TAMPEREEN
AMMATTIKORKEAKOULU

VERTAISMYYNTIMOBIILISOVELLUS

Jyri Jaakkola

Juha Sivonen

Opinnäytetyö
Tammikuu 2016
Tietotekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikka
Ohjelmistotekniikka

JAAKKOLA JYRI & SIVONEN JUHA:
Vertaismyyntimobiilisovellus

Opinnäytetyö 50 sivua, joista liitteitä 0 sivua
Tammikuu 2016

Idea sovellukselle lähti liikkeelle nykyisten vertaismyyntipalveluiden yksityishenkilölle tuottamasta vaivasta. Tämän päivän ohjelmistojen käyttäminen on työlästä. Hakujen tekeminen, tuotteiden lisääminen ja yhteydenottaminen myyjään on hankalaa. Tämän sovelluksen tarkoituksena on poistaa ylimääräinen työ tästä kaikesta hankaluudesta. Lähdettiin kehittämään mobiilisovellusta, jolla tarjotaan paranneltu vaihtoehto jo olemassa oleville palveluille.

Opinnäytetyössä esitellään prototyypille valitut oleellisimmat ohjelmiston toiminnallisuudet ja esitellään kuinka niiden tulee toimia. Lisäksi työssä esitellään käytettyjä teknologioita ja työkaluja, joita sovelluksen kehityksessä käytettiin ja kerrotaan miten itse ohjelmisto on toteutettu. Toteutuksesta kerrotaan laitteen ja palvelimen välisestä kommunikoinnista ja versionhallinnasta. Opinnäytetyön loppuosassa kerrotaan ohjelmiston kehitysvaiheen testaamisesta ja esitellään jatkokehitysideoita.

Opinnäytetyöprosessiin ollaan tyytyväisiä sovelluksen prototyypin kehittämisen ja työn raportoinnin osalta. Vertaismyyntimobiilisovellus prototyyppi ja taustalla toimiva tietokantarajapinta toteutettiin hyvälle tasolle. Prosessista löytyy myös kehitettävää. Suurempi määrä suunnittelua ohjelmistoprojektin alkuvaiheessa helpottaisi käytännön toteutuksen tekemistä ja vältyttäisiin joiltakin ongelmilta ja saman työn useampaan kertaan tekemiseltä.

Asiasanat: mobiilisovellus, vertaismyynti

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Information Technology
Software Engineering

JAAKKOLA JYRI & SIVONEN JUHA:
Peer selling mobile application

Bachelor's thesis 50 pages, appendices 0 pages
January 2016

The idea of the thesis was invented when person was frustrated of the current peer sale systems on the market right now. The current systems are complicated in terms of product searches, adding new product for sale and communicating with the seller of the interesting product. This is why the developing process of this mobile application was started. The goal was to make a new peer selling system with better features and simpler to use than the systems in the market now.

To start with, the thesis introduces the features and how they should work in the working prototype. For the prototype purposes there are only the most substantial and the most important features that user would need in peer selling system. The thesis also covers the technologies and tools used to make the prototype system. The thesis covers how the system communication works between the mobile device and the server to get the data from database to the mobile device. The thesis also covers how the system is and will be tested and how the process will continue from the point that it is currently standing.

The thesis process has been a success in a big picture. The peer selling mobile application and database interface were made to good point. Nevertheless, like always, there are points that could have been done better. For example in the starting point of the project, the planning was made a bit roughly if at all. In that way there would have been less problems and less parts made multiple times.

Key words: mobile application, peer selling

SISÄLLYS

ERITYISSANASTO

Back-end	Palvelin pää, tietoa käsittelevä ohjelmiston osa
Bugi	Virhe ohjelmistokoodissa
ClearCase	Maksullinen keskitetty versionhallintatyökalu
Commit	Uusi päivitys versionhallinnassa
CVS	Concurrent Versions System, SCM-työkalu
Debugger	Debug-työkalu
DMA	Direct Memory Access
Front-end	Ohjelmiston käyttäjälle esitettävä osa
Hashtag	Avainsana, aihetunniste
Intellisense	Microsoftin ohjelmistokoodin ennakoiva tekstinsyöttö
Lambda-ilmaisu	Anonyymi funktio
Natiivi	Alustalle tarkoitettu
Perforce	Maksullinen versionhallintatyökalu
Plugin	Ohjelmiston lisäosa
Polymorfismi	Monimuotoisuus
Refaktorointi	Ohjelmistokoodin siistimistä
Repository	Versionhallinnan tietovarasto
SCM	Supply chain management, palveluiden ja tavaroiden kulun valvonta.
Staging area	Git hakemistossa oleva tiedosto, joka sisältää tallennettavia tietoja
Subversion	Avoimen lähdekoodin SCM-työkalu
Web-API	Palvelimella toimiva ohjelmallinen rajapinta
WPF	Windows Presentation Foundation

1 JOHDANTO

Vertaismyyntimobiilisovelluksen idea lähti liikkeelle yksityishenkilön tyytymättömyydestä nykyisiin verkon vertaismyyntipalveluihin. Nykyisillä ohjelmistoilla tehtävät haut vaativat suuren määrän täytettäviä hakuehtoja ja myytävien tuotteiden lisääminen palveluihin on aikaa vievää. Yhteydenotto myyjiin onnistuu vain sähköpostilla, ellei myyjä ole laittanut puhelinnumeroaan näkyviin.

Tehtävä vertaismyyntimobiilisovellus on tarkoitettu helpottamaan tarpeettomien tavaroiden myymistä. Tuotteiden lisäämisessä palveluun tullaan painottamaan tuotteen mahdollisimman helppoa ja nopeaa myyntiin saattamista. Tuotteita selatessa ei tarvitse täyttää lukuisia hakuehtoja, vaan haku tapahtuu täyttämällä muutama tärkeimmistä hakuehdoista. Tärkeimpiin hakuehtoihin tulevat kuulumaan haluttu maksimietäisyys nykyisestä sijainnista, maksimi hinta ja halutut hakusanat tuotteelle. Yhteydenotto myyjään tullaan tekemään helpoksi ohjelmistosta löytyvällä chat-ominaisuudella.

Opinnäytetyön tarkoituksena on kehittää toimiva sovellusprototyyppi, jolla voidaan helpottaa yksityisillä henkilöillä lojumaan jääneiden tavaroiden myymistä.

Opinnäytetyön toisessa luvussa käydään läpi sovelluksen ideansyntymistä, tutustutaan ohjelmiston vaatimuksissa määriteltyihin pakollisiin toimintoihin, sovelluksen markkinointiin sekä otetaan kantaa markkinarakoon pintapuoleisesti ja esitellään lyhyesti pahimpia kilpailijoita.

Kolmannessa luvussa paneudutaan sovelluksen arkkitehtuurin suunnitteluun tutkimalla suunnittelussa hyväksikäytettyjä erilaisia arkkitehtuurisia kaavioita Philippe Kruchten:in 4+1 malliin pohjautuen.

Neljännessä luvussa käydään läpi vertaismyyntimobiilisovelluksen toteutuksessa käytettyjä tekniikoita ja työkaluja, jonka jälkeen viidennessä luvussa käydään läpi itse ohjelmiston toimintamallia sekä omia päätelmiä ohjelmiston kehittämisestä.

Kuudennessa luvun aiheena on ohjelmiston toimivuuden testaaminen. Luvussa kerrotaan ohjelmiston testauksesta kehitysvaiheessa sekä miten valmista prototyyppiä on tarkoitus testata.

Viimeisessä luvussa käsitellään ohjelmiston jatkokehityssuunnitelmia. Esittelyssä on jo suunnitteluvaiheilla olevat toiminnot, sekä pohditaan mahdollisia muita lisäominaisuuksia.

2 VERTAISMYYNTIMOBIILISOVELLUS

Tässä luvussa perehdytään vertaismyyntimobiilisovelluksen prototyypiltä vaadittaviin ominaisuuksiin ja kuinka niiden on tarkoitus toimia. Lisäksi luvussa pohditaan mihin kohderyhmään mainonta ja itse sovellus kohdennetaan. Pohdinnassa on myös tämän hetkinen markkinatilanne tämän kaltaisilla palveluilla ja vertaismyyntimobiilisovelluksen sopivuus näille markkinoille.

2.1 Prototyypin ominaisuudet

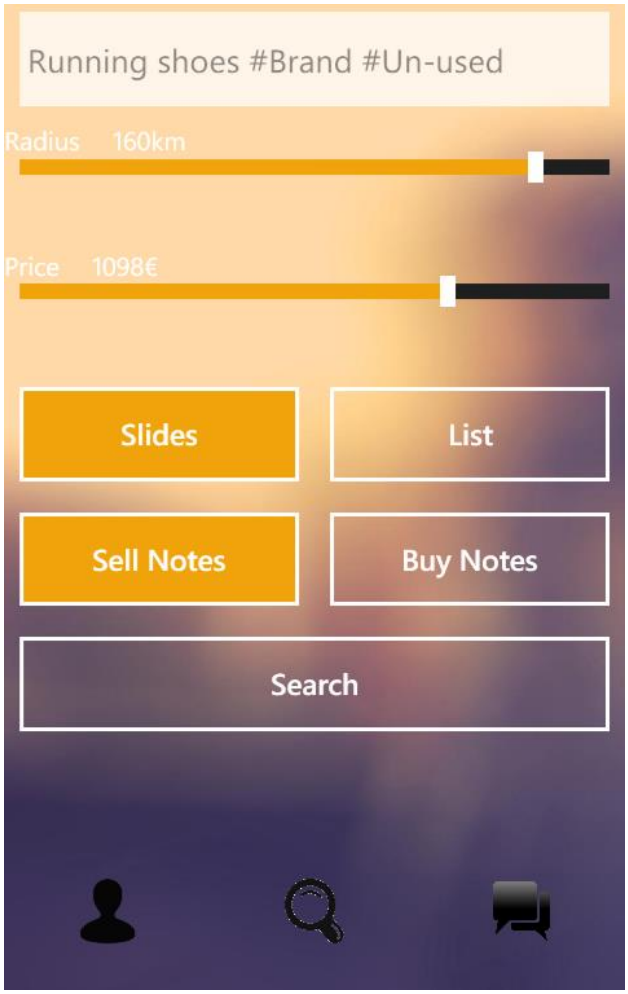
Vertaismyyntimobiilisovelluksen prototyypiltä vaadittavat ominaisuudet tulevat olemaan vain ohjelmiston tärkeimmät ja oleellisimmat toiminnallisuudet. Nämä ominaisuudet ovat tuotteiden selaus, lisäys, lisääminen suosikkeihin, poisto selaustuloksista, ilmianto ja chat. Ominaisuudet käydään tarkemmin läpi alaluvuissa.

Kyseiset ominaisuudet on valittu prototyyppiin, koska niiden koetaan olevan ohjelmiston perustoiminnallisuuden ja käyttömukavuuden kannalta kaikkein tärkeimmät. Ominaisuuksia suunnitellaan ja lisätään tulevaisuudessa, kun se koetaan tarpeelliseksi. Palvelun käyttäminen tulee vaatimaan sisäänkirjautumisen. Luvussa esiintyvät kuvat ovat prototyypin kehitysvaiheesta, eikä niitä tule pitää viimeisteltyinä versioina.

2.1.1 Tuotteiden selaus

Ohjelmassa tuotteiden selaus tehdään mahdollisimman helpoksi ja nopeaksi. Yleisistä hakuehdoista karsitaan suurin osa pois ja jätetään vain oleellisimmat, jotta haku olisi mahdollisimman helppoa ja nopeaa. Vertaismyyntimobiilisovellus tulee mahdollistamaan myös uudenlaisen selauksen, jolla voi hakea tuotteita lähietäisyydeltä. Ominaisuudessa tullaan hyödyntämään laitteen omaa paikallistamisominaisuutta tuotetta lisätessä ja tuotteita hakiessa. Selaustuloksista selviää suoraan kuinka kaukana kyseinen tuote on nykyisestä sijainnista. Tuotteiden selaukseen käyttäjän tarvitsee syöttää vain haluamansa hakusanat, hashtagit, maksimihinta ja maksimietäisyys.

Kuten alla olevasta kuvasta näkee (kuva 1), hakuehtoja ei ole paljon. Tämä mahdollistaa sen, että hakujen tekoon ei kulu turhan paljon aikaa. Jättämällä hakusanakentän tyhjäksi käyttäjä voi hakea tuotteita suodattamalla tulokset ainoastaan hinnan ja etäisyyden perusteella.

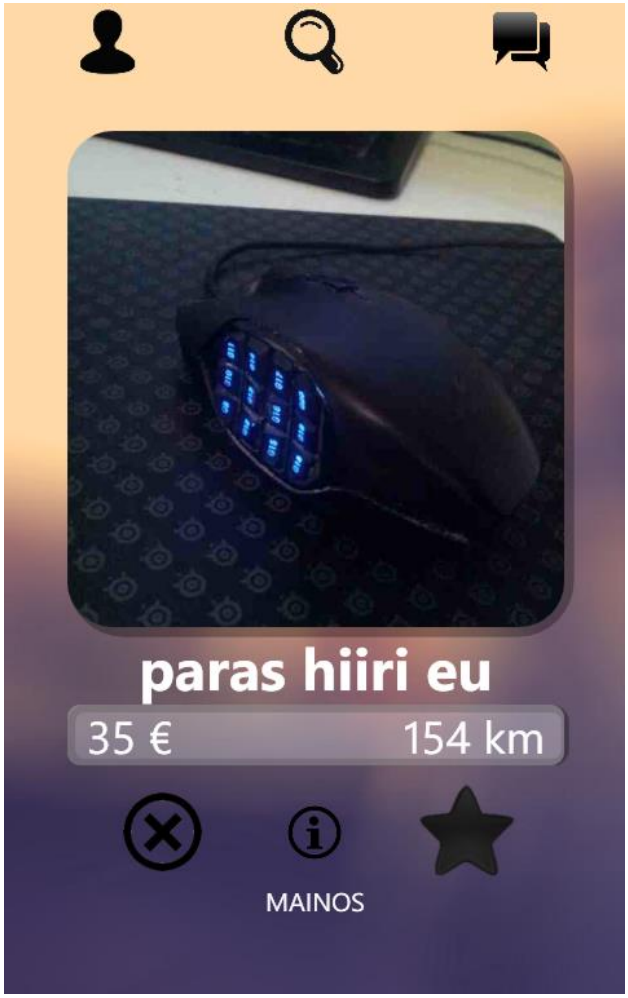


KUVA 1. Hakuehdot.

Tuotteiden selauksiin halutaan kaksi erilaista selaustapaa, moderni tuotekortti-selaus (kuva 2), sekä perinteinen listaselaus (kuva 3). Näillä eri selaustavoilla saavutetaan miellyttävä käyttäjäkokemus laajemmalle käyttäjäkannalle. Eri selaustapojen välillä vaihtaminen tapahtuu valitsemalla selaustapa hakuehtoja (kuva 1) täytettäessä. Selaustavan vaihtaminen ei prototyypin kohdalla onnistu ilman uuden haun tekemistä.

