



Tarkkaavaisuuspainotteinen tehtävienhallintasovellus

Tero Lahti

OPINNÄYTETYÖ
Toukokuu 2022

Tieto- ja viestintäteknikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintäteknikka
Ohjelmistotekniikka

LAHTI, TERO:

Tarkkaavaisuuspainotteinen tehtävienhallintasovellus

Opinnäytetyö 22 sivua
Toukokuu 2022

Opinnäytetyön tarkoituksena oli kehittää Android-pohjainen mobiilisovellus, jonka avulla voisi helpottaa tarkkaavaisuusongelmista kärsivien elämänhallintaa, erityisenä painotuksena ADHD. Opinnäytetyöntekijän kokemus- ja tietopohjaa aiheesta täydennettiin lukuisista eri lähteistä koostettujen tietojen avulla. Työn tavoitteina olivat myös uuden oppiminen sekä yksinkertainen ja jatkokehitettävyydeltään hyvä sovellustoteutus.

Opinnäytetyössä perehdyttiin Googlen luomiin Dartiin, Flutteriin ja Firebaseen, jotka ovat tehokkaita, nykyaikaisen ohjelmistokehityksen vaatimuksia noudattavia työkaluja. Niiden avulla kehitettiin helppokäyttöinen mobiilisovellus, joka tarjoaa sekä käyttäjäystävällisen tehtävälistan että yksinkertaisen keskittymisajastimen. Toimintoja ei haluttu tehdä liian monipuolisiksi, koska monimutkaisuus voisi korottaa sovelluksen käyttöönottokynnystä, etenkin sen pääasialliseksi aiotussa kohderyhmässä. Erityisesti jatkokehityksen kannalta ajatellen toteutettiin sovellukseen myös mahdollisuus kirjautua Google-tilin avulla.

Opinnäytetyön tuloksena saatiin yksinkertainen ja toimiva tehtävienhallintasovellus, jonka kehitysprosessi antoi työlle lisäarvoa. Sovelluksella on myös erinomaiset jatkokehitysmahdollisuudet, kuten vaikkapa käyttötilastoihin perustuvan käyttäjäpalautteen toteuttaminen. Opinnäytetyö tarjoaa sellaista tietoa, jonka avulla ohjelmistokehittäjän on mahdollista kehittyä modernien mobiilisovellusten luomisessa sekä oppia huomioimaan keskittymisvaikeuksista kärsivät käyttäjät paremmin.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Software Engineering

LAHTI, TERO:
Attention-Focused Task Management Application

Bachelor's thesis 22 pages
May 2022

The purpose of the thesis was to develop an Android-based mobile application to help people with attention problems, particularly ADHD, manage their lives. The experience and knowledge base of the thesis researcher was supplemented with information from various sources. The objectives of the thesis were also to learn new things and to develop a simple and scalable application.

The technologies the thesis focuses on, are Google's Dart, Flutter and Firebase, which are powerful tools that meet the requirements of modern software development. They were used to develop an easy-to-use mobile application that provides both a user-friendly to-do list and a simple focus timer. The aim was not to make the functions too complex, as complexity could increase the adoption threshold, especially for the intended main target group. For further development, the possibility to log in using a Google account was also implemented.

The thesis resulted in a simple and functional task management application, the development process of which added value to the thesis. The application also has excellent potential for further development, such as the implementation of user feedback based on usage statistics. The thesis provides information that will enable software developers to improve their skills in creating modern mobile applications and to learn how to better cater for users with concentration difficulties.

Key words: ADHD, Android, Dart, Flutter

SISÄLLYS

1	JOHDANTO	6
2	AKTIIVISUUDEN JA TARKKAAVUUDEN HÄIRIÖ	7
3	VALITUT TEKNIIKAT.....	10
3.1	Dart	10
3.2	Flutter.....	10
3.3	Firestore	11
4	MOBIILISOVELLUS.....	12
4.1	Sovelluksen rakenne ja toiminta.....	12
4.2	Koodiesimerkkejä.....	18
5	YHTEENVETO JA POHDINTA.....	21
	LÄHTEET.....	22

LYHENTEET JA TERMIT

ADHD	Attention Deficit Hyperactivity Disorder, eli aktiivisuuden ja tarkkaavuuden häiriö
Android	Linux-ytimeen perustuva käyttöjärjestelmä, joka on erittäin yleinen älypuhelimissa, tableteissa yms. älylaitteissa
Dart	Googlen kehittämä tehokas, monialustaiseen ohjelmistokehitykseen tehty ohjelmointikieli
Firebase	Googlen pilvipohjainen backend-palvelu mobiili- ja verkkosovellusten kehitykseen ja käyttöön
Flutter	Dart-pohjainen sovelluskehitysteknologia
JSON	JavaScript Object Notation on selkokielen, avoimen standardin tiedostomuoto
NoSQL	Not only SQL. Perinteisistä relaatiotietokannoista eroava, hyvin skaalautuva tietokantatyyppejä
SDK	Software Development Kit, eli ohjelmistokehityspaketti sisältää joukon ohjelmistokehityksen työkaluja yhdessä paketissa
Widget	Eräs ohjelmistokomponentti, Flutterilla tehdyt sovellukset muodostuvat näistä komponenteista

1 JOHDANTO

Julkinen keskustelu erilaisista tarkkaavaisuuden ja keskittymisen häiriöistä on yleistynyt. Useat tunnetut henkilöt ovat viime vuosina julkisesti kertoneet siitä, miten esimerkiksi aktiivisuuden ja tarkkaavuuden häiriö ADHD on vaikuttanut heidän elämäänsä. Teknologiayhteiskunta on synnyttänyt sosiaalisen median ja pikaviestintäsovellusten kaltaisia lieveilmiöitä. Näiden digitaalisten mediapalveluiden on havaittu neurotyypillisillä ihmisillä aiheuttavan samankaltaisia oireita, kuin ADHD-diagnoosin saaneilla on. (Ra, Cho & Stone. 2018) Onkin loogista, että ADHD-diagnoosin saaneilla on havaittu saman ilmiön aiheuttamaa oireiden pahenemista. (Shuai ym. 2021)

Tämän opinnäytetyön tarkoitus oli lähestyä aihetta siellä, missä digitaalista mediaa kulutetaan. Erilaiset mobiililaitteet, kuten älypuhelimet ja tabletit ovat yleistyneet mittavasti, ja niistä on tullut monille sekä viestinnän että viihteen keskuksia. Ongelmia on syytä hoitaa siellä, missä ne ilmenevät, ja siksi opinnäytetyössä kehitettiin mobiilisovellusta, jolla kohdata näitä haasteita tehokkaasti.

