

Juhani Halkola

# Android-käyttöjärjestelmällä toteutettu liikuntapäiväkirjasovellus

Metropolia

Ammattikorkeakoulu

Insinööri (AMK)

Tekijä(t) Otsikko Sivumäärä Aika	Juhani Halkola Android-käyttöjärjestelmällä toteutettu liikuntapäiväkirjasovellus 32 sivua 2.12.2014
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja	Lehtori Kimmo Saurén
<p>Tässä insinööriyössä toteutettiin liikuntapäiväkirjasovellus nimeltä KuntoJäbä. Tavoitteena on kehittää toimiva sovellus, jolla pystyy merkitsemään liikuntasuorituksia muistiin. Tämän sovelluksen voi ladata GitHubista tai Google Play-sovelluskaupasta.</p> <p>KuntoJäbä-sovellus toteutettiin Java ohjelmointikielellä Eclipse-ohjelmointiympäristöä käyttäen. GitHubia käytettiin versionhallintaan ja varmuuskopiointiin. Työssä myös kerrotaan yleisesti Android-käyttöjärjestelmästä ja Android-sovellusten sekä Android-sovelluskauppojen tietoturvallisuudesta.</p> <p>Insinööriyön tuloksena kehitettiin KuntoJäbä-sovellus, jolla voi merkitä omia liikuntasuorituksia muistiin. Käyttäjä pystyy näkemään päivinä milloin on harrastanut liikuntaa ja mitä liikuntaa. Sovelluksen käyttämiseen ei tarvitse internet-yhteyttä ja tiedot tallennetaan puhelimen sisäiseen muistiin.</p> <p>Lopussa pohditaan jatkokehitystä kyseiselle sovellukselle. Jatkokehitys on tärkeää sovelluksen elinkaarelle. Hyvät sovellukset syntyvät vuosien kehityksen ja testauksen seurauksena. KuntoJäbä-sovellus näyttää vain listana merkatut liikuntasuoritukset, mutta tulevaisuudessa se voisi näyttää suoritukset diagrammeina ja eri kuukausina.</p>	
Avainsanat	Android, fitness, Java, Git

Author(s) Title Number of Pages Date	Juhani Halkola Physical activity diary application created for Android Operating System 32 pages 2 December 2014
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor	Kimmo Saurén, Senior Lecturer
<p>The subject of this thesis was to develop an exercise diary application called KuntoJäbä. The goal was to develop a working application that would log exercise performance into the memory. This application can be downloaded from Github or Google Play app store.</p> <p>The KuntoJäbä application was implemented in the Java programming language using the Eclipse programming environment. GitHub was used in version control and for backup. This thesis gives an overview of the Android operating system and the security of Android applications and Android application stores.</p> <p>In this thesis the KuntoJäbä Android application was created to make it possible for a user to mark physical performance in the memory. The user is able to see the days when he has exercised and what kind of exercise has been done. The application does not require an Internet connection. The data are stored in the phone's internal memory.</p> <p>Further development of the application was also discussed in the thesis. Further development is important for an application's life cycle. Good applications are created over years of development and testing. The KuntoJäbä application shows only a list of the marked physical performance, but in the future it might display performance as diagrams and as different months.</p>	
Keywords	Android, fitness, Java, Git

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Kuntokeskusyritystoiminta Suomessa	2
3	Yleistä tietoa Android-käyttöjärjestelmästä	3
3.1	Androidin arkkitehtuuri	8
3.1.1	Sovellustaso	9
3.2	Aktiviteetit	11
4	Kehitysympäristö	14
5	Liikuntapäiväkirjasovelluksen kehittäminen	16
5.1	KuntoJäbä-sovellus	16
5.2	Sovellusprojektin tallentaminen GitHubiin	23
5.3	Android-sovelluksen testaaminen	24
5.4	Ongelmatilanteita	25
6	Sovelluksen julkaisu	28
7	Yhteenveto	28
	Lähteet	30
	Liitteet	

## Lyhenteet

ADT	Android Development Tools. Androidin kehitystyökalut Eclipse-kehitysympäristöön.
API	Application Programming Interface. Ohjelmointirajapinta, jonka mukaan eri ohjelmat voivat keskustella keskenään.
APK	Android application package file. Tiedostomuoto Android sovelluksien jakamiseen.
IDE	Integrated development environment. Integroitu ohjelmointiympäristö, jolla ohjelmoija suunnittelee ja toteuttaa ohjelmistoja.
OSI	Open Source Initiative. Järjestö joka edistää avoimen lähdekoodin ohjelmistojen käyttöä.
SDK	Software Development Kit. Ohjelmointirajapinta, jonka mukaan eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja keskenään.
SGL	Skia Graphics Library. Avoimen lähdekoodin graafinen kirjasto, joka on kirjoitettu C++ -ohjelmointikielellä.
Widget	Aloituskäyttöön pyörivä ohjelma, joka on yleensä pieni osa Android-sovellusta.

## 1 Johdanto

Kunto-urheilun suosio on kasvanut viime vuosien aikana hyvin paljon. Tämä näkyy kävijämäärän lisääntymisellä kuntosaleilla. Myös erilaiset treeniblogit ovat valloittaneet internetin. Kuntosaliharrastus ei ole nää niin tuntematon, vaan siitä on tullut suomalaisten harrastus iästä ja sukupuolesta riippumatta.

Markkinoilta löytyy paljon sovelluksia, joilla pystyy seuraamaan omia liikuntasuorituksia tai juoksulenkin pituutta. Näitä sovelluksia ovat esimerkiksi Sports Tracker, Simple Workout Log ja My Workout Journal. Tässä opinnäytetyössä tavoitteena on tehdä Android-sovellus, jonka avulla voi merkitä omia liikuntasuorituksia ylös. Markkinoilla on samantyyppisiä sovelluksia, mutta monet niistä ovat aika monimutkaisia tai vaikeita käyttää. Esimerksi sovelluksissa ei voinut merkitä päivän kohdalle, että liikuntasuoritus on suoritettu, vaan piti merkata kuinka monta hauiskäntöä on tehty ja millä painolla.

Tavoitteena on kehittää yksinkertainen Android-sovellus, jolla pystyy helposti merkitsemään liikuntasuorituksia ylös ja julkaista sovellus Google Play-palvelussa. Sovelluksen pystyy myös lataamaan GitHubista.

## 2 Kuntokeskusyrittöstoiminta Suomessa

Kansallisen liikuntatutkimuksen 2009–2010 mukaan kuntosaliharjoittelu on kolmanneksi suosituin liikuntamuoto suomalaisten 19–65-vuotiaiden keskuudessa. Suosituin liikuntaharrastus on kävelylenkkeily ja toiseksi suosituin pyöräily. Kansallinen liikuntatutkimus tehdään neljän vuoden välein. Vuoden 2005–2006 kansallisen liikuntatutkimuksen perusteella kuntosaliharjoittelu on kasvanut 36 % verrattuna vuoden 2009–2010 kansalliseen liikuntatutkimukseen, uusia harrastajia oli 189 000, joka teki kuntosaliharjoittelusta eniten kasvaneen lajin. [1.]

Jos kuntosalin palveluihin kuuluu muita liikuntapalveluita, sitä kutsutaan kuntokeskukseksi. Muita liikuntapalveluita voivat olla esimerkiksi ryhmäliikuntatunnit, vesiurheilumahdollisuus ja erilaiset palloilulajit.

Kuntokeskuksissa voi harjoittaa erilaisia liikuntaharrastuksia aina CrossFitistä joogaan. Kaikissa kuntokeskuksissa ei voi harrastaa kaikkia lajeja. CrossFitä voi harrastaa Suomessa tietyissä saleissa. Ensimmäinen CrossFit-sali tuli Suomeen vuonna 2007. CrossFit on kuntoilumuoto, jossa tehdään erilaisia harjoitteita fyysisen kunnon kohottamiseksi. [2.]

Omien liikuntasuorituksien analysoiminen on helppoa kuntosalilaitteiden ansiosta. Juoksumatot, soutulaitteet ja kuntopyörät näyttävät harjoittelun tehonkulutuksen watteina. Laitteet mittaavat myös käytetyn ajan ja matkan. CrossFit-harjoituksissa ajan ottaminen on tärkeää, koska monet harjoittelutohjelmot perustuvat siihen, että tiettyä liikettä toistetaan esimerkiksi minuutin ajan.

Vuonna 2014 kuntokeskukset SATS ja ELIXIA yhdistyivät ja muodostivat kuntokeskusketjun SATS ELIXIA:n. SATS ELIXIA tarjoaa mobiililaitteille sovelluksen, jolla pystyy esimerkiksi varaamaan ryhmäliikuntatunteja ja merkitsemään omia harjoituksia ylös. Jos on SATS ELIXIAN jäsen, niin sovelluksen kaikki toiminnot on käytössä, muille käyttäjille sovellus näyttää vain, että missä on lähin SATS ELIXIA kuntokeskus sijaitsee kartalla. Halvin jäsenyys maksaa 71 euroa kuukaudessa, mukaan tulee vielä 59 euron liittymismaksu. [3;4.]

### 3 Yleistä tietoa Android-käyttöjärjestelmästä

Andy Rubin perusti yhtiön nimeltä Android Inc Palo Alto, Kaliforniassa lokakuussa vuonna 2003. Yhtiö kehitti käyttöjärjestelmää älypuhelimille vielä tuolloin. [5.]

