



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Mikael Meinander

Tehtävälistan tyylivaihtoehdot työnoh- jauksen mobiilisovelluksessa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

27.4.2020

Tekijä Otsikko	Mikael Meinander Tehtävälistan tyylivaihtoehdot työnohjauksen mobiilisovelluksessa
Sivumäärä Aika	33 sivua 27.4.2020
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Mobile Solutions
Ohjaaja	Yliopettaja Hannu Markkanen
<p>Insinööriyön tarkoituksena oli suunnitella ja kehittää Android-sovellukseen työnohjaukselle uudenlainen tehtävälista hyödyntäen uusimpia tyyliohjeita. Työssä pyrittiin parantamaan tehtävälistan toiminnallisuutta, koska siitä puuttui tarpeellisia ominaisuuksia. Sen ulkonäköä ja toiminnallisuutta pyrittiin parantamaan ja tekemään siitä uusimpien käytäntöjen mukainen. Tiedon määrää ja ymmärrettävyyttä pyrittiin lisäämään luomalla uusia näkymiä ja lisäämällä toimintoja.</p> <p>Uudet, tarvittavat ominaisuudet olivat etukäteen tiedossa, mutta lähestymistyyli ja toteutus-tapa olivat itse päätettävissä. Sovellus on tehty Android Studiolla, joten päädyttiin käyttämään Googlen luomaa Material Design -viitekehystä. Tarkoituksena oli hyödyntää tyyllisesti ja koodillisesti uusimpia Material Designin tukemia komponentteja. Työn toteutuksessa hyödynnettiin Design Thinking -tyyppistä lähestymismallia.</p> <p>Lopputuloksena syntyi toimiva tehtävälista, joka hyödynsi muun muassa kelluvaa toimintapainiketta ja korttinäkymää. Uudet komponentit ja tyyliuntu antavat käyttäjälle paremman käyttökokemuksen animaatioilla ja päivitetyllä ulkonäöllä. Kuvakkeita ja komponentteja päivitettiin sopimaan Material Design -malliin. Tehtävälistalle lisättiin uutena ominaisuutena lisätietojen tallentaminen tehtävässä käytetyistä tarvikkeista. Tehtävän tietojen tarkasteleminen ja muuttaminen helpottui parantamalla navigaatiota ja myös uusien näkymien ansiosta.</p> <p>Insinööriyössä luodut sovelluksen komponentit ovat hyödynnettävissä sovelluksen muillakin osa-alueilla kuin tehtävälistalla, joten yleiskehityksenä nämä asiat toivat lisäarvoa yritykselle. Päivitetyt komponentit ja tyyli pitävät sovelluksen uudenveroisena, jolloin pysytään kehityksen aallonharjalla. Uusien aktiviteettien näkymät tekivät sovelluksesta entistä ymmärrettävämmän ja helpommin lähestyttävän, kun käyttäjälle näytettävä tiedon määrä kasvoi ja selkeytyi.</p>	
Avainsanat	Android, tehtävälista, Material Design, työnohjaus, Java

Author Title	Mikael Meinander Design options of task list to supervision of work in a mobile application
Number of Pages Date	33 pages 27 April 2020
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Professional Major	Mobile Solutions
Instructors	Hannu Markkanen, Principal Lecturer
<p>The purpose of this final year project was to develop and implement a new work list to an Android application. The goal was to make the application modern and more appealing. New features required designing and creating new activities to show more information on the screen without forgetting to improve user experience.</p> <p>The methods used in this final year project utilized the newest styles and components that Google offers in its Material Design guidelines. The new features were predefined, which helped designing and executing them. The application is made with Android Studio and the workflow was carried out with Design Thinking.</p> <p>The outcome was a mobile work task list, that has new features, such as floating action button and cards implemented. These new components improved user experience and enabled new features in the application. The application now uses more animations. Its icons were updated to ones that Material Design recommends.</p> <p>These new components themselves are a huge asset to the company. Those components make the application more modern and are ready to be used also elsewhere in the application. In addition, they follow the newest Material Design guidelines. In conclusion, the new task list was able to show not only more information but also to show it in a distinctly clearer way than before.</p>	
Keywords	Android, work list, Material Design, supervision of work, Java

Sisällys

Lyhenteet

1	Johdanto	1
2	Tyylivaihtoehdot	2
2.1	Material Design -tyylikieli	3
2.2	Flat Design -tyylikieli	4
2.3	Flat Remix -tyylikieli	7
2.4	Tyylin valinta	8
2.5	Listat	9
2.6	Materiaalikorttinäkymä	9
2.7	Kelluva toimintapainike	11
3	Uuden tehtävälistan edut ja toteuttaminen	12
3.1	Prototyyppi	12
3.2	Listat	16
3.3	Kuvakkeet	17
3.4	Materiaalikorttinäkymä	22
3.5	Kelluva toimintapainike	24
3.6	Yläpalkin kuvakkeet	25
3.7	Kirjauslomake	26
4	Tehtävälistan toteutus REST-rajapinnalla	28
4.1	Tietokanta	28
4.2	Palvelin	28
5	Uuden ja vanhan tehtävälistan vertailu	29
5.1	Tehtävälista	29
5.2	Tyyli	30
5.3	Uudet ominaisuudet	30
6	Yhteenveto	31

Lyhenteet

DP	Density-independent Pixels. Tiheysvapaa pikseli, jonka abstrakti koko perustuu näytön tiheyteen. Voidaan kutsua myös DIP:ksi.
DIP	Density-independent Pixels. Tiheysvapaa pikseli, jonka abstrakti koko perustuu näytön tiheyteen. Voidaan kutsua myös vapaammin DP:ksi.
FAB	Floating Action Button. Kelluva toimintapainike suorittaa yleisimmän tapahtuman näytöllä. Painike tuntuu kelluvan, koska se ei reagoi muihin näytön komponentteihin.
GML	Geography Markup Language. XML-kieliopin merkintäkieli, jolla esitetään maantieteellisiä kohteita.
HDPI	High density Pixels per Inch. Noin 213–240 kuvapistettä tuumassa. Laitteen pikselitiheys.
REST	Representational State Transfer. Arkkitehtuurimalli ohjelmointirajapinnoissa. Perustuu HTTP-protokollaan.
WFS	Web Feature Service. Määrittelee rajapinnan paikkatietokohteiden lukemiselle, luomiselle ja muokkaamiselle alustariippumattomasti.
XHDPI	eXtra High density Pixels per Inch. Noin 240–320 kuvapistettä tuumassa. Laitteen kuvapistetiheys.
XML	Extensible Markup Language. Tiedonsiirron rakenteellinen merkintäkieli.
XXHDPI	eXtra eXtra High density Pixels per Inch. Noin 320–480 kuvapistettä tuumassa. Laitteen kuvapistetiheys.
XXXHDPI	eXtra eXtra eXtra High density Pixels per Inch. Noin 480–640 kuvapistettä tuumassa. Laitteen kuvapistetiheys.

1 Johdanto

Insinööriyön tarkoituksena on tutkia, suunnitella ja toteuttaa vaihtoehtoja työnohjauksessa käytettävän tehtävälistan uudistamiseksi tuotteistetuksi työkalupakiksi Android-sovellukselle. Työssä uudelleenkehitetään tehtävälistaa mobiililaitteille, tarkkaan ottaen Android-laitteille, jo olemassa olevien ja tiedettyjen kenttätyön tarpeiden pohjalta, joita vanhassa tehtävälistassa hyödynnetään. Työssä isoon osaan nousee tehtävälistan suunnittelu visuaaliseksi ja toimivaksi kokonaisuudeksi. Uutta tehtävälistaa verrataan vanhaan tehtävälistaan ja tutkitaan uusien ominaisuuksien parasta toteuttamiskeinoa uudella toimivalla tyyllillä.

Insinööriyö tehdään suomalaiselle paikkatietoon ja liikkuvaan työhön erikoistuneelle Geometrix Oy:lle, joka tarjoaa sovellustaan asiakkailleen, jotka toimivat muun muassa kenttätöiden parissa [1]. Yrityksen ydinosaamista on karttojen ja datan yhdistäminen ja tuominen asiakkaan hyötykäyttöön [2]. Geometrix Oy on perustettu vuonna 2002, ja sen omistajana sekä toimitusjohtajana toimii Olli Alanko. Yritys sijaitsee kirjoitushetkellä Helsingissä, Pitäjänmäessä. Henkilöstöä yrityksellä on 12. [3.]

Kenttätöissä tarvitaan helppokäyttöistä sovellusta, jossa on laajat toimintamahdollisuudet valvoa, suorittaa ja dokumentoida työtehtäviä ja niihin liittyviä asioita, kuten työsuoritteita. Tässä insinööriyössä suunnitellaan uusi ja moderni ulkonäkö tehtävälistalle, joka koostuu työsuoritteista, sekä toteutetaan tarvittavat vanhat toiminnallisuudet vanhasta tehtävälistasta tähän uuteen tehtävälistaan. Uusia ominaisuuksia on tarkoitus lisätä mahdollisuuksien ja tarpeiden mukaan. Niitä on tiedossa muutamia, kuten tehtävälle uudet materiaali-, kone- ja tyyppikirjaukset. Näille ominaisuuksille täytyy kehittää käyttöliittymä, palvelinpuolen logiikka ja tietokantatuki. Vanhat käytettävyyden kannalta tärkeät ominaisuudet on säilytettävä. Tällaisia ovat esimerkiksi mahdollisuus käyttää Android-laitetta hansikkaat kädessä. Tämä tarkoittaa tärkeiden painikkeiden entistä parempaa esille tuomista, jotta ne näkyvät selvästi näytöllä ja niiden käyttäminen on mahdollisimman helppoa ja nopeaa. Kenttätöissä sovelluksen käyttämisen tulee olla suoraviivaista, ja sen on tarkoitus helpottaa ja nopeuttaa tiedonkeruuta.

Uudelleensuunnittelun ja tekemisen perimmäinen syy on sovelluksen työnohjauksen vanhanaikainen ulkonäkö, jossa kaikkia ominaisuuksia ei pystytä tuomaan tarpeeksi

hyvin esille eikä tilaa uusille tarvittaville ominaisuuksille ole siellä, mihin ne haluttaisiin. Sovellus on luotu 2010-luvun alkupuolella, mutta sitä kehitetään edelleen. Yleiseen Android-kehitykseen ja suunnitteluun liittyvät Googlen Material Design -ohjeet ovat muuttuneet paljon niistä ajoista [4]. Tämän huomaa esimerkiksi, jos vertaa, minkälaisia matkapuhelimia oli kymmenen vuotta sitten ja minkälaisia on nyt. Esimerkkinä käsiteltävän sovelluksen vanhanaikaisesta ulkonäöstä on työnohjauksen osion lista, jonka kaikkia alkioita ei näe. Jotta listaa osaa selata oikein, on tiedettävä, että se jatkuu alemmas. Myöskään uusien ominaisuuksien lisäämiselle ei ole järkevää paikkaa, eikä logiikka toimi tämän hetken toteutuksen kanssa hyvin yhteen. Insinööriyön tarkoituksena on tehdä niin sanottu kasvojenkohotus sovellukselle, jotta käyttäjät pitäisivät entistä enemmän käyttökokemuksesta ja jotta uusien käyttäjien saaminen uudella, modernilla ulkoasulla onnistuisi vielä helpommin ja saadaan tuki uusille ominaisuuksille uusilla asiakkaila.

