OULUN AMMATTIKORKEAKOULU

Taneli Ridell

MOBIILISANELU-SOVELLUKSEN PAKKAUSALGORITMIN JA TALLENNUSMEKANISMIN PARANTAMINEN

MOBIILISANELU-SOVELLUKSEN PAKKAUSALGORITMIN JA TALLENNUSMEKANISMIN PARANTAMINEN

Taneli Ridell
Opinnäytetyö
Syksy 2017
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, ohjelmistokehitys

Tekijä: Taneli Ridell
Opinnäytetyön nimi: Mobiilisanelu-sovelluksen pakkausalgoritmin ja tallennusmekanismin parantaminen
Työn ohjaajat: Veikko Tapaninen, Lasse Haverinen, Juha Ahola
Työn valmistumislukukausi ja -vuosi: Syksy 2017
Sivumäärä: 31 + 1 liitettä

Opinnäytetyön aiheena oli Medanets Oy:n Mobiilisanelu-sovelluksen pakkausalgoritmin ja tallennusmekanismin parantaminen. Mobiilisanelu-sovellus on erityisesti lääkäreiden ja hoitajien käyttöön kehitetty Windows Phone 8 -sovellus, joka mahdollistaa saneluiden toteuttamisen ajasta ja paikasta riippumatta. Lähtökohtainen tarve työlle syntyi äänitallenteiden hitaasta prosessoinnista tallenteen koon kasvaessa.

Työn tavoitteena oli siis löytää vaihtoehtoinen tallennusformaatti äänitallenteille nykyisen WAV-formaatin tilalle sekä implementoida valittu pakkausalgoritmi sovellukseen. Pakkausalgoritmin tuli olla nopea ja muuntaa tallenne pienempään formaattiin, kuitenkin niin että puhe säilyisi hyvin selkeänä tallenteessa. Formaatin piti myös soveltua käytettäväksi eri verkkoselaimilla, jotta tallenteiden kuuntelu palvelimelta olisi mahdollisimman helppoa.

Lopputuloksena päädyttiin käyttämään MP3-formaattia sen erittäin laajan tuen takia. Implementoinnin jälkeen sovelluksen toiminta tallenteiden prosessoinnissa nopeutui huomattavasti. Työn tavoitteet siis saavutettiin, mutta jatkokehitykselle on edelleen tarvetta.

Jatkokehityksessä voitaisiin kiinnittää enemmän huomiota mobiililaitteen muistinkulutuksen kasvuun sovellusta käytettäessä.

Asiasanat: Windows Phone 8, äänentallennus, ohjelmointi

ABSTRACT

Oulu University of Applied Sciences
Information technology, software development

Author: Taneli Ridell
Title of thesis: Improving mobile dictation application's compression algorithm and storage mechanism
Supervisors: Veikko Tapaninen, Lasse Haverinen, Juha Ahola
Term and year when the thesis was submitted: Autumn 2017
Pages: 31 + 1 Appendice

The subject of this thesis was improving a mobile dictation application called Mobiilisanelu made by a company named Medanets. Mobiilisanelu is a Windows Phone 8 -application specifically made for doctors and nurses for making dictations anywhere and at any time.

The goal of this thesis was to find an alternative audio format for dictations made with the application, which uses WAV as its main format. The need for changing the audio format came from problems with audio file size and its processing to a server taking needlessly long time. The format also needed to be widely supported in different browsers for easy access to the dictations stored on a server.

As a result of this project, MP3 -format was chosen as the format to use due to its wide support with different platforms. After the implementation of this format the applications performance increased vastly. However this put more requirements on the mobile device used, mainly concerning its memory usage.

For further development extra emphasis should be paid to the application's memory usage.

Keywords: Windows Phone 8, Audio recording, Programming

SISÄLLYS

SANASTO

Dekoodaus	Pakkauksenhallinnan osa, joka hoitaa datan muuntamisen takaisin alkuperäiseen muotoonsa pakatusta muodosta.
Enkoodaus	Pakkauksenhallinnan osa, joka hoitaa datan pakkaamisen.
Kernel	Toiselta nimeltään käyttöjärjestelmän ydin, joka määrittelee käyttöjärjestelmän rakenteen, luokituksen ja ominaisuudet. Ydin voi olla kooltaan muutaman tuhannen tai jopa miljoonien koodirivien pituinen.
Koodekki	Toiselta nimeltään pakkauksenhallinta on algoritmi tai tietokoneohjelma, joka pakkaa ja purkaa ääni- tai kuvasignaalia tai muuten muuntaa datavuota tai signaalia.
MP3	Toiselta nimeltään MPEG-1 Audio Layer 3 on MPEG-1-tandardiin perustuva häviöllinen äänenpakkausmenetelmä, joka on pitkään ollut suosituin tiedostomuoto esimerkiksi musiikin jakelussa.
PCM	Englanninkielinen lyhenne sanoille Pulse Code Modulation (suom. pulssikoodimodulaatio). Menetelmä, jolla sähköinen äänitaajuussignaali koodataan digitaaliseen muotoon ottamalla signaalista tasaisin väliajoin näytteitä.
WAV	Tiedostomuoto, jota käytetään äänen tallentamiseen. Se on Microsoftin ja IBM:n käyttämä standardi.

ZIP	Yksinkertainen tiedostojen pakkausmenetelmä, joka pakkaa jokaisen tiedoston erikseen. Tämä nopeuttaa tiedostojen lukua, kun yhtä tiedostoa etsiessä ei tarvitse ladata muiden tiedostojen dataa.
Wrapper	Ohjelmakoodia, joka toimii välikätenä ohjelman koodin ja ulkoisen koodin välillä. Wrappereitä käytetään yleensä esimerkiksi eri ohjelmointikielien hyödyntämiseen saman ohjelman sisällä tai vaihtoehtoisesti tietoturvan lisäämiseksi.

1 JOHDANTO

Työn lähtökohtana on parantaa jo olemassa olevaa Medanets Oy:n tekemää Mobiilisanelu-sovellusta, joka on suunniteltu toimimaan Windows Phonella. Sovellus on suunniteltu käyttöön esimerkiksi lääkäreille mukana kuljetettavaksi. Lääkäri voi tehdä potilaskohtaiset sanelut ajasta ja paikasta riippumatta. Sanelut siirtyvät automaattisesti halutulle palvelimelle, kunhan puhelimella on käytössään toimiva nettiyhteys.

Sovellus itsessään on jo tässä vaiheessa käyttökelpoinen, mutta koska sanelut voivat olla jopa tunnin mittaisia, saneluiden prosessoinnissa ja lähettämisessä palvelimelle menee liikaa aikaa tallennusformaatin ja pakkauksen takia. Tämän takia on tarve tutkia vaihtoehtoisia tallennusmuotoja saneluille ja implementoida tehokkaampi tapa käsitellä tallenteita.

2 TYÖN TAVOITTEET

2.1 Mobiilisanelu-sovelluksen esittely

Mobiilisanelu-sovellus on suunniteltu erityisesti lääkäreiden ja hoitajien käyttöön, jolla he voivat sovelluksen avulla tehdä sanelunsa älypuhelimella, josta sovellus lähettää sanelut automaattisesti oikeaan paikkaan sanelun purkajille. Ratkaisu tuo ennen kaikkea joustavuutta työn tekemiseen sekä tehostaa hoitohenkilökunnan toimintaa. Lisäksi tarve kalliiden sanelulaitteiden hankinnalle vähenee. (1.)

