

Teemu Järvisalo

Ideawall-sovelluksen toteutus ja dokumentaatio

Opinnäytetyö
Tietotekniikan koulutusohjelma

2017



**Kaakkois-Suomen
ammattikorkeakoulu**

Tekijä/Tekijät Teemu Järvisalo	Tutkinto Insinööri (AMK)	Aika Elokuu 2017
Opinnäytetyön nimi		44 sivua
Ideawall-sovelluksen toteutus ja dokumentaatio		56 liitesivua
Toimeksiantaja		
Kaakkois-Suomen Ammattikorkeakoulu		
Ohjaaja		
Lehtori Niina Salmi		
Tiivistelmä		
<p>Tässä opinnäytetyössä käsitellään Ideawall-sovelluksen suunnittelu ja toteutus. Opinnäytetyö toimii myös dokumentaationa sovelluksen ylläpidolle ja mahdolliselle jatkokehitykselle. Sovellus kehitettiin pelillistetyksi tehtävälustaksi STARPABS-hankkeen MOOC opintokurssille, johon voi osallistua opiskelijoita mistä maasta tahansa.</p> <p>Ideawall-sovellus on verkkoselaimella toimiva tehtävälusta, jossa opiskelijat luovat startup-ideoita. Nämä ideat toimivat samalla sovelluksen ryhminä. Sovelluksen tarkoituksena on, että opiskelijat löytävät omaa osaamistaan vastaavia ryhmiä ja ryhmät löytävät niille sopivia opiskelijoita. Löytämistä helpottavat sovellukseen syötettävät tiedot taidoista ja kiinnostuksen kohteista. Näillä kriteereillä sovellus automaattisesti suosittelee sekä opiskelijoita ryhmille että ryhmiä opiskelijoille. Sovelluksessa muodostuneet ryhmät suorittavat opintokurssilla myöhemmin annettavia tehtäviä.</p> <p>Sovellus kehitettiin Kaakkois-Suomen ammattikorkeakoulun Kouvolan kampuksen game design- ja Kotkan kampuksen peliohjelmoinnin-opintolinjojen yhteistyöprojektina. Kouvolan game design -ryhmä suoritti sovelluksen toiminnan ja ulkoasun suunnittelun. Kotkan ryhmä suoritti sovelluksen käytännön toteutuksen. Ideawall-sovellus ohjelmointiin Ruby-ohjelmointikielellä ja käyttää Ruby on Rails -sovelluskehystä. Sovellus käyttää myös useita lisäosia ja laajennuksia sen toiminnallisuuden parannuksiin.</p> <p>Ideawall-sovellukselle sopivan sovelluskehysten valinnalla päästiin projektin lähes kaikkiin tavoitteisiin sovelluksen toiminnasta. Sovellus jäi kuitenkin vielä kesken ja vaatii jatkokehitystä, jotta se soveltuisi kurssin tehtävälustana. Opinnäytetyön dokumentaatio-osassa perehdytään sovelluksen ohjelmakoodin eri toimintoihin ja toteutuksiin, joka helpottaa sovelluksen mahdollista jatkokehitystä.</p>		
Asiasanat		
Ruby, Ruby on Rails, pelillistäminen, verkkosovellus		

Author (authors)	Degree	Time
Teemu Järvisalo	Bachelor of Engineering	June 2017
Thesis Title		44 pages
Implementation and documentation of Ideawall application		56 pages of appendices
Commissioned by	South-Eastern Finland University of Applied Sciences	
Supervisor	Niina Salmi, Senior Lecturer	
Abstract	<p>The objective of this thesis was to implement and document a web application called Ideawall. The documentation will be used for maintaining and possible further development of the application. The web application was designed for STARPABS project as a gamified platform for completing course tasks for a MOOC.</p> <p>Ideawall is a web application where students create startup ideas. The ideas will also operate as groups for the application. The purpose of the application is to assist students with finding a group that is suitable for their skills and interests and also assist the groups to find suitable students which have the required skills. Based on these characteristics, the application then suggests automatically the matches for both to the student and to the group. The groups formed in the application are then used for later tasks in the course.</p> <p>The application was developed as a collaborative project by two groups of students from two campuses of South-Eastern Finland University of Applied Sciences, game design students from the campus of Kouvola and game programming students from the campus of Kotka. The game design group were tasked with the graphical design and designing the functionality of the application. The game programming group were tasked to carry out the practical solution of the application. Ideawall was programmed in Ruby programming language and uses Ruby on Rails framework with several extensions and add-ons.</p> <p>The application succeeded in most of its objectives with the right choice of its framework. However, it remained unfinished and will need further development to be used in the MOOC. The documentation section of the thesis focuses on the code and how it functions in the application to aid in the possible further development.</p>	
Keywords	Ruby, Ruby on Rails, web application, gamification	

SISÄLLYS

LYHENTEET & TERMIT	6
1 JOHDANTO	7
2 RUBY-OHJELMOINTIKIELI.....	8
3 RUBY ON RAILS -SOVELLUSKEHYS.....	9
3.1 MVC-arkkitehtuuri	9
3.2 Action cable -moduuli	10
4 SOVELLUKSEN LISÄOSAT JA LAAJENNUKSET	10
5 SUUNNITTELU.....	12
6 IDEAWALL-SOVELLUS	13
7 SOVELLUKSEN FRONT END	16
7.1 Opiskelijan näkymät ja toiminnot	16
7.1.1 Etusivu	16
7.1.2 Profiilinäkymä.....	21
7.1.3 Sisäänkirjautuminen.....	22
7.1.4 Käyttäjän luominen.....	22
7.1.5 Idean luominen.....	24
7.1.6 Ideanäkymä.....	25
7.1.7 Idean asetukset.....	27
7.2 Ylläpitäjän näkymät.....	28
7.2.1 Näytä tapahtumat	29
7.2.2 Luo ajastin.....	30
7.2.3 Muokkaa ajastimia	30
7.2.4 Käyttäjälista.....	31
7.3 Muita näkymiä.....	32
7.3.1 Alert-ilmoituslaatikko	32
7.3.2 Tooltip-vihjelaatikko.....	34
7.4 Evästeet.....	35
8 IDEAWALL BACKEND	35

8.1	Ohjaimet	36
8.1.1	Registrations controller	37
8.1.2	Application controller	37
8.1.3	Comments controller	37
8.1.4	Events controller	38
8.1.5	Ideas controller.....	38
8.1.6	Invites controller	38
8.1.7	Notifications controller	39
8.1.8	Profiles controller	39
8.1.9	Sessions controller	39
8.2	Tietokanta.....	39
8.3	Ideawall -sovelluksen näyttö Eliademy -verkko-opinnotsivustolla.....	39
9	JATKOKEHITYS.....	40
10	PÄÄTELMÄT JA LOPPUYHTEENVETO.....	41
	LÄHTEET.....	43
	KUVALUETTELO	44
	LIITTEET	
	Liite 1. Ideawall hallinnoijan käyttöohje	
	Liite 2. Ideawall käyttäjäongelmat	
	Liite 3. Ideawall tietokannan rakenne	
	Liite 4. Ideawall Back end -dokumentaatio	

LYHENTEET & TERMIT

AJAX	Lyhenne sanoista Asynchronous JavaScript and XML. Se on kokoelma erilaisia internetteknikkoja, jolla voidaan mm. päivittää dataa palvelimen ja verkkoselaimen välillä ilman verkkosivun kokonaista päivitystä.
Back end	Back end -nimeä käytetään ohjelmointialan terminä, jolla kutsutaan palvelimella suoritettavaa sovelluksen ohjelmakoodia.
DRY	Lyhenne sanoista Don't Repeat Yourself. Kyseessä on ohjelmistokehityksen yleinen ohjelmointimalli. Mallin tarkoituksena on pitää ohjelmakoodi optimoituna.
Front end	Front end -nimeä käytetään ohjelmointialan terminä, jolla kutsutaan käyttäjän verkkoselaimella suoritettavaa ohjelmakoodia.
JavaScript	Verkkoseilain-ympäristössä käytettävä dynaaminen ohjelmointikieli.
MOOC	Lyhenne sanoista Massive Open Online Course. Se on internetissä toteutettu etäopintokurssi, joka on avoin kaikille opiskelijoille ja siihen voi käytännössä liittyä loputon määrä osallistujia.
Sovelluskehys	Sovelluskehys tarkoittaa ohjelmistotuotetta, joka muodostaa rungon sen päälle rakennettavalle sovellukselle. Sovelluskehys tunnetaan paremmin ohjelmointialan terminä nimellä Framework.
Tag	Ideawall-sovelluksessa käytettävä vapaasti kirjoitettava avainsana, jota käytetään sovelluksen hakutoiminnossa.
WebSocket	Kommunikaatioprotokolla, jolla palvelin ja verkkoselain voivat ottaa yhteyden toisiinsa.

1 JOHDANTO

STARPABS tarkoittaa intohimoista startup-toiminnan kehittämistä Itämeren alueella. Kaakkois-Suomen ammattikorkeakoulu, Kotkan-Haminan seudun kehittämisyhtiö Cursor Oy, Tallinnan yliopiston ja Stockholm School of Economics Rigan konsortio sai nelivuotisen hankkeen EU:n keskeisen Itämeren alueen ohjelmasta.

Hankkeen päätavoite on synnyttää reilu kymmenen ylikansallisesti omistettua tietointensiivistä, opiskelijalähtöistä startup-yritystä.

Hanke perustuu siihen, että suomalaisilla, virolaisilla ja latvialaisilla on toisiaan täydentävää osaamista ja kulttuurista pääomaa (Lindeman 2015).

Ideawall-sovellus sai alkunsa STARPABS-hankkeesta, jonka yhtenä tavoitteena oli kehittää vuorovaikutteinen MOOC. Tähän vuorovaikutteiseen kurssiin luotiin pelillistetty, verkkoselaimella toimiva tehtävälusta nimeltään Ideawall.

Ideawall-sovellus rakennettiin Xamkin Kouvolan game design -opiskelijoista koostuvan ryhmän ja Xamkin Kotkan peliohjelmoinnin opiskelijoista koostuvan ryhmän kanssa. Xamkin Kouvolan ryhmä suoritti sovelluksen suunnittelun ja grafiikan. Ryhmään kuuluivat Ksenia Shanina, Sanja Kunttu ja Niina Sirparanta. Käytännön toteutuksen suoritti Xamkin Kotkan ryhmä, johon kuului Teemu Järvisalo ja Mikko Raussi. Lisäksi projektin ohjauksesta ja etenemisestä vastasivat Suvi Pylvänen, Niina Salmi, Sabine Suorsa ja Marianne Roslund. Kaakkois-Suomen ammattikorkeakoulun tietohallinto asetti Ideawallin omalle palvelimelleen.

Opinnäytetyön tavoitteena oli toteuttaa Ideawall-sovellus ja dokumentoida kehitystyö jatkokehitystä ja ylläpitoa varten. Ylläpitoa varten opinnäytetyöhön liitettäisiin ohjeet sovelluksen käytöstä (liite 1). Jatkokehitystä varten liitettäisiin dokumentaatio ohjelmakoodin selventämiseksi (liite 4).

Sovellus ohjelmoitiin kesän 2016 työharjoittelun aikana. Ohjelmoinnissa käytettiin tekstinmuokkausohjelmaa Sublime Text 3, johon oli lisätty HTML5- ja Ruby on Rails -lisäosat. Harjoittelun jälkeen Ideawall -sovellusta jatkokehitetiin opetussuunnitelmaan kuuluvilla projektitunneilla. Sovellusta on kaikkiaan työstetty 8 kuukauden ajan.