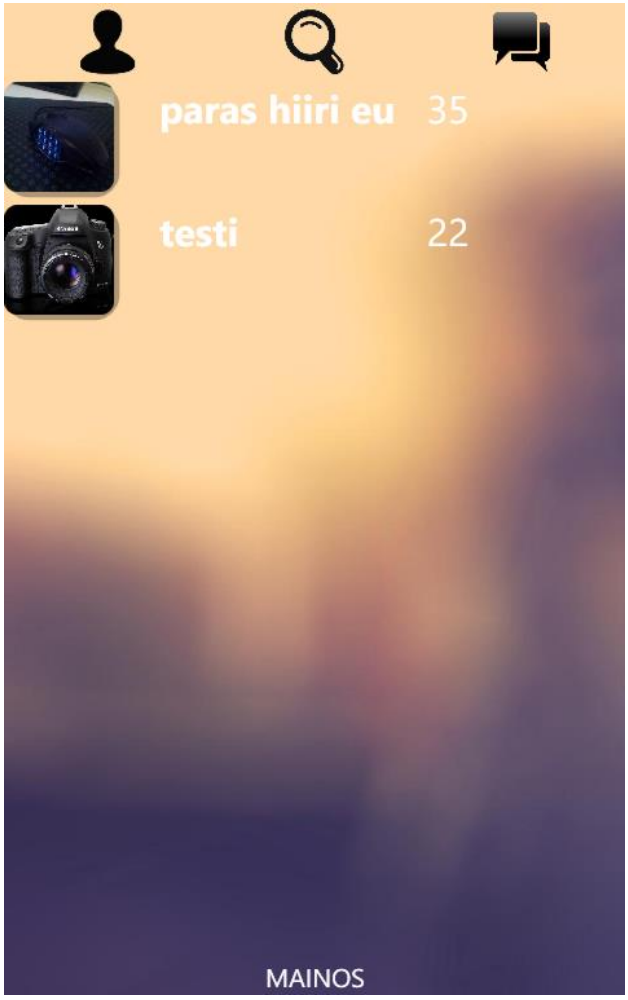
Tuotekortti-selauksessa (kuva 2) tuotteet tulevat näkymään yksittäin, isommalla kuvalla kuin listaselauksessa, ja niitä voi selata pyyhkäisemällä näytöllä oikealle tai vasemmalle.

Kun käyttäjä pääsee viimeiseen tuotteeseen ja tämän jälkeen pyyhkäisee vielä kerran vasemmalle, eli seuraavaan tuotteeseen, alkaa tuotteiden selaus alusta. Tässä selaustavassa nähdään nopeasti tuotteen nimi, hinta ja etäisyys käyttäjän nykyisesti sijainnista.



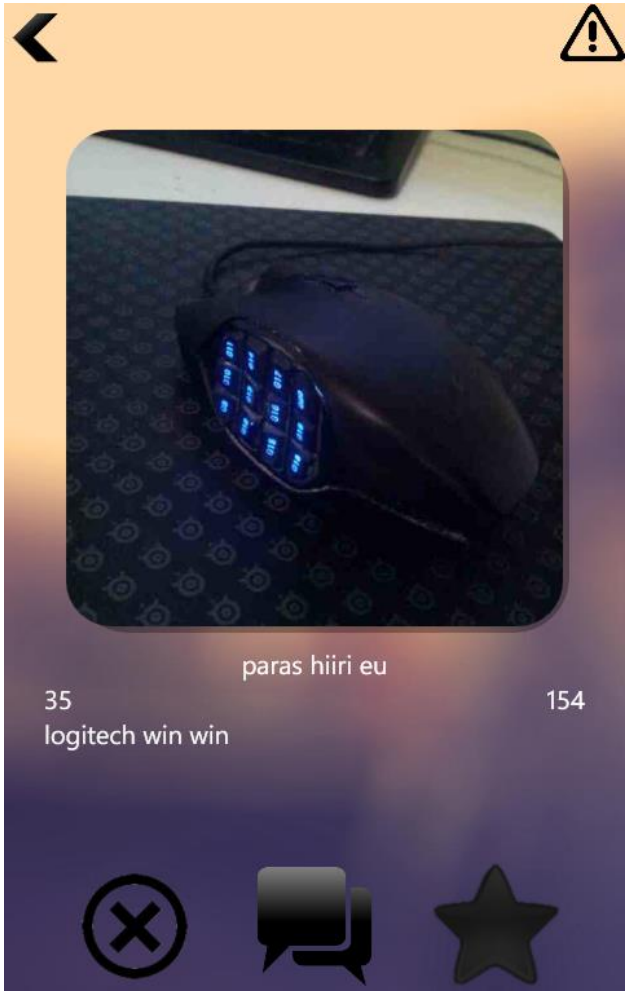
KUVA 2. Tuotekortti-selaus.

Perinteisemmässä listaselauksessa (Kuva 3) tuotteet tulevat olemaan selkeässä ja yksinkertaistetussa listassa ja käyttäjä pystyy selaamaan laajempaa tuotemäärää kerralla. Tällä selaustavalla tuotteita voidaan käydä nopeammin läpi ja löytää etsitty tuote paljon helpommin. Tässä selaustavassa tuotteesta näkyy vähemmän tietoja kuin aikaisemmassa tuotekortti-selauksessa. Ainoat näytettävät tiedot ovat tuotteen pieni myyntikuva, nimi ja hinta.



KUVA 3. Listaselaus

Alla olevassa kuvassa (kuva 4) on avattuna tuotesivu. Tämä näkymä tulee aukeamaan aina, kun avataan toisen käyttäjän lisäämä tuote-ilmoitus. Tuotesivulla voidaan poistaa tuote selaustuloksista, ilmiantaa asiaton tuote, lisätä tuote suosikkeihin ja aloittaa chat. Tuotesivulla tuotteen kuvia voi selata samaan tapaan kuin korttiselauksessa pyyhkäisemällä vasemmalle tai oikealle. Myös tuotesivulla käyttäjän päästyä viimeiseen kuvaan alkaa kierros alusta. Tuotesivulla nähdään kaikki tuotteesta tallennetut tiedot selkeästi käyttäjälle esitettynä.

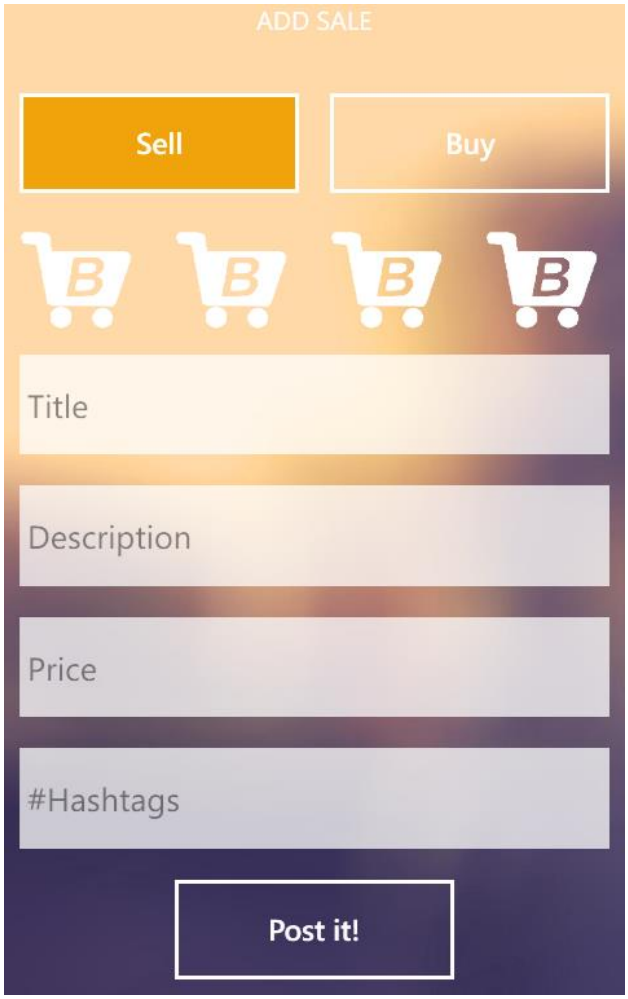


KUVA 4. Tuotesivu.

2.1.2 Tuotteiden lisäys

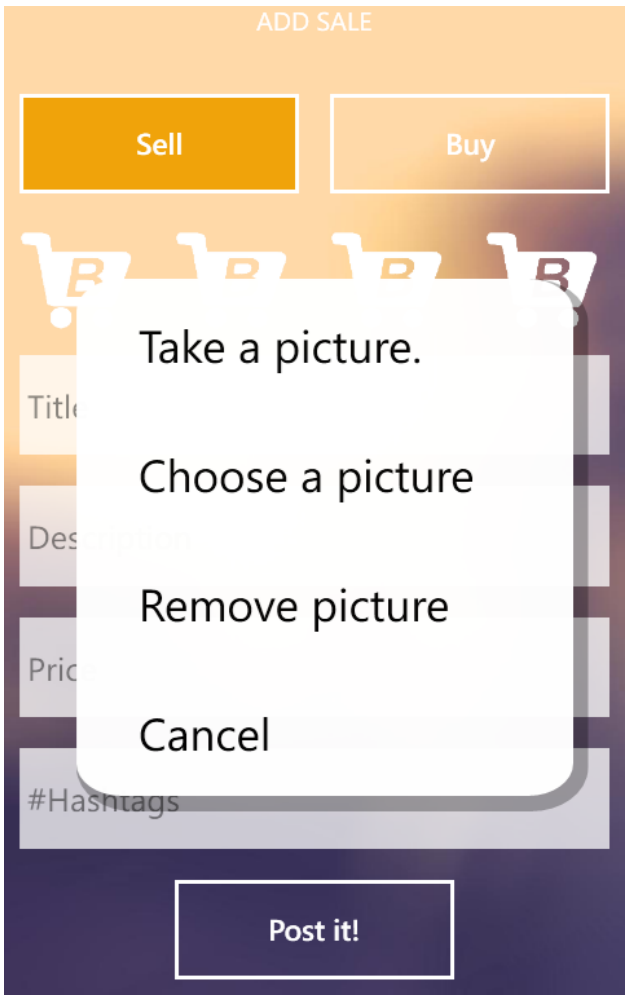
Tuotteiden lisääminen palveluun on suunniteltu mahdollisimman helpoksi ja nopeaksi. Tarkoituksena on poistaa päänvaiva tarpeettomaksi jääneiden tuotteiden myyntiin saattamisesta. Näin kenelläkään tavarat eivät jää lojumaan ympäriinsä sen takia, että myyntiin saattaminen on liian työlästä. Tuotteesta tulee olemaan mahdollisuus lisätä neljä (4) kuvaa, otsikko, kuvaus, hinta ja hashtagit. Näistä pakollisia tietoja tulevat olemaan vain otsikko ja hinta. Muut tiedot tulevat olemaan suositeltavia täytettäväksi, mutta eivät tule olemaan välttämättömiä tuotteen lisäämiseksi palveluun. Esimerkiksi tuotetta hyvin kuvaavien hashtagien lisääminen parantaa todennäköisyyttä tuotteen löytymisestä hakutuloksista.

Kuvien lisääminen ilmoitukseen onnistuu painamalla kuvassa (kuva 5) näkyviä ostoskärryjen kuvia. Tuotteiden lisäämiseen käytettävä näkymä on pidetty mahdollisimman yksinkertaisena ja helppolukuisena. Tämä onnistuu hyvin, koska tuotteen lisääminen ei vaadi turhan monia tietoja.



KUVA 5. Tuotteen lisääminen.

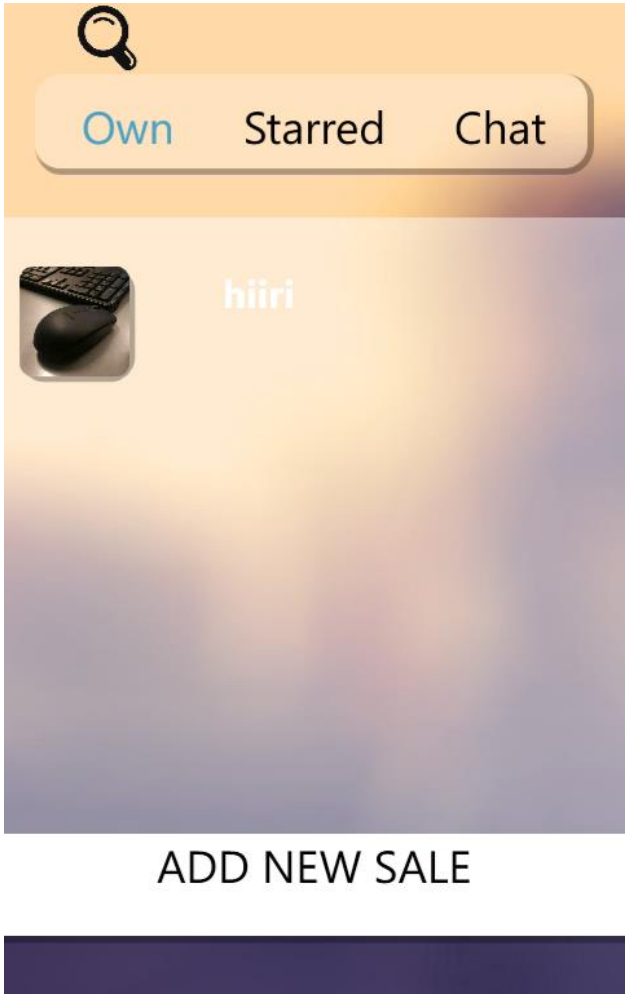
Käyttäjän painaessa ostoskärryn kuvaa avautuu valikko (kuva 6), josta valitaan haluaako käyttäjä ottaa uuden kuvan vai valita jo olemassa olevista kuvista. Jos käyttäjä haluaa ottaa uuden kuvan, käynnistyy puhelimen kamerasovellus automaattisesti. Mikäli käyttäjä on jos lisännyt kuvan ja painaa tätä kuvaa, aukeaa sama valikko kuin ostoskärryn kuvaakin painettaessa. Nyt käyttäjällä on mahdollisuus korvata kuva toisella kuvalla tai poistaa valittu kuva.



KUVA 6. Valikko kuvan lisäämistä varten

Kun tuote on lisätty palveluun, tuote tallennetaan myös puhelimen muistiin. Omien tuotteiden selaaminen on mahdollista Omat tuotteet -välilehdeltä (kuva 7). Omien tuotteiden lista on hyvin samankaltainen kuin listaselauksessa (kuva 3) ja toimimaan myös samalla periaatteella. Tältä sivulta löytyy myös painike uuden tuotteen lisäämiseen. Tuotteen lisäys -painiketta painettaessa siirrytään tuotteen lisäyssivulle (kuva 5).

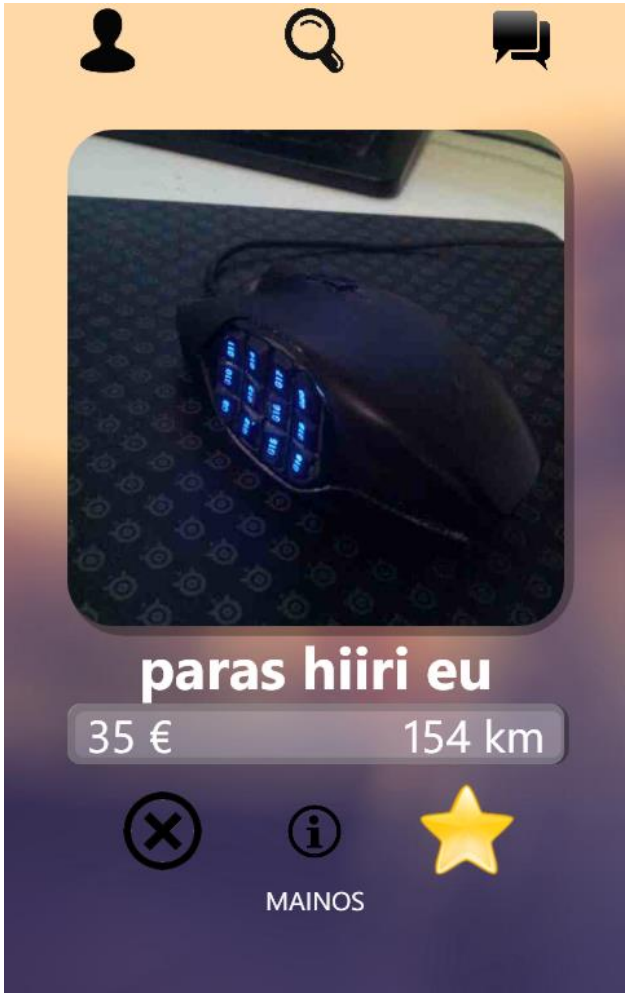
Omia tuotteita avaaminen tapahtuu painamalla haluamaansa tuotetta. Tämä avaa tuotteen omana sivunaan. Tällä sivulla voi esimerkiksi poistaa tuotteen, mikäli sen haluaa ottaa palvelusta pois.