Sanelusovelluksen avulla voidaan hoitotapahtuman yhteydessä ottaa myös kuvia, jotka lähetetään suoraan sairaalan kuva-arkistoon. Kuvat voidaan myös tarpeen mukaan linkittää saneluihin tai lähettää yksittäin. Esimerkiksi plastiikkakirurgit, ihotautilääkärit, konsultaatioita tekevät lääkärit, ensihoidon henkilökunta sekä kotihoitajat voivat tehdä potilaaseen liittyvät sanelut ja kuvan lähetykset suoraan hoitotapahtuman yhteydestä. (1.)

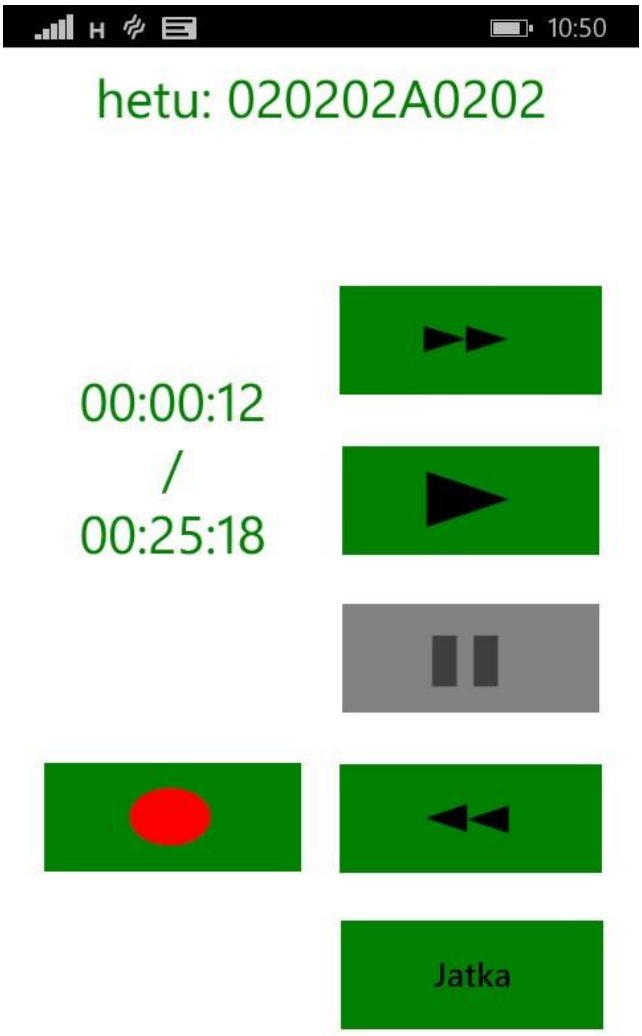
2.2 Nykyinen sovelluksen toiminta

Käytössä jo oleva sanelusovellus tallentaa sanelut WAV-formaatissa. Vaikka WAV voi tallentaa ääntä millä tahansa koodausmenetelmällä pakattuna, yleisin tapa äänen tallentamiseen on häviötön PCM-muoto. Tästä johtuen tallenne WAV-formaatissa on huomattavasti suurempi kuin ajallisesti samanmittaiset pakatut äänitiedostot.

Mobiilisanelu-sovellus (kuva 1) hyödyntää puhelimen mikrofonia tallentaakseen pääasiallisesti puhetta tiedostoon. Kun tallenne on valmis, sanelu pakataan ZIP-tiedostoon, joka lähetetään halutulle Linux-pohjaiselle palvelimelle salattuna.

KUVA 1. Mobiilisanelun päävalikko

Sanelusovelluksen käyttötarkoituksena on antaa lääkärille mahdollisuus tehdä saneluita missä tahansa ja milloin tahansa. Kun sanelu aloitetaan, käyttäjä laittaa puhelimestaan sanelusovelluksen päälle, painaa nauhoitusnappia ja tekee haluamansa sanelun. Sanelun päättyessä käyttäjä lopettaa tallennuksen ja puhelin pakkaa tallennetun WAV-tiedoston ZIP-tiedostoon ja tämän jälkeen lähettää tallenteen Linux-palvelimelle. Sovellus tarjoaa myös mahdollisuuden keskeyttää nauhoituksen, jatkaa nauhoista sekä kelata tallennetta aikaisempaan kohtaan keskustelussa ja nauhoittaa aiemman puheen päälle.



KUVA 2. Mobiilisanelun nauhoitusnäkymä

2.3 Sovelluksen ongelmakohdat

Lähetys toimii sujuvasti, kun käytössä on nopea verkkoyhteys ja tallenne ei ole kovin pitkä. Hitaammissa verkoissa tallenteen lähettämiseen kuluu liian kauan aikaa, koska tallenteet voivat olla jopa tunnin mittaisia. Pitkissä jopa tunnin mittaisissa tallenteissa pakkaamiseen ja pakkauksen jälkeen tallenteen lähettäminen palvelimelle vie huomattavan kauan aikaa, jopa 10 minuuttia, riippuen puhelimessa sillä hetkellä olevan verkkoyhteyden nopeudesta. WAV-tiedoston koko kasvaa liian isoksi, minkä takia on tarve pakata sanelu tiiviimpään tilaan, jotta tiedoston lähettämisessä palvelimelle ei menisi turhan kauan aikaa, jos käytössä ei ole nopeaa verkkoyhteyttä. Tiiviimmän pakkauksen

takia ohjelmalla menee huomattavan kauan aikaa päästä takaisin valmiustilaan eli tilaan, jossa voidaan tallentaa uusia saneluita. Tällä hetkellä ongelmaa lievitetään pakkaamalla WAV-tiedosto ZIP-tiedostoksi, joka pienentää kokoa vähän. Tiedostojen pakkaukseen ja lähetykseen kuluvan ajan vertailu alla olevissa taulukoissa (taulukot 1 ja 2).

Sanelun kesto	Sanelun koko (keski-määrin Mt)	Sanelun lähetys WLAN-verkosta sekuntia (n. 11 Mbit/s)	Sanelun lähetys WLAN-verkosta sekuntia (54 Mbit/s)	1 Mbit/s liittymän lähetysaika sekuntia	2 Mbit/s liittymän lähetys-aika sekuntia	Full rate liittymän lähetysaika (21 Mbit/s) sekuntia
1 min	1,8	1,31	0,27	14,4	7,2	0,69
5 min	9	6,55	1,33	72	36	3,43
10 min	18	13,09	2,67	144	72	6,86
15 min	27	19,64	4,00	216	108	10,29
30 min	54	39,27	8,00	432	216	20,57
60 min	108	78,55	16,00	864	432	41,14

TAULUKKO 1. Sanelun lähettämiseen kuluva aika teoreettisillä maksiminopeuksilla

Sanelun kesto	Sanelun työstö(pakkaus) aika keskimäärin sekuntia
1 min	1,5
5 min	7,5
10 min	15,0
15 min	22,5
30 min	45,0
60 min	90,0

TAULUKKO 2. Sanelun pakkaukseen kuluva aika nykyisellä implementaatiolla

Sanelun kesto	Sanelun lähetys WLAN-verkosta sekuntia (n. 11 Mbit/s) + pakkaus	Sanelun lähetys WLAN-verkosta sekuntia (54 Mbit/s) + pakkaus	1 Mbit/s liittymän lähetysaika sekuntia + pakkaus	2 Mbit/s liittymän lähetysaika sekuntia + pakkaus	Full rate liittymän lähetys-aika (21 Mbit/s) sekuntia + pakkaus
1 min	2,81	1,77	15,90	8,70	2,19
5 min	14,05	8,83	79,50	43,50	10,93
10 min	28,09	17,67	159,00	87,00	21,86
15 min	42,14	26,50	238,50	130,50	32,79
30 min	84,27	53,00	477,00	261,00	65,57
60 min	168,55	106,00	954,00	522,00	131,14

TAULUKKO 3. Sanelun pakkaukseen ja lähetykseen kuluva aika käytännössä

Nykyaikaisella mobiililaitteella tiedostojen pakkaus on nopeaa, eikä sen käyttämä aika ole hidastavin tekijä tallenteen siirtämisessä palvelimelle, koska nykyaikaisten mobiililaitteiden suorituskyky on koko ajan lähempänä halvemman kuluttajaluokan kannettavien tietokoneiden tehoa (2).

