



Software Development Final Project
Wearable using Movesense Hardware

Design Document

Student: Yinglong Yu

Student Number: C00253508

Supervisor: Joseph Kehoe

Submission Date: 17/04/2023

Table of content

1. Introduction	3
2. Technology selections	4
2.1 Docker	4
2.2 Java	5
2.3 Movesense sensor API	5
2.4 Android Studio	6
2.5 Firebase real-time database	6
2.6 The advantage of the combination above Technologies	7
3. Application Flow Chart	8
4. Application screenshots	9
4.1 New Account Sign Up & Sign In	9
4.2 Menu interface	11
4.2 User Details	12
4.3 Physical Rehabilitation training Schemes selection	13
4.4 Rehabilitation Scheme instructions	14
4.5 Linear Acceleration line chart	16
4.6 Remote diagnosis	18
4.7 User Personal Details	20
5. Sequence diagram	21
5.1 Login sequence diagram	21
5.2 Manage Rehabilitation Site Selection	22
5.3 Remote diagnosis sequence diagram	22
5.4 Measurement data analysis	23
6. Movesense sensor raw data transfer screenshot	23
7. Firebase firestore of the application data structure	24
8. Bibliography	26

1. Introduction

This document is a design document based on the Research Report document and Functional Specification with more specific application design and software application development details.

Wearable using Movesense Hardware as a Rehabilitation mobile application, the primary function is to help users formulate personal physical rehabilitation schemes and manage relevant training at home.

As the first iteration of the project application design document. this document sets up an application development scheme and will be modified during further development.

The following content illustrates the project application development progression in detail, such as the application development platform, programming languages, application different function screenshots, application class diagram sequence diagram, etc.

2. Technology selections

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

2.1 Docker



Figure 1. Docker image screenshot

Docker is a software platform that allows programmers 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. This enables the application to run in a variety of locations, such as on-premises, in a public or private cloud.

2.2 Java



Figure 2. Java screenshot

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.

Owing to the Movesense sensor programming language used Java, so the further project application programming language changed from Dart to Java in order to have better cooperation between the project application and Movesense sensor, and can reduce the application development debugging difficulty.

2.3 Movesense sensor API

All the Movesense devices implement the Movesense API. This REST-like API is defined, managed, and released by the Movesense organization.

The Movesense API enables developers to programmatically interact with the Movesense open sensor development platform for measuring movement and physiology. The API offers GET, PUT, POST, and DELETE actions for exchanging data with the Movesense sensor, with functions for accelerometer, gyroscope, magnetometer, temperature, ECG, HR, and more.

Owing to the project application development needs the cooperation with hardware support-Movesense sensor, so the Movesense sensor API is necessary for the development progression.

2.4 Android Studio

Android Studio provides app builders with an integrated development environment (IDE) optimized for Android apps.

Android Studio features

- With Multipreview API
- Compose design tools
- IntelliFgent code editor
- Flexible build system
- Android app bundle
- Easily emulate any device

2.5 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

2.6 The advantage of the combination above Technologies

As the Android application program language, increasing the possibility of software interaction is Java's greatest feature.

The Android Studio is designed as an Android application suitable development environment with powerful technology features support.

The Firebase real-time database allows developers to manage data in real-time without worrying about the networking code part, which can increase the application development efficiency.

3. Application Flow Chart

The Flow Chart below describes the application operation progression and the relationship between the application's different functions.

