SAVONIA

OPINNÄYTETYÖ - **AMMATTIKORKEAKOULUTUTKINTO**
TEKNIIKAN JA LIIKENTEEN ALA

INTEGRATING 3RD PARTY MHEALTH APP INTO REACT NATIVE APP

TEKIJÄ: Jiri Tukiainen

SAVONIA-AMMATTIKORKEAKOULU

OPINNÄYTETYÖ
Tiivistelmä

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä Jiri Tukiainen	
Työn nimi React Native -sovelluksen integrointi kolmannen osapuolen mHealth -sovellukseen	
Päiväys 8.6.2021	Sivumäärä/Liitteet 32
Toimeksiantaja/Yhteistyökumppani Nursie Health Oy	

Tiivistelmä

Tämän opinnäytetyön tavoitteena oli perehtyä kahden isoimman mobiilikäyttöjärjestelmän, Androidin ja iOS:n mHealth-sovellusmarkkinoihin. Tavoitteena oli kerätä tarpeeksi tietoa olemassa olevista sovelluksista, jotta voitiin päättää, mikä sovellus on toimeksiantajan kannalta järkevin vaihtoehto integraatiolle. Tämän selvityksen pohjalta toteutettiin integraatio toimeksiantajan sovelluksen ja yhden kolmannen osapuolen sovelluksen välille.

Teoriaosuudessa perehdyttiin mHealthiin käsitteenä ja sen tarjoamiin mahdollisuuksiin terveydenhuollolle ja mHealth-sovelluksien käyttäjille, sen markkinoihin, haasteisiin ja muihin osa-alueisiin. Lisäksi käytiin läpi iOS:n ja Androidin mHealth-sovellustarjontaa ja selvitettiin, mitkä ovat isoimmat toimijat ja minkälaisia käyttäjämääriä näillä sovelluksilla on. Viimeiseksi teoriaosuudessa käsiteltiin React Nativea sovelluskehityksen osalta. Työn käytännön osuudessa toteutetaan alustava integraatio toimeksiantajan sovelluksen ja Apple Healthin välille react-native-health kirjaston avulla. Apple Health valittiin, koska sen kautta on mahdollista saada dataa myös muista sovelluksista, sillä se kokoaa lukuisien eri sovelluksien datan yhteen. Lisäksi se on yksi iOS:n oletussovelluksista, joten tällä integraatiolla on mahdollisuus tavoittaa suuria käyttäjämääriä.

Opinnäytetyön tuloksena on kirjaston onnistunut käyttöönotto, jonka avulla voidaan hakea Apple Healthista dataa ja viedä sinne dataa. Lisäksi kirjastoon toteutettiin uusia toiminnallisuuksia, joita se ei vielä tukenut.

Avainsanat
React Native, iOS, Android, mHealth

SAVONIA UNIVERSITY OF APPLIED SCIENCES

THESIS
Abstract

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author Jiri Tukiainen			
Title of Thesis Integrating React Native Application Into Third-Party mHealth Application			
Date	8 June 2021	Pages/Appendices	32
Client Organisation /Partner Nursie Health Oy			

Abstract

The goal of this thesis was to investigate the mHealth application markets of the two biggest mobile operating systems, iOS, and Android. The aim was to gather enough knowledge of existing applications, so the decision, which application to integrate into, could be made. Based on this investigation, the integration between the client's application and one third-party application was carried out.

mHealth and its possibilities for healthcare providers and application users and other aspects such as the markets and challenges were addressed in the theory part of this thesis. The selection of the biggest mHealth applications available on iOS and Android and their download counts were also covered. Finally, the library used in the client's application, React Native, was introduced. In the practical part of this thesis, an integration between Apple Health and the client's application was done by using a library called react-native-health. Apple Health was chosen as it is possible to get data from multiple third-party applications through it. It is also one of the stock applications on iOS, so the userbase is high.

As a result of this thesis, the library was taken into use successfully and reading and writing data between the two apps was implemented. Additionally, non-existing functionalities were added to the library.

Keywords
React Native, iOS, Android, mHealth

TABLE OF CONTENTS

LIST OF ABBREVATIONS 5

1 INTRODUCTION 6

2 MHEALTH 7

2.1 What is mHealth? 7

2.2 Solutions for healthcare 7

2.3 Market potential 8

2.4 Awareness and adoption of mHealth apps 10

2.5 Problems and challenges of mHealth applications 10

2.5.1 Clinical challenges 10

2.5.2 Technical challenges 11

3 STATE OF MHEALTH ON IOS & ANDROID 13

3.1 iOS 13

3.1.1 Apple Health & HealthKit 13

3.2 Android 14

3.2.1 Google Fit 15

3.3 Other apps 16

4 REACT NATIVE 17

4.1 An overview 17

4.2 Pros and cons versus native development 17

4.3 Getting into React Native 18

5 INTEGRATION 19

5.1 Findings of theory part 19

5.2 Requirements 19

5.3 Implementation 20

5.3.1 Overview of the library and how to use it 20

5.3.2 Handling similar data from two sources 22

5.3.3 Adding new functionality 23

6 CONCLUSION 28

7 SOURCES 30

LIST OF ABBREVATIONS

JSON

JavaScript Object Notation. It is a syntax for writing data in human-readable format.

HL7

Health Level 7. A set of international standards for moving data between healthcare applications and providers.

API

Application Programming Interface. An API defines how and what interactions work between multiple applications

FDA

The United States Food and Drug Administration. An organization that supervises food safety, medications, cosmetics and other substances and products.

MDR

The European Union Medical Device Regulation. A regulation that covers the clinical investigation and sale of medical devices

CLI

Command-line interface. CLI is a command line program used for executing various functions by inputting text commands.

1 INTRODUCTION

This thesis is commissioned by Nursie Health, a health technology company I work at, that combines practical nursing, digitalization, and self-care services. The company is developing a self-care application called MyNursie to enable all-inclusive and long-term self-care assistance straight from one's mobile phone. Goal of this thesis is to provide enough knowledge for the company about the state of mHealth applications on the two major mobile operating systems, iOS, and Android, and help the company to pick the right third-party apps to which we can integrate to with our own solution. The integration between third-party application and MyNursie provide our users even further control of their own health. (Nursie Health, 2021.)

The theory part covers aspects such as market size, real-world usage, and awareness of mHealth applications, downsides of present applications and technical and clinical challenges. The goal is to gain enough knowledge of the mHealth field, so we can draw the conclusion which application or applications should we integrate to, based on the value they provide not only for us, but for our users. In this thesis, mHealth is covered as a whole to give a clear basis of what it includes and what is its state nowadays. When comparing the mHealth apps available on iOS and Android, the focus will be on fitness and activity tracking related apps. This is because these kinds of apps and the data they provide will be the most beneficial for Nursie Health's own solution and the other types of mHealth apps are a bit out of scope for the time being.

In the practical part of this thesis, a proof-of-concept of an integration into one third-party app will be carried out. The aim is to conclude what application can provide us the most value at this time and the theory part will act as a basis for the decision.

2 MHEALTH

2.1 What is mHealth?

mHealth or mobile health is defined as "*medical and public health practice supported by mobile devices, such as mobile phones, patient monitoring devices, personal assistants (PDAs), and other wireless devices*". Lifestyle and wellbeing apps with medical device or sensors connections and personal guidance systems are also included in mHealth. Most notable examples of mHealth solutions are communication, information and motivation apps and tools. The features these apps offer range from medication reminders to fitness and dietary suggestions. mHealth apps often support adding physiological measurements, such as heart rate, blood glucose level, blood pressure and body temperature. (WHO, 2011; European Commission, 2014.)

2.2 Solutions for healthcare

mHealth could be one of the solutions that resolve the new challenges the healthcare field is facing. In the United States alone, over half of physicians have had discussions about mHealth with their patients and around a quarter of patients have consulted them about mHealth, so the interest and potential are there. As mobile phones and wireless connectivity are starting to become more and more ubiquitous, mHealth can truly harness its potential and become the game-changer in the coming years tackling hardships such as the growing elderly population and the rise in budgetary limitations. (Rowland, Fitzgerald, Holme, Powell, & McGregor, 2020.)

A significant prospect of mHealth is that it can provide the means to detect chronic diseases at an early stage. By enabling self-assessment for users and sharing the data with healthcare providers, early prevention could be possible. It is found out that over 50 million people have already used mHealth applications for self-triage in recent years. By improving and paying attention to prevention, also a greater quality of life and even longer life expectancy can become possible to achieve. A direct benefit from using mHealth applications for disease prevention is that the healthier the population is, the less financial pressure they put on the healthcare system. The table below illustrates the evidence-backed cases where mHealth apps can provide value to users (Figure 1). (Rowland, Fitzgerald, Holme, Powell, & McGregor, 2020; European Commission, 2014.)

