

Karo Kolari

# Ecoupon-mobiilikuponkisovelluksen liikeidea ja mainostajan web-käyttöliittymä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinööriytyö

25.3.2015

Tekijä Otsikko Sivumäärä Aika	Karo Kolari Ecoupon-mobiilikuponkisovelluksen liikeidea ja mainostajan web-käyttöliittymä 37 sivua + 7 liitettä 25.3.2015
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	ohjelmistotekniikka
Ohjaaja	lehtori Olli Hämäläinen
<p>Insinööriyön tavoitteena oli kuvata Ideja Oy:ssä toteutuksessa olleen Ecoupon-mobiilikuponkisovelluksen liikeidea ja sovelluksen mainostajan web-käyttöliittymän toteutusta.</p> <p>Ecoupon-mobiilikuponkisovelluksen tarkoituksena oli osittain korvata monien yritysten markkinoinnissa käyttämiä paperisia kuponkeja ja tarjouksia. Mainostajan käyttöliittymällä yritykset pystyisivät vaivattomasti lisäämään yrityksensä kuponkeja ja tarjouksia Ecoupon-mobiilikuponkisovellukseen. Ecouponissa kuluttajat pystyisivät näkemään yritysten lisäämiä etuja älypuhelimeen ladattavalla sovelluksella ja web-sivujen kautta.</p> <p>Web-käyttöliittymä toteutettiin PHP-ohjelmointikielellä ja Symfony2-ohjelmointikehyksellä. Tekniikat valittiin projektiryhmän osaamisen ja kokemusten perusteella. Käyttämällä tekniikoita, joista projektiryhmällä oli aiempaa kokemusta, toivottiin projektin etenemisen nopeutuvan. Tämä oli tärkeää, koska projektiryhmässä haluttiin saada tuotteen ensimmäinen versio mahdollisimman aikaisessa vaiheessa julkaistuksi.</p> <p>Projektissa tutustuttiin Idejan mobiilikuponkisovelluksen kilpailijoihin, minkä aikana saatiin selville kilpailijoiden heikkouksia ja vahvuuksia. Tutustumalla kilpailijoiden tuotteisiin Ideja löysi ratkaisuja, joita kannatti hyödyntää oman tuotteen kehittämisessä ja markkinoinnissa.</p> <p>Mobiilikuponkisovellukselle Ideja ei kuitenkaan saanut tarpeeksi rahoitusta ja mobiilimarkkinoilta alkoi löytyä yhä useampia kilpailijoita mobiilikuponkisovellukselle. Nämä edellä mainitut asiat johtivat siihen, että sovelluksen kehittäminen päätettiin lopettaa.</p>	
Avainsanat	liikeidea, mobiilikuponki, PHP, Symfony2, web-käyttöliittymä

Author Title	Karo Kolari Mobile Ecoupon application and advertiser web interface
Number of Pages Date	37 pages + 7 appendices 25 March 2015
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor	Olli Hämäläinen, Senior Lecturer
<p>The aim of this thesis was to describe a business idea of the Ecoupon mobile coupon application and the implementation of the application's advertiser-side web interface in Ideja Oy.</p> <p>During the project competing products were discovered and analyzed. The analysis showed what their strengths and weaknesses are. By studying the competitors, Ideja found solutions that could be used in their product development and marketing.</p> <p>The Ecoupon mobile coupon application was designed to partially replace paper coupons and paper promotions used by many companies in marketing. Using the advertiser's web interface, companies could easily add benefits and promotions to the mobile coupon application. Consumers would be able to see the companies' benefits via a smartphone application and web pages.</p> <p>The web interface was made using the PHP programming language and the Symfony2 software framework. Techniques were chosen based on the project group's knowledge and experience. By using techniques, with which the project group had previous experience, the group hoped to speed up the project development. This was important because the project group wanted to get the first version of the product released as early as possible.</p> <p>In the end Ideja did not get enough funding for its mobile coupon application and at the same time new competing mobile coupon applications were found in the mobile marketing field. These issues led to terminating the mobile coupon application development.</p>	
Keywords	business idea, mobile coupon, PHP, Symfony2, web interface

## Sisällys

1	Johdanto	1
2	Ecoupon-mobiilikuponkisoituksen liikeidea ja kilpailijat	3
2.1	Liikeidea	3
2.2	Kilpailijat	3
3	Mobiilimarkkinointi	8
4	Tuotteen vaatimukset ja käyttäjätarinat	10
4.1	Vaatimukset	10
4.2	Käyttäjätarinat	11
4.2.1	Ylläpitäjän käyttäjätarinat mainostajan web-käyttöliittymässä	13
4.2.2	Mainostajan käyttäjätarinat mainostajan web-käyttöliittymässä	14
4.2.3	Kuluttajan käyttäjätarinat Ecoupon-mobiilikuponkisoituksessa	15
5	Toteutusratkaisut ja arkkitehtuuri	16
6	Käytetyt teknologiat ja ohjelmistot	18
7	Projekti	22
8	Mainostajan web-käyttöliittymän toteutus	25
9	Yhteenveto	33
	Lähteet	35
	Liitteet	
	Liite 1. Benefit-luokka	

## 1 Johdanto

Tämän opinnäytetyön tarkoituksena on kuvata Ideja Oy:ssä toteutuksessa olleen Ecoupon-mobiilikuponkisovelluksen liikeideaa ja siihen suunniteltuja ominaisuuksia. Samalla työssä kuvataan sovelluksen mainostajan web-käyttöliittymän toteutusta.

Ideja Oy on vuonna 2013 perustettu yritys, jonka toimiala on ohjelmistojen suunnittelu ja tuotanto. Ideja Oy:n Ecouponin ohjelmoinnista vastaavat henkilöt työskentelivät ennen Idejan perustamista Metropolia Ammattikorkeakoulun Karhukopla-projektissa työharjoittelijoina. Karhukopla-projektissa ylläpidettiin ja tuotettiin useita Metropolia Ammattikorkeakoulun käytössä olevia web-sovelluksia, joiden ylläpito siirtyi Idejan perustamisen jälkeen Ideja Oy:n hoidettavaksi.

Ecoupon sai alkunsa projektiryhmän miettiessä, miten kupongit olisivat aina kuluttajien käsien ulottuvilla ja miten kuluttajat muistaisivat paremmin käyttää yritysten tarjoamia etuja. Varsinkin paperiset kupongit voivat helposti unohtua kotiin, jäädä kokonaan käyttämättä tai kadota. Ratkaisuksi tähän projektiryhmässä päädyttiin älypuhelimissa ja web-sivulla toimivaan mobiilikuponkisovellukseen.

Ideja Oy:n mobiilikuponkisovelluksen tarkoituksena on osittain korvata monien yritysten markkinoinnissa hyödykseen käyttämiä paperisia kuponkeja ja tarjouksia. Mainostajan web-käyttöliittymän avulla yritykset voisivat vaivattomasti lisätä yritykselle kuponkeja ja tarjouksia mobiilikuponkisovellukseen. Mobiilikuponkisovelluksessa kuluttajat pystyisivät näkemään useiden eri yritysten etuja samasta paikasta. Kuluttajalla olisi myös mahdollisuus suodattaa yritysten etuja sijainnin, paikkakunnan tai kategorian mukaan älypuhelimelle tarkoitettulla sovelluksella ja web-sovelluksella.

Nykypäivänä älypuhelin on todella suosittu ja lähes välttämätön väline, joka kulkee lähes poikkeuksetta useimpien ihmisten mukana. Älypuhelimissa toimivaa Ecouponia käyttämällä kuluttaja voisi luopua paperisten kuponkien käytöstä, koska ne olisivat saatavilla sähköisessä muodossa.

Mainostajan puolen käyttöliittymän toteutuksessa käytetään PHP-ohjelmointikielellä toimivaa Symfony2-ohjelmistokehystä ja Doctrinen ORM-kirjastoa. Sovelluksen tekemisessä ohjelmointityökaluna käytetään NetBeans-kehitysympäristöä, tietokantapalvelimena toimii MySQL ja web-palvelimena on Apache.

Edellä mainittujen aiheiden lisäksi tässä opinnäytetyössä kuvaillaan projektiryhmän työskentelyä, kilpailijoita, mobiilimarkkinoiden mahdollisuuksia ja projektin onnistuneisuutta.

## 2 Ecoupon-mobiilikuponkisovelluksen liikeidea ja kilpailijat

Idejassa pohdittiin, miten Ecoupon-mobiilikuponkisovellusta tulisi markkinoimaan yrityksille ja kuluttajille. Idejan toiveissa oli päästä mobiilikuponkimarkkinoille ensimmäisenä, jolloin tuotteen markkinoinnin kanssa olisi voitu toimia vapaammin.

### 2.1 Liikeidea

Ecoupon-mobiilikuponkisovellus päätettiin toteuttaa Ideja Oy:ssä, jossa sen tarkoituksena oli tuoda Idejalle pitkällä aikavälillä jatkuvaa kassavirtaa. Ecoupon-mobiilikuponkisovellusta oli tarkoitus markkinoida mainospalveluna yrityksille, jotka myivät tuotteita ilman välikäsiä kuluttajille. Samaan aikaan Ecoupon-mobiilikuponkisovellusta oli tarkoitus markkinoida kuluttajille mahdollisuutena nähdä haluamiensa yritysten tarjoukset ja kupongit suoraan älypuhelimien sovelluksella tai web-sivujen kautta.

Ecoupon-mobiilikuponkisovelluksen tarjouksia ja kuponkeja kutsuttiin eduiksi. Etuja yritykset olisivat voineet lisätä erillisellä mainostajille tehdyllä web-sovelluksella. Yritysten rekisteröidyttä mainostajiksi web-sovellukseen olisivat yritykset saaneet lisätä mobiilikuponkisovelluksen etuja ennalta sovitun ajan ilmaiseksi. Ilmaisen käytön tarkoituksena oli houkutella mahdollisimman monta yritystä käyttämään palvelua. Ilmaisen käytön jälkeen palvelu oli tarkoitus muuttaa maksulliseksi. Projektiryhmässä oli mietitty kahta erilaista vaihtoehtoa, joilla yrityksiä olisi voitu laskuttaa. Ensimmäisessä vaihtoehdossa laskuttaminen olisi tapahtunut sen mukaan, kuinka moni kuluttaja olisi katsonut Ecouponia käyttävän yrityksen etuja. Toisessa vaihtoehdossa laskuttaminen olisi tapahtunut sen perusteella, kuinka kauan yrityksen mainos olisi näkynyt Ecouponissa. Lopullista ratkaisua projektiryhmässä ei tehty siitä, miten yrityksiä olisi laskutettu mainospalvelun käytöstä tai miten mainospalvelu olisi hinnoiteltu.

### 2.2 Kilpailijat

Ennen sovelluksen kehittämisen aloitusta Idejassa oletettiin, että Ecoupon olisi Suomen markkinoilla ensimmäisenä ja jopa etuajassa. Lähimpänä Ecouponin ideaa näytti olevan yhdysvaltalainen Groupon, jonka toiminta keskittyy edullisten lahjakorttien myyntiin toisin kuin Idejassa toteutuksessa olleessa mobiilikuponkisovelluksessa, jonka tarkoituksena

oli jakaa ilmaisia etuja kuluttajille tarjousten ja kuponkien muodossa. Idejan tarkoituksena oli saada pidemmällä aikavälillä tuottoa yrityksiensä esille laittamien etujen kautta. Ideja löysi muita jo Suomen markkinoilla toimivia mobiilikuponkisovelluksia vasta kun Ecoupon-sovellus oli lähes kokonaan suunniteltu loppuun ja mainostajan käyttöliittymä oli lähes kokonaan valmiina. Nämä sovellukset olivat Cardu, Yamii ja Incentz. Lisäksi löytyi muita älypuhelimella toimivia sovelluksia, joissa tarjottiin kuponkeja ja muita etuja. Näiden sovellusten pääasiallinen toiminta ei kuitenkaan keskittynyt mobiilietujen tarjoamiseen. Yksi tällainen yritys oli Tassa.fi, jonka pääasiallisena tarkoituksena on tarjota tietoa kuluttajan lähellä olevista palveluista sekä ohessa antaa tarjouksia ja kuponkeja pieneen osaan lähialueen palveluista. Tieto jo markkinoilla olevien sovellusten olemassaolosta merkitsi sitä, että Ideja ei olisi ensimmäisenä mobiilikuponkimarkkinoilla. Mikään näistä yrityksistä ei ollut kuitenkaan päässyt nousemaan kuluttajien merkittävään tietoisuuteen, minkä vuoksi markkina-osuuksia ei ollut vielä jaettu ja mainospalvelun kehitystä kannatti jatkaa.