Insinööriyössä ei käsitellä sovelluksen muita toimintoja eikä asiakkaita, vaan insinööriyön rajauksena on uuden tehtävälistan integrointi ja parantaminen vanhan tilalle. Tämä sisältää REST-rajapinnan ja tietokannan muutokset. Pääasiallinen insinööriyö koostuu siis työnohjauksen uudelleen tehtävälistan toteuttamisen työvaiheista sekä käytännön vertailusta ja toteuttamisesta uuden ja vanhan tehtävälistan välillä. Tämä painottuu käyttöliittymäpuoleen, mutta läpi käydään myös hieman palvelin- ja tietokantapuolta tarvittavilta osin, jotta työn kokonaiskuva tulee paremmin esille.

2 Tyylivaihtoehdot

Geometrix Oy:n kehittämä sovellus, johon tehtävälista toteutetaan, on nimeltään Mobilenote. Se on toteutettu Android-puhelimille käyttäen Javaa ja Kotlinia, jotka ovat keskenään yhteensopivia kehityskieliä [5]. Tämä rajaa tehtävälistan tyylin suunnittelua, eikä muiden kielten tyyliasuuntauksia tarvitse tutkia. Androidia voi kehittää Javan ja Kotlinin lisäksi esimerkiksi React Nativella [6], joka on JavaScript-pohjainen ja näin ollen tukee eri kirjastoja ja eri tyyliasuuntia kuin Mobilenotessa käytettävät kielet. Myös tyyli suunnan valitseminen helpottuu, kun rajausta on tehty valmiiksi. Koska Mobilenotea alettiin kehittää ennen vuotta 2014, jolloin Google julkaisi I/O-konferenssissa virallisen Android-kehityksen tyyli suunnan, Material Designin [7], sovellus sisältää paljon itsekehitettyjä kuvakkeita ja tyyliä, jotka eivät enää ole Googlen sovelluskehityksen kannalta suositeltuja.

Google rohkaisee Android-kehityksessä käytettäväksi Material Design -suuntaviivoja, jotta sovelluskehitykseen tuodaan yhtenäinen tyyli.

2.1 Material Design -tyylikieli

Material Design on Googlen kehittämä tyylikieli, jota voi noudattaa verkko- ja mobiilisovelluskehityksessä. Material Design on suunniteltu toimimaan support.android.com-kirjaston kanssa, jota Mobilenote käyttää käyttöliittymäelementeissään. Uusin versio Material Designille on com.google.android.material-kirjastossa [8]. Mobilenotea ei ole vielä konfiguroitu käyttämään android.*-tyypin kirjastoja, joilla on uusin tuki tuleville kehitykselille, kun taas support.android.com-kirjastoa ei enää ylläpidetä ja päivitetä. Material Design otettiin käyttöön Android 5.0:ssa, Lollipopissa [9]. Sen ominaisuuksia ovat väreilyefekti, 3D-näkymä, varjot ja valaistus, aktiveetti siirtymien animaatiot ja skaalautuvat vektorit xml-tiedostoissa [4].

Väreilyefekti (engl. ripple effect) tekee komponenteista herkemmän tuntuisia lisäämällä niihin väreilymäisen ominaisuuden, jota voisi rinnastaa siihen, kun vesipisara putoaa veteen, jolloin kosketuspinnasta lähtee etenemään väreilymäinen aaltoliike. Androidissa tämä aaltoliike jatkuu kosketuskohdasta aina komponentin reunoille. Väreilyefektiä tukevat painikkeet, valintaruudut ja muut kosketushallitsimet. [4.]

3D-näkymä tekee käyttöliittymään lisää ulottuvuutta, kun aiempien x- ja y-akselien (vaaka- ja pystysuuntaisten akselien) lisäksi näytöllä tuetaan myös z-akselin suuntaista liikettä, eli kohtisuoran akselin toimintaa. Tämä tekee syvyyttä käyttöliittymään, kun näytöllä olevat elementit pystyvät nousemaan toisten yläpuolelle. Uutena asiana ovat myös tästä aiheutuvat varjot ja valaistukset, jotka toteutuvat automaattisesti z-akselilla tapahtuvasta liikkeestä. Elementin nostamista (engl. elevation) voi korostaa kasvattamalla nostamisen arvoa, jolloin varjostus on tummempaa ja näin tuo vahvemman kontrastin näytölle, jolloin käyttäjä huomaa helpommin komponentin olevan irrallaan muista elementeistä. [4.]

Aktiveettien siirtymät (engl. activity transitions) ja muiden kosketusanimaatioiden tukeminen tekevät käyttöliittymästä aiempaa elävämmän, kun siirtymät eri näkymien ja painikkeiden koskettamisesta käynnistävät pehmeään animaation seuraavaan näkymään.

[10.] Animaatiota tuetaan myös, kun komponenttien tila muuttuu, eli kun uusi tieto saadaan ladatuksi ja painiketta voidaan käyttää, voidaan tämä muutos animoida. Tällöin kuvake mahdollisesti muuttuu ja näin siirtymä pystytään kommunikoimaan käyttäjälleen helpommin. Tästä esimerkkinä pysäytyspainikkeesta painaminen ja muuttaminen soittamispainikkeeksi. Uusia animaatiotyylejä on useita: komponentit voidaan saada liikkumaan esimerkiksi sivuttain, pystysuunnassa tai menemään itsensä ympäri.

Vektorityyppisten kuvakkeiden käyttö on myös lisätty aiempien bittikarttojen rinnalle. Tämä mahdollistaa kuvan skaalautuvuuden ilman laadun heikkenemistä. Vanhat bittikartat ovat edelleen saatavilla ja tuettuina. Erona näillä on bittikarttojen monistaminen tallennustilaan eri pikselitiheyttä tukeville näytöille, esimerkiksi hdpi-, xhdpi-, xxhdpi- ja xxxhdpi-näytöille täytyy kullekin olla omat bittikartat tallennettuna, mikä kasvattaa sovelluksen viemää tallennustilan kokoa, kun taas vektoreita käytettäessä selvittää yhdellä kuvakkeella. Pikselitiheys tarkoittaa sitä, että kahden puhelimen näytön fyysinen koko voi olla sama, mutta näytölle mahtuu neliötuumaa kohden esimerkiksi 240 kuvapistettä (engl. pixel) ja toiselle näytölle 480 kuvapistettä. Tässä tapauksessa vaaditaan tiheämpi tarkkuus tiheyspikseleissä, jossa 240 kuvapisteen puhelin voi käyttää hdpi-näyttöä tukevaa bittikarttaa ja 480 kuvapisteen puhelin xxxhdpi-näytölle tarkoitettua bittikarttaa. [11.] Haittapuolena vektoreissa on vaaditun laskentatehon lisääntyminen, kun vektoria joudutaan matemaattisesti laskemalla muuttamaan isommaksi tai pienemmäksi, mikä lisää suorittimen käyttöä. Lisääntynyt suoritinkuorma lisää myös akun käyttöä.

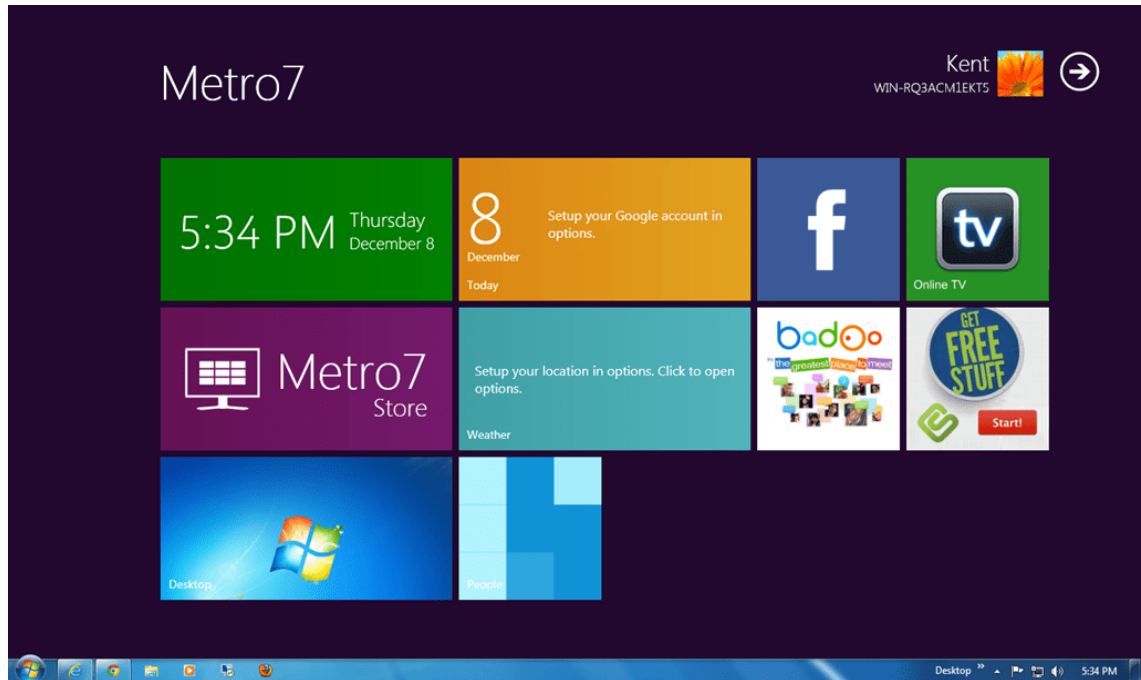
Material Designin tavoite on tehdä komponenteista korttimaisempia, ja tätä on pyritty noudattamaan muissakin Googlen sovelluksissa Material Designin julkistamisen jälkeen. Kortteja käsitellään tarkemmin luvussa 2.6.

2.2 Flat Design -tyylikieli

Flat Design on minimalisuuteen pyrkivä tyylikieli. Sen käyttöä popularisoi Allan Grinshstein 2000-luvun alkupuolella [12]. Grinshsteinin mukaan tyylikkäämpiä käyttöliittymiä ovat ne, jotka saavat eniten aikaan pienimmällä määrällä elementtejä [13]. Flat Design -tyylikielen tarkoitus on saada käyttöliittymä riisutuksi kaikista turhista hienouksista, kuten varjoista, vinoista reunoista, tekstuureista ja gradienteista, jotka saavat aikaan kolmiulotteisuutta, koska Flat Design pyrkii kaikin tavoin kaksiulotteisuuteen. Tämän tyyliuunnun

tunnusmerkki on tuoda kaikki elementit tasaiselle alustalle kaksiulotteisesti, ilman niiden tunnistettavuuden tai käytettävyyden puutteita, sekä käyttää kirkkaita värejä. [14.]

Flat Designia käytetään muun muassa Microsoftin Metro-designissa, josta on esimerkki kuvassa 1.



Kuva 1. Microsoftin Metro-design on käytössä Windows 7 -käyttöjärjestelmässä [15].

Kuvassa 1 näkyvä Microsoftin kehittämä Metro-design-tyyppinen näkymä on Windows-käyttöjärjestelmälle. Sen rakenneosat ovat värikkäitä, reunattomia, erittäin yksinkertaisia laatikoita. Tilaa on jätetty reunoille runsaasti.

Myös Apple käyttää tasaisen alustan elementtejä omassa iOS 7 -käyttöjärjestelmässään, kuten kuvassa 2 on esitelty.



Kuva 2. iPhoneen sääsovellus iOS 7- ja iOS 6 -käyttöjärjestelmillä [16].

Flat Designin käytön näkee etenkin Applen iPhoneen sääsovelluksesta, jossa oikealla puolella oleva vanhempi tyyli iOS 6 -käyttöjärjestelmässä on kuvakkeiden osalta tehty kolmiulotteiseksi ja viikonpäivien taustavärit on tehty gradienttimaisesti tummasta vaaleampaan, kun taas iOS 7 -versiossa kuvakkeet on tehty Flat Designin tyyppisesti kaksiulotteisina ja tausta on samaa kirkasta sinistä. Korttimaisuudesta luovuttiin ja siirryttiin käyttämään koko tilan kattavaa tyyliä. [16.] Ylimääräiset asiat on yritetty karsia pois, kuten lämpötilan astemerkki.

