

Antti Haukipuro

**MEDIAMEMO-ANDROID-SOVELLUS**

# **MEDIAMEMO-ANDROID-SOVELLUS**

Antti Haukipuro  
Opinnäytetyö  
Kevät 2014  
Tietotekniikan koulutusohjelma  
Oulun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu  
Tietotekniikan koulutusohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

---

Tekijä: Antti Haukipuro  
Opinnäytetyön nimi: MediaMemo-Android-sovellus  
Työn ohjaaja: Juha Alakärppä  
Työn valmistumislukukausi ja -vuosi: Kevät 2014  
Sivumäärä: 46

---

Oppinäytetyön tavoitteena oli suunnitella ja toteuttaa Android-sovellus, jonka avulla voidaan tallentaa muistioita. Muistioiden tallennuksen tuli olla mahdollisimman sujuvaa, eikä siitä saa aiheutua käyttäjälle turhia latausaikoja, vaikka muistion koko olisi suuri. Työ tehtiin Softeri Oy:lle, joka toimi projektin tilaajana ja toimeksiantajana.

Työ aloitettiin tutustumalla Android-ohjelmointiin, PHP-ohjelmointiin sekä toimeksiantajan ohjelmistoalustaan. Ensimmäinen vaihe työstä oli Web-sovellus, johon rakennettiin sekä muistioiden hallintapaneeli että rajapinnat kommunikoidaan Android-sovelluksen kanssa. Toisessa vaiheessa valmistettiin Android-sovellus. Työkaluina projektissa toimivat NetBeans ja Eclipse.

Työ eteni aikataulun mukaan. Opinnäytetyön tuloksena syntyi MediaMemo-Android-sovelluksen versio 1.0. Sovelluksen rinnalle asennettiin myös Web-sovellus, josta voidaan hallita tallennettuja muistioita.

---

Asiasanat: Android, HTML5, rajapinnat, mobiilisovellukset

## ABSTRACT

Oulu University of Applied Sciences  
Degree programme in Information Technology, option of Software development

---

Author: Antti Haukipuro  
Title of thesis: MediaMemo-Android application  
Supervisor: Juha Alakärppä  
Term and year when the thesis was submitted: Spring 2014  
Pages: 46

---

The objective of this thesis was to design and develop Android application which allows user to save memos. Uploading memos had to be fluent and downloading/uploading shouldn't cause any problems to user. Thesis was made for Softeri Ltd. which was the subscriber and the client of the project.

Thesis started by getting familiar with Android programming, PHP programming and client's platform. The first phase of the project was to make Web application where one can control his/her memos. Web application also includes Web services which are used by the Android application. On the second phase the Android application was developed. Developing tools in this project were NetBeans and Eclipse.

Progress of thesis went as planned. The result of this project was MediaMemo-Android application version 1.0. In addition of Android application, Web application was implemented for memo control.

---

Keywords: Android, HTML5, Web services, Mobile applications

## **ALKULAUSE**

Haluan kiittää Softeri Oy -yrityksen henkilökuntaa avusta ja mahdollisuudesta tehdä opinnäytetyöni yritykselle. Lisäksi haluan kiittää työni ohjaajaa Juha Alakärppää ja viestinnän opettajaa Tuula Hopeavuorta tuesta ja avusta.

Oulunsalossa 27.5.2014

Antti Haukipuro

# SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	8
1 JOHDANTO	10
2 ANDROID-SOVELLUSKEHITYS	11
2.1 Java	11
2.2 Android SDK	11
2.3 Ohjelmointiympäristöt	11
3 KEHITYSTYÖSSÄ KÄYTETYT TEKNIIKAT	13
3.1 PHP-ohjelmointi	13
3.2 HTML5	14
3.3 Pilvipalvelut	17
3.3.1 Käyttöönottomallit	18
3.3.2 Palvelumallit	19
3.4 Web-rajapinnat	20
3.5 Käyttöliittymät	21
3.5.1 Web-sovelluksen käyttöliittymät	21
3.5.2 Android-sovelluksen käyttöliittymät	22
4 TOTEUTUSRATKAISUT	25
4.1 WEB-sovellus	25
4.1.1 Näkymät	25
4.1.2 Ohjelma-asetukset	27
4.1.3 Muistion tallennus	28
4.1.4 Tiedoston liittäminen	29
4.1.5 Tietokanta	30
4.1.6 Rajapinnat	31
4.2 Android-sovellus	32
4.2.1 Näkymät	32
4.2.2 Muistiotyyppien tallennus	38

4.2.3 Rajapintojen käyttö	40
4.2.4 Tiedoston lähetys	40
4.2.5 Tietokanta	41
5 POHDINTA	42
LÄHTEET	43

## SANASTO

AJAX	Lyhenne sanoista Asynchronous Javascript and XML. Useiden Web-kehitysteknologioiden käyttöä muodostettaessa entistä tehokkaampaa ja vuorovaikutteisempaa internetsivustoa.
ADT	Lyhenne sanoista Android Developer Tools. Lisäosa Eclipse-ohjelmistoympäristöön, joka sisältää työkalut Android-sovelluskehitystä varten.
Android	Googlen käyttöjärjestelmä mobiililaitteille.
Android SDK	Sisältää API-kirjastot ja kehitystyökalut Androidille kehittämistä varten. Sisältää ADT:n.
CSS	Lyhenne sanoista Cascading Style Sheets. CSS on WWW-dokumenteille kehitetty tyyliohjeiden laji.
Dalvik	Virtuaalikone, jonka avulla Android-sovellukset ajetaan.
Eclipse	Ohjelmointiympäristö, jota käytetään mm. Android-sovelluskehityksessä.
GPLv2	Vapaiden ohjelmistojen julkaisemiseen tarkoitettu lisenssi, joka antaa kenelle tahansa oikeuden käyttää, kopioida, muuttaa ja jakaa edelleen ohjelmia ja niiden lähdekoodia
HTML	Hypertext Markup Language on avoimesti standardoitu kuvauskieli, jolla voidaan kuvata hyperlinkkejä sisältävää tekstiä eli hypertekstiä.
Java	Sun Microsystemsin kehittämä järjestelmäriippumaton oliopohjainen



ohjelmointikieli.

JavaScript	Web-ympäristössä käytettävä komentosarjakieli. Mahdollistaa dynaamisen toiminnallisuuden lisäämisen internetsivuille.
jQuery	JavaScript-kirjasto, joka tekee JavaScript-komentosarjojen kirjoittamisesta helpompaa.
PHP	PHP: Hypertext Preprocessor on ohjelmointikieli, jota käytetään erityisesti Web-palvelinympäristöissä dynaamisten Web-sivujen luonnissa.
phpMyAdmin	Selaimen kautta käytettävä alustariippumaton tietokannan hallintatyökalu.
SQLite	Relaatiotietokantajärjestelmä, jossa koko tietokanta tallennetaan yleensä yhteen tiedostoon, joten se soveltuu erityisesti sovellusohjelmiin.
W3C	WWW-standardien kehittäjä ja ylläpitäjä
XML	XML (Extensible Markup Language). Sitä käytetään yleisesti tiedon siirtoon.

# 1 JOHDANTO

Tein opinnäytetyöni Softeri Oy:ssä, joka on myös työn tilaaja. Sovelluksen idea syntyi osana asiakasprojektia, johon sovellus liitetään. Myöhemmin sovelluksesta tulee osa Softeri Oy:n tuoteperhettä.

Opinnäytetyön aiheena oli suunnitella ja toteuttaa Android-sovellus, jonka avulla käyttäjä voi tallentaa muistioita ja liittää niitä erilaisiin kokonaisuuksiin. Sovelluksen käytön tulee olla mahdollisimman yksinkertaista sekä vanhojen muistioiden selauksen sujuvaa.

Sovelluksella käyttäjä tallentaa kuva-, video-, ääni- ja PDF-muistioita missä ja milloin vain. Muistiot tallentuvat pilvipalveluun, josta ne voidaan ladata koska tahansa.

Opinnäytetyö koostuu kahdesta suuremmasta osasta. Ensimmäinen osa on Web-sovellus, jossa käyttäjä voi hallita ja selata puhelimestansa pilvipalveluun lähetettyjä muistioita. Web-sovellus mahdollistaa myös uusien muistioiden luonnin. Rajapinnat, joita käytetään muistioiden lataamiseen ja listaamiseen Android-sovelluksessa, löytyvät myös Web-sovelluksesta. Toinen osa on Android-sovellus, jossa muistioiden tallennus suoritetaan.

Ohjelmasta julkaistaan versio, joka on kenen tahansa vapaasti ladattavissa ja se voidaan myös liittää helposti Softerin tarjoamaan Tribuni-toiminnanohjausjärjestelmään.

## **2 ANDROID-SOVELLUSKEHITYS**

Android on mobiililaitteille suunniteltu ohjelmistopino, joka sisältää käyttöjärjestelmän, väliohjelmistoja ja käyttäjän perusohjelmia. Androidissa käytetään avoimen lähdekoodin GPLv2-lisensioitua Linux-käyttöjärjestelmäydintä. Ohjelmakoodi Androidille kirjoitetaan Java-ohjelmointikielellä ja se käyttää Googlen kehittämiä Java-kirjastoja. Java-ohjelmat Androidilla ajetaan Dalvik-virtuaali-koneella. (1.)

### **2.1 Java**

Java on Sun Microsystemsin kehittämä, nykyään Oraclen omistama laaja teknologiaperhe ja ohjelmistoalusta, johon kuuluu muun muassa laitteistoriippumaton oliopohjainen ohjelmointikieli sekä ajoaikainen ympäristö virtuaalikoneineen ja luokkakirjastoineen. Java-alusta on käytössä noin 3,8 miljardissa laitteessa matkapuhelimista supertietokoneisiin. (2; 3.) Tällä hetkellä Androidille ei voi kirjoittaa alkuperäisohjelmia muilla ohjelmointikielillä kuin Javalla.

### **2.2 Android SDK**

Android SDK on ohjelmistokehityspaketti, jonka avulla voidaan tehdä sovellusohjelmia Android-käyttöjärjestelmälle. Paketti sisältää paljon esimerkkisovelluksia, kehitystyökalut, emulaattorin testausta varten ja tarvittavat kirjastot, jotta voidaan rakentaa Android-sovellus. (4.)

### **2.3 Ohjelmointiympäristöt**

Androidille kehittämistä varten on toistaiseksi hyvin rajoitettu määrä toimivia ohjelmointiympäristöjä. Tutuin ja toistaiseksi paras vaihtoehto Android-sovelluksia kehittäville on Eclipse, mutta Googlen tarjoama Android Studio on varmasti tulossa rinnalle ja vielä ohitse.

## **Eclipse**

Eclipse on yleisesti tunnettu ohjelmointiympäristönä Java ohjelmoijille. Eclipsen vahvuus on se, että sitä voidaan laajentaa erilaisilla lisäosilla, esimerkkinä Android-työkaluilla.

Android-kehitystyökalulisäosa (ADT) Eclipselle tarjoaa kaikki tarvittavat työkalut Android-sovellusten kehittämiseen. ADT:n avulla voidaan ajaa, testata, etsiä virheitä ja rakentaa latauspaketti Android-sovelluksesta.