2 RUBY-OHJELMOINTIKIELI

Ruby on ohjelmointikielenä suhteellisen uusi. Sen ensimmäinen julkinen versio julkaistiin vuonna 1995. Rubyn kehittäjä, Yukihiro Matsumoto, kuvailee suunnitelleensa kielensä *inhimilliseksi*. Hän perustelee sen sillä, että Ruby mahdollistaa ohjelmoijan luomaan enemmän käskyjä vähemmällä kirjoitusvalla (Stewart 2001).

Ruby on sekoitus Matsumoton lempiohjelmointikielien parhaista puolista (Perl, Smalltalk, Eiffel, Ada, and Lisp) (About Ruby 2017). Ruby eroaa muista ohjelmointikielistä siten, että se on aidosti olioihin erikoistunut ohjelmointikieli. Useimmissa moderneissa ohjelmointikielissä on lisätty mahdollisuus olio-ohjelmoinnille, mutta niitä ei alun perin ole suunniteltu siihen tarkoitukseen. Ruby on suunniteltu alusta lähtien olio-ohjelmointiin. Kaikki Ruby-ohjelmointikielessä käsitellään olioina. (Matsumoto 2009.)

```
1 5.times { print "We *love* Ruby -- it's outrageous!" }
```

Kuva 1. Esimerkki Ruby-ohjelmakoodista

Esimerkkinä kuvan 1 Ruby-ohjelmakoodi, joka tulostaa tekstin viidesti. Useimmat ohjelmointikieliet käsittelevät numerot aina lukuyksikköinä, mutta Ruby hyödyntää numerot myös olioina, joilla voi olla omia metodeja.

Ruby on tällä hetkellä yksi maailman 10. suosituimmista ohjelmointikielistä. Sen suuri suosio on paljolti Ruby on Rails -sovelluskehiksen ansiota selainpohjaisten sovellusten ohjelmoinnissa ja Ruby on Rails -sovelluskehikseen lisätyistä ohjelmista. Ruby on myös täysin ilmainen, vapaasti muokattavissa ja levitettävissä. (About Ruby 2017.)

3 RUBY ON RAILS -SOVELLUSKEHYS

Ruby on Rails on verkkosivujen ohjelmointiin suunniteltu sovelluskehys, joka hyödyntää Ruby-ohjelmointikieltä. Vuoden 2004 julkaisunsa jälkeen se on yksi tämän päivän suosituimpia sovelluskehysä verkkosivujen alustana. (Hartl 2016.)

3.1 MVC-arkkitehtuuri

Ruby on Rails -sovelluskehysten toiminnan taustalla piilee sen ominainen MVC-arkkitehtuuri. Nimi juontuu sanoista Model, View, Controller, jotka tarkoittavat mallia, näkymää ja ohjainta. Tämän arkkitehtuurin hyötyjä ovat:

- Toimintalogiikan eristäminen käyttöliittymälogiikasta.
- Helpottaa ohjelmakoodin pysymistä DRY-mallin mukaisena.
- Luo selkeän rakenteen mihin mikäkin ohjelmakoodi kuuluu, mikä helpottaa ylläpitoa.

Model-datamalli

Model-datamalli edustaa sovelluksen tietoja Ruby-ohjelmakielen oliona Ruby on Rails -sovelluskehyksessä ja sisältää säännöt, kuinka tietoja manipuloidaan. Model-datamalleja käytetään tietokannan taulujen käsittelyyn. Yhtä taulua tietokannassa kuvaa yksi datamalli sovelluksen sisällä.

Näkymät

Näkymät edustavat sovelluksen käyttöliittymää. Ruby on Rails -sovelluskehyksessä näkymät ovat yleensä HTML-tiedostoja, joihin on sisällytetty Ruby-ohjelmakoodia. Sisällytetyn ohjelmakoodin tehtävänä on datan esittäminen näkymässä. Näkymät hoitavat datan esittämisen verkkoselaimeen tai muuhun ohjelmaan, joka suorittaa kutsun sovellukseen.

Ohjaimet

Ohjaimet suorittavat kaikki mallin ja näkymän välissä tapahtuvat asiat. Ohjainten tehtävä on käsitellä verkkoselaimelta tulevat kutsut, tuoda mallilta tarvittava data ja tuoda haluttu data näkymään esitettäväksi. (Ruby on Rails Guides 2017.)

3.2 Action cable -moduuli

Action Cable -moduuli yhdistää WebSocket-kommunikaatioprotokollan saumattomasti Ruby on Rails -sovelluskehityksen sovellukseen. Se mahdollistaa sovellukseen reaaliaikaisten ominaisuuksien kehittämisen Ruby-ohjelmointikielellä. (Ruby on Rails Guides 2017.)

Ideawall-sovellus hyödyntää tätä moduulia monissa toiminnoissaan, jotka luovat käyttökokemuksesta dynaamisemman. Moduuli helpottaa huomattavasti reaaliaikaisten toimintojen ohjelmoinnin Ruby on Rails -sovelluskehitykselle.

4 SOVELLUKSEN LISÄOSAT JA LAAJENNUKSET

Ideawall-sovellus hyödyntää monissa verkkoselaimiin pohjautuvissa sovelluksissa hyväksi todettuja lisäosia ja laajennuksia. Näiden lisäosien käyttäminen helpotti ja nopeutti Ideawall-sovelluksen kehitystä ja samalla tekivät siitä paremman ja turvallisemman käyttää.

Devise

Ideawall-sovellus hyödyntää Devise-lisäosaa käyttäjien sisään kirjautumiseen ja varmennukseen. Devise on valmis ratkaisu käyttäjän kirjautumisen varmennukseen. Devise-lisäosa on monen vuoden ja lukemattomien iteraatioiden tulos, joka on erittäin suosittu Ruby on Rails -sovelluskehityksen kanssa työskenteleville sovelluskehittäjille. Devise-lisäosaa käytetään niin noviisien tekemien sivujen kuin monien tuhansien käyttäjien sivujen kirjautumisratkaisuna. Devise sisältää 10 moduulia, jotka ovat:

- Database Authenticatable -moduuli, joka sisältää salasanojen kryptauksen tietokantaan.
- Omniauthable-moduuli, joka mahdollistaa kirjautumisen Facebook- tai Google-käyttäjätileillä.
- Confirmable-moduuli, joka toimii sähköpostivarmennuksena uusille tileille.
- Registerable-moduuli, joka on sisäänrakennettu rekisteröintipalvelu käyttäjätileille.
- Rememberable-moduuli, jolla voidaan tallentaa kirjautuminen evästeeksi, jolloin käyttäjän ei tarvitse kirjautua uudelleen selaimen suljetuana.
- Trackable-moduuli, joka kirjaa ylös käyttäjän kirjautumismäärän, ajan ja IP-osoitteen.
- Timeoutable-moduuli, joka kirjaa käyttäjän ulos automaattisesti tietyn ajan kuluttua.
- Validatable-moduuli, jolla voi muokata millainen käyttäjien varmennuksen tulisi olla, esimerkiksi salasanan tietty pituus ja monimutkaisuus.
- Lockable-moduuli, jolla voidaan lukita käyttäjätili, jos väärää kirjautumisia on tullut liikaa. (Devise 2017.)

Omniauth

Omniauth-lisäosa on ohjelmakirjasto, jonka tarkoituksena on mahdollistaa käyttäjien kirjautuminen sisään käyttämällä toisen palveluntarjoajan tunnuksia. Esimerkiksi sisäänkirjautuminen Facebook- tai Google-tiliä käyttäen. Käyttäjän kirjautuessa sisään hänet ohjataan kirjautumaan kunkin palvelun tarjoajan sivuston kautta. Omniauth-kirjautuminen toimii siten, että kunkin palveluntarjoajan sivustolle rekisteröidään *applikaatio*, joka tällöin käyttäjän kirjautuessa antaa takaisin tarvittavat tiedot sovellukselle siitä, että käyttäjä voidaan varmentaa kirjautuneeksi sisään. (Omniauth 2016.)

jQuery

jQuery on JavaScript-ohjelmakirjasto, jonka tehtävänä on yksinkertaistaa JavaScript-ohjelmointikielen kirjoittamista ja mahdollistaa saman JavaScript-ohjelmakoodin toiminta kaikilla alustoilla. (York 2009, 19.)

Bootstrap

Bootstrap on verkkoselainsovelluksen käyttöliittymän suunnittelua helpottava sovelluskehys. Bootstrap koostuu JavaScript-ohjelmakoodista ja CSS-tyyliohjeista ja sisältää useimmat tärkeät työkalut, joilla voidaan kehittää valmis ratkaisu verkkosovelluksen käyttöliittymälle. (Jakobus & Marah 2016, 6.)

Paper.js

Paper.js on HTML5 Canvas -sovelluskehys. Se mahdollistaa helpomman ja tehokkaamman tavan kirjoittaa JavaScript -ohjelmakoodia HTML5 Canvas -ympäristöön. (Paper.js 2017.)

Will_paginate

Will_paginate -lisäosa on ohjelmakirjasto, jonka tarkoituksena on helpottaa datan esittämistä Ruby on Rails -sovelluskehyksessä järjestelemällä ne selataviksi sivuiksi (will_paginate 2017).

5 SUUNNITTELU

Projektin alkuvaiheessa ei tiedetty millainen alusta olisi paras, jossa käyttäjät voisivat interaktiivisesti suorittaa kurssin tehtäviä tavalla, joka muistuttaisi pelin pelaamista. Projektissa päädyttiin ratkaisuun, joka olisi verkkoselaimella käytettävä sovellus. Sovellus tultaisiin räätälöimään juuri tätä tarkoitusta varten, koska samanlaisia olemassa olevia ratkaisuja ei ollut olemassa.

Kouvolan ryhmän tehtävänä oli suunnitella Ideawall-sovelluksen toiminta ja ulkoasu. Suunnitelmista laadittiin yksityiskohtaiset dokumentaatiot, jotta sovelluksen toteuttaminen olisi mahdollisimman helppoa ja väärinymmärryksiltä välttyttäisiin. Ideawall ottaisi vaikutteita monista suosituista verkkoselaimella käytettävistä sovelluksista, jotka antaisivat käyttäjille tutun vaikutelman käyttökokemuksesta.

Kotkan ryhmä sai vapaasti keksiä tällaisen sovelluksen toteutuksen alusta lähtien. Toteutuksessa päädyttiin käyttämään Ruby on Rails -sovelluskehystä.

Ohjelmakoodin kirjoittamiseen käytettiin tekstinmuokkausohjelmaa Sublime Text 3. Versionhallintaan käytettiin TortoiseSVN -versionhallintaohjelmistoa.

Ruby on Rails -sovelluskehys valittiin, koska lähes kaikki Ideawall-sovelluksen toiminnoista oli mahdollista toteuttaa sen mukana tulleilla työkaluilla ja lisäosilla. Lisäksi sovelluskehysten käytön oppiminen oli todella helppoa verkossa suoritettavien kurssien ja kattavan oppimismateriaalin ansiosta.

Suunnittelun vaiheet

Sovelluksen suunnittelu ja toteutus alkoivat samanaikaisesti. Ohjelmoinnin osalta tämä tarkoitti sitä, että sovelluksen rakenne ja tietokanta suunniteltiin sovelluksen sisällöstä riippumattomaksi. Ideawall-sovelluksen taidot ja kategoriat on suunniteltu tietokantaan siten, että niitä voitaisiin muuttaa milloin tahansa sovelluksen kehitysvaiheessa. Taitojen ja kategorioiden määrä ei ole rajoitettu tai ennalta määrätty. (Liite 3.)

Kommunikaation ylläpitäminen oli tärkeää, jotta sovelluksen tavoitteet pysyivät yhtenäisenä ryhmän kaikkien jäsenten kesken. Kotkan ja Kouvolan ryhmien kommunikaatiota pidettiin yllä viikoittaisten Skype-kokousten avulla. Ryhmät myös kokoontuivat kerran kuukaudessa samaan tilaan keskustelemaan sovelluksen etenemisestä. Näiden kokoontumisten avulla voitiin paremmin tehdä päätöksiä siitä, mitä ominaisuuksia ja toimintoja oli mahdollista tehdä ja realistista toteuttaa käytettävissä olevilla resursseilla.