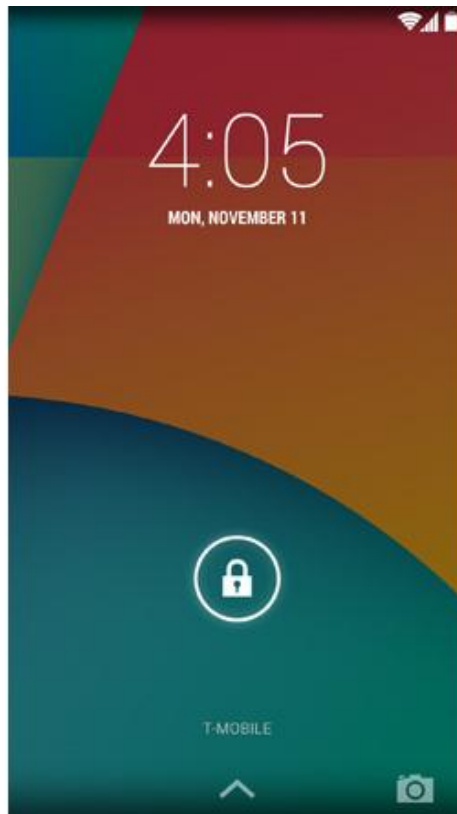
Google osti yhtiön Android Inc. elokuussa 2005. Toinen suuri yhdysvaltainen yritys Apple julkaisi 9. tammikuuta 2007 oman älypuhelimien iPhone, joka tuli myyntiin Yhdysvalloissa 29. kesäkuuta 2007. Vuoden lo0,pussa 84 yrityksen muodostama yritysliittymä Open Handset Alliance julkisti Androidin ensimmäisen tuotteen, päämääränä kehittää avointa standardia matkapuhelimille. Yleisen standardin vahvuus on siinä, että yksi ja sama sovellus toimii kaikilla puhelimilla, eikä sovellusta tarvitse kehittää yksitellen jokaiselle puhelimelle. Ensimmäinen älypuhelin HTC Dream, joka pyöri Android-käyttöjärjestelmällä, tuli myyntiin vuoden 2008 loppupuolella. Vielä tänäkin päivänä Open Handset Alliance vastaa Android-käyttöjärjestelmän kehityksestä. [6.]

Android syrjäytti johtavan Nokian Symbian-mobiilikäyttöjärjestelmän vuonna 2010. [7.]

#### Androidin ominaisuudet

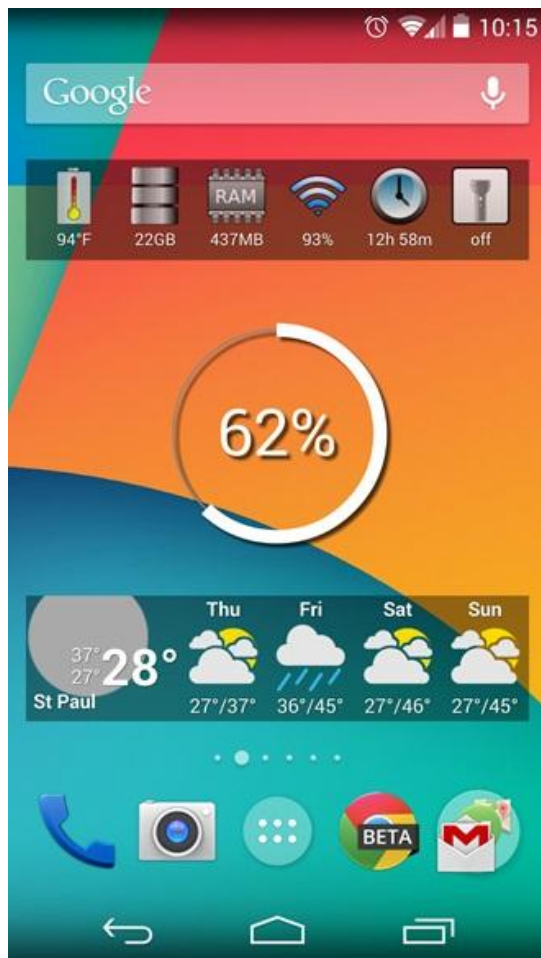
Android perustuu avoimeen lähdekoodiin, eli kehittäminen ja käyttäminen on täysin ilmaista kaikille. Androidin käyttöliittymä perustuu erilaisiin kosketuksiin. Androidilla varustettu älypuhelin lukee ruutua kädellä pyyhkäisemällä, näpäyttämällä ja nipistämällä. Androidin sisäinen laitteisto sisältää kiihtyvyyssanturin, gyroskoopin ja läheisyysanturin. Jotkut sovellukset käyttävät näitä antureita. Esimerkiksi kun käyttäjä puhuu puhelimessa, näyttö sammuu läheisyysanturin ansioista ja näin voi säästää akkua. [8.]





Kuva 1. Android KitKat 4.4-lukitusnäyttö.

Kun Androidilla varustetun älypuhelimien käynnistää, tulee näkyviin kuvan 1 mukainen lukitusnäyttö. Näytön voi avata lukkoa pyyhkäisemällä. Lukitusnäyttöä voi myös vaihtaa niin, että käyttäjä voi halutessaan syöttää pin-koodin tai vaikka piirtää kuvion. Lukitusnäytön voi myös avata niin, että kamerasovellus käynnistyy samalla.



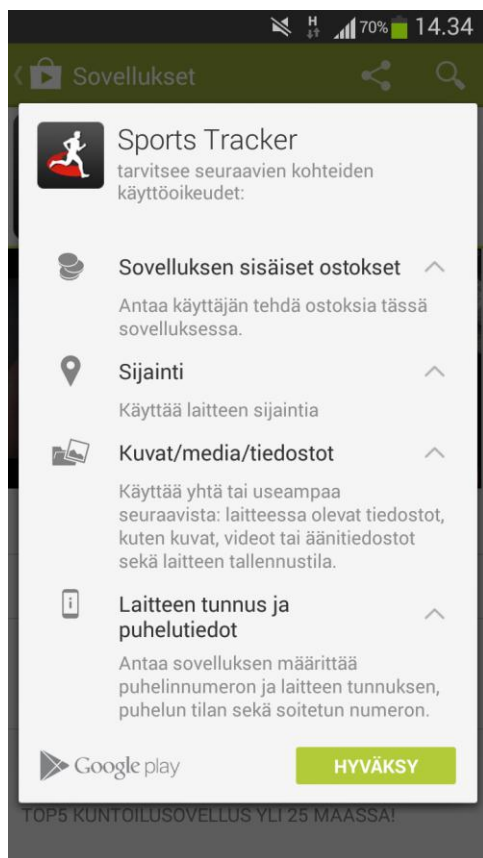
Kuva 2. Android KitKat 4.4 aloitusnäyttö.

Kuvassa 2 on Android Kitkat 4.4-aloitusnäyttö, joka tulee näkyviin, kun lukitusnäyttö on avattu. Aloitusnäyttöön voi halutessaan laittaa pienoishelmia. Kuvassa 2 on aloitusnäyttöön asennettu säätä näyttävä pienoishjelma. Pienoishjelmien vahvuus on siinä, että ne näyttävät nopeasti tietoa eikä käyttäjän tarvitse avata sovellusta. Pienoishjelmat voivat viedä tilaa parhaimmillaan koko näytön tai sitten yhden kuvakkeen verran. Android 1.5-versiossa oli ensimmäiset pienoishjelmat. [9.]

#### Androidin käyttöjärjestelmän tietoturvasuus ja yksityisyys

Vaikka Android-käyttöjärjestelmä on suunniteltu täysin avoimeksi, ei tietoturvasuutta ole unohdettu. Monikerroksinen turvasuus takaa juostavuuden avoimelle alustalle, mutta samalla hyvän suojan kaikille käyttäjille.

Android-sovellukset toimivat omassa hiekkalaatikkomaisessa ympäristössä ja käyttäjä näkee aina, mitä käyttöoikeuksia sovellus haluaa käyttää.

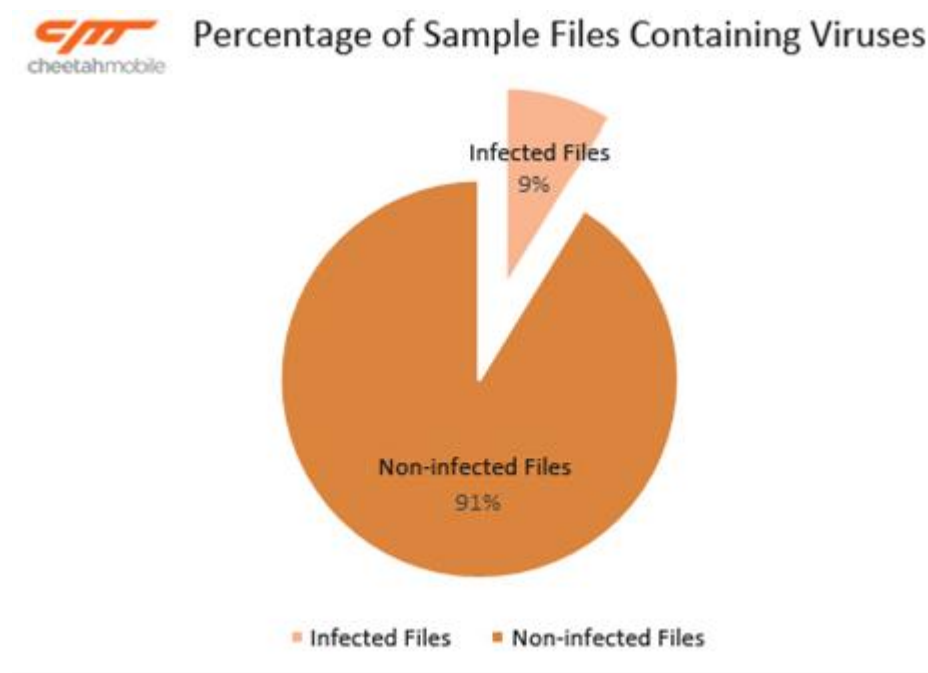


Kuva 3. Sports Tracker -sovellus kysyy käyttöoikeuksia asennuksessa.