## **Android Studio**

Android Studio on uusi kehitystyökalu Androidille ja se pohjautuu IntelliJ IDEA -kehitystyökaluun, jonka vuoksi se on hyvin kevyt ja nopea (5). Sovellus on vielä keskeneräinen ja saattaa sisältää paljon virheitä, jotka haittaavat sovelluskehitystä. Ohjelmasta on julkaistu versio 0.5.2, josta puuttuu vielä paljon ominaisuuksia, jotka esimerkiksi Eclipse jo tarjoaa.

Android Studion puolesta kuitenkin puhuu seikka, että Googlen omat Android-kehittäjät kehittävät sovelluksensa Android Studiolla, joten on vain ajan kysymys, milloin Android Studio kukistaa Eclipsen toiminnallisuudellaan (5). Toiseksi Google saattaa lakata päivittämistä Eclipsen ADT-lisäosaa.

## 3 KEHITYSTYÖSSÄ KÄYTETYT TEKNIIKAT

### 3.1 PHP-ohjelmointi

PHP on avoimen lähdekoodin Web-palvelinympäristön ohjelmointikieli. Se on tehokas työkalu dynaamisten ja vuorovaikutteisten internetsivustojen tekoon yhdistettynä HTML:n kanssa. PHP on laajalti käytetty, ilmainen ja tehokas vaihtoehto verrattuna kilpailijoihinsa, kuten esimerkiksi Microsoftin ASP:hen. (7.)

Ohjelmointikielen lisäksi PHP-ympäristössä on laaja luokkakirjasto. Ohjelmointikielenä se on komentosarjakieli, jossa ohjelmakoodi tulkitaan vasta ohjelman suoritusvaiheessa. (7; 8.) PHP:n suosiosta kertoo se, että sitä hyödyntävät muun muassa Wikipedia ja Facebook

#### Syntaksi

PHP:n upottaminen HTML:ään on todella yksinkertaista (kuva 1). PHP:n käyttö edellyttää `<?php`-merkin ohjelmakoodin alkuun ja loppuun `?>`-merkin. Jos kuitenkin halutaan käyttää PHP:tä ainoastaan tekstin tulostamiseen, voidaan käyttää aloitusmerkinä `<?=`, jolloin **echo**-tulostuskomentoa ei tarvita. Kaikki ohjelmakoodi sisällytetään näiden merkkien sisään. Jos tiedosto sisältää ainoastaan PHP-ohjelmointikieltä, on suositeltavaa, että dokumentti sisältää aloitus- ja lopetusmerkin vain kerran. (7.)

PHP:ssä muuttujaa ei tarvitse alustaa, eikä sille tarvitse määrittää tyyppiä. Muuttujien eteen merkataan aina \$-merkki. Esimerkiksi \$value voi olla numeroarvo tai tekstiä ja tyyppin määrittää se, mitä siihen asetetaan.

```
<!DOCTYPE html>
<html>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
    <br/>
    <?= '<b>Hello again</b>' ?>
  </body>
</html>
```

*KUVA 1. PHP:n käyttö HTML-dokumentissa. Koodi tulostaa p-tagiin tekstin "Hello World". Tämän alapuolelle tulostetaan teksti "Hello again" lihavoituna.*

## 3.2 HTML5

HTML5 voi tarkoittaa kolmea eri asiaa, jotka kytkeytyvät toisiinsa:

- uusinta versiota HTML-merkintäkielelle (katso 4.1 HTML)
- yleisnimitystä nykyaikaisille Web-tekniikoille
- sovellusten toteuttamista Web-ohjelmoinnin tekniikoilla (JavaScript, CSS, HTML) (9).

HTML5 sisältää paljon uusia ominaisuuksia, jotka eivät tarjoa ainoastaan runsasta mediatukea, vaan myös tuen tehdä entistä tehokkaampia ja vuorovaikutteisempia Web-sovelluksia (9).

HTML5 on jatkuvan kehityksen alla ja se saa uusia ominaisuuksia alinomaan. Tämän vuoksi eri selaimilla, kuten Firefox, Safari ym., on täysi työ vastata näiden ominaisuuksien tuesta. (10; 11.)

### HTML

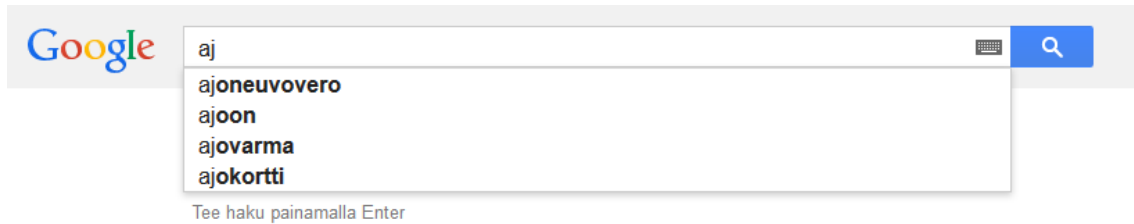
HTML on WWW-sivun kuvauskieli. HTML on ensimmäinen asia, joka tulee kotisivun tekoa opiskelevan eteen, mutta kokeneemman ohjelmoijan käsissä se muuttuu vain lopulliseksi muodoksi, mihin tieto esimerkiksi tietokannoista siirretään (12).

### Ajax

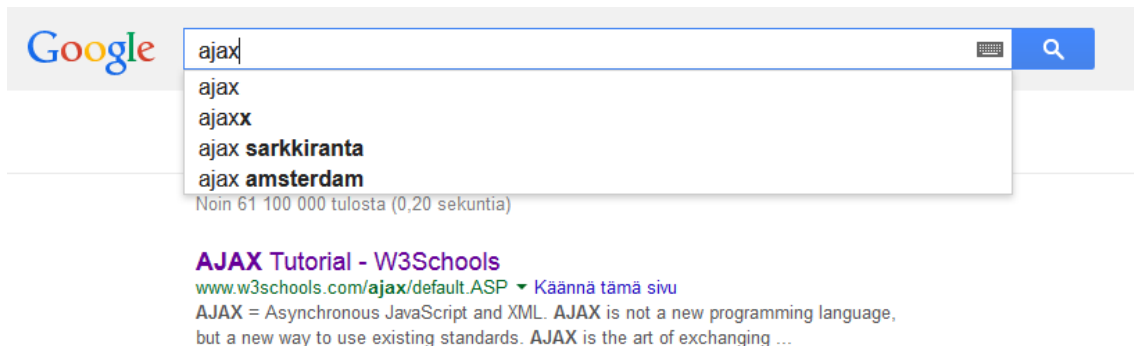
Ajax tarkoittaa useiden Web-kehitysteknologioiden käyttöä muodostettaessa entistä tehokkaampaa ja vuorovaikutteisempaa internetsivustoa. Ajax ei ole itsessään oma teknologia vaan termi, joka kuvaa lähestymistapaa joka yhdistää useita eri Web-kehitystekniikoita: HTML, CCS, JavaScript, DOM, XML, XSLT ja XMLHttpRequest. (13.)

Ajaxin peruseriaate on, että koko internetsivua ei tarvitse päivittää, vaikka käyttäjä tekisi jonkin muutoksen (13). Hyvä esimerkki tästä on Google hakukoneen hakukenttä. Kun käyttäjä alkaa kirjoittamaan hakukenttään hakusanaansa, Google ehdottaa käyttäjälle listan hakusanoja (kuva 2). Hakutulos päivittyy käyttäjälle sitä mukaa, kun hakusanaa kirjoitetaan (kuva 3).

Kaikki tämä tapahtuu taustalla eikä siitä aiheudu käyttäjälle turhaa odottelua ja latailua.



*KUVA 2. Googlen hakukenttään kirjoitettaessa Google ehdottaa yleisesti haettuja hakusanoja.*



*KUVA 3. Kun hakukenttään on kirjoitettuna kokonainen sana, päivittyy hakutulos kentän alapuolelle.*

## JavaScript

JavaScript on yleinen ohjelmointikieli, jota voidaan sisällyttää verkkosivuille niiden toiminnallisuuden lisäämiseksi. JavaScriptillä saadaan internetsivut eloon lisäämällä sivuille vuorovaikutteisuutta. Sen avulla voidaan esimerkiksi lisätä dynaamista toiminnallisuutta sivuille. Ohjelmia, joita JavaScriptillä tehdään, kutsutaan nimellä Scripts, joka tarkoittaa komentosarjoja. (14; 15.) Kirjoitettu komentosarja ei tarvitse kääntämistä. Kun se on kirjoitettu, se on valmis käytettäväksi. JavaScriptillä voidaan tehdä mm. seuraavia asioita:

- muokata HTML-sivua, kirjoittaa tekstiä sille, lisätä tai poistaa tageja, muokata elementtien tyylejä
- suorittaa komentosarjoja tapahtumista: hiiren klikkauksesta ja liikkeestä ym.

- käyttää Ajax-komentoja
- asettaa ja vastaanottaa evästeitä (14).

Huonoja puoliakin JavaScriptistä löytyy. Esimerkkinä jos käyttäjällä on auki useampi välilehti, jossa käytetään JavaScriptiä, voi toinen JavaScript tiedosto vaikuttaa toiseen välilehteen. Ongelma esiintyy erityisesti silloin, jos välilehdet toimivat saman verkkotunnuksen alla. Palvelun tarjoajan kuitenkin pitää ottaa tällaiset asiat huomioon komentosarjoissaan. Kolme tärkeintä ominaisuutta JavaScriptissä ovat

1. täysi integraatio HTML:n ja CSS:n kanssa
2. yksinkertainen toteuttaminen
3. selainten kattava tuki. (15.)

Kuvassa 4 on yksinkertainen esimerkki, kuinka JavaScriptillä voidaan asettaa tekstiä HTML-dokumentissa löytyvään elementtiin.

```
document.getElementById('elementti1').value = 'Terve';
```

*KUVA 4. HTML-dokumentissa olevaan elementtiin nimeltä elementti1 asetetaan tekstiksi "Terve".*

## **jQuery**

jQuery on ilmainen avoimen lähdekoodin JavaScript-kirjasto. Sen avulla HTML-dokumentin muokkaaminen, tapahtumakäsittely, animaatiot ja Ajax-kutsujen käyttö on paljon yksikertaisempaa ja helppokäyttöisempää. Sen tarkoitus on siis helpottaa JavaScriptin käyttöä. jQuery voidaan ladata ilmaiseksi jQueryn kotisivuilta. jQuery on erillinen JavaScript-tiedosto ja voidaan ottaa käyttöön omilla sivuilla linkittämällä sivu paikalliseen jQuery-tiedostoon tai johonkin julkisen palvelimen kopioon. Tänä päivänä jQuery on maailman suosituin JavaScript-kirjasto. (16; 17.)

Kuvassa 5 nähdään kuinka jQueryllä ajetaan sama tulostus kuin kuvassa 4.