Suunnitelmaa voitiin muuttaa myös helposti ilman, että valmiina olevan sovelluksen rakennetta olisi tarvinnut huomioida. Sovellukselle suoritettiin suunnittelun alkupuolella käyttäjättestaus sovelluksen paperille tulostetulla versiolla, joka mahdollisti käyttäjäryhmän hiomisen.

6 IDEAWALL-SOVELLUS

Ideawall on verkkoselaimella toimiva tehtäväalusta, jonka tarkoitus on saada kurssiin liittyneet opiskelijat luomaan ryhmiä ja työskentelemään toistensa kanssa riippumatta mistä maasta he ovat. Monikansallisen käyttäjäryhmän takia Ideawall-sovellus tehtiin englanninkieliseksi, jotta mahdollisimman moni

käyttäjä ymmärtäisi sen toiminnan. Opiskelijat saavat Ideawall-sovelluksen tehtävän suoritetuksi, kun heillä on lisättynä sovellukseen jokin oma idea tai he kuuluvat toisen opiskelijan tekemään ryhmään.

Ideawall-sovellus sisällytettiin toimimaan Eliademyn kurssisivuille, jotta kaikki kurssiin osallistuneet opiskelijat löytäisivät sovelluksen. Sisällyttämisessä avataan pieni ikkunan verkkosivun sisälle, jossa toinen verkkosivu voidaan esittää. Tätä samaa yleistä tekniikkaa käytetään mm. verkkomainonnassa. (ks. Ideawall-sovelluksen näyttö Eliademy verkko-opinnot -sivustolla).

Ideawall-sovelluksessa annetaan opiskelijalle valmiit työkalut ryhmien muodostamiseen. Opiskelijat valitsevat keksimäänsä ideaan kategorian ja sen jälkeen järjestelmä kysyy, millaisia taitoja tarvittaisiin ryhmätyöskentelyyn. Uutta käyttäjää tehtäessä, sovellus kysyy opiskelijalta, mitkä aiheet häntä voisivat kiinnostaa ja mitä taitoja hänellä on. Opiskelijan valitsemia taitoja ja kiinnostavia kategorioita käytetään tämän jälkeen kriteereinä, joilla sovellukseen kirjautuneet opiskelijat voivat löytää hänen taitojaan tarvitsevan ryhmän. Yhdistävät kriteerit ovat esitettynä kuvassa 2.



Kuva 2. Ideawall-sovelluksen yhdistävät tekijät

Ideawall-sovelluksen kategoriat ja taidot on valittu niin, että opiskelijan olisi helppo löytää joukosta joku hänen ominaisuuksiinsa liittyvä asia. Kategorioihin kuuluvat:

- Culture
- Smart Cities
- Bioeconomy
- Well-being
- Impact business
- Education
- Other.

Idean väri määräytyy sen kategoriansa mukaan, jolloin ideoiden selaaminen on selkeämpää ja ideat ovat helpommin erotettavissa. Kategorian väriä käytetään myös monessa muussa sovelluksen osassa, jossa kategoria on mainittuna tai esitettynä.

Taitojen ensisijaisena tarkoituksena on saada opiskelijat luomaan ryhmiä ideoiden ympärille. Uuden idean ja uuden opiskelijan tulee valita sovelluksen antamalta listalta vähintään kolme sopivaa taitoa. Taidot on valittu sovellukseen monipuolisesti laajoista kokonaisuuksista, jolloin sopivan taidon valitseminen helpottuisi opiskelijalle. Taidot on jaettu listalla viiteen ryhmään ja jokaisessa ryhmässä on 10 taitoa, jolloin taitojen kokonaismäärä on 50 taitoa. Taitojen Ryhmät ovat:

- Health sciences
- Social sciences
- Humanities
- Natural sciences
- Technical sciences.

Ideawall-sovelluksessa on idean ja opiskelijan profiiliinsa valittuihin taitoihin ja kategorioihin perustuva automaattinen suosittelu. Automaattinen suosittelu listaa sopivia ideoita opiskelijoille ja sopivia opiskelijoita idean luojalle, jolloin ryhmien muodostaminen tapahtuisi usein ja olisi helppoa. Opiskelija voi vapaasti kirjoittaa omaan käyttäjäprofiiliin ja omaan ideaansa Tag-hakusanoja, jotka otetaan huomioon sovelluksen hakutoimintoa käytettäessä. Opiskelija voi käyttää sovelluksen hakutoimintoa apuna, jos automaattisesti suositeltuja ideoita tai ideaan suositeltuja opiskelijoita ei valmiiksi löydy.

7 SOVELLUKSEN FRONT END

Sovelluksen käyttöliittymäpuolta kutsutaan nimellä Front end. Sen kautta käyttäjä suorittaa kaikki toiminnot sovellukselle verkkoselaimen avulla. Ideawall-sovelluksen Front end käsittää opiskelijan ja ylläpitäjän näkymät ja toiminnot. Front end hyödyntää HTML5-merkintäkieltä, CSS-tyyliohjeita ja JavaScript-ohjelmointikieltä, kuvatiedostoja, äänitiedostoja ja fontteja, jotka sovelluksen palvelin lähettää käyttäjän verkkoselaimeen.

7.1 Opiskelijan näkymät ja toiminnot

Opiskelijan tehdessä käyttäjätilin Ideawall-sovellukseen hänellä on mahdollisuus tehdä monia toimintoja monissa eri näkymissä. Opiskelijan näkymät ja toiminnot on suunniteltu siten, että sovellusta on helppoa ja vaivatonta käyttää.

7.1.1 Etusivu

Etusivu on se näkymä, joka Ideawall-sovelluksesta latautuu käyttäjälle ensimmäisenä (kuva 3). Opiskelijalla on mahdollisuus suorittaa etusivun toiminnot, jotka ovat:

- ideoiden selaus
- hakutoiminto
- pallonäkymän toiminnot
- listanäkymän toiminnot
- sivupalkin toiminnot
- ideoiden suositus
- nähdä ilmoitukset.



Kuva 3. Kuva Ideawall-sovelluksen etusivusta

Ideoiden selaus

Ideawall-sovelluksen etusivulla näytetään kaikki muiden käyttäjien lisäämät ideat. Ideat ovat oletuksena järjestettyinä niiden lisäysajankohdan mukaan. Ideoiden listaukseen käytetään Will_paginate-lisäosaa, joka jakaa ideat usealle välilehdelle. Ideat ovat listattuna siten, että yhdellä välilehdellä näytetään pallonäkymässä enintään 10 ideaa ja listanäkymässä enintään 25 ideaa. Ideoiden tietoja päivitetään etusivulla reaaliajassa käyttäen Ruby on Rails -sovellyshäytöksen Action Cable -ominaisuutta.

Hakutoiminto

Hakupalkki ilmestyy etusivulle käyttäjän sisäänkirjautumisen jälkeen. Hakuominaisuudella voidaan hakupalkkiin kirjoittamalla etsiä ideoita ja käyttäjiä. Hakukriteereinä käytetään idean tai käyttäjän nimeä ja Tag-hakusanaa. Käyttäjä pääsee katsomaan sille tarkoitettua näkymää valitsemalla haetun linkin.

Pallonäkymä

Ideawall-sovelluksen pallonäkymä käyttää pallojen piirtämiseen HTML5 Canvas -vektoriympäristöä ja Paper.js-ohjelmakirjastoa (kuva 3). Palloissa esitetään idean kategoria sen reunojen värillä. Pallojen koko on riippuvainen siitä, kuinka monta tykkäystä ja kommenttia se on saanut. Pallot on otsikoitu idean nimen mukaan ja otsikon alle on listattuna ideaan liitetyt Tag-hakusanat. Tag-hakusanojen alla on esitetty ryhmän koko.

Listanäkymä

Listanäkymässä idean tiedot ovat esitettyä yksityiskohtaisemmin. Tietoja ovat: idean luoja, tykkäysten ja kommenttien määrä numeroina, lyhyt kuvaus ideasta. Listanäkymä on esitettyä kuvassa 4.

The screenshot shows the Ideawall application interface. At the top, there is a blue header with a lightbulb icon and the text "Ideawall". Below the header, there is a navigation menu on the left with the following items: "filter", "All ideas", "Newest ideas", "Top Rated ideas", "idea categories", "Culture", "Smart Cities", "Bioeconomy", "Well-being", "Impact business", "Education", and "Other".

The main content area displays several idea cards. The first card is titled "MOBILE APP FOR GET-TOGETHERS" with tags "Well-being" and "funtimes". It is by Mikko Kiiski and has 0 likes and 1 comment. The description is: "This would be a very fast way to know when your friends are up for doing something (specific) instead of having a group discussion of schedules and trying to reach people who might be busy. Their free time and activity interest would be seen clearly in the app". The date is 23 February 2017.

The second card is titled "ECO HOUSES" with tags "Smart Cities", "eco", "green", and "city". It is by mantti 123 and has 0 likes and 1 comment. The description is: "Basically city streets are narrow and there is too much traffic to construct. Old houses have little to nothing isolation for heat. I've have read add on isolation and solar panels are developing all the time, so add those to the old apartment buildings and you have eco friendly old houses in the cities. It's is expensive I know :(". The date is 23 February 2017.

The third card is titled "MOBILE APP FOR CATCHING A BUS IN KOTKA" with tags "Smart Cities", "transportation", and "technology". It is by Juha Salmi and has 0 likes and 1 comment. The description is: "Saves time, reliefs stress and less "late from the class excuses" because of the bus.". The date is 22 February 2017.

The fourth card is titled "CELEBRATION FOR TRADITIONAL FOOD" with tags "Culture", "traditonal food", "frendship", and "culture". It is by exchange student and has 0 likes and 1 comment.

Kuva 4. Kuva Ideawall-sovelluksen listanäkymästä

Sivupalkki

Sivupalkkiin on lisätty käyttäjälle hyödyllisiä toimintoja ideoiden selaamiseen. Sivupalkin toiminnot on jaettu kahteen ryhmään: toiminnot ja kategoriat. Toiminnoista voi valita Ideawall-sovelluksen listaamaan ideoita seuraavasti:

- All ideas
- Newest ideas
- Top Rated ideas.

All ideas -toiminto toimii oletuksena ja listaa vain kaikki sovellukseen lisätyt ideat. Newest ideas -toiminto listaa ideat lisäysajankohdan mukaan. Top Rated ideas -toiminto listaa ideat sen mukaan, kuinka monta kommenttia ja tykkäystä ne ovat saaneet.

Toiminnon valinnan jälkeen käyttäjä voi valita, mitä ideoita näytetään toiminnon valitsemilla kriteereillä haluamistaan kategorioista. Kategorioita voi valita yhden tai useampia kerrallaan.

Käyttäjä voi halutessaan piilottaa sivupalkin valitsemalla sivupalkin laidalla olevan nuolen. Sivupalkin näkymä tallennetaan evästeeksi, jotta käyttäjän tullessa seuraavan kerran sivulle, sivupalkki pysyy edelleen piilotettuna.

