

Saara Salminen

# Android-sovellus verensokeriarvojen seurantaan

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinööriytyö

25.4.2018

Tekijä Otsikko	Saara Salminen Android-sovellus verensokeriarvojen seurantaan
Sivumäärä Aika	25 sivua 25.4.2018
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tietotekniikan koulutusohjelma
Ammatillinen pääaine	Ohjelmistotekniikka
Ohjaajat	Lehtori Juha Kämäri
<p>Opinnäytetyön tarkoituksena oli kehittää sovellus omaseurannassa mitattujen verensokeriarvojen tallentamiseen. Lähtökohtana toteutuksessa oli helppokäyttöisyys ja selkeys sekä informatiivisuus. Sovelluksen kohderyhmäksi valikoituivat lapsipotilaat huonon hoitotasapainon ja taudin yleisyyden vuoksi. Sovellus on tarkoitettu käytettäväksi Android-käyttöjärjestelmää käyttävälle puhelimelle.</p> <p>Työssä käsitellään sovelluksen kehitys määrittelystä lopulliseen tuotteeseen ja jatkokehitysehdotuksiin. Sovellus ja sen käyttäminen esitellään lyhyesti.</p> <p>Sovelluksen toteutuskieli on Java ja kehitysympäristönä toimi Android Studio ja sen tarjoama emulaattori ja muut työkalut. Toteutuksessa hyödynnettiin SQLite-tietokantaa. Määritelmää päivitettiin työn aikana.</p> <p>Opinnäytetyö onnistui kohtalaisesti. Suoranaista kirjallista aikataulua ei ollut, joten ajankäyttö ei ollut erityisen organisoitua. Lopputuloksena oli kuitenkin toimiva sovellus.</p>	
Avainsanat	diabetes, verensokeri, Android

Author Title	Saara Salminen An Android application for monitoring blood sugar levels
Number of Pages Date	25 pages 25 April 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructors	Juha Kämäri, Senior Lecturer
<p>The purpose of this thesis was to develop an application for storing blood sugar level values that a patient measures by themselves. The premises for the implementation were ease of use, clarity and informativity. The main target audience for the application are children because of their poor therapeutic equilibrium and because of the prevalence of the disease in Finland. The application is meant to be used on phones using Android operating system.</p> <p>This thesis discusses the development of the application from its specification to the product and further development. The application itself is presented and its use is discussed in brief.</p> <p>The application was developed using Java programming language. The development environment was Android Studio and the emulator and tools it provides. The implementation utilizes a SQLite database. The application specifications had to be updated during the development.</p> <p>The thesis was moderately successful. There was no clear written timetable, so time management wasn't very efficient. However, the end product is a working application.</p>	
Keywords	diabetes, blood sugar, Android

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Projektin tausta ja tarve	1
2.1	Diabetes mellitus	2
2.2	Diabetes mellitus lapsipotilailla	4
3	Projektin tavoitteet, kohderyhmät ja hyödynsaajat	4
4	Työprosessi	5
4.1	Määrittely	5
4.2	Riskit	5
4.3	Suunnittelu	6
4.4	Toteutus	6
4.5	Testaus	7
5	Käytetyt tekniikat ja toteutus	7
5.1	Java	7
5.2	Android SDK	7
5.3	Android Studio	8
5.4	XML	8
5.5	SQLite	8
5.6	Sovelluksen toiminta	9
5.6.1	MainActivity	9
5.6.2	BloodSugarActivity	10
5.6.3	Hiilihydraattitietovisa	15
5.6.4	SQLite	19
5.7	Tietovisan kysymykset ja vastaukset	20
6	Jatkokehitysehdotukset	22
7	Lopuksi	24
	Lähteet	26

## Lyhenteet

API	Application programming interface. Ohjelmointirajapinta.
IDE	Integrated Development Environment. Sovelluskehitysympäristö.
LADA	Latent Autoimmune Diabetes in Adults. Diabeteksen tyyppi.
mmol	Millimooli. Kansainvälisen yksikköjärjestelmän mukainen mittayksikkö ainemäärälle n.
MODY	Maturity-Onset Diabetes of the Young. Diabeteksen tyyppi.
SDK	Software Development Kit. Joukko työkaluja, joita käytetään ohjelmistokehityksessä.
XML	Extensible Markup Language. Tiedon kuvausmuoto.

## 1 Johdanto

Opinnäytetyön aiheena on sokeritason seurantaan käytettävä sovellus Android-puhelimelle. Sovellus on suunnattu lapsille. Kirjoittajaa aihe kiinnostaa siksi, että Suomessa on enemmän diabetesta sairastavia lapsia kuin missään muualla maailmassa eikä sairauden hoitotasapainoa yleensä saavuteta nuorilla potilailla [Kivelä & Keskinen: 2016]. Aihe valikoitui kiinnostavuuden takia; kirjoittaja opiskelee sekä tietotekniikkaa että sairaanhoitotyötä. Diabetesta sairastavien lapsipotilaiden heikon hoitotasapainon vuoksi on mielekästä kehittää uusia menetelmiä, jotka ovat mahdollisimman monen potilaan käytettävissä. Tämä oli myös yksi syy aiheen valituksi tulemiselle ja sen rajaamiselle lapsipotilaisiin.

Opinnäytetyön tavoitteena on tuottaa verensokerin seurantasovellus Android-puhelimille. Sovelluksen tulisi olla helppokäyttöinen, sillä kohteena ovat lapsipotilaat. Helppokäyttöisyys yritetään saavuttaa yksinkertaistetulla käyttöliittymällä ja helpposelitteisillä valikoilla. Lapset hyötyvät sokeritasapainon seurannasta tulevaisuuttaan ajatellen. Jos sairauden omahoito tehdään lapsuudesta asti tutuksi, voidaan mahdollisesti ehkäistä aikuisiällä ilmenevien liitännäissairauksien ilmaantuminen.

Työssä perehdytään diabetekseen, sokeriarvojen seurantaan ja hiilihydraattien laskemiseen. Sovellus kehitetään Javalla käyttäen Android APIa. Sovellukseen tallennetut sokeriarvot tallennetaan niin, että niistä on mahdollista nähdä tallennuksen ajankohta ja tallennettu arvo. Tärkeimpiä tietolähteitä ovat Android API, Suomen Diabetesliiton julkaisut, Terveyskirjasto ja Käypä hoidon-suositukset.

## 2 Projektin tausta ja tarve

Projektin toteutti opiskelija, joka opiskelee sekä ohjelmistotekniikkaa että sairaanhoitotyötä. Tekijä kiinnostui aiheesta hoitotyön harjoittelun aikana. Hän huomasi, että useat insuliinihoitoista diabetesta sairastavat henkilöt pitivät kirjaa veren glukoosipitoisuudestaan joko paperilla tai jättivät arvojen kirjaamisen kokonaan.

Tehdessään taustatyötä projektin toteuttaja huomasi erityisesti lapsipotilaiden hoitotasapainon olevan heikko. Kivelän ja Keskinen tutkimuksen mukaan internetin, älypuhelimien

ja opetuspelien käyttö vaikutti lasten hoitotasapainoon. [Kivelä & Keskinen: 2016.] Nykypäivänä lapsille kuitenkin hankitaan puhelin käyttöön jo nuorella iällä. DNA:n kyselytutkimuksen mukaan osa lapsista saa ensimmäisen puhelimensa aloittaessaan opintonsa ala-asteella, ja jo 86 %:lla 8-vuotiaista lapsista on älypuhelin [DNA:n Koululaistutkimus 2016: Koululaisten matkapuhelinten hankinta ja käyttö: 2016]. Android-käyttöjärjestelmän älypuhelimeen ei kuitenkaan löytynyt suomenkielistä sovellusta veren glukoosipitoisuuden kirjaamiseen.

Diabeteksen yhteydessä seurattavan veren glukoosipitoisuuden arvojen tallentamiseen käytettävä älypuhelinsovellus tuntui mielenkiintoiselta ja sopivan haastavalta projektilta toteuttaa.

