



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Alexi Anttila

# Älykello terveydenhuollon laitteena

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Opinnäytetyö

12.5.2019

Tekijä(t) Otsikko	Aleksi Anttila Älykello terveydenhuollon laitteena
Sivumäärä Aika	43 sivua + 1 liite 12.5.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tietotekniikan koulutusohjelma
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Lehtori Simo Silander
<p>Tämän opinnäytetyön tarkoitus oli selvittää älykelloihin vaikuttavia terveydenhuollon laitteiden lakisääteisiä vaatimuksia EU:n ja Suomen alueella sekä tutkia Android Wear -kellon tarjoamia sovellusrajapintoja ja toteuttaa yksikertainen Android Wear -sovellus. Työssä perehdyttiin terveydenhuollon alan lakisääteisiin määräyksiin, EU:n laatimiin terveydenhuollon sovelluksia koskeviin direktiiveihin ja standardeihin sekä terveydenhuollon sovellusten tuoteluokitteluun.</p> <p>Android Wear -älykellolle voi toteuttaa direktiivien ja standardien mukaisen turvallisen sovelluksen käyttäen Googlen tarjoamia sovellusrajapintoja. Rajapinnat tarjosivat monia eri tapoja, jolla käyttäjän pystyi luotettavasti tunnistamaan, sekä tavan siirtää ja vastaanottaa tietoa muista laitteista.</p> <p>Työssä opin tuntemaan terveydenhuollon sovellusvaatimukset sekä opin soveltamaan vaatimuksia Android Wear -sovelluksen suunnittelussa ja tekemisessä. Työn aikana terveydenhuollon vaatimusten laajuus ja vaikeus sovelluskehityksessä yllättivät, mutta tiedon soveltaminen sovellussuunnittelussa auttoi paremmin ymmärtämään oleelliset vaatimukset ja sen miksi ne ovat tärkeitä.</p>	
Avainsanat	Terveydenhuollon laite, Älykello, Android Wear

Author(s) Title	Aleksi Anttila Smartwatch as a Medical Device
Number of Pages Date	43 pages + 1 appendices 12 May 2019
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Specialisation option	Software Engineering
Instructor(s)	Simo Silander, Senior Lecturer
<p>The purpose of this thesis was to clarify the medical device regulations affecting the smart watches in the EU and Finland and investigate application interfaces for Android Wear devices. The thesis explored the statutory provisions of the healthcare sector, the EU directives and standards on medical devices, and the product classification of medical devices and software's.</p> <p>Implementing safe application, in compliance with directives and standards, for Android Wear smart watches can be done by using application interfaces provided by Google. Interfaces provide many ways for the app to indentify/authorize the user, and transfer or receive information from other devices.</p> <p>During the work, I learned about the medical device requirements for software's, and learned to apply requirement during designing the Android Wear application. The scope and difficulty of the medical device requirements surprised me, but when designing the application I learned to understand the essential requirements and why the regulations are important.</p>	
Keywords	Medical device, smartwatch, Android Wear

## Sisällys

1	Johdanto	1
2	Terveydenhuollon laitteita koskevat vaatimukset	2
2.1	MD-, IVD- ja AIMD-direktiivit	3
2.2	Potilastietoihin liittyvät lait	4
2.3	EU:n yhdenmukaistetut standardit	4
2.4	Käyttötarkoitus	5
2.5	Tuoteluokittelu	5
2.5.1	Terveydenhuollon laitteet (MD)	5
2.5.2	In vitro diagnostiset -laitteet (IVDMD)	7
2.6	Laadunhallintajärjestelmä (QMS)	7
2.7	Kliininen arviointi	8
2.8	CE merkintä	9
2.9	Laiterekisteröinti	11
2.10	Vaaratilanneilmoitukset	11
3	Terveydenhuollon ohjelmiston luokittelu	14
3.1	Terveydenhuollon ohjelmiston määrittelykriteerit	14
3.2	Itsenäisen ohjelmiston luokittelu	16
3.3	Moduulit	16
4	Älykello terveydenhuollon laitteena	17
5	Vaatimustenmukainen Android Wear -sovellus	18
5.1	Kellotaulu	19
5.2	Komplikaatio	19
5.3	Minisovellus	22
5.3.1	Käyttöliittymä	22
5.3.2	Tietokanta	28
5.3.3	Käyttäjän tunnistaminen	30
5.3.4	Tiedonsiirto	34
6	Yhteenveto	39
	Lähteet	40
	Liitteet	
	Liite 1. Valviran tuoteluokkamäärittelyt	

## Lyhenteet ja käsitteet

API	Application Programming interface eli ohjelmointirajapinta on määritelmä, jonka mukaan ohjelmat voivat kommunikoida.
HTTP	HypertextTransfer Protocol. Verkkoselaimien ja WWW-palvelimien käyttämä tiedonsiirtoprotokolla.
UDP	User Datagram Protocol. Verkkoselaimien ja WWW-palvelimien käyttämä tiedonsiirtoprotokolla.
TCP	Transmission Control Protocol. Verkkoselaimien ja WWW-palvelimien käyttämä tiedonsiirtoprotokolla.
UUID	Universally unique identifier. 128-bittinen numero, jota käytetään tunnistamaan tietoja tietokonejärjestelmissä.
ISO	International Organization for Standardization. Kansainvälinen standardisimisjärjestö.
Käyttötarkoitus	Tarkoittaa käyttöä, johon terveydenhuollon laite valmistajan merkinnöissä, käyttöohjeessa tai myynninedistämistä koskevassa aineistossa annettavien tietojen mukaan on tarkoitettu.
Markkinoille saattaminen	Terveydenhuollon laitteen ensimmäistä käyttöön saattamista maksua vastaan tai korvauksetta laitteen jakelemiseksi tai käyttämiseksi Euroopan unionin alueella riippumatta siitä, onko laite uusi, täysin kunnostettu tai uudelleenkäsitelty.
Kliininen arviointi	Valmistajan kliinisten tietojen perusteella tekemä arviointi, jolla vahvistetaan, että laitteen ominaisuudet ja suorituskyky ovat vaatimusten mukaisia laitteen tavanomaisissa käyttöolosuhteissa.

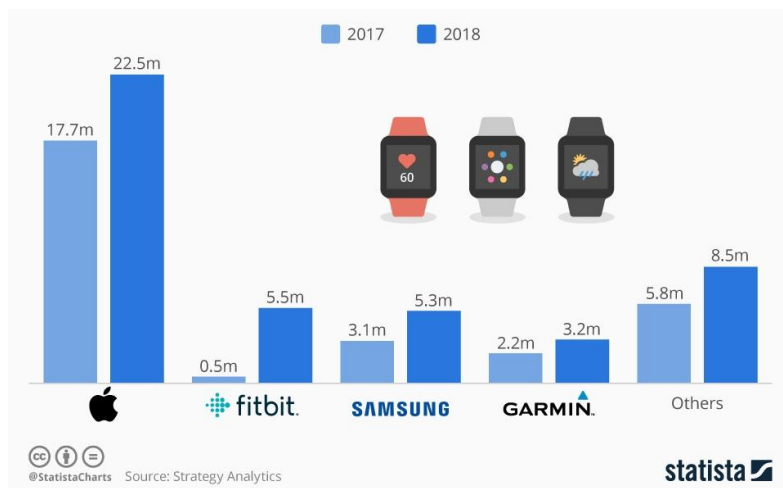
**Käyttöönotto** Käyttöönotolla tarkoitetaan vaihetta, jolloin terveydenhuollon laite on loppukäyttäjän saatavilla ja valmis käytettäväksi ensimmäistä kertaa käyttötarkoituksensa mukaisesti Euroopan unionin alueella.

**Ilmoitettu laitos** Euroopan unionin jäsenvaltion nimeämää ja Euroopan komissiolle ilmoitettua laitosta, jolla on oikeus tehdä vaatimustenmukaisia arviointeja.

**Turvallisuustiedote** Valmistajan tai sen edustajan asiakkaille ja ammattimaiselle käyttäjälle lähettämä korjaavia toimenpiteitä koskeva tiedonanto.

## 1 Johdanto

Terveydenhuoltoon tarkoitettujen mobiilisovellusten määrä on viime vuosina ollut rajussa kasvussa. Zion marketin julkaisemassa raportissa arvioitiin, että vuonna 2018 terveydenhuoltoon tarkoitettujen mobiilisovellusten markkina-arvo oli noin 8,0 miljardia dollaria. Samassa raportissa todettiin, että markkina-arvon ennustetaan kasvavan 111,1 miljardiin dollariin vuoteen 2025 mennessä [1].



Kuva 1. Älykellojen markkinaosuus vuosina 2017 ja 2018. [2]

Terveydenhuollon sovellusten markkina-arvon kova kasvu ja älykellojen erinomainen sopivuus terveydenhuollon laitteiksi on kasvattanut älykellojen kysyntää joka vuosi. Apple ja Fitbit ovat viime vuosina laajentaneet älykellomarkkinoita suuresti (katso kuva 1), sekä Google on vihdoin panostamassa lisää Android Wear -kelloihin. Googlen Android Wear -kellojen kehitys on parin viime vuoden aikana junnannut paikoillaan, koska Googlen yhteistyökumppani Qualcomm ei ole julkaissut uutta järjestelmäpiiriä kelloille, edellinen versio oli liian vanhanaikainen. Tämä on vihdoin muuttumassa, sillä Qualcomm julkaisi Android Wear -kelloihin tarkoitettua uuden piirin syksyllä 2018 [3].

Työn tarkoituksena on selvittää ja tutustua terveydenhuollon sovellusten lakisääteisiin vaatimuksiin EU:n ja Suomen alueella sekä selvittää, miten Android Wear -kelloon voi tehdä terveydenhuollon kannalta turvallisen sovelluksen ja toteuttaa yksinkertainen Android Wear -sovellus.

## 2 Terveydenhuollon laitteita koskevat vaatimukset

Suomen terveydenhuollon laite- ja tarvikelaissa (629/2010) on määritellyt terveydenhuollon laite seuraavasti:

*Terveydenhuollon laitteella ja tarvikkeella tarkoitetaan instrumenttia, laitteistoa, välinettä, ohjelmistoa, materiaalia tai muuta yksinään tai yhdistelmänä käytettävää laitetta tai tarviketta, jonka sen valmistaja on tarkoittanut käytettäväksi ihmisen*

- a) *sairausten diagnosointiin, ehkäisyyn, tarkkailuun, hoitoon tai lievitykseen*
- b) *vamman tai vajavuuden diagnosointiin, tarkkailuun, hoitoon, lievitykseen tai kompensointiin,*
- c) *anatomian tai fysiologisen toiminnon tutkimiseen, korvaamiseen tai muunteluun; taikka*
- d) *hedelmöitymisen säätelyyn. [6.]*

Tämä tarkoittaa, että kaikki laitteet ja sovellukset, jotka on tarkoitettu lääkinälliseen hoitoon, ehkäisyyn tai sairauksen diagnosointiin, määritellään terveydenhuollon laitteiksi.

EU-direktiiveissä käytetään englanniksi termiä "medical device" terveydenhuollon laitteista. Suora käänös tästä olisi lääkinällinen laite, mutta Valviran ja Suomen laissa käytetään termiä terveydenhuollon laite. Tämän takia kaikissa tulevilla luvuilla käytetään termiä terveydenhuollon laite. [5.]

Ohjelmistotuotannossa rajatapauksia saattaa tulla helposti vastaan, minkä takia on hyvä tuntee terveydenhuollon laitedirektiivit ja lainsäädännöt. Tärkeimmät direktiivit, jotka laitteen pitää huomioida, ovat MD-, IVD- ja AIMD-direktiivit. Mutta laitteen käyttötarkoituksen ja määrittelyn mukaan siihen voi liittyä myös useita muita direktiivejä ja lainsäädäntöjä. Terveydenhuollon laitedirektiivit sisältävät luettelon kaikista olennaisista vaatimuksista, joiden avulla laitteen valmistaja voi osoittaa, että kaikki direktiivissä mainitut vaatimukset on käyty läpi ja arvioitu. Jos laite täyttää terveydenhuollon laitteen kriteerit, tuotteen täytyy seurata direktiivejä ja asetuksia ja sen tulee olla CE-merkitty. Jos taas tuote ei täytä terveydenhuollon laitekriteerejä, tuote ei saa olla CE-merkitty terveydenhuollon laitteeksi. [5.]



Laitteen valmistaja tai edustaja ei voi saattaa laitetta markkinoille ennen kuin terveydenhuollon laite täyttää kaikki sille asetetut vaatimukset: laitteen käyttötarkoitus on määriteltävä sille asetettujen direktiivien mukaisesti, ja kaikki laitteeseen vaikuttavat säädökset ja direktiivit on huomioitu. Terveydenhuollon laitteen tuoteluokitus on tehty direktiivien mukaisesti, myös kaikki asiakirjat vaatimustenmukaisuuden osoittamiseksi on olemassa ja saatavilla. Laitteen vaatiessa on myös olemassa asianmukainen laadunhallintajärjestelmä ja ilmoitetun laitoksen hyväksyntä. Lopuksi laitteen vaatimustenmukaisuustodistus on allekirjoitettu, tuotteeseen on kiinnitetty CE-merkki ja ilmoitus tuoterekisteriin on tehty. [5.]

Ennen terveydenhuollon laitteen käyttöönottoa, joka voidaan tehdä vasta tuotteen markkinoille saattamisen jälkeen, tulee valmistajan tai edustajan suunnitella, valmistaa, pakata ja merkitä laite siten, että se soveltuu valmistajan tarkoittamaan käyttötarkoitukseen. Laitteet tulee myös käyttötarkoituksen mukaisesti käytettynä saavuttaa sille suunniteltu toimivuus ja suorituskyky. Laitteen asianmukainen käyttö ei saa myöskään vaarantaa potilaan, käyttäjän tai muun henkilön terveyttä tai turvallisuutta eikä ympäristöä. [5.]

## 2.1 MD-, IVD- ja AIMD-direktiivit

EU on asettanut kolme direktiiviä, jonka avulla se säätelee terveydenhuollon laitteita. Direktiiveissä määritellään tavoitteet, jotka jokaisen jäsenvaltion on toteutettava kansallisesti omilla lakisäädöksillä. Suomessa EU:n direktiivien vaatimukset on otettu huomioon terveydenhuollon laite- ja tarvikelaissa (laki 629/2010). Tämä takaa, että noudattamalla Suomen terveydenhuollon laite- ja tarvikelakia, tulee EU:n asettamat vaatimukset noudatettua. [5.]