KUVA 7. Omat tuotteet

2.1.3 Tuotteiden lisääminen suosiksi

Tuotteiden joukosta hyväksi havaittujen tuotteiden lisääminen suosikkeihin tulee olemaan mahdollista painamalla korttiselauksessa tai tuotesivulla näkyvää tähteä. Mikäli tuotetta ei ole lisätty suosikkeihin näytetään tähti mustana. Suosikiksi lisätyissä tuotteissa tämä kyseinen tähti näkyy keltaisena (kuva 8). Suosikki-tähteä painettaessa tähti muuttuu hiljalleen läpinäkyväksi ja uudelleen näkyväksi samalla väriään muuttaen, ilmaisten tällä tavalla käyttäjälle suosikiksi lisäämisen tai poistamisen onnistuneen. Omat suosikit löytyvät helposti suosikit-välilehdeltä (kuva 9).



KUVA 8. Suosikki tähti

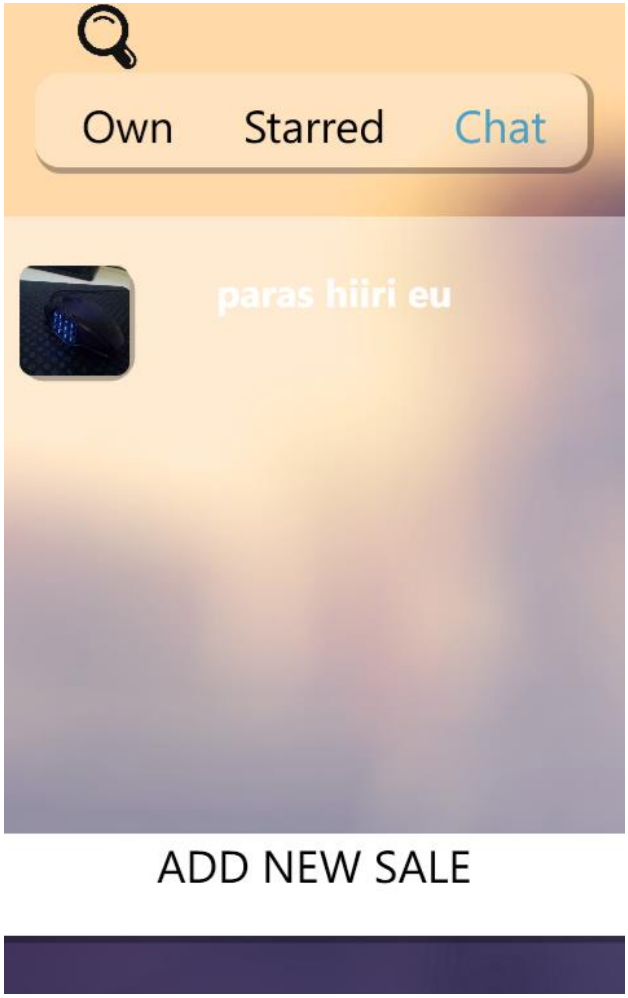
Suosikiksi lisättyjen tuotteiden avaaminen tulee onnistumaan painamalla haluttua tuotetta kuvassa (kuva 9) näkyvästä listasta. Tämä avaa tuotesivun jossa näkyy tuotteesta tarkemmat tiedot. Lista on hyvin samankaltainen kuin omien tuotteiden lista. Ainoana erona näiden listojen välillä on, että suosikkien listassa tuotteesta näkyy myös suoraan hinta. Omien tuotteiden listauksessa tuotteen hinta ei näy suoraan tuotelistauksessa.

Suosikiksi lisätyt tuotteet eivät tule katoamaan käyttäjältä, vaikka tämä vaihtaisi laitetta. Suosikkilista tullaan päivittämään aina käyttäjän kirjautumisen yhteydessä palvelun tietokannasta.

KUVA 9. Suosikit-lista

2.1.4 Chat

Ohjelmistosta oleva chat-ominaisuus on tarkoitettu helppoon yhteydenpitoon ostajien ja myyjien välillä. Jokainen käynnissä oleva chat tulee löytymään chat-välilehdeltä (kuva 10). Chatin vanhennettua se katoaa itsestään chat-välilehdeltä, eivätkä viestit ole enää luettavissa. Mahdollisia syitä chatin vanhentumiseen ovat: tuotteen poistuminen myynnistä, chat-viestien vanhentuminen. Chat-viesteille on tietokannassa asetettu elinaika, jonka jälkeen ne poistetaan automaattisesti.



KUVA 10. Chat-välilehti

Chat-välilehdellä tuotetta painettaessa tulee avautumaan kyseisen tuotteen omistajan kanssa Chat-ikkuna (kuva 11). Chat-välilehdellä tuotteiden listaus on vastaava kuin muissakin listanäkymissä.

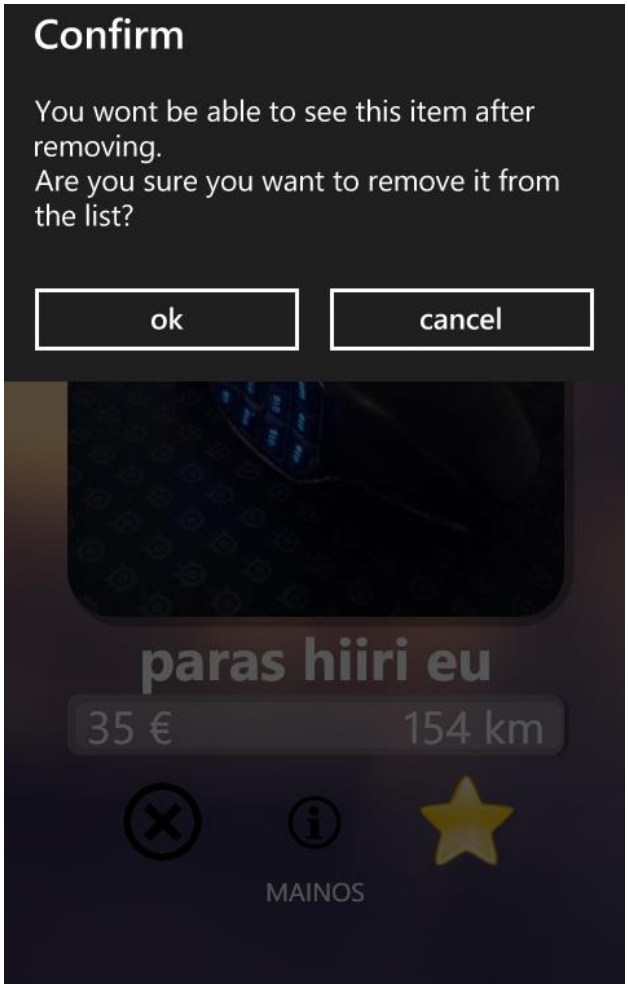
Chat-ikkunassa (kuva 11) näkymä on pyritty pitämään yksinkertaisena ja helppolukuisena. Ylimpänä näkyy kenen kanssa käyttäjä on keskustelemassa ja tämän alla tuotteen nimi, josta kauppaa ollaan käymässä. Tämän jälkeen näkyvät chat-viestit sijoittuvat niin, että käyttäjän omat viestit ovat oikealla ja keskustelukumppanin viestit tulevat olemaan vasemmalla.

KUVA 11. Chat-ikkuna.

2.1.5 Tuotteiden poisto selaustuloksista

Tuote-ilmoituksien poistaminen selaustuloksista tulee olemaan mahdollista painamalla tuotesivulla tai tuotekortti-selauksessa näkyvää rasti-ikonia (kuva 4). Tämän toiminnon tarkoitus mahdollistaa käyttäjille turhien ilmoitusten piilottamista selaustuloksista, mikä parantaa käyttäjien selauskokemusta. Tämä ominaisuus ei tee itse tuotteelle mitään, vaan tämän ominaisuuden avulla käyttäjä pystyy piilottamaan tuotteen omista hakutuloksistaan. Mikäli käyttäjä tekee useampia toisistaan vain vähän poikkeavia hakuja, ei tule sama käyttäjälle turha tuote näkymään uudestaan.

Rasti-ikonia painettaessa tulee avautumaan vahvistusviesti (kuva 12), jolla varmistetaan, että käyttäjä todella haluaa piilottaa tuote-ilmoituksen hakutuloksista. Näin vältytään vahingossa tapahtuvilta piilotuksilta.

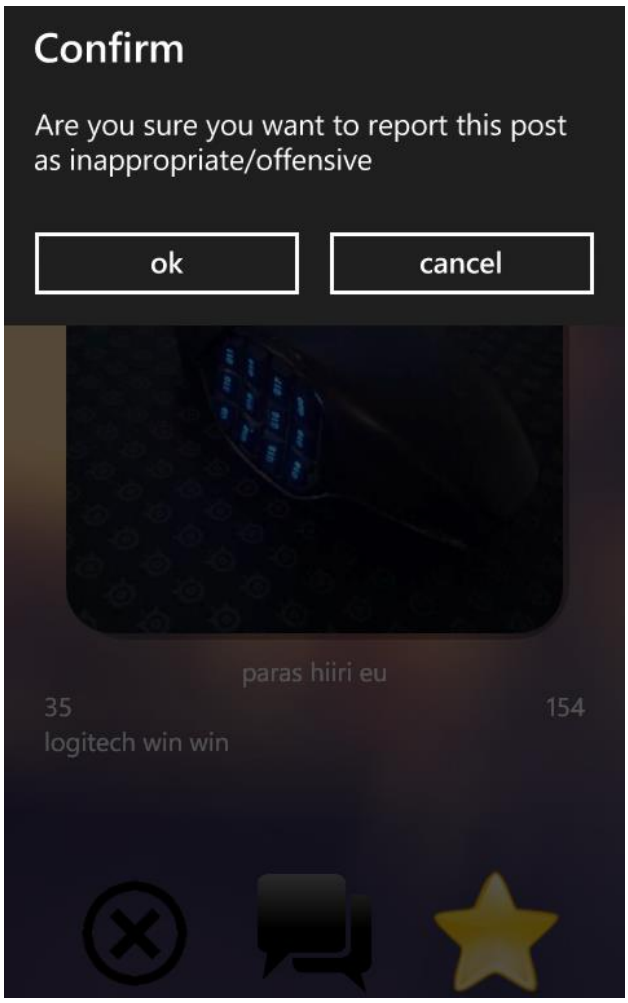


KUVA 12. Tuotteen piilottaminen hakutuloksista.

2.1.6 Tuotteiden ilmianto

Mikäli palvelusta löytyy käyttäjien mielestä asiatonta sisältöä, käyttäjillä on mahdollisuus ilmiantaa nämä ilmoitukset. Mikäli yksittäistä tuotetta ilmiannetaan tarpeeksi monta kertaa eri käyttäjien toimesta, tuote piilotetaan hakutuloksista automaattisesti siihen asti, että palvelun ylläpitäjät tarkistavat kyseisen ilmoituksen. Ilmiantaminen tehdään mahdollisimman nopeaksi, jottei käyttäjältä kuluisi tähän suhteettoman paljon aikaa. Tämän tarkoituksena on kannustaa ihmisiä auttamaan pitämään palvelu puhtaana asiattomista ilmoituksista, luoden näin mukavamman käyttäjäympäristön.

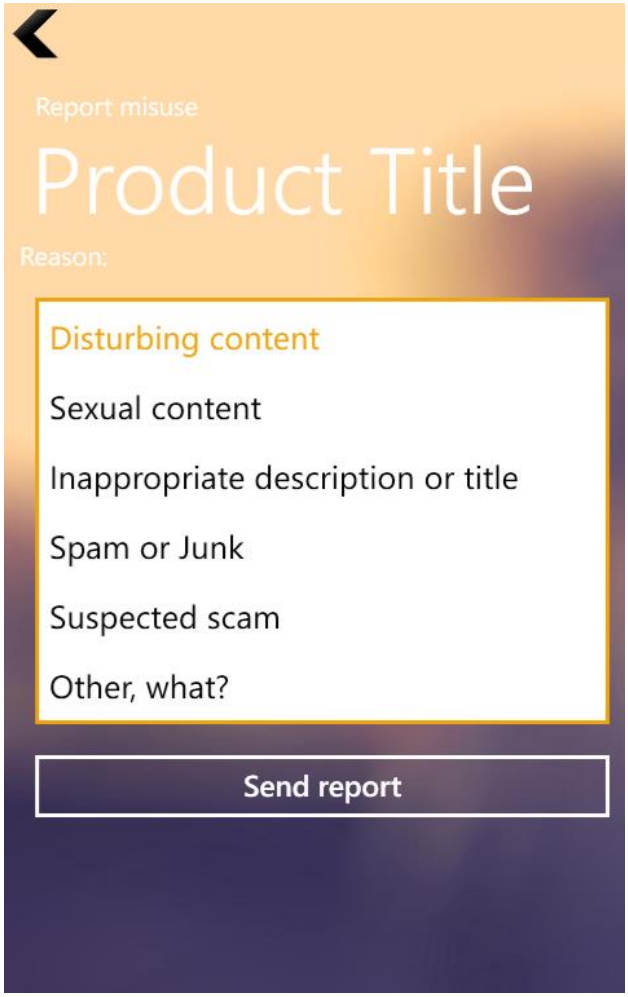
Ilmiantaminen aloitetaan painamalla tuotesivulla näkyvää huutomerkki-ikonia (kuva 4). Tämän painaminen näyttää varmennusikkunan (kuva 13), jolla varmistetaan, että käyttäjä haluaa varmasti ilmiantaa kyseisen tuotteen.



KUVA 13. Ilmianto-kuvaketta painettu.

Käyttäjän painettua varmennusikkunasta ok-painiketta, käyttäjä siirretään suoraan ilmiantosivulle (kuva 14). Ilmianto-sivulla on pidetty elementtien määrä minimissä ja näin ollen pidetty sivun ulkoasu selkeänä ja helppolukuisena.

Ilmianto onnistuu helposti pudotusvalikosta valittavalla syyllä (kuva 14) ja mikäli sopivaa vaihtoehtoa ei yleisimmistä vaihtoehdoista löydy, voi käyttäjä valita vaihtoehdon ”muu syy” ja kirjoittaa tämän syyn valittuna ollessa näkyvään tekstikenttään oman valintaisen, kuvaavan syyn.



KUVA 14. Ilmiannon syy.

2.2 Markkinat

Vertaismyyntimobiilisovelluksen kaltaiselle sovellukselle on tilaajan mielestä hyvin tilaa markkinoilla, sillä sovelluksen käyttö tehdään niin yksinkertaiseksi, että se onnistuu kokemattomimmaltakin. Vertaismyyntimobiilisovellus voi tavoittaa uusia käyttäjiä ja saada jo olemassa olevien palvelujen käyttäjiä siirtymään käyttämään vertaismyyntimobiilisovellusta korostamalla sen uusia ominaisuuksia. Tällaisia ominaisuuksia ovat esimerkiksi lokaatiopalvelun keskeinen käyttö ja sovelluksen sisäinen chat, joita olemassa olevista kauppapaikoista ei löydy. Markkinointia ajatellen suurin haaste on saada muiden palveluiden jo vakiintuneet käyttäjät kokeilemaan ja mahdollisesti vaihtamaan kilpailevasta palvelusta käyttämään vertaismyyntimobiilisovellusta.

2.2.1 Kohderyhmä

Ohjelmistolle ei ole suunniteltu varsinaista kohderyhmää, vaan ohjelmiston tarkoituksena on olla luonteva ja sopiva jokaiselle ohjelmiston käyttäjälle. Ohjelmiston graafinen käyttöliittymä on kuitenkin suunniteltu pääsääntöisesti silmällä pitäen 15-45 vuotiaita käyttäjiä. Ohjelmiston vaihtuva värimaailma pyritään pitämään sellaisena, ettei se suosi esimerkiksi yhtä pientä ikähaarukkaa, vaan on mahdollisimman silmää miellyttävä jokaiselle käyttäjälle. Värimaailmaa päivitetään ylläpitäjien toimesta esimerkiksi vuodenaikojen ja juhlapyhien mukaan.