Idejassa tutkittiin kilpailijoiden tekemiä tuotteita uusien kilpailijoiden löydyttyä. Mikäli tuotteissa havaittiin käytetyn hyviä ratkaisuja, näistä otettiin mallia Idejan mobiilikuponkisovelluksen kehityksessä. Uusien ominaisuuksien lisääminen kesken projektin oli helppoa Idejassa käytetyn Scrum-pohjaisen kehittämistavan vuoksi.

## Groupon

Groupon on yhdysvaltalainen kuluttajille lahjakortteja ja tarjouksia välittävä yritys, jolla on noin 50 miljoonaa asiakasta 48:ssa eri maassa [1]. Grouponin toimintaidea perustuu suuriin ostajamääriin, joiden ansiosta se pystyy tarjoamaan edullisesti yritysten lahjakortteja ja tarjouksia kuluttajille. Kuluttajille tarjota lahjakortteja ja tarjouksia Groupon kutsuu diileiksi. Grouponin tuotto syntyy siitä, että lisää uusia asiakkaita haluavat yritykset ostavat näkyvyyttä Grouponin sovelluksesta tarjoamalla kuluttajille edullisia diilejä. Jokaisesta myydyistä diilistä Groupon saa osuuden itselleen.

Grouponin toiminta perustuu siihen, että se tarjoaa diilejä ilmaisella mobiilisovelluksella ja web-sivulla. Groupon ei suoraan kilpaile mobiilikuponkeja tarjoavien yritysten kanssa, vaikka samalla mobiilimarkkina-alueella toimiikin. Diilin tarjoaminen eroaa mobiilikuponkien eduista siten, että diiliä tehdessä kuluttaja ostaa suoraan Grouponin sovelluksesta



lahjakortin kolmantena osapuolena toimivan yrityksen tuotteeseen, joka myöhemmin voidaan lunastaa diilissä sovitulla tavalla. Mobiilikuponkisovelluksissa kuluttajan ei tarvitse etukäteen ostaa mitään. [2; 3.]

Ideja sai Grouponin mobiilisovelluksesta hyviä vinkkejä mobiilisovelluksen käyttöliittymän kehittämiseen, vaikka Grouponin toiminnan idea oli erilainen. Idejassa otettiin esimerkiksi mallia, miten Grouponin mobiilisovelluksessa tarjoukset ja lahjakortit olivat esitetty.

## Yamii

Yamii on suomalaisen LBB Solutions Oy:n kehittämä web- ja mobiilisovellus, joka on ladattavissa iPhonelle Applen App Storesta ja Android-puhelimille Google Playsta. Yamiiin palveluun yritykset voivat lisätä omia alennuksia saadakseen omalle yritykselle lisää näkyvyyttä. Mobiilisovellus etsii sijainnin perusteella lähiympäristön alennukset ja välittää tiedon suoraan kuluttajan puhelimeen. [4; 5.]

Yamii-sovelluksessa on paljon samanlaisia ratkaisuja kuin Idejan mobiilikuponkisovellukseen oli alun perin suunniteltu tehtävän. Muista kilpailijoista poiketen LBB Solutionsin sovelluksessa on mainostajilla oma Merchant-tietosivu. Merchant-tietosivusta kuluttaja näkee yrityksestä useita eri tietoja, kuten yrityksen muut tarjoukset, yhteystiedot, oletuskuvan, yrityksen lisäämiä kuvia liikepaikasta ja yrittäjän oman kuvauksen liikkeestään. Tietosivun avulla yritykset pystyvät erottumaan toisista yrityksistä itse sovelluksen sisällä. Tämä tapahtuu yrityksen Merchant-tietosivulle lisättävillä yrityksen omilla kuvilla ja kuvauksella. Ideja koki tämän ominaisuuden tärkeäksi ja päätti tehdä omaan mobiilikuponkisovellukseen vastaavanlaisen ominaisuuden.

Idejassa oltiin sitä mieltä, että Yamiilla oli paras kokonaisuus kaikista suomalaisilla markkinoilla olevista mobiilikuponkisovelluksista. Yamiin sovelluksen hyvästä kokonaisuudesta huolimatta Androidin Google Play -sovelluskaupan ja iPhonen App Store -sovelluskaupan latausmäärät olivat melko vähäisiä. Yamiin sovelluksessa etujen järjestely oli hankalaa ja mainostajat pääosin pieniä ravintola-alan yrittäjiä.

## Cardu

Cardu on suomalaisen Sunduka Oy:n kehittämä mobiilisovellus. Siinä käyttäjällä on eri yritysten kanta-asiakaskortit, jäsenkortit ja etukortit yhdessä paikassa. Näiden korttien avulla kuluttajat voivat saada tietoa eri toimijoiden tarjouksista ja mobiilikupongeista. Cardun idea jäsenkorttien tarjoamisesta ei ole täysin sama kuin mobiilikuponkisovelluksessa, mutta se kilpailee samalla mobiilimarkkina-alueella.

Cardu tekee Keskon kanssa yhteistyötä ja on saanut Tekesiltä tukea, joten se oli varteenotettava kilpailija Idejassa suunnitellulle mainospalvelulle [6; 7]. Kun Cardun olemassaolo tuli Idejassa selville olivat sen latausmäärät todella pieniä Google Play -sovelluskaupassa. Ideja myös testasi Cardua käyttämällä palvelun etuja useissa liikkeissä. Liikkeissä huomattiin, että niiden työntekijät eivät tieneet liikkeellä olevan etuja Cardussa. Cardun sovelluksen alhaiset latausmäärät ja Cardun heikko tunnettuus antoi Idejalle uskoa oman mainospalvelun menestymisen mahdollisuuksiin. Reilu vuosi sen jälkeen, kun Ideja oli saanut selville Cardun olemassaolon, ovat sen latausmäärät kasvaneet Google Play -sovelluskaupassa. Insinööriyötä kirjoittaessa Cardua on ladattu Google Play -sovelluskaupassa yli 50 000 kertaa, ja se on ollut esillä useissa Keskon liikkeissä. Cardua käyttää nykyään Keskon lisäksi moni muu iso ketju, kuten SOL, LähiTapiola ja Hong Kong -tavaratalot [8].

Idejan projektiryhmän mielestä Cardun ongelma oli, että se vaatii koko ajan paikannustiedon olevan käytössä. Mikäli paikannus ei ole käytössä sovellus pyytää laittamaan paikannuksen päälle. Projektiryhmän mielestä toinen huono asia Cardussa oli, että se vaatii käyttäjältä kirjautumisen, ennen kuin sovellusta pystyy käyttämään. Projektiryhmä koki, että pakollinen kirjautuminen kasvattaa kuluttajan kynnystä ottaa sovellus käyttöön. Eduille ei myöskään ole omia kategorioita ja yritysten tarjoamia kortteja pitää selailla loputtoman tuntuista listasta, jotta löytäisi etsimänsä kortin.

## Incentz

Incentz on suomalaisen Incentz Oy:n tekemä mobiilisovellus. Incentzin ideana on auttaa kuluttajia löytämään tarjouksia paikallisiin kauppoihin ja ravintoloihin. Sovelluksen ominaisuuksina ovat tarjousten näkyminen käyttäjän sijainnin perusteella, tarjousten näkyminen eri kielillä, mahdollisuus lisätä yritykselle useita toimipisteitä ja mahdollisuus jul-

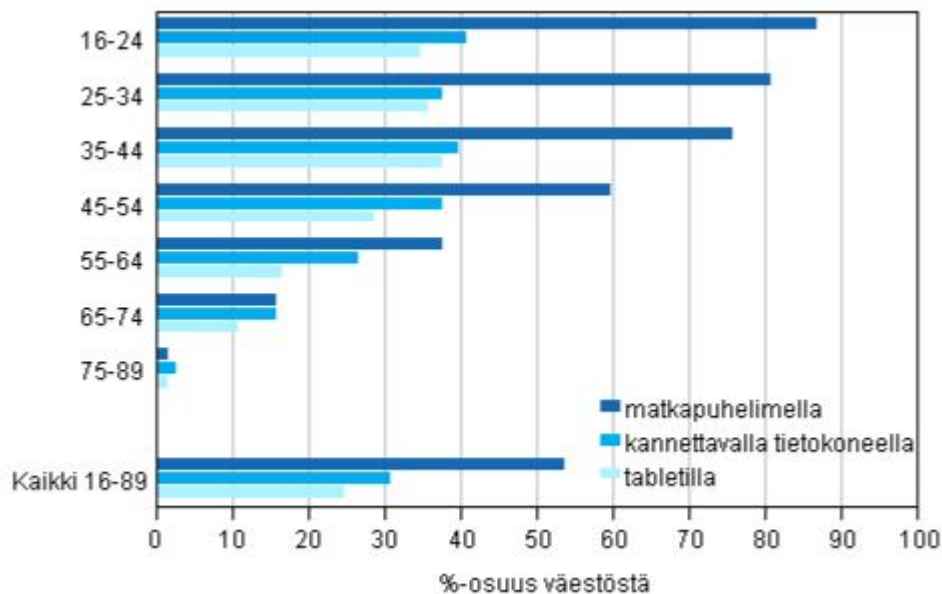
kaista tarjouksia Incentzin Facebook-sivulla. Incentz on ainoa Idejan kartoittamista kilpailuvista mobiilikuponkisovelluksista, jolla on palvelun hinnat esillä yrityksen omalla verkkosivulla. Incentz on hinnoitellut tarjouksensa kolmen eri hintaluokan paketin mukaisesti. Mitä arvokkaampi paketti on kyseessä, sitä enemmän yritys voi lisätä sovellukseen nähtäviksi tarjouksia, yrityksen toimipisteitä, saada näkyvyyttä Incentzin verkkosivulla ja Facebook-sivulta. Kuluttajat voivat sovelluksessa tallentaa tarjouksia, hakea tiettyä yritystä ja hyödyntää hakusanoja tai paikkaa tarjousten etsimisessä. [9.]

Incentzissä on hyödynnetty sosiaalisen median käyttöä markkinoinnissa. Sillä on omat Twitter- ja Instagram-tilit sekä Facebook-yhteisö, joissa Incentz jakaa oman mobiilisovelluksen etuja. Ideja päätti Incentzin tavoin jakaa etuja sosiaalisessa mediassa ja päätti tehdä oman sosiaalisen median strategian. Ideja loi oman Facebook-ryhmän ja Twitter-tilin, joiden kautta oli mahdollista jakaa kuluttajille säästöä tuovia etuja. Etujen jakamiskriteerinä oli iso tarjousprosentti ja tuotteen edullisuus verrattuna markkinoiden keskihintaan. Mikäli tuote oli useammassa paikassa tarjouksessa, tällöin valittiin aina edullisin tarjous. Edellä mainitut jakamiskriteerit asetettiin, jotta kuluttajat kiinnostuisivat Idejan Facebook-yhteisöstä sekä Twitter-tilistä, ja suosittelisivat niitä myös muille kuluttajille. Ideja olisi mainostanut Ecoupon-mobiilikuponkisovellusta ja sen tarjouksia sosiaalisen median sovellusten kautta, jos sen sovellus olisi valmistunut.

### 3 Mobiilimarkkinointi

Mobiilimarkkinointi tarkoittaa palveluita, jotka käyttävät hyödyksi matkapuhelinta osana markkinointia. Mobiilimarkkinointi pitää sisällään mainonnan lisäksi yritysten mobiilisivut, QR-koodit, kanta-asiakassovellukset ja kännykässä toimivat verkkokaupat. [10, s.2.]