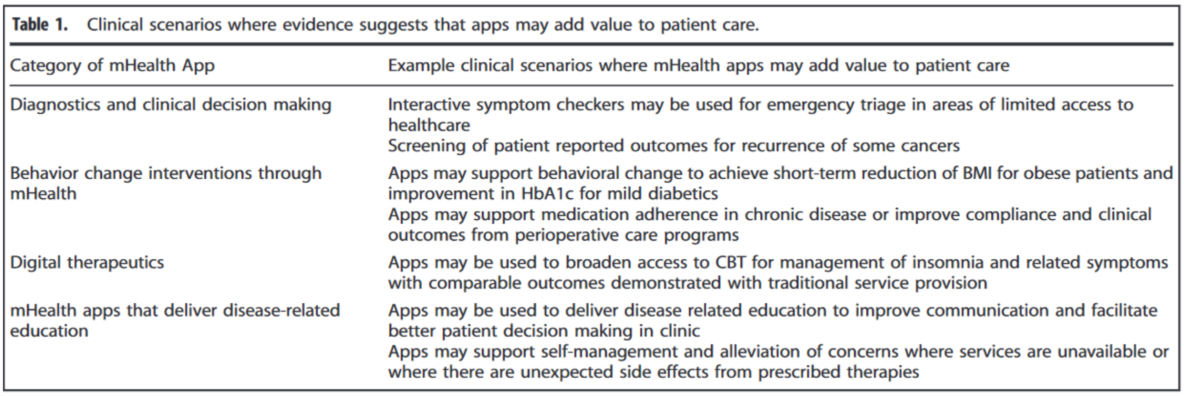


Table 1. Clinical scenarios where evidence suggests that apps may add value to patient care.

Category of mHealth App	Example clinical scenarios where mHealth apps may add value to patient care
Diagnostics and clinical decision making	Interactive symptom checkers may be used for emergency triage in areas of limited access to healthcare Screening of patient reported outcomes for recurrence of some cancers
Behavior change interventions through mHealth	Apps may support behavioral change to achieve short-term reduction of BMI for obese patients and improvement in HbA1c for mild diabetics Apps may support medication adherence in chronic disease or improve compliance and clinical outcomes from perioperative care programs
Digital therapeutics	Apps may be used to broaden access to CBT for management of insomnia and related symptoms with comparable outcomes demonstrated with traditional service provision
mHealth apps that deliver disease-related education	Apps may be used to deliver disease related education to improve communication and facilitate better patient decision making in clinic Apps may support self-management and alleviation of concerns where services are unavailable or where there are unexpected side effects from prescribed therapies

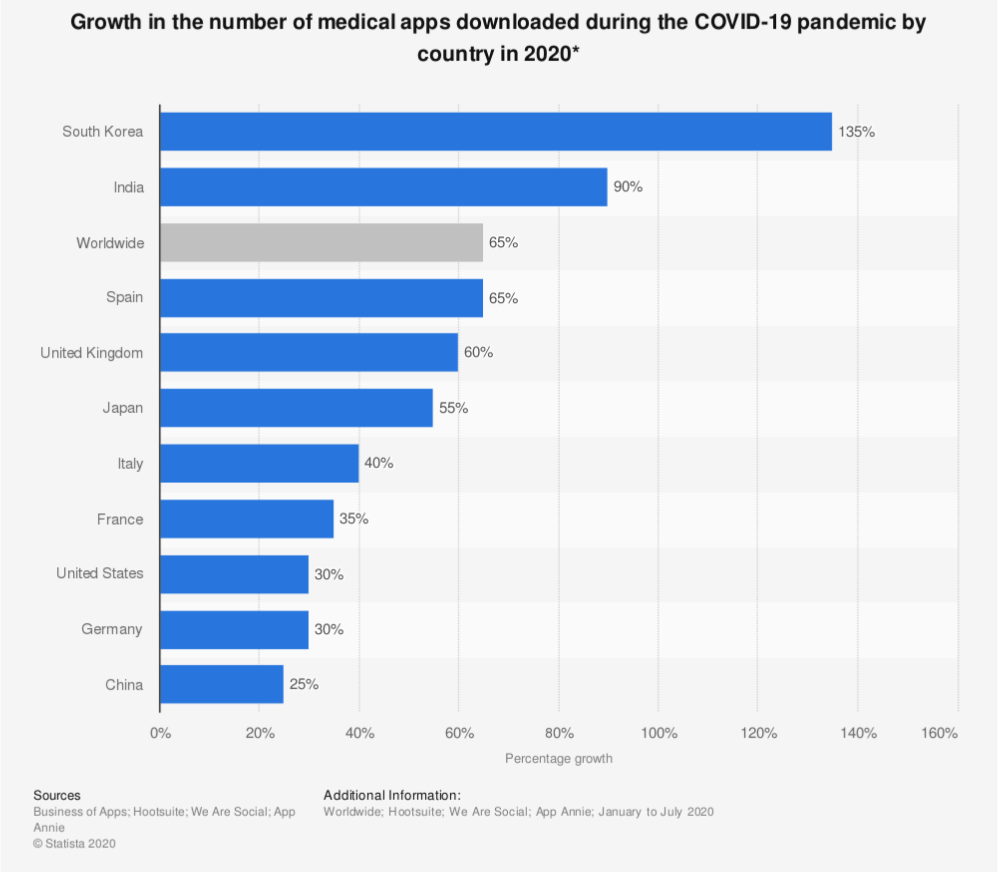
Figure 1. Evidence supported scenarios where mHealth apps can be beneficial (Rowland, Fitzgerald, Holme, Powell, & McGregor, 2020).

mHealth also has the capability to make the delivery of healthcare services more efficient. It can help by reducing unneeded consultations and provide guidance to professionals on treatment and medication. Apps that allow communication and exchange of data between the patient and the healthcare provider can help utilize the healthcare workforce better since it has been found out that a big portion of healthcare related appointments and consultations could be done remotely. They could even be done by the patients themselves with an app that could help manage a chronic disease by providing a way for monitoring and guidance. This would ultimately lead to lower healthcare costs and improved patient comfort. By enabling patients to monitor themselves, a large amount of health data would be generated that could be further processed and analysed, providing the healthcare professional a better picture of the patients' overall condition. (European Commission, 2014.)

As already stated in the previous chapter, allowing patients to monitor their health through mHealth apps and sensors empowers them and makes them more autonomous as they now can live more independently in their home environment and have greater control of their health. Gaining insights on their health through apps can raise patients' awareness of their condition and make it easier to cope with. mHealth solutions also make it easier for users to follow, for instance, a diet or medication. This transition into patient-centric healthcare is already ongoing and existing infrastructures need to rethink their composition. The most important aspect of this reformation is to be able to receive patient-generated data. (European Commission, 2014.)

2.3 Market potential

mHealth is becoming a prominent way of delivering healthcare globally and the market is larger than ever before, with expected market size of 189 billion dollars by 2025, and the global smartphone penetration has made specifically apps the main driving force of mHealth. The recent pandemic has also played a role in the adoption towards mHealth, as the downloads of medical apps surged globally (Figure 2). The market is mostly divided in two categories of apps: medical apps and fitness apps, the former of which is dominating the market with around 71% share. The share of fitness apps is expected to see great growth in the upcoming years. (Stewart, Global digital health market by major segment 2015-2025, 2021; Grand View Research, 2021.)


Growth in the number of medical apps downloaded during the COVID-19 pandemic by country in 2020*

Sources
Business of Apps; Hootsuite; We Are Social; App Annie
© Statista 2020

Additional Information:
Worldwide; Hootsuite; We Are Social; App Annie; January to July 2020

Figure 2. Growth in medical app downloads during COVID-19 in 2020 (Stewart, 2020).

It is important to notice that while the availability of mHealth apps and downloads have increased rapidly in the past few years, the market share is dominated by a select few apps that total the most users and downloads. There were around 325 000 mHealth apps available in 2017 but only 4% of the publishers were able to get downloads exceeding 1 million, and 83% had less than 10 000 monthly active users. (Research2Guidance, 2017.)

Mobile cellular subscribers reaching an 8 billion mark in 2019, the potential market now covers most of the globe. Since wireless networks are so widespread, even the poorer countries can benefit from mHealth solutions – in fact, 70% of wireless subscribers live in middle- and low-income countries. From a marketing point of view, mHealth app publishers must recognize the differences what mHealth can offer for high-income countries versus developing countries. In high-income countries, mHealth is driven forward by the need to cut healthcare costs, while in low-income countries it can provide the means for people to access primary care. (European Commission, 2014; WHO, 2011; International Telecommunication Union (ITU), 2021.)

2.4 Awareness and adoption of mHealth apps

As mHealth apps are becoming an integral part of healthcare field, it is important to evaluate how they are perceived by the general population and how willing people are to adopt them into their daily lives. It is not only vital to inform the public about mHealth solutions, but also the people working in the healthcare field. They are in the key position to recommend or even prescribe apps and highly influence the usage and adoption of mHealth into patients' daily lives. The success of mHealth solutions depends how disposed the patients are to commit using them and therefore it is vital to understand what factors affect it. (Deng, Hong, Ren, Zhang, & Xiang, 2018; Reem, et al., 2017.)

Some aspects that have been identified to positively affect users' commitment to mHealth apps are how trustworthy, how useful, and how easy to use the app is. Negative aspects include privacy concerns, performance issues, usability issues and price-value factor. In addition to these aspects that root from the app itself, other factors such as age and the presence of a chronic illness in an individual also affect their trust towards mHealth solutions. It has been found out that older and people with a chronic condition have higher trust towards mHealth apps. (Deng, Hong, Ren, Zhang, & Xiang, 2018.)

2.5 Problems and challenges of mHealth applications

2.5.1 Clinical challenges

The massive potential that mHealth apps have is held up by the lack of clinical guidance on how these applications should be utilized. As the mHealth market and app availability is rapidly growing, a significant adverse effect is that only a small portion of these apps are proven to be effective and rely on evidence-based medicine. (Larson, 2018.)