Suomen terveydenhuollon laki kattaa EU-vaatimukset hyvin, mutta on silti hyvä tuntea alkuperäiset direktiivitekstit. EU:n asettamat kolme direktiiviä ovat MD-, IVD- ja AIMD-direktiivit, jotka on määriteltävä dokumenteissa:

- *Lääkinnällisistä laitteista annettu neuvoston direktiivi 93/42/ETY (MD)*
- *In vitro -diagnostiikkaan tarkoitetuista lääkitämisistä laitteista annettu Euroopan parlamentin ja neuvoston direktiivi 98/79/EY (IVD)*
- *Aktiivisia implantoitavia lääkitämisistä laitteita koskeva jäsenvaltioiden lainsäädännön lähentämisestä annettu neuvoston direktiivi 90/385/ETY (AIMD). [6.]*

Seuraavissa luvuissa määritellyt vaatimukset ovat Suomen lain ja EU-direktiivien määrittämiä ja toteuttamalla tuote niiden pohjalta voi tuotteen tulkita vaatimusten täyttäväksi terveydenhuollon laitteeksi.

## 2.2 Potilastietoihin liittyvät lait

Sovellukset voidaan myös tulkita terveydenhuollon laitteiksi, kun sovellusta käytetään yksin tai yhdessä muiden terveydenhuollon laitteiden kanssa hankimaan tietoja potilaan terveydentilan tai sairauden havaitsemiseksi, diagnosoimiseksi, valvomiseksi tai hoitamiseksi. Tämä sisältää myös mobiililaitteisiin suunnitellut sovellukset, potilastietojärjestelmät ja potilastietojärjestelmien osat. [7.]

Suomessa sosiaali- ja terveydenhuollon asiakastietojen sähköisen käsittelyn laki (159/2007) määrittelee yleiset vaatimukset tietojärjestelmille ja niiden valmistajille. Laki sisältää säännökset sosiaali- ja terveydenhuollon asiakastietojen sähköisen käsittelyn yleisistä vaatimuksista. Vaatimusten tarkoituksena on turvata potilastietojen käytettävyyttä, eheys ja säilyminen sekä asiakkaan yksityisyyden suoja. [8.]

Asiakastietojen sähköisen käsittelyn laki vaatii, että asiakkaan potilastiedot tulee olla aina saatavilla, kun asiakas tarvitsee niitä. Asiakastietojen tulee myös säilyä muuttumattomina eikä kukaan ulkopuolinen saa päästä käsiksi niihin. Asiakastietojen ylläpitäjän täytyy pitää rekisteriä omien asiakastietojärjestelmiensä ja asiakasrekisteriensä käyttäjistä sekä näiden käyttöoikeuksista sekä tunnistaa luotettavasti asiakas, sosiaali- ja terveydenhuollon palvelujen antaja tai muu asiakastietojen käsittelyn osapuoli. [8.]

## 2.3 EU:n yhdenmukaistetut standardit

Terveydenhuollon laitteisiin liittyy valtava määrä standardeja, useat näistä voivat olla laitteen valmistajalle hyödyllisiä taloudellisesti sekä auttaa pääsemään nopeammin ja varmemmin markkinoille. Osa kyseisistä standardeista on saanut erikoisaseman, koska niitä seuraamalla varmistetaan vaatimustenmukaisuuden toteutuminen hyvin yksityiskohtaisesti. Euroopan unionin hyväksymiä standardeja kutsutaan yhdenmukaistetuiksi standardeiksi. Nämä voi tunnistaa etuliitteestä "EN". Suomessa hyväksytyt standardit voi taas tunnistaa "SFS"-etuliitteestä. Jos standardilla on molemmat etuliitteet, se on

hyväksytyt sekä EU:ssa että Suomessa. Suomessa Suomen standardisoimisliitto SFS ry laatii ja hyväksyy standardit. [5; 9.]

EU:n säädökset ja näihin nojaavat kansalliset säädökset sisältävät ainoastaan tuotteita koskevat olennaiset vaatimukset ja vaatimustenmukaisuuden osoittamiseksi tarvittavat asiat. EU-direktiiveissä ei viitata yksilöityihin standardeihin, vaan kehoitetaan käyttämään yhdenmukaistettuja standardeja. Tämän takia terveydenhuollon laitedirektiiveissä ei vaadita käyttämään tiettyjä standardeja. Laitteen valmistajan pitää jollakin tapaa osoittaa, että direktiivien vaatimukset täyttyvät. Olemassa oleviin standardeihin viittamalla valmistaja voi helposti osoittaa, että kaikki olennaiset vaatimukset on täytetty. Käytännössä tämä tarkoittaa sitä, että vain laitteet, jotka täyttävät standardien vaatimukset, hyväksytään terveydenhuollon laitteiksi kansainvälisille markkinoille. [5.]

## 2.4 Käyttötarkoitus

Käyttötarkoitus on määritelty Suomen terveydenhuollon laite- ja tarvikelaissa. Lain mukaan käyttötarkoitus tarkoittaa käyttöä, johon terveydenhuollon laite valmistajan merkinnöissä, käyttöohjeessa tai myynninedistämistä koskevassa aineistossa annettavien tietojen mukaan on tarkoitettu. Terveydenhuollon laitteen valmistaja kertoo käyttötarkoituksella, mihin laitetta voi käyttää ja mihin ei. Tällä laitteen valmistaja voi siirtää vastuun käyttäjälle, mikäli hän käyttää laitetta käyttötarkoituksen vastaisesti. [5.]

Laitteen käyttötarkoitus määrittää laitteen vaatimusten laajuuden ja tiukkuuden. Mitä laajempi käyttötarkoitus laitteella on määritelty, sitä isompi taakka valmistajalla on osoittaa, että kaikki alueet on otettu huomioon käyttötarkoituksessa. Laitteen tuoteluokka määräytyy käyttötarkoituksen perusteella, tuoteluokka vaikuttaa vaatimusten tiukkuuteen. Mitä korkeampi laiteluokka on, sitä tiukemmat ja kattavammat riskienhallinnan ja kliinisten tutkimusten pitää olla. [5.]

## 2.5 Tuoteluokittelu

### 2.5.1 Terveydenhuollon laitteet (MD)

Terveydenhuollon laitteiden tuoteluokka määräytyy MD-direktiivin liitteen yhdeksän asetettujen luokittelusääntöjen mukaan. Direktiivin 9 artiklassa terveydenhuollon laitteet jaetaan neljään eri tuoteluokkaan, jotka ovat luokka I, luokka IIa, luokka IIb ja luokka III. Luokan I laitteista pitää ilmoittaa, jos laite sisältää mittaustoiminnon (Im) tai laite

on steriili (Is). Laitteet luokitellaan sen mukaan, miten suuri riski niiden käyttötarkoituksen mukaisen käytöstä aiheutuu käyttäjälle. Luokan III laitteilla on suurin riski ja luokan I pienin. [5.]

MD-direktiivin liite yhdeksän sisältää 18 eri luokitussääntöä. Näiden avulla pyritään arvioimaan laitteen riskien vakavuutta. Riskiarviointiin vaikuttaa neljä eri tekijää: kuinka kauan laite on kosketuksissa käyttäjään ja mihin osaan kehoa se on kosketuksissa, laitteen invasiivisuuden aste ja onko laite aktiivinen. [5.]

Laite luokitellaan riskialttiimmaksi, mitä pidemmän ajan se on yhtäjaksoisesti kosketuksissa potilaaseen. Kosketusaika tulkitaan tilapäiseksi, kun laite on kosketuksessa vähemmän kuin 60 minuuttia, lyhytaikaiseksi, jos vähemmän kuin 30 päivää ja pitkäaikaiseksi, jos enemmän kuin 30 päivää. Yhtäjaksoisella käytöllä tarkoitetaan laitteen todellista keskeytyksetöntä käyttöä käyttötarkoituksessaan. Jos laite korvataan identtisellä laitteella tai jos samaa laitetta käytetään heti uudelleen samaan käyttötarkoitukseen, katsotaan, että käyttö on yhtäjaksoista. [5.]

Laitteen invasiivisuusaste vaikuttaa luokitukseen. Laite tulkitaan invasiiviseksi, jos se viedään kokonaan tai osittain kehon sisään luonnollisen aukon kautta tai kirurgisella toimenpiteellä. Myös laite, joka on implantoiduttu osittain ihmiskehoon kirurgisen toimenpiteen avulla ja jätetty paikoilleen toimenpiteen jälkeen vähintään 30 päiväksi luetaan invasiivisiksi laitteiksi. [5.]

Terveydenhuollon laitteen luokitukseen vaikuttaa myös, onko se aktiivinen laite vai ei. Terveydenhuollon laite on aktiivinen laite, kun se käyttää jotain ulkoista energialähdettä. Itsenäiset ohjelmistot luetaan aktiivisiksi laitteiksi, koska ne käyttävät jotakin ulkoista energialähdettä, esim. puhelimen akkua. [5.]

Laitteen sijainti ihmiskehossa vaikuttaa laitteen riskiluokitukseen. Jos laite on kosketuksissa ihmiskehon toiminnan kannalta kriittisellä alueella, riskiluokitus on aina korkeampi. Esimerkiksi laite, joka on kosketuksissa keskushermoston tai verenkiertoelimistöön, aiheuttaa huomattavasti isomman riskin potilaalle. [5.]

Valviran asettamat vaatimukset terveydenhuollon laitteille on esitetty liitteen luvussa yksi.

### 2.5.2 In vitro diagnostiset -laitteet (IVDMD)

In vitro -diagnostiikka tarkoittaa potilaasta tai terveestä henkilöstä otetun lääketieteellisen näytteen laboratoriotutkimusta. In vitro -diagnostiikka laitteet ovat siis laitteita, jotka on tarkoitettu suorittamaan in vitro -tutkimuksia. [5.]

IVD-direktiivin liite kaksi jakaa in vitro -diagnostiikkalaitteet kolmeen ryhmään. Laite on joko luettelo A tai B:n laite tai näiden luetteloiden ulkopuolella jäävä laite. Lisäksi IVD-laitteet jaetaan käyttötarkoituksensa mukaisesti itse suoritettavan testauksen laitteisiin, suorituskyvyn arvioinnin laitteisiin ja muihin laitteisiin. IVD-direktiivin liitteen kaksi luettelon A kuuluvilla laitteilla on ankarammat vaatimukset, kuin luettelon B laitteilla. [5; 16.]

Valviran asettamat vaatimukset In vitro -diagnostiikkalaitteille on esitetty ensimmäisen liitteen luvussa kaksi.

### 2.6 Laadunhallintajärjestelmä (QMS)

Direktiivien täyttämällä laadunhallintajärjestelmällä laitteenvalmistaja pystyy todistamaan, että tuote täyttää kaikki siihen kohdistuvat vaatimukset laitteen elinkaaren aikana sekä normaali- että ongelmatilanteissa. Sertifioitua laadunhallintajärjestelmää ei vaadita kaikilta terveydenhuollon laitevalmistajilta. Laadunhallintajärjestelmän pakollisuus määräytyy tuotteen tuotelukituksen mukaan, mutta useasti suositellaan että kaikille terveydenhuollon laitteille toteutettaisi laadunhallintajärjestelmä. [5.]

Laadunhallintajärjestelmä on pakollinen terveydenhuollon laitteiden tuoteluokan IIb-, IIa- ja III-laitteissa sekä In vitro -diagnostiikka A- ja B-listan laitteissa. Ainoastaan luokan I terveydenhuollon laitteilta ei vaadita laadunhallintajärjestelmää, mutta tulevaisuudessa direktiivit voivat vaatia myös luokan I laitteilta sitä. Laadunhallintajärjestelmän olemassaolo helpottaa myös CE-merkin hakuprosessia. Tämän takia suositellaan, että kaikki terveydenhuollon laitteiden valmistaja tekisivät laadunhallintajärjestelmän. [5.]

Laadunhallintajärjestelmä tulee täyttää kaikki vaatimukset, jotka laitedirektiiveissä on määritelty. Vaatimusten määrittelyyn on luotu eurooppalainen standardi, SFS-EN ISO 13845. Käyttämällä standardia laadunhallintajärjestelmän laadinnassa, laitteenvalmistaja voi helposti osoittaa ja ilmoitettu laitos pystyy arvioimaan, että laite täyttää kaikki lakisääteiset vaatimukset. [10.]

ISO-13845-standardin pohjalta laadittu laadunhallintajärjestelmä on prosessimainen toimintamalli, ja tämän takia linjassa ISO-9001-standardin kanssa. Erona ISO-9001- ja ISO-13845-standardien välillä on se, että ISO-13845 ottaa huomioon terveydenhuollon laitteiden vaatimat lakisääteiset vaatimukset. Tämän takia joitakin osia on voitu poistaa tai lisätä ISO-13845-standardista verrattuna ISO-9001-standardiin. ISO-13845-standardin päämäärä on vaatimustenmukaisuuden varmistaminen, kun taas ISO-9001-standardi korostaa jatkuvaa parantamista. [5; 10.]

Sertifioidun laadunhallintajärjestelmän saamiseksi tulee laitteen valmistajan suorittaa auditointi. Ilmoitettu laitos auditoi laadunhallintajärjestelmän ja myöntää hyväksytyin auditoinnin jälkeen laadunhallintajärjestelmän EY-todistuksen, joka on voimassa vain määrätyn ajan. Ilmoitettu laitos suorittaa myös vuosittaisia seuranta-auditointeja todistuksen saamisen jälkeen. EY-todistus myönnetään valmistajakohtaisesti ja muille valmistajan tuotteille ei vaadita uutta auditointia, ellei tuotteen vaatimustenmukaisuuden arviointi noudata eri menettelyä. Uudet tuotteet on kuitenkin lisättävä todistukseen, ennen kuin sitä voidaan soveltaa. [5.]

## 2.7 Kliininen arviointi

Laitteen soveltuvuus käyttötarkoitukseensa on varmistettava kliinisellä arvioinnilla. Kliinisen arvioinnin tulee sisältää arvio sivuvaikutusten ja haitta-hyötysuhteen hyväksyttävyydestä. Arvioinnilla vahvistetaan, että laitteen ominaisuudet ja suorituskyky ovat vaatimustenmukaisia laitteelle tarkoitetuissa käyttöolosuhteissa. [5; 11.]

Direktiivin 93/42/ETY Liite 10 määrittelee, mistä tarvittavien tietojen tulee olla kliiniseen arviointiin. Tiedot voivat olla peräisin jostakin muusta terveydenhuollon laitteen tai tarvikkeen kliinisestä laitetutkimuksesta. Tiedot voivat myös olla muusta virallisesta kliinisestä tutkimuksesta tai muusta tutkimuksesta, joka on raportoitu tieteellisessä kirjallisuudessa. Tiedot voivat olla myös kliinisiä kokemuksia kuvaavista julkaisuista tai julkaisemattomista raporteista. Näissä tapauksissa laitevalmistajan pitää osoittaa, että laitteet tai tarvikkeet ovat samankaltaisia raportin tai julkaisun laitteen kanssa. [11.]

Jokaisesta markkinoille saatetusta terveydenhuollon laitteesta on tehtävä kliininen arviointi. Jos kyse on implantoitavasta tai III tuoteluokan laitteesta, on tuotteelle tehtävä kliininen laitetutkimus. Laitevalmistajan ei tarvitse tehdä kliinistä laitetutkimusta, jos tuotteen valmistaja pystyy vakuuttavasti perustella, miksi kliinisen arvioinnin voi tehdä aikaisempiin kliinisiin tietoihin nojautuen. [5; 11.]

