

Android-pohjaisen lintukirjasovelluksen suunnittelu

Teemu Lakela

Tietotekniikan opinnäytetyö
Ohjelmistosuunnittelu
Insinööri (AMK)

KEMI 2013

ALKUSANAT

Kiitokset Antti Niemelälle opinnäytetyön mainiosta aiheesta.

TIIVISTELMÄ

KEMI-TORNION AMMATTIKORKEAKOULU, Tekniikka

Koulutusohjelma:	Tietotekniikka, ohjelmistotekniikka
Opinnäytetyön tekijä:	Teemu Lakela
Opinnäytetyön nimi:	Android-pohjaisen lintukirjasovelluksen suunnittelu
Sivuja (joista liitesivuja):	54 (0)
Päiväys:	18.3.2013
Opinnäytetyön ohjaaja:	Antti Niemelä, projektipäällikkö
<p>Opinnäytetyön tavoitteena oli suunnitella Android-mobiililaitteilla toimiva lintukirjasovellus, joka antaisi laitteen käyttäjälle helpon, mukana kulkevan alustan, lintujen tietojen ja kuvien selaamiseen ja tallentamiseen.</p> <p>Aihe valittiin, koska tällä hetkellä Android on suosituin käyttöjärjestelmä älypuhelimissa, mikä tekee siitä hyvän alustan tutustua matkapuhelinsovellusten kehittämiseen. Myöskään työn kuvauksen mukaista sovellusta ei ollut saatavilla Android-sovelluksia jakelevassa Play-kaupassa, joten kysyntää kuvatun kaltaiselle sovellukselle saattaisi löytyä, jopa kaupallisessa mielessä.</p> <p>Työ aloitettiin tutkimalla Android-alustan taustatietoja ja ohjelmistokehityksen teoriaa. Myöhemmin tutustuttiin Eclipse-kehitysympäristöön, jossa sovelluksen varsinainen ohjelmointi tapahtuu. Teoriaa kokeiltiin käytännössä ohjelmoimalla osia sovelluksesta ja testaamalla niitä emulaattorissa. Lopuksi kirjoitettiin suunnitelma sovelluksen toteuttamisesta tätä dokumenttia varten.</p> <p>Opinnäytetyö epäonnistui alkuperäisessä tavoitteessaan, jossa sovellus oli tarkoitus myös kehittää pelkän suunnittelemisen sijaan. Työstä muodostui kuitenkin hyvä teoriapaketti Android-ohjelmistokehitykseen.</p>	
Asiasanat: Android, käyttöjärjestelmät, ohjelmistokehitys, matkapuhelimet.	

ABSTRACT

KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES, Technology

Degree programme:	Information Technology
Author:	Teemu Lakela
Thesis title:	Designing Android Based Bird Book Application
Pages (of which appendixes):	54 (0)
Date:	18 March 2013
Thesis instructor:	Antti Niemelä, Project Manager
<p>The objective of this study was to design a bird book application running on Android mobile devices. The application would give the user an easy mobile solution to save and browse information and pictures of different birds.</p> <p>The subject was selected because of the popularity of Android. As the world's most used mobile operating system, Android would be the best platform to learn mobile programming skills. Also when the subject was selected, one could not find a single similar application from the official Android application distributor, Google Play Store.</p> <p>The study was started with background research of the Android platform and software development. Later, the Eclipse IDE (Integrated Development Environment) was used to test the theory in practice by programming some of the application's functions and by testing them in the Android emulator. In the end, the designs of the application were written down to this document.</p> <p>This study failed in its original objective, which was to design and develop the application, instead of just designing it. In any case, the work formed a nice package containing theory of Android software development.</p>	
<p>Keywords: Android, operating systems, software development, mobile phones.</p>	

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
KÄYTETYT MERKIT JA LYHENTEET	6
1 JOHDANTO	8
2 ANDROID-ALUSTA	9
2.1 Lyhyt historia	9
2.2 Käyttöliittymä	11
2.3 Tietoturva	16
2.4 Versiohistoria	18
2.5 Ongelmia	21
3 ANDROIDIN ARKKITEHTUURI	23
3.1 Linux kernel	23
3.2 Libraries - kirjastot	24
3.3 Application Framework - ohjelmistokehys	25
3.4 Applications - sovellukset	27
4 OHJELMAN RAKENNE	28
4.1 Activity	28
4.2 Service	32
4.3 Content Providers	32
4.4 Broadcast Receiver	33
4.5 Asettelut	33
5 SOVELLUKSEN SUUNNITTELU	36
5.1 Graafinen käyttöliittymä	36
5.2 Tietokanta	41
5.3 Edistyneet toiminnot	43
5.4 Käytettävät työkalut	44
6 POHDINTA	46
LÄHTEET	49

KÄYTETYT MERKIT JA LYHENTEET

ADT	Android Development Tools, liitännäinen Eclipse-kehitysympäristöön, joka mahdollistaa Android-ohjelmien kehittämisen
AOSP	Android Open Source Project
APK	Application Package File, tiedostopaketti, josta Android-sovellus asentuu
EDGE	Enhanced Data rates for GSM Evolution, matkapuhelinten pakettikytkentäinen tiedonsiirtotekniikka
ER	Entity Relationship, malli/kaavio tietokokoelman käsitetason kuvauksessa
GB	Giga Byte, gigatavu
GPS	Global Positioning System, satelliittipaikannusjärjestelmä
GSM	Global System for Mobile Communications, matkapuhelinjärjestelmä
HSPA	High Speed Packet Access
HTML5	Hypertext Markup Language, yleisesti nettisivujen koodaamiseen käytetty kuvauskieli
IMAP4	Internet Message Access Protocol, sähköpostien lukemiseen tarkoitettu protokolla
LTE	Long Term Evolution, edistynyt 3G-tekniikka matkapuhelinverkkojen datan siirtonopeuksien kasvattamiseen
mAh	Milliampeeritunti
Mpx	Megapikseli
NFC	Near Field Communications, radiotaajuisen etätunnistuksen mahdollistava tekniikka
PIN	Personal Identification Number, tunnusluku
POP3	Post Office Protocol version 3, sähköpostin hakemiseen tarkoitettu protokolla
RAM	Random Access Memory, käyttömuisti
SDK	Software Developer Kit, ohjelmistokehitykseen tarkoitettu työkalupaketti
SIM	Subscriber Identity Module, älykortti matkapuhelinliittymän käyttäjätietojen tallentamiseen
SMTP	Simple Mail Transfer Protocol, protokolla sähköpostiviestien välittämiseen sähköpostipalvelimien kesken
SSL	Secure Socket Layer, tietoverkkosalausprotokolla
UAC	User Account Control, käyttäjätilien valvonta

USB	Universal Serial Bus, sarjaväyläarkkitehtuuri oheislaitteiden liittämiseen tietokoneeseen
VoIP	Voice over Internet Protocol, äänen siirto reaaliaikaisesti IP-protokollaa käyttävän verkon välityksellä
Wi-Fi	Wireless Fidelity, langaton lähiverkkotekniikka

1 JOHDANTO

Tämän opinnäytetyön aiheena oli suunnitella ja kehittää Android-käyttöjärjestelmää käyttävillä mobiililaitteilla toimiva lintutietokirjasovellus. Ohjelma antaisi laitteen käyttäjälle helpon, mukana kulkevan, elektronisen alustan, johon kerätä tietoa ja kuvia linnuista. Ohjelman edistyneet toiminnot mahdollistavat myös lintujen tunnistamisen perinteistä paperista kirjaa helpommin.

Android on tällä hetkellä käytetyin käyttöjärjestelmä niin kutsutuissa älypuhelimissa. Se on dokumentoitu laajasti, joten esimerkkejä ja ohjeita ongelmiin löytyy helposti. Tämä tekee Androidista mainion alustan tutustua matkapuhelinsovellusten kehittämiseen.

Ensisijainen syy aiheen valinnalle oli oma kiinnostukseni Android-pohjaisiin laitteisiin. Itse aiheen keksi opinnäytetyön ohjaaja, Antti Niemelä. Aihetta valittaessa Android-sovelluksia jakelevasta ja myyvästä Play-sovelluskaupasta ei löytynyt yhtäkään kattavaa lintutietokirjasovellusta, joten kysyntää tämänkaltaiselle sovellukselle löytyisi. Tarvittaessa sovellus olisi siis mahdollista kaupallistaa laittamalla se myyntiin Play-kauppaan.

Työn tavoitteena on ohjelmoida toimiva pohja ohjelmalle, jossa käyttäjä voi tallentaa ja hallita tekstiä ja kuvia matkapuhelimessaan. Kyseessä on siis pieni tietokantaohjelma yksinkertaisella graafisella käyttöliittymällä. Jatkossa ohjelmaan on mahdollista lisätä uusia ominaisuuksia.

2 ANDROID-ALUSTA

Android on amerikkalaisen ohjelmistoyhtiö Googlen johtaman Open Handset Alliancen kehittämä avoin, Linux-pohjainen ohjelmistopino mobiililaitteille, kuten matkapuhelimille ja tableteille. Se sisältää käyttöjärjestelmän, väliohjelmistoja, kirjastoja ja perusohjelmia. /32/

Avoimen lähdekoodinsa ansiosta laitevalmistajat ovat vapaita käyttämään Androidia puhelimissaan. Esimerkiksi siinä missä Applen valmistamia iPhoneja on saatavilla ainoastaan kalleimmassa yli 600 euron, hintaluokassa, Android-puhelimia löytyy aina alle 100 euron halpamalleista, yli 600 euron huippumalleihin. Tämän ansiosta Android onkin levinnyt niin laajalle ja vuoden 2012 kolmanteen neljännekseen mennessä, alustasta on kasvanut suurin älypuhelinlaluusta 75 %:n markkinaosuudella. /31/

2.1 Lyhyt historia

Android Inc. perustettiin lokakuussa 2003 Palo Altossa, Kaliforniassa. Yhden yrityksen perustajan, Andy Rubinin, mukaan suunnittelufilosofiana oli kehittää “älykkäämpi mobiililaitte, joka on tietoinen käyttäjänsä sijainnista ja mieltymyksistä”. Noihin aikoihin Androidin kehityksestä ei kantautunut julkisuuteen sen tarkempaa tietoa kuin, että yritys kehittää ohjelmistoja matkapuhelimille. /26/

Elokuussa vuonna 2005 hakukoneyhtiönä tunnettu Google kaikessa hiljaisuudessa osti Androidia kehittävän startup-yritys Android Inc:in, palkaten samalla yhtiön tärkeimmät työntekijät. Pian hankinnan jälkeen alettiin spekuloidaan Googlen mahdollisista pyrkimyksistä matkapuhelinlalualla uudella mobiilikäyttöjärjestelmällä. /26/

Marraskuussa 2007 Google ilmoitti kehittävänsä uutta avoimen lähdekoodin alustaa mobiililaitteille yhteistyössä useiden monikansallisten teknologia- ja matkapuhelinyhtiöiden, kuten teleoperaattori T-Mobilen, matkapuhelinvalmistaja HTC:n ja mm. matkapuhelinkomponentteja valmistavan Qualcommin kanssa. Kehitystyö tapahtui Open Handset Alliance -nimisen liittouman alla. Liittouma koostui alkajaisiksi 34 yrityksestä, joiden tavoitteena oli kehittää alusta, joka tarjoaisi paremman käyttökokemuksen aikaisempiin laitteisiin verrattuna ja joka avoimuutensa ansiosta mahdollistaisi uusien tuotteiden ja palveluiden tarjoamisen asiakkaille

pienemmillä kustannuksilla ja nopeammalla aikataululla. Nykyään Open Handset Alliance koostuu 84 yrityksestä. /32/, /45/

Ensimmäinen Android-pohjainen (versio 1.0) älypuhelin, HTC Dream, julkaistiin lokakuussa 2008. Ilman liittymää puhelimen hinta Yhdysvalloissa oli 399 dollaria. Kuuden kuukauden aikana laitetta myytiin Yhdysvalloissa miljoona kappaletta. /30/

Taulukossa 1 vertaillaan lyhyesti HTC Dreamin ominaisuuksia silloiseen älypuhelinmarkkinajohtajaan ja Androidin pahimpaan kilpailijaan, Applen toisen sukupolven iPhone 3G:hen.

Taulukko 1. HTC Dream ja Apple iPhone 3G vertailu. /13/, /58/

	HTC Dream	Apple iPhone 3G
Julkaisupäivä	22. lokakuuta 2008	11. kesäkuuta 2008
Fyysiset mitat	117,7 x 55,7 x 17,1 mm	115,5 x 62,1 x 12,3 mm
Paino	158 g	133 g
Näyttö	3,2”, 320 x 480 px, kapasitiivinen kosketusnäyttö	3,5”, 320 x 480 px, kapasitiivinen kosketusnäyttö
Keskusyksikkö	528 MHz ARM11	620 MHz, RISC ARM
Keskusmuisti	192 MB RAM	128 MB eDRAM
Massamuisti	Tuki maksimissaan 16 GB microSD-muistikortille	8 tai 16 GB sisäistä flash-muistia mallin mukaan
Akku	1150 mAh, irrotettava	1150 mAh, sisäänrakennettu
Kamera	3,2 Mpx	2,0 Mpx
Yhteydet	GSM/GPRS/EDGE UMTS/HSDPA Wi-Fi (802.11 b/g) Bluetooth 2.0+EDR extUSB A-GPS	GSM/GPRS/EDGE UMTS/HSDPA Wi-Fi (802.11 b/g) Bluetooth 2.0 USB 2.0 A-GPS

2.2 Käyttöliittymä

Android-matkapuhelimen käyttöliittymä koostuu suuresta kosketusnäytöstä, joka reagoi usean pisteen kosketukseen (multi-touch), fyysisestä virtapainikkeesta ja äänenvoimakkuutta säätävästä keinukytkimestä, sekä muutamasta näytön alapuolella olevasta näppäimestä. Uudemmat Android-versiot tukevat myös USB-oheislaitteita, kuten hiiriä ja näppäimistöjä.