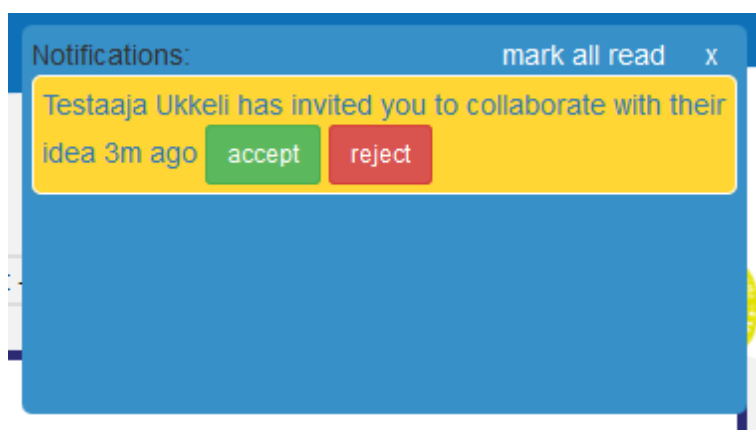
Idean suositus

Idean suositus -näkymään pääsee valitsemalla Recommended-painikkeen etusivulta. Idean suositus -näkyvässä käyttäjälle näytetään etusivulle suosituskriteereihin kuuluvat ideat. Kriteereinä toimivat käyttäjän profiiliinsa lisäämät taidot ja kiinnostavat kategoriat. Suositeltu idea listautuu tähän näkymään, jos joku näistä kriteereistä täsmää ideaan, joka etsii samoja taitoja tai kuuluu samaan kategoriaan.

Ilmoitukset

Käyttäjälle tulevat ilmoitukset näkyvät napsauttamalla kirjekuorta esittävää kuvaketta. Kuvassa 5 esitetystä näkyvässä listautuvat kaikki käyttäjälle tulleet

ilmoitukset, joita sovellus on lähettänyt käyttäjälle. Ilmoitukset päivittyvät reaaliajassa käyttäen Ruby on Rails -sovelluskehityksen Action Cable -ominaisuutta. Käyttäjä näkee uusien ilmoitusten määrän kirjekuori-kuvakkeelle ilmestyvän numeron avulla. Uuden ilmoituksen saapuessa käyttäjä myös saa verkkoselaimeensa Alert-ilmoituslaatikon (ks. Muita näkymiä). Saapunut ilmoitus häviää listalta, kun käyttäjä on valinnut sen tai "mark all read" -painikkeen, joka tyhjentää kaikki listatut ilmoitukset.



Kuva 5. Kuva käyttäjälle tulleesta kutsusta

Käyttäjä saa ilmoituksia seuraavista toiminnoista:

- käyttäjä on kommentoinut ideaasi
- käyttäjä on tykännyt ideaasi
- käyttäjä on poistunut ryhmästäsi
- idean luoja on poistanut sinut ryhmästä
- käyttäjä on hylännyt kutsusi
- käyttäjä on hyväksynyt kutsusi.

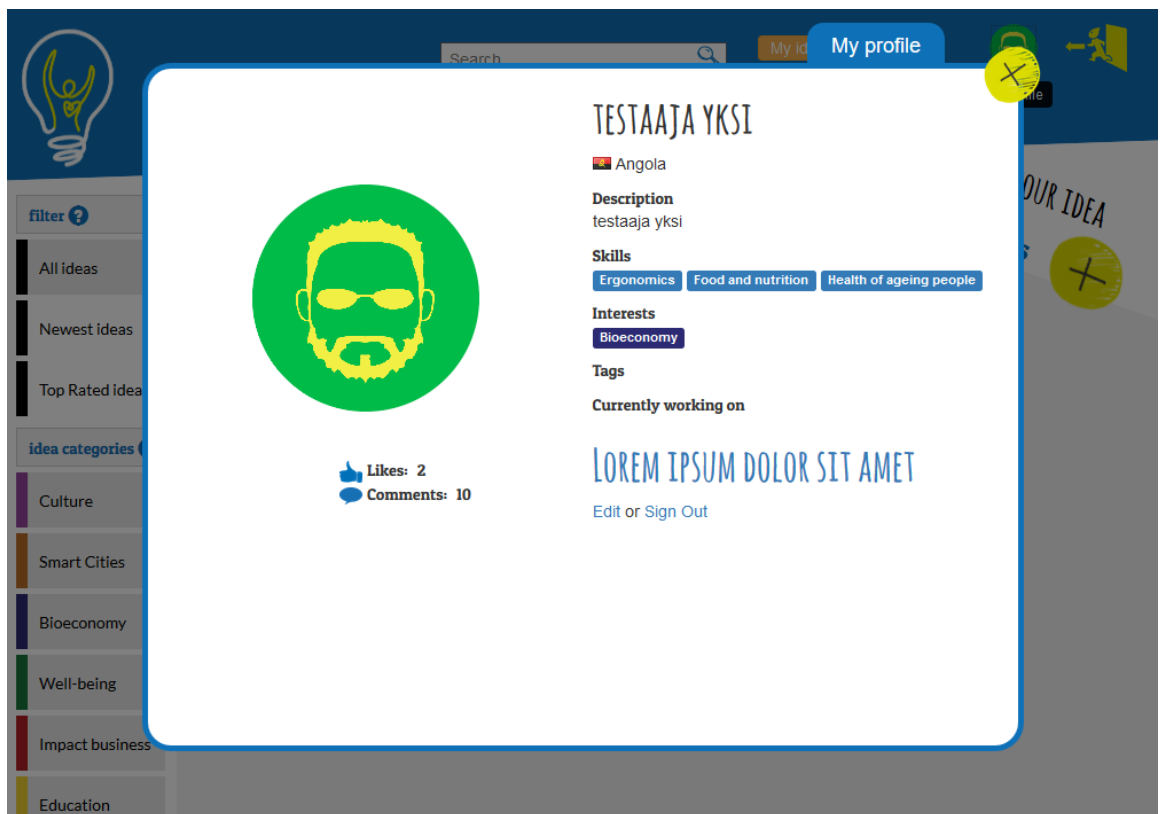
Jokaisella käyttäjällä on Action Cable -ominaisuudelle tarkoitettu kuuntelukanava nimeltä NotificationsChannel, jonka kautta ilmoitukset saapuvat käyttäjälle. Käyttäjän kirjautuessa sisään, jokaiselle käyttäjälle määritellään oma kuuntelukanava sovelluksen palvelimella. Sovelluksen Front end -puolella käyttäjän verkkoselain tilaa kuuntelukanavan sovelluksen palvelimelta. Tämän jälkeen Ideawall-sovellus voi lähettää ilmoituksia reaaliajassa.

Ilmoitukset-näkymään tulevat myös muilta käyttäjiltä tulevat kutsut. Käyttäjät voivat kutsua toisia käyttäjiä omiin ryhmiinsä tai suositella itseään olemassa

olevaan ryhmään. Kutsun hyväksyminen lisää käyttäjän idean ryhmään. Hylkääminen poistaa kutsun ilmoitukset-näkymästä.

7.1.2 Profiilinäkymä

Käyttäjän profiilinäkymä avautuu valitsemalla yläpalkissa olevan My profile -painikkeen tai toisen käyttäjän profiilin linkkiä. Profiilinäkymässä on esitettynä Ideawall-sovelluksen käyttäjän tiedot.

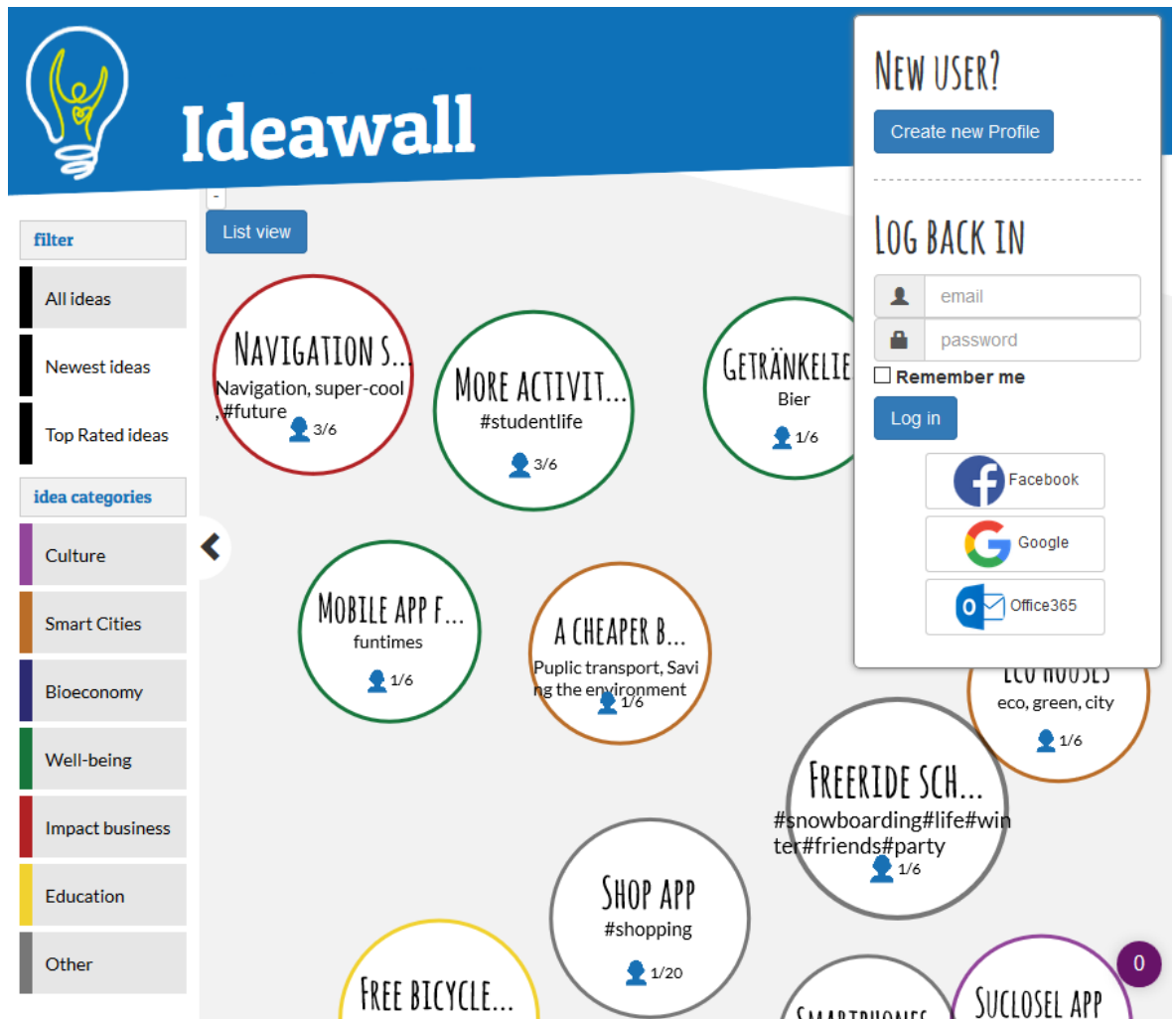


Kuva 6. Kuva Ideawall-sovelluksen profiilinäkymästä

Profiilinäkymästä muut käyttäjät voivat nähdä käyttäjän nimen, kotimaan, kuvauksen, taidot, kiinnostavat kategoriat, Tag-hakusanat, käyttäjän ryhmän, kommenttien ja tykkäysten lukumäärä. Käyttäjän ollessa kirjautuneena hän voi profiilinäkymän kautta käydä muokkaamassa tietojaan tai kirjautua ulos. (Kuva 6.)

7.1.3 Sisäänkirjautuminen

Sisäänkirjautumisnäkyminen avautuu valitsemalla sovelluksen yläpalkissa olevan Login-painikkeen, näkymä on esitettyä kuvassa 7. Sisäänkirjautumisnäky-
mästä käyttäjä voi tehdä uuden käyttäjätilin, kirjautua sisään sovellukseen teh-
dyllä käyttäjällä tai kirjautua sisään käyttäen Facebook-, tai Google-käyttäjätä-
liä.