Sports Tracker-kuntoilusovellus on sovellus, joka mittaa kuljetun matkan GPS:n avulla. Kuvassa 3 Sports Tracker-kuntoilusovellus kysyy käyttäjältä käyttöoikeuksia asennuksen yhteydessä. Sports Tracker-sovellus vaatii käyttäjältä esimerkiksi puhelimen sijainnin jakamista. Sovelluksessa tämä käy ilmi, että käyttäjä näkee millaisen matkan on taittanut kartalla. Matkan näyttäminen kartalla edellyttää sijaintitiedon jakamisen. Monet käyttäjät eivät juuri katso, mitä käyttöoikeuksia sovellus haluaa käyttää. Kun käyttöoikeudet on hyväksytty, sovellus ei enää kysy uudestaan samoja oikeuksia. Käyttäjä on itse vastuussa laitteen tietoturvasta. [10.]

Peruskäyttäjä voi ihmetellä, miksi Sports Tracker tarvitsee puhelutietoja, mutta niiden avulla sovellus tietää puhuuko käyttäjä puhelua. [11.]

Vuoden 2011 loppuun mennessä Androidin haittaohjelmien määrä on kasvanut räjähdysmäisesti.



Kuva 4. 9 % kaikista kerätyistä Android-näytteistä oli haittaohjelmia. [12.]

Kuvassa 4 havainnollistetaan Cheetah Mobilen keräämää 24,4 miljoonaa näytettä puolen vuoden ajalta vuoden 2014 tammikuusta lähtien. Kerätyistä näytteistä 2,2 miljoonaa oli saastuneita, mikä tarkoittaa noin yhdeksää prosenttia.

Vaikka Android haittaohjelmia on 2,2 miljoonaa, niistä 99,86 % oli muista digitaalisista sisältöpalveluista kuin Google Play -sovelluskaupasta. Yleensä kolmannen osapuolen sisältöpalvelut eivät tarkasta käyttäjien laittamia sovelluksia viruksilta.

Google kehitti Google Bouncerin helmikuussa 2012. Sen tehtävänä on poistaa haittaohjelmia Google Play -sovelluskaupasta. Se pyörii Google Playssa jatkuvasti ja analysoi uusia ja vanhoja sovelluksia haittaohjelmista. Google Playn käyttäjien ei tarvitse erikseen valtuuttaa Google Bounceria. Bouncer tutkii uusien ja vanhojen sovelluksien lisäksi myös kehittäjien tilit. [13.]

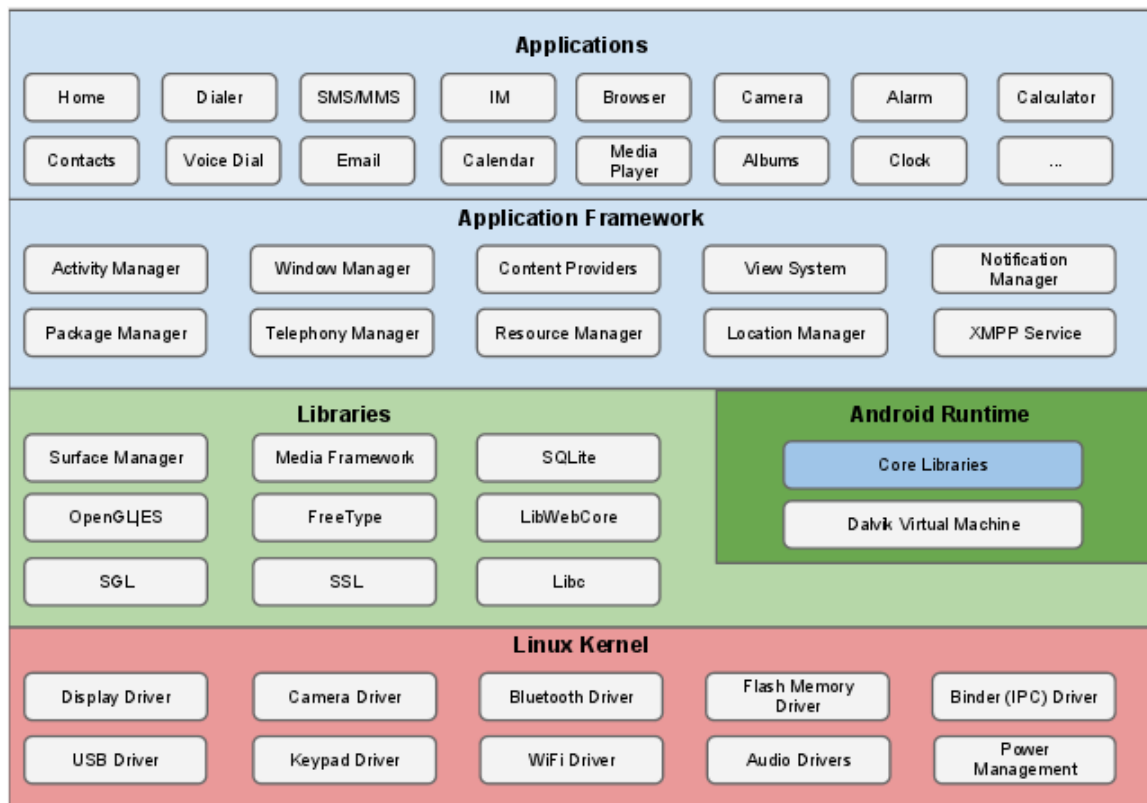
Vaikka Bouncer on vähentänyt haitallisten ohjelmien määrää, ei se täysin pysty kaikkia haittaohjelmia paikantamaan. Kaksi tietoturvatutkijaa testasi Bouncerin ominaisuuksia. He saivat selville seuraavia asioita Google Bouncerista:

- Bouncer käyttää QUME-virtualisointi ohjelmaa.
- Bouncer tarkistaa palveluun ladattua uutta sovellusta vain viiden minuutin ajan.
- Bouncer tekee vain dynaamisia tarkistuksia. Jos sovellus ei pyöri tarkistuksen aikana, se ei läpäise seula.
- Googlen määritetty IP-osoitealue Bouncerille on paljastettavissa, koska tarkistuksen aikana sovelluksella on yhteys Internetiin.

### 3.1 Androidin arkkitehtuuri

Android-mobiilikäyttöjärjestelmä perustuu Linux-ytimen ympärille. Androidin kehitykseen on käytetty yli 12 miljoonaa koodiriviä, josta yli 3 miljoonaa on XML koodia, yli 2,8 miljoonaa riviä on C-koodia ja yli 2,1 miljoonaa riviä on Java-koodia. [14.]

Kuvassa 5 on esitetty Androidi-käyttöjärjestelmän arkkitehtuuri.



Kuva 5. Android-käyttöjärjestelmän arkkitehtuuri. [10.]

### 3.1.1 Sovellustaso

Sovellustasolla ajetaan kaikki Android-käyttöjärjestelmän sovellukset, kuten verkkoselain, kalenteri ja herätyskello. Myös itse kirjoitetut ja ladatut ohjelmat pyörivät tällä tasolla. Taso on peruskäyttäjälle tuttu, koska se on Androidin näkyvä osa.

#### Sovelluskehys

Androidin avoin kehitysympäristö antaa rajapinnan sovelluskehittäjille mahdollisuuden tehdä erilaisia sovelluksia Androidille. Sovelluskehys tarjoaa sovelluskehittäjille esimerkiksi seuraavia palveluita:

- Toiminnanhallinta (Activity Manager) päättää, mikä näkymä käyttäjälle näytetään, ja tarpeen vaatiessa se pysäyttää tai lopettaa näkymiä.

- Ilmoitusten hallinta (Notification Manager) mahdollistaa ilmoitusten näyttämisen laitteen näytölle tai ruudun yläpalkille.
- Puhelinpalvelun hallinta (Telephony Manager) hoitaa äänipuhelut.
- Sisällön tarjoajan (Content Provider) avulla eri sovellukset voivat lukea tai käyttää toisten sovelluksien tietoja. [10.]

### Kirjastot

Android sisältää myös C/C++-kirjastoja, joihin sovelluskehittäjät pääsevät sovelluskehityksen kautta:

- Mediakirjastot (Media Framework) mahdollistavat tuen ääni-, video- ja kuvaformaateille.
- SGL on taustalla pyörivä 2D-grafiikkamoottori.
- SQLite on tehokas ja kevyt relaatiotietokanta.
- LibWebCore on Androidin verkkoselain moottori.

### Android Runtime

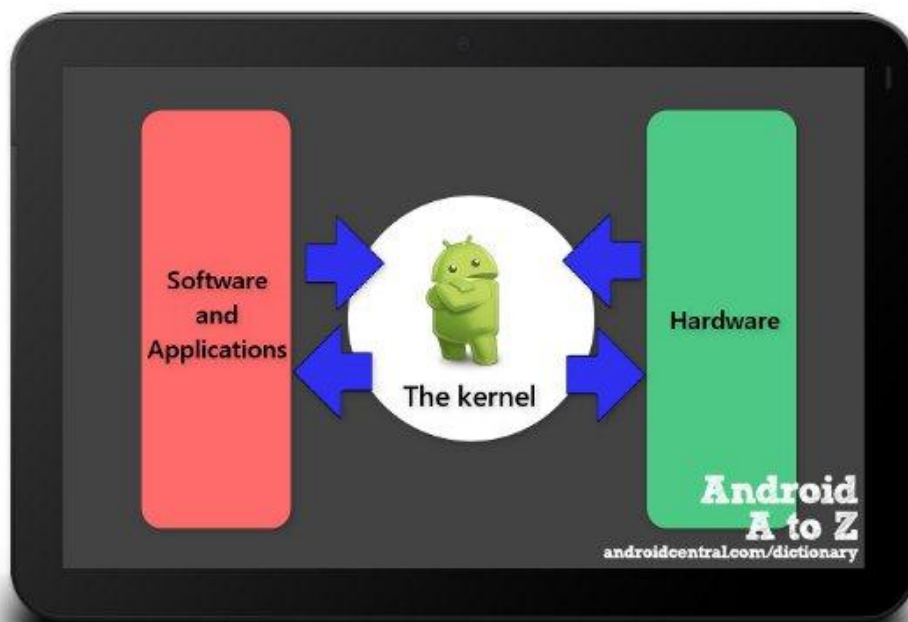
Android Runtime-kerros sisältää Androidin ydinkirjastoja ja Dalvik-virtuaalikoneen. Kaikki Android sovellukset ajetaan omana prosessina Dalvik-virtuaalikoneella. Vaikka monta prosessia olisi ajossa samaan aikaan, se ei haittaa Dalvik-virtuaalikonetta, koska se on suunniteltu ajamaan useita prosesseja samaan aikaan ja tehokkaasti.