Näppäimet

Laitteen näytön alapuolelta löytyvät Home-, Menu- ja Back-näppäimet, sekä joistakin puhelinmalleista myös Search-näppäin. Näppäimet voivat olla tyypiltään mekaanisia kytkimiä, jotka naksuvat painettaessa, tai kapasitiivisia näppäimiä, jotka reagoivat kevyeen kosketukseen. /7/

Alkaen Androidin versiosta 3.0, nämä näppäimet on mahdollista toteuttaa myös ohjelmallisesti (software button), jolloin ne piirretään laitteen näytön alareunaan. Suurinäyttöisissä tableteissa ohjelmapohjaisia näppäimiä käytetään aina.

Home-näppäin

Home-näppäin palauttaa käyttäjän kotinäyttöön ja siirtää näytöllä sillä hetkellä olleen aktiivisen ohjelman taustalle pysähtyneeseen tilaan. Kun kotinäppäin painetaan hetkeksi pohjaan, avautuu multitask-näkymä, jossa käyttäjä voi navigoida viimeksi käytettyjen ohjelmien välillä. Ohjelmista esitetään pieni esikatselukuva siitä näytöstä, jossa käyttäjä viimeksi oli poistuessaan ohjelmasta.

Menu-näppäin

Menu-näppäin avaa ohjelmakohtaisen Menu-valikon, jonne on yleensä sijoitettu ohjelman kannalta tärkeitä toimintoja, joita on päästävää käyttämään helposti. Näppäimen pohjassa pitäminen avaa yleensä hakutoiminnon, mikäli ohjelmasta sellainen löytyy.

Back-näppäin

Back-näppäintä käytetään navigointiin kun käyttäjä haluaa palata takaisin päin ohjelman näkymähierarkiassa. Ohjelmat koostuvat (lähes aina) useista näkymistä, jotka esittävät tietoa käyttäjälle ja/tai sisältävät käyttöliittymän. Eri toiminnot vaativat uusia näkymiä, joita ohjelma avaa tarvittaessa. Uuden näkymän avautuessa vanha näkymä siirtyy ohjelman taustalla olevan abstraktin ”näkymäpinon” päällimmäiseksi. Kun Back-näppäintä painetaan, nyt aktiivisena oleva näkymä suljetaan ja pinossa päällimmäisenä oleva näkymä avautuu uudelleen. Tätä pinoa puretaan niin kauan, kunnes viimeisen näkymän ja Back-näppäimen painalluksen myötä ohjelma sammuu.

Graafinen käyttöliittymä

Androidin graafinen käyttöliittymä muodostuu seuraavista näytöistä: kotinäytöstä, ilmoitusalueesta ja sovellusvalikosta. Graafinen tyyli ja toiminnot vaihtelevat eri valmistajien ja mallien välillä, mutta perustoiminta on samanlaista laitteesta riippumatta. Seuraavissa kuvissa esimerkkinä käytetään Androidin versiota 4.0 ja Samsungin TouchWiz-käyttöliittymää.

Kotinäyttö

Androidin kotinäyttö toimii lähtöpisteenä laitteen käytölle ja siihen palaaminen onnistuu aina Koti-näppäimellä. Näille näytöille käyttäjä voi asettaa pikakuvakkeita usein käytettyihin ohjelmiin tai widgettejä, eli pienoishjelmiä. Kotinäyttöjä voi olla useampi rinnakkain ja niiden välillä liikutaan horisontaalisilla pyyhkäisyillä.



Kuva 1. Androidin kotinäyttö.

Widgeteillä voidaan ohjata ohjelmia, muuttaa puhelimen asetuksia nopeasti (esim. kytkeä radioita päälle ja pois) tai esittää tietoa. Kuvassa 1 näkyy Fancy Widget -niminen pienoisojelma, joka näyttää kellonajan ja säätilan laitteen sijainnin perusteella.

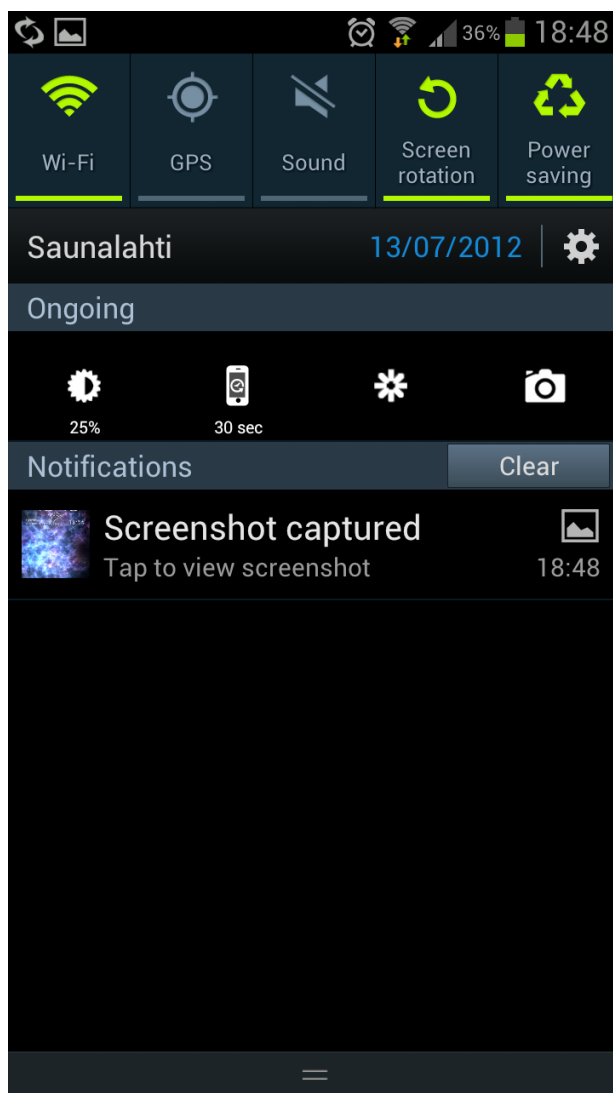
Kotinäytön alareunassa sijaitsee telakka, josta yleensä löytyvät tärkeimpien ohjelmien pikakuvakkeet. Tässä tapauksessa puhelin, osoitekirja, tekstiviestit ja sähköposti. Keskimmäisenä on painike, joka avaa sovellusvalikon.

Näytön yläreunassa on ilmoituspalkki. Koska se ei varsinaisesti kuulu kotinäyttöön, tarkastellaan sitä tarkemmin seuraavassa kappaleessa.

Kotinäytön taustalla näkyy taustakuva, joka voi olla staattinen tai animoitu kuva (Live Wallpaper).

Ilmoituspalkki ja -alue

Ilmoituspalkki esittää tärkeää tietoa järjestelmän ja ohjelmien tilasta. Ilmoituspalkin oikealla puolella näkyy järjestelmään liittyvää tietoa kuten kellonaika, akun varaus, sekä matkapuhelin- ja WLAN-verkkojen vahvuudet. Vasemmalle puolelle ilmestyy ohjelmien toiminnasta kertovia kuvakkeita.



Kuva 2. Ilmoituspalkki avattuna eli ilmoitusalue.

Ilmoituspalkki avautuu kuvan 2 mukaisesti koko näytön kokoiseksi ilmoitusalueeksi pyyhkäisemällä alaspäin näytön yläreunasta. Ilmoitusalueen yläosasta löytyy pikavalintoja tärkeisiin asetuksiin, kuten Wi-Fi- ja GPS-yhteyksien käynnistämiseen ja

puhelimien mykistämiseen. Valikko rullautuu sivusuunnassa ja osa valinnoista on piilossa näytön ulkopuolella. Nämä pikavalinnat ovat laitteen valmistajan oma lisäys käyttöliittymään, eikä niitä löydy Androidin perusversiosta.

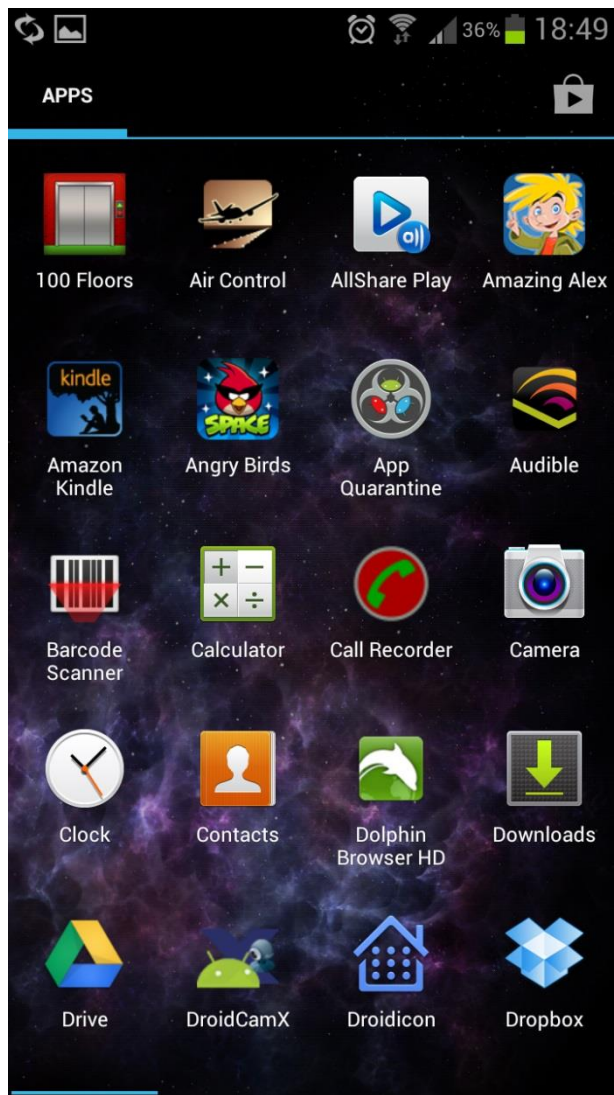
Pikavalintojen alla olevan rivin vasemmalla puolella näkyy, minkä matkapuhelinoperaattorin verkkoa puhelin käyttää. Oikealla puolella näkyvät päivämäärä sekä pikakuvake puhelimen asetuksiin.

Seuraavana on kolmannen osapuolen kehittämä sovellus, joka ei vakiona kuulu ilmoitusvalikkoon, mutta se on hyvä esimerkki Androidin muokattavuudesta. Se antaa käyttäjän lisätä omia pikavalintoja ja asetuksia ilmoitusalueelle, jolloin niitä pääsee muuttamaan nopeasti ilman Asetukset-valikon selaamista. Tässä tapauksessa pikavalintoina on näytön kirkkauden asteittainen säätö, näytön sammumisen aikaviiveen säätö, kytkin kameran salaman käynnistämiseen, joka toimii taskulamppuna, sekä pikakuvake kameran käynnistämiseen.

Alimpana on varsinainen ilmoitusalue, jonne ohjelmien uudet ilmoitukset, kuten saapuneet sähköposti- ja tekstiviestit, ilmestyvät ja jossa niitä voi esikatsella. Kun ilmoitusta painaa, yleensä ilmoituksen tehnyt ohjelma avautuu siihen näyttöön, jossa ilmoitukseen voi reagoida tarkemmin.

Sovellusvalikko

Sovellusvalikossa näkyvät laitteeseen asennetut sovellukset aakkosjärjestyksessä. Valikko rullautuu yleensä sivusuunnassa, mikäli kaikki sovellukset eivät mahdu näytölle kerralla. Kuvan 3 alareunassa näkyvä sininen nelikulmio näyttää, että sovellusvalikko jatkuu vielä ainakin kolmen sivun verran.



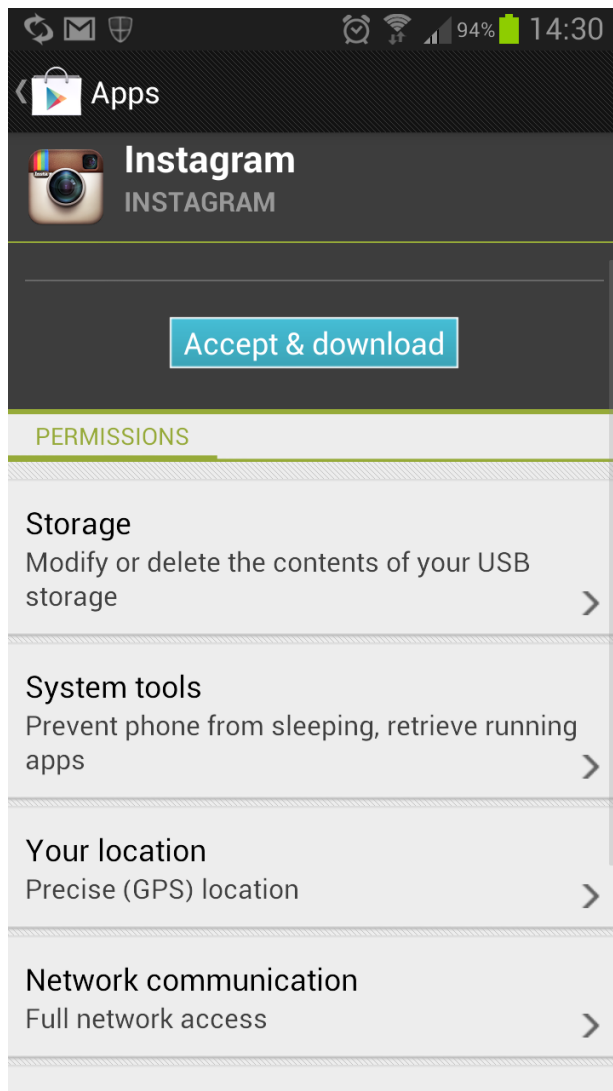
Kuva 3. Sovellusvalikko.

