



**Software Development Final Project**  
**Wearable using Movesense Hardware**

**Software Development Project**  
**Research report**

**Student: Yinglong Yu**

**Supervisor: Joseph Kehoe**

**Submission Date: 17/04/2023**

## Table of content

<b>1. Abstract</b>	<b>4</b>
<b>2. Introduction</b>	<b>5</b>
<b>3. Business Idea</b>	<b>6</b>
3.1. The rehabilitation training application	6
<b>4. Target Users</b>	<b>7</b>
4.1. Healthy person	7
4.2. The Patient	7
4.3. The Medical Institution's expert	8
<b>5. Project Description</b>	<b>9</b>
5.1. The application user operating progress	9
5.1.1. Wear the Movesense Active sensor	9
5.1.2. Connect Application	9
5.1.3. Tracking healthy and movement indicators	10
5.1.4. Remote diagnosis with experts	10
5.2. The medical care expert's operation progress	10
5.2.1. Login rehabilitation training application's website	10
5.2.2. Remote diagnosis with the patient	10
<b>6. The Research on Hardware</b>	<b>11</b>
6.1. Movesense sensor	11
6.2. Movesense architecture	12
6.3. The data dealing	12
6.4. The Movesense sensor basic facts	12
6.4.1. Class IIa medical device accessory	13
6.4.2. Measurement capabilities	13
<b>7. The Technical Highlights of Movesense Active Sensor</b>	<b>15</b>
7.1. The Sensor Programming Basic	15
7.1.1. Language: C/C++ however with limitations	15
7.1.1.1. Dynamic memory allocation	15
7.1.1.2. Standard Template Library	16
7.1.2. All hardware and low-level access is via Movesense Representational state transfer (REST) API	16
7.1.2.1. Representational state transfer	16
7.1.2.2. Publish / Subscribe	16
7.1.3. Fully asynchronous	17
7.1.3.1. Call structure	17
7.1.3.2. Automatic power optimization	17
7.2. Internal memory	17
7.3. Size and Weight	17
<b>8. Similar application Research</b>	<b>19</b>
8.1. My Autonom Health (Movesense medical sensor showcase)	19
8.2. The Sports Injury Rehabilitation	20
8.2.1. Body Injuries selection	20

8.2.2. The personal Daily rehabilitation training schedule	21
8.2.3. The training phase of rehabilitation	21
8.2.4. The rehabilitation training program management	22
<b>9. The project application design scheme</b>	<b>23</b>
9.1. The connector	23
9.2. The sensor	23
9.3. The mobile	23
9.4. The cloud	24
9.5. The cooperation between those facilities	24
<b>10. Technologies Research</b>	<b>25</b>
10.1. Application about Native	25
10.1.1.1. Kotlin	26
10.1.1.2. Java	26
10.1.1.3. Objective-C	27
10.1.1.4. Swift	27
10.2. Web Application	28
10.3. Cross-Platform Native Frameworks	29
10.3.1. Flutter	30
10.3.1.1. The Dart Language	31
10.3.1.2. The Flutter engine	31
10.3.1.3. The Flutter Foundation library	32
10.3.2. Xamarin	32
10.3.2.1. C# lanaguage	33
10.3.3. React Native	33
10.3.3.1. Javascript	34
10.4. SQL	35
10.5. No-SQL	35
10.6. SQLite	35
10.7. Firebase real-time database	36
10.8. GitHub	37
10.9. Doxygen	37
10.10. Docker	39
<b>Conclusion</b>	<b>40</b>
<b>References</b>	<b>41</b>

# 1. Abstract

Nowadays, people are more likely to pay attention to their health status, and when people talk about the topic of “Wearable”, it can relate to a watch on the wrist or a necklace on the neck, with science and technology development, the “health” and “wearable” can be associated naturally in nowadays, for example, there are variable smartwatches in the market that allow the user to measure basic heart rate and other movement indicators.

Wearable medical sensors as the more professional wearable sensors can help people to monitor various health and rehabilitation training movement indicators accurately and with more operation possibilities.

## 2. Introduction

This project research will illustrate a report on “Wearable using Movesense Hardware” that contains

- The research details of wearable Movesense sensors(hardware, technology, etc)
- The project application development’s relevant aspects(development platform, technology, program language, etc).

Investigating the App’s market target user and the currently released similar app in the market will also be discussed.

## 3. Business Idea

### 3.1. The rehabilitation training application

As a software application product using the “Movesense Active” as hardware technical support.

It allows the application’s user to create a personal health&sports profile and create relevant rehabilitation training phases based on instructions.

This application can help users record and track their various health&sports indicators. it enables users to manage rehabilitation training progression, when one of the training phases is completed, the second phase will become available with a higher difficulty level than the first phase. In addition, it also can be used by medical care experts as a remote diagnosis service.

The rehabilitation training application can be used in different physical parts of the rehabilitation training programs, such as injured joint rehabilitation, posture correction, and so on

## 4. Target Users

The rehabilitation training application is a mobile platform offering personal health and movement data tracking, analysing, and remote diagnosis services.

There are three main target user categories for this project application.

### 4.1. Healthy person

The person who needs to monitor various health and movement indicators for self-health check, and sports data tracking purposes.

There is a popular trend nowadays, more and more people pay more attention to their body's health and daily exercise performance, such as heartbeat rate, blood pressure, daily steps, average running speed, etc.

### 4.2. The Patient

The patient needs rehabilitation training and measuring health indicators. There exists a higher proportion of senior people who need to do rehabilitation training for recovering physical body function, and the sporter and fitness personnel have possibly gotten injured during the competition and training progression.

### 4.3. The Medical Institution's expert

Medical institutions such as hospitals, and clinics have relevant physical rehabilitation training experts, experts can use the rehabilitation training application for remote diagnosis of patients' physical rehabilitation status purposes.

The medical care expert can use the rehabilitation training application one-to-one to monitor a patient's health indicators and the progression of the rehabilitation training.



## 5. Project Description

The user can wear the Movesense sensor on different parts of the body, such as the chest, wrist, ankle, etc.

The project application can connect with the Movesense Active sensor by BlueTooth, and it can read the raw data that is delivered from the wearable sensor and then transfers raw data into various readable charts and summaries for application users and medical care experts.

The application-specific operation progress under below:

### 5.1. The application user operating progress

#### 5.1.1. Wear the Movesense Active sensor

The patient wears the Movesense Active sensor on their wrist, ankle, or chest by using the sensor for various accessories.

