

Tomi Kenkkilä

## **ALOITTELEVAN YRITYKSEN MOBIILISOVELLUS**

## **ALOITTELEVAN YRITYKSEN MOBIILISOVELLUS**

Tomi Kenkilä  
Opinnäytetyö  
Syksy 2022  
Tietotekniikan tutkinto-ohjelma  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu

Tietotekniikan tutkinto-ohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

---

Tekijä: Tomi Kenkkilä

Opinnäytetyön nimi: Aloittelevan yrityksen mobiilisovellus

Työn ohjaaja(t): Eino Niemi

Työn valmistumislukukausi ja -vuosi: Syksy 2022

Sivumäärä: 29

---

Opinnäytetyön tavoitteena oli tutkia aloittavan pienyrityksen IT-tarpeita ja toteuttaa niiden pohjalta mobiiliapplikaatio Android-käyttöjärjestelmälle. Työssä tehtiin kysely, joka kohdistettiin sovelluksen kohderyhmille.

Pääasiallisesti sovelluksessa keskityttiin ajanhallinnan ja inventaarion toteutukseen sekä niiden yhteistoimintaan. Työ tehtiin lähtökohtaisesti omaan käyttöön harrastukseen, josta on myös mahdollisuus aloittaa pienyritystoiminta.

Android-sovelluksen toiminnallisuus luotiin Java-kielellä käyttäen Android Studio -nimistä kehitysympäristöä.

Työ rajattiin vastaamaan pääosin tuotannollisten yritysten tarpeita, eikä siinä myöskään otettu kantaa virallisiin byrokraattisiin tarpeisiin, kuten kirjanpitoon tai verotukseen liittyviin asioihin.

---

Asiasanat: Android, SQLite, pienyritykset, Java, mobiiliohjelmointi

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Information Technology, Option of Software Development

---

Author: Tomi Kenkkilä  
Title of thesis: Mobile application for start-up company  
Supervisor: Eino Niemi  
Term and year when the thesis was submitted: Autumn 2022  
Number of pages: 29

---

In this thesis the topic was to research IT needs for start-up companies and develop an Android mobile application to meet these needs. Research contained inquiry which was targeted for the target demographic.

The application was mainly focused on the schedule and inventory as well as the way their ways to cooperate in this kind of application. This project was made mainly for hobby use.

Android application was programmed using java language with IDE called Android Studio.

---

Keywords: Android, SQLite, small business, inquiry, Java, mobile programming

# SISÄLLYS

LYHENTEET JA TERMIT .....	6
1 JOHDANTO .....	8
2 YRITYKSEN TARPEET .....	9
2.1 Ajanhallinta .....	9
2.2 Inventaario .....	9
3 SOVELLUKSEN KEHITYSYMPÄRISTÖ .....	10
3.1 Android-ohjelmointi .....	10
3.1.1 Java .....	10
3.1.2 Aktiviteetit .....	11
3.1.3 Käyttöliittymä .....	12
3.2 SQLite .....	13
4 PROJEKTIN TOTEUTUS .....	14
4.1 Tarpeiden kartoitus .....	14
4.1.1 Kyselyt .....	14
4.1.2 Päätelmät .....	16
4.2 Sovelluksen toteutus .....	16
4.2.1 Tietokanta .....	17
4.2.2 Inventaario .....	20
4.2.3 Aikataulu-näkymä .....	22
4.2.4 Toimivuus yleisesti .....	23
5 JATKOKEHITYS .....	27
6 YHTEENVETO .....	28
LÄHTEET .....	29

## LYHENTEET JA TERMIT

Android	Käyttöjärjestelmä, joka on yleinen mobiililaitteissa
Android Studio	Androidin virallinen ohjelmointiympäristö
ContentValues	Androidissa käytetty luokka, jolla voidaan tallentaa tietoa
CustomListAdapter	Muokattu versio adapter-oliosta, jolla voidaan asettaa dataa näkyviin
IDE	Ohjelmointiympäristö, eli ohjelma tai joukko ohjelmia, joilla ohjelmointia toteutetaan.
Intent	Android-ohjelmoinnissa käytetty olio, jolla avataan uusi aktiviteetti
Java	Ohjelmointikieli, jota käytetään Android-ohjelmoinnissa
Kotlin	Ohjelmointikieli, jota käytetään Android-ohjelmoinnissa
Layout	Android-ohjelmoinnissa graafinen asetelma
ListView	Androidissa graafinen tapa listan luontiin
Luokka	Android-ohjelmoinnissa ohjelmiston tiedosto, joka luo oliolle rakennusohjeet
Olio	Ohjelmiston perusyksikkö, joka sisältää tietoa ja toiminnallisuutta
OnCreate	Android-ohjelmoinnissa aktiviteetin luonnissa ensimmäinen funktio, joka suoritetaan
SQLite	Tietokannan hallintajärjestelmä

Widget	Yksinkertainen ja helppokäyttöinen ohjelmistosovellus tai komponentti
XML	Merkintäkielien standardi, joka määrittää tietojen merkintämuodon loogisella rakenteella

# 1 JOHDANTO

Opinnäytetyön tavoitteena oli tutkia aloittavan pienyrityksen IT-tarpeita ja toteuttaa niiden pohjalta Android-mobiilisovellus, jolla voi helpottaa yrityksen toimintaa.

Yrityksen aloittamisen kynnyks on madaltunut viime vuosien aikana huomattavasti ja yhä useampi henkilö on siirtynyt laskuttamaan työstään joko yrityksen tai kevytyrittäjyyden kautta. Jo pelkästään tammi-maaliskuussa vuonna 2022 perustettiin 13 470 uutta yritystä, joista 7 850 olivat luonnollisten henkilöiden perustamia (1). Tilastokeskuksen tilastoista (2) voi myös huomata, että yritysten perustaminen on lisääntynyt viime vuosina ja tässä ei edes näy kevytyrittäjien määrää, joita oli vuonna 2021 yhteensä noin 150 000 (3).

Aiheen keksin itse ajatellen omaa harrastustani ja siitä mahdollisesti yrityksen perustamista varten. Vastaavanlaisia pikkuohjelmia on olemassa yrityksille useita, mutta niiden toiminnallisuus on pääosin kohdistettu jo perustetuille suhteellisen vakaille yrityksille ja lisäksi niiden käytöstä joutuu maksamaan kuukausittain. Sovelluksen pääasiallisena tarkoituksena on kohdistua helpottamaan nimenomaan tuotannollisia yrityksiä, joissa tuotetaan tai myydään fyysisiä tuotteita. Työssä tehdään myös kysely, joka pyritään kohdistamaan sovelluksen mahdolliselle kohdeyleisölle.

Sovelluksen teossa keskitytään pääosin aikataulun ja inventaarion hallintaan sekä niiden yhteiseen toimintaan.