2.3 Tietoturva

Turvallisuussyistä Android-sovellukset ajetaan ns. hiekkalaatikossa ja oletuksena niillä ei ole oikeuksia tehdä mitään, jolla olisi vaikutusta käyttöjärjestelmän, käyttäjän tai muiden ohjelmien toimintaan. Tämän vuoksi ohjelman kehittäjä on määritellyt AndroidManifest-tiedostossa mitä lisäoikeuksia ohjelma tarvitsee toimiakseen oikealla tavalla. Näitä lisäoikeuksia ovat esimerkiksi:

- lupa käyttää Wi-Fi-yhteyttä ja laitteen muita radioita
- pääsy osoitekirjaan (sisältää tietoa ihmisistä, jota ohjelmat voivat käyttää eri tavoin)
- luku/kirjoitus-oikeus ulkoiseen muistikorttiin. /42/, /47/

Ohjelman tarvitsemat oikeudet on syytä määrittellä tarkasti, koska järjestelmä antaa ohjelman toimia ainoastaan niiden oikeuksien rajoissa mitkä xml-tiedostossa on määriteltä. Nämä oikeudet esitetään käyttäjälle ennen ohjelman asennusta kuvan 4 mukaan.



Kuva 4. Ohjelman tarvitsemat lisäoikeudet esitetään käyttäjälle ennen kuin ohjelma ladataan ja asennetaan Play-kaupasta.

Käyttäjä hyväksyy ohjelman vaatimat oikeudet ainoastaan kerran ja jatkossa ohjelma on vapaa toimimaan oikeuksiansa mukaan. Tällä on haettu sujuvaa käyttökokemusta, jossa käyttäjää ei jatkuvasti häiritä esiin ponnahtavilla tietoturvailmoituksilla. Vastakkaisena esimerkkinä toimikoon Windowsien UAC-turvaominaisuus, joka myöntää ohjelmalle korotetut oikeudet ainoastaan hetkeksi. Tämä saattaa aiheuttaa sen, että käyttäjä joutuu vähän väliä hyväksymään ohjelman lisäoikeuspyyntöjä joko nappia painamalla tai hankalasti salasanaansa kirjoittamalla. /47/

2.4 Versiohistoria

Androidin virallinen kehitys tapahtuu Android Open Source Project (AOSP) nimen alla. Lähdekoodi on Apache lisenssi 2.0 alaista, eli se on vapaasti saatavilla, jolloin kuka tahansa voi muokata Androidista omiin tarpeisiinsa sopivan version. /37/

Vaikka lähdekoodi onkin avointa, Android-laitteiden on täytettävä tietyt yhteensopivuusstandardit, jotta niillä olisi oikeus käyttää virallista Android tuotemerkkiä ja päästäkseen osaksi Android ekosysteemiä, eli lähinnä Google Play sovelluskauppaa. Vaatimukset on määritelty Android Compatibility Definition dokumentissa. Vaatimuksilla pyritään varmistamaan sovelluskaupasta ladattujen sovellusten toimivuus kaikilla laitteilla. /22/

Seuraavassa taulukossa näkyy Androidin versiohistoria ja kuinka järjestelmä on kehittynyt vuosien saatossa ja saanut uusia ominaisuuksia. Versionumeron jälkeen suluissa on version "koodinimi". Huhtikuusta 2009 lähtien jokainen Androidin uusi versio on nimetty jälkiruuan tai makeisten mukaan. Koodinimet etenevät aakkosjärjestyksessä. /15/

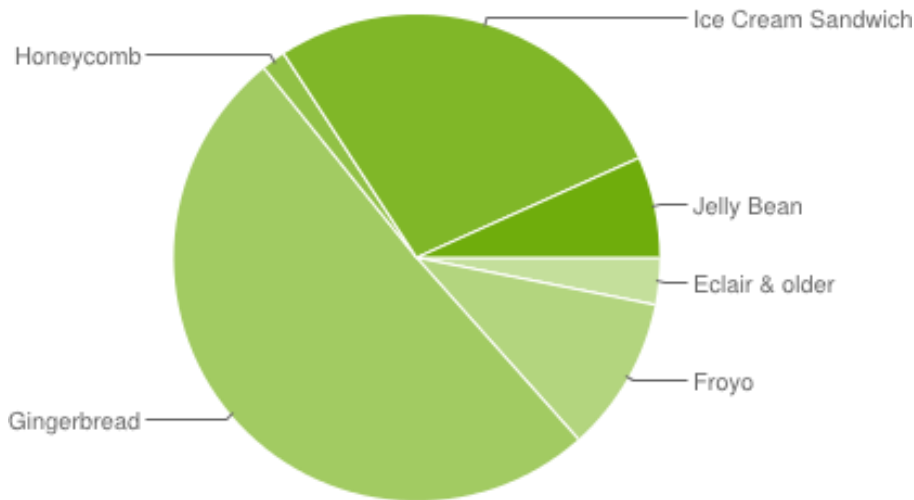
Taulukko 2. Androidin versiohistoria. /15/

Versio	Julkaisu-päivä	Ominaisuuksia
1.0	23.9.2008	<ul style="list-style-type: none"> • Android Market -sovelluskauppa • Internet-selain • Yksinkertainen kameratuki • Gmail- ja sähköpostipalvelintuki (POP3, IMAP4, SMTP) • Googlen perus internetpalvelut, kuten Yhteystiedot, Kalenteri ja Kartat • Yksinkertainen mediasoitin ja YouTube-sovellus
1.6 (Donut)	15.9.2009	<ul style="list-style-type: none"> • Puhesyntetisaattori (tietokoneääni lukee tekstiä) • Android Market: parannettu haku ja kehittäjien

		mahdollista lisätä kuvakaappauksia sovelluksista
2.0/2.1 (Éclair)	26.10.2009	<ul style="list-style-type: none"> • Parannettu tuki tilien synkronoinnille • Exchange-sähköpostituki • Bluetooth 2.1 tuki • Lisäominaisuuksia kameraan, kuten tuki salamalle, digitaalinen zoom, makrokuvaus ja erilaisia säätömahdollisuuksia • Kehittyneempi virtuaalinäppäimistö sanastolla ja ennakoivalla tekstinsyötöllä • Parannuksia internet-selaimeen: uusi käyttöliittymä ja HTML5-tuki • Live taustakuvat: animoituja taustakuvia kotinäytölle
2.2 (Froyo)	20.5.2010	<ul style="list-style-type: none"> • Suorituskyvyn optimointia • Wi-Fi- ja USB-tethering, eli mahdollisuus jakaa puhelimen 3G-verkkoyhteys muiden laitteiden kanssa joko langattomasti Wi-Fi-yhteydellä tai USB-kaapelin avulla • Android Market: automaattinen sovellusten päivitys • Adobe Flash tuki: mahdollista toistaa nettisivuilla hyvin yleisesti käytettyjä flash-videoita. • Tuki korkearesoluutioisille näytöille
2.3 (Gingerbread)	6.12.2010	<ul style="list-style-type: none"> • Uusi yksinkertaistetumpi ja nopeampi käyttöliittymä • Parempi virranhallinta sovellusten osalta • Tuki isoille näytöille ja resoluutioille • VoIP tuki: puhelut internetin välityksellä • NFC tuki: mahdollisuus lukea NFC-tunnisteita

3.0 (Honeycomb)	22.2.2011	<ul style="list-style-type: none"> • Ensimmäinen tableteille suunniteltu ja optimoitu Android, ei saatavilla matkapuhelimille • Uudelleen suunniteltu graafinen käyttöliittymä • Uudistettu moniajovalikko • Tuki moniydinprosessoreille
3.1	10.5.2011	<ul style="list-style-type: none"> • Parannuksia käyttöliittymään • Tuki USB-lisälaitteille (esim. näppäimistöt, hiiret ja peliohjaimet) • Widgettien koko säädettävissä
4.0 (Ice Cream Sandwich)	19.10.2011	<ul style="list-style-type: none"> • Tuo version 3.0 ominaisuudet matkapuhelimiin ja korvaa edellisen, vain tableteille tarkoitettun, version • Ohjelmistopohjaiset valikkonäppäimet • Näytön lukituksen avaaminen kasvotunnistuksella • Uusi internet-selain • Android Beam: NFC-pohjainen tiedonsiirtojärjestelmä, jossa kahta laitetta koskettamalla muodostetaan yhteys laitteiden välille tiedonsiirtoa varten • 1080p, eli ns. Full HD -tasoinen videokuvaus • Adobe Flash -tuki loppuu
4.1 (Jelly Bean)	9.7.2012	<ul style="list-style-type: none"> • Merkittäviä parannuksia laitteen suorituskykyyn ja erityisesti nopeuteen • Kehittyneempi ilmoitusalue

Kuvassa 5 näkyvät eri Android versioiden markkinaosuudet joulukuussa 2012. /48/



Kuva 5. Androidin eri versioiden suhteelliset markkinaosuudet joulukuussa 2012. /48/

2.5 Ongelmia

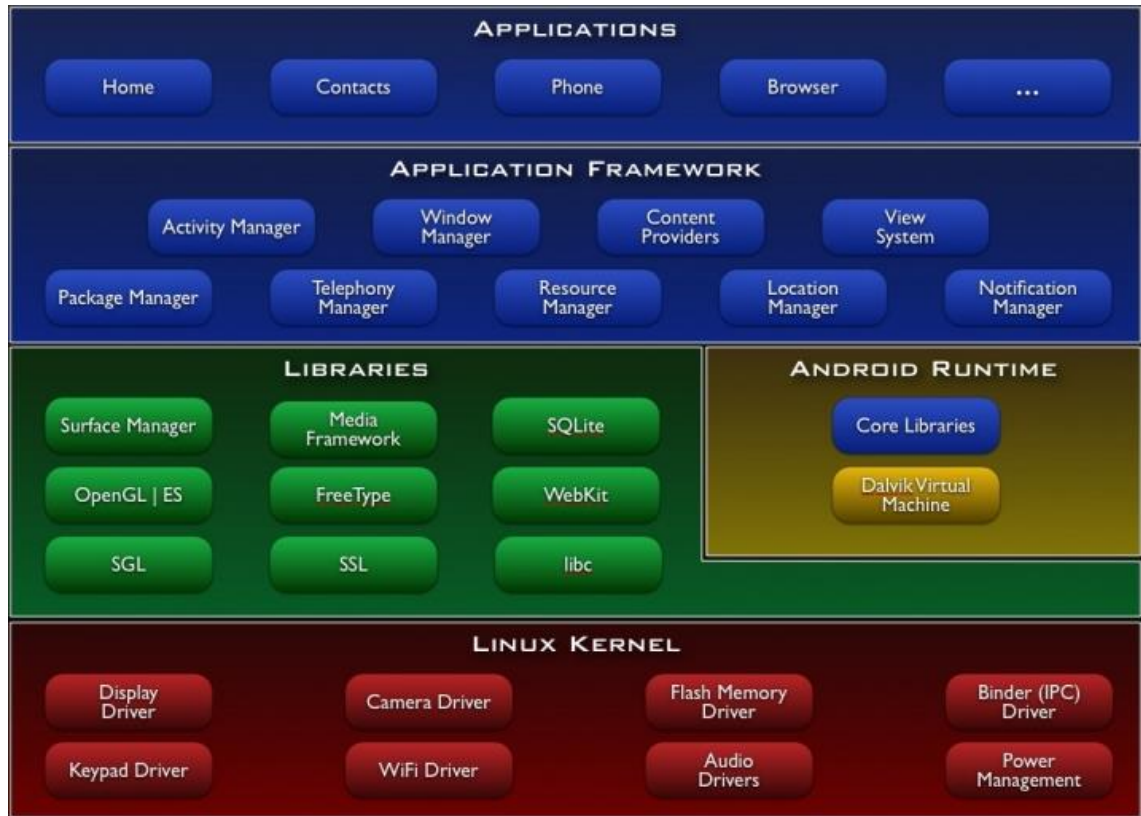
Androidin avoimuus ja nopea kehitys aiheuttavat myös ongelmia. Harva laitevalmistaja tyytyy varustamaan puhelimensa Androidin perusversiolla, vaan vähintäänkin kustomoi puhelimen käyttöliittymän omanlaisekseen erottuakseen muista valmistajista. Tämä aiheuttaa sen, että päivitysten saaminen puhelimelle hidastuu, koska valmistaja joutuu erikseen muokkaamaan suuremmat Googlen julkaisemat viralliset Android-päivitykset omalle laitteelleen sopiviksi. Pahimmassa tapauksessa päivitystuki lakkaa kokonaan, koska valmistaja ei halua tuhjata resurssejaan ”vanhojen” puhelinten ylläpitämiseen. /23/

Puhutaan sirpaloitumisesta, joka tarkoittaa, että markkinat ovat täynnä erilaisia laitteita, eri Android versioilla. Tästä aiheutuu ongelmia sekä ohjelmistokehittäjille että normaaleille käyttäjille. Kehittäjä ei voi optimoida ohjelmiaan toimimaan jokaisella laitteella, joka voi aiheuttaa ohjelmien toimimattomuutta tai suorituskykyongelmia. Käyttäjän kannalta taas uuden puhelimen hankkiminen voi olla riskialtista, koska päivitysten saaminen tulevaisuudessa on epävarmaa. Edellisen kappaleen kuvasta 5 selviää, että noin puolet Android-laitteista käyttää yli kaksi vuotta vanhaa 2.3 versiota. /57/

Vanhemmat Android-versiot aiheuttavat myös suorituskykyongelmia jopa kalliimman hintaluokan puhelimissa.

3 ANDROIDIN ARKKITEHTUURI

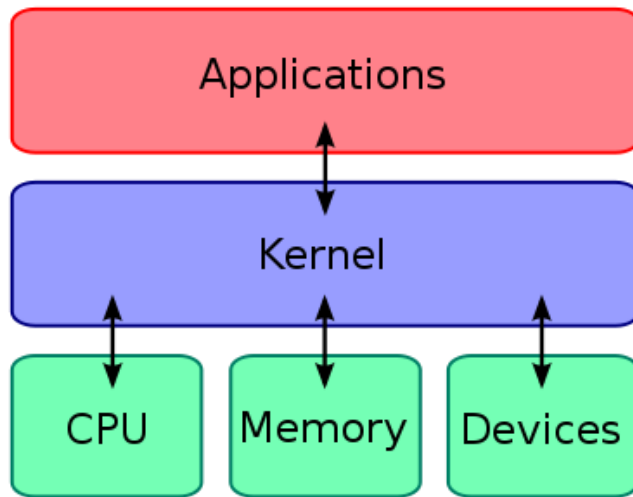
Tässä kappaleessa käydään tarkemmin läpi mistä Android-ohjelmistopino rakentuu tarkastelemalla kuvassa 6 esiintyviä järjestelmän komponentteja.