2.2.2 Kilpailijat

Huuto.net

Huuto.net on vuonna 1999 Helsingissä perustettu, tällä hetkellä suomen suurin ja toiseksi suosituin vertaiskauppapaikka. Huuto.netissä voidaan myydä ja ostaa tuotteita suoraan muilta palvelun käyttäjiltä. Huuto.net oli alun perin nettihuutokauppa, mutta nykypäivänä Huuto.netissä on mahdollista julkaista tavallisia kiinteän hinnan myynti-ilmoituksia. Huuto.netissä on myös mahdollista tehdä tuotteista tarjouksia edullisemmalla hinnalla, kuin ilmoituksessa ilmoitettu myyntihinta. Huuto.netissä olevat ilmoitukset on kategorisoitu tuoteryhmittäin ja palvelu tarjoaa niiden etsimiseen useita hakumahdollisuuksia.

Vuonna 2012 huutonetin palvelun kautta solmittiin yli 3,2 miljoonaa varmennettua kauppaa, joiden yhteenlaskettu myyntihinta oli jopa 89 miljoonaa euroa. Vuonna 2013 kauppojen määrä väheni 2,6 miljoonaan ja näiden kauppojen yhteenlaskettu summa oli 76 miljoonaa euroa, eli 13 miljoonaa vähemmän kuin aiempana vuonna. Palvelussa voi kerrallaan olla tarjolla yli 1,5 miljoonaa kohdetta. (Markkinointi&Mainonta: Huuto.net 15 vuotta; Huuto.net: Blogit.)

Tori.fi

Tori.fi on internetissä palveleva, kuluttajille maksuton osto- ja myyntikauppapaikka, Tori.fi julkaisee yksityishenkilöiden luokiteltuja ilmoituksia ilman maksua. Alkukeväästä

vuonna 2013 Tori.fi ohitti Huuto.netin suosikkina internetin vertaiskauppapaikoista ja sijoittuu nyt suomalaisten verkkosivujen vertailussa neljänneksi suosituimmaksi yli 40 miljoonalla viikoittaisella sivulatauksellaan (TNS Metrix, vko 4, 2014). Joulukuussa 2009 avattu Tori.fi kuuluu norjalaiselle Schibsted konsernille. Sivusto on osa konsernin yli 30 maassa toimivaa kategorioittain luokiteltujen ilmoitusten verkkosivustoketjua, joka on toteutettu ruotsin suosituimman Blocket.se palvelun pohjalta. Kyseisen konsernin verkkokauppa on suosituimpana myös Ranskassa ja Espanjassa. (Kauppalehti: Tori.fi.)

2.2.3 Erot kilpailijoihin

Vertaismyyntimobiilisovellus tulee erottumaan kilpailijoistaan yksinkertaisuudellaan ja suoraviivaisuudellaan. Ohjelmistossa laajojen toiminnallisuuksien sijaan keskitytään käyttöliittymän yksinkertaisuuteen ja ohjelmiston käytön helppouteen. Vertaismyyntimobiilisovelluksen tärkeimpiä ominaisuuksia tulee olemaan se, että sen käyttö onnistuu kaikilta älypuhelinten omistajilta ilman suurempia ongelmia. Lisäksi suurimpana erona kilpailijoihin nähden tulee olemaan sovellukseen integroitu chat-ominaisuus, joka tulee nopeuttamaan ostajan ja myyjän välistä kommunikaatiota selvästi.

3 ARKKITEHTUURI

Tässä kappaleessa perehdytään sovelluksen arkkitehtuuriseen suunnitteluun ja toteutukseen, tutkitaan sovelluskehityksessä käytettyjä toimintatapoja sekä perehdytään suunnittelu vaiheessa luotuihin UML-kaavioihin. Arkkitehtuuria tarkastellaan Philippe Kruchtenin suunnittelemalla 4+1 näkymää suunnittelumalliin pohjautuvalla tavalla. Philippe Kruchtenin malli valittiin sen selkeyden vuoksi. Läpikäytävät näkymät ovat;

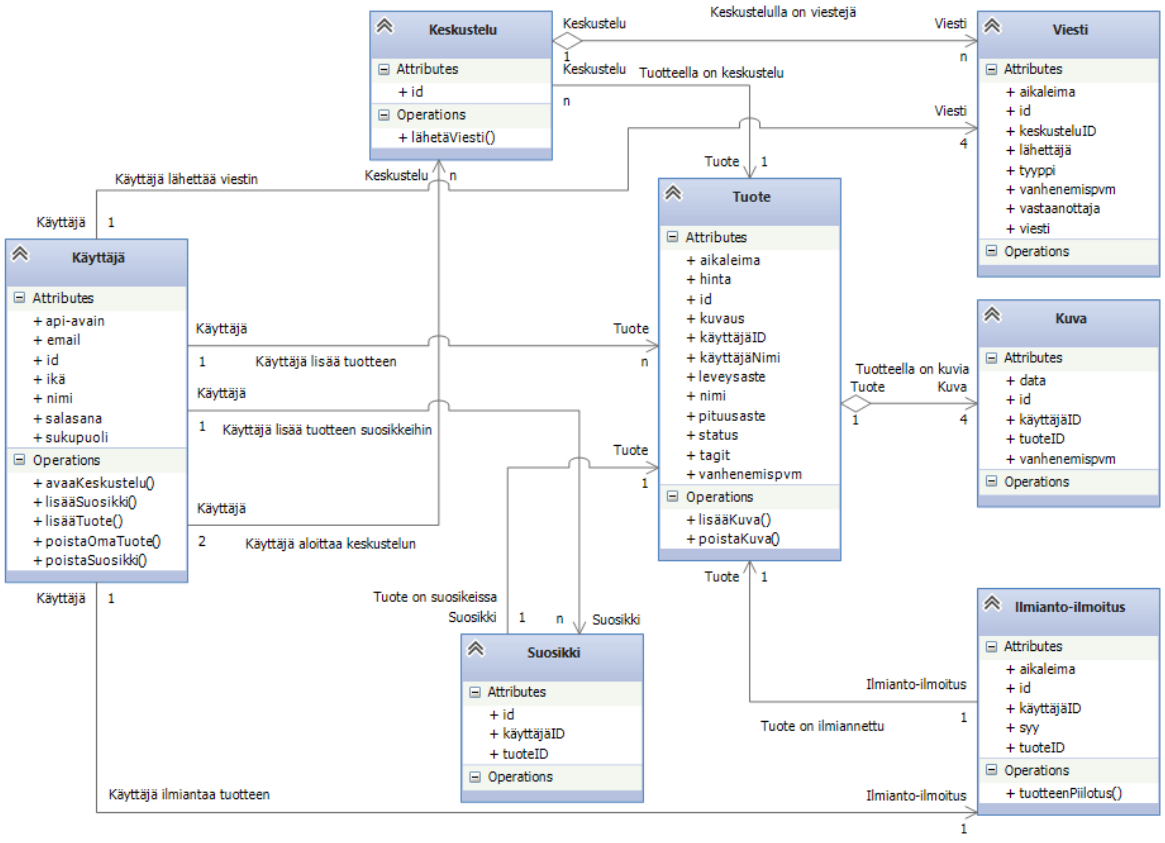
Looginen näkymä, joka sisältää toiminnallisuudet, jotka järjestelmä mahdollistaa käyttäjille. UML-kaaviot, joita käytetään loogisessa näkymässä, ovat mm. luokkakaavio, kommunikaatiokaavio ja sekvenssikaavio.

Prosessinäkymä, joka tarkastelee järjestelmän dynaamisia aspekteja, selittää järjestelmän prosesseja, kuinka ne kommunikoivat keskenään ja järjestelmän ajonaikaista käyttäytymistä. Prosessinäkymä tarkastelee lisäksi samanaikaisuutta, jakelua, integraattoreita, suorituskykyä ja skaalautuvuutta. Prosessinäkymää kuvataan muun muassa UML-aktiviteettikaaviolla.

Skenaariossa arkkitehtuurin määrittelyä kuvaillaan pienillä käyttötapauskaavioilla. Skenaariot kuvaavat vuorovaikutus sekvenssejä olioiden ja prosessien välillä. Niitä käytetään identifioimaan arkkitehtuurisia elementtejä ja kuvaamaan sekä varmentamaan arkkitehtuurista suunnittelua. Tämä näkymä tunnetaan myös nimellä käyttötapausnäkymä. (Philippe Kruchten: 4+1 view model.)

3.1 Looginen näkymä

Ohjelmistokehityksessä luokkakaavio on staattinen rakennekaavio, joka kuvailee järjestelmän staattista rakennetta esittämällä sen luokkia, operaatioita ja olioiden välisiä yhteyksiä. Luokkakaavio on oliopohjaisen suunnittelun tärkein kulmakivi. Kaavioita voi käyttää myös tiedon mallintamiseen. Alla nähdään vertaismyyntimobiilisovelluksen luokkakaavio (kuva 15).



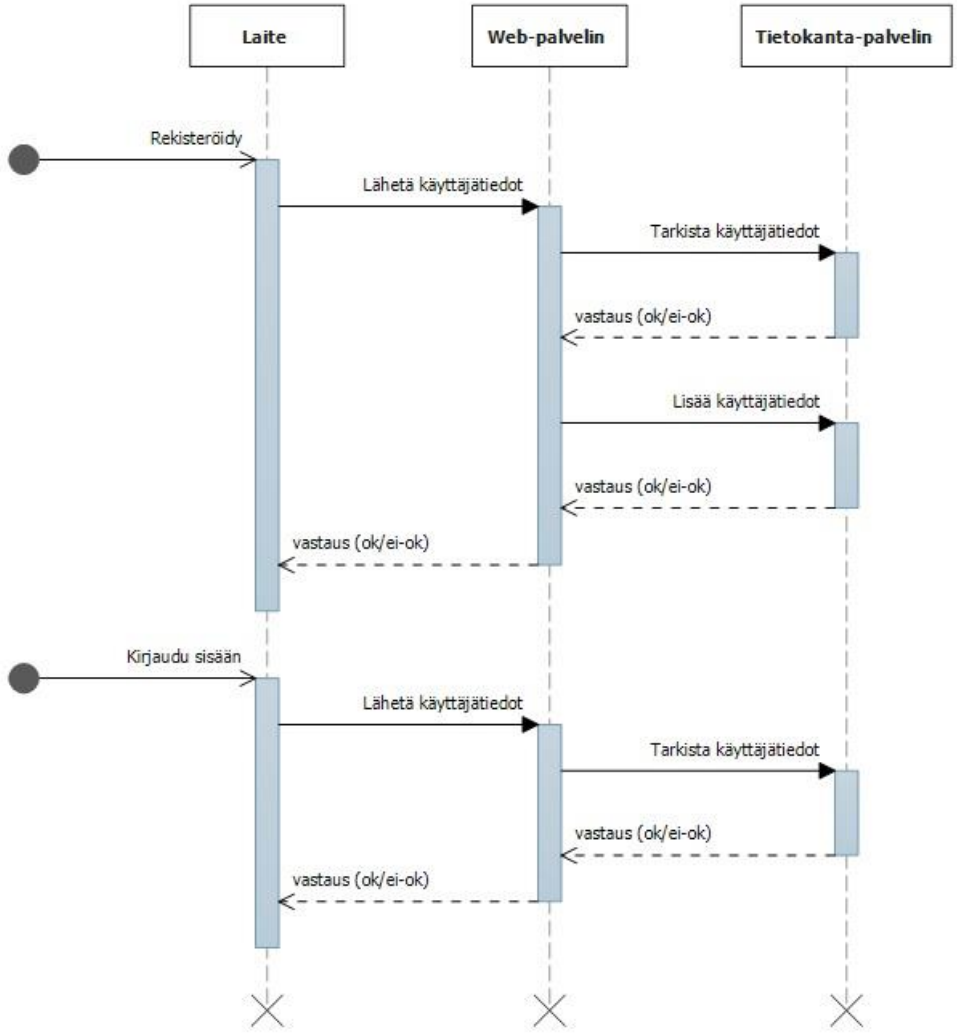
KUVA 15. Luokkakaavio.

Katsotaan esimerkiksi käyttäjän ja tuotteen välistä yhteyttä. Tuotteen ja käyttäjän välinen viiva tarkoittaa näiden välistä yhteyttä, joka kertoo sen, että tuotetta ei voi olla ilman sen lisännyttä käyttäjää. Yhteyden ohessa esitetyt numero yksi (1) ja kirjain n ilmoittaa näiden välisestä yhteydestä sen, että käyttäjällä voi olla useita tuotteita, mutta tuotteella voi olla vain yksi käyttäjä. Kaaviossa on yhteyksiä esitetty myös katkoviivoilla, joka ilmoittaa että luokka on riippuvainen toisesta niin, ettei sitä voi olla olemassa ilman luokkaa, josta se on riippuvainen.

Kuten luokkakaaviosta (kuva 15) käy ilmi, on käyttäjä hyvin keskeisessä osassa vertaismyyntimobiilisovellusta. Jokainen luokka on jollain tavalla riippuvainen käyttäjästä. Tämän jälkeen tärkeänä osana sovellusta tulevat tietenkin tuotteet. Järjestelmän ylläpidon helpottamiseksi on myös lisätty käyttäjille mahdollisuus ilmiantaa tuotteita, mikä vähentää ylläpitäjän tarvetta jatkuvasti tarkistaa tuotteiden sääntöjen mukaisuutta.

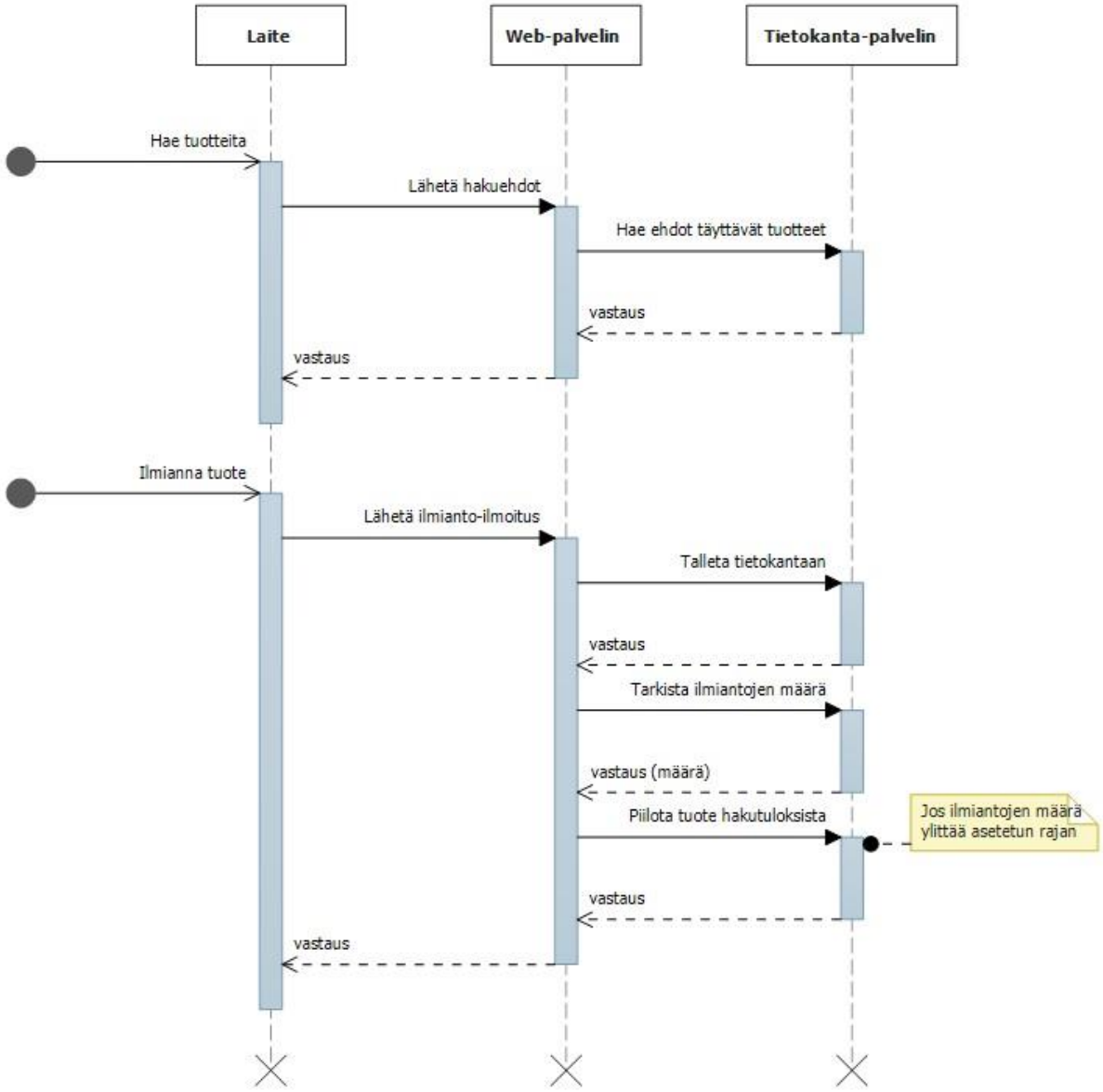
Sekvenssikaavio on vuorovaikutuskaavio, joka näyttää kuinka ja missä järjestyksessä prosessit operoivat toistensa kanssa. Sekvenssikaavio näyttää olioiden vuorovaikutukset aikajärjestyksessä. Se kuvaa skenaarioon kuuluvat oliot ja luokat sekä vuorovaikutuksessa lähetetyt viestit, jotka ovat välttämättömiä toiminnallisuuden ylläpitämiseksi.