## 2.1 Diabetes mellitus

Diabetes on sairaus, jossa veren glukoosipitoisuus on pysyvästi yli 7,0 mmol/l, vaikka potilas olisi paastonnut yön yli. Se on aikaisemmin jaettu kahteen eri tyyppiin: tyyppiin 1 ja tyyppiin 2. Diabetesta on kuitenkin olemassa useampaa eri tyyppiä kuin edellämainittu. Diabeteksen tyypit ovat:

- Tyypin 1: haima ei tuota insuliinia, sillä sitä tuottavat solut ovat tuhoutuneet autoimmuuniprosessin johdosta. Johtaa hoitamattomana kuolemaan.
- Tyypin 2: insuliinin vaikutus potilaassa on heikentynyt ja sen tuotanto haimassa on alentunut. Taudin puhkeamiseen vaikuttavat ylipaino, kohonnut verenpaine tai rasva-aineenvaihdunnan häiriö tai molemmat (metabolinen oireyhtymä).
- Raskausdiabetes: raskaudenaikainen veren glukoosipitoisuuden kasvaminen, joka palautuu normaaliksi synnytyksen jälkeen.
- LADA: muistuttaa tyypin 2 diabetesta, mutta insuliinin puutos kehittyy nopeammin, kuin tyypissä 2. Alkaa useimmiten, kun potilas on yli 35 -vuotias.
- MODY: nuorten diabetestyyppi, joka alkaa ennen 25 vuoden ikää. Tyyppiin liittyy yli kymmenen eri geenin mutaatio.

- Sekundaarinen diabetes: sairauden aiheuttaa esimerkiksi haimatulehdus tai muu haiman sairaus.
- Neonataalidiabetes: sairaus todetaan alle kuuden kuukauden ikäisellä potilaalla. Sairaus voi olla pysyvä tai ohimenevä.
- Mitokondriaalinen diabetes: potilas on perinyt äidiltään harvinaisen mitokondriaalisen DNA:n mutaation. Taudinkuvaan liittyy myös heikentynyt kuulo, neurologiset oireet, lihaksiston oireet ja potilaan pieni koko. [Diabetes. Käypä hoito -suositus 2016.]

Tässä työssä keskitytään vain tyyppiin 1. Tyypin 1 diabeteksessa haiman insuliinia tuottavat solut ovat tuhoutuneet autoimmuuniprosessin johdosta, eikä potilaalla ole lainkaan omaa insuliinituotantoa. Tämän tyypin diabetes alkaa tyypillisesti nuoruusiässä. [Mustajoki: 2015.]

Tyypin 1 diabetekseen sairastuneen henkilön glukoosiarvoilla on yleiset tavoitteet. Paastoarvon tulisi olla sairastuneella alle 7 mmol/l ja aterian jälkeisen glukoosipitoisuuden arvo tulisi olla alle 10 mmol/l. Arvon ylittyessä on kyse hyperglykemiasta, eli korkeasta verensokerista. [Diabetes. Käypä hoito -suositus 2016.] Glukoosiarvon alaraja on 4,0 mmol/l. Kun glukoosiarvo menee tämän arvon alle, on kyse hypoglykemiasta, eli alhaisesta verensokerista. Hypoglykemian itsehoitona toimii nopeasti imeytyvien hiilihydraattien pikainen nauttiminen. Nopeasti imeytyviä hiilihydraatteja saa muun muassa hunajasta, tuoremehusta ja veteen liuotetusta sokerista. Jos alhainen verensokeri on johtanut tajuttomuuteen eli insuliinisokkiin, tulee pakalle soittaa ambulanssi. [Mustajoki 2018.]

Diabeetikon tulee itsenäisesti seurata verensä glukoosipitoisuutta mittaamalla verensokeri. Verensokeri voidaan mitata sormenpäältä (kapillaariglukoosimittaus) tai kudosgluukoosia mittaavan sensorin avulla. Mittauksien tiheyden päivää kohti määrittelee lääkäri, mutta pääsääntöisesti mittaus tulisi tehdä aamulla herätessä ennen ateriaa ja illalla nukkumaan mennessä. Aamun ja illan mittausten tarkoitus on seurata, miten veren glukoosipitoisuus vaihtelee paaston aikana pitkävaikutteista insuliinia käytettäessä. Lisäksi verensokeri tulee mitata aina, jos epäilee hypo- tai hyperglykemiaa. Lisäksi esimerkiksi lääkemuutosten tai akuutin sairauden sattuessa mittausten aikaväli tulisi arvioida uudelleen. [Diabetes. Käypä hoito -suositus 2016.]



## 2.2 Diabetes mellitus lapsipotilailla

Suomessa diabetes mellitus on lapsilla yleisempi kuin missään muualla maailmassa. Vuoteen 2005 asti alle 15-vuotiaiden diabeetikkojen määrä kasvoi vuosittain. [Kivelä & Keskinen: 2016.] Suomessa oli vuonna 2014 noin 4000 diabetesta sairastavaa lasta [Jalanko: 2014].

Kivelän ja Keskinen mukaan suomessa noin 54 – 84 % lapsista ei saavuta toivottavaa hoitotasapainoa [Kivelä & Keskinen: 2016]. Hoitotasapainoa kuvataan mittaamalla soke-rihemoglobiiniarvoa (HbA1c-arvo), joka kuvaa verensokerin tasoa pitkällä aikavälillä. Mitä korkeampi arvo on, sitä korkeampi verensokeritaso on ollut [Eskelinen: 2016]. Vaikutuksia heikkoon hoitotasapainoon Kivelä ja Keskinen hakivat hoitoyksiköistä ja insuliinin annostelutavoista, mutta näillä ei ollut selvää heikentävää vaikutusta. Parhaiten hoitotasapainossa vaikuttivat olevan kahden vanhemman perheessä olevat lapset, vaikkakin yksinhuoltajaperheissäkin saavutettiin hyvä hoitotasapaino, kun käytössä oli perhe-suuntautunut käyttäytymisterapia. Myös internetin, älypuhelimien ja opetuspelien käytöllä havaittiin olevan vaikutusta hoitoon. Kirjoittajien mukaan iso osa diabetesta sairastavien lasten vanhemmista hakee sairauteen liittyvää tietoa internetistä. Lääkehoitoon liittyvillä opetuspeleillä potilaat perheineen ovat saaneet lisätietoa sairaudesta, mutta pitkäaikaiset vaikutukset hoitoon ovat jääneet vähäisiksi. Pelien vapaaehtoisuus ja tiedon saatavuus internetistä ovat vaikeuttaneet potilaiden sitouttamista hoitoon. [Kivelä & Keskinen: 2016.]

## 3 Projektin tavoitteet, kohderyhmät ja hyödynsaajat

Android on maailman suosituin käyttöjärjestelmä puhelimille ja muille mobiililaitteille. [Android.com, 2018]. Opinnäytetyön tavoitteena onkin kehittää Android-puhelimelle toteutettava Java-pohjainen sovellus. Tavoitteena on kerrata Android-ohjelmointia ja saada lisäharjoitusta. Tavoitteena on myös tehdä sovellus, jota nuoremmatkin lapset oppisivat helposti käyttämään. Sovelluksen on myös tarkoitus ohjata lapsia hiilihydraattien laskemisessa, jotta lääkehoito onnistuisi paremmin.

Opinnäytetyön hyödynsaajia ovat lapset ja heidän vanhempansa. Myös terveydenhuollon henkilökunta kuten lääkärit ja hoitajat hyötyvät sovelluksesta, sillä siihen tallennetaan arvokasta tietoa potilaan verensokeriarvoista.

## 4 Työprosessi

Opinnäytetyön tekeminen aloitettiin loppuvuodesta 2015 aiheen valinnalla. Opinnäytetyö toteutettiin kokonaisuudessaan etätöinä.

### 4.1 Määrittely

Aluksi tehtiin määrittely. Käyttäjryhmäksi valikoituivat lapset aiemmin mainitun huonon hoitovasteen vuoksi. Käyttöliittymän tulisi näin ollen olla mahdollisimman selkeä ja helpokäyttöinen, sillä käyttäjät ovat iältään nuoria. Tietovisaan ei käyttäjien iän vuoksi laiteta aikarajaa kysymykseen vastaamiselle, kun ei voi olla varma käyttäjän lukunopeudesta. Glukoosiarvojen tallennus tehdään tiedostoon. Tiedostoon tallennetaan päivämäärä, kellonaika ja glukoosiarvo. Tiedon onnistuneesta tallentamisesta tiedostoon tulee jonkinlainen ilmoitus käyttäjälle.

Sovellus ottaa osaltaan kantaa glukoosiarvoon. Jos lapsen syöttämä arvo on alle 4 mmol/l, tulee sovelluksen ehdottaa syömistä. Jos lapsen syöttämä arvo on yli 10 mmol/l, tulee sovelluksen kysyä, tarvitseeko lapsi insuliinia. Glukoosiarvon ollessa viitearvon puitteissa sovellus ei ilmoita lisätietoa käyttäjälle.