Dalvik-virtuaalikone tukeutuu Linux-kernelin ominaisuuksiin, kuten alhaiseen muistinhallintaan ja prosessien ketjuttamiseen.

### Linux-ydin

Linux Kernel eli käyttöjärjestelmäydin on tietokoneen käyttöjärjestelmän alin osa, joka mahdollistaa laitteen muiden ohjelmien toiminnan. Ensimmäinen Android 1.0 versio käytti 2.6.25 Linux-ydintä (Kernel). Uusin Android 4.4.4 toimii 3.4.0 Linux-ytimellä. [15;16.]

Linux ytimen (Kernel) tehtävä on toimia virtualisointikerroksena laitteiston ja ohjelmistokerroksen välillä. Kuvassa 6 voi nähdä, kuinka Linux-ydin toimii näiden tasojen välissä. [17.]



Kuva 6. Linux-ydin [17.]

Linux-ytimessä pyörii paljon tärkeitä ydinpalveluita, kuten muistinhallinta, prosessien hallinta, ajurit, virranhallinta ja tietoturva.

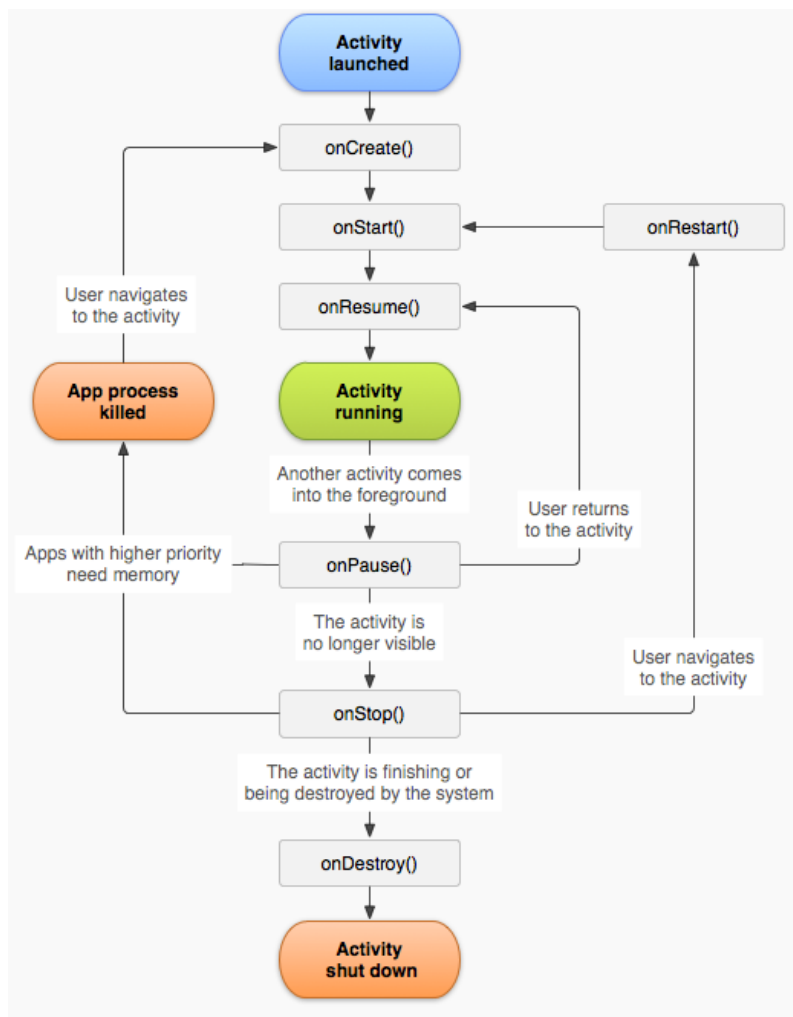
### 3.2 Aktiviteetit

Aktiviteetti on yksi sovelluskomponentti Android-sovelluksissa. Sen tehtävä on näyttää käyttäjälle sovelluksen käyttöliittymän näkymää. Näkymässä käyttäjä voi tehdä erilaisia toimintoja, kuten soittaa puheluita tai ottaa kuvia. Aktiviteetin luoma näkymä voi olla koko ruudun peittävä tai sitten pieni ikkuna muitten näkymien päällä. Vain yksi aktiviteetti voi olla kerrallaan aktiivinen ja reagoida käyttäjän syötteisiin.



Aktiviteetin elämänkaaren voi jakaa kolmeen eri pätilaan:

1. Aktiviteetti käynnistetään (onResume). Käynnistyksen jälkeen aktiviteetti on näkyvissä ja muiden aktiviteettien päällimmäinen näkymä.
2. Jos toinen aktiviteetti B käynnistetään, se menee osittain päällimmäiseksi ja alla oleva aktiviteetti A pysäytetään. Tässä tilassa Aktiviteetti A on keskeytynyt (onPause) ja säilyttää kaikki tietonsa, mutta jos muistia on erittäin vähän, Android-käyttöjärjestelmä saattaa lopettaa tämän aktiviteetin vapauttaakseen muistia lisää.
3. Aktiviteetti pysäytetään (onStop). Pysäytetty aktiviteetti säilyttää kaiken tietonsa, mutta ei ole enää näkyvissä käyttäjälle. Android-käyttöjärjestelmä saattaa lopettaa sovelluksen, jos muistia on liian vähän.



Kuva 7. Aktiviteetin elinkaari ja siirtymävaiheiden metodit. [18.]

Kuvassa 7 kuvataan aktiviteetin koko elinkaarta. Koko elinkaari sijoittuu onCreate()- ja onDestroy()-kutsujen väliin. Suorakulmiot kuvassa 7 esittävät eri metodeja. Ovaalin muotoiset kuviot kuvastavat aktiviteetin erilaisia tiloja.

```
public class Splash extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);
        Thread splash_timer = new Thread() {
            public void run() {
                try {
                    sleep(1500);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                } finally {
                    Intent openMainActivity = new Intent(
                        "com.insinoorityo.kuntojaba.MAIN");
                    startActivity(openMainActivity);
                }
            }
        };
        splash_timer.start();
    }
    @Override
    protected void onPause() {
        super.onPause();
        finish();
    }
}
```

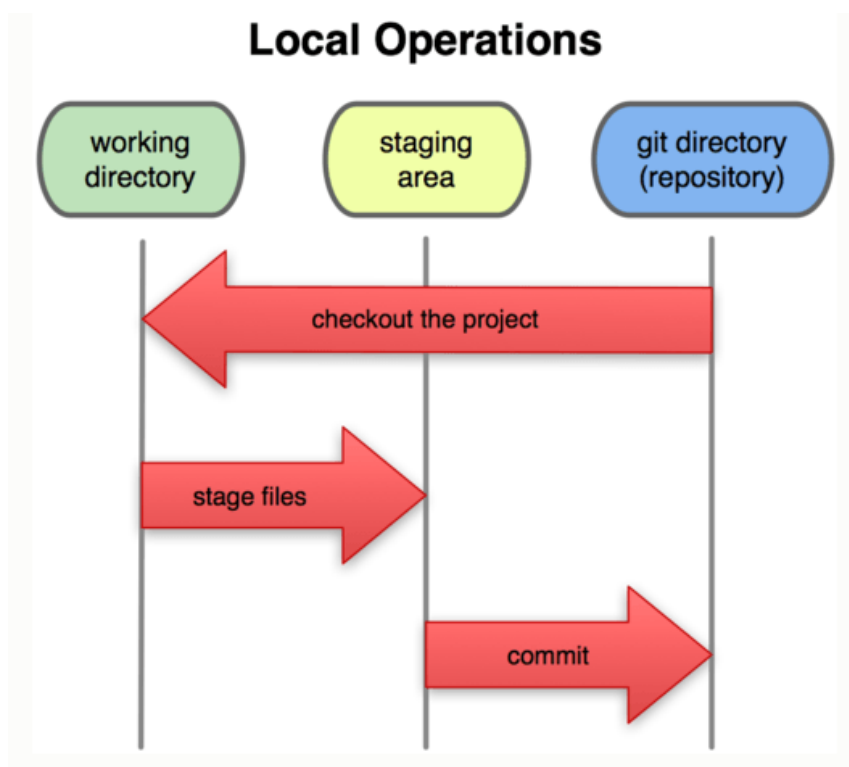
#### Esimerkkikoodi 1. Ote Splash-luokasta

Esimerkkikoodissa 1 Splash-luokka näyttää sovelluksen logoa puolitoista sekuntia ja sen jälkeen käynnistää pääluokan eli varsinaisen ohjelman. Kun pääluokka käynnistyy, tämä tulee Splash-aktiviteetin päälle ja takaisin-nappulaa painamalla päästiin logoa näyttävään näkymään takaisin. Splash-näkymä halutaan kuitenkin näytettävän vain kerran, kun sovellus käynnistetään eikä siihen tarvitse palata enää. Kun pääluokka avautuu sovelluksessa peittäen näkyvän Splash-aktiviteetin niin onPause() metodi tunnistaa tämän ja finish() metodi lopettaa Splash-aktiviteetin.