Kuva 6. Androidin arkkitehtuuri. /11/

3.1 Linux kernel

Android perustuu Linuxin kernelin versio 2.6:een. Kerneli on käyttöjärjestelmän ydin, joka kuvan 7 mukaisesti yhdistää tietokoneella ajettavat ohjelmat (software) tietokoneen laitteistoon eli “rautaan” (hardware). /25/



Kuva 7. Kernelin suhde ohjelmistoihin (Applications) ja laitteistoon (CPU, Memory, Devices). /33/

Androidin tapauksessa kerneli sisältää muistin- ja virranhallintaan liittyvät ohjelmat, sekä laitteen perustoiminnan kannalta tärkeitä laitteistoajureita (device driver). Ajuri on pieni ohjelma, joka kertoo käyttöjärjestelmälle kuinka se kommunikoi laitteiston kanssa. /25/, /33/

3.2 Libraries - kirjastot

Kirjasto on itsenäinen kokoelma koodia tai tietoa, jota ohjelmat voivat käyttää. Kirjasto voi esimerkiksi kertoa laitteella kuinka se käsittelee tietynlaista tietoa. Käydään seuraavaksi läpi Androidin peruskirjastot taulukossa 3. /36/

Taulukko 3. Kirjastot. /2/, /4/, /8/, /20/, /34/, /44/, /59/, /62/, /63/, /64/

Surface Manager	Vastaa ikkunoiden esittämisestä laitteen näytöllä.
OpenGL ES ja SGL	Grafiikkakirjastot kaksi- (2D) ja kolmiulotteisen (3D) grafiikan piirtämistä varten. Grafiikalla tarkoitetaan tässä tapauksessa kaikkea mitä laitteen näytöllä näkyy. Suurin osa ohjelmista käyttää kaksiulotteista grafiikkaa (käyttöliittymä, valikot jne.) ja jotkin kehittyneemmät ohjelmat, kuten pelit, voivat käyttää 3D-grafiikkaa.
Media Framework	Sisältää tarvittavat kirjastot ja koodekit median (kuva, video, ääni) toistamista varten. Koodekki on ohjelma, joka pakkaa

	ja purkaa kuva- tai äänisignaalia. Yksinkertaisesti sanottuna, se mahdollistaa erilaisten mediatiedostojen avaamisen.
FreeType	Käsittelee fontteja.
SSL	Secure Socket Layer, nykyiseltä nimeltään TLS (Transport Layer Security), on salausprotokolla tietoliikenteen suojaamiseen internetissä.
SQLite	Tietokannanhallintajärjestelmä, jota Android käyttää tietojen tallentamiseen. Koko järjestelmä on tallennettu yhteen, kooltaan hyvin pieneen C-kirjastoon, joka tekee järjestelmästä hyvin kevyen. Tämän vuoksi SQLite onkin suosittu sulautetuissa järjestelmissä.
WebKit	Avoimen lähdekoodin selainmoottori, jonka avulla internetselain muodostaa web-sivun lähdekoodin perusteella.
Core Libraries	Java-ohjelmointikielen peruskirjastot, joita tarvitaan, koska Android-sovellukset kirjoitetaan Javalla.
Dalvik Virtual Machine	Virtuaalikone, jonka sisällä kaikki Android-sovellukset ajetaan. Virtuaalikone on ”ohjelmistopohjainen tietokone”, jota ajetaan toisen tietokoneen sisällä. Virtuaalikone voi emuloida tietynlaista laitteistoa tai esimerkiksi käyttöjärjestelmää.

3.3 Application Framework - ohjelmistokehys

Ohjelmistokehys on ”perusta”, jonka päälle ohjelma rakennetaan. Kehys voi sisältää valmiita ohjelman osia tai toiminnallisuuksia, tai työkaluja, jopa kehitysympäristön, ohjelmien kehitykseen. Androidin käyttämän ohjelmistokehysten yksityiskohdat löytyvät taulukosta 4. /54/

Taulukko 4. Ohjelmistokehys. /8/, /10/, /40/, /41/, /43/, /46/, /51/, /55/, /61/

Activity Manager	Huolehtii ohjelman tärkeimpien rakennuspalikoiden, aktiviteettien, elinkaarista. Tähän kuuluu: <ul style="list-style-type: none"> aktiviteetin käynnistyminen ja tarvittavien resurssien (muisti) varaaminen
------------------	---

	<ul style="list-style-type: none"> • varatun muistin hallinta, jota tarvittaessa vapautetaan muuhun käyttöön • aktiviteetin sisältämän tiedon hallinta • aktiviteetin sulkeminen ja resurssien vapauttaminen. <p>Asiaa käsitellään tarkemmin kappaleessa 4.1 Activity.</p>
Package Manager	Kokoelma työkaluja, jotka automaattisesti asentavat tai poistavat ohjelmia, ja pitävät kirjaa asennetuista ohjelmista ja niiden riippuvuuksista.
Telephony Manager	Tarjoaa ohjelmille tietoa puhelimen eri tiloista: onko se kytkeytyneenä matkapuhelinverkkoon, verkon tyyppin (esim. EDGE, HSPA, LTE), teleoperaattorin nimen ja SIM-kortin tilan (esim. onko kortti avattu PIN-koodilla).
Content Providers	Tekniikka, jolla Android-sovellukset jakavat tietoa keskenään.
Resource Manager	Valitsee ohjelmalle sopivan asettelun (layout) ja grafiikat ohjelman resursseista. Jotta ohjelma näyttäisi hyvältä sekä pienen matkapuhelimen että suuremman tabletin näytöllä, ohjelman käyttämiin resursseihin määritellään erityyppisiä asetteluja ja erikokoisia grafiikoita erilaisia ja kokoisia näyttöjä varten.
View System	Sisältää komponentteja sovellusten käyttöliittymien rakentamiseen, kuten painikkeita, tekstilaatikoita, merkkijonona jne.
Location Manager	Pitää kirjaa laitteen sijainnista, jota ohjelmat voivat käyttää eri tavoin. Sijainti määritellään GPS:n, Wi-Fi:n tai matkapuhelinverkon maston, johon laite on kirjautunut, perusteella. Satelliittipohjainen GPS on tarkin ja mahdollistaa laitteen sijainnin määrittämisen muutaman metrin tarkkuudella. Wi-Fi:n ja maston perusteella paikantaminen antavat laitteelle summittaisen sijainnin satojen metrien tarkkuudella.
Notification Manager	Ilmoittaa käyttäjälle ilmoituksella, värinällä tai valoilla kun jotain tärkeää on tapahtunut. Ilmoitus voi tulla esimerkiksi

	saapuneesta sähköpostiviestistä, kalenteriin asetetusta tapahtumasta tai ohjelmalta, joka vaatii käyttäjän huomiota.
--	--

3.4 Applications - sovellukset

Korkeimmalla tasolla ovat tavalliset sovellukset. Puhelin sisältää vakiona perussovelluksia, kuten kotinäytön, yhteystiedot, sekä sovellukset puheluiden soittamisen ja tekstiviestien lähettämiseen.

Käyttöjärjestelmä ei kuitenkaan tee eroa eri sovellusten (varusohjelmat vs. kolmannen osapuolen kehittämät sovellukset) välillä, mikä on eräs Androidin vahvuuksista. Tämä tarkoittaa sitä, että käyttäjä voi halutessaan korvata minkä tahansa sovelluksen toisella vastaavalla. Puhelimen varusohjelmat ovat yleensä ominaisuuksiltaan vaatimattomia, joten tehokäyttäjät saattaa haluta vaihtaa ne monipuolisempiin vaihtoehtoihin. /14/

Android-sovellusten virallinen jakelu- ja myyntikanava on Google Play -kauppa (entiseltä nimeltään Android Market). Suomessa Play-kaupasta on toistaiseksi saatavilla vain Android-sovelluksia, mutta mm. Yhdysvalloissa kaupasta saa ostettua myös e-kirjoja, elokuvia, tv-sarjoja ja musiikkia. Vuoden 2012 viimeisellä neljänneksellä, Playn valikoima koostui 675 000 Android-sovelluksesta. /27/, /28/

4 OHJELMAN RAKENNE

Android-sovellukset kirjoitetaan Java-kielellä. Lähdekoodi käännetään kehitysympäristössä ja pakataan yhteen APK-tiedostoon, josta ohjelma asentuu laitteeseen. APK-tiedosto sisältää mm. käännetyt Java-luokkatiedostot, jotka sisältävät sovelluksen varsinaisen koodin, resurssit (esim. sovelluksen käyttämät grafiikka- ja äänitiedostot) sekä AndroidManifest.xml -tiedoston. /12/

AndroidManifest.xml on eräänlainen yhteenveto sovelluksen sisällöstä ja sen vaatimuksista, jota järjestelmä tarvitsee ennen kuin sovellus voidaan asentaa. Tiedostoon on merkitty esimerkiksi sovelluksen sisältämät ja käyttämät aktiviteetit (joista lisää tietoa seuraavassa kappaleessa), sovelluksen tarvitsemat lisäoikeudet ja vähimmäis-Android API -taso, eli Androidin versio, jonka sovellus vaatii. /56/

Android-sovellukset rakentuvat neljästä komponentista:

- Activity
- Service
- Content Providers
- Broadcast Receivers. /12/

Activity, jatkossa aktiviteetti, on näistä neljästä tärkein, koska jokainen ohjelma, jossa on graafinen käyttöliittymä, rakentuu sellaisista. Muita kolmea komponenttia käytetään ohjelman erityistarpeiden mukaan. Myös tämän opinnäytetyön kannalta aktiviteetti on tärkein osa-alue, joten se käsitellään seuraavissa kappaleissa muita tarkemmin. /12/

4.1 Activity

Android-sovellukset rakentuvat aktiviteeteista. Yksinkertaisesti esitettynä, yksi aktiviteetti on se mitä käyttäjä juuri sillä hetkellä näkee laitteensa näytöllä. Aktiviteetti voi olla koko näytön kokoinen ikkuna täydellisellä graafisella käyttöliittymällä, tai pieni ponnahdusikkuna, joka odottaa käyttäjältä vahvistusta johonkin. Kun sovellus

käynnistetään, avautuu *main*-aktiviteetiksi määritelty aktiviteetti, joka näyttyy käyttäjälle ensimmäisenä. /3/

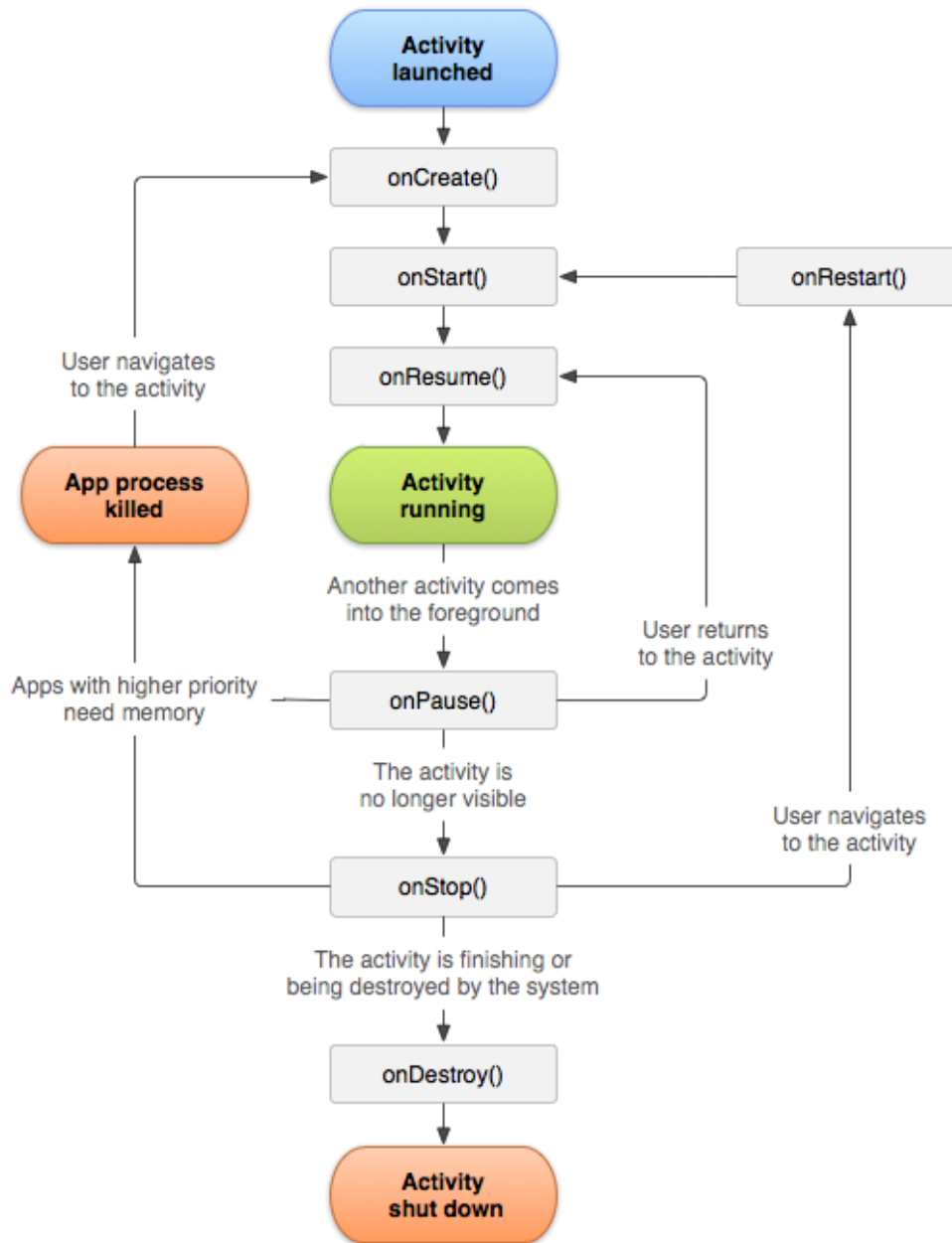
Aktiviteetti voi käynnistää uusia aktiviteetteja, jotka suorittavat muita toimintoja eli laitteen näytölle avautuu uusia näkymiä. Kun uusi aktiviteetti avautuu näytölle, sitä edeltävä aktiviteetti siirtyy taustalla olevan aktiviteettipinon (back stack) päällimmäiseksi joko Paused- tai Stopped-tilaan. Kun käyttäjä painaa laitteen Back-painiketta, näytöllä oleva aktiivinen aktiviteetti suljetaan lopullisesti ja taustalla olevasta pinosta nostetaan esiin ”pinon päällimmäisenä oleva” aktiviteetti. /3/

Aktiviteetilla on kolme erilaista tilaa, jossa se voi olla:

- *Resumed*: aktiviteetti on käynnissä ja aktiivisena näytöllä vieden koko näytön pinta-alan.
- *Paused*: aktiviteetti on taustalla pysähtyneessä tilassa silloin kun toinen aktiviteetti on käynnissä, mutta ei vie koko näyttöalaa. Aktiviteetti ja sen sisältämä tieto on tallennettu ja pyritään säilyttämään muistissa, koska on todennäköistä, että käyttäjä palaa tähän aktiviteettiin (eli näkymään) pian. Aktiviteetti saatetaan kuitenkin sulkea jos muisti käy äärimmäisen vähiin.
- *Stopped*: aktiviteetti on taustalla pysähtyneessä tilassa silloin kun toinen aktiviteetti on käynnissä ja vie koko näyttöalan. Aktiviteetti on edelleen ladattuna muistiin, mutta se suljetaan tarvittaessa jos muistia tarvitaan toisaalla.