Sanelusovelluksen nykyisessä implementaatiossa tallenne pakataan ZIP-muotoon mahdollisimman hyvällä pakkaussuhteella. Se kutistaa tiedoston koon noin 70 %:iin alkuperäisestä tiedostosta. Pakkauksen kestossa menee sitä kauemmin, mitä isompi tiedosto pakataan. Tästä johtuen tiedoston koko ja yhteyden nopeus määräävät pääasiassa, kuinka nopeasti tallenne saadaan lähetettyä palvelimelle nauhoituksen loputtua.

2.4 Sovelluksen optimointi

Lähtökohtaisesti sovelluksen tallenteet pitäisi saada joko tallennettua suoraan vähemmän tilaa vievään ääniformaattiin tai konvertoitua jälkikäteen, kun tallenne on ensin tallennettu raakoina PCM-paketteina tai WAV-formaatissa käytettävään mobiililaitteeseen.

Kun halutaan pakata ääntä digitaalisesti mahdollisimman vähän tilaa vievään formaattiin, joudutaan turvautumaan häviöllisiin ääniformaatteihin. Häviöllisissä ääniformaateissa menetetään yleisesti iso osa dataa, joten on tärkeää, ettei ääntä pakata liian tiiviiseenkään muotoon, jotta tallenne pysyisi vielä

kuuntelukelpoisena. Tästä tulee tarve tutkia vaihtoehtoisia tallennusmuotoja saneluille ja implementoida tehokkaampi tapa käsitellä tallenteita.

3 WINDOWS PHONE -YMPÄRISTÖ

Windows Phone on Microsoftin kehittämä käyttöjärjestelmä mobiililaitteille. Windows Phone on suunniteltu korvaamaan vanha Windows Mobile -käyttöjärjestelmä. Windows Phoneksi kutsutaan yleisesti kaikkia mobiililaitteita, joissa on Windows Phone 7 -käyttöjärjestelmä tai sen jälkeen tullut versio. Windows Mobile –sovellukset eivät ole yhteensopivia Windows Phone -käyttöjärjestelmän kanssa., Myöskään Windows Phone 7 -käyttöjärjestelmässä ei voida suorittaa Windows Phone 8 -sovelluksia, vaan molemmille on tehtävä omat versionsa. Vaihtoehtoisesti voidaan suunnitella sovellus Windows Phone 7:lle, jolloin jäädään ilman Windows Phone 8:n ominaisuuksia, mutta saadaan yhteensopivuus molemmille laitteille. (3.)

3.1 Kehitys

Oletettavasti uuden Windows Phone -mobiiliympäristön kehitys aloitettiin jo vuonna 2004 koodinimellä Photon, jolloin suunnitteilla oli iso päivitys sillä hetkellä olleeseen Windows Mobile -ympäristöön. Kehitys oli kuitenkin hidasta ja vuonna 2008 Microsoft uudisti Windows Mobilen kehitysryhmää ja alkoi suunnittelemaan kokonaan uutta käyttöjärjestelmää mobiiliympäristöönsä. Alun perin käyttöjärjestelmän julkaisu piti tapahtua jo vuonna 2009, mutta sen sijaan saatiin päivitys vanhaan ympäristöön nimellä Windows Mobile 6.5. (3.)

Koska Windows Phonen kehitys oli erittäin nopeaa, kun sitä alettiin toteuttamaan, jouduttiin hylkäämään taaksepäin yhteensopivuus aiemman Windows Mobilen kanssa. Windows Phone 7 julkaistiin julkisesti marraskuussa 2010 Yhdysvalloissa. Lokakuussa 2012 Microsoft julkaisi Windows Phone 8:n. Windows Phone 8 korvasi aiemmin Windows CE -pohjaisen arkkitehtuurin Windows NT -pohjaisella kernelillä, joka on isolta osin samanlainen komponenteiltaan kuin Windows 8:n ydin. Tästä johtuen Windows 8:lle toteutetut applikaatiot on helpompi siirtää toimimaan Windows Phone 8 -alustalla. (3.)

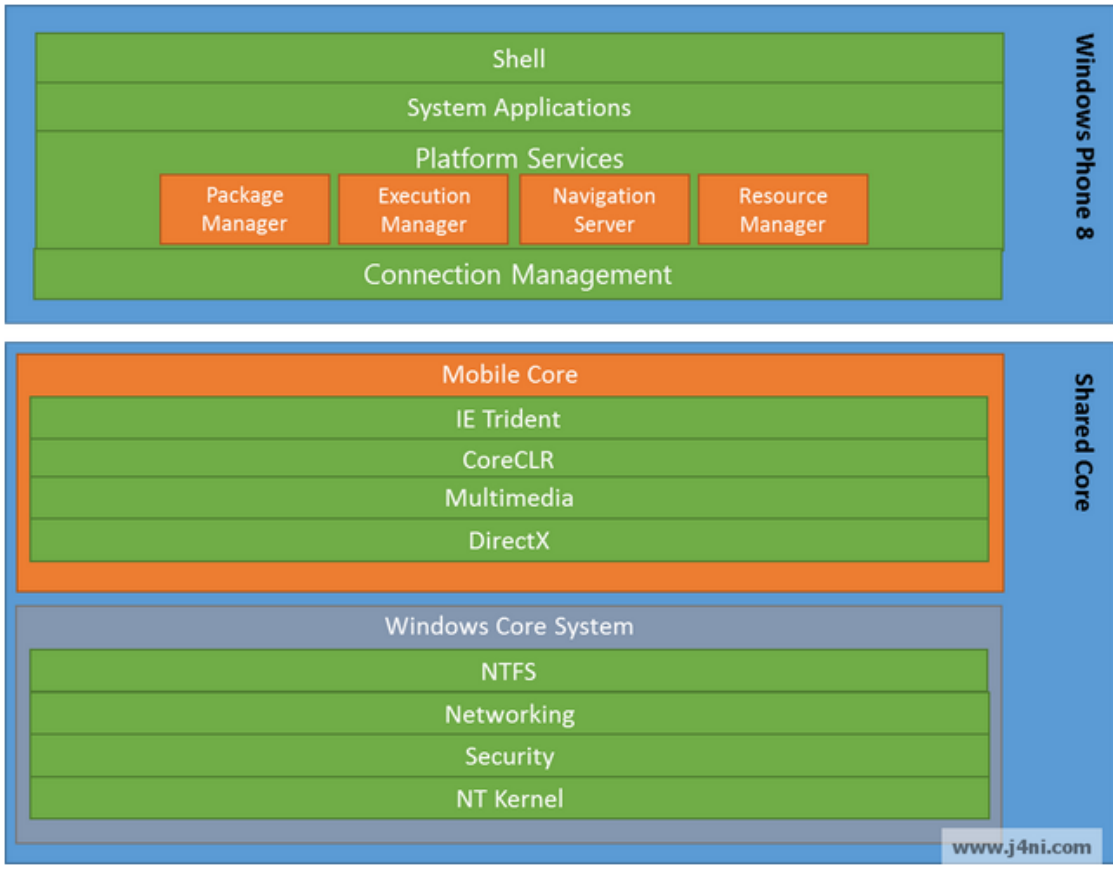
Windows Phone 8 toi myös mukanaan uudistetun laitteistotuen. Tämä mahdollisti moniytimisten prosessorien käytön sekä mahdollisuuden käyttää

aiempaa korkeamman resoluution näyttöjä. Päivitys paransi Windows Phonen kilpailukykyä Googlen ja Applen vastaavien älypuhelimien kanssa. (3.)