Flat Designia on arvosteltu sen huonosta käyttäjäystävällisyydestä. Varjojen puute tekee painikkeista usein taustaan sekoittuvan, eikä käyttäjä tiedä, miten jonkin elementin kanssa voi olla vuorovaikutuksessa, eli onko se painettava vai ei. Tämän vuoksi Flat

Design on kehittynyt ja siihen usein viitataan Flat Design 2.0:na eli Flat Designin versiona, johon on lisätty varjoja ja väri variaatioita, kuten kuvassa 3 nähdään. [17.]

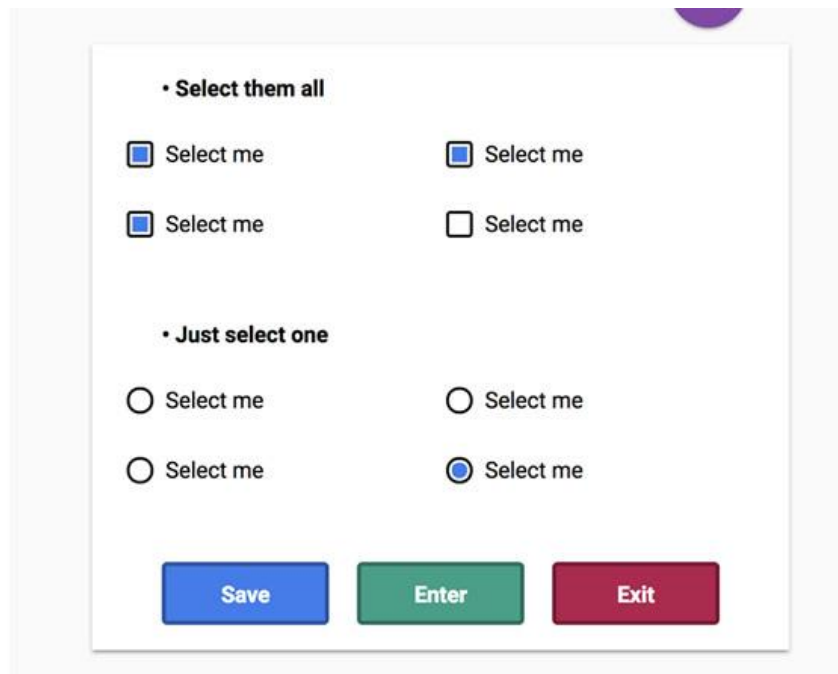


Kuva 3. Varjon käyttäminen Flat Design 1.0:ssa ja 2.0:ssa [18].

Tämä tyyli on kolmiulotteista ja vastaa jo suurelta osin Material Designia [17]. Flat Design 2.0:n tavoitteena on saada käyttäjä ymmärtämään painikkeiden tilaa paremmin aiempaan versioon verrattuna. Varjojen lisäksi kuvakkeissa voidaan käyttää eri värejä, kirkkauteen tai gradienttiin katsomatta. [18.] Flat Design muuttuu siis versiossa 2.0 vapaammaksi ja käyttäjäystävällisemmäksi eikä ole enää täysin sidottu tasaisuuteensa.

2.3 Flat Remix -tyylikieli

Flat Remix on alun perin kehitetty todella kevyeksi CSS-kirjastoksi erittäin nopealle sovelluskehitykselle, mutta sen tyyliä voi soveltaa muuallakin kuin Javascript-maailmassa [19]. Esimerkki Flat Remixistä on esitetty kuvassa 4.



Kuva 4. Flat Remix -komponentteja [19].

Flat Remix on CSS-kirjastona etukäteen määritellyt tyylit painikkeille ja lomakkeille, mutta niiden rekonstruktio Androidille on mahdollista Android Studion omalla muokkausohjelmalla. Kuvassa 4 nähtävät komponentit ovat kirkkaita ja selkeitä, ylimääräistä tekstiä ei juuri ole. Komponenttien suurin ero Flat Designiin verrattuna on niiden reunojen vahvistaminen selkeästi tummalla värillä.

2.4 Tyylin valinta

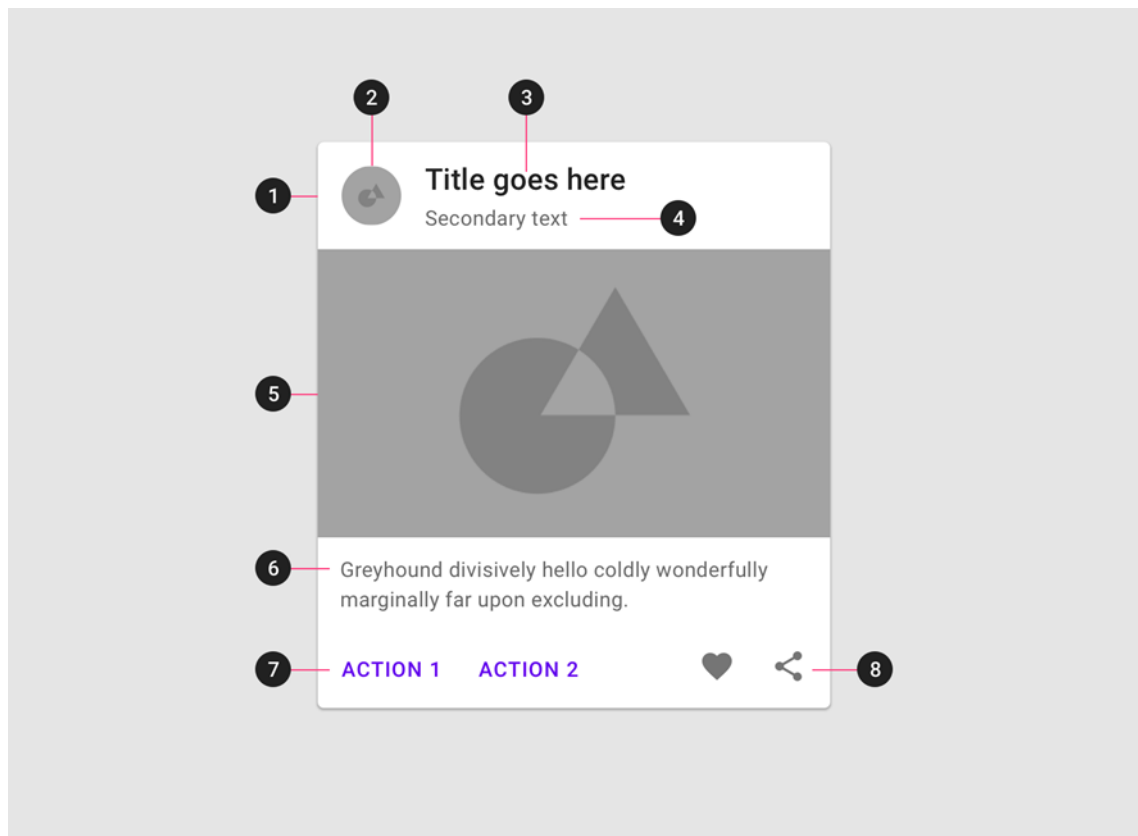
Eri vaihtoehdoista päädyttiin käyttämään Flat Designin uusinta versiota, jota suositellaan käyttämään Android-sovelluskehityksessä: Material Designia. Sen koettiin olevan käytännöllisin vaihtoehto kenttätöissä käytettäväksi, koska painikkeiden tilan ymmärtäminen onnistuu siinä paremmin kuin Flat Design 1.0:ssa. Material Design on myös uusin versio Flat Designista, ja se on Androidin suositteluin ja nykyaikaisin suuntaus. Sen tarjoamat ominaisuudet lisäävät käyttäjystävällisyyttä ja tuovat selkeyttä näytöllä tapahtuviin asioihin. Material Design määrittelee komponenttien ja niiden alkioden koon ja tyylin. Tämä lisää helppoutta elementtien käyttöönotossa, kun vain muistaa seurata suuntaviivoja tarkasti, jotta virheiltilä välttyttäisiin. Näitä suuntaviivoja pyrittiin noudattamaan insinööri-työssä mahdollisimman hyvin.

2.5 Listat

Material Design on määritellyt listan alkioden koon tietyinkokoiseksi, 48 tiheysvapan pikselin (engl. Density-independent Pixels, lyhennettynä dp tai dip) suuruiseksi. Suunnittelussa tämä on keskeinen tekijä käyttöliittymän kannalta: jos tekstin saa listan alkiossa yhdelle riville, on kooksi määritelty tämä 48 dp, mutta jos sillä on pieni kuvake, sen korkeus on 56 dp tai isommalla kuvakkeella jopa 72 dp. Kahdelle riville sijoittuva teksti tekee alkion korkeudesta Material Designissa 64 dp:n tai isolla kuvakkeella 72 dp:n suuruisen. Kolmirivinen alkio on kaikissa tapauksissa, kuvalla tai ilman, aina 88 dp:n suuruinen. [20.] Avattu ja suljettu listan alkio on 48 dp:n suuruinen, mikä tekee siitä listan alkion pienimmän mahdollisimman koon [21].

2.6 Materiaalikorttinäkymä

Materiaalikorttinäkymä on suurin Material Designin tuoma tyyli muutos komponenttien suhteen, koska elementtejä yritetään muuttaa korttimuotoon [22]. Kortin tarkoitus on tuoda nopealla vilkaisulla tietoa käyttäjälle, mutta ei liian tarkasti. Kortissa on usein avaimismahdollisuus lisätietoa varten, tai jotain tietoa voidaan kätkeä vaihtoehtoisesti *laajenna* ja *piilota* -tyyppisten nappien taakse. Tässä uudessa näkymässä on mahdollista korostaa reunan, taustan ja edustan värejä entistä tehokkaammin [22]. Myös itse käyttö on sulavampaa verrattuna itse kehitettyihin korttityyppeihin komponentteihin, kun tuettuna on enemmän ominaisuuksia eikä niitä tarvitse itse koodailla tai kehitellä. Esimerkiksi kortin reunan väriä tai kokoa voi vaihtaa helposti niitä tukevilla funktioilla. Kortista on esimerkki kuvassa 5.



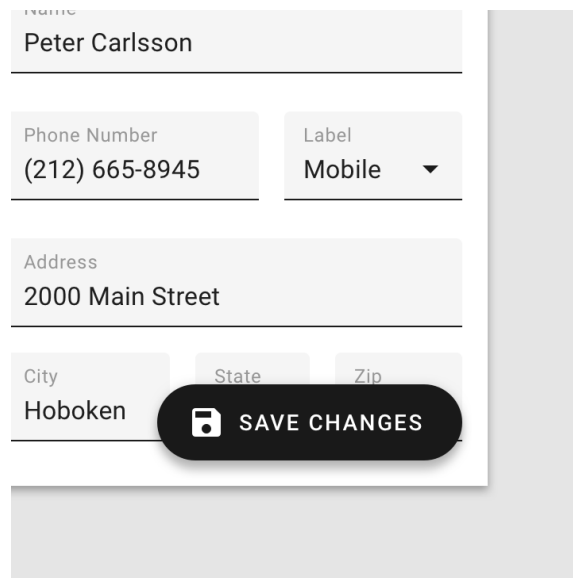
Kuva 5. Material Design -kortti [22].

Kortin ominaisuudet on numeroitu kehittäjien avuksi. Kortin yläosassa, kuvan yläpuolella määritellään tärkeimpiä tietoja, kuten otsikko, apuotsikko ja logo. Keskiosassa on sisällön paikka, jota kuvataan otsikossa. Sisältö voi olla esimerkiksi kuva, video tai kuvaaja. Kortin alaosassa on pidempi selite, painikkeet ja kuvakkeet. Kortin kaikki ominaisuudet ovat valinnaisia ja niiden järjestys voi vaihdella. [22.] Usein nähdään kortteja, joissa sisältö laitetaan ylimmäiseksi ja muut tiedot sen alle. Kortin ominaisuuksia on myös sen laajeneminen ja pienentäminen, jolloin käyttäjälle ei paljasteta kaikkia tietoja heti aluksi.