Kliinisistä laitetutkimuksista on tehtävä ilmoitus Valviralle ja sen maan viranomaisille, jossa tutkimus suoritetaan ennen tutkimuksen aloittamista. Ilmoitusta ei tarvitse tehdä CE-merkinnällä varustetusta terveydenhuollon laitteesta, jos tutkimuksessa ei poiketa valmistajan määrittelemästä käyttötarkoituksesta ja käyttöohjeista. Aktiivisella implantoitavalla laitteella tehtävästä tutkimuksesta tulee tehdä ilmoitus aina. [5; 11.]

Tuoteluokan Ila ja IIb pitkäaikaisen invasiivisen, tuoteluokan III ja implantoitavan laitteen kliinisestä laitetutkimuksesta on tehtävä ilmoitus Valviraan 60 päivää ennen tutkimuksen käynnistämistä. Tutkimuksen voi aloittaa, jollei Valvira ole tätä ennen pyytänyt lisäselvitystä tai ilmoittanut kieltävänsä tutkimukset ihmisen terveydelle tai turvallisuudelle aiheutuvan vaaran vuoksi. Muille laitteilla riittää, että ilmoitus on jätetty Valviralle ennen tutkimusten käynnistämistä. [11.]

Tutkimuksen aikana todetuista vaaratilanteista on välittömästi ilmoitettava Valviralle ja muun maan viranomaisille, jos tutkimus tehdään Suomen ulkopuolella. Säteilyä synnyttävien laitteiden käyttöön liittyvistä poikkeavista tapahtumista ja vaaratilanteista on lisäksi ilmoitettava Säteilyturvakeskukselle. Jos tutkimuksen aikana epäillään laitteen aiheuttaneen haittaa potilaalle, on asianmukaiset yksityiskohdat merkittävä tehtävätutkimuspöytäkirjaan ja potilaan sairaskertomukseen. [5.] [11.]

Toimeksiantajan on annettava selvitys tutkimuksen tuloksista Valviralle mahdollisimman pian ja viimeistään yhden vuoden kuluttua tutkimuksen päättymisestä. Kaikkien tutkimukseen osallistuneiden tutkijoiden on allekirjoitettava kyseinen loppuraportti. [11.]

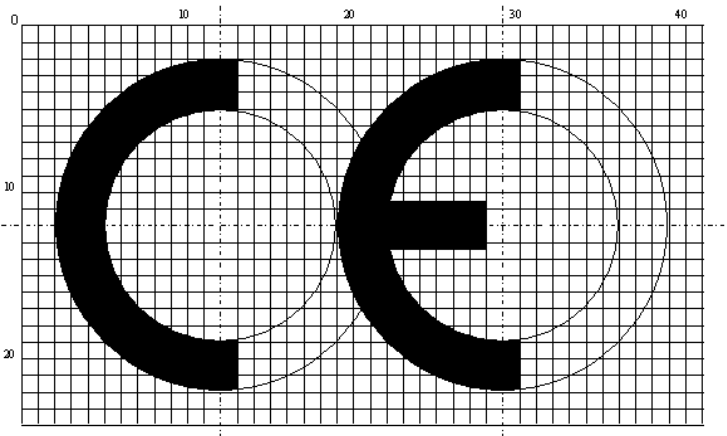
Valmistaja tai valtuutettu edustaja on velvoitettu säilyttämään kliinisiä tutkimuksia koskevat asiakirjat, joiden perusteella vaatimustenmukaisuuden arviointi on mahdollista. Asiakirjoja on säilytettävä vähintään 5 vuoden ajan tutkimuksen päättymisestä ja implantoitavien laitteiden osalta 15 vuoden ajan. [11.]

## 2.8 CE merkintä

Valviran 2/2011 määräyksessä on säädelty CE-merkinnän käytöstä. CE-merkinnällä valmistaja osoittaa, että terveydenhuollon laite täyttää kaikki sitä koskevat vaatimukset ja on valmis markkinoille saattamiseen. Vaatimukset sisältävät terveydenhuollon laitetta koskevat direktiivit ja mahdolliset muut direktiivit ja lainsäädännöt jotka koskevat laitetta. Tämä tarkoittaa, että pelkästään AIMD-, MD- ja IVD-direktiivien vaatimusten noudattaminen ei riitä CE-merkinnän saamiseksi. Tuotteen valmistajan on tunnistettava

mahdolliset muut direktiivit ja lainsäädännöt, jotka saattavat vaikuttaa tuotteeseen ja toteutettava laite käyttäen niitä. [5; 13.]

CE-merkintä on kiinnitettävä näkyvällä, helposti luettavalla ja pysyvällä tavalla laitteeseen tai sen steriiliin tuotepakkaukseen sekä laitteen käyttöohjeeseen. Mikäli merkintä laitteeseen tai tuotepakkaukseen ei onnistu, CE-merkinnän on oltava laitteen käyttöohjeessa. In vitro -diagnostiikka laitteessa CE-merkinnän on oltava myös myyntipakkauksessa. Ohjelmistossa CE-merkin tulee olla joko ohjelman käynnistyssivulla tai kohdassa jossa valmistaja antaa yleistä tietoa tuotteesta ja sen versionumerosta. [5; 13.]



Kuva 2. Virallinen CE-merkki. CE-merkin kuvasuhde pitää pysyä samana, jos merkin kokoa muuttaa. [12]

CE-merkin koolle, mallille ja tavalle, jolla CE-merkki kirjoitetaan, on määritelty minimivaatimukset direktiiveissä (katso kuva 2). CE-merkin kokoa saa muuttaa, kunhan se on suurempi kuin 5 mm korkeudeltaan, mutta kuvasuhteen pitää pysyä aina samana. Jos vaatimustenmukaisuuden osoittamiseen on käytetty ilmoitettua laitosta, on liitettävä laitoksen nelinumeroinen tunnusnumero CE-merkin kanssa. Jos laitosta ei ole käytetty, ei tarvitse numeroa liittää CE-merkkiin. [5; 13.]

Tuotteeseen ei saa kiinnittää sellaisia merkintöjä, jotka muistuttavat CE-merkintää tai heikentävät CE-merkin näkyvyyttä tai luettavuutta. CE-merkinnällä ei saa varustaa yksilölliseen käyttöön valmistettua laitetta, kliinisiin tutkimuksiin tarkoitettua laitetta, toimenpidepakkausta, järjestelmää, suorituskyvyn arviointiin tarkoitettua laitetta eikä terveydenhuollon toimintayksikön omana laitevalmistuksena valmistettua laitetta. Sterilointipalvelun tuottaja ei myöskään saa kiinnittää steriloimaansa laitteeseen CE-merkintää. [13.]



Mikäli CE-merkintä on tehty väärin perusteiden tai jos se puuttuu, Valvira voi kieltää laitteen valmistuksen, myynnin, maastaviennin tai muun jakelun riippumatta siitä, onko jakelu maksullista vai ilmaista, tai velvoittaa valmistajan ryhtymään tarpeellisiin toimenpiteisiin laitteen saattamiseksi säännösten ja määräysten mukaiseksi määräajassa tai asettaa laitteen saatavuudelle ja käytölle ehtoja. [13.]

## 2.9 Laiterekisteröinti

Laitteen täytettyä CE-merkin vaatimukset valmistajan on vielä ilmoitettava laite laiterekisteriin. Ilmoituksen tekeminen on osa direktiivien vaatimuksia, se on tehtävä ennen kuin laite voidaan saattaa markkinoille. Suomessa toimiva yritys tekee ilmoituksen Valviraan ja tarvittaessa myös muiden EU-jäsenvaltioiden toimintavaltaisille viranomaisille. [5.]

Ilmoitus on toimitettava hyvissä ajoin valvonnan tarkoituksenmukaisen hoitamisen mahdollistamiseksi. Laitteen rekisteröinti-ilmoitus on toimitettava Valviralle ennen markkinoille saattamista, palvelun tuottamista tai maahantuonnin aloittamista. Oma-valmistuksena valmistetusta laitteesta riittää ilmoitus ennen laitteen käyttöönottoa. [5; 14.]

Ilmoituksen tekijän on oltava oikeutettu edustamaan yritystä tai elinkeinonharjoittajaa sekä oltava vastuussa ilmoitukseen annettavista tiedoista. Ilmoituksen tiedot on oltava ajan tasalla ja tiedon muuttuessa siitä on tehtävä muutosilmoitus. Muutosilmoitus on tehtävä ennen kuin muutoksen tiedot astuvat voimaan. Rekisteröinnintekijän on myös tehtävä poistoilmoitus, kun laitteen valmistus loppuu tai valmistus siirtyy toiselle yritykselle. [5; 14.]

Ilmoitukseen vaadittavat tiedot riippuvat toimijasta ja toimitetavasta laitteesta. Ilmoitukseen vaaditaan rekisteröintitiedot, valmistajan, valtuutetun edustajan tai maahantuojan yhteystiedot, laitetiedot ja ilmoitetun laitoksen todistuksen tiedot, jos ilmoitettua laitosta on käytetty. [5; 14.]

## 2.10 Vaaratilanneilmoitukset

Markkinoille saattamisen jälkeen laitteen valmistajalla on vielä useita velvoitteita, mutta terveydenhuollon laitedirektiivien kannalta tärkein on potilasturvallisuus. Potilasturvan keskeisempänä osana ovat vaaratilanneilmoitukset, jotka ovat pakollisia kaikille tervey-

denhuollon laitevalmistajille. Vaaratilanneilmoituksilla pyritään estämään potilas- ja käyttäjävahingot sekä minimoimaan vahinkojen laajuus ongelmatilanteissa. [5.]

Terveydenhuollon laite- ja tarvikelaki 629/2010 pykälä 15 määrittelee vaaratilanteen tapahtumaksi, joka johtaa tai olisi voinut johtaa potilaan, käyttäjän tai muun henkilön terveyden vaarantumiseen ja joka johtui terveydenhuollon laitteen ominaisuuksista, suorituskyvyn poikkeamasta tai häiriöstä, riittämättömästä merkinnästä tai riittämättömästä tai virheellisestä käyttöohjeesta. [6.]

Valmistajat ja valtuutetut edustajat, joiden markkinoille saattama terveydenhuollon laite on aiheuttanut vaaratilanteen Suomessa, ovat velvollisia tekemään vaaratilanneilmoitukset Valviraan. Lisäksi säteilyä synnyttävien laitteiden käyttöön liittyvistä poikkeavista tapahtumista ja vaaratilanteista on ilmoitettava Säteilyturvakeskukseen. Suomalaisen valmistajan on myös ilmoitettava Valviralle, jos vaaratilanne tapahtuu Euroopan unionin ulkopuolella. Vaaratilanteen tapahtuessa muussa Euroopan unionin valtiossa, vaaratilanneilmoitus tehdään kyseisen valtion toimivaltaiselle viranomaiselle. Myös terveydenhuollon laitteen ilmoitetulle laitokselle on suositeltavaa tehdä ilmoitus. [5.]

Toiminnanharjoittajalla on velvollisuus ilmoittaa laitteen valmistajalle tai valtuutetulle edustajalle tietoonsa tulleesta vaaratilanteesta, jonka on todettu tai epäilty johtuneen terveydenhuollon laitteesta olevasta viasta tai puutteellisuudesta. Toiminnanharjoittajan on ilmoituksen jälkeen avustettava viranomaisia vaaratilanteen selvittämisessä raportoimalle tarvittavat tiedot ja toimimaan saatujen ohjeiden mukaisesti. [5.]

Jos tekniset tai lääketieteelliset syyt johtavat valmistajan järjestelmällisesti poistamaan samaa tyyppiä olevat laitteet markkinoilta tai tekemään järjestelmällisiä korjaavia toimenpiteitä, valmistajan tai valtuutetun edustajan on ilmoitettava syyt sekä korjaavat toimenpiteet viranomaiselle. Toimenpiteistä tulee myös ilmoittaa asiakkaille ja ammattimaisille käyttäjille turvallisuustiedotteella. [5.]

Kun valmistaja tai valtuutettu edustaja saa tiedon vaaratilanteesta, on ilmoitus tehtävä mahdollisimman nopeasti. Vaaratilanteita varten on oltava valmiina selkeä toimintaohje, jolla varmistetaan nopea tiedonhankinta asiakkailta ja yhteistyökumppaneilta sekä tiedon nopea eskalointi oikealle henkilölle. Ilmoitusta on mahdollista täydentää jälkikäteen, jos selvitystyön valmistumisen jälkeen selviää uutta tietoa. [5.]

Ilmoittamisen määräajat ovat Euroopan unionissa kolmitasoiset ja Valvira on määritellyt ne Suomessa. Vaaratilanteen vakavuus määrää aikaikkunan pituuden. Valvira on asettanut vakavalle kansanterveyden uhalle kahden päivän ilmoitusikkunan. Ilmoitus pitää tehdä kahden päivän sisällä siitä, kun laitevalmistaja on saanut tiedon uhasta. Jos laitteen käyttö on johtanut kuolemaan tai vakavaan terveydentilan heikentymiseen, pitää siitä tehdä ilmoitus 10 vuorokauden kuluttua, kun laitevalmistaja saa tiedon. Muissa tapauksissa on ilmoitettava välittömästi, kun valmistaja pystyy todistamaan, että laite oli yhteydessä johonkin tapahtumaan. [15.]

Valviran vaaratilanneilmoitusmääräyksessä 1/2010 mainitaan kaikki tiedot jotka valmistajan tai valtuutetun edustajan tulee kirjata ylös ilmoituksessa. Ilmoituksessa annetaan valmistajan yhteystiedot ja tilanteessa, jossa valmistaja on EU:n ulkopuolelta, annetaan valtuutetun edustajan yhteystiedot. Yhteyshenkilön tiedot tulee myös antaa, jos muu henkilö tekee ilmoituksen valmistajan puolesta. Vaaratilanteen aiheuttaneesta laitteesta annetaan laiteryhmää kuvaavan nimikkeen koodi, laitteen tuoteluokka ja laitteen yksilöintiin ja tunnistamiseen tarvittavat tiedot. Ilmoitetun laitoksen tunnusnumero tulee myös antaa, jos laitteenvaatimustenmukaisuuden arviointiin käytettiin ilmoitettua laitosta. Vaaratilannetapahtumasta kerrotaan aika, paikka ja annetaan tapahtumakuvaus sekä tiedot tapahtuman seurauksista potilaalle, käyttäjälle tai muulle henkilölle. Tilanne, jossa potilaan sukupuoli, ikä tai paino ovat voineet vaikuttaa vaaratilanteeseen, tulee myös kertoa. [5; 15.]