Tarve oli vahvasti sisäsyntyinen ja juonsi opinnäytetyön tekijän omasta, aikuisiällä saadusta ADHD-diagnoosista. Missään vaiheessa tarkoitus ei kuitenkaan ollut luoda täydellistä työkalua tai tehdä yhtäkään terveystavittämää sen toimintojen suhteen. Kantavana ajatuksena olikin luoda pohja sellaiselle ajanhallintaa ja keskittymistä tukevalle mobiilisovellukselle, jolla olisi kehityspotentiaalia myös tulevaisuudessa. Käytettävät tekniikat valittiin myös tehokkaan ja monialustaisen kehityksen näkökulmasta.

Erilaisia ajan- ja tehtävienhallintasovelluksia oli olemassa jo aiemmin, ja niitä tehdään varmasti myös tulevaisuudessa. Yleistä niille tuntuu kuitenkin olevan liiallinen monimutkaisuus, joka helposti kyllästyttää käyttäjänsä, etenkin, jos käyttäjällä on jo valmiiksi ongelmia keskittymiskyvyn ja tarkkaavaisuuden suhteen. Koska opinnäytetyö on kehitetty itsenäisesti, ilman sidonnaisuuksia mihinkään työnantajaan, sille ei myöskään mikään kaupallinen taho asettanut esivaatimuksia ja siten rajoittanut sen kehityssuuntaa tai yksinkertaisuuden ja selkeyden tavoitteita.

2 AKTIIVISUUDEN JA TARKKAAVUUDEN HÄIRIÖ

Opinnäytetyössä kehitetyn sovelluksen käyttötarkoituksen painopiste on erityisesti aktiivisuuden ja tarkkaavuuden häiriö eli ADHD. Se on kehityksellisen neuropsykiatrian yleisin häiriö (Kemi. 2019. s. 9), joka alentaa toimintakykyä muun muassa tarkkaamattomuuden kautta. ADHD:n esiintyvyys on lapsilla noin 3,6–7,2 prosenttia, ja aikuisuuteen siitä aiheutuva haittavaikutus säilyy 30–70 prosentilla (Kemi. 2019. s. 10.)

Vaikka monilla oireet lievenevät iän myötä, ovat lapsien tukemiseen suunnatut keinot yhtä lailla sovellettavissa aikuisten tukemiseen. Kuviossa 1 on esitelty luokanopettajan keinoja ADHD-oireisen oppilaan tukemiseen. Opinnäytetyössä kehitetty mobiilisovellus pyrkii vastaamaan näihin tarpeisiin soveltuvilta osin, ja autamaan mm. toiminnanohjauksessa.

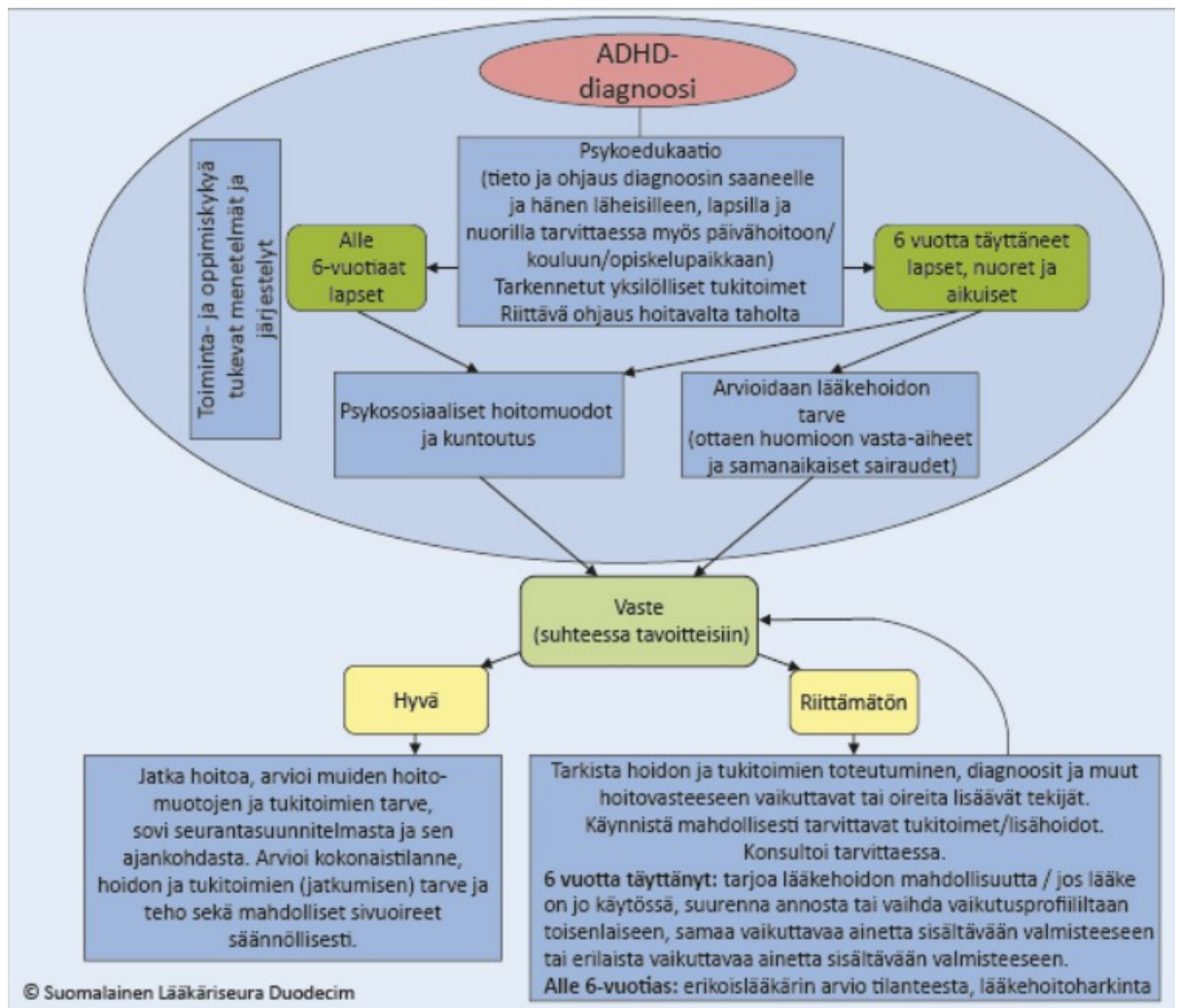
Luokanopettajan keinoja ADHD-oireisen oppilaan oppimisen ja koulunkäynnin tukemiseen	
Opetuksessa	Oppimisympäristössä
<ul style="list-style-type: none"> ▪ Katsekontakti ▪ Opettajan oma asenne ▪ Rauhallisuus ▪ Johdonmukaisuus ▪ Selkeä ja vaiheittainen ohjeistus ▪ Ymmärtämisen varmistaminen ▪ Toiminnanohjauksen tukeminen ▪ Opetuksen strukturointi ▪ Rutiinit ▪ Visuaalisuus ▪ Onnistumisten huomioiminen ▪ Kannustus ja palautteen antaminen ▪ Ennakointi ▪ Hienomotoriikan hyödyntäminen 	<ul style="list-style-type: none"> ▪ Mahdollisuus valita työskentelypaikka ▪ Monipuolinen oppimisympäristö ▪ Mahdollisuus liikkua ▪ Sopiva ärsykkeiden määrä ▪ Valaistus ▪ Keskittymisen apuvälineet