There are regulatory models and organizations that aim to ensure that apps meet minimal standards of safety and proven effectiveness, most notable of which are FDA and MDR. A downside is that these regulations only match a small portion of mHealth apps, those that can be considered medical devices. For example, the FDA only considers apps to be medical devices if the app is used *"for the diagnosis of disease or other conditions, or the cure, mitigation, treatment, or prevention of disease, or is intended to affect the structure or any function of the body of man*". This means that apps that only coach the users, encourage them to manage their condition or only track their health data are not regulated. (Larson, 2018; U.S Food And Drug Administration, 2019.)

2.5.2 Technical challenges

The change mHealth can provide for healthcare globally does not come without challenges. Factors like unreliable technologies, data privacy and security, usability, non-uniformity in the technologies and the inadequacy in end-user education need to be addressed. (Gurupur & Wan, 2017.)

Usability is a key element when designing an app especially for healthcare purposes. Generally, it can be split into 5 different components (Figure 3).

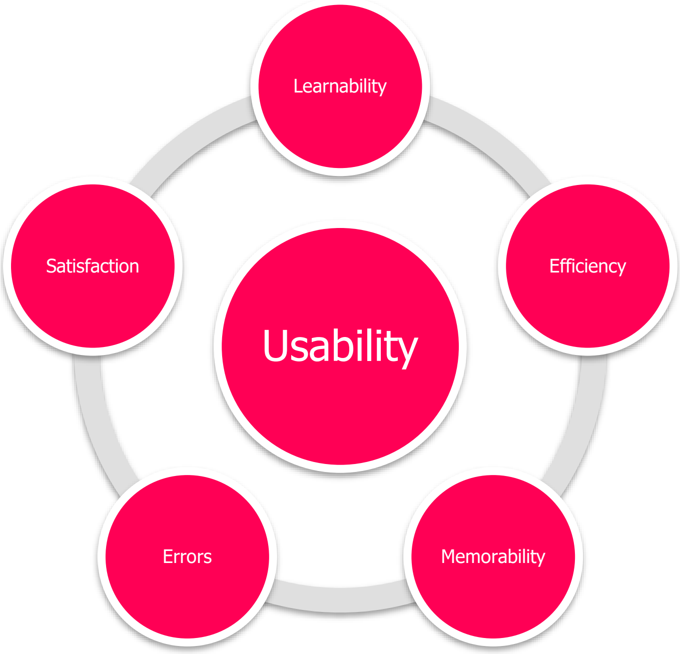


Figure 3. Usability and its components (Adapted from Nielsen, 2012).

Learnability defines how effortlessly the users can achieve the task at hand. After they have learned how the design works, efficiency comes into play. Its purpose is to find out how fast they can now perform tasks. Memorability means how long does it take for the users to gain proficiency using the design after not using it for an indefinite amount of time. When the design is in use, the errors users are making need to be observed. It is important to track what kind, how many, how serious errors users make and how they recover from them. Finally, satisfaction measures how much users enjoy using the design. When talking about mHealth apps, it has been found out that out of these 5 components, satisfaction is perceived as the number one factor for people in the know of mHealth as well as the users. Learnability and efficiency were also found out to be important, although less so than satisfaction. mHealth suppliers should be aware what the users expect from the app and what factors cause them to keep using the app or abandon it. Strengthening the connectivity between the developers and users, the uptake of mHealth applications is much higher. Therefore, it is important to address these key components that usability consists of and make sure that the design supports the cause. (Nielsen, 2012; Liew, Zhang, See, & Ong, 2019.)

Data security is a must for mobile apps, especially for those that tackle with health data. Mobile apps are often cross-platform and that causes there to be multiple possible loopholes and vulnerabilities. In a 2020 report, Intertrust analysed 100 mHealth applications found in the major app stores for security issues. They found out that 71% of the apps analysed have at least one high level vulnerability and 85% of COVID tracking apps leak the data they collect. Alarming aspect is that according to Intertrust, 83% of the high-level vulnerabilities could have been prevented by in-app protection. Healthcare organisations and developers should make security a top priority when developing mHealth applications. (Isakova, 2021; Intertrust, 2020.)

Interoperability and integrations also pose a wide range of technical difficulties for mHealth apps. Regarding these two themes, one prominent solution is to decide on standards that developers and healthcare providers should follow. A promising and widely used set of standards is the HL7. It provides multiple flexible standards and directives that are implemented in numerous healthcare systems. These allow the information to be distributed and processed in a uniform way across all platforms. (Isakova, 2021; Gurupur & Wan, 2017.)

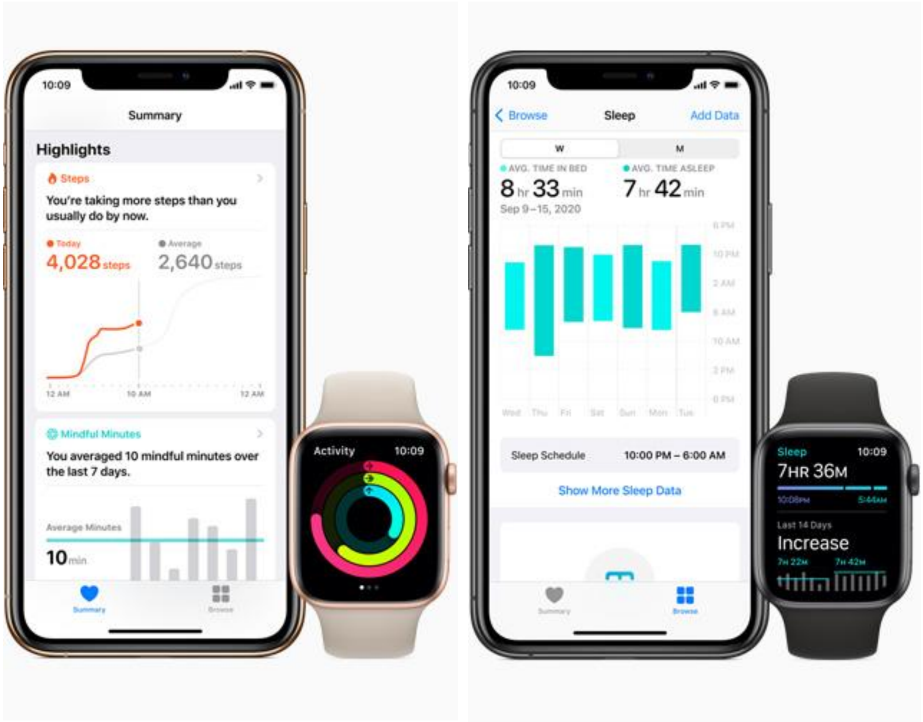
3 STATE OF MHEALTH ON IOS & ANDROID

3.1 iOS

iOS is the operating system that runs on Apple's mobile devices, that include iPhone, Apple Watch, and iPad. It is the second-most popular operating system after Android, with approximately 15% market share in 2020. iOS was initially released in 2007 along with the first iPhone. The effectiveness of iOS paved the road for the success of the iPhone, and it became the single most successful product ever released after selling almost 218 million units in 2018. iOS and the iPhone revolutionized the way of using mobile devices by combining multiple functionalities within a single device, such as camera, internet browsing, media capabilities and more. (Apple, Inc., 2007 as cited in Kenton, 2020; Vailshery, 2018 as cited in Kenton, 2020.)

3.1.1 Apple Health & HealthKit

Apple Health is the official health and fitness app of Apple and one of the stock apps on iPhones. It combines the data gathered from the iPhone and Apple Watch and a wide range of third-party wearables and apps. This is its most prominent feature as users are not required to open multiple apps to see their health data, since Apple Health shows it all. Apple Health has an extensive variety of options out of the box, such as allergy information, medical conditions, and blood type to name a few. The third-party integrations extend its coverage even further. Figure 4 shows what Apple Health looks like on iPhones. (Nield, 2020; O'Boyle, 2019.)



You're taking more steps than you usually do by now.

You averaged 10 mindful minutes over the last 7 days.

Figure 4. User Interface of Apple Health (Apple Inc., 2021).

Healthkit is the framework used to make third-party apps compatible with Apple Health. It provides a comprehensive API to share data between Apple Health and other apps. Healthkit provides a predefined and constrained set of data types and units, which provides great interoperability and no custom data types are allowed. (Apple Inc., 2021; O'Boyle, 2019.)

Other mentionable initiatives of Apple are CareKit and ResearchKit. CareKit is a framework for building apps that support more personalized healthcare. It can be used to implement features such as custom care plans, daily progress tracking and generating trends. ResearchKit is another framework to help develop apps that provide data for research purposes. Surveys, custom tasks, and consents are some of the features that ResearchKit provides. Both frameworks can be used together with HealthKit to create even more meaningful and complex apps. (Horton, 2020; Apple Inc., 2021.)

3.2 Android

Android is a mobile operating system, that was originally developed by Android Inc. It was then acquired by Google in 2005. Its primary use is to power mobile devices such as phones and tablets, some other use cases for Android include cars, televisions, and wearable devices. Developers can create apps with Android SDK and the apps are usually published through Google Play, the official distribution service for Android. The source code for Android is released as open source, although Android variants more often come packaged with proprietary software, depending on the manufacturer. It is the most popular mobile operating system and held a smartphone market share of around 84% in 2020. (Chen, 2021; International Data Corporation, 2021.)