/3/

Kuvassa 8 näkyvät aktiviteetin eri tilat ja järjestelmäfunctiot. Järjestelmä kutsuu näitä funktioita kuvan mukaisella tavalla aina kun aktiviteetti muuttaa tilaansa.



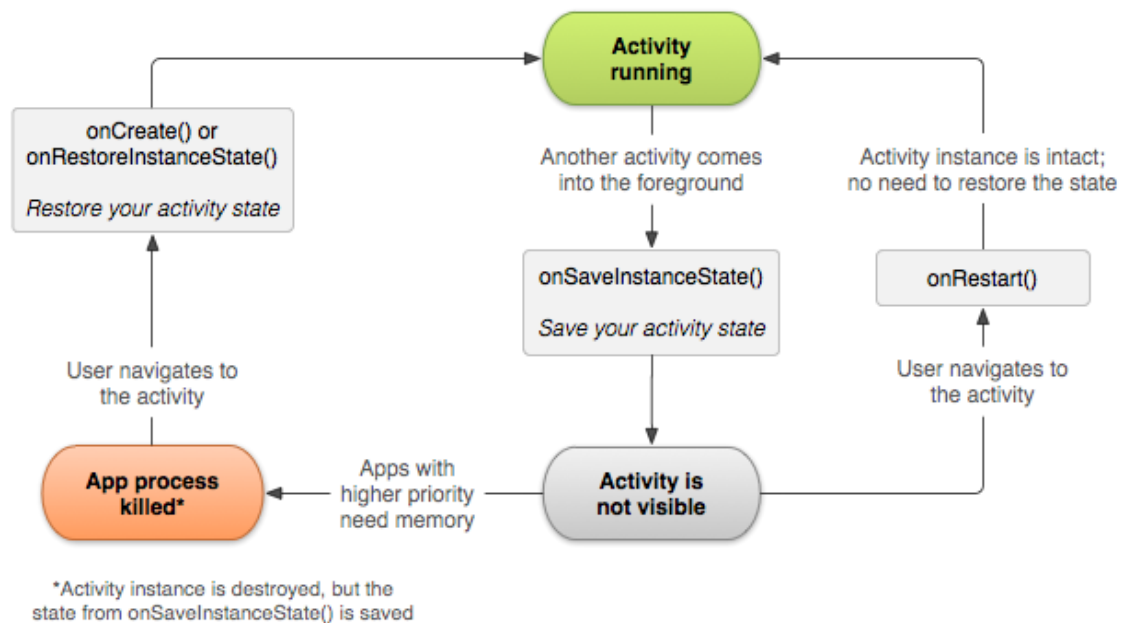
Kuva 8. Aktiviteetin elinkaari. /3/

Muistinhallinta

Aktiviteetit liittyvät vahvasti ohjelmien muistinkäyttöön. Kun ohjelma käynnistetään, se ladataan matkapuhelimen RAM-muistiin. Koska laitteessa on rajattu määrä muistia (tyypillisesti 256 MB – 2 GB) ja älypuhelimissa on tavanomaista ajaa useita sovelluksia samanaikaisesti, muistia ei todennäköisesti suoraan riitä kaikille sitä tarvitseville ohjelmille. Tämän vuoksi järjestelmän on jatkuvasti huolehdittava, että tärkeimmillä eli ajossa olevilla ohjelmilla, on käytössään tarpeeksi muistia. Muistia vapautetaan tarpeen vaatiessa sulkemalla taustalla olevia muita ohjelmia, etenkin sellaisia, joita ei ole

käytetty pitkään aikaan. Myös ajossa olevan ohjelman osia, eli taustalle jääneitä aktiviteetteja voidaan sammuttaa jos muisti käy kriittisen vähiin. /49/

Aktiviteetit säilyvät pinossa vaikka niitä olisi suljettu muistin vapauttamisen vuoksi. Jos käyttäjä palaa aktiviteettiin joka on jo ehditty sulkea, aktiviteetti yksinkertaisesti käynnistetään uudelleen. Mikäli aktiviteetti ei sisältänyt mitään käyttäjän lisäämää tietoa, kuten kirjoitettua tekstiä tekstilaatikoissa, käyttäjä ei edes huomaa, että aktiviteetti suljettiin ja käynnistettiin uudelleen. Toisaalta jos suljettu aktiviteetti sisälsi jotain tietoa, tämä tieto menetettiin siinä vaiheessa kun aktiviteetti siivottiin pois muistista. Tässä tapauksessa esimerkiksi kaikki tekstikentät olisivat tyhjentyneet. Aktiviteetti on kuitenkin mahdollista ohjelmoida tallentamaan tietonsa, jotka ladataan automaattisesti kun aktiviteetti uudelleenkäynnistyy. /52/



Kuva 9. Aktiviteetin sisältämän tiedon palautus jos aktiviteetti suljettiin muistin puutteen vuoksi. /52/

Aktiviteetti suljetaan ja käynnistetään uudelleen myös silloin kun laitteen näytön orientaatio muuttuu. Järjestelmä tuhoaa aktiviteetin ja rakentaa sen uudelleen siltä varalta, että näytölle on saatavina uudet asetustiedostot, esimerkiksi vaihtoehtoinen asetustiedosto, jossa aktiviteetin käyttöliittymä on rakennettu ja optimoitu vaakasuoralle näytölle. /52/

4.2 Service

Palvelu on taustaprosessina pyörivä ohjelman komponentti, joka suorittaa pitkäkestoisia operaatioita. Palvelulla ei ole käyttöliittymää ja oletuksena se jatkaa toimintaansa loputtomasti. Esimerkiksi musiikkisoitin voi toimia palveluna ja soittaa musiikkia taustalla samalla kun käyttäjä tekee jotain muuta toisella, etualalla olevalla, ohjelmalla. /53/

Palvelut ovat melko hyvin turvassa sulkemiselta kun muisti käy vähiin. Yleensä järjestelmä sulkee palveluita vain kun aktiivisena oleva aktiviteetti tarvitsee lisää muistia. Palvelu voidaan myös asettaa toimimaan etualalla (foreground service), mikä antaa sille parhaan mahdollisen suojan sulkemisesta vastaan. Tällöin palvelulle asetetaan ilmoitus ilmoitusalueelle. /53/

Palvelulla on kaksi erilaista muotoa:

- *Started*. Ohjelma (aktiviteetti) kutsuu palvelua suorittamaan jonkin operaation, esimerkiksi lataamaan tiedoston internetistä. Palvelu käynnistyy, suorittaa operaation ja jatkaa käymistä loputtomasti, riippumatta siitä onko sen kutsunut ohjelma enää olemassa. Palvelu on hyvä määrittää sulkemaan itsensä kun se on suorittanut tehtävänsä. /53/
- *Bounded*. Palveluun voidaan "sitoa" ohjelman komponentti (esim. aktiviteetti), jolloin komponentti pystyy olemaan vuorovaikutuksessa palvelun kanssa. Sidottua palvelua käytetään kun halutaan käyttää palvelun toimintoja muista ohjelman komponenteista tai silloin kun halutaan jakaa toiminnallisuutta ohjelmien kesken. /16/

4.3 Content Providers

Content Providereita käytetään tiedon jakamiseen muille ohjelmille, koska tietoturvasyistä ohjelmat eivät pääse suoraan vaikuttamaan toisiinsa. Tiedon jakaminen tapahtuu palvelinmaisesti, jossa asiakas (ohjelma, joka haluaa käyttää toisen ohjelman tietoa) lähettää pyynnön palvelimelle (ohjelmalle, jolla haluttu tieto on). Palvelin, eli

Content Provider, hakee tiedon varsinaisesta lähteestä, esimerkiksi tietokannasta, ja palauttaa sen asiakkaalle. /9/, /18/, /19/

4.4 Broadcast Receiver

”Lähetin-vastaanottimia” käytetään tiedon välittämiseen järjestelmän laajuisesti. Ne koostuvat kahdesta osasta: lähetin (broadcast, suomeksi lähettää) lähettää viestin, johon vastaanotin (receiver, ohjelman komponentti) reagoi. Yleensä nämä viestit ovat ilmoituksia “tärkeistä” tapahtumista järjestelmässä, kuten näytön sammumisesta, puhelimen orientaation muuttumisesta (pystyasennosta vaaka-asentoon tai toisinpäin) tai akun vähäisestä varauksesta. Kun ohjelma on asetettu kuuntelemaan tietynlaisia lähetyksiä, viestin kuullessaan se voi reagoida halutulla tavalla. /6/, /17/

4.5 Asettelut

Asetteluilla (layout) rakennetaan aktiviteetin graafinen käyttöliittymä. Asettelu määrää miten näytölle sijoitettavat käyttöliittymän komponentit (napit, tekstilaatikot jne.) asetuvat. Asettelun valintaan vaikuttavat laitteen näytön koko ja orientaatio. Matkapuhelimen pienen näytön takia käyttöliittymän on oltava hyvin tiivis. Tabletissa taas tilaa on käytettävissä runsaammin, joten kaikki käyttöliittymä on suunniteltava täysin eri tavalla kuin matkapuhelimen tapauksessa. Ohjelmaan on mahdollista luoda samasta näkymästä useita erilaisia versioita eri laitteita varten käyttämällä erilaisia asetteluja. /21/

Erilaisia asetteluja on käytettävissä neljä kappaletta:

- lineaarinen (linear)
- suhteellinen (relative)
- ruudukko (grid)
- lista (list). /35/

Lineaarinen asettelu

Linearisessa asettelussa komponentit asettuvat peräkkäin vaaka- tai pystysuorasti asetellulle annettujen attribuuttien mukaan. Esimerkiksi jos asetellulle on annettu attribuutti "horizontal" ja näytölle asetetaan kaksi nappia, ne asettuvat vaakasuorasti vierekkäin kuvan 10 tapaan. Jos attribuutti taas olisi "vertical", napit olisivat päällekkäin. /38/



Kuva 10. Lineaarinen asettelu. /38/

Pääasettelun sisälle voidaan luodaan "aliasetteluja", joilla kaikilla on omat attribuuttinsa. Kuvassa 10 taustalla oleva suuri, vaaleansininen nelikulmio on horisontaalinen pääasettelu, joka vie laitteen koko näytön tilan. Sen sisällä näkyvät kolme sinistä suorakaidetta ovat aliasetteluja joiden jokaisen sisään voidaan sijoittaa lisää asetteluja tai käyttöliittymän komponentteja. Esimerkiksi jos oletetaan, että nämä aliasettelut olisivat vertikaalisia, niiden sisään asetetut komponentit asettuisivat päällekkäin. Lopputuloksena käyttäjälle siis näkyisi kolme vierekkäistä saraketta, jossa napit yms. olisivat päällekkäin. /38/

Suhteellinen asettelu

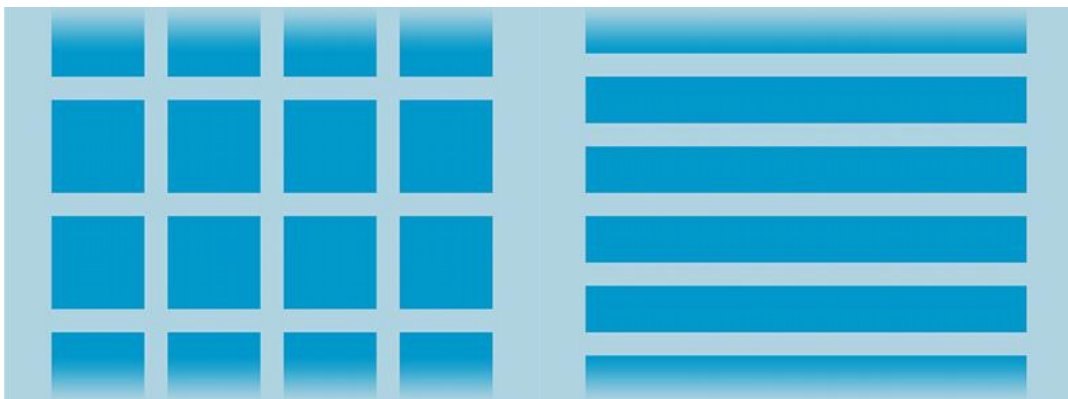
Suhteellisessa asettelussa komponentit asettuvat suhteessa toisiin komponentteihin. Tarkastellaan esimerkkinä kuvaa 11 ja nimetään kuvassa olevat siniset nelikulmiot alkaen suurimmasta pienimpään, N1:ksi, N2:ksi ja N3:ksi. N2:lla on attribuutti "below of N1", joka asettaa sen N1-nelikulmion alapuolelle. N3 taas asettuu N2:den oikealle puolelle attribuutilla "right of N2". /50/



Kuva 11. Suhteellinen asettelu. /50/

Ruudukko ja lista

Ruudukko- ja lista-asettelut nimiensä mukaan asettelevat komponentit joko ruudukoksi tai listaksi. Ruudukkoa voidaan käyttää esimerkiksi näyttämään useita kuvia kerrallaan ja listaa näyttämään merkintöjä tietokannasta. /29/, /39/



Kuva 12. Ruudukko ja lista. /29/, /39/

Kaikkia neljää asettelua voidaan käyttää sekaisin samassa näytössä käyttämällä aliasetteluja. Näin komponentit on mahdollista sijoittaa mahdollisimman tarkasti. On kuitenkin suositeltavaa välttää käyttämästä liikaa aliasetteluja, koska niiden suuri määrä saattaa hidastaa näytön piirtymistä. /35/

5 SOVELLUKSEN SUUNNITTELU

Lintukirjasovellusta käytetään lintujen tietojen tallennukseen ja selaamiseen. Linnusta tallennetaan kuva sekä erilaista tekstipohjaista tietoa, kuten perustiedot, elinalue ja ravinto. Lintujen selaaminen onnistuu joko sovelluksen sisäistä sanahakua käyttämällä, tai selaamalla listaa, josta löytyvät kaikki sovellukseen tallennetut linnut.