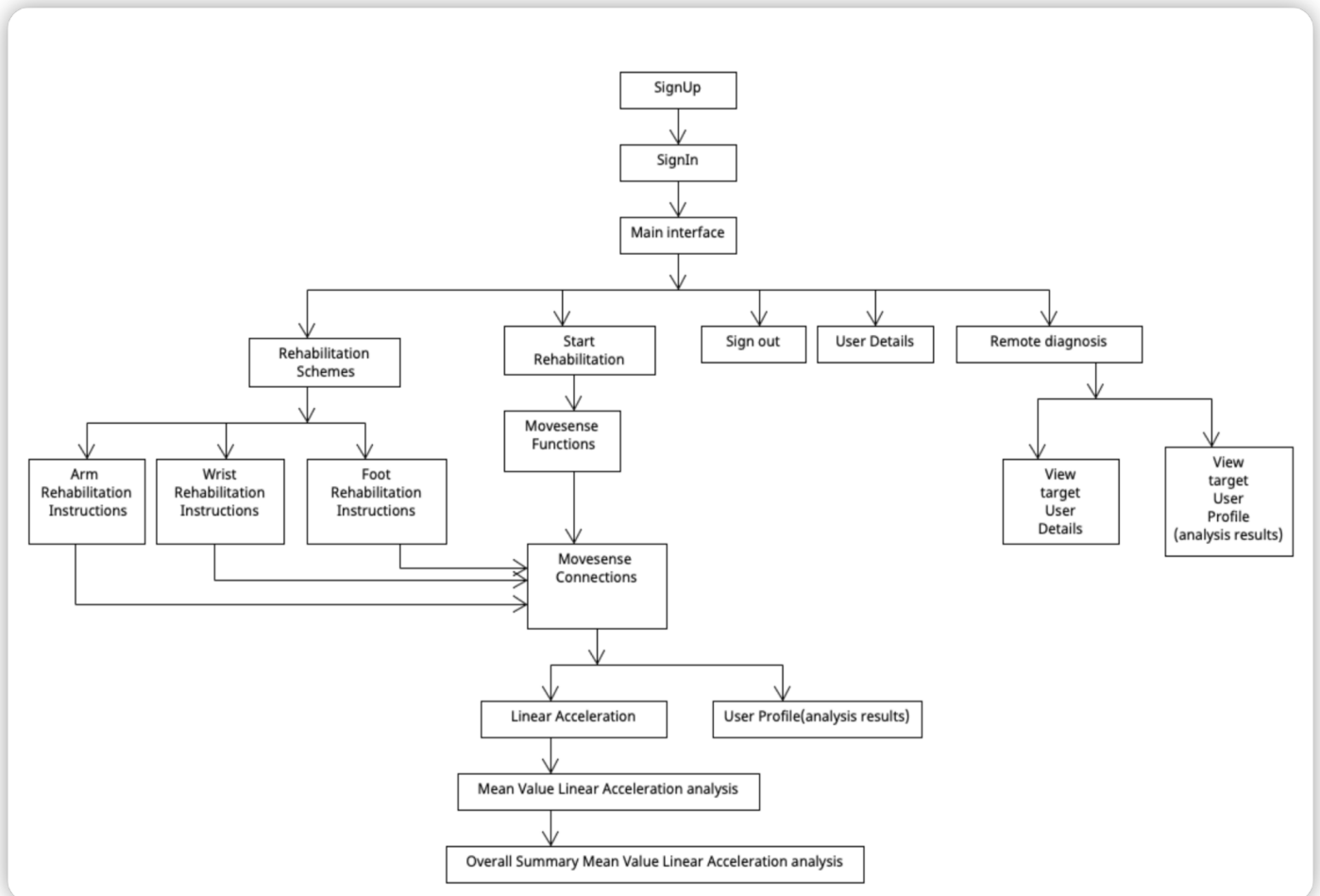


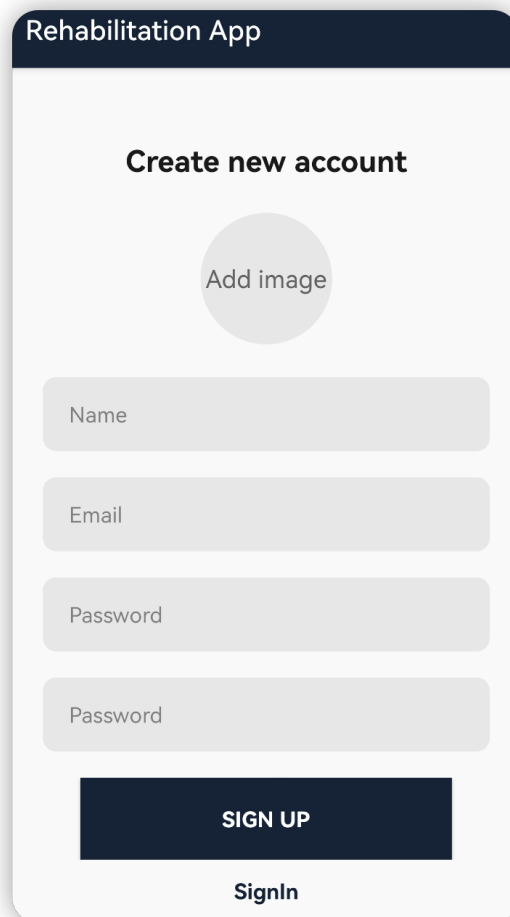
Figure 3. Application Flow Chart Example

