

Tuomas Manninen

# Keskustelusivuston ja sitä tukevan Windows Phone -sovelluksen toteuttaminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

14.4.2013

Tekijä(t) Otsikko	Tuomas Manninen Keskustelusivuston ja sitä tukevan Windows Phone -sovelluksen toteuttaminen
Sivumäärä Aika	42 sivua 14.4.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Lehtori Simo Silander Lehtori Jorma Rätty
<p>Insinöörityön tavoitteena oli kehittää keskustelusivusto ja sitä tukeva Windows Phone -sovellus. Chat-sivuston tarkoituksena on tarjota keskustelupaikka, johon ihmiset voivat rekisteröidä oman tunnuksen ja osallistua keskusteluun erilaisissa keskusteluhuoneissa. Rekisteröityneet käyttäjät voivat liittyä keskusteluun myös Windows Phone -sovelluksen kautta.</p> <p>Työ vaati sivuston eri osien suunnittelun ja toteutuksen, tietokannan luomisen sekä mobiilisovelluksen suunnittelun ja toteutuksen. Sivusto toteutettiin käyttäen web-kieliä html, php sekä JavaScript. Windows Phone -sovellus toteutettiin Microsoftin Visual Studio -ympäristössä C#-kielellä.</p> <p>Käytännön osan lisäksi työssä selvitettiin ja listattiin erilaisia keskustelusivutyyppejä, joista otettiin vaikutteita itse toteutettavaan sivustoon ja mobiilisovellukseen.</p>	
Avainsanat	Chat, php, Windows Phone

Author(s) Title Number of Pages Date	Tuomas Manninen Development of a chat site and a supporting Windows Phone application 42 pages 14 April 2013
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Simo Silander, Senior Lecturer Jorma Rätty, Senior Lecturer
<p>The target of this Bachelor's thesis was to develop a chat site and a Windows Phone application that supports the site. The purpose of the chat site is to offer a place where people can register own account for themselves and join conversations in variety of chat rooms. Registered users can also join to these conversation by the Windows Phone application that was developed for this use.</p> <p>To complete the thesis different parts of the site and mobile application was needed to design and develop. Also database for various information needed to build for the use of both the site and the mobile application. The site was developed using web languages Html, Php and JavaScript. The Windows Phone application was developed in Microsoft's Visual Studio -programming environment using C#-language.</p> <p>In addition to the practical part of the thesis some of the most common chat types were studied and listed. This information was used for the design of the chat site and the mobile application.</p>	
Keywords	Chat, php, Windows Phone

## Sisällys

1	Johdanto	1
2	Chat yleisesti	1
2.1	Historiaa	1
2.2	Chatin eri kasvot	2
2.2.1	Huonejaoteltu chat	2
2.2.2	Shoutbox	4
2.2.3	Alapalkki-chat	6
3	Zat-sivusto	8
3.1	Esittely	8
3.2	Suunnittelu	11
3.2.1	Sivuston eri osat	11
3.2.2	Tietojen säilöminen	13
3.2.3	Yksityiskeskustelu	14
3.3	Toteutus	15
3.3.1	Käytetyt ohjelmat sekä ohjelmointikielet	15
3.3.2	Kirjautumisjärjestelmä	17
3.3.3	Huonelistaus	17
3.3.4	Chathuone	18
3.3.5	Yksityiskeskustelu	21
3.3.6	MySQL-tietokanta	22
3.4	Kehitysideoita	24
4	ZatApp-mobiilisovellus	24
4.1	Esittely	24
4.2	Suunnittelu	27
4.2.1	Sovelluksen eri osat	27
4.2.2	Tiedonsiirto	30
4.3	Toteutus	31
4.3.1	Käytetyt ohjelmat sekä ohjelmointikielet	31
4.3.2	Tiedonsiirto tietokannan ja sovelluksen välillä	32

4.3.3	Kirjautuminen	34
4.3.4	Huonelistaus	35
4.3.5	Viestien lukeminen ja kirjoittaminen	37
4.3.6	Keskustelijoiden listaaminen	39
4.4	Kehitysideoita	39
5	Yhteenveto	40
	Lähteet	42

## Lyhenteet

.NET	Microsoftin kehittämä virtuaalinen ajoympäristö ja abstraktiokerros Windows-käyttöjärjestelmän alaisuuteen.
CSS	Cascading Style Sheets. Tyyli-merkintäkieli, joka mahdollistaa WWW-sivujen ulkoasun määrittelyn.
C#	C sharp. Microsoft-yhtiön .NET-konseptia varten kehittämä ohjelmointikieli, joka kehitettiin yhdistämään C++:n tehokkuus ja Java-kielen helppokäyttöisyys.
GPL	GNU General Public License. Vapaa ohjelmistolisenssi, joka antaa oikeuden käyttää, kopioida, muuttaa ja jakaa edelleen ohjelmistoa ja sen lähdekoodia. Lisenssi edellyttää, että samat vapaudet säilyvät myös ohjelmiston muunnelluissa versioissa.
HTML	HyperText Markup Language. Avoimesti standardoitu merkintäkieli, jolla voidaan kuvata ja määritellä WWW-sivuja.
PHP	Hypertext Preprocessor. Ohjelmointikieli, jota käytetään erityisesti Web-palvelinympäristöissä dynaamisten web-sivujen luomisessa.
SSH	Secure Shell. Salattuun tietoliikenteeseen tarkoitettu protokolla, jonka yleisin käyttötapa on ottaa etäyhteys SSH-asiakasohjelmalla SSH-palvelimeen, jotta päästään käsiksi toiseen koneeseen.
XAML	Extensible Application Markup Language. Microsoftin luoma deklaraatiivinen kieli, joka perustuu vahvasti XML-kieleen. XAML:a käytetään kuvaamaan jäsenneltyjä arvoja ja olioita.

## 1 Johdanto

Insinööriyön aiheena on keskustelusivuston sekä sitä tukevan Windows Phone -sovelluksen toteuttaminen. Sivustolle rekisteröityneet käyttäjät voivat kirjautua sivustolle ja osallistua keskusteluihin eri keskusteluhuoneissa. Käyttäjä voi myös osallistua samaisiin keskusteluihin erikseen toteutetulla Windows Phone -sovelluksella. Internetissä käytävistä keskusteluista käytetään usein Suomessakin englanninkielistä nimitystä chat, jota myös tässä työssä jatkossa käytetään.

Sivusto on toteutettu HTML-, CSS-, PHP- sekä JavaScript-ohjelmointikielillä. Etenkin JavaScriptin jQuery-kirjastoa käytettiin sivuston toteutukseen ja erityisesti kirjaston Ajax-funktioista oli paljon apua. Lueteltujen kielten lisäksi sivusto vaatii MySQL-tietokannan luomisen.

Windows Phone -sovellus on toteutettu C#-kielen erikseen Windows Phone -ohjelmointiin tarkoitettujen ohjelmointikirjastojen avulla. Sovellus käyttää samaa tietokantaa kuin sivusto, joka vaatii erillisten PHP-tiedostojen kirjoittamisen luku- ja kirjoitusoperaatioita varten. Näiden tiedostojen ja sovelluksen välillä kulkeva tieto kulkee JSON-muodossa.

Työ esittelee aluksi yleisesti chat-sivustoja esitellen mm. erilaiset chat-tyypit. Toteutetusta chat-sivustosta sekä -sovelluksesta tarkastellaan erikseen suunnitteluprosessia sekä itse toteutusvaihetta. Myös käytetyt tekniikat ja työkalut esitellään. Lisäksi pohditaan mahdollisia kehitysideoita, jotka tekisivät sivustosta ja mobiilisovelluksesta entistäkin paremmat. Lopun yhteenveto kokoaa koko prosessin kasaan ja siinä käydään läpi muun muassa mahdolliset ylä- ja alamäet, jotka kohdattiin insinööriyön aikana.

## 2 Chat yleisesti

### 2.1 Historiaa

Kaikkia chatteja yhdistää niiden perustoiminallisuus: viestikenttä, johon käyttäjä kirjoittaa halutun viestinsä, sekä viestilistaus, jossa viestit listataan aikajärjestykseen.

Ensimmäinen chat-järjestelmä oli vuonna 1974 Doug Brownin ja David R. Woolleyn luoma PLATO-ympäristössä toimiva Talkomatic [1]. Se tarjosi useita kanavia, joihin pystyi liittymään enimmillään viisi keskustelijaa. Heidän kirjoittamansa viestit ilmestyivät kaikkien keskustelijoiden luettavaksi kirjain kirjaimelta. Internetin edeltäjässä ARPANETissä oli myös samoihin aikoihin chat nimeltään PLANET [2]. Suurempaan suosioon chatit tulivat suomalaisen Jarkko Oikarisen vuonna 1988 kehittämän IRCin kautta [3].

## 2.2 Chatin eri kasvot

Kun selailee erilaisia sivustoja, niin huomaa nopeasti, että hyvin monesta löytyy chat jossakin muodossa. Sivustolla voi olla oma chat-osio erilaisine nk. huoneineen. Huone on nimitys sivulle, jossa tietty ryhmä keskustelijoita kirjoittaa viestejä toisilleen reaaliajassa [4]. Sivu on siis eräänlainen huone, jolla on oma tarkoitusperänsä esimerkiksi aiheen kautta ja jonne keskustelijat voivat "astua sisään" klikkaamalla huonetta vastaavaa linkkiä erikseen linkeille varatusta listauksesta. Chatin ei kuitenkaan tarvitse olla suuri rakennelma erilaisia huoneita, vaan se voi olla vain pieni viestilaatikko sivun laidassa. Sivusto voi myös rakentua kokonaan chatin ympärille jättäen muut osa-alueet pieniksi tai jopa olemattomiksi. Chat-tyypin valintaan näyttääkin vaikuttavan yhtenä asiana sen tuleva tarkoitus: kuinka moni chatissa keskustelee samanaikaisesti, onko heillä rekisteröidyt nimet tai nimimerkit ja halutaanko keskustelun aiheita mahdollisesti rajata. Seuraavissa alaluvuissa käydään läpi näitä erilaisia chat-tyyppejä esimerkkisivustoihin.

### 2.2.1 Huonejaoteltu chat

Kun halutaan kattava chat-osio laajalle käyttäjäkunnalle, on hyvä jaotella keskustelijat eri huoneisiin aihepiirien tai jonkin muun kriteerin perusteella. Näin voidaan tietenkin vain ohjalla käyttäjiä keskustelemaan esimerkiksi urheilusta elokuvien sijaan. Keskustelijat itse muodostavat keskustelun aiheen joko tarkoituksella tai sattumalta. Tavoitteena voikin olla kerätä tietyn kriteerin omaavat ihmiset yhteen, jolloin syntyy helposti luontevaa keskustelua. Mainitut kriteerit voivat olla kiinnostuksen aiheita kuten jokin harrastus, paikkakunta tai esimerkiksi tietty ikähaarukka. Ihmisten ei oleteta välttämättä puhuvan näistä asioista chat-huoneessa, mutta heillä on kuitenkin jokin yhdistävä tekijä alkuun pääsemiseksi. [5.]



Huonejaoteltu chat voidaan rajoittaa vain rekisteröityneille käyttäjille. Tunnuksellaan käyttäjä voi jakaa nimimerkinsä lisäksi muitakin tietojaan ja mahdollisesti kuviansakin. Rekisteröityminen vie kuitenkin aikaa, mikä on sekä hyvä että huono asia. Näin karsitaan pahimmat satunnaiset häiriköt, jotka voisivat helposti häiritä muita asiattomilla puheillaan ilman rekisteröityä nimimerkkiä. Käyttäjien valvonta ja mahdollinen rankaiseminen on myös helpompaa rekisteröityjen tunnusten parissa. Moni potentiaalinen kävijä voi kuitenkin kääntyä chat-huoneen "ovella" pois, koska ei halua kokea rekisteröitymisestä koituvaa vaivaa. Chatin ylläpitäjän onkin päätettävä, haluaako hän paljon keskustelijoita, joiden joukossa voi olla nimettömiä häiriköitä vai vähemmän, mahdollisesti asiallisemmin käyttäytyviä, joihin on helpompi pitää kontrollia ja jotka voivat jakaa tietojaan. Sivustolle voidaan tietenkin sallia sekä rekisteröityneet että rekisteröitymättömät, nk. anonyymit keskustelijat. Tässä tapauksessa rekisteröityneille voidaan antaa tiettyjä etuja ja lisäominaisuuksia. Tietenkin rekisteröityneet käyttäjätkin voidaan jakaa keskenään eri asemiin esimerkiksi kuukausimaksujen perusteella. Yksi tapa on myös jättää rekisteröityminen kokonaan pois, jolloin kaikki keskustelijat ovat huoneissa automaattisesti luodulla tai sillä kerralla syöttämällään nimimerkillä.

The screenshot shows the Suomi24 website's chat interface. At the top, there is a navigation bar with the Suomi24 logo and various menu items: ETUSIVU, KESKUSTELU, UUTISET, YHTEISÖ, TREFFIT, TORI, and CHAT. Below the navigation bar, there is a sidebar on the left with a list of chat categories: RYHMÄT, AJANVIETE, IKÄRYHMÄT, INTERNATIONAL CHATS, KAUPUNGIT & ALUEET, LESBO, GAY & BI, SEKSI, SINKUT JA SUHTEET, USKONTO, and YHTEISÖT. The main content area is titled "Suomi24 Chat" and displays a list of chat rooms. The first room is "Hautausmaa (1)" with a description: "Kalman Kehdossa levottomat sielut löytävät viimeisen leposijansa ... kunnes täydenkuun aikoihin on aika lähteä saalistamaan. Hautausmaan kotisivu". The second room is "Kahvila (2)" with a description: "Jutustelua mukavassa kahvilassa." Below the list, it indicates "4 chattaajaa alueella".

Kuva 1. Suomi24 Chatin huonelistaus [6.]