Kuva 7. Kuva Ideawall-sovelluksen sisäänkirjautumisnäkyimestä

7.1.4 Käyttäjän luominen

Ideawall-sovelluksen uuden käyttäjän luominen tapahtuu valitsemalla Create new profile -painikkeen sisäänkirjautumisnäkyimestä. Sovellus sen jälkeen pyytää käyttäjältä sähköpostia ja salasanaa tai kirjautumaan sisään käyttäen

Facebook- tai Google-käyttäjätiliä. Tietojen syöttämisen jälkeen käyttäjä ohjataan uuden käyttäjän lomakkeelle (kuva 8), johon käyttäjä syöttää omat tietonsa. Tietoihin kuuluvat:

- nimi
- kotimaa
- lyhyt kuvaus
- profiilikuva
- vähintään kolme taitoa
- yksi tai useampi kiinnostava kategoria
- valinnainen, vapaasti kirjoitettava Tag-hakusana.

Kuva 8. Kuva Ideawall-sovelluksen uuden käyttäjän lomakkeesta

Lomake on jaettu kolmeen osaan. Lomakkeen kulkua kuvaavat ikkunan alalaidalla sijaitseva sivunumero, sekä interaktiivinen edistymispalkki. Palkki kertoo käyttäjälle mitä lomakkeelta vielä puuttuu, jotta voidaan siirtyä seuraavalle sivulle. Ensimmäisessä lomakkeen osassa syötetään nimi, kotimaa, lyhyt kuvaus ja profiilikuva. Toisessa lomakkeen osassa valitaan taidot, jotka kuvaavat käyttäjän osaamista. Kolmannessa lomakkeen osassa käyttäjä valitsee kiinnostavat kategoriat ja lisää vapaasti kirjoitettavia Tag-hakusanoja. Tag-haku-

sanaa käytetään hyödyksi vain hakutoimintoa käytettäessä, jotta toisen käyttäjän nimeä ei välttämättä tarvitse tietää, kun hakutoimintoa käytetään. Lomakkeen täytön jälkeen käyttäjäprofiili on valmis.

7.1.5 Idean luominen

Ideawall-sovelluksen uuden idean luominen tapahtuu valitsemalla Add your idea -painikkeen. Tämän jälkeen käyttäjä ohjataan uuden idean lomakkeelle (kuva 9). Tietoihin kuuluvat:

- idean nimi
- idean lyhyt kuvaus
- idean tarkoitus
- idean kategoria
- idean Tag-hakusanat
- vähintään kolme kappaletta ideaan tarvittavia taitoja.

YOUR IDEA

Fill in the required information to submit your idea. You can always go back to change anything or edit the idea when you've submitted it to the ideawall.

Title

idea title

Description

Few short words what is your idea.

Idea details

What the idea is about? What problems does it solve?

1/3

Next >

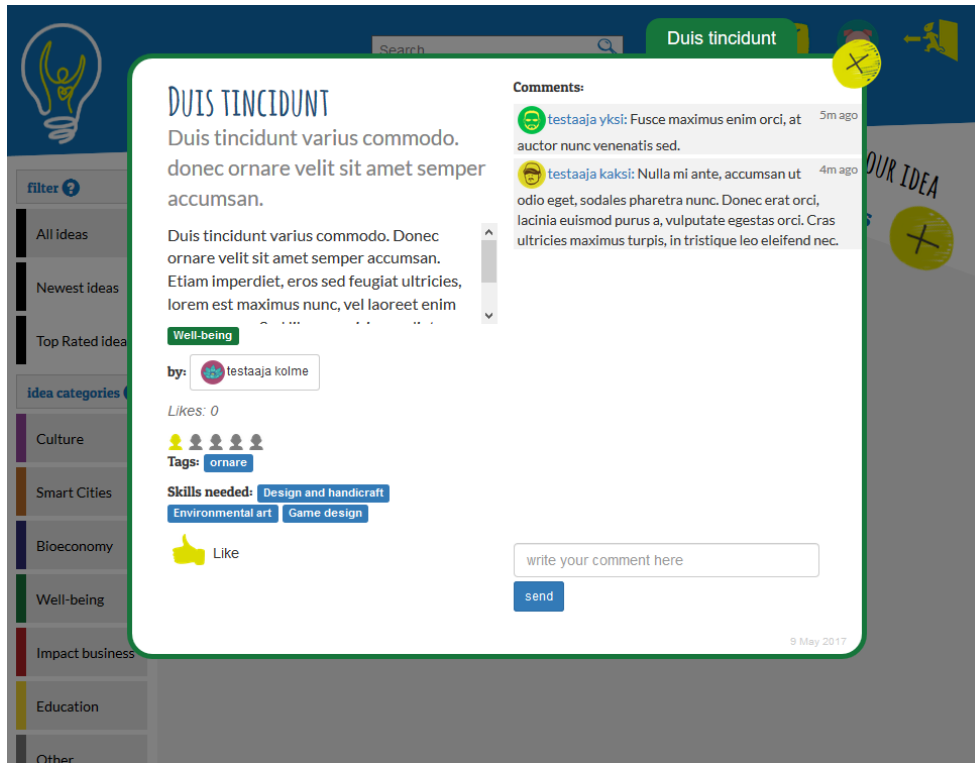
Required: Title Description Idea details

Kuva 9. Kuva Ideawall-sovelluksen uuden idean lomakkeesta

Uuden idean lomake on jaettu kolmeen osaan, kun kaikki osat ovat täytetty idea lisätään Ideawall-sovellukseen. Lomakkeen ensimmäisessä osassa pyydetään idean nimi, lyhyt kuvaus ja idean tarkoitus. Lomakkeen toisessa osassa pyydetään mihin kategoriaan idea kuuluu ja vähintään yksi Tag-hakusana. Lomakkeen kolmannessa osassa pyydetään valitsemaan vähintään kolme taitoa, joita idean toteuttamiseen vaaditaan.

7.1.6 Ideanäkymä

Käyttäjä ohjataan ideanäkymään valitsemalla oman tai toisen käyttäjän luoman idean (kuva 10). Ideanäkymässä on esitettynä idean tiedot, joita käyttäjä on ideaan lisännyt. Käyttäjät voivat myös tykätä ja lähettää kommentteja idealle. Ideanäkymä on jaettu kahteen osaan, josta vasen puoli on käytetty idean tietojen esittämiseen ja oikea puoli on varattu ideaan kirjoitettujen kommenttien esittämiseen. Ideanäkymässä on myös esillä ryhmän koko ja ryhmän jäsenten määrä, tykkäysten määrä, Tag-hakusanat ja tarvittavat taidot. Taidot, jotka täsmäävät idean tarvittavien taitojen ja käyttäjän osaavien taitojen kanssa on merkitty keltaisella pohjalla ja eroavat muista tarvittavista taidoista, jotka ovat sinisellä pohjalla. Värien avulla käyttäjä huomaa profiilinsa taitojen täsmäävän idean kanssa ja mahdollisesti soveltuvansa idean ryhmään.



Kuva 10. Kuva Ideawall-sovelluksen ideanäkymästä

Kommentin luominen tapahtuu kirjoittamalla halutun tekstin uuden kommentin tekstikentälle ja valitsemalla send-painikkeen. Kommentti ilmestyy muille käyttäjille, kun ideanäkymä ladataan uudelleen. Kommentin lähettäjälle kommentti ilmestyy JavaScript-ohjelmakoodin avulla heti näkyviin lähetysnapin valinnalla. Kommentin lähettäjä voi muokata tai poistaa lähettämänsä kommentin halutessaan. Uusi kommentti lähettää aina ilmoituksen idean luojalle.

Käyttäjä, jolla ei ole omaa ryhmää voi lähettää pyynnön idean luojalle haluavansa liittyä idean ryhmään. Käyttäjälle ilmestyy ideanäkymään Request to join -painike silloin, kun käyttäjä ei omista ideaa ja ei ole toisessa ryhmässä. Toiminto lähettää pyynnön idean luojalle, joka kysyy käyttäjän hyväksymistä ryhmään. Hyväksymisen jälkeen käyttäjä on lisätty kuulumaan idean ryhmään.

Käyttäjät voivat lähettää idean luojalle tykkäyksen hyvästä ideasta valitsemalla tykkäys-painikkeen. Tykkäys lähettää idean luojalle aina ilmoituksen uudesta tykkäyksestä. Idean saamien tykkäysten määrä on esitettyä ideanäkymässä. Tykkäysten määrä on myös näkyvillä käyttäjän profiilinäkymässä. Tykkäysten ja kommenttien määrä vaikuttaa idean kokoon pallonäkymässä, sekä nostaa ideaa ylemmäs Top Rated ideas -listaustoiminnassa.

7.1.7 Idean asetukset

Idean luoja ja idean ryhmässä olevat käyttäjät voivat siirtyä ideanäkymästä idean asetuksiin. Ideanäkymän vierellä olevan idean asetukset -välilehteä valitsemalla voidaan siirtyä tähän näkymään.



Kuva 11. Kuva ideawall-sovelluksen idean asetukset -näköymästä

Idean asetukset -näköymästä idean luojalla on mahdollisuus poistaa ryhmänsään olevia käyttäjiä, lisätä suositeltuja käyttäjiä idean ryhmään, muokata ideaa, lukita idean ja poistaa idean. Muut ryhmässä olevat käyttäjät voivat tästä näköymästä ainoastaan poistaa itsensä ryhmästä (kuva 11).

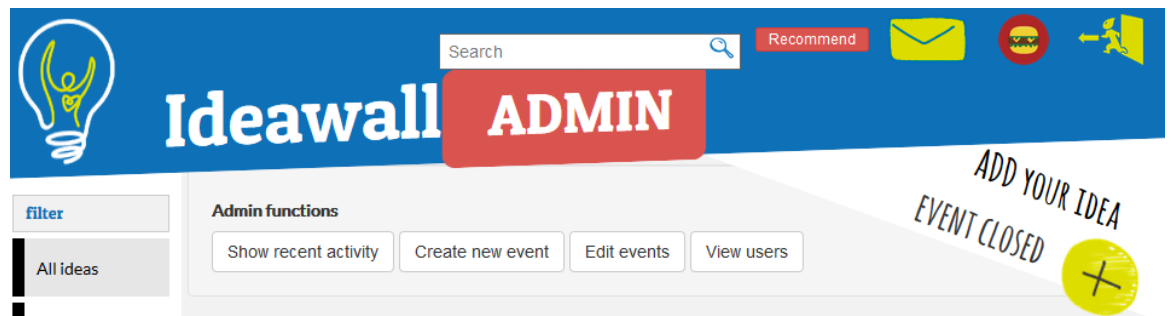
Idean lukitseminen estää muiden käyttäjien mahdollisuuden lähettää idean luojalle pyyntöjä lisätä heidät idean ryhmään. Idean luoja voi tällä viestiä muille käyttäjille, ettei etsi enää uusia ryhmän jäseniä ideaansa (kuva 12).



Kuva 12. Kuva lukitusta ideasta

7.2 Ylläpitäjän näkymät

Ylläpitäjällä on tavallisen käyttäjän kaltainen näkymä, mutta ylläpitäjällä on näiden toimintojen lisäksi näkyvillä ylläpitäjän työkalupalkki, jossa on ylläpitäjälle tarpeellisia työkaluja (kuva 13).



Kuva 13. Kuva Ideawall-sovelluksen ylläpitäjän työkalupaneelista

Näiden näkymien lisäksi ylläpitäjällä on mahdollisuus luoda rajaton määrä uusia ideoita, sekä muokata tai poistaa mikä tahansa idea. Ylläpitäjä voi myös muokata ideoiden ryhmiä siirtämällä ryhmissä olevia käyttäjiä toisiin ryhmiin tai poistaa käyttäjiä ryhmästä.

7.2.1 Näytä tapahtumat

Ylläpitäjä voi nähdä viimeisimmät käyttäjien tekemät tapahtumat Näytä tapahtumat -näköymästä (kuva 14). Tapahtumiin kuuluvat:

- käyttäjän kirjautuminen
- käyttäjän muokkaaminen
- uuden käyttäjän luominen
- uuden idean luominen
- idean muokkaaminen
- idean tykkäys
- kommentin luominen.