KUVIO 1. Luokanopettajan keinoja ADHD-oireisen oppilaan oppimisen ja koulunkäynnin tukemiseen. (Kemi. 2019. s. 42).

Erityisesti opinnäytetyössä kehitetyllä sovelluksella pyritään tuomaan rutiineja ja struktuuria käyttäjiensä elämiin. Tehtävälistan ja keskittymisajastimen avulla se kykenee toimimaan keskittymisen apuvälineenä, ja auttamaan abstraktioiden visualisoinnissa.

Koska ADHD ei ole sairaus tai muukaan parannettavissa oleva lääketieteellinen tila, täytyy sen kanssa yrittää tulla toimeen. ADHD:n selkeimpiä taustasyitä on poikkeavuus välittäjäaine dopamiinin säätelyssä. (adhdutuksi.fi. n.d.) Dopamiinin taso voi vaikuttaa mielialaan, tarkkaavaisuuteen, motivaatioon ja liikkeeseen. Dopamiinilla on rooli myös palkitsijana, kun tekee jotakin miellyttävää, sen määrä aivoissa kasvaa. (adhdfoundation.org.au. 2020.) Sosiaalinen media palkitsee käyttäjää erilaisilla reaktioilla tämän jakamiin sisältöihin, mikä saa aivojen palkitsemisvälittäjäaineet liikkeelle (chadd.org. n.d.). Koska sosiaalista mediaa käytetään usein mobiililaitteilla, voi samalla mobiilialustalla toimiva sovellus mahdollisesti auttaa oireiden hallinnassa.

Virallisten hoitosuosituksen mukainen hoito on toki aina ykkösprioriteetti, mutta yleisesti on hyvä olla monipuolinen valikoima apukeinoja käytettävissä. Kuviossa 2. on esiteltyä virallisten suositusten mukainen hoitokaavio. Siinä on pintapuolisesti kuvattuna ADHD-diagnosoidun hoidon rakennetta ja vuorovaikutuksellisuutta, missä sopiva mobiilisovelluskin voisi varmasti olla osana kokonaisuutta. Laajemmasta käytöstä on kuitenkin aina syytä keskustella omasta hoidosta vastaavan ammattilaisen kanssa.



KUVIO 2. Käypä hoito -työryhmän tekemä ADHD:n hoitokaavio. (kaypahoito.fi 2017)

3 VALITUT TEKNIIKAT

3.1 Dart

Dart on Googlen kehittämä, vuonna 2011 julkaistu ohjelmointikieli, joka kehitettiin nopeuttamaan ohjelmistokehittäjien työtä. Se on tehokkaasti optimoitu, monialustainen kieli, jonka syntaksi muistuttaa muun muassa C-kieltä, Javaa ja JavaScriptiä. Dart-koodi voidaan kääntää konekieliseksi, minkä ansiosta se on erittäin nopea alustasta riippumatta. Tämä mahdollistaa korkeaa suorituskykyä vaativien sovellusten kehittämisen eri käyttöjärjestelmiä ja prosessoriarkkitehtuureja käytäville laitteille yhden ja saman koodipohjan avulla.

Konekielisen käännöksen lisäksi Dart voidaan kääntää myös JavaScript-kielille, joka on ylivoimaisesti yleisin ohjelmointikieli internetissä (w3techs.com. 2022. 19.5.2022) Kääntämismahdollisuudella voidaan taata, että Dart-kielillä ohjelmoidut sovellukset toimivat hyvin web-alustalla käytettynä.

3.2 Flutter

Flutter on avoimeen lähdekoodiin perustuva SDK eli ohjelmistokehityspaketti, joka on tehty pääosin Dart-kielillä, joka on myös Flutterissa itsessään käytettävä ohjelmointikieli. Sen ensimmäinen vakaa versio on julkaistu vuonna 2018 (docs.flutter.dev. 19.5.2022.)

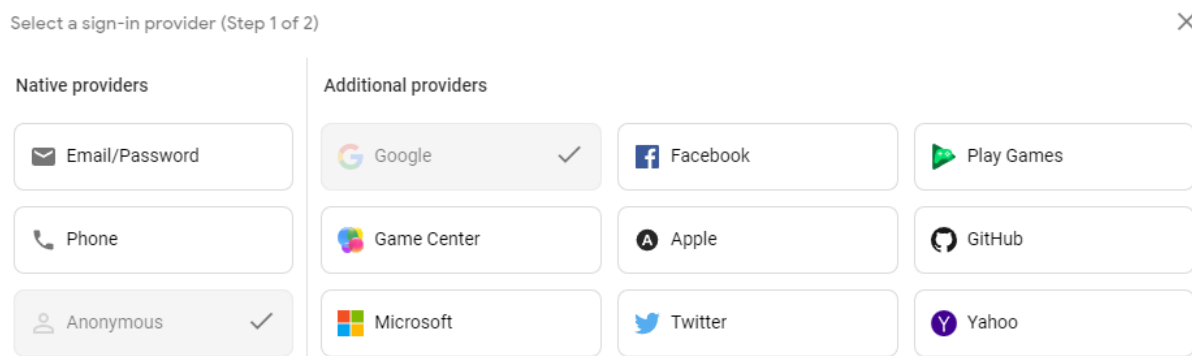
Flutter ammentaa pohjanaan olevan Dart-ohjelmointikielen tehokkuudesta mahdollistaen nopean ja tehokkaan ohjelmistokehityksen. Flutter on alun perin erityisesti suurimmille mobiilialustoille, eli Googlen Androidille ja Applen iOS:lle, kohdistuvaan ohjelmistokehitykseen suunniteltu. Mutta sen lisäksi, että yhden koodipohjan päältä voidaan tehdä pikselintarkasti valmistajiensa suunnittelukieltä noudattavia mobiilisovelluksia, tarjoaa Flutter paljon muutakin. Sillä voidaan tehdä myös Windows-, Linux-, macOS- ja Fuchsia OS-sovelluksia.