## **2 YRITYKSEN TARPEET**

Aloittavalla yrityksellä on usein monia erilaisia tarpeita riippuen alasta. Tässä työssä keskityttiin pääosin tuotannollisten yritysten tarpeisiin, koska ne olivat selkeimmin toteutettavissa yleiseen käyttöön. Toinen syy tuotannolliseen keskittymiseen oli puhtaasti se, että työn tekijän oma harrastus on rinnastettavissa tuotannolliseen yritystoimintaan. Tällaisilla aloittelevilla yrityksillä saattaa olla useita erilaisia tarpeita, mutta pääimmäisenä tarpeena yleensä on ajan- ja resurssienhallinta.

### **2.1 Ajanhallinta**

Kyselyiden ja alkukartoituksen perusteella ajanhallinta on keskeinen osa yritystoiminnan kirjauksia ja myös sellainen asia, joka usein toteutetaan puhelimella esimerkiksi käyttäen puhelimen omaa kalenteria. Yleensä aloittelevalle yritykselle riittää yksinkertainen kalenterisovellus, mutta niiden toiminta ei ole lähtökohtaisesti suunniteltu tuotannolliseen yrityskäyttöön.

### **2.2 Inventaario**

Usein tuotannollisilla yrityksillä on tarve pitää kirjaa myytävien sekä käytettävien tavaroiden määrästä sekä siitä, mihin niitä on käytetty. Inventaarion pitämisellä huolehditaan myös siitä, että yrityksen tulot ja menot pysyvät selkeästi järjestyksessä. On melko yleistä, että inventaario hoidettaisiin yrityksessä käyttäen paperista kirjanpitoa tai Excel-taulukkoita. Näissä menetelmissä ei suoranaisesti ole mitään vikaa, mutta paperinen kirjanpito voi suuremmilla tietomäärillä olla haastavaa. Excelin tai vastaavan taulukkosovelluksen käyttö sen sijaan on suotavaa, mutta puhelimella niiden nopeasti käyttäminen voi olla haastavaa. Sovelluksessa inventaariossa keskityttiin pääosin yleisesti tuotteiden helppoon tallennukseen ja selkeään listaamiseen.

### 3 SOVELLUKSEN KEHITYSYMPÄRISTÖ

Työssä kehitettiin mobiilisovellus Android-alustalle, jonka ohjelmointiympäristönä käytettiin Android Studiota. Työhön valittiin vähittäisvaatimukseksi Android 10, mikä tarkoittaa sitä, että sovellus ei toimi puhelimilla, joissa on käytössä vanhempi Android-versio. Version valintaan vaikutti sen yleisyys nykyisin käytetyissä puhelimissa sekä sen tarjoamat toiminnallisuudet.

#### 3.1 Android-ohjelmointi

*Android on älypuhelimille, taulutietokoneille ja älytelevisioille suunniteltu ohjelmistopino, joka sisältää käyttöjärjestelmän, väliohjelmistoja ja käyttäjän perusohjelmia. Siinä käytetään avoimen lähdekoodin Linux-käyttöjärjestelmän ydintä. Androidia kehitti alun perin Android Inc., jonka Google myöhemmin osti. Nykyisin sen kehittämisestä vastaa Open Handset Alliance Googlen johdolla. (4.)*

Android on nykyään maailma suosituin käyttöjärjestelmä puhelimissa ja siksi Google kehittääkin sitä jatkuvasti paremmaksi. Tämä on sinällään hyvä asia käyttäjän näkökulmasta uusien ominaisuuksien ja turvapäivitysten myötä, mutta ohjelmoijalle se luo haasteita jatkuvan kehityksen tarpeen takia. Google kuitenkin tarjoaa kattavat dokumentaatiot Androidista sekä useista siihen liittyvistä työkaluista ja ratkaisuista.

##### 3.1.1 Java

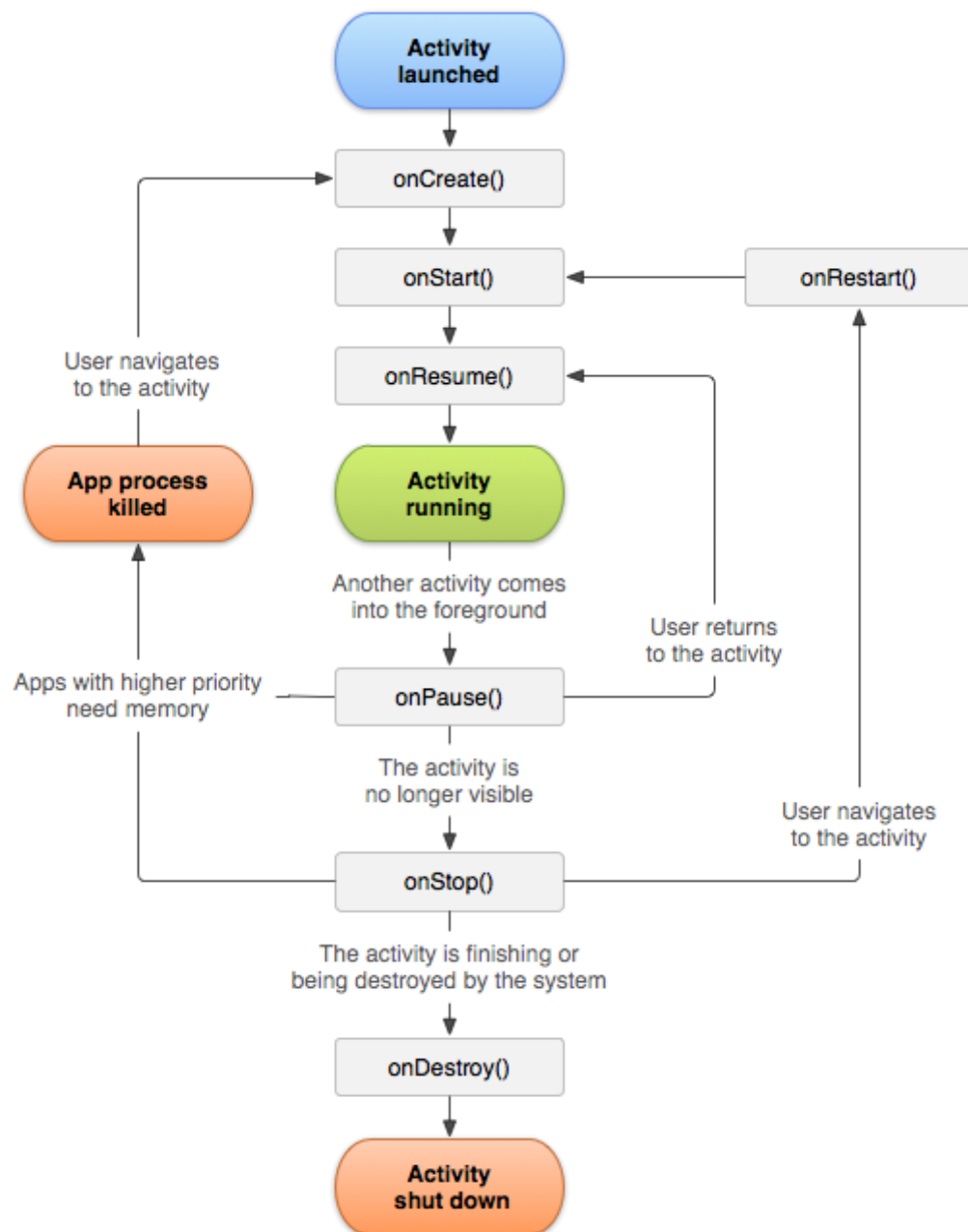
Java on Sun Microsystemsin kehittämä teknologiaperhe ja ohjelmistoalusta, johon kuuluu laitteistoriippumaton oliopohjainen ohjelmointikieli. Java on julkaistu vuonna 1995 ja se on nykyään Oraclen omistuksessa. (5.) Java on pitkään ollut suosituin kieli Android-ohjelmoinnissa ja sitä käytetään edelleen laajalti, vaikka Androidin omistama Google nykyään suosittelee Kotlinia ohjelmointikieleksi.