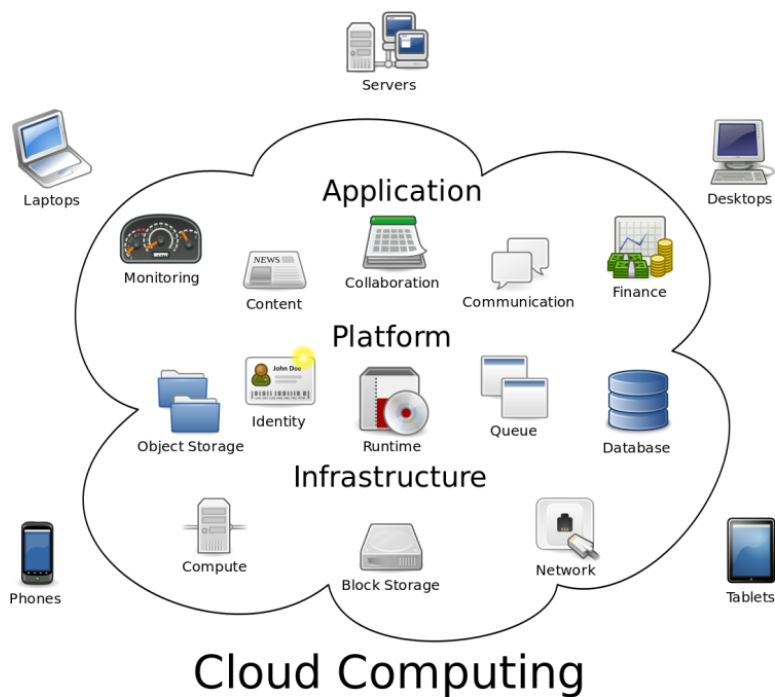
```
$("#elementti1").val("Terve");
```

*KUVA 5. jQueryllä toteutettu sama tulostus kuin kuvassa 4*



### 3.3 Pilvipalvelut

Pilvipalvelu tarkoittaa palvelun tarjoamista pilvessä eli ulkopuolisella palvelimella. Käyttäjä ei kuitenkaan voi nähdä eikä hallita palvelun teknisiä yksityiskohtia. Palvelu voi olla virtuaalipalvelin, talletusalusta, tietokanta, sovellus tai ohjelma. Pilveen voi olla yhteydessä useammasta laitteesta yhtäaikaaisesti. Pilvipalveluiden avulla käyttäjän ei tarvitse suorittaa sovellusta omalta koneelta, sillä se pyörii palveluntarjoajan pilvessä. Pilvipalvelut ovat yleiskäyttöisiä palveluita, jonka vuoksi ne tarjoavat yleensä hyvin dokumentoidut rajapintakuvaukset, mikä taas auttaa ohjelmistokehittäjiä pysymään trendin mukana. Kuvassa 6 on esitetty kuinka käyttäjä näkee pilvipalvelun toiminnan.



(18.)

*KUVA 6. Kuvassa on esitetty pilvipalvelun yleiskatsaus käyttäjän näkökulmasta. Useita eri palveluilta voi olla tarjoajan pilvessä (19).*

Kaikkia pilvipalveluita yhdistää tekijä, että palvelun avaaminen tai siihen liittyminen vie käyttäjältä vain hetken. Palveluihin liittyminen sekä niiden käyttö on yleensä ilmaista, mutta vain tiettyyn rajaan asti. Esimerkiksi Dropbox-tiedostojen synkronointisovelluksen tallennustila on rajallinen, mutta käyttäjä voi maksua vastaan saada tililleen lisää tallennustilaa.

### 3.3.1 Käyttöönottomallit

Käyttöönottomalli kertoo, missä palvelun tarjoajan laitteet sijaitsevat ja kuka on niiden ylläpitäjä. Pilvipalveluista on erilaisia käyttöönottomalleja vastaamaan käyttäjän tarpeita. Neljä yleisintä mallia ovat

1. yksityinen pilvi
2. julkinen pilvi
3. yhteisöllinen pilvi
4. hybridipilvi, joka on yhdistelmä muista malleista.

#### **Yksityinen pilvi**

Yksityisessä pilvessä kaikkea operoi yksi ja sama organisaatio, joka vastaa järjestelmien ja laitteiston ylläpidosta. Yhteys pilveen muodostetaan lähiverkon tai laajaverkon kautta. Oikein tehtynä yksityinen pilvi voi kehittää yrityksen liiketoimintaa, mutta pahimmillaan se voi olla suuri tietoturvariski yritykselle, sillä tiedon fyysinen sijainti löytyy yhdestä osoitteesta. (20.)

#### **Julkinen pilvi**

Julkisen ja yksityisen pilven arkkitehtuurissa ei ole juurikaan eroa. Julkisessa pilvessä kaikki fyysiset laitteet löytyvät palveluntarjoajan tiloista. Palveluntarjoaja vastaa myös laitteiston ylläpidosta. Yhteys pilveen muodostetaan internetin välityksellä. (21.)

#### **Yhteisöllinen pilvi**

Yhteisöllinen pilvi jaetaan yhteisön kesken. Yhteisöllinen pilvi tarjoaa turvallisemman tavan säilöä tietoa verrattuna julkiseen pilveen. Laitteiston huollosta ja ylläpidosta vastaavat yhteisön jäsenet tasapuolisesti. (22.)

#### **Hybridipilvi**

Hybridipilvi on yhdistelmä kahdesta tai useammasta eri käyttöönottomallista. Käyttöönottomallit eivät sulaudu toisiinsa vaan tuovat omat vahvuutensa pilveen. Tämän vuoksi pilvi on entistä joustavampi käyttäjänsä tarpeille. (23.)

### 3.3.2 Palvelumallit

Pilvipalveluiden tarjoajat tarjoavat palveluitansa yleensä muutaman perusmallin mukaisesti:

- infrastructure as a service (IaaS)
- platform as a service (PaaS)
- software as a service (SaaS) (24; 25).

Näiden mallien avulla käyttäjä pääsee käsiksi palveluntarjoajan palveluihin. (Kuva 7.)

#### **SaaS**

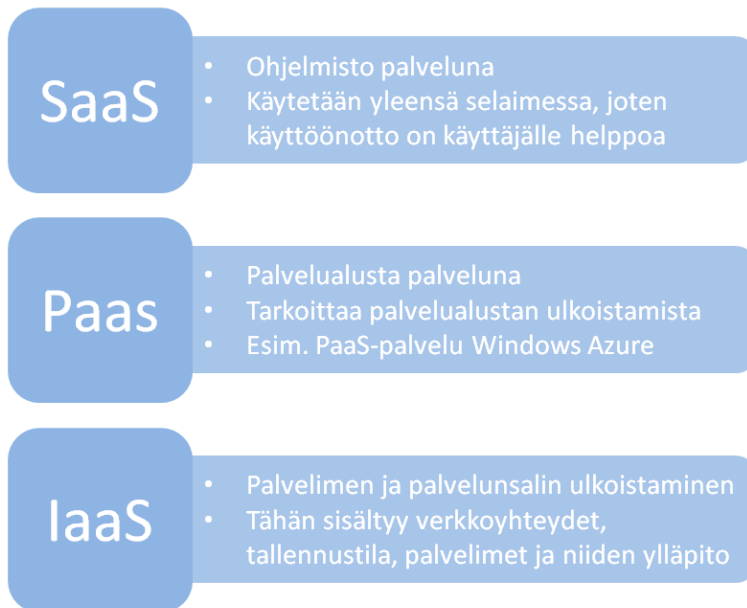
SaaS tarkoittaa ohjelmiston hankkimista palveluna. Yleensä käytöstä maksetaan sen käytön laajuuden mukaan. Tuotantoympäristö on sama kaikille palvelun asiakkaille. Ohjelmisto toimii yleensä Internet-selaimella, minkä vuoksi sen käyttöönotto on käyttäjälle helppoa. (26.)

#### **PaaS**

PaaS tarkoittaa palvelualustan ulkoistamista. PaaS:llä kehittäjät voivat ladata omia sovelluksiaan osaksi kokonaisuutta. (27.)

#### **IaaS**

IaaS tarkoittaa palvelimen ja palvelinsalien ulkoistamista. Tähän sisältyy kaikki, mitä palvelun pystyttäminen tarvitsee: verkkoyhteydet, tallennustilan, palvelimet ja niiden ylläpidon. (28.)



*KUVA 7. Pilvipalvelun palvelumallit ja niiden yleispiirteet (26; 27; 28)*

### **3.4 Web-rajapinnat**

Web-rajapinnat eli Web Services on W3C:n mukaan ohjelmistojärjestelmä, joka mahdollistaa elektronisten laitteiden vuorovaikutuksen tietoverkon yli.

Rajapinnat ovat palveluita, jotka ovat avoimia muillekin kuin palvelun tarjoajalle. Yleisin tarkoitus rajapinnoilla on tarjota dataa yritysten välillä, sovelluksille ja peruskäyttäjille. Vaikka rajapinnat tarjoavat tietoa, se ei tarkoita sitä, että niiden käyttäjä näkisi miten ne toimivat ja mitä niiden taustalla pyörii. (29.)

Yritysten järjestelmät on voitu tehdä eri ohjelmointikielillä, jolloin tarvitaan yhteinen tapa lähettää tietoa toistensa järjestelmiin. Yleensä tieto siirtyy XML-tiedostoina.

Rajapinnat koostuvat kolmesta osasta: palvelun tarjoajasta, palvelun käyttäjästä ja palveluhakemistosta. Nämä tekijät kommunikoivat keskenään eri XML-pohjaisten protokollien avulla. (30; 31.)

#### **Rajapinnan kuvaus**

Rajapintojen tarjoajilla tulee olla kattavat rajapintakuvaukset, jotta niitä pystyisi käyttämään sujuvasti. Niistä tulee löytyä ainakin seuraavat asiat:

- Kuinka palvelun käyttäjä saa palvelun tarjoajalta haluamansa tiedon?
- Mitkä parametrit ovat pakollisia tiedon saamiseksi?
- Minkälainen rakenne lähetettävässä tiedostossa tulee olla (yleensä XML-tiedosto)?
- Mitä mikäkin palautettu virhe tarkoittaa? Tämän avulla käyttäjä voi paikantaa helposti, mistä oma virhe löytyy. (30.)

### 3.5 Käyttöliittymät

#### 3.5.1 Web-sovelluksen käyttöliittymät

Web-sovellus rakennetaan toimeksiantajan valmistaman toiminnanohjausjärjestelmän ohjelmistoalustan päälle. Web-sovelluksen käyttöliittymät toteutetaan HTML5-tekniikoin eli käyttäen HTML:ää, CSS:ää ja JavaScriptiä. Käyttöliittymän taustalla toiminnot toteutetaan PHP:llä. Aloituskäytönä on kirjautumisikkuna, jossa kysytään käyttäjätunnus ja salasana. Sisäänkirjautumisen jälkeen käyttäjä ohjataan etusivulle, jossa käyttäjä näkee ilmoituksia, joita voivat lisätä vain järjestelmän pääkäyttäjät.