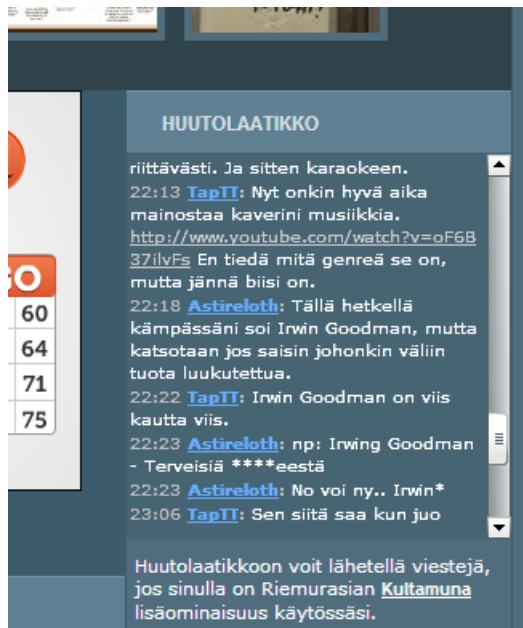
Tullessaan huonejaoteltuun chatiin käyttäjä ohjataan yleensä ensimmäiseksi huonelistaukseen (mahdollisen sisäänkirjautumisen jälkeen). Tässä kohtaa eri chat-

palvelut eroavat eniten toisistaan. Huoneet voidaan lajitella lukemattomiin kategorioihin ja niiden alakategorioihin. Esimerkiksi kuvassa 1 näkyvässä Suomi24:n chatissa on yhdeksän ryhmäksi kutsuttua huonekategoriaa, joissa on tietty määrä erikseen nimettyjä huoneita. Esimerkiksi Ajanviete-ryhmässä on kolme eri huonetta. City.fi-sivuston chat on taas jaoteltu vain kahteen ryhmään [7]: kaksi huonetta sisältävään Yleiset huoneet-ryhmään sekä Kaupunkitreffit-ryhmään, jossa on neljä eri paikkakunnan mukaan nimettyä huonetta. Huoneita ei välttämättä aina jaotella kuitenkaan eri kategorioihin, vaan ne listataan allekkain aakkos- tai muuhun järjestykseen. Tällöin nimen lisäksi huoneesta on kerrottu ainakin sisällä olevien keskustelijoiden määrä ja ehkä jokin lyhyt kuvaus huoneesta.

Keskustelijoiden määrä onkin yksi tärkeimmistä käyttäjälle näytettävistä tiedoista, oli huonejaottelu minkälainen hyvänsä. Kukaan ei halua keskustella yksikseen, joten yleensä keskustelijat kerääntyvät sinne, missä on eniten ihmisiä paikalla. Tämän takia huonemäärän tulisi olla suhteessa keskustelijoiden määrään, koska keskustelijat harvemmin jakaantuvat tasan huoneiden kesken. On siis turhaa ylläpitää kymmeniä huoneita, jos keskustelijoita on kerrallaan vain kourallinen parissa eri huoneessa.

### 2.2.2 Shoutbox

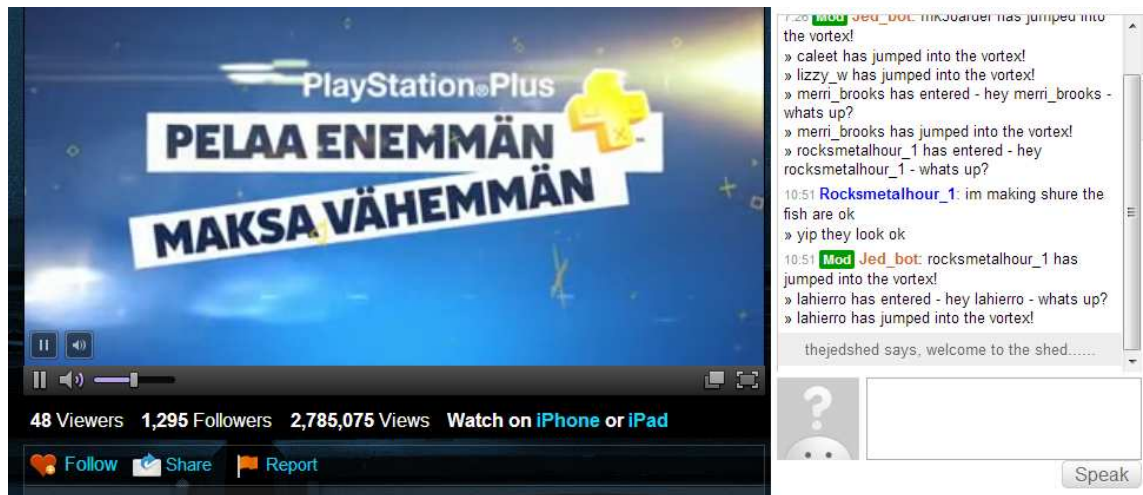
Shoutbox, tai suomalaisittain huutolaatikko, on tarkoitettu yleensä vain lisäominaisuudeksi etusivulle tai muuhun melko keskeiselle paikalle. Sen tarkoituksena ei ole ylläpitää aktiivista keskustelua; nimensä mukaisesti laatikkoon kirjoitetaan vain satunnaisia asioita, joita käyttäjälle juolahtaa mieleen. [8.]



Kuva 2. Riemurasian huutolaatikko [9.]

Kooltaan huutolaatikko on pieni ja se näyttää vain muutaman viestin kerrallaan, joista näytetään perinteisesti lähetysaika, lähettäjä sekä tietenkin itse viesti. Tällainen laatikko on esimerkiksi Riemurasia-sivuston etusivulla (kuva 2). Huutolaatikkoon näkee käytettävän sekä nimimerkkien kanssa että ilman niitä. Jos huutolaatikkoon voi kirjoittaa ilman rekisteröityä nimimerkkiä, laatikkoon on varattu viestikentän lisäksi ylimääräinen tekstikenttä, johon käyttäjä voi antaa nimensä. Joissakin huutolaatikoissa käyttäjälle luodaan automaattisesti generoitu nimimerkki sen hetkisen istunnon ajaksi. Tällainen nimimerkki koostuu yleensä sanasta kuten anonymous, guest tai user sekä numerosta tämän perässä. Tietenkin suomalaisilla sivustoilla voidaan käyttää suomenkielisiä vastineita.

Jos huutolaatikkoon voi kirjoitella ilman sisäänkirjautumista, käytössä on monesti ns. kuvavarmennus, kansainvälisemmin CAPTCHA [10]. Siinä käyttäjälle näytetään pieni kuva, jossa on satunnaisia kirjaimia tai numeroita, jotka hänen on tunnistettava ja kirjoitettava tekstikenttään viestin kirjoituksen yhteydessä. Ideana on luoda kyseisestä kuvasta sen verran epäselvä, että ihmissilmä pystyy erottamaan kirjaimet tai numerot toisin kuin jokin roskapostiohjelma. Näin saadaan ihmisten välistä keskustelua ilman, että tietokoneohjelma roskaa huutolaatikon omilla viesteillään.



Kuva 3. Huutolaatikko Justin.tv -videosivustolla [11.]

Huutolaatikon näköistä chat-laatikkoa näkee käytettävän myös esimerkiksi nk. livestream - videoiden yhteydessä (kuva 3). Sivun siinä näyttää suoraa videokuvaa, jonka vieressä on pieni huutolaatikon näköinen laatikko, johon käyttäjät voivat "huudella" kommenttejaan katsoessaan videota. Katsojien määrän ollessa suuri viestejäkin tulee helposti niin paljon, että laatikon viestivirta näyttää enemmänkin perinteiseltä chat-keskustelulta. Tässäkin tapauksessa voidaan silti puhua huutolaatikosta, koska pääpaino on videon katselussa, ei aktiivisessa keskustelussa.

### 2.2.3 Alapalkki-chat

Facebookin vuonna 2008 tunnetuksi tekemä nk. alapalkki-chat (eng. footer chat) sijaitsee nimensä mukaisesti sivustolle luodussa alapalkissa [12]. Palkki on toteutettu aina paikallaan olevaksi, eli vaikka sivua vieritetään ylös tai alas, niin alapalkki sijaitsee silti kiinteästi sivun alalaidassa. Tämän tyylisiä chatteja näkee jo muillakin sivuilla, kuten Microsoftin Hotmail-sähköpostipalvelussa (kuva 4). Alun perin ko. chat oli vain kahden kaverin väliseen keskusteluun, mutta nykyään samaan keskusteluun voi liittää useita kavereita. Näitä keskusteluita voi olla käynnissä useampi samanaikaisesti, ja se onkin yksi tämän chat-tyyppin eduista.



**Kuva 4. Alapalkki-chat Hotmail- (vas.) ja Facebook-sivustoilla [13.]**

Facebookin keskusteluikkuna on muuttunut vuosien varrella, mutta pääominaisuudet ovat pysyneet melko samanlaisina[14]. Nykyään ikkunan avoin näkymä koostuu kuvassa 4 näkyvällä tavalla viestikentästä sekä viestilistauksesta, jossa on jokaisen viestin kohdalla kirjoittajan profiilikuva, itse viesti sekä kirjoitusaika, joka näkyy vain jos käyttäjä vie hiiren kursorin viestin päälle. Myös hymiöille on varattu oma kohta ikkunan alakulmasta.

Keskusteluikkuna voidaan pienentää pieneksi palkiksi, jolloin siitä näkyy vain kaverin tai kavereiden nimi, joiden kanssa keskustellaan. Ikkuna voidaan myös sulkea kokonaan x-painikkeesta, jolloin se häviää alapalkista. Sivustolla on erikseen kaverilista, josta kaverin nimeä klikkaamalla alapalkkiin ilmestyy kyseisen kaverin ja käyttäjän välinen keskusteluikkuna. Kun käyttäjä lähettää kaverilleen viestin, tämän alapalkkiin ilmestyy myös ko. keskusteluikkuna, ellei se ole jo valmiiksi auki. Viestejä voi lähettää myös silloin, kun kaveri ei ole kirjautuneena. Tällöin viesti ilmestyy hänelle keskusteluikkunaan seuraavan sisäänkirjautumisen yhteydessä.

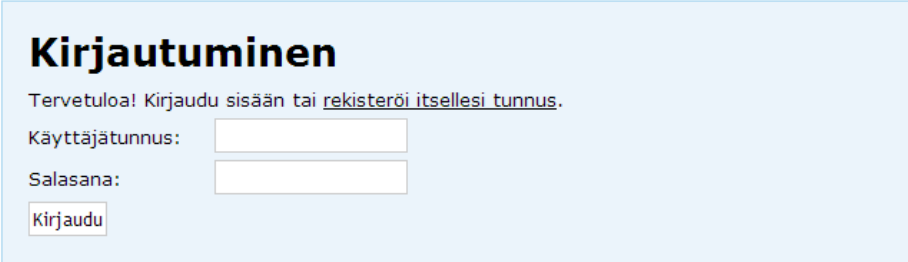
Em. toiminnallisuudet koskevat siis Facebookin chat-palkkia, mutta samantyylistä tekniikkaa näkee käytettävän muillakin sivuilla. Sen etuina on mahdollisuus käydä useampaa keskustelua samanaikaisesti sekä kevyesti ja yleensäkin sivua selatessa se

mahdollistaa keskustelun jättämisen taustalle ilman, että keskusteluikkunaa tarvitsee sulkea ja avata uudelleen. Alapalkki-chat sopii siis kahden tai korkeintaan muutaman ihmisen väliseen keskusteluun, mutta jos sille haluttaisiin suurempi rooli, niin sellaisenaan se ei välttämättä toimisi. Tietenkin verrattuna huonejaoteltuun chattiin, jokaisella chat-huoneella voisi olla oma keskusteluikkunansa alapalkissa ja näin ollen keskustelija voisi olla läsnä monessa chat-huoneessa samanaikaisesti. Tosin jos kaikki keskustelijat voisivat tehdä näin, niin luultavasti ennen pitkään he kaikki olisivat kaikissa huoneissa ja näin ollen huonejaotteluun ei olisi tarvetta ollenkaan. Mutta nykyisellään alapalkki-chat toimii hyvin sivuston kuin sivuston lisätoiminnallisuutena eikä ole mahdoton ajatus, että sille voitaisiin antaa jollakin sivustolla suurempikin rooli, kunhan sitä kehitetään oikeaan suuntaan.

### 3 Zat-sivusto

#### 3.1 Esittely

Insinöörityön käytännön osan toinen puoli koostuu chat-sivuston toteutuksesta, kun taas toinen puolisko sisältää sivustoa tukevan mobiilisovelluksen toteuttamisen. Chat-sivusto sai työnimekseen ytimekkään ja kuvaavan nimen, Zat. Tavoitteena oli luoda perustoiminnallisuuksiltaan yksinkertainen ja toimiva, mutta kuitenkin jotain uutta tarjoava keskustelupaikka.



**Kirjautuminen**

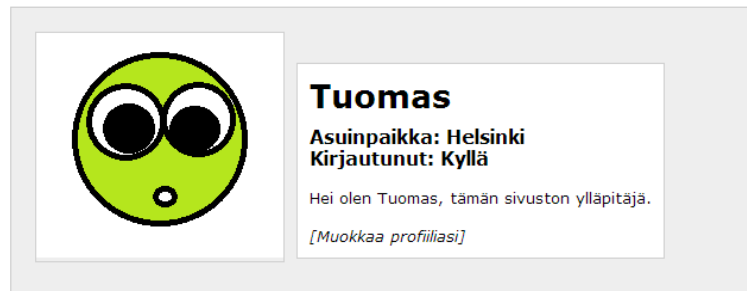
Tervetuloa! Kirjaudu sisään tai [rekisteröi itsellesi tunnus](#).

Käyttäjätunnus:

Salasana:

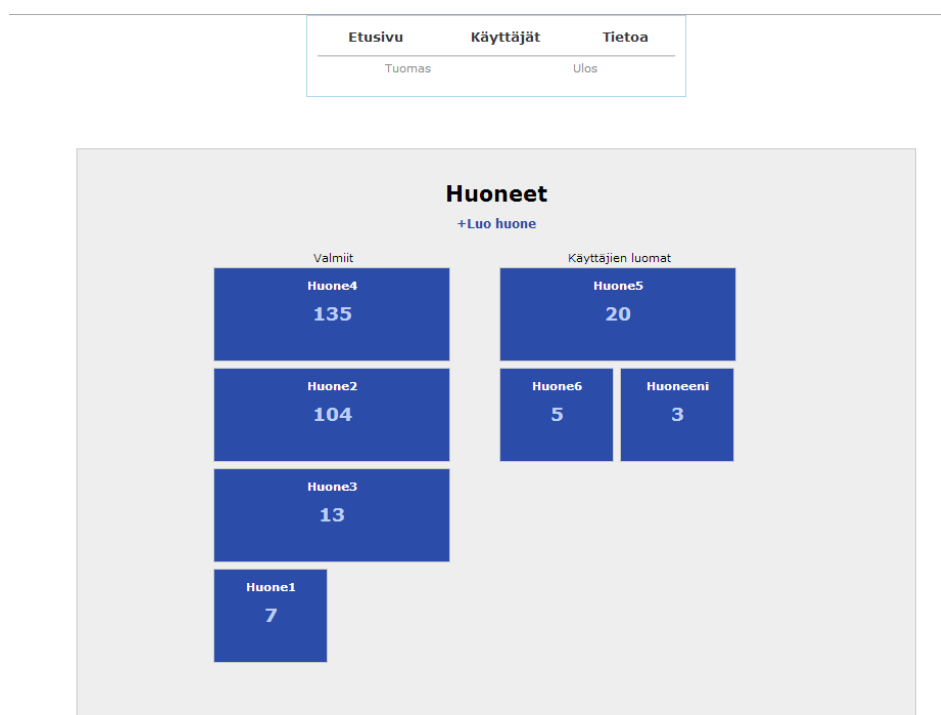
**Kuva 5. Zat-sivuston sisäänkirjautumislomake**

Jotta käyttäjä pääsee käyttämään Zatia, hänellä tulee olla rekisteröity tunnus. Tämän tunnuksen luominen onnistuu rekisteröintilomakkeella, johon on linkki kirjautumissivulla (kuva 5). Antamalla kirjautumissivulla rekisteröimänsä tunnuksen ja salasanan käyttäjä pääsee sisälle sivustolle.



