

Happeninki

Sovellus dokumentti

Tekijä	Jari Lybeck
Tarkastajat	
Hyväksyjät	
Version	0.5

MUUTOS HISTORIA

Versio	Pvm	Muokkaaaja	Muutoksen kuvaus
0.1	01.11.2015	Jari Lybeck	Ensimmäinen versio
0.2	08.11.2015	Jari Lybeck	Järjestelmän tietosisältö kappale lisätty
0.3	08.11.2015	Jari Lybeck	Relaatiotietokantakaavio lisätty
0.4	22.11.2015	Jari Lybeck	Alustava käyttöohje lisätty
0.5	18.12.2015	Jari Lybeck	Dokumentaatiot päivitetty lopullista palautusta varten

CONTENTS

1	JOHDANTO	4
2	YLEISKUVA JÄRJESTELMÄSTÄ	5
2.1	KÄYTTÖTAPAAVIO.....	5
2.2	KÄYTTÄJÄRYHMÄT	5
2.3	KÄYTTÖTAPAUKSET	6
2.3.1	<i>Järjestäjän käyttötapaukset.....</i>	<i>6</i>
2.3.2	<i>Osallistujan käyttötapaukset.....</i>	<i>6</i>
3	JÄRJESTELMÄN TIETOSISÄLTÖ	7
4	RELAATITOKANTAKAAVIO	10
5	JÄRJESTELMÄN YLEISRAKENNE.....	11
6	KÄYTTÖLIITTYMÄ KOMPONENTIT	12
7	ASENNUSOHJE	12
8	KÄYTTÖOHJE.....	13
9	ESITTELYTEKSTI	13