Korttien uudenlainen käsittely tuo käyttäjän lähemmäs sovellusta. Korttia voi käsitellä sen omilla painikkeilla tai esimerkiksi raahaamalla korttia jommallekummalle sivulle, jos sovellus on luotu tukemaan sitä. Myös korttien järjestäminen onnistuu painamalla korttia pitkään. Tätäkään ei tueta automaattisesti, mutta se tuo luonnollisen lähestymisen kortteihin, jos ominaisuus on lisätty sovellukseen. [22.]

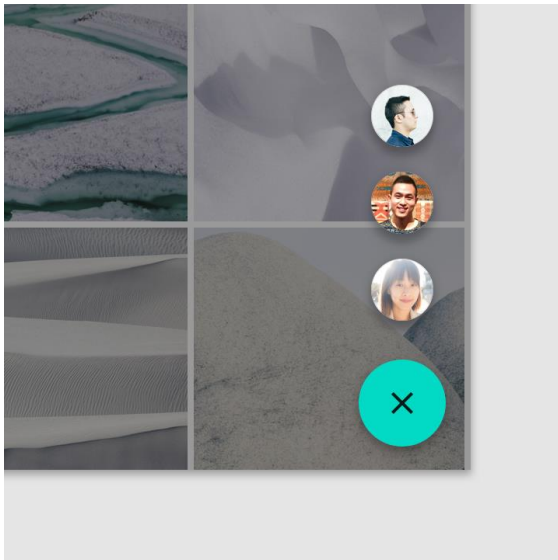
2.7 Kelluva toimintapainike

Kelluva toimintapainike (engl. Floating Action Button, FAB) on näytön päälle rakentuva komponentti, joka tuntuu kelluvan näytön päällä. Se on irrallaan muista näytön komponenteista, eli komponentit eivät voi reagoida sen kanssa. Kelluva toimintapainike tuodaan usein näytön oikeaan alaosaan, ja se on 16 dp irti ala- ja sivureunasta. Se voidaan sijoittaa muuallekin, kuten keskelle näyttöä. Kelluva toimintapainike on tyypillisesti reunoilta pyörästetty tai jopa ympyrän muotoinen. Sillä voi olla kolme tyyliä: tavallinen, mini tai laajennettu. [23.] Laajennetusta toimintapainikkeesta on esimerkki kuvassa 6.



Kuva 6. Laajennettu kelluva toimintapainike [23].

Laajennetun kelluvan toimintapainikkeen ero tavalliseen ja mini-versioon on sen sisältämä teksti sekä kuvakkeen käyttämisen vapaaehtoisuus. Tavallisen kelluvan toimintapainikkeen koko on 56 x 56 dp, minin 40 x 40 dp ja laajennetun 48 dp korkea ja kuvan kanssa vähintään 68 dp leveä; leveys riippuu tekstin pituudesta. Ilman kuvaa laajennettu kelluva toimintapainike on vähintään 32 dp leveä. [23.] Kuvan 7 oikeassa alakulmassa on nähtävissä avattu tavallisenkokoinen toimintapainike, jonka yläpuolella on listattuna kolme minikokoista kelluvaa toimintapainiketta.



Kuva 7. Avautunut tavallisenkokoinen ja kolme minikokoista kelluvaa toimintapainiketta [23].

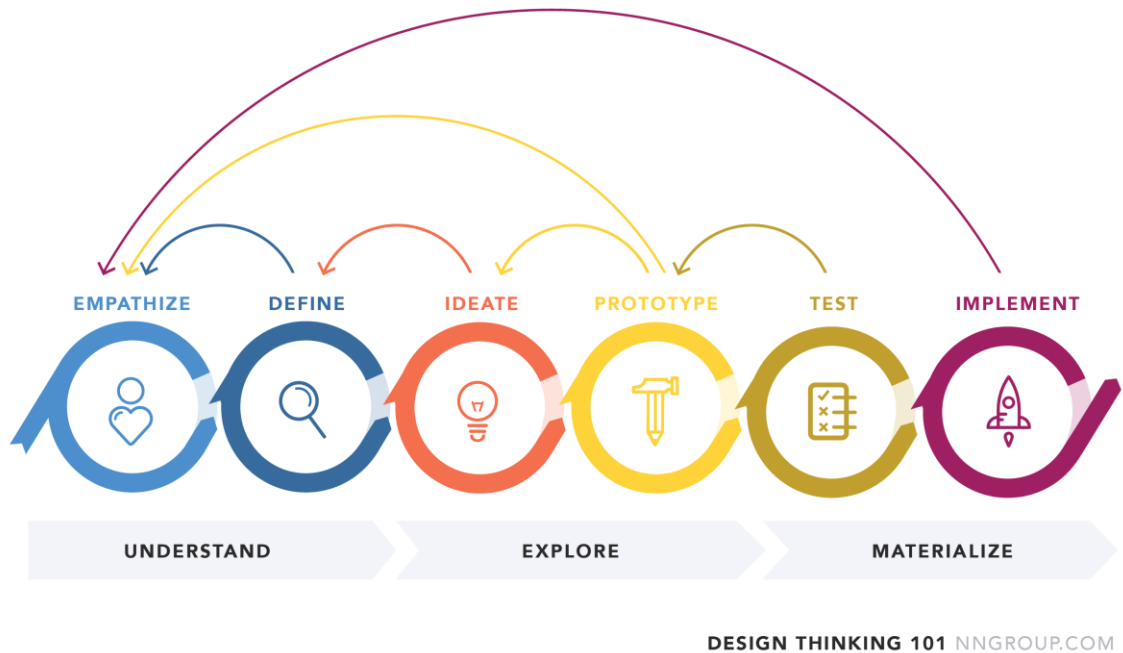
Kelluvasta toimintapainikkeesta aikaansaattavan tapahtuman tulee olla positiivinen pää-tapahtuma, kuten suosikkeihin lisääminen. Esimerkiksi äänen säätäminen tai asetusten avaaminen eivät voi olla sen tapahtumia. Sen painaminen voi avata lisävaihtoehtoja, jotka ovat usein minikokoisia kelluvia toimintapainikkeita. Näitä voi enimmillään olla kuusi, ja suositeltu minimimäärä on kolme, kuten kuvassa 7. Ne voivat avautua animaatiolla, joka tekee käytöstä ja niiden huomaamisesta tyylikkäämpää ja helpommin huomattavaa. Avautuvilla minikokoisilla kelluvilla toimintapainikkeilla voi olla myös kuvatekstit auttamassa niiden ymmärtämistä. [23.]

3 Uuden tehtävälistan edut ja toteuttaminen

3.1 Prototyyppi

Tehtävälistaa lähdettiin toteuttamaan Design Thinking -tyyppisellä ajattelulla kuvan 8 mukaisesti, eli aloitettiin eläytymisellä, josta edettiin hiljalleen prototyyppivaiheeseen, joka oli määrittelyn ja ideoinnin lopputuloksena toteuttaa uusi tehtävälista halutuilla ominaisuuksilla. Design Thinking -ajattelumalliin kuuluu vaiheiden läpikäyminen useaan otteeseen eikä suoraan ensimmäisellä kerralla loppuun asti vieminen [24]. Tämä takaa

laadukkaamman lopputuloksen. Näin ollen on luonnollista palata välillä taaksepäin ja parantaa ominaisuuksia.



Kuva 8. Design Thinking -prosessin vaiheet [24].

Prototyypin tarkoituksena on osoittaa hyvät ja huonot asiat ja sen jälkeen testata niitä, ideoida uudestaan, minkä jälkeen taas muokataan prototyyppiä. Elinkaaren jatkumo jatkuu taaksepäin menevällä askelilla aina käyttöönottoon asti, minkä jälkeenkin palataan tekemään vielä muutoksia, jotka sitten vievät loppuun asti.

Prototyypille asetettiin muutama vaatimus ennen suunnittelun ja koodauksen varsinaista aloittamista:

- Tehtävälistan tehtävien tärkeimpien tietojen tulee näkyä helposti.
- Tehtävälle voidaan lisätä materiaali-, konetunti- ja henkilötuntikirjauksia alikohteina.
- Mahdollisuus luoda uusi alikohde (tehtävä tai jokin muu) tehtävän juurikohdelle (yleensä jokin omaisuuskohte, esimerkiksi liikuntapaikka tai työmaa).
- Mahdollisuus luoda samanlainen tehtävä lennosta (tämä on erikoistapaus edellisestä).

- Tehtävän toiminnallinen kontrollointi on tehtävä helpoksi (kommentointi, kartalla näyttäminen, aloittaminen, keskeyttäminen ja lopettaminen).

Nämä vaatimukset eivät olennaisesti rajoita suunnittelua, etenkin koodauksen suhteen. Käyttöliittymässä ne tulee kuitenkin ottaa huomioon, eli täytyy tehdä jonkinlainen lista, jossa näkyy muutama tärkeä tieto tehtävästä nopealla silmäyksellä. Sen lisäksi tarvitaan tehtävälle hallinnallisia ominaisuuksia, kuten kohteiden luomista ja tilan muuttamista.

Tässä vaiheessa työssä harkittiin laajennettavaa listanäkymää sekä aiempaa tavallista listanäkymää pienin parannuksin. Toteutuksessa haluttiin kuitenkin kokeilla jotain uutta ja selvitettiin, millaiselta laajennettava listanäkymä näyttäisi. Tästä rakennettiin ensimmäinen prototyyppi. On mahdollista, että se tuo käyttöliittymähaasteita, koska tehtävien lukumäärä listalla voi olla erittäin suuri, jolloin niiden hallinnointi voi osoittautua vaikeaksi, jos on liian monta liikkuvaa osaa.

Prototyypissä haluttiin kokeilla tehtävää valittaessa esiin tulevien uusien toimintojen käyttämistä vanhojen ominaisuuksien rinnalla, jotta nähdään käyttöliittymän kapasiteetti. Jo olemassa olevia toimintoja ovat