Määrittelyä tarkennettiin tarpeen mukaan suunnittelun aikana.

### 4.2 Riskit

Opinnäytetyön suurimpana riskinä pidettiin tiedon ja ohjelmakoodin katoamista. Kirjallinen tuotos ja ohjelmakoodi tallennettiin molemmat moneen eri paikkaan aina niitä muutettaessa. Tiedostot löytyvät opinnäytetyön kirjoittajan omalta työasemalta, Metropolian verkkolevyiltä, pilvipalvelusta, USB-muistitikulta ja verkkotallennuksena.

Toisena suurena riskinä pidettiin ajankäytön hallintaa. Organisoinnin puute ja toisen opintolinjan tehtävien kuormittavuus huolestuttivat. Ajankäytön ongelmien mahdollinen esiintyminen saattaisi johtaa kolmanteen riskiin: opinnäytetyön jäämiseen kesken.

Viimeisenä riskinä ja ongelmakohtana pidettiin osaamistasoa. Vaikka opinnäytetyön yhtenä tavoitteena on kerrata ja oppia lisää Android-ohjelmoinnista, voi tehtävä osoittautua vaikeaksi toteuttaa.

#### 4.3 Suunnittelu

Sovelluksen suunnittelu aloitettiin määrittelyn pohjalta. Suunnittelu aloitettiin glukoosiarvon tallentamisesta. Tiedostomuodoksi valikoitui .txt. Sovelluksen sivulla päätettiin esittää päivämäärä ja kellonaika sen mukaan, milloin aktiviteetti on käynnistetty. Tämän lisäksi päätettiin lisätä mahdollisuus muuttaa päivämäärää ja kellonaikaa poikkeustapausten varalta. Tällaisia tapauksia voivat olla esimerkiksi puhelimen akun loppuminen ennen glukoosiarvon tallennusta, tai puhelimen puuttuminen syystä tai toisesta tilanteessa, jolloin arvo pitäisi tallentaa.

Tietovisasta päätettiin tehdä monivalintainen: annetaan yksi kysymys, johon on kolme eri vastausvaihtoehtoa. Vastauksen jälkeen pitäisi näyttää oikea vastaus ja antaa piste oikeasta vastauksesta. Tietovisan kysymykset ja vastaukset esitellään luvussa 8.

#### 4.4 Toteutus

Sovellus toteutettiin käyttämällä Android Studio -nimistä ohjelmistoa. Käyttöjärjestelmäversion tasoksi valikoitui API-taso 25. Glukoosiarvon tallentamiselle varattiin oma aktiviteetti, jonka ulkoasu toteutettiin XML-tiedoston avulla. Aktiviteettiin toteutettiin tiedon kellonajan muuttaminen, päivämäärän muuttaminen ja tiedon tallentaminen tiedostoon. Ennen glukoosiarvon tallennusta tehdään arvolle tarkistus. Arvon tulee olla numeerinen eikä tyhjää kenttää hyväksytävä vastaukseksi.

Tietovisa toteutettiin omana aktiviteettinaan. Tämänkin ulkoasu toteutettiin XML-tiedoston avulla. SQLite-tietokantajärjestelmä valittiin kysymysten tallentamiseen. Kysymyksiä varten luotiin oma luokka, johon määriteltiin kysymysoliolle muuttujat. Aktiviteetissa näkyy pelille otsikko, pisteiden lukumäärä, kysymys ja kysymysvaihtoehdot. Vastaus luki-taan ennen oikean vastauksen ilmoittamista. Oikea vastaus ilmoitetaan tekstinä kysymyksen tilalla. Oikeita vastauksia per kysymys on yksi. Jos kysymyksiä on jäljellä, siirrytään seuraavaan kysymykseen painikkeen painalluksella. Jos kysymyksiä ei enää ole, lopetetaan tietovisa painikkeen painalluksella ja palataan alkuun.

## 4.5 Testaus

Sovellusta testattiin pääsääntöisesti käyttämällä Android Studion tarjoamaa emulaattoria. Emulaattorissa käytettiin eri puhelimia ja käyttöjärjestelmäversioita. Suurin osa testauksesta toteutettiin Android-käyttöjärjestelmän versiolla 7.1.1.

Testauksessa huomioon otettiin esimerkiksi väärää tietotyyppiä olevien arvojen tallentamisen yrittäminen ja tyhjän arvon tallentamisen yritys. Tietovisassa yritettiin saada tallennettua ylimääräisiä vastauksia. Testissä yritettiin myös saada tallennettua vastaus ilman, että yhtään vastausta oli valittu. Visassa testattiin myös mahdollisuus vaihtaa vastaus toiseen vastauksen lukitsemisen jälkeen. Tietovisasta poistuminen kesken kaiken onnistuu myös ongelmitta. Testeissä ei ilmennyt virhetilanteita.

## 5 Käytetyt tekniikat ja toteutus

Tässä luvussa esitellään opinnäytetyössä käytetyt tekniikat ja ohjelmisto, ja käydään läpi sovelluksen toiminta.

### 5.1 Java

Java on yleiskäyttöinen oliopohjainen korkean tason ohjelmointikieli, joka on suunniteltu mahdollisimman yksinkertaiseksi. Java on C- ja C++-kielten sukulaiskieli. Javan on tarkoitus olla niin helppokäyttöinen kieli, että suurin osa ohjelmoijista pystyy käyttämään sitä sujuvasti; siihen kuuluu muun muassa automaattinen roskienkeruu. [Oracle 2018.]

### 5.2 Android SDK

Android SDK sisältää työkaluja, joita käytetään Android-sovellusten kehityksessä. Nämä työkalut on jaettu useaan osaan:

- SDK Tools sisältää työkalut ohjelmistokehitykseen ja virheiden etsimiseen ja poistamiseen (debugging) [Android Studio 2017a].
- SDK Build Tools sisältää työkalut ohjelmiston koontiin [Android Studio 2017b].

- SDK Platform Tools sisältää työkaluja jotka toimivat rajapintana sovellusalueeseen. Tällaisia työkaluja ovat muun muassa fastboot ja systrace. Työkalut ovat välttämättömiä Android-sovelluksen kehittämisessä. [Android Studio 2017c.]
- Android Emulator on osa Android SDK:ta. Sen avulla pystyy testaamaan kehitettävää sovellusta käytännössä. [Android Studio 2017d.]

### 5.3 Android Studio

Android Studio on virallinen IDE Android-sovellusten kehitykseen. Se perustuu IntelliJ IDEA IDE:hen, joskin ohjelmiston päälle on lisätty paljon muita Android-sovellusten kehittämisessä käytettäviä ominaisuuksia. [Android Studio 2018.] Android Studio sisältää muun muassa nopean Android-emulaattorin, työkaluja ja viitekehukset sovelluksen testaukseen sekä kääntämisen työkaluja. Kääntämisen peruskivenä IDE:ssä toimii Gradle Build System. Android Studiosta on versiot Windowsille, Mac OS X:lle ja Linuxille. [Android Studio.]

### 5.4 XML

XML, eli Extensible Markup Language, on yksinkertainen tiedon kuvausmuoto. Se suunniteltiin tiedon tallentamista ja siirtoa ajatellen. XML ei kuitenkaan tee mitään, vaan sitä käytetään ainoastaan tiedon kuvaamiseen. Tietoa kuvataan elementtien avulla [W3C 2016]. Android-sovelluskehityksessä XML on käytössä käyttöliittymän ulkoasun suunnittelussa.

### 5.5 SQLite

SQLite on tietokantajärjestelmä. Se on tietokantajärjestelmänä poikkeuksellinen, sillä se ei käytä erillistä tietokantapalvelinta, vaan on ikään kuin sulautettu sitä käyttävään ohjelmaan. Sen ei siis tarvitse avata erillistä yhteyttä tietokantaan. Tietokanta pidetään koneen muistissa. SQLite on tekijänoikeudeton, joten sitä voi käyttää veloitusetta yksityisessä tai kaupallisessa käytössä. [About SQLite.]

## 5.6 Sovelluksen toiminta

Sovelluksen ohjelmointiin käytettiin Javaa. Kehitysympäristönä toimi Android Studio ja siihen liittyvä puhelinemulaattori, jolla sovellusta testattiin käytännössä. Sovellusta testattiin myös fyysisellä päätelaitteella. Käytössä oli tällöin OnePlus 5.