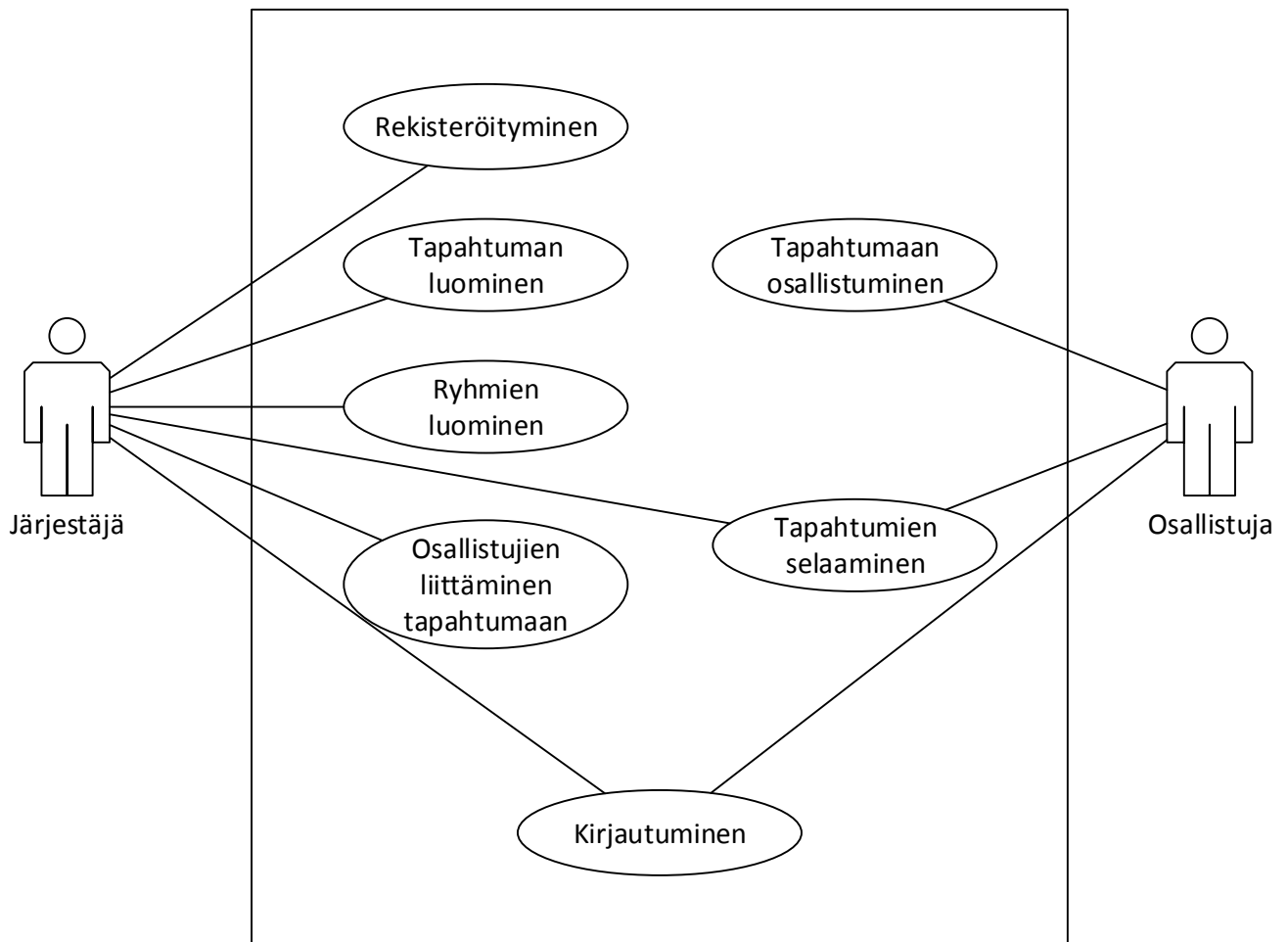
1 Johdanto

Happeninki on web-pohjainen sovellus, jolla voidaan luoda ja hallita tapahtumia sekä rekisteröidä osallistujia niihin. Sovelluksessa toteutetaan muutama yksinkertainen hallintanäyttö sekä näyttö jolla voidaan liittää henkilöt tapahtumaan. Henkilöt liitetään tapahtumaan sähköpostiosoitteella, jolloin kyseiselle henkilölle lähtee sähköposti viesti ja linkki tapahtumaan mihin hänet on liitetty. Tällöin henkilö voi linkin takaa löytyvän näytön kautta ilmoittaa osallistuuko tapahtumaan vai ei.

Sovellus on tarkoitus toteuttaa Heroku pilvipalveluun ja tekniikoina käytetään Java ohjelmointikieltä sovelluslogiikan hallitsemiseen sekä HTML ja JavaScript kieliä käyttöliittymän toteutukseen. Tietokantana käytetään PostgreSQL tietokantaa, joka myös toimii Heroku alustalla. Erinäisiä JavaScript kirjastoja saatetaan käyttää sovelluksen toteutuksessa esim. JQuery.

2 Yleiskuva järjestelmästä

2.1 Käyttötapauskavio



2.2 Käyttäjärühmät

Järjestäjä

Tapahtuman järjestäjä on henkilö, joka rekisteröityy ja luo tapahtumia Happeninki sovellukseen. Hän myös kutsuu osallistujat tapahtumaan.

Osallistuja

Tapahtuman osallistuja on henkilö, jolle on lähetetty kutsu tapahtuman järjestäjän toimesta. Hänen ei tarvitse rekisteröityä järjestelmään.

2.3 Käyttötapaukset

2.3.1 Järjestäjän käyttötapaukset

Tapahtuman luominen:

Järjestäjä luo tapahtuman syöttämällä tapahtuman tarvitsemat perustiedot, kuten tapahtuman nimen, kuvauksen sekä tapahtuman ajankohdan. Tapahtuman voi myös merkitä toistuvaksi esimerkiksi joka viikon keskiviikkona klo 07:30. Järjestäjä voi määritellä muistutus sähköpostien lähetys tiheyden ja ajankohdan, kun tapahtuman ajankohta alkaa lähestyä.

Ryhmiä luominen:

Järjestäjä voi luoda ryhmiä, johon lisätä henkilöitä ja näiden sähköposteja. Esimerkkeinä ryhmistä voisi olla Perhe, Kaverit tai Työkaverit.