Regarding Android's wearable and fitness tracking market, a big update came from Google on May 18, 2021, as they announced collaboration with Samsung to unify Google's WearOS and Samsung's TizenOS wearable operating systems into a single platform. This might make it possible for Android to have a single ecosystem for fitness trackers and wearables the same ways as iOS does with Apple Health. It is stated that the update does not only concern Samsung and Google as "*all device makers will be able to add a customized user experience on top of the platform, and developers will be able to use the Android tools they already know and love to build for one platform and ecosystem.*" (Killburn, 2021.)

3.2.1 Google Fit

Google Fit is Google's official fitness tracker app that runs on both Android and iOS. It supports tracking activity using the sensors on a phone and wearables that run Google's Wear OS. Some third-party wearables and apps are also supported via syncing between the app and Google Fit. Compared to Apple Health, Google Fit is a much simpler app and provides more basic functionalities, such as steps and activity monitoring. Setting personal goals for activity is a feature that makes Google Fit a bit more interactive. You can set goals for steps and Heart Points, a relatively new feature utilizing recommendations from World Health Organization and American Heart Association which tracks cardio focused activities and scores them. Figure 5 shows the user interface of Google Fit. (Nield, 2020; Google, 2021.)

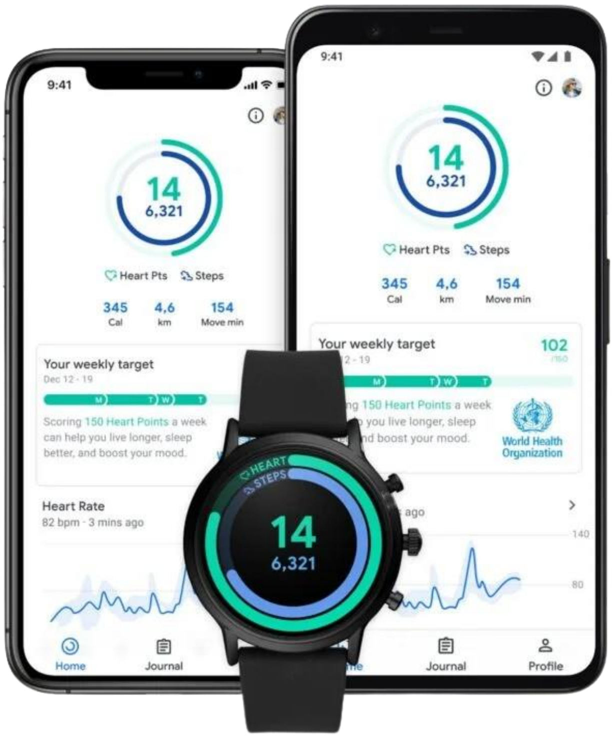


Figure 5. User Interface of Google Fit (Google, 2020).

3.3 Other apps

Alongside Google Fit and Apple Health, there is a wide range of fitness and health trackers out there. Some of the biggest apps and their download counts, as found out in Google Play Store include:

- Samsung Health (1,000,000,000+)
- Mi Fit (50,000,000+)
- Fitbit (50,000,000+)
- Garmin Connect (10,000,000+)
- Polar Flow (1,000,000+)
- Suunto (1,000,000+)
- Withings Health Mate (1,000,000+)
- Oura (100,000+)

(Google, 2021.)

4 REACT NATIVE

4.1 An overview

React Native is a framework used for mobile development and it is developed by Facebook Inc. for developing iOS and Android applications with React and the native capabilities the app’s platform provide. Under the hood, React Native utilises JavaScript to access platform specific interfaces. Figure 6 below demonstrates the native counterparts for some of React Native’s components. Traditionally, Android applications are developed either in Kotlin or Java while in iOS development, Swift or Objective-C are the available languages. React Native invokes platform specific capabilities using JavaScript. As a result, the apps developed with React Native perform as their native counterparts. It comes with a ready-made set of Native Components for iOS and Android, that are called Core Components. It is also possible to create one’s own Native Components. In addition to these components, users can write Native Modules, that invoke platform specific APIs to access features not available by default. (Facebook, Inc., 2021.)

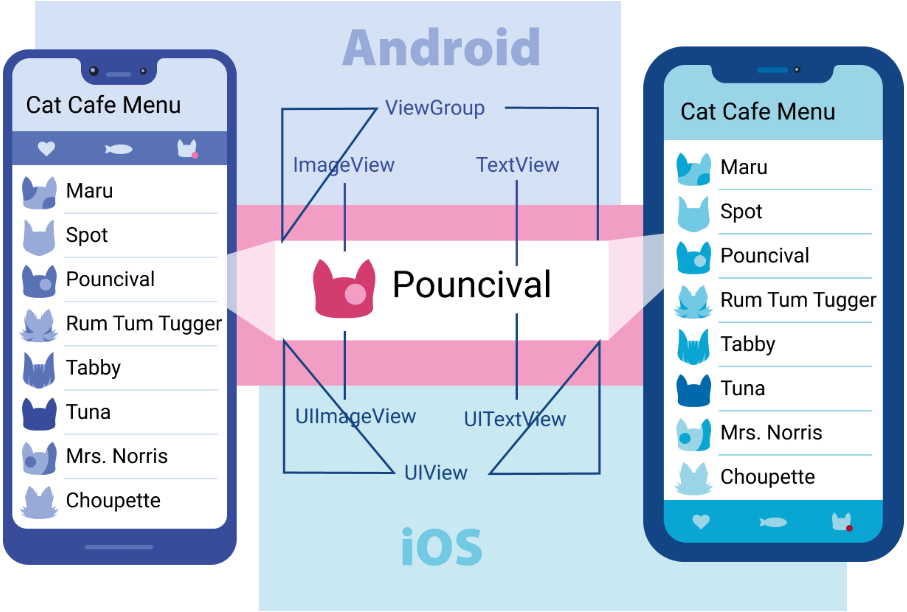


Figure 6. Native views of iOS and Android (Facebook, Inc., 2021).

4.2 Pros and cons versus native development

Possibly the greatest advantage of React Native is how fast and efficient it is to write an app for both iOS and Android. It provides a single JavaScript codebase that works on both platforms and minimizes the need for platform-specific development. The fact that it is JavaScript based framework makes it very easy for a company to find developers. JavaScript is hugely popular as found out by Stack Overflow that 67.7% of respondents in their 2020 Developer Survey use JavaScript. (Trnka, 2018; Stack Overflow, 2020.)

Native development on iOS and Android provides its own positive sides. While JavaScript is an interpreted and untyped language, the languages used in native development are strongly typed and compiled. This provides much more reliable, predictable, and safer development experience. Though the same behaviour can be achieved on React Native using TypeScript or linting tools, it is still something to keep in mind. (Trnka, 2018.)

Caveats of React Native include the lack of native elements and API's and limited third-party libraries. React Native does not support all native capabilities out of the box, and while one can write their own library or native module to achieve the desired outcome, native iOS/Android development skills are needed. The third-party library field lacks behind Android and iOS and that provides its own challenges to write wrappers for native libraries. (Trnka, 2018; Bullington, 2019.)

4.3 Getting into React Native

There are currently two ways to set up one's own React Native project, either by using Expo CLI or React Native CLI. The choice depends on the experience of the developer and which of these solutions support the needs. (Kruhlyk, 2020; Facebook, Inc., 2021.)

Expo is essentially a combination of tools and additional features wrapped around vanilla React Native and its goal is to enable rapid development without the need to spend much time setting up a development environment. Expo does not require setting up platform-specific IDEs as is the case with React Native CLI, and the overall installation process is much more straightforward. (Kruhlyk, 2020; Facebook, Inc., 2021.)

React Native CLI is for developers with more experience in mobile development. To use it, one needs to configure Xcode or Android Studio, depending on the target OS. The advantage over Expo CLI is reduced app size and access to native modules that are written in Objective-C, Swift, Java or Kotlin. Expo also ties developers to a certain version of React Native and the whole Expo infrastructure, so React Native CLI provides more freedom and control over the project. React Native's official documentations are of good quality and provide a relatively easy way for anyone interested in mobile development without the need of native development skills. (Kruhlyk, 2020; Facebook, Inc., 2021.)

5 INTEGRATION

5.1 Findings of theory part

In the theory part we found out that the market is blooming for mHealth solutions. More and more people are adapting them into their daily lives, and it is expected that the field of mHealth plays a significant role in delivering healthcare globally in the upcoming years.

There are many types of mHealth applications available, from remote appointments to patient monitoring, self-triage and overall wellbeing and fitness trackers. For the time being, Nursie Health is most interested in the latter, wellbeing, and fitness apps. These apps can provide us important metrics about people's health, such as blood pressure, heart rate, physical activity levels and other measurements and metrics. It was already discussed in the company beforehand, that a solution that could give the access to most data would be ideal for the initial integration into third-party apps. As such, Apple Health was concluded to be the most beneficial target since it integrates with numerous third-party applications. This makes it possible to fetch data from various sources as Apple Health acts as a central repository for all the data. It efficiently mitigates the need to create and implement multiple integration solutions for different apps.