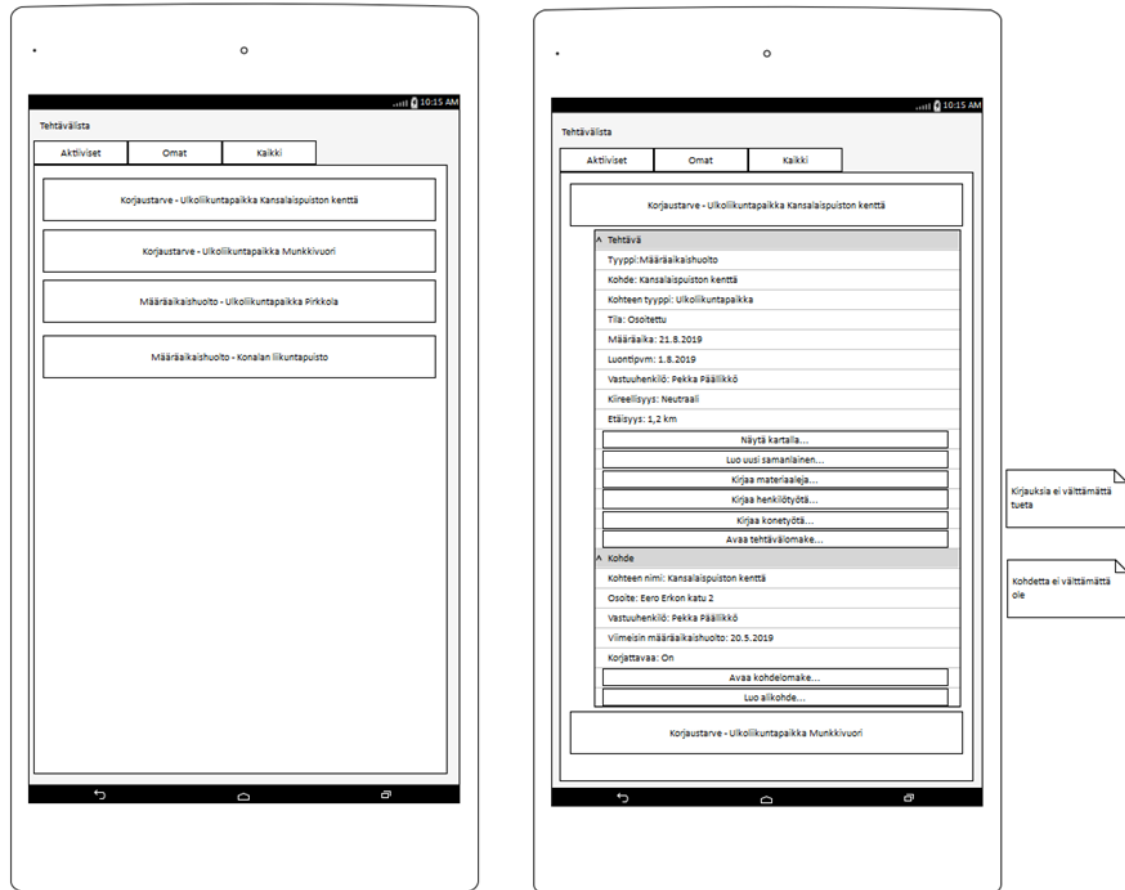
- Aloita.
- Keskeytä.
- Lopeta.
- Liitekuvat.
- Kommentoi.
- Navigoi.

Uusia ominaisuuksia ovat

- Kirjaa materiaaleja.
- Kirjaa henkilötyötä.
- Kirjaa konetyötä.
- Avaa tehtävälomake
- Luo uusi samanlainen tehtävä.

Näiden uusien ominaisuuksien sijoittaminen tehtävälle oli haasteellista niiden lukumäärän vuoksi, sillä puhelimen näytölle mahtuu rajattu määrä toimintoja, ennen kuin käyttöliittymästä tulee liian sekava. Tämä ongelma ei tuntunut vielä prototyypistä suunnitelmassa haittaavan tai rajoittavan mitään.

Kuvassa 9 nähdään prototyypin tuloksia laajennettavaa listanäkymää käytettäessä.



Kuva 9. Tehtävälistan alustava prototyyppi.

Kuvan 9 tehtävää, korjaustarvetta, painettaessa avautuvat tehtävän tarkemmat tiedot sisennettynä. Toisella sisennyksellä tehdyt vaihtoehdot taas ovat tehtävän hallinnoinnin painikkeita. Tässä vaiheessa todettiin, että prototyypistä nähdään hyvin, mikä toimii ja mikä ei. Huomattiin, että vaihtoehtoja on liikaa ja sisennykset eivät näytä käyttäjän silmään kovin hyvältä. Päädettiin palaamaan Design Thinking -prosessin kohtaan ideointi.

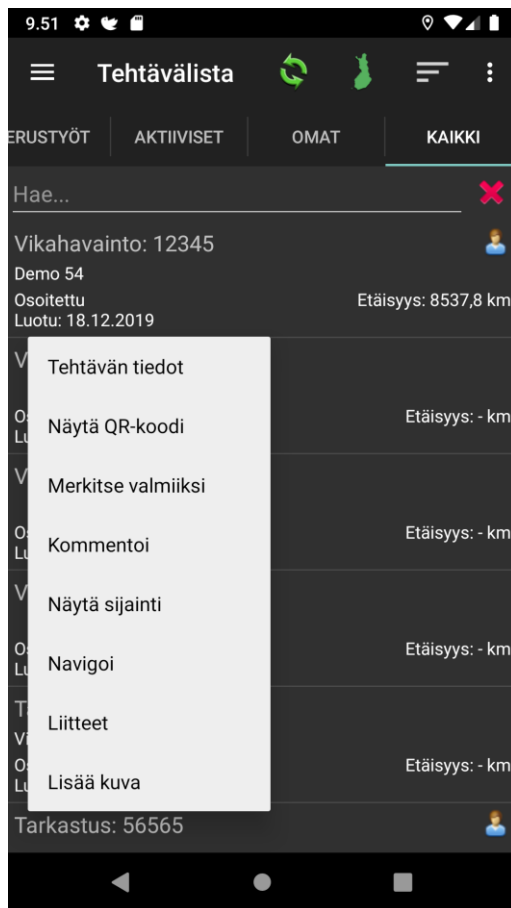
Päätettiin jatkaa seuraavilla muutoksilla:

- Näytetään tehtävälisan tiedoissa enemmän tietoa kuin pelkkä otsikko, eli samat tiedot kuin nytkin on tehtävälisassa (numero, tehtävän otsikko, osoituspäivämäärä, luontipäivämäärä, etäisyys, tila- tai osoituskuvake).
- Kun tehtävälisasta avataan tehtävä, avautuu uusi näkymä, jossa otsikossa on samat tiedot kuin tehtävälisallakin, ja sen alla on tiivistetyssä muodossa, yhdessä komponentissa tehtävän tarkat tiedot.
- Avatussa tehtävälomakkeessa on kelluva toimintapainike, jossa on valittavana tehtävälle tai kohteelle suoritettavat toiminnot (esimerkiksi aloita, keskeytä ja lopeta).
- Yhdistetään kaikki kirjaukset (materiaali-, henkilötyö- ja konetyökirjaus) *kirjaa*-painikkeeksi, josta avautuu uusi lomake, jossa valitaan kirjaustyyppi ja syötetään kirjauksen parametrit.

Lähdettiin rakentamaan lopullista versiota tehtävälisasta, kuitenkin prototyypin ja testaamisen kautta. Jos jokin todettiin toimivaksi, se pidettiin eli implementoitiin, joka on Design Thinking -prosessin viimeinen kohta. Mikään ei ole välttämättä lopullista, vaan aina voidaan palata taaksepäin ja ideoida ja parantaa toiminnallisuuksia, kuten Design Thinking -ajattelumalliin kuuluu [24].

3.2 Listat

Insinöörityössä oli tarkoitus käyttää paljon listanäkymiä, sillä työtehtäviä on työlistassa yleensä valittavana paljon. Tehtävälisalla voi myös tehdä tehtävään tai kohteeseen erilaisia kirjauksia. Enimmillään erilaisia työtehtäviä voi olla yli sata. Tämän vuoksi sovellukseen jätettiin alkuperäinen hakuominaisuus, jossa tehtäviä pystyy hakemaan tehtävälisasta hakusanalla. Haku suodattaa tehtävän nimen ja tehtävän tietojen perusteella tulokset. Erityisesti listanäkymää oli tarkoitus käyttää tehtävien listauksessa, mutta mahdollisuuksien mukaan paljon muuallakin. Kuvassa 10 nähdään alkuperäinen tehtävälis, josta on jo hieman parannettu tekstin jäsentelyä tasoittamalla marginaaleja. Tehtävä on myös valittuna ja tehtävän vaihtoehtoiset toiminnot ovat esillä.



Kuva 10. Alkuperäisessä tehtävälissässä valittu tehtävä.

Alkuperäisessä tehtävälissässä valitulla tehtävällä on paljon vaihtoehtoja kuvan 10 mukaisesti. Siinä ei ole kuvakkeita helpottamassa valinnan tekemistä, mikä on silmälle raskainta, kun ei ole indikaattoreita osoittamassa, mitä voisi tehdä. Vaihtoehtolista jatkuu myös pidemmälle kuin mitä kuvassa näkyy, eikä sivupalkkiin ilmesty vieritysindikaattoria, ellei siihen koske. Tämän takia osa toiminnoista voi jäädä käyttäjältä täysin huomaamatta. Uusi tehtävälissa toteutettiin siten, että nämä haitat saatiin poistettua.

3.3 Kuvakkeet

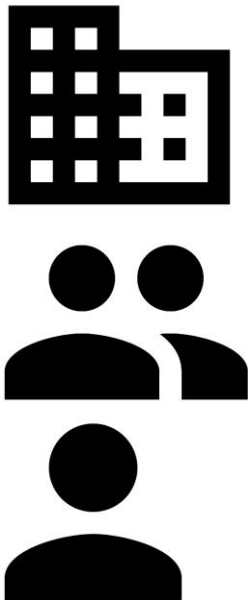
Tehtävälisan kuvakkeiden huomattiin tarvitsevan päivitystä. Päätettiin tehdä uudet Material Designin ja Androidin tukemat kuvakkeet vanhojen itse tehtyjen kuvakkeiden tilalle. Tämä lisää sen ymmärtämistä, kenelle tehtävät on tarkoitettu ja missä tilassa ne ovat. Kuvassa 11 näkyvät alkuperäiset kuvakkeet, jotka olivat täysin itse määriteltäviä eikä niissä ollut yhteneväisyyttä Androidin tukemiin virallisiin kuvakkeisiin, joita Material

Design tarjoaa [25]. Kuvakkeet eivät noudattaneet Material Designin määritelmiä, joissa kielletään kuvakkeen kallistaminen, kääntäminen ja kolmiulotteisuuden tekeminen, kuten oli tehty palikkaa muistuttavassa kuvakkeessa, joka kuvaa organisaatiolle osoitettua tehtävää [26].



Kuva 11. Alkuperäiset tehtävänosoituksen kuvakkeet.

Kuvassa 12 nähdään uudistetut kuvakkeet, joissa alin kuvake tarkoittaa yhdelle ihmiselle tarkoitettua tehtävää, keskimäinen kuvake ryhmälle ja ylin kuvake organisaatiolle kuvattua tehtävää. Kuvakkeiden viralliset nimet vastaavat täysin kuvakkeiden sanomaa eli ihmistä, ryhmää ja toimialaa. Kuvakkeiden isona parannuksena oli bittikarttakuva muuttaminen vektorikuvaksi, eli niistä tehtiin Arie Kaufmanin mukaan tietokoneella koordinaattistoon sidottuja olioita, joita pystyy matemaattisesti skaalaamaan isommaksi tai pienemmäksi laadun kärsimättä, mitä on nyt jo havaittavissa kuvassa 11 [27 s. 86]. Tyyliä muutettiin myös yksivärisiseksi, mikä on Material Designia mukailevampaa.



Kuva 12. Uuden tehtävänosoituksen kuvakkeet.

Värejä käytettiin aiemminkin, mutta niitä selvennettiin lisäämällä niille sopivimmat kuvakkeet pelkkien väripisteiden sijaan, jotka kuvassa 13 nähdään. Kuvakkeet helpottavat väripisteiden asiayhteyden ymmärtämistä. Kuvan 13 hierarkia ylhäältä alas: erittäin kiireellinen, kiireellinen ja kiireetön.



