ominaisuuksia, kuten kalenterin skrollaaminen pienellä näytöllä sivuittain, mikä ei toimi.

SQL Kyselyissä käytännössä ainoa testauskohde on kyselyiden haluttu haluttu toiminta eikä kyselyn toimiminen. Esimerkiksi usean taulun liitosten oikeellisuus tilanteeseen pitäisi varmistaa.

GET-Parametrit Vaikka kaikki reititys onkin kunnossa, niin GET-parametrien toiminta pitäisi varmistaa esimerkiksi viikkonavigaatioissa.

Ideoita

- Ryhmähallinta
- Erilaiset tuonti- vienti- ja synkronointimahdollisuudet: .ics, caldav, google calendar...
- Erilaiset näkymät, kuten kuukausi-, päivä- ja kohdenäkymät.
- CSS-tyylit ja kotinäkymän markup vaativat uudelleenkatsauksen, jos järjestelmän visuaalista ilmettä lähtisi kehittämään.
- Kohteiden alikäytetyille kentille (tärkeys, hälytys, osallistujat jne.) voisi keksiä jotain käteviä käyttötapauksia.
- Kohteille metadata, joka unohtui lisätä heti alussa
- Tuki kryptatulle sisällölle (CryptJS)