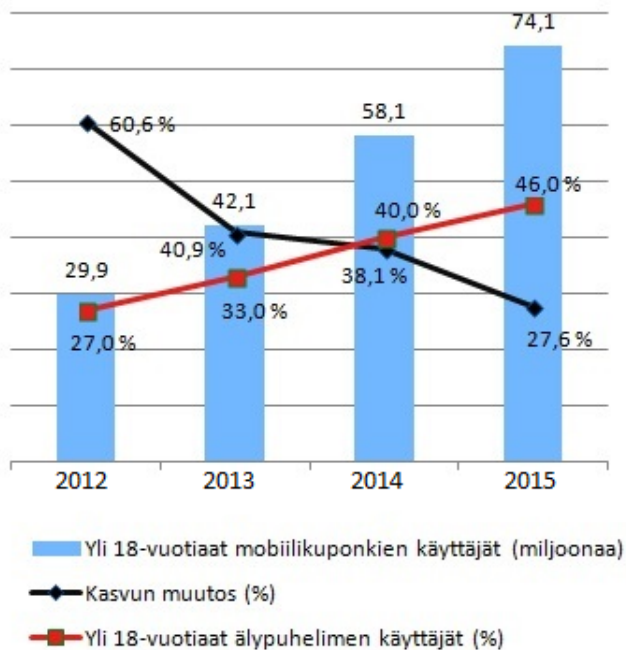
Ideja Oy:ssa selvitettiin mobiilimarkkinoiden mahdollisuuksia Suomessa. Selvityksessä tuli selville, että mobiilimarkkinointi on kasvanut Suomessa todella nopeasti viimeisten vuosien aikana. Vuonna 2013 Mainonnan neuvottelukunnan teettämän tutkimuksen mukaan mobiilimarkkinoiden arvo Suomessa oli 17 miljoonaa euroa ja mobiilimarkkinoiden arvo oli kasvanut 69 prosenttia edellisestä vuodesta [11]. Idejassa pääteltiin, että markkinoiden kasvuun on vaikuttanut suomalaisten älypuhelimien käyttäjien kasvaminen noin 70 prosenttiin. Toinen merkittävä vaikutus mobiilimarkkinoiden kasvuun uskottiin olevan Internetin lisääntyneellä käytöllä kodin ja työpaikan ulkopuolella. Kuviosta 1 nähtävän tilaston mukaan 53 prosenttia suomalaisista on käyttänyt Internetiä matkapuhelimella työn ja kodin ulkopuolella [12; 13, s.8].



Kuvio 1. Internetin käyttö matkapuhelimella, kannettavalla tietokoneella ja tabletilla muualla kuin kotona tai työpaikalla 2014 kaavio. [13, s.8.]

Mobiilikuponkien kasvupotentiaalia varten tehdyn selvityksen hyvät luvut Suomen mobiilimarkkinoiden kasvamisesta vahvistivat Idejan alkuperäisen uskomuksen mobiilimarkkinoiden hyvästä kannattavuudesta ja mahdollisuuksista. Idejassa tulkittiin lukujen pe-

rusteella, että kasvu tulee jatkumaan mobiilimarkkinoilla hyvänä ja yhä useammat ihmiset tulevat lisäämään Internetin käyttöä matkapuhelimilla. Kuviosta 1 näkyvästä tilastosta selviää 16–24-vuotiaiden nuorten muita ikäryhmiä isompi 90 prosentin käyttäjämäärä matkapuhelimen Internetin käytöstä kodin ja työpaikan ulkopuolella. Tämä tieto lisäsi Idejan luottamusta mobiilimarkkinoiden kasvuun myös pitemmällä aikavälillä. Idejassa oletettiin myös, että älypuhelimien määrä tulee jonkin verran kasvamaan nykyisestä määrästä sitä mukaan kuin loputkin kuluttajat uusivat vanhoja matkapuhelimiaan.



Kuvio 2. Yli 18-vuotiaat yhdysvaltalaiset mobiilikupongi käyttäjät, sekä niiden kasvun muutos ja älypuhelimien määrä vuosina 2012–2015.[14.]

Suomessa mobiilikupongeista ei ollut saatavilla minkäänlaisia tilastoja, koska mobiilikupongeja tarjoavia sovelluksia ei Suomen markkinoilla juuri ollut olemassa tai ne eivät olleet kovin suosittuja. Yhdysvalloista mobiilikupongien käytöstä on saatavissa tilastoja. Kuviosta 2 voi päätellä, että Yhdysvalloissa mobiilikupongien käyttö on nousussa ja niiden kasvuennuste on erinomainen. Idejassa Yhdysvaltojen tilastoja mobiilikupongien käytöstä ei pidetty suoraan takuuna mobiilikupongien menestys mahdollisuuksiin Suomessa, mutta se vahvisti Idejan olettamusta mobiilikupongien menestys mahdollisuuksista myös Suomessa. Tärkeintä Idejalle oli päästä mobiilikupongimarkkinoille ensimmäisten joukossa, jotta se pääsisi hyödyntämään mobiilikupongien oletettua menestystä.

## 4 Tuotteen vaatimukset ja käyttäjätarinat

Projektissa Ideja käytti omaa ketterään ohjelmistokehitykseen pohjautuvaa kehitystapaa. Ketterä ohjelmistokehitys käsittää joukon useita eri ohjelmistokehitysmenetelmiä [15]. Yhteistä näille on kehittää tuotetta sykleissä, jolloin potentiaalisesti valmista sovellusta pystytään esittelemään säännöllisesti [15]. Tällöin sovellukseen on mahdollista lisätä myös uusia ominaisuuksia kehityksen ollessa vielä käynnissä [15]. Idejassa käytetyn oman ketterään ohjelmistokehitykseen pohjautuvan kehitystavan vuoksi tuotteen vaatimuksia ei etukäteen juuri tehty, vaan vaatimukset tarkentuvat projektin edetessä [16]. Ketterissä menetelmissä projekti kannattaa aloittaa käyttäjätarinoiden tekemisellä [16]. Käyttäjätarina on toiminnallinen vaatimus, josta selviää kuka tekee, mitä tekee ja miksi tekee [16]. Idejan oma kehitystapa käsitellään myöhemmin luvussa 7.

### 4.1 Vaatimukset

Ecoupon-mobiilikuponkisovelluksen mainostajaksi haluavan toimijan piti olla Suomessa rekisteröity yritys. Ilman Suomessa rekisteröityä yritystä mainostajaksi halunnut toimija ei olisi voinut rekisteröityä mainostajan palveluun, jolla se olisi voinut lisätä yrityksen etuja Ecoupon-mobiilikuponkisovellukseen. Rekisteröityminen palvelun mainostajaksi olisi tapahtunut rekisteröitymistä varten tehdyllä web-lomakkeella. Rekisteröitymisen yhteydessä yrityksen olisi pitänyt täyttää oman yrityksen toimipisteen tiedot ja yrityksen yhteyshenkilön henkilökohtaiset tiedot. Rekisteröitymisen jälkeen ylläpidon olisi pitänyt tarkistaa uuden rekisteröityneen yrityksen tietojen oikeellisuus ja tarkistaa y-tunnuksen avulla yrityksen olevan olemassa. Vasta tämän jälkeen yritys olisi päässyt lisäämään etuja siihen erikseen tarkoitettulla sovelluksella. Epäilyttävissä rekisteröitymisissä, kuten rekisteröityneen yrityksen virheellisesti antamien tietojen vuoksi, ylläpidon olisi pitänyt suoraan ottaa yhteyttä yritykseen ja tarvittaessa vaatia yritykseltä lisätietoja.

Tulevaisuudessa mainostajien todentaminen oli tarkoitus hoitaa pankkitunnistautumisen avulla, jolloin mainostaja olisi päässyt heti rekisteröitymisen jälkeen lisäämään etuja Ecouponissa nähtäviksi. Todennuksen avulla pyrittiin vähentämään mainosten lisäämiseen mahdollisesti kohdistuvia väärinkäytöksiä. Esimerkiksi mainostajan web-sovellukseen olisi voinut rekisteröityä henkilö, joka olisi laittanut olemattoman yrityksen etuja Ecoupon-mobiilikuponkisovellukseen ilkeällä tavalla. Tämän lisäksi pahaa tahtova yri-

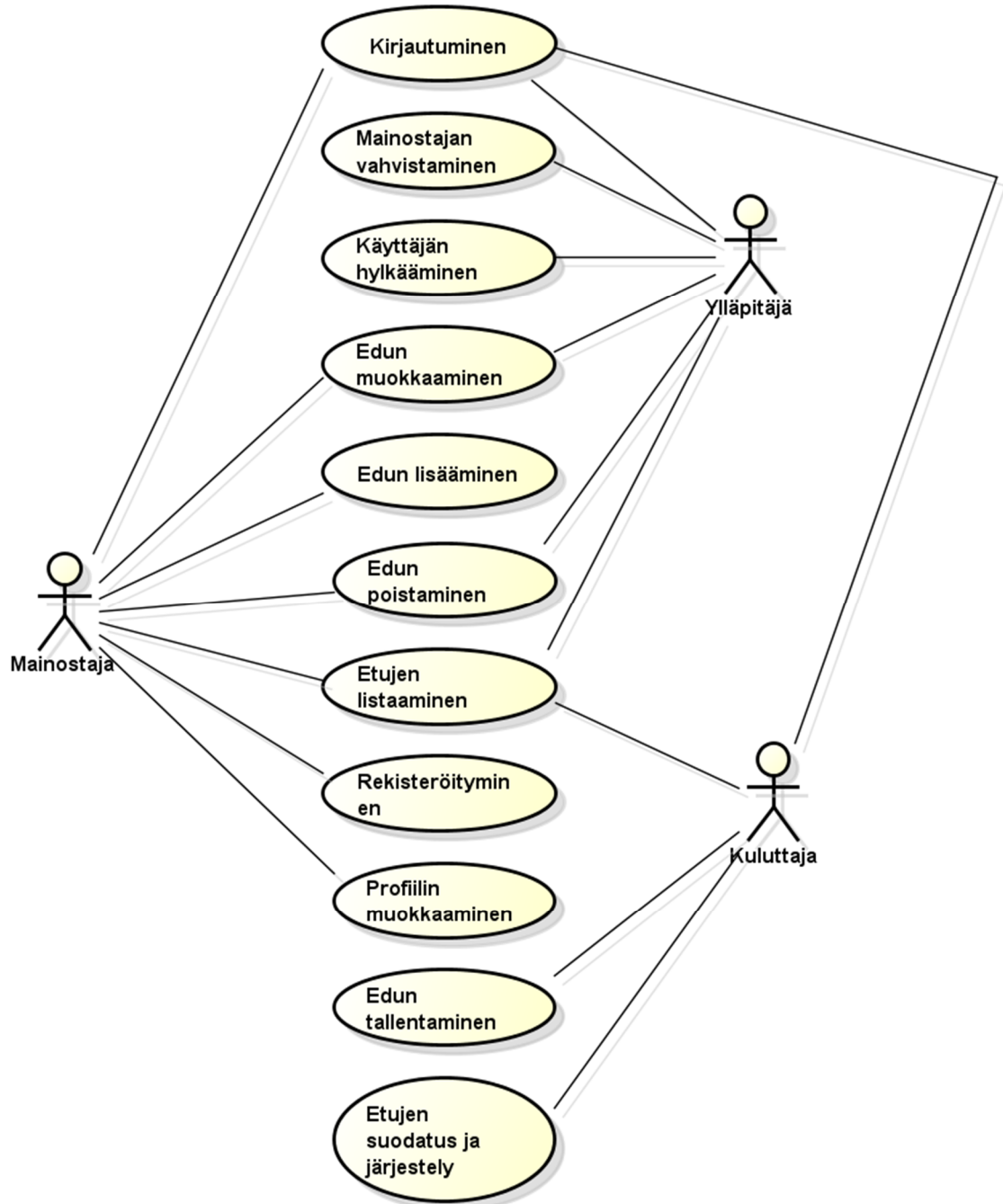
tys olisi voinut lisätä väärää etuja kilpailevan yrityksen nimissä omien etujen edistämiseksi. Tällainen tilanne olisi vähentänyt kuluttajan luottamusta ja palvelun uskottavuutta. Pahimmassa tapauksessa harhaan johtavasta mainonnasta olisi voinut joutua vastaamaan oikeusteitse.

#### 4.2 Käyttäjätarinat

Idejassa käytetyn ketterän ohjelmistokehitystavan tarkoituksena oli päästä mahdollisimman nopeasti kehittämään sovellusta. Käyttäjätarinoita jaettiin osiin, suunniteltiin ja spesifioitiin tarkemmin projektin edetessä. Tämä myös mahdollisti uusien ominaisuuksien lisäämisen, kun projekti eteni.

## Käyttötapauskaavio

Mainospalvelun käyttötapauskaavio on nähtävissä kuviossa 3, jossa näkyy eri käyttäjäroolien käyttötapaukset. Käyttötapauskaavio on tehty projektin käyttäjätarinoiden pohjalta.