**Kuva 6. Käyttäjän profiilisivu**

Jokaisella rekisteröityneellä käyttäjällä on oma profiilisivu (kuva 6), jolla listataan hänen antamansa tiedot, kuten asuinpaikkakunta sekä profiilikuva, jonka käyttäjä on mahdollisesti ladannut palvelimelle rekisteröitymisen yhteydessä.

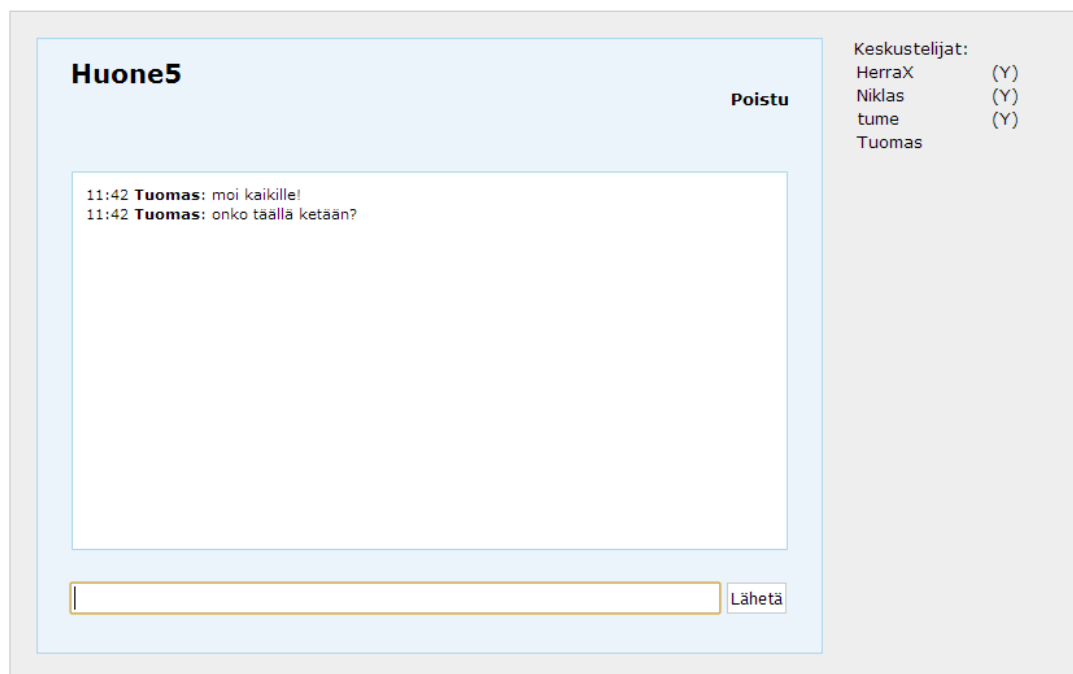


**Kuva 7. Etusivun huonelistaus**

Zatin etusivu koostuu kokonaan kuvassa 7 näkyvästä huonelistauksesta, jossa chat-huoneet on jaoteltu kahteen kategoriaan: valmiisiin huoneisiin, jotka sivuston ylläpitäjä on valmiiksi luonut, ja käyttäjien itse luomiin huoneisiin. Listausta koostuu

suorakulmioista, joista jokainen edustaa yhtä chat-huonetta. Suorakulmiossa kerrotaan huoneen nimi sekä siellä sisällä olevien keskustelijoiden sen hetkinen lukumäärä. Tämä lukumäärä vaikuttaa suorakulmion kokoon eli mitä enemmän keskustelijoita huoneessa on, sitä suurempi suorakulmiosta tulee ja näin se saa sitä enemmän näkyvyyttä. Listausta päivittyy säännöllisin väliajoin automaattisesti eli suorakulmioiden koko elää keskustelijoiden liittyessä ja poistuessa huoneista. Huonelistauksessa on myös linkki, jonka kautta käyttäjä pääsee luomaan oman huoneen.

Etusivu	Käyttäjät	Tietoa
Tuomas		Ulos



**Kuva 8. Chat-huone**

Jokaista suorakulmiota klikkaamalla pääsee dynaamisesti luotuun chat-huoneeseen (kuva 8). Sen näkymä kertoo huoneen nimen, sisällä olevat keskustelijat sekä kirjoitetut viestit. Klikkaamalla keskustelijan nimen vieressä olevaa ikonia, käyttäjä voi tarkastella keskustelijan profiilisivua tai avata yksityiskeskusteluikkunan.





**Kuva 9. Avattu yksityiskeskustelupalkki**

Tämä yksityiskeskusteluikkuna (kuva 9) avautuu näkymättömään alapalkkiin, jossa käyttäjä voi käydä samanaikaisesti yksityiskeskustelua sekä itse huoneen keskeistä keskustelua. Yksityiskeskusteluita voi olla käynnissä enimmillään 4-6 samanaikaisesti riippuen käyttäjän ruudunkoosta.

### 3.2 Suunnittelu

Zatin suunnittelu lähti liikkeelle ulkoisten asioiden kautta: mitä näkyviä osia sivustossa on ja mitä kautta käyttäjä liikkuu näiden välillä. Zatista haluttiin alusta alkaen sisällöltään ja värimaailmaltaan yksinkertainen ja ehkä jopa tietyllä tavalla tutun oloinen. Käyttäjän ei tarvitsisi klikkailla usean polun kautta päästäkseen keskusteluun, vaan tie sinne tarjottaisiin heti sivulle kirjautuessa.

#### 3.2.1 Sivuston eri osat

Zat suunniteltiin alusta alkaen sivustoksi, joka vaatisi rekisteröitymisen ennen kuin käyttäjä pääsisi millään tavalla itse keskusteluun ja muihin tietoihin käsiksi. Vaadittiin siis rekisteröintiosio, jossa käyttäjältä kysytään tarvittavat tiedot, jotka säilötään talteen. Syötteet tarkistetaan ja esimerkiksi toisella käyttäjällä olevaa nimimerkkiä ei voi rekisteröidä, vaan jokaisen nimimerkin täytyy olla uniikki.

Kirjautumisosioon siirrytään aina, jos käyttäjä yrittää johonkin sivuston osaan ilman kirjautumista. Tällöin käyttäjältä kysytään nimimerkkiä sekä salasanaa, jotka tarkistetaan syöttämisen jälkeen ja jonka perusteella joko sallitaan tai estetään sisäänkäynti.

Sivusto tarvitsi myös oman osionsa huonelistaukselle, johon oli tarkoitus keksiä jotakin erilaista verrattuna yleisimpiin huonelistauksiin. Kahtena kriteerinä käytettiin selkeää huoneen koon eli keskustelijoiden määrän esittämistä, sekä käyttäjien mahdollisuutta luoda omia huoneita. Näiden kriteerien perusteella huoneet päätettiin jakaa kahteen sarakkeeseen sen mukaan, onko huoneen luonut ylläpito vai yksittäinen käyttäjä.

Huoneet suunniteltiin myös kuvattavan listauksessa niin, että niiden edustamien kuvakkeiden koko määräytyy ja muuttuu portaittain sisällä olevien keskustelijoiden määrän mukaan. Yleensä huoneet näkee listattavan jonkinlaiseen tekstimuotoiseen taulukkoon allekkain, joten tässä haettiin graafisempaa näkökulmaa listaukseen. Yhtenä suurimpana tavoitteena oli tuoda suosituimmat huoneet selkeästi esille, jotta käyttäjät näkevät heti missä keskustelu on kuumimmillaan. Toisaalta oman huoneen luoja rohkaistaan näin keräämään huoneeseensa paljon keskustelijoita, jotta huone pääsisi listan kärkipäähän ja saisi enemmän ns. palstatilaa.

Sivuston kaiketi tärkein osa on chat-osio, jossa kaikki keskustelu käydään. Se suunniteltiin näyttämään ulkoisesti perinteiseltä sisältäen suuren viesti-ikkunan, viestin kirjoittamiseen varatun tekstikentän sekä listauksen huoneessa olevista keskustelijoista. Tässä kohtaa ei siis haluttu tehdä suuria innovatiivisia muutoksia, koska perinteinen näkymä on todettu monessa yhteydessä toimivaksi tavaksi toteuttaa tämänkaltainen chat-palvelu. Yksityiskeskustelun osalta haluttiin kuitenkin tuoda jotain erilaista tähän perinteiseen ja nähtyyn näkymään, josta enemmän kohdassa 3.2.3 Yksityiskeskustelu.

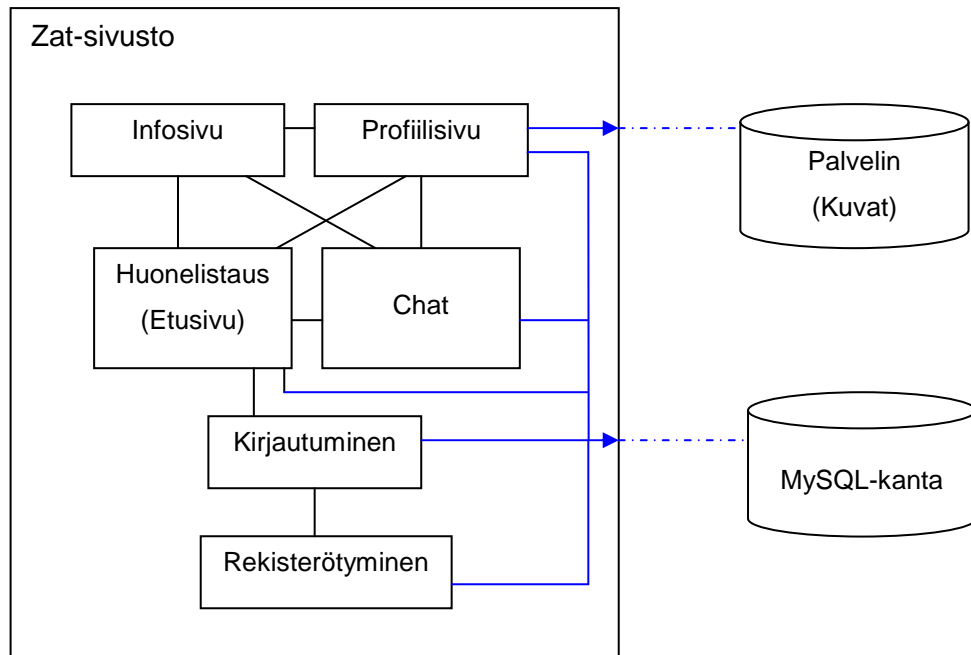
Koska sivusto vaatii rekisteröinnin ja näin ollen käyttäjätunnukset, oli loogista suunnitella myös profiilisivu-osa, jossa käyttäjä näkee oman profiilisivunsa tietoineen. Toki käyttäjät näkevät toistensa profiilisivut oman sivunsa lisäksi. Sivulle ei suunniteltu suurta roolia, joten se sisältää vain muutaman käyttäjän mahdollisesti antaman perustiedon.

Lueteltujen osioiden lisäksi sivustolle suunniteltiin kaksi pientä, informatiivista osiota, jotka ovat käyttäjälistaus sekä ns. infosivu. Käyttäjälistauksessa listataan nimensä mukaisesti kaikki sivuston käyttäjät ja siitä käy myös ilmi, ketkä ovat kirjautuneina sillä hetkellä. Infosivu pitää sisällään perustietoa Zatista ja lyhyen käyttöohjeen sivuston eri osista.

### 3.2.2 Tietojen säilöminen

Zat vaatii monenlaisen tiedon säilömistä. Käyttäjätilien tiedot oli saatava talteen, kuten myös keskusteluhuoneiden nimet ja muut tärkeät tiedot. Suurin yksittäinen määrä tietoa tulisi kuitenkin keskusteluviestien myötä, joilla on pelkän viestin lisäksi useita muita yksilöllisiä tietojaan. Nämä kaikki tiedot oli saatava talteen ja säilömiseen oli kolme vartenotettavaa vaihtoehtoa: MySQL-tietokanta, teksti/html-tiedosto sekä näiden yhdistelmä eli osa tiedoista tallennettaisiin tietokantaan ja osa tiedostoihin.

Tiedostot sopivat hyvin viestien talletukseen, koska ne voidaan tallettaa html-elementtien kera suoraan siinä muodossa, missä ne tullaan esittämään sivulla viestilistauksessa. Toisaalta viestien moderointi eli poistaminen ym. muokkaaminen on huomattavasti helpompaa MySQL-tietokannasta. Tietokanta mahdollistaa myös helpomman tavan toteuttaa halutun viestimäärän säilöminen. Eli jos halutaan säilöä esimerkiksi sata uusinta viestiä, niin uusien viestien tullessa on helppo poistaa vanhoja pois tietokannasta. Myös käyttäjätiedot tallettavat luontevimmin MySQL-tietokantaan, jossa niitä on helppo hallita turvallisesti. Näin päädyttiin yhden talletusmuodon käyttämiseen, koska myös huoneiden omat tiedot oli helpoin tallettaa tietokantaan.



**Kuva 10. Sivuston rakennekaavio**

Lopulta siis kaikki sivuston ns. muuttuva data suunniteltiin siten, että ne sijoitetaan MySQL-tietokantaan käyttäjien profiilikuvia lukuunottamatta, jotka sijoitettaisiin palvelimelle omaan kansioon php-tiedostojen oheen. Nämä kaksi säilöntäsjaintia näkyvät kuvassa 10, josta käy ilmi myös sivuston osa-alueet ja niiden suhteet. Mustat viivat osien välillä kuvaavat navigaation keinoja osien suhteita eli viivaa pitkin käyttäjä siirtyy osasta toiseen. Siniset viivat kuvaavat suhdetta säilöttyihin tietoihin eli kaikki osat infosivua lukuunottamatta käyttävät MySQL-tietokantaa, sekä profiilisivu käyttää yksinään kuvia palvelimelta.

### 3.2.3 Yksityiskeskustelu

Jotta kaksi käyttäjää voisivat keskustella keskenään ilman, että muut näkevät heidän kirjoittamansa viestit, oli sivulle suunniteltava yksityiskeskusteluominaisuus. Tapoja tämän toteuttamiseen on useita ja loppujen lopuksi kolme vartenotettavaa vaihtoehtoa erottui muista.