### 3.1.2 Aktiviteetit

Aktiviteetti on ohjelmassa oleva komponentti, joka toimii pohjana ohjelmassa tapahtuvalle interaktiiviselle toiminnalle. Aktiviteetti voi olla esimerkiksi ohjelman päävalikko. Aktiviteetilla on oma elämänsykli (kuva 1), jonka pohjalta aktiviteetti toimii.

Aktiviteetti-luokka alkaa onCreate()-metodilla, jonka muokkaaminen on pakollista ja jossa määritellään aktiviteetille oma käyttöliittymätiedosto sekä alustetaan aktiviteetissa käsiteltävät tiedot. Muita aktiviteetin metodeita ei ole pakko muokata, mutta niiden avulla voidaan määritellä, miten aktiviteetti käyttäytyy tietyissä tilanteissa.

Aktiviteetin elämänsyklistä (kuva 1) voidaan huomata, että aktiviteetti säilyttää tietonsa, vaikka käyttäjä navigoi uuteen aktiviteettiin. Tämän huomaa esimerkiksi sovellusta käyttäessä, jos käyttäjä siirtyy seuraavaan aktiviteettiin ja sen jälkeen puhelimen takaisin-näppäimellä palaa takaisin aktiviteettiin. Tällaisessa tilanteessa on mahdollista, että sovelluksessa tapahtuvat muutokset eivät ole päivittyneet aktiviteettiin, mikäli sovelluksessa ei ole huomioitu tällaista navigointitapaa. Myös esimerkiksi tietojen tallennuksen tulisi tapahtua aktiviteetissa, ennen kuin aktiviteetti päättyy onDestroy()-metodiin, jolloin aktiviteetissa olevat tiedot tuhoutuvat ja siihen takaisin palatessa suoritetaan onCreate()-metodi uusiksi.



KUVA 1 Aktiviteetin elämänkaari (6)

### 3.1.3 Käyttöliittymä

Android-kehityksessä käytetään XML-tiedostoja graafisten tietojen luontiin. XML-tiedostoon luodaan layout, jonne luodaan graafinen käyttöliittymä aktiviteetille. Asetelmaan määritellään, miten aktiviteetissa näkyvät erilaiset komponentit (painikkeet, tekstit, kuvat, valikot,jne). Tyypillisesti jokaiselle aktiviteetille luodaan oma XML-tiedosto.

Layout-tyylejä on erilaisia ja niiden valintaan vaikuttaa aktiviteetin haluttu toiminta. Android Studioissa on käytössä constraint-layout vakiona, mutta myös aiemmin suuressa suosiossa olleet linear- ja relative-layout ovat edelleen paljon käytettyjä. Linear-layoutissa eli lineaarisessa asettelussa graafiset elementit asetellaan pysty- ja vaakasuunnassa, relative-layoutissa eli relatiivisessa asettelussa elementit asetellaan toisiinsa suhteessa, mikä helpottaa sovelluksen skaalausta. Constraint-layout on periaatteeltaan hyvin samanlainen kuin relatiivinen asettelu, mutta kehitelty versio siitä ja siksi myös suorituskykyisempi. Constraint-layoutin tekeminen on myös helpompaa visuaalisessa editorissa kuin relatiivisen asettelun.

### **3.2 SQLite**

SQLite on kevyt avoimen lähdekoodin relaatiotietokantajärjestelmä, jolla voidaan tallentaa tietoa erilliseen tiedostoon ja josta voidaan hakea tallennettua tietoa erilaisilla kyselyillä (7). Erityisesti mobiiliapplikaatioissa SQLite on suosittu, koska sitä käyttäessä tietojen tallennus tapahtuu eristetyssä tilassa, eikä sovelluksessa tapahtuvien virhetilanteiden pitäisi vaikuttaa tiedon eheyteen. SQLite on myös vakiona käyttövalmiina useimmissa käyttöjärjestelmissä, mukaan lukien Androidissa, jolle sovellus toteutettiin. SQLite on myös käyttövalmiina Androidin mukana, eli sitä ei tarvitse erikseen asentaa ja sen käytöstä löytyy kattavat dokumentaatiot.

## **4 PROJEKTIN TOTEUTUS**

### **4.1 Tarpeiden kartoitus**

Projektin alussa tutkittiin yritysten erilaisia tarpeita, joita voidaan toteuttaa mobiiliapplikaatiossa järkevästi ja samalla karsittiin sellaisia tarpeita pois, joiden toteutukseen tarvittaisiin ulkopuolisia järjestelmiä tai joiden toteutuksessa joutuisi ottamaan huomioon lakitekniset tarpeet. Tarpeiden tutkimisen kautta saatiin myös työn aihe rajattua selkeästi. Tarpeiden kartoituksessa päädyttiin lopputulokseen siitä, että mobiilisovelluksen olisi hyvä sisältää ainakin ajanhallintaan tarvittavat toiminnot sekä jonkinlaisen inventaariototeutuksen.

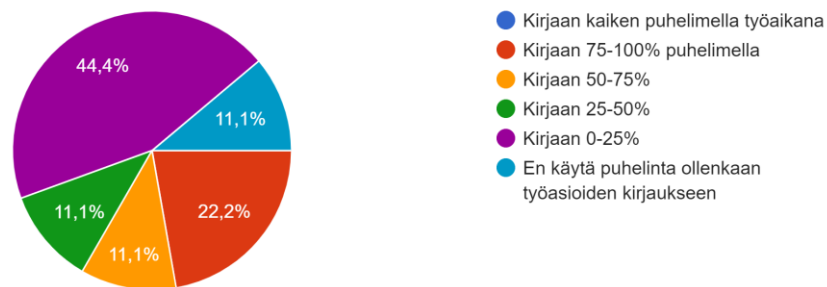
#### **4.1.1 Kyselyt**

Työ piti sisällään myös pienimuotoisen kyselyn muutamille yrityksille, millä pyrittiin järkevöittämään sovelluksen toimintaa ja varmistamaan sovelluksen toiminnan tarpeet. Kyselyn luontiin käytettiin Google Forms -työkalua ja kysely jaettiin muutamille yrityksille, joiden toiminnan koin osuvan sovelluksen kohderyhmään. Kyselyn kohderyhmäksi valittiin tuttuja pienyrittäjiä sekä heidän työntekijöitään. Kyselyyn vastasi yhdeksän henkilöä, jolloin otannan laajuus ei ole koivin kattava, mutta kuitenkin riittävä sovelluksen kehityksen kannalta.