3.2 Ohjelmistoalusta

Microsoftin tavoitteena on ollut parantaa yhteensopivuutta mobiili- ja työpöytäkäyttöjärjestelmien välillä Windows Phone 8:aa suunnitellessa. Tästä johtuen Windows Phone 8:n ytimenä toimii sama järjestelmä kuin Windows 8:ssa, mutta karsittuna. Tämän lisäksi siinä on oma ytimensä, jossa on tarvittavat ominaisuudet mobiilikäyttöön.

Jaettu ydin näkyy kuvassa 3 alla. Mobiiliytimessä on samanlaista koodia kuin Windows 8:n alustalla, mutta se ei ole enää täysin samanlainen kokonaisuus, vaan siitä on tehty sopivampi mobiilikäyttöön. Toinen osa jaetusta ytimestä on karsittu versio Windows 8:n ytimestä. Jaetun ytimen yläpuolella on Windows Phone 8:n oma järjestelmä, joka koostuu muun muassa valmiista sovelluksista kuten ”Ihmiset” tai ”Musiikki & videot”.



KUVA 3. Windows Phone 8 Kernel (4)

Paketinhallinta on vastuussa ohjelmien koko elinkaaresta asennuksesta poistoon ja metadatan hallinnoinnista. Suoritushallinta hallitsee sovellusten ja niiden taustasovellusten toimintaa. Navigaatiopalvelin hallitsee kaikkea liikettä puhelimen näkyvällä osuudella eli päättää, mikä sovellus on päällimmäisenä näkyvillä milläkin hetkellä. Resurssienhallinta tarkkailee kaikkien sovellusten muistinkulutusta ja rajoittaa tai jopa poistaa prosesseja, jotta puhelimen käyttö olisi sujuvaa. (4.)

4 DIGITAALINEN ÄÄNEN TALLENNUS JA PAKKAUS

Digitaalisella äänen tallennuksella tarkoitetaan äänisignaalin kaappaamista analogisena ja muuntamista digitaaliseksi käsiteltävämpään muotoon. Tästä esimerkkinä on mikrofoni, joka vastaanottaa puhetta eli akustista signaalia eli signaalia ilman pitkittäistä aaltoliikettä. Puhe muunnetaan mikrofonilla analogiseen muotoon ja jälleen eteenpäin digitaaliseksi AD-muuntimen avulla.

Pakkauksessa taas pyritään joko säilyttämään tallennettu digitaalinen ääniraita häviöttömänä eli mahdollisimman vähäisillä äänen vääristymillä ja pienentämään sen kokoa tiedostona, tai vaihtoehtoisesti pyritään mahdollisimman vähän tilaa vievään ratkaisuun äänenlaadun kustannuksella.

4.1 AD/DA-muuntimet

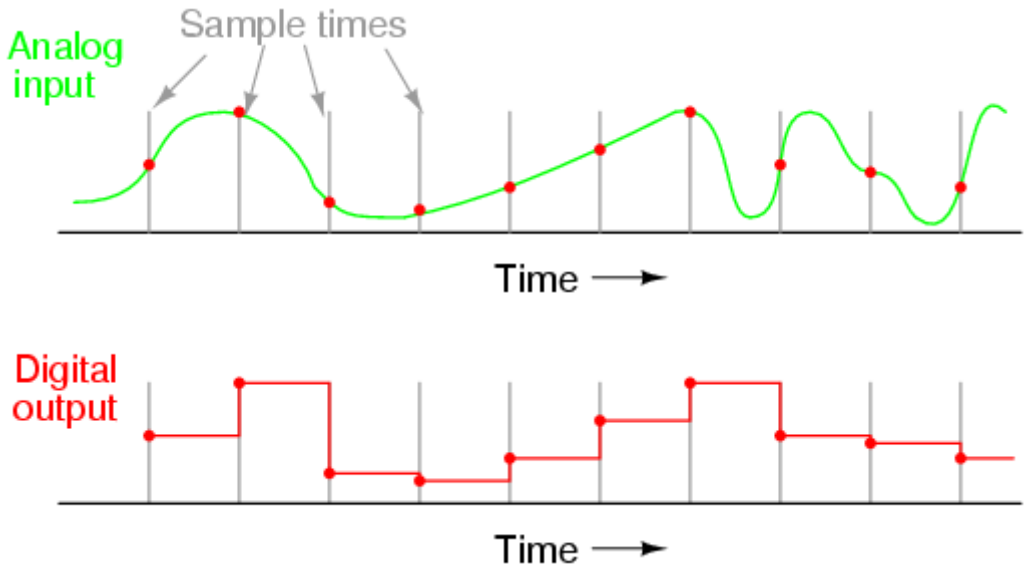
4.1.1 Analogia-digitaalimuunnos

Analogisen signaalin muuntaminen digitaaliseen muotoon tehdään analogia-digitaalimuuntimella eli AD-muuntimella. Muunnos tapahtuu ensin suodattamalla signaalista suurtaajuiset häiriöt. Tämän jälkeen suodatetusta signaalista otetaan näytteitä halutuin välein ja signaalin arvo näytteiden välillä jätetään ottamatta huomioon. Kun näytteet on otettu, tehdään kvantisointi eli kutakin näytettä asetetaan vastaamaan sitä lähinnä oleva digitaalinen arvo. Tämän jälkeen saadut näytearvot koodataan tallennusta tai siirtoa varten sopivaan muotoon. (7, s. 20.)

Muunnosprosessissa menetetään osa alkuperäisestä tiedosta näytteenotossa ja kvantisoinnissa. Kvantisoinnissa syntynyttä virhettä kutsutaan kvantisointivirheeksi tai kvantisointikohinaksi. Jotta analogia-digitaalimuunnos antaisi muunnettavan analogisen signaalin riittävän tarkan digitaalisen vastineen, muunnoksen täytyy täyttää kaksi ehtoa. Näytteitä pitää ottaa riittävän tiheästi eli näytevälin tulee olla riittävän pieni ja käytössä pitää olla riittävän monta erilaista kvantisoitua näytearvoa eli koodaus on tehtävä riittävän monella bitillä. Erittäin hyvään äänenlaatuun riittävä bittiarvo on yleisesti 16 MP3:lla. (5.)