ADMIN

Search

Recommend

Admin functions

Show recent activity Create new event Edit events View users

IDEAWALL ACTIVITY

Please note: Activity only shows the most recent changes happening in the database, it does not log all the changes happening to the same database entry

1 - 25 of 49

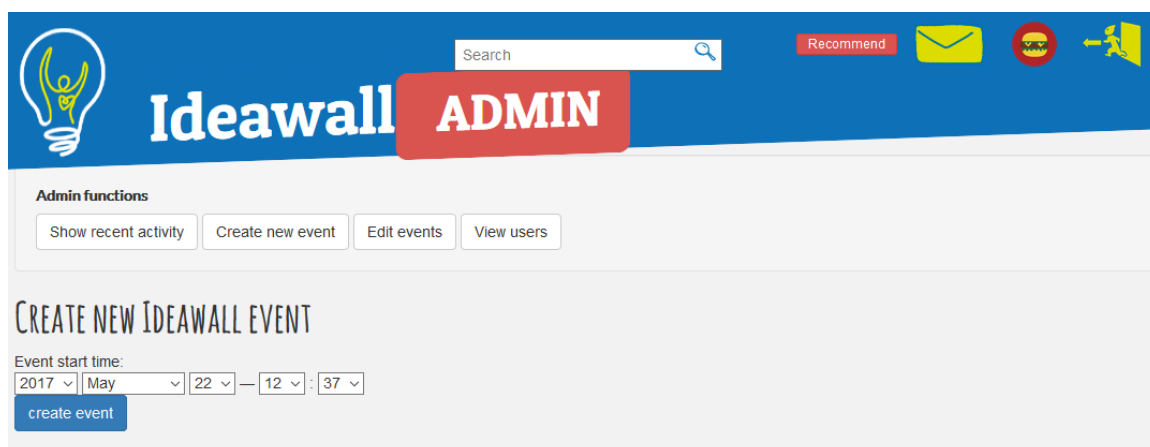
← Previous 1 2 Next →

Time	User	Content	Idea	Type
2017-05-22 12:31:16 UTC	Testaaja Ukkeli	Idea edit	Maecenas vel ligula dui.	Idea
2017-05-22 12:30:30 UTC	Testaaja Ukkeli	Idea edit	Etiam et sagittis arcu.	Idea
2017-05-22 12:29:42 UTC	Testaaja Ukkeli	Idea edit	Lorem ipsum	Idea
2017-05-22 12:27:29 UTC	testaaja admin	User most recent sign in/update-		User
2017-02-24 10:56:59 UTC	Testaaja Ukkeli	User most recent sign in/update-		User
2017-02-24 10:40:01 UTC	testi henkilö	New Comment	Maecenas vel ligula dui.	Comment
2017-02-24 10:39:59 UTC	testi henkilö	New Comment	Maecenas vel ligula dui.	Comment
2017-02-24 10:39:56 UTC	testi henkilö	New Comment	Maecenas vel ligula dui.	Comment
2017-02-24 10:39:33 UTC	testi henkilö	New Comment	Maecenas vel ligula dui.	Comment
2017-02-24 10:39:31 UTC	testi henkilö	New Comment	Maecenas vel ligula dui.	Comment
2017-02-24 10:37:27 UTC	testi henkilö	New Comment	Maecenas vel ligula dui.	Comment
2017-02-24 10:37:27 UTC	testi henkilö	New Comment	Maecenas vel ligula dui.	Comment

Kuva 14. Kuva Ideawall-sovelluksen Näytä tapahtumat -näköymästä

7.2.2 Luo ajastin

Luo ajastin -näköymästä voidaan luoda alkamaan uusi ajastettu toiminto sovellukseen (kuva 15). Ajastimeen on valmiiksi ohjelmoitu ajankohta, jolloin uusien ideoiden luominen on estetty ja käyttäjät voivat ainoastaan liittyä olemassa oleviin ryhmiin. Ajastimen tarkoituksena on saada kaikki käyttäjät kuulumaan johonkin ryhmään.



Kuva 15. Kuva Ideawall-sovelluksen Luo ajastin -näköymästä

7.2.3 Muokkaa ajastimia

Muokkaa ajastimia -näköymässä on listattuna kaikki sovellukseen luodut ajastimet (kuva 16). Listattuja ajastimia voidaan tarpeen tullen poistaa listalta tai ottaa toisen ajastimen käyttöön.

ADMIN

Search

Recommend

Admin functions

Show recent activity Create new event Edit events View users

EVENT LIST:

List of Ideawall Timed events. Only one event can be active.

event id:	start time:	close time:
1	2017-02-18 07:30:00 UTC	2017-02-25 07:30:00 UTC
<input type="checkbox"/> active		
edit		
delete		
2	2017-02-20 10:58:00 UTC	2017-02-27 10:58:00 UTC
<input type="checkbox"/> active		
edit		
delete		
3	2017-02-23 09:42:00 UTC	2017-03-02 09:42:00 UTC
<input checked="" type="checkbox"/> active		
edit		
delete		

Kuva 16. Kuva Ideawall-sovelluksen Muokkaa ajastimia -näköystä

7.2.4 Käyttäjalista

Käyttäjalista-näkymässä on listattuna kaikki sovelluksen tavalliset käyttäjät. Listalta näkyy, onko käyttäjä luonut tai liittynyt ryhmään, jolloin tiedetään käyttäjän saaneen Ideawall-sovelluksen tehtävän valmiiksi (kuva 17).

ADMIN

Search

Recommend

Admin functions

Show recent activity Create new event Edit events View users

Registered users in Ideawall

Name	Email address	Has groupidea	Finished setup	created:
Testaaja Ukkeli	testaaja@testaaja.testfalse	false	true	Maecenas vel ligula dui.
testi henkilö	test@test.test	false	true	-

Kuva 17. Kuva Ideawall-sovelluksen Käyttäjalista -näköystä

7.3 Muita näkymiä

Ideawall-sovelluksessa käytetään myös näkymiä, joita käytetään kaikilla aikaisemmin mainituilla näkymillä. Nämä näkymät ohjelmoitiin parantamaan sovelluksen käytettävyyttä.

7.3.1 Alert-ilmoituslaatikko

Ideawall-sovellukseen lisättiin alert-ilmoituslaatikot havainnollistamaan käyttäjälle, että käyttöliittymällä valittujen painikkeiden toiminnot on suoritettu loppuun tai ilmoittaa mahdollisista virhetilanteista. Ilmoituslaatikon näyttäminen tapahtuu kolmessa vaiheessa:

Vaihe 1. Palvelin lähettää käyttäjän verkkoselaimeen JavaScript-ohjelmakoodin (kuva 18) AJAX-kutsulla (kuva 19).

```
1 makealert("You can only participate in one idea.");
```

Kuva 18. Käyttäjän verkkoselaimelle lähetettävän ilmoituslaatikon JavaScript-ohjelmakoodi

```
40     else
41       respond_to do |format| # user already has an idea
42         format.js { render 'already_have_idea' }
43     end
```

Kuva 19. Ilmoituslaatikon AJAX-kutsu

Vaihe 2. Käyttäjän verkkoselain suorittaa *app.index.js* -tiedostosta *makealert*-funktion, jossa tehdään ilmoituslaatikko näytettäväksi verkkoselaimelle (kuva 20).

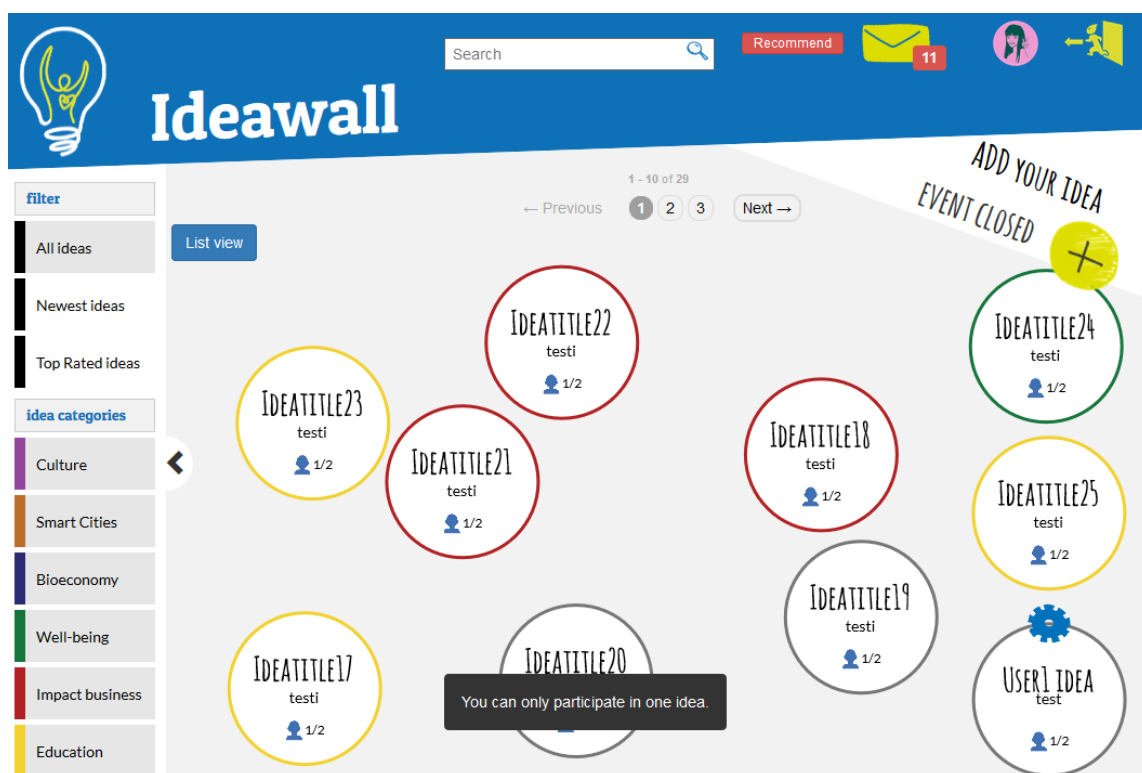

```

29 function makealert(text){
30     $("body").append("<div class='alert alert-notice show'>"+text+"</div>");
31
32     setTimeout(function() {
33         $(".alert").removeClass("show");
34         $(".alert").remove();
35     }, 3000);
36 }

```

Kuva 20. app.index.js -tiedoston JavaScript-ohjelmakoodi ilmoituslaatikosta

Vaihe 3. Verkkoselain näyttää esitetyn ilmoituslaatikon (kuva 21), joka suorittaa CSS-tyyliohjeissa määritetyn animaation (kuva 22).



