

ANDROID-TOIMINNANOHJAUSOVELLUS



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeenlinnan korkeakoulukeskus, tietojenkäsittelyn koulutusohjelma

Syksy 2020

Joel Närhi

Tietojenkäsittelyn koulutusohjelma
Hämeenlinnan korkeakoulukeskus

Tekijä	Joel Närhi	Vuosi 2020
Työn nimi	Android-toiminnanohjaussovellus	
Työn ohjaaja	Lasse Seppänen	

TIIVISTELMÄ

Opinnäytetyön aiheena on tuottaa kehityssuunnitelma Android-sovellukselle sekä tuottaa prototyyppiversio sovelluksesta. Android-sovelluksen käyttötarkoitus on auttaa perheitä, joiden lapsilla on haasteita oman toiminnan ohjauksen kanssa. Sovellus hyödyntää kuvakortteja, joilla pyritään ohjaamaan lapsi haluttuun toimintaan.

Opinnäytetyön tekijällä ei ole aiempaa kokemusta Android-sovelluskehityksestä, joten myös kehitysmenetelmiä ja työkaluja esitellään opinnäytetyön edetessä.

Opinnäytetyön tuloksena on prototyyppiversio sovelluksesta, jolla voi esittää rajallisen määrän toimintakortteja. Sovellus asennettiin virtuaaliseen Androidiin tietokoneella, sovellusta ei testattu tosielämän tilanteissa.

Avainsanat Android, ohjelmistokehitys, toiminnanohjaus (psykologia)

Sivut 29 sivua, joista liitteitä 0 sivua

Degree Programme in Business Information Technology
Hämeenlinna university centre

Author	Joel Närhi	Year 2020
Subject	Android Executive functions app	
Supervisors	Lasse Seppänen	

ABSTRACT

The purpose of this thesis is to create a development plan and a prototype for an Android application. The purpose of the application is to support families, who's children are strugglinsg with executive functions skills. The way the application will achieve this is by displaying visual cards that implicate what is expected of them.

The maker of this thesis does not have previous experience of Android application development therefore, I will also include chapters of development methods and tools to the thesis.

The deliverables of this thesis are the prototype version of the app and a general direction for further development of the app. The assignee of this thesis is the student himself.

Keywords Android, app development, executive functions (psychology)

Pages 29 pages including appendices 0 pages

SISÄLLYS

1	JOHDANTO.....	5
2	TOIMINNANOHJAUS JA ANDROID-SOVELLUSKEHITYS.....	6
2.1	Toiminnanohjaus.....	6
2.2	Sovelluskehitys.....	7
2.2.1	Kehitysmallit.....	7
2.2.2	Sovelluksen suunnittelu.....	7
2.2.3	Vaatimusmäärittely.....	8
2.3	Android.....	8
2.3.1	Sovelluskehitys Androidille.....	9
3	KEHITYSSUUNNITELMA.....	10
3.1	Sovelluksen tarve.....	10
3.2	Sovelluksen toiminnot.....	10
3.2.1	Kuvakortin näyttäminen.....	11
3.2.2	Kuvakirjasto.....	11
3.2.3	Kuvien löytäminen.....	11
3.2.4	Ajastintoiminto.....	12
3.3	Materiaalin lähde ja käyttöluvat.....	12
3.4	Jatkokehitettävät toiminnot.....	12
3.4.1	Kalenteritoiminto.....	13
3.4.2	Kuvasarja -toiminto.....	13
3.4.3	Tulostustoiminto.....	13
3.4.4	Usean laitteen tuki.....	14
4	SOVELLUSKEHITYS.....	15
4.1	Android Studio-kehitysympäristö.....	15
4.2	Valittu kehitysmenetelmä.....	15
4.3	Vaatimusmäärittely.....	16
4.3.1	Toiminnalliset vaatimukset.....	16
4.3.2	Ei-toiminnalliset vaatimukset.....	17
4.4	Käyttöliittymäsuunnittelu.....	17
4.4.1	Käyttöliittymä.....	17
4.4.2	Valikkorakenne.....	17
4.5	Graafinen ilme.....	18
5	PROTOTYYPPI, SOVELLUSVERSIO 1.....	20
5.1	Sovelluskehitys Android Studiossa.....	20
5.2	Alkunäkymä.....	20
5.3	Kuvat-näkymä.....	21
5.4	Kategoriat-näkymä.....	21
5.5	Sarjat-näkymä.....	22
5.6	Ajastin-näkymä.....	23

6	PROTOTYYPPI, SOVELLUSVERSIO 2	24
6.1	Päänäkymä	24
6.2	Kuvan vaihtaminen sovelluksessa	25
6.3	Käyttöliittymän suunnittelu	26
6.4	Jatkokehityskohteet	27
7	YHTEENVETO	28
	LÄHTEET	29

1 JOHDANTO

Opinnäytetyön tarkoituksena on luoda kehityssuunnitelma ja prototyyppi-versio sovelluksesta, jonka avulla tuetaan lapsen oman toiminnanohjauksen kehitystä.

Omalla toiminnanohjauksella tarkoitetaan kykyä keskittyä annettuun tehtävään ja saada se toteutettua siihen tarkoitettuun ajassa. Tämä on kyky, joka kehittyy lapsilla vuosien saatossa normaalisti. Sen kehitys on kuitenkin vaihtelevaa ja vaatii joskus tukea. (HUS, n.d.)

Yksi tapa auttaa lapsia ja huoltajia arjen haastavissa tilanteissa on toiminnanohjaus korteilla, joissa esitetään selkeällä kuvalla haluttua tilannetta, toimintaa. Kortin avulla lapsen huoltaja voi ohjata lapsen huomion haluttuun toimintaan visualisoidun avulla.

Opinnäytetyön tekijälläkin on ollut näitä kortteja käytössä, mutta niiden suuren määrän vuoksi oikean kortin löytäminen nopeasti juuri oikeassa tilanteessa on haastavaa ja usein mahdotonta. Tämän ongelman ratkaisemiseksi päätin kehittää sovelluksen, jota voisi käyttää puhelimella. Puhelin on melkein aina saatavilla, jolloin myös kortit olisivat melkein aina käden ulottuvilla sovelluksessa.

Tutkimuskysymykset:

- Millainen sovellus on helppokäyttöinen ja hyödyllinen tukemaan lapsen omatoiminnanohjausta
- Mitä toiminnot sovelluksessa ovat tärkeitä, ja mitä voi jättää myöhemmään kehitysvaiheeseen tai kokonaan pois?
- Mitä sovelluskehityksen menetelmiä pitäisi noudattaa?

2 TOIMINNANOHJAUS JA ANDROID-SOVELLUSKEHITYS

Tässä luvussa kerrotaan toiminnanohjauksesta ja sovelluskehityksestä. Mistä lähtökohdista sovellusta lähetään kehittämään ja mitä tarvetta sillä pyritään täyttämään? Luvussa puhutaan myös Androidista alustana ja käyttöliittymäsuunnittelusta.

2.1 Toiminnanohjaus

Oman toiminnanohjauksessa on kyse toimisesta oikein erilaisissa tilanteissa. Kyky on luonnollinen ihmisille ja se kehittyy lapsille jo varhaisella iällä. Kehitys ei kuitenkaan ole välitöntä vaan se on prosessi, joka kestää vuosia. Kyky kehittyy lapsilla vaihtelevalla nopeudella ja oikein toimiminen voi olla joillekin lapsille haastavampaa. He tarvitsevat tukea huoltajilta omassa toiminnassaan. (HUS, n.d.)

Toiminnanohjaukseen sisältyy sellaisia käsitteitä kuin aloitteellisuus, joustavuus, suunnittelu ja inhibitio. Aloitteellisuus on kyky käynnistää toiminta tai suunnitella sitä. Joustavuudessa on kyse kyvystä muuntaa toimintaansa ulkoisten ärsykkeiden perusteella tai tilanteen mukaan. Suunnittelu on tavoitteen ja suunnitelman muodostamista, se edellyttää aloitteellisuutta sekä joustavuutta. Inhibitio on kyky vastustaa ja säädellä sisäisiä impulsseja tai ulkoisia ärsykeitä, jotka voivat olla häiritseviä. (HUS, n.d.)

Tyypillinen esimerkki lapsen oman toiminnanohjauksen haasteesta voi olla kyvyttömyys toimia annetun ohjeen mukaisesti. Näitä tapauksia ilmenee arjessa, kun pitäisi suorittaa aamutoimet, kuten hammaspesu ja pukeutuminen, mutta nuori lapsi siirtyykin leikkimään leluillansa, eikä toimi kuten huoltaja odottaa. Tilanne aiheuttaa toistuvana harmia huoltajalle ja voi turhauttaa lasta, joka ei ymmärrä huoltajan harmistuneisuutta.