#### 5.1.2. Connect Application

Firstly, the patient needs to activate the Movesense Active sensor and then use the mobile phone to connect with the Movesense Active sensor.

#### 5.1.3. Tracking healthy and movement indicators

The patient uses the project application to monitor the health status and movement data with friendly readable charts and

summaries and also can follow the app instructions to achieve rehabilitation training.

#### 5.1.4. Remote diagnosis with experts

The patient shares the procession data of rehabilitation training schemes with medical care experts as a remote one-to-one diagnosis.

### 5.2. The medical care expert's operation progress

#### 5.2.1. Login rehabilitation training application's website

The medical care expert login the rehabilitation training application's website by using a desktop computer or tablet personal computer.

#### 5.2.2. Remote diagnosis with the patient

The medical care expert can review the patient-shared health indicators data and rehabilitation training complete progression for one-to-one remote diagnostics with the patient.

## 6. The Research on Hardware

### 6.1. Movesense sensor

The project's development purpose is to develop a medical care rehabilitation training application that allows users to use the medical sensor as a hardware device support for personal body health indicators detection and rehabilitation training use. so the hardware device needs to have in-built elements for monitoring health indicators measurement and movement data detection.

“The Movesense: Open wearables & sensor development platform for sports, health, research manufacturing, and more. Made by the sports watch company Suunto. ” (“Movesense grows independent from Suunto”)



Figure 1 Movesense Active sensor screenshot  
*(“Suunto open development environment offers rapid prototyping of Bluetooth LE motion sensing solutions”)*

The Movesense Active sensor as a measurable device with programmable, wireless, electrocardiogram (ECG) characteristics can measure users' health and movement indicators data to medical applications and host devices for healthcare solution usage.

This project application selects the Movesense medical sensor as hardware support, the architecture as below.

## 6.2. Movesense architecture

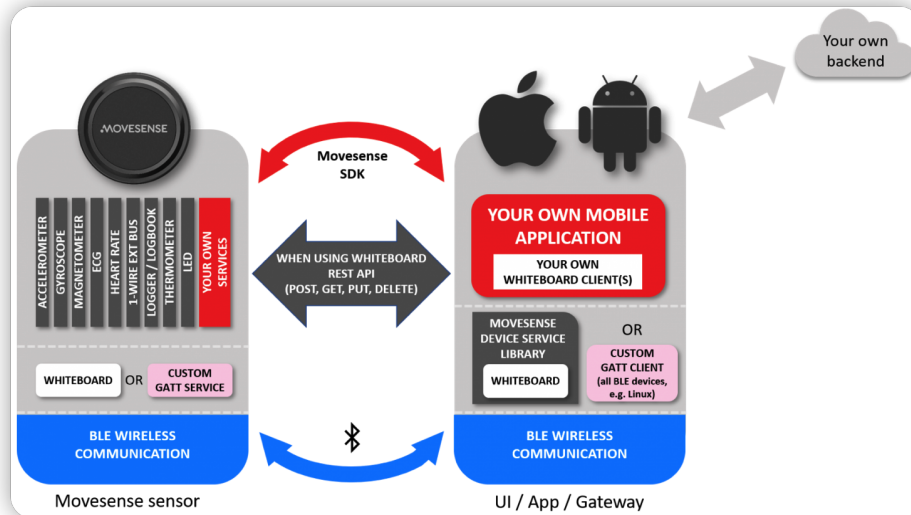


Figure 2. The architecture of the Movesense sensor  
(*Movesense Medical Sensor — Wearable Sensor for Physiological Parameters*)

The Movesense architecture allows developers to have high freedom and flexibility to develop various custom applications through offered software development kit.

## 6.3. The data dealing

The Movesense sensor is capable of measuring, processing, and sending measurement data wirelessly to smartphones via Bluetooth 4.0 / 5.0 radio depending on the firmware version.

## 6.4. The Movesense sensor basic facts

### 6.4.1. Class IIa medical device accessory

- Wristband
- Clothing clip

- Surface mount
- Heart rate chest strap
- Electrocardiogram patch electrode
- Heart rate electrode element for shirts & tops

Class IIa device definition: “Devices that are installed in the body in the short term (usually 60 minutes to 30 days)” (Chmielewska)



Figure 3. The Movesense sensor accessory screenshot  
 (“Movesense sensor accessory”)

#### 6.4.2. Measurement capabilities

- In-built 9-axis movement sensor: acceleration, gyroscope, magnetometer.
- Heart rate (bpm).
- R-R intervals(the time elapsed between two successive R waves of the QRS signal on the electrocardiogram).
- Single channel electrocardiogram (non-medical), Bluetooth heart rate profile.
- Temperature measurement.

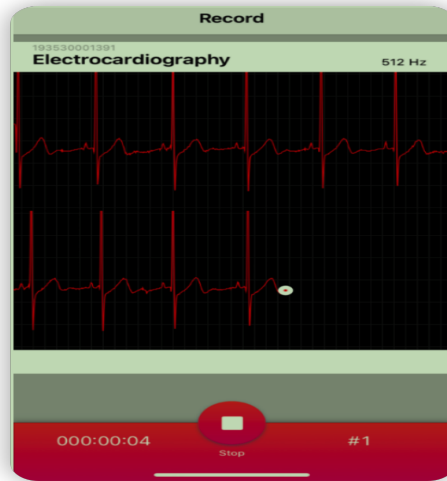


Figure 4. Real-time view of single channel ECG recorded with the chest belt sensor Movesense

*(“Rik Vullings, Valentina Corino and Linda M. Eerikäinen”)*

## 7. The Technical Highlights of Movesense Active Sensor

### 7.1. The Sensor Programming Basic

As the hardware device, Movesense Sensor provides open API and developer tool that allows developers to build custom apps, such as health indicator monitor, rehabilitation training app, etc.

#### 7.1.1. Language: C/C++ however with limitations

- No dynamic memory allocation
- No Standard Template Library (STL)
- Limited resources(Ram: 64kb, Flash: 512 kb)

##### 7.1.1.1. Dynamic memory allocation

The operating system offers the program a block of main memory when it requests.