5.2 Requirements

The ultimate goal of the integration is to provide additional value and data to Nursie Health's application, MyNursie. Our application can greatly benefit from activity, heart rate and other types of data to provide a more meaningful experience for our users. As found out in previous chapters, more and more people track their health using mHealth solutions and this makes integrating into an existing app the best option for us, since the data is already there.

Due to a tight schedule, the scope of this integration is limited to fetching data from Apple Health into MyNursie and writing data to Apple Health. The aim is to see how straightforward the process is, are there any caveats when two applications are possibly creating the same type of data and ultimately, which data is prioritized. This will act as the basis for a more in-depth integration process that is to be implemented in the near future.

5.3 Implementation

At the time of writing this thesis, a relatively new library called react-native-health was available and seemed promising. It is gathering over 1000 downloads weekly, is updated frequently and supports a wide range of measurements the HealthKit offers. This library is based on React Native's NativeModule system. Essentially it allows us to access HealthKit APIs using plain JavaScript. If we were to create our own solution for Apple Health integration from scratch, it would have been more or less the same as react-native-health and done by utilising the NativeModule system to invoke HealthKit APIs. The fetching and writing of data between the two apps are handled using this library. Additionally, the process of implementing a feature not yet present in it is demonstrated as it might prove to be vital to know how this library works, since the maintainers might abandon the project at any time. This way the development of this library can be continued within the company. (Macelai, 2021.)

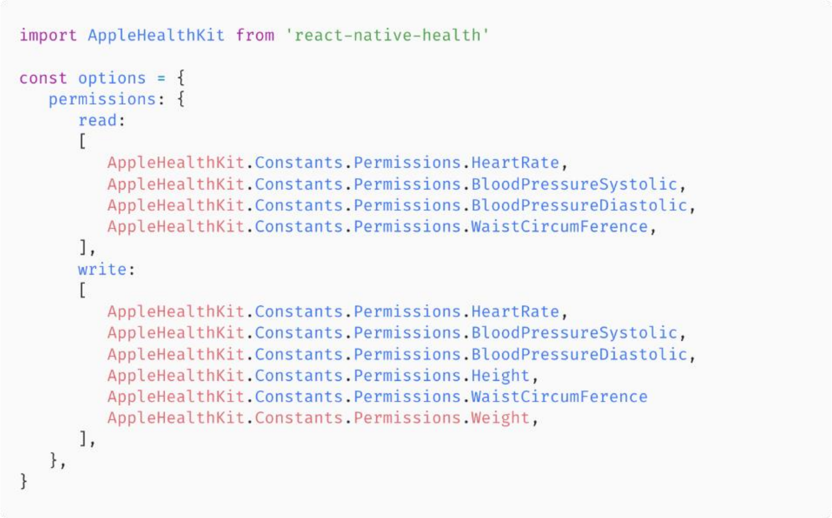
5.3.1 Overview of the library and how to use it

The core features the library offers are methods that can be used to fetch various types of data from Apple Health. These among others include:

- Information about vitals
- Workouts
- Mindfulness sessions
- Sleep
- Dietary data
- Laboratory tests such as blood alcohol and blood glucose content
- Fitness data including steps, swimming, cycling etc.

(Macelai, 2021.)

To make use of the library, the first step is to ask for the user’s permissions to collect data, using an object that contains the wanted permissions (Figure 7).

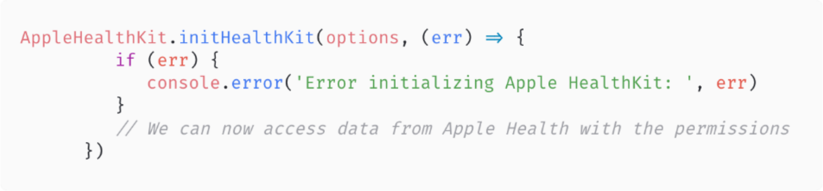


```
import AppleHealthKit from 'react-native-health'

const options = {
   permissions: {
      read:
      [
         AppleHealthKit.Constants.Permissions.HeartRate,
         AppleHealthKit.Constants.Permissions.BloodPressureSystolic,
         AppleHealthKit.Constants.Permissions.BloodPressureDiastolic,
         AppleHealthKit.Constants.Permissions.WaistCircumFerence,
      ],
      write:
      [
         AppleHealthKit.Constants.Permissions.HeartRate,
         AppleHealthKit.Constants.Permissions.BloodPressureSystolic,
         AppleHealthKit.Constants.Permissions.BloodPressureDiastolic,
         AppleHealthKit.Constants.Permissions.Height,
         AppleHealthKit.Constants.Permissions.WaistCircumFerence
         AppleHealthKit.Constants.Permissions.Weight,
      ],
   },
}
```

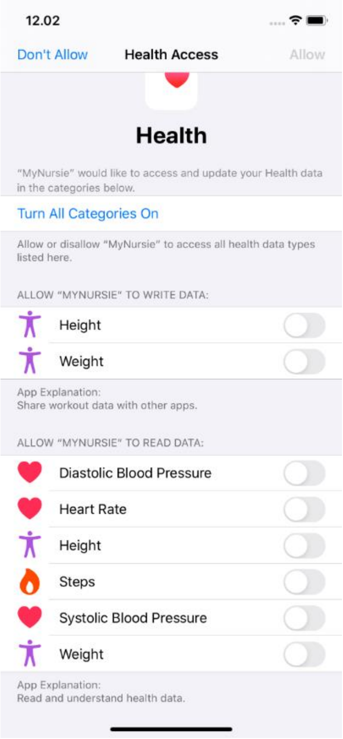
Figure 7. Setting up permissions for react-native-health.

We can then pass these permissions to the initialization function, and upon the next launch of the app, Apple Health will open and ask the user to accept or deny some or all these permissions (Figure 8 and Figure 9).



```
AppleHealthKit.initHealthKit(options, (err) => {
      if (err) {
        console.error('Error initializing Apple HealthKit: ', err)
      }
      // We can now access data from Apple Health with the permissions
    })
```

Figure 8. Initializing HealthKit with permissions.



12.02

Don't Allow Health Access Allow

Health

“MyNursie” would like to access and update your Health data in the categories below.

Turn All Categories On

Allow or disallow “MyNursie” to access all health data types listed here.

ALLOW “MYNURSIE” TO WRITE DATA:

Height

Weight

App Explanation:
Share workout data with other apps.

ALLOW “MYNURSIE” TO READ DATA:

Diastolic Blood Pressure

Heart Rate

Height

Steps

Systolic Blood Pressure

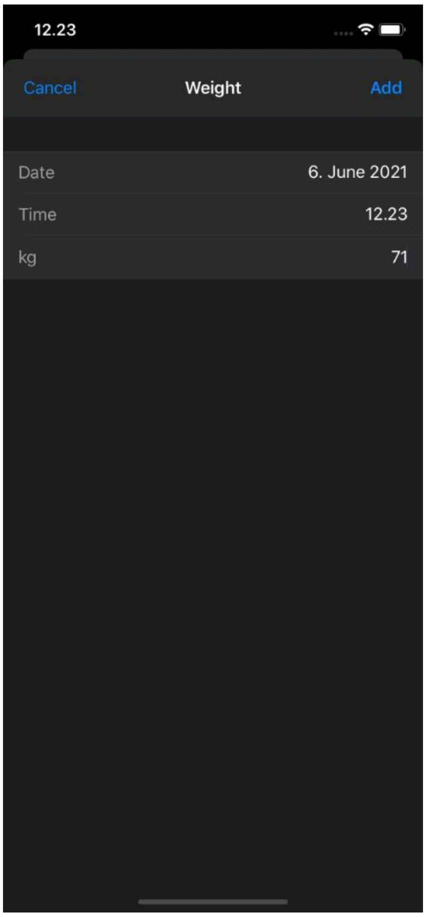
Weight

App Explanation:
Read and understand health data.

Figure 9. Apple Health prompting the user to accept permissions.

5.3.2 Handling similar data from two sources

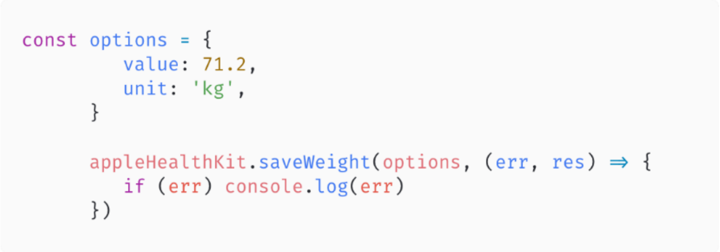
As MyNursie and Apple Health might support adding similar measurements such as height and weight, we can inspect how Apple Health handles same types of data coming from two sources. We will first add our weight directly via Apple Health (Figure 10).



Date 6. June 2021
Time 12.23
kg 71

Figure 10. Adding weight in Apple Health.

We will then add another weight result with react-native-health, using an options object with hardcoded value and unit (Figure 11).