Flutter koostuu käytännössä widgeiteistä eli pienistä sovellus- tai käyttöliittymä-komponenteista. Ne ovat tavallaan eräänlaisia rakennuspalikoita, joista voi rakentaa erilaisia sovelluksia. Mutta koska Flutter perustuu avoimeen lähdekoodiin, voi kuka tahansa tarkastella, miten nuo palikat on rakennettu, ja muokata niitä omiin tarkoituksiinsa sopiviksi. Yhtä lailla Flutter-kehittäjä voi myös rakentaa omia widgeitejään ja siten tuoda sovellukseensa täsmälleen ne ominaisuudet ja toiminnot, jotka hän haluaa.

3.3 Firebase

Googlen Firebase on pilvipalveluihin perustuva alusta mobiili- ja web-sovelluksille. Se on nykyään integroitu moniin Googlen palveluihin, ja se tarjoaa muun muassa kaksi erilaista reaaliaikaisesti käytettävää NoSQL-tietokantavaihtoehtoa, joista toinen on JSON- ja toinen dokumenttipohjainen.

Firebase mahdollistaa pilvi-infrastruktuurin skaalautuvuuden käytännössä mihin kokoluokkaan tahansa. Se palvelee helposti niin yksittäisen harrastelijan kuin suuren monikansallisen yrityksenkin tarpeita. Tietokantapalveluiden lisäksi se mahdollistaa esimerkiksi käyttäjien autentikoinnit, markkina- ja suorituskyky-analytiikan, viestinnän ja erilaiset testaukset. Lisäksi Firebase on helposti integroitavissa Flutter-sovelluksen kanssa. Tarjolla on lukuisia kirjautumisvaihtoehtoja, joista sovellukseen valittiin kaksi (Kuva 1.)

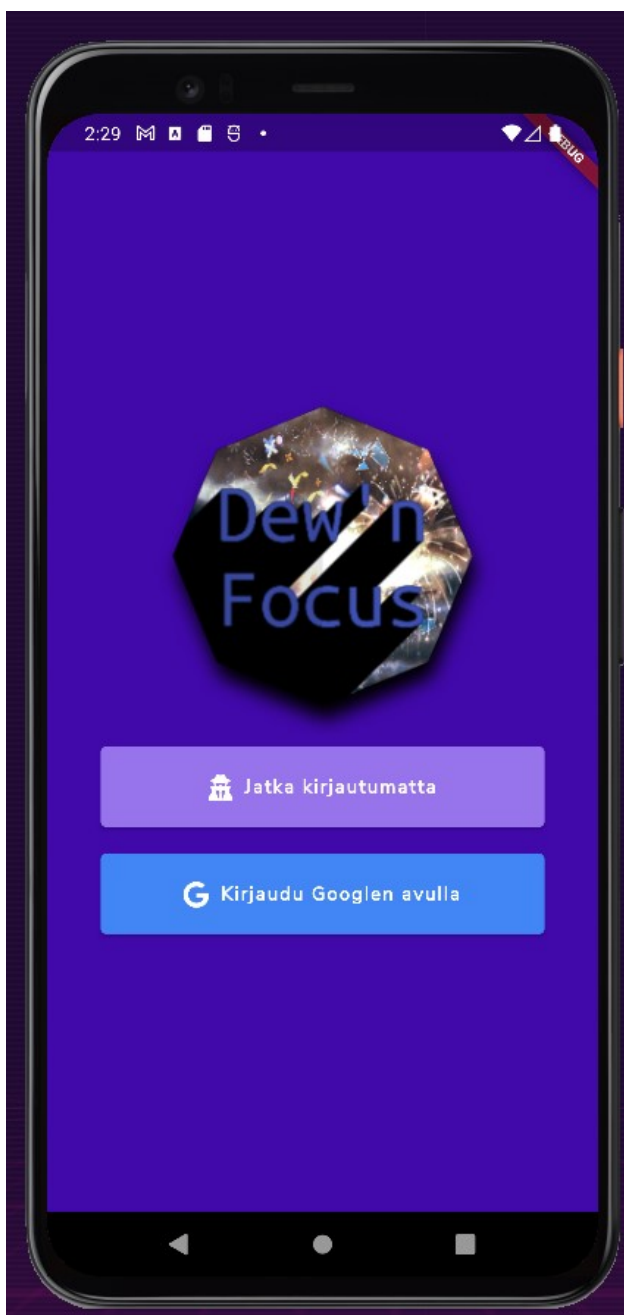


KUVA 1. Opinnäytetyössä kehitettyyn sovellukseen Firebasen hallintakonsolista valitut ja valittavissa olevat kirjautumisvaihtoehdot.

4 MOBIILISOVELLUS

4.1 Sovelluksen rakenne ja toiminta

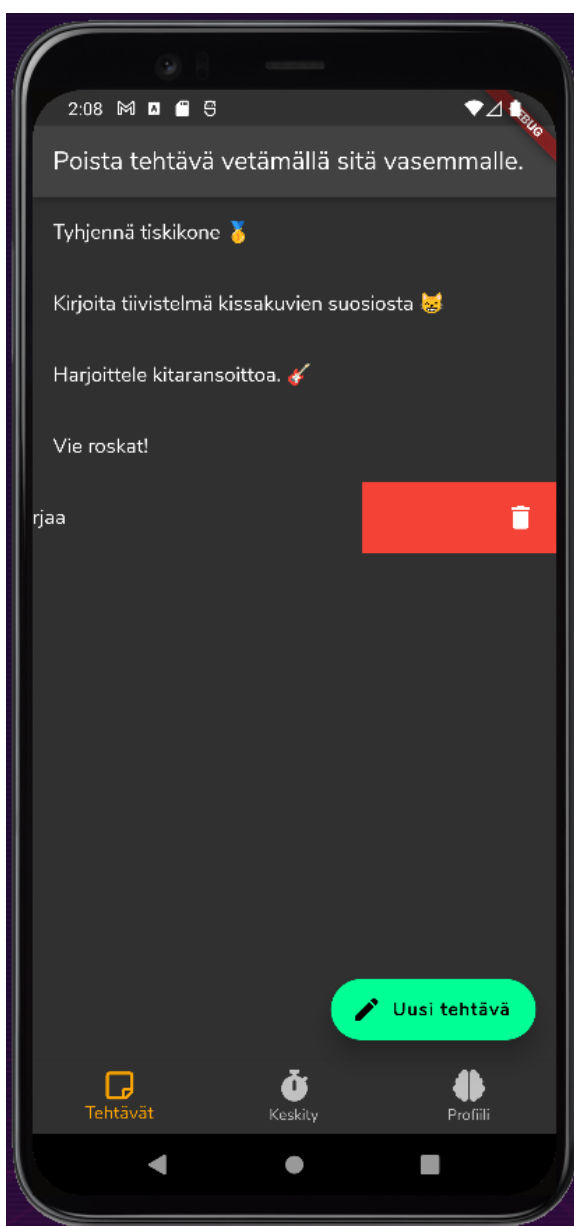
Alusta asti oli selvää, että sovellukseen tulisi tehdä kirjautumismahdollisuus, mutta sen ei tulisi olla pakollista. Tästä johtuen olikin luonnollista, että ohjelma käynnistyisi kirjautumisruutuun (Kuva 2.)