Osallistujien liittäminen tapahtumaan:

Järjestäjä voi liittää osallistujia tapahtumaan valitsemalla ensin tapahtuman ja tämän jälkeen lisäämällä tapahtumalle henkilöiden sähköpostiosoitteita. Tämän lisäksi järjestäjä voi myös lisätä luomiansa ryhmiä tapahtumaan.

Tapahtumien selaaminen:

Järjestäjä voi selata tapahtumia, joita on luonut.

Muita käyttötapauksia: Rekisteröityminen, kirjautuminen

2.3.2 Osallistujan käyttötapaukset

Tapahtumaan osallistuminen:

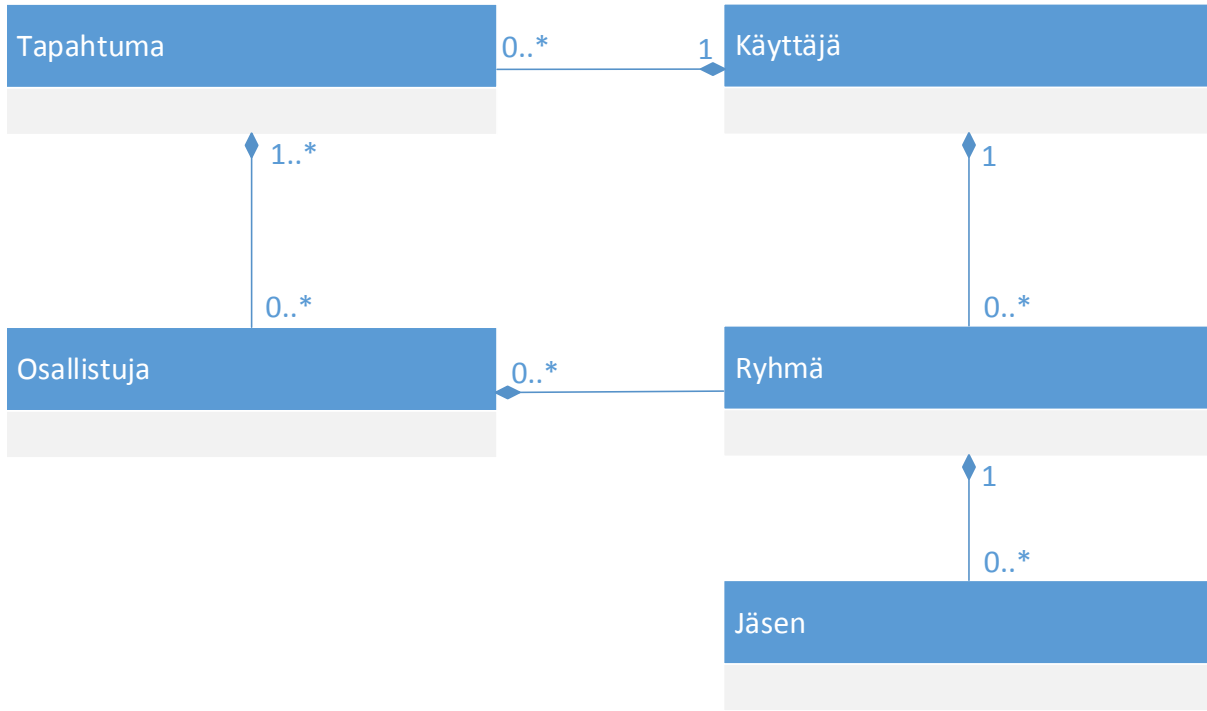
Sähköpostin saatuaan osallistuja voi vahvistaa, että hyväksyykö vai hylkääkö kutsun tapahtumaan sekä lisätä omat kommenttinsa, jotka on kaikkien tapahtumaan kutsuttavien nähtävillä.

Tapahtumien selaaminen:

Osallistuja voi selata tapahtumia, joihin hän on osallistunut.