#### **Muistiot**

Muistioita käyttäjä pääsee hallitsemaan muistioiden hallinnasta. Muistioiden hallinnassa ensimmäinen näkymä käyttäjälle on lista hänen muistioistaan. Näkymässä on hakukenttä aiheilla hakua varten sekä hakusanalla hakua varten. Näkymästä tulee olla nopeaa ja helppoa luoda uusi muistio valitsemalla uuden muistion tyyppi. Muistiolistasta yksittäinen muistio voidaan poistaa, muokata tai esikatsella ja sen tiedot voidaan avata näkyville.

Kun lisätään uutta muistiota, aukeaa näkyviin muistiolomake, jossa täytetään muistiota koskevat tiedot. Tietoja ovat aihe, otsikko, teksti (lyhyt kuvaus) ja tiedosto. Sama lomake on käytössä, jos muistiota muokataan.

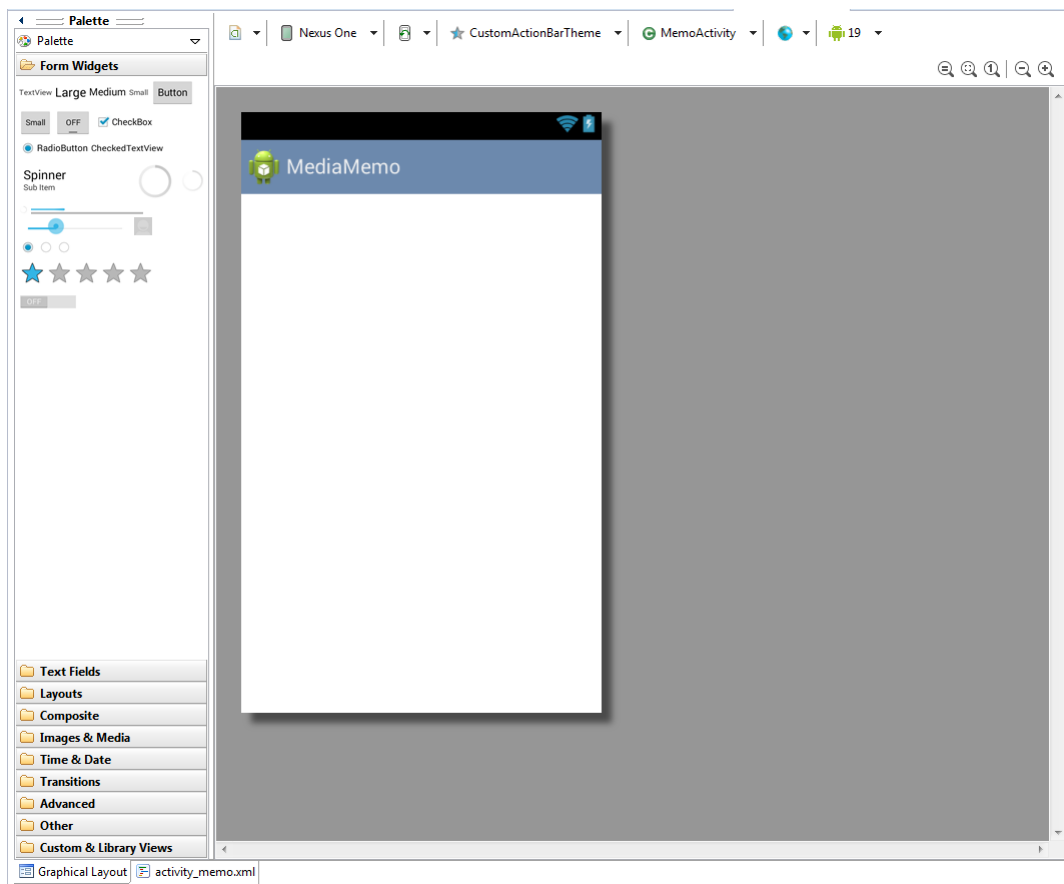
Muistion tietoja tarkasteltaessa aukeaa näkymä, jossa on listattu kaikki tiedot muistiosta. Sivulla näkyy myös esikatselukuva. Näkymästä pitää olla helppo päästä muokkaamaan tarkasteltavaa muistiota sekä palaamaan muistion selaukseen.

## Ylläpito

Ylläpidon ominaisuuksiin ja näkymiin pääsevät ainoastaan pääkäyttäjätason käyttäjät. Ohjelmistoalusta tarjoaa ylläpidon ominaisuudet ja näkymät. Ylläpidon ominaisuudet on jaoteltu välilehtiin. Ylläpito-osio on osa toimeksiantajan tuotetta, joten tarkempi kuvaus jätetään pois.

### 3.5.2 Android-sovelluksen käyttöliittymät

Android-sovelluksen näkymät rakennetaan Eclipsessä ADT:n mukana tulevalla graafisella suunnittelutyökalulla (kuva 8).



*KUVA 8. ADT-paketin mukana tuleva näkymien suunnittelutyökalu*

Sovellukseen tulee kolme erilaista päänäkymää: muistiolista-, hallintavalikko- ja muistiolomakenäkymä. Näkymät sisältävät paljon toiminnallisuutta, joten osa toiminnoista löytyy painikkeen takaa. Näkymät suunnitellaan niin, että niiden toiminnallisuuksia on helppo käyttää ja toimintamalli on entuudestaan tuttu Android-käyttöjärjestelmää käyttäville.

## Muistioliista

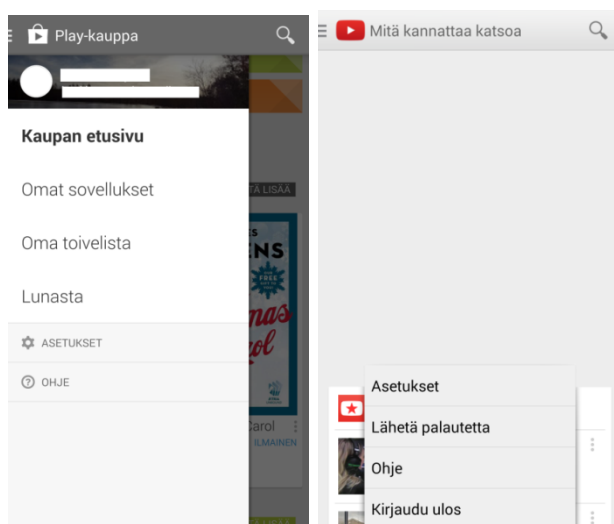
Ensimmäinen näkymä, joka tulee näkyviin käyttäjälle, on lista hänen lisäämistään muistiosta. Muistioita pitää voida suodattaa aiheella tässä näkymässä. Listanäkymästä pitää olla helppo päästä lisäämään uusia muistiota valitsemalla muistiotyyppi.

Muistioliistan yksittäistä muistiota voidaan

- esikatsella
- avata
- poistaa
- muokata
- tarkastella ominaisuudet-näkymässä.

## Hallintavalikko

Hallintavalikko on näkymä, joka on käyttäjältä piilossa, mutta sen saa vedettyä näkyviin pyyhkäisemällä näytön vasemmasta reunasta oikealle. Tällaiset valikot ovat yleistyneet Android sovelluksissa viime aikoina, sillä niihin saa huomattavasti enemmän toiminnallisuutta kuin esimerkiksi normaalin valikkonapin taakse asetettuun valikkoon (kuva 9).



*KUVA 9. Vasemmalla on kuva hallintavalikosta (Googlen Play Kauppa -sovellus). Oikealla kuva tavallisesta valikosta, joka aukeaa Androidin menu-painikkeesta (Googlen Youtube-sovellus).*

Valikkosta voidaan luoda uusi muistio samaan tapaan kuin aloitusnäkömäästäkin. Valikkosta löytyy myös lista viimeksi käytetyistä aiheista, joihin muistioita on liitetty.

### **Muistiolomake**

Muistiolomake on näkymä, jossa tallennetaan yksittäinen muistio. Muistiolle voidaan tässä näkymässä syöttää otsikko ja teksti. Näkymä sisältää myös tiedoston esikatselukuvakkeen ja tallennuspainikkeen.



## 4 TOTEUTUSRATKAISUT

### 4.1 Web-sovellus

Projektin tekeminen aloitettiin Web-sovelluksen tekemisestä. Se selkeyttäisi ohjelman toimintatapaa ja rajapinnat, joita Android-sovellus käyttää, tulevat Web-sovelluksesta. Myös Android-sovelluksessa käytettävät ohjelma-asetukset asetetaan Web-sovelluksessa.

Web-sovellus rakennettiin toimeksiantajan ohjelmistoalustan päälle, jonka käyttö oli minulle tuttua jo entuudestaan. Ensimmäisenä siivosin ohjelmistoalustan runkoa poistamalla turhat moduulit ja tietokannasta turhat taulut. Sitten pystyitin ohjelmaan muistioiden hallitsemista varten oman moduulin. Moduulin pystyttämisen vaiheista tässä dokumentissa ei voida mainita enempää.

#### 4.1.1 Näkymät

##### Muistioiden selaus

Web-sovelluksen muistioidenhallintanäkymistä ensimmäisenä rakennettiin selausnäkyvä. Selausnäkyvässä käyttäjä voi hakea lisäämiänsä muistiota aiheen tai hakusanan mukaan. Hakusana ja aihe lähetetään Ajax-kutsuna kun kentiin kirjataan jotain (kuva 10). Ajax-kutsu palauttaa näkymän muistioista, jotka vastaavat hakuehtoihin. Hakutulos muodostetaan PHP-tiedostossa, joka suorittaa tietokantahaun ja tulostaa hakutuloksen HTML-tiedostoon. Aihetta haettaessa ohjelma ehdottaa kannasta löytyviä aiheita. Samalla Ajax-kutsu hakee muistioita, jotka sopivat aiheeseen. Tämä ominaisuus on toteutettu jQueryn autocomplete-funktiota käyttäen (kuva 11).

```
$("#hakusana").keyup(function() {  
    haeTiedot(BASE_URI + "muistiot/haemuistiotajax", 200);  
});
```

*KUVA 10. jQueryä käyttäen toteutettu komentosarja, joka lähettää Ajax-kutsun joka kerta, kun hakusana-kenttään kirjoitetaan jotain.*

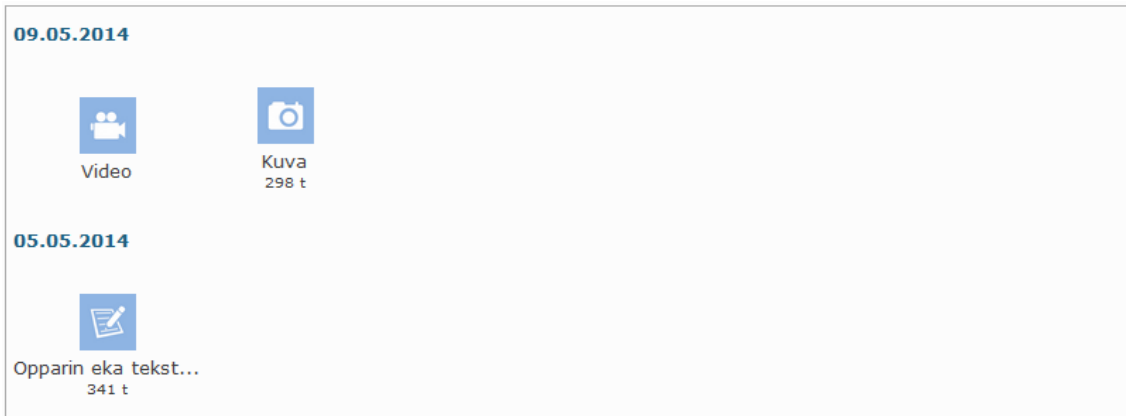


*KUVA 11. Autocomplete-funktio näyttää Ajax-kutsulla palautettujen ”op” merkkeihin täsmäävät aiheet, jotka haetaan tietokannasta.*

Muistiot listataan hakutuloksena hakukenttien alapuolelle. Muistiot näytetään siten, että uusin on ensimmäisenä ja samana päivänä lisätyt muistiot niputetaan päivämääräotsikon alle (kuva 12). Jos muistioista muodostuu neljä riviä tai enemmän, jaetaan hakutulos useammalle sivulle. Klikattaessa muistiota hiiren oikealla painikkeella aukeaa näkyviin valikko, josta muistio voidaan

- avata esikatseltavaksi
- avata uudelle välilehdelle tarkkailtavaksi
- poistaa
- muokata
- avata ominaisuudet-ikkunaan.