KUVA 2. Mobiilisovelluksen kirjautumisruutu. Kirjautumisvaihtoehdot anonyymi kirjautuminen ja Google-kirjautuminen.

Anonyymin kirjautumisen lisäksi oli luonnollista valita kirjautumistavaksi Google-tili. Tulevaisuudessa sovellukseen voidaan helposti lisätä muita kirjautumistapoja, mikäli asia katsotaan tarpeelliseksi. Mikäli sovellus halutaan saataville Applen sovelluskauppaan, täytyy myös Apple-tilillä kirjautuminen lisätä.

Kun käyttäjä on kirjautunut sisään, joko anonyymisti tai Google-tilin avulla, avautuu oletuksena tehtävälisteruutu, joka näkyy kuvassa 3. Tehtävälisteruutu on valittu ensimmäiseksi ruuduksi, koska ajatuksena on syöttää siihen tehtävät ja sitten tarpeen mukaan vaihtaa keskittymisajastimeen.



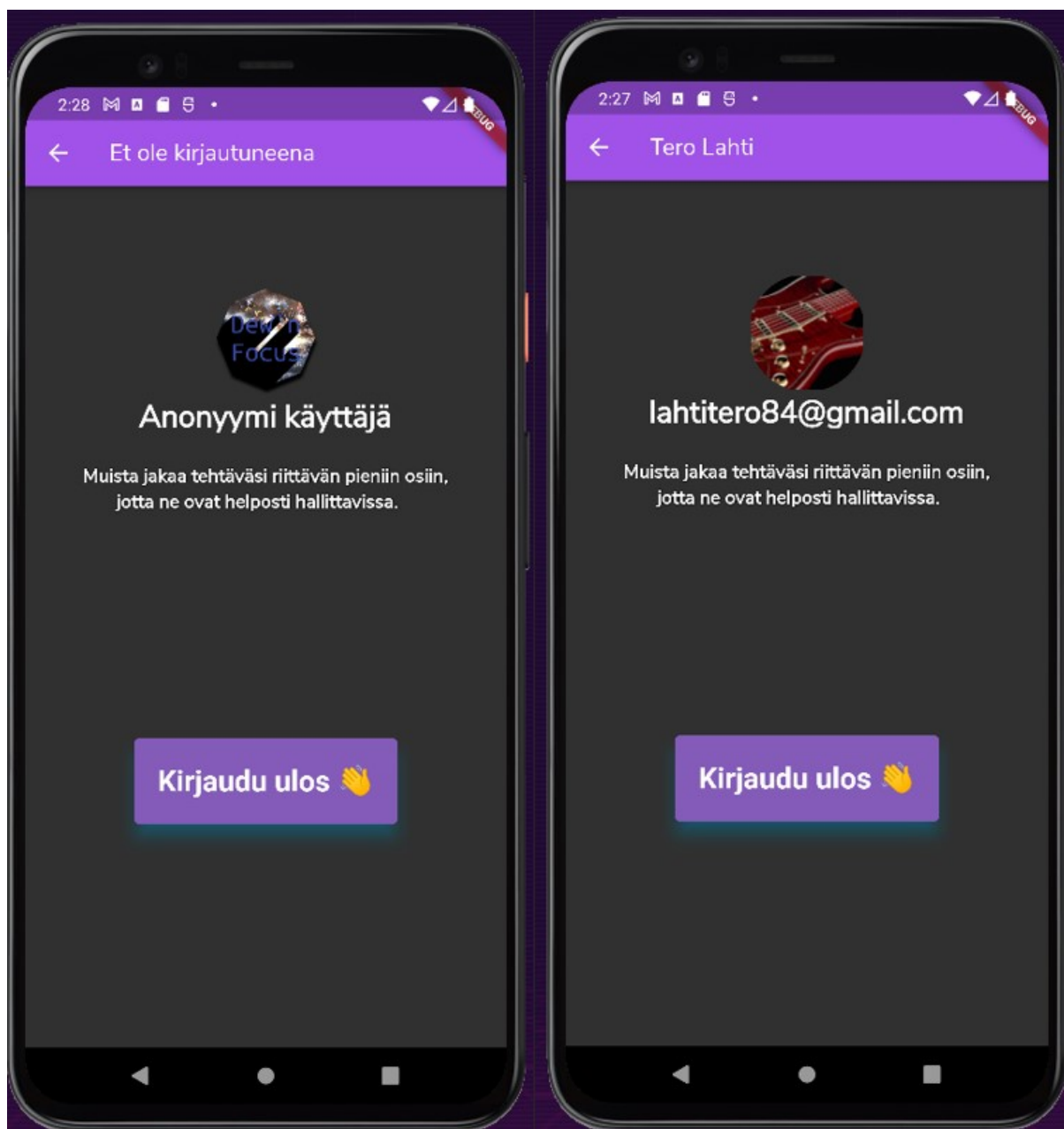
KUVA 3. Tehtävälisteruutu. Kuvassa näkyy erilaisia tehtäviä ja myös punainen palkki, jonka sisällä on valkoinen roskakorin kuva. Tuo palkki ilmaisee, että käyttäjä on juuri pyyhkäisemässä suoritettua tehtävää pois listalta.

Tehtävälistan valikkopalkissa on valittavissa myös kaksi muuta ruutua, jotka ovat keskittymisajastin ja profiili. Keskittymisajastin näkyy kuvassa 4. Se tarjoaa yksinkertaisesti 25 minuutin keskittymisajan, joka voidaan joko aloittaa kesken kaiken alusta tai nollata ja pysäyttää. On mahdollista, että siihen lisätään varoen ominaisuuksia jatkokehityksessä. Sovelluksen ei kuitenkaan tulisi ikinä tulla liian monimutkaiseksi, jotta ominaisuudet eivät vie liikaa käyttäjän huomiota.