Kun haluttu lintu valitaan esimerkiksi listasta, avautuu näkymä, jossa on selkeä kuva linnusta, sekä tiedot. Menu-valikon kautta lintua voidaan muokata tai poistaa merkintä. Lintukirja käyttää paikallista, itse laitteessa sijaitsevaa tietokantaa. Tämä tarkoittaa, että käyttäjä itse huolehtii sisällön keräämisestä ja tallentamisesta sovellusta varten.

Seuraavissa kappaleissa käydään tarkemmin läpi mitä toimintoja sovelluksessa on ja kuinka ne toimivat. Kuvat ovat ehdotuksia miltä ohjelman graafinen käyttöliittymä voisi näyttää matkapuhelimen näytöllä. Koska sovelluksen vaatimusmääritettyä jouduttiin muuttamaan työn aikana, kappaleessa 5.3 käydään vielä läpi minkälainen sovelluksen alun perin piti olla. Kappaleen voi myös ajatella sisältävän kuvauksen sovelluksen edistyneemmistä toiminnoista.

5.1 Graafinen käyttöliittymä

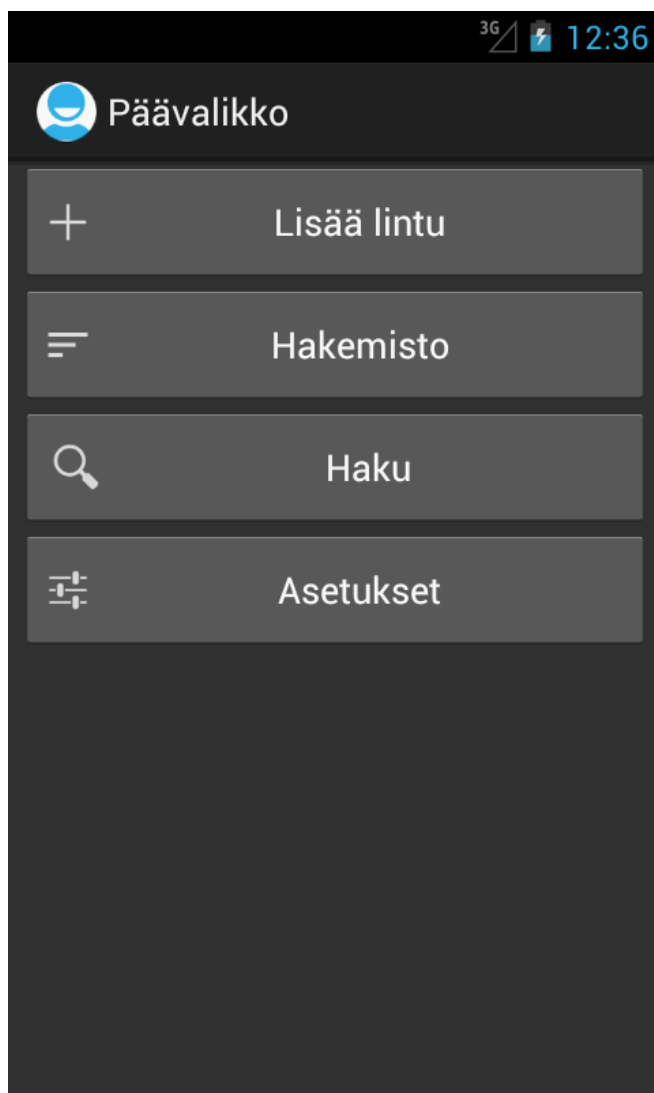
Sovelluksen graafinen käyttöliittymä koostuu useista näkymistä, johon sen toiminnot on jaettu. Tämä selkeyttää sovelluksen käyttöä ja tekee siitä käyttäjäystävällisen. Suunniteltuja toimintoja varten tarvitaan ainakin seuraavat näytöt:

- päävalikko, josta löytyvät painikkeet toisiin näkyymiin siirtymistä varten
- haku
- aakkosellinen hakemisto, eli listanäkymä, jossa näkyvät lintutietokannan kaikki merkinnät
- linnun yksityiskohtaiset tiedot ja kuva
- uuden linnun tietojen syöttäminen.

Lisäksi mahdollinen Asetukset-näkymä, jossa sovelluksen asetuksia, esimerkiksi kieli- ja varmuuskopiointiasetuksia, voidaan muuttaa helposti. Näkymistä palataan edelliseen näkymään laitteen Back-näppäimellä.

Päävalikko

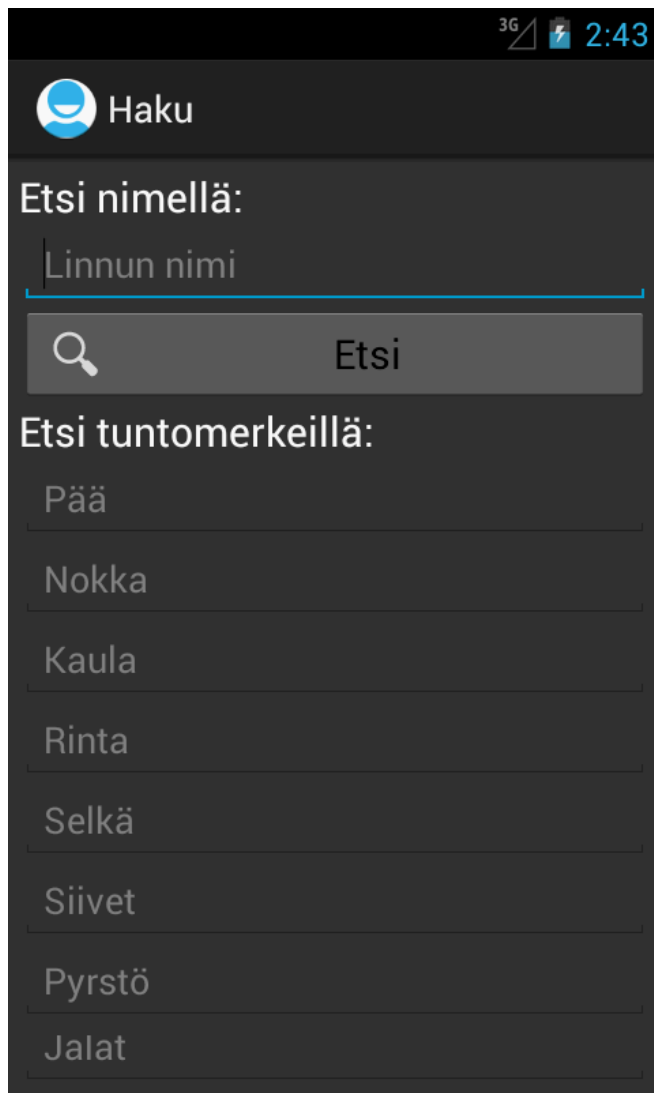
Sovellus luonnollisesti käynnistyy päävalikkoon. Kuvan 13 päävalikosta löytyvät selkeät painikkeet, joilla ohjelman varsinaisia toimintoja päästään käyttämään nopeasti.



Kuva 13. Päävalikko.

Haku

Kuvassa 14 esitetään hakunäkymä, joka koostuu kahdesta osasta. Näytön yläreunassa oleva ”Etsi nimellä”-perushaku etsii tietokannasta linnun nimen mukaisia merkintöjä.

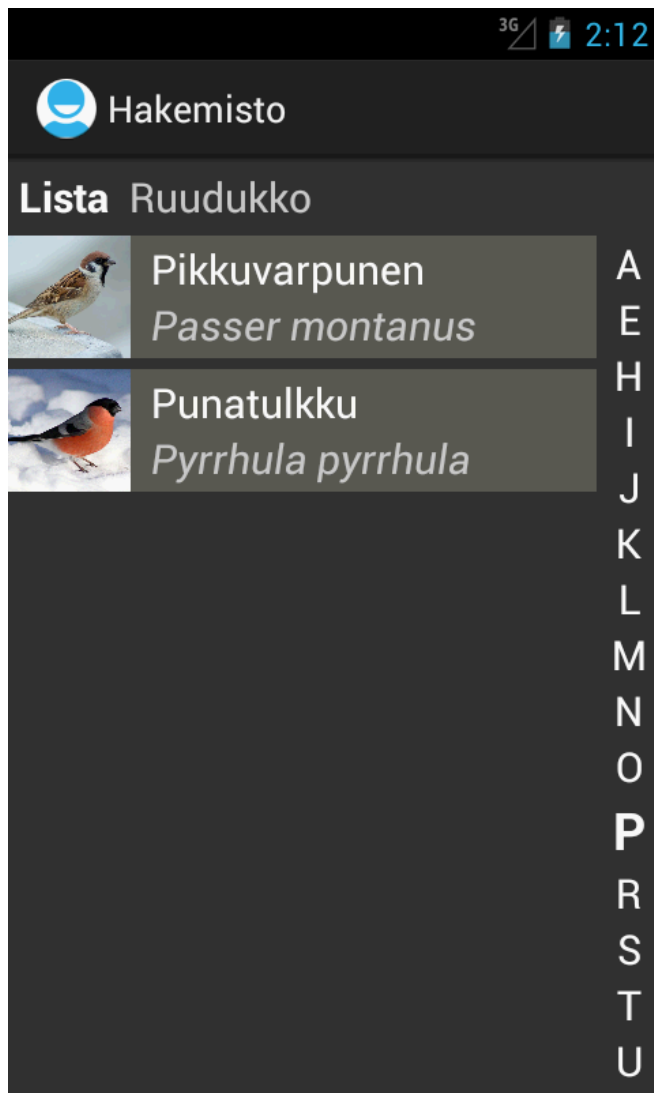


Kuva 14. Hakunäkymä.

Lintujen etsiminen on mahdollista myös linnun tuntomerkkien perusteella käyttämällä ”Etsi tuntomerkeillä”-hakua. Linnun eri ruumiinosista voidaan tallentaa tietokantaan yksinkertaisia, yhden sanan mittaisia, kuvauksia, kuten värejä, muotoja, kuvioita ja niin edelleen. Hakua varten jokaiselle ruumiinosalle on oma tekstilaatikkonsa, johon haettava ominaisuus kirjoitetaan. Ohjelma yhdistää nämä hakusanat ja etsii tietokannasta vain ne merkinnät, jotka sopivat tuntomerkkien yhdistelmään.

Aakkosellinen hakemisto

Aakkosellinen hakemisto on listanäkymä, joka listaa kaikki tietokannassa olevat linnut aakkosjärjestykseen.



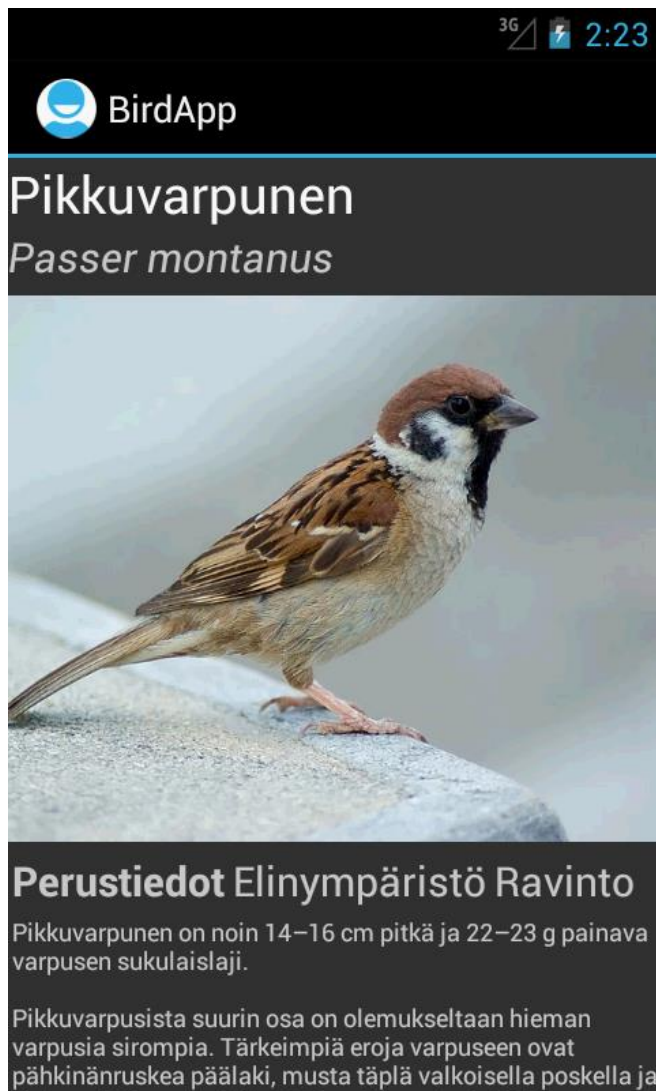
Kuva 15. Aakkosellinen hakemisto.