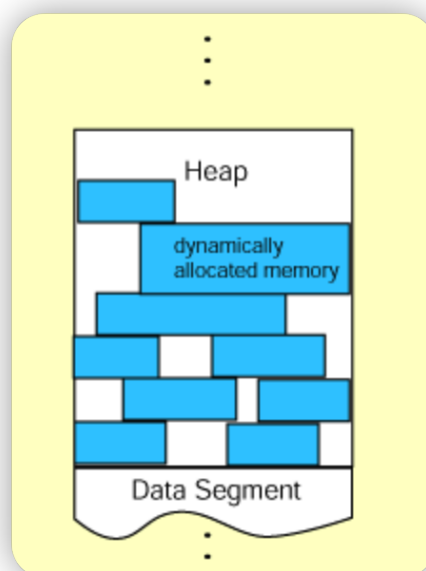


Figure 5. Dynamic memory allocation structure  
(Patanaik)

### 7.1.1.2. Standard Template Library

The Standard Template Library (STL) is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc.

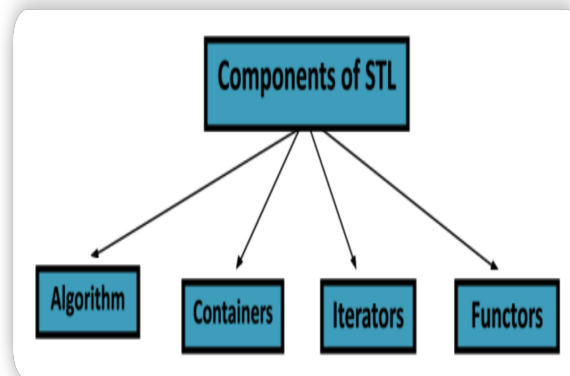


Figure 6. STL Components

*(Great Learning team)*

### 7.1.2. All hardware and low-level access is via Movesense Representational state transfer (REST) API

- GET/POST/PUT/DELETE
- Publish-subscribe extension for data streams

#### 7.1.2.1. Representational state transfer

REST is a software architecture style for the World Wide Web proposed to facilitate the transfer of information between different software and programs.

#### 7.1.2.2. Publish / Subscribe

Publish / Subscribe is one of several messaging paradigms supported by Universal Messaging.



### 7.1.3. Fully asynchronous

- Call structure
- Automatic power optimization

#### 7.1.3.1. Call structure

The data of the header and the arrangement can be transmitted with a call

#### 7.1.3.2. Automatic power optimization

Use electronic design automation tools to optimize the power digital design consumption.

## 7.2. Internal memory

The sensor has in-built memory for logging data, reducing user transfer(store, retrieve) data operating times.

The Movesense sensor contains 386 kB Electrically Erasable Programmable Read-Only Memory (EEPROM) that is organized as a circular buffer when the memory gets full, the old data gets overwritten.

## 7.3. Size and Weight

The small and lightweight sensor can sustain mechanical impacts and submersion.

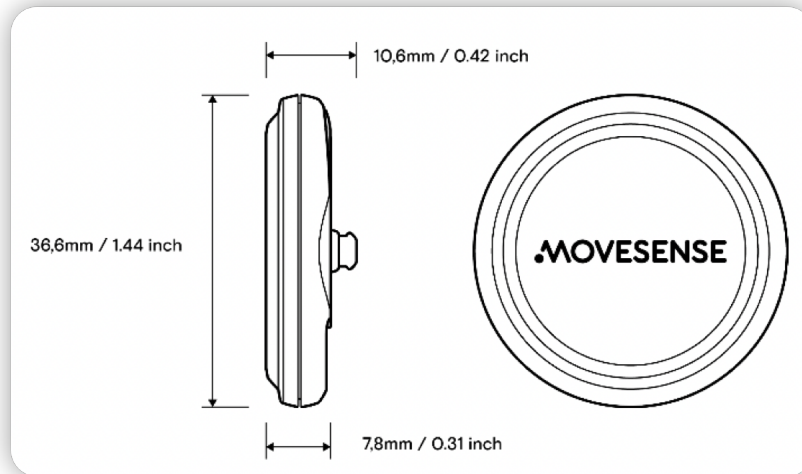


Figure 7. Movesense Sensor details

(*MOVESENSE SENSOR*)

- 36.6mm/1.44" dia. x 10.6mm/0.42" thick
- Weight 9.4g / 0.33oz with battery
- Water resistant to 30m/100ft

## 8. Similar application Research

### 8.1. My Autonom Health (Movesense medical sensor showcase)



Figure 8. My Autonom Health Application screenshot  
(Autonom Health — Movesense.)

My Autonom Health is a 24 hours measurement heart rate application that uses Movesense Medical Sensor as hardware support, “an application with a reliable and highly sensitive screening instrument with which the general health of a person can be ascertained”. (“Autonom Health — Movesense”)

My Autonom Health can measure the heart beats during the whole day and displays the data by using a LifeFire diagram.

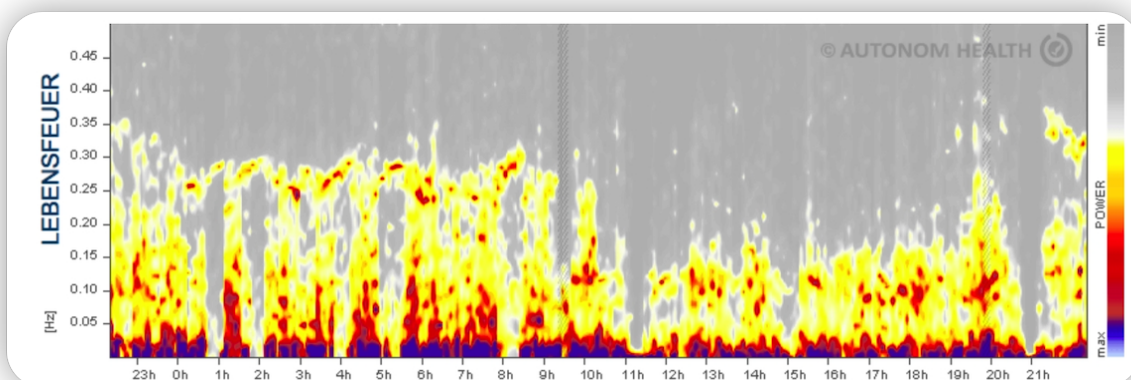


Figure 9. LifeFire screenshot  
(Autonom Health — Movesense.)

The diagram represents the users' holistic health status and extends the variety of numerical analysis parameters.

## 8.2. The Sports Injury Rehabilitation



Figure 10. The screenshot of the app logo from the google play store  
(*Sports Injury Rehabilitation - Apps on Google Play.*)

The Sports injury rehabilitation application offers rehabilitation training instructions to users for step-by-step finish the progress of various body parts.