Tampereen kaupunki tarjoaa työkaluja lasten huoltajille, joiden avulla voi tunnistaa oman toiminnanohjauksen haasteista kärsivän lapsen sekä oppaita, miten saada arki sujuvammaksi lapsen kanssa, jolla on näitä haasteita. Yksi näistä oppaista opastaa käyttämään visuaalisia kuvakortteja apuvälineenä. Näillä on todettu olevan suuri merkitys oppimisen kannalta. (Lehtisare, n.d.)

Kuvakorttien käyttöä on suositeltu myös opinnäytetyön tekijälle Ylöjärven kunnan varhaiskasvatuksen ammattilaisen, Mirja Laitisen, toimesta ja niiden on todettu olevan hyödyllisiä arjessa. Korttien käytössä on kuitenkin haasteena, että ne ovat harvoin siellä missä niitä tarvitaan tai sitten juuri oikea kortti puuttuu pakasta.

2.2 Sovelluskehitys

Sovelluksella tarkoitetaan äylaitteella tai tietokoneella suoritettavaa ohjelmaa, joka on kehitetty tiettyyn käyttötarkoitukseen. Sovellukset ja niiden käyttäminen on nykyisin arkipäivää valtaosalle maapallon väestöstä ja sovelluksien kehittäminen ja ylläpito kattaa suuren liikevaihdon maailmanlaajuisesti.

2.2.1 Kehitysmallit

Sovelluskehitykseen on useita eri lähestymistapoja ja malleja. Yksi niistä on vesiputousmalli, joka on vaiheittainen prosessi. Mallissa siirrytään vaiheesta toiseen. Vesiputousmallin vaiheet ovat vaatimusten määrittely, suunnittelu, toteutus, integraatio, testaus, asennus ja ylläpito. (Sommerville, 2016, s. 48)

Toinen suosittu sovelluskehityksen malli on ketterä kehitys, jossa sovelluskehitys pilkotaan pieniin helposti hallittaviin kokonaisuuksiin. Sen ideana on tuottaa iteratiivisesti toimivia kokonaisuuksia, jokaisen kehitysvaiheen lopulla on toimiva sovellus. Jokaisessa iteraatiossa joko kehitetään sovellukseen uusi toiminto tai keskitytään korjaamaan tai jatkokehittämään jotain olemassa olevaa toimintoa. Yksi mallin vahvuuksista on joustavuus ja mahdollisuus saada toimiva versio sovelluksesta valmiiksi nopeasti. (Sommerville, 2016, s. 74)

Kolmas sovelluskehityksen malli on prototyypimenetelmä. Prototyyppi on varhainen tai ensimmäinen versio sovelluksesta, joka on kehitetty esittelemaan jotain sovelluksen toimintoja. Prototyypin tarkoituksena on myös havaita mahdollisia ongelmia sovelluksen käytössä mahdollisimman varhaisessa vaiheessa. Prototyypillä voidaan myös ennakoida sovellukseen mahdollisesti tarvittavia muutoksia. Prototyypistä voi myös päätellä mitä ominaisuuksia kannattaa sisällyttää sovellukseen tai mitä jättää pois. (Sommerville, 2016, s. 62)

2.2.2 Sovelluksen suunnittelu

Sovelluksen suunnittelu on sovelluskehityksen vaihe, jossa suunnitellaan sovellukseen lisättävät toiminnot ja kartoitetaan tarpeet. Jossain tapauksissa sovellussuunnittelu ja kehitys ovat erillisiä prosesseja, usein kuitenkin ketterässä kehityksessä pyritään yhdistämään nämä kaksi toimintoa. Varsinkin pienemmissä projekteissa usein tehdään sovellussuunnitelmaa samaan aikaan sovelluksen kehityksen kanssa. Sovellussuunnitelma elää kehityksen edetessä ja suunnitelmasta ei välttämättä tehdä pysyvää dokumentaatiota vaan suunnitelmaa kirjataan lehtiöihin tai valkotaululle. (Sommerville, 2016 s. 56)

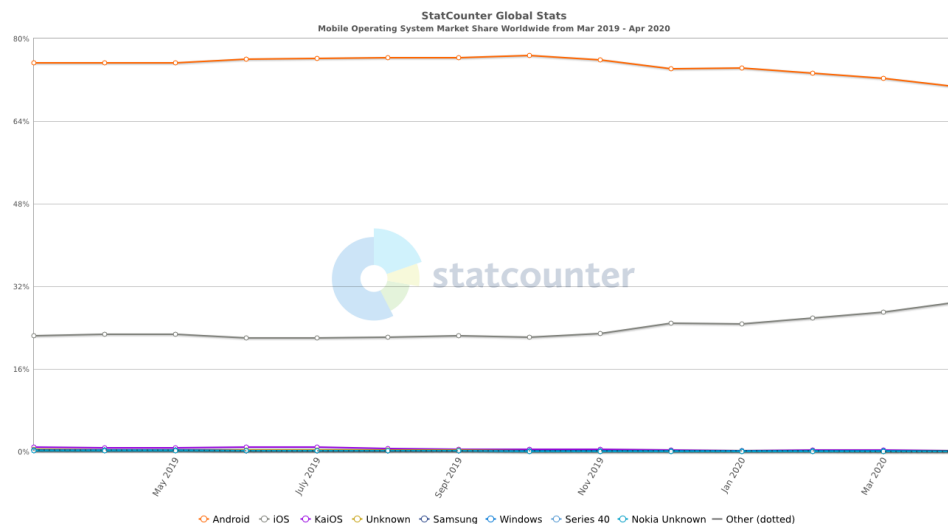
2.2.3 Vaatimusmäärittely

Vaatimusmäärittelyssä on otettava huomioon toiminnalliset ja ei-toiminnalliset vaatimukset. Toiminnalliset vaatimukset ovat toimintoja, joita sovelluksen odotetaan suorittavan. Toiminnallisia vaatimuksia ovat miten sovellus reagoi käyttäjän syötteeseen tai mitä tehdä, kun ajoitettu toiminto päättyy. Jossain tapauksissa nämä määrittelyt voivat olla myös sitä, miten sovelluksen tulee toimia.

Ei-toiminnalliset vaatimukset ovat sovelluksesta riippumattomia vaatimuksia, kuten budjetti ja aikataulu. Nämä joudutaan kuitenkin ottamaan huomioon sovelluskehityksessä ja nämä kartoitetaan ja dokumentoidaan tarvittaessa vaatimusmäärittelydokumenttiin.

2.3 Android

Android on yksi maailman suosituimmista mobiilikäyttöjärjestelmistä, jolla on maailmanlaajuisesti noin 70 % käyttöaste kuten kuvasta 1. näkyy. Toiseksi suosituin käyttöjärjestelmä mobiililaitteilla on IOS 28 % markkinaosuudella.



Kuva 1 Androidin markkinaosuus

Android-käyttöjärjestelmän suosio perustuu osittain siihen, että se on kehitetty avoimen lähdekoodin Linux -käyttöjärjestelmän päälle ja sen omistava Google tarjoaa käyttöjärjestelmää puhelinvalmistajille ilmaiseksi. Googlekin hyötyy järjestelystä, sillä jokaisessa Android-käyttöjärjestelmää käyttävässä puhelimesta ja tabletissa on esiasennettuna Googlen sovelluksia.

Android-käyttöjärjestelmästä löytyy myös esiasennettuna Googlen sovelluskauppa Google Play. Sovelluskaupan kautta on saatavilla erittäin laaja ja monipuolinen kirjasto sovelluksia ja pelejä. Se on kehittäjille helppo tapa toimittaa heidän kehittämänsä sovellus lukemattomien käyttäjien saataville.

Android tarjoaa avoimen kehitysympäristön eli sovellusta kehittäessä voi hyödyntää helposti Android-käyttöjärjestelmän toimintoja kuten hälytyksiä, ilmoituksia, sijaintipalveluita sekä paljon muita. Androidilla on myös mahdollista hyödyntää käyttöjärjestelmän toimintoja, kuten kameraa.

2.3.1 Sovelluskehitys Androidille

Sovelluskehitystä varten Android-järjestelmille vaaditaan kehitysympäristö, eli tietokoneelle asennettava ohjelmisto. Kehitysympäristön avulla suunnitellaan käyttöliittymä ja kirjoitetaan ohjelman käyttämä koodi, sekä lisätään ohjelman käyttämät tiedostot. Ohjelmointi suoritetaan käyttäen Java tai Kotlin -ohjelmointikieltä.

Android-kehityksessä tarvitaan myös tietokoneelle asennettavaa virtuaalista Android-laitetta, joka asennetaan emulaattorina, ohjelmana, joka emuloi Android-puhelinta tietokoneella. Tämän avulla voidaan testata kehitettävää sovellusta. Vaihtoehtoisesti voidaan myös yhdistää Android-laitte tietokoneeseen ja suorittaa kehitettävää sovellusta suoraan laitteessa. Tämä kuitenkin vaatii allekirjoittamattomien sovellusten suorittamisen sallimista Android-laitteessa. Tämä ei myöskään ole käytännöllistä jos haluaa testata sovellusta useilla erilaisilla laitteilla.