Kuvan 15 esimerkissä listassa näytetään linnusta pieni esikatselukuva, sekä suomalainen ja tieteellinen nimi. Jotta listan selaaminen olisi helpompaa, näkymässä on erilaisia valintoja, joilla lista voidaan järjestellä eri tavoin, esimerkiksi suomalaisen tai tieteellisen nimen alkukirjaimen mukaan. Tiettyyn alkukirjaimeen hyppääminen onnistuu oikeassa reunassa olevasta valitsimesta. Lista voidaan esittää myös ruudukkonäkymässä, jolloin linnusta näytetään ainoastaan kuva.

Listanäkymän merkintää painamalla avautuu näkymä (jota käsitellään tarkemmin seuraavassa kappaleessa), jossa esitetään linnun yksityiskohtaiset tiedot ja isompi kuva. Kun merkintää painetaan pidempään, näytölle avautuu pieni ponnahtusikkuna, josta merkintä voidaan poistaa tai sitä voidaan muokata.

Lintunäkymä

Kuvan 16 mukainen lintunäkymä avautuu, kun tietty lintu valitaan listanäkymästä tai haun listauksesta.



Kuva 16. Linnun kuva ja yksityiskohtaiset tiedot jaoteltuna välilehtiin.

Näytön ylälaudassa näkyvät linnun nimi suomeksi ja latinaksi. Nimien alapuolella on oletuskuva, jota painamalla kuva avautuu koko näytön kokoiseksi. Jos linnusta on

useampia kuvia, niiden välillä liikutaan sivuttaispyyhkäisemällä kuvaa. Tätä varten kuvassa tulisi olla jonkinlainen merkki, että kuvia on saatavilla useampia.

Linnun kuvan alapuolella on välilehtinäköymä, joka rullautuu sivusuunnassa jos kaikki välilehdet eivät mahdu näytölle. Välilehden sisältämä tieto näkyy sen alapuolella olevassa tekstiosiossa, joka rullautuu pystysuunnassa.

Menu-näppäintä painamalla avataan työkaluvalikko, jossa on ainakin Muokkaa-komento, jolla linnun tietoja on mahdollista muokata. Muokkaus-toiminnon käyttö muuttaa näkymän samantyylliseksi mitä seuraavassa ”Lisää uusi lintu”-kappaleessa käsitellään. Näytöstä poistuminen takaisin aakkoselliseen hakemistoon onnistuu laitteen Back-näppäimellä.

Lisää uusi lintu

Lisää uusi lintu -näköymä on rakenteeltaan samanlainen kuin edellisen kappaleen kuvan 16 lintunäköymä. Kohdat, joissa aiemmin näkyi tekstiä, on tässä näköymässä korvattu tekstikentillä, johon käyttäjä itse lisää haluamansa tiedot. Tekstikentässä näkyvä vaalealla värillä kirjoitettu vihje kertoo mitä kenttään on tarkoitus kirjoittaa. Näköymän lopusta löytyy Tallenna-painike, joka tallentaa merkinnän tietokantaan.

5.2 Tietokanta

Tietokanta on hyvin yksinkertainen ja se sisältää Birds-nimisen taulun, joka sisältää ainakin seuraavat tallennettavan linnun attribuutit:

- `_id`: yksilöllisen, automaattisesti generoidun, tunnistein (`_id`) tietokannan hallintaa varten
- `name_fi`: suomenkielinen nimi
- `name_la`: latinankielinen nimi
- `description`: perustiedot
- `feeding`: ravinto
- `distribution`: elinalue

- notes: muistiinpanot, johon käyttäjä voi lisätä vapaamuotoista tekstiä.

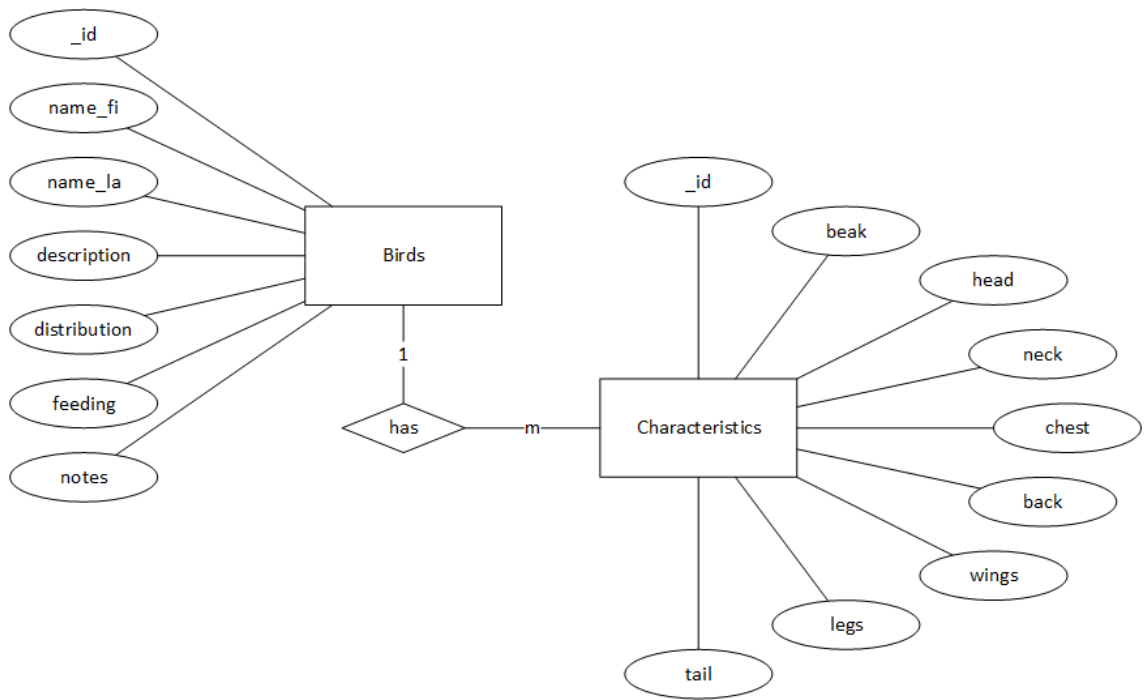
Tietokantaan lisätään myös valmius tallentaa yksityiskohtia linnusta. Ominaisuus helpottaa vaikeasti tunnistettavan linnun löytämistä, koska käyttäjä voi tehdä hakuja linnun ruumiinosien ominaisuuksien perusteella.

Linnun jokaiselle ruumiinosalle (nokka, pää, kaula, rinta, selkä, siivet, pyrstö ja jalat) voidaan antaa useita lyhyitä ja kuvaavia arvoja, kuten väri, kuvio tai muoto. Näitä detaileja varten on oma hakunsa. Hakuun voidaan kirjoittaa esimerkiksi linnun nokan kohdalle “pitkä”, “terävä” ja “keltainen”, jolloin ohjelma listaa kaikki linnut, joiden nokka on näiden kolmen ominaisuuden mukainen.

Yksityiskohtia varten on oma Characteristics-taulunsa, josta löytyvät seuraavat attribuutit:

- _id (tunniste hallintaa varten)
- beak (nokka)
- head (pää)
- neck (kaula)
- chest (rinta)
- back (selkä)
- wings (siivet)
- legs (jalat)
- tail (pyrstö).

Kuvassa 17 oleva ER-diagrammi kuvaa tietokantaa.



Kuva 17. ER-diagrammi tietokannasta.

5.3 Edistyneet toiminnot

Tämä kappale sisältää kuvauksen minkälainen lintukirjan oli alun perin tarkoitus olla, ennen kuin sen vaatimusmäärittelyä muutettiin. Kappaleen kuvaamia ominaisuuksia voidaan myös ajatella sovelluksen edistyneempinä ominaisuuksina, koska ne lisäävät sovelluksen käytettävyyttä merkittävästi. Ohjelman perusrakenne pysyy samana, mutta sovellus käyttää internet-yhteyden välityksellä hyväkseen palvelimella sijaitsevaa etätietokantaa, jossa aineistoa säilytetään. Etätietokanta tekee sovelluksen käytöstä kätevämpää, koska sen sijaan, että käyttäjä keräisi kaiken aineiston laitteeseensa itse, palvelimelta on mahdollista ladata valmista aineistoa sisältävä tietokanta. Käyttäjän ei myöskään tarvitse huolehtia ”oman” tietokantansa varmuuskopioinnista, koska tietokanta on aina tallessa palvelimella.

Etätietokantaa voidaan käyttää kahdella tavalla:

1. Online-tila. Kun laite on internetissä, joko 3G- tai WiFi-yhteyden kautta, lintukirja-sovellus käyttää palvelimella olevaa tietokantaa suoraan, lataamalla sieltä aineistoa laitteeseen reaaliaikaisesti sitä mukaa kun käyttäjä tarkastelee merkintöjä. Online-tila mahdollistaa sen, että sovellus käyttää aina ajan tasalla olevaa tietoa.

2. Offline-tila. Tietokanta voidaan myös kokonaisuudessaan kopioida laitteeseen niitä tilanteita varten kun internet-yhteyden käyttäminen ei ole mahdollista. Tietokannan koosta riippuen, sen lataaminen voi kestää pitkiäkin aikoja. Kun tietokanta on kerran kopioitu laitteeseen, kaikki merkinnät ovat selattavissa ilman internet-yhteyttä.

Jotta sovellus olisi käyttäjäystävällinen, offline-tilan toiminnan, eli palvelimen ja paikallisen tietokannan synkronoinnin, tulisi olla mahdollisimman automaattista. Tätä varten sovelluksen sisäistä tietokannanhallintaa olisi muokattava niin, että sovellus osaisi itse tarkistaa mitä merkintöjä paikallisesta tietokannasta jo löytyy ja tarvitseeko näitä merkintöjä päivittää. Tämän jälkeen mahdolliset päivitykset ja kokonaan uudet merkinnät haettaisiin palvelimelta. Jotta päivitysten tarkastaminen olisi mahdollista, tietokannan tauluihin tulisi lisätä uusi päivämäärä-attribuutti, joka kertoo milloin merkintää on viimeksi muokattu. Tämän ajankohdan perusteella sovellus osaisi tarkistaa ovatko paikallisen tietokannan merkinnät ajan tasalla.

Palvelin pohjainen tietokanta nostaa esiin kysymyksen, kuka vastaa aineiston keräämisestä tietokantaan. Eräs ratkaisu voisi olla websivu, jonka kautta tietokantaa hallitaan. Sivun sisältäisi tarvittavan käyttöliittymän uusien merkintöjen lisäämiseen, sekä vanhojen muokkaukseen ja poistoon. Sivun vaatisi kirjautumisen ja luotettaviksi todetut henkilöt saisivat tunnukset käyttää sivustoa uuden aineiston lisäämiseen. Näin aineistoa kertyisi vähitellen, sitä nopeammin mitä enemmän sivustolla olisi ylläpitäjiä.

5.4 Käytettävät työkalut

Ohjelman kirjoittamiseen suosittelen käytettäväksi Eclipse-kehitysympäristöä, jota myös Google suosittelee Android-kehitykseen. Eclipse on vapaan lähdekoodin kehitysympäristö, joka tukee useita eri ohjelmointikieliä, mm. Javaa, jota tässä projektissa tarvitaan. /1/

Android-kehitystä varten Eclipseen asennetaan Googlen ohjeiden mukaan ADT-lisäosa (Android Development Tools), joka integroi hyödyllisiä Androidiin liittyviä toimintoja

Eclipsen omaan käyttöliittymään. Lisäksi tarvitaan Android SDK, joka sisältää mm. emulaattorin ja debuggerin ohjelmien testaukseen, sekä tarvittavan dokumentaation.

/24/

6 POHDINTA

Opinnäytetyön aihe osoittautui odotettua haastavammaksi. Työn alkuperäisenä tavoitteena oli suunnitella ja kehittää toimiva sovellus, ei pelkkä suunnitelma sellaisesta. Lintukirjan oli tarkoitus perustua käyttäjien tuottamaan sisältöön, jota säilytettäisiin palvelimilla ja joka siirrettäisiin verkon yli itse laitteeseen tarpeen vaatiessa. Yhteisö myös vastaisi sisällön moderoinnista äänestysjärjestelmällä, jossa käyttäjät antaisivat plus- ja miinus-pisteitä lisätyille teksteille ja kuville. Epäsopiva aineisto poistuisi automaattisesti saatuaan liikaa negatiivisia pisteitä.

Projektin alussa olin melko positiivisella mielellä. Tutkin Android kehitykseen liittyvää teoriaa ja hahmottelin sovellusta Eclipse-kehitysympäristössä. Suunnitelmana oli ensiksi luoda sovellukseen toimiva ja yksinkertainen graafinen käyttöliittymä tarvittavine näkymineen, jonka jälkeen alkaisin lisätä varsinaisia toiminnallisuuksia. Käyttöliittymän luominen olikin aika helppoa ja käyttöliittymän komponenttien määrittäminen ja säätäminen XML-tiedostoissa oli mielenkiintoista. Itse ohjelman toiminnallisuus, eli Java-koodaaminen, olikin kokonaan toinen juttu.

Vaikeudet alkoivat kun aikaan piti saada jotain toimivaa. Android-ohjelmat kirjoitetaan Java-ohjelmointikielellä, josta itsellä on kokemusta kaksi, hyvin alhaisen tason tiedot antavaa, kurssia. Kurkseista oli aikaa vuosia, eikä toinen niistä aikoinaan edes mennyt kovin hyvin. Voidaan siis sanoa, että tietämykseni Javasta oli lähellä nolla, mikä ei luvannut hyvää tätä projektia ajattelen.