4. Application screenshots

The project application with different functional interfaces that designs and shows in the below draft design screenshots.

The project application will be based on the below draft design to develop and improve.

4.1 New Account Sign Up & Sign In



The image shows a mobile application interface for creating a new account. At the top, there is a dark blue header with the text "Rehabilitation App". Below the header, the main content area is white and contains the following elements: a bold heading "Create new account", a circular placeholder for a profile picture with the text "Add image" inside, four text input fields labeled "Name", "Email", "Password", and "Password" (repeated), a dark blue button with the text "SIGN UP", and a link labeled "SignIn" at the bottom.

Figure 4. New Account SignUp

The application user needs to create a new account before using the application.

In the create new account interface, the user fills in the relevant required information, such as email address, password, name, and so on, then presses the SignUp button, and the account data is transferred to the database. A new account is created.

After the user has a personal account, then the user can log into the application and use the application by inputting the account name and password.

Rehabilitation App

Welcome Back
Login to continue

Email

Password

SIGNIN

Create new account

Figure 5. New Account SignIn

4.2 Menu interface

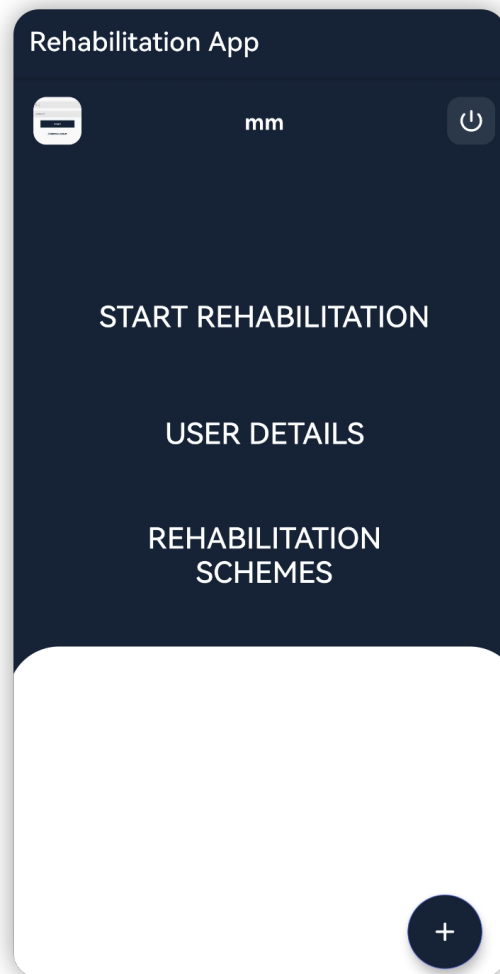


Figure 6. Rehabilitation app menu interface screen

The menu interface screen shows the project application involves various functions, such as Start Rehabilitation, User Details, Rehabilitation Schemes, Remote Diagnosis, SignOut.

When click one of above function, then the current page will redirect the new page with more functions.

4.2 User Details

Rehabilitation App

Please enter below information

Age:

please enter your age here

Gender

please enter your gender here

Illness History

IllnessHistory

Rehabilitation Plan

please enter your plan details

SAVE AND UPDATE

Figure 7. The user profile screen

The user can enter the 'User Details' interface to review the user's personal information details, such as Age, Gender, Illness History, Rehabilitation Plan. in the meantime, if the user wants to Save and Update button.