Kyselyn pääteemana oli kysyä työntekijöiltä työtehtävien kirjaukseen liittyviä asioita. Aluksi kyselyssä kartoitettiin sitä, kuinka usein työntekijät ylipäätään käyttävät puhelinta työtehtävien kirjauksiin (kuva 2) ja sen jälkeen sitä minkälaisissa työtehtävissä (kuva 3).

Kuinka usein huomaat tekeväsi työasioiden kirjauksia puhelimellasi?

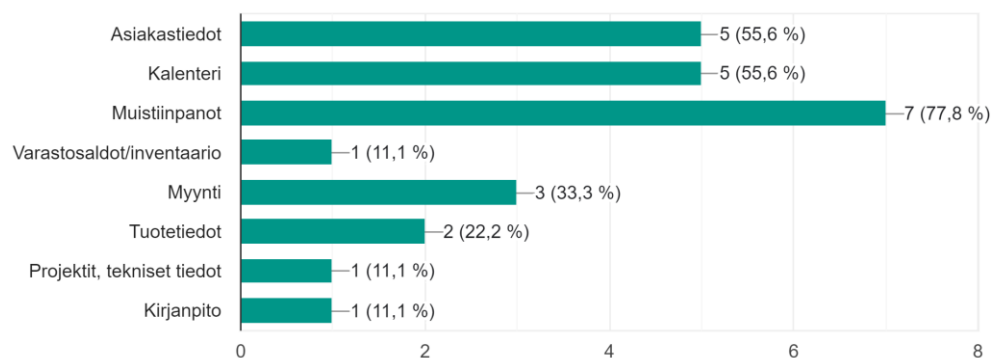
9 vastausta



KUVA 2 Kyselyn tulos työasioiden kirjauksesta puhelimella.

Millaisissa työtehtävissä joudut kirjaamaan töitä ylös? (Puhelimella, tietokoneella, paperille jne.)

9 vastausta

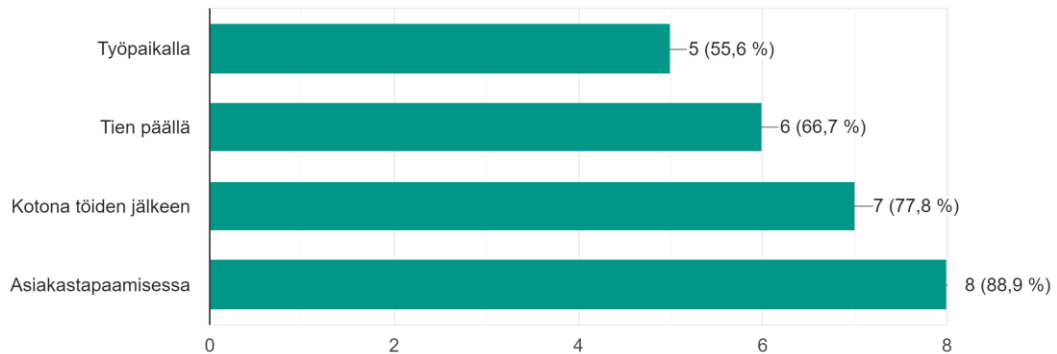


KUVA 3 Kyselyn tulos työtehtävistä, joita kirjataan.

Kyselyssä pyrittiin myös selvittämään, missä paikoissa työntekijät yleensä kirjaavat työtapahtumia (kuva 4), koska puhelimen käytön oleellisuus näkyy selkeämmin tilanteissa, joissa muilla laitteilla kirjaus on vaikeampaa.

Millaisissa tilanteissa kirjaat työtapauksia puhelimella? Valitse ne vaihtoehdot, joissa pääosin.

9 vastausta



KUVA 4 Kyselyn tulos kirjauspaikoista

#### 4.1.2 Päätelmät

Kyselyjen vastaukset tukevat ainakin osittain (kuva 3) tarpeiden kartoituksessa huomattuja tarpeita. Yli puolet vastaajista käyttää kalenteria työssään, mikä ei sinällään ole yllättävää yritystoiminnassa. Inventaarion osalta vastaajien toimiala luonnollisesti vaikutti siihen miten he hallinnoivat tuotteitaan ja siitä syystä myös kyselyyn tuli inventaarion kaltaisia vastauksia erilaisilla nimikkeillä (kuva 3). Toteutetun mobiilisovelluksen kannalta ajateltuna osa vastauksista liittyi epäsuorasti sovelluksessa toteutettuun inventaarioratkaisuun, jonka käyttöä voidaan soveltaa ainakin tuotetietojen ja projektien listaukseen, mikäli erilliselle inventaariolle ei ole tarvetta.