4.1.2 Digitaali-analogiamuunnos

Toistossa tai siirron jälkeen digitaalisessa muodossa oleva tieto muutetaan yleensä takaisin analogiseen muotoon. Tähän tarvitaan DA-muunnin. Digitaali-analogiamuunnoksessa tapahtuvat seuraavat asiat. Koodattu tieto dekoodataan digitaalisiksi arvoiksi eli luvuiksi. Luvuista muodostetaan vastaavat analogiset signaaliarvot eli jännitteet ja sijoitetaan ne peräkkäin siten, että niiden aikaväli on sama kuin näyteväli oli analogia-digitaalimuunnoksessa. Kutakin signaaliarvoa venytetään pitopiirillä täyttämään väli seuraavaan arvoon asti (Kuva 4). Näin saatua signaalia suodatetaan, poistaen signaalin "kulmikkuus". (7, s. 21.)



KUVA 3. Analoginen ja digitaalinen signaali (6)

4.2 Mahdolliset pakkausmenetelmät Mobiilisanelu-sovellukselle

4.2.1 MPEG-1 Audio Layer 3 (MP3) -formaatti

MP3-formaatti on ollut laajassa käytössä jo 1990-luvun alkupuolelta asti. Lähes kaikki tietokoneella toimivat mediasoittimet tunnistavat ja toistavat sitä, vaikka niiden pääasiallinen tehtävä olisi toistaa jotain muuta ääniformaattia. Esimerkiksi Itunes on suunniteltu tukemaan AAC:ta ja Windows Media Player WMA:ta. (8.)

MP3:a käytetään erityisesti digitaalisen äänen tallentamiseen, kun halutaan säästää tilaa niin, että äänenlaatu on lähes verrannollinen vastaavaan tallenteeseen CD:llä, mutta tyypillisesti tiedostokoko on jopa vain kymmenesosa alkuperäisestä tallenteesta. (8.)

Tallenne saadaan pienempään kokoon käyttämällä koodekkia, kuten MP3. Koodekki poistaa alkuperäisestä tallenteesta osat, jotka ovat käytännössä mahdottomia kuulla ihmiskorvalla. Koodekkeja, jotka toimivat tällä periaatteella, kutsutaan häviöllisiksi formaateiksi. Näihin formaatteihin muunnettaessa menetetään osa taajuuksista tallenteessa, eikä niitä voida enää siihen palauttaa. Tiedoston kokoon voidaan edelleen vaikuttaa enkoodaamalla eri bittinopeudella, jolloin mitä isommalla bittinopeudella enkoodataan, sitä vähemmän menetetään alkuperäisestä tallenteesta eli toisin sanoen päästään lähemmäksi alkuperäisen tallenteen äänenlaatua. (8.)

Tyypillinen bittinopeus MP3:lle on välillä 128—320 kilobittiä sekunnissa, ja miniminä hyväksytylle äänenlaadulle musiikissa on pidetty 128 kilobittiä sekunnissa stereona eli käyttäen kahta äänilähdettä. Bittinopeudella siis ilmaistaan, kuinka monta kilobittiä sekunnissa tallenne vie tilaa. Isommalla bittinopeudella säilytetään suurempi osa otetuista näytteistä, jolloin kokonaisuudessaan MP3-koodekki tarjoaa mahdollisuuden enkoodata tallenteen myös VBR-tilassa (Variable Bit Rate), jolloin tallenteen eri kohdat tallentuvat eri bittinopeudella, riippuen kohdan monimutkaisuudesta. Tällä metodilla saadaan säästettyä isompi osa alkuperäisestä tallenteesta sekä saadaan myös hyödynnettyä koodekin tarjoama ”ylimääräisen” datan karsiminen. (8.)

4.2.2 Advanced Audio Coding (AAC) -formaatti

AAC-formaatti on myös standardisoitu, häviöllinen koodekki. Se on suunniteltu MP3:n seuraajaksi, ja yleensä sillä päästään parempaan äänenlaatuun enkoodauksen jälkeen samanlaisilla bittinopeuksilla. AAC:llä on mahdollisuus enkoodata eri tavalla sen mukaan, minkälaista dataa tallenne sisältää, ja sillä on omat algoritmit, jota se hyödyntää muun muassa puheelle ja musiikille. (9.)

4.2.3 Opus-formaatti

Opus on täysin vapaa ja monipuolinen koodekki. Se on erityisesti suunniteltu puheelle ja musiikille internetissä, jossa on tarve siirtää ääntä alhaisella viiveellä eli saada data mahdollisimman nopeasti vastaanottajalle. Sillä voidaan enkoodata tallenteita bittinopeuksilla 6–510 kbit/s VBR-tilassa. Opus ei ole vielä niin laajalle levinnyt kuin esimerkiksi MP3 tai AAC, mutta nykyään jo monista tietokoneella olevista mediasoittimista tuki opuksella enkoodattujen tallenteiden toistamiseen löytyy. (10.)

4.2.4 Speex-formaatti

Speex on myös ilmainen koodekki, joka on suunniteltu erityisesti puheen siirtoon internetin välityksellä. Se on Opusta vanhempi koodekki, jolle löytyy erittäin hyvä tuki varsinkin ohjelmissa, jotka käyttävät VoIP:a (Voice over Internet Protocol). Nykyään kuitenkin Opus on syrjäyttämässä sitä laajemmilla ominaisuuksillaan ja paremmalla äänenlaadulla. (11.)

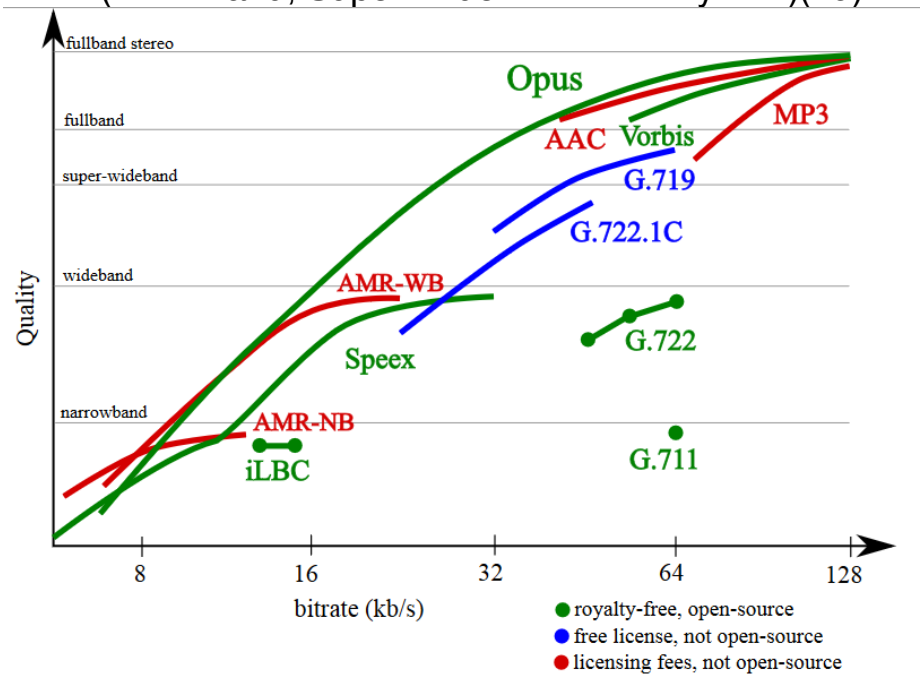
4.3 Pakkausmenetelmien vertailu

Lähtökohtana työlle oli löytää koodekki, jonka äänenlaatu on tarpeeksi hyvä puheelle myös kohtuullisen matalilla bittinopeuksilla, jotta sanelun käsittelyyn kuluva aika olisi mahdollisimman lyhyt. Hyvänlaatuinen tallenne on vielä mahdollista saavuttaa 64 kbit/s:n bittinopeudella, kunhan tallennetaan vain yhdellä kanavalla. Matalilla bittinopeuksilla kaksikaistaisella nauhoituksella voi olla negatiivisia vaikutuksia äänenlaatuun, vaikka muuten se onkin suositellumpi audiotallenteen muoto. Koodekin piti olla myös soveltuva implementoitavaksi Windows Phone 8.0 -alustalle tehtyyn sovellukseen ja sitä uudempiin versioihin.