Yksityisviestit voitaisiin sijoittaa muun huonekeskustelun sekaan, mutta tietenkin niin, etteivät ulkopuoliset keskustelijat näkisi niitä. Yksityisviestien väri voisi olla poikkeava normaalien viestien väristä, jolloin nämä erottuisivat toisistaan paremmin. Tällainen käytäntö on käytössä esimerkiksi suomi24.fi:n chatissa yhtenä vaihtoehtona. Tämä

tapa on kuitenkin hieman liian yksioikoinen ja kömpelö, koska viestien kirjoitukseen käytetään samaa tekstikenttää, jolloin käyttäjän pitää klikkailla edes takaisin yleisen ja yksityiskeskustelun välillä. Puhumattakaan tilanteesta, jossa käyttäjä on yleisen huonekeskustelun lisäksi yksityiskeskustelussa useamman käyttäjän kanssa. Täytyisi keksiä jokin toinen tapa.

Yksi vaihtoehto yksityisviesteille oli käytäntö, jossa jokaiselle yksityiskeskustelulle avattaisiin oma chat-ikkuna uuteen selainikkunaan tai -välilehteen. Tällöin huonekeskustelu pysyisi yhdessä ikkunassa ja yksityiskeskustelu, tai useampi sellainen, omassa ikkunassaan. Tässäkin tulee kuitenkin eteen turhaa klikkailua edes takaisin ja useamman keskustelun seuraaminen on kaiken lisäksi hankalaa tällä käytännöllä eli oli löydyttävä vielä jokin erilainen tapa yksityiskeskusteluun.

Ratkaisuksi valittiin luvussa 2.2.3 esitelty, Facebookin tunnetuksi tekemä, alapalkki-chat. Tässä mallissa jokaiselle yksityiskeskustelulle luotaisiin oma chatpalkki sivun alalaitaan, johon käyttäjä voi kirjoittaa ja vastaanottaa viestejä samalla, kun hän seuraa huoneen yhteistä keskustelua. Alapalkki-chat ei ollut helpoin tai yksioikoinen tapa toteuttaa yksityiskeskustelu, mutta siitä huolimatta ja osittain siitä syystä tähän malliin päädyttiin.

### 3.3 Toteutus

#### 3.3.1 Käytetyt ohjelmat sekä ohjelmointikielät

Sivustoa suunniteltaessa ja toteuttaessa pyrittiin käyttämään kevyitä, avoimen lähdekoodin ohjelmistoja, jotka tukevat hyvin tarvittavia ohjelmointikieliä ja -tekniikoita. Avoimen lähdekoodin ohjelmistot on julkaistu avoimen lähdekoodin lisenssillä. Tämän lisenssin käyttöehdoissa sallitaan ohjelmiston vapaa jaettavuus. Itse ohjelmiston lisäksi jaetaan myös lähdekoodi, mikä mahdollistaa ohjelman vapaan jatkokehityksen. [15.]

Ohjelmointityöhön käytettiin Notepad++ -teksti- ja -lähdekoodieditoria, joka on GPL-lisenssoitu, avoimen lähdekoodin ohjelma [16]. Notepad++ tukee monia eri ohjelmointikieliä, joihin lukeutuu työn toteuttamiseen valitut, myöhemmin esiteltävät, HTML-, CSS-, JavaScript- sekä PHP-ohjelmointikielät.

Tiedonsiirto palvelimen sekä työpöydän välillä tapahtui WinSCP-ohjelman avulla. WinSCP on FTP-asiakasohjelma, joka käyttää SSH:tä tiedostojen siirrossa. WinSCP on Notepad++:n tapaan GPL-lisensioitu. [17.]

MySQL-tietokannan luontiin ja ylläpitoon käytettiin avoimen lähdekoodin phpMyAdmin-ohjelmistoa, joka on selaimen kautta käytettävä MySQL-tietokannan hallintatyökalu [18.]. Nimensä mukaisesti ohjelmisto on kirjoitettu PHP:llä. Työhön käytetty Metropolian SQL-tietokanta sisälsi valmiiksi phpMyAdmin-hallintapaneelin, joten oli loogista valita se työkaluksi tietokannan työstämiseen.

Sivuston rakenteeseen ja ulkoasuun käytettiin HTML- ja CSS-merkintäkieliä. HTML on WWW-sivujen määrittelyyn tarkoitettu merkintäkieli. Kielestä löytyvillä elementeillä rakennetaan WWW-sivuja, joita internetiselaimet tulkitsevat ja muodostavat käyttäjälle näkymän sivusta. [19.] CSS-tyylimerkintäkielellä voidaan määritellä muun muassa WWW-sivujen ulkoasu. CSS-määrytykset sijaitsevat yleensä erillisessä tiedostossa, jolloin WWW-sivun ulkoasua voidaan muuttaa muokkaamalla vain CSS-tiedostoa eli itse sivun sisältöön tai rakenteeseen ei tarvitse koskea. [20.]

Jotta Zatista saatiin dynaaminen kaikkine muuttuvine tietoineen, käytettiin HTML:n ja CSS:n lisäksi PHP:tä. PHP-ohjelmointikielen komentoja voidaan upottaa suoraan HTML-sivujen sisään, mutta ne eivät kuitenkaan näy tarkasteltaessa selaimella sivun lähdekoodia. Tämä johtuu siitä, että PHP-koodi on tulkittava kieli, joka suoritetaan aina palvelimella ennen kuin sivu päättyy selaimen. [21.]

Muutamiin yksityiskohtiin, kuten huonelistaukseen ja yksityiskeskusteluun, käytettiin JavaScriptiä sekä sillä toteutettua jQuery-ohjelmistokirjastoa. JavaScript on ohjelmointikieli selainskriptien luomiseen. Selain suorittaa nämä skriptit paikallisesti (vrt. palvelimella suoritettava PHP) , mikä mahdollistaa nopean reagoinnin käyttäjän toimintaan. [22.] JQuery-ohjelmistokirjasto on suunniteltu yksinkertaistamaan selainohjelmointia, mikä toteutuu sen tarjoamalla valmiilla funktioilla. JQuery on parhaimmillaan etenkin animaatioiden ja Ajax-sovelluksien toteutuksessa, joihin on siis tarjolla yksinkertaiset ja valmiit funktiot. JQuery on erillinen JavaScript-tiedosto, joka sisällytetään WWW-sivuun linkittämällä se paikallisena kopiona tai julkisella palvelimella sijaitsevana kopiona. [23.]

### 3.3.2 Kirjautumisjärjestelmä

Zatin kirjautumisjärjestelmä on toteutettu hyödyntäen istuntoja, jotka PHP mahdollistaa. Kirjautumisen yhteydessä istuntoihin tallennetaan käyttäjän käyttäjätunnus, salasana sekä tieto siitä, että käyttäjä on kirjautuneena sisään. Sivusto tarkistaa tämän tiedon jokaiselle sivulle mentäessä, minkä perusteella se suorittaa jommankumman kahdesta vaihtoehdosta. Jos istunto ei sisällä tietoa kirjautumisesta, käyttäjälle ei näytetä mitään muuta kuin erillisessä tiedostossa sijaitseva kirjautumislomake, joka liitetään sivulle. Jos istunto palauttaa tiedon kirjautuneesta käyttäjästä, tälle näytetään sivun sisältö normaalisti ilman kirjautumislomaketta.

Kun käyttäjä kirjautuu sivustolle täytettyään kuvassa 6 näkyvän kirjautumislomakkeen, hänet ohjataan index.php-etusivulle, jossa sivu samantien tarkistaa, onko käyttäjätunnus ja salana tallessa. Jos on, käyttäjän tiedot siirtyvät tarkistukseen checklogin.php - sivulle, josta ei kuitenkaan näy itse käyttäjälle mitään. Jos tiedot täsmäävät, käyttäjä ohjataan takaisin etusivulle (kuva 7), josta hän voi jatkaa haluamaansa sivuston osaan.

Käyttäjän kirjautuessa ulos istunto tietoineen tuhoetaan, jolloin hän näkee sivuston tarkistuksen myötä kirjautumislomakkeen.

### 3.3.3 Huonelistaus

Etusivulla sijaitseva huonelistaus koostuu suorakulmioista, jotka kukin edustavat yksittäistä chat-huonetta. Suorakulmioiden koko määräytyy huoneessa olevien keskustelijoiden mukaan. Jos keskustelijoiden lukumäärä on suurempi tai yhtä kuin 10, asetetaan suorakulmion leveydeksi 210 pikseliä. Muussa tapauksessa leveys on 90 pikseliä. Keskustelijoiden määrän ollessa 50 tai sitä suurempi, suorakulmion korkeudeksi asetetaan 167. Määrän ollessa alle 50, suorakulmion korkeus on 70. Näin huoneita vastaaville suorakulmioille saadaan kolme eri kokovaihtoehtoa, joiden perusteella käyttäjä näkee helposti suosituimmat huoneet.

Koon lisäksi jokaista huonetta vastaava suorakulmio jaotellaan jompaan kumpaan kahdesta ryhmästä: valmiiksi luotuihin tai käyttäjien luomiin huoneisiin. Tämä onnistuu tarkistamalla huoneen luojan id - valmiiksi luoduilla huoneilla se on 0, kun taas käyttäjien luomilla huoneilla se on huoneen luoneen käyttäjän id.

Jotta käyttäjä pysyisi jatkuvasti selvillä huoneiden keskustelijamääristä, listaus päivitetään säännöllisin väliajoin näyttämään sen hetkinen tilanne. Tämä toteutetaan jQuery:n avulla.

```
<script type="text/javascript">
var auto_refresh = setInterval(function () {
$('#roomlisting').load('roomlisting.php').fadeIn("slow");
}, 4000); // päivitetään chat-huonelistaus 4 sekunnin välein
</script>
```

#### Koodiesimerkki 1. Huonelistauksen päivitysfunktio

Koodiesimerkin 1 funktio hakee neljän sekunnin välein dokumentista roomlisting id:n omaavan elementin, jonka se korvaa roomlisting.php-sivun sisällöllä. Kyseinen sivu hakee huoneiden tiedot ja listaa huoneet aiemmin mainituilla tavalla kahteen ryhmään erikokoisiksi suorakulmioiksi.

#### 3.3.4 Chat-huone

Sivuston suurin osio on chathuone, joka sijaitsee chat.php-tiedostossa. Huoneen latautuessa näkyviin sivu tekee runsaasti erilaisia tietokantahakuja ja -päivityksiä. Aluksi tietokannassa sijaitsevaa huoneessa olevien keskustelijoiden lukumäärää kasvatetaan yhdellä. Samoin päivitetään käyttäjän tietoihin ko. huone, jossa hän on läsnä. Tämä tieto tarvitaan, jotta tietokannasta voidaan hakea kaikki huoneessa olevat käyttäjät. Lueteltujen toimenpiteiden lisäksi itse keskusteluviestit haetaan kannasta.

```
//Ladataan huoneen viestit chatlog.php:n kautta
function loadLog(){
$.ajax({
url: "chatlog.php?roomid="+<?php echo $roomid;?>,
type: 'get',
dataType: 'html',
async: false,
cache: false,
success: function(data) {
$('#chatbox').html(data);//lisätään ladatut viestit chatboxiin
$('#chatbox').scrollTop($('#chatbox').attr("scrollHeight"));//skrollataan alas
//viimeisimpään viestiin
}
});
}
setInterval (loadLog, 2500); //Päivitetään chatbox 2,5 sekunnin välein
```

#### Koodiesimerkki 2. jQuery-funktio viestilistaukseen



```

<?php
include "base.php";
$roomid = $_GET["roomid");//ajaxin välittämä huoneen id, jossa keskustellaan
$chatmessages = mysql_query("SELECT * FROM chatmessages ORDER BY id");
for( $laskuri=0 ; $laskuri<mysql_num_rows($chatmessages) ; $laskuri++ )//listataan viestit
{
    if(mysql_result( $chatmessages, $laskuri, "room") == $roomid) {
        $sentby = mysql_result( $chatmessages, $laskuri, 'sentby');
        $users_result = mysql_query("SELECT Username FROM users WHERE UserID = '". $sentby. "'");
        $fetchedname = mysql_fetch_array($users_result);
        $sendername = $fetchedname["Username"];
        $time = new DateTime(mysql_result( $chatmessages, $laskuri, 'time'));
        echo '<div class="msgln">'.date_format($time, "H:i").' <b>'. $sendername. '</b>: '
            .mysql_result( $chatmessages, $laskuri, "message").'</div>';
    }
}
?>

```

### Koodiesimerkki 3. Viestien haku tietokannasta

Keskusteluviestien hakutoimenpide on sijoitettu erilliseen chatlog.php-tiedostoon. 2,5 sekunnin välein suoritettava jQuery-skripti (koodiesimerkki 2) lähettää Ajax-kutsun avulla kyseiselle chatlog.php-sivulle parametrina huoneen id:n, jonka perusteella sivu osaa hakea tietokannasta oikeat viestit (koodiesimerkki 3). Ajax-funktio saa paluuarvona html-muotoon listatut viestit, jotka se lisää niille varattuun chatbox-elementtiin, joka näkyy kuvassa 8.

Huoneessa olevat käyttäjät listataan sivun oikeassa laidassa sijaitsevaan listaan, jossa jokaisesta käyttäjästä näytetään nimimerkki, joka toimii linkkinä käyttäjän profiilisivulle, sekä Y-kirjain, jota klikkaamalla käyttäjän kanssa voi aloittaa yksityiskeskustelun. Kirjainta klikatessa suoritetaan jQuery-funktio, joka tarkistaa aluksi onko yksityiskeskustelupalkki jo luotu eli onko keskustelu jo käynnissä. Enemmän yksityiskeskustelun toimintaperiaatteesta on kuitenkin sille varatussa luvussa 3.3.5.