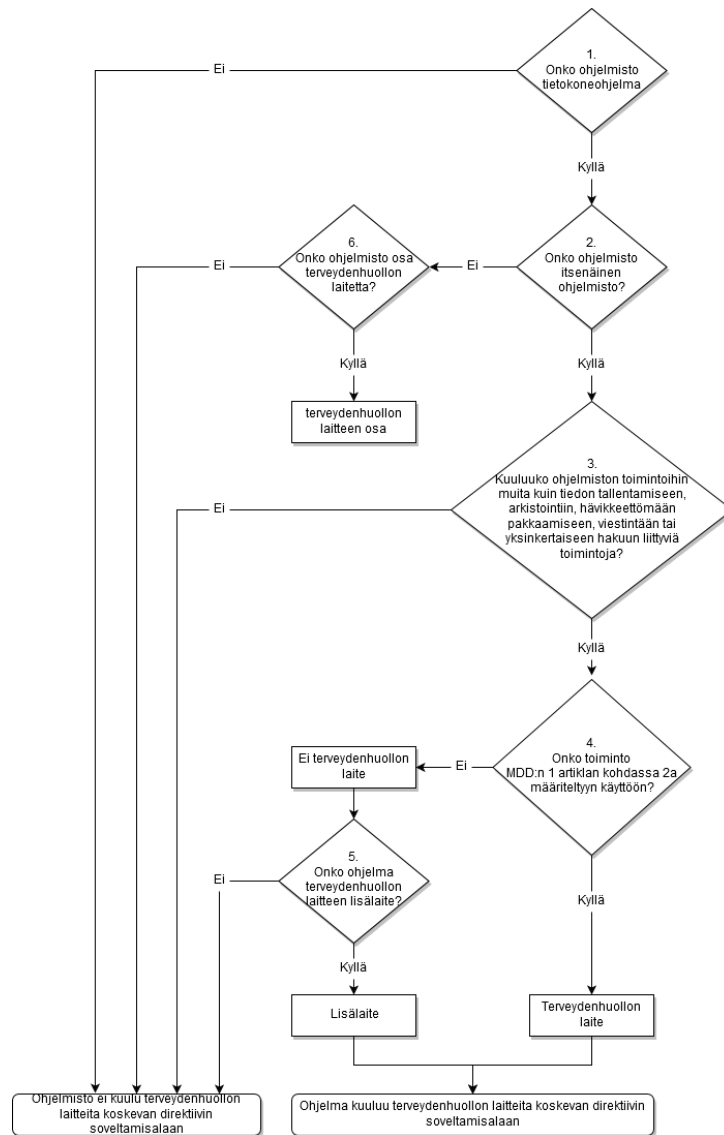
Markkinoilta poistamisen tai korjaavien toimenpiteiden kohteena olevan laitteen osalta ilmoitetaan samat tiedot kuin vaaratilanneilmoituksessakin. Lisäksi ilmoitetaan toimenpiteisiin johtaneet syyt ja mahdolliset vaikutukset sekä markkinoilta poistamisen tai korjaavien toimenpiteiden toteuttamisaikataulu. Jos toimenpiteistä ilmoitetaan käyttäjille, tulee turvallisuustiedote sisällyttää ilmoitukseen ja antaa viranomaisille kaikki mahdolliset muut tarvittavat tiedot ja asiakirjat, jotka saattavat auttaa viranomaista jäljittämään ilmoitusta koskevan laitteen markkinoilta poistamista tai korjaavien toimenpiteiden tekemistä. [5.]

### 3 Terveydenhuollon ohjelmiston luokittelu

#### 3.1 Terveydenhuollon ohjelmiston määrittelykriteerit

Ohjelmisto voi olla terveydenhuollon laite, jos se on itsenäinen. Itsenäisellä ohjelmistolla tarkoitetaan ohjelmistoa, joka ei kuulu terveydenhuollon laitteeseen, kun se tuodaan markkinoille. Itsenäisellä ohjelmistolla on oltava lääkinällinen tarkoitus, jotta se voidaan luokitella terveydenhuollon laitteeksi. Ainoastaan valmistajan kuvailema käyttötarkoitus vaikuttaa määrittelyyn ja luokitteluun eikä laitteelle annettu nimi. Itsenäinen ohjelmisto, joka ei täytä terveydenhuollon laitteen määritelmää mutta jonka valmistaja on tarkoittanut terveydenhuollon laitteen lisälaitteeksi, kuuluu MD-direktiivin soveltamisalaan. Vaikka laite ei täytä terveydenhuollon laitteen määritelmää tai se ei kuulu direktiivien soveltamisalaan, siihen saatetaan silti soveltaa muita yhteisön sääntöjä ja/tai kansallisia lainsäädäntöjä. [17.]

Kuvan 3 kaavio opastaa vaiheista, joilla voi selvittää, onko itsenäisen ohjelmisto terveydenhuollon laite ja kuuluuko se MD-direktiivin soveltamisalaan.



Kuva 3. Kaavio, joka auttaa määrittelemään ohjelmiston terveydenhuollon laitteeksi. [17]

Ensimmäinen vaihe on selvittää, onko ohjelmisto tietokoneohjelma. Jos ohjelmisto ei ole tietokoneohjelma, se ei ole terveydenhuollon laite vaan digitaalinen dokumentti. Tämän jälkeen kannattaa selvittää, onko tietokoneohjelma itsenäinen ohjelmisto vai ei. Vain itsenäinen ohjelmisto voi olla terveydenhuollon laite. Vaikka ohjelmisto ei ole itsenäinen, se voi silti olla osa jotain muuta terveydenhuollon laitetta. [17.]

Itsenäinen ohjelmisto, joka vain varastoi, arkistoi ja viestii tietoja tai toteuttaa yksinkertaisen tiedon haun, ei ole terveydenhuollon laite. Muissa tapauksissa ohjelmisto mahdollisesti tulkitaan terveydenhuollon laitteeksi mukaan lukien tapaukset, joissa ohjelmisto muuttaa tietoa esittämistarkoitusta varten ja sitä käytetään lääkinnällisessä käytössä. [17.]

Ohjelmisto on terveydenhuollon laite, jos valmistaja on määritellyt käyttötarkoituksen MD-direktiivin yksi ja kaksi artikloissa olevien käyttötarkoitusten mukaan. Muulloin ohjelmisto ei ole terveydenhuollon laite tai se on terveydenhuollon laitteen lisäosa. [17.]

### 3.2 Itsenäisen ohjelmiston luokittelu

Direktiivien mukaan itsenäinen ohjelmisto on aktiivinen terveydenhuollon laite. Itsenäinen ohjelmisto vaatii aina ulkoisen virtalähteen. Esimerkkinä on puhelinsovellus, joka käyttää puhelimen akkua ohjelmiston toimimiseen. Aktiivisiin terveydenhuollon laitteisiin sovelletaan MD-direktiivin liitteen yhdeksän sääntöjä 9-12. Liitteen yhdeksän soveltamissääntöjen kohdan 2.3 mukaan laitetta ohjaava tai sen toimintaan vaikuttava ohjelmisto kuuluu välittömästi samaan luokkaan. [17.]

Säännön 9 mukaan aktiiviset terapeuttiset laitteet, jotka on tarkoitettu tuottamaan tai vaihtamaan energiaa, kuuluvat luokkaan IIa. Jos aktiivinen laite tuottaa energiaa ihmiskehoon tai suorittaa energiansiirtoa mahdollisesti vaarallisella tavalla, laite kuuluu luokkaan IIb. [17.]

Säännön 10 mukaan diagnosointiin tarkoitettu aktiivinen laite kuuluu luokkaan IIa, jos se on tarkoitettu tekemään näkyväksi radiofarmaseuttisten tuotteiden jakaantumisen elävien organismien tutkimuksissa tai mahdollistamaan elintärkeiden fysiologisten toimintojen määrittämisen tai suoran tarkastuksen. Tilanteessa, jossa tarkasteltujen fysiologisten parametrien vaihtelu saattaa aiheuttaa potilaalle välitöntä hengenvaaraa, laite kuuluu luokkaa IIb. [17.]

Säännön 11 mukaan aktiivinen laite, joka on tarkoitettu annostelevaan kehossa lääkkeitä, biologisia nesteitä tai muita aineita tai poistamaan niitä kehosta, kuuluvat luokkaan IIa. Mikäli tästä voi aiheutua potilaalle vaaraa, se kuuluu luokkaan IIb. [17.]

Säännön 12 mukaan kaikki muut aktiiviset laitteet kuuluvat luokkaan I. Luokkaan I kuuluvassa itsenäisessä ohjelmistossa voi olla myös mittaustoimintoja. [17.]

### 3.3 Moduulit

Itsenäiset ohjelmistot voivat koostua useammasta moduulista ja jokaisella moduulilla ei välttämättä ole lääketieteellistä tarkoitusta. Tässä tapauksessa on kyseenalaista, voiko koko laite olla kokonaisuudessaan CE-merkitty. Moduulien, jotka kuuluvat terveyden-

huollon laitteiden direktiivien soveltamisalaan, on täytettävä kaikki terveydenhuollon laitteen direktiivin vaatimukset ja niissä on oltava CE-merkki. [17.]

Valmistajan on identifioitava terveydenhuollon laitteita koskevien direktiivien soveltamisalaan kuuluvien moduulien rajat ja rajapinnat käyttötarkoituksen perusteella. [17.]

Jos terveydenhuollon laitteita koskevan direktiivin soveltamisalaan kuuluvia moduuleja on tarkoitettu käytettäväksi yhdessä koko ohjelmistorakenteen muiden moduulien tai muiden laitteiden kanssa, yhdistelmän tulee kokonaisuudessaan toimia turvallisesti, eikä se saa haitata terveydenhuollon laitteita koskevan direktiivin soveltamisalaan kuuluvien moduulien toimintaa. [17.]

#### 4 Älykello terveydenhuollon laitteena

Älykello on ranteessa pidettävä puettava tietokone/laite, englanniksi wearable computer tai lyhyesti wearable. Puettavalla tietokoneella tarkoitetaan elektronista laitetta tai tietokonetta, joka on sisällytetty vaatteisiin tai tarvikkeisiin ja sitä voi mukavasti käyttää keholla. Puettavat laitteet voivat suorittaa monia samoja tehtäviä kuin älypuhelimet tai kannettavat tietokoneet, mutta ne ovat koko ajan käyttäjän saatavilla ja käyttövalmiina. Tämän takia puettavat laitteet tarjoavat mahdollisuuden hyödyntää monia erityyppisiä sensoreita fysiologisen tilan valvontaan, mikä ei ole mahdollista kannettavilla laitteilla. [18.]

Puettaville tietokoneille/laitteille ei ole tarkkaa lakisäätelistä määrittystä, mutta EU:n komission on julkaisut artikkelin "Smart Wearables Reflection and Orientation Paper", jossa lakimiehet ja asiantuntijat ovat antaneet virallisen ehdotuksen puettavan laitteen määrittämiselle. Virallisesti tulkitaan, että puettava laite on ylläpidettävä laskennallinen laite tai sensori, joka voi havaita henkilön ja/tai hänen ympäristönsä. Puettava laite voi kommunikoida muiden laitteiden kanssa joko suoraan käyttämällä puettavan laitteen omaa tuettua verkkoa tai jonkin muun laitteen kautta. Kaikki käyttäjältä kerätty tieto tai hänen ympäristöstä käsitellään joko lokaalisti tai ulkoisella palvelimella, ja tulokset esitetään lopulta käyttäjälle. [19.]

Kysymykseen "onko älykello terveydenhuollon laite" vastaaminen MD-direktiivin pohjalta on helppoa, mutta todellisuudessa tämä saattaa olla hankalampaa. MD-direktiivi tulkitsee kaikki laitteet terveydenhuollon laitteeksi, jos ne on tarkoitettu lääkinälliseen

käyttöön. Ongelmana tässä on, että MD-direktiivi ei selkeästi määrittele, mitä lääkinnällinen käyttö tarkoittaa. [20.]

MD-direktiivin vaatimukset auttavat selvittämään, onko laite tarkoitettu lääkinnälliseen käyttöön. Tähän vaikuttaa kaksi tekijää: laitteen käyttötarkoitus ja onko se tarkoitettu käytettäväksi yhdessä muun laitteen kanssa vai kokonaan erikseen. Laitteen käyttötarkoitus määräytyy laitteen valmistajan ilmoittamien laitemerkintöjen, käyttöohjeen ja/tai markkinointiaineiston antamien tietojen perusteella, eli se miten valmistaja kuvaillee ja markkinoi laitetta, vaikuttaa, onko laite terveydenhuollon laite vai ei. [20.]

Toinen osa vaatimusta, onko laite tarkoitettu käytettäväksi yksin vai yhdessä muun laitteen kanssa, on myös tärkeä. Tämä määrittelee, onko laite terveydenhuollon laite vai lisälaitte. Älykello voidaan tulkita lisälaitteeksi, sillä se saattaa tarvita toisen laitteen toimiakseen tai laajentaa sen toimintaa. Älykellossa oleva ohjelmisto määrittää, onko laite lisälaitte vai kokonaan oma terveydenhuollon laite. Lisälaitteet eivät ole terveydenhuollon laitteita, mutta kuuluvat MD-direktiivin soveltamisalaan. Lisälaitteen etuna on, että se voidaan tuoteluokitella erikseen terveydenhuollon laitteesta. [20.]

## 5 Vaatimustenmukainen Android Wear -sovellus

Android Wear on puettaville tietokoneille ja älykelloille Googlen tekemä Android-käyttöjärjestelmä. Käyttöjärjestelmä julkaistiin 18. maaliskuuta 2014 ja se vaatii käyttäjää parittamaan kellon puhelimen kanssa. Kello toimii Android- ja iOS-puhelimien kanssa, Android-puhelimeissa pitää olla versio 4.3 tai uudempi ja iOS-puhelimeissa versio 8.2 tai uudempi.

Android Wear -sovelluksia voi käyttää kokonaan itsenäisesti, tai ne voivat olla riippuvaisia puhelimesta. Android Wear tukee kolmea eri tapausta, sovellus on kokonaan itsenäinen, osittain itsenäinen ja kokonaan riippuvainen. Jos sovellus on kokonaan tai osittain riippuvainen, kutsutaan sovellusta "standalone"-sovellukseksi. [21.]

Työssä toteutettiin Android Wear -sovellus, joka koostuu minisovelluksesta, kellotaulusta ja komplikaatioista. Sovelluksen tarkoitus on auttaa potilasta seuraamaan hänen diabetesta. Sovelluksella käyttäjä pystyy syöttämään verensokerin, hiilihydraattin ja insuliiniarvon, näkemään viimeisimmät arvot ja vaihtamaan sovelluksen asetuksia. Minisovellus suunniteltiin "standalone"-sovellukseksi, sillä vain itsenäinen sovellus voi-



daan tulkita terveydenhuollon laitteeksi. Jos taas sovellus vaatisi kumppanisovelluksen, se tulkitaan terveydenhuollon laitteen lisälaitteeksi.