Android-sovelluskehitys eroaa tavanomaisesta sovelluskehityksestä tietokoneille. Tietokoneille kehittäessä voidaan huolehtia usein vain itse sovelluksen toiminnasta, käyttöjärjestelmä tarjoaa tarvittavat resurssit sovellukselle, joita voi hyödyntää melko vapaasti. Android-kehityksessä pitää huomioida alustan toiminta tarkemmin, jos sovellus vuotaa muistia, on vaarana koko käyttöjärjestelmän kaatuminen. Android-laitteilla on usein rajallisemmat resurssit kuin tietokoneita käytettäessä. (Murphy, 2011, s. 2)

3 KEHITYSSUUNNITELMA

Luvussa kerrotaan tarkemmin sovelluksen tarpeesta sekä siihen suunnitelluista toiminnoista ja niiden tarpeista. Luvussa kerrotaan myös sovelluksessa käytettävien toimintakorttien lähteestä ja niiden käyttöluvista.

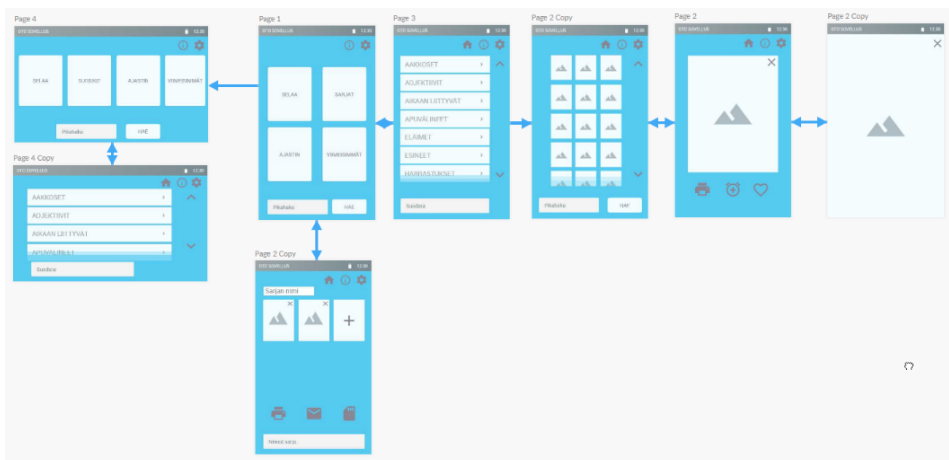
3.1 Sovelluksen tarve

Sovelluskehityksen innoittajana toimii usein jokin tarve. Tässä tapauksessa usean lapsen huoltajalla on tarve auttaa heitä arjen haastavissa tilanteissa, käyttäen oman toiminnanohjauksen tukemiseen tarkoitettuja visuaalisia kortteja. Kortit ovat kuitenkin epäkäytännöllisiä, paperista leikattuja ja laminoituja pieniä lappuja, jotka ovat usein sekaisin tai hukassa.

Sovelluskehityksen perusidea on korvata näiden korttien toiminnallisuus sovelluksessa, jossa kaikki on aina tallessa ja helposti saatavilla, sekä haetavissa ja järjestetty aiheittain.

3.2 Sovelluksen toiminnot

Sovelluksen ensimmäisessä versiossa keskityttiin saamaan aikaan toimiva käyttöliittymä ja mahdollisuus esittää ruudulla kuvakortteja. Sovellukseen suunniteltuja perustoimintoja olivat kuvakorttien löytäminen kategorian tai haun avulla. Sovelluksen käyttäjäksi on visioitu huoltajaa, joka pitäisi laitetta kädessään tai asettaisi sen lapsen näkyville, mutta tarkoitus on kuitenkin, että huoltaja käyttää sovellusta. Jatkossa käyttäjästä puhuttaessa tarkoitetaan huoltajaa, eikä lasta.



Kuva 2 Varhainen käyttöliittymäsuunnitelma

3.2.1 Kuvakortin näyttäminen

Sovelluksen perimmäinen käyttötarkoitus on kuvakortin näyttäminen ruudulla. Toiminnon tarkoitus on kiinnittää lapsen huomio kuvaan ja auttaa häntä ymmärtämään häneltä toivottua toimintaa. Vaikka kuvan esittäminen on itsessään yksinkertaista, tarkoitus on saada juuri tilanteeseen sopiva kuva ja sen löytämisen on oltava helppoa huoltajalle. Kuvakortin esittämisen lisäksi ruudulla on tarkoitus myös lukea selkokielellä, mistä toiminnassa on kyse.

Kuvakortin näyttämiseen liittyy myös sovelluksen muita toimintoja, kuten ajastintoiminto, josta kerrotaan lisää myöhemmin tässä luvussa.

3.2.2 Kuvakirjasto

Sovelluksen lopulliseen versioon on tarkoitus tuoda saataville suuri kirjasto kuvakortteja, joten niiden järjestäminen ja kategoriointi on tärkeää. Sovellukseen luodaan oma sivu, josta voi selata helposti kuvia aiheittain. Jokaiselle kuvalle on tarkoitus antaa ominaisuuksina kategoriat joihin kuva voi kuulua. Sovelluksessa kategoriat täytetään kuvilla, joiden ominaisuuksista löytyy sopiva kategoria. Joten yksi kuva voi kuulua useampaan eri kategoriaan.

3.2.3 Kuvien löytäminen

Sovelluksen käyttötarkoituksen mukaisesti on tärkeää löytää nopeasti tilanteen vaatiessa sopiva kuva näytettäväksi. Tätä varten sovellukseen suunnitellaan muutama eri toiminto kuvien löytämiseksi. Ensinnä aiheiden mukainen luettelo. Sovelluksen päävalikosta pääsee suoraan näkymään, jossa on listattuna aiheittain eri toimintoja. Aiheet on järjestetty aakkosjärjestyksessä luetteloksi, jota voi vierittää käyttöliittymästä vetämällä sormella luetteloa alemmaksi.

Kun jotain aihetta napsauttaa, siirrytään sovelluksessa ruudukkonäkymään, jossa on esikatselukuvina lueteltu aiheeseen sopivat kuvat. Tätä näkymää voi myös vierittää sormella, koska kaikki kuvat eivät voi mahtua yhtä aikaa ruudulle. Kun jotain haluttua kuvaa napsauttaa sormella, siirrytään sovelluksessa kuvan esitystoimintoon.

Toinen toiminto on sovelluksen päänäköymän alalaidassa oleva hakukenttä, johon voidaan kirjoittaa haluttu hakutermi. Kun hakutermi on syötetty ja napsautettu hae -painiketta ruudulle tulee vastaava luettelo esikatselukuviista, joista valintaa napsauttamalla pääsee taas suoraan kuvan katselutoimintoon.

3.2.4 Ajastintoiminto

Ajastintoiminnan toteutuksessa voidaan mahdollisesti hyödyntää Android-käyttöjärjestelmän olemassa olevaa ajastinta sovelluksen sisällä. Jos tämä ei ole mahdollista, joudutaan ajastintoiminto ohjelmoimaan itse. Sovelluksessa voidaan antaa näytöllä esitettävälle kuvalle aika, jossa lapsen on tarkoitus suoriutua annetusta tehtävästä. Ajastin voi auttaa joitain lapsia keskittymään annettuun tehtävään, mutta se voi myös mahdollisesti häiritä toisia lapsia. On huoltajan vastuulla tunnistaa, onko ajastimen käytöstä lapselle hyötyä vai haittaa.

Toiminnossa on käyttäjän tarkoitus asettaa nopeasti aika toiminnolle. Aika asetetaan minuuteissa ja sekunneissa. Toiminnon lopuksi sovelluksessa voidaan toistaa merkkiäni ja korvata ajastin ruudulta tekstillä ”aika loppui”. Ajastin on tarkoitus toteuttaa visuaalisesti yksinkertaisella ulkoasulla, jotta se ei häiritse sovelluksen oleellisesta asiasta eli esitettävästä kuvakortista.

3.3 Materiaalin lähde ja käyttöluvat

Sovelluksessa käytettävä kuvamateriaali haetaan papunetin kuvapankista. Materiaalia saa käyttää Creative Commons ”Nimeä-Epäkaupallinen-Jaasamoin” -lisenssillä. Tämän lisenssin perusteella saatavaa kuvakirjastoa voi käyttää vapaasti, kunhan ei saa itse siitä kaupallista hyötyä ja merkitsee kuvien lähteet selkeästi. Papunetin kuvapankista on saatavilla yhteensä yli 30 000 kuvaa, joista suurin osa on erilaisia piirroksia.



Kuva 3 papunet -kuvakirjasto

3.4 Jatkokehittävät toiminnot

Sovellukseen on suunniteltu toimintoja, joita ei pyritä saamaan mukaan prototyyppivaiheeseen. Nämä toiminnot ovat toteutukseltaan haastavampia sekä aikaa vieviä toteuttaa. Toimintojen jatkokehitys tulevissa

versioissa on mahdollista ja todennäköistä, jos sovelluksen kehitystä jatketaan prototyypivaihetta pidemmälle.