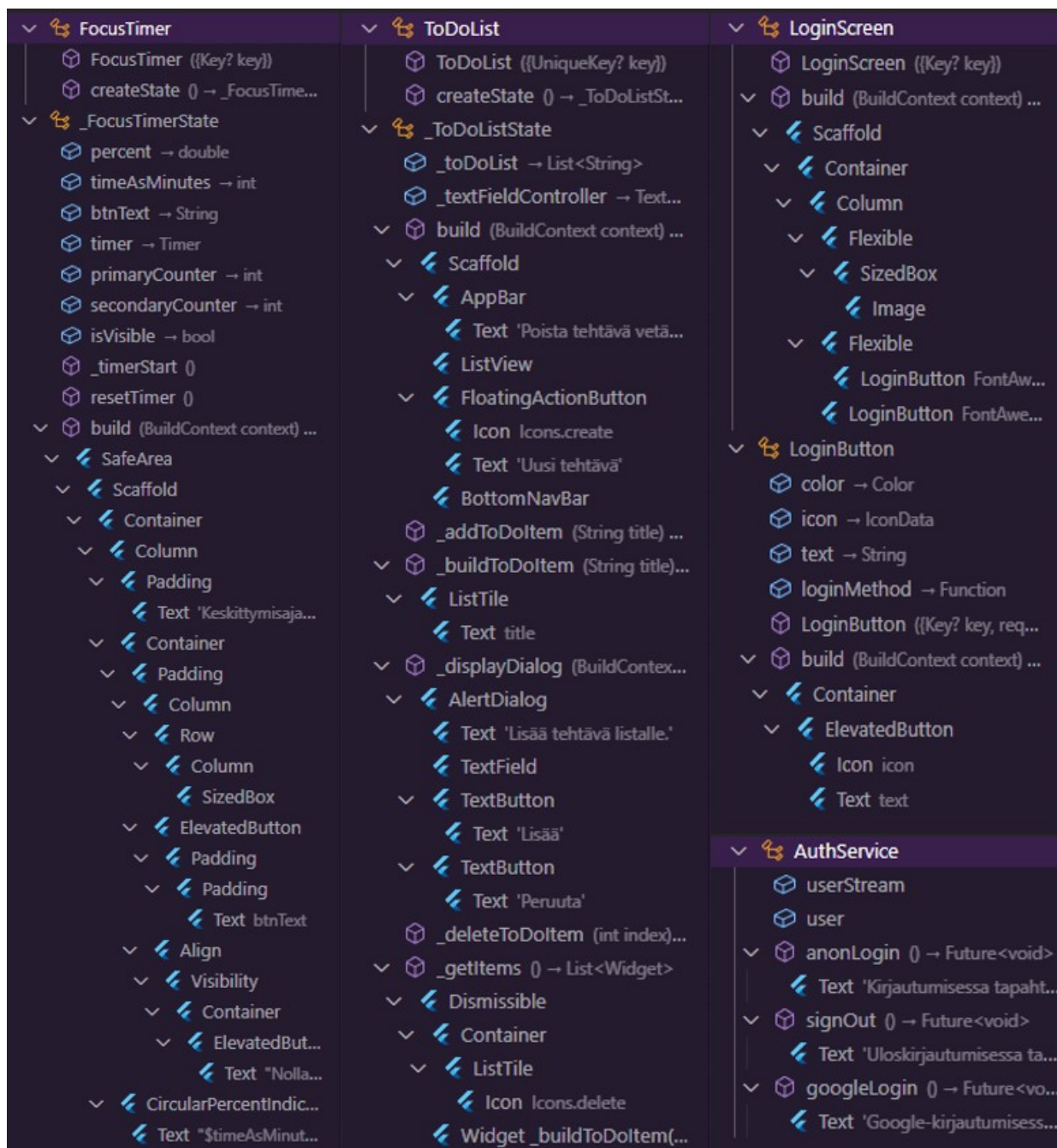
KUVA 4. Keskittymisajastin. Puoliympyrän muotoinen edistymispalkki täyttyy sinisellä sitä mukaa, kun aika kuluu.

Profiilinäkymässä (kuva 5.) näytetään joko Googlella kirjautuneen sähköposti-osoite ja kyseisen Google-tilin avatar-kuva tai sitten sovelluksen logo ja tieto siitä, että käyttäjä on anonyymi. Molemmissa tapauksissa ruutu tarjoaa myös mahdollisuuden kirjautua ulos sovelluksesta, jolloin päädytään takaisin kirjautumisruutuun.



KUVA 5. Profiiliruutu. Vasemmalla anonyymin käyttäjän profiiliruutu ja oikealla Googlella kirjautuneen käyttäjän näkemä ruutu. Jatkokehitysvaiheessa profiiliruutuun voidaan tuoda esimerkiksi käyttäjätilastoja tai lisää tekstimuotoisia neuvoja ja lainauksia.

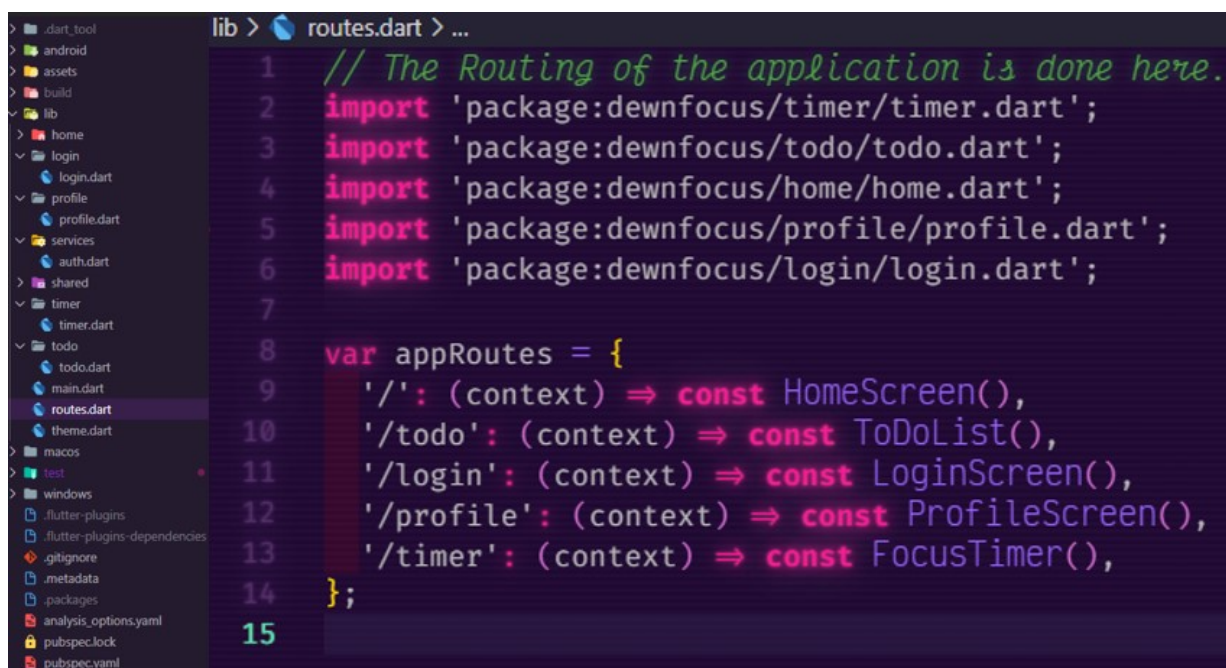
Ohjelmakoodin näkökulmasta Flutter-sovellus on käytännössä kokoelma widgettejä, jotka on koostettu sovelluksen toiminnan kannalta sopivaan puumaiseen hierarkiaan. Tätä on esitelty kuvassa 6.