Käydään esimerkkinä läpi käyttäjän rekisteröityminen (kuva 16). Käyttäjä haluaa rekisteröityä vertaismyyntimobiilisovellukseen, joten käyttäjä menee laitteella rekisteröitymissivulle ja täyttää rekisteröitymislomakkeen. Lomakkeelle laitetut tiedot lähetetään web-palvelimelta löytyvälle web-API:lle, joka tarkistaa tietojen olevan oikeassa formaatissa, jonka jälkeen web-palvelin tarkistaa tietokantapalvelimelta ovatko käyttäjätiedot sopivat tietokantaan. Tässä kohtaa ongelmana voi käytännössä tulla vain jo käytössä oleva sähköpostiosoite. Web-palvelimen saadessa positiivinen vastaus tietokannasta suorittaa web-palvelin käyttäjätietojen lisäämisen tietokantaan ja tämän onnistuessa informoi laitteelle rekisteröitymisen onnistuneen. Kirjautumisprosessi toimii vastaavalla menettelyproseduurilla.



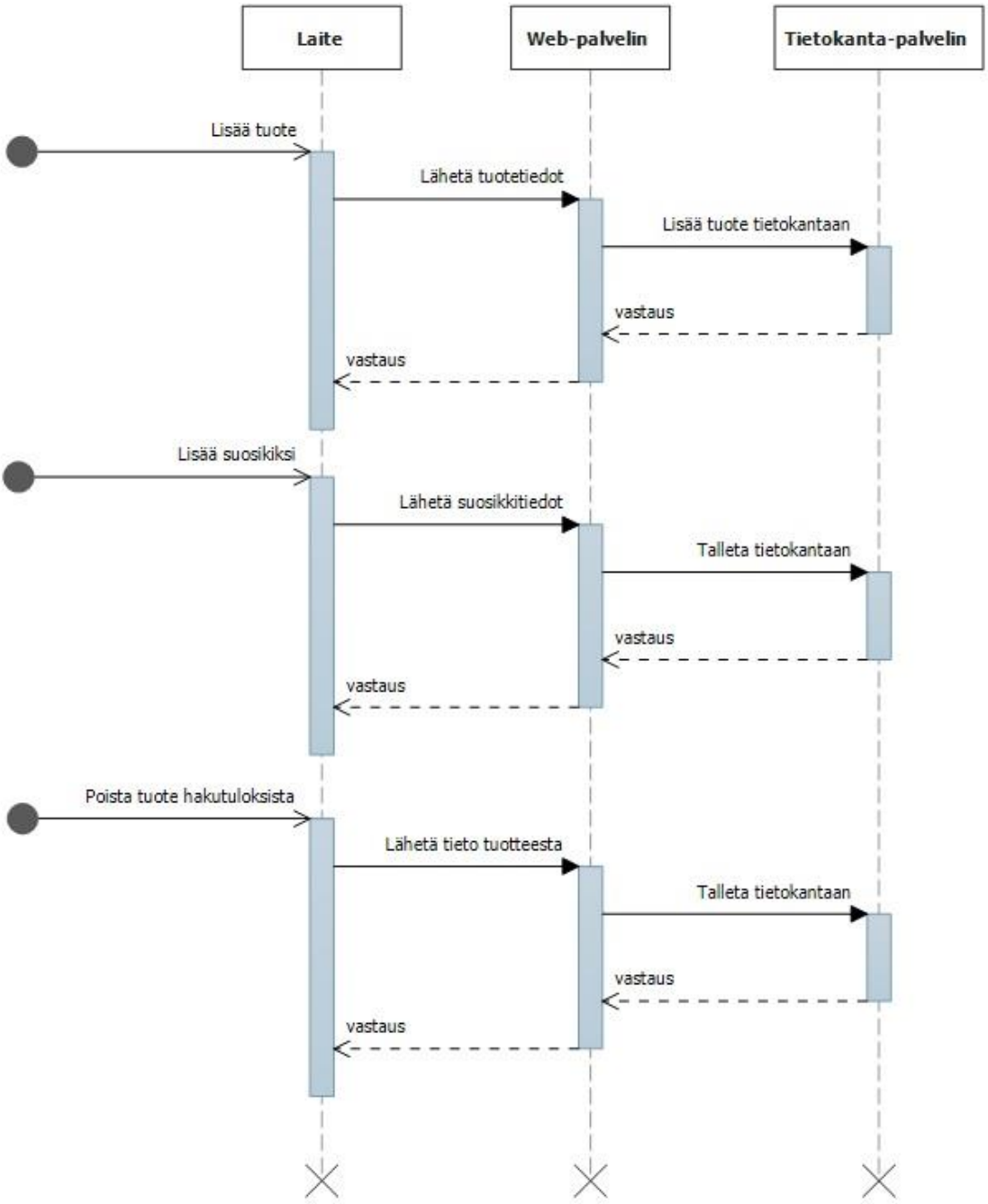
KUVA 16. Sekvenssikaavio käyttäjän kirjautumisesta

Tuotteiden ilmiannossa (kuva 17), sen jälkeen kun web-palvelin on suorittanut ilmiannon lisäämisen tietokantaan ja saanut positiivisen vastauksen suorittaa se vielä toisen kyselyn. Web-palvelin tarkistaa juuri raportoidun tuotteen ilmiantojen kokonaismäärän ja mikäli tämä määrä ylittää ylläpitäjän toimesta määritellyn rajan, suorittaa web-palvelin toimenpiteen, jolla kyseistä tuotetta ei tule enää löytymään hakutuloksista, kuitenkaan poistamatta tuotetta kokonaan. Ylläpitäjä voi tarkistaa tuotteen rauhassa ja päättää tämän jälkeen jatkotoimenpiteistä.



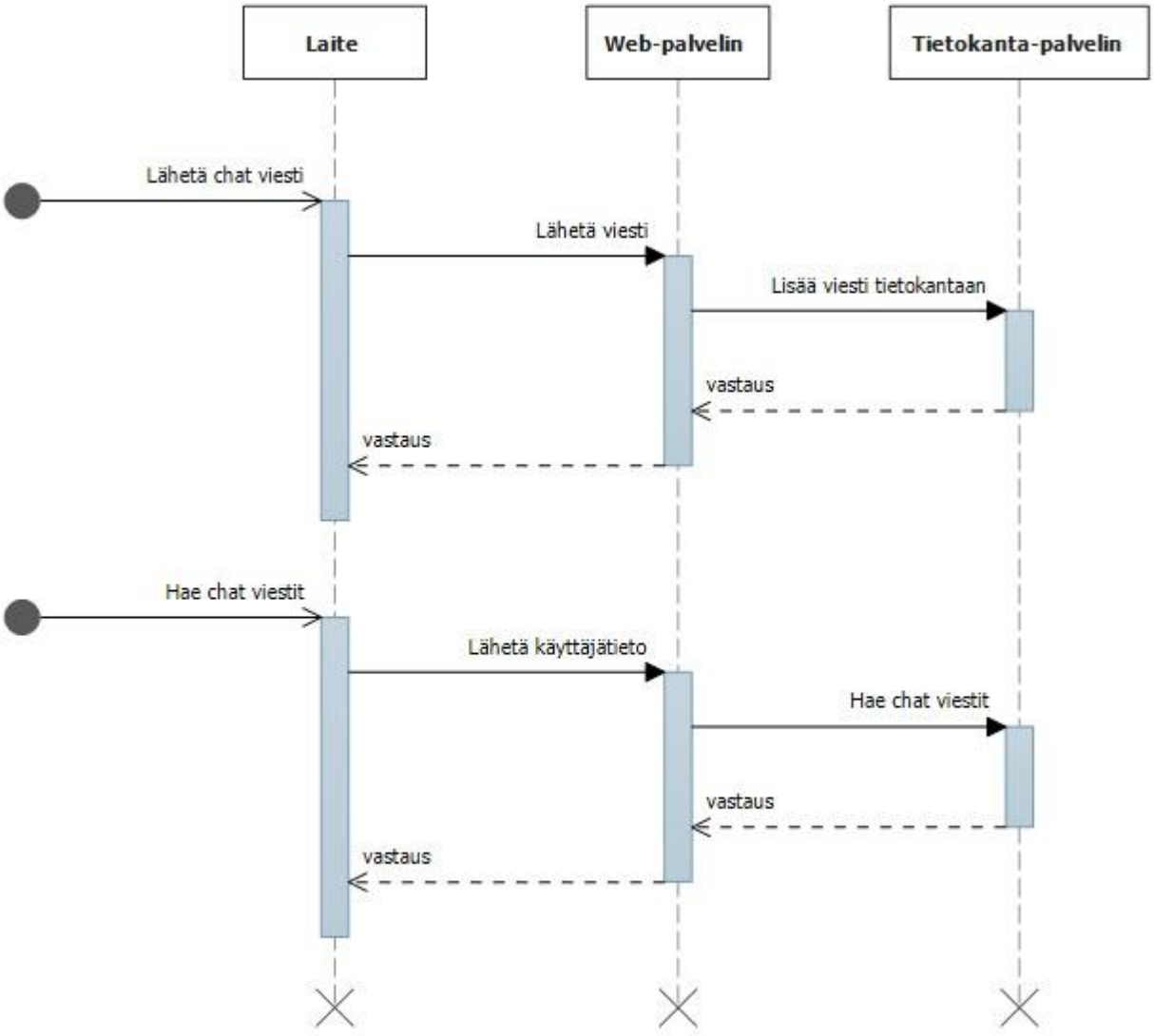
KUVA 17. Sekvenssikaavio tuotteiden hausta ja ilmiannosta

Tuotteiden lisäys, suosikiksi asettaminen ja hakutuloksista poistaminen (kuva 18) ovat järjestelmän kannalta yksinkertaisia prosesseja ja prosessien sekvenssit toteuttavat jo aiemmin tutuksi käynyttä kaavaa.



KUVA 18. Tuotteiden lisääminen, suosikiksi lisääminen ja poisto hakutuloksista

Chat viestien lähetys- ja hakuprosessi (kuva 19) toteuttavat tutuksi käynyttä yksinkertaista sekvenssiä. Käyttäjä lähettää laitteella pyynnön web-palvelimelle, joka suorittaa toimenpiteet tietokantaan ja antaa vastauksen takaisin laitteelle.



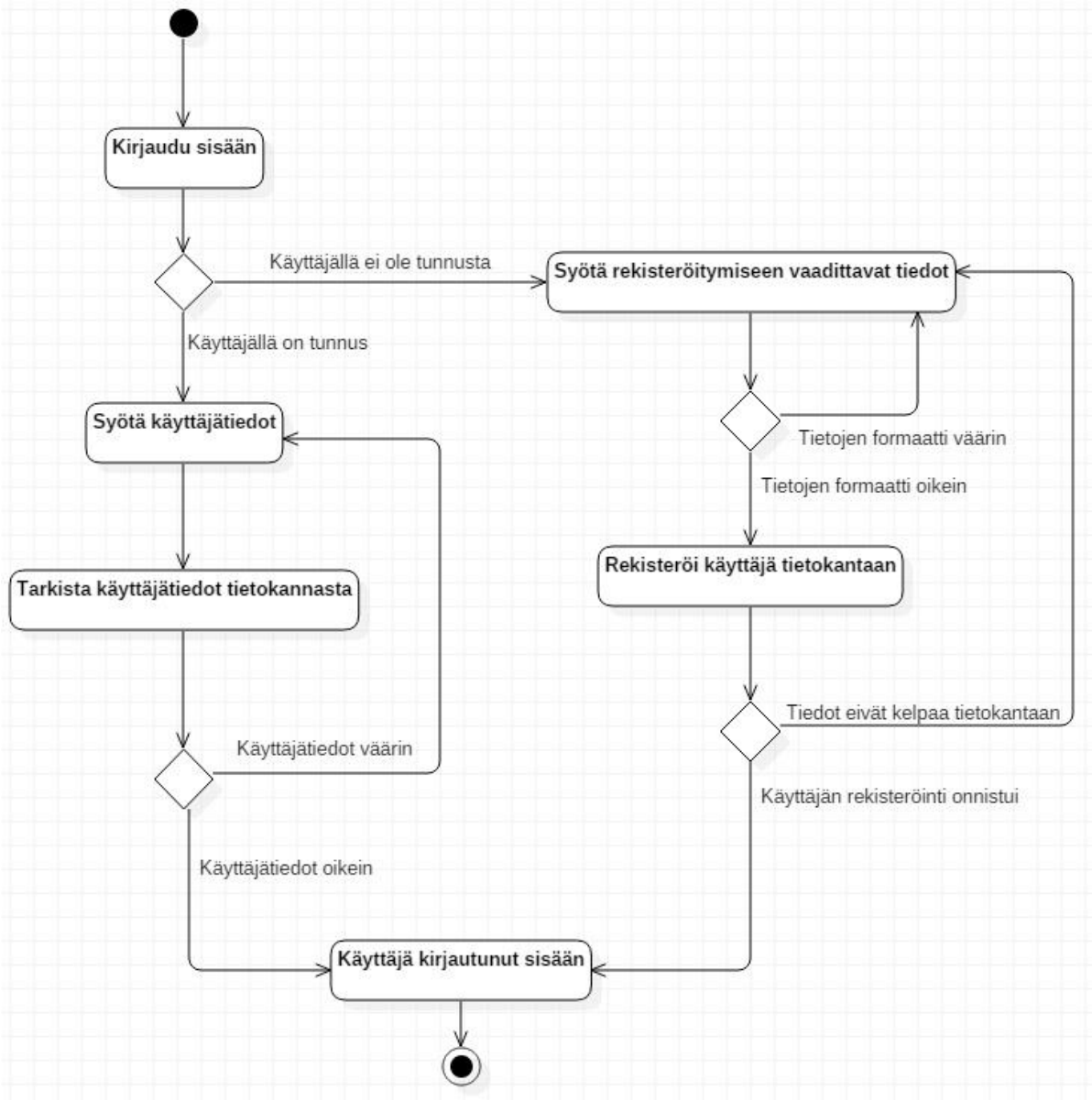
KUVA 19. Sekvenssikaavio chatistä

3.2 Prosessinäkymä

Aktiviteettikaaviot ovat graafisia esityksiä prosessien kulun tapahtumista. Niistä ilmenee askel kerrallaan kuinka prosessit etenevät ja millaisia valintoja mahdollisesti prosessien aikana tapahtuu. Alla on kuvattu vertaismyyntimobiilisovelluksen keskeisimmät prosessit aktiviteettikaavioina.