3.4.1 Kalenteritoiminto

Sovelluksen käytön näkökulmasta on oleellista, että tilanteeseen sopiva toimintokortti olisi heti saatavilla, kun sovelluksen avaa. Siihen nähden toiminto, jossa voi määrittellä etukäteen arkipäivän kulun, voisi olla hyödyllinen. Toiminto tarkistaisi sovellusta avattaessa, mikä päivä ja kellonaika on ja tuon tiedon perusteella toisi ennalta määritetyn kuvan suoraan ruudulle. Sovelluksen käyttäjä voisi määrittellä, että arkipäivisin klo 7 halutaan näyttää aamupala -kuvaa tai klo 7:30 pukeutuminen kuvaa.

Toiminto hyödyntäisi Android-käyttöjärjestelmän kalenteria, ja vaatisi sovellukselle myönnettävän käyttöluvan kalenterille. Toiminto on hankala toteuttaa, eikä se ole oleellinen sovelluksen perustoiminnan kannalta. Sen takia se jätetään sovelluksen kehityssuunnitelmaan toteutettavaksi myöhemmässä vaiheessa.

3.4.2 Kuvasarja -toiminto

Sovellukseen halutaan kehittää toiminto, jossa käyttäjä voi luoda usean kortin sarjan. Kun lapsi on suoriutunut yhdestä tehtävästä, voidaan siirtyä suoraan seuraavaan ennalta määritellyyn toimintoon. Ilman että sovelluksen käyttäjän tarvitsee etsiä seuraavaa korttia. Tämä helpottaa sovelluksen käyttöä, kun lapselle esitetään usein samoja toimintoja. Tähän liittyen on tarkoitus myös kehittää käyttäjälle mahdollisuus merkitä yksittäisiä kuvakortteja tai sarjoja suosikeiksi, jolloin niihin olisi helppo ja nopea pääsy sovelluksesta.

3.4.3 Tulostustoiminto

Vaikka sovelluskehityksen tarve onkin korvata irralliset tulostetut laput, jotka ovat helposti hukassa, niiden käytössä on silti etunsa ja niitä voi käyttää monipuolisesti, esimerkiksi A4 -kokoiselle pohjalle tulostetulle viikkokalenterille voi eri päiville liimata kuvakortteja, jotta lapsi tietää mitä tiettyinä päivinä tehdään.

Tätä vastaavaa toimintoa on sovellukseen hankala lisätä, koska sovelluksen ruutu ei ole aina nähtävillä ja puhelinta käytetään moneen eri käyttötarkoitukseen. Kiertotienä tälle haasteelle sovellukseen, voisi kehittää mahdollisuuden tulostaa siinä luotuja kuvasarjoja. Tämän toiminnon lisäys sovellukseen on kuitenkin epätodennäköistä, koska se jää sovelluksen peruskäyttötarkoituksen ulkopuolelle ja toiminto on helpompi suorittaa

tietokoneella sieltä löytyvillä sovelluksilla, joten toiminto ei toisi merkittävää lisäarvoa sovellukselle.

3.4.4 Usean laitteen tuki

Prototyypiversiota kehitetään toimimaan vain Android-laitteella ja tietyllä näyttökoolla. Sovelluksen jatkokehityksen kannalta voisi olla oleellista luoda myös IOS-versio sekä tuki erikokoisille laitteille. Sovelluksen käyttö voisi olla useassa tapauksessa helpompaa tabletilaitteella, jossa suuren näytön ansiosta saa suuremman kuvan näytölle, tai muutaman kuvan yhtä aikaa. Usean erilaisen laitteen tuki tuo omat haasteensa kehitykselle ja se jätetään prototyypiversion kehityksestä pois tuosta syystä.

4 SOVELLUSKEHITYS

Tässä luvussa kerrotaan tarkemmin sovelluskehityksessä käytetyistä työkaluista ja menetelmistä.

4.1 Android Studio-kehitysympäristö

Sovelluksen prototyyppiversio kehitetään käyttäen Android Studio-kehitysympäristöä, joka on Googlen julkaisema työkalu sovelluksien kehittämiseksi Android-alustalle. Sovelluksen kehityksessä on tarkoitus käyttää Googlen julkaisemia kehitystapoja, jotka löytyvät Android-kehitysooppaasta. Näitä noudattamalla voidaan varmistaa sovelluksen yhteensopiavuus erilaisilla Android-laitteilla.

Sovellus prototyyppi ohjelmoidaan käyttäen Java-ohjelmointikieltä, koska se on entuudestaan tuttu sovelluksen kehittäjälle. Toinen vaihtoehto olisi ollut Kotlin-kieli, joka on jopa suositumpi Android kehittäjien keskuudessa. Android Studio-kehitysohjelmisto on saatavilla ilmaiseksi Googlen tarjoamana. Kehitysympäristön lisäksi tarvitaan koneelle asennettava virtuaalinen Android-laite ohjelmiston testausta ja suoritusta varten. Tässä tapauksessa asennettiin käyttöön virtuaalinen Nexus 5X -Android-laite.

4.2 Valittu kehitysmenetelmä

Oman toiminnanohjauksen tukisovelluksen kehitysmalliksi valittiin prototyyppimalli. Mallissa aletaan kehittää sovellusta hyvin vähäisellä alustavalla selvitystyöllä ja dokumentaatiolla. Sovelluksen idea ja käyttötarkoitus on toki selvillä. Myös käyttöliittymän ensimmäisestä versiosta oli luonnoksia tehtynä. Kun käyttöliittymä on nähtävillä, voidaan sen päälle alkaa rakentamaan toiminnallisuuksia sovellukseen.

Prototyyppimallinen kehitys soveltuu monesta syystä tämän sovelluksen kehitysmalliksi, sen avulla voidaan visualisoida sovelluksen tarpeita ja toimintoja suunnittelemalla ensin käyttöliittymä. Käyttöliittymän on todettu myös olevan hyvin tärkeä tämän sovelluksen helppokäyttöisyyden kannalta.

4.3 Vaatimusmäärittely

Vaatimusmäärittely on ohjelmistokehityksen ensimmäinen ohjaava dokumentti, jossa määritellään

- sovelluksen käyttötarkoitus ja miten se saavutetaan
- mitä vaatimuksia onnistuneella sovelluskehityksellä on
- mitä ominaisuuksia sovelluksen pitää sisältää ja mitä vaatimuksia ne asettavat sovellukselle.

Vaatimusmäärittely on tärkeää toteuttaa huolellisesti, jotta sovelluskehitys pysyy asetetuissa raameissa. Vaatimusmäärittelyvaiheen tuloksena on dokumentti, johon koko loppukehitys nojautuu.

Vaatimusmäärittelyn tueksi pyritään tunnistamaan sovelluksen eri käyttäjät ja minkälaisia tarpeita heillä on. Tarpeiden perusteella voidaan luoda käyttäjätarinoita, joista taas pyritään saamaan esille sovellukseen kehitettäviä ominaisuuksia. Sovelluskehityksessä joudutaan usein tasapainottelemaan erilaisten tarpeiden välillä, jotkin ominaisuudet, joita eri käyttäjät kaipaavat voivat olla ristiriidassa keskenään. Jos tällaisia vaatimuksia tulee esiin, tulee sovelluksen kehittäjien pystyä selvittämään mitä tarpeita painotetaan.

Vaatimusmäärittelyssä otettiin kantaa sovelluskehitykseen pitkällä tähtäimellä, joten vaatimusmäärittelydokumenttiin lisättiin vaatimuksia, joita ei ole tarkoitukseen toteuttaa sovelluksen prototyyppiversiossa.

4.3.1 Toiminnalliset vaatimukset

Toiminnallisia vaatimuksia sovellukselle, joita löydettiin vaatimusmäärittelyä tehdessä, olivat mm. helppokäyttöisyys ja sovelluksen käyttöliittymän selkeys.

Sovellusta suunnitellaan käytettäväksi arjen hektisissä tilanteissa, joissa oikean kuvakortin saaminen esille pitää olla nopeaa ja vaivatonta. Tarkoitus on helpottaa huoltajan arkea, kun hän tukee lapsen oman toiminnanohjausta. Tarkoitus ei ole, että huoltaja eksyy selaamaan puhelintaan, kun on oletettavasti haastava arjen tilanne käsillä, jossa kuvakortin tuomaa tukea tarvitaan.

Esimerkki toiminnallisesta vaatimuksesta, joka liittyy toimintoon, jota ei ole tarkoitukseen lisätä prototyyppi versioon sovelluksesta on sovelluksen pääsy säätietoihin, jotta kuvakorttia pukeutumisesta tarjottaessa voidaan ottaa sääolosuhteet huomioon. Lämpötila- ja sadetiedon perusteella sovellus osaa ehdottaa oikeantyyppisten vaatteiden pukemiseen liittyvää kuvakorttia.