KUVA 6. Sovelluksen joidenkin osien hierarkkinen rakenne visualisoituna.

Vaikka kyseessä on verrattain yksinkertainen sovellus, löytyy pinnan alta monimutkaisempia rakenteita kuin peruskäyttäjähän ehkä osaisi olettaa. Kuvassa 7 näkyy sovelluksen hakemistorakenne ja reititys, eli miten ja mihin ruutuihin sovelluksessa navigoidaan.

Flutter tarjoaa kaksi tapaa navigoida sovelluksen ruutujen välillä. Nämä ovat imperatiivinen Navigator 1.0 ja deklaratiiivinen Navigator 2.0. Opinnäytetyön sovellus käyttää ensin mainittua, koska se soveltuu tähän käyttötarkoitukseen paremmin. Navigator 2.0 nostaisi turhaan sovelluksen kompleksisuutta ja tarvittun koodin määrää tarjoamatta kuitenkaan mitään etuja käyttökokemuksen suhteen. Mitä monimutkaisempi sovellus, sitä enemmän siihen mahtuu virheitä.



```
lib > routes.dart > ...
1 // The Routing of the application is done here.
2 import 'package:dewnfocus/timer/timer.dart';
3 import 'package:dewnfocus/todo/todo.dart';
4 import 'package:dewnfocus/home/home.dart';
5 import 'package:dewnfocus/profile/profile.dart';
6 import 'package:dewnfocus/login/login.dart';
7
8 var appRoutes = {
9   '/': (context) => const HomeScreen(),
10  '/todo': (context) => const ToDoList(),
11  '/login': (context) => const LoginScreen(),
12  '/profile': (context) => const ProfileScreen(),
13  '/timer': (context) => const FocusTimer(),
14 };
15
```

KUVA 7. Kuvassa vasemmalla sovelluksen kansiorakenne ja oikealla sovelluksen sisäinen reititys.

4.2 Koodiesimerkkejä

Dart-koodin lähtöpiste on aina main-funktio, joka on tyypiltään void, eli se ei palauta mitään arvoa. Opinnäytetyössä kehitetyn sovelluksen pääfunktio löytyy kuvan 8 riveiltä 11–14.

```

6 import 'package:dewnfocus/routes.dart';
7 import 'package:flutter/material.dart';
8 import 'package:dewnfocus/theme.dart';
9 import 'package:firebase_core/firebase_core.dart';
10
11 Run | Debug | Profile
12 void main() {
13   WidgetsFlutterBinding.ensureInitialized();
14   runApp(const App());
15 }
16
17 class App extends StatefulWidget {
18   const App({Key? key}) : super(key: key);
19
20   @override
21   _AppState createState() => _AppState();
22 }
23
24 class _AppState extends State<App> {
25   final Future<FirebaseApp> _initialization = Firebase.initializeApp();
26
27   @override
28   Widget build(BuildContext context) {
29     return FutureBuilder(
30       future: _initialization,
31       builder: (context, snapshot) {
32         if (snapshot.hasError) {
33           return const Text(
34             'VIRHE!',
35             ); // This error should never happen, but the error message should exist just in case.
36         }
37
38         if (snapshot.connectionState == ConnectionState.done) {
39           return MaterialApp(
40             routes: appRoutes,
41             theme: appTheme,
42             ); // MaterialApp
43
44           // While waiting for init to complete:
45           return const Center(
46             child: Text(
47               'Ladataan 🌐',
48               textDirection: TextDirection
49                 .ltr, // This tells flutter to use the Left To Right text direction.
50               style: TextStyle(
51                 fontSize: 35, fontWeight: FontWeight.w700, color: Colors.white), // TextStyle
52             ), // Text
53           ); // Center
54         },
55       ); // FutureBuilder
56     }
57 }

```

KUVA 8. Opinnäytetyössä kehitetyn Dew'n Focus-sovelluksen pää- eli main-funktio ja sen hyödyntämää koodia.

Jokaisessa Flutter-projektissa on pubspec.yaml-tiedosto. Siihen kirjataan viitteet kaikkiin kirjastoihin, joita kyseinen Flutter-sovellus tarvitsee. Siihen laitetaan myös edellä mainittujen kirjastojen versionumerot. Se on hyvin tarkka sisältönsä oikeanlaisesta asettelusta. Jo pelkkä vääränlainen tekstin sisennys aiheuttaa virheilanteen. Tabulaattorin käyttö on kokonaan kiellettyä kyseistä tiedostoa muokattaessa.

Tämän opinnäytetyön tuloksena syntyneen sovelluksen pubspec.yaml-tiedosto on esitelty kuvassa 9. YAML-kieltä käytetään usein asetustiedostoissa, kuten Flutterinkin kohdalla tehdään.

```
1  name: dewnfocus
2  description: The Software part of Tero Lahti's Bachelor's Thesis (May 2022, Tampere University of Applied Sciences.)
3
4  version: 1.0.0+1
5
6  environment:
7    sdk: "≥2.17.0 <3.0.0"
8
9  dependencies:
10   flutter:
11     sdk: flutter
12
13   firebase_core: 1.17.0
14   cloud_firestore: ^3.1.17
15   firebase_auth: 3.3.18
16   google_sign_in: 5.3.1
17   crypto: 3.0.2
18   font_awesome_flutter: 10.1.0
19   google_fonts: 2.3.2
20
21  dev_dependencies:
22   flutter_test:
23     sdk: flutter
24
25   flutter_lints: ^1.0.0
26   flutter_native_splash: ^2.2.1
27   flutter_launcher_icons: ^0.9.2
28   percent_indicator: ^4.2.2
29
30  flutter:
31    uses-material-design: true
32
33    assets:
34      - assets/
35      - assets/icon.png
36
37  flutter_native_splash:
38    color: "FF6600"
39    image: assets/splash.png
40    ios: false
41
42  flutter_icons:
43    android: true
44    ios: false
45    image_path: "assets/icon.png"
```

KUVA 9. Pubspec.yaml-tiedosto sisältää viittaukset sovelluksen tarvitsemiin kirjastoihin.