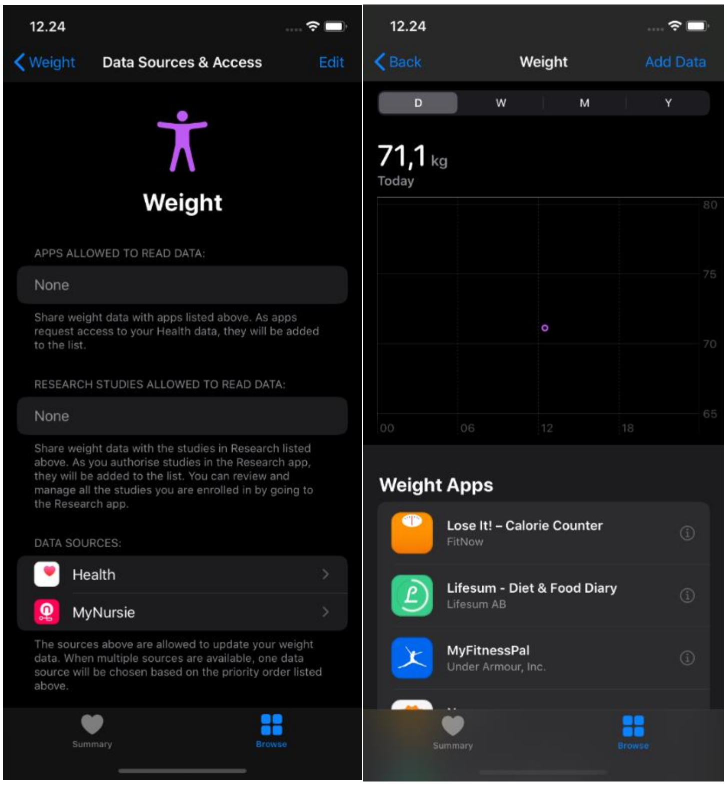


```
const options = {
          value: 71.2,
          unit: 'kg',
      }

      appleHealthKit.saveWeight(options, (err, res) => {
          if (err) console.log(err)
      })
```

Figure 11. Storing weight using react-native-health.

After adding data from both applications, if we inspect it in Apple Health, we can see that the data is in fact coming from two sources, and Apple Health displays the average of the measurements (Figure 12).



APPS ALLOWED TO READ DATA:

None

Share weight data with apps listed above. As apps request access to your Health data, they will be added to the list.

RESEARCH STUDIES ALLOWED TO READ DATA:

None

Share weight data with the studies in Research listed above. As you authorise studies in the Research app, they will be added to the list. You can review and manage all the studies you are enrolled in by going to the Research app.

DATA SOURCES:

Health

MyNursie

The sources above are allowed to update your weight data. When multiple sources are available, one data source will be chosen based on the priority order listed above.

71,1 kg
Today

Weight Apps

Lose It! – Calorie Counter
FitNow

Lifesum - Diet & Food Diary
Lifesum AB

MyFitnessPal
Under Armour, Inc.

Figure 12. Viewing the inputted weight data in Apple Health.

5.3.3 Adding new functionality

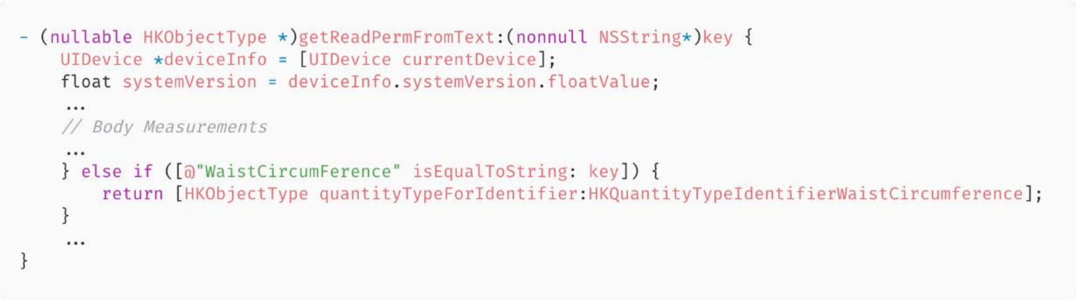
react-native-health is already a quite comprehensive library with plenty of useful functions out of the box, but we can leverage it further by adding features that do not yet exist. For demonstration purposes, implementing functionality for reading waist circumference from Apple health to MyNursie is covered.

The files that need modifying are following:

- RCTAppleHealthKit+TypesAndPermissions.m
- RCTAppleHealthKit+Methods_Body.m
- RCTAppleHealthKit+Methods_Body.h
- RCTAppleHealthKit.m
- Permissions.js
- index.d.ts

HealthKit documentation states, that waist circumference is a quantity sample. In short, this means that waist circumference samples are quantities with value and unit. The unit will be a length type, more closely covered under HKUnit in the documentation. It is also mentioned that these samples measure discrete values, using HKStatisticsQuery.

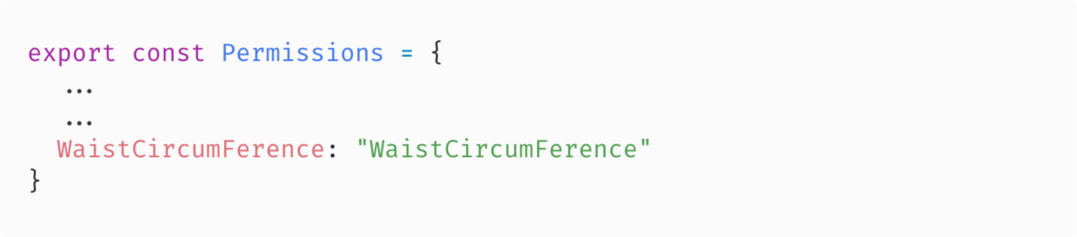
When enough information about the measurement at hand is gathered, we can start building our own function to read and write waist circumference functions between the two apps. Starting by setting permissions for read and write rights to this type of data, RCTAppleHealthKit+TypesAndPermissions.m needs to be edited.
Waist circumference is a body measurement, so the read right will be added under getReadPermFromText (Figure 13).



```
- (nullable HKObjectType *)getReadPermFromText:(nonnull NSString*)key {
    UIDevice *deviceInfo = [UIDevice currentDevice];
    float systemVersion = deviceInfo.systemVersion.floatValue;
    ...
    // Body Measurements
    ...
    } else if ([@"WaistCircumFerence" isEqualToString: key]) {
        return [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierWaistCircumference];
    }
    ...
}
```

Figure 13. Adding read permission for waist circumference.

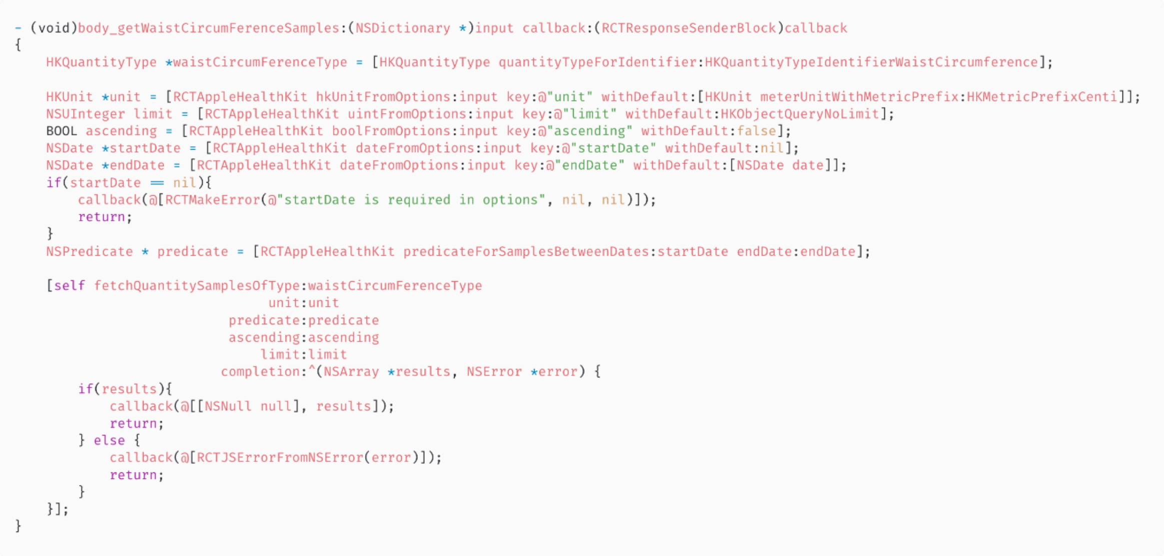
An entry for interacting with permissions using JavaScript is also required, and it will be placed into Permissions.js (Figure 14).



```
export const Permissions = {
   ...
   ...
  WaistCircumFerence: "WaistCircumFerence"
}
```

Figure 14. Adding entry to Permissions.js.

After the according permissions are added, a function for fetching waist circumference samples from Apple Health can be implemented. The core logic will reside in RCTAppleHealthKit+Methods_Body.m. Based on the earlier inspection of waist circumference in the official documentation, the method body_getWeightSamples can be used as a basis for this use case, as the types are a close match (Figure 15).