Muistion tiedot -välilehti aukeaa, jos muistiota klikataan hiiren vasemmalla painikkeella. Muistioita voidaan myös poistaa raahaamalla muistio näkymän yläreunassa olevaan roskakorikuvakkeeseen. Ominaisuus on toteutettu JavaScriptillä käyttäen jQuerya. Ennen poistoa ohjelma varmistaa käyttäjältä, haluaako hän varmasti poistaa muistion.



*KUVA 12. Hakutulokset, jossa muistiot on listattu lisäyspäivämäärän alle.*

### **Muistion tiedot**

Muistion tiedot -näkyvässä käyttäjälle avautuu tiedot hänen valitsemastaan muistiosta. Muistion tiedot haetaan tietokannasta ja siitä luodaan muistio-olio. Tietoja ovat otsikko, teksti, muistion tyyppi, tiedoston nimi, tiedoston koko, lisäyspäivämäärä ja esikatselukuva. Esikatselukuva näkyy toistaiseksi ainoastaan kuva- ja tekstimuistiolle. Näkymän oikeasta yläreunasta käyttäjä pääsee muokkaamaan muistion tietoja kynäkuvakkeesta.

### **Muistiolomake**

Muistiolomake aukeaa muistioiden selausnäkyvästä, josta valitaan muistiotyyppi. Ohjelmassa lomake aukeaa uudelle välilehdelle. Muistiosta luodaan uusi olio ja siitä tallennetaan väliaikainen rivi muistiotauluun. Mikäli lomake peruutetaan, poistetaan muistorivi tietokannasta. Lomakkeella on kentät muistion aiheelle, otsikolle, tekstille ja tiedoston valinnalle. Mikäli muistiota muokataan, avataan sama lomake. Muistiosta luodaan olio, jonka tiedoilla lomake täytetään. Tiedot päivittyvät muistiolle tallennuspainiketta painamalla. Jos taas lomake peruutetaan, muistio pysyy muuttumattomana.

#### **4.1.2 Ohjelma-asetukset**

Ohjelma-asetukset on ohjelmistoalustan tarjoama ominaisuus, jossa voidaan tallentaa esimerkiksi muistiot-moduulin erilaisia arvoja. Näitä asetuksia voidaan käyttää missä vain ohjelman sisällä. Asetukset lähetetään myös Android-

sovellukselle käyttäen ohjelman rajapintoja. Ohjelma-asetuksissa määritellään muistioille taulukon 1 mukaiset asiat.

*TAULUKKO 1. Ohjelman ohjelma-asetukset ja niiden selitykset*

Nimi	Vaihtoehdot	Kuvaus
Liitosvaihtoehdot	1 = Aihe 2 = Työ 3 = Huolto 4 = Työmaa 5 = Kauppa 6 = Asiakas	Määrittää, mihin kokonaisuuteen muistion voi liittää. Aihevaihtoehtoa käytetään opinnäytetyössäni. Muut vaihtoehdot ovat valittavissa sovelluksen asiakasversiossa.
Muistion maksimikoko	Numero	Tähän tekstikenttään syötetään muistion maksimikoko tavuina. Esimerkiksi 1000000, joka on 1 Mt.
Tuettavat tiedostotyytit	1 = Teksti 2 = Kuva 3 = Video 4 = Ääni 5 = Pdf	Listasta valitaan yksi tai useampi tiedostotyyppi. Asetus määrittää, mitä tyyppisiä käyttäjillä on käytössä. Ohjelman näkymät muokkautuvat asetuksen mukaan.
Omat muistiot	Epätosi/Tosi	Asetus määrittää, näytetäänkö haussa vain omat muistiot. Opinnäytetyössä asetus on "Tosi".
Omat aiheet	Epätosi/Tosi	Asetus määrittää, voidaanko käyttää vain itse lisättyjä aiheita. Opinnäytetyössä asetus on "Tosi"

#### **4.1.3 Muistion tallennus**

Muistiota on viittä erityyppistä ja niiden tallennus Web-sovelluksessa käsitellään samalla tavalla. Muistioiden tallennus suoritetaan muistiolomakkeelta, niin kuin aiemmin on käynyt selväksi. Käyttäjä täyttää lomakkeen tiedot ja valitsee liitettävän tiedoston käyttäjän laitteelta. Kun kaikki kentät on täytetty ja painettu tallennuspainiketta, lähettää muistiolomake tietonsa PHP-metodille, joka hoitaa muistion tallennuksen tietokantaan. Olion tiedot päivitetään lomakkeelta tulleiden tietojen perusteella. Muistion lisäysajankohdaksi tallennetaan tallennusajankohta. Tämän jälkeen muistio löytyy käyttäjän muistiolistasta.

#### 4.1.4 Tiedoston liittäminen

Kun muistion perustiedot (otsikko ja teksti) on tallennettu, voidaan muistioon liittää valittu tiedosto liitteeksi. Tiedostoja voidaan liittää ainoastaan yksi muistiota kohti. Tiedosto ladataan lomakkeelta löytyvästä input-kentästä, jonka tyyppi on file. Jotta HTML-lomakkeelta voidaan ottaa vastaan tiedostoja, täytyy sen form-merkkiin määrittää enctype="multipart/form-data". Tiedosto tallennetaan palvelimelle ja kantaan tallennetaan viittaus tähän tiedostoon. Kannan tiedosto-rivistä tallennetaan viittaus myös muistioon, jotta muistiolle löydetään sille kuuluva tiedosto.

Muistion tiedosto tallennetaan muistion tiedostoille määrättyyn kansioon. Lomakkeelta lähetetty tiedosto löytyy \$\_FILES-taulukosta, joka sisältää tiedostoa koskevat tiedot. Jotta tiedostoja voidaan tallentaa palvelimella olevaan kansioon, täytyy kansiolle antaa täydet kirjoitus- ja lukuoikeudet (chmod 777). Tallennettaessa ensiksi tarkastetaan, että tiedosto on asetettu lomakkeella. Jos tiedosto löytyy, se tallennetaan muistio-nimiseen kansioon palvelimella. Tallennuksen hoitaa uploadFiles-metodi, joka ottaa vastaan kansion nimen, jonne tiedosto tallennetaan. uploadFiles-metodi tallentaa kantaan tiedostoa koskevat tiedot ja viittaukset muistioon. Itse tiedosto ladataan palvelimelle käyttäen PHP:n move\_uploaded\_file-metodia, joka ottaa vastaan tiedoston nimen ja sijainnin. Näillä tiedoilla metodi tallentaa tiedoston palvelimelle. Myös itse tiedosto tarvitsee täydet kirjoitus- ja lukuoikeudet. Tiedoston tallennus on havainnollistettu kuvassa 13.

```
if (!empty($_FILES['muistio']['name'])) {  
    if (!$this->muistio->uploadFiles("muistio")) {  
        Message::set("Muistion tiedoston tallennus epäonnistui.", Message::ERROR);  
    }  
}
```

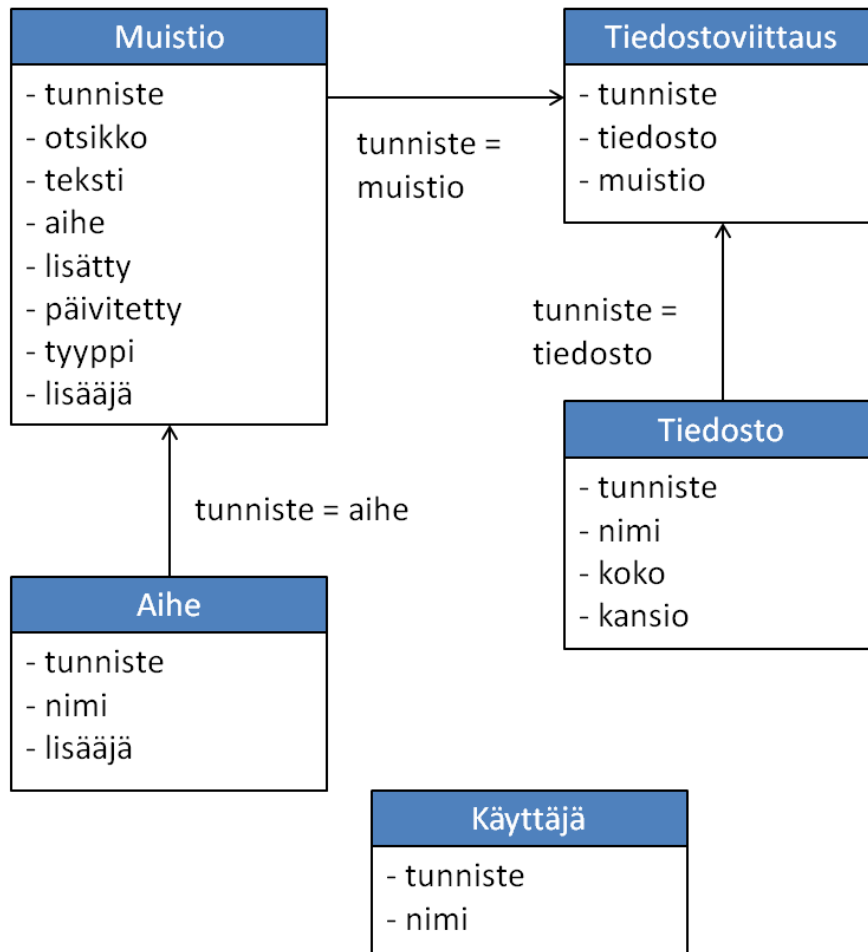
*KUVA 13. Tiedoston tallennus palvelimelle Web-sovelluksessa.*

#### 4.1.5 Tietokanta

Web-sovelluksen tietokanta on toteutettu phpMyAdmin työkalua käyttäen. Ohjelmistoalusta tarvitsee itsessään tietokannasta tauluja käyttöönsä. Tässä projektissa näiden lisäksi tarvitaan taulut

- aiheille
- muistioille
- tiedostoille
- tiedostojen viittaukselle
- käyttäjille.