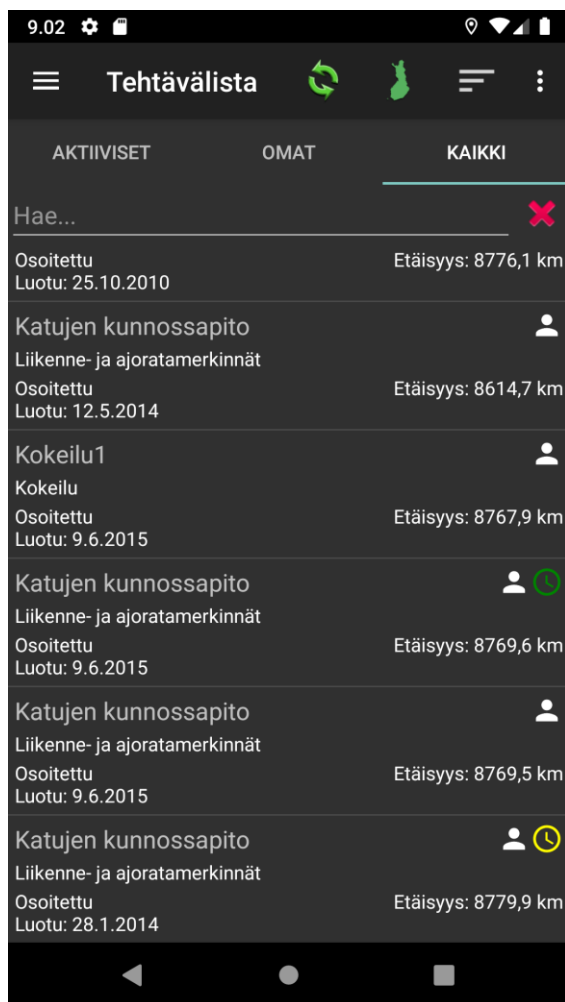
Kuva 13. Tehtävän kiireellisyyden värit vanhassa tehtävälisäyksessä.

Pelkät värit eivät paljon kerro tehtävän kiireellisyydestä, ellei käytetä apukuvakkeita tai kansainvälisesti tunnettuja symboleja hyödyksi. Punainen yhdistetään usein vaarallisuuteen ja huomioväriksi, kuten stop-merkeissä. Keltainen kuvaa usein kansainvälisesti epävarmuutta ja ristiriitaisuutta. Musta on usein yhdistetty suruun, kun taas vihreä on liikennevaloissakin sovitun mallin mukaisesti turvallisuuden väri. Tämän vuoksi musta väri vaihdettiin vihreään kiireettömän tehtävän kohdalla. Värit toimivat hyvin yhteen taustan kanssa, sillä teemoitus on harmaan sävyinen, jolloin ne erottuvat tarpeeksi. Värit auttavat käyttäjäliittymässä hahmottamaan tärkeitä asioita. Kun värit liitetään kuvan 14 mukaisesti aikaan, tulee käsitys kiireellisyydestä. Punainen kello luonnollisesti kuvaa erittäin kiireellistä tehtävää, kun taas vihreä kuvaa paljon kiireettömämpää tilannetta.



Kuva 14. Tehtävän kiireellisyyden värikuvakkeet uudessa tehtävälistassa.

Myös tehtävälistan ulkoasua hiottiin hieman: marginaaleja ja yhteneväisyyttä komponenttien välillä selkiytettiin ja tehtiin symmetrisemmin. Kuvassa 15 nähdään tehtävälistaa uusilla kuvakkeilla ja hiotun jäljen tulosta.



Kuva 15. Tehtävälista, jossa pienet kuvakkeet on päivitetty uudempiin, paremmin soveltuviin kuvakkeisiin.

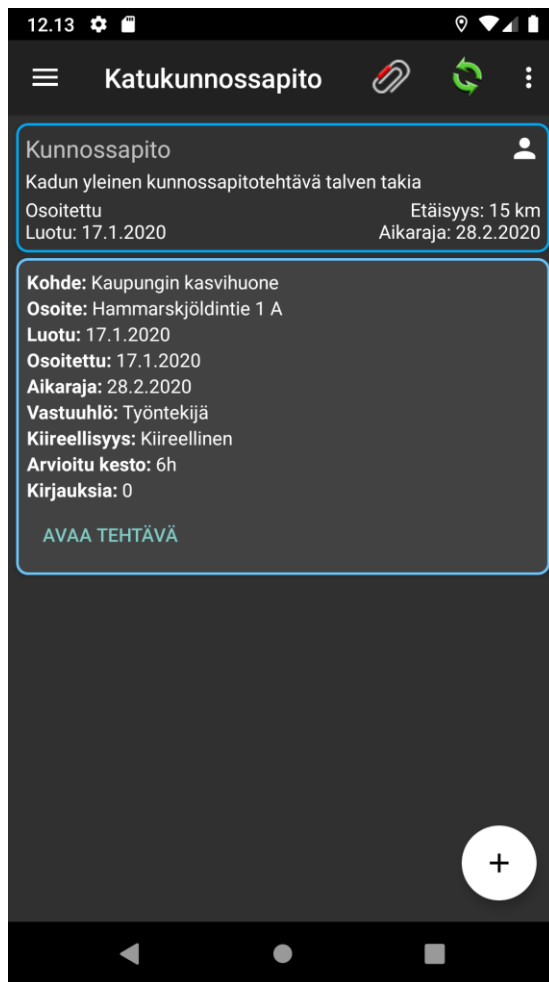
Kun tehtävä on valittu listanäkymästä, avautuu uusi näkymä, jossa on luetteloitu tehtävän tietoja. Aluksi suunniteltiin laajennettavaa listanäkymää näille tiedoille, mutta ajatuksesta luovuttiin kuvassa 9 (s. 15) nähtävän digitaalisen mallin jälkeen, koska se todettiin kehnoksi: näkymästä tuli liian pitkä ja monimutkainen, eikä sen käyttö tuntunut luontevalta ja sujuvalta. Painikkeita, joista pystyi hallinnoimaan tehtävää tai sen tietoja, tuli myös liian suuri määrä näytölle hankalasti hahmotettaviksi. Myös painikkeet tuntuvat häviävän näytölle, eikä selkeyttä onnistuttu lisäämään tarpeeksi. Ajatuksena laajennettava lista on hieno ja moderni, mutta käytännöllisyyden saaminen halutulle tasolle osoittautui erittäin haastavaksi eikä laajennettava lista toiminut halutulla tavalla.

Listan alkion koon määrittely Material Designin mukaan, kuten luvussa 2.5 todettiin, venytti näkymää entisestään, kun haluttiin käyttää laajennettavaa listaa. Aiemmin sovelluksessa oli käytetty vastaavassa tilanteessa näkymiä, joissa ei ollut käytetty suunnittelun suuntaviivoja, eli rivit olivat liian lähellä toisiaan käytettävyyden kannalta. Nyt näytölle eivät mahdu edes yhden tehtävän tiedot ja sillä olevat toimintapainikkeet. Tavoitteena oli myös, että tehtävälisan muita tehtäviä pystyisi kontrolloimaan samalla tavalla. Todettiin, että tämä ei ole käytännöllinen ratkaisu käyttöliittymän ja käyttäjäystävällisyyden kannalta, ja päätettiin siirtyä kokeilemaan uudelle aktiviteetille avautuvaa materiaalikorttinäkymää.

3.4 Materiaalikorttinäkymä

Kun huomattiin, että laajennettava lista ei toimi toivotulla tavalla, siirryttiin käyttämään materiaalikorttinäkymää (engl. Material Card View).

Kuvassa 16 näkyy hyvin korttimaisuus: kortin taustaväri tulee hyvin esille ilman suurempia muutoksia. Ylempää korttia on korostettu niin reunoilta kuin taustavärin puolestakin. Pää tiedot näkyvät ylemmässä korttinäkymässä: tehtävän otsikko tai numero, tehtävän kuvaus sekä ajankohta, jolloin se on luotu ja sen aikaraja. Tietoja ei ole toistettu alemmassa kortissa, paitsi päivämäärätiedot. Muita tietoja ei tarvita kahteen kertaan; se tekisi kortista liian pitkän, koska tehtävän kuvaus voi olla monta kymmentä riviä pitkä. Päivämäärät ovat tehtävän kannalta olennaisia, eikä niitä ole samalla tavalla mainittu peräkkäin, vaan ne on ylemmässä kortissa sijoitettu eri puolille. Kuvan 16 tietokortissa luonti-, osoitus- ja aikarajapäivämäärät on listattu peräkkäin lineaarisesti.



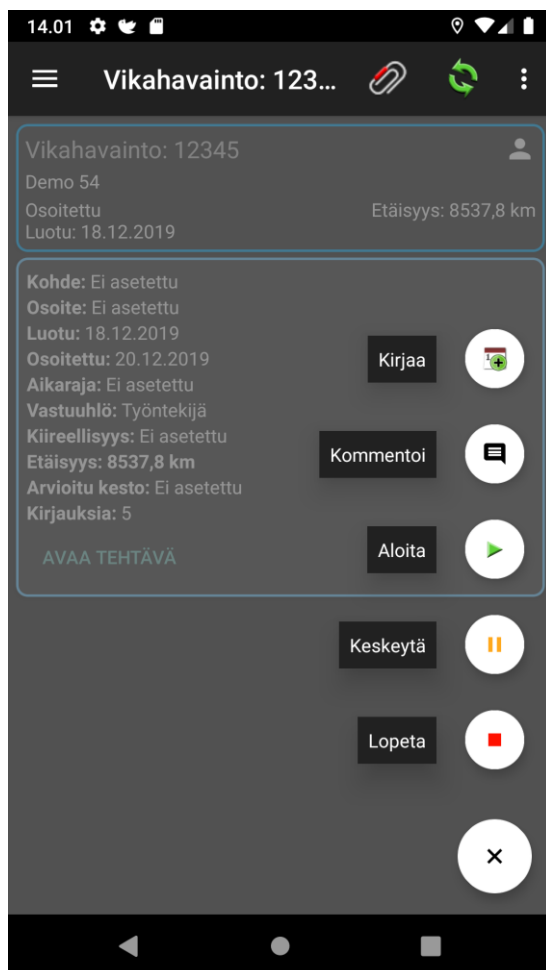
Kuva 16. Tehtävlistasta valittu yksittäisen tehtävän näkymä, jossa käytettynä kahta korttinäkömää.

Kuvan 16 ylempään materiaalikorttiin on laitettu tehtävän tärkeimmät tiedot ja alla olevaan materiaalikorttiin on koottu kaikki toisarvoiset, mutta myös käyttäjälle hyödylliset tiedot. Ylemmän näkymän reunoja on korostettu astetta tummemmalla värillä. Tämä on yleistä Android-sovelluskehityksessä, jossa käytetään teemoituksessa vallitsevaa väriä (engl. primary color) ja toissijaista väriä (engl. secondary color).

Materiaalikortin alaosassa on myös yksi suhteellisen huomaamaton painike, jonka taustaväri on sama kuin materiaalikortin, mikä tekee siitä vähemmän silmille hyppäävän. Siitä voi avata tehtävän tiedot, josta pääsee katsomaan tehtävän pääkohdetta sekä selaa-
maan tehtävälle tehtäviä alikohteita, kuten kirjauksia.

3.5 Kelluva toimintapainike

Kuvassa 16 (s. 23) näkyy oikealla alhaalla puhelinsovelluksissa tyypillinen komponentti, kelluva toimintapainike. Siitä painaessa avautuu jokaiselle tehtävälle erikseen muunneltavat toimintapainikkeet, kun taas yläpalkissa on kaikille tehtäville yhteiset, yleiset toiminnot. Näitä erikseen muunneltavia painikkeita, joita näkyy kuvassa 17, on muun muassa kirjaus, josta pystyy uutena kehitettävänä ominaisuutena tekemään tehtävälle materiaali-, henkilö- ja konetyyppikirjauksia. Nämä ovat aivan uusia toimintoja, eikä niitä ole aiemmin tuettu sovelluksessa. Kelluvan toimintapainikkeen toimintoja ovat luonnollisesti tehtävän hallinnointiin liittyvät aloitus-, keskeytys- ja lopetuspainikkeet. Muita painikkeita ei välttämättä tähän kelluvan toimintapainikkeen alle tehdä, sillä Material Designissa on määritelty avautuvan enintään kuusi vaihtoehtoa [28]. On kuitenkin mahdollista, että niitä tulee lisää, jos asiakkaiden tarpeet muuttuvat tai niitä koetaan tarpeelliseksi lisätä helpottamaan tehtävän hallinnointia. Yhdelle painikkeelle on vielä tilaa, mutta muut toiminnot olisi sen jälkeen sijoitettava muualle.