## 4 Kehitysympäristö

### Git-versionhallintaohjelmisto

Git on versionhallintaohjelmisto, joka on tehty toimimaan hajautetusti. Versionhallinnalla tarkoitetaan sitä, että vanhempaan versioon voi palata myöhemmin. Ohjelmistokehityksessä tämä on todella tärkeää, koska kun kehittäjät tekevät päivityksiä ja muutoksia sovellukseen niin voidaan selvittää, missä kohtaa sovellus vielä toimi. Hajautetuttu versionhallinta on erittäin tehokas, koska jokainen käyttäjä tallentaa itselleen oman kopion kotikoneelle. Näin, jos yhdeltä häviää projekti, se löytyy muilta käyttäjiltä. Git on nopea, koska muutokset eli *commitit* tehdään paikallisesti. [19.]



Kuva 8. Paikalliset operaatiot. [20.]

Kuvassa 8 voidaan nähdä kolme pääasiallista tilaa, missä Gitin tiedostot voivat olla. Git-hakemisto tärkein paikka, mihin kaikki ohjelmoijat laittavat omat muutokset. Työskentelyhakemisto on yksi versio otettu pääasiallisesta Git-hakemistosta, mihin voi tehdä omat muutokset. Kun muutokset on tehty, merkitään tiedostot, mihin on tehty

muutoksia. Lavastuksen jälkeen tiedostot liitetään Git-hakemistoon *commit*-komennolla.

Suljettu BitKeeper-ohjelmisto oli Linux-ytimen versionhallintaohjelmisto, mutta vuonna 2005 se lopetettiin. Linuxin kehittäjä Linus Torvalds kehitti avoimen versiohallintaohjelmiston – Gitin. Nykyään Gitin ylläpitäjä on Junio Hamano. [21.]

#### GitHub-sivusto

Toisin kuin Git, joka perustuu komentoliittymään, on GitHub internetsivu, jonne ohjelmoijat voivat tallentaa omia ohjelmointiprojektejaan kaikkien nähtäväksi. GitHub on pääsääntöisesti tarkoitettu ohjelmakoodille, mutta sinne voi tallentaa myös muita tiedostoja. GitHub toimii graafisena käyttöliittymänä. Oman käyttäjätilin voi luoda maksutta, mutta jos haluaa omasta projektista yksityisen, johon muut eivät pääse käsiksi, niin täytyy maksaa.

GitHubin vahvuus on siinä, että muihin avoimen lähdekoodin projekteihin on helppo ottaa osuutta. Ennen piti ladata ensin avoimen lähdekoodin projekti, sitten täytyi tehdä muutokset. Muutokset piti myös kirjoittaa muistiin, että alkuperäinen omistaja näkee ne helposti ja sitten lähettää uusi projekti omilla lisäyksillä sähköpostitse projektin alkuperäiselle omistajalle.

GitHubin avulla avoimen lähdekoodin projektin omistaja näkee heti, jos joku haluaa tehdä muutoksia hänen projektiinsa ja samalla, mitä toinen käyttäjä haluaa tehdä. Samalla projektin alkuperäinen omistaja näkee toisen käyttäjän muut lisäykset muihin projekteihin. [22.]

#### Eclipse ohjelmointiympäristö

Eclipse on ohjelmointiympäristö, joka auttaa ohjelmoijaa tekemään sovelluksia. Eclipse tarkistaa ohjelmakoodin oikeellisuuden. Siinä on myös lauseen ennakoiva tekstinsyöttö. Eclipse on kirjoitettu Java-ohjelmointikielellä, mutta Eclipsessä voi myös muilla ohjelmointikielillä kehittää sovelluksia.

Eclipse-kehitysympäristöön voi asentaa ADT:n eli Androidin kehitystyökalut. Nämä saa ilmaiseksi [developer.android.com](http://developer.android.com)-sivustolta. Työkalut sisältävät tarvittavat työkalut Android-sovelluksien kehitykseen. Android sovelluskehittäminen onnistuu myös hyödyntämällä Android NDK-pakettia, jossa Javan ja XML:n sijaan voi ohjelmoida C/C++-ohjelmointikielellä. Mukaan tulee emulaattori, jonka hitaus saattaa aiheuttaa pitkiä odotusaikoja. Nopein tapa on ajaa kirjoitettu ohjelma omalle Android laitteelle. [23;24.]

Google julkaisi vuoden 2013 maaliskuussa Androidille Android Studio-kehitysympäristön, joka pohjautuu IntelliJ IDEA -kehitystyökaluun. Android Studio on vielä beetetestausvaiheessa, joten kaikki toiminnallisuudet eivät vielä toimi tai toimivat vajavaisesti. Ohjelman uusin versio on 0.8.6. [25.]

Android Studio on Googlen oma ohjelma, joka hyvin varmasti tulee syrjäyttämään Eclipsen tulevaisuudessa.

## 5 Liikuntapäiväkirjasovelluksen kehittäminen

### 5.1 KuntoJäbä-sovellus

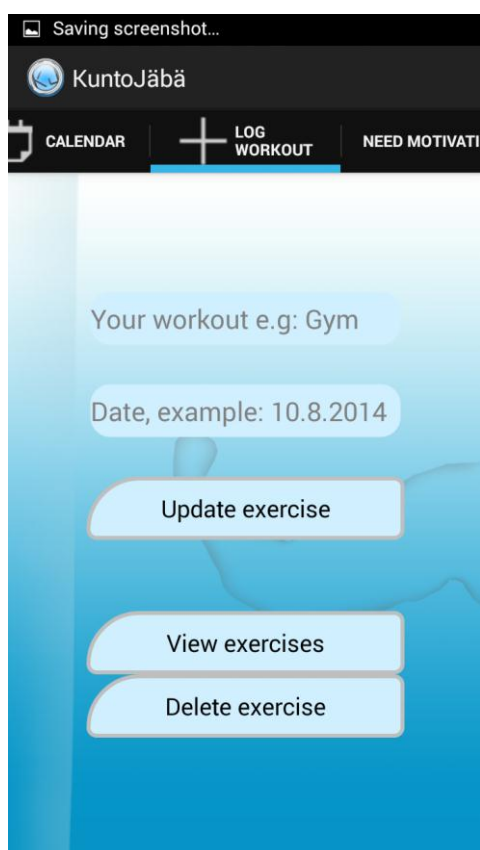
KuntoJäbä-sovellus on Android-käyttöjärjestelmälle kehitetty sovellus, jolla pystyy merkitsemään omia liikuntasuorituksia muistiin. Merkityt liikuntasuoritukset nähdään allekkain listana päivämäärä ensimmäisenä ja sen perässä liikuntasuoritus. KuntoJäbä-sovellus näyttää myös kalenterin näkymän korostaen, mikä päivä on parhaillaan menossa. KuntoJäbä-sovelluksessa on ominaisuus, jolla nappia painamalla saa motivoivan mietelauseen ruudulle näkyviin.

Android-käyttöliittymän suunnittelu on haastavaa, koska puhelimen ruutu on pieni. Käyttöliittymällä tarkoitetaan sitä osaa, jolla käyttäjä pystyy antamaan komentoja ohjelmalle. [26.]

Monet käyttäjät lopettavat sovelluksen käyttämisen, jos se on liian monimutkainen. Guenther Beyerin mukaan käyttäjällä on mielessään kolme kysymystä, kun hän käynnistää sovelluksen:

1. Missä kohtaa sovellusta olen nyt?
2. Mitä voin tehdä tässä vaiheessa sovellusta?
3. Mitä voin tehdä tuonnempana?

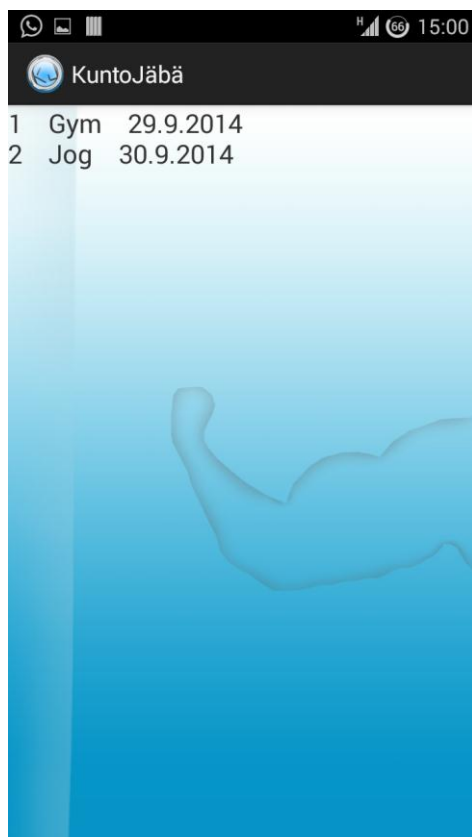
Näihin kysymyksiin vastaamalla voi vakuuttaa käyttäjälle, että sovellus on hyvä. Guenther Beyer on Android-käyttöjärjestelmä ja suunnittelu asiantuntija, joka perusti AndroidIcons.com nettisivuston kehittäjiä varten, mistä voi ladata ilmaisia ikoneja. [27.]



Kuva 9. Kuvakaappaus KuntoJäbä-sovelluksesta.