### 8.2.1. Body Injuries selection

The user can select the different injury body parts for rehabilitation training.

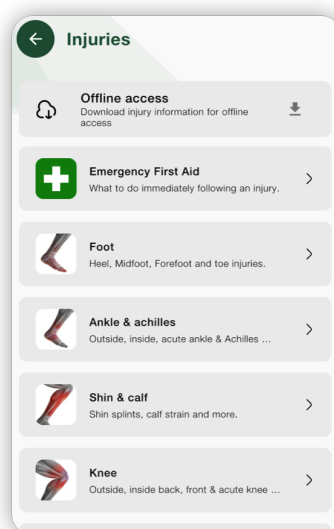


Figure 11. Rehabilitation training options for various body parts

(Sports Injury Rehabilitation - Apps on Google Play.)

### 8.2.2. The personal Daily rehabilitation training schedule

The personally changeable rehabilitation exercise scheme allows users to flow the training and modify the plan as well.

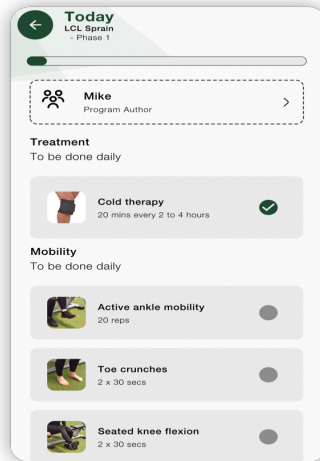


Figure 12. Rehabilitation schedule screenshot  
(Sports Injury Rehabilitation - Apps on Google Play.)

### 8.2.3. The training phase of rehabilitation

There are different phases for a user to complete the progression as below picture.

When the user finished one phase, then the next phase is available for the user to complete, with the step-by-step principle will showing a more clear training plan for the user.

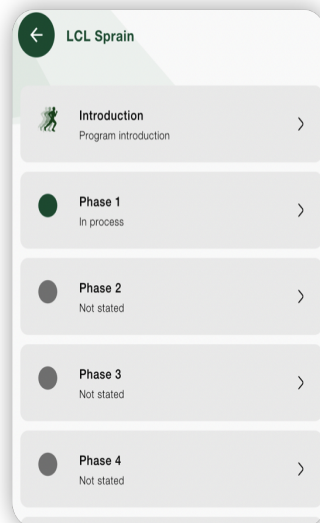


Figure 13. The Phase of rehabilitation screenshot

(Sports Injury Rehabilitation - Apps on Google Play.)

#### 8.2.4. The rehabilitation training program management

The personal program management function allows the user to manage their rehabilitation plan, the right-side clickable button can manage the current program.

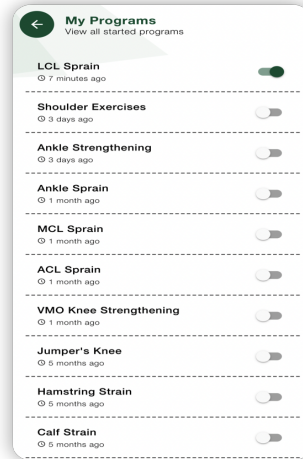


Figure14. The program management  
(Sports Injury Rehabilitation - Apps on Google Play.)

## 9. The project application design scheme

The project application development design scheme contains a connector, sensor, mobile, and cloud.

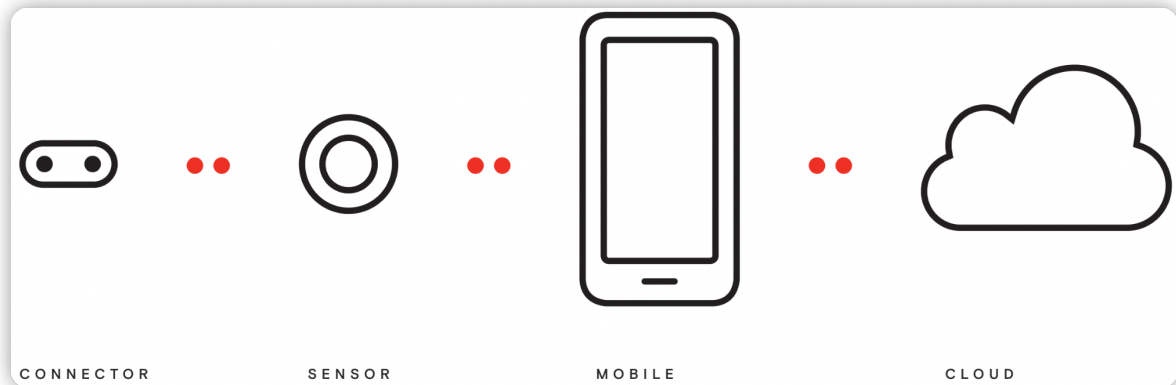


Figure 15. The project application of different parts' relationship  
(*“Overview — Movesense”*)

### 9.1. The connector

The smart version connector has an individual ID which has an awareness characteristic with the sensor and also can suitably equip on different places, such as the wrist, chest, ankle, etc.

### 9.2. The sensor

The Movesense sensor with small size, long endurance, IP68 Protection, and track movement locus characteristics, the rehabilitation user wears the sensor to complete the training progression.

### 9.3. The mobile

As the application platform involves Android and IOS applications and plays a deliver role in the data transformation, in detail that the mobile receives the data from the sensor and visualizes the raw data into readable charts and summaries.

The mobile can upload the relevant data to the cloud for medical care expert remote diagnosis patient health and rehabilitation training status usage.

#### 9.4. The cloud

The cloud as a backend data storage part of the project software application also takes responsibility for displaying the users' various healthy indicators and the rehabilitation training complete progression in the web browser for doctor telemedicine.

#### 9.5. The cooperation between those facilities

- The connector as the accessory of the Movesense sensor supports the Movesense sensor equipped in different places.
- The Movesense sensor is the measurement apparatus for the rehabilitation training mobile application, offering a variety of measurement raw data.
- The mobile application is the data display platform for the visible health and movement indicators data summaries. In addition, as a bridge between the Movesense sensor user and the medical care expert, the user's measurement health and movement data is delivered to the cloud.
- The cloud is the data storage platform for the Movesense sensor measurement data and mobile application data



## 10. Technologies Research

The software application development necessary platform and technologies will be illustrated below, which involve the development language, database, framework, and so on.