Kuva 17. Kelluva toimintapainike, joka on avattu painamalla.

Kuvassa 17 on painettu toimintapainiketta kerran, mikä käynnistää animaation, jossa painike muuttuu sulkevaksi painikkeeksi ja tehtävän tausta himmenee, jotta tapahtumapainikkeet tulevat enemmän esille. Painikkeille on lisätty ymmärtämistä helpottavat tekstit. Teksteistä painamalla ei tapahdu niiden kuvaamia toimintoja, painikkeet ainoastaan käynnistävät tapahtumat.

3.6 Yläpalkin kuvakkeet

Kuvassa 17 näkyy ylhäällä navigointipalkissa kolme kuvaketta, joista saa tehtyä tehtävälle toimintoja. Material Design rajoittaa yläpalkkiin oikeaan laitaan tulevien kuvakkeiden määrän kolmeen, ja niiden järjestyksen tulee olla käyttömäärältään vasemmalta

oikealle järjestyksessä [29]. Jos kuvakkeita on enemmän kuin kolme, ne tulee piilottaa pystysuuntaisen kolmipistekuvakkeen taakse [29], kuten nyt on tehty.

Ensimmäisenä vasemmalta on liitteet-painike, josta tehtävään saa lisättyä tai selattua liitteitä. Keskellä yläpalkissa on päivityspainike, josta saa ladattua tehtävän uudelleen, jos se ei ole latautunut kunnolla tai jos haluaa tarkistaa, ovatko tehtävän tiedot muuttuneet. Oikeanpuolimmaisena on lisätoiminnot-painike, josta avautuvat painikkeet, jotka eivät mahdu ylävälilehdelle. Niitä ovat tässä esimerkissä karttapainike, josta saa sijoitettua ilman paikkatietoa olevan tehtävän reaaliaikaisesti kartalle tai tarkastettua sen sijainnin, sekä navigointi, josta pystyy navigoimaan kyseisen tehtävän sijainnille.

3.7 Kirjauslomake

Uutena ominaisuutena tehtävälisälle lisättiin kirjauslomake, joka käyttää uusia Material Designin TextInputEditText- ja TextInputLayout-kenttiä ja myös uusia materiaalipainikkeita. Lomake avautuu yksittäisen tehtävän kelluvan toimintapainikkeen kirjaa-painikkeesta. Kuvassa 18 nähdään lomakkeen yläosassa omassa laatikossa mustalla taustalla joitakin tehtävän tietoja, jotka ovat oleellisia kirjaamiselle siten, ettei lomakkeelta tarvitse poistua tarkistaakseen jonkin yksityiskohdan. Otsikko väritetään vihreäksi, koska ollaan tekemässä uutta. Vihreää väriä käytetään tarkoittamaan uutta ja jatkamista. Tallennuspainikkeessa on käytetty uutta Material Designin -painiketyyppiä, materiaalipainiketta (engl. Material Button). Siinä on uudenlaisia visuaalisia komponentteja, joita voi hyödyntää. Tässä on käytetty painettaessa värin vaihtumista täyttymisanimaatiolla, joka parantaa käyttökokemusta selventämällä, mitä käyttäjä painoi. [30.] Täyttymisvärinä on käytetty aksenttiväriä, eli sinivihreää, RGB-asteikolla nimellä hex #80CBC4 tunnettua väriä, joka on sama kuin esimerkiksi kuvassa 18 nähtävä Lisätieto-kentän otsikon aksenttiväri, joka on samalla aksenttiväri koko sovelluksessa. Lisätiedot-kenttä on väriltään sinertävä, koska sen kenttä on valittu eli aktiivinen. Muuten se on harmaa, kuten esimerkiksi samassa kuvassa nähtävä Määrä-kenttä. Virheellisesti täytettyä kirjauslomaketta lähetettäessä täyttöväri on punainen.

8.55

Demo 54
12345
Vastuuhenkilö: Työntekijä
Kirjauksia: -

Uusi kirjaus

Tyyppi

Konetyö

Konetyyppi

Kiskopyörätraktori

Määrä: 5 Yksikkö: h

Aloitusaika: 11.02.2020 08:54 Lopetusaika: 12.02.2020 01:54

Lisätiedot

Lisätietoja ajelusta

+ TALLENNA

Kuva 18. Mobiilisovelluksen kirjaamislomake.

Kirjauslomakkeella syötettävissä kentissä käytetään Material Designin `TextInputLayout`ia [31] ja `TextInputEditText`ä [32]. `TextInputEditText` on alkio `TextInputLayout`ille [32]. `TextInputLayout` antaa koskiessa kelluvan apukentän `TextInputEditText`in syöttökentälle, jonka oma aputeksti katoaa, kun käyttäjä alkaa kirjoittaa [31]. Näin käyttäjä näkee kentän tiedot myös kirjoittamisen jälkeen, jolloin on helppo tarkastella, mitä tietoja ollaan syöttämässä. Näitä kelluvia apukenttiä ovat kuvassa 18 nähtävät määrä, yksikkö, aloitusaika, lopetusaika ja lisätiedot. Kenttien syötteet tehtiin `TextInputEditText`ille.

4 Tehtävälistan toteutus REST-rajapinnalla

REST-rajapinnan hienouksia on sen helppokäyttöisyys alustasta riippumatta. Sen käytetyimpiä tilattomia operaatioita ovat saaminen, lisääminen, muokkaaminen ja poistaminen (engl. GET, POST, PUT, DELETE), joilla hallinnoidaan verkkopalveluiden dataa. [33.]

4.1 Tietokanta

Uusille ominaisuuksille, kuten materiaali-, henkilö- ja konetyyppikirjaukset, oli kehitettävä uusi taulu PostgreSQL-tietokantaan. Taulun sarakkeissa on linkitetty tehtävä kirjauksiin vieraalla avaimella (engl. foreign key). Sarakkeissa on listattu myös muita oleellisia tietoja tehdystä kirjauksesta, kuten kirjauksen tyyppi, määrä, yksikkö ja haluttaessa lisätiedot kirjauksesta. Tähän käytettiin PostgreSQL-tietokantaa, koska sillä voi tehdä SQL-kutsuja maantieteellisille kohteille, kuten tehtäville, PostGIS-laajennuksen avulla [34].

Tehtävä-tauluun tehtävä kirjaus ei päivitä taulun rivejä, kuten päivytyspäivämäärää, kirjausten lukumäärä -saraketta lukuun ottamatta. Toisin sanoen, tehtävälistalla ei huomaa eroa tehtävien järjestyksessä. Tämä toteutustapa valittiin, koska tehtävälista on riippuvainen tehtävien tilojen muutoksista, kuten tehtävän aloittamisesta tai lopettamisesta, ei sille tehtävien lisäysten takia. Tehtävälistalla voi järjestää tehtävät niiden muokkauspäivän mukaan.

4.2 Palvelin

Palvelimen tehtävä on käyttöliittymän yhdistäminen tietokantaan, joten tässä tehtävälistan uudessa kirjaamisominaisuudessa tarvittiin myös palvelinpuolen kehittämistä. Palvelimelle laajennettiin REST-tukea uusille ominaisuuksille. Siinä on määritetty erityyppisten kutsujen lisäävän ja poistavan sekä muokkaavan tietokannassa olevia tietoja tai lähettävän niitä tietokantaan tai käyttöliittymään tapauskohtaisesti.

Erikoisen palvelinpuolen kehittämisestä teki tehtävälistan tuleminen työnohjauksen puolelta, WorkManagementista, joka on erillinen palvelin Mobilenoten pääasiallisesta palvelimesta, MobilenoteServicestä. Tehtävät tallennetaan työnohjauksen palvelimelle, mutta

kirjaukset ovat itsessään alikohteita tehtävän kohteelle eli MobilenoteService:ssä olevalle kohteelle. Tämän takia tehtävän kirjaukset tallennetaan työnohjaukseen tehtävän kirjauksina sekä alikohteena pääkohteelle, jolloin ne näkyvät karttanäkymässä, kun tarkastellaan kohteiden alikohteita. Kirjaus lähetetään MobilenoteServiceen esimerkkikoodin 1 mukaisesti.

```
public void insertToService(HttpServletRequest request, TaskEntryDTO entryDTO,
TaskManager tm) throws Exception {
    Task task = tm.getTask(entryDTO.getTask());
    String value = task.getWorkId(); //tyo_id
    SimpleFeature entryFeature = null;
    if (value.startsWith("tarkastus")) { //example: tarkastus.5
        entryFeature = createInspectionInstance(entry, value);
    } else if (value.startsWith("vika")) { //example: vikahavainto.7
        entryFeature = createFaultInstance(entry, value);
    }
    WfsClient client = new WfsClient("http://localhost:8180/MNService/wfs",
Collections.emptyMap(), new FeatureIO(), OGCContext.WFS_1_1_0_Context);
    client.addHeader("Authorization", "Bearer " + request.getAttribute(
"fi.geometrix.application.openid.accessToken"));
    TransactionRequest wfsRequest = new TransactionRequest("1.1.0", null,
Collections.singletonList(new InsertOperation(Collections.singletonList(
entryFeature))));
    wfsRequest.post(client);
}
```

Esimerkkikoodi 1. Kirjauksen lähettäminen MobilenoteServiceen.

Kirjauksen lähettämiseen käytetään WFS-rajapintaa, joka on alustariippumaton tapa lähettää tai lukea paikkatietokohteita tietoverkon läpi HTTP-kutsuilla. WFS-rajapinnassa tieto koodataan ja lähetetään GML-tyyppinä, joka on XML-kieliopin tukema merkintätapa maantieteellisille kohteille. [35.]

5 Uuden ja vanhan tehtävälisan vertailu

5.1 Tehtävälisa

Tehtävälisastaan tehtiin paljon muutoksia. Osa parantaa käytettävyyttä, kuten kuvakkeet, värit ja tyylin yhtenäistäminen, kun taas toiset ovat enemmän uusien ominaisuuksien tukemiseen tehtyjä parannuksia. Tehtävälisatojen ulkonäkö onnistuttiin pitämään yhtenäisenä, jotta alkuperäisille käyttäjille ei tule vierauden tuntua uuden tehtävälisan käyttöönotosta, kun vaihto uuteen tapahtuu.

Toiminnallisuus ei muuttunut kuitenkaan tehtävälissä uusien ominaisuuksien lisäämistä lukuun ottamatta. Tehtäväliselle pääsee samalla tavalla kuin aiemminkin, mutta tehtävän valitseminen avaa valikon sijaan kokonaan uuden tehtävän hallintanäkymän. Uusi näkymä sisältää uusia toimintoja, kuten kirjauksien tekemisen. Alkuperäiset toiminnot kuten tehtävän tilan muuttaminen ja liitteiden selaaminen säilyvät, vaikka niiden paikka onkin muuttunut näytöllä.

5.2 Tyyli

Tehtävälisöjen tyyliä tapahtui eniten muutoksia. Tyyliä yhtenäistettiin Googlen Material Designin mukaiseksi. Vanhoja itse suunniteltuja kuvakkeita päivitettiin virallisiin kuvakkeisiin. Marginaaleja ja pehmennyksiä (engl. padding) tyyliä tyyliä kuvakkeiden ohella sopimaan Googlen ohjeistukseen ja näyttämään paremmalta. Tekstit saattoivat olla joko liian lähellä toisiaan tai lähellä näytön reunoja, jolloin käyttäjäkokemus heikkeni. Värejä lisättiin kuvakkeisiin ja materiaalikortille helpottamaan tärkeiden asioiden huomaamista näytöllä.