Kuvio 3. Mainospalvelun käyttötapauskaavio.



#### 4.2.1 Ylläpitäjän käyttäjätarinat mainostajan web-käyttöliittymässä

Idejan projektiryhmän jäsenten oli tarkoitus toimia mainostajan web-käyttöliittymässä ylläpitäjinä. Projektin alussa ylläpitäjä-roolille kirjoitettiin käyttäjätarinat, joiden uskottiin olevan ylläpidon kannalta keskeisiä. Ylläpitäjälle tehdyt käyttäjätarinat ovat seuraavanlaisia:

##### *Kirjautuminen*

Ylläpitäjänä kirjaudun mainostajan web-käyttöliittymään, jotta voin ylläpitää sovelluksen etuja ja käyttäjiä.

##### *Mainostajan vahvistaminen*

Ylläpitäjänä vahvistan rekisteröityneen yrityksen mainostajaksi, jotta mainostaja voi käyttää sovellusta.

##### *Käyttäjän hylkääminen*

Ylläpitäjänä hylkään rekisteröityneen yrityksen, koska yrityksen tiedot ovat puutteelliset.

##### *Edun muokkaaminen*

Ylläpitäjänä muokkaan rekisteröityneen mainostajan etua, koska edussa on väärää tietoa.

##### *Edun poistaminen*

Ylläpitäjänä poistan rekisteröityneen mainostajan edun, koska etu on sopimaton.

##### *Etujen listaaminen*

Ylläpitäjänä listaan sovelluksen edut, jotta löydän ylläpidettävän edun.

#### 4.2.2 Mainostajan käyttäjätarinat mainostajan web-käyttöliittymässä

Yritysten, jotka olisivat lisänneet etuja Ecouponiin, oli tarkoitus toimia mainostajan web-käyttöliittymässä mainostajina. Projektin alussa projektiryhmässä kirjoitettiin mainostajaroolille käyttäjätarinat, joiden uskottiin olevan mainostajan kannalta keskeisiä. Mainostajalle tehdyt käyttäjätarinat ovat seuraavanlaisia:

##### *Rekisteröityminen*

Mainostajana täytän rekisteröitymislomakkeen, jotta saan oikeuden lisätä etuja Ecouponiin.

##### *Kirjautuminen*

Mainostajana kirjaudun mainostajan web-käyttöliittymään, jotta pääsen lisäämään etuja.

##### *Edun lisääminen*

Mainostajana lisään edun Ecouponiin, jotta saan sen kuluttajille nähtäväksi.

##### *Edun poistaminen*

Mainostajana poistan julkaisemattoman edun, koska edussa olevaa tuotetta ei ole saatavissa.

##### *Edun muokkaaminen*

Mainostajana muokkaan julkaisematonta etua, koska siinä on väärää tietoa.

##### *Etujen listaaminen*

Mainostajana listaan itse lisäämäni edut, jotta voin helposti selata lisäämiäni etuja.

##### *Profiilin muokkaaminen*

Mainostajana muokkaan oman profiilin tietoja.

#### 4.2.3 Kuluttajan käyttäjätarinat Ecoupon-mobiilikuponkisovelluksessa

Henkilöt, jotka olisivat käyttäneet Ecouponin etuja, oli tarkoitus kutsua kuluttajiksi. Projektin alussa projektiryhmä kirjoitti kuluttaja-roolille käyttäjätarinat, joiden se oletti olevan keskeisiä kuluttajille. Kuluttajalle tehdyt käyttäjätarinat ovat seuraavanlaisia:

##### *Etujen listaaminen*

Kuluttajana listaan mainostajien lisäämät edut, koska haluan nähdä edullisia tuotteita.

##### *Edun tallentaminen*

Kuluttajana kirjaudun Ecoupon-mobiilikuponkisovellukseen tallentamaan mainostajan lisäämän edun, koska olen kiinnostunut edusta ja voin käyttää sen myöhemmin.

##### *Etujen suodatus ja järjestely*

Kuluttajana valitsen suodattimia ja järjestystavan etujen listaamisessa, jotta löydän itseäni kiinnostavat edut.

##### *Kirjautuminen*

Kuluttajana kirjaudun Ecoupon-mobiilikuponkisovellukseen, jotta voin tallentaa etuja.

## 5 Toteutusratkaisut ja arkkitehtuuri

### Web-käyttöliittymän toteutustekniikat

Ennen web-käyttöliittymän toteutusta Idejassa kartoitettiin teknologioita, joilla käyttöliittymä voitaisiin toteuttaa. Lopulta päädyttiin toteuttamaan mainostajan käyttöliittymä web-sovelluksena, jotta kaikki sovellukseen mainostajaksi haluavat pystyisivät helposti lisäämään etuja useimmilla laitteilla ja milloin vain. Pelkkä mobiilisovellus olisi rajoittanut päätelaitteet ainoastaan älypuhelimiin ja tabletteihin, joilla olisi ollut myös hankala lisätä etuja. Myös työpöytäsovellusta ei pidetty sopivana tähän tarkoitukseen, koska mainostajilla oletettiin olevan käytössä useita erilaisia päätelaitteita. Työpöytäsovellus toimisi ainoastaan niillä työasemilla, jotka tukivat toteutusratkaisua. Tämän lisäksi työpöytäsovellus pitäisi aina ladata ja asentaa uudelle työasemalle, jolla etuja haluttaisiin lisätä Ecouponiin.

Vaihtoehtoina mainostajan sovelluksen toteutustekniikoiksi olivat PHP, C# ja Java. Projektin toteuttaminen C#-ohjelmointikielellä ei ollut kannattava vaihtoehto, koska kenelläkään projektiryhmän jäsenistä ei ollut sen käytöstä merkittävää kokemusta. Kokonaan uuden asian opetteleminen olisi hidastanut projektin aloitusta ja projektin mahdollisen myöhästymisen takia sen katsottiin olevan taloudellisesti kannattamaton vaihtoehto. Kaikki projektiryhmän jäsenet olivat käyttäneet Javaa aikaisemmissa projekteissa, mutta puolelle projektiryhmän jäsenistä Javan käyttö web-ohjelmoinnissa ei ollut tuttua. PHP-ohjelmointikielen käyttö oli projektiryhmälle entuudestaan tuttua, koska kaikki projektiryhmän jäsenet olivat käyttäneet sitä aiemmin Karhukopla-projektissa. Projektin toteutustekniikan varten otettaviksi vaihtoehdoiksi jäivät siis lopulta PHP ja Java. Lopullisen valinnan perusteeksi tuli projektiryhmän kokemus ja toteutusteknologioiden ohjelmistokehykset, joissa PHP-ohjelmointikielelle vaihtoehtoina olivat Zend- ja Symfony2- ohjelmistokehykset ja Javalta Spring Framework -ohjelmistokehys. Näihin kolmeen ohjelmistokehysvaihtoehtoon päädyttiin, koska niiden käytöstä oli kokemusta entuudestaan projektiryhmän sisällä. Lopullisessa valinnassa päädyttiin käyttämään Symfony2-ohjelmistokehystä, koska projektiryhmän mielestä se oli helposti omaksuttavissa hyvän dokumentaation ansiosta ja sillä oli iso yhteisö taustalla. Lisäksi jokaisella projektiryhmän jäsenellä oli aiempaa kokemusta PHP-ohjelmointikielen käytöstä web-ohjelmoinnissa, joten oli luontevaa käyttää projektin toteutuksessa PHP-ohjelmointikielellä toimivaa Symfony2-sovelluskehystä.

## Sovelluksen arkkitehtuuri

Sovellus rakentuu Symfony2-sovelluskehiksen sisältämään arkkitehtuuriin, jossa on hyödynnetty malli-näkymä-ohjain-arkkitehtuuria (model-view-controller, MVC). Malli-näkymä-ohjain-arkkitehtuurin ideana on käyttöliittymän ja ohjelman toimintalogiikan erottaminen toisistaan, jolloin esimerkiksi käyttöliittymää on helpompi muokata ja koodin luetavuudesta tulee selkeämpää. Arkkitehtuurissa on kolmenlaisia osia: mallit (model), näkymät (view) ja ohjaimet (controller). Ohjainten tehtävä on yhdistää mallit ja näkymät toisiinsa. [17; 18.]

## 6 Käytetyt teknologiat ja ohjelmistot

### Ohjelmistokehys

Ohjelmistokehystenä projektissa käytettiin kansainvälisesti tunnettua ja tunnustettua Symfony2-ohjelmistokehystä web-projekteille. PHP-ohjelmointikielelle tehty Symfony2-ohjelmistokehys on nopeasti omaksuttavissa ammattimaiseen käyttöön ja sen kehittämisen taustalla on SensioLabs yhtiö. SensioLabs on vuonna 1998 perustettu ranskalainen ohjelmistotalo. SensioLabsin Symfony2:lle tarjoaman pitkäaikaistuen lisäksi Symfony2:lle on mahdollista saada tukea esimerkiksi postituslistojen ja IRC:n kautta. [19.]

### Olio-relaatio-mallinnus

Projektissa käytettiin PHP-ohjelmointikielelle tehtyä Doctrine ORM olio-relaatio-mallinnusta (Object-relation mapper), jonka kautta olioiden mallit tallennetaan relaatiotietokantoihin kuten MySQL, PostgreSQL tai Microsoft SQL. Doctrine ORM toimii Doctrine tietokanta-abstraktiotason (Database Abstraction Layer, DBAL) päällä, joka abstrahoi PHP:n tietokantaohjelmointimallin (PHP Data Objects, PDO). PDO on rajapinta, jota hyödyntämällä pystytään käyttämään eri tietokantoja PHP:lla toteutettujen sovellusten tietojen tallentamiseen. [20.]

### Tehtävienhallintaohjelmisto

Projektin käytössä oli vuonna 2002 perustetun australialaisen Atlassian-yrityksen tekemä Jira-tehtävienhallintaohjelmisto, joka on käytössä tuhansissa ohjelmistoalan projekteissa ympäri maailman [21.] Jirassa kirjataan projektin tehtävistä tikettejä, joiden avulla tehtäviä on helppo hallita.

Jiran käytössä Ideja käytti työnkulkua, jonka se oli kokenut projektiryhmän aiemmissa projekteissa hyväksi ja toimivaksi. Idejassa käytetyn työnkulun keskeisenä pyrkimyksenä oli saada varmistettua mahdollisimman hyvä koodin laatu.

## Jiran työnkulku vaiheittain

### 1. *Suunnittelu*

Jirassa luotiin projektissa toteutettavista ominaisuuksista tiketit. Jira-tiketin toteutus suunniteltiin suunnitelupalaverissa, minkä jälkeen se laitettiin tikettijonoon. Tikettijono sisälsi Jirassa tehtyjä tikettejä projektin toteuttamattomista ominaisuuksista. Tikettijonosta nostettiin seuraavaksi tehtäviä Jira-tikettejä niiden prioriteetin mukaan tai silloin, kun ne olivat projektin etenemisen kannalta parasta toteuttaa.

### 2. *To do*

Kehittäjä laittoi Jira-tiketin To do -tilaan silloin, kun kehittäjä otti Jira-tiketin omaan työjonoon.

### 3. *In progress*

Kehittäjä laittoi tiketin In progress -tilaan silloin, kun aloitti tekemään Jira-tiketin toteutusta.

### 4. *In testing*

Kun kehittäjä oli saanut toteutettua Jira-tiketin, hän siirsi sen testattavaksi In testing -tilaan. Tämän jälkeen toinen kehittäjä pystyi ottamaan In testing -tilassa olevan Jira-tiketin testattavaksi. Jira-tiketin toteuttanut kehittäjä ei saanut ottaa omaa tuotosta testattavaksi. Jos testauksessa ei löytynyt Jira-tiketin toteutuksesta vikaa, se suljettiin. Mikäli toteutuksessa löydettiin In testing -tilassa vikaa, Jira-tiketti palautettiin To do -tilaan alkuperäiselle kehittäjälle ja siihen kommentoitiin testauksessa löydetyt puutteet.