As the software application operating environment, the platform selection will play an important role in the whole flowing software development progress, owing to the different platform's unique characteristics, the software development scheme, cost, development length, and even the target market also can be influenced.

### 10.1. Application about Native

The native application is a software program that is written to work on a particular device or platform.

There are two main mobile operating system platforms.

- IOS is Apple company-developed operating system.
- Android as an operating system is developed by Google company.

Android with higher customizability and file transfer easier than IOS.



Figure 16. Native App example image screenshot  
(*Native Vs Hybrid App Development In 2022*, n.d.)

There are several programming languages for developing a native application, for example.

- Kotlin
- Java
- Objective-C
- Swift

Android mobile application development language as below.

#### 10.1.1.1. Kotlin

“Kotlin is a free, open-source, statically typed “pragmatic” programming language”. (“Kotlin for server side”)

The purpose of designing Kotlin is to interoperate with Java and Java Virtual Machine.

It combines object-oriented and functional programming features, and over 60% of professional Android developers use it.

```
// Hello World Program

fun main(args : Array<String>) {
    println("Hello, World!")
}
```

Figure 17. “Hello World” Kotlin code example

(“When I can’t even start Console.WriteLine(“Hello world!”))

#### 10.1.1.2. Java

Java is a popular programming language and as core technology is used to develop mobile applications, web applications desktop applications, and so on. Java has an extensive library and a variety of tools that allow the development progress more effective.

```
// Your First Program

class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Figure 18. “Hello World” Java code example  
 (“When I can't even start Console.WriteLine("Hello world!")”)

IOS mobile application development language is as below.

#### 10.1.1.3. Objective-C

Objective-C was the primary programming language for developing IOS apps and OS X. It is a superset of the C programming language and uses object-oriented capabilities.

```
#import <Foundation/Foundation.h>

int main(int argc, const char * argv[]) {
    @autoreleasepool {
        // insert code here...
        NSLog(@"Hello, World!");
    }
    return 0;
}
```

Figure 19. “Hello World” Objective-C code example  
 (“When I can't even start Console.WriteLine("Hello world!")”)

#### 10.1.1.4. Swift

Swift as a programming language supports multiple programming paradigms and compilations, similar to Objective-C also can write software based on Apple’s variety of device operating systems.

```
// Swift "Hello, World!" Program
print("Hello, World!")
```

Figure 20. "Hello World" Swift code example  
 ("When I can't even start Console.WriteLine("Hello world!"))

As a comparison between different program languages.

Different program languages with code structure and syntax respectively to achieve the same output "Hello world", the Swift code example only needs one line to print the "Hello world", which is much easy than the other three examples. however, application programming language selection also needs to consider other features.

## 10.2. Web Application

The web application is an application program operating in a web browser interface and stored on a remote server.



Figure 21. Web application development example image screenshot  
 (Native Vs Hybrid App Development In 2022.)

Web application development can divide into two parts, the front end, and the back end.

The front end display as a website, and the development front end involves the technology as below.

- Javascript
- XML,
- HTML5
- CSS, and so on.

The back-end development relevant programming languages are as below.

- .NET
- C#
- PHP
- JAVA
- Python

Based on the software development scheme, only one platform for all target users is not a reality, so the Cross-Platform Native should be considered and it is a good way to develop software with high cost performance.

### 10.3. Cross-Platform Native Frameworks

There are three main cross-platform native popular frameworks.

- Flutter
- Xamarin
- React Native

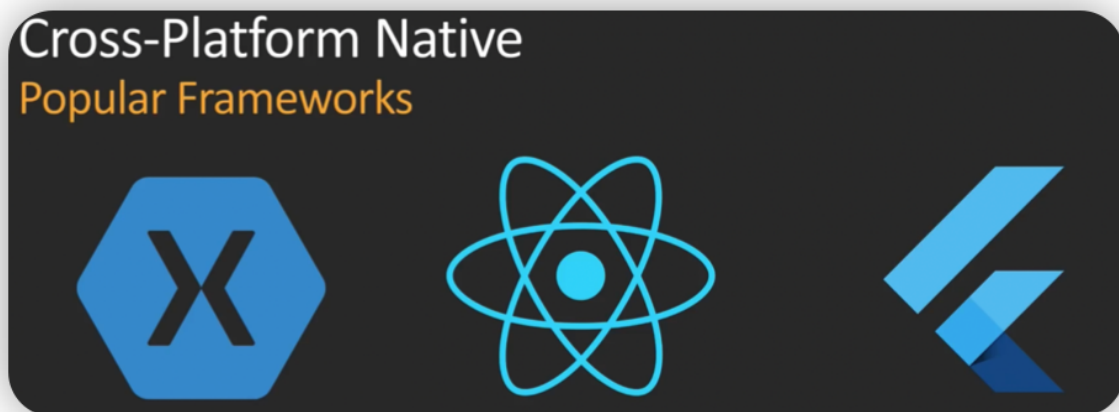


Figure 22. Cross-Platform Native Popular Frameworks screenshot

(Native Vs Hybrid App Development In 2022.)

### 10.3.1. Flutter

Flutter is a software development kit for mobile application usage developed by Google, as a cross-platform framework suitable for different operating systems, such as Android, IOS, Windows, macOS, Linux, and so on.

Cross-Platform Native  
Flutter

Framework Owner	Google	✓
UI / UX	Bespoke	✓
Development Speed	Fast	✓
Maintenance Cost	Medium	✓
Performance	High	✓
Programming Language	Dart	✓
Maturity	Young	✓
Feature Access	Complete*	✓
Open Source	Yes	✓

\*Updates released via NPM

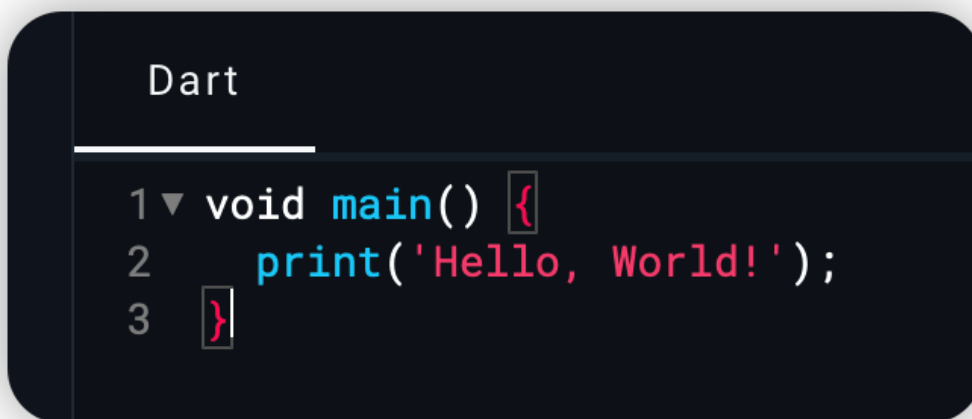
Figure 23. Flutter details screenshot  
(Native Vs Hybrid App Development In 2022.)

The Flutter components are as below.

- Dart Language
- Flutter engine
- Foundation library

#### 10.3.1.1. The Dart Language

The Dart language is a client-optimized language for fast app development on any platform with free and open-source advantages and is supported by Google.