Kuva 21. Kuva Ilmoituslaatikosta.

```

973 .show {display:none;}
974
975 .show {
976     visibility: visible !important;
977
978     -webkit-animation: fadein 0.5s, fadeout 0.5s 2.5s;
979     animation: fadein 0.5s, fadeout 0.5s 2.5s;
980 }
981
982 @-webkit-keyframes fadein {
983     from {bottom: 0; opacity: 0;}
984     to {bottom: 30px; opacity: 1;}
985 }
986
987 @keyframes fadein {
988     from {bottom: 0; opacity: 0;}
989     to {bottom: 30px; opacity: 1;}
990 }
991
992 @-webkit-keyframes fadeout {
993     from {bottom: 30px; opacity: 1;}
994     to {bottom: 0; opacity: 0;}
995 }
996
997 @keyframes fadeout {
998     from {bottom: 30px; opacity: 1;}
999     to {bottom: 0; opacity: 0;}
1000 }

```

Kuva 22. Ilmoituslaatikolle application.css -tiedostoon määritelty animaatio

7.3.2 Tooltip-vihjelaatikko

Tooltip-vihjelaatikon tehtävänä on kertoa lyhyesti, mitä hiiren osoittimen alla olevan painikkeen toiminto tekee. Useat informaationäytöt ja toiminnot Ideawall-sovelluksessa hyödyntävät kuvakkeita toiminnon viestimiseen. Tooltip-vihjelaatikot lisättiin, jotta ne täydentäisivät käyttäjän tietoisuutta painikkeen toiminnasta tai informaatiosta, johon hiiri on osoitettuna. Tooltip-vihjelaatikot ovat Bootstrap-lisäosan ominaisuus. Vihjelaatikon näyttäminen tapahtuu kahdessa vaiheessa:

Vaihe 1. Verkkoselain suorittaa JavaScript-ohjelmakoodilla kirjoitetun Bootstrap-lisäosan `.tooltip()` -funktion (kuva 23).

```

3 $('[data-toggle="tooltip"]').tooltip();

```

Kuva 23. Bootstrap-lisäosan `.tooltip()` -funktio

Vaihe 2. Käyttäjälle näytetään vihjelaatikko kaikkiin HTML-elementteihin, joihin se on määritelty (kuva 24). Määrittely tehdään lisäämällä "data-toggle" ja "title"-määrittelyt HTML-elementtiin (kuva 25).



Kuva 24. Tykkäys-painikkeen vihjelaatikko

```
95 <div id="likebutton" class="btn" data-toggle="tooltip" title="Good idea? Send a like!" data-behavior="like-idea" data-idea="<%= @idea.id %>"><%= image_tag 'ticon-yellow', width:'32', height:'32', id:'likeimg' %>Like</div>
```

Kuva 25. Vihjelaatikon määrittely tykkäys-painikkeella.

7.4 Evästeet

Ideawall-sovellus tallentaa käyttäjän verkkoselaimen evästeeksi seuraavat asiat:

- etusivun tila pallonäkymässä tai listanäkymässä
- sivupalkin näkyminen tai piilotus
- käyttäjän kirjautuminen, jos kirjautuessa valittiin "Remember me" -toiminto.

Evästeet salataan Ruby on Rails -sovelluskehityksen standardin mukaan.

(Ruby on Rails Guides 2017.)

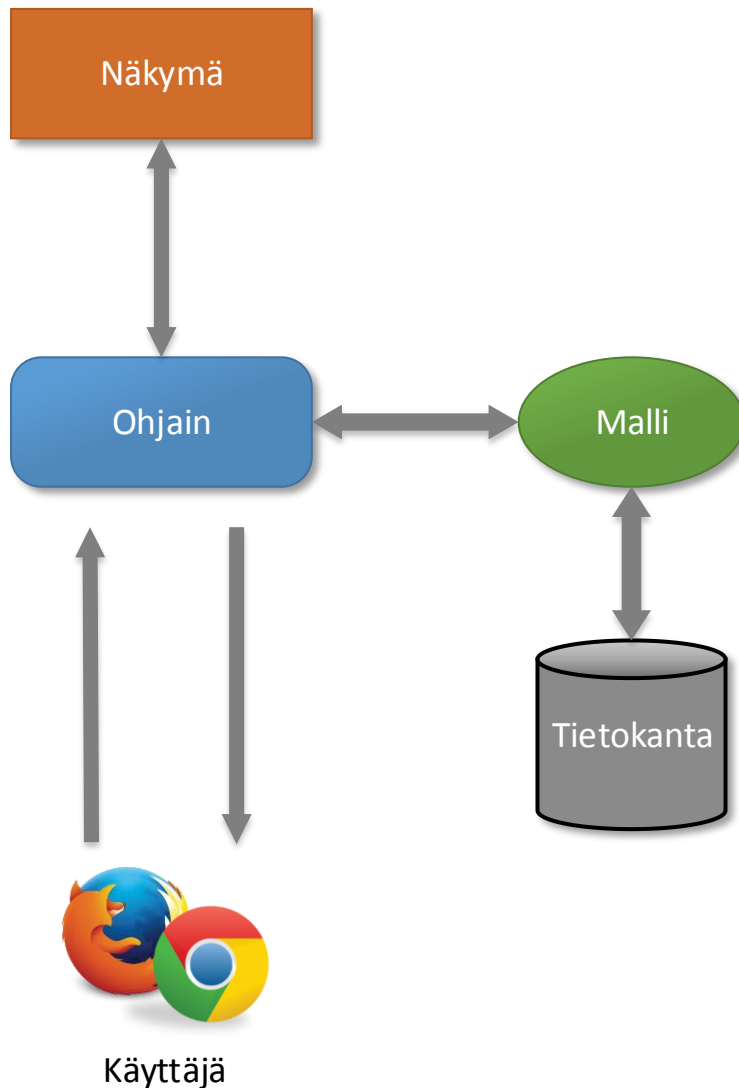
8 IDEAWALL BACKEND

Sovelluksen Back end -osaksi kutsutaan sitä osaa, joka toimii palvelimella.

Back end -osa voi koostua myös monesta eri palvelimesta, jotka yhdessä vastaavat sovelluksen toiminnoista. Esim. sovelluksen tietokanta ja käyttäjälle jaettava toimintalogiikka voivat toimia täysin eri palvelimilla.

8.1 Ohjaimet

Ruby on Rails -sovelluskehyksessä, kaikki sovelluksen palvelimella ajettavat funktiot ohjelmoidaan ohjaimiin. Hyvänä sääntönä pidetään, että yhdelle sovelluksen tietokannassa olevalle asialle tai yhdelle sovelluksen näkymälle on oma ohjaimensa. Ohjainten toiminta sovelluksessa on esitettyä kuvassa 26.



Kuva 26. Kuva Ruby on Rails -sovelluskehiksen sovelluksen toiminnasta

Käyttäjän tehdessä toiminnon sovelluksessa, hänen verkkoselaimensa lähettää kutsun sovelluksen palvelimelle. Kutsu otetaan vastaan ja siirretään sovelluksen reittien osoittamaan ohjaimen. Ohjain tämän jälkeen suorittaa sille an-

netut tehtävät. Tehtävä voi olla uuden näkymän näyttö, jonka Template-sivumuotti muunnetaan HTML-merkinnäksi. Usein dynaamisissa sovelluksissa ohjain on vuorovaikutuksessa Model-datamallien kanssa, jotka vastaavat tietoja sovelluksen tietokannassa. Ohjain ensin hakee tarvittavat tiedot Model-datamallista, sen jälkeen kääntää näkymän HTML-merkinnäksi ja lopulta lähettää vastauksen takaisin käyttäjän verkkoselaimeen. (Hartl 2016.)

8.1.1 Registrations controller

Registrations-ohjain on yksi Devise-lisäosan mukana tulleista ohjaimista, johon on lisätty. Tähän on lisätty Ideawall-sovelluksen käyttäjälle kuuluvat asiat mukaan. Näihin kuuluvat mm. käyttäjän muokkauksen ottamaan huomioon taidot, Tag-hakusanat ja kategoriat. Ohjaimessa on ohjelmoituna säännöt, miten nämä asiat päivitetään tietokantaan. (Liite 4/5.)

8.1.2 Application controller

Application-ohjain on se, joka ajetaan aina oletuksena ennen muita ohjaimia. Ohjaimiin kuuluvat mm. varmennukset, jotta käyttäjä pysyy muistissa ja kirjautuneena eri näkymissä ja evästeiden pitäminen muistissa eri näkymien välillä. Tämä ohjain myös ohjaa käyttäjän näyttämään virhenäkymän, jos näytettävä tieto tai verkkoselaimen osoite ei täsmää sovellukseen. (Liite 4/5.)

8.1.3 Comments controller

Comments-ohjain sisältää kaikki sovelluksen toiminnot, jotka kuuluvat kommenteille. Näihin toimintoihin kuuluvat: Kommentin luominen, muokkaus ja kommentin poistaminen. (Liite 4/5.)

8.1.4 Events controller

Events-ohjain sisältää Ideawall-sovelluksen ajastimeen kuuluvat toiminnot ja myös admin-käyttäjän työkalupalkin näkymien toiminnot. Nämä toiminnot ovat: Näytä tapahtumat, luo ajastin, muokkaa ajastimia, käyttäjälista. (Liite 4/6.)

8.1.5 Ideas controller

Ideas-ohjain toimii ideanäkymien näyttämisen, ideoiden muokkauksen ja myös etusivun toimintojen ohjaimena. Tämä on myös koko sovelluksen massiivisin ohjain. (Liite 4/9.) Sen toimintoihin kuuluvat:

- etusivun näyttäminen
- ajastimen hakeminen
- uuden idean lisäys
- idean muokkaus
- idean suositus
- idea settings- näkymän näyttäminen
- hakutoiminto
- sivupalkin filter -toiminto
- idean tykkäys- toiminto
- ideanäkymän -näyttäminen
- idean lukitus -toiminto
- idean poistaminen
- ryhmän jäsenen poistaminen
- poistuminen ryhmästä
- ilmoitusten näyttäminen.

8.1.6 Invites controller

Invites-ohjaimen kuuluvat kaikki käyttäjien tekemiin kutsuihin liittyvät toiminnot. Toiminnoista löytyvät säännöt mitä tapahtuu, kun käyttäjä hyväksyy tai hylkää kutsun tai käyttäjä lähettää idean luojalle suosituksen ryhmään liittymiseen. (Liite 4/19.)

8.1.7 Notifications controller

Notifications-ohjaimen kuuluvat toiminnot mitä tapahtuu, kun käyttäjälle saapunut ilmoitus valitaan ja kun ilmoitus on luettu. Ohjain muun muassa tallentaa tietokantaan tiedot ilmoituksen lähettäjistä ja vastaanottajasta, sekä tiedosta siitä, onko ilmoitus luettu. (Liite 4/20.)

8.1.8 Profiles controller

Profiles-ohjaimessa on ohjelmoituna kaikki profiiliin liittyvien toimintojen funktiot. Näitä toimintoja ovat: profiilinäkymän näyttäminen, profiilin poistaminen, profiilin muokkaus, profiilin tietojen täyttäminen. (Liite 4/20.)

8.1.9 Sessions controller

Sessions-ohjain sisältää Omniauth-lisäosan toiminnot käyttäjän sisään kirjautumiseen. Ohjaimen tehtävänä on suorittaa varmennukset ja tallentaa tarvittavat tiedot tietokantaan. (Liite 4/22.)

8.2 Tietokanta

Ideawall-sovelluksen tietokantana käytetään Ruby on Rails -sovelluskehityksen oletuksena käytettävää SQLite-tietokantaa. Tietokanta soveltui hyvin kaikkiin tarvittaviin toimintoihin, joten tarvetta sen vaihtamiseen ei ollut. (Liite 3.)

8.3 Ideawall -sovelluksen näyttö Eliademy -verkko-opinnotsivustolla

Eliademy -verkko-opinnotsivustolle sisällyttäminen vaati, että Ideawall-sovellukseen määriteltiin asetukset, jotta verkkoselaimet saavat luvan noutaa sisällytettyjä sivuja. Sisällytettyjen sivujen näyttäminen on tietoturvasyistä oletuksena estetty ja vaatii palvelimelta erityisen säännön (Mozilla Developer Network 2017).

9 JATKOKEHITYS

Ideawall-sovellus jäi monelta osin vielä kesken. Viimeisimpien käyttäjätestausten perusteella, sovellukseen tulisi kehittää vielä monia toimintoja, jotta sovellus olisi mahdollisimman helppokäyttöinen (Liite 2).