## Versionhallintajärjestelmä

Versionhallintajärjestelmänä Ideja käytti vapaan lähdekoodin Apache Subversion -versionhallintajärjestelmää, jonka käyttäminen tapahtui sille tehdyn TortoiseSVN-käyttöliittymän kautta [22; 23]. TortoiseSVN tarjoaa Apache Subversion -versionhallintajärjestel-

mään mukavan ja helpon käyttöliittymän [23]. TortoiseSVN on kehitetty vapaiden ohjelmistojen julkaisemiseen tarkoitettun GPL-lisenssin alla ja se on käytettävissä myös kaupallisessa tarkoituksessa [23]. Sitä ei ole tarkoitettu käytettäväksi millään yksittäisellä ohjelmointiympäristöllä, ja sen sai liitettyä Idejan valitseman NetBeans-kehitysympäristöön [24]. TortoiseSVN-käyttöliittymän etuna oli sen selkeä käyttöliittymä ja kaikkien projektiryhmän jäsenien aikaisempi kokemus sen käytöstä.

### Kehitysympäristö

Kehitysympäristönä toimi Oraclen hallinnoima vapaan lähdekoodin NetBeans-kehitysympäristö [25]. Siitä löytyy SVN-tuki ja hyvät työkalut PHP-ohjelmointikielellä työskentelemään varten [24; 25]. Projektiryhmällä oli pitkäaikainen kokemus NetBeansin käytöstä kehitysympäristönä, joten se oli luonnollinen valinta kehitysympäristöksi tässä projektissa.

### Käyttöliittymätekniikka

Sovelluksen käyttöliittymän tekemisessä hyödynnettiin Foundation responsive frontend-ohjelmistokehystä, jolla tehdyt web-sivujen käyttöliittymät ovat helposti skaalattavissa mobiililaitteille. Foundation on vapaan lähdekoodin projekti, jonka kehittämisestä vastaa yhdysvaltalainen yritys nimeltä Zurb. [26; 27.]

Foundationissa on käytetty hyödyksi JavaScript jQuery-kirjastoa, css-tyylimäärittelyä ja html-merkintäkieltä. Sen hyöty käyttöliittymänkehityksessä perustuu Foundationissa valmiina oleviin käyttöliittymäkomponentteihin. Valmiita käyttöliittymäkomponentteja käyttämällä kehittäjän ei tarvitse käyttää niin paljon resursseja käyttöliittymän tekemiseen kuin perinteisellä tavalla. Tällöin säästyy paljon kehittämisessä käytettävää aikaa, kun käyttöliittymän jokaista elementtiä ei tarvitse erikseen suunnitella. [27.]

### Yksikkötestit

Sovelluksen tekemisessä hyödynnettiin yksikkötestejä, joiden tarkoituksena oli varmistaa sovelluksen lähdekoodissa esiintyvien metodien sekä funktioiden toimivuus ja laatu. Yksikkötestit tehtiin PHP-ohjelmointikielelle tehdyllä PHPUnit-yksikkötestausohjelmistokehyksellä, jonka on tehnyt saksalainen Sebastian Bergmann. PHPUnit pohjautuu xUnit-arkkitehtuuriin, jota käytetään useissa yksikkötestausohjelmistokehyksissä. [28.]



Yksikkötestejä käytettiin sovelluksen luokkien metodien testaamiseen. Tarkoituksena oli tehdä vähänkin monimutkaisille luokkien metodeille testit, ennen kuin niitä alettiin toteuttaa. Testin toteutuksesta vastaavan ei itse tarvinnut päättää, pitkö yksikkötesti tehdä uudelle luokan metodille, vaan tämä asia oli päätetty aiemmin suunnitelupalaverissa, jossa toteutettavat ominaisuudet spesifioitiin tarkasti.

```
/**
 * @Assert\True(message="benefit.endDate.mustBeLaterThanStartDate")
 * @return boolean
 */
public function isEndDateLaterThanStartDate()
{
    return $this->endDate === null || $this->startDate === null || $this->endDate >= $this->startDate;
}
```

Kuvio 4. Esimerkki edun metodista, jossa palautetaan tieto siitä, onko edun päättymisaika myöhemmin kuin aloitusaika.

```
/**
 * @test
 */
public function entity_should_fail_validation_if_end_date_is_earlier_than_start_date()
{
    $benefit = $this->getFixtureFactory()->getUnpersisted('BenefitBundle\Entity\Benefit', array(
        'startDate' => new DateTime('2112-01-02'),
        'endDate' => new DateTime('2112-01-01'),
        'publishDate' => new DateTime('2111-01-01'),
        'placesOfBusiness' => $this->getPlacesOfBusiness(),
    ));

    $this->assertNotValidEntity($benefit);
}
```

Kuvio 5. Esimerkki yksikkötestistä, jossa odotetaan validoinnin epäonnistumista, jos edun päättymisaika on ennen aloitusaikaa.

Kuviossa 4 on nähtävissä toteutus liitteessä 1 tarkemmin esiteltävän Benefit-luokan metodista `isEndDateLaterThanStartDate()`. Tällä metodilla tarkastetaan, onko edun päättymisaika myöhemmin kuin aloitusaika. Kuviossa 5 on metodia `isEndDateLaterThanStartDate()` varten luotu yksikkötesti `entity_should_fail_validation_if_end_date_is_earlier_than_start_date()`. Yksikkötestissä haetaan metodilla `getFictureFactory()` instanssi edusta `$benefit`-muuttujaan. Samalla instanssille annetaan tarvittavat arvot testaamista varten. Tässä testissä instanssille annetaan aloitusaika, päättymisaika, julkaisuaika ja toimipisteet. Toimipisteet annetaan instanssille, koska `FictureFactory` ei saa itse luotua niitä. Lopulta `$benefit`-muuttujaan tallennettu instanssi annetaan metodille `assertNotValidEntity()`. Tämä metodi tarkistaa edun `isEndDateLaterThanStartDate()` metodin assertin avulla, että instanssi ei ole validi etu.

## 7 Projekt

Projektissa työskenteli kuusi henkilöä, joista viidellä henkilöllä oli yhteistä kokemusta ohjelmien tekemisestä ja suunnittelusta. Nämä viisi henkilöä olivat aluksi vastuussa tuotteen teknisestä toteuttamisesta. Tämän lisäksi projektissa oli mukana henkilö, jolla oli aiempaa kokemusta yrittämisestä. Hän keskittyi tuotteen markkinoimiseen ja yrityksen juoksevien asioiden hoitamiseen.

Kaikki tuotteen toteuttamisessa mukana olleet projektiryhmän jäsenet olivat projektin alussa mukana suunnittelemassa Ecoupon-mobiilikuponkisovelluksen ja sen mainostajan web-käyttöliittymän ominaisuuksia. Tätä suunnitteluvaihetta kesti noin kaksi viikkoa, ennen kuin projektiryhmässä siirryttiin suunnittelun ohessa tekemään tuotteen toteutusta.

Projektin toteutuksessa käytettiin Idejan omaa ketteriin kehitysmenetelmiin pohjautuvaa kehitysmenetelmää. Käytössä oli projektiryhmän aiemmissa projekteissa käytetty kehitysmenetelmä, jossa kerran viikossa suunnittelupalaverin yhteydessä valittiin sprintteihin toteutettavia Jira-tikettejä. Suunnitteluvaiheessa pyrittiin siihen, että Jira-tiketit olivat toteuttamiskelpoisia ja jokainen teknistä toteutusta tekemässä ollut projektiryhmän jäsen pystyi ne toteuttamaan.

### Viikkopalaveri

Viikkopalaveri oli kerran viikossa pidettävä palaveri, jossa käytiin läpi sovelluksen ominaisuuksia. Viikkopalaverissa pohdittiin seuraavaksi projektissa toteutettavia ominaisuuksia ja ominaisuuksia, jotka eivät toimineet tai joita ei ollut järkevä toteuttaa. Näiden lisäksi viikkopalaverissa mietittiin kokonaan uusia ominaisuuksia, jotka tuotteen kehityksestä puuttuivat, mutta olivat olennaisia sovelluksen toiminnan kannalta. Tämän lisäksi viikkopalaverissa pohdittiin tuotteen markkinointia ja yritykseen liittyviä hallinnollisia asioita.

## Sprintti

Sprintti on enintään kuukauden pituinen aikaraja, jonka sisällä tuotetaan “valmiin” määritelmän täyttävä, käyttökelpoinen ja potentiaalisesti julkaisukelpoinen tuoteversio [29, s.7]. Idejan projektissa päätettiin pitää sprinttejä, koska haluttiin puuttua kehityksessä esiintyviin mahdollisiin ongelmiin ennen kuin ne kasvaisivat liian suuriksi tai hidastaisivat huomattavasti projektin etenemistä. Sprintti kesti yleensä noin viikon, jonka aikana tehtiin ainoastaan sprinttiin otettuja Jira-tikettejä. Sprintin päätyttyä käytiin läpi, mitä kukin projektin ohjelmoinnissa mukana ollut jäsen oli saanut tuotettua. Mikäli sprintin aikana havaittiin ongelmia syntyneen, ongelmiin pystyttiin puuttumaan välittömästi.

## Suunnittelupalaveri

Suunnittelupalaveri pidettiin viikoittain, johon osallistuivat kaikki projektiryhmän jäsenet. Suunnittelupalavereissa suunniteltiin sovelluksen uusia ominaisuuksia, sekä kirjoitettiin näistä tarkat spesifioinnit Jira-tiketteihin. Näin jokainen projektiryhmän toteutuksen kanssa työskennellyt henkilö osasi ominaisuudet tarvittaessa toteuttaa. Tämän lisäksi suunnittelupalaverissa valittiin seuraavan viikon sprinttiin valmiiksi spesifioituja Jira-tikettejä.

## Hyväksymispalaverien käyttöönotto ja markkinointiin keskittyminen

Kun projekti oli ollut käynnissä noin neljä kuukautta, haluttiin projektin toimintaa tehostaa. Tähän päädyttiin, koska projekti tuntui etenevän liian hitaasti ja kehityksen puolivälissä Ideja löysi useampia vastaavanlaisia mobiilikuponkisovelluksia. Projektiryhmä muutti työskentelytapoja niin, että viikkopalaveri pidettiin tarvittaessa ja suunnittelupalaveri jaettiin kolmeen osaan.

Suunnittelupalaverin ensimmäisessä osassa kaksi henkilöä tarvittaessa suunnittelivat ominaisuuksia ja valitsivat valmiiksi suunniteltuja ominaisuuksia seuraavaan sprinttiin. Tämän jälkeen kaksi henkilöä tekivät tarkat spesifioinnit sprinttiin valituista ominaisuuksista niin, että ne olivat toteuttamiskelpoisia. Lopuksi pidettiin hyväksymispalaveri, jossa käytiin läpi seuraavassa sprintissä toteutettavat ominaisuudet. Tämä mahdollisti sen, että ominaisuuksien suunnittelussa ja spesifioinnissa ei tarvinnut olla kaikkien projektin jäsenien paikalla. Näin resursseja pystyttiin siirtämään itse tuotteen toteutuksen tekemiseen.

Hyväksymispalaverin tarkoituksena oli, että ominaisuuksien valinta- ja spesifointipalaverien ulkopuolelle jääneet projektiryhmän jäsenet saisivat tietoa uusista suunnitelluista ominaisuuksista. Siinä valinta- ja spesifointipalaverien ulkopuolelle jääneet henkilöt hyväksyivät uudet toiminnallisuudet sellaisinaan tai pystyivät kysymään lisätietoja ja antamaan parannusehdotuksia suunnitelluista uusista ominaisuuksista.

Suunnittelupalaverin muutosten lisäksi projektiryhmä siirsi myös toisen projektiryhmän jäsenen tekemään markkinointitöitä. Tähän vaikutti markkinointiin laitettut vähäiset resurssit, rahoittajan puuttuminen ja yrityksille tehtävän markkinoinnin hidas edistyminen.

### Oma osuuteni projektissa

Oma panokseni Ecoupon-mobiilikuponkisovelluksen toteuttamisessa oli pääasiallisesti toteuttaa mainostajan web-liittymään suunniteltuja ominaisuuksia. Olin projektin alussa aina mukana suunnittelupalavereissa, mutta projektin myöhemmässä vaiheessa vähemmän. Syynä tähän oli Idejan pyrkimys tehostaa projektin työskentelyä. Projektissa pyrityn tehostamisen jälkeen keskityin suurimmaksi osaksi mobiilikuponkisovelluksen mainostajan web-käyttöliittymän toteutuksen tekemiseen. Osallistuin myös hyväksymispalaveriin, joissa hyväksyttiin suunnittelupalavereissa suunnitellut ominaisuudet, jotka aiottiin ottaa seuraavaan sprinttiin. Ennen projektin loppumista olin toteuttamassa sosiaalisen median markkinointia, jota kautta projektiryhmä aikoi markkinoida mobiilikuponkisovellusta. Näiden lisäksi tein useita erilaisia selvityksiä asioista, joita projektiryhmä olisi voinut hyödyntää sovelluksen toteutuksessa projektin myöhemmässä vaiheessa. Näitä olivat mainostajan tunnistautuminen pankkitunnuksilla, XHTML:n käyttöönotto ja Google Maps -integraatio.