4.3.2 Ei-toiminnalliset vaatimukset

Sovelluksen suunnitteluvaiheessa kartoitetaan myös ei-toiminnalliset vaatimukset, eli sovelluksen toiminnan kannalta tulevat vaatimukset. Jotka eivät liity suoraan sovelluksen käyttäjään tai käyttöön. Sovelluksen toiminnallisia vaatimuksia löytyi kartoituksessa vähemmän, koska sovelluskehitys lähtee omasta tarpeesta eikä suunnitelmissa ole sovelluksen julkaisua yleisesti, ei ole budjettia tai aikataulua. Ainoastaan prototyypiversion kehitys opinnäytetyötä varten on selkeä ei-toiminnallinen vaatimus.

4.4 Käyttöliittymäsuunnittelu

Käyttöliittymällä tarkoitetaan rajapintaa, jonka kautta sovelluksen käyttäjä on vuorovaikutuksessa sovelluksen kanssa. Käyttöliittymä sisältää kaikki näytöllä olevat elementit, joiden avulla käyttäjä voi vaikuttaa sovellukseen. Toimivan käyttöliittymän edellytyksenä on selkeys, käyttäjän pitää pystyä ymmärtämään helposti mikä on eri elementtien käyttötarkoitus. Jos käyttöliittymä on suunniteltu huonosti, käyttäjä turhautuu. Hän ei löydä tarvitsemiaan toimintoja ja sovelluksen käyttö hankaloituu tai estyy.

Ensimmäisen prototyypiversion käyttöliittymä suunniteltiin piirtämällä se kuvankäsittelyohjelmassa, myöhemmin tuon piirroksen perusteella luotiin käyttöliittymä Android Studio kehitysympäristössä. Tässä vaiheessa lisättiin painikkeille toiminnallisuus, jossa niitä napsauttamalla siirryttiin sovelluksen näkymistä toiseen.

Toiminnanohjaussovelluksen toisen prototyypiversion käyttöliittymä suunniteltiin samalla kun prototyypisovellusta luotiin Android Studiossa. Tämän etuna oli mahdollisuus testata käyttöliittymän toimivuutta samalla kun prototyypisovellusta rakennettiin. Tämän seurauksena päädyttiin tekemään toinenkin versio prototyypisovelluksesta, kun todettiin että ensimmäinen käyttöliittymäversio ei ole riittävän selkeä ja helppokäyttöinen. Sovelluksen oleellisin käyttötarkoitus oli piilotettuna painikkeiden taakse.

4.4.1 Käyttöliittymä

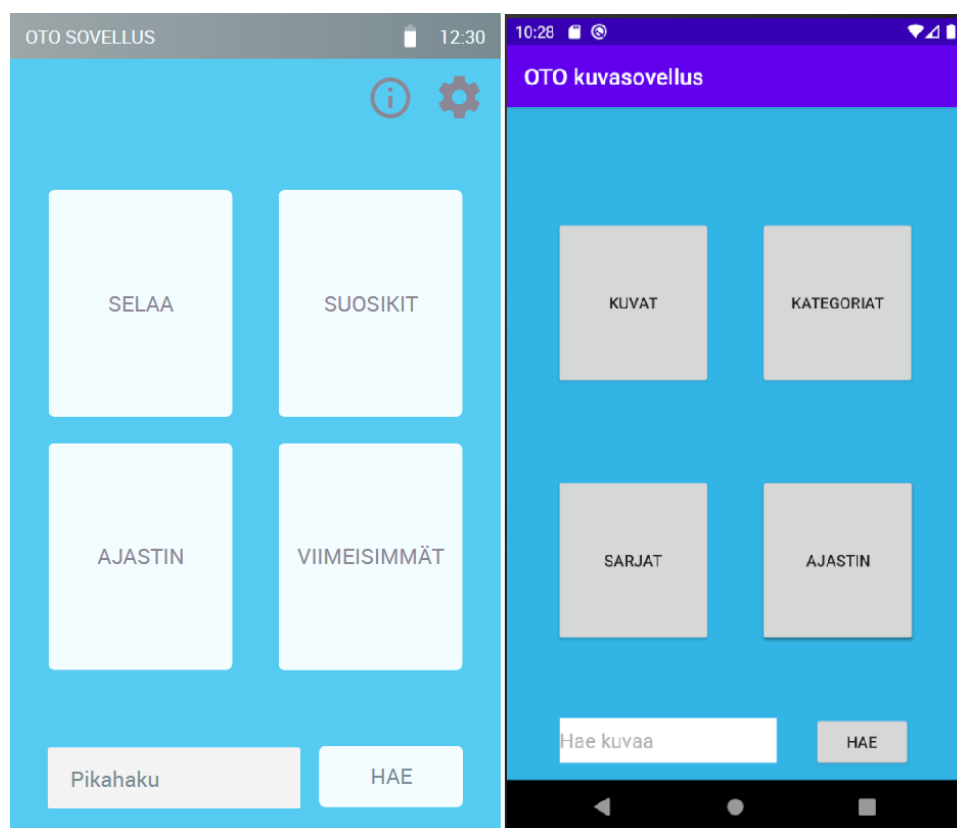
Käyttöliittymänsuunnittelun lähtökohtana oli suunnitella mahdollisimman yksinkertainen ja selkeä käyttöliittymä, josta löytää tarvittavat toiminnot nopeasti ja helposti. Käyttöliittymäsuunnittelussa olivat etusijalla toiminnallisuus, käytettävyyden sekä selkeys. Käyttöliittymän on tarkoitus olla myös johdonmukainen navigaation ja toiminnallisuuden kannalta.

4.4.2 Valikkorakenne

Ensimmäisen prototyypiversion alkunäkymä koostuu neljästä pääkategoriarista, suosikit, ajastin ja viimeisimmät. Sen lisäksi näkymän alalaidassa on hakukenttä. Sovelluksessa jokaista toimintoa vastaa oma näkymä, kun

käyttäjää painaa jotain painiketta tai käyttää hakua, ohjataan hänet toimintoa vastaavalle sivulle.

Navigointi tapahtuu Android-käyttöjärjestelmän omilla navigointipainikkeilla, paluu -painike johtaa aina edelliselle sivulle. Käyttöliittymäsuunnitelmassa oikeaan yläkulmaan oli tarkoitus lisätä Info ja asetukset -painikkeet. Näitä ei kuitenkaan lisätty prototyypiversioon, selkeyden lisäämiseksi.



Kuva 4 Käyttöliittymä piirros ja prototyyppi 1

Selaa -painikkeella siirrytään kuvakategoriat sisältävälle sivulle, jonka taakaa voi selata kaikkia eri kuvakategorioita. Luettelo voi vierittää sormella ylös tai alaspäin, jolloin vain lista ruudulla liikkuu häipyen ruudun yläpäässä ja ilmestyen ruudun alapäästä. Kun kategoriata napsautetaan, siirrytään kategoriata vastaavalle sivulle, josta löytyy aihetta vastaavat kuvat.

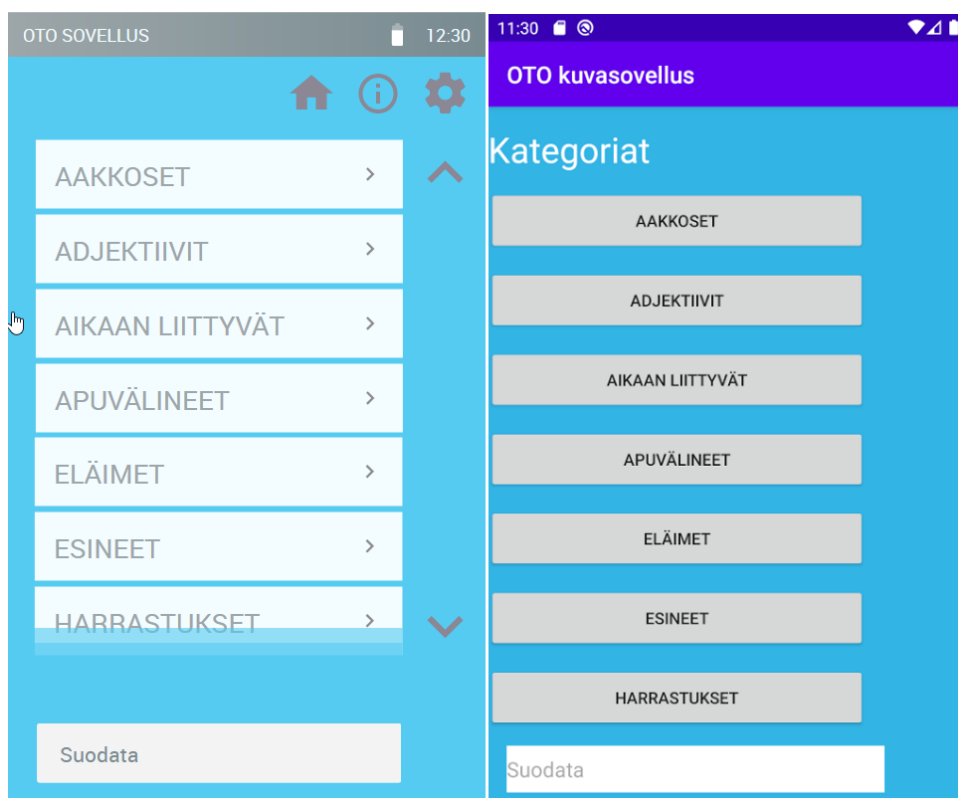
Selaa -sivulla on myös alalaidassa suodata -tekstikenttä, kun tätä napsautetaan, aukeaa ruudulle näkyviin Android-järjestelmän näppäimistö, johon voi syöttää kategorian nimen.