Sovelluksen ohjelmakoodin kannalta paljon olisi vielä paranneltavaa. Esimerkiksi Ideas-ohjain on niin täynnä toimintoja, että ohjelmakoodin jatkokehityksen kannalta se tulisi purkaa useammaksi pienemmäksi ohjaimeksi. Ohjelmakoodin lukeminen ja ymmärtäminen parantuisi, jos Ideas-ohjaimen toiminnot olisivat loogisemmin lajiteltuna.

Ylläpitäjän työkalut jäivät kesken ja niiden toiminta pitäisi viimeistellä. Tällä hetkellä työkalujen hyödyllisyys voi olla kyseenalainen, koska ylläpitäjällä ei ole hyvää näkymää tarkistaa kaikkien opiskelijoiden tehtävän suorittamisen tila.

Sovelluksen ajastin-ominaisuuden toiminta jäi viimeistelemättä. Sen käyttö sovelluksessa voi estää opiskelijoiden tehtävän teon, koska sovellus ei kerro tarpeeksi selkeästi ajastimen toimintaa. Alkuperäisessä suunnitelmassa opiskelijat ohjataan ajastimen avulla etsimään ryhmiä. Sovellus kuitenkin vain estää opiskelijoiden ideoiden tekemisen ilman erillistä ilmoitusta siitä, mitä muuta sovelluksella pitäisi tehdä.

Komentointi jäi viimeistelemättä. Sovellukseen olisi voinut tehdä ominaisuuden toisen opiskelijan kommenttiin vastaamisen, jolloin kommenttialueen keskustelujen seuraaminen olisi selkeämpää.

Lähes kaikki sovelluksen vapaat tekstikentät jäivät rajoittamatta. Rajoittamattomien tekstikenttien käytössä on aina vaarana, että sovelluksen käyttöliittymä hajoaa käyttökelvottomaksi, koska käyttäjä joko tahallaan tai tietämättömästi syöttää liikaa merkkejä, esimerkiksi idean nimeen.

10 PÄÄTELMÄT JA LOPPUYHTEENVETO

Lähtökohtaisesti minulla oli vähäinen kokemus verkkosivujen ohjelmoinnista. Olin tehnyt kaksi yksinkertaista verkkosivua käyttäen PHP- ja JavaScript-ohjelmointikieltä. Otin tehtäväkseni tutustukseni, millainen sovelluskehys soveltuisi parhaiten tämän sovelluksen toteuttamiseen. Vaihtoehtoja oli monia. Suurimpana kriteerinä oli etsiä sovelluskehys, joka soveltuisi parhaiten sovellukselle, jossa itse käyttäjät luovat nettisivun sisällön. Ideavall näyttäisikin täysin tyhjältä, jos se ei sisältäisi yhtään ideaa. Tutustuessani sovelluskehyyksiin useat ammattilaisfoorumit suosittelivat Ruby on Rails -sovelluskehysten käyttöä. Sovelluskehysellä on hyvä historia useimpien suosittujen verkkosovellusten pohjana, joissa käyttäjät luovat verkkosivujen sisällön. Punnitsin myös muita mahdollisia ratkaisuja. Muun muassa Laravel-sovelluskehystä, jota ohjelmoidaan PHP-ohjelmointikielellä ja Django-sovelluskehystä, joka toimii Python-ohjelmointiskriptauksella. Ruby on Rails -sovelluskehysten kattava ohjeistus sai minut lopulta päättämään sen käyttöön.

Projekti muuntautui paljon yrittäessäni toteuttaa suunnitellut asiat mahdollisimman yksinkertaisesti. Muun muassa sovelluksen oli alun perin tarkoitus siirtää automaattisesti käyttäjiä ryhmiin. Automaattinen järjestelmä olisi ollut aivan liian monimutkainen toteuttaa, jotta se olisi tehnyt haluttavat asiat. Automaattinen järjestelmä olisi myös todennäköisesti liittänyt käyttäjiä ryhmiin, jotka eivät olisi heitä kiinnostaneet tai, joihin he eivät olisivat halunneet liittyä. Tämä olisi johtanut käyttäjän turhautumiseen, koska järjestelmä ei olisi antanut käyttäjää valita itse ryhmäänsä. Järjestelmä olisi yksinkertaisesti tehnyt asioita käyttäjää vastaan, eikä olisi palvellut heitä halutulla tavalla. Vaihtoehtoisesti suunnittelin sivustolle järjestelmän, jolla käyttäjä voi kutsua muita omaan ryhmäänsä tai käyttäjä voi suositella itseään olemassa olevaan ryhmään. Sovellus myös suosittelisi käyttäjälle ryhmää, joka olisi käyttäjää kiinnostavasta aiheesta tai tarvitsisi käyttäjän taitoja. Käyttäjät voivat itse vaikuttaa mihin ryhmään kuuluvat, jolloin he löytävät todennäköisesti sellaisen ryhmän, joka heitä kiinnostaa.

Käyttäjien kirjautuminen myös muuttui paljon alkuperäisistä suunnitelmista. Paras mahdollinen tapaus olisi, jos käyttäjää ei tarvitsisi kirjata sisään tai käyttäjä kirjautuisi valmiiksi olemassa olevilla tunnuksilla toisen palvelun kautta.

Alkuperäinen suunnitelmani oli, että käyttäjät kirjautuisivat sisälle käyttäen jo valmiiksi olemassa olevia tunnuksia. Facebook- ja Google-käyttäjätilit kattaisivat käytännössä kaikki Ideawall-sovelluksen opiskelijat. Käyttämällä jo olemassa olevia tunnuksia säästyttäisiin siltä, ettei järjestelmän tietokantaan tarvitsisi tallentaa sähköposti- ja salasana-yhdistelmää, joka olisi tietoturvallisesti paras ratkaisu. Käyttäjätestauksessa kuitenkin huomattiin, että käyttäjät eivät halunneet käyttää näitä tunnuksia, koska he kokivat, että heidän julkiset tilinsä olisivat enemmän näkyvillä ja se johtaisi heidän mielestä huonompaan tietoturvaan. Tästä syystä sivustolle rakennettiin kirjautumisjärjestelmä, jonne käyttäjä voi luoda oman tunnuksensa käyttäen mitä tahansa sähköpostitiliä. Kirjautumiseen käytetään erittäin suosittua ja pitkään kehitettyä Devise-lisäosaa (ks. Lajennukset ja lisäosat). Devise kattaa kaiken, mitä käyttäjän kirjautumiseen vaaditaan. Käyttäjä voi kuitenkin edelleen käyttää olemassa olevaa Facebook- ja Google-käyttäjätiliä, jolloin salasanaa ei tallenneta tietokantaan.

Vaikka projektia tehtiinkin ryhmässä, päädyin lopulta tekemään projektia yksin kesän harjoittelun jälkeen. Työmäärä projektissa ei yllättänyt, mutta työskentely yksin näin suuren kokonaisuuden parissa tekee viimeistelyjen ominaisuuksien teosta todella hidasta. En olisi kuvitellut, että tekisin projektin viimeistelyä vielä vuodenkin päästä.

Ruby on Rails -sovelluskehysessä on suhteellisen korkea oppimiskynnys, koska se vaatii Ruby -ohjelmointikielen opetteluun ja paljon sisäistämistä, miten vältetään kaikki sudenkuopat, jotka johtuvat sovelluskehysen automatisoidusta tyylistä tehdä tietyt asiat. Se on kuitenkin todella mukava ohjelmointiympäristö ja kokemuksen kartuttua suurempienkin ominaisuuksien aikaansaaminen on yllättävän nopeaa. Projekti opetti minulle, mitä verkkosivujen kehittäminen vaatii niin käyttäjältä, verkkoselaimelta, ohjelmakoodilta, kuin palvelimelta.

LÄHTEET

- About Ruby. 2017. Ruby Programming Language. WWW-dokumentti. Saatavissa: <https://www.ruby-lang.org/en/about/> [viitattu 7.3.2017].
- Devise. 2017. WWW-dokumentti. Saatavissa: <http://devise.plataforma-tec.com.br/> [viitattu 12.4.2017].
- Hartl, M. 2016. E-kirja. Ruby on Rails Tutorial: Learn Web Development with Rails.
- Jakobus, B & Marah, J. 2016. E-kirja. Mastering Bootstrap 4.
- Lindeman, A. 2015. Startup-intohimoa Itämeren alueelle. Verkkolehti. Saatavissa: <http://www2.kyamk.fi/Koskinen/062015/startup.html> [viitattu 6.5.2017].
- Matsumoto, Y. 2001. E-kirja. Ruby in a Nutshell.
- Mozilla Developer Network. 2017. X-Frame-Options. 2017. WWW-dokumentti. Saatavissa: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options> [viitattu 11.5.2017].
- Omniauth. 2017. Github. WWW-dokumentti. Saatavissa: <https://github.com/omniauth/omniauth/wiki> [viitattu 12.4.2017].
- Paper.js. 2017. WWW-dokumentti. Saatavissa: <http://paperjs.org/about/> [viitattu 12.4.2017].
- Ruby on Rails Guides. 2017. Rails Guides. WWW-dokumentti. Saatavissa: <http://guides.rubyonrails.org/> [viitattu 12.4.2017].
- Stewart, B. 2001. An Interview with the Creator of Ruby. WWW-dokumentti. Päivitetty 11.29.2001. Saatavissa: <http://www.linuxdevcenter.com/pub/a/linux/2001/11/29/ruby.html> [viitattu 7.3.2017].
- Will_paginate. 2017. Github. WWW-dokumentti. Saatavissa: https://github.com/mislav/will_paginate [viitattu: 11.5.2017].
- York, R. 2009. E-kirja. Beginning JavaScript and CSS Development with jQuery.

KUVALUETTELO

- Kuva 1. Esimerkki Ruby-ohjelmakoodista
- Kuva 2. Ideawall-sovelluksen yhdistävät tekijät
- Kuva 3. Kuva Ideawall-sovelluksen etusivusta
- Kuva 4. Kuva Ideawall-sovelluksen listanäkymästä
- Kuva 5. Kuva käyttäjälle tulleesta kutsusta
- Kuva 6. Kuva Ideawall-sovelluksen profiilinäkymästä
- Kuva 7. Kuva Ideawall-sovelluksen sisäänkirjautumisnäköymästä
- Kuva 8. Kuva Ideawall-sovelluksen uuden käyttäjän lomakkeesta
- Kuva 9. Kuva Ideawall-sovelluksen uuden idean lomakkeesta
- Kuva 10. Kuva Ideawall-sovelluksen ideanäkymästä
- Kuva 11. Kuva ideawall-sovelluksen idean asetukset -näköymästä
- Kuva 12. Kuva lukitusta ideasta
- Kuva 13. Kuva Ideawall-sovelluksen ylläpitäjän työkalupaneelistä
- Kuva 14. Kuva Ideawall-sovelluksen Näytä tapahtumat -näköymästä
- Kuva 15. Kuva Ideawall-sovelluksen Luo ajastin -näköymästä
- Kuva 16. Kuva Ideawall-sovelluksen Muokkaa ajastimia -näköymästä
- Kuva 17. Kuva Ideawall-sovelluksen Käyttäjalista -näköymästä
- Kuva 18. Käyttäjän verkkoselaimelle lähetettävän ilmoituslaatikon JavaScript-ohjelmakoodi
- Kuva 19. Ilmoituslaatikon AJAX-kutsu
- Kuva 20. app.index.js -tiedoston JavaScript-ohjelmakoodi ilmoituslaatikosta
- Kuva 21. Kuva Ilmoituslaatikosta.
- Kuva 22. Ilmoituslaatikolle application.css -tiedostoon määritellyt animaatio
- Kuva 23. Bootstrap-lisäosan .tooltip() -funktio
- Kuva 25. Vihjelaatikon määrittely tykkäys-painikkeella.
- Kuva 24. Tykkäys-painikkeen vihjelaatikko
- Kuva 26. Kuva Ruby on Rails -sovelluskehiksen sovelluksen