## **8 Mainostajan web-käyttöliittymän toteutus**

Mainostajan web-käyttöliittymän ensimmäisen versio ehdittiin tehdä valmiiksi lähes kokonaan Idejassa ennen kuin projektista päätettiin luopua. Käyttöliittymän toteutuksen keskeisimmät ominaisuudet saatiin tehtyä valmiiksi kokonaisuudessaan.

### Rekisteröityminen

Ennen kuin yritys olisi pystynyt lisäämään omia etuja Ecoupon-mobiilikuponkisovellukseen, olisi yrityksen pitänyt rekisteröityä mainostajaksi sovelluksessa. Rekisteröitymisessä mainostajaksi yrityksen edustajan olisi pitänyt täyttää rekisteröitymislomake, jonka pohjalta ylläpito olisi tehnyt päätöksen, hyväksytäänkö yritys mainospalvelun mainostajaksi tai pyydetäänkö yritykseltä lisätietoja rekisteröitymiseen liittyen. Viime kädessä yrityksen rekisteröityminen mainostajaksi olisi voitu hylätä, mikäli yritys ei olisi vaikuttanut luotettavalta.

Rekisteröidy  
 Kirjaudu sisään

**Henkilötiedot**

Etunimi *	Sähköpostiosoite *
Sukunimi *	Salasana *
Puhelinnumero *	Salasana uudestaan *

**Yrityksen tiedot**

Virallinen nimi (esim. HOK-Elanto) *	Y-tunnus *
--------------------------------------	------------

**Liikkeen tai toimipisteen tiedot**

Nimi (esim. Alepa, Suvela) *	WWW-sivu	
Puhelinnumero	Katuosoite *	
Sähköpostiosoite	Postinumero *	Toimipaikka *
Aukioloajat		
Kuvaus		
Logo		
<input type="button" value="Selaa..."/> Ei valittua tiedostoa.		

**Käyttöehdot**

Olen lukenut ja hyväksyn käyttöehdot \*

Kuva 1. Mainostajan rekisteröitymislomakkeen näkymä.

Sovelluksessa rekisteröitymiseen vaadittavat pakolliset tiedot näkyvät kuvassa 1 punaisina tähtinä. Lomakkeen kenttien tiedot tarkistetaan käyttöliittymän sekä palvelimen puolella. Lisäksi osaan lomakkeiden kentistä tehtiin omat filtrit, joilla tarkastettiin kenttiin syötettävät arvot. Tällä pyrittiin estämään käyttäjää syöttämästä virheellisiä arvoja sovellukseen. Kuvassa 2 rekisteröityminen ei mene läpi, koska käyttäjä on syöttänyt y-tunnuksen väärässä muodossa.

Y-tunnus \*

2323-2

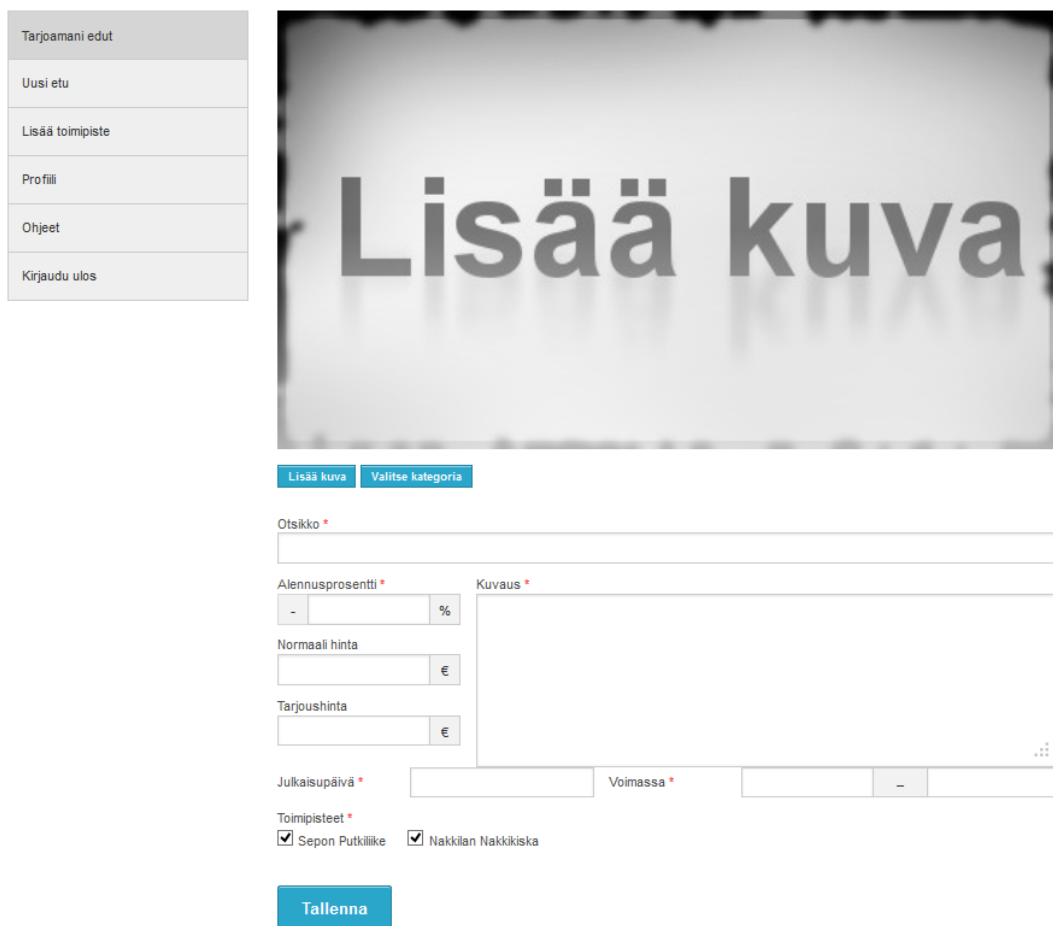
Annettu arvo ei ole kelvollinen Y-tunnus.

Kuva 2. Mainostajan rekisteröitymislomakkeen y-tunnus-kenttä.

## Edun lisääminen

Kun yritys olisi rekisteröitynyt sovellukseen ja ylläpito olisi hyväksynyt yrityksen sovelluksen mainostajaksi, olisi yritys pystynyt lisäämään etuja. Edut ovat tarjouksia tai kuponkeja, jotka olisivat näkyneet kuluttajapuolen mobiilisovelluksessa.

Tarjouksen lisääminen sovellukseen tapahtuu kirjautumalla mainostaja käyttäjänä ja valitsemalla kuvan 3 näkyvästä valikosta ”Uusi etu” ja sen alapuolelle aukeavan valikon kohdasta ”Tarjous”. Tarjousta lisättäessä mainostaja pystyy määrittelemään tarkasti mainoksen tiedot kuvan 3 näkyvän lomakkeen mukaisesti.



The screenshot shows a web interface for adding a new offer. On the left is a sidebar menu with the following items: Tarjoamani edut, Uusi etu, Lisää toimipiste, Profiili, Ohjeet, and Kirjaudu ulos. The main content area features a large image placeholder with the text 'Lisää kuva'. Below the image are two buttons: 'Lisää kuva' and 'Valitse kategoria'. The form fields include: Otsikko (Title), Alennusprosentti (Discount percentage) with a minus sign and percentage symbol, Normaali hinta (Normal price) and Tarjoushinta (Offer price) both with Euro symbols, Julkaisupäivä (Release date) and Voimassa (Valid until) with a minus sign, and two checkboxes for 'Toimipisteet' (Locations): 'Sepon Putkiliike' and 'Nakkilan Nakkikiska'. A 'Tallenna' (Save) button is at the bottom.

Kuva 3. Tarjouksen lisäämisen näkymä mainostajan web-käyttöliittymässä.

Kupongin lisääminen tapahtuu lähes samalla tavalla kuin tarjouksen lisääminen, mutta siinä on pieniä eroja. "Uusi etu"-kohdan alapuolelle aukeavan valikon kohdasta valitaan "Kuponki". Kupongin ympärillä näytetään kuvassa 4 nähtävällä tavalla katkoviivat ja saksien kuva. Saksien ja katkoviivojen näkymisen tarkoituksena on helpottaa erottamaan kuponki tarjouksesta. Tämän lisäksi kuponki eroaa tarjouksesta siten, että mainostaja voi lisätä rajoituksen jaossa olevien kuponkien kappalemäärän.

Kuva 4. Kupongin lisäämisen näkymä mainostajan web-käyttöliittymässä.

Mainostajalla on myös mahdollisuus muokata omia etuja ennen kuin tarjous tai kuponki on julkaistu. Edun voi myös poistaa ennen kuin se on julkaistu, mutta julkaisun jälkeen mainostaja ei voi enää itse poistaa tai muokata etua. Tällöin kuluttaja ei olisi tullut harhautetuksi eduilla, joita olisi muutettu kesken edun voimassaoloajan. Ainoa tapa poistaa etu julkaisun jälkeen, olisi ollut yhteydenotto sovelluksen ylläpitoon.



## Toimipisteen lisääminen

Mainostajalla voi Idejan mobiilikuponkisovelluksessa olla useita toimipisteitä. Toimipisteen lisääminen tapahtuu kuvassa 5 nähtävällä lomakkeella, jonka kirjautunut mainostaja saa näkyviin oikealla puolella näkyvästä navigaatiopalkin valinnasta ”Lisää toimipiste”. Lisätessä toimipistettä mainostajan pitää täyttää vähintään punaisella tähdellä merkityt pakolliset tiedot, joita ovat toimipisteen nimi ja osoitetiedot. Mainostajalla on myös mahdollisuus lisätä toimipisteelle oma logo.

Tarjoamani edut	Nimi (esim. Alepa, Suveila) *	WWW-sivu	
Uusi etu	Puhelinnumero	Katuosoite *	
Lisää toimipiste	Sähköpostiosoite	Postinumero *	Toimipaikka *
Profiili	Aukioloajat		
Ohjeet	Kuvaus		
Kirjaudu ulos	Logo <input type="button" value="Selaa..."/> Ei valittua tiedostoa. <input type="button" value="Lisää"/>		


Kuva 5. Toimipisteen lisäämisen näkymä mobiilikuponkisovelluksessa.

## Etujen listaaminen

Navigaatiovalikon kohdassa ”Tarjoamani edut” mainostajalle listataan kaikki voimassa-olevat edut, jotka mainostaja on itse lisännyt yritykselle. Edut jaotellaan julkaisemattomien ja julkaistujen etujen mukaan kuvassa 6 nähdyllä tavalla. Julkaisemattomat edut näkyvät listauksessa ensimmäisenä, koska mainostaja pystyy muokkaamaan näiden etujen tietoja. Julkaistujen mainosten tietoja mainostaja ei voi enää muokata. Mainostaja voi myös erikseen listata itse lisäämänsä julkaisemattomat, julkaistut tai erääntyneet edut.

Tarjoamani edut
<a href="#">Julkaisemattomat</a>
<a href="#">Julkaistut</a>
<a href="#">Erääntyneet</a>
Uusi etu
Lisää toimipiste
Profiili
Ohjeet
Kirjaudu ulos

### Julkaisemattomat




Syno 55\" Smart TV  
1600,00 €

-40%

---

### Julkaistut



Legendaarinen K-Espoo muki  
30,00 €

-30%

Kuva 6. Etujen listauksen näkymä mainostajan web-käyttöliittymässä.

## Ylläpidon toiminnallisuudet

Ylläpidolla olisi ollut mahdollisuus hylätä rekisteröityneitä yrityksiä ja poistaa etuja, jotka esimerkiksi eivät olisi täyttäneet keskuskauppakamarin mainonnan eettisen neuvoston sääntöjä. Sovelluksessa mainostajana toimiva yritys olisi voinut lähettää ylläpidolle pyynnön poistaa virheellinen etu, mikäli yritys olisi esimerkiksi vahingossa lisännyt edun väärälle päivämäärälle. Mainostaja-käyttäjäroolissa ei voi poistaa sovelluksessa voimassa nähtäviä etuja. Tarkoituksena oli estää sovelluksen väärinkäyttöä ja sitä ettei kuluttajat olisi tulleet hämääntyneiksi etujen katoamisista palvelusta.