Alkuperäisten vaihtoehtojen MP3, AAC, Opus ja Speex äänenlaadun vertailu näkyy kuvassa 5. Narrowband (suom. kapea kaista) tarkoittaa pienellä taajuusalueella toimivaa ääntä. Wideband (suom. laaja kaista) tarkoittaa taas laajempaa taajuusaluetta ja Super Wideband vielä laajempaa taajuusaluetta. Fullband (suom. täysi kaista) kattaa noin koko ihmisen kuuloalueen. Edellä mainittujen audiokoodekkien tarkemmat taajusalueet löytyvät alta. (12.)

- 300–3 400 Hz = Narrowband
- 50–7 000 Hz = Wideband
- 50–14 000 Hz = Super Wideband
- 20–20 000 Hz = Fullband

(”Wideband, Super-Wideband and beyond”)(13)



KUVA 5. Audiokoodekkien äänenlaadun ja bittinopeuden vertailu (13)

4.4 Valittu pakkausmenetelmä

Näistä vaihtoehdoista lopulta päädyttiin käyttämään MP3-formaattia. Syynä tähän oli sen kohtuullisen helppo toteutus jälkikäteen lisättynä sekä tuki tiedoston toistoon suoraan nettiselaimessa käytetyllä verkkopalvelimella, ilman että tiedostoformaattia tarvitsi enää muuttaa. Myös MP3:n äänenlaatu on erittäin hyvä muunnoksen jälkeen.

Huonoina puolina MP3:ssa on sen hinta, joka on 0,75 dollaria/dekooderi ja 2,50 dollaria/enkooderi, mutta kuitenkin minimissään 15 000 dollaria kalenterivuodelta. (14.)

Yksityiset käyttäjät voivat kuitenkin muuntaa tiedostonsa MP3-formaattiin omaan käyttöönsä ilmaisena. Tästä johtuen useat ilmaiset ohjelmat, jotka tukevat MP3-muunnoksia, siirtävät koodekin lisäämisen ohjelmaan loppukäyttäjän vastuulle. (14.)

5 KEHITYSPROSESSI

Työssä olennaisena osana oli kehittää jo ennaltaan toimivaa sovellusta Windows Phone 8:lle, jonka johdosta tarvittavat työkalut olivat jo selvillä. Koska työssä käytettiin Microsoftin Windows-pohjaista laitetta, Visual Studio 2012 ja tämän ohjelman työkalupaketti Windows Phone SDK 8.0 olivat olennaisessa osassa työn kehityksessä. Prosessi käy läpi tyypilliset vaiheet ohjelmistokehityksessä ja pyrkii avaamaan työn erinäisiä haasteita.

5.1 Visual Studio 2012

Microsoft Visual Studio on Microsoftin ohjelmankehitysympäristö, joka tukee useita eri ohjelmointikieliä kuten Visual Basic, C++, C# ja F#. Sillä voidaan tehdä muun muassa web-, Windows- ja mobiilisovelluksia. Siihen voidaan integroida monenlaisia täydennyksiä, jotka avustavat erilaisten projektien tekemisessä. Visual Studiolla on erityisen helppoa tehdä graafisia käyttöliittymiä, mikä isolta osin selittää sen suuren suosion. (15.)

5.2 Windows Phone SDK 8.0

Windows Phone SDK 8.0 on joukko työkaluja, jotka antavat mahdollisuuden Windows Phone 8 -sovellusten kehittämiseen Visual Studio -kehitys-ympäristössä. Työkalupaketin mukana tulee Visual Studio 2012 Express -versio, jos käyttäjällä ei jo ennestään ole joko Visual Studio 2012:ta tai uudempaa versiota. Visual Studion mukana tulee integroituna työkalut, jotka on erityisesti tehty Windows Phone 8 -sovellusten kehittämiseen ja julkaisun helpottamiseen. (16.)

Lisäksi paketissa tulee mukana emulaattorit, joilla voidaan simuloida erilaisia fyysisiä mobiililaitteita tietokoneessa, jos kehittäjällä ei ole sovelluksen käyttöön soveltuvaa mobiililaitetta. Näiden lisäksi mukana tulee Microsoft Expression Blend for Windows Phone, joka avustaa sovelluksen käyttöliittymän suunnittelussa erityisesti Windows Phone -laitteille. (16.)

5.3 Uuden pakkausmenetelmän implementointi

Pakkausmenetelmän implementointi osoittautui erittäin hankalaksi Windows Phone 8 -käyttöjärjestelmällä. Vakioltaan alusta tukee tallennusta formaateille WAV, AAC, sekä AMR. Mobiilisanelu-sovellus tallentaa äänen PCM-muodossa WAV-formaattiin 256 kbit/s. Äänenlaatu näin tallennettuna on erittäin hyvä, mutta tiedosto vie paljon tilaa, jonka takia on tullut tarve pakata WAV –tiedosto pienempään tilaan pakattaessa ZIP-tiedostoon. PCM-muodossa tallennetta on myös erittäin helppo käsitellä, kuten sovelluksessa tehdään muun muassa nauhoitettaessa aiemman tallenteen osan päälle.

AMR- ja AAC-formaattien ongelmaksi koitui hankala muokattavuus jälkeenpäin, sekä yhteensopivuus eri selainten mahdollisuuksissa toistaa sitä suoraan verkkopalvelimelta.

Koska Windows Phone 8:n valmiit enkooderit eivät soveltuneet sanelusovelluksen parantamiseen, päädyttiin MP3-formaattin, joka oli yhteensopiva kaikkien selainten kanssa sekä toimi suoraan palvelimelle lähetettynä ilman muokkausta. Koska sanelusovellukseen valmiiksi tehdyt tallenteen muokkausmahdollisuudet eivät olleet yhteensopivia MP3-formaatin kanssa, päädyttiin ratkaisuun, jossa tallenne muutetaan WAV-formaatista MP3-formaattin, kun käyttäjä on valmis tallentamaan nauhoituksen. Tällä menetelmällä päästiin lopputulokseen, jossa valmis 256 kbit/s:n bittinopeuksinen WAV-tiedosto muutettiin 64 kbit/s:n bittinopeuksiseen MP3-tiedostoon, ja näin saatiin tallenne pienennettyä neljännesosaan alkuperäisestä sekä säilytettiin tarpeeksi hyvä äänenlaatu puheelle MP3-enkoodauksen avulla.

Kun tallenne saatiin pienennettyä neljännesosaan, tallennetta ei ollut enää tarvetta pakata pienemmäksi pakattaessa ZIP-tiedostoon. Tallenteen enkoodaamiseen kuluva aika oli lyhyempi, kuin aika mitä tarvittiin ZIP-tiedoston pienempään kokoon pakkaamiseen, joten aikaa säästettiin tallenteen pakkaamisessa pienempään kokoon, jonka johdosta lähettämien palvelimelle sujuu nopeampaa.