```
Dart
1 void main() {
2     print('Hello, World!');
3 }
```

Figure 24. “Hello World” Dart code example

#### 10.3.1.2. The Flutter engine

The Flutter engine is developed primarily in C++, and it is Google’s software development kit with a single codebase for developing fast user experience for mobile, web, and desktop platforms, the Flutter framework, provides a timely update, reactive framework and a rich set of layout, and foundation widgets.

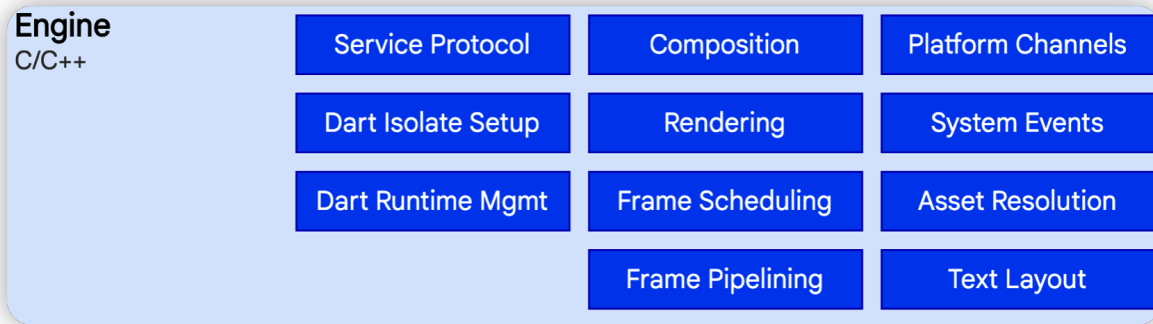


Figure 25. The Flutter engine layer (“Flutter architectural overview | Flutter”)

### 10.3.1.3. The Flutter Foundation library

The foundation library uses dart as a programming language, with APIS to communicate with the Flutter engine. It also provides basic classes and functions to construct applications.

### 10.3.2. Xamarin

Xamarin is a Microsoft-owned software company, it is a Cross- Platform development software, and it uses C# programing to build a sharable codebase.

The screenshot shows the Xamarin details page. On the left, it says 'Cross-Platform Native' with the Xamarin logo. Below this are the Microsoft logo (four colored squares) and the Visual Studio logo (purple infinity symbol), separated by a plus sign. On the right, there is a table of attributes:

Framework Owner	Microsoft	✓
UI / UX	Native	✓
Development Speed	Fast	✓
Maintenance Cost	Low	✓
Performance	High	✓
Programming Language	.NET, C#, F#	✓
Maturity	Mature	✓
Feature Access	Complete*	✓
Open Source	Yes	✓

\*Updates released through Visual Studio

Figure 26. Xamarin details screenshot (Native Vs Hybrid App Development In 2022.)



Xamarin has a variety of tools that allows the developer to program the native Android, iOS, and Windows apps and share code across different platforms, such as Windows, macOS, and Linux.

#### 10.3.2.1. C# language

C# is an object-oriented programming language derived from C and C++, It inherits the advantage of C and C++ and removes the complexity in the meantime.

```
C# Program.cs > {} wow
1  using System;
2  using System.Collections.Generic;
3
4  namespace wow
5  {
6      class Program
7      {
8          Console.WriteLine("Hello world!");
9      }
10 }
```

Figure 27. "Hello World" C# code example  
("When I can't even start Console.WriteLine("Hello world!")")

#### 10.3.3. React Native

React Native is an open-source mobile application development framework developed by Meta Platforms Owned by Facebook. It is a cross-platform framework, which supports the development of native applications for iOS, Android, Windows, and macOS.

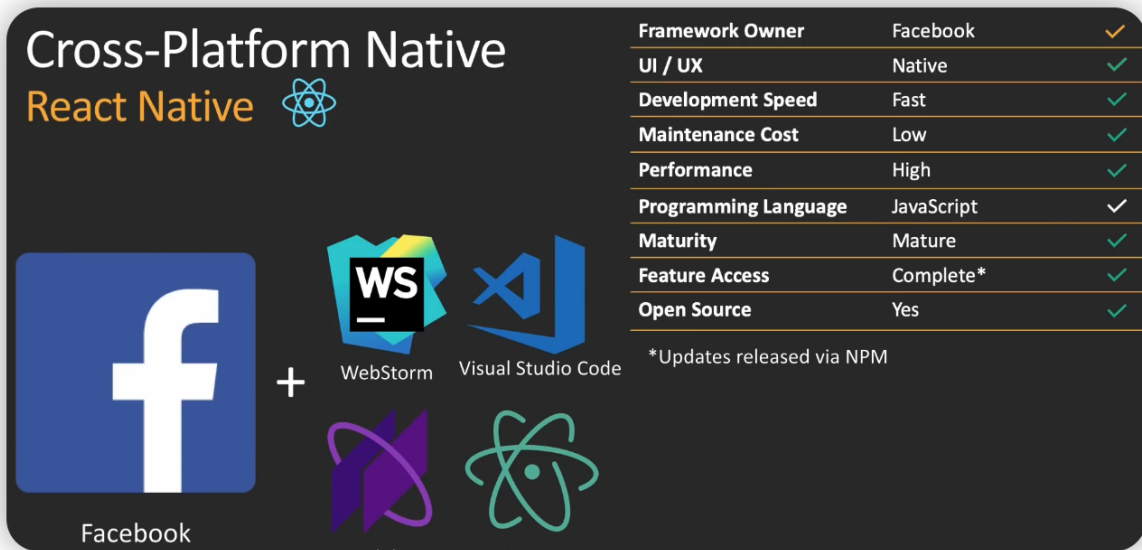


Figure 28. React Native Framework details screenshot  
(Native Vs Hybrid App Development In 2022.)

React Native is written in JavaScript and combines with Hyper Text Markup language, and it is used in both frontend and backend, which allows developers to program the full-stack web application.

### 10.3.3.1. Javascript

JavaScript is a programming language based on prototypes and first-class functions. It supports object-oriented principles, imperative programming, and functional programming.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JavaScript Hello World Example</title>
  <script>
    alert('Hello, World!');
  </script>
</head>
<body>
</body>
</html>

```

Figure 29. JavaScript “Hello World” code example

## 10.4. SQL

Structured Query Language (SQL) is a standardized domain-specific programming language that is used to manage data held in a relational database management system, such as create, update, modify, delete, receive, update, upload data, and so on.