Vahvistamattomat mainostajat

---

Yritykset


---

Ohjeet

---

Kirjaudu ulos

### Julkaisemattomat



-40%

**Syno 55" Smart TV**  
1600,00-€

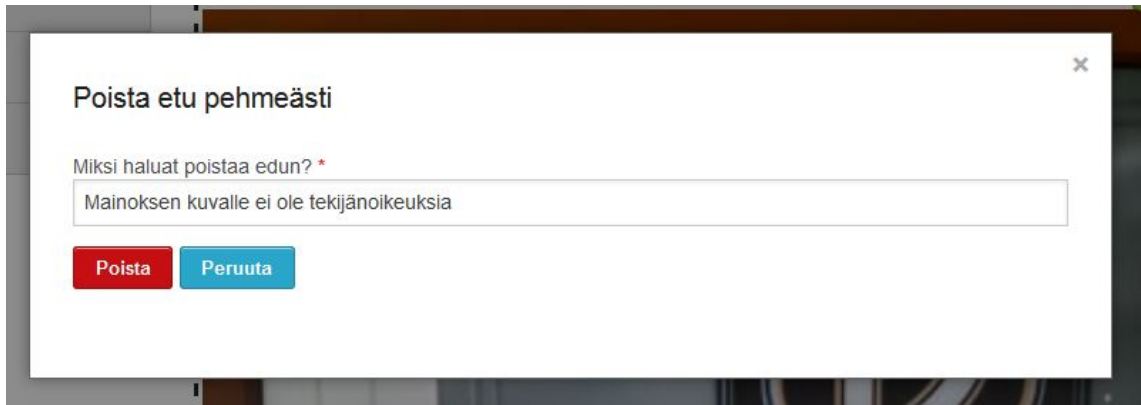
✂

<b>Voimassa</b>	<b>Julkaisupäivä</b>	<a href="#">✕</a>
27.03.2015 – 31.03.2015	19.03.2015	<a href="#">Poista pehmeästi</a>
<b>Kategoria</b>	<b>Toimipisteet</b>	
Elektroniikka > Televisiot	Sepon Putkiliike Nakkilan Nakkikiskka	
<b>Kuvaus</b>		
Laadukas Syno 55" Smart TV 4K-resoluutiolla.		

Kuva 7. Edun näkymä mainostajan web-käyttöliittymässä ylläpitäjänä.

Ylläpitäjä näkee mainostajien kaikki edut ja tarvittaessa voi tarkastella etuja tarkemmin. Kuvassa 7 on ylläpitäjän näkymä, josta näkyy yrityksen lisäämän edun tiedot. Samassa näkymässä ylläpitäjä voi poistaa edun tarpeen vaatiessa. Ylläpitäjä-käyttäjäroolissa tehtävää edun poistamista kutsuttiin pehmeäksi poistamiseksi, sillä pehmeästi poistamalla etu ei poistunut palvelusta kokonaan. Idejassa ajateltiin, että etua ei ollut hyvä poistaa kokonaan palvelusta. Silloin kuluttaja olisi voinut hämmentyä, ettei löydä aiemmin näke-

määnsä tai tallentamaansa etua. Kuvassa 8 ylläpitäjä on painanut edun pehmeää poistoa ja esille on tullut näkymä, jossa ylläpitäjältä pyydetään vahvistus edun poistamiselle. Vahvistuksessa ylläpitäjä voi vielä peruuttaa edun poistamisen tai antaa syyn edun poistamiselle ja poistaa edun. Kun etu on poistettu pehmeästi, edun kuva sumenee ja sen päälle tulee punainen viiva, jossa selitetään miksi etu on poistettu. Näin kuluttajalle tai mainostajalle ei olisi jäänyt epäselvyyttä mistä syystä etu on poistettu palvelusta.



Poista etu pehmeästi

Miksi haluat poistaa edun? \*

Mainoksen kuvalle ei ole tekijänoikeuksia

Poista Peruuta

Kuva 8. Edun poistamisen näkymä web-käyttöliittymässä ylläpitäjänä.

## 9 Yhteenveto

Mobiilikuponkisovelluksen ideaa pidettiin yleisesti toimivana, ja se sai paljon kehuja projektiryhmän teettämässä kyselyssä. Kyselyssä selvitettiin ihmisten mahdollista kiinnostusta käyttää Ecouponia. Myös sovelluksessa käytetyt tekniikat olivat huolellisesti valittuja ja tuotetun koodin laatu saatiin pidettyä hyvänä. Tähän vaikutti Idejassa käytössä olleet työskentelytavat, jotka olivat tarkasti määriteltyjä. Projektityöskentely oli usein hidasta. Tämä johtui projektiryhmän lähes päivittäisistä palavereista, jotka venyivät usein puolen päivän mittaisiksi. Tämä ei kuitenkaan ollut iso ongelma projektissa. Isompi ongelma oli, että projektiryhmällä ei ollut tarpeeksi markkinointiosaamista ja taloudellista pääomaa sovelluksen markkinointia varten. Projektiryhmä haki tukea Keksintösäätiöltä, mutta sieltä sanottiin, että pelkälle hyvälle it-sovellukselle tai sen idealle ei ole mahdollista saada patenttia. Patentti oli taas edellytys rahallisen tuen saamiselle Keksintösäätiöltä. Ainoa asia, mihin it-sovelluksessa voi saada patentin, on riittävän monimutkainen algoritmi. Keksintösäätiön lisäksi Ideja selvitti mahdollisuutta saada tukea muista paikoista ja harkitsi mahdollisuutta yrittää saada yksityinen rahoittaja projektille. Nämä vaihtoehdot eivät kuitenkaan edenneet selvitystyötä pidemmälle.

Tuotteen menestymisen heikot näkymät ja löydetyt kilpailijat vaikuttivat negatiivisesti projektiryhmän toimintaan. Tämä näkyi projektiryhmän motivaation laskuna ja sovelluksen kehityksen hidastumisena. Lopulta projektiryhmän kesken pidettiin kokous, jossa Ecouponin kehitys päätettiin lopettaa.

Projektin aikana Ideja sai valmiiksi koko järjestelmän kattavan suunnitelman, lähes julkaisuvalmiin mainostajan käyttöliittymän ja pääsi alkuun Ecoupon-mobiilikuponkisovelluksen tekemisessä. Markkinoinnissa Ideja sai tehtyä selvityksen mahdollisista kilpailijoista ja käytti sosiaalista mediaa avuksi saadakseen eduista kiinnostuneita ihmisiä seuraamaan Idejan Facebook-yhteisöä. Tämän lisäksi Ideja sai useita yrityksiä esirekisteröitymään Ecouponin käyttäjiksi, kun se jalkautui esittelemään Ecoupon-mobiilikuponkisovelluksen ideaa eri yrityksille.

Mikäli Ecoupon olisi valmistunut ja tullut suosituksi, se ei olisi ollut suojassa isommilta toimijoilta. Mikään ei estä isompaa toimijaa kopioimasta pienemmän toimijan suosittua tuotetta ja sen liikeidea. Tämän jälkeen isompi toimija voisi hyödyntää omaa isompaa markkinointikoneistoa ja ottaa sen avustuksella markkina-alueen haltuun. Projektin alussa projektiryhmässä tehtiin liian optimistisia arvioita sovelluksen menestymisen

mahdollisuuksista. Lopullisissa laskelmissa projektiryhmässä päädyttiin siihen, ettei sovellus olisi voinut toimia pääasiallisena tulonlähteenä Suomen markkina-alueella projektiryhmän kaikille henkilöille. Mikäli Ecouponista kuitenkin olisi saatu kannattava, olisi se tarvinnut ulkopuolista rahoitusta ja se olisi pitänyt Suomen lisäksi lanseerata ulkomailla. Pelkkä hyvin tehty potentiaalinen tuote, ei pärjää kuluttajamarkkinoilla, missä isot yritykset ovat vahvoilla.

## Lähteet

- 1 Groupon Works. 2012. Verkkodokumentti. Groupon, Inc. <<http://www.groupon-works.fi/>>. Luettu 24.2.2015.
- 2 Tietoa Grouponista. 2015. Verkkodokumentti. Groupon, Inc. <<http://www.groupon.fi/about-us>>. Luettu 25.4.2014.
- 3 Carlson, Nicholas. 2011. Business Insider - INSIDE GROUPON: The Truth About The World's Most Controversial Company. Verkkodokumentti. <<http://www.businessinsider.com/inside-groupon-the-truth-about-the-worlds-most-controversial-company-2011-10>>. Luettu 25.4.2014.
- 4 Salo, Pia. 2013. Kiinasta Suomeen startup-yrittäjäksi. Verkkodokumentti. T-Lehti. <<http://www.t-lehti.fi/artikkelit/kiinasta-suomeen-startup-yrittajaksi/>>. Luettu 12.4.2014.
- 5 Yamii. 2015. Yamii. Verkkodokumentti. <<http://yamii.fi/>>. Luettu 24.2.2015.
- 6 Lappalainen, Elina. 2013. Näiden 31 nuoren yrityksen kasvuun Tekes uskoo. Verkkodokumentti. Talouselämä. <<http://www.talouselama.fi/Kasvuyritykset/naiden+31+nuoren+yriyksen+kasvuun+tekes+uskoo/a2165126>>. Luettu 26.2.2015.
- 7 K-Plussa-pisteet älypuhelimien ruudulle. 2013. Tiedote. Sunduka Oy. <[http://www.cardu.com/wp-content/uploads/2014/11/K-Plussa-pisteet\\_30052013.pdf](http://www.cardu.com/wp-content/uploads/2014/11/K-Plussa-pisteet_30052013.pdf)>. Luettu 26.2.2015.
- 8 Cardu. 2015. Verkkodokumentti. Sunduka Oy. <<http://www.cardu.com/>>. Luettu 26.2.2015.
- 9 Incentz. 2014. Verkkodokumentti. Incentz. <<http://incentz.com/web/s/how-it-works.html>>. Luettu 12.4.2014.
- 10 Luhtala, Marko. 2013. Mitä on mobiilimarkkinointi. Verkkodokumentti. NearMe Services. <<http://mobiilimarkkinointiopas.fi/oppaat/mita-on-mobiilimarkkinointi.pdf>>. Luettu 3.3.2015.
- 11 Mainonnan neuvottelukunta. 2013. Tiedote. TNS Gallup Oy. <[http://www.tns-gallup.fi/doc/uutiset/Markkinointiviestinnan\\_maara\\_2012.pdf](http://www.tns-gallup.fi/doc/uutiset/Markkinointiviestinnan_maara_2012.pdf)>. Luettu 2.3.2015.
- 12 Kuluttaja on kännykän päässä, onko suomalainen kauppa?. 2014. Verkkodokumentti. TNS Gallup. <<http://www.tns-gallup.fi/uutiskirje2014/10/otsikko1>>. Luettu 2.3.2015.