Muita käyttötapauksia: Kirjautuminen

3 Järjestelmän tietosisältö



Tietokohde: Käyttäjä

<i>Attribuutti</i>	<i>Arvojoukko</i>	<i>Kuvaus</i>
Nimi	Merkkijono, max.100	
Tunnus	Merkkijono, max.100	Käyttäjätunnus järjestelmään kirjautumista varten
Salasana	Merkkijono, max.100	SHA-256 hash salasanasta.
Sähköposti	Merkkijono, max.100	
Varmistettu	Boolean, true / false	Onko käyttäjä varmentanut itsensä.
Muokkaus päivämäärä	Timestamp	Muokkauksen ajanhetki
Viimeksi kirjautunut	Timestamp	

Tietokohde: Ryhmä

<i>Attribuutti</i>	<i>Arvojoukko</i>	<i>Kuvaus</i>
Nimi	Merkkijono, max.100	Ryhmän nimi esim. Kaverit.
Kuvaus	Merkkijono, max.1000	
Muokkaus päivämäärä	Timestamp	Muokkauksen ajanhetki

Tietokohde: Jäsen

<i>Attribuutti</i>	<i>Arvojoukko</i>	<i>Kuvaus</i>
Nimi	Merkkijono, max.100	
Kuvaus	Merkkijono, max.1000	
Sähköposti	Merkkijono, max.100	
Muokkaus päivämäärä	Timestamp	Muokkauksen ajanhetki