SQL examples such as “DB2”, “MySQL”, “Oracle Database”, etc.

The project application involves database management and data analysis, MySQL can be useful in the Movesense Sensor measurement data storage aspect.

## 10.5. No-SQL

“NoSQL is a non-relational database, meaning it allows different structures than a SQL database and more flexibility to use a format that best fits the data.” (Anderson)

No-SQL examples such as “Mongo DB”, “Redis”, “Couch DB”, etc.

## 10.6. SQLite

SQLite is a database with server-less and self-contained characteristics and does not require a client and server architecture to interact over a network.

SQLite supports datatypes that are limited such as Blob, Integer, Null, Text, Real.

The size of the SQLite library is very small around 250 KB, and SQLite can be stored as a single file that is not needed to configure.

Owing to the characteristics, the following situations will suit using SQLite:

- Small standalone apps development
- Project with less scalability requirement
- Quick directly edit database file
- Basic testing and development usage

## 10.7. Firebase real-time database

As is a cloud-hosted NoSQL database, Firebase supports iOS, Android, Web, C++ and Unity platforms and allows developers to store and sync data in real-time.

Compare with traditional databases need to work with a request/response model.

Fire the Realtime Database uses data synchronization and subscriber mechanisms instead of typical HTTP requests.

The below situations suit using Firebase real-time database:

- Short development time
- In-time management database
- Cross-platform product development

## 10.8. GitHub

“GitHub is a code hosting platform for version control and collaboration. It lets developers and others work together on projects from anywhere”.

Developers can use GitHub to manage project file systems easily and safely, GitHub allows the user to upload and download files and folders, and as a positive result, GitHub can be used as a private cloud for developers working anywhere and anytime. (“GitHub”)

```
$ git init
Initialized empty Git repository in /tmp/tmp.IMBYSY7R8Y/.git/
$ cat > README << 'EOF'
> Git is a distributed revision control system.
> EOF
$ git add README
$ git commit
[master (root-commit) e4dcc69] You can edit locally and push
to any remote.
1 file changed, 1 insertion(+)
create mode 100644 README
$ git remote add origin git@github.com:cdown/thats.git
$ git push -u origin master
```

Figure 30. A representative git session, demonstrating the initialization of a repository, and the addition of a file and a remote.

*(Down, Chris)*

## 10.9. Doxygen

“Doxygen is a documentation generator and static analysis tool for software source trees.”

It is written in C++ and supports various other popular programming languages such as C, Objective-C, C#, PHP, Java, and Python.

(“Doxygen: Doxygen”)

The developer can use the “doxywizard” command to call the graphical user interface frontend for running and configuring Doxygen, as a result, the developer’s project code can be summarized into an HTML file, and the structure of the code and the relevance of comments can be reviewed.

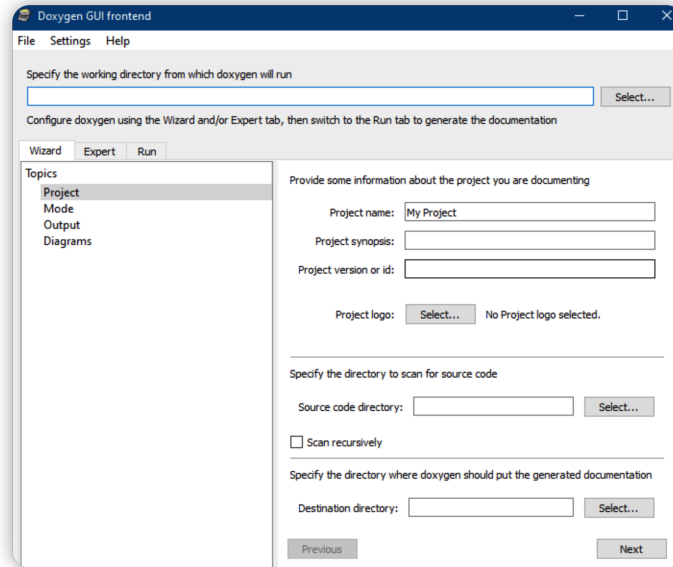


Figure 31. The main windows of Doxygen  
(Avenwedde)

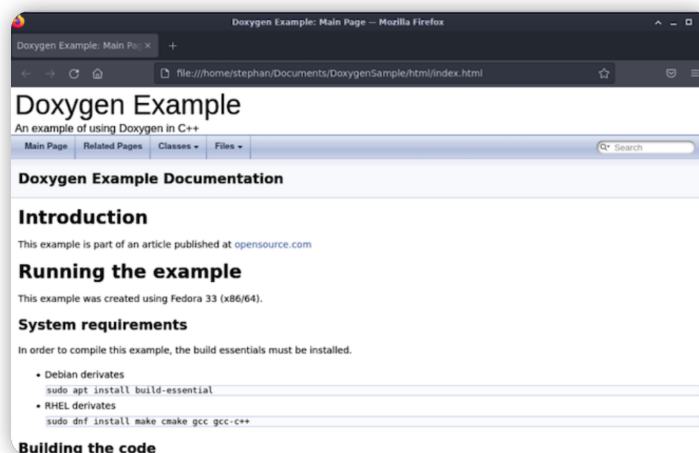


Figure 32. The main page of Doxygen  
(Avenwedde)

## 10.10. Docker



Figure 33. The Docker Logo

Docker is a software platform that allows developers to build, test, and deploy applications quickly.

In the project application design, Docker plays a role as a Movesense sensor programming environment.

“Docker can package an application and its dependencies in a virtual container that can run on any Linux, Windows, or macOS computer”.  
(Docker)

This enables the application to run in a variety of locations, for example, in a public or private cloud.