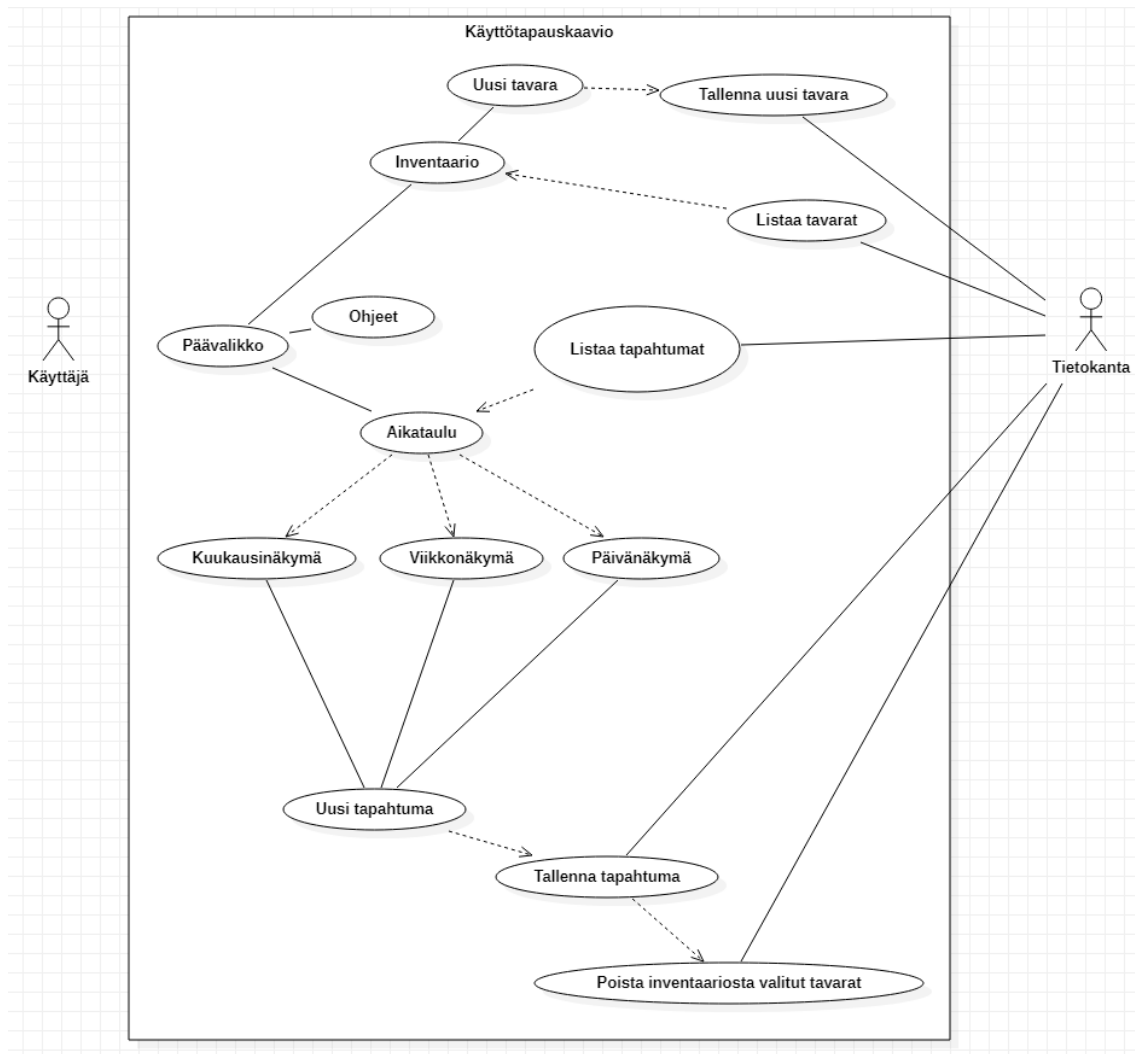
Vastausten hajanaisuuteen saattoi vaikuttaa se, että osa vastanneista työskentelee samassa yrityksessä, jonka takia työtehtävien erilaisuus vaikuttaa vahvasti vastauksiin.

#### 4.2 Sovelluksen toteutus

Sovelluksen toteutus alkoi inventaarion toimintojen luomisella ja ideoinnilla sovelluksen käyttötarkoituksesta. Aluksi sovelluksen oli tarkoitus olla yleishyödyllinen työkalu aloittelevan yrityksen käyttöön, mutta työn edetessä se muuttui pääosin keskittymään tuotannollisen yrityksen tarpeisiin. Tämä muutos näkyy konkreettisesti siten, että sovelluksen käyttäjällä täytyy olla jokin fyysinen tuote ja fyysiset kuluvat välineet, joiden inventaariosta pidetään kirjaa.



Sovelluksen toteutuksen aikana tuli tarpeelliseksi tehdä käyttötapauskaavio (kuva 5) helpottamaan sovelluksen vaadittujen toimintojen dokumentointia ja selkeyttämään sovelluksen kehitystä.



KUVA 5. Sovelluksen käyttötapauskaavio

Käyttötapauskaaviosta näkee selkeästi sovelluksen toiminnan, vaaditut toiminnot ja toimintojen hierarkian.

#### 4.2.1 Tietokanta

Sovellukseen toteutettiin tietokanta tiedon tallennukseen. Tietokantana käytettiin SQLiteä, jonka kirjasto on valmiiksi Androidissa. Tietokantaan tallennettiin inventaarion, kalenterin ja inventaarion varausten tiedot. Tietokanta luodaan erillisessä DBHandler-luokassa, joka pitää sisällään tarvittavat toiminnot tietokannan hallintaan.

Koodiesimerkissä 1 luodaan tietokantaan tarvittavat taulut onCreate-metodissa, joihin käyttäjä pystyy tallentamaan tietoa ja joista tietoa pystytään hakemaan erilaisilla DBHandler-luokkaan tehdyillä funktioilla.

```
// Luodaan tietokantaan eri taulut
@Override
public void onCreate(SQLiteDatabase DB) {
    DB.execSQL("create Table tuotteentiedot(name TEXT primary key,
    paikka TEXT, saldo TEXT)");
    DB.execSQL("create Table aikataulu(id INTEGER PRIMARY KEY AUTOIN-
    CREMENT, tehtava TEXT, epochdays LONG, epochseconds INTEGER)");
    DB.execSQL("create Table varatutinventaariotuotteet(name TEXT,
    paikka TEXT, saldo TEXT, tehtavaId INTEGER)");
}
```

*KOODI 1. DBHandler-luokan koodi, joka luo tarvittavat taulut tietokantaan.*

Koodiesimerkissä 2 on funktio, jolla lisätään tuotteita inventaarion tauluun. Ensimmäiseksi asetetaan käsiteltäväksi tietokannaksi olemassa oleva tietokanta, jonka jälkeen luodaan ContentValues-olio, johon tallennetaan halutut tiedot. Lopuksi käytetään SQLiten insert-metodia, jolla siirretään ContentValues-olion tiedot haluttuun tauluun, tässä tapauksessa tuotteentiedot-nimiseen tauluun, joka on inventaarion taulun nimi tietokannassa. Insert-metodia käytetään siten, että siihen ensimmäiseksi tiedoksi annetaan halutun taulun nimi. Toinen tieto, joka on tässä tapauksessa "null", kertoo funktiolle, että ContentValues-olion tietojen ollessa tyhjät, tauluun ei lisätä mitään. Viimeinen eli kolmas tieto on itse ContentValues-olio, joka sisältää aiemmin määritellyt tiedot, jotka halutaan tauluun. Koodiesimerkin 2 metodin käytön jälkeen saadaan luotua taulu inventaarion tiedoille (kuva 6).

```
//Lisätään inventaarioon tuote
public Boolean insertTuotteentiedot(String name, String paikka, String
saldo) {
    SQLiteDatabase DB = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put("name", name);
    contentValues.put("paikka", paikka);
    contentValues.put("saldo", saldo);
    long result = DB.insert("tuotteentiedot", null, contentValues);
    if (result == -1) {
        return false;
    } else {
        return true;
    }
}
```