### 5.6.1 MainActivity

Sovelluksen käynnistyessä aukeaa ensimmäinen aktiviteetti – päänäkymä. Päänäkymässä on kaksi painiketta, joita painamalla siirrytään muihin aktiviteetteihin. Painikkeet on nimikoitu aktiviteettejä kuvaavasti.



Kuva 1. MainActivity:n painikkeet [Pixabay].

Molemmat painikkeet luotiin XML-tiedoston avulla seuraavasti:

```
<Button
    android:id="@+id/bs_button"
    android:layout_height="wrap_content"
    android:layout_column="0"
    android:layout_row="0"
    android:onClick="startBloodsugar"
    android:text="@string/blood_s" />
```

Esimerkkikoodi 1. Painikkeen luominen XML-tiedostossa

Molemmille painikkeille luotiin onClick-kuuntelija. Painikkeen painalluksesta käynnistyy MainActivity-luokassa olevan metodi, joka on painikekohtainen. Esimerkkikoodissa olevan bs\_button-painikkeen painaminen kutsuu startBloodsugar-metodia, joka käynnistää uuden aktiviteetin esimerkkikoodin 2 mukaisesti.

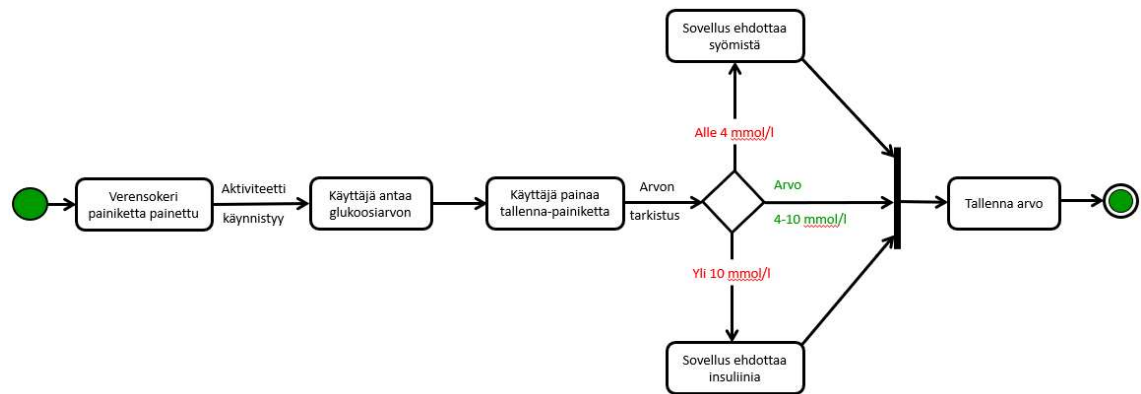
```
public void startBloodsugar(View view){
    Intent bs_intent = new Intent(this, BloodSugarActivity.class);
    startActivity(bs_intent);
}
```

Esimerkkikoodi 2. Aktiviteetin käynnistäminen MainActivity-luokassa.

Metodi startActivity saa parametrina intent-olion, joka sisältää tiedon siitä, kumpi aktiviteetti seuraavaksi käynnistetään. Kyseisessä esimerkissä seuraavaksi käynnistetään BloodSugarActivity-aktiviteetti.

## 5.6.2 BloodSugarActivity

Sokeritason seuranta varten luotiin oma aktiviteetti nimeltä BloodSugarActivity. Aktiviteetin perustason toiminta on esitetty kuvassa 1. Kuvassa ei kuvata mitään mahdollista virhetilannetta. Aktiviteetti käynnistyy, kun käyttäjä painaa painiketta MainActivity:n puolella. Tämän jälkeen käyttäjä antaa mittaamansa verensokeriarvon ja painaa tallennuspainiketta. Tällöin ohjelma arvioi annetun arvon perusteella, mitä tehdä seuraavaksi. Jos arvo on alle 4 mmol/l, ohjelma ehdottaa ruokailua. Arvon ollessa yli 10 mmol/l ehdottaa ohjelma insuliinin pistämistä. Jos arvo on näiden kahden arvon välillä, ohjelma ei ehdota toimintaa. Arvon arvioinnin jälkeen ohjelma tallentaa annetun arvon. Tiedon tallentumisesta ilmoitetaan käyttäjälle.

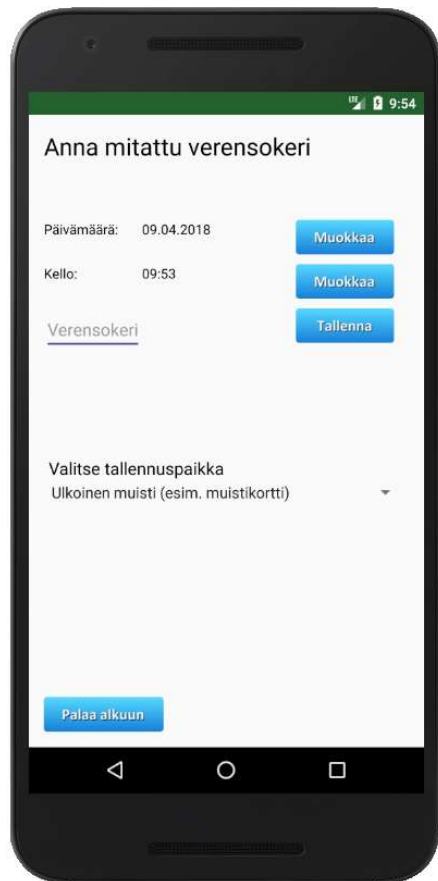


Kuva 2. Sovelluksen toiminta ilman virhetilanteita.

Aktiviteetin alustuksessa sille luotiin XML-tiedostosta ulkoasu. Tämän jälkeen ikkunassa näkyvät painikkeet, pudotusvalikko, ja tekstikentät alustettiin ohjelmakoodissa XML-komponenteiksi R-luokan avulla ja painikkeille määritettiin kuuntelijat. Kuuntelijoille lisättiin tarvittavat toiminnallisuudet:

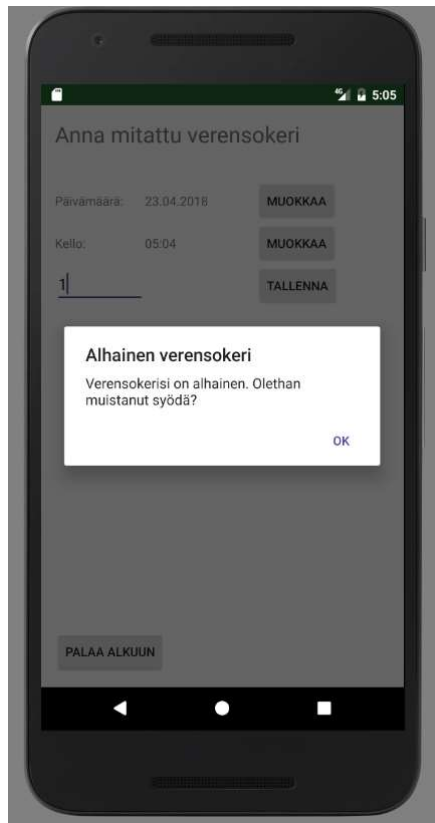
- Tallenna-painikkeen painaminen aloittaa annetun arvon tarkastelun. Tarkastuksen lopuksi luku tallennetaan.
- Päivämäärän muokkaamisen painike päästää käyttäjän muokkaamaan päivämäärää.
- Kellonajan muokkaamisen painike päästää käyttäjän muokkaamaan kellon-aikaa.
- Pudotusvalikosta käyttäjä voi valita, tallennetaanko tiedosto laitteen sisäiseen muistiin vai ulkoiseen muistiin.
- Palaa alkuun -painike palauttaa käyttäjän edeltävään aktiviteettiin.





Kuva 3. BloodSugarActivity:n näkymä.

Tallenna-painikkeen painamisen jälkeen käynnistyvä tarkastus katsoo, että verensokeriarvoksi määritetty luku on desimaaliluku eikä esimerkiksi kirjain tai jokin muu merkki. Jos annettu arvo on virheellinen, ohjelma pyytää käyttäjään antamaan glukoosiarvon numerona ja ilmoittaa, että numeron on oltava suurempi kuin 0. Virhetilanne tulee myös tapauksessa, jossa käyttäjä jättää kentän täysin tyhjäksi ja yrittää tallentaa tietoa. Tarkastuksessa katsotaan myös, onko arvo niin korkea tai matala, että käyttäjän tarvitsee reagoida siihen. Käyttäjälle ilmoitetaan tällaisen tilanteen sattuessa. Liian korkealla ja liian matalalla arvolla ilmoitukset ovat erilaisia.