## Conclusion

This report has preliminarily completed the research on the project application target user, hardware relevant details, software development platform, and technology, A basic plan for project application development is formed.

This project is based on “Movesense” as wearable hardware to develop a rehabilitation training application to transform the unformed monitoring data from the wearable Movesense sensor into a visualizable interface for helping senior patient rehabilitation training and data sharing with the medical care expert as a remote diagnosis.

Based on the project application requirement and research results, the development technologies decide to use Android Studio as the Movesense sensor programming environment and cooperate it with Java as the programming language, because the Movesense sensor official development kit and Movesense-mobile-API uses Java as the programming language. As a result, using Java for project application development will be more suitable and efficient.



## References

Swift.org - Welcome to Swift.org, available on

<https://www.swift.org>. Accessed 12 November 2022.

“About Objective-C.” *Apple Developer*, 17 September 2014, available on

<https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>. Accessed 12 November 2022.

Anderson, Benjamin. “SQL vs. NoSQL Databases: What's the Difference?” *IBM*, 12 June 2022, available on <https://www.ibm.com/cloud/blog/sql-vs-nosql>. Accessed 12 November 2022.

“Autonom Health — Movesense.” *Movesense*, available on

<https://www.movesense.com/showcase/showcase-autonom-health/>. Accessed 12 November 2022.

Avenwedde, Stephan. “Document your source code with Doxygen on Linux.”

*Opensource.com*, 26 May 2022, available on

<https://opensource.com/article/22/5/document-source-code-doxygen-linux>. Accessed 13 November 2022.

Chmielewska, Matylda. “EU MDR: everything you need to know about Medical Device Regulation.” *Spyrosoft*, 2021, available on

<https://spyro-soft.com/blog/eu-mdr>. Accessed 13 November 2022.

Down, Chris. “Git.” *Wikipedia*, 28 10 2013, available on

[https://en.wikipedia.org/wiki/Git#/media/File:Git\\_session.svg](https://en.wikipedia.org/wiki/Git#/media/File:Git_session.svg). Accessed 13 November

2022.

“Doxygen: Doxygen.” *official website*, available on

<https://doxygen.nl>. Accessed 13 November 2022.

“Flutter architectural overview | Flutter.” *Flutter documentation*, available on

<https://docs.flutter.dev/resources/architectural-overview>. Accessed 13 November 2022.

Great Learning team. “Standard Template Library in C++.” *Great Learning*, 25 March

2021, available on

<https://www.mygreatlearning.com/blog/standard-template-library-in-c/>. Accessed 13

November 2022.

“Hello World.” *GitHub Docs*, available on

<https://docs.github.com/en/get-started/quickstart/hello-world>. Accessed 13 November 2022.

“Kotlin for server side.” *Kotlin*, 7 December 2021, available on

<https://kotlinlang.org/docs/server-overview.html>. Accessed 13 November 2022.

Lipponen, Petri. “Introduction to Movesense Programming.” *Movesense*, 6 November

2018, available on

<https://www.movesense.com/wp-content/uploads/2018/11/Introduction-to-Movesense-2018-11-06.pdf>. Accessed 13 November 2022.

Minnick, Brandon. “Choosing The Best Mobile App Framework.” *SlideShare*, available on

<https://www.slideshare.net/BrandonMinnickMBA/>. Accessed 15 November 2022.

“The Movesense-API.” *The Movesense-API*, available on

<https://bitbucket.org/movesense/movesense-device-lib/src/master/MovesenseCoreLib/resources/movesense-api/>. Accessed 15 November 2022.

“Movesense Medical Sensor — Wearable sensor for physiological parameters.”

available on <https://www.movesense.com/movesense-medical-sensor/>. Accessed 15

November 2022.

“MOVESENSE SENSOR.” *Movesense*, 12 2020, available on

<https://www.movesense.com/wp-content/uploads/2020/12/Movesense-Spec-Sheet-12-2020>

[0.pdf](#). Accessed 16 November 2022.

“Native Vs Hybrid App Development In 2022.” *Addevice*, available on <https://addevice.io/blog/native-vs-hybrid-app-development-in-2020/>. Accessed 16 November 2022.

“Overview — Movesense.” *Movesense*, available on <https://www.movesense.com/overview/>. Accessed 16 November 2022.

Patanaik, Chhanda. “Memory Allocation with COBOL - Mainframe....” *LinkedIn*, 12 May 2017, available on <https://www.linkedin.com/pulse/memory-allocation-cobol-chhanda-charan-patanaik>. Accessed 18 November 2022.

“Sports Injury Rehabilitation - Apps on Google Play.” *Google Play*, available on [https://play.google.com/store/apps/details?id=com.sportsinjuryclinic.rehab.program&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.sportsinjuryclinic.rehab.program&hl=en_US&gl=US). Accessed 18 November 2022.

“Validation Study Proves Movesense Medical's Accuracy in Measuring Heart Rate Variability — Movesense.” *Movesense*, Movesense, 22 April 2022, available on <https://www.movesense.com/news/2022/04/validation-study-proves-movesense-medicals-accuracy-in-measuring-heart-rate-variability/>. Accessed 18 November 2022.

“Wearable technology — Movesense.” *Movesense*, available on <https://www.movesense.com/technology/>. Accessed 18 November 2022.

“When I can't even start Console.WriteLine("Hello world!"); : r/csharp.” *Reddit*, 21 March 2021, available on [https://www.reddit.com/r/csharp/comments/m9vizt/when\\_i\\_cant\\_even\\_start\\_consolewritelinehello\\_world/](https://www.reddit.com/r/csharp/comments/m9vizt/when_i_cant_even_start_consolewritelinehello_world/). Accessed 18 November 2022.

Docker 2013, Docker overview, available on [https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)) Accessed 13 January 2023.

Docker 2023, Docker official website, available on <https://www.docker.com/>. Accessed 13 January 2023.

S, Edward. “What to Use – SQLite vs MySQL? In-Depth Analysis.” *Hostinger*, 21 December 2022, available on

<https://www.hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference/>. Accessed 17 February 2023.

Djermanović, Dean. "Saving Data on Android, Chapter 12: Introduction to Firebase Realtime Database." *Kodeco*, available on

<https://www.kodeco.com/books/saving-data-on-android/v1.0/chapters/12-introduction-to-firebase-realtime-database>. Accessed 17 February 2023.