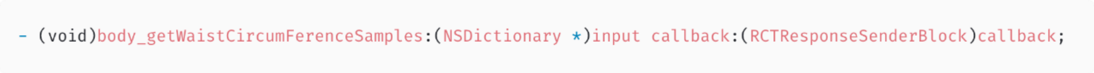
```
- (void)body_getWaistCircumFerenceSamples:(NSDictionary *)input callback:(RCTResponseSenderBlock)callback
{
    HKQuantityType *waistCircumFerenceType = [HKQuantityType quantityTypeForIdentifier:HKQuantityTypeIdentifierWaistCircumference];

    HKUnit *unit = [RCTAppleHealthKit hkUnitFromOptions:input key:@"unit" withDefault:[HKUnit meterUnitWithMetricPrefix:HKMetricPrefixCenti]];
    NSUInteger limit = [RCTAppleHealthKit uintFromOptions:input key:@"limit" withDefault:HKObjectQueryNoLimit];
    BOOL ascending = [RCTAppleHealthKit boolFromOptions:input key:@"ascending" withDefault:false];
    NSDate *startDate = [RCTAppleHealthKit dateFromOptions:input key:@"startDate" withDefault:nil];
    NSDate *endDate = [RCTAppleHealthKit dateFromOptions:input key:@"endDate" withDefault:[NSDate date]];
    if(startDate == nil){
        callback(@[RCTMakeError(@"startDate is required in options", nil, nil)]);
        return;
    }
    NSPredicate * predicate = [RCTAppleHealthKit predicateForSamplesBetweenDates:startDate endDate:endDate];

    [self fetchQuantitySamplesOfType:waistCircumFerenceType
                                unit:unit
                           predicate:predicate
                           ascending:ascending
                               limit:limit
                          completion:^(NSArray *results, NSError *error) {
        if(results){
            callback(@[[NSNull null], results]);
            return;
        } else {
            callback(@[RCTJSErrorFromNSError(error)]);
            return;
        }
    }];
}
```

Figure 15. Function for fetching samples from Apple Health.

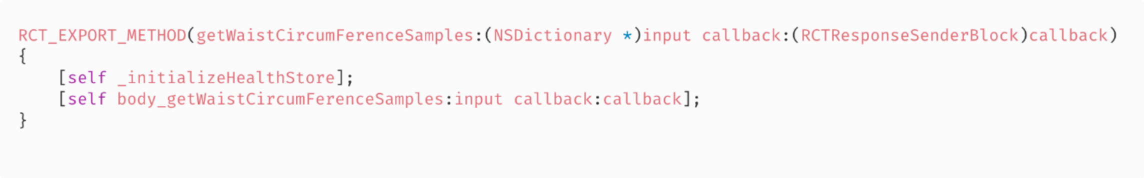
The only parts needing modificiation were the HKQuantityType and the unit, which was adjusted to use metric with a centimeter prefix. As the .m files are used for private parts of classes, the .h file is used for public declarations (Figure 16).



```
- (void)body_getWaistCircumFerenceSamples:(NSDictionary *)input callback:(RCTResponseSenderBlock)callback;
```

Figure 16. Declaring the function in RCTAppleHealthKit+Methods_Body.h.

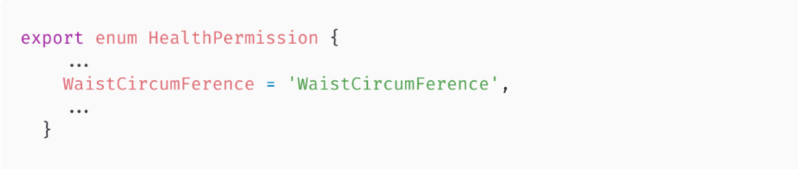
The next step is to export the native method to be used with JavaScript. The library does not currently do this automatically, so it must be done manually in the RCTAppleHealthKit.m file. It is simply done by using RCT_EXPORT_METHOD macro (Figure 17).



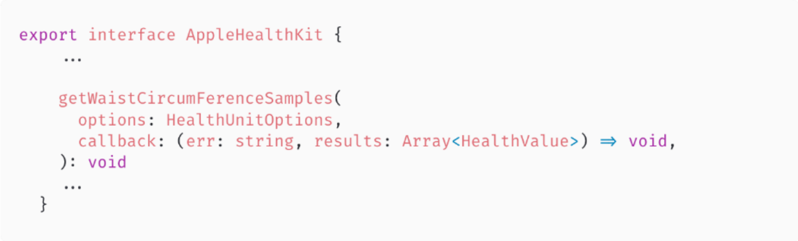
```
RCT_EXPORT_METHOD(getWaistCircumFerenceSamples:(NSDictionary *)input callback:(RCTResponseSenderBlock)callback)
{
    [self _initializeHealthStore];
    [self body_getWaistCircumFerenceSamples:input callback:callback];
}
```

Figure 17. Exporting the function for React Native.

The final steps are to add the function to the AppleHealthKit export in index.d.ts file and add the waist circumference permission to the HealthPermission constants (Figure 18 and 19).



```
export enum HealthPermission {
    ...
    WaistCircumFerence = 'WaistCircumFerence',
    ...
  }
```

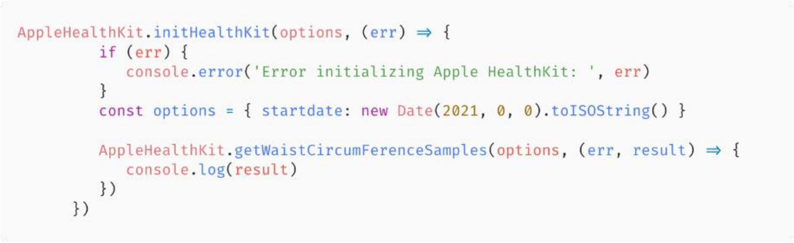


```
export interface AppleHealthKit {
    ...

    getWaistCircumFerenceSamples(
      options: HealthUnitOptions,
      callback: (err: string, results: Array<HealthValue>) => void,
    ): void
    ...
  }
```

Figure 18 and 19. Adding final exports.

The function is now accessible in JavaScript. For demonstration purposes we will call it straight after initializing HealthKit (Figure 20).

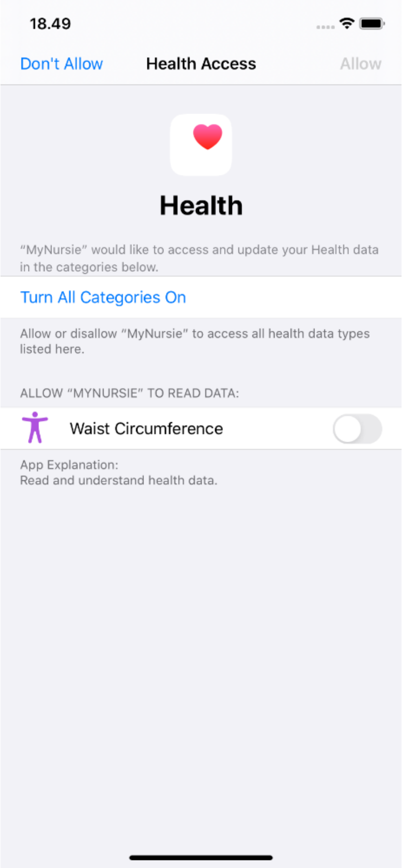


```
AppleHealthKit.initHealthKit(options, (err) => {
            if (err) {
               console.error('Error initializing Apple HealthKit: ', err)
            }
            const options = { startdate: new Date(2021, 0, 0).toISOString() }

            AppleHealthKit.getWaistCircumFerenceSamples(options, (err, result) => {
               console.log(result)
            })
        })
```

Figure 20. Calling waist circumference function in JavaScript.

Upon launching MyNursie with the newly configured function and calling it on start, Apple Health will detect that MyNursie is initializing Apple Health and requesting permissions to, as already seen in the introduction of the library (Figure 21).



“MyNursie” would like to access and update your Health data in the categories below.

Allow or disallow “MyNursie” to access all health data types listed here.

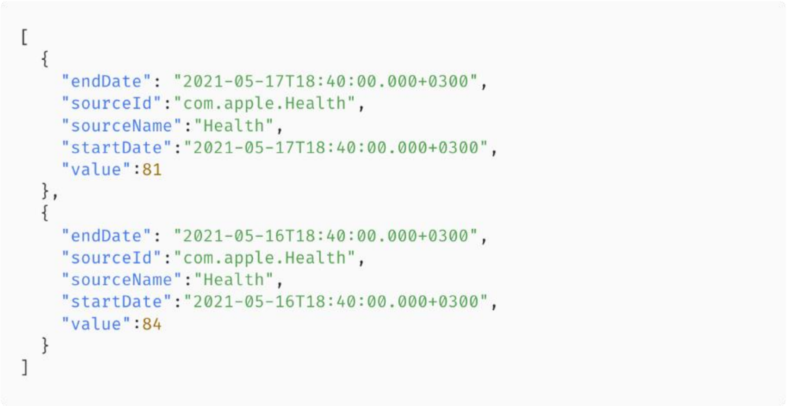
ALLOW “MYNURSIE” TO READ DATA:

Waist Circumference ○

App Explanation:
Read and understand health data.

Figure 21. Giving MyNursie permission to read data.

After the according permissions have been given, the result will be logged. A JSON array is returned consisting of the waist circumference values in the given span of time (Figure 22).



```
[
  {
    "endDate": "2021-05-17T18:40:00.000+0300",
    "sourceId":"com.apple.Health",
    "sourceName":"Health",
    "startDate":"2021-05-17T18:40:00.000+0300",
    "value":81
  },
  {
    "endDate": "2021-05-16T18:40:00.000+0300",
    "sourceId":"com.apple.Health",
    "sourceName":"Health",
    "startDate":"2021-05-16T18:40:00.000+0300",
    "value":84
  }
]
```

Figure 22. Waist circumference results returned by the function.