*KOODI 2. DBHandler-luokan koodi, jolla inventaarion tauluun tallennetaan uudet tuotteet.*

name	paikka	saldo
Filter	Filter	Filter
Skalpeli	Hylly 3	170
Petrimalja	Hylly 5	110
Ruisku	Hylly 3	75
Vinyylihanskat	Taso 2	28
magneettisauva	Hylly 4	3
Lasipurkki 300ml	Taso 0	25
Lasipurkki 500ml	Taso 0	10
Injektioneula	Hylly 3	200
Kookoskuitu	Taso 0	2
Pelletti	Varasto	25
Kukkaruukku	Hylly 3	80
Laatikko 50x50x100	Hylly 3	456
Laatikko 40x40x70	Hylly 3	300
Teippirulla	Hylly 4	10

KUVA 6. Inventaarion tiedot tietokannassa

SQLite-tietokannalla ei ole virallista tapaa päiväyksien tallentamiseen (8), minkä vuoksi päädyin käyttämään Unix-aikaa, eli toisin sanoen Epoch-aikaa päiväyksien tallentamiseen tauluun. Epoch-ajalla tarkoitetaan ajan tallennustapaa, joka ilmaisee ajan sekunteina ajanhetkestä 1.1.1970 kello 0.00.00 UTC. Kalenteritapahtumien taulusta (kuva 7) voi huomata, että tässä tapauksessa Epoch-ajan tallennus on jaettu kahteen eri tietoon. Ensimmäinen eli epochdays kertoo tallennetun päivän ja toinen eli epochseconds tallennetun kellonajan tuolle päivälle. Sovelluksessa on päädytty tähän ratkaisuun, koska useissa sovelluksen metodeissa tarvitaan vain toista tietoa, jolloin sitä ei tarvitse jokaisessa kohdassa erikseen eritellä.

id	tehtava	epochdays	epochseconds
Filter	Filter	Filter	Filter
1		19260	25581
2	20 Ruukun siirto	19261	37853
3	testi	19261	70406
4	Laatikoiden myynti	19275	51489
5	Laatikoiden myynti	19275	51511

KUVA 7. Kalenteritapahtumien tiedot tietokannassa

Varattujen tuotteiden tapahtumatiedot löytyvät tehtavaId:n (kuva 8) avulla, joka sitoo taulujen tiedot toisiinsa.

name	paikka	saldo	tehtavaId
Filter	Filter	Filter	Filter
Ruukku	Hylly 3	20	2
Laatikko	varasto	20	4
Laatikko	varasto	20	5

KUVA 8. Varattujen tuotteiden tiedot tietokannassa

#### 4.2.2 Inventaario

Sovelluksessa inventaarion ideana on helpottaa käyttäjän järjestelmällisyyttä ja antaa käyttäjälle mahdollisuus lisätä uusia tavaroita inventaarioon sekä myös seurata niiden kulutusta. Sovelluksen inventaario ei sisällä tietoa tuotteiden arvosta, eli sitä ei voi suoraan sellaisenaan käyttää viralliseen kirjanpitoon.

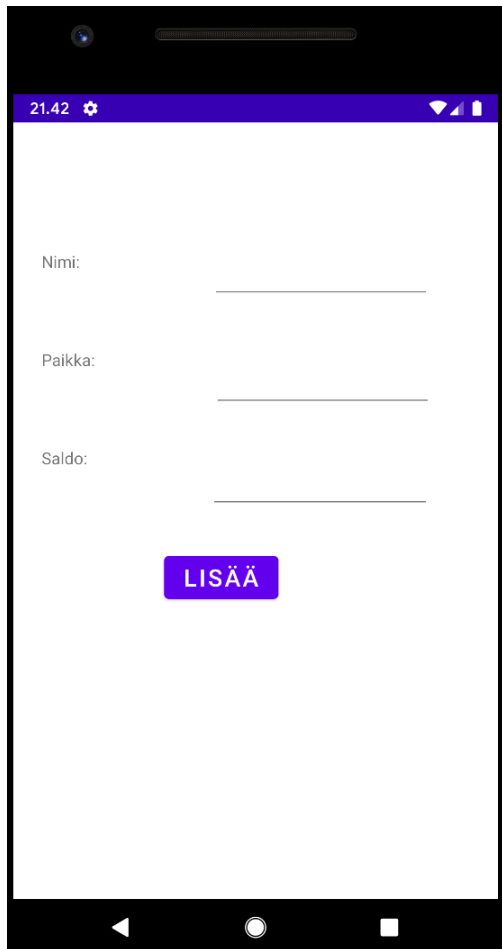
Inventaario-näkymässä tietokannasta haetaan DBHandler-luokan avulla inventaarion tavarat CustomAdapterilla toteutettuun ListViewiin, joka muodostaa listan (kuva 9) kaikista tallennetuista tiedoista näkyviin. Inventaario-aktiviteetissa (kuva 9) voi myös avata AddItemToInventory-aktiviteetin (kuva 10) painamalla kuvassa (kuva 9) näkyvää "Lisää Tuote"-näppäintä, jossa käyttäjä voi lisätä uuden tavarat tietokantaan. Tavaraa tietokantaan lisätessä sille annetaan tiedoiksi nimi, paikka ja saldo ja tiedot lisätään tietokantaan käyttäen insert-metodia (koodi 2). Tavarat nimi on tietokannassa määritelty uniikiksi avaimeksi, eli muita samannimisiä tuotteita ei voi lisätä

tietokantaan. Inventaarionäkymässä (kuva 9) tuotelistauksesta voidaan myös poistaa tuotteita painamalla niitä pitkään pohjassa, jolloin sovellus vielä varmistaa käyttäjältä, halutaanko tuote varmasti poistaa. Kun tuote poistetaan, se katoaa listasta sekä myös tietokannasta. Tuotteiden listanäkymän järjestystä voidaan myös muuttaa klikkailemalla tuotelistauksen yläpuolella näkyviä tekstikenttiä, jolloin tuotelistaus järjestyy aakkos- tai määräjärjestyksessä halutun tiedon mukaan.



Paikka	Nimi	Saldo
Hylly 1	Skalpeli	100
Hylly 3	Lasipurkki 500ml	25
Varasto	Laatikko 50x50	30
Varasto	Laatikko 30x30	30
Hylly 2	Pakkausteippi	2

KUVA 9 Inventaarion päänäköymä



KUVA 10 Inventaarioon uuden tavaran lisäys -näkymä

#### 4.2.3 Aikataulu-näkymä

Sovelluksen aikataulu-näkymä toteutettiin normaalin kalenterin tavoin, eli aikataulu-näkymässä ensimmäisenä näkyy kuukausinäkymä, josta saa auki viikko- tai päivänäkymän ja näiden näkymien välillä voi liikkua sulavasti. Kuukausi- ja viikkonäkymässä myös erikseen listautuu valitun päivän kooste tapahtumista ja näkymien kautta pystyy lisäämään uusia tapahtumia halutuille päiville.