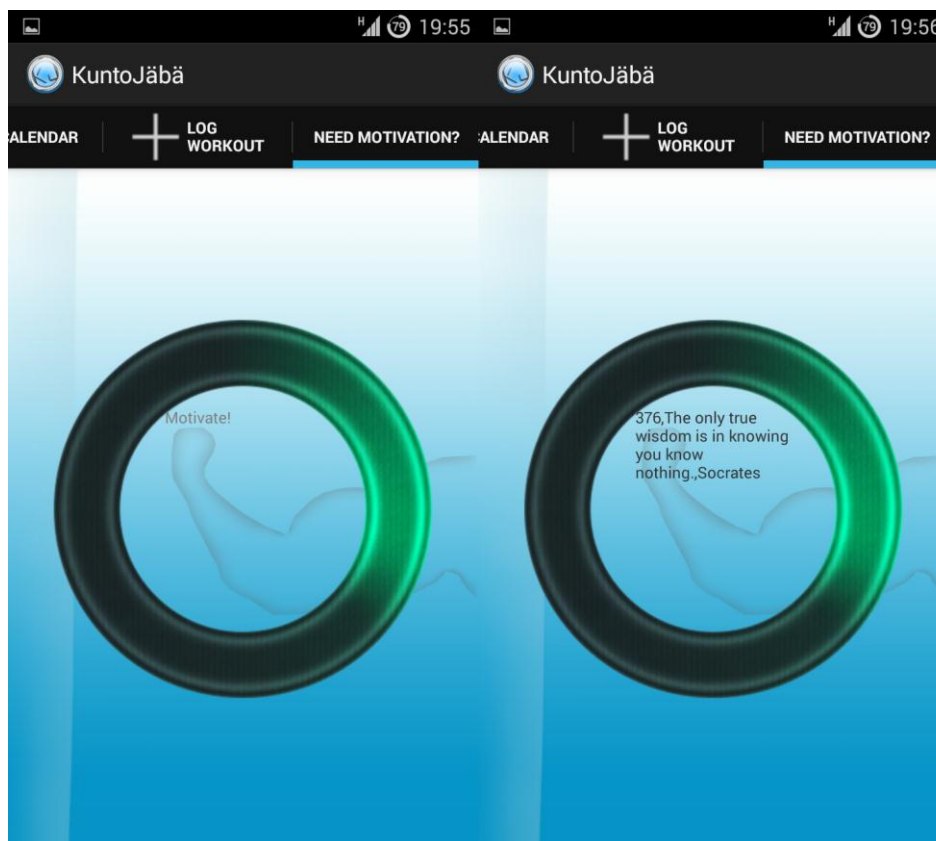
KuntoJäbä-sovelluksen käyttöliittymä toteutettiin kuvan 9 mukaisesti. Se on tarpeeksi yksinkertainen ja sovellus sisältää kolme välilehteä, joissa kaikilla on omat toimintonsa. Calendar-välilehdellä on näkyvä vain perinteinen kalenteri ilman muita toimintoja. Kuvassa 9 esitetyssä välilehdessä käyttäjä voi syöttää liikuntasuorituksen ja

päivämäärän muistiin. Samalla välilehdellä voi myös tarkastella kaikkia liikuntamerkintöjä ja myös halutessaan poistaa niitä.



Kuva 10. Kuvankaappaus KuntoJäbä -sovelluksesta.

Kun liikuntasuorituksia on kirjattu ylös, niitä voin katsoa kuvan 10 mukaisesti. Tieto tallennetaan puhelimen sisäiseen muistiin eli internet-yhteyttä ei tarvitse tässä sovelluksessa. Tallentaminen on toteutettu SQL-tietokantaa hyödyntäen. [28.]



Kuva 11. Kuvankaappaus KuntoJäbä -sovelluksesta.

Sovelluksen viimeinen välilehti on kuvassa 11. Siinä käyttäjä voi painaa rengasta ja saa vastineeksi jonkun mietelauseen. Lainattu virke tulee sattumanvaraisesti yli 500 virkkeen Excel-tiedostosta [34]. Tiedosto sijaitsee sovelluksessa itsessään, joten internet-yhteyttä ei tarvitse.

Android-käyttöjärjestelmä perustuu avoimeen lähdekoodiin, mutta pelkkä lähdekoodin muokkaaminen ei tarkoita, että se olisi avoin. Open Source Initiative järjestö keksi termin *open source* eli avoin lähdekoodi. Open Source Initiativen määritelmän mukaan ohjelma on avointa lähdekoodia hyödyntävä, jos se täyttää seuraavat kriteerit: [29.]

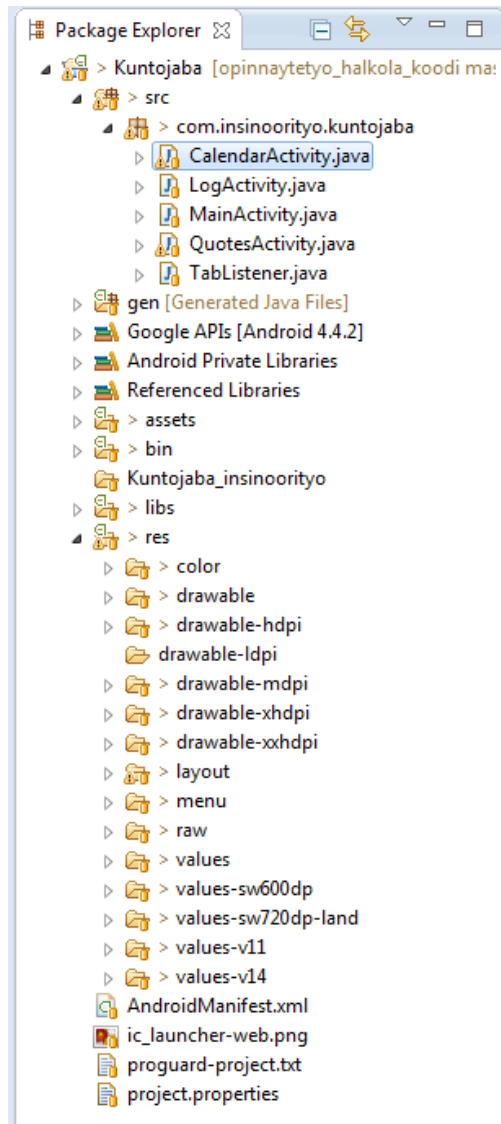
1. Ohjelman täytyy olla vapaasti levitettävissä ja välitettävissä.
2. Lähdekoodin täytyy tulla ohjelman mukana tai olla vapaasti saatavissa.
3. Myös johdettujen teosten luominen ja levitys pitää sallia.
4. Lisenssi voi rajoittaa muokatun lähdekoodin levittämistä vain siinä tapauksessa, että lisenssi sallii korjaustiedostojen ja niiden lähdekoodin levittämisen. Voidaan myös vaatia, ettei johdettua teosta levitetä samalla nimellä tai versionumerolla kuin lähtöteosta.
5. Yksilöitä tai ihmisryhmiä ei saa asettaa eriarvoiseen asemaan.
6. Käyttötarkoituksia ei saa rajoittaa.



7. Kaikilla ohjelman käsiinsä saaneilla on samat oikeudet.
8. Lisenssi ei saa olla riippuvainen laajemmasta ohjelmistokokonaisuudesta, jonka osana ohjelmaa levitetään, vaan ohjelmaan liittyvät oikeudet säilyvät, vaikka se irrotettaisiin kokonaisuudesta.
9. Lisenssi ei voi asettaa ehtoja muille ohjelmille. Ohjelmaa saa levittää myös yhdessä sellaisten ohjelmien kanssa, joiden lähdekoodi ei ole avointa.
10. Lisenssin sisällön pitää olla riippumaton teknisestä toteutuksesta. Oikeuksiin ei saa liittää varauksia jakelutavan tai käyttöliittymän varjolla.[29.]

Koska Android on ilmainen, kehitystyökalut voi ladata ilmaiseksi internetistä. Ensiksi pitää valita kehitysympäristö, itse valitsin Eclipsen, koska se on minulle tuttu ja siihen löytyy Android Development Tools (ADT) lisäosa. Kolmas asia, joka tarvitaan on Android Software Development Kit (SDK), joka sisältää kehitys-, paketointityökalut ja emulaattorin.

Kun tarvittavat työkalut on asennettu, voidaan luoda uusi Android-projekti Eclipseen. Eclipse kysyy luomisen yhteydessä sovelluksen nimen, tarvittavan Android-version ja väriteeman.



Kuva 12. Kuvakaappaus Android-projektista Eclipsessä.

Kuvassa 12 on Android-ohjelman perusrakenne. Android-projektin kannalta tärkeimmät hakemistot ovat: src-, bin-, res- ja libs-hakemistot sekä AndroidManifest.xml-tiedosto. [30.]

Src-hakemiston alla on ohjelmani lähdetiedostot eli aktiviteetit ja varsinaisen ohjelman. [30.]

Bin-hakemisto sisältää APK-asennustiedoston, jonka Android-laite osaa asentaa käyttöönsä. [30.]

Res-hakemisto sisältää useita alihakemistoja. Tärkeimmät alihakemistot ovat layout-hakemisto, joka määrittelee ohjelmani ulkoasun ja drawable-kansio, missä sijaitsee kaikki kuvat ja ikonit ohjelmaani varten. Raw-alihakemistoon olen laittanut ”motivation\_database\_new.csv” -tiedoston, joka sisältää sitaatteja sovellustani varten. [30.]

AndroidManifest.xml määrittelee ohjelman yhteensopivuuden eri Android versioille. Tämä tiedosto sisältää myös kaikki sovelluksen käyttämät aktiviteetit, palvelut, yleisvastaanottajat(broadcast receivers) ja sisällöntuottajat (content providers). Tiedosto määrittää ohjelman tarvitsemat oikeudet. Olen määritellyt sovellukseni KitKat 4.4 Android versiolle, mikä on uusin Android versio. [30.]

Koska Androidia kehitetään jatkuvasti siitä syntyy uusia versioita, jotka sisältävät uusia ominaisuuksia ja uudet versiot korjaavat virheitä, myös tietoturvaa parannetaan.

Version	Codename	API	Distribution
2.2	Froyo	8	0.7%
2.3.3 - 2.3.7	Gingerbread	10	13.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	10.6%
4.1.x	Jelly Bean	16	26.5%
4.2.x		17	19.8%
4.3		18	7.9%
4.4	KitKat	19	20.9%

Kuva 13. Androidin versiot. (Luotu: 12.8.2014) [31.]

Kuvassa 13 on esitelty eri Android versioiden nimet, ohjelmointirajapintakerrosluvut (API) ja jakauma. KuntoJäbä -sovellus sisältää fragmenttejä, jotka vaativat toimiakseen Android version 3.0 tai suuremman. [31.]