## 5.1 Kellotaulu

Käyttäjän yksityissuojan kannalta on tärkeää huomioida, että kello on paljon intiimimpi ja helpommin esillä kuin puhelin, koska kello on yleensä käyttäjällä ranteessa kiinni. Yksityisyyden suojaamiseksi kannattaa piilottaa henkilökohtaiset tiedot, kun kello ei ole käytössä. Tähän Android Wear tarjoaa "Always on" -tilaa. Kello menee "Always on" -tilaa, kun se ei ole pariin sekuntiin käytössä ja tämän jälkeen se muuttaa kaikki värit mustavalkoiseksi. Kellotaulu havaitsee, kun kello menee "always on" -tilaan, ja tällöin se voi estää henkilökohtaisten tietojen piirtämisen kellotauluun.

### Kellotaulu - Aktiivinen      Kellotaulu - Always On



Kuva 4. Yksinkertainen digitaalinen kellotaulu, jossa on kaksi komplikaatioita.

## 5.2 Komplikaatio

Tiedon esittämiseen Android Wear -kellossa Google tarjoaa kehittäjille mahdollisuuden luoda omia kellotauluja ja komplikaatioita. Komplikaatio on jokin muu ominaisuus ajan lisäksi, joka on lisättyä kellotauluun, esimerkkinä kellon akkuindikaattori tai sähköpostien määrän näyttäminen.

Jos kehittäjä haluaa lisätä lisätoiminnallisuuksia kellotauluun, näitä Google kutsuu komplikaatioiksi. Komplikaatiot ovat kellotauluriippumattomia, ja käyttäjä voi lisätä min-kä tahansa komplikaation mihin tahansa kellotauluun, jos kellotauluun on määritetty kenttä komplikaatiolle ja sen tyyppi on tuettu.

Koodiesimerkissä 1 luodaan kellotauluun 2 komplikaatiokenttää kellon ylä- ja alareunaan.

```
private void initializeComplications() {
    mActiveComplicationDataSparseArray = new SparseArray<>(COMPLICATION_IDS.length);

    ComplicationDrawable topComplicationDrawable =
        new ComplicationDrawable(getApplicationContext());

    ComplicationDrawable bottomComplicationDrawable =
        new ComplicationDrawable(getApplicationContext());

    mComplicationDrawableSparseArray = new SparseArray<>(COMPLICATION_IDS.length);

    mComplicationDrawableSparseArray.put(TOP_COMPLICATION_ID, topComplicationDrawable);
    mComplicationDrawableSparseArray.put(BOTTOM_COMPLICATION_ID, bottomComplicationDrawable);

    setComplicationsActiveAndAmbientColors (mWatchBandHighlightColor);
    setActiveComplications (COMPLICATION_IDS);
}
```

Koodiesimerkki 1. Komplikaatioiden luominen.

Komplikaatiot piirretään kellontauluun onDraw()-metodin sisällä. Sovellus kutsuu koodiesimerkissä 2 esitettyä metodia onDraw()-metodin. Koodiesimerkin metodi piirtää komplikaatiot kellotauluun.

```
private void drawComplications(Canvas canvas, long currentTimeMillis) {
    int complicationId;
    ComplicationDrawable complicationDrawable;

    for(int complicationId1 : COMPLICATION_IDS) {
        complicationId = complicationId1;
        complicationDrawable = mComplicationDrawableSparseArray.get(complicationId);
        complicationDrawable.draw(canvas, currentTimeMillis);
    }
}
```

Koodiesimerkki 2. Komplikaatioiden piirtäminen kellotauluun.

Kun komplikaatio havaitsee, että sen tiedot ovat muuttuneet, se kutsuu koodiesimerkissä 3 olevaa metodia. Tämän avulla kellotaulu voi päivittää itsensä.

```
@Override
public void onComplicationDataUpdate(int complicationId, ComplicationData complicationData) {
    mActiveComplicationDataSparseArray.put(complicationId, complicationData);
    ComplicationDrawable complicationDrawable = mComplicationDrawableSparseArray.get(complicationId);
    complicationDrawable.setComplicationData (complicationData);

    invalidate();
}
```

Koodiesimerkki 3. Kellotaulun komplikaation päivittäminen.

Käyttäjän yksityisyysuojan parantamiseksi ja Google-ohjeistusten noudattamiseksi komplikaatiot päivitetään "always on" -tilaan, kun kello siirtyy inaktiiviseen tilaan. "Always on" -tilassa kellotaulu käyttää vain värinä mustaa ja valkoista. Yli 90 % näytöstä pitää olla mustana. Jos komplikaation tietoja ei halua näyttää "Always on" -tilassa, helppoin tapa on se tehdä muuttamalla komplikaation arvo tekstien väri samaksi kuin tausta. Koodiesimerkissä 4 päivitetään komplikaatiot "always on" -tilaan.

```

@Override
public void onAmbientModeChanged(boolean inAmbientMode) {
    super.onAmbientModeChanged(inAmbientMode);
    Log.d(TAG, "onAmbientModeChanged: " + inAmbientMode);
    mAmbient = inAmbientMode;
    updateWatchPaintStyles();

    ComplicationDrawable complicationDrawable;
    for (int complicationId : COMPLICATION_IDS) {
        complicationDrawable = mComplicationDrawableSparseArray.get(complicationId);
        complicationDrawable.setInAmbientMode(mAmbient);
    }

    updateTimer();
}


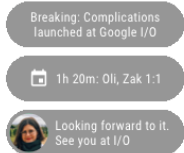


```

Koodiesimerkki 4. Komplikaatioiden asettaminen "Always on" -tilaan.

Komplikaatioiden luomiseen Google on luonut oman API:n, Complication API. API määrittelee, miten komplikaatioin voi luoda ja miten siihen voi sitoa datan. Taulukossa 1 on listattu kaikki eri komplikaatiotyylit ja tietokentät, jotka niihin voi lisätä.

Taulukko 1. Lista Android Wear:n komplikaatiotyypeistä. [27]

Tyyppi	Vaaditut kentät	Lisä kentät	Esimerkki kuva
SHORT_TEXT	Lyhyt teksti	Ikoni "Burn in" -suojausikoni Lyhyt teksti	
ICON	ikoni	"Burn in" -suojausikoni	

RANGED_VALUE	Arvo Min arvo Max Arvo	Ikoni "Burn in" -suojausikoni Lyhyt teksti Lyhyt otsikko	
LONG_TEXT	Pitkä teksti	Pitkä otsikko Ikoni "Burn in" -suojausikoni Pieni kuva	
SMALL_IMAGE	Pieni kuva		
LARGE_IMAGE	Iso kuva		

### 5.3 Minisovellus

#### 5.3.1 Käyttöliittymä

Googlen "Material Guideline" -dokumentissa on määritelty, miltä Googlen sisäänkirjautumisen tulee näyttää. Kuvassa 5 on esitetty sisäänkirjautumissivu. Sivun linjassa Googlen antamien suositusten kanssa. Koodiesimerkkissä 5 on esitetty, miten sisäänkirjautumissivu on toteutettu XML-tiedostossa.

## Sisäänkirjautuminen



Kuva 5. Minisovelluksen sisäänkirjautumissivu.

```
<androidx.wear.widget.BoxInsetLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_height="match_parent"
    android:layout_width="match_parent">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="5dp"
        app:boxedEdges="all"
        tools:ignore="MissingPrefix">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/textView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/wear_signin_title"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent" />

            <TextView
                android:id="@+id/textView2"
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_marginTop="8dp"
                android:text="@string/wear_signin_description"
                app:layout_constraintBottom_toTopOf="@+id/sign_in_button"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/textView" />

            <com.aleksi_anttila.myhealth.ui.WearGoogleSignInButton
                android:id="@+id/sign_in_button"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent" />

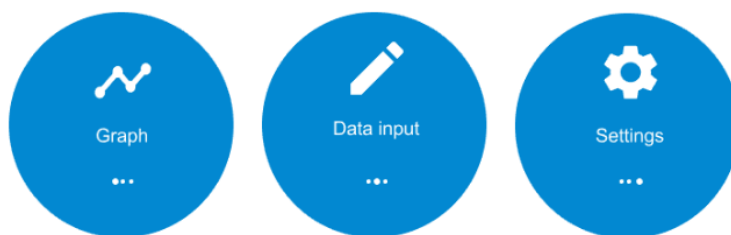
        </androidx.constraintlayout.widget.ConstraintLayout>
    </FrameLayout>
</androidx.wear.widget.BoxInsetLayout>
```

Koodiesimerkki 5. Sisäänkirjautumissivun XML-tiedosto.

Sisäänkirjautumislogiikka sovellustasolla on selitetty luvussa 5.3.3.

## Navigation Drawer

Navigointi sivujen välillä



Kuva 6. Navigointi eri sivujen välillä minisovelluksessa.

Navigointi eri sivujen välillä sovelluksen sisällä toteutettiin käyttäen NavigationDraweria. Kuvassa 6 on esitetty, miltä navigaatio näyttää. Käyttäjä voi avata navigaatiopalkin pyyhkäisemällä ylöspäin näytöllä ja valita eri sivun liikkumalla oikealle tai vasemmalle. Koodiesimerkissä 6 on toteutettu navigaatiopalkki XML-tiedostossa.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.wear.widget.drawer.WearableDrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/black"
    tools:context=".MainActivity">

    <androidx.core.widget.NestedScrollView
        android:id="@+id/content"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fillViewport="true">

        <FrameLayout
            android:id="@+id/content_frame"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />
    </androidx.core.widget.NestedScrollView>

    <androidx.wear.widget.drawer.WearableNavigationView
        android:id="@+id/top_navigation_drawer"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/grey"
        app:navigationStyle="multiPage" />

</androidx.wear.widget.drawer.WearableDrawerLayout>
```

Koodiesimerkki 6. Navigaatio Drawerin XML-tiedosto.

Koodiesimerkissä 7 selitetään, miten java-koodissa luodaan navigaatiopalkki ja miten sivun vaihto tehdään. Sovelluksen sivujen luontiin käytettiin fragmentteja ja fragment manager vaihtaa päällimmäisen fragmentin uuteen, kun käyttäjä vaihtaa sivua.

Kun käyttäjä vaihtaa sivua, järjestelmä kutsuu sovelluksen `onItemSelected()`-metodia, joka vaihtaa sivun tekemällä transaktion käyttämällä `FragmentManager`-luokkaa.

```

@Inject
FragmentManager fragmentManager;

@BindView(R.id.top_navigation_drawer)
WearableNavigationView wearableNavigationDrawer;

private NavigationAdapter adapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    AmbientModeSupport.attach(activity: this);

    ((MainApplication) getApplication()).getApplicationComponent()
        .plus(new ActivityModule(this))
        .inject(activity: this);

    adapter = createAdapter();
    wearableNavigationDrawer.setAdapter(adapter);
    wearableNavigationDrawer.getController().peekDrawer();
    wearableNavigationDrawer.addOnItemSelectedListener(this);

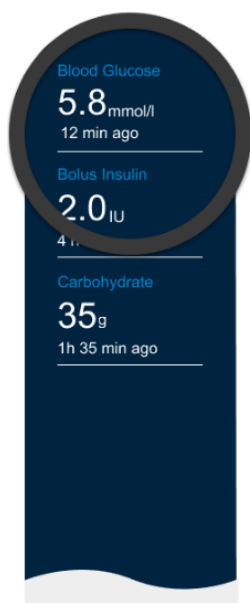
    onItemSelected(position: 0);
}

@Override
public void onItemSelected(int position) {
    NavigationItem item = adapter.getItem(position);
    fragmentManager.beginTransaction().replace(R.id.content_frame, item.getFragment()).commit();
}

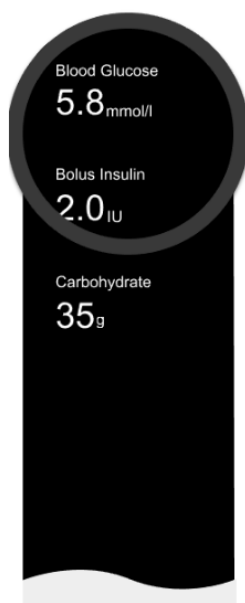
```

Koodiesimerkki 7. Navigaatiopalkin luonti ja sivun vaihto kooditasolla.

Graph - Aktiivinen



Graph - Always On



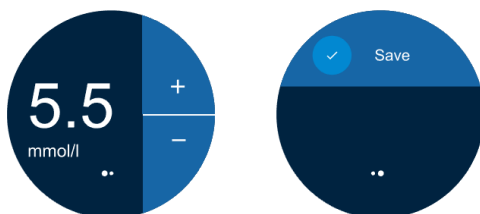
Kuva 7. Minisovelluksen "graph"-sivu, aktiivinen ja "always on" versio.

"Graph"-sivu toteutettiin käyttäen vertikaalista WearableRecyclerView'tä (ks. koodiesimerkki 8). Lista sisältää X määrän viimeisimpiä arvoja, joka määräytyy eri datatyyppien mukaan. Yksi listaelementti näyttää datatyyppin nimen, datatyyppin yksikön, viimeisimmän käyttäjän syöttämän arvon ja kuinka kauan viimeisimmästä arvosta. Kuvassa 7 on esitetty lista viimeisimmistä arvoista, kun käyttäjä on syöttänyt verensokeri-, hiilihydraatti- ja insuliiniarvot.

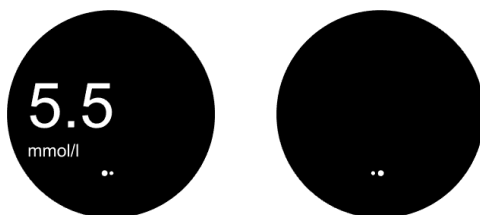
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.wearable.widget.WearableRecyclerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/graph_root"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scrollbars="vertical" />
```

Koodiesimerkki 8. "Graph"-sivun XML-tiedosto.

#### Tiedon syöttö - Aktiivinen



#### Tiedon syöttö - Always-On



Kuva 8. Minisovelluksen tiedonsyöttösivu, aktiivinen ja "always on" versio.

Datan syöttösivu toteutettiin käyttäen Viewpageriä, ViewPager näyttää data input -sivun jokaiselle datatyyppille ja lopussa on arvojen tallennussivu. Yhden datan syöttöelementin ulkonäkö ja tallennussivu on esitetty kuvassa 8.