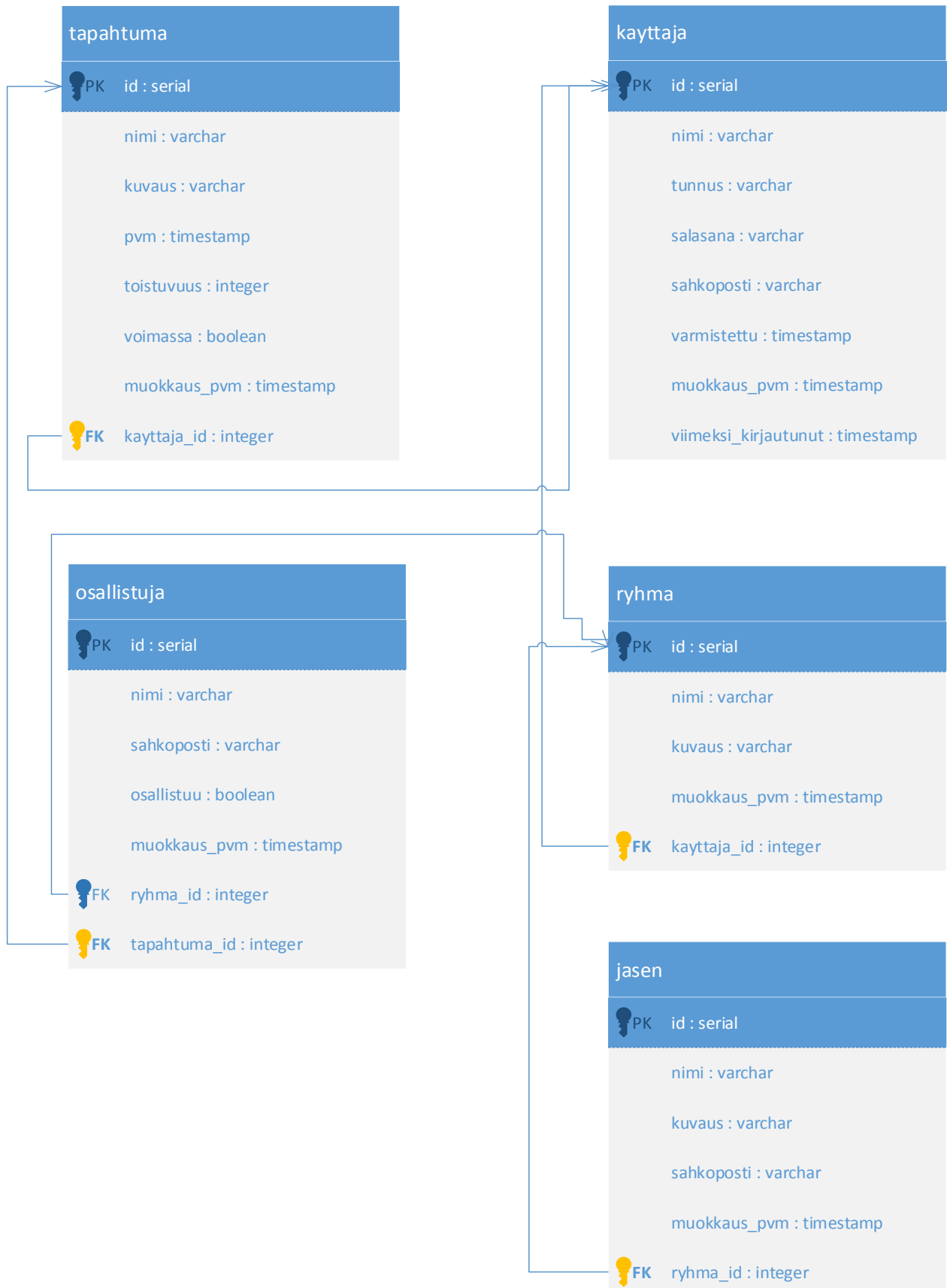
Tietokohde: Tapahtuma

<i>Attribuutti</i>	<i>Arvojoukko</i>	<i>Kuvaus</i>
Nimi	Merkkijono, max.100	
Kuvaus	Merkkijono, max.1000	
Päivämäärä	Timestamp	Tapahtuman ajanhetki
Toistuvuus	Integer	Tapahtuman toistuvuus päivinä. Esim. arvo 7 tarkoittaa 7 päivän välein ja tämä arvo lisätään Päivämäärä attribuuttiin.
Voimassa	Boolean, true / false	Onko tapahtuma aktiivinen?
Muokkaus päivämäärä	Timestamp	Muokkauksen ajanhetki

Tietokohde: Osallistuja

<i>Attribuutti</i>	<i>Arvojoukko</i>	<i>Kuvaus</i>
Nimi	Merkkijono, max.100	
Sähköposti	Merkkijono, max.100	
Osallistuu	Boolean, true / false	Osallistuuko käyttäjä tapahtumaan?
Muokkaus päivämäärä	Timestamp	Muokkauksen ajanhetki

4 Relaattiotietokantakaavio



5 Järjestelmän yleisrakenne

Järjestelmän ohjelmakoodi sijaitsee **code/Happeninki** hakemistossa. Ohjelmakoodin hakemistorakenne noudattaa Maven projektin rakennetta jolloin käyttöliittymä sekä ohjauslogiikka löytyy **src/main/webapp** hakemiston alta. JAVA ohjelma koodi löytyy **src/main/java** hakemiston alta.

Ohjelma on pyritty tekemään MVC mallin mukaisesti, jolloin Model osio löytyy **src/main/java/fi/soha/model** hakemistosta. View:tä vastaa **src/main/webapp/*.jsp** sivut sekä **src/main/webapp/js** koodi. Controller on ohjelmoitu yhteen JSP sivuun jonka nimi on **kontrolleri.jsp**.