- 13 Väestön tieto- ja viestintätekniiikan käyttö 2014. Verkkodokumentti. Tilastokeskus. <[http://www.stat.fi/til/sutivi/2014/sutivi\\_2014\\_2014-11-06\\_fi.pdf](http://www.stat.fi/til/sutivi/2014/sutivi_2014_2014-11-06_fi.pdf)>. Luettu 2.3.2015.
- 14 Majority of US Internet Users Will Redeem Digital Coupons in 2013. Verkkodokumentti. eMarketer. <<http://www.emarketer.com/Article/Majority-of-US-Internet-Users-Will-Redeem-Digital-Coupons-2013/1010313>> Luettu 4.3.2015.
- 15 Sulautettujen järjestelmien ketterä käsikirja. 2014. Verkkodokumentti. Technology Research Center. <<http://trc.utu.fi/embedded/kasikirja/1/1/>>. Luettu 21.3.2015.
- 16 Ketteryys haltuun: Yleisimmät ketterät käytännöt. 2011. Verkkodokumentti. Sinenen Meteoriitti. <<http://www.meteoriitti.com/Artikkelisarjat/Ketteryys-haltuun/Ketteryys-haltuun-Yleisimmat-ketterat-kaytannot/>>. Luettu 21.3.2015.
- 17 Reenskaug, Trygve. 1979. THING-MODEL-VIEW-EDITOR. Xerox.
- 18 Koskimies, Kai ja Mikkonen, Tommi. 2005. Ohjelmistoarkkitehtuurit. Verkkodokumentti. <<http://www.cs.tut.fi/~kk/Ohjelmistoarkkitehtuuri.pdf>>. Luettu 13.4.2014.
- 19 6 good reasons to use Symfony. 2014. Verkkodokumentti. Symfony. <<http://symfony.com/six-good-reasons>>. Luettu 20.2.2014.
- 20 Welcome to the Doctrine Project. 2015. Verkkodokumentti. Doctrine Team. <<http://www.doctrine-project.org/>>. Luettu 20.2.2015.
- 21 Jira. 2015. Verkkodokumentti. Atlassian. <<https://www.atlassian.com/software/jira>>. Luettu 20.2.2015.
- 22 Apache Subversion. 2015. Verkkodokumentti. The Apache Software Foundation. <<http://subversion.apache.org/>>. Luettu 7.3.2015.
- 23 TortoiseSVN. 2014. Verkkodokumentti. The TortoiseSVN team. <<http://tortoisesvn.net/about.html>>. Luettu 19.4.2014.
- 24 Using Subversion Support in NetBeans IDE. 2015. Verkkodokumentti. Oracle. <<https://netbeans.org/kb/docs/ide/subversion.html>>. Luettu 20.2.2015.
- 25 NetBeans IDE Features: PHP Development. 2015. Verkkodokumentti. Oracle. <<https://netbeans.org/features/php/index.html>>. Luettu 20.2.2015.
- 26 Foundation. 2015. Verkkodokumentti. ZURB. <<http://foundation.zurb.com/>>. Luettu 7.3.2015.



- 27 Kramer, Jen. 2014. Responsive Design Frameworks: Just Because You Can, Should You?. Verkkodokumentti. Smashing Magazine. <<http://www.smashingmagazine.com/2014/02/19/responsive-design-frameworks-just-because-you-can-should-you/>>. Luettu 14.9.2014.
- 28 Bergmann, Sebastian. 2015. Welcome to PHPUnit!. Verkkodokumentti. <<https://phpunit.de/index.html>>. Luettu 8.3.2015.
- 29 Ken, Schwaber and Jeff, Sutherl. 2013. The Scrum Guide. Verkkodokumentti. <<http://scrumwell.files.wordpress.com/2014/03/scrum-guide-2013-fi-v1-1.pdf>>. Luettu 14.9.2014.

## Benefit-luokka

```

<?php

namespace Oranki\BenefitBundle\Entity;

use Oranki\ImageBundle\Entity\Image;
use Oranki\ImageBundle\Entity\UploadedImage;
use Oranki\CategoryBundle\Entity\Category;
use Oranki\UserBundle\Entity\Advertiser;
use Oranki\PlaceOfBusinessBundle\Entity\PlaceOfBusiness;
use Symfony\Component\Validator\Constraints as Assert;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\ORM\Event\LifecycleEventArgs;
use Doctrine\ORM\Event\PreUpdateEventArgs;
use Doctrine\ORM\Mapping as ORM;
use JMS\Serializer\Annotation as Serializer;
use DateTime;

/**
 * @ORM\Entity
 * @ORM\InheritanceType("JOINED")
 * @ORM\DiscriminatorColumn(name="type", type="string")
 * @ORM\DiscriminatorMap({"benefit" - "Benefit", "coupon" - "Coupon"})
 * @ORM\Table("Benefit")
 * @ORM\HasLifecycleCallbacks
 */
class Benefit
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     * @Serializer\Groups({"published"})
     * @var integer
     */
    private $id;

    /**
     * @ORM\Column(type="string")
     * @Assert\NotBlank()
     * @Serializer\Groups({"published"})
     * @var string
     */
    private $name;

    /**
     * @ORM\Column(type="date")
     * @Assert\NotBlank()
     * @var DateTime
     */
    private $startDate;

    /**
     * @ORM\Column(type="date")
     * @Assert\NotBlank()
     * @var DateTime
     */
    private $endDate;

    /**
     * @ORM\Column(type="date")
     * @Assert\NotBlank()
     * @var DateTime
     */
    private $publishDate;
}

```

```

/**
 * @ORM\Column(type="float", nullable=true)
 * @var float
 * @Assert\Range(min=0.01, minMessage="benefit.retailPrice.greaterThanZero")
 */
private $retailPrice;

/**
 * @ORM\Column(type="float", nullable=true)
 * @var float
 * @Assert\Range(min=0, minMessage="benefit.offerPrice.min")
 */
private $offerPrice;

/**
 * @ORM\Column(type="float")
 * @Assert\NotBlank()
 * @Assert\Range(min=1, minMessage="benefit.discountPercent.min",
 * max=100, maxMessage="benefit.discountPercent.max")
 * @var float
 */
private $discountPercent;

/**
 * @ORM\Column(type="text")
 * @Assert\NotBlank()
 * @Serialize\Groups({"published"})
 * @var string
 */
private $description;

/**
 * @ORM\Column(type="boolean")
 * @Assert\NotNull()
 * @var boolean
 */
private $deleted = false;

/**
 * @ORM\Column(type="string", nullable=true)
 * @var string
 */
private $deleteReason;

/**
 * @ORM\ManyToOne(targetEntity="Oranki\ImageBundle\Entity\Image", cascade={"persist"})
 * @ORM\JoinColumn(nullable=false)
 * @Assert\NotNull(message="benefit.image.notNull")
 * @Assert\Valid()
 * @var Image
 */
private $image;

/**
 * @ORM\ManyToMany(targetEntity="Oranki\PlaceOfBusinessBundle\Entity\PlaceOfBusiness", inversedBy="benefits")
 * @ORM\JoinColumn(nullable=false)
 * @Assert\NotNull()
 * @Assert\Count(min=1, minMessage="benefit.placesOfBusiness.atLeastOne")
 * @var ArrayCollection
 */
private $placesOfBusiness;

/**
 * @ORM\ManyToOne(targetEntity="Oranki\CategoryBundle\Entity\Category", inversedBy="benefits")
 * @ORM\JoinColumn(nullable=false)
 * @Assert\NotNull(message="benefit.category.notNull")
 * @var Category
 */
private $category;

public function __construct()
{
    $this->placesOfBusiness = new ArrayCollection();
}

```

```
/**
 * @return integer
 */
public function getId()
{
    return $this->id;
}

/**
 * @return string
 */
public function getName()
{
    return $this->name;
}

/**
 *
 * @param string $name
 */
public function setName($name)
{
    $this->name = $name;
}

/**
 * @return DateTime
 */
public function getStartDate()
{
    return $this->startDate;
}

/**
 * @param DateTime $startDate - null)
 */
public function setStartDate(DateTime $startDate - null)
{
    $this->startDate = $startDate;
}

/**
 * @return DateTime
 */
public function getEndDate()
{
    return $this->endDate;
}

/**
 * @param DateTime $endDate
 */
public function setEndDate(DateTime $endDate - null)
{
    $this->endDate = $endDate;
}

/**
 * @return float
 */
public function getRetailPrice()
{
    return $this->retailPrice;
}

/**
 * @param float $retailPrice
 */
public function setRetailPrice($retailPrice)
{
    $this->retailPrice = $retailPrice;
}
}
```

```
/**
 * @return float
 */
public function getOfferPrice()
{
    return $this->offerPrice;
}

/**
 * @param float $offerPrice
 */
public function setOfferPrice($offerPrice)
{
    $this->offerPrice = $offerPrice;
}

/**
 * @return float
 */
public function getDiscountPercent()
{
    return $this->discountPercent;
}

/**
 * @param float $discountPercent
 */
public function setDiscountPercent($discountPercent)
{
    $this->discountPercent = $discountPercent;
}

/**
 * @return string
 */
public function getDescription()
{
    return $this->description;
}

/**
 * @param string $description
 */
public function setDescription($description)
{
    $this->description = $description;
}

/**
 * @return DateTime
 */
public function getPublishDate()
{
    return $this->publishDate;
}

/**
 * @param DateTime $publishDate
 */
public function setPublishDate(DateTime $publishDate = null)
{
    $this->publishDate = $publishDate;
}

/**
 * @return boolean
 */
public function isDeleted()
{
    return $this->deleted;
}
```

```

/**
 * @param boolean $deleted
 */
public function setDeleted($deleted)
{
    $this->deleted = $deleted;
}

/**
 * @return string
 */
public function getDeleteReason()
{
    return $this->deleteReason;
}

/**
 * @param string $deleteReason
 */
public function setDeleteReason($deleteReason)
{
    $this->deleteReason = $deleteReason;
}

/**
 * @return Image
 */
public function getImage()
{
    return $this->image;
}

/**
 * @param Image $image
 */
public function setImage(Image $image = null)
{
    $this->image = $image;
}

/**
 * @return ArrayCollection
 */
public function getPlacesOfBusiness()
{
    return $this->placesOfBusiness;
}

/**
 * @param PlaceOfBusiness $placeOfBusiness
 */
public function addPlaceOfBusiness(PlaceOfBusiness $placeOfBusiness)
{
    if (!$this->placesOfBusiness->contains($placeOfBusiness)) {
        $this->placesOfBusiness->add($placeOfBusiness);
        $placeOfBusiness->addBenefit($this);
    }
}

/**
 * @param PlaceOfBusiness $placeOfBusiness
 */
public function removePlaceOfBusiness(PlaceOfBusiness $placeOfBusiness)
{
    $this->placesOfBusiness->removeElement($placeOfBusiness);
}

/**
 * @return boolean
 */
public function hasPlacesOfBusiness()
{
    return !$this->placesOfBusiness->isEmpty();
}

```

```

/**
 * @return Category
 */
public function getCategory()
{
    return $this->category;
}

/**
 * @param Category $category
 */
public function setCategory(Category $category = null)
{
    if ($this->category !== $category) {
        $this->category = $category;

        if ($this->category !== null) {
            $this->category->addBenefit($this);
        }
    }
}

/**
 * @return boolean
 */
public function isPublished()
{
    return $this->getPublishDate() <= new DateTime('today');
}

/**
 * @Assert\True(message="benefit.offerPrice.mustBeLessThanRetailPrice")
 * @return boolean
 */
public function isOfferPriceLessThanRetailPrice()
{
    return $this->offerPrice --- null || $this->offerPrice < $this->retailPrice;
}

/**
 * @Assert\True(message="benefit.discountPercent.imbalanced")
 * @return boolean
 */
public function isDiscountPercentValid()
{
    if ($this->discountPercent --- null || $this->offerPrice --- null || !$this->retailPrice) {
        return true;
    }

    static $errorMargin = 5;
    $calculated = 100 - $this->offerPrice / $this->retailPrice * 100;

    return abs($this->discountPercent - $calculated) <= $errorMargin;
}

/**
 * @Assert\True(message="benefit.endDate.mustBeLaterThanStartDate")
 * @return boolean
 */
public function isEndDateLaterThanStartDate()
{
    return $this->endDate --- null || $this->startDate --- null || $this->endDate >= $this->startDate;
}

/**
 * @Assert\True(message="benefit.startDate.mustBeLaterThanPublishDate")
 * @return boolean
 */
public function isStartDateLaterThanPublishDate()
{
    return $this->startDate --- null ||
        $this->publishDate --- null || $this->startDate >= $this->publishDate;
}

```

```
/**
 * @Assert\False(message="benefit.publishDate.cannotBeInPast")
 * @return boolean
 */
public function isPublishDateInPast()
{
    return $this->publishDate !== null as $this->publishDate < new DateTime('today')
}

/**
 * @ORM\PreRemove
 * @param LifecycleEventArgs $event
 */
public function preRemove(LifecycleEventArgs $event)
{
    $image = $this->getImage();

    if ($image instanceof UploadedImage) {
        $event->getEntityManager()->remove($image);
    }
}

/**
 * @ORM\PreUpdate
 * @param PreUpdateEventArgs $event
 */
public function preUpdate(PreUpdateEventArgs $event)
{
    if ($event->hasChangedField('image')) {
        $oldImage = $event->getOldValue('image');

        if ($oldImage instanceof UploadedImage) {
            $event->getEntityManager()->remove($oldImage);
        }
    }
}

/**
 * @return Advertiser
 */
public function getAdvertiser()
{
    return $this->placesOfBusiness->first()->getAdvertiser();
}

/**
 * @return boolean
 */
public function isCoupon()
{
    return $this instanceof Coupon;
}
}
```