Tehtävälistan tehtävät poistetaan pyyhkäisemällä niitä vasemmalle. Pyyhkäisy on useimmille mobiililaitteiden käyttäjille varsin tuttu ja luonnolliselta tuntuva ele, joten se on sopiva suurimmalle osalle sovelluksen potentiaalisia käyttäjiä. Jatkokehityksessä olisi kuitenkin hyvä toteuttaa myös jokin vaihtoehtoinen keino, jotta sovelluksen saavutettavuutta voitaisiin parantaa.

```

void _deleteToDoItem(int index) {
    // Delete the item at the given index
    setState(() {
        _todoList.removeAt(index);
    });
}

List<Widget> _getItems() {
    return _todoList.map((String item) {
        return Dismissible(
            key:
                UniqueKey(), // Ensures that the key is unique and that the widget can be removed from the widget tree when needed.
            onDismissed: (DismissDirection direction) {
                if (direction == DismissDirection.endToStart) {
                    // Swipe from left to right to delete it.
                    _deleteToDoItem(_todoList.indexOf(item));
                }
            },
            background: Container(
                color: Colors.red,
                child: const ListTile(
                    trailing: Icon(Icons.delete),
                ), // ListTile
            ), // Container
            child: _buildToDoItem(item),
        ); // Dismissible
    }).toList();
}

```

KUVA 10. Tehtävälistan pyyhkäisyfunktion avulla voidaan tehdä tehtävät pyyhkäistä pois listalta.

5 YHTEENVETO JA POHDINTA

Tämä opinnäytetyö syntyi tekijänsä sisäsyntyisestä tarpeesta pureutua henkilökohtaisestikin koskettavaan ongelmaan, joka vaivaa lukemattoman monia ihmisiä niin Suomessa kuin muuallakin maailmassa. Opinnäytetyön tekeminen oli myös ammatillisesti hyvin opettavaista, koska aiempi kokemus Flutterista ja Dartista oli lähes olematon. Kokonaisuutena kyse on siis sekä ammatillisesta että inhimillisestä kasvusta.

Välillä kului tunteja lukien valittujen teknologioiden dokumentaatioita ja säätäen jotakin pientä yksityiskohtaa. Toisaalta tunteja kului myös hyvin pienten bugien korjaamiseen. Bugien, jotka ironista kyllä, olivat selkeästi oman keskittymiskyvyn herpaantumisen aiheuttamia. Haasteita tuotti myös olosuhteiden luoma paine saada työ valmiiksi. Kiire on kuitenkin varsin yleistä työelämässä, joten sen kanssa on vain tultava toimeen, aivan kuten ADHD:nkin kanssa.

Loppujen lopuksi sovellus sekä valmistui että alustavan, hyvin suppean testauksen pohjalta toimii odotetusti. Työssä käytettävistä tekniikoista ei juuri ollut aiempaa kokemusta, minkä vuoksi työ oli osin haastavaa. Valituissa tekniikoissa yhdistyy monialustainen yhteensopivuus ja tehokkuus, jotka ovat erinomaisia piirteitä sovellukselle, jolle asetettiin alusta asti yhtenä tärkeimmistä vaatimuksista jatkokehitettävyys. Jo sovelluksen nykyisen version kehityksen aikana kehittyi monia parannusideoita, mutta niiden toteuttaminen ei vielä ollut ajankohtaista eikä opinnäytetyön tavoitteiden mukaista. Sovellusta voisi viedä eteenpäin esimerkiksi esittämällä sovelluksessa tilastoja käyttäjän suorituksista, sekä palkitsemalla käyttäjää esimerkiksi pisteillä. Käyttökielen valinta tulisi myös toteuttaa varhaisessa vaiheessa, jotta erikielisten käännösten mukaan tuominen olisi helpompaa myös myöhemmissä vaiheissa.

Jatkokehitysideat jäivät kuitenkin kypsymään ja joskus tulevaisuudessa sovellus löytynee ilmaisena Googlen ja Applen sovelluskaupoista, koska taloudelliset voitot eivät missään vaiheessa kuuluneet tämän opinnäytetyön tavoitteisiin. Sen tarjoamat immateriaaliset voitot ovat kuitenkin kiistattomia.

LÄHTEET

adhdfoundation.org.au. Is there a link between ADHD and dopamine? 2020. Luettu 18.5.2022. <https://adhdfoundation.org.au/wp-content/uploads/2020/06/Dopamine.pdf>

ADHD:n hoitokaavio. Käypä hoito - Käypä hoito -työryhmä ADHD (aktiivisuuden ja tarkkaavuuden häiriö). 2017. Luettu 17.5.2022. <https://www.kaypahoito.fi/imk01099>

adhdtutuksi.fi. Mikä on ADHD. n.d. Luettu 17.5.2022. <https://adhdtutuksi.fi/mika-on-adhd/>

chadd.org. What Social media Does to an ADHD Brain. n.d. Luettu 17.5.2022. <https://chadd.org/adhd-weekly/what-social-media-does-to-an-adhd-brain/>

docs.flutter.dev. n.d. Luettu 2022. <https://docs.flutter.dev/resources/faq#what-makes-flutter-unique>

Kemi, E. 2019. Luokan Luokanopettajien käsityksiä alakouluikäisten ADHD-diagnosoitujen oppilaiden oppimisen ja koulunkäynnin tukemisesta. Luettu 18.5.2022. <http://jultika.oulu.fi/files/nbnfioulu-202001211048.pdf>

Ra, C. Cho, J. Stone, M. 2018. Association of Digital Media Use With Subsequent Symptoms of Attention-Deficit/Hyperactivity Disorder Among Adolescents. Luettu 17.5.2022. <https://jamanetwork.com/journals/jama/article-abstract/2687861>

Shuai, L. He, S. Zheng, H. Wang, Z. Qiu, M. Xia, W. Cao, X. Lu, L. Zhang, J. 2021. Influences of digital media use on children and adolescents with ADHD during COVID19 pandemic. Luettu 17.5.2022. <https://globalizationandhealth.biomedcentral.com/track/pdf/10.1186/s12992-021-00699-z.pdf>

w3techs.com. 2022. Usage statistics of JavaScript as client-side programming language on websites. Luettu 19.5.2022. <https://w3techs.com/technologies/details/cp-javascript>