## 5.2 Sovellusprojektin tallentaminen GitHubiin

Git-versiohallinta ohjelman voi ladata heidän virallisilta internet sivuilta. Mukaan tulee kaksi versiota: graafinen ja komentorivipohjainen käyttöjärjestelmä. On suositeltavaa käyttää jälkimmäistä, koska graafisessa käyttöjärjestelmässä ei ole kaikkia toimintoja.

Ensiksi pitää mennä hakemistoon, missä sovellusprojekti sijaitsee:

```
cd /"oikea polku missä projekti sijaitsee"
```

Seuravaaksi alustetaan kansio Gitille sopivaksi:

```
git init
```

Seuraavaksi lisätään versionhallinta piiriin kaikki tiedostot mitkä kansiossa on. Piste tarkoittaa, että kansion kaikki tiedostot lisätään. Tiedostoja voisi lisätä myös yksitellen korvaamalla pisteen tiedoston nimellä.

```
git add .
```

Git tietää nyt mihin tiedostoihin on koskettu eli tehdään seuraavaksi muutos eli *commit*:

```
git commit -m "Ensimmäinen committini"
```

Ensin tehtiin GitHubissa uusi koodiarkisto ja tämän jälkeen seuraavalla komennolla saadaan yhteys tiedoston ja internetsivun välille:

```
git remote add github https://github.com/Juhanijh/  
KuntoJaba.git
```

Projektin siirtokomento GitHubiin:

```
git push origin master
```

Nyt projekti on GitHubissa kaikkien saatavilla. Ilman alkuperäisen projektin omistajan lupaa kukaan ei voi tehdä muutoksia projektiin. Muut käyttäjät voivat ladata projektin ja

tehdä muutoksia sitä kautta, mutta ei suoraan GitHubiin ellei projektin omistaja sitä valtuuta.

Komennolla `Git pull` saa projektin uusimman version GitHubista. Näin projektia voi jatkaa myös toiselta koneelta ja projektia voi tehdä useampi henkilö kerrallaan. Muutoksien jälkeen on aina hyvä työntää projekti takaisin GitHubiin käyttäen komentoa `Git push origin master`.

### 5.3 Android-sovelluksen testaaminen

Kirjoitettu koodi ajettiin Samsung Galaxy S3-älypuhelimeen, koska se toimii paljon nopeammin kuin emulaattorilla. Android-versio on koodinimellä kulkeva Jelly Bean 4.3. Ohjelmistokehityspaketin mukana tulee emulaattori, missä voi virtuaalisesti ajaa sovellus, tämä on kuitenkin hitaampi tapa kuin ajaa kirjoitettu koodi suoraan puhelimeen paikallisesti.

Ennen kuin oman ohjelmakoodin voi ladata Android-laitteelle, pitää asetuksista kytkeä pois asetus, joka estää muitten kuin virallisen sovelluskaupan asentamista laitteelle. Android laitteissa on oletuksena estetty muiden kuin virallisten sovellusten asentaminen tietoturvariskin takia. Tämän saa pois seuraavasti: Asetukset -> Turvallisuus -> Tuntemattomat lähteet. Nyt asennuspaketin voi asentaa laitteelle.

```
import com.example.KuntoJaba.R;

public class SQLView extends Activity{

    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        Log.d("sqlview", "1");
        setContentView(R.layout.sqlview);
        Log.d("sqlview", "2");
        TextView tv = (TextView)findViewById(R.id.tv_workout_data);
        LogActivity info = new LogActivity(this);
        Log.d("sqlview", "3");
        info.open();
        String data = info.getData();
        info.close();
        tv.setText(data);
    }
}

```

Application	Tag	Text
system_process	ActivityManager	STARTI ba/cc
com.example.KuntoJaba	sqlite	5:
com.example.KuntoJaba	sqlview	1
com.example.KuntoJaba	sqlview	2
com.example.KuntoJaba	sqlview	3

Kuva 14. LogCatin hyödyntämistä Eclipsessä.

Kuvassa 14 on luokan SQLViewin ohjelmistokoodi, johon on lisätty log-merkintöjä sovelluksen testausta varten. Kuten kuvassa 14 nähdään, sovellus suorittaa kaikki koodirivit eikä sovellus kaadu tai jätä suorittamatta ohjelmistokoodia. LogCatia hyödyntäen pystyy myös selvittämään jonkin muuttujan arvon. Tämä on tärkeää esimerkiksi tietokantaan tallentaessa, jotta oikea arvo menee oikeaan paikkaan.

#### 5.4 Ongelmatilanteita

KuntoJäbä-sovelluksen kehityksessä pyrittiin yksinkertaisuuteen. Kuntojäbä-sovellus sisältää tästä syystä vain kolme välilehteä. Ensimmäinen välilehti näyttää kalenterinäkymän. Toinen välilehti näyttää näkymän, jossa käyttäjä voi merkitä liikuntasuorituksensa ylös. Viimeinen välilehti sisältää näkymän, jossa käyttäjä saa nappia painamalla motivaatiolauseen ruudulle noin 500 erilaisesta lauseesta.

Kalenterinäkymä tuotti ongelmia, koska Android ei tarjoa kalenterinäkymää, jota voisi räätälöidä KuntoJäbä-sovellukselle sopivaksi. Esimerkiksi Androidin tarjoamissa

kalenterinäkymissä ei voi vaihtaa tietyn päivän taustaväriä niin kuin kuvassa 15 alhaalla.

August 2014						
Mo n	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Kuva 15. Kalenterinäkötehtävä Grid Viewiä käyttäen.

Kuvassa 15 kalenterinäkötehtävä näkötehtävä on tehty Grid Viewiä ja ArrayAdapteria käyttäen. Grid View luo näkötehtävän ja ArrayAdapter täyttää ruudukon solut numeroilla ja tekstillä. Ideana oli, että kun käyttäjä on merkinnyt urheilusuorituksen ylös kalenterinäkötehtävässä, se päivä muuttuisi vihreäksi. Näin käyttäjä voisi helposti katsoa liikuntasuorituksia kalenterista, mikä myös motivoisi, jos ei olisi hetkeen käynyt harrastamassa liikuntaa.

Lopulliseen sovellukseen tätä näkötehtävää ei tullut, koska sitä ei saatu toimimaan kunnolla. Kun sovelluksen lopetti, vihreällä merkityt päivät muuttuivat takaisin harmaiksi. Erilaisia ratkaisuja yritettiin, mutta lopulta päädyttiin Androidin tarjoamaan CalendarView-luokkaan, joka näyttää perinteisen kalenterinäkötehtävän ilman lisätoimintoja. Päivien taustaväriä ei voi muuttaa näkötehtävässä. Kalenterinäkötehtävän luominen on työlästä, koska pitää ottaa huomioon, että kuukausissa on 28–31 vuorokautta ja helmikuussa on 29 vuorokautta 28 vuorokauden sijaan jos on karkausvuosi. CalendarView-luokka tekee kaiken.





## 6 Sovelluksen julkaisu

### Google-Play -sisältöpalvelu

Google-Play, entinen Android Market on digitaalinen sisältöpalvelu, josta voi ladata sovelluksien lisäksi myös e-kirjoja, lehtiä, musiikkia, tv-ohjelmia ja elokuvia. Palvelut ovat maakohtaisia. Esimerkiksi Suomessa ei ole lehti- ja tv-ohjelma-palveluita. [32.]

Google-Playssa on yli 1,3 miljoonaa sovellusta ja niistä noin 0,2 miljoonaa on haitallisia. Google-Play on yleensä esiasennettu Android-käyttöjärjestelmää käyttäviin laitteisiin. Google Playn käyttö vaatii internetyhteyttä, joko mobiiliverkon kautta tai WLAN-yhteydellä. Koska kyseessä on Googlen tarjoama palvelu joten sovellusten lataaminen vaatii myös Google-tilin, mikä on ilmainen. Sovellukset voivat olla ilmaisia tai maksullisia. Yleensä ilmaiset sovellukset sisältävät mainoksia. [33.]

## 7 Yhteenveto

Sovellukset mobiililaitteille on viime vuosina ollut suuressa kasvussa. Myös fitness-buumi on puhjennut kokonaisvaltaisesti. Kaupoista löytyy terveyttä edistäviä ruoka-aineita ja ihmiset kirjoittavat urheilusuorituksistaan omissa blogeissa. Vaikka markkinoilla on paljon sovelluksia, joilla pystyy esimerkiksi mittaamaan juoksulenkin pituutta ja keskinopeutta. Insinööriyön tarkoituksena oli tehdä yksinkertainen Android-sovellus, millä pystyy merkitsemään liikuntasuorituksia ylös. Sovelluksen nimi on KuntoJäbä.

KuntoJäbä-sovelluksessa tiedon tallentamiseen on käytetty apuna Androidin tarjoamaa SQLite-relaatiotietokantaa. KuntoJäbä-sovelluksessa se on toteutettu MVC-arkkitehtuuria käyttäen. MVC-arkkitehtuurin ideana on jakaa ohjelma kolmeen osaan: malliin, näkymään ja käsittelijään. KuntoJäbä-sovelluksessa LogActivity-luokka hoitaa tiedon tallentamisen. SQLView-luokka näyttää tallennetun tiedon. SQLite-luokka hoitaa käyttäjältä tulevat käskyt eli toimii käsittelijänä.

Sovelluksen rungon olisi voinut tehdä eri tavalla, jolla olisin ehkä saanut kaikki toiminnot toimimaan. KuntoJäbä-sovellus toteutettiin käyttämällä fragmenttejä.

KuntoJäbä-sovelluksen lähes kaikki kuvat on tehty Adobe Photoshop-kuvankäsittelyohjelmalla. Androidille-sovelluskehitys on melko helppoa, koska kehitykseen tarvittavat työkalut saa ilmaiseksi ja internet on täynnä Android-ohjelmointi oppaita. Vaikeaa on keksiä sovellusta, mitä ei olisi olemassa. Google Play-sovelluskaupasta löytyy yli 1.3 miljoonaa sovellusta, joten sovelluksia löytyy peleistä taskulamppu-sovellukseen asti.