Aiheista tallennetaan nimi ja aiheen lisääjä (käyttäjä). Muistioista tallennetaan otsikko, teksti, viittaus aiheeseen, lisäyspäivä, muistion lisääjä (käyttäjä) ja muistion tyyppi. Loput taulut liittyvät vahvasti ohjelmistoalustaan. Kuvassa 14 on hahmoteltu tietokannan rakennetta ja sidoksia. Käytetyt sarakkeiden nimet eivät vastaa todellisuutta, sekä rakenne on vain suuntaa antava.



KUVA 14. Hahmotelma ohjelman tietokantarakenteesta. Lisääjä-sarake tauluissa Aihe ja Muistio viittaavat Käyttäjä-taulun tunniste-sarakkeeseen.

#### 4.1.6 Rajapinnat

Rajapinnat ovat Android-sovelluksen käyttöä varten. Rajapinnat ovat samalla palvelimella kuin itse Web-sovellus. Taulukosta 2 käy ilmi, mitä eri rajapintoja Android sovellus käyttää ja mitä ne tekevät. Tiedostojen tarkkoja nimiä ei tässä dokumentissa voida ilmoittaa. Kaikkien rajapintojen käyttö vaatii todennuksen eli käyttäjätietojen lähetyksen.

TAULUKKO 2. Android-sovelluksessa käytettävät rajapinnat ja niiden selitykset.

Kuvaus	Selite
Kirjautuminen	Vastaanottaa käyttäjätunnuksen ja salasanan ja palauttaa kirjautumisen tuloksen.
Muistiot	Palauttaa käyttäjän tallentamat muistiot JSON-muodossa. Voidaan suodattaa aiheella sekä hakusanalla.
Aiheet	Palauttaa käyttäjän tallentamat aiheet JSON-muodossa. Tarvitsee hakusanan, jotta voidaan suodattaa hakua.
Viimeaikaiset aiheet	Palauttaa listan käyttäjän viimeksi käytetyistä aiheista.
Ohjelma-asetukset	Lataa Web-sovelluksen määritetyt ohjelma-asetukset.
Muistion tallennus	Vastaanottaa muistion tiedot ja tallentaa ne tietokantaan. Käytetään myös muistion muokkauksessa.
Muistion poisto	Vastaanottaa muistion tunnisteen, jonka avulla muistio voidaan poistaa tietokannasta.

## 4.2 Android-sovellus

Sovellus toimii Android 4.0-versiossa ja sitä uudemmissa. Vanhemmat versiot jätettiin tarkoituksella pois, jotta voidaan käyttää uusimpia Androidin ominaisuuksia. Vanhemmat Android-versiot eivät välttämättä tue kaikkia ominaisuuksia, joita Android 4.0 ja uudemmat tarjoavat.

### 4.2.1 Näkymät

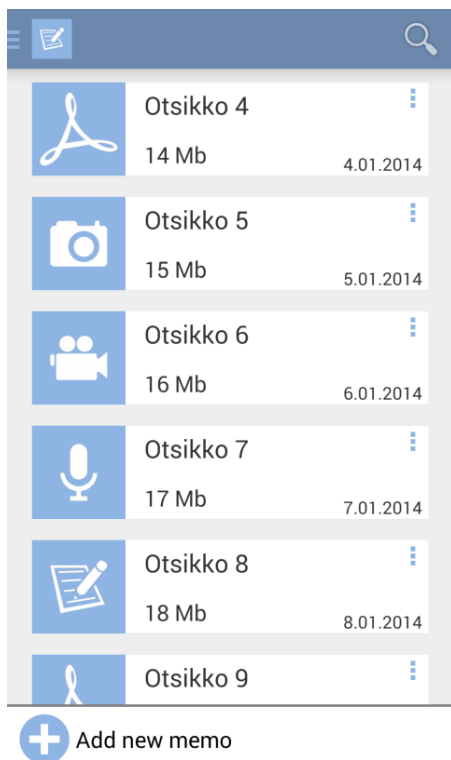
Sovelluksen tekeminen aloitettiin näkymien suunnittelusta. Näkymien tuli olla mahdollisimman yksinkertaisia ja kuitenkin sisältää tarvittavat toiminnallisuudet. Sovelluksen perusideana oli, että uuden muistion tallentaminen onnistuu vaivatta ja helposti.

#### Muistioiden selaus

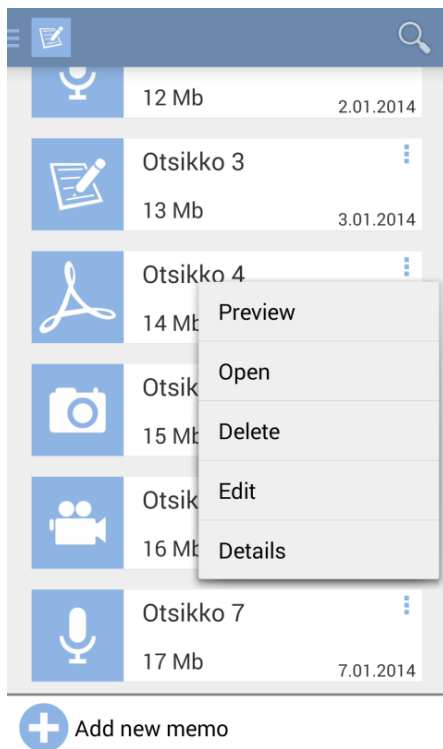
Ensimmäisenä näkymistä suunniteltiin tallennettujen muistioiden selausnäkyvän. Näkymä ei kuitenkaan ole pelkästään vanhojen muistioiden selausta varten, vaan siitä pääsee lisäämään uusia muistiotakin. Kun muistiot on ladattu rajapintojen kautta laitteelle, ne lisätään listaan (kuva 15), jossa yksittäisestä muistiosta näytetään otsikko, tiedoston koko, päiväys ja



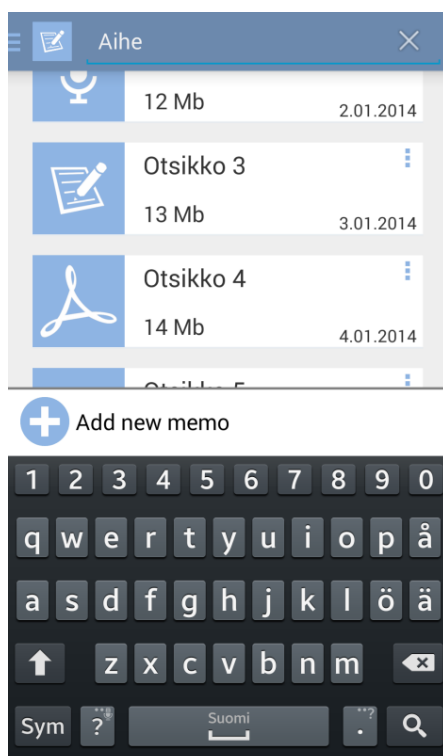
muistiotyyppi kuvana. Listassa olevasta muistiosta voidaan avata lisävalikko (kuva 16) jonka takaa löytyvät seuraavat toiminnot muistiolle: esikatselu, muistion avaus, muistion poisto, muistion muokkaus ja ominaisuuksien avaus. Muistiolistaa voidaan suodattaa näkymän toimintapalkista ruudun yläosassa (kuva 17). Suurennuslasikuvakkeen takaa aukeaa tekstikenttä, johon voidaan syöttää aiheen nimi. Ruudun alareunassa on painike, josta aukeaa käyttäjälle kuvakkeet uuden muistion luomista varten (kuva 18).



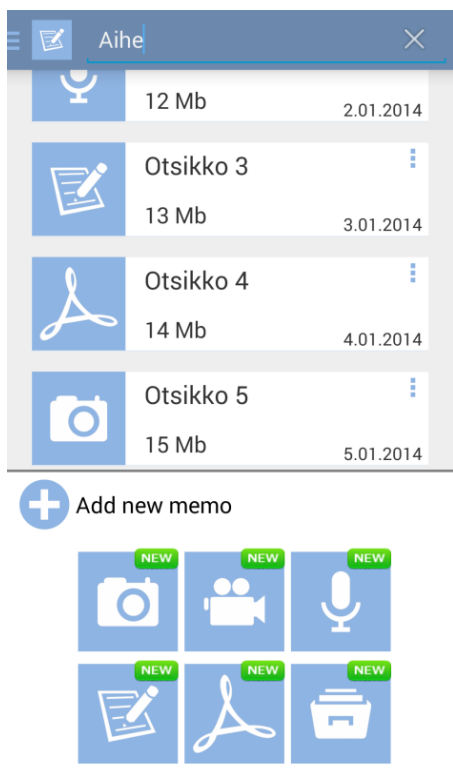
*KUVA 15. Muistiolistanäkymä, johon käyttäjän tallentamat muistiot on listattu.*



KUVA 16. Muistion lisävalikko, joka sisältää listan eri toimintoja muistiolle.



KUVA 17. Aihe-haku muistioille löytyy näytön yläosasta.

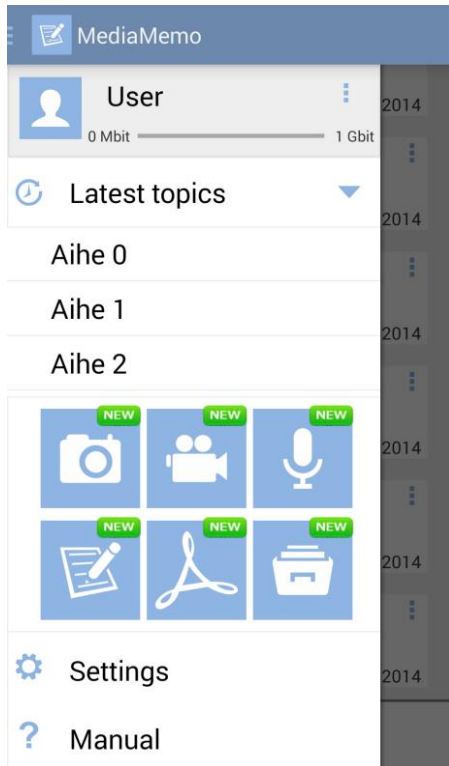


KUVA 18. Uuden muistion lisäyspainikkeet.