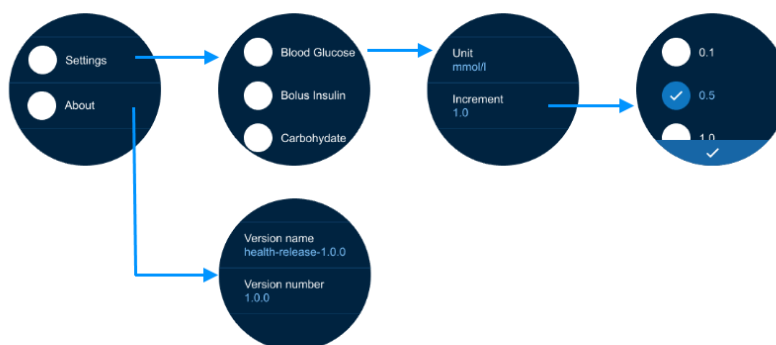
```

<?xml version="1.0" encoding="utf-8"?>
<androidx.viewpager.widget.ViewPager
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/data_input_root"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

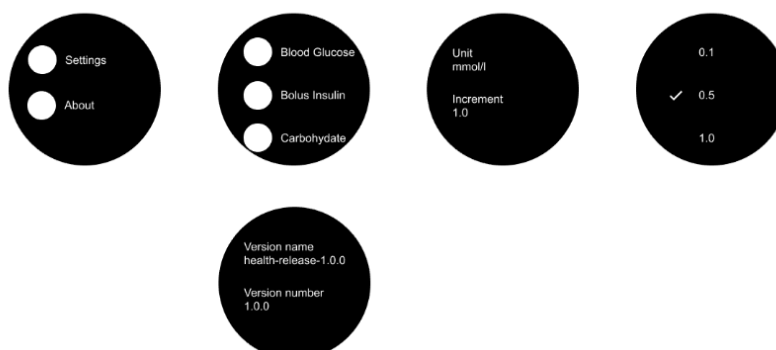
```

### Koodiesimerkki 9. Data syöttö -sivun XML-tiedosto

#### Asetukset - Aktiivinen



#### Asetukset - Always On



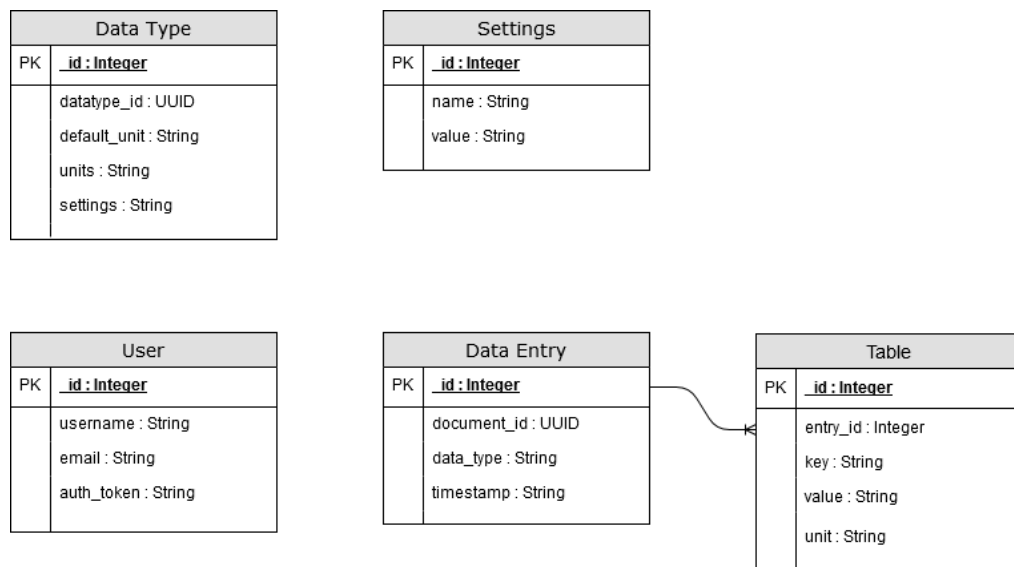
Kuva 9. Minisovelluksen asetussivu, aktiivinen ja "always on" versio.

Settings-sivu ja sen alisivut toteutettiin myös käyttämällä `WearableRecyclerView`'tä. Kun käyttäjä valitsee alisivun, sovellus lisää uuden fragmentin pinon ja näyttää seuraavan sivun. Takaisin navigoidessa pinosta poistetaan päällimmäinen, ja edellinen sivu näytetään. Kuvassa 9 on esitetty settings- ja about-sivu sekä navigointijärjestys settings-alasivuilla.

### 5.3.2 Tietokanta

Jotta käyttäjän syöttämät verensokeri-, hiilihydraatti- ja insuliiniarvot, sekä asetukset pysyvät tallessa ja ovat käytettävissä jokaisella käyttökerralla, tiedot tallennetaan tietokantaan. Tietokannasta voidaan lukea kaikki syötetyt arvot ja asetukset, kun sovellus tarvitsee niitä. Paikallinen tietokanta mahdollistaa offline-käytön, laitteen ei tarvitse hakea tietoja palvelimelta, koska tarvittavat tiedot ovat tallessa tietokannassa. Syötetyt arvot ovat tallessa tietokannassa ja ne voidaan lähetetään palvelimelle, kun sovellus havaitsee, että laite on yhteydessä verkkoon.

Tietokannan luomiseen ja hallintaan käytettiin Googlen uutta Room-kirjastoa, joka on osa uusia arkkitehtuurikomponentteja. Kirjasto hoitaa olio- ja relaatiomallin kommunikaation SQLite-tietokannan ja sovelluksen välillä. Kirjaston avulla voi luoda vakaamman ja tehokkaamman SQL-tietokannan sovellukseen. Koodiesimerkki 10 on luotu Room-kirjastolla DataEntry-luokka. Kirjasto osaa luokan annotaatioiden perusteella luoda tietokantataulun, joka vastaa kuvan 10 "Data Entry" -taulua. "ColumnInfo"-annotaatio määrittelee taulun kolumnin ja "Entry"-annotaatio sisällä määritelty "tableName"-kenttä taulun nimen. Taulun automaattisesti kasvavan indeksikentän voi määrittää käyttämällä "PrimaryKey"-annotaatiota, ja asettamalla "autoGenerate"-arvon todeksi.



Kuva 10. Sovelluksen tietokantamalli.

```

@Entity(tableName = TABLE_NAME,
    indices = {@Index(
        value = {DataEntry.Column.DOCUMENT_ID},
        unique = true)})
public class DataEntry {
    static final String TABLE_NAME = "data_entry";

    static class Column {
        static final String DOCUMENT_ID = "document_id";
        static final String ENTRY_TYPE = "entry_type";
        static final String TIMESTAMP = "timestamp";
    }

    @PrimaryKey(autoGenerate = true)
    private int id;

    @ColumnInfo(name = Column.DOCUMENT_ID)
    private String documentId = UUID.randomUUID().toString();

    @ColumnInfo(name = Column.ENTRY_TYPE)
    private String entryType;

    @ColumnInfo(name = Column.TIMESTAMP)
    private OffsetDateTime timestamp;

    public DataEntry() {}

    public DataEntry(DataEntry dataEntry) {...}

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getDocumentId() { return documentId; }

    public void setDocumentId(String documentId) { this.documentId = documentId; }

    public String getEntryType() { return entryType; }

    public void setEntryType(String entryType) { this.entryType = entryType; }

    public OffsetDateTime getTimestamp() { return timestamp; }

    public void setTimestamp(OffsetDateTime timestamp) { this.timestamp = timestamp; }
}

```

Koodiesimerkki 10. Room-kirjastolla "Data Entry" -tietokantataulun luonti.

Jokaiselle kuvassa 10 määritellylle taululle luotiin luokka käyttämällä Room-kirjaston annotaatioita. DataEntry taulun one-to-many-vierasavainyhteys DataEntryAttributeen luodaan Room-kirjastolla paketoimalla kummatkin luokat uuden POJO-luokan sisälle (ks. koodiesimerkki 11).

```

public class DataEntryWithAttributes {

    @Embedded
    public DataEntry entry;

    @Relation(parentColumn = "id", entityColumn = ENTRY_ID)
    public List<DataEntryAttribute> attributes;

    public DataEntryAttribute getAttribute(String attribute) {
        return Iterables.find(attributes,
            input -> input != null && TextUtils.equals(input.getAttribute(), attribute),
            defaultValue: null
        );
    }
}

```

Koodiesimerkki 11. One-to-many-vierasavainyhteyden luominen Room-kirjastolla.

```

@Dao
public interface DataEntryDAO {

    @Insert
    void insert(DataEntry dataEntry);

    @Update
    void update(DataEntry dataEntry);

    @Query("SELECT * FROM data_entry WHERE document_id = :documentId LIMIT 1")
    DataEntry queryForId(String documentId);
}

```

Koodiesimerkki 12. SQL-tietokantakyselyjen tekeminen Room-kirjastolla

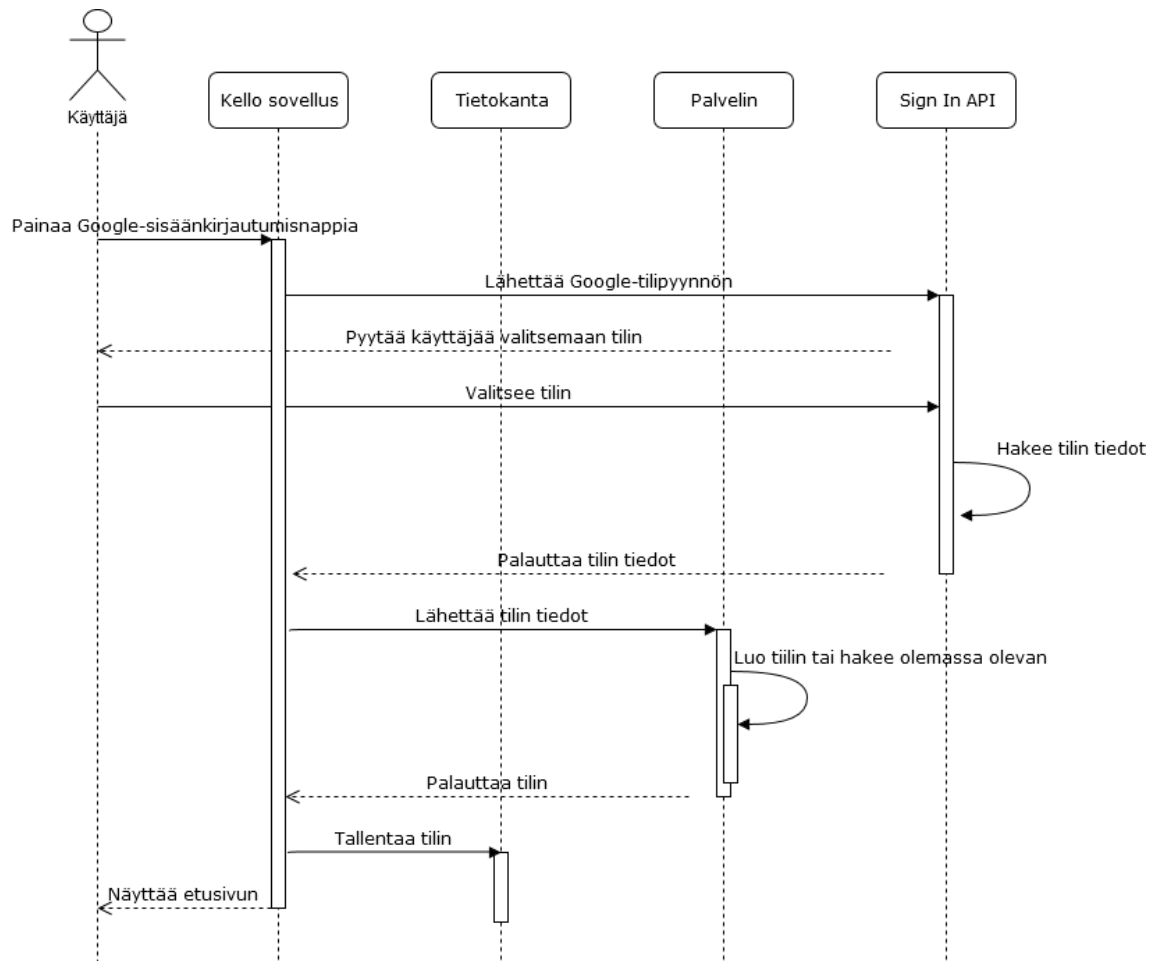
Luokka, joka hakee tai päivittää tietokantaa, suorittaa operaatiot käyttämällä rajapinnan määrittämiä metodeja. Koodiesimerkissä 12 on määritelty arvon lisääminen, päivittäminen ja yksinkertainen tietokantakysely.

### 5.3.3 Käyttäjän tunnistaminen

Asiakastietojen sähköisen käsittelyn laki vaatii terveydenhuollon laitteilta, että käyttäjä on tunnistettu luotettavasti. Käyttäjän tunnistamisen voi tehdä monella tapaa Android Wear -sovelluksessa. Tuetut tavat helpoimmasta vaikeimpaan toteuttaa ovat:

1. kirjautuminen käyttäen Google-tiliä
2. autentikaatiotunnisteiden siirtäminen käyttäen "Data layer" API:a
3. OAuth 2.0 -autentikaation suorittaminen puhelimessa
4. autentikaatio kolmannesapuolen palvelussa ja sieltä saadun koodin syöttäminen kelloon.

Sovelluksessa sisäänkirjautuminen toteutettiin käyttämällä vaihtoehtoa yksi.



Kuva 11. Sekvenssikaavio sisäänkirjautumisesta.

Kuvassa 11 näkee Google-sisäänkirjautumisvaiheet. Kirjautuminen alkaa, kun käyttäjä painaa Google-nappia. Koodissa kutsutaan koodiesimerkissä 13 määriteltyä metodia.

```

@OnClick(R.id.sign_in_button)
private void signIn() {
    if (googleApiClient == null || !googleApiClient.isConnected()) {
        return;
    }
    Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(googleApiClient);
    startActivityForResult(signInIntent, REQUEST_CODE_SIGN_IN);
}
  
```

Koodiesimerkki 13. Google-tilin valitsemisen aloittaminen koodissa.

Tämän jälkeen käyttöjärjestelmä antaa käyttäjän valita haluamansa tilin, joka palautetaan sovellukseen kutsumalla `onActivityResult()`-metodia. Koodiesimerkissä 14 käsitellään saadun tilin tiedot ja aloitetaan tilin tietojen siirto palvelimeen. Tilinsiirto on hidas prosessi, joten tilinsiirtoa ei voi tehdä pääsäikeessä. Tämän takia tilinsiirto tehdään taustaprosessissa, koodiesimerkki 15 `JobService`-luokka suorittaa tiedonsiirron taustäsäikeessä.

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_SIGN_IN) {
        GoogleSignInResult signInResult = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        if (signInResult.isSuccess()) {
            GoogleSignInAccount account = signInResult.getSignInAccount();
            if (account != null) {
                PersistableBundle extras = new PersistableBundle();
                extras.putString(ACCOUNT_EMAIL, account.getEmail());
                extras.putString(ACCOUNT_AUTH_CODE, account.getServerAuthCode());

                JobInfo jobInfo = new JobInfo.Builder(JOB_ID, new ComponentName(pkg, this, SignInJob.class))
                    .setRequiredNetworkType(JobInfo.NETWORK_TYPE_UNMETERED)
                    .setExtras(extras)
                    .build();

                jobScheduler.schedule(jobInfo);
            }
        }
    }
}
```

Koodiesimerkki 14. Google-tilin käsittely ja tiedonsiirron aloitus.