4.5 Graafinen ilme

Sovelluksen graafiseen ilmeeseen ei painosteta päättötyötä tehdessä paljoakaan, koska kyseessä on prototyypisovellus, eikä sen ole tarkoitus päätyä laajempaan käyttöön. Silti sovelluksen käytön selkeys on jo tässä

vaiheessa ensisijaisen tärkeää, sillä sen korjaaminen jälkikäteen voi olla hyvinkin haastavaa.

Sovelluksen kehityksessä on tarkoitus hyödyntää Android-kehityksen parhaita käytäntöjä, joilla taataan sovelluksen selkeä ja helposti ymmärrettävä ilme. Sovellukseen ei ole tarkoitus lisätä mitään, mikä voisi vetää huomiota pois tärkeästä sisällöstä eli toiminnanohjauskuvista.



Kuva 5 Kategoriat näkymä suunniteltu ja toteutus

5 PROTOTYYPPI, SOVELLUSVERSIO 1

Android Studio kehitysympäristössä luotiin kaksi prototyyppiversiota sovelluksesta. Tässä luvussa käydään ensimmäiseen versioon kehitetystä käyttöliittymästä ja ominaisuuksista.

5.1 Sovelluskehitys Android Studiossa

Android Studiolla sovelluksen kehitys alkaa uuden projektin luomisesta, kun uutta projektia luodaan, sovellus luo automaattisesti kaikki sovelluspaketin tarvitsemat tiedostot. Näihin sisältyy käyttöliittymäsuunnittelua varten tarvittava alkuaktiviteetti ja siihen liittyvä `activity_main.xml` -tiedosto. Aluksi näkymässä on vain valkoinen tausta ja ”Hello world!” -tervehdysteksti. Nämä tulee korvata itse suunnitellulla käyttöliittymällä.

Käyttöliittymään lisätään elementtejä joko graafisen liittymän kautta, raahaamalla elementit näkymään ja muokkaamalla niiden ominaisuuksia, tai suoraan kirjoittamalla XML-koodia `activity_main.xml` -tiedostoon. Helpompi tapa on tietysti käyttäen graafista käyttöliittymää, jota hyödynnettiin prototyyppien rakennettaessa.

Kaikille käyttöliittymään lisättäville elementeille, oli ne sitten kuvia, tekstikenttiä tai painikkeita tulee antaa uniikki ID-tunniste, jonka avulla niihin voi viitata ohjelman koodissa. Tämän lisäksi elementeille tulee määritellä ominaisuuksia kuten elementin koko, sijainti ja suhde muihin elementteihin. Sovelluksessa painike elementit liitettiin toisiinsa, siten että niiden suhteellinen etäisyys toisiinsa ja näytön reunoihin pysyy näytön koosta riippumatta.

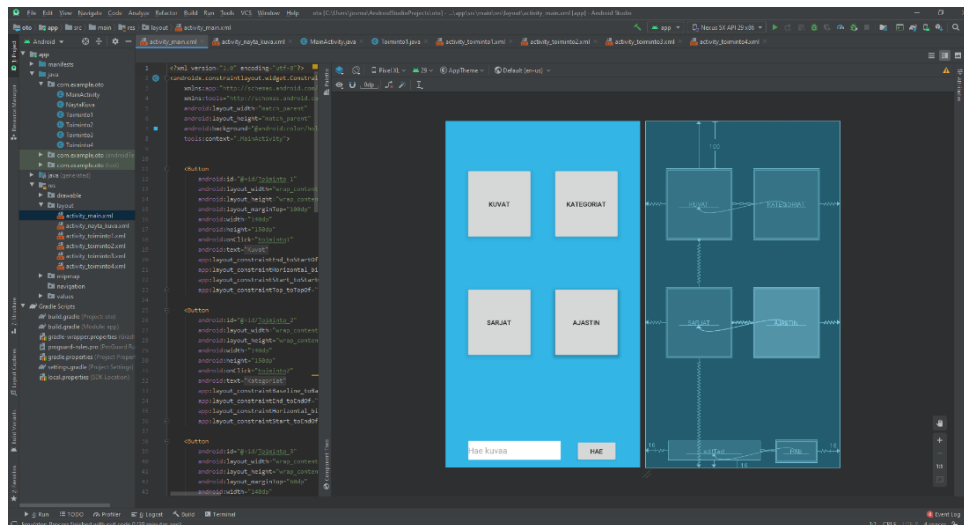
Ohjelman toiminnallisuus kirjoitetaan Java-koodilla `MainActivity`-tiedostoon. Tiedostossa myös tuodaan ohjelman käyttämät koodikirjastot ohjelmaan. Jokaista sovelluksen näkymää vastaa oma XML- ja Java-tiedosto, jotka pitävät sisällään näkymän käyttöliittymän ja ohjelmakoodin.

Ohjelman käytössä olevat tiedostot tulee tuoda projektiin, ne säilytetään ohjelman kansiorakenteessa `res\drawable` -hakemiston alla. Jotta sovellukseen tulevia kuvia voi käyttää sovelluksessa, täytyy niihin viitata koodissa alkuviitteellä `R.drawable.tiedostonimi`, koodissa ei ole tarpeen käyttää tiedostopäätettä.

5.2 Alkunäkymä

Ensimmäisessä versiossa aloitusnäkyssä oli navigointivalikko, sekä hakukenttä. Tarkoituksena oli käyttää suuria selkeitä navigointipainikkeita sovelluksen sisällä eri ruutujen väliseen siirtymiseen. Ruudun alalaidassa

olevaa hakukenttää ja painiketta oli tarkoitus käyttää esitettävien kuvien hakemiseen.



Kuva 6 Sovelluksen alku näkymä Android Studiassa

5.3 Kuvat-näkymä

Sovelluksen ensisijainen toiminto on kuvakorttien esittäminen. Sovelluksen päävalikosta pääsee siirtymään kuvat-toimintoon ensimmäisestä painikkeesta. Kun painiketta on painettu, tulee näkyville luettelo prototyyppi-sovellukseen lisätyistä kuvista. Kuvia tulee kolme kappaletta riville. Pienet kuvat toimivat samanaikaisesti esikatselukuvina ja painikkeina siirtymiselle kuvan katselu-toimintoon.

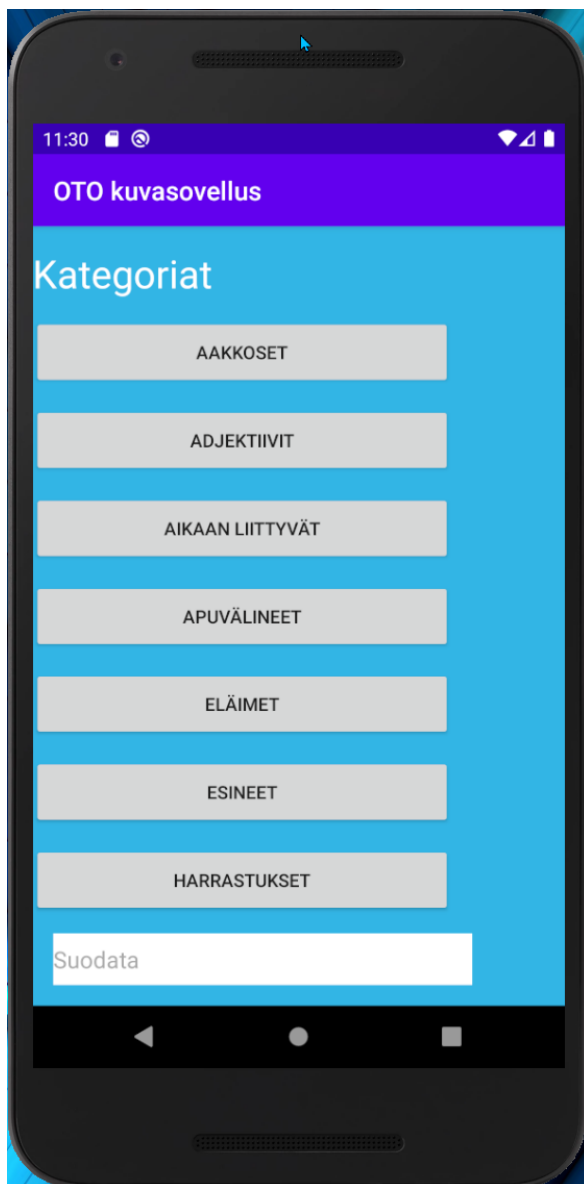
Kuvat-näkymässä näytettävät kuvat lajitellaan kategorioiden aakkosjärjestykseen. Kun näkymään siirrytään kuvat -painikkeen kautta, näkyville tulee kaikki sovellukseen lisätyt kuvat. Ruudulla näytetään niin monta riviä, kun esikatselukuvia mahtuu kerrallaan näkyviin. Näkymää vieritetään sormella ylöspäin, jolloin näytettävät kuvat ruudulla vaihtuvat. Kun haluttua kuvaa napsautetaan sormella, siirrytään näyttämään ruudulla vain kuvaa.