Kuva 4. Ilmoitus, jos verensokeri on liian alhainen.

Ruudulla aktiviteettia käynnistettäessä näkyvät kellonaika ja päivämäärä ovat puhelimen nykyhetken mukaiset. Käyttäjälle annetaan mahdollisuus muuttaa mittauspäivämäärää ja kellonaikaa, jos syystä tai toisesta mittaus tuloksen merkitseminen järjestelmään unohdetaan tai jos merkkauksen ajankohta on mittausajasta muusta syystä poikkeava. Käyttäjän painaessa kellonajan vieressä olevaa muokkaa-painiketta (kuva 3) siihen liitetty kuuntelija laukaisee kellonajan muuttamiseen liittyvän setTime-metodin. Metodille lähetetään parametrina näytöllä näkynyt kellonaika kahtena eri kokonaislukuna: tunnit ja minuutit. Näiden avulla näkymän kellonaika asetetaan TimePickerin kellonajaksi. TimePickerin ulkoasu näyttää kellotaululta. Kellotaulusta valitaan haluttu tunti ja minuutti, jonka jälkeen painetaan Ok-painiketta. Tällöin kellonaika päivittyy date-muuttajaan, jonka jälkeen se päivitetään myös näytölle.

```

public void setTime(int hour, int min) {
    TimePickerDialog timePicker = new TimePickerDialog(this, new
    TimePickerDialog.OnTimeSetListener() {
        public void onTimeSet(TimePicker picker, int h, int m) {
            date.set(Calendar.HOUR_OF_DAY, h);
            date.set(Calendar.MINUTE, m);
            time_tw.setText(TIME_FORMAT.format(date.getTime()));
        }
    }, hour, min, true);
    timePicker.show();
}

```

### Esimerkkikoodi 3. Kellonajan muuttaminen TimePickerin kautta

Sovelluksessa oleva päivämäärän muuttaminen tapahtuu lähes identtisellä tavalla, mutta TimePicker-luokan sijaan käytetään DatePicker-luokkaa. Päivämäärää muutettaessa käyttäjä painaa päivämäärän vieressä olevaa muokkaa-painiketta näytöllä. Tällöin painikkeeseen liitetty kuuntelija laukaisee ajan muuttamiseen liitetyn setDate-metodin. Metodille annetaan parametrina näytöllä näkynyt päivämäärä kolmena erillisenä kokonaislukuna: päivä, kuukausi, vuosi. Näiden avulla DatePickerissä näkyvä päivämäärä asetetaan samaksi kuin näytöllä näkynyt päivämäärä. DatePickerin ulkoasu näyttää kalenterilta. Kalenteri näyttää yhden kuukauden kerrallaan. Käyttäjä valitsee kalenterista oikean päivämäärän ja painaa ok-painiketta. Tällöin päivämäärä päivitetään date-muuttajaan, jonka jälkeen se päivittyy myös näytölle.

Käyttäjän on mahdollista valita, minne tiedoston tallentaa. Valittavina vaihtoehtoina pudotusvalikossa ovat puhelimen sisäinen muisti ja ulkoinen muisti. Oletuksena valittuna on puhelimen sisäinen muisti. Siinä annettu verensokeriarvo tallennetaan sovelluksen asennuskansioon .txt-tiedostoon. Vain tiedoston luonut sovellus pääsee käsittelemään sitä. Ulkoiseen muistiin tallennettua tiedostoa pääsee muokkaamaan kuka tahansa. Ulkoiseen muistiin kirjoittamiseen ja sen lukemiseen tarvittavat oikeudet ilmoitetaan AndroidManifest-tiedostossa. Kun käyttäjä tallentaa ensimmäistä kertaa ulkoiseen muistiin, kysytään tallentamiseen lupa.

Tallennus tapahtumana alkaa, kun käyttäjä klikkaa tallenna-painiketta ja kuuntelija laukaisee siihen liittyvän metodin. Metodissa tarkastetaan ensin aiemmin esitetyllä tavalla, että annettu arvo on validi ja tämän jälkeen tarkastetaan, että arvo on raja-arvojen sisäpuolella. Jos annettu arvo ei ole validi, palataan takaisin kuuntelemaan painikkeen painalluksia. Jos arvo on alle 4 mmol/l, kysyy sovellus, onko käyttäjä muistanut syödä. Jos arvo on yli 10 mmol/l, sovellus kysyy käyttäjältä, tarvitsisiko tämä insuliinia. Arvon ollessa

viitearvojen mukainen ei järjestelmä anna ilmoitusta. Lopuksi metodi putkittaa tallennuksen onCreate:n ulkopuoliseen metodiin, jossa arvo lopulta tallennetaan. Tallennus suoritetaan käyttämällä FileOutputStream-luokan muuttujaa, jonka avulla tiedoston loppuun lisätään tekstiä. Jos tallennustiedostoa ei ollut aiemmin olemassa, luo olion alustaminen uuden tiedoston annettuun polkuun. Polku on haettu käyttämällä Environment-luokan muuttujia ja metodeita.

```
String documents = Environment.DIRECTORY_DOCUMENTS;
try{
    file = new File(Environment.getExternalStoragePublicDirectory(documents),
filename);
    outputStream = new FileOutputStream(file, true);
    outputStream.write(save_data.getBytes());
    outputStream.close();
    Toast toast = Toast.makeText(this, "Tallennus onnistui",
Toast.LENGTH_SHORT);
    toast.show();
}
}
```

Esimerkkikoodi 4. Tallentamisen try-lohko.

Tallennuksen jälkeen sovellus ilmoittaa tallennuksen onnistumisesta lyhytkestoisella Toastilla ”tiedon tallennus onnistui!”. Toastin käyttö on esitelty esimerkkikoodissa 4. Jos tallennus ei onnistu, antaa sovellus tästä ilmoituksen, joka pitää kuitata painamalla ok-painiketta.

15	Päivä ja kellonaika: 11.04.2018, 07:12	Verensokeri: 6.0
16	Päivä ja kellonaika: 11.04.2018, 08:10	Verensokeri: 6.7
17	Päivä ja kellonaika: 11.04.2018, 11:24	Verensokeri: 7.9
18	Päivä ja kellonaika: 11.04.2018, 15:45	Verensokeri: 4.7
19	Päivä ja kellonaika: 11.04.2018, 19:55	Verensokeri: 3.8
20	Päivä ja kellonaika: 11.04.2018, 20:20	Verensokeri: 5.7
21	Päivä ja kellonaika: 11.04.2018, 23:12	Verensokeri: 7.0
22	Päivä ja kellonaika: 12.04.2018, 07:23	Verensokeri: 4.5

Kuva 5. Sovelluksen avulla tiedostoon tallennettuja verensokeriarvoja.

Tiedostoon tallennetaan tieto järjestyksessä kuvan 5 osoittamalla tavalla: päiväys, kellonaika ja glukoosiarvo.

### 5.6.3 Hiilihydraattitietovisa

Peliin päästään painamalla painiketta sovelluksen aloitusnäkyssä. Painikkeen painallus laukaisee MainActivity:ssa startGame-metodin. Metodi saa parametrina intent-olion,

joka kertoo, mikä aktiviteetti seuraavaksi käynnistetään. Avautuvassa näkymässä on vielä erillinen aloituspainike, joka käynnistää tietovisa-aktiviteetin nimeltä QuizActivity.

Tietovisassa käytettäviä kysymyksiä varten luotiin oma Question-luokka, jonka pohjalta kysymysolioita voitiin luoda. Olion ilmentymämuuttujia ovat kysymys, kolme erillistä vastausta ja oikea vastaus. Kysymys ja vastaukset ovat tyypiltään merkkijonoja, ja oikea vastaus on tyypiltään kokonaisluku.

Tietovisa-aktiviteetin käynnistyessä sen ulkoasu haetaan XML-tiedostosta, näkymässä näkyvät painikkeet ja tekstikentät alustetaan R-luokan avulla ja määrättiin painikkeille kuuntelijat.

```
DataBaseHelper helper = new DataBaseHelper(this);  
list = helper.getQuestions();  
total = list.size();  
Collections.shuffle(list);
```

Esimerkkikoodi 5. Tietokannan luominen, kysymysten hakeminen ja niiden sekoittaminen.