#### Jatkokehitys

KuntoJäbä-sovelluksen kehitystä voisi jatkaa pidemmälle tekemällä siihen käyttäjäprofiilit. Tällä hetkellä sovellus tallentaa kaiken tiedon vain puhelimen sisäiseen muistiin. Käyttäjäprofiilien ansiosta käyttäjät voisivat kilpailla urheilusuorituksista muiden käyttäjien kanssa. Jos käyttäjä A on esimerkiksi käynyt lenkillä 12 kertaa kuukauden aikana ja käyttäjä B on käynyt vain 6 kertaa, tällä voisi olla motivoiva vaikutus käyttäjään B.

Erilaiset peleistä tutut saavutukset voisivat olla hyvä lisä sovellukseen. Jos käyttäjä käy esimerkiksi neljä kertaa viikon aikana harrastamassa jotain liikuntaa, käyttäjää voisi palkita ”saavutuksilla” tai pisteillä tästä suorituksesta. Käyttäjät voisit vertailla pisteitä toistensa kanssa.

Sovelluksen laajentaminen muille mobiilikäyttöjärjestelmille lisäisi sen latausmääriä. PhoneGap-ohjelmistokehityksen ansiosta tämä olisi mahdollista. KuntoJäbä-nimen vaihtaminen voisi myös tuoda lisää latausmääriä, nimen perusteella ei selviä millainen, sovellus on kyseessä. Jos nimen vaihtaisi esimerkiksi liikuntapäiväkirja-sovellukseksi, niin käyttäjille ilmenee heti millainen, sovellus on kyseessä.

## Lähteet

- 1 Kansallinen Liikuntatutkimus 2009-2010. 2013. Verkkodokumentti. Sport. < [http://www.sport.fi/system/resources/W1siZiIsIjIwMTMvMTAvMjQvMTRfMThfNDdfMTcwX0xpaWt1bnRhdHV0a2ltdXNfYWlrdWlzbGlpa3VudGFfMjAwOV8yMDUwLnBkZiJdXQ/Liikuntatutkimus\\_aikuisliikunta\\_2009-2010.pdf](http://www.sport.fi/system/resources/W1siZiIsIjIwMTMvMTAvMjQvMTRfMThfNDdfMTcwX0xpaWt1bnRhdHV0a2ltdXNfYWlrdWlzbGlpa3VudGFfMjAwOV8yMDUwLnBkZiJdXQ/Liikuntatutkimus_aikuisliikunta_2009-2010.pdf) >. Luettu 4.8.2014.
- 2 CrossFit Suomessa. 2014. Verkkodokumentti. Crossfitsuomi. < <http://www.crossfitsuomi.fi/> >. Luettu 26.8.2014.
- 3 Tämä on SATS ELIXIA. 2014. Verkkodokumentti. Sats Elixia. < <http://www.sats.fi/sats-yrityksena/tama-on-sats/> >. 25.8.2014. Luettu 29.8.2014.
- 4 Tarjonta ja hinnat. 2014. Verkkodokumentti. Sats Elixia. < <http://www.sats.fi/jaseneksi/valitse-jasenyys/> >. Luettu 29.8.2014.
- 5 Google Buys Android for Its Mobile Arsenal. 2005. Verkkodokumentti. Businessweek. < <http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>>. 16.8.2005. Luettu 10.6.2014.
- 6 Applen iPhone myyntiin kesäkuun lopussa. 2007. Verkkodokumentti. Helsinginsanomat. < <http://www.hs.fi/talous/artikkeli/Apllen+iPhone+myyntiin+kes%C3%A4kuun+lopusa/1135227752503>>. 4.6.2007. Luettu 10.6.2014.
- 7 Google topples Nokia from smartphones top stop. 2011. Verkkodokumentti. Reuters. < <http://uk.reuters.com/article/2011/01/31/oukin-uk-google-nokia-idUKTRE70U1YT20110131>>. 31.1.2011. Luettu 13.6.2014.
- 8 Proximity Sensor on Android Gingerbread. 2014. Verkkodokumentti. Developer Android. < [http://developer.android.com/guide/topics/sensors/sensors\\_position.html](http://developer.android.com/guide/topics/sensors/sensors_position.html)>. Luettu 13.6.2014.
- 9 What is a widget?. 2012. Verkkodokumentti. Androidcentral. < <http://www.androidcentral.com/what-is-a-widget>>. 7.2.2012. Luettu 19.6.2014.
- 10 Android Security Overview. 2014. Verkkodokumentti. Source Android. < <http://source.android.com/devices/tech/security/> >. Luettu 15.7.2014.
- 11 Sovelluksen käyttöoikeuksien tarkastelu. 2014. Verkkodokumentti. Support Google. < [https://support.google.com/googleplay/answer/6014972?p=app\\_permissions&rd=1](https://support.google.com/googleplay/answer/6014972?p=app_permissions&rd=1) >. Luettu 4.8.2014.

- 12 2014 Half Year Security Report. 2014. Verkkodokumentti. Cheetahmobile. < <http://www.cmcm.com/blog/2014-07-18/186.html> >. 18.7.2014. Luettu 4.8.2014.
- 13 A Look at Google Bouncer. Verkkodokumentti. < <http://blog.trendmicro.com/trendlabs-security-intelligence/a-look-at-google-bouncer/> >. 20.7.2012. Luettu 28.7.2014.
- 14 How many lines of code does it take to create the Android OS?. 2010. Verkkodokumentti. Security Intelligence Blog. < <http://www.gubatron.com/blog/2010/05/23/how-many-lines-of-code-does-it-take-to-create-the-android-os/> >. 23.5.2010. Luettu 21.7.2014.
- 15 Android version history. 2014. Verkkodokumentti. Wikipedia. < [http://en.wikipedia.org/wiki/Android\\_version\\_history](http://en.wikipedia.org/wiki/Android_version_history) >. 22.7.2014. Luettu 22.7.2014.
- 16 Google Rolling Out Android 4.4.4 Update (KTU84P) With A Security Fix, Factory Images/Binaries Up For Nexus Devices. 2014. Verkkodokumentti. Android Police. <<http://www.androidpolice.com/2014/06/19/google-rolling-out-android-4-4-4-update-ktu84p-with-a-security-fix-factory-imagesbinaries-up-for-nexus-devices/>>. 19.6.2014. Luettu 22.7.2014.
- 17 Android A to Z: What is a kernel?.2012. Verkkodokumentti. Androidcentral. < <http://www.androidcentral.com/android-z-what-kernel> >. 23.1.2012. Luettu 22.7.2014.
- 18 Activities. 2014. Verkkodokumentti. Developer Android. < <http://developer.android.com/guide/components/activities.html> >. Luettu 15.9.2014.
- 19 1.1 Alkusanat – Versionhallinnasta. Verkkodokumentti. < <http://git-scm.com/book/fi/Alkusanat-Versionhallinnasta> >. Luettu 18.7.2014.
- 20 1.3 Getting Started – Git Basics. 2014. Verkkodokumentti. Git. < <http://git-scm.com/book/en/Getting-Started-Git-Basics> >. Luettu 25.7.2014.
- 21 1.2 Getting Started – A Short History of Git. 2014. Verkkodokumentti. Git. < <http://git-scm.com/book/en/Getting-Started-A-Short-History-of-Git> >. Luettu 25.7.2014.
- 22 What Exactly Is GitHub Anyway? 2012. Verkkodokumentti. Techcrunch. < <http://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/> >. 14.6.2012. Luettu 25.7.2014.
- 23 ADT Plugin. 2014. Verkkodokumentti. Developer Android. < <http://developer.android.com/tools/sdk/eclipse-adt.html> >. Luettu 18.7.2014.

- 24 Android NDK. 2014. Verkkodokumentti. Developer Android. <  
<http://developer.android.com/sdk/installing/studio.html> >. Luettu 29.8.2014.
- 25 Android Studio. 2014. Verkkodokumentti. Developer Android. <  
<http://developer.android.com/sdk/installing/studio.html> >. Luettu 29.8.2014.
- 26 User interface. 2014. Verkkodokumentti. Webopedia. <  
[http://www.webopedia.com/TERM/U/user\\_interface.html](http://www.webopedia.com/TERM/U/user_interface.html) >. Luettu 29.9.2014.
- 27 10 Tips For Android UI Design. 2011. Verkkodokumentti. Phandroid. <  
<http://phandroid.com/2011/05/11/10-tips-for-android-ui-design/> >. 11.5.2011.  
Luettu 29.9.2014.
- 28 Saving Data in SQL Databases. 2014. Verkkodokumentti. Developer Android. <  
<http://developer.android.com/training/basics/data-storage/databases.html> >. Luettu 30.9.2014.
- 29 The Open Source Definition. Verkkodokumentti. Open Source initiative. <  
<http://opensource.org/docs/osd> >. Luettu 22.7.2014.
- 30 Managing Projects. 2014. Verkkodokumentti. Developer Android. <  
<http://developer.android.com/tools/projects/index.html> >. Luettu 29.8.2014.
- 31 Platform Version. 2014. Verkkodokumentti. Developer Android. <  
<http://developer.android.com/about/dashboards/index.html> >. 12.8.2014. Luettu 1.9.2014.
- 32 Country availability for apps & digital content. 2014. Verkkodokumentti. Support Google. <  
<https://support.google.com/googleplay/answer/2843119?hl> >. Luettu 29.8.2014.
- 33 Number of Android applications. 2014. Verkkodokumentti. Appbrain. <  
<http://www.appbrain.com/stats/number-of-android-apps> >. 16.9.2014. Luettu 17.9.2014.
- 34 Favorite Quotes. 2014. Verkkodokumentti. Litemind. <  
<https://litemind.com/favorite-quotes/> >. Luettu 10.10.2014.