Jotta käyttäjä voisi osallistua keskusteluun muutenkin kuin vain lukemalla viestejä, hänen täytyy pystyä kirjoittamaan ja lähettämään omat viestinsä. Tämä onnistuu tekstikentän ja lähetysnapin avulla.

```
//Chatviestin lähetys
$("#submitmsg").click(function(){
    var clientmsg = $("#usermsg").val();
    var clientroomid = <?php echo $roomid;?>;
    $.post("post.php", {text: clientmsg, roomid: clientroomid});
    $("#usermsg").attr("value", "");
    return false;
});
```

#### Koodiesimerkki 4. Viestin lähetys

Kun käyttäjä on kirjoittanut kenttään haluamansa viestin ja painaa lähetysnappia tai näppäimistön Enter-painiketta, suoritetaan koodiesimerkin 4 jQuery-funktio, joka ottaa viestin ja huoneen id:n talteen muuttujiin ja lähettää ne eteenpäin post.php-tiedostolle, joka hoitaa tietokantaan tallennuksen.

```
<?php
include "base.php";
session_start();
if(isset($_SESSION['Username'])) {
    $idresult = mysql_query("SELECT UserID FROM users WHERE Username = '".$_SESSION['Username']."'");
    $fetchedid = mysql_fetch_array($idresult);
    $myid = $fetchedid['UserID'];
    $time = date("Y-m-d H:i:s");
    $text = $_POST['text'];
    $roomid = $_POST['roomid'];
    $sqladd = mysql_query("INSERT INTO `chatmessages`(`id`, `message`, `sentby`, `time`, `room`)
VALUES (NULL, '".stripslashes(htmlspecialchars($text))."', '$myid.', '$time.', '$roomid.')");
}
?>
```

#### Koodiesimerkki 5. Viestin talletus tietokantaan

Tiedosto tallentaa kantaan saatujen muuttujien lisäksi lähetysajan sekä lähettäjän id:n koodiesimerkissä 5 näkyvällä tavalla. Viestien päivitysfunktio hakee enintään 2,5 sekunnin kuluttua tallennetut tiedot näkyviin viesti-ikkunaan.

Huoneesta poistumiseen on varattu Poistu-näppäin viesti-ikkunan oikeaan ylälaitaan. Sitä klikattaessa käyttäjältä kysytään haluaako hän varmasti poistua huoneesta. Jos käyttäjä valitsee OK-vaihtoehdon, tietokannassa sijaitsevasta huoneen keskustelijalukumäärästä vähennetään yksi sekä käyttäjän tiedoista poistetaan läsnäolotieto. Tämän jälkeen käyttäjä ohjataan etusivulle.

### 3.3.5 Yksityiskeskustelu

Yksityiskeskustelun avulla kaksi käyttäjää voivat lähettää toisilleen viestejä, joita muut huoneessa läsnäolevat käyttäjät eivät näe. Tämä ominaisuus toteutettiin hyvin samaan tapaan kuin normaalikin chat-keskustelu.

```
//Luodaan yksityiskeskustelupalkki
function priv(id){
    if($("#messages"+id).length == 0) {//tarkistetaan onko yksityiskeskustelu jo auki
        $.ajax({
            url: "privmessages.php?toid="+id,
            type: 'get',
            dataType: 'html',
            async: false,
            cache: false,
            success: function(data) {
                $("#privatebar").append('<td width="300px" id="messages'+id+'></td>');
                $("#messages"+id).html(data);//lisätään itse privapalkki luotuun soluun
                $('#slideText'+id).scrollTop($('#slideText'+id).attr("scrollHeight"));//skrollataan alas
                //viimeisimpään viestiin
            }
        });
        setInterval(function(){ updatepriv(id);}, 2500);
    }
}
```

#### Koodiesimerkki 6. jQuery-funktio yksityiskeskustelupalkin luomiseen

Kuten chat-huoneen yhteydessä mainittiin, käyttäjän klikatessa käyttäjälistauksessa olevaa Y-kirjainta suoritetaan jQuery-funktio (koodiesimerkki 6), joka tarkastaa ensin, onko yksityiskeskustelupalkki jo luotu. Jos palkkia ei löydy, kutsutaan Ajax-funktion avulla yksityisviestit listaavaa php-tiedostoa, joka keskustelukumppanin id:n perusteella listaa viestit ja palauttaa ne funktiolle. Funktio luo tarvittavat elementit ja lisää lopulta haetut viestit siististi näkyviin. Tämän lisäksi funktio asettaa erillisen päivitysfunktion hakemaan ja listaamaan yksityisviestit 2,5 sekunnin välein.

```
//tarkastetaan onko yksityisviesteissä viestejä, joita käyttäjä ei ole nähnyt
function checkPriv(){
    $.ajax({
        url: "checkpriv.php",
        type: 'get',
        dataType: 'html',
        async: false,
        cache: false,
        success: function(data) {
            script = $(data).text();
            eval(script);//suoritetaan paluuarvona mahd. saatu skripti eli luodaan
            //yksityiskeskustelupalkki vastaamaan oikeaa keskustelijaa
        }
    });
}
```

#### Koodiesimerkki 7. Tarkastetaan mahdolliset yksityisviestit, joita käyttäjä ei ole nähnyt

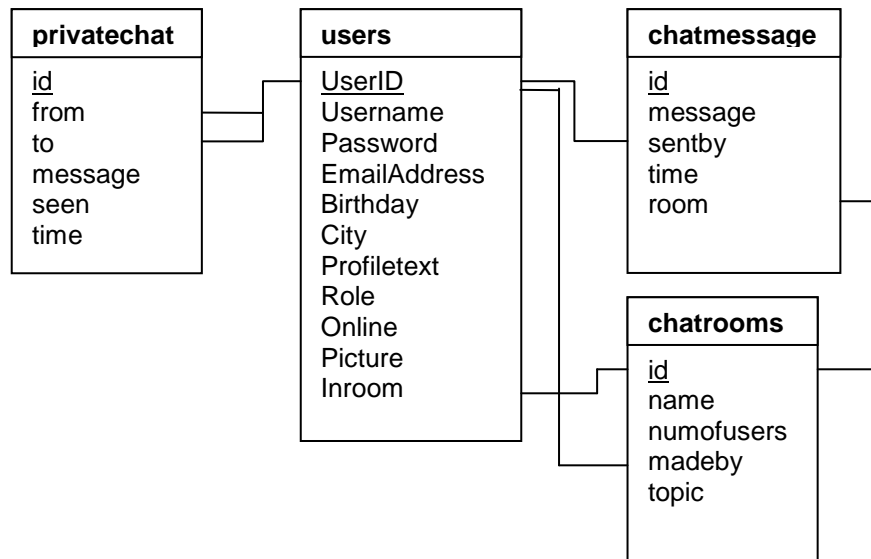
Yksityiskeskustelu vaatii toimiakseen myös toisen jQueryn ja Ajaxin yhteistyötä vaativan funktion (koodiesimerkki 7). Tämä kyseinen funktio tarkistaa 2,5 sekunnin välein, onko käyttäjälle lähetetty yksityisviestejä, joita hän ei ole nähnyt. Jos sellainen viesti löytyy, niin sen lähettäjää vastaava yksityiskeskustelupalkki luodaan ellei sellaista ole jo luotu aiemmin. Näin käyttäjä saa tiedon, että joku on aloittanut yksityiskeskustelun hänen kanssaan.

Uuden yksityisviestin lähetys on toteutettu samaan tapaan kuin yleisten chatviestien lähetys. Kun käyttäjä kirjoittaa tekstikenttään viestinsä ja painaa lähetysohjetta tai vaihtoehtoisesti näppäimistöön Enter-painiketta, suoritetaan jQuery-funktio, joka lähettää privpost.php-tiedostolle viestin sisällön sekä vastaanottajan id:n. Tiedosto tallentaa tiedot tietokantaan, josta ne ladataan näkyviin muiden viestien kera yksityisviestipalkkiin.

Yksityisviestipalkin sulkemiseen on varattu palkin oikeaan kulmaan X-painike, jota klikkaamalla suoritetaan palkin sulkemisfunktio. Funktio käytännössä vain poistaa palkkia vastaavan elementin html-koodista, jolloin sen saa näkyviin taas klikkaamalla Y-kirjainta käyttäjälistauksessa.

### 3.3.6 MySQL-tietokanta

Luvussa 3.2.2 tuli ilmi, että kaikki Zatin käyttämä data päätettiin säilöä MySQL-tietokantaan. Kuvan 1 osoittamalla tavalla data jaettiin neljään tauluun.



**Kuva 11. Tietokannan relaatiokaavio**

Users-taulu sisältää kaikkien käyttäjätilien tiedot, joita ovat id, kirjautumiseen käytettävät käyttäjätunnus sekä salasana, käyttäjän antamat henkilökohtaiset tiedot, joita ovat sähköpostiosoite, syntymäaika, asuinpaikka ja oma teksti sekä käyttäjän rooli sivustolla, hänen kirjautumistilansa sekä tieto siitä, missä huoneessa käyttäjä on keskustelemassa. Myös käyttäjän profiilikuvan tiedostonimi on tallennettu käyttäjän tietoihin, jotta oikea kuva voidaan näyttää käyttäjän profiilisivulla.

Chatrooms-tauluun on sijoitettu chathuoneiden tiedot eli huoneen nimi, id, huoneessa olevien keskustelijoiden lukumäärä, huoneen luoneen käyttäjän id sekä huoneen aihe teksti.

Chatmessages-taulussa säilytetään kaikki huoneissa kirjoitetut viestit. Kullakin viestillä on itse viestitekstin lisäksi oma id, sekä tieto siitä, kenen kirjoittama ja lähettämä viesti on, lähetysaika sekä sen huoneen id, jossa viesti on kirjoitettu.

Yksityisviestit tallennetaan privatechat-tauluun, jossa jokaisesta viestistä on tallessa id, lähettäjän ja vastaanottajan id:t, itse viesti, lähetysaika sekä tieto siitä, onko viesti luettu vai ei. Jälkimmäisin tieto voi olla joko 0, jolloin viestiä ei ole vielä luettu tai 1, jolloin viesti on luettu.

Kuten kuva 11 osoittaa, taulut ovat toisiinsa liitoksissa huone- sekä käyttäjät:n kautta.

### 3.4 Kehitysideoita

Zat-sivustosta tuli monella tapaa sellainen kuin alunperin oli tarkoitus. Toteutuksen yhteydessä tuli kuitenkin muutamia asioita, joita voitaisiin ottaa huomioon mahdollisessa jatkokehityksessä.

Sivuston ulkoasu on tietenkin eräs osa-alue, jota voi kehittää loputtomiin. Zatin kohdalla entistäkin pelkistetympi ja sirompi ulkoasu voisi viedä koko sivustoa parempaan suuntaan. Myös navigaatiota sivuston osien välillä tulisi mietittyä uudelleen, vaikka se on melko toimiva jo nykyisellään.

Profiilisivussa on paljon kehitysmahdollisuuksia, koska tällä hetkellä se kertoo käyttäjistä vain muutaman tiedon. Jos käyttäjälle mahdollistettaisiin julkaista entistä enemmän tietoa itsestään tekstin ja kuvan muodossa, sivustosta tulisi entistä monimuotoisempi ja yhteisöllisempi nykytrendin mukaisesti.

Tietenkin suorituskyky on myös aina yksi tärkeä kehityskohde, jota saisi paremmaksi etenkin viestien latauksen yhteydessä. Olisi hyvä kartoittaa, entisten lisäksi, mahdolliset vaihtoehdot, joilla viestit saataisiin nopeasti ja kevyesti näkyviin tietokannasta.

Loppujen lopuksi Zat toimii jo ilman jatkokehitystä hyvin, mutta jos siitä haluttaisiin laajempi, suuremman yleisön sivusto, niin se vaatisi joitakin em. ulkoisia sekä rakenteellisia muutoksia.

## 4 ZatApp-mobiilisovellus

### 4.1 Esittely

Zatin tueksi toteutettiin ZatApp-niminen mobiilisovellus Windows Phone 7 -matkapuhelimille. Zat-sivustoahan pystyy selaamaan nykyisillä älypuhelimilla selaimen kautta sujuvasti, mutta käyttökokemuksesta haluttiin luoda entistä parempi ja kevyempi

ZatAppin avulla. Se on ikäänkuin riisuttu malli Zatista, jota on tarkoitus käyttää silloin, kun käyttäjä ei pääse osallistumaan keskusteluun tietokoneellaan. ZatAppista lähetettyjä viestejä ei erotella mitenkään eli sekä Zatissa että ZatAppissa näkyy kummastakin järjestelmästä lähetetyt viestit.



**Kuva 12. Kirjautumislomake**

Zatin tapaan ZatApp vaatii aluksi sisäänkirjautumisen tunnuksella, joka on aiemmin luotu Zatin rekisteröitymislomakkeella. Kun käyttäjä antaa tunnuksensa ja salasanansa kirjautumissivulla (kuva 12) ja klikkaa Kirjaudu-painiketta, järjestelmä tarkistaa niiden oikeellisuuden tietokannasta ja ilmoittaa, jos tiedot ovat virheelliset. Onnistuneen kirjautumisen jälkeen ZatApp näyttää käyttäjälle Zatista tutun huonelistauksen.



**Kuva 13. Huonejaottelu valmiisiin ja käyttäjien luomiin huoneisiin**

Kuten kuvasta 13 käy ilmi, huoneet on jaoteltu valmiisiin ja käyttäjien luomiin huoneisiin, jotka on listattu kahdelle eri sivulle. Käyttäjä voi liikkua näiden sivujen välillä liu'uttamalla sormeaan puhelimen ruudulla sivulta toiselle tai klikkaamalla sivun otsikkoa. Klikkaamalla huonetta vastaavaa suorakulmiota käyttäjä pääsee sisään huoneeseen.



**Kuva 14. Chat-huone**



ZatAppissa huone koostuu kahdesta näytettävästä sivusta. Ensimmäisellä sivulla käyttäjälle näytetään huoneeseen lähetetyt viestit, ja hän voi kirjoittaa sekä lähettää oman viestinsä ruudun alareunan tekstikentän kautta. Mobiiliuden tuomana ominaisuutena käyttäjä voi myös ilmoittaa sen hetkisen olinpaikkansa sille varatulla näppäimellä. Toiselle sivulle on listattu huoneessa sillä hetkellä olevat keskustelijat. Näidenkin sivujen välillä käyttäjä voi liikkua liuttamalla sormeaan ruudulla puolelta toiselle. Kumpikin sivuista päivittyy tietyin väliajoin näyttäen reaaliaikaiset tiedot. Suurin ero Zat-sivustoon verrattuna on yksityiskeskustelu, jota ZatAppiin ei toteutettu ollenkaan.