Alla olevassa kuvassa (kuva 20) on kuvattu käyttäjän kirjautuminen palveluun aktiviteettikaavioina. Aluksi käyttäjä pyrkii kirjautua palveluun, palvelu kysyy käyttäjätunnusta, mikäli käyttäjällä ei tunnusta ole, ohjataan käyttäjä rekisteröitymään palveluun. Käyttäjän lähettäessä rekisteröintitiedot, tarkistetaan niiden oikeellisuus web-palvelimella, mikäli

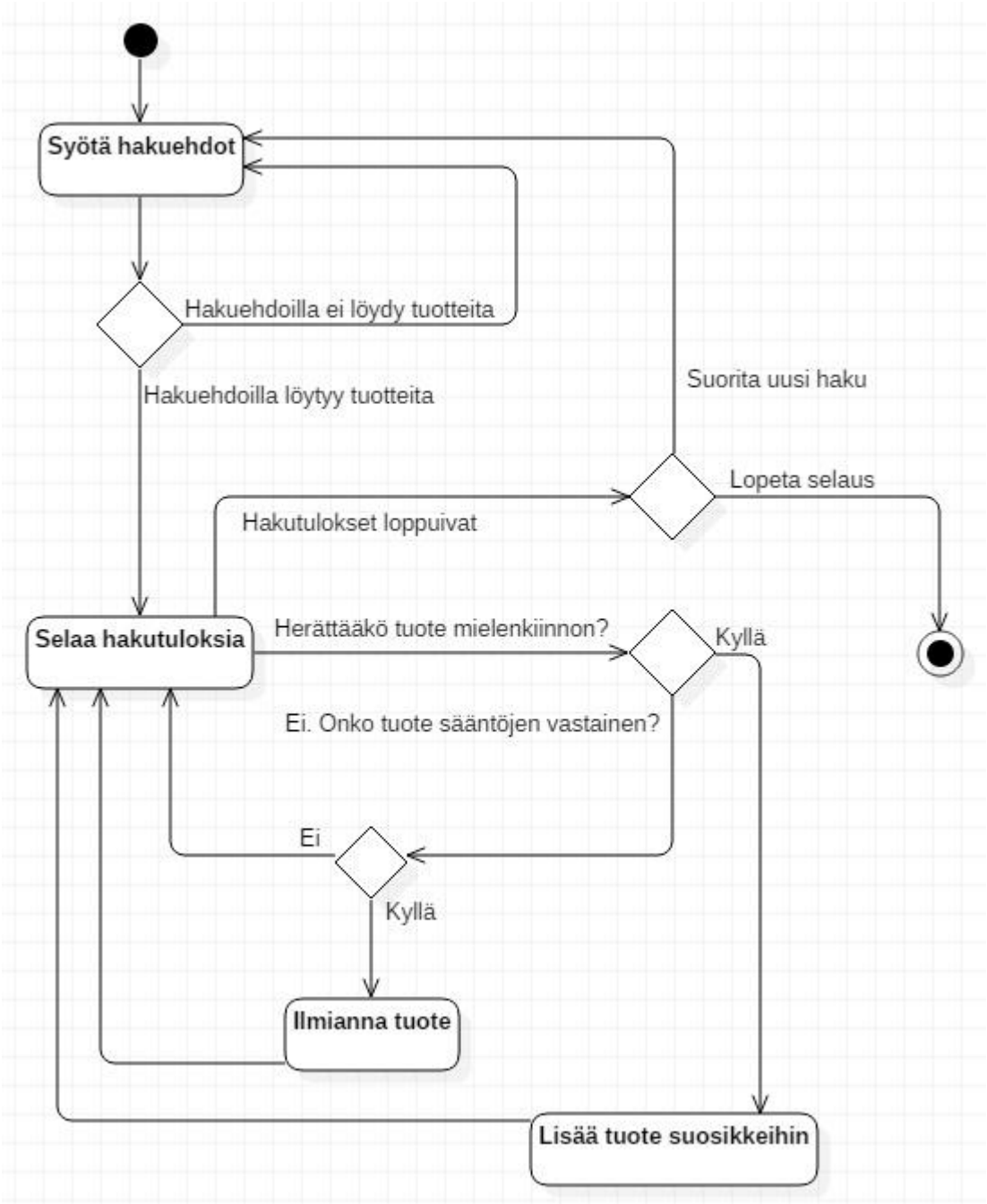
tiedot ovat hyväksyttävät, käyttäjä tallennetaan tietokantaan ja käyttäjä pääsee näin kirjautumaan tunnuksellaan palveluun. Mikäli käyttäjällä kuitenkin on jo tunnus, voi hän käyttää tätä kirjautumiseen. Kirjautumistiedot tarkistetaan web-palvelimella, mikäli tiedot ovat oikein, käyttäjä kirjautuu sisään palveluun.



KUVA 20. Aktiviteettikaavio sisäänkirjautumisesta

Kuvassa (kuva 21) on kuvattuna tuotteiden hakuun ja selailuun liittyvä prosessi aktiviteettikaaviona. Tuotteiden selaaminen alkaa sillä, että käyttäjä syöttää hakuehdot hakupalkissa ja suorittaa haun. Mikäli näillä hakuehdoilla löydetään tuotteita, pääsee käyttäjä selaamaan hakutuloksia. Aina uuden tuotteen kohdalla käyttäjällä on mahdollisuus lisätä

tuote suosikkeihinsa, mikäli se herättää käyttäjän mielenkiinnon ja toisaalta mikäli tuote vaikuttaa käyttäjän mielestä jotenkin sääntöjen tai hyvän maun vastaisilta, voi käyttäjä ilmiantaa tuotteen. Käyttäjä voi jatkaa tuotteiden selausta kunnes on selannut kaikki tuotteet, jonka jälkeen käyttäjä voi suorittaa uuden hau tai lopettaa tuotteiden selaamisen.

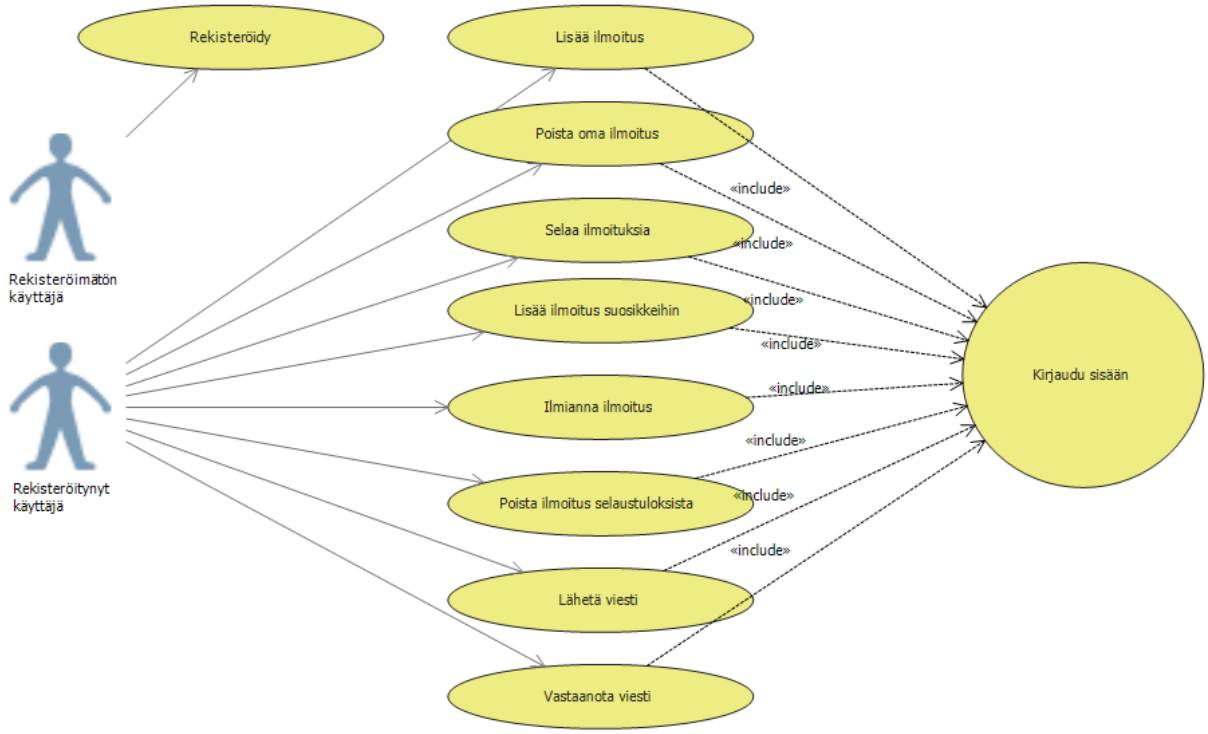


KUVA 21. Aktiviteettikaavio tuotteiden selaamisesta.

3.3 Skenaariot

Ohjelmistotekniikassa käyttötapauskaavio listaa mahdolliset käyttötapaukset kullekin käyttäjälle. Tässä tapauksessa on vain kaksi käyttäjätyyppiä.

Alla olevasta kaaviosta (kuva 22) ilmenee kaikki vertaismyyntimobiilisovelluksen toiminnot. Kaaviosta näkee, että kaikki vertaismyyntimobiilisovelluksen toiminnot vaativat käyttäjän rekisteröitymisen ja sisäänkirjautumisen muiden toimintojen käyttämiseen.



KUVA 22. Käyttötapauskaavio

4 KÄYTETYT TEKNOLOGIAT JA TYÖKALUT

Tässä luvussa käydään läpi vertaismyyntimobiilisovelluksen toteutuksessa käytettävistä teknologioista ja työkaluista.

4.1 C#

Ohjelmisto on tarkoitus toteuttaa natiivina Windows Phone alustalle, joten C# oli siihen ainoa mahdollinen valinta. Mikäli kyseessä ei olisi natiivisovellus, vaihtoehtoja olisi huomattavasti useampia.

C# (lausutaan c sharp) on ohjelmointikieli, joka on suunniteltu .NET-Frameworkissä toimivien ohjelmien luomiseen. C# on yksinkertainen, tehokas, tyyppiturvallinen ja oliopohjainen. Useat C#:n innovaatiot mahdollistavat nopean ohjelmistokehityksen säilyttämällä C-tyylisten ohjelmointikielien ilmaisukyvyn sekä eleganttiuden.

```
203                 data.iss5 = new ArrayList();
204
205                 //Parse first record
206                 int intSize = sizeof(int);
207                 int size1 = Marshal.SizeOf(firstRecord);
208                 //int size2 = Marshal.SizeOf(firstRecordType);
209                 //int size3 = sizeof(firstRecordType);
210                 byte[] buf = new byte[size1];
211                 byte[] buf2 = new byte[160];
212                 buf = b.ReadBytes(Marshal.SizeOf(firstRecord));
213                 firstRecord = (firstRecordType) RawDeserialize(buf,
    firstRecord.GetType());
```

KUVA 23. Esimerkki C#-ohjelmointikielestä

C#:n syntaksi on kovin tarkka, mutta kuitenkin helppo ja yksinkertainen oppia. C# syntaksi on välittömästi tunnistettavissa kaikille ohjelmistokehittäjille, jotka ovat olleet tekemisissä C, C++ tai Javan kanssa. Ohjelmistokehittäjät, jotka osaavat jotain näistä ohjelmointikielistä, pystyvät tyypillisesti työskentelemään tuottavasti C# parissa hyvin lyhyen ajan kuluttua. C# syntaksi yksinkertaistaa C++ ohjelmointikielen monimutkaisuuksia ja tarjoaa tehokkaita ominaisuuksia kuten nollattavia arvotyyppejä, listauksia, delegaatioita, lambda-ilmaisuja ja DMA, joita ei löydy Javasta. C# tukee yleisiä metodeja ja tyyppejä, jotka mahdollistavat lisääntyneen tyyppiturvallisuuden, suorituskyvyn ja iteraattorit.

Olioihin perustuvana ohjelmointikielenä, C# tukee kapseloinnin, periytymisen ja polymorfismin konseptia. Kaikki muuttujat ja metodit, mukaan lukien päämetodi, ohjelmiston aloituspisteet ovat kapseloituna luokkamäärittelyissä. Luokka voi periytyä suoraan yhdestä sen äitiluokasta, mutta se voi myös sisällyttää kuinka monta tahansa rajapintaa. Metodit, jotka kumoavat virtuaalisia metodeja äitiluokassa vaativat ohitus avainsanan välttääkseen vahingollista uudelleen määrittelyä. C#:ssa tietue on kuin kevyt luokka, se on pinojakoinen tyyppi, joka voi sisällyttää rajapintoja mutta ei tue periyttämistä. (Microsoft: Introduction to the C#.)

4.2 XAML

XAML on tarkoitettu natiivien Windows Phone sovellusten staattisen käyttöliittymien rakentamiseen, joten se on ainoa kieli tukemaan C# natiivinohjelmiston kehityksessä.

XAML on osa Microsoftin WPF:ää. WPF on kategoria ominaisuuksia Microsoftin .NET Framework 3.5:sta, jotka käsittelevät visuaalista esitystapaa Windows pohjaisista ohjelmistoista ja web-selain pohjaisista asiakasohjelmistoista. WPF pohjaisia ohjelmistoja voidaan suorittaa Windows Vistalla tai Windowsin aiemmissa versiossa, joissa Microsoft .NET framework 3.5 on asennettu tai Internet Explorer 7.0 asennettuna mikäli kyseessä on Web-selainpohjainen ohjelmisto.

```
<UserControl
    x:Class="NotReferencedFormsLibrary.UserControl1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="300" VerticalAlignment="Stretch">
    <Grid Background="{StaticResource redBrush}">
        <Border Margin="50" Background="{StaticResource greenBrush}">
            <TextBlock
                Foreground="{StaticResource yellowBrush}"
                Text="See both colors are available"
                TextAlignment="Center"
                VerticalAlignment="Center"
                FontSize="16"
                TextWrapping="Wrap" />
        </Border>
    </Grid>
</UserControl>

```

KUVA 24. Esimerkki XAML-kielestä

WPF käyttää XAMLia visuaalisesti näyttävien käyttöliittymien rakentamiseen merkintäkielellä ohjelmointikielen kuten C# sijaan. XAMLilla voi luoda käyttöliittymäelementit kuten kontrollerit, tekstit, kuvat, muodot, animaatiot ja muut. Koska XAML on deklaratiivinen kieli, se tarvitsee ohjelmistokoodin toimimaan taustalle, mikäli ohjelmistoon halutaan lisätä ajonaikaista logiikkaa. Pelkkää XAMLia käyttämällä voit vain luoda ja ainmoida käyttöliittymä elementtejä. Mikäli elementit halutaan vastaavan käyttäjän syötteisiin, tarvitaan ohjelmistokoodia taustalle. Ohjelmistokoodi XAML ohjelmistolle säilytetään erillisessä tiedostossa. Tämä erottaminen käyttöliittymän suunnittelun ohjelmistokoodista mahdollistaa ohjelmoijien sekä suunnittelijoiden toimia lähemmin yhteistyössä samassa projektissa viivästymättä toisiaan. (Microsoft: XAML.)