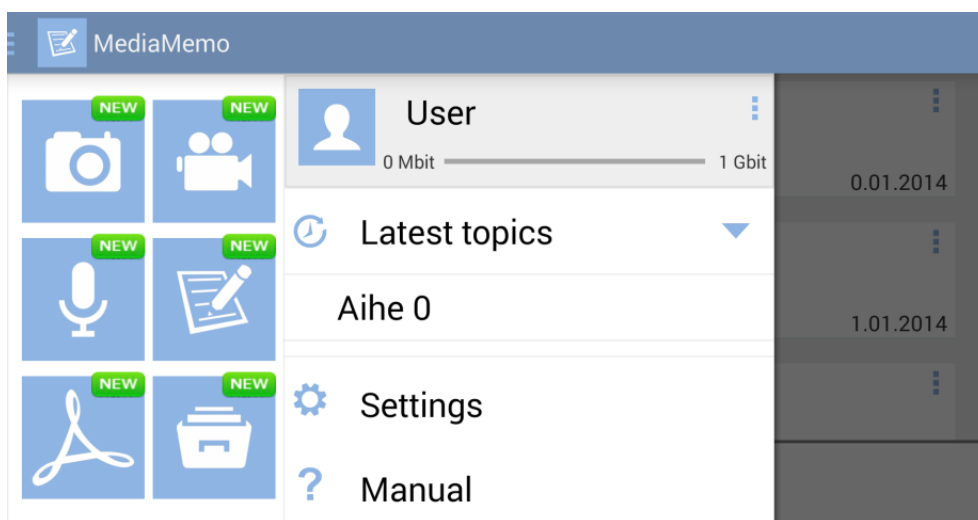
### Hallintavalikko

Hallintavalikko saadaan avattua, kun ollaan muistioden selausnäkymässä. Käyttäjän pitää vetää sormella näytön vasemmasta reunasta oikealle. Valikon yläosaan tulostetaan käyttäjän tiedot. Tiedoissa näkyy käyttäjän nimi ja tallennustila. Tallennustilassa ilmaistaan, kuinka paljon käyttäjä on käyttänyt tallennustilaa ja kuinka paljon hänellä on sitä yhteensä käytössä. Käyttäjätietojen yläreunasta löytyy lisävalikko, josta käyttäjä voi kirjautua ulos sovelluksesta. Käyttäjätietojen alla on lista käyttäjän viimeksi käyttämistä aiheista. Listan saa piilotettua otsikon vierestä olevasta nuolikuvaakkeesta. Listan alta löytyvät samat painikkeet kuin muistioden selausnäkymän alareunasta. Näiden painikkeiden alta löytyvät painikkeet avaamaan sovelluksen asetukset ja sovelluksen käyttöohjeet.

Hallintavalikosta piti rakentaa kaksi erilaista näkymää. Toista käytetään, kun puhelin on pystyasennossa (kuva 19), ja toista, kun puhelin on vaakasennossa (kuva 20).



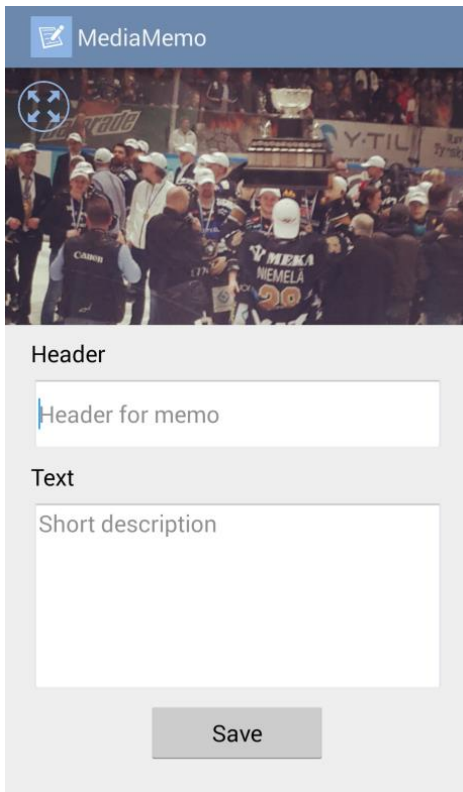
KUVA 19. Hallintavalikko laitteen ollessa pystyasennossa.



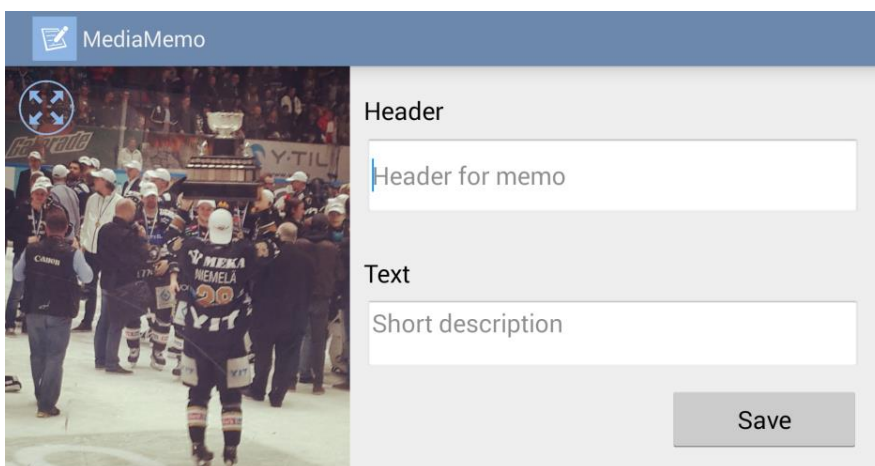
KUVA 20. Hallintavalikko laitteen ollessa vaakasennossa.

## Muistiolomake

Muistiolomake avataan, kun muistion tyyppi on valittu. Lomakenäkymä on yksikertainen ja siinä on ainoastaan muutama elementti: tiedoston esikatseluikkuna, tekstikenttä otsikolle, tekstikenttä muistion kuvaukselle ja tallennuspainike. Muistiolomakenäkymästäkin piti rakentaa omat versiot pysty- (kuva 21) ja vaaka-asennolle (kuva 22).



*KUVA 21. Muistiolomakenäkymä laitteen ollessa pystyasennossa.*



*KUVA 22. Muistiolomakenäkymä laitteen ollessa vaaka-asennossa.*

## 4.2.2 Muistiotyyppien tallennus

Sovelluksella voidaan tallentaa erityyppisiä muistioita ja ne käsitellään ohjelmassa eri tavalla. Sovellus hyödyntää Android järjestelmään valmiiksi asennettuja ohjelmia. Esimerkiksi kamerakuvan kaappausta varten käynnistetään Android-laitteesta löytyvä kamera. Android-järjestelmään valmiiksi asennettuja mediasovelluksia voidaan avata samaan tapaan kuten siirtymällä ohjelman sisällä näkymästä toiseen. Koodissa (kuva 23; kuva 24) näkyy, kuinka sovelluksessa voidaan siirtyä näkymästä toiseen ja kuinka avataan kameran sovellus käytettäväksi. Jotta tiedosto voidaan palauttaa ohjelman käyttöön, täytyy mediasovellukset käynnistää `startActivityForResult`-metodia. Tällöin kun palataan takaisin sovellukseen, tiedosto voidaan ottaa vastaan.

```
Intent intent = new Intent(this, MemoFormActivity.class);
startActivity(intent);
```

*KUVA 23. Ohjelmakoodin pätkä avaa sovelluksen sisällä toisen ns. activityn eli uuden näkymän käyttäjälle.*

```
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
startActivityForResult(intent, Constants.CAMERA_RESULT);
```

*KUVA 24. Ohjelmakoodin pätkä avaa kameran käytettäväksi sovellukselle.*

### Kuvien ottaminen

Kun käyttäjä valitsee muistiotyypiksi kuvan, ensimmäiseksi käyttäjälle aukeaa Androidin kameran sovellus. Jotta kuva tallennettaisiin tiedostoon, täytyy kameranavaus Intent-oliolle määrittää tiedosto mihin se tallennetaan (kuva 25). Kun kameralla on otettu kuva, se voidaan joko hylätä tai tallentaa. Hylättäessä käyttäjä voi ottaa uuden kuvan hylätyn tilalle. Tallennettaessa kuva palautetaan MediaMemo-sovellukselle. Kuvasta luodaan tilapäinen kuvatiedosto puhelimeen. Kun käyttäjä on syöttänyt muut muistion tiedot ja painanut tallennuspainiketta, kuva lähetetään palvelimelle. Kun lataus on valmis, tilapäinen kuva voidaan poistaa puhelimesta.

```
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
intent.putExtra(MediaStore.EXTRA_OUTPUT, TÄHÄN_TIEDOSTO);
startActivityForResult(intent, Constants.CAMERA_RESULT);
```

*KUVA 25. Kameran sovelluksen käynnistäminen ja kuvan tallentaminen valittuun tiedostoon.*

### **Videon tallennus**

Videomuistion tallennus hoituu pitkälti samalla tavalla kuin kuvamuistionkin. Ensimmäisenä käyttäjälle avataan Androidin kameran sovellus. Käyttäjä kuvaa videon kameralla, jonka jälkeen hylkää tai tallentaa sen. Tallennuksessa siitä tallennetaan tilapäinen videotiedosto puhelimen muistiin. Kun muistio tallennetaan, tilapäisvideotiedosto lähetetään palvelimelle. Tämän jälkeen tiedosto poistetaan puhelimesta.

### **Äänen nauhoitus**

Äänimuistion tallennukseen käytetään myös Android-laitteesta löytyvää sovellusta. Käyttäjälle avataan hänen puhelimestaan löytyvä nauhoitussovellus. Nauhoituksen jälkeen palautetusta äänestä luodaan tilapäinen äänitiedosto puhelimen muistiin. Tämän jälkeen se lähetetään palvelimelle. Kun lähetys on valmis, tilapäinen äänitiedosto poistetaan.

### **Tekstin kirjoittaminen tiedostoon**

Tekstimuistion tallentaminen poikkeaa muista siinä määrin, että tiedosto luodaan sovelluksessa itsessään. Tekstinkäsittelylomakkeella on tekstikenttä, johon kirjoitetaan teksti. Kun painetaan liittämiskuvaketta, sovellus luo tilapäistekstitiedoston, johon tallennetaan tekstikentän teksti. Kun käyttäjä tallentaa muistion se lähetetään palvelimelle, jonka jälkeen tiedosto voidaan poistaa puhelimesta.

### **PDF-tiedoston valinta**

Jos käyttäjä valitsee muistion tyyppiä PDF, käyttäjälle aukeaa lista puhelimen muistista löytyvistä PDF-tiedostoista. Käyttäjä valitsee listasta yhden tiedoston, josta luodaan tilapäinen tiedosto. Tallennuksen jälkeen tiedosto lähetetään palvelimelle, jonka jälkeen tiedosto voidaan poistaa puhelimesta.

## **Tiedoston valinta puhelimesta**

Käyttäjä voi myös valita puhelimestaan halutun tiedoston. Käyttäjä voi valita puhelimesta tuettavan tiedostotyyppin tiedoston. Kuva- ja videotiedoston käyttäjä voi valita suoraan galleriasta. Ääni-, teksti- ja PDF-tiedoston käyttäjä voi valita tiedostojen selaussovelluksella. Valitusta tiedostosta luodaan tilapäinen tiedosto muistiolle. Tiedoston lähetyksen jälkeen se voidaan poistaa puhelimen muistista. Alkuperäiselle tiedostolle ei tehdä mitään.

### **4.2.3 Rajapintojen käyttö**

Sovellus käyttää hyväksi Web-sovelluksessa olevia rajapintoja. Jokaisessa näkymässä käytetään rajapintojen välittämää tietoa. Joistakin näkymistä myös lähetetään tietoa rajapintojen kautta.