```

public class SignInJob extends JobService {

    public static final int JOB_ID = 6008;

    public static final String ACCOUNT_EMAIL = "EXTRA_EMAIL";
    public static final String ACCOUNT_AUTH_CODE = "EXTRA_AUTH_TOKEN";

    @Inject
    AccountSyncStep syncStep;

    private SignInTask task;

    @Override
    public void onCreate() {
        super.onCreate();
        ((MainApplication) getApplication()).getApplicationComponent().inject( service: this);
    }

    @Override
    public boolean onStartJob(JobParameters jobParameters) {
        PersistableBundle bundle = jobParameters.getExtras();
        if(task != null) {
            task.cancel( mayInterruptIfRunning: true);
        }

        task = new SignInTask(syncStep);
        task.execute(bundle.getString(ACCOUNT_EMAIL), bundle.getString(ACCOUNT_AUTH_CODE));

        return false;
    }

    @Override
    public boolean onStopJob(JobParameters jobParameters) {
        return false;
    }

    private static class SignInTask extends AsyncTask<String, Void, Void> {
        private final AccountSyncStep syncStep;
        private SignInTask(AccountSyncStep syncStep) {...}

        @Override
        protected Void doInBackground(final String... params) {
            String email = params[0];
            String authToken = params[1];
            syncStep.signIn(email, authToken);

            return null;
        }
    }
}

```

Koodiesimerkki 15. JobService, joka hallitsee sisäänkirjautumisen taustasäikeessä.

Palvelimen kanssa kommunikointi tapahtuu käyttäen https-protokollaa. Retrofit-kirjasto mahdollistaa http-kutsujen helpon tekemisen. Koodiesimerkissä 16 suoritetaan sisäänkirjautuminen palvelimen kanssa käyttäen REST-rajapintoja, jotka on määritelty koodiesimerkissä 17. Palvelimelle lähetetään sähköposti ja autentikointitietue ja palvelin palauttaa onnistuneesta kirjautumisesta tiedon sovellukseen, joka tämän jälkeen luo tietokantaan käyttäjän profiilin.

```

public class AccountSyncStep {
    private static final String TAG = "AccountSyncStep";

    @Inject
    @Named("api")
    Config config;

    @Inject
    AccountRepository repository;

    @Inject
    ServiceGenerator serviceGenerator;

    private String clientId;
    private String clientSecret;
    private String scope;

    @Inject
    public void init() {
        clientId = config.getString( path: "clientId");
        clientSecret = config.getString( path: "clientSecret");
        scope = config.getString( path: "scope");
        repository.createAndroidAccount();
    }

    public void signIn(final String username, final String authToken) {
        OAuthService oAuthService = serviceGenerator.createService(OAuthService.class, authProvider: null);
        oAuthService.getAccessToken(clientId, clientSecret, scope, username, authToken)
            .enqueue(new Callback<AccessToken>() {
                @Override
                public void onResponse(Call<AccessToken> call, Response<AccessToken> response) {
                    if (response.isSuccessful()) {
                        AccessToken accessToken = response.body();
                        repository.setUsername(username);
                        repository.setAccountId(accessToken.getAccountId());
                        repository.createAndroidAccount();
                    } else {
                        ErrorResponse error = serviceGenerator.errorResponseAs(ErrorResponse.class, response);
                        if (error != null) {
                            Log.v(TAG, msg: "getAuthToken OAuth request failed: " + error.getErrorDescription());
                        }
                    }
                }

                @Override
                public void onFailure(Call<AccessToken> call, Throwable t) {
                    Log.e(TAG, msg: "getAuthToken OAuth request failed", t);
                }
            });
    }
}

```

Koodiesimerkki 16. Sisäänkirjautumisen suorittaminen palvelimen kanssa.

```

public interface OAuthService {

    @POST("v1/oauth2/access_token")
    @FormUrlEncoded
    Call<AccessToken> getAccessToken(@Field("client_id") String clientId,
                                    @Field("client_secret") String clientSecret,
                                    @Field("scope") String scope,
                                    @Field("username") String username,
                                    @Field("password") String password);

    @POST("v1/oauth2/access_token")
    @FormUrlEncoded
    Call<AccessToken> refreshAccessToken(@Field("client_id") String clientId,
                                        @Field("client_secret") String clientSecret,
                                        @Field("scope") String scope,
                                        @Field("refresh_token") String refreshToken);
}

```

Koodiesimerkki 17. Palvelimen määrittämät REST-rajapinnat

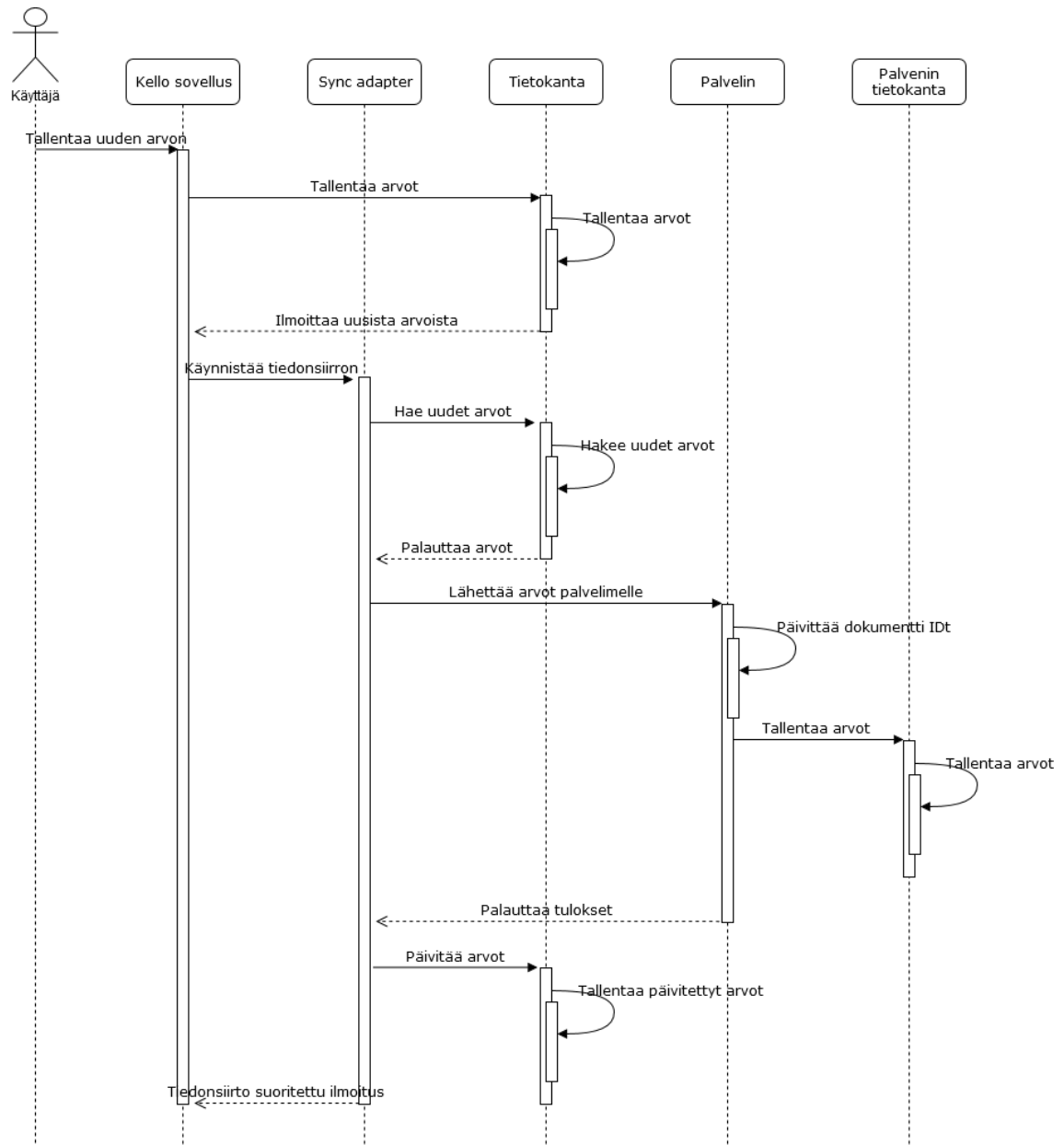
### 5.3.4 Tiedonsiirto

Asiakastietojen sähköisen käsittelyn laki (luku 2.2) vaatii, että käyttäjällä on aina pääsy tarvittaviin tietoihin ja tiedon pitää olla eheää. Käyttäjän tiedot pitää siis aina olla saatavilla ja ajan tasalla jokaisessa laitteessa. Android Wear tukee kahta eri tapaa tiedonsiir-



toon. Ensimmäinen, suositeltu tapa, on siirtää dataa käyttäen kellon tai puhelimen Wi-Fi:ä/verkkoa, ja toinen tapa on käyttää "Data layer"-API:a. Tässä työssä toteutetaan tiedonsiirto käyttäen WiFi/verkkoyhteyttä. Sisäänkirjautumisluvussa (luku 5.3.3) käytettiin tätä tapaa tilin lähettämiseen palvelimelle.

Kello yrittää mahdollisimman paljon käyttää puhelimen WiFi:ä tai dataverkkoa, kun se suorittaa verkko-operaatioita säästääkseen virtaa. Tällöin kello siirtää kaiken tiedon Bluetoothin välityksellä kellosta puhelimeen ja sen jälkeen suorittaa verkko-operaation käyttäen puhelimen verkkoa. Kun kello käyttää Bluetooth-yhteyttä, on hyvä muistaa, että Bluetooth-verkon tiedonsiirtonopeus on vain 4 kB/s [29]. Tämä rajoittaa verkon käytön vain perustiedonsiirtoon. Isojen kuvien ja tiedostojen siirtäminen palvelimista kelloon on tämän takia hidasta.



Kuva 12. Sekvenssikaavio tiedonsiirrosta.

Tiedonsiirron prosessi on kuvattuna kuvassa 12. Kun sovellus havaitsee, että tietokannassa on muuttunut jotain, se käynnistää tiedonsiirron, joka tapahtuu taustalla. Tämän jälkeen sovellus hakee uudet tiedot ja lähettää ne REST-rajapinnan yli palvelimelle, joka tallentaa ne omaan tietokantaan ja tekee niihin tarvittavat muutokset. Kun palvelin on valmis, se lähettää vastauksen sovellukseen.

Tiedonsiirtoon käytettiin Retrofit-kirjastoa. Kirjaston avulla voi tehdä palvelinkutsuja käyttäen http-protokollaa. Käyttämällä OkHttp3-kirjastoa Retrofitin kanssa sovelluksella pystyy tekemään HTTPS-pyyntöjä palvelimelle. OkHttp3-kirjastolla voi lukea tiedostos-

ta SSL-sertifikaatin ja ottaa se käyttöön. Koodiesimerkissä 18 on luettu SSL-sertifikaatti tiedostosta.

```
private void loadCerts(InputStream keystoreStream, String keystorePassword) {
    try {
        KeyStore keyStore = KeyStore.getInstance("BKS");
        keyStore.load(keystoreStream, keystorePassword.toCharArray());

        TrustManagerFactory tmf = TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init(keyStore);

        TrustManager[] trustManagers = tmf.getTrustManagers();
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(km: null, tmf.getTrustManagers(), random: null);

        trustManager = (X509TrustManager) trustManagers[0];
        sslSocketFactory = sslContext.getSocketFactory();
    } catch (KeyStoreException | CertificateException | NoSuchAlgorithmException | IOException | KeyManagementException e) {
        Log.e(TAG, msg: "Failed to load keystore", e);
        throw new RuntimeException("Failed to load keystore", e);
    }
}
```

Koodiesimerkki 18. SSL-sertifikaatin lukeminen tiedostosta.

Kun SSL-sertifikaatti on luotu, Retrofit client voi käyttää aina sitä tekemään pyyntöjä palvelimelle. Koodiesimerkissä 19 luodaan Retrofit-client käyttämällä SSL-sertifikaattia. Koodiesimerkin alussa OkHttpClientiin lisätään tuki käyttäjän tunnistamiseksi käyttämällä käyttöoikeustietuetta. Tämän jälkeen OkHttpClientiin lisätään SSL-sertifikaatti ja tiedonsiirron ajankatkaisut, ajankatkaisujen suuruudet on luettu asetus-tiedostosta. Lopuksi OkHttpClient lisätään Retrofit-luokkaan ja luokalle määritellään json-luokkakonvertterit. Kun tämä on tehty, voi Retrofit-luokkaa käyttämällä tehdä REST-rajapintapyyntöjä palvelimeen.

```

public <S> S createService(@NonNull Class<S> serviceClass,
                        @Nullable final AuthorizationProvider authProvider,
                        @NonNull Interceptor... extraInterceptors) {
    OkHttpClient.Builder clientBuilder = new OkHttpClient.Builder()
        .addInterceptor(chain -> {
            String authHeader = null;
            if(authProvider != null) {
                authHeader = authProvider.getAccessToken();
            }

            Request.Builder requestBuilder = chain.request().newBuilder()
                .header( name: "Accept", value: "application/json")
                .header( name: "Authorization", authHeader == null ? "" : authHeader);
            return chain.proceed(requestBuilder.build());
        })
        .authenticator((route, response) -> {
            Log.d(TAG, msg: "authenticate()");

            if(authProvider == null || cloudAuthenticator == null) {
                return null;
            }

            String newAccessToken;
            try {
                newAccessToken = reuseConcurrentRefreshResult.execute(
                    () -> cloudAuthenticator.refreshToken(authProvider.getRefreshToken())
                );
            } catch (IOException e) {
                throw e;
            } catch (Exception e) {
                Log.v(TAG, msg: "Unexpected exception thrown while executing refreshToken", e);
                throw new RuntimeException("Unexpected exception thrown while executing refreshToken", e);
            }

            if(newAccessToken == null) {
                return null;
            }

            return response.request().newBuilder()
                .header( name: "Authorization", newAccessToken)
                .build();
        });

    for (Interceptor interceptor : extraInterceptors) {
        clientBuilder.addInterceptor(interceptor);
    }

    if(sslSocketFactory != null) {
        clientBuilder.sslSocketFactory(sslSocketFactory, trustManager);
    }

    if (apiConfig == null) {
        throw new RuntimeException("Configuration not defined before instantiating a service");
    }

    long connectTimeout = apiConfig.getDuration( path: "timeout.connect", TimeUnit.MILLISECONDS);
    clientBuilder.connectTimeout(connectTimeout, TimeUnit.MILLISECONDS);

    long readTimeout = apiConfig.getDuration( path: "timeout.read", TimeUnit.MILLISECONDS);
    clientBuilder.readTimeout(readTimeout, TimeUnit.MILLISECONDS);

    long writeTimeout = apiConfig.getDuration( path: "timeout.write", TimeUnit.MILLISECONDS);
    clientBuilder.writeTimeout(writeTimeout, TimeUnit.MILLISECONDS);

    OkHttpClient client = clientBuilder.build();

    Converter.Factory objectConverterFactory = serviceClass.isAnnotationPresent(StrictJsonDeserialization.class)
        ? strictConverterFactory() : looseConverterFactory();
    Retrofit.Builder builder = new Retrofit.Builder()
        .baseUrl(formatApiBaseUrl(apiConfig.getString( path: "url")))
        .addConverterFactory(ScalarsConverterFactory.create())
        .addConverterFactory(objectConverterFactory);

    if(callbackExecutor != null) {
        builder.callbackExecutor(callbackExecutor);
    }