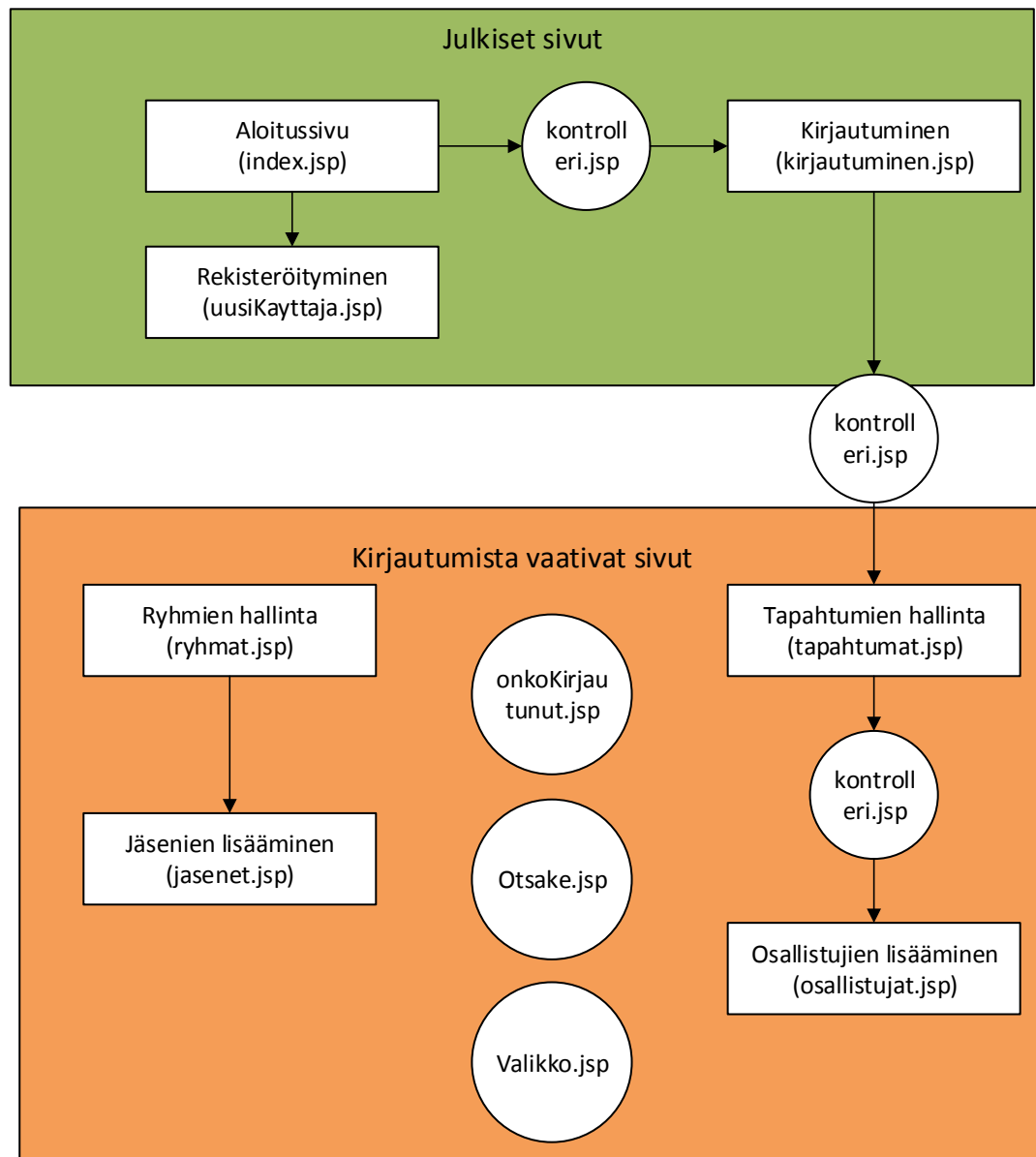
JAVA koodi on pyritty pitämään suhteellisen yksinkertaisena ja periaatteena on ollut pitää tietokanta haut yhdessä luokassa: **HappeninkiDAO.java** tiedostossa. **HappeninkiService.java** luokan tarkoituksena on ollut piilottaa kutsut DAO luokkaan ja myöskin tätä kautta luodaan model oliot DAO luokan metodeja varten. Service luokan metodeja kutsutaan luonti, päivitys ja poisto tilanteissa kontrolleri.jsp sivulla ja haku metodeja kutsutaan käyttöliittymä JSP sivuilta suoraan.