Muistioiden selausnäkymässä muistiot haetaan rajapinnan kautta ja asetetaan muistiolistaan. Hallintavalikossa käyttäjätiedot haetaan rajapintojen avulla tietokannasta, kuten myös viimeksi käyttämät aiheetkin. Muistiolomakkeella rajapintoja käytetään tiedon lähettämistä varten. Kun luodaan uusi muistio sen otsikko, kuvaus ja tiedosto lähetään palvelimelle.

### **4.2.4 Tiedoston lähetys**

Tiedostojen lähetys palvelimelle suoritetaan taustalla. Tällöin siitä ei aiheudu odottelua käyttäjälle. Lähetys aloitetaan ottamalla ensiksi yhteys palvelimelle. Sitten suoritetaan todentaminen lähettämällä käyttäjän tiedot. Seuraavaksi ohjelma lähettää tiedon mitä metodia se haluaa käyttää. Tämän jälkeen sovellus aloittaa tiedoston lähettämisen (kuva 26). Kun lähetys on valmis, yhteys palvelimelle katkaistaan.



```

bytesAvailable = fileInputStream.available();
bufferSize = Math.min(bytesAvailable, maxBufferSize);
buffer = new byte[bufferSize];

// Read file
bytesRead = fileInputStream.read(buffer, 0, bufferSize);

while (bytesRead > 0) {
    outputStream.write(buffer, 0, bufferSize);
    bytesAvailable = fileInputStream.available();
    bufferSize = Math.min(bytesAvailable, maxBufferSize);
    bytesRead = fileInputStream.read(buffer, 0, bufferSize);
}

outputStream.writeBytes(lineEnd);
outputStream.writeBytes(twoHyphens + boundary + twoHyphens+ lineEnd);

fileInputStream.close();
fileInputStream.close();

```

*KUVA 26. Tiedosto lähetetään while-silmukassa. Yhteys suljetaan lähetyksen päätyttyä.*

#### 4.2.5 Tietokanta

Toisin kuin Web-sovelluksessa Android-sovelluksien käyttämä sisäinen tietokanta on toteutettu käyttäen Androidin tukemaa SQLite-tietokantaa. Tietokantarakenne noudattaa samaa rakennetta kuin Web-sovelluksen tietokanta.

Puhelimen omaan tietokantaan tallennetaan rajapintojen kautta haettua tietoa. Kantaan rakennetaan omat taulut

- muistioille
- aiheille
- käyttäjälle
- viimeksi käytetyille aiheille.

Muistio-tauluun tallennetaan rajapinnan kautta välitetyt käyttäjän muistiot. Myös uudet muistiot, jotka tallennetaan puhelimesta, säilötään tähän tauluun.

## 5 POHDINTA

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa Softeri Oy:lle Android-sovellus, jolla käyttäjä voi tallentaa erityyppisiä muistioita. Sovelluksen piti olla myös helposti liitettävissä osaksi Softerin toiminnanohjausjärjestelmää. Sovelluksen perusideana oli sen yksinkertaisuus ja nopea käytettävyys tilanteessa kuin tilanteessa.

Työ alkoi työajan ja eri työtehtävien suunnittelulla. Koska aikaa ei projektin tekemiseen ollut paljoa, työajan suunnittelu nousi tärkeään rooliin. Jaoin työtehtävät pieniin osiin ja arvioin niille alustavan keston tunteina. Tällä tavoin pysyin itse selvillä, mitä milloinkin kannattaisi tehdä. Koodaustyön tekemisen aloitin Web-sovelluksen tekemisestä. Kun olin asentanut ohjelmistoalustan kehitysversion, pystyin rakentamaan tietokannan suunnitelmien pohjalta.

Web-sovellus rakentuikin nopeasti, ja sen jälkeen keskityin Android-sovelluksen toteutukseen. Sovelluksen teko sujui ilman suurempia ongelmia. Kaikkiin kysymyksiin Androidista löytyi nopeasti vastaukset Googlen hakukoneen avulla.

Vaikka työn aikataulu olikin tiukka, sovellus saatiin valmiiksi aikataulun puitteissa eikä siitä tarvinnut jättää pois mitään suunniteltuja ominaisuuksia. Sovellus tullaan lataamaan jossain vaiheessa Google Play -kauppaan. Ensiksi sovellus kuitenkin otetaan käyttöön Softerin henkilökunnalla, jonka avulla voidaan testata sovellusta ja saada uusia kehitysideoita.

Aihe oli mielestäni mielenkiintoinen ja kokonaisuus oli aikatauluun nähden aika suuri. Lopputuloksena syntyi kuitenkin toimiva versio sovelluksesta, johon olen erittäin tyytyväinen.

## LÄHTEET

1. Weiss, Todd R. 2014. Google Gives Free Online Training for Maps SDK for iOS. Saatavissa: <http://www.eweek.com/blogs/upfront/google-gives-free-online-training-for-maps-sdk-for-ios.html>. Hakupäivä 21.4.2014.
2. Learn About Java Technology. Oracle. Saatavissa: <http://www.java.com/en/about/>. Hakupäivä 22.4.2014.
3. What is Java technology and why do I need it? Oracle. Saatavissa: [http://www.java.com/en/download/faq/whatis\\_java.xml](http://www.java.com/en/download/faq/whatis_java.xml). Hakupäivä 22.4.2014.
4. Introduction. Google. Saatavissa: <http://developer.android.com/tools/workflow/index.html>. Hakupäivä 22.4.2014.
5. Getting Started with Android Studio. 2014. Google. Saatavissa: <http://developer.android.com/sdk/installing/studio.html>. Hakupäivä 22.4.2014.
6. What is PHP? 2014. PHP. Saatavissa: <http://www.php.net/manual/en/intro-what-is.php>. Hakupäivä 21.4.2014.
7. What can PHP do? 2014. PHP. Saatavissa: <http://www.php.net/manual/en/intro-whatcando.php>. Hakupäivä 21.4.2014
8. 1. Introduction. 2014. W3C. Saatavissa: <http://www.w3.org/TR/html5/introduction.html#introduction>. Hakupäivä 24.4.2014.
9. What is HTML5? w3schools.com. Saatavissa: [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp). Hakupäivä 24.4.2014.
10. Magnoni, Fábio. 2014. Introduction to HTML5. Mozilla Developer Network. Saatavissa: <https://developer.mozilla.org/en->

- [US/docs/Web/Guide/HTML/HTML5/Introduction\\_to\\_HTML5](http://US/docs/Web/Guide/HTML/HTML5/Introduction_to_HTML5). Hakupäivä 24.4.2014.
11. Pilgrim, Mark. Dive Into HTML5. Saatavissa: <http://diveintohtml5.info/>. Hakupäivä 25.4.2014.
  12. HTML. sivut.web. Saatavissa: <http://www.sivut.org/html/>. Hakupäivä 15.5.2014.
  13. Sheppy, Jeremie. 2014. AJAX. Mozilla Developer Network. Saatavissa: <https://developer.mozilla.org/en-US/docs/Glossary/AJAX>. Hakupäivä 28.4.2014.
  14. Kantor, Ilya. 2011. Overview: JavaScript, Flash, Java, Silverlight and ActiveX – What is JavaScript? JavaScript Tutorial. Saatavissa: <http://javascript.info/tutorial/overview#what-is-javascript>. Hakupäivä 28.4.2014.
  15. JavaScript-asetukset vuorovaikutteisten verkkosivujen avaamiseen. 2014. Mozilla support. Saatavissa: [https://support.mozilla.org/fi/kb/javascript-asetukset#w\\_mikac-on-javascript](https://support.mozilla.org/fi/kb/javascript-asetukset#w_mikac-on-javascript). Hakupäivä 28.4.2014
  16. What is jQuery? 2014. jQuery Foundation. Saatavissa: <http://jquery.com/>. Hakupäivä 28.4.2014.
  17. jQuery Tutorial . 2014. w3schools.com. Saatavissa: <http://www.w3schools.com/Jquery/default.asp>. Hakupäivä 28.4.2014.
  18. Knorr, Eric – Gruman, Galen. 2011. What cloud computing really means. InfoWorld. Saatavissa: <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>. Hakupäivä 30.4.2014.
  19. Sam Johnston. 2009. File:Cloud computing.svg. Saatavissa: [http://en.wikipedia.org/wiki/File:Cloud\\_computing.svg](http://en.wikipedia.org/wiki/File:Cloud_computing.svg). Hakupäivä 30.4.2014.

20. Private cloud. Webopedia. Saatavissa:  
[http://www.webopedia.com/TERM/P/private\\_cloud.html](http://www.webopedia.com/TERM/P/private_cloud.html). Hakupäivä  
30.4.2014.
21. What is Public cloud? Interoute. Saatavissa:  
<http://www.interoute.com/cloud-article/what-public-cloud>. Hakupäivä  
30.4.2014.
22. Rouse, Margaret. 2012. Community cloud. Saatavissa:  
<http://searchcloudstorage.techtarget.com/definition/community-cloud>.  
Hakupäivä 30.4.2014.
23. Rouse, Margaret 2013. Hybrid cloud. Saatavissa:  
<http://searchcloudcomputing.techtarget.com/definition/hybrid-cloud>.  
Hakupäivä 30.4.2014.
24. Fulton, Scott M. 2012. New Open Group Cloud Standard Introduces  
"XaaS" - Something as a Service. Saatavissa:  
[http://readwrite.com/2012/01/19/new-open-group-cloud-  
standard#awesm=~oDsNL2yx5P1OTW](http://readwrite.com/2012/01/19/new-open-group-cloud-standard#awesm=~oDsNL2yx5P1OTW). Hakupäivä 1.5.2014.
25. Pilvilaskennasta ja pilvipalveluista. 2013. Saatavissa:  
[http://sange.fi/~atehwa/cgi-  
bin/piki.cgi/pilvilaskennasta%20ja%20pilvipalveluista](http://sange.fi/~atehwa/cgi-bin/piki.cgi/pilvilaskennasta%20ja%20pilvipalveluista). Hakupäivä 1.5.2014.
26. What is SaaS? Interoute. Saatavissa: <http://www.interoute.com/what-saas>.  
Hakupäivä 5.5.2014.
27. What is PaaS? Interoute. Saatavissa: <http://www.interoute.com/what-paas>.  
Hakupäivä 5.5.2014.
28. What is IaaS? Interoute. Saatavissa: <http://www.interoute.com/what-iaas>.  
Hakupäivä 5.5.2014.
29. Web Services Glossary - Introduction. 2004. W3C. Saatavissa:  
<http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>. Hakupäivä  
6.5.2014.

30. Rouse, Margaret. 2007. Web services (application services). Saatavissa: <http://searchsoa.techtarget.com/definition/Web-services>. Hakupäivä 6.5.2014.
31. Danske Bank. 2014. Palvelukuvaus. Web Services -yhteyskäytäntö ja PKI-turvaratakisu. Saatavissa: <http://www.danskebank.fi/PDF/fi/Yritysasiakkaat/Maksuliike/WebServiceDescriptionFI.pdf>. Hakupäivä 9.5.2014.