6 CONCLUSION

The goal of this thesis was to gather an overview and deeper understanding of mHealth applications and their markets to help Nursie Health decide, which applications we can benefit from. Based on these findings, an initial integration into Apple Health was done as a proof-of-concept. The result demonstrates that integrating into Apple Health is feasible and relatively simple to carry out in the near future.

Due to a relatively hectic spring in the company, the schedule was tight, and the goal was set to be a proof-of-concept instead of a more in-depth integration. The result is not as comprehensive as it could have been due to the mentioned schedule issues. Luckily, this does not impact the development of Nursie Health's application, because the timeline for integrating into third-party applications was already set to be not earlier than the end of 2021.

Other major challenge was the difficulty in comparing the different mHealth applications available for iOS and Android. Gathering market and revenue data about fitness tracking applications turned out to be tedious. Google Play Store and Apple's App Store did not necessarily provide much data beside download counts and reviews. During the making of the thesis, it was found out that to analyse competing apps more closely, there are platforms such as App Annie, Apptopia and Statista. They offer details such as average revenue per user, monthly active users, user churn rate and more. These solutions however would have been quite expensive to use, and the free plans did not offer much. In the future, it might prove to be useful for the company to subscribe to some of these platforms to acquire more details about the markets and competing applications.

Although there were challenges with the schedule and the comparison of mHealth applications, this thesis provided a lot of information and acts as a basis for a more complex integration. It showed what the integration process is like and helps me to keep developing our app with the integration in mind. Since the background work is already done, the real integration will be easier to carry out.

This thesis did not necessarily provide me new programming experience, but more importantly helped me to gather a deeper understanding and knowledge of mHealth applications. This aids me to see our application in a different light now that I know about various aspects of mHealth applications and which factors make a good or bad mHealth application. The process of making this thesis will also help me as a developer to be able to make competent suggestions on improving our application, now that I have a deeper understanding of the field we are entering with our solution.

7 SOURCES

Apple Inc. (2021). *CareKit*. Retrieved from https://developer.apple.com/design/human-interface-guidelines/carekit

Apple Inc. (2021). *About the HealthKit Framework*. Retrieved from https://developer.apple.com/documentation/healthkit/about_the_healthkit_framework

Apple Inc. (2021). *Building tools to advance research and care. One app at a time.* Retrieved from https://www.researchandcare.org/

Apple Inc. (2021). *iOS - Health*. Retrieved from https://www.apple.com/ios/health/

Apple Inc. (2021). *ResearchKit*. Retrieved from http://researchkit.org/

Apple, Inc. (2007). *iPhone Premieres This Friday Night at Apple Retail Stores*. Retrieved from https://www.apple.com/newsroom/2007/06/28iPhone-Premieres-This-Friday-Night-at-Apple-Retail-Stores/

Bullington, A. (2019). *Lessons I learned while building in React Native*. Retrieved from https://www.freecodecamp.org/news/lessons-i-learned-while-building-in-react-native-917cb7bb5993/

Chen, J. (2021). *Android Operating System*. Retrieved from https://www.investopedia.com/terms/a/android-operating-system.asp

Deng, Z., Hong, Z., Ren, C., Zhang, W., & Xiang, F. (2018). *What Predicts Patients'Adoption Intention Toward mHealth Services in China: Empirical Study.* Retrieved from https://mhealth.jmir.org/2018/8/e172/PDF

Dogtiev, A. (2021). *Top App Analytics Companies (2021)*. Retrieved from https://www.businessofapps.com/marketplace/app-analytics/

European Commission. (2014). *Green Paper on mobile Health ("mHealth")*. Retrieved from https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=5147

Facebook, Inc. (2021). *Core Components and Native Components*. Retrieved from https://reactnative.dev/docs/intro-react-native-components

Facebook, Inc. (2021). *Setting up the development environment*. Retrieved from https://reactnative.dev/docs/environment-setup

Google. (2020). *New Updates from Google Fit.* Retrieved from https://support.google.com/fit/thread/38325431/new-updates-from-google-fit?hl=en

Google. (2021). *Google Fit*. Retrieved from https://www.google.com/fit/

Google. (2021). *Google Play*. Retrieved from https://play.google.com/store

Grand View Research. (2021). *mHealth Apps Market Size, Share & Trends Analysis Report By Type (Fitness, Medical), By Region (North America, APAC, Europe, MEA, Latin America), And Segment Forecasts, 2021 - 2028*. Retrieved from https://www.grandviewresearch.com/industry-analysis/mhealth-app-market

Gupta, D. (2018). *React Native vs Native: What to choose for App Development*. Retrieved from https://appinventiv.com/blog/react-native-vs-native-apps/

Gurupur, V., & Wan, T. (2017). *Challenges in implementing mHealth interventions: a technical perspective*. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5583043/pdf/mh-03-2017.07.05.pdf

Horton, N. (2020). *Introductions: HealthKit, ResearchKit and CareKit*. Retrieved from https://medium.com/kinandcartacreated/introductions-healthkit-researchkit-and-carekit-d72e2ac9ce2

International Data Corporation. (2021). *Smartphone Market Share*. Retrieved from https://www.idc.com/promo/smartphone-market-share/os

International Telecommunication Union (ITU). (2021). *Mobile cellular subscriptions*. Retrieved from https://data.worldbank.org/indicator/IT.CEL.SETS

Intertrust. (2020). *How secure are today's mobile healthcare and medical apps?* Retrieved from https://www.intertrust.com/resources/healthcare-app-security-report-2020/

Isakova, T. (2021). *Privacy and Security in mHealth applications [guide]*. Retrieved from https://www.businessofapps.com/insights/privacy-and-security-in-mhealth-applications-guide/

Kenton, W. (2020). *Apple iOS*. Retrieved from https://www.investopedia.com/terms/a/apple-ios.asp

Killburn, B. (2021). *What's new for Wear*. Retrieved from https://blog.google/products/wear-os/wear-io21/

Kruhlyk, Y. (2020). *Expo vs Vanilla React Native: What to Choose for Your Project*. Retrieved from https://apiko.com/blog/expo-vs-vanilla-react-native/

Larson, R. S. (2018). *A Path to Better-Quality mHealth Apps.* Retrieved from https://mhealth.jmir.org/2018/7/e10414/PDF

Liew, M., Zhang, J., See, J., & Ong, Y. (2019). *Usability Challenges for Health and Wellness Mobile Apps: Mixed-Methods Study Among mHealth Experts and Consumers*. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6372932/

Macelai, V. (2021). *React Native Health*. Retrieved from https://docs.react-native-health.com/

Nield, D. (2020). *A beginner's guide to Google Fit and Apple Health*. Retrieved from https://www.popsci.com/beginners-guide-google-fit-apple-health/

Nielsen, J. (2012). *Usability 101: Introduction to Usability*. Retrieved from https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Nursie Health. (2021). *MyNursie*. Retrieved from https://www.mynursie.com/en/home/

Nursie Health. (2021). *Nursie Health*. Retrieved from https://www.nursiehealth.com/

O'Boyle, B. (2019). *Apple Health app and HealthKit: What are they and how do they work?* Retrieved from https://www.pocket-lint.com/apps/news/apple/131130-apple-healthkit-and-health-app-how-they-work-and-the-medical-records-you-need

Reem, K., Aliki, P., Muhammad, I., Zahra, H., Pedro, B., & Jennifer, B. (2017). *Awareness and Use of mHealth Apps: A Studyfrom England.* Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5597158/pdf/pharmacy-05-00033.pdf

Research2Guidance. (2017). *mHealth Economics – How mHealth App Publishers Are Monetizing Their Apps.* Retrieved from https://research2guidance.com/product/mhealth-economics-how-mhealth-app-publishers-are-monetizing-their-apps/

Rowland, P. S., Fitzgerald, E. J., Holme, T., Powell, J., & McGregor, A. (2020). *What is the clinical value of mHealth for patients?* Retrieved from https://www.nature.com/articles/s41746-019-0206-x.pdf

Stack Overflow. (2020). *2020 Developer Survey*. Retrieved from https://insights.stackoverflow.com/survey/2020

Stewart, C. (2020). *Growth in the number of medical apps downloaded during the COVID-19 pandemic by country in 2020*. Retrieved from https://www.statista.com/statistics/1181413/medical-app-downloads-growth-during-covid-pandemic-by-country/

Stewart, C. (2021). *Global digital health market by major segment 2015-2025*. Retrieved from https://www.statista.com/statistics/387867/value-of-worldwide-digital-health-market-forecast-by-segment/

Trnka, D. (2018). *Mobile App Development: React Native vs Native (iOS, Android)*. Retrieved from https://medium.com/mop-developers/mobile-app-development-react-native-vs-native-ios-android-49c5c168045b

U.S Food And Drug Administration. (2019). *Policy for Device Software Functions and Mobile Medical Applications.* Retrieved from https://www.fda.gov/media/80958/download

Vailshery, L. S. (2018). *Global Apple iPhone sales from 3rd quarter 2007 to 4th quarter 2018 (in million units)*. Retrieved from https://www.statista.com/statistics/263401/global-apple-iphone-sales-since-3rd-quarter-2007/

WHO. (2011). *mHealth: New horizons for health through mobile technologies: second global survey on eHealth*. Retrieved 3 12, 2021, from https://www.who.int/goe/publications/goe_mhealth_web.pdf