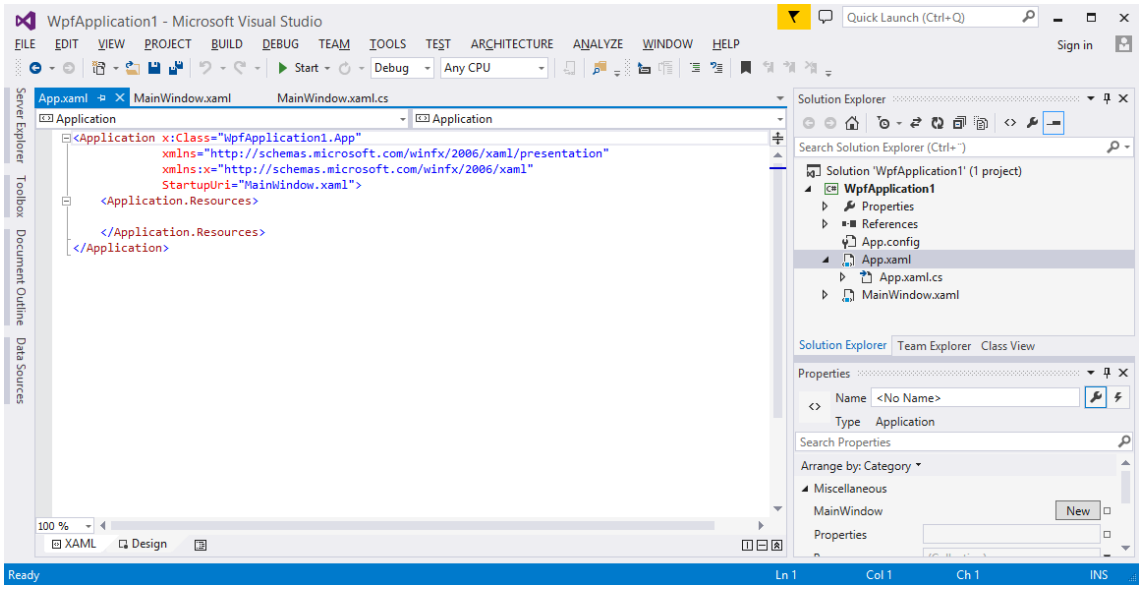
4.3 Visual Studio

Visual Studio tarjoaa kattavan tuen C# ja XAML -kielille ja Windows Phone kehitykselle, joten se oli paras valinta kehitystyökaluksi.

Microsoft Visual Studio on integroitu kehitysympäristö, jonka Microsoft on kehittänyt. Sitä käytetään ohjelmistojen kehittämiseen Microsoft Windows käyttöjärjestelmään. Visual Studiolla voi myös kehittää web-sivuja, web-ohjelmistoja ja web-palveluita. Visual Studio käyttää Microsoftin kehitysalustoja kuten Windows API, Windows Forms, Windows Presentation Foundation, Windows Store ja Windows Silverlight.

Visual Studiossa on sisäänrakennettuna koodieditori, joka tukee Microsoftin IntelliSenseä, sekä ohjelmistokoodin refaktorointia. Muut sisäänrakennetut ominaisuudet sisältävät työkalut mm. graafiselle lomakesuunnittelulle, web-suunnittelulle, luokkien suunnitteluun ja tietokantarakenteen suunnitteluun. Visual Studioon voi liittää useita erilaisia pluginejä, joilla voi tehostaa ja parantaa Visual Studion toimivuutta lähes jokaisella tasolla, kuten esimeriksi lisätä tuen versionhallinnalle. (Microsoft: Visual Studio.)

Alla olevassa kuvassa (kuva 25) näkyy ruutukaappaus Visual Studion XAML kehitys näkymästä. Oikeassa laidassa näkyy projektin tiedosto hierarkia.



KUVA 25. Ruutukaappaus Visual Studiosta

4.4 MongoDB

MongoDB valittiin vertaismyyntimobiilisovelluksen tiedonvarastointijärjestelmäksi MongoDB:n joustavuuden, skaalautuvuuden ja nousevan suosion takia (DB-Engines: Ranking). MongoDB:stä uskotaan olevan hyötyä myös tulevaisuudessa ja sen oppiminen koetaan hyväksi.

MongoDB on NoSQL-tietokannaksi luokiteltava dokumentti-orientoitunut tietokanta. MongoDB välttää perinteistä relaatiotietokannan rakennetta suosimalla JSON-tyyppisiä dokumentteja ja dynaamisia malleja, joka tekee tietyntyyppisten ohjelmissa datan integroinnista nopeampaa ja helpompaa. MongoDB on ilmainen vapaanlähdekoodin ohjelma, joka on toteutettu C++, JavaScript ja C kielillä. MongoDB toimii useilla alustoilla ja monet suuretkin verkkosivustot ja palvelut ovat ottaneet sen käyttöön palvelinpään ohjelmaksi. Marraskuussa 2015 MongoDB oli neljänneksi suosituin tietokanta hallintajärjestelmä (DB-Engines: Ranking).

MongoDB:n keskeisiin ominaisuuksiin kuuluvat muun muassa sen dokumentti-orientoituneisuus, joka tarkoittaa sitä että tietoja ei jaeta useampaan relaatiorakenteeseen vaan tiedot tallennetaan minimaaliseen määrään dokumentteja. Toinen keskeinen ominaisuus on Ad hoc –kyselyt, eli tarpeen mukaan mukautuvat kyselyt. MongoDB:stä voidaan suorittaa kyselyitä kentän mukaan, aluekyselyitä ja kyselyitä regular expression:ien avulla.

Tuloksena voidaan palauttaa kaikkien kenttien sijaan myös jotkin tietyt kentät dokumentista. MongoDB tukee myös dokumenttien indeksointia.

MongoDB tarjoaa myös korkean saatavuuden kopiosarjoja. Kopiosarja koostuu kahdesta tai useammasta datan kopiosta ja jokainen kopio voi toimia primäärisenä tai sekundäärisenä kopiona milloin tahansa. Datan primäärikopio suorittaa kaikki kirjoitukset ja luvut oletuksena, kun taas sekundäärinen kopio ylläpitää primäärisen datan kopiota. Siinä tapauksessa että primäärikopio pettää otetaan sekundäärinen kopio käyttöön muuttamalla se primääriseksi.

MongoDB:ssä on myös ominaisuuksia kuorman tasoittamiseksi. MongoDB voi toimia myös useammalla palvelimella, tasoittamalla kuormaa palvelinten välillä tai kopioimalla datan ja pitämällä järjestelmän toimintakuntoisena mahdollisen laitteisto-ongelman sattuessa. Automaattinen konfigurointi on helppo ottaa käyttöön ja uusien koneiden lisäys toiminnassa olevaan tietokantaan onnistuu helposti.

MongoDB:tä voi käyttää myös tiedostojärjestelmänä ja käyttää hyväksi kuorman tasoitusta ja kopiointiominaisuutta tallennettaessa tiedostot useammalle koneelle. MongoDB käyttää tiedostojen tallentamisessa Grid File System –funktiota, joka pilkkoo isommat tiedostot useaksi pienemmäksi palaksi ja tallentaa ne. Useamman koneen järjestelmässä tiedostojen varmennus onnistuu helposti ja näin saadaan tasaisesti kuormaa jakava ja virheistä selviävä järjestelmä.

MongoDB on siis monipuolinen ja mukautuva tietokantajärjestelmä, joka ei rajoita tietokanta rakennetta niin paljon kuin perinteiset SQL-relaatiotietokannat. MongoDB on myös yksi suorituskykyisimmistä NoSQL tietokannoista mitä on tarjolla. United Software Associates julkaisi testin tulokset, jossa MongoDB päihitti muut suositut NoSQL tietokannat, osassa testeistä jopa selvällä erolla (United Software Associates: NoSQL Systems Benchmarking). (MongoDB Inc.: The MongoDB 3.0 Manual.)

4.5 PHP

PHP valittiin palvelinpäässä käytettäväksi ohjelmointikieleksi sen takia, että PHP:n käytöstä löytyi eniten kokemusta. Näin ollen toteuttaminen pystyttiin aloittamaan nopeasti ja tehokkaasti, ilman suurempia viivästyksiä.

PHP on palvelinpään skriptauskieli joka on suunniteltu web-kehitykseen mutta on käytetty myös yleiskäyttöisenä ohjelmointikielenä. PHP oli alun perin lyhenne sanoista Personal Home Page, mutta nykyisin sillä tarkoitetaan PHP: Hypertext Preprocessor:ia. PHP koodia on helppo sekoittaa HTML koodin joukkoon. PHP on yleensä prosessoitu palvelimelta löytyvällä tulkilla ja tästä saatu tulos lähetetään käyttäjälle. PHP:n avulla pystyy generoimaan esimerkiksi HTML koodia, tai kuten vertaismyyntimobiilisovelluksen tapauksessa JSON:ia.

PHP pystyy käytännössä mihin tahansa. Vaikka PHP onkin pääasiassa fokusoitunut palvelinpään skriptaukseen, pystyy se kaikkeen mihin mikä tahansa CGI ohjelma pystyy, kuten datan keruu, dynaamisen sivun sisällön generointiin tai lähettämään ja vastaanottamaan evästeitä. Kolme pääaluetta, joissa PHP skriptejä käytetään, ovat palvelinpään skriptaus, komentorivi skriptaus ja työpöytäsovellusten kirjoittamiseen. PHP:ta voidaan käyttää kaikissa merkittävissä käyttöjärjestelmissä kuten esimerkiksi Linux, Microsoft Windows ja Mac OS X. PHP:lla koodattaessa on siis vapaus valita käyttöjärjestelmä ja palvelin vapaasti ja voit lisäksi valita myös perinteisen koodauksen tai olio-ohjelmoinnin tai näiden sekoituksen. (PHP.net: PHP.net documentation.)

4.6 JSON

JSON valikoitui web-APIn tuottamaksi vastausformaatiksi JSON:in helppolukuisuuden takia. JSON on helppolukuista sekä ihmiselle, että tietokoneelle. Ja koska web-APIa tulee käyttämään useat erilaiset laitteet, oli JSON oiva valinta sillä sen käsittely onnistuu helposti lähes millä alustalla ja ohjelmointikielellä tahansa.

JSON eli JavaScript Object Notation on kevytrakenteinen datan siirtoformaatti. Se on ihmisille helppolukuista ja sitä on helppo kirjoittaa. JSON on koneelle helppoa parsia ja helppoa tuottaa. JSON on tekstiformaatti, joka on täysin kieliriippumaton, mutta käyttää

yleissopimuksia, jotka ovat tuttuja C-kieliperheiden koodaajille, kuten C, C++, C#, Java, JavaScript, Perl, Python ja monien muiden. Nämä ominaisuudet tekevät JSON:ista täydellisen kielen tiedonvaihtoon.

JSON on rakennettu käytännössä kahdelle rakenteelle, valikoimalle nimi-arvo-pareja ja järjestetty lista arvoja. Nimi-arvo parit ovat eri ohjelmointikielissä ymmärretty esimerkiksi objekteina, tietueina, avaimellisina listoina tai assosiatiivisina taulukkoina. Järjestetyt listat arvoina taas ovat useimmissa ohjelmointikielissä esimerkiksi taulukkoja, vektoreita tai listoja. Nämä ovat universaaleja datarakenteita. Käytännössä kaikista moderneissa ohjelmointikielissä on tuki näille muodossa tai toisessa. On siis järkevää että dataformaatti on vaihdettavissa ohjelmointikielien välillä, joista nämä rakenteet löytyvät.

JSON on formaatiltaan seuraavan kaltainen. Objekti on järjestelemätön valikoima nimi-arvo pareja. Objekti alkaa {-merkillä ja päättyy }-merkillä. Jokaisen nimen perästä löytyy kaksoispiste, jonka jälkeen löytyy arvo, joka voi olla myös objekti. Nimi-arvo parit on eroteltu pilkulla. (JSON.org: Introducing JSON.)

5 TOTEUTUS

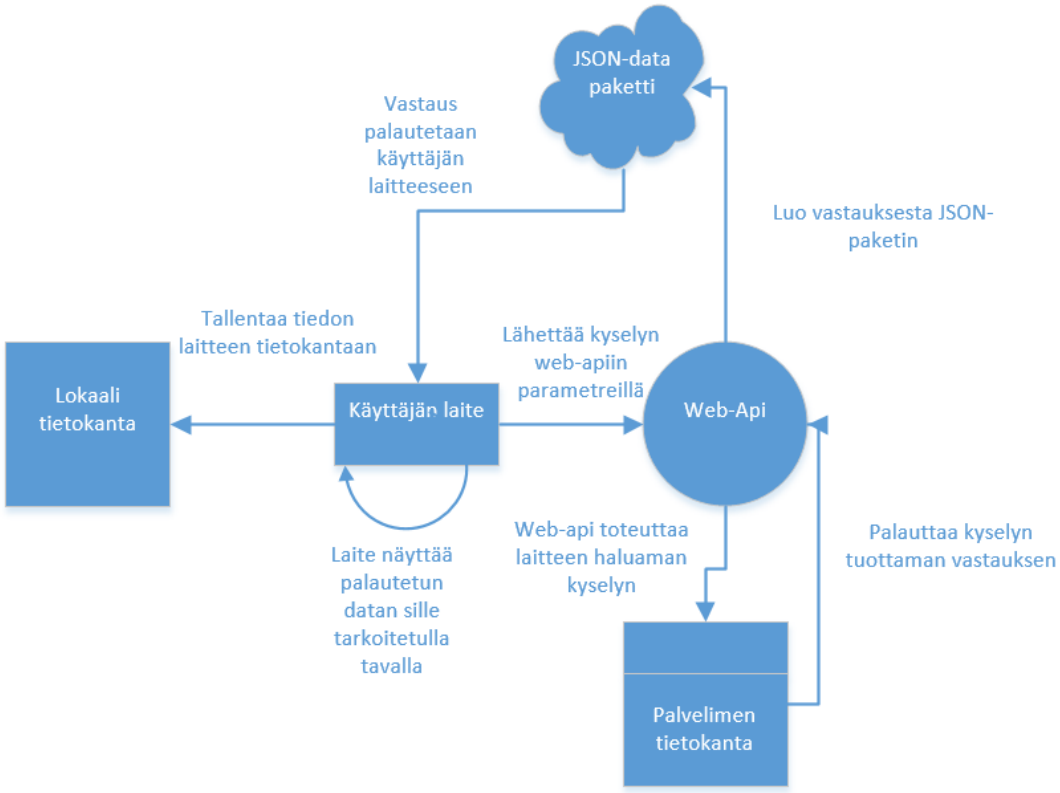
Tässä luvussa tarkastellaan palvelun suurimpien osien teknistä toteutusta, sekä pohditaan tuntemuksia itse kehitys prosessista.

5.1 Tietokannan ja laitteen välinen kommunikointi

Laitteen ja tietokannan välinen kommunikointi (kuva 26) suoritetaan web-palvelimen kautta. Tällä ratkaisulla mahdollistetaan kaikille alustoille saman tietokantarajapinnan käyttäminen. Tällä tavalla tehdessä tietokantaan ja sen rajapintaan tehdyt muutokset toimivat suoraan jokaisella alustalla, eikä vain yhdellä. Tämä tapa lisää myös palvelun tietoturvaa, sillä suoraa yhteyttä ei tietokantaan laitteesta oteta. Tällä tavalla käyttäjät eivät pysty suoraan selvittämään tietokannan IP-osoitetta.

Tietokantakyselyt ovat palvelussa tarkkaan rajattu, eivätkä käyttäjät pääse luomaan omia kyselyitään mahdollisilla haavoittuvuuksien hyväksi käytöillä. Arkaluontoisemmat haut tietokannasta on lisäksi suojattu jokaiselle käyttäjälle uniikilla rajapinta-avaimella, joka luodaan käyttäjän luonnin yhteydessä automaattisesti. Tämä avain tarkistetaan jokaisessa tietokantakyselyssä, jolla on mahdollista tehdä haittaa palvelun käyttäjille tai palvelun ylläpitäjille. Rajapinta-avaimen ja käyttäjätunnuksen täytyy olla oikeita, jotta palvelu antaa suorittaa esimerkiksi tuotteiden poistamisen. Tietokantakyselyt, joissa ei ole mahdollista haitan riskiä, kuten tavallinen tuotteiden haku, ei ole suojattu tämän rajapinta-avaimen avulla.