5.4 Toimivuuden testaus palvelimen kanssa

Ohjelman toimivuuden testauksessa käytiin läpi kaikki yleisimmät skenaariot, joita loppukäyttäjä voi ohjelmalla toteuttaa. Eri testiskenaariot ja niiden tulokset löytyvät tarkemmin liitteestä 1: Testiskenaariot. Testien tarkoituksena oli kartoittaa ohjelman toimivuutta yleisimmissä skenaarioissa ja varmistaa, että kaikki toimii hyvin muutosten jälkeenkin. Lisäksi koetettiin tallenteen maksimikokoa ja ohjelman käyttäytymistä, kun sitä käytetään äärirajoillaan.

Testeissä käytiin läpi yleisimmät tilanteet, joita normaalissa sovelluksen käytössä voi tapahtua, esimerkiksi verkkoyhteys häviää jossain sanelun teon vaiheessa ja sanelu lähetetään vasta, kun verkkoyhteys löydetään uudestaan. Testauksen aikana havaittiin yksi ongelma, joka johtui käyttöönotetusta pakkausmetodista. Puhelimen muisti ei riittänyt käsittelemään yli puolen tunnin mittaisia saneluita, joten toistaiseksi sanelun pituus rajoitettiin 15 minuuttiin.

Palvelimen päässä toiminta vain yksinkertaistui. Tämä johtui puhelimen päässä suoritettavasta enkoodauksesta.

6 SOVELLUKSEEN JA PALVELIMEEN TEHDYT MUUTOKSET

Koska työn tarkoituksena oli vain optimoida tallenteen käsittelyä, sovelluksen toiminta pysyi lähes samana. Muutoksista voidaan mainita tallenteen käsittelyn toiminnan yhtenäistäminen, pakkauksen tiivistyksen poisto ja uutena osana tallenteen enkoodaaminen pienempään formaattiin koon puolesta kuitenkaan paljoa laadusta tinkimättä.

Tallenteen käsittelyn toiminnan yhtenäistämisellä saavutettiin samanlainen toiminta ohjelman eri vaihtoehdoissa käsitellä tallennetta, mikä taas yksinkertaisti uuden toiminnollisuuden lisäämistä sovellukseen.

Tallenteen tiiviimpään tilaan pakkaamista ei enää koettu tarpeelliseksi, kun tallenne saatiin huomattavasti pienempään kokoon muuntamalla se MP3-muotoon. Näin säästettiin aikaa ZIP-muotoon pakkaamisessa, ja lopulta aikaa lähetyksessä.

Enkoodaamisen toteutus tapahtui lisäämällä LAME-nimisen enkooderin DLL-tiedostot Sanelusovellus-projektiin. Koska nämä kirjastotiedostot ovat käytettävissä vain C-kielellä, jouduttiin turvautumaan wrapperiin, jonka tehtävänä on toimia välikätenä kirjastotiedostojen ja ohjelman välillä.

Palvelimen päässä toiminta pysyi samanlaisena, koska valittu ääniformaatti suoraviivaisti toimintaa entisestään.

7 YHTEENVETO

Työn päätarkoituksena oli tutkia, onko mahdollista käyttää parempaa pakkausmenetelmää kuin nykyinen WAV Medanets Oy:n Mobiilisanelu-sovellukseen Windows Phone 7/8:lle, sekä implementoida uusi pakkausmenetelmä, jos sopiva löytyy. Alun perin tutkittavia menetelmiä olivat MP3, Opus ja Speex, ja myöhemmin myös AAC.

Lähtökohtana Sanelusovelluksen ongelmakohdaksi osoittautui pidemmän tallenteen koko. Pitkissä jopa tunnin mittaisissa tallenteissa pakkaamiseen ja pakkauksen jälkeen tallenteen lähettäminen palvelimelle vei huomattavan kauan aikaa, jopa 10 minuuttia, riippuen puhelimessa sillä hetkellä olevan verkkoyhteyden nopeudesta.

Ongelmaa lähdettiin ratkaisemaan tutkimalla eri ääniformaattien soveltuvuutta. Pakkausalgoritmin piti olla nopea ja muuntaa tallenne pienempään muotoon, kuitenkin niin että puhe säilyisi hyvin selkeänä tallenteessa. Vaihtoehtoja oli näennäisesti monia. Useimpien formaattien ongelmaksi tuli joko hankala tai liikaa aikaa vievä toteutus Windows Phone 8 -käyttöjärjestelmälle sekä tallenteiden formaatin yhteensopivuus eri verkkoselaimissa, kun ne ovat palvelimella.

Lopulta päädyttiin MP3-formaattiin sen erittäin laajan tuen takia. Lähes kaikki tallenteita toistavat ohjelmat tukevat MP3-formaattia.

Lopputuloksena ohjelman toiminta nopeutui huomattavasti. Nykyisestä toteutuksesta johtuen muistin käyttöön olisi voitu kiinnittää enemmän huomiota. Puhelinmalleissa, joissa puhelimen sisäistä muistia on käytössä vain vähän, tallenteen maksimipituudeksi jää vain noin 15 minuuttia. Jatkokehityksessä Sovellukseen muistin käyttöä parannettaessa voitaisiin päästä lähes tunnin pituisiin tallenteisiin vastaavissa puhelimissa.

Työn alkuperäiset tavoitteet siis täytettiin, mutta työn tutkimusosasta tuli odotettua pitempi prosessi. Tämä johtui Windows Phone 8:n käyttöjärjestelmän puutteellisesta tuesta eri ääniformaattien käsittelyssä.

LÄHTEET

1. Sanelut ajasta ja paikasta riippumatta. Medanets Oy. Saatavissa: http://www.medanets.com/fi/ratkaisut/mobiilisanelu.html. Hakupäivä 25.11.2014.

2. Processors: Computer vs mobile. Techadvisory. Saatavissa: http://www.techadvisory.org/2013/12/processors-computer-vs-mobile/. Hakupäivä 11.2.2015.

3. Rubino, Daniel 2014. This is why Microsoft keeps “starting over” with windows phone. Windows Central. Saatavissa: http://www.windowscentral.com/why-microsoft-keeps-starting-over-windows-phone, Hakupäivä 18.9.2017.

4. Nevalainen, Jani 2012. Windows Phone 8 Kernel Architecture. .Net Junkyard. Saatavissa: http://j4ni.com/blog/?p=107. Hakupäivä 18.9.2017.

5. Sample rate & Bit depth. Applied Acoustic Systems. Saatavissa: https://www.applied-acoustics.com/techtalk/sampleratebitdepth/. Hakupäivä 18.9.2017.

6. Analoginen ja digitaalinen signaali. Saatavissa: http://sub.allaboutcircuits.com/images/04253.png. Hakupäivä 18.9.2017.

7. Haltsonen, Seppo – Levomäki, Jaakko – Rautanen, Esko T. 2006 .Digitaalitekniikka. Helsinki. Edita.

8. Understanding the MP3 format. Crutchfield. Saatavissa: http://www.crutchfield.com/S-8rRH5klp7Zb/learn/learningcenter/home/mp3.html. Hakupäivä 18.9.2017.

9. What Is AAC Format? How to Convert Video/DVD to AAC Format on Windows 10. 2017. Digiarty. Saatavissa: www.winxdvd.com/resource/aac.htm. Hakupäivä 18.9.2017.