Käyttöliittymän toteutuksessa on käytetty apuna seuraavia JavaScript kirjastoja:

- JQuery
- JQuery-ui
- JQuery-mmenu
- JQuery-validate

Kirjautuminen tapahtuu HappeninkiService.kirjaudu metodin kautta, jolloin sovellus käy tarkistamassa syötetyn salasanan SHA-256 HASHia tietokantaan tallennettuun hashiin. Jos tällainen löytyy tallennetaan tiedot käyttäjästä HttpSession attribuuttiin "kayttaja". onkoKirjautunut.jsp sivu on lisätty jokaisen käyttöliittymän JSP sivun alkuun, joka käy tutkimassa löytyykö tällainen sessiomuuttuja ja sen mukaan päästää henkilön sivulle.

6 Käyttöliittymä komponentit



Navigaatio palkista (vasen yläkulma), pääsee tapahtumat.jsp ja ryhmat.jsp sivuille suoraan.

7 Asennusohje

Sovellus on suunniteltu asennettavaksi Heroku ympäristöön, jossa on provisioitu PostgreSQL tietokanta add-on. PostgreSQL ympäristöön pääsee esimerkiksi SQLDeveloper työkalulla, mutta tämän asennus ja stepit ovat tämän dokumentin ulkopuolella.

1. Luo Heroku tili ja asenna heroku toolbet koneellesi
2. Luo hakemisto, johon haluat asentaa sovelluksen

3. Mene hakemistoon käyttöjärjestelmän shellillä
4. `git clone` <https://github.com/jalybeck/Happeninki>
5. `cd code/Happeninki`
6. heroku login
7. heroku create
8. `git commit -am "Initial commit"`
9. `git push heroku master`
10. heroku open

Aja tietokannan luontilauseet(code/database) Herokussa sijaitsevaan PostgreSQL tietokantaan käyttäen esim. SQLDeveloper työkalua.

8 Käyttöohje

Sovelluksen URL on <https://happeninki.herokuapp.com>.

Kirjautuminen valmiilla tunnuksilla: Testi / Testi1234

Uuden käyttäjän luonti tapahtuu linkistä Uusi käyttäjä.

Kun olet luonut käyttäjän voit kirjautua järjestelmään, jolloin sinut ohjataan Tapahtumien hallinta sivulle, jossa voit syöttää tapahtumia.

Olemassa olevan tapahtuman voi päivittää klikkaamalla tapahtuma riviä, jolloin se värjäytyy vihreällä.

Vasemmassa yläkulmassa on navigointi menu nappi, josta pääsee siirtymään tapahtumat sivulle sekä ryhmien luonti sivulle.

9 Esittelyteksti

Ajattelin ihan kurssin alussa toteuttaa sovelluksen, joka olisi hyödyttänyt meidän työporukkaa sähköpostien ylläpidossa. Tiesin nimenhuuto.comin, joten ajattelin, että olisi helppoa toteuttaa samankaltainen, mutta hieman eri flavourilla oleva palvelu.

Java on kieli, jota olen koodannut pitkään sekä Tomcat on myös minulle hyvin tuttu, joten oli luontevaa lähteä hakemaan JSP/JAVA ratkaisua. En ollut koskaan tehnyt mitään isompaa pilvipalveluiden päälle, kuten Heroku, joten asia tuntui mielenkiintoiselta. Herokun opettelu oli yllättävänkin helppoa ja tutoriaalit / dokumentaatio oli sen verran hyvin toteutettu, että sain Heroku palvelun pystyyn PostgreSQL tietokannalla muutamassa tunnissa.

Tietokanta toteutus oli mielestäni kaikista helpoin asia tässä kurssissa. Haastavinta oli käyttöliittymän tekeminen, koska en ole koskaan ollut siinä kovin vahva. Myöskin eri javascript frameworkkien käyttöönotto tuntui jokseenkin haastavalta ja luin melko paljon eri frameworkkien heikkouksista ja vahvuuksista.

Mielestäni kurssi oli kaiken kaikkiaan erittäin mielenkiintoinen sen monipuolisuuden takia. Vaikka kurssin nimi onkin Tietokantojen Harjoitustyö, niin opin paljon HTML5/javascript/css koodausta sekä pilvipalveluista.