Tapahtumaa tehdessä (kuva 13) käyttäjä voi myös valita pudotusvalikosta tapahtumaan liittyvät työvälineet, joita tapahtumassa halutaan käyttää tai jostain muusta syystä halutaan tapahtuman yhteydessä poistaa inventaariosta. Nämä työvälineet vähentyvät suoraan inventaarion tiedoista ja ne lisätään varattujen välineiden tietoihin.

Kalenterin aktiviteeteissa tietokannasta haetaan tehtyjen tapahtumien nimet, päivämäärät ja kellonajat, jotta saadaan kalenterin tapahtumat näytettyä käyttäjälle. Tapahtumien

päivämäärätiedot on tallennettu tietokantaan epochpäivissä ja epochsekunneissa, eli vaikka käyttäjä matkustaisi toiselle aikavyöhykkeelle, saadaan tietokannasta tarkka tieto tapahtuman ajankohdasta.

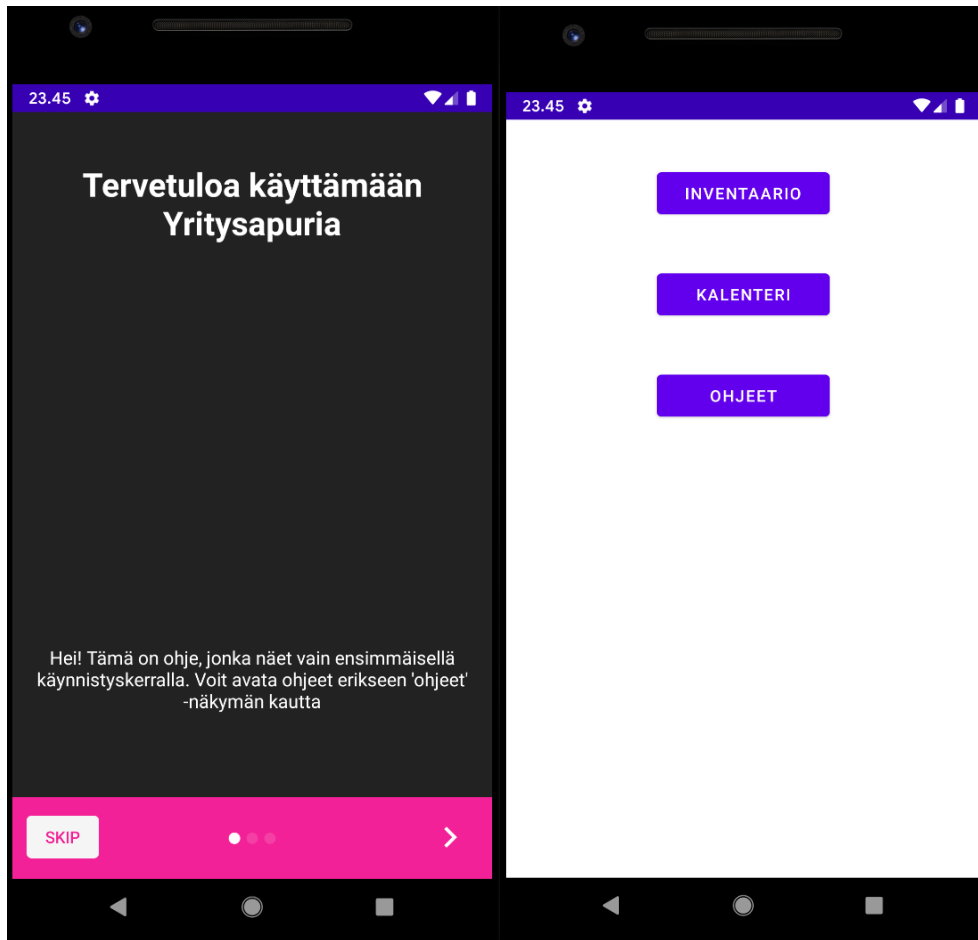
Kalenterin aktiviteettien kautta avataan EventEdit-aktiviteetti (kuva 13) painamalla painiketta "Uusi tapahtuma" (kuva 12), jolla voidaan luoda uusia tapahtumia. Tapahtumaa tehdessä käytetään DBHandler-luokan metodeita, joilla haetaan aluksi onCreate-metodissa tietokannasta kaikkien inventaarion tavaroiden tiedot ja näytetään käyttäjälle niiden nimet listana. Kun listasta valitsee tavarat, näyttää aktiviteetti valitun tavarat saldon. Tapahtumassa valituille tavaroille valitaan haluttu määrä ja sen jälkeen tapahtumaa tallentaessa DBHandler-luokan metodeilla lisätään valitut tuotteet ja halutut määrät varattuihin tavaroihin ja poistetaan sama haluttu määrä inventaariosta.

#### 4.2.4 Toimivuus yleisesti

Ensimmäisen käynnistykseen yhteydessä sovelluksessa näkyvät aluksi ohjeet sovelluksen toiminnoista ja yleisesti sovelluksen käytöstä. Seuraavilla käynnistyskerroilla ohjeet saa näkyviin vielä erikseen ohjeet-näkymän kautta. Ohjeiden toiminta toteutettiin käyttäen AppIntro-kirjastoa (9), jolla pystyttiin luomaan selkeästi toimiva diaesitys ohjelman toiminnasta (kuva 11). Tämän jälkeen sovelluksessa aukeaa päävalikko (kuva 11), joka pitää sisällään painikkeet inventaarion, aikataulun ja ohjeiden näkymien avaukseen. Inventaario-painiketta painaessa päävalikossa käytetään Intent-luokan oliota (koodi 3), jolla siirrytään seuraavaan näkymään laittaen intent-oliolle arvoiksi nykyinen näkymä, josta siirrytään, sekä näkymä johon siirrytään.

```
Intent intent = new Intent(MainActivity.this, Inventaario.class);
MainActivity.this.startActivity(intent);
```

*Koodi 3. Intent-olio, jolla avataan inventaario-aktiviteetti.*



KUVA 11 Vasemmalla Ohjeet-näkymä ja oikealla Päävalikko-näkymä

Inventaario-näkymä sisältää painikkeen uuden tuotteen luomiselle sekä listan, joka näyttää inventaarion tuotteet. Painike uuden tuotteen luomiselle avaa uuden näkymän samalla tavalla kuin koodiesimerkissä 3. Uutta tuotetta luodessa käyttäjä antaa tuotteelle nimen, paikan ja kappalemäärän ja tallentamisen jälkeen tuote näkyy inventaario-näkymän tuotelistauksessa.