Peli käynnistyy esimerkkikoodin 4 mukaisesti. Ensin alustetaan muuttuja nimeltä helper, joka on DataBaseHelper-luokkaan kuuluva olio. Muuttuja saa tietokannasta kaikki kysymykset, joita tietovisassa käytetään. Nämä kysymykset annetaan list-muuttujalle. List-muuttuja on List-luokan olio, joka ottaa vastaan Question-luokan olioita. Total-muuttujaan otetaan talteen tieto siitä, montako kysymystä list-muuttuja sai. Kysymykset vielä sekoitetaan niin, etteivät ne tule pelaajalle aina samassa järjestyksessä.



Kuva 6. Tietovisan näkymä. Osaan kysymyksistä on vastattu oikein, jolloin pisteitä on kertynyt.

Pelin näkymässä on otsikko, pisteytys, kysymys, vastausvaihtoehdot ja vastauksen lukitsemiseen käytettävä painike. Otsikko pysyy pelin ajan samana. Kysymys haetaan nextQuestion-metodin avulla. Myös pelin alkaessa ensimmäinen kysymys haetaan näin. Metodissa muutetaan ensin valintanappuloiden vieressä oleva teksti oletusväriksi (musta). Tämän jälkeen Question-olioista koostuvasta list-muuttujasta haetaan tämänhetkinen kysymys current-olioon ja asetetaan näkymässä näkyvä kysymys ja vastausvaihtoehdot sen mukaiseksi. Counter-muuttuja kertoo, missä kysymyksessä mennään, joten sitä voidaan kasvattaa yhdellä, kun tämänhetkinen kysymys on jo otettu talteen. Lopuksi asetetaan boolean-tyyppin muuttuja answerDone epätodeksi ja vaihdetaan painikkeen nimeksi ”Lukitse vastaus.”

```

private void nextQuestion(){
    radioGroup.clearCheck();
    radio1.setTextColor(defaultRadioColour);
    radio2.setTextColor(defaultRadioColour);
    radio3.setTextColor(defaultRadioColour);

    if (counter<total){
        current = list.get(counter);
        question.setText(current.getQuestion());
        radio1.setText(current.getAnswer1());
        radio2.setText(current.getAnswer2());
        radio3.setText(current.getAnswer3());

        counter++;
        answerDone = false;
        lockAnswer.setText("Lukitse vastaus");
    } else {
        finished();
    }
}

```

#### Esimerkkikoodi 6. Kysymksen asettaminen ennen vastausta

Lukitse vastaus -painikkeen painaminen laukaisee metodin, jolla tarkistetaan vastaus. Jos käyttäjä painaa painiketta ennen vastauksen valintaa, annetaan virheilmoituksena teksti: valitse vastaus. Vastauksen lukitsemisen kysymys vaihtuu tekstiksi, joka kertoo, mikä on oikea vastaus kysymykseen. Tällöin myös painikkeen teksti muuttuu. Tilalle tulee teksti "Seuraava kysymys."



Kuva 7. Käyttäjä on vastannut kysymykseen oikein.

Oikeaa vastausta auttaa hahmottamaan myös vastausvaihtoehdon tekstin muuttuminen vihreäksi. Väärissä vaihtoehdoissa vastaus pysyy mustana. Siirryttäessä seuraavaan ky-

symykseen kasvaa pistemäärä - jos vastaus on ollut oikea - kysymys ja vastausvaihtoehdot vaihtuvat uusiin ja valintanappulan valinta tyhjenee. Tietovisan loppuessa pisteet siirretään näkymään pelin aloitusnäkyeseen GameActivityyn.

#### 5.6.4 SQLite

Tietovisaa varten luotiin oma luokka nimeltä DataBaseHelper. Luokka käsittelee SQLite-tietokantaa. Kysymyksiä varten luotiin Question-luokka, joka sisältää kysymysten käsittelyn. Tietovisan käynnistyessä alustetaan esimerkikoodin 4 mukaisesti DataBaseHelper-luokan muuttuja helper. Ensimmäisen käynnistyksen yhteydessä sen alustaminen luo tietokannan ja lisää sinne kysymyksiä. Muilla käynnistyskerroilla tietokantaa ei luoda uudelleen, ellei sen versionumero ole muuttunut. Muuttujan alustamisen jälkeen tietokannasta haetaan kaikki kysymykset list-muttuunaan.

Tietokannassa on quiz-niminen taulu, joka sisältää kentät kysymykselle, vastausvaihtoehdoille ja oikealle vastaukselle. Tauluun lisätään peliä käynnistettäessä viisi kysymystä vastauksineen ja vastausvaihtoehtoineen.

```
private void insertQuestions(){
    Question q1, q2, q3, q4, q5;
    q1 = new Question("Mikä näistä sisältää eniten hiilihydraatteja?",
"Banaani", "Iso omena", "Kerroshampurilainen", 3);
    addToDB(q1);
    q2 = new Question("Mikä näistä sisältää vähiten hiilihydraatteja
desilitrassa?", "Tuoremehu", "Maito", "Limppari", 2);
    addToDB(q2);
    q3 = new Question("Kummassa on enemmän hiilihydraatteja?", "Molem-
missa saman verran", "Hillomunkki", "Korvapuusti", 1);
    addToDB(q3);
    q4 = new Question("Missä näistä on eniten hiilihydraatteja?", "Pe-
runa (keskikokoinen)", "Keitetty riisi (1 dl)", "Makaroni (1 dl)", 2);
    addToDB(q4);
    q5 = new Question("Mikä sisältää eniten hiilihydraatteja?", "Pika-
nuudelit", "Hampurilainen", "Sipsit (5 dl)", 1);
    addToDB(q5);
}
```

Esimerkkikoodi 7. Kysymysten lisääminen tauluun. Konstruktorin mukainen kysymysolion luominen.

Esimerkkikoodi 6 esittelee kysymysten ja vastausten lisäämisen tietokantaan. Tietokannalle annetaan järjestyksessä kysymys, vastausvaihtoehdot ja oikean vastauksen numero. Tämän jälkeen jokainen kysymys laitetaan yksitellen tietokantaan kutsumalla addToDB-metodia ja antamalla sille parametrina kysymys esimerkikoodin 7 mukaisesti.



Ensin metodissa luodaan luokan ContentValues muuttuja cv, jolle annetaan avain-arvo-pareja tietokantaan menevästä tiedosta.

```
private void addToDB(Question q){
    ContentValues cv = new ContentValues();
    cv.put(QTable.QUESTION, q.getQuestion());
    cv.put(QTable.ANSWER1, q.getAnswer1());
    cv.put(QTable.ANSWER2, q.getAnswer2());
    cv.put(QTable.ANSWER3, q.getAnswer3());
    cv.put(QTable.CORRECT_ANSWER, q.getCorrectAnswer());
    db.insert(QTable.TABLE_NAME, null, cv);
}
```

Esimerkkikoodi 8. AddToDB-metodin sisäinen ohjelmakoodi.

DataBaseHelper-luokka käsittelee tietokantaa. Luokassa on myös metodi, jonka avulla tietovisan kysymykset saadaan peliä käsittelevän luokan tietoon. Metodissa tietokannasta haetaan kysymykohtaisia arvoja yksi kerrallaan ja lisätään ne Question-luokan muuttujaan. Kun muuttuja on saanut kaikki tarvitsemansa parametrit, lisätään se list-muuttujaan. Kyseinen muuttuja on List-luokan olio, joka ottaa vastaan Question-luokan olioita. Metodin lopuksi palautetaan kysymysoliot sisältävä list-muuttuja.

## 5.7 Tietovisan kysymykset ja vastaukset

Kysymysten valmistaminen tietovisaan oli haastavaa. Kysymykset piti pitää selkeinä ottaen huomioon pääasiallisen kohderyhmän ikäjakauman. Kysymyksiin oli alun perin suunniteltu myös vaikeampia kysymyksiä, joissa joutuisi mahdollisesti laskemaan kertolaskuja. Näistä luovuttiin myös. Myös kieliasu piti pitää mahdollisimman selkeänä. Tämä oli hankalaa, sillä usein kysymyksiin näytti tulevan mukaan joko kiertoilmaisuja tai turhaa kapulakieltä.

Loppujen lopuksi kysymyksiä kertyi viisi. Viidestä kysymyksestä kolme on käytännössä identtisiä idealtaan, mutta niiden muotoilu on hieman erilainen. Kolmen kysymyksen samankaltaisuuden pystyi perustelemaan sillä, että niille valikoituneet vastausvaihtoehdot tarkoitettiin herättämään ajatuksia visailijassa.