## 4.2 Suunnittelu

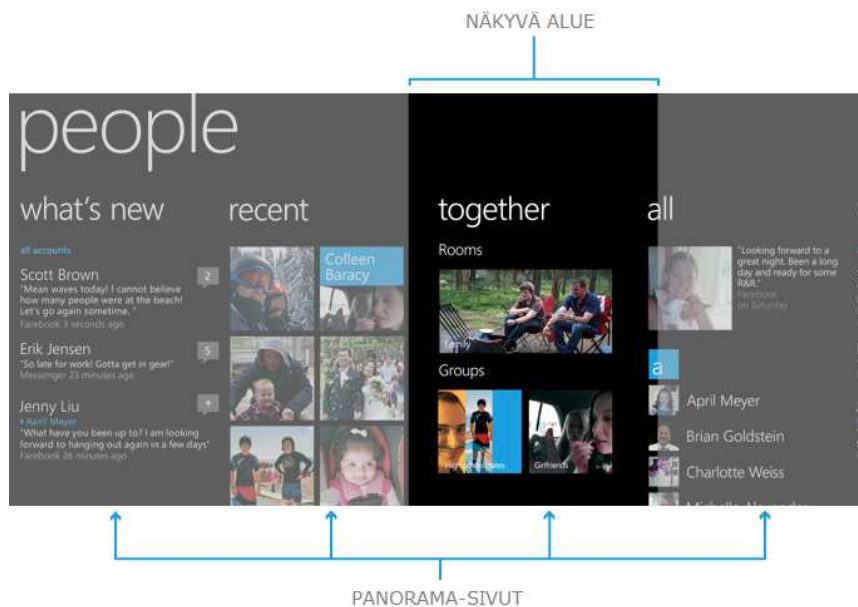
Kuten Zat-sivuston kohdallakin, ZatAppin suunnittelu lähti liikkeelle ulkoisten asioiden määrittelystä: mitä kaikkea tietoa käyttäjälle halutaan näyttää ja miten nämä tiedot on jaoteltu ruudulle. Koska puhelimen ruutu on huomattavasti tietokoneen ruutua pienempi, täytyi ruudulla näytettäviä asioita jakaa tarkemmin osiin verrattuna Zatin käyttöliittymään. Tietenkin mobiilisovelluksen mahdollisuudet tietokantatyypin datan käsittelyyn oli selvitettävä, jotta viestien lukeminen ja kirjoittaminen sujuisi mahdollisimman mutkattomasti.

### 4.2.1 Sovelluksen eri osat

ZatApp suunniteltiin Zatin tueksi, joten siirtyminen Zatin käyttämisestä ZatAppin pariin tulisi olla mahdollisimman helppoa ja luonnollista. Jotta tämä toteutuisi käytännössä, Zatin ulkoasu ja toimintaperiaate tuotiin ZatAppiin lähes sellaisenaan. Toisin sanoen käyttäjä kohtaa samat asiat samoissa vaiheissa käyttäessään kumpaa tahansa ympäristöä. Kuten Zatisa, ZatAppiinkin suunniteltiin tietty polku, jonka kautta käyttäjä etenee lopulliseen päämääräänsä eli keskusteluun.

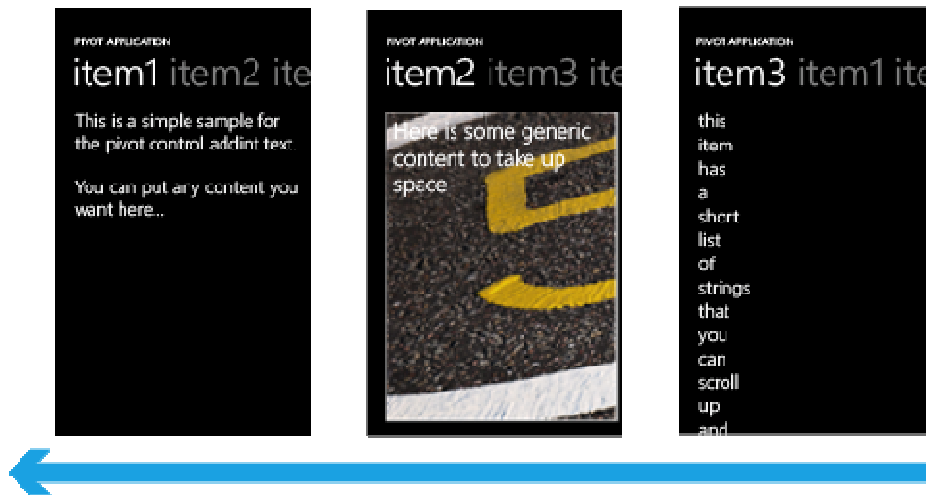
Aluksi sovelluksen auetessa tulee kysyä käyttäjänimi ja salasana, joiden oikeellisuus tarkistetaan. Jos tiedoilla löytyy rekisteröity käyttäjätunnus, käyttäjä ohjataan ns. etusivulle eli huonelistaukseen. Huonelistauksessa listataan keskusteluhuoneet samalla tavalla kuin Zatisa, eli suorakulmioihin, jotka kukin vastaavat tiettyä keskusteluhuonetta. ZatAppiin haluttiin Zatin tapaan myös mahdollisuus jaotella huoneet valmiisiin ylläpitäjien luomiin sekä käyttäjien itse luomiin huoneisiin.

Matkapuhelimien rajallinen näytön koko esti kuitenkin täysin samantyyppisen asettelun, kuin mitä Zatin yhteydessä oli suunniteltu. Windows Phone -ympäristö mahdollistaa kuitenkin muutaman eri mallin näkymän eräänlaiseen laajentamiseen, joista kaksi olivat potentiaalisia vaihtoehtoja: Panorama- ja Pivot-käyttöliittymämalli.



**Kuva 15. Panorama-käyttöliittymämalli [24.]**

Ensimmäinen vaihtoehto näkymän laajentamiseen oli Panorama-käyttöliittymämalli, jossa näkymän todellinen leveys on usean ruudunmitan levyinen. Käytännössä tämä tarkoittaa sitä, että käyttöliittymään on aseteltu kuvassa 15 näkyvällä tavalla peräkkäin useita sovelluksen sivuja, joista muodostuu yhdessä panoraamamainen, leveä kaistale. Käyttäjän vetäessä sormeaan ruudun puolelta toiselle, vaihtuu ruudulla näytettävä sivu sulavasti seuraavaan tai edelliseen. Panorama sopii hyvin tilanteisiin, joissa paljon dataa sisältävä näkymä halutaan pitää yhtenäisenä erillisistä sivuista huolimatta.[24.]



**Kuva 16. Pivot-käyttöliittymämalli [25.]**

Hyvin samankaltainen, mutta hieman Panoramaa yksinkertaisempi käyttöliittymämalli on Pivot. Se muistuttaa paljon tietokoneiden internetselainten välilehtimallia, jossa eri sivut ovat omissa välilehdissään ja lehden otsikkoa klikatessa siirrytään tarkastelemaan tätä vastaavaa sivua. Kuten kuvasta 16 nähdään, Pivot-käyttöliittymässä sovelluksen sivut on sijoitettu Panoraman tapaan peräkkäin, mutta siirtyminen näiden välillä ei tapahdu niin sulavasti ja yhtenäisesti kuin Panoramassa. Pivot-käyttöliittymää on hyvä käyttää, jos sivu sisältää erityyppistä tietoa, jotka halutaan erottaa selkeästi toisistaan erillisillä sivuilla, mutta joiden välillä voidaan kuitenkin liikkua helposti. [25.]

Huonelistauksessa päädyttiin lopulta käyttämään Pivot-käyttöliittymämallia, koska sillä saadaan selkeämmin erotettua kaksi eri huonekategoriaa toisistaan. Jos käytettäviä sivuja olisi ollut enemmän, Panorama olisi ollut vahvempi vaihtoehto, mutta kahdella sivulla Pivot tarjoaa selkeämmän ratkaisun. Pivot-malli sopii vieläkin paremmin itse keskusteluosioon, jolloin oli loogista käyttää samaa mallia koko sovelluksessa.

Huonetta vastaava suorakulmiota klikatessaan käyttäjä siirtyy itse keskusteluosioon eli sisälle huoneeseen. Huone sisältää samat tiedot ja osat kuin Zatin vastaava. Viestiosa sisältää huoneessa kirjoitetut viestit sekä viestin kirjoittamiseen varatun tekstikentän. Tämän lisäksi käyttäjälle näytetään lista huoneessa läsnäolevista keskustelijoista. Nämä tiedot on jaettu Pivot-mallin avulla kahteen osaan:

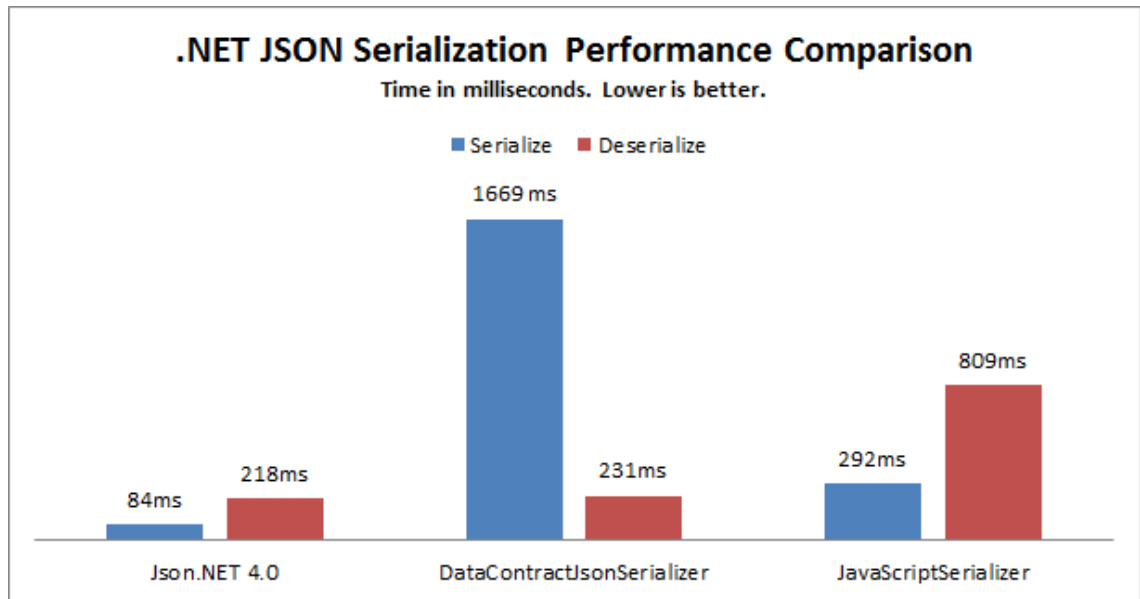
ensimmäisellä sivulla näytetään viestiosa viesteineen ja toisella sivulla listataan keskustelijat.

#### 4.2.2 Tiedonsiirto

Zatin toteutuksen yhteydessä luotiin tietokanta käyttäjätietoja, viestejä ja huonetietoja varten. ZatApp käyttää luonnollisesti näitä samoja tietoja ja tallentaa myös uutta tietoa kantaan, kuten käyttäjän kirjoittamat viestit. Tätä varten oli selvitettävä Windows Phonen mahdollisuudet toimia MySQL-tietokannan kanssa.

Windows Phone ei mahdollista suoraa yhteyttä SQL-tietokantaan samaan tapaan kuin tietokone. Tämä tarkoittaa sitä, että Windows Phonen kirjastot eivät sisällä sellaisia funktioita, joilla voitaisiin ottaa suora yhteys tietokantaan ja lukea tai kirjoittaa sinne dataa. [26.] Täytyi siis löytää jokin keino päästä käsiksi tietokantaan muutaman mutkan kautta.

Ainoa varteenotettava vaihtoehto mobiilisovelluksen ja tietokannan väliseen tiedonsiirtoon oli JSON (JavaScript Object Notation), joka on yksinkertainen tiedonsiirtomuoto käytettäväksi nimensä mukaisesti JavaScript-ohjelmissa. Tästä huolimatta JSON on JavaScriptistä riippumaton eli sitä voidaan käyttää muillakin kielillä. Sekä Php:stä, että Windows Phonen C#:sta löytyy tuki JSON:lle [27.]. C#:n osalta ei kuitenkaan tyydytty valmiiseen JSON-kirjastoon, vaan tähän tarkoitukseen ladattiin ja lisättiin Json.NET-ohjelmakehys, joka on tavallista JSON:a yksinkertaisempi syntaksiltaan ja nopeampi suorituskyvyiltään. [28.]



**Kuva 17. JSON-toimintojen suorituskykyvertailu [28.]**

Kuvasta 17 näkyy vertailu Json.NET 4.0:n, valmiin JSON:n (kuvassa keskimäinen) sekä JavaScriptin JSON:n välillä. Siniset pylväät edustavat Serialize-toimintoon kuluvaa aikaa millisekunneissa. Serialize muuntaa ohjelman käyttämät oliot JSON-muotoon. Punaiset pylväät näyttävät päinvastaiseen toimintoon, Deserializeen, kuluvan ajan. Deserialize muuntaa JSON-muotoisen tiedon ohjelman käyttämäksi olioksi. Kuten kuvasta siis käy ilmi, Json.NET on kummallakin osa-alueella kilpailijoitaan nopeampi ja näin erinomainen valinta mobiilisovellukseen, jossa on rajalliset resurssit.

Json.NET ei kuitenkaan yksinään riitä, jotta saataisiin yhteys valmiina olevaan tietokantaan. Yksi mutka tarvittiin vielä ja se oli Php. Kuten edellä mainittiin, Php:stä löytyy myös tuki JSON:lle. Suunniteltiin siis muutama Php:llä toteutettava apusivu, joiden avulla hoidetaan tiedonsiirto tietokannan ja JSON:n välillä.

## 4.3 Toteutus

### 4.3.1 Käytetyt ohjelmat sekä ohjelmointikielet

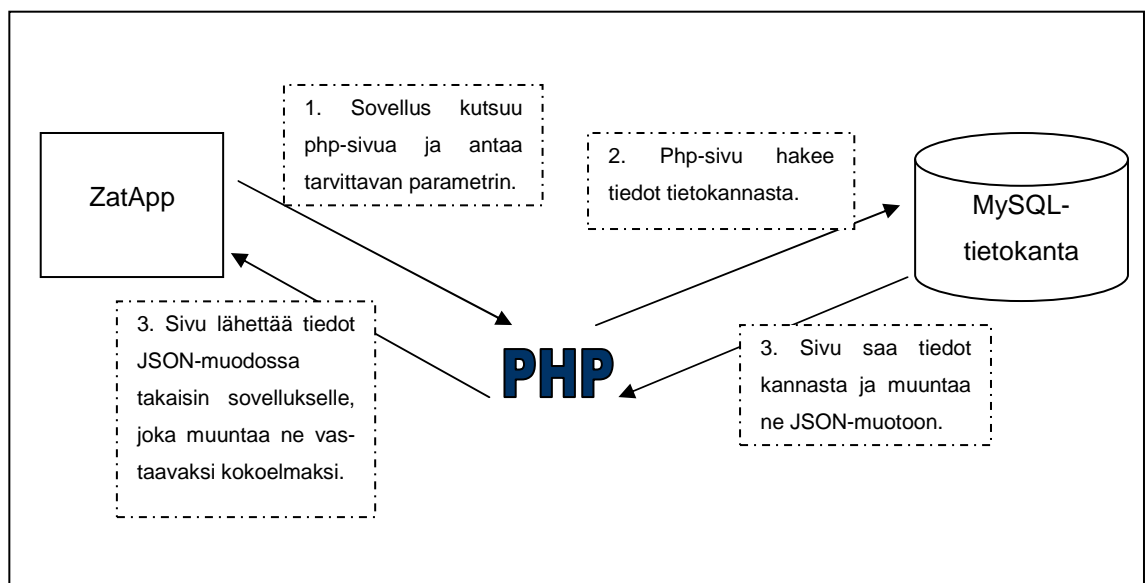
ZatApp-sovelluksen toteuttamiseen käytettiin Visual Studio 2010 Express for Windows Phone -työkalua, joka on Microsoftin tarjoama, nimenomaan Windows Phone -kehitykseen tarkoitettu maksuton versio Visual Studiosta [29]. Sovelluksen käyttöliittymä luotiin työkalun tarjoamilla valmiilla elementeillä, jotka kuvataan

koodipuolella XAML:lla. XAML on Microsoftin luoma deklarativinen kieli, joka perustuu vahvasti XML-kieleen [30]. Sovelluksen syvempi toiminta ohjelmoitiin käyttäen C#-kieltä, joka on myös Microsoftin kehittämä ohjelmointikieli. Sekä XAML, että C# kehitettiin .NET-konseptia varten, johon myös Windows Phone -ohjelmistokehitys pohjaa. [31.]