Painikkeet saivat uutta tyyliä Material Designin myötä. Väriilyefekti painikkeista painettaessa lisää käyttökokemusta ja värikorostus auttaa huomaamaan, oliko painallus positiivinen vai negatiivinen. Kirjauksen lisäämisessä vihreän sävyinen täyttöväri painikkeessa viestii kirjauksen olevan hyväksyttävä. Punainen taas viestii virheellisyydestä.

Uutena ominaisuutena sovellukseen tullut kelluva toimintapainike lisäsi käyttömahdollisuuksia ja toi näytölle lisää tilaa esimerkiksi harventamalla yläpalkin toimintavalikkoa, kun tärkeimpiä toimintoja voitiin siirtää nopeasti käytössä olevaan kelluvaan toimintapainikkeeseen. Nopean siitä tekee näytöllä kelluminen, kun ei enää tarvitse selata yläpalkista vaihtoehtoja. Kelluvan toimintapainikkeen kuvakkeet tuovat hyvän lisän käyttökokemukseen.

5.3 Uudet ominaisuudet

Uusina ominaisuuksina tehtiin tehtävän kirjaukset, josta pystytään laskemaan kustannuksia tehtävän suorittamiseen ajan ja tarvikkeiden kertomana tai tarkastelemaan tarvittuja asioita vastaaville tehtäville tulevaisuudessa. Kun tiedetään etukäteen määritelty

tuntihinta esimerkiksi kaivinkoneelle, voidaan laskea käytetty rahasumma, jos konetta on tarvittu kokonaisen tai useamman työpäivän ajan. Aika ei ole ainut käytettävä suure, vaan osalla kirjaustyypeistä hinta koostuu ajan lisäksi painosta, kuten esimerkiksi rakenustarvikkeiden tai -materiaalien laskemisessa.

Uutena näkymänä tullut yksittäisen tehtävän tarkastelu antaa mahdollisuuden listata entistä enemmän tietoa tehtävästä helposti luettavaksi. Aiemmin näiden tietojen määrää oli rajoitettu näytöllä. Nyt selattavissa ovat kaikki tehtävän tiedot selvemässä näkymässä. Uudessa näkymässä on myös kelluva toimintapainike, joka tuo käytettävyyteen lisäyksen. Sen avautumisanimaatio on myös uutta; aiemmin animaatioita ei ole käytetty.

6 Yhteenveto

Insinööriyössä kehitettiin uudenlainen tehtävälista käytettäväksi paikkatietoon liittyvään kenttätyönohjaukseen. Työ koostui uuden tehtävälistan suunnittelusta ja kehittämisestä uuden tiedon pohjalta, mistä edettiin Design Thinking -tyyppistä ajattelua käyttäen eri työvaiheiden kautta prototyyppivaiheen läpi, aina työn tuotantoon asti viemiseen. Suunnittelu pohjautui perimmältään alkuperäiseen tehtävälistaan, mutta uusia ominaisuuksia tuli paljon. Insinööriyö oli haasteellinen, koska käytettävänä oli paljon tietoa, joka täytyi opiskella ja omaksua, jotta sitä pystyi hyödyntämään.

Kaikkia Material Designin linjauksia ei sisällytetty vielä tämän työn toteutukseen, ja tulevia kehityskohteita on jo siten valmiiksi tunnistettu ja odottamassa jatkotyötä. Keskeisimmät suuntaviivat pystyttiin toteuttamaan aiempaa työlistasta selkeästi paremmin. Puutteina huomattiin, että jos tehtävälistan yksittäisen tehtävän yläpalkin teksti on liian pitkä, ei loppua tulisi piilottaa kolmella pisteellä, vaan yläpalkkia tulisi laajentaa isommaksi, jotta teksti mahtuu kokonaisuudessaan siihen. Mobilenote-sovelluksen kirjastojen päivittäminen androidX:ään tulisi myös tehdä, jolloin saadaan tuki uusimmille Material Design -komponenteille ja saadaan käyttöön entistä enemmän ominaisuuksia vanhoille komponenteille.

Parhaina ominaisuuksina voi mainita Googlen Material Designin mukaisen suunnittelun ja sen uudet komponentit, jotka olivat koko työn perusta. Huolellisen suunnittelun tuloksena tehdyt komponentit ovat hyvä pohja Mobilenote-sovelluksen jatkokehitykselle.

Tehtävälistan suunnittelussa ja käytännössä käytettiin useita vanhoja ominaisuuksia, jotka tuotiin esille uudella, helpommalla tavalla. Uutena isona ominaisuutena lisättiin kirjaukset, joita oli kolmea tyyppiä: materiaali-, henkilö- ja konetyypit.

Kirjauksille tuli rakentaa palvelinpuolen logiikka, tietokantatuki ja mobiilipuolen käyttöliittymä. Yhdessä ne muodostavat uuden tehtävälistan, joka otetaan käyttöön Mobilenote-sovelluksessa vuoden 2020 alusta alkaen. REST-rajapinta mahdollistaa myös helpon integraation selainsovellukseen.

Lähteet

- 1 Etusivu. Verkkoaineisto. Geometrix Oy. <<https://www.geometrix.fi/>>. Luettu 16.2.2020.
- 2 Meistä. Verkkoaineisto. Geometrix Oy. <https://www.geometrix.fi/?page_id=39>. Luettu 16.2.2020.
- 3 Asiakastietorekisterin taloustiedot. Verkkoaineisto. Suomen asiakastieto Oy. <<https://www.asiakastieto.fi/yritykset/fi/geometrix-oy/17790520/taloustiedot>>. Luettu 16.2.2020.
- 4 Material Design for Android. Verkkoaineisto. Google LLC. <<https://developer.android.com/guide/topics/ui/look-and-feel/>>. Luettu 23.3.2020.
- 5 Using Kotlin for Android Development. Verkkoaineisto. Kotlin Foundation. <<https://kotlinlang.org/docs/reference/android-overview.html>>. Luettu 29.2.2020.
- 6 React Native. Verkkoaineisto. Facebook Inc. <<https://reactnative.dev/>>. Luettu 29.2.2020.
- 7 Savov, Vlad. 2014. Google's next big Android redesign is coming in the fall. Verkkoaineisto. Vox Media LLC. <<https://www.theverge.com/2014/6/25/5841302/google-announces-the-next-android/>>. 25.6.2014. Luettu 30.3.2020.
- 8 AndroidX Overview. Verkkoaineisto. Google LLC. <<https://developer.android.com/jetpack/androidx/>>. Luettu 15.3.2020.
- 9 Android Lollipop. Verkkoaineisto. Google LLC. <<https://developer.android.com/about/versions/lollipop/>>. Luettu 29.2.2020.
- 10 Material Design for Android. Verkkoaineisto. Google LLC. <<https://developer.android.com/guide/topics/ui/look-and-feel#animations>>. Luettu 23.3.2020.
- 11 App resources overview. Verkkoaineisto. Google LLC. <<https://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources>>. Luettu 15.3.2020.
- 12 Pratas, Antonio. 2013. Flat Design: An In-Depth Look. Verkkoaineisto. <<https://www.awwwards.com/flat-design-an-in-depth-look.html>>. 4.7.2013. Luettu 21.3.2020.

- 13 Grinshtein, Allan. 2012. The Flat Design Era. Verkkoaineisto. <<https://layer-vault.tumblr.com/post/32267022219/flat-interface-design/>>. 25.9.2012. Luettu 21.3.2020.
- 14 May, Tom. 2018. The beginner's guide to flat design. Verkkoaineisto. <<https://www.creativebloq.com/graphic-design/what-flat-design-3132112/>>. 10.12.2018. Luettu 21.3.2020.
- 15 Chen, Kent. 2011. Metro7 Brings Metro to Windows 7 To Make it Look Like Windows 8. Verkkoaineisto. <<https://www.nextofwindows.com/metro7-brings-metro-to-windows-7-to-make-it-look-like-windows-8/>>. 8.12.2011. Luettu 1.4.2020.
- 16 Wong, Kar Loong. 2013. Apple Design Goes Flat with iOS 7. Verkkoaineisto. <<https://www.bignerdranch.com/blog/apple-design-goes-flat-with-ios-7/>>. 17.9.2013. Luettu 22.3.2020.
- 17 Moran, Kate. 2015. Flat Design: Its Origins, Its Problems, and Why Flat 2.0 Is Better for Users. Verkkoaineisto. <<https://www.nngroup.com/articles/flat-design/>>. 27.9.2015. Luettu 21.3.2020.
- 18 Flat 2.0: Why Your Website Needs This Design Update. 2018. Blue compass Interactive LLC. Verkkoaineisto. <<https://www.bluecompass.com/blog/flat-20-why-your-website-needs-this-design-update/>>. 27.3.2018. Luettu 22.3.2020.
- 19 Flat Remix CSS Library. Verkkoaineisto. Dani Ruiz. <<https://drasite.com/flat-remix-css/>>. Luettu 1.4.2020.
- 20 Lists. Verkkoaineisto. Google LLC. <<https://material.io/components/lists/#specs>>. Luettu 18.2.2020.
- 21 ExpandableListView. Verkkoaineisto. Google LLC. <<https://developer.android.com/reference/android/widget/ExpandableListView/>>. Luettu 18.3.2020.
- 22 MaterialCardView. Verkkoaineisto. Google LLC. <<https://material.io/develop/android/components/material-card-view/>>. Luettu 18.2.2020.
- 23 Buttons: floating action button. Verkkoaineisto. Google LLC. <<https://material.io/components/buttons-floating-action-button/>>. Luettu 28.3.2020.
- 24 Gibbons, Sarah. 2016. Design Thinking 101. Verkkoaineisto. <<https://www.nngroup.com/articles/design-thinkin/>>. 31.7.2016. Luettu 3.4.2020.
- 25 Icons. Verkkoaineisto. Google LLC. <<https://material.io/resources/icons/?style=baseline/>>. Luettu 7.3.2020.

- 26 System icon metrics. Verkkoaineisto. Google LLC. <<https://material.io/design/iconography/system-icons.html#system-icon-metrics/>> Luettu 7.3.2020.
- 27 Kaufman, Arie. 1993. Rendering, Visualization and Rasterization Hardware (Focus on Computer Graphics). E-kirja. Springer-Verlag.
- 28 Buttons: floating action button. Verkkoaineisto. Google LLC. <<https://material.io/components/buttons-floating-action-button/#types-of-transitions/>>. Luettu 23.2.2020.
- 29 Anatomy. Verkkoaineisto. Google LLC. <<https://material.io/components/app-bars-top/#anatomy/>>. Luettu 18.2.2020.
- 30 Material Button. Verkkoaineisto. Google LLC. <<https://material.io/develop/android/components/material-button/>>. Luettu 18.2.2020.
- 31 TextInputLayout. Verkkoaineisto. Google LLC. <<https://developer.android.com/reference/android/support/design/widget/TextInputLayout>>. Luettu 15.3.2020.
- 32 TextInputEditText. Verkkoaineisto. Google LLC. <<https://developer.android.com/reference/android/support/design/widget/TextInputEditText>>. Luettu 15.3.2020.
- 33 What is REST. Verkkoaineisto. Restapitutorial. <<https://www.restapitutorial.com/lessons/whatisrest.html/>>. Luettu 18.2.2020.
- 34 Chapter 1. Introduction. Verkkoaineisto. PostGIS. <https://postgis.net/docs/manual-dev/postgis_introduction.html/>. Luettu 14.3.2020.
- 35 WFS reference. Verkkoaineisto. Open Source Geospatial Foundation. <<https://docs.geoserver.org/latest/en/user/services/wfs/reference.html/>>. Luettu 14.3.2020.