Kysymykset ovat:

- Mikä näistä vaihtoehdoista sisältää eniten hiilihydraatteja?

- Mikä näistä vaihtoehdoista sisältää vähiten hiilihydraatteja?
- Kummassa on enemmän hiilihydraatteja?

Ensimmäisenä mainittua kysymystä on kolme kappaletta. Vastaukset kuitenkin eriävät toisistaan. Vastausvaihtoehdot näihin kolmeen kysymykseen ovat

- banaani, keskikokoinen omena, mandariini
- peruna, keitetty riisi, makaroni
- pikanuudelit, hampurilainen, sipsit.

Vastausvaihtoehdoissa näkyy sovelluksessa sama annosmäärä kuin taulukossa 1.

Kysymysten tarkoituksena oli saada lapsi miettimään syömäänsä ruokaa. Tämä on tärkeää varsinkin silloin, jos lapsi joutuu käyttämään pikavaikutteista insuliinia ruoan jälkeen. Tällöin pitäisi arvioida tarvittava insuliinin määrä ruokailua edeltävän verensokerin, syötyjen hiilihydraattien ja fyysisen aktiivisuuden mukaan. [Diabetes. Käypä hoito -suositus 2016.] Koulua aloittavalle diabetesta sairastavalle lapselle Sosiaali- ja terveysministeriön tekemän selvityksen mukaan nimetään henkilö tai henkilöitä, jotka huolehtivat lapsen ruokailusta ja mahdollisesta lisäruoasta [Toimintamalli diabetesta sairastavan lapsen koulupäivän aikaisesta hoidosta: 15]. Näin joku on auttamassa lasta arvioimaan syödyt hiilihydraatit, mutta lapsen tulisi oppia myös itse suorittamaan arviointia.

Vastauksiksi valitut tuotteet ja niiden hiilihydraattipitoisuudet saatiin Diabetesliiton ylläpitämästä arviointitaulukosta.

Taulukko 1. Sovelluksessa käytettyjen tuotteiden hiilihydraattimäärät grammoina annosta kohden [Apua hiilihydraattien määrän arviointiin].

Elintarvike/ruoka	Annos	Hiilihydraatit grammoina
Banaani, keskikokoinen	1 kpl	20
Hampurilainen	1 kpl	30
Hillomunkki	1 kpl	30
Korvapuusti	1 kpl	30

Maito	1 dl	5
Makaroni	1 dl	15
Mandariini	1 kpl	5
Omena, keskikokoinen	1 kpl	10
Peruna, keskikokoinen	1 kpl	10
Perunalastut	5 dl	30
Pikanoudelit	1 pussi	50
Riisi, keitetty	1 dl	20
Tuoremehu	1 dl	10
Virvoitusjuoma	1 dl	10

Osalla tuotteista hiilihydraattien määrä ei ollut Diabetesliiton arviointitaulukossa tarkka. Esimerkiksi riisin kohdalla hiilihydraattien määrä desilitraa kohden oli arvioitu olevan noin 15-20 grammaa. Myös perunalastujen kohdalla oli epätarkka arvio hiilihydraattien määrästä: 30-35 grammaa/annos. [Apua hiilihydraattien määrän arviointiin.]

Kysymyksissä käytetyt elintarvikkeet valikoituivat melko sattumanvaraisesti ajatellen sitä, mitä lapset ja nuoret syövät. Kouluruokana tarjotaan useimmiten perinteisiä suomalaisia ruokia, joista hiilihydraatteja saa nimenomaan vastauksiksi valituista pastasta, perunasta ja riisistä. Lisäksi mukaan päätettiin ottaa nuorten suosimia pikaruokia ja makeita välipaloja. Mukaan otettiin myös muutama erilainen juoma, jotta lapsi huomaa, että myös juomien hiilihydraatit tulee laskea mukaan ruokailun aikana nautittuun määrään.

## 6 Jatkokehitysehdotukset

Insinöörityönä tehtävä sovellus saatiin alkuperäisten vaatimusten mukaisena valmiiksi. Kuitenkin tehtävän tekemisen aikana kirjoittajalle tuli mieleen erilaisia jatkokehitysmahdollisuuksia, joita ei kuitenkaan ajan puitteissa ollut mahdollista toteuttaa välittömästi.

Glukoosiarvojen tallentamisen ensimmäisenä jatkokehitysideana on kauniimpi käyttöliittymä. Sovelluksen ollessa suunnattu lapsille ja nuorille voisi tämän ottaa käyttöliittymäsuunnittelussa huomioon; lapsille ja nuorille voisi olla erilaiset käyttöliittymät. Ne voisivat myös olla värimaailmoiltaan muokattavissa. Käyttöliittymän olisi myös hyvä olla joustavampi erilaisia laitteita ajatellen. Näin sovellusta voisi käyttää esimerkiksi puhelimella ja

tabletilta. Monen laitteen käyttämiseen liittyen kehitysehdotuksena on myös pilvipalveluun tallentaminen. Tällöin eri laitteilla voisi lisätä samaan tiedostoon arvoja eikä jokaiselta laitteelta tarvitsisi erikseen hakea tiedostoa.

Tiedoston tallentaminen eri formaattiin voisi myös olla käytännöllistä. Esimerkiksi Excel-tiedostoon tallentaminen auttaisi tiedon käsittelyssä: verensokerin mittausarvoista voisi piirtää viivadiagrammin, joka näyttäisi sokerin heilahtelut helpommin kuin pelkkien lukujen seuraaminen. Tiedon tallennuspaikan voisi myös antaa valita. Vaihtoehtoina voisi olla laitteen sisäinen muisti, ulkoinen muisti ja pilvipalvelu. Tällä hetkellä tieto tallentuu ainoastaan laitteen sisäiseen muistiin ja ulkoiseen muistiin. Sisäiseen muistiin tallennettuun tiedostoon ei pääse käsiksi sovelluksen ulkopuolelta helposti.

Lääketieteen kannalta olisi hyvä vaihtoehto, jos mitatut verensokeriarvot voisi siirtää suoraan hoitavalle lääkärille. Tämä kuitenkin asettaisi melkoisia vaatimuksia sekä sovellukselle että terveydenhuollon ohjelmistoille, eikä vaihtoehto näin ollen olisi kovin todennäköinen. Verensokerin viitearvojen muuttaminen olisi kuitenkin hyvä lisä ohjelmaan. Tämä vaatisi salasanan asettamisen ohjelman sisälle, ettei lapsi pääsisi muuttamaan asetuksia itse.

Tällä hetkellä sovelluksesta puuttuu myös mahdollisuus muokata aiemmin annettuja arvoja. Jos käyttäjällä sattuu virhepainallus, eikä hän huomaa sitä, tulee tiedostoon tallennetuksi väärä arvo, jota ei tällä hetkellä voi muuttaa.

Hiilihydraattitietovisan ensimmäisenä jatkokehitysideana on kauniimpi käyttöliittymä myös tietovisaa ajatellen. Ulkoasu on tällä hetkellä erittäin pelkistetty, mutta selkeä. Kysymyksiä voisi selventää myös kuvien lisääminen vastausvaihtoehdoille. Esimerkiksi leivälle sopisi kuva paahtoleipäviipaleesta. Tämä auttaisi hahmottamaan annoskokoja. Kuvien lisäksi myös teksti on hyvä olla näkyvillä.

Tietovisan nykyinen sisältö on rajallinen. Kysymysten lisääminen sovelluksesta käsin olisikin hyvä lisäominaisuus. Tällöin lapsi voisi oppia hallitsemaan itselleen mieluisten ruokien hiilihydraattipitoisuuksia. Kysymysten lisääminen pitäisi sekin laittaa salasanan taakse, jotta lapsi ei pääse omatoimisesti lisäämään asioita tietokantaan.

## 7 Lopuksi

Opinnäytetyön tavoitteena oli kehittää Android-puhelimille Java-pohjainen sovellus. Tavoitteena oli kerrata Android-ohjelmointia ja saada lisäharjoitusta. Tavoitteena ollut sovellus valmistui, joskin käyttöliittymässä on vielä parantamisen varaa. Tavoitteena oli kuitenkin saada aikaan toimiva sovellus, joten tässä mielessä tehtävä on suoritettu. Alun perin aiheeseen liittyvän kurssin jälkeenkin Android-ohjelmointi tuntui vaikealta, mutta opinnäytetyön tekeminen on selkeyttänyt kurssilta jääneitä kysymyksiä. Opinnäytetyön tuloksena on sovellus, jonka avulla diabeetikko voi tallentaa puhelimensa muistiin mitaamiaan glukoosiarvoja ja harjoitella hiilihydraattimäärien vertailua keskenään eri ruoissa.