Päävalikosta aikataulu-painiketta painettaessa siirrytään aikataulu-näkymään intent-oliolla. Sovelluksessa on kolme erilaista aikataulu-näkymää, joista kuukauden aikataulu-näkymä on ensimmäinen, jonka saa auki. Kuukausi-näkymä (kuva 12) näyttää kalenterin tavoin kuukauden päivät ja näyttää oletusarvoisesti sen hetkisen päivän tapahtumat. Käyttäjä voi myös päivämääriä painelemalla tarkistaa kyseisten päivien tapahtumia ja luoda tapahtumia niille. Kuukausi-näkymästä voi myös halutessaan siirtyä viikko- tai päivä-näkymään. Viikko-näkymä on hyvin samanlainen kuin kuukausi-näkymä, sillä erotuksella, että se näyttää vain yhden viikon kerrallaan. Päivä-näkymässä kellonajat listautuvat 0–24-periaatteella ja tapahtumat näkyvät listauksessa tuntien kohdalla.





KUVA 12 Aikataulun kuukausi-näkymä



KUVA 13 Uuden tapahtuman teko

Aikataulu-näkymässä on myös uusi tapahtuma -painike, joka avaa näkymän uuden tapahtuman luontiin. Uutta tapahtumaa luodessa (kuva 13) käyttäjä asettaa tapahtumalle nimen sekä kellonajan. Jos tapahtumalle ei aseteta kellonaikaa, tulkitsee sovellus asetetuksi kellonajaksi sen hetkisen kellonajan. Uutta tapahtumaa luodessa voi tapahtumaan myös lisätä inventaarion tuotteita, jotka tallentuvat tietokantaan varattuihin tuotteisiin ja joiden määrä vähenee inventaarion tiedoista. Uuden tapahtuman tallennuttua sovellus siirtyy takaisin aikataulun siihen näkymään, josta uuden tapahtuman -näkyä alunperin avattiin.

## 5 JATKOKEHITYS

Sovelluksen jatkokehityksessä sovellukseen voisi tehdä toimintoja, jotka mahdollistaisivat sovelluksen virallisen käytön kirjanpidossa sekä muissa virallisissa yrityksen toimintaan liittyvissä asioissa. Myös erilaisten laskureiden lisääminen voisi helpottaa tietojen kirjaamisessa samalla ja lisäisi inventaarioon ylläpitoon enemmän toiminnallisuutta. Sovellukseen oli tarkoitus alunperin myös toteuttaa tuotteille omat tapahtumahistoriat ja näihin tehtiin tietokantaan liittyvät toiminnotkin valmiiksi, kuten varattujen inventaariotuotteiden tiedoista näkyy (kuva 8).

Nykyisellään sovellukseen tallennetut tiedot tallentuvat laitteen sisäiseen muistiin ja niiden käsittely on hieman hankalaa muissa ympäristöissä, kuten esimerkiksi tietokoneella. Tätä voisi kehittää siten, että tietokannan tiedot pystyttäisiin helposti siirtämään eri muodossa muihin järjestelmiin, kuten yrityksen käyttämiin pilvipalveluihin tai kiinteästi erillisenä helppolukuisena tiedostona tietokoneelle. Tietojen tallennuksen voisi myös toteuttaa siten, että samaa tietoa pystyisi muokkaamaan usealta laitteelta samaan aikaan.

Jatkokehityksessä olisi tarpeellista myös tutkia laajemmin yritysten tarpeita, esimerkiksi tekemällä laajemman kyselyn tai olemalla tiiviisti yhteistyössä jonkin yrityksen kanssa. Toisena vaihtoehtona olisi muokata kyseistä sovellusta siten, että sen saisi vastaamaan tietyn yrityksen omia tarpeita.

## 6 YHTEENVETO

Opinnäytetyön tekeminen osoittautui odotettua vaikeammaksi. Työn alkuperäisenä tavoitteena oli toteuttaa kattava mobiilisovellus yrityskäyttöön sisältäen useita erilaisia toimintoja liittyen yritysten alkuvaikeuksiin. Lopulta kuitenkin sovellus jäi erittäin suppeaksi verrattuna alkuperäisen suunnitelman laajuuteen. Sovelluksen oli tarkoitus sisältää mahdollisuus tietokannan tietojen varmuuskopiointiin sekä useita erilaisia laskureita, joilla voitaisiin helpottaa inventaarion ylläpitoa. Suunnitelmissa oli myös tehdä sovelluksesta visuaalisesti näyttävämpi, mutta loppujen lopuksi päädyin tekemään siitä mahdollisimman yksinkertaisen näköisen aikatauluongelmien vuoksi.

Alun perin sovelluksen oli myös tarkoitus kohdistua kaikenlaisille yrityksille aputyökaluna, mutta projektin edetessä tuli huomattua, että suurin osa toiminnoista sopii paremmin nimenomaan tuotannollisen yrityksen käyttöön. Tästä syystä projektin tavoitetta muutettiin siten että se kohdistui pelkästään tuotannollisiin yrityksiin ja siksi projektin laajuus lyheni samalla.

Työn edetessä hain jatkuvasti tietoa erilaisista ratkaisuihin ja tekniikoista työn tarpeisiin liittyen ja muokkasinkin sovellusta havaittuani uusia ratkaisuja. Tästä syystä kuitenkin itse raportin tekeminen hidastui ja sen sisällön jatkuva muokkaaminen sovelluksen kehityksen mukana turhautti. Näiden vaikeuksien takia myös aikataulut työn valmistumisessa venyivät ja lopullinen työ ei vastannut odotettua. Työn tekoa kuitenkin helpotti, kun ohjaava opettaja jaksoi kärsivällisesti ja aktiivisesti järjestää katselmointeja ja lisäksi motivoivalla puheellaan edisti mielenkiintoani työn loppuun viemisessä.

## LÄHTEET

1. Tilastokeskus 2022. tiedote uusista yrityksistä tammi-maaliskuussa 2022. Hakupäivä 31.10.2022. <https://stat.fi/julkaisu/cku2b5ujs82g70c52dxelbyji>
2. Tilastokeskus 2022. Aloittaneet ja lopettaneet yritykset. Hakupäivä 11.9.2022 <https://stat.fi/tilasto/aly>
3. Tilastokeskus 2021. Kevytyrittäjien määrä kasvoi koronavuonna. Hakupäivä 31.10.2022. <https://www.stat.fi/tietotrendit/artikkelit/2021/kevytyrittajien-maara-kasvoi-koronavuonna/>
4. Wikipedia 2022. Android. Hakupäivä 24.9.2022 <https://fi.wikipedia.org/wiki/Android>
5. Wikipedia 2022. Java. Hakupäivä 18.09.2022. <https://fi.wikipedia.org/wiki/Java>
6. Android Developer. Activity Lifecycle. Hakupäivä 19.12.2022. <https://developer.android.com/guide/components/activities/activity-lifecycle>
7. Viikonvalo 2015. SQLite. Hakupäivä 18.09.2022. <https://www.viikonvalo.fi/SQLite/>
8. Tableplus 2018. SQLite how to use datetime value. Hakupäivä 16.10.2022. <https://tableplus.com/blog/2018/07/SQLite-how-to-use-datetime-value.html>
9. Github 2022. AppIntro. Hakupäivä 21.11.2022 <https://github.com/AppIntro/AppIntro>