Microsoftin ympäristön lisäksi sovelluksen toteuttamiseen vaadittiin myös Php:ta, jota ohjelmoitiin Zat-sivuston yhteydessä esitellyllä Notepad++-editorilla. C#:n ja Php:n väliseen tiedonsiirtoon valittiin JSON-muotoisten tietojen käsittelyyn kehitetty, luvussa 4.2.2 esitelty, avoimen lähdekoodin Json.NET-ohjelmistokehitys.

#### 4.3.2 Tiedonsiirto tietokannan ja sovelluksen välillä

Suunnitteluvaiheessa todettiin, että oli löydettävä jokin tapa jakaa tietoa sovelluksen ja MySQL-tietokannan välillä. Tähän tarkoituksen Visual Studioon lisättiin Json.NET-kirjasto, jolla tiedot muunnetaan JSON-muotoon ja päinvastoin. Itseasiassa ZatAppin kohdalla käytettiin ainoastaan tätä päinvastastaista tapaa eli JSON-muotoiset tiedot muunnetaan sovelluksen käyttämiksi objekteiksi. Seuraavassa eri tilanteet, jotka vaativat tiedonsiirtoa sovelluksen ja kannan välillä.

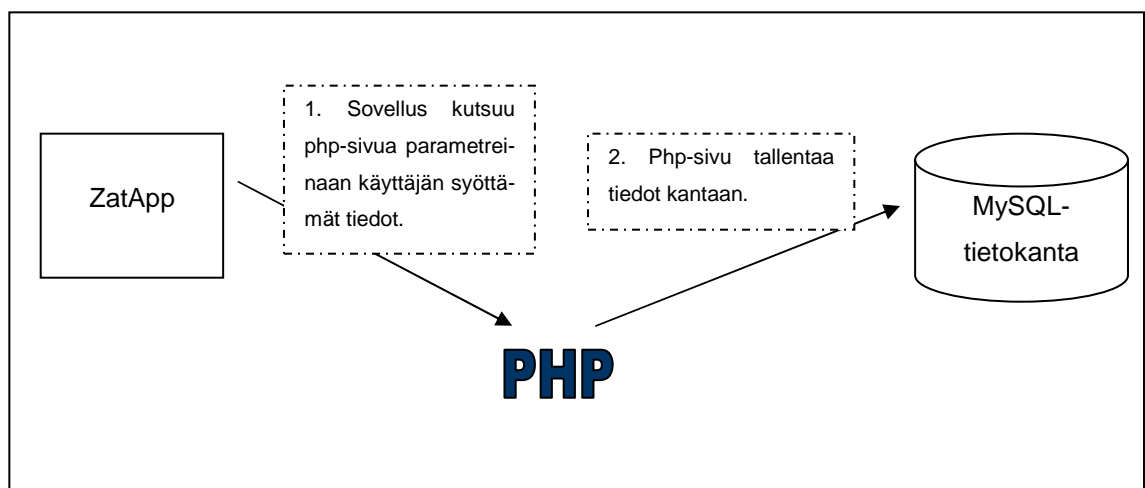


Kuva 18. Tiedonhaku tietokannasta sovelluksen käyttöön

Kuva 18 havainnollistaa tilanteen, jossa ZatApp-sovellus tarvitsee tietoa tietokannasta, ja sen saamiseksi se ottaa yhteyttä tiettyyn php-sivuun. Tällainen tilanne on kyseessä, kun listataan huoneet, viestit tai huoneessa olevat käyttäjät. Käytetään esimerkkinä viestilistausta. Kuten kuvasta näkyy, sovellus kutsuu aluksi php-sivua, jolle se kutsun yhteydessä antaa tässä esimerkkitapauksessa parametrina sen hetkisen huoneen id:n. Tämän jälkeen php-sivu hakee kannasta huoneen id:tä vastaavat viestit, jotka se muuntaa JSON-muotoon. Lopulta sivu lähettää JSON-muotoiset viestit ZatApp-sovellukselle, jotka se muuntaa sen käyttämään muotoon eli tässä tapauksessa viestikokoelmaksi.

Keskustelijoiden listaaminen toimii täsmälleen samalla tavalla kuin viestilistaus eli parametrina annetaan sen hetkisen huoneen id, mutta viestien sijaan php-sivu hakee kannasta käyttäjät. Huonelistaus on toteutettu muuten samaan tapaan, mutta mitään parametria ei anneta kutsuttaessa php-sivua. Sivun siis hakee kaikkien huoneiden tiedot ja lähettää ne JSON-muodossa sovellukselle, joka muuntaa ne huonekokoelmaksi.

Kirjautuminen hoituu lähes samalla tavalla eli sivulle annetaan parametreina käyttäjän syöttämät käyttäjätunnus ja salasana, joiden oikeellisuuden sivu tarkistaa tietokannasta. Php-sivu palauttaa sovellukselle paluuarvona merkkijonon "OK" tai "FAIL" sen mukaan, löytyykö kannasta annettu käyttäjätunnus-salasana -yhdistelmä. JSON:a ei siis tarvita ollenkaan.



Kuva 19. Tiedonlähetyksovellukselta tietokantaan

Kirjoitettujen viestien tallennus tietokantaan tapahtuu kuvan 19 osoittamalla tavalla. Kuten listaustoiminnoissakin myös tallennuksessa sovellus kutsuu tarkoitukseen luotua php-sivua, jolle se antaa parametreja. Tässä tapauksessa parametreina ovat kirjoitettu viesti merkkijonona, sen huoneen id, jossa viesti kirjoitettiin sekä viestin kirjoittajan käyttäjäid. Sivun selvittää vielä sen hetkisen ajan päivämäärineen, jonka se lisää saamiensa parametrien kera tietokantaan.

### 4.3.3 Kirjautuminen

Kuten esittelyluvussa kerrotaan, sovelluksen käynnistyessä käyttäjä saa puhelimensa ruudulle kirjautumislomakkeen. Lomake sisältää omat tekstikentät tunnuksen ja salasanan syöttämiseksi sekä kirjautumispainikkeen.

```
private void loginButton_Click(object sender, RoutedEventArgs e)
{
    if (usernameBox.Text == "") { MessageBox.Show("Anna käyttäjänimi"); }
    else if (passwordBox.Password == "") { MessageBox.Show("Anna salasana"); }
    else {
        client.DownloadStringCompleted +=
            new DownloadStringCompletedEventHandler(
                client_DownloadStringCompleted);
        client.DownloadStringAsync(new Uri("http://users.metropolia.fi/~m0703151/Membership/wp7_auth.php?usr=" +
            usernameBox.Text + "&psw="+ passwordBox.Password));
    }
}
```

#### Koodiesimerkki 8. Kirjautumistietojen lähetys

Kun käyttäjä on antanut tietonsa ja klikkaa kirjautumispainiketta, siirrytään painikkeen tapahtumankäsittelijään (koodiesimerkki 8). Tämä käsittelijä tarkastaa aluksi, ettei kumpikaan tekstikentistä ole jäänyt tyhjäksi, minkä jälkeen se kutsuu autentikointiin luotua php-sivua parametreinaan annetut käyttäjätunnus ja salasana. Tämän jälkeen php-sivu hakee saamallaan tiedoilla käyttäjätunnusta ja sellaisen löytyessä palauttaa sovellukselle merkkijonon "OK".



```

void client_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e) {
    if (e.Error == null) {
        if (e.Result == "OK") {
            App.ViewModel.setUsername(usernameBox.Text);
            NavigationService.Navigate(new Uri("/RoomPage.xaml", UriKind.RelativeOrAbsolute));
        }
        else {
            MessageBox.Show("Kirjautuminen epäonnistui. Tarkista käyttäjänimi ja salasana.");
        }
    }
}
}

```

#### Koodiesimerkki 9. Php-sivulta saadun vastauksen käsittely

Sovellus on siirtynyt tällä välin kirjautumispainikkeen tapahtumankäsittelijästä odottamaan vastausta php-sivulta. Kun vastaus saadaan ja jos siinä ei ilmene virheitä, tarkastetaan, onko saatu paluarvo "OK" eli löytyikö tunnuksella ja salasanalla rekisteröity käyttäjä (koodiesimerkki 9). Jos näin on, käyttäjä siirretään huonelistaukseen (RoomPage.xaml). Muussa tapauksessa käyttäjälle kerrotaan kirjautumisen epäonnistuneen.

#### 4.3.4 Huonelistaus

Huonelistauksessa listataan kaikki Zatin huoneet kahdelle eri sivulle riippuen siitä, onko huone valmis ylläpitäjän luoma vai normaalin käyttäjän luoma huone.

```

clientti = new WebClient();
clientti.DownloadStringCompleted +=
    new DownloadStringCompletedEventHandler(clientti_DownloadStringCompleted);
clientti.DownloadStringAsync(new Uri("http://users.metropolia.fi/~m0703151/Membership/wp7_rooms.php"));
timer = new DispatcherTimer(); //ajastin huonelistan lataukseen
timer.Interval = TimeSpan.FromSeconds(7); //asetetaan ajaksi 7 sekuntia
timer.Tick += new EventHandler(loadRooms);
timer.Start();

```

#### Koodiesimerkki 10. Huonelistauksen lataus ja ajastus

Sovelluksen siirtyessä huonelistaus-näkymään, se kutsuu välittömästi php-sivua, joka hakee tietokannasta kaikki luodut keskusteluhuoneet. Tämän jälkeen luodaan vielä ajastin, joka suorittaa php-sivun kutsufunktion seitsemän sekunnin välein (koodiesimerkki 10).

```

        $rows[] = array ('id' => $id ,
                        'name' => $name,
                        'numofusers' => $numofusers,
                        'madeby' => $madeby
                        );
    }
    echo "{ 'rooms':";
    echo json_encode($rows);
    echo "}";

```

#### Koodiesimerkki 11. Huoneiden tiedot taulukosta JSON-muotoon

Php-sivu hakee tietokannasta huoneiden tiedot, jotka se listaa yksitellen taulukkoon koodiesimerkissä 11 näkyvällä tavalla. Tämän jälkeen se tulostaa taulukon JSON-muodossa, jotta ZatApp-sovellus voi lukea ja tulkita tiedot.

```

void clientti_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e){
    if (e.Error == null){
        RoomList r1 = JsonConvert.DeserializeObject<RoomList>(e.Result);//muutetaan saatu json-objekti RoomList-viestikokelmaksi
        App.ViewModel.LoadRooms(r1);//lähetetään huonekokoelma ViewModellille, joka lisää huoneet yksitellen näkymään
    }
}

```

#### Koodiesimerkki 12. JSON-muotoisten tietojen muunnos sovelluksen käyttämäksi huonekokoelmaksi

Php-sivun palauttavat JSON-muotoiset huonetiedot muutetaan koodiesimerkissä näkyvällä Deserialize-toiminnolla huonekokoelmaksi, joka lähetetään ViewModeliin luodulle LoadRooms-funktiolle.

```

public void LoadRooms(RoomList r1)
{
    this.Rooms1.Clear();
    this.Rooms2.Clear();
    for (int i = 0; i < r1.Rooms.Count; i++){
        if (r1.Rooms[i].Madeby.Equals("0")){
            this.Rooms1.Add(new ItemViewModel() { LineOne = r1.Rooms[i].Name, LineTwo = r1.Rooms[i].Id });
        }
        else{
            this.Rooms2.Add(new ItemViewModel() { LineOne = r1.Rooms[i].Name, LineTwo = r1.Rooms[i].Id });
        }
    }
    this.IsDataLoaded = true;
}

```

#### Koodiesimerkki 13. Huoneiden jaottelu kahteen ryhmään

Huonelistausta varten on aiemmin luotu kaksi ObservableCollection-kokoelmaa - toinen valmiille huoneille ja toinen käyttäjien luomille huoneille. Koodiesimerkin 13

LoadRooms-funktio jaottelee ja lisää huoneet Madeby-tiedon perusteella jompaankumpaan kokoelmaan.

```
<ListBox x:Name="FirstListBox" Margin="0,0,-12,0" ItemsSource="{Binding Rooms1}" Height="515">
  <ListBox.ItemsPanel>
    <ItemsPanelTemplate>
      <toolkit:WrapPanel ItemHeight="200" ItemWidth="200" />
    </ItemsPanelTemplate>
  </ListBox.ItemsPanel>
  <ListBox.ItemTemplate>
    <DataTemplate>
      <StackPanel Orientation="Horizontal" Margin="0,0,0,0">
        <Button Tag="{Binding LineTwo}" Content="{Binding LineOne}" Height="190" Width="190"
          Click="Button_Click" FontSize="15" Foreground="White" Background="#FF0D67CE"
          BorderBrush="#FF004390" />
      </StackPanel>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>
```

#### Koodiesimerkki 14. Huonelistauksen käyttöliittymä kuvattuna XAML-kielellä

Huonelistauksessa haetaan huonekokoelmista huoneet kahteen ListBox-listakenttään, joista toinen koodiesimerkissä 14 XAML-kielellä kuvattuna. Jokaista huonetta vastaamaan luodaan painike, jonka tunnisteeksi annetaan huoneen id sekä näkyväksi tekstiksi huoneen nimi. Tunniste on huoneeseen siirtymistä varten. Huonepainikkeilla on yhteinen tapahtumankäsittelijä, jonne siirrytään käyttäjän klikatessa jotakin näistä painikkeista. Tunnisteen, eli huoneen id:n perusteella, tiedetään, mitä painiketta käyttäjä klikkasi. Näin hänet osataan ohjata oikeaan keskusteluhuoneeseen.