Alla olevasta kaaviosta (kuva 26) selviää millä tavalla laite kommunikoi tietokannan kanssa. Kaaviossa lähdetään liikkeelle käyttäjän laiteesta, joka ottaa yhteyden web-APIin ja ilmoittaa haluamansa kyselyn ja tarvittavat parametrit. Web-APIssa tarkistetaan vaatiiko kysely rajapinta-avaimen tarkistusta. Mikäli kaikki parametrit ovat kunnossa, web-API suorittaa kyselyn tietokannasta. Tietokanta palauttaa vastauksen web-APIin ja web-API muodostaa vastauksesta oikean muotoisen JSON-objektin, jonka web-API palauttaa vastauksena takaisin käyttäjän laitteeseen. Käyttäjän laitteessa palautettu data tallennetaan paikalliseen tietokantaan ja asetettaan näytille kyseiselle datalla tarkoitetulla tavalla.



KUVA 26. Palvelimen ja laitteen välinen kommunikointi.

5.2 Versionhallinta

Ohjelmistokehityksen versionhallintatyökaluksi valittiin Git, joka on ilmainen, vapaan-lähdekoodin versionhallintajärjestelmä. Git oli luonteva valinta versionhallinta työkaluksi, koska siitä oli aiempaa käyttökokemusta, jonka seurauksena versionhallinnan opetteluun ei kulunut ylimääräistä aikaa. Git on suunniteltu käsittelemään kaikki pienistä todella suuriin projekteihin nopeasti ja tehokkaasti. Git on hyvin laajalti käytetty versionhallintatyökalu ja sen osaamiselle on suurta kysyntää ohjelmistosuunnittelu alalla.

Git on helppo oppia, se käyttää vähän resursseja ja toimii nopeasti. Git peittoaa muut SCM työkalut, kuten Subversion, CVS, Perforce ja ClearCase, ominaisuuksillaan. Ominaisuuksia ovat esimerkiksi kevyt paikallinen haaroittaminen ja mahdollisuus usean henkilön samanaikaisen työskentelyn. (Git, 2015.)

Versionhallinta ohjelmistonkehityksessä toteutettiin siten, että perustettiin ilmainen yksityinen repository Bitbucket-palveluun, joka tarjoaa helppoa ja nopeaa tapaa saada oma versionhallintajärjestelmä yhdessä Gitin kanssa. Tässä kyseisessä projektissa versionhallinnointi ei ollut erityisen aktiivista. Versionhallintaan tallennettiin isoja paketteja muutoksia, pienten yksittäisten muutosten sijaan. Näin toimittaessa versionhallinta toimii lähinnä varmuuskopiona projektille. Näin toimittiin, koska ohjelmistoa kehitettiin vain yhdeltä tietokoneelta, joten välitöntä tarvetta pienten muutosten lähettämisellä pääreposito-ryyn ei nähty.

Varsinaista tarvetta versionhallinnalle ei ole tässä projektissa tähän mennessä ilmennyt, sillä ongelmia ei ilmennyt uusien versioiden kanssa, josta syystä tarvetta ei ole ollut palata käyttämään aiempaa versiota. Mikäli ilmenee ongelmia uusimpien committien kanssa työskentelyssä, Git mahdollistaa helpon paluun aiempaan versioon ja tämän version voi asettaa uusimmaksi versioksi. Tällöin kehitystyötä jatketaan siitä mihin kyseinen committi on jäänyt. Ongelmattomuudesta huolimatta versionhallinta on hyvä tapa, niin suurissa kuin pienissäkin projekteissa. Versionhallinnan käyttäminen helpottaa projektien kehittämistä missä on useita kehittäjiä mukana, koska useat käyttäjät voivat tallentaa tällöin muutoksensa samaan paikkaan. Tällöin ei erillistä aikaa vievää tiedostojen yhdistelyä tarvitse tehdä. Lisäksi koskaan ei voi olla varma millaisia ongelmia matkan aikana ilmenee.

5.3 Omat päätelmät ohjelmiston kehityksestä

Ohjelmiston front-endin ohjelmointi lähti aluksi hieman verkkaisesti liikkeelle, sillä aiempaa kokemusta kyseisistä tekniikoista ei ollut. Hetken tutustumisen ja testaamisen jälkeen projekti kuitenkin lähti yllättävän nopeasti etenemään. C# syntaksi oli huomattavan yksinkertainen oppia. Suurin ongelma kehitystyössä oli löytää ajan tasalla olevia ohjeita ja apuja. Suurin osa korjausehdotuksista vastaaviin ongelmiin olivat aiempaan versioon tarkoitettuja, eivätkä näin toimineet uusimmalla alustalla, jolle kyseistä ohjelmistoa kehitetään. Visual studio tuki hyvin C#:illa kehittämistä ja sen ennustavasta tekstinsyötöstä oli suuri apu etsiessä esimerkiksi uusista kirjastoista oikeaa funktiota. Kaiken kaikkiaan ohjelmistokehittäminen Windows Phone alustalle oli hyvin mieluista vastoinkäymisistä huolimatta.

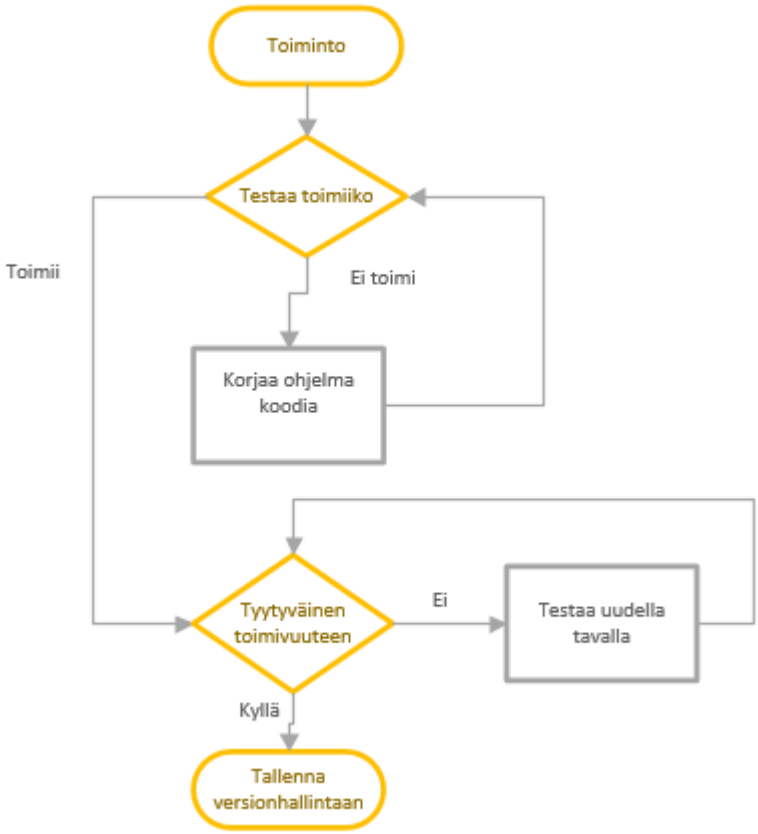
Ohjelmiston back-endin, eli käytännössä web-APIn, toteuttamisessa ensisijaisena tavoitteena oli saada nopeasti toimintaan saatavia ominaisuuksia. Tämän takia ohjelmointikielen valintaan ei kulunut käytännössä ollenkaan aikaa, vaan päädyttiin käyttämään kaikista tutuinta tehtävään soveltuvaa ohjelmointikieltä PHP:ta. Vaikkakin PHP osaaminen oli vahvaa, niin ei toteutus kuitenkaan ollut täysin ongelmatonta. Data oli varastoituna MongoDB:hen, mikä oli projektin alkuvaiheessa täysin uusi tietokantatyyppi. Lisäksi välillä pieniä ongelmia aiheutti tulostetun JSON-datan saaminen oikeaan muotoon. Kaiken kaikkiaan web-APIn toteuttaminen oli hyvin mieluinen tehtävä.

Kehityksen aikana opittiin paljon uusia asioita, jotka tukevat ammattiosaamista ja vahvistivat itseluottamusta uusien asioiden nopeasta oppimisesta. Esimerkiksi mobiilikehityksestä tulee varmasti olemaan hyötyä tulevaisuudessa sillä mobiililaitteet yleistyvät ja kehittyvät jatkuvasti, joten niille tarvitaan jatkuvasti parempia ja kehittyneempiä ohjelmistoja.

6 TESTAUS

6.1 Kehitysvaiheen testaus

Ohjelmiston kehitysvaiheessa testaaminen on jatkuvaa. Ennen jokaista päivitystä versionhallintaa uusia ominaisuuksia testataan laajasti useaan otteeseen useilla eri syötteillä ja tarkastellaan tietoja, joita puhelimesta lähtee, ja mitä puhelimeen saadaan vastauksena. Tällä tavalla pyritään mahdollisimman bugi-vapaan ohjelmakoodin tuottamiseen (kuva 27).



KUVA 27. Vuokaavio ohjelmiston kehitysvaiheessa tapahtuvasta testauksesta.

Tarkastellaan esimerkiksi tuotteen lisäämisen testausta. Testaus prosessissa kokeiltiin tuotteen lisäämistä ilman kuvia, yhdellä, kahdella sekä neljällä kuvalla joka on kuvien maksimimäärä. Näin voidaan olla hyvin varmoja siitä, että ohjelma toimii myös kolmella kuvalla, ilman että tarvitsee tätä erikseen testata. Tämä voidaan päätellä siitä, että tuotteen lisääminen toimii ilman kuvia, yhdellä kuvalla, useammalla kuin yhdellä kuvalla ja maksimi määrällä kuvia, on todennäköistä, että tuotteen lisääminen toimii kaikilla kuva määrillä kyseisten kohtien välillä, koska toiminnon toimintaperiaate ei muutu oli kyseessä kaksi tai kolme kuvaa.

Kehitysvaiheessa on ilmennyt vikoja, joita ei ole onnistuttu toistamaan debuggerin ollessa käynnissä. Tällaiset ongelmat ovat huomattavasti vaikeampia paikallistaa sekä korjata. Osa tämän tyyppisistä virheistä on edelleen prototyypissä ja niitä yritetään korjata, kun toimenpiteille löytyy tarpeeksi aikaa.

6.2 Valmiin prototyypin testaus

Valmiin prototyypin testaus on vasta suunnitteluvaiheessa. Tarkoituksena on useamman henkilön voimin useammalla laitteella testata ominaisuus kerrallaan, miten kukin ominaisuus käyttäytyy kun sitä toistetaan uudelleen ja uudelleen. Laitteita käytettävissä on tällä hetkellä vain kaksi, joten testaaminen tulee viemään aikaa, ellei lisälaitteita hankita. Testauksessa käytetään apuna myös muiden alustojen ohjelmistoja. Näitä apuna käyttäen voidaan esimerkiksi kokeilla, kuinka ohjelma käyttäytyy, kun tulee useita viestejä samanaikaisesti ja dataa liikkuu keskustelujen välillä enemmän.

7 JATKOKEHITYSSUUNNITELMAT

Ohjelmisto on jatkuva kehitysprojekti, joka ei missään vaiheessa ole täysin valmis. Tarkoituksena on kuunnella käyttäjien toiveita ja kehitysideoita. Mikäli jotkin käyttäjien ehdottamat lisätoiminnot tai ominaisuudet koetaan hyväksi ideaksi, voidaan nämä toiminnot ottaa kehityksen alle. Tässä luvussa keskitytään jo tässä vaiheessa suunnitteilla oleviin lisäominaisuuksiin.

7.1 Hakuvahti

Palveluun suunnitellaan hakuvahdin toteuttamista. Hakuvahdin toimintaperiaate on tallentaa käyttäjän tekemä haku. Hakua toistetaan palvelimella aina asetetun ajan välein ja jos hakutuloksiin tulee uusia hakuehtoja vastaavia tuotteita, käyttäjä saa tästä ilmoituksen ohjelmiston kautta. Tämän toiminnon tarkoituksena helpottaa käyttäjiä, jotka etsivät jotain tietynlaista tuotetta. Käyttäjän ei tarvitse jatkuvasti kirjautua ohjelmaan tekemään hakuja haluamillaan hakuehdoilla, vaan lisättyjen tuotteiden valvonta suoritetaan palvelimella automaattisesti. Käyttäjille on tarkoitus mahdollistaa yksi aktiivinen hakuvahti kerrallaan.

7.2 Sosiaalisenmedian tunnuksilla kirjautuminen

Ohjelmistoon on suunnitteilla mahdollistaminen sisäänkirjautuminen kaikilla yleisimmillä sosiaalisenmedian käyttäjätunnuksilla. Tämä helpottaisi käyttäjien ohjelmiston käyttöönottoa. Tällä tavalla lasketaan käyttäjien kokeilukynnystä, koska käyttäjän ei tarvitse täyttää rekisteröitymistietoja vaan kirjautuminen onnistuu yhdellä napin painalluksella. Mikäli käyttäjä ei ole kirjautuneena sosiaalisenmedian palveluun käytettävällä laitteella, ohjelma avaa valitun sosiaalisen median sisäänkirjautumisikkunan ja käyttäjä pääsee palveluun täytettyään sosiaalisenmedian käyttäjätietonsa.

7.3 Automaattinen yhteystietojen jakaminen

Tarkoituksena on mahdollistaa käyttäjälle tallentaa jaettavaksi tarkoitetut yhteystiedot, jotka voi jakaa helposti keskustelun kautta, yhden napin painalluksella. Tämän toiminnon tarkoituksena on helpottaa ja nopeuttaa puhelinnumeroiden ja sähköpostien jakamista, mikäli käyttäjä haluaa näitä yhteystietoja hyödyntää kaupan käynnissä. Näiden tietojen jakaminen ei ole tarkoitettu mitenkään pakollisiksi, eikä tallennettuja yhteystietoja jaeta ilman käyttäjän lupaa tai halua jakaa..

LÄHTEET

DB-Engines, 2015. Ranking. DB-Engines Ranking. Luettu 5.11.2015. http://db-engines.com/en/ranking

Git, 2015. Luettu 3.11.2015. https://git-scm.com/

Huuto.net, 2013. Blogit. Vertaiskauppa kasvoi uuteen ennätykseen. Luettu 10.10.2015. http://blogit.huuto.net/kehitys/2013/04/22/vertaiskauppa-kasvoi-uuteen-ennatykseen/

JSON.org, 2015. Introducing JSON. Luettu 16.10.2015. http://www.json.org

Kauppalehti, 2014. Tori.fi. Tori.fi nousi markkinajohtajaksi. Luettu 10.10.2015. http://www.kauppalehti.fi/5/i/yritykset/lehdisto/cision/tiedote.jsp?selected=kaikki&oid=20140201/13916856169530

Markkinointi&Mainonta, 2014. Huuto.net 15 vuotta. Huuto.net 15 vuotta – viime vuonna kauppoja 26 miljoonaa. Luettu 10.10.2015. http://www.marmai.fi/uutiset/huutonet+15+vuotta++viime+vuonna+kauppoja+26+miljoonaa/a2245128

Microsoft, 2015. Introduction to the C#. Introduction to the C# and the .NET Framework. Luettu 16.10.2015. https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx

Microsoft, 2015. Visual Studio. Visual Studio IDE. Luettu 16.10.2015. https://msdn.microsoft.com/fi-fi/library/dn762121.aspx

Microsoft, 2015. XAML. What is XAML? Luettu 16.10.2015. https://msdn.microsoft.com/en-us/library/cc295302.aspx

MongoDB Inc., 2015. The MongoDB 3.0 Manual. Luettu 16.10.2015. http://docs.mongodb.org

Philippe Kruchten, 1995. 4+1 view model. Architectural Blueprints – The “4+1” View Model of Software Architecture. Luettu 28.10.2015. http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf

PHP.net, 2015. PHP.net documentation. Luettu 16.10.2015. http://www.php.net

United Software Associates, 2015. NoSQL Systems Benchmarking. HIGH PERFORMANCE BENCHMARKING: MongoDB and NoSQL Systems. Luettu 16.10.2015. http://info-mongodb-com.s3.amazonaws.com/High%2BPerformance%2BBenchmark%2BWhite%2BPaper_final.pdf