5.4 Kategoriat-näkymä

Kategoriat sivulla näytetään lista aiheista, joihin kuvia voidaan rajata. Valitsemalla näkymästä jokin kategoria siirrytään kuvat näkymälle, tällä kertaa näytölle tulee vain valitun kategorian kuvat. Toiminnon tarkoitus on rajoittaa esitettäviä kuvia tiettyyn aihealueeseen ja näin helpottaa sopivan kuvan löytämistä sovelluksesta.

Myös kategoriat sivun alalaidassa on hakukenttä, tämän avulla voidaan rajata näytettäviä kategorioita helposti. Kun kenttään alkaa kirjoittaa tekstiä, piilotetaan käyttöliittymästä painikkeet, jotka eivät sisällä syötettyä tekstiä. Kun painike poistuu näkymästä, siirtyvät sen alla olevat painikkeet

täyttämään vapautuneen tilan. Näin käyttöliittymä ei hajoa, vaan pysyy helppokäyttöisenä.



Kuva 7 Sovelluksen kategoriat näkymä emulaattorissa

5.5 Sarjat-näkymä

Sarjat näkymässä sovelluksen käyttäjä voi näyttää ja luoda itse useamman kuvakortin sarjoja. Sarjojen tarkoituksena on linkittää toisiinsa liittyviä kuvakortteja loogiseksi sarjaksi, jolloin voidaan opastaa lasta monivaiheisessa toiminnassa.

Kun sarjaa esitetään ruudulla, on näkyvissä yksi kuvakortti kerrallaan. Seuraavaan kuvakorttiin siirrytään pyyhkäisemällä sormella ruutua vasempaan tai oikeaan, riippuen halutaanko siirtyä edelliseen vai seuraavaan kuvaan.

5.6 Ajastin-näkymä

Sovellukseen suunniteltiin lisättävän ajastin toiminto, joka liittyisi kuvakortin esittämiseen. Toiminnon tarkoituksena on antaa huoltajalle mahdollisuus määritellä aika jossa kuvakortin ohjaama toiminto tulisi suorittaa. Toiminnanohjaus sovelluksen suunnitteluvaiheessa visioitiin, että kuvakortin päälle lisättäisiin jonkinlainen visuaalinen elementti, joka selkeästi havainnollistaisi lapselle ajan kulkua, mutta jättäisi kuitenkin kuvakortin helposti näkyviin.

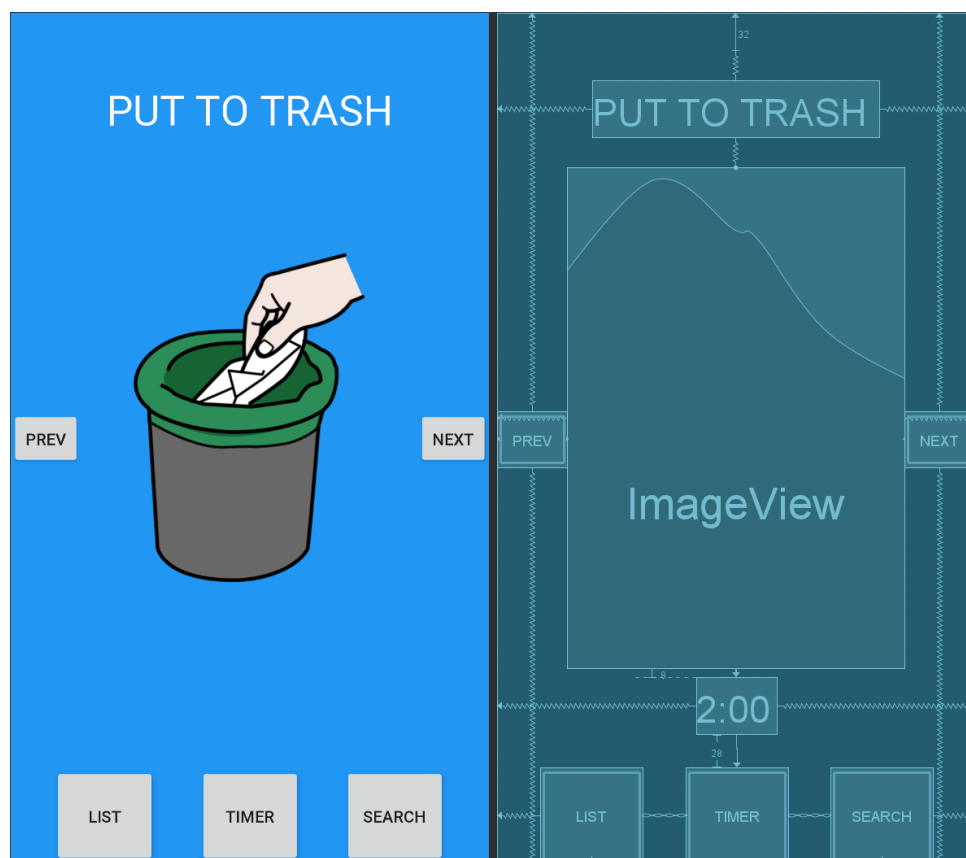
Tämä toiminnallisuus jäi lopulta toteuttamatta ensimmäiseen prototyyppiversioon. Toiminnon lisäys jää kuitenkin suunnitelmiin myöhempää sovellusversiota varten.

6 PROTOTYYPPI, SOVELLUSVERSIO 2

Tässä luvussa käydään läpi toisen prototyypin kehitystä ja vaiheita. Ensimmäisen prototyypin ongelmien vuoksi päädyttiin luomaan kokonaan uusi prototyypin versio sovelluksesta uutena Android Studio-projektina.

6.1 Päänäkymä

Tässä prototyypin versiossa keskityttiin saamaan sovelluksen käyttäjä, heti sovelluksen avattuaan suoraan toimintaan. Sovelluksen päänäkymä koostuikin heti esitettävästä kuvakortista. Tämä kuvakortti oli prototyypin versioon ennalta valittu ja siinä käytettiin tilantäyttäjänä toimintokorttia, jossa roska laitetaan roskikseen. Tarkoitus olisi, että käyttäjä voisi itse valita mikä kuva näkyy ensimmäisenä, kun sovelluksen avaa tai että siihen aukeaa viimeksi käytetty kuva lopullisessa sovellusversiossa.



Kuva 8 Päänäkymä v2 sovelluksessa

Päänäkymässä kuvakortin yläpuolella on myös kuvan toimintaan liittyvä opasteksti, joka liittyy kuvakortin toimintaan. Esimerkki kuvaan liittyvä

teksti olisi tässä tapauksessa ”Heitä roskeen”. Kuvan molemmilla puolilla on painikkeet kuvan vaihtamiseksi, edellinen ja seuraava. Näistä napsauttamalla ruudulla näytettävä kuva ja teksti vaihtuvat. Sovelluksessa pysytään kuitenkin samalla ruudulla koko ajan, vain sisältö vaihdetaan. Toiminto on ohjelmoitu Java -ohjelmointikielellä.

Päänäkymässä on ruudun alalaidassa kolme toimintopainiketta, lista, ajastin ja haku. Prototyypisovelluksessa näistä kahdella siirryttäisiin toiselle sivulle sovelluksen sisällä ja ajastin painike tuo näkyville kuvakortin alle ajastimen.

Toisessa prototyypissä pyrittiin vähentämään sovellussivujen määrää mahdollisimman pieneksi. Tarkoituksena oli selkeyttää navigointia sovelluksessa sekä saada suorituskykyä optimoiduksi. Tämän takia sovelluksen ensimmäistä ruutua käytetään sovelluksen päätarkoitukseen ja pyrittiin saamaan siihen mahdollisimman paljon sovelluksen toimivuudesta.

Lista ja hakunäkymät vastaavat edellisen prototyypiversion sivuja, eikä niitä käydä tässä luvussa sen tarkemmin läpi.

6.2 Kuvan vaihtaminen sovelluksessa

Kuvan ja siihen liittyvän tekstin vaihtaminen, siirtymättä toiselle ruudulle sovelluksessa, toteutettiin listojen avulla. Luotiin kaksi listaa, joista toinen sisältää viittaukset sovelluksen sisälle importoituihin kuviin. Nämä listattiin viitaten niiden sovelluksen sisäiseen osoitteeseen, esim. R.drawable.ajaa_pyoralla joka viittaa toimintakuvaan, jossa on polkupyörän kuva. Sovelluksessa on toinen lista, jossa on seliteteksti jokaiselle kuvalle, polkupyöräkuvalla on selitetekstinä ”Ajaa pyörällä”. Molemmissa listoissa opasteteksti ja siihen liittyvä kuva ovat samassa järjestyksessä. Eli polkupyörällä ajaminen toimintoon liittyvällä kuvalla ja tekstillä on järjestysnumerona nolla.