#### 4.3.5 Viestien lukeminen ja kirjoittaminen

Viestien listaaminen ja lukeminen on toteutettu hyvin samalla tavalla kuin huoneiden listaus. Kuten koodiesimerkissä 10, tiettyä php-sivua kutsutaan, mutta tällä kertaa annetaan lisäksi parametri. Tämä parametri on sen huoneen id, jonka viestejä käyttäjä on lukemassa. Viestilistaukselle asetetaan myös oma ajastin, joka määrittää, kuinka usein viestilistaus suoritetaan.

Php-sivun toiminta viestilistauksessa on myös erittäin samanlainen kuin huonelistauksen yhteydessä eli sivu hakee ja taulukoi huonetta vastaavat viestit koodiesimerkissä 11 näkyvällä tavalla. Tämän jälkeen sivu tulostaa taulukon sisällön JSON-muodossa, jonka sovellus lopulta saa paluuarvona itselleen.

```

if (e.Error == null)
{
    MessageList m2 = JsonConvert.DeserializeObject<MessageList>(e.Result); //muutetaan saatu json-objekti MessageList-viestikokelmaksi
    App.ViewModel.LoadRoom(m2); //lähetetään viestikokoelma ViewModellille, joka lisää viestit yksitellen näkymään
    if((FirstListBox.Items.Count != 0)) {
        Dispatcher.BeginInvoke(() => FirstListBox.ScrollIntoView(FirstListBox.Items.Last())); //rullataan listBox viimeiseen eli uusimpaan viestiin
    }
}

```

### Koodiesimerkki 15. JSON-muodossa olevat tiedot viestikokoelmaksi

Sovellus muuntaa koodiesimerkissä 15 näkyvällä tavalla saadun JSON-objektin viestikokoelmaksi, jonka se lähettää ViewModelin funktiolle. Tämä funktio toimii samalla tavalla kuin huonelistauksen vastaava (koodiesimerkki 13), eli se lisää jokaisen viestin yksitellen ObservableCollection-kokoelmaan, josta viestilistauksen näkymä hakee viestit näkymään käyttäjälle luettavaksi.

Viestin kirjoitus muistuttaa toiminnaltaan kirjautumista, mutta sillä eroavaisuudella, että käyttäjän syöttämät tiedot tallennetaan lopulta tietokantaan.

```

private void sendButton_Click(object sender, RoutedEventArgs e)
{
    if (messageBox.Text != ""){
        string message = messageBox.Text;
        string usr = App.ViewModel.username;
        sendClient.DownloadStringAsync(new Uri("http://users.metropolia.fi/~m0703151/Membership/wp7_send.php?roomid="
        + id + "&user="+usr+"&text=" + message));
        messageBox.Text = "";
        clientti.DownloadStringAsync(new Uri("http://users.metropolia.fi/~m0703151/Membership/wp7.php?roomid=" + id));
    }
    else{
        MessageBox.Show("Viestisi on tyhjä täynnä.");
    }
}

```

### Koodiesimerkki 16. Kirjoitetun viestin lähetys kohti tietokantaa

Kun käyttäjä on kirjoittanut haluamansa viestin tekstikenttään ja klikkaa sen vieressä olevaa ok-painiketta, sovellus siirtyy painikkeen tapahtumankäsittelijään, joka on esitetty koodiesimerkissä 16. Tapahtumankäsittelijä tarkistaa aluksi, ettei käyttäjä ole painanut painiketta vahingossa ja näin ollen lähettämässä tyhjää viestiä. Jos viesti ei ole tyhjä, niin sekä kirjoitettu viesti, että käyttäjän käyttäjätunnus haetaan omiin muuttujiinsa. Nämä lähetetään parametreina huoneen id:n kera php-sivulle. Tämän jälkeen sivu lisää tiedot tietokantaan, putsattuaan ensin viesti-merkkijonon mahdollisista html-tageista yms. Tämän jälkeen sovellus listaa viestit juuri lähetetyn viestin kera aiemmin esitellyllä tavalla käyttäjälle luettavaksi.

#### 4.3.6 Keskustelijoiden listaaminen

Keskustelijoiden listaamiseen käytettiin jälleen hyväksi havaittua tapaa huoneiden sekä viestien listauksen yhteydestä. Kun chat-huone ladataan näkyviin, suorittaa sovellus keskustelijoiden hakemisen tietokannasta php-sivun avulla. Kuten viestihaun yhteydessä, tässäkin tapauksessa php-sivulle annetaan parametrina ko. huoneen id, jotta sivu osaa hakea juuri siinä huoneessa läsnäolevat keskustelijat. Tämän jälkeiset vaiheetkin ovat hyvin samanlaisia viestilistauksen kanssa eli sovellus muuntaa saadun, keskustelijat sisältävän, JSON-objektin keskustelijakokoelmaksi, jotka lisätään ViewModelin kautta näkymään oikealle Pivot-sivulle.

```

timer = new DispatcherTimer();//ajastin uusien viestien&keskustelijoiden lataukseen
timer.Interval = TimeSpan.FromSeconds(5);//asetetaan ajaksi 5 sekuntia
timer.Tick += new EventHandler(loadStuff);
timer.Start();
}

void loadStuff(object sender, EventArgs e)
{
    clientti.DownloadStringAsync(new Uri("http://users.metropolia.fi/~m0703151/Membership/wp7.php?roomid=" + id));//viestit
    clientti2.DownloadStringAsync(new Uri("http://users.metropolia.fi/~m0703151/Membership/wp7_inroom.php?roomid=" + id));//keskustelijat
}

```

#### Koodiesimerkki 17. Keskustelijoiden listaaminen ajastetusti

Keskustelijoiden haku ja listaaminen suoritetaan säännöllisin väliajoin kuvassa 17 näkyvällä ajastimella, jota käytetään myös viestilistaukseen. Ajastimelle määritetään aika (tässä tapauksessa 5 sekuntia) , jonka välein se suorittaa sille annetun funktion (koodiesimerkin 17 loadStuff-funktio) eli kutsuu php-sivuja.

#### 4.4 Kehitysideoita

ZatApp-sovellus toteutettiin vain Zatin tueksi, joten siihen määriteltiin alun perin vähemmän ominaisuuksia verrattuna Zat-sivustoon. Joitakin ominaisuuksia olisi kuitenkin hyvä lisätä ZatAppiin mahdollisessa jatkokehitysvaiheessa.

Profiilisivut olisi yksi melko pieni muutos, jolla saataisiin ZatAppia lähemmäs itsenäistä sovellusta. Tällä hetkellä ZatAppissa ei voi tarkastella käyttäjien tietoja lähemmin, joten se toisi hyvää monimuotoisuutta sovellukseen. Myös yksityiskeskustelun mahdollisuutta voitaisiin tutkia eli saataisiinko se liitettyä järkevästi sovellukseen vai rajoittaisiko puhelimen rajallinen ruutukoko liikaa sitä.

Sekä ulkoasu- että suorituskykyparannukset ovat myös tärkeät osa-alueet jatkokehityksessä, joten varsinkin viestiliikenne puhelimen ja tietokannan välillä tulisi saada mahdollisimman hyvin optimoiduksi. Tietenkin olisi hyvä tutkia, onko edes järkevää ylläpitää erillistä sovellusta vai saavutettaisiinko sama lopputulos muuttamalla Zat-sivusto huomioimaan mobiililaitteet. Erityisesti html5- ja css3-kielten myötä responsiivisuusominaisuudet ovat helpommin huomioitavissa verkkosivustoilla [32] joten tämä olisi yksi jatkokehitysvaihtoehto. Tässä insinööriyössä haluttiin kuitenkin tutustua ja paneutua Windows Phone -kehitykseen, joten siltä osin responsiiviset sivustot eivät olleet vaihtoehtoina.

## 5 Yhteenveto

Insinööriyössä toteutettiin sekä Zatiksi nimetty chat-sivusto että sitä tukeva Windows Phone -sovellus nimeltään ZatApp. Pääpaino työssä oli sivuston toteutuksessa, koska alusta alkaen oli tiedossa, että mobiilisovellus tulisi käyttämään samaa tietokantaa ja osittain samanlaisia php-sivuja kuin sivusto. Tästä syystä nämä asiat tuli toteuttaa hyvin ja tarkasti, koska moni asia oli lopulta niistä riippuvainen. Näin myös tapahtui, ja sivustosta tuli ehjä kokonaisuus, jota Windows Phone -sovellus tukee juuri niin kuin oli tarkoitus.

Zat-sivuston kohdalla eniten haasteita tuotti yksityiskeskustelu, joka haluttiin toteuttaa JavaScriptillä, josta ei kuitenkaan ollut paljoa aiempaa kokemusta. Näin ollen se vaati paljon itseopiskelua toteutuksen aikana, mikä hidasti työn valmistumista. JavaScript oli kuitenkin oikea valinta, koska se toimii toteutetussa sivustossa hyvin ja se on myös hyvä hallita yleisestikin, koska sillä on suuri rooli kaikenlaisilla web-sivustoilla.

ZatApp-sovelluksen suurin kysymysmerkki oli alun perin tiedonsiirto tietokannan ja sovelluksen välillä, mutta siihen löydettiin yllättävänkin helposti ratkaisu Json.NETin muodossa. Json.NET vaati jonkin verran kokeilemista ja tarkempaa perehtymistä, mutta se kannatti, koska se palvelee erittäin hyvin apuvälineenä tiedonsiirrossa. ZatAppista saatiin toteutetuksi muutenkin hyvin toimiva kokonaisuus, joka täyttää sille asetetut tavoitteet ja vaatimukset.

Sivusto ja mobiilisovellus tulevat tuskin julkiseen käyttöön nykyisellään, mutta pienellä jatkokehityksellä, ominaisuuksien lisäämisellä ja tarvittavalla markkinoinnilla näistä

kahdesta saataisiin varmasti hyvin palveleva ja haluttu keskustelualusta niin verkkoon kuin mobiililaitteisiinkin. Tämänkaltaisen palvelu olisikin sitä parempi, mitä useampi käyttäjä käyttäisi sitä aktiivisesti. Mutta jos katsotaan insinööriyötä oppimisprosessina sekä asetettujen tavoitteiden täyttämisenä, niin näissä asioissa onnistuttiin ja saatiin juuri se, mitä haluttiinkin.

## Lähteet

- 1 Online chat. Verkkodokumentti <[http://en.wikipedia.org/wiki/Online\\_chat](http://en.wikipedia.org/wiki/Online_chat)> Luettu 14.4.2013
- 2 Chatti. Verkkodokumentti <<http://fi.wikipedia.org/wiki/Chatti>> Luettu 14.4.2013
- 3 Irc. Verkkodokumentti <<http://en.wikipedia.org/wiki/Irc>> Luettu 14.4.2013
- 4 Chat room. Verkkodokumentti <<http://www.businessdictionary.com/definition/chat-room.html>> Luettu 14.4.2013
- 5 Margaret Rouse, Chat room. Verkkodokumentti <<http://searchsoa.techtarget.com/definition/chat-room>> Luettu 14.4.2013
- 6 Suomi24.fi Chat. Verkkodokumentti <<http://chat.suomi24.fi/login.cgi?gid=6&category=1>> Luettu 14.4.2013
- 7 City.fi Chat. Verkkodokumentti <<http://www.city.fi/chat>> Luettu 11.2.2013
- 8 Shoutbox. Verkkodokumentti <<http://en.wikipedia.org/wiki/Shoutbox>> Luettu 14.4.2013
- 9 Riemurasia. Verkkodokumentti <<http://www.riemurasia.net>> Luettu 14.4.2013
- 10 CAPTCHA. Verkkodokumentti <<http://en.wikipedia.org/wiki/CAPTCHA>> Luettu 14.4.2013
- 11 Justin.tv. Verkkodokumentti <<http://www.justin.tv>> Luettu 14.4.2013
- 12 Facebook features - Chat. Verkkodokumentti <[http://en.wikipedia.org/wiki/Facebook\\_features#Chat](http://en.wikipedia.org/wiki/Facebook_features#Chat)> Luettu 14.4.2013
- 13 Aemon Malone, Hotmail pushes Facebook chat integration worldwide. Verkkodokumentti <<http://www.digitaltrends.com/computing/hotmail-pushes-facebook-chat-integration-live-worldwide/>> Luettu 14.4.2013
- 14 Kelsey, Todd. 2010. Social networking spaces: from Facebook to Twitter and everything in between. Apress.
- 15 Open Source Initiative, 2013. The Open Source Definition. Verkkodokumentti <<http://www.opensource.org/docs/osd>> Luettu 14.4.2013
- 16 About Notepad++. Verkkodokumentti <<http://notepad-plus-plus.org/>> Luettu 14.4.2013
- 17 Introducing WinSCP. Verkkodokumentti <<http://winscp.net/eng/docs/introduction>> Luettu 14.4.2013
- 18 Bringin MySQL to the web, phpMyAdmin. Verkkodokumentti <[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)> Luettu 14.4.2013



- 19 HTML. Verkkodokumentti. <<http://en.wikipedia.org/wiki/HTML>> Luettu 13.3.2013
- 20 Cascading Style Sheets. Verkkodokumentti <[http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)>Luettu 13.3.2013
- 21 PHP. Verkkodokumentti. <<http://en.wikipedia.org/wiki/Php>> Luettu 13.3.2013
- 22 Javascript. Verkkodokumentti. <<http://en.wikipedia.org/wiki/JavaScript>> Luettu 13.3.2013
- 23 JQuery. Verkkodokumentti <<http://en.wikipedia.org/wiki/JQuery>> Luettu 13.3.2013
- 24 Panorama control for Windows Phone. Verkkodokumentti <[http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941104\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941104(v=vs.105).aspx)> Luettu 13.3.2013
- 25 Pivot control for Windows Phone. Verkkodokumentti <[http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941098\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff941098(v=vs.105).aspx)> Luettu 13.3.2013
- 26 Wei-Meng Lee, Web Access in Windows Phone 7 Apps. Verkkodokumentti <<http://mobiforge.com/developing/story/web-access-windows-phone-7-apps>> Luettu 14.4.2013
- 27 JSON. Verkkodokumentti <<http://en.wikipedia.org/wiki/JSON>> Luettu 14.4.2013
- 28 James Newton-King, Json.NET. Verkkodokumentti <<http://james.newtonking.com/projects/json-net.aspx>> Luettu 10.3.2013
- 29 C Sharp (Programming language). Verkkodokumentti <[http://en.wikipedia.org/wiki/C\\_Sharp\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language))> Luettu 14.4.2013
- 30 Microsoft Visual Studio Express 2010. Verkkodokumentti <[http://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio\\_Express#2005-2010\\_products](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Express#2005-2010_products)> Luettu 14.4.2013
- 31 Extensible Application Markup Language. Verkkodokumentti <<http://en.wikipedia.org/wiki/XAML>> Luettu 14.4.2013
- 32 Katrien De Graeve, Responsive Web Design. Verkkodokumentti <<http://msdn.microsoft.com/en-us/magazine/hh653584.aspx>> Luettu 14.4.2013