10. Opus Interactive Audio Codec. 2017. Opus. Saatavissa: www.opus-codec.org. Hakupäivä 18.9.2017.

11. Speex: A Free Codec For Free Speech. 2016. Speex. Saatavissa: www.speex.org. Hakupäivä 18.9.2017.

12. Wideband, Super-Wideband and beyond. Saatavissa: http://blog.voxygen.co.uk/?p=1132. Hakupäivä 27.1.2015.

13. Quality versus bitrate. Saatavissa: http://www.opus-codec.org/comparison/quality.png. Hakupäivä 18.9.2017.

14. Mp3 Licensing. Saatavissa: http://mp3licensing.com/royalty/. Hakupäivä 15.4.2015.

15. Introducing Visual Studio. Microsoft Developer Network. Saatavissa: http://msdn.microsoft.com/en-us/library/fx6bk1f4%28v=vs.100%29.aspx. Hakupäivä 18.9.2017.

16. Introducing Windows Phone SDK 8.0. 2012. The Visual Studio Blog. Saatavissa: http://blogs.msdn.com/b/visualstudio/archive/2012/10/30/introducing-windows-phone-sdk-8-0.aspx. Hakupäivä 18.9.2017.

TESTISKENAARIOT

Testausta varten muutamia testicaseja: -

Sanelun teko offline -tilassa ja yhdistäminen myöhemmin
>>Sanelu, tallentuu normaalisti, paketti näkyy hetken aikaa puhelimen unsent -sivulla, sitten häviää.
>> Antaa käyttäjälle väärän kuvan tilanteesta, jos ei mitään viittausta siitä että verkkoyhteys ei ole päällä.
>> Onko mahdollisuutta kertoa tästä käyttäjälle?
>>online-tila päällä, paketti näkyy unsent -sivulla ja lähtee normaalisti palvelimelle.

Pitkän sanelun teko offline -tilassa (Niin iso tallenne että se ei mahdu sovelluksen ”background transferiin”.
>>Toimii normaalisti.
>> Todennäköisesti paketteja ei ollut background transferin ulkopuolella.

Kuinka monta pakettia mahtuu background transferiin? Pystyikö varmistamaan että osa paketeista ei mahtunut background transferiin? >> Puhelinverkossa maksimi 5mt, wifi 20mt ja wifi (kun laite latauksessa) 100mt
>> Background transfer: yhtäaikaisesti paketteja voidaan lähettää 2 kappaletta. Jonossa voi olla 25 pakettia/aplikaatio. MAX sanelu koko on siis 125megaa background transferissa.
>> 10min mp3 on 5 Mt. Tästä johtuen saneluita pitäisi tehdä yli 250minuuttia että sovelluksen background transfer –toiminto tulisi täyteen.

Tee useita saneluita offline -tilassa (esim. 5 kpl) ja saneluiden jälkeen vaihdetaan online –tilaan.
>> Ohjelman ylälaitaan ilmestyy "No data network available" useamman tallennetun sanelun jälkeen.
>> Aika pitkä viive. Tämä tieto tulee tod.näk. background transferista, kun sovellus on varmistanut yhteyden puuttuvan.
>> Online -tilassa kaikki sanelut lähetettiin onnistuneesti.

Tee sanelu ja kesken lähetyksen aseta yhteydet offline -tilaan
>> Tiedostoja ei lähetetty ofline -tilassa, tiedostot lähetettiin oikein ja poistettiin puhelimesta vaihtaessa online –tilaan.
>> Jos sanelu on paloiteltu ja vain osa paketeista lähetetään säilytetään lähetystä odottavia osatiedostoja /tmp hakemistossa. Hakemistosta ei löytynyt yhtään osatiedostoa lähetysten jälkeen.
>> Tarkistettiin että Tanelin sanelu tanelisanelija_2014-11-12-14-36-14.mp3 oli kahdessa osassa lähetetty.

Tee sanelu ja keskeytä enkoodaus sulkemalla sovellus. Yhdistä sovellukseen uudestaan.

Esimerkki monisivuisesta liitteestä. Sivunumero tulee automaattisesti ylätunnisteeseen, liitteen numero täytyy vaihtaa.

>> OK , Tallenne lähetetään onnistuneesti.
>> Havaittiin myös että kesken jääneet sanelut lähetetään automaattisesti seuraavan käynnistyksen yhteydessä. Hyvä asia ettei saneluita poisteta.
>> Jatkokehitys. Lippu milloin käyttäjä on painanut lähetä, muussa tapauksessa mahdollisuus jatkaa keskeytynyttä sanelua.

Keskeytä enkoodaus ja keskeytyksen jälkeen aloita heti uusi enkoodaus.
>> Tallenteen käsittely onnistuu samaan aikaan kun nauhoitetaan jo uutta tallennetta.
>> OK

Tee maksimi pituinen sanelu ja aloita heti uusi sanelu.
>> 31 minuuttia pitkä (58 megaa .wavia). Automaattisesti loppui. Tallenna näppäin kumminkin näkyvissä ja sanelua pystyi jatkamaan. Sanelun kokonaisaikaan tuli joka painalluksella sekuntti lisää ja pysähtyi automaattisesti sen jälkeen. Kun sanelua yritti lähettää, sanelua ei käännetty mp3-tiedostoksi. Puhelimeen jäi .xml ja .wav tiedostot, joita ei käsitelty. Seuraava noin 5 minuutin sanelu onnistui ongelmitta.
>> Ei testattu 29 minuutin sanelua jossa käyttäjä lopettaa sanelun vertauskohtana. Pisin Tanelilta löytyvä .wav on 15 minuuttia pitkä. Taneli vertaa 58 Mt ja 31 minuuttia pitkää sekä 15 minuuttia pitkää xml sekä wav tiedostoa keskenään ja laittaa ne sovelluksen käytössä olevaan kansioon(”Isolated Storage”). Sanelut eivät mene jostain syystä työkalun kautta oikeaan paikkaan niin että sovellus löytäisi ne. Huomiona että saneluissa jotka eivät siirry on metadata.xml tiedostossa DataLength 0 kun onnistuneissa siinä on sanelun koko.

LISÄTESTI: Taneli testaa tehdä 30min sanelusovelluksen debug moodissa ja katsoo tuleeko virheilmoitusta silloin näkyviin tai breakpointin avulla näkisi mihin liian pitkä tallenne jumittuu. Mp3-pakkaamisessa ilmeisesti käytetään sen verran enemmän muistia tai .wav tallentuu keskusmuistiin kahteen kertaan. Taneli selvittää vielä ongelmaa viimeisenä tehtävänään.

Testaa useamman kuvan ottaminen ja lähettämisen toiminta.
>> OK, tarkistettu, kuvat löytyivät palvelimelta.

Testaa lähteekö kuva automaattisesti jos kuvanoton keskeyttää ennen lähetä -näppäimen painamista.
>> Kuva ei jäänyt puhelimeen, eikä sitä lähetetty palvelimelle. Eli toisin sanoen kuva katosi kokonaan, koska se ei päässyt lähetykseen asti.

LISÄTESTI: - Testaa että lähettyykö sanelu background transferista automaattisesti ilman, että sovellukseen palataan lähetyksen aloittamisen jälkeen. Katkaise verkkoyhteys, tee lyhyt sanelu, katso että sanelu menee

background transferiin, odota 10 min, verkkoyhteys takaisin, katso palvelimelta menikö sanelu perille.
>> Juha testasi. Siirtyi OK.