Kun sovelluksessa painaa edellinen tai seuraava -painiketta, ohjelman koodi tarkistaa sillä hetkellä olevan toiminnon järjestysnumeron opasteteksti kentästä. Edellinen painiketta painaessa, vähennetään luvusta yksi, paitsi jos se on nolla, jolloin luvuksi asetetaan listan viimeisen kuvan luku. Seuraava painike toimii samalla logiikalla, paitsi kun saavutaan listan viimeiseen lukuun, asetetaan luvuksi nolla ja esitetään ruudulle listan ensimmäinen kuva. Näyte koodista 1, jolla siirrytään seuraavaan kuvaan.

```
public void nxtImage(View v) {
    image = (ImageView) findViewById(R.id.img_focus);
    text = (TextView) findViewById(R.id.txt_todo);
    Button button = (Button) findViewById(R.id.btn_next);
    image.setImageResource(imageId[kuvid]);
    text.setText(texts[kuvid]);
    if (kuvid == 5) {
        kuvid = 0;
    }
    else {
        kuvid++;
    }
}
```

Koodi 1

6.3 Käyttöliittymän suunnittelu

Sovelluksen käyttöliittymän ja toimivuuden suunnittelu Android Studio-kehitysympäristössä kulkevat lähekkäin. Sovelluksen toimivuuden kannalta on oleellista, että ruudulla näytettävät elementit ovat oikeilla kohdilla eri kokoisilla ruuduilla. Sovelluksen prototyypit suunniteltiin Android Studiossa ja testattiin sen vierelle asennetulla virtuaalisella Nexus 5X -puhelimella, jonka ruudun koko on 5.2 tuumaa ja näytön tarkkuus on 1920 x 1080 pikseliä. Tämä vastaa noin keskivertopuhelimen kokoa.

Käyttöliittymän kaikki elementit sidottiin toisiinsa ketjuilla, eli niiden välinen etäisyysuhde pysyy näytöllä toisiinsa verrattuna saman suhteisena. Vaikka prototyyppisovellus suoritettaisiinkin laitteella, jossa on suurempi näyttö, pysyisivät elementit oikeilla paikoillaan. Suurempi näyttö tai kuvataarkkuus saattaisivat aiheuttaa, hieman ylimääräistä tyhjää tilaa elementtien väliin.

Osa ruudulla näkyvistä elementeistä sidottiin tiettyyn sijaintiin. Sovelluksessa näkyvä kuva on aina keskellä ruutua, siirtymispainikkeet ovat aina kuvan vasemmalla ja oikealla puolella melkein ruudun reunassa kiinni ja toimintapainikkeet löytyvät ruudun alalaidasta.

Käyttöliittymässä on tilaa myös ajastimelle, joka ei ole oletuksena näkyvissä. Vain jos painaa ruudulla olevaa ajastin -painiketta, tulee se näkyviin toimintakorttikuvan alle.

6.4 Jatkokehityskohteet

Prototyypisovelluksessa keskityttiin pääasiassa käyttöliittymän suunnitteluun sekä perustoimivuuden testaamiseen. Jatkokehittettäviä kohteita sovellukselle prototyypiversion perusteella löytyy useita. Sovelluksen kautta on tarkoitus pystyä hyödyntämään laajaa kuvakirjastoa. Prototyypiversion tallennettiin yksinkertaisena listana vain muutamia kuvia. Toimivassa sovelluksessa pitäisi kiinnittää enemmän huomioita kuvakirjaston hallittavuuteen. Yksinkertaiset listat eivät ole toimiva keino tiedon hallittavuuden kannalta.

Ratkaisu tähän haasteeseen olisi luoda sovellukseen XML-tiedostomuotona kirjastot kuvakorteista. Jokaisella toimintokortilla olisi tiedoissaan ID, tiedostosijainti, korttiin liittyvä opasteteksti sekä kategoriat joihin kuva sisältyy. Sovelluksesta löytyviin lista-, esikatselu- ja hakunäkymiin voisi hakea sisällön näiden tietojen perusteella. Toiminto oli kuitenkin niin laaja, että se rajattiin prototyypisovelluksen toteutuksen ulkopuolelle.

Prototyypisovelluksen ulkonäkö ja käyttöliittymä ovat myös melko pelkistetyt, tämä on osittain tarkoituksenmukaista, jotta sovelluksen kohteen huomio saataisiin suunnattua toimintokorttiin. Tässä olisi kuitenkin vielä varaa jatkokehitykselle. Yksi idea on, että jos ruudulla näkyvää kuvaa napsauttaisi sormella, siirryttäisiin esittämään kyseistä toimintakorttia ja siihen liittyvää opastetekstiä kokoruudun koolta. Tästä näkymästä voisi palata napsauttamalla sormella kuvaa uudelleen. Tätä kokonäytön tilaa voitaisiin myös hyödyntää, kun käytetään ajastintoimintoa. Kohteen huomiota ei häirittäisi sovelluksen toimintaan liittyvillä painikkeilla, vaan vain oleellinen olisi näkyvissä.

7 YHTEENVETO

Sovelluskehityksessä on hyvin tärkeää tiedostaa heti alkuvaiheessa suunnitellun sovelluksen laajuus ja käyttötarkoitus. Ensisijaisesti näiden tekijöiden tulisi ohjata sovelluskehitystä oikeaan suuntaan. Jos näihin tekijöihin ei kiinnitä heti alkuvaiheessa riittävästi huomiota voi helposti käyttää aikaa ja resursseja epäoleellisten toimintojen tai ominaisuuksien kehittämiseen.

Android-sovelluksen kehittäminen poikkeaa yllättävän paljon ”tavallisten” tietokonesovellusten kehittämisestä. Suunnitelman kannalta lähestyminen on lähestulkoon sama, mutta prototyyppisovelluksen kehityksessä tuli hyvin paljon uusia asioita esiin ja yllätyin uuden opittavan määrystä.

Prototyyppisovelluksen tekeminen opetti myös paljon Android-sovelluskehityksestä, moni asia nojautuu ennalta määriteltyihin rakenteisiin ja tapoihin. Voidaan hyödyntää paljon valmiita kirjastoja esimerkiksi sovelluksen navigoinnin rakentamiseksi. Silti vaikka nämä kirjastot ovat olemassa, niiden hyödyntäminen omassa sovelluksessa ei ole välittömästi yksinkertaista tai helppoa.

Kun ensimmäisen prototyypin kehityksessä päästiin navigoinnin suunnitteluun, huomattiin ongelmia käytettävyydessä. Sovellus ei ollut suunnitellun mukaisesti selkeä ja helppokäyttöinen. Sovelluksen käyttötarkoituksen mukainen nopea pääsy kuvan esittämiseen tilanteen vaatiessa ei toteutunut halutulla tavalla.

Toinen prototyyppiversio oli toiminnallisesti laajempi ja siinä toteutui suunnitelman mukainen nopea pääsy kuvan esittämiseen, lisäksi toisen prototyyppiversion käyttöliittymä tuntui selkeämmältä ja helppokäyttöisemmältä. Jatkokehitykselle jäi kuitenkin vieläkin tilaa.

LÄHTEET

HUS. (n. d.) Toiminnanohjaus. Haettu 19.4.2020 osoitteesta <https://www.hus.fi/sairaanhoito/sairaanhoitopalvelut/foniatria/lapsen-neuropsykologinen-arvio/toiminnanohjaus/Sivut/default.aspx> (1)

HUS. (n. d.) Toiminnanohjaus. Haettu 19.4.2020 osoitteesta <https://www.hus.fi/sairaanhoito/sairaanhoitopalvelut/foniatria/lapsen-neuropsykologinen-arvio/toiminnanohjaus/Sivut/default.aspx> (2)

Lehtisare (n.d.) Toimiva arki visuaalisin keinoin opas. Haettu 19.4.2020 osoitteesta <https://www.tampere.fi/sosiaali-ja-terveyspalvelut/lapsiperheiden-palvelut/nepsy/selviytymiskeinot/oppaat.html>

Sommerville, I. (2016) *Software Engineering, Tenth edition* Essex, England: Person Education Limited (1)

Sommerville, I. (2016) *Software Engineering, Tenth edition* Essex, England: Person Education Limited (2)

Sommerville, I. (2016) *Software Engineering, Tenth edition* Essex, England: Person Education Limited (3)

Murphy, M. (2011) *The Busy Coder's Guide to Android Development*, USA: CommonsWare (1)

Murphy, M. (2011) *The Busy Coder's Guide to Android Development*, USA: CommonsWare (2)