    Retrofit retrofit = builder.client(client).build();

    return retrofit.create(serviceClass);
}

```

Koodiesimerkki 19. Retrofit client'in luonti käyttäen SSL-sertifikaattia.

## 6 Yhteenveto

Työn tavoitteena oli tutkia terveydenhuollon laitevaatimuksia sovelluskehityksessä ja toteuttaa yksinkertainen Android Wear -sovellus, jossa yritettiin huomioida terveydenhuollon vaatimukset. Opin työn aikana paljon terveydenhuollon laitevaatimuksista ja nyt ymmärrän perusvaatimukset terveydenhuollon sovelluskehitykselle.

Työn alussa pystyin vain oletamaan, että terveydenhuollon sovelluskehitys on monimutkainen asia, mutta nyt tutustuttuani asiaan tiedän tarkasti, miten hankaia ja laajoja terveydenhuollon vaatimukset ovat. Jos en olisi rajannut projektia hyvin yleiselle tasolle, pelkästään terveydenhuollon laatu järjestelmästä voisi kirjoittaa kokonaisen opinnäytetyön.

Koska rajasin projektissa sovelluksen kehityksen pelkästään Android Wear -sovellukseen, en luonut kokonaisuuteen omaa palvelinpään sovellusta. Jatkokehitystä varten sovelluksella voisi luoda palvelintuen ja mahdollisesti kumppanipuhelinsovelluksen.

## Lähteet

- 1 Zion Market Research. 24.1.2019. Global mHealth Apps Market Will Reach USD 111.1 Billion By 2025: Zion Market Research.. Verkkoaineisto. <<https://www.globenewswire.com/news-release/2019/01/24/1704860/0/en/Global-mHealth-Apps-Market-Will-Reach-USD-111-1-Billion-By-2025-Zion-Market-Research.html/>>. Luettu 4.5.2019.
- 2 Statista. Kuva. <<https://www.statista.com/chart/15035/worldwide-smartwatch-shipments/>>. Luettu 4.5.2019.
- 3 Qualcomm. 10.9.2018. Qualcomm Snapdragon Wear 3100 Platform Supports New Ultra-Low Power System Architecture for Next Generation Smartwatches. Verkkoaineisto. <<https://www.qualcomm.com/news/releases/2018/09/10/qualcomm-snapdragon-wear-3100-platform-supports-new-ultra-low-power-system>>. Luettu 4.5.2019.
- 4 Valvira. 21.6.2017. Terveysteknologian laitteet ja tarvikkeet. Verkkoaineisto. <[https://www.valvira.fi/terveydenhuolto/terveysteknologia/tuotteen\\_markkinoille\\_s\\_aattaminen/terveydenhuollon\\_laitteet\\_ja\\_tarvikkeet/](https://www.valvira.fi/terveydenhuolto/terveysteknologia/tuotteen_markkinoille_s_aattaminen/terveydenhuollon_laitteet_ja_tarvikkeet/)>. Luettu 13.4.2019.
- 5 T. Ståhlberg. 1.14.2015. Terveysteknologian laitteiden lakisäätöiset määräykset kansainvälisillä markkinoilla. Verkkoaineisto. <1. [https://www.businessfinland.fi/globalassets/julkaisut/terveydenhuollon\\_laitteiden\\_akisaateiset\\_maaraykset\\_opas.pdf](https://www.businessfinland.fi/globalassets/julkaisut/terveydenhuollon_laitteiden_akisaateiset_maaraykset_opas.pdf)>. Luettu 13.4.2019.
- 6 Finlex. 24.6.2010. Laki terveydenhuollon laitteista ja tarvikkeista. Verkkoaineisto. <<https://www.finlex.fi/fi/laki/alkup/2010/20100629/>>. Luettu 14.4.2019.
- 7 Valvira. 10.7.2015. Ohjelmistot ja tietojärjestelmät. Verkkoaineisto. <<https://www.valvira.fi/terveydenhuolto/terveysteknologia/tietojarjestelmat/>>. Luettu 19.4.2019.
- 8 Finlex. 2.9.2007. Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä. Verkkoaineisto. <<https://www.finlex.fi/fi/laki/ajantasa/2007/20070159/>>. Luettu 19.4.2019.
- 9 Suomen Standardisoimisliitto SFS ry. Standardi tutuksi. Verkkoaineisto. <[https://www.sfs.fi/julkaisut\\_ja\\_palvelut/standardi\\_tutuksi/](https://www.sfs.fi/julkaisut_ja_palvelut/standardi_tutuksi/)>. Luettu 20.4.2019.
- 10 Suomen Standardisoimisliitto SFS ry. Terveysteknologian tuotteiden laadunhallintajärjestelmän vaatimukset kuvaava standardi on uudistettu. Verkkoaineisto. <[https://sales.sfs.fi/fi/index/tuoteuutiset/terveydenhuollonlaitteidenlaatustandardie\\_sitellaan6.6.2016.html.stx/](https://sales.sfs.fi/fi/index/tuoteuutiset/terveydenhuollonlaitteidenlaatustandardie_sitellaan6.6.2016.html.stx/)>. Luettu 14.4.2019.
- 11 Valvira. 7.9.2010. Määräys 3/2010 - Terveysteknologian laitteilla ja tarvikkeilla tehtävät kliiniset tutkimukset. Verkkoaineisto. <<https://www.valvira.fi/-/maaraykset-3-2010-terveydenhuollon-laitteilla-ja-tarvikkeilla-tehtavat-kliiniset-tutkimukset/>>. Luettu 13.4.2019.
- 12 European Commission. Kuva. <[https://ec.europa.eu/growth/single-market/ce-marking\\_en/](https://ec.europa.eu/growth/single-market/ce-marking_en/)>. Luettu 19.4.2019.

- 13 Valvira. 30.3.2011. Määräys 2/2011 - CE-merkinnän käyttö terveydenhuollon laitteessa ja tarvikkeessa. Verkkoaineisto. <<https://www.valvira.fi/-/maarays-2-2011-ce-merkinnan-kaytto-terveydenhuollon-laitteessa-ja-tarvikkeessa/>>. Luettu 13.4.2019.
- 14 Valvira. 7.9.2010. Määräys 2/2010 - Terveydenhuollon laitteesta ja tarvikkeesta tehtävät laiterekisteri-ilmoitukset. Verkkoaineisto. <<https://www.valvira.fi/-/maarays-2-2010-terveydenhuollon-laitteesta-ja-tarvikkeesta-tehtavat-laiterekisteri-ilmoitukset/>>. Luettu 18.5.2019.
- 15 Valvira. 7.9.2010. Määräys 1/2010 - Terveydenhuollon laitteesta ja tarvikkeesta tehtävät valmistajan vaaratilanneilmoitukset. Verkkoaineisto. <<https://www.valvira.fi/-/maarays-1-2010-terveydenhuollon-laitteesta-ja-tarvikkeesta-tehtavat-valmistajan-vaaratilanneilmoitukset/>>. Luettu 14.4.2019.
- 16 Valvira. 30.3.2011. Määräys 1/2011 - Terveydenhuollon laitteen ja tarvikkeen vaatimustenmukaisuuden arviointi. Verkkoaineisto. <<https://www.valvira.fi/-/maarays-1-2011-terveydenhuollon-laitteen-ja-tarvikkeen-vaatimustenmukaisuuden-arviointi/>>. Luettu 20.4.2019.
- 17 Valvira. 13.3.2012. MEDDEV 2.1/6 - Terveydenhuollossa käytettävien itsenäisten ohjelmistojen määrittely- ja luokitteluoheje lääkinnällisten laitteiden sääntelyn puitteissa. Verkkoaineisto. <[https://www.valvira.fi/documents/14444/37132/sw\\_luokitteluoheje\\_2012-03-13.pdf/](https://www.valvira.fi/documents/14444/37132/sw_luokitteluoheje_2012-03-13.pdf/)>. Luettu 21.4.2019.
- 18 K. j. A. M. Tehrani. Wearable Devices Magazine. 26.3.2014. Wearable Technology and Wearable Devices: Everything You Need to Know. Verkkoaineisto. <<http://www.wearabledevices.com/what-is-a-wearable-device/>>. Luettu 21.4.2019.
- 19 A. Lymberis. European Commission. 2.12.2016. Smart Wearables Reflection and Orientation Paper. Verkkoaineisto. <<https://ec.europa.eu/digital-single-market/en/news/european-commission-seeks-input-reflection-and-orientation-paper-smart-wearables/>>. Luettu 22.4.2019.
- 20 D. M. Charlotte Ryckman. Covington. 22.1.2019. Are Wearables Medical Devices Requiring a CE-Mark in the EU?. Verkkoaineisto. <<https://www.insidemedicaldevices.com/2019/01/are-wearables-medical-devices-requiring-a-ce-mark-in-the-eu/>>. Luettu 21.4.2019.
- 21 Google LLC. Standalone Wear apps. Verkkoaineisto. <<https://developer.android.com/training/wearables/apps/standalone-apps/>>. Luettu 22.4.2019.
- 22 Google LLC. Package and distribute Wear apps. Verkkoaineisto. <<https://developer.android.com/training/wearables/apps/packaging/>>. Luettu 22.4.2019.
- 23 Google LLC. Sign in. Verkkoaineisto. <<https://designguidelines.withgoogle.com/wearos/patterns/sign-in.html/>>. Luettu 28.4.2019.

- 24 Google LLC. Authentication in Wear. Verkkoaineisto.  
<<https://developer.android.com/training/wearables/apps/auth-wear/>>. Luettu 28.4.2019.
- 25 Google LLC. Send and sync data on Wear. Verkkoaineisto.  
<<https://developer.android.com/training/wearables/data-layer/>>. Luettu 4.5.2019.
- 26 Google LLC. Protect your data and applications. Verkkoaineisto.  
<[http://services.google.com/fh/files/misc/security\\_one\\_pager.pdf/](http://services.google.com/fh/files/misc/security_one_pager.pdf/)>. Luettu 4.5.2019.
- 27 Google LLC. Add complications to a watch face. Verkkoaineisto.  
<<https://developer.android.com/training/wearables/watch-faces/adding-complications/>>. Luettu 28.4.2019.
- 28 Google LLC. Network access and syncing on Wear. Verkkoaineisto.  
<<https://developer.android.com/training/wearables/data-layer/network-access/>>. Luettu 28.4.2019.
- 29 Google LLC. Sync data items on Wear. Verkkoaineisto.  
<<https://developer.android.com/training/wearables/data-layer/data-items/>>. Luettu 28.4.2019.



## Liite 1: Valviran tuoteluokkamäärytykset

### 1 Terveysthuollon laitteet (MD)

#### 1.1 Luokan III laitteet

Valmistajan on CE-merkinnän kiinnittämiseksi noudatettava:

- a) MD-direktiivin liitteessä II tarkoitettua EY-vaatimustenmukaisuusvakuutusta (täydellinen laadunvarmistusjärjestelmä) koskevaa menettelyä tai
- b) MD-direktiivin liitteessä III tarkoitettua EY-tyyppitarkastusta koskevaa menettelyä yhdessä:
  - i. MD-direktiivin liitteessä IV tarkoitettua EY-tarkastusta koskevan menettelyn kanssa tai
  - ii. MD-direktiivin liitteessä V tarkoitettua EY-vaatimustenmukaisuusvakuutusta (tuotannon laadunvarmistus) koskevan menettelyn kanssa.

#### 1.2 Luokan IIa laitteet

Valmistajan on CE-merkinnän kiinnittämiseksi noudatettava MD-direktiivin liitteessä VII esitetyn EY-vaatimustenmukaisuusvakuutusta koskevan menettelyn kanssa:

- a) MD-direktiivin liitteessä IV tarkoitettua EY-tarkastusta koskevaa menettelyä,
- b) MD-direktiivin liitteessä V tarkoitettua EY-vaatimustenmukaisuusvakuutusta (tuotannon laadunvarmistus) koskevaa menettelyä tai
- c) MD-direktiivin liitteessä VI tarkoitettua EY-vaatimustenmukaisuusvakuutusta (tuotteiden laadunvarmistus) koskevaa menettelyä. Näiden menettelyjen soveltamisen sijasta valmistaja voi myös noudattaa kohdan alakohdassa a esitettyä menettelyä.

#### 1.3 Luokan II b laitteet

Valmistajan on CE-merkinnän kiinnittämiseksi noudatettava:

- a) MD-direktiivin liitteessä II esitettyä EY-vaatimustenmukaisuusvakuutusta (täydellinen laadunvarmistusjärjestelmä) koskevaa menettelyä, tällöin ei sovelleta MD-direktiivin liitteessä II olevaa 4 kohtaa, tai
- b) MD-direktiivin liitteessä III tarkoitettua EY-tyyppitarkastusta koskevaa menettelyä yhdessä:
  - I. MD-direktiivin liitteessä IV tarkoitettua EY-tarkastusta koskevan menettelyn kanssa,
  - II. MD-direktiivin liitteessä V tarkoitettua EY-vaatimustenmukaisuusvakuutusta (tuotannon laadunvarmistus) koskevan menettelyn kanssa tai iii) MD-direktiivin liitteessä VI tarkoitettua EY-vaatimustenmukaisuusvakuutusta (tuotteiden laadunvarmistus) koskevan menettelyn kanssa.

### 2 In vitro diagnostiset –laitteet (IVDMD)

#### 2.1 IVD-direktiivin liitteen II A-luettelon laitteet

Valmistajan on CE-merkinnän kiinnittämiseksi noudatettava:

- a) VD-direktiivin liitteessä IV esitettyä EY-vaatimustenmukaisuusvakuutukseen liittyvää menettelyä (täydellinen laadunvarmistus) tai
- b) IVD-direktiivin liitteessä V esitettyä EY-tyyppitarkastukseen liittyvää menettelyä yhdessä IVD-direktiivin liitteessä VII esitettyyn EY-vaatimustenmukaisuusvakuutukseen liittyvän menettelyn kanssa (tuotannon laadunvarmistus).

## 2.2 IVD-direktiivin liitteen II B-luettelon laitteet

Valmistajan on CE-merkinnän kiinnittämiseksi noudatettava:

- a) IVD-direktiivin liitteessä IV esitettyä EY-vaatimustenmukaisuusvakuutukseen (täydellinen laadunvarmistusjärjestelmä) liittyvää menettelyä tai
- b) IVD-direktiivin liitteessä V esitettyyn tyyppitarkastukseen liittyvää menettelyä yhdessä:
  - i. IVD-direktiivin liitteessä VI esitettyyn EY-tarkastukseen liittyvän menettelyn kanssa tai
  - ii. IVD-direktiivin liitteessä VII esitettyyn EY-vaatimustenmukaisuusvakuutukseen (tuotannon laadunvarmistus) liittyvän menettelyn kanssa.