Yhtenä opinnäytetyön tavoitteena oli luoda sovellus, jota lapsi osaa käyttää. Ajan puitteissa ei ollut kuitenkaan mahdollista järjestää tilaisuutta, jossa lapset olisivat päässeet testaamaan sovellusta puhelimillaan. Myös lapsilta saatu palaute sovelluksen ulkoasusta jäi puuttumaan.

Ennen opinnäytetyön tekemisen alkamista tekijä oli käynyt lähes kaikki tietotekniikan tutkinto-ohjelman vaatimat kurssit. Aiheen keksiminen opinnäytetyölle oli vaikeaa, sillä kirjoittajalla ei ollut asiakasyritystä, jolle tehdä opinnäytetyö. Työn idea muodostuikin vähitellen toisten opintojen ohella muodostuneista ideoista ja pohdinnoista. Kirjoittaja opiskelee sairaanhoitoa. Opintojen aikana suoritetaan useita työharjoittelujaksoja. Näiden aikana huomio kiinnittyi siihen, että diabetesta sairastavien henkilöiden verensokerin omaseuranta oli heikolla tasolla. Lastenkin kanssa seuranta toteutettiin niin, että verensokeriarvo merkittiin pieneen vihkoseen, joka hävisi helposti. Idea glukoosiarvojen seurantaan keskittyvästä sovelluksesta lähti näistä huomioista.

Aihetta rajattiin ja hiottiin opettajan ohjauksessa. Opinnäytetyötä tehtiin pelkästään vapaa-ajalla. Kirjoittaja ei suunnitellut projektille tarkkaa aikataulua, vaan projekti eteni pienten välietappien kautta eteenpäin. Tämä näkyikin toteutuksen tekemisessä voimakkaasti. Ajoittain projekti eteni vauhdilla, kun taas ajoittain työskentely takkusi, kun jokin ongelma ei ottanut ratketakseen. Ajankäyttöön vaikuttivat myös toisen linjan opinnot ja työssäkäynti samanaikaisesti. Alkuperäinen suunnitelma opinnäytetyön tekemisestä viikonloppuisin ei pitänyt.

Sovelluksen määrittely muuttui opinnäytetyön aikana. Kaikki alkuperäiset suunnitelmat eivät olleet ajan puitteissa mahdollisia toteuttaa ja osaa ideoista muokata, jotta kokonaisuus säilyi eheänä. Glukoosiarvojen tallentamisen osio onnistui hyvin. Alkuperäisen suunnitelman mukaan käyttäjä olisi vain avannut sovelluksen, valinnut arvojen tallennuksen, syöttänyt arvon tyhjäan kenttään ja tallentanut arvon puhelimeen. Tällöin päivämääräksi ja kellonajaksi olisi valikoitunut tallennushetki. Myöhemmin kuitenkin huomattiin, ettei arvoa aina välttämättä pääse kirjaamaan juuri mittaushetkellä; käyttäjälle voi tulla esimerkiksi odottamaton puhelu tai jotakin muuta voi sattua. Tällöin arvo olisi tallentunut väärälle ajankohdalle. Näin ajankohdan muuttaminen valikoitui mukaan sovelluskehityksen alkuvaiheessa.

Tietovisaosion määrittely muuttui sovelluskehityksen aikana huomattavasti enemmän. Alkuperäinen suunnitelma pelistä, jossa henkilö arvioi hiilihydraattien määrää grammoina usean eri vaihtoehdon joukosta vaikutti liian vaikealta lapsen ymmärtää. Ideaksi jäi myös vaihtoehto ruoan määrän esittämisestä kuvana; jos kuva on käyttäjälle epäselvä, on mahdotonta arvioida, mikä ruoka edes ylipäänsä on kyseessä.

Vaikka opinnäytetyön tekemisessä kesti, koen, että saavutin asettamani tavoitteet.

## Lähteet

About SQLite. SQLite. Verkkodokumentti. <<http://sqlite.org/about.html>>. Luettu 24.4.2018.

Android.com. 2018. Luettu 8.4.2018.

Android Studio. Verkkodokumentti. <<https://developer.android.com/studio/intro/index.html>>. Luettu 18.4.2018.

Android Studio 2018. Verkkodokumentti. <<https://developer.android.com/studio/intro/>>. Luettu 20.4.2018.

Android Studio 2017a. SDK Tools Release Notes. Verkkodokumentti. <<https://developer.android.com/studio/releases/sdk-tools.html>>. Luettu 18.4.2018.

Android Studio 2017b. SDK Build Tools Release Notes. Verkkodokumentti. <<https://developer.android.com/studio/releases/build-tools.html>>. Luettu 18.4.2018.

Android Studio 2017c. SDK Platform Tools Release Notes. Verkkodokumentti. <<https://developer.android.com/studio/releases/platform-tools.html>>. Luettu 18.4.2018.

Android Studio 2017d. Emulator Release Notes. Verkkodokumentti. <<https://developer.android.com/studio/releases/emulator.html>>. Luettu 24.4.2018.

Apua hiilihydraattien määrän arviointiin. Diabetesliitto. Verkkodokumentti. <<https://www.diabetes.fi/hiilihydraattitaulukko>>. Luettu 15.2.2018.

Diabetes. Käypä hoito -suositus 2016. Suomalaisen Lääkäriseuran Duodecimin, Suomen Sisätautilääkäreiden yhdistyksen ja Diabetesliiton Lääkärineuvoston asettama työryhmä. Helsinki: Suomalainen Lääkäriseura Duodecim. Verkkodokumentti. <<http://www.kaypahoito.fi/web/kh/suosituksset/suositus?id=hoi50056>>. Luettu 17.11.2016.

DNA:n Koululaistutkimus 2016: Koululaisten matkapuhelinten hankinta ja käyttö. Verkkodokumentti. <[https://www.dna.fi/documents/15219/31276591/DNA+Koululaistutkimus+2016\\_laaja+raportti\\_media.pdf/bbf90491-ba98-46c3-8303-7b6f5019b28e](https://www.dna.fi/documents/15219/31276591/DNA+Koululaistutkimus+2016_laaja+raportti_media.pdf/bbf90491-ba98-46c3-8303-7b6f5019b28e)>. Luettu 28.8.2016.

Eskelinen, Seija 2016. Hemoglobiini HbA1c (B-HbA1c). Terveyskirjasto. Verkkodokumentti. <[http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p\\_artikkeli=snk03092](http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=snk03092)>. Luettu 18.8.2016.

Jalanko, Hannu 2014. Diabetes lapsella. Terveyskirjasto. Verkkodokumentti. <[http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p\\_artikkeli=dlk00114](http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=dlk00114)>. Luettu 18.8.2016.

Kivelä, Laura; Keskinen, Päivi 2016. Lapsen diabetestasapainon parantaminen – Asenne on tärkein. Diabetes ja lääkäri 1/2016. Luettavissa myös verkossa osoitteessa <<http://diabeteslehti.diabetes.fi/lapsen-diabetestasapainon-parantaminen-assenne-on-tarkein/>>. Luettu 18.8.2016.

Mustajoki, Pertti 2015. Diabetes (sokeritauti). Terveyskirjasto. Verkkodokumentti. <[http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p\\_artikkeli=dlk00011](http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=dlk00011)>. Luettu 28.8.2016.

Mustajoki, Pertti 2018. Alhainen verensokeri (hypoglykemia) diabetesta sairastavalla. Terveyskirjasto. Verkkodokumentti. <[http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p\\_artikkeli=dlk00757](http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=dlk00757)>. Luettu 23.4.2018.

Oracle 2018. Java Language Specification. Verkkodokumentti. <<https://docs.oracle.com/javase/specs/jls/se8/html/jls-1.html>>. Luettu 17.4.2018.

Pixabay. Verkkodokumentti. <<https://pixabay.com/en/strawberry-red-rosa-network-pink-2850845/>>. Viitattu 15.2.2018.

Toimintamalli diabetesta sairastavan lapsen koulupäivän aikaisesta hoidosta. Helsinki: Sosiaali- ja terveysministeriön selvityksiä 2010:9. Saatavilla myös verkossa osoitteessa <<http://julkaisut.valtioneuvosto.fi/handle/10024/72726>>.

W3C 2016. Extensible Markup Language (XML). Verkkodokumentti. <<https://www.w3.org/XML/>>. Luettu 18.4.2018.