Koodia kirjoittaessani, Javan puutteellisesta hallinnasta seurasi, että suurimman osan ajasta etsin internetistä harjoituksia ja valmiita ratkaisuja, joita olisin voinut käyttää omassa projektissani, toteuttaakseni yhden, tietyn ohjelman toiminnallisuuden. Useimmiten en edes ymmärtänyt mitä kopioimani koodin pätkä teki, joten sen soveltaminen omassa projektissani tapahtui yritys-erehdys -menetelmällä. Oli äärimmäisen turhauttavaa käyttää tuntikausia aikaa yksittäisen ratkaisun etsimiseen ja sitten todeta, ettei se sopinutkaan tai toiminut odotetulla tavalla.

Edellä mainittujen vaikeuksien vuoksi opinnäytetyön vaatimusmääritettyä jouduttiin muuttamaan ensimmäisen kerran, koska totesin, etteivät taitoni tai ymmärrykseni riittäisi näin monimutkaiseen ohjelmaan.

Palvelimella oleva etätietokanta, joka vastaisi lintukirjan sisältämän aineiston säilyttämisestä, pudotettiin pois työn vaatimuksista. Palvelin pohjainen tietokanta olisi vaatinut omat, mahdollisesti selainpohjaiset, hallintajärjestelmänsä, joka taas olisi lisännyt työn, ja sitä myötä uusien opeteltavien asioiden, määrää liikaa.

Etätietokannan poistamisen myötä kuoli myös ajatus yhteisön tuottamasta sisällöstä. Oli siis tyytyminen paikalliseen tietokantaan itse puhelimesta, johon puhelimen käyttäjä itse lisäisi kaiken haluamansa. Tämä taas tappoi koko sovelluksen perimmäisen idean eli sovellus degeneroitui tavalliseksi matkapuhelimessa toimivaksi muistiosovellukseksi. Sovellus olisi kuitenkin edelleen hyvää harjoitusta Android-kehittämisestä tietokantoineen ja käyttöliittymineen.

Puuhailin aikani paikallisen SQLite-tietokannan parissa tutkien internetin tarjoamia tutoriaaleja. Tavoitteena oli luoda yksinkertainen graafinen käyttöliittymä, joka toimisi yhdessä tietokannan kanssa, jolloin käsissä olisi perustoimiva matkapuhelinsovellus.

Ongelmat alkoivat taas, enkä saanut näennäisesti yksinkertaista tietokantaa toimimaan. Netti oli täynnä aihetta käsitteleviä harjoituksia, mutta en koskaan täysin ymmärtänyt esimerkkikoodia. Viimeisenä oljenkortena minulla oli tarkoitus käyttää tietokannan pohjana Googlen tarjoamaa Notepad-harjoitusta, jolla esitellään tietokantojen käyttöä Androidissa. Tässä kohtaa petyin hieman, koska Notepad-harjoitus oli vanhentunut.

Android on edelleen kehittyvä alusta ja tavat, joilla sovellusten ominaisuuksia toteutetaan kooditasolla, muuttuvat ajan myötä. Kehitysympäristö saattaa ilmoittaa, että käyttämäsi tapa on vanhentunut ja tee asia mieluummin näin. Näin tuntui käyvän virallisen harjoituksen kanssa, koska aina kun muutin esimerkkikoodia omiin tarkoituksiini sopivaksi, kehitysympäristö valitti vanhentuneista työskentelytavoista. Lopulta luovutin tietokannankin suhteen, koska en saanut sitä toimimaan muokattuna.

Tietokannan toteuttamisen epäonnistumisen myötä opinnäytetyön vaatimusmäärittelyä laskettiin jälleen ja työ muuttui pelkäksi suunnitelmaksi lintukirja-sovelluksen toteuttamisesta. Tässä vaiheessa olin jo hahmotellut päässäni melko selvästi miltä ohjelma näyttäisi ja miten se toimisi. Tätä dokumenttia varten suunnittelin hahmotelmat

ohjelman käyttöliittymästä käyttäen Eclipseä ja ottamalla kuvia näkymistä emulaattorin kautta. Tietokannan suunnitelmaksi riitti yksinkertaisen ER-kaavion piirtäminen.

Loppusanoina sanottakoon, että olin hieman pettynyt tähän projektiin. Ehkä olin alun perinkin liian toiveikas, että saisin ohjelman toimimaan ja kenties jopa jakoon Play-sovelluskauppaan. Toisaalta taas sovelluksen toimivaksi saaminen oli epätodennäköistä, koska yli neljän vuoden insinöörikoulutuksen aikana emme koskaan ohjelmoineet vähänkään monimutkaisempia, toimivia, ohjelmia.

Opinnäytetyö oli kuitenkin mukava kurkistus Android-kehityksen maailmaan ja taustoja tutkiessa tuli perehdyttyä asioihin mistä tämän päivän käytetyimmät älypuhelimet rakentuvat.

LÄHTEET

/1/ About Eclipse Foundation. Hakupäivä 9.1.2013

<<http://www.eclipse.org/org/#about>>

/2/ About SQLite. Hakupäivä 6.2.2013

<<http://www.sqlite.org/about.html>>

/3/ Activities. Hakupäivä 6.9.2012

<<http://developer.android.com/guide/components/activities.html>>

/4/ Androidology - Part 1 of 3 - Architecture Overview. Hakupäivä 13.2013

<<http://www.youtube.com/watch?v=Mm6Ju0xhUW8>>

/5/ Android ActivityManager vs WindowManager. Hakupäivä 6.2.2013

<<http://stackoverflow.com/questions/9238078/android-activitymanager-vs-windowmanager>>

/6/ An Android coder's introduction to broadcast receivers. Hakupäivä 5.2.2013

<<http://www.techrepublic.com/blog/app-builder/an-android-coders-introduction-to-broadcast-receivers/1173>>

/7/ Android 4.1 Compatibility Definition. Hakupäivä 13.11.2012

<http://static.googleusercontent.com/external_content/untrusted_dlcp/source.android.com/en//compatibility/4.1/android-4.1-cdd.pdf>

/8/ Android Architecture - The Key Concepts of Android OS. Hakupäivä 6.2.2013

<<http://www.android-app-market.com/android-architecture.html>>

/9/ Android: Content Providers (Part 01). Hakupäivä 11.2.2013

<http://www.youtube.com/watch?v=3a2NYXZR_Z4>

/10/ Android Programming: Understanding the Activity Life Cycle. 11.5.2010.

<<http://answers.oreilly.com/topic/2692-android-programming-understanding-the-activity-life-cycle/>>

/11/ App Framework. Hakupäivä 13.2.2013

<<http://developer.android.com/about/versions/index.html>>

/12/ Application Fundamentals. Hakupäivä 6.9.2012

<<http://developer.android.com/guide/components/fundamentals.html>>

/13/ Apple iPhone 3G. Hakupäivä 6.2.2013

<http://www.gsmarena.com/apple_iphone_3g-2424.php>

/14/ Android Overview. Hakupäivä 5.2.2013

<http://www.openhandsetalliance.com/android_overview.html>

/15/ Android Version History. Hakupäivä 24.1.2013

<http://en.wikipedia.org/wiki/Android_version_history>

/16/ Bound Services. Hakupäivä 10.9.2012

<<http://developer.android.com/guide/components/bound-services.html>>

/17/ Chapter 11. Broadcast Receivers. Hakupäivä 5.2.2013

<http://ofps.oreilly.com/titles/9781449390501/Android_Broadcast_Receivers.html>

/18/ Content Providers. Hakupäivä 5.2.2013

<<http://developer.android.com/guide/topics/providers/content-providers.html>>

/19/ Content Provider Basics. Hakupäivä 5.2.2013

<<http://developer.android.com/guide/topics/providers/content-provider-basics.html>>

/20/ Dalvik VM Internals. Hakupäivä 13.2.2013

<<https://sites.google.com/site/io/dalvik-vm-internals>>

/21/ Designing for Multiple Screens. Hakupäivä 25.1.2013

<<http://developer.android.com/training/multiscreen/index.html>>

/22/ Frequently Asked Questions. Hakupäivä 14.2.2013

<<http://source.android.com/faqs.html#compatibility>>

/23/ Fragmentation? Just Buy the Nexus Already. Hakupäivä 13.2.2013

<<http://lifehacker.com/5924547/frustrated-by-android-fragmentation-just-buy-the-nexus-already>>

/24/ Get the Android SDK. Hakupäivä 9.1.2013

<<http://developer.android.com/sdk/index.html>>

/25/ Google Android Architecture. Hakupäivä 2.7.2012

<<http://electronics.howstuffworks.com/google-phone2.htm>>

/26/ Google Buys Android for Its Mobile Arsenal. Hakupäivä 11.2.2013

<<http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>>

/27/ Google Play. Hakupäivä 11.2.2013

<http://en.wikipedia.org/wiki/Google_Play>

/28/ Google Play Matches Apple's iOS With 700,000 Apps. Hakupäivä 14.2.2013

<<http://www.tomsguide.com/us/Google-Play-Android-Apple-iOS,news-16235.html>>

/29/ Grid View. Hakupäivä 25.1.2013

<<http://developer.android.com/guide/topics/ui/layout/gridview.html>>

/30/ HTC G1 Dream: 1 million in the USA. Hakupäivä 6.2.2013

<http://www.thedigitalnewsroom.com/en/news/2242/htc_g1_dream_1_million_in_the_usa.htm>

/31/ IDC: Android Market Share Reached 75% Worldwide In Q3 2012. Hakupäivä 11.2.2013

<<http://techcrunch.com/2012/11/02/idc-android-market-share-reached-75-worldwide-in-q3-2012/>>

/32/ Industry Leaders Announce Open Platform for Mobile Devices

<http://www.openhandsetalliance.com/press_110507.html>

/33/ Kernel (computing). Hakupäivä 5.2.2013

<[http://en.wikipedia.org/wiki/Kernel_\(computing\)](http://en.wikipedia.org/wiki/Kernel_(computing))>

/34/ Codec. Hakupäivä 6.2.2013

<<http://desktopvideo.about.com/od/glossary/g/codec.htm>>

/35/ Layouts. Hakupäivä 5.2.2013

<<http://developer.android.com/guide/topics/ui/declaring-layout.html>>

/36/ library. Hakupäivä 6.2.2013

<<http://www.webopedia.com/TERM/L/library.html>>

/37/ Licenses. Hakupäivä 24.1.2013

<<http://www.apache.org/licenses/>>

/38/ Linear Layout. Hakupäivä 11.1.2013

<<http://developer.android.com/guide/topics/ui/layout/linear.html>>

/39/ List View. Hakupäivä 25.1.2013

<<http://developer.android.com/guide/topics/ui/layout/listview.html>>

/40/ Location Strategies. Hakupäivä 4.9.2012

<<http://developer.android.com/guide/topics/location/strategies.html>>

/41/ Location source and accuracy. Hakupäivä 4.9.2012

<<http://support.google.com/gmm/bin/answer.py?hl=en&answer=81873>>

/42/ Manifest.permission. Hakupäivä 13.2.2013

<<http://developer.android.com/reference/android/Manifest.permission.html>>

/43/ NotificationManager. Hakupäivä 4.9.2012

<<http://developer.android.com/reference/android/app/NotificationManager.html>>

/44/ OpenGL ES - The Standard for Embedded Accelerated 3D Graphics. Hakupäivä 13.2.2013

<<http://www.khronos.org/opengles/>>

/45/ Open Handset Alliance – FAQ. Hakupäivä 5.2.2013

<http://www.openhandsetalliance.com/oha_faq.html>

/46/ Package management system. Hakupäivä 6.2.2013

<http://en.wikipedia.org/wiki/Package_manager>

/47/ Permissions. Hakupäivä 7.9.2012

<<http://developer.android.com/guide/topics/security/permissions.html>>

/48/ Platform Versions. Hakupäivä 11.1.2013

<<http://developer.android.com/about/dashboards/index.html>>

/49/ RAM - random access memory. Hakupäivä 5.2.2013

<<http://www.webopedia.com/TERM/R/RAM.html>>

/50/ Relative Layout. Hakupäivä 11.1.2013

<<http://developer.android.com/guide/topics/ui/layout/relative.html>>

/51/ Resources Overview. Hakupäivä 5.2.2013

<<http://developer.android.com/guide/topics/resources/overview.html>>

/52/ Saving activity state. Hakupäivä 5.2.2013

<<http://developer.android.com/guide/components/activities.html#actstate>>

/53/ Services. Hakupäivä 7.9.2012

<<http://developer.android.com/guide/components/services.html>>

/54/ Software framework. Hakupäivä 6.2.2013

<http://en.wikipedia.org/wiki/Software_framework>

/55/ TelephonyManager. Hakupäivä 19.7.2012

<<http://developer.android.com/reference/android/telephony/TelephonyManager.html>>

/56/ The AndroidManifest.xml File. Hakupäivä 27.2.2013

<<http://developer.android.com/guide/topics/manifest/manifest-intro.html>>

/57/ The many faces of a little green robot. Hakupäivä 13.2.2013

<<http://opensignal.com/reports/fragmentation.php>>

/58/ T-Mobile G1. Hakupäivä 6.2.2013

<http://www.gsmarena.com/t_mobile_g1-2533.php>

/59/ Transport Layer Security. Hakupäivä 3.7.2012

<http://en.wikipedia.org/wiki/Transport_Layer_Security>

/60/14/ understanding the android media framework. Hakupäivä 6.2.2013

<<http://rxwen.blogspot.fi/2010/01/understanding-android-media-framework.html>>

/61/ View. Hakupäivä 6.2.2013

<<http://developer.android.com/reference/android/view/View.html>>

/62/ Virtual Machine. Hakupäivä 13.2.2013

<http://en.wikipedia.org/wiki/Virtual_machine>

/63/ WebKit Project Goals. Hakupäivä 6.2.2013

<<http://www.webkit.org/projects/goals.html>>

/64/ What is FreeType?. Hakupäivä 3.7.2012

<<http://freetype.org/freetype2/index.html>>