4.3 Physical Rehabilitation training Schemes selection

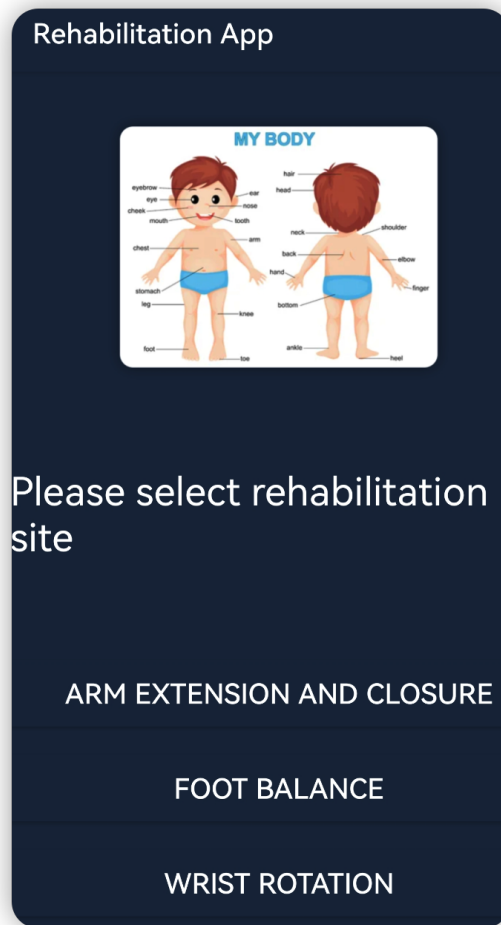


Figure 8. Rehabilitation training site selection screen

In this screen, the application user can based on their individual body injury status select rehabilitation sites for the following training.

4.4 Rehabilitation Scheme instructions

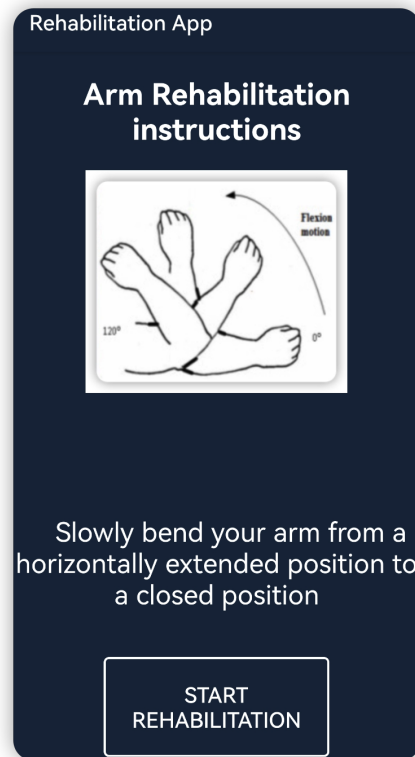


Figure 9. Arm Rehabilitation Scheme Instructions.

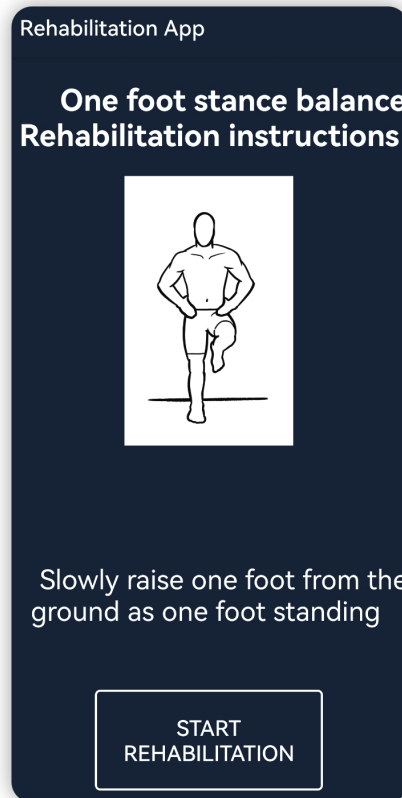


Figure 10. Foot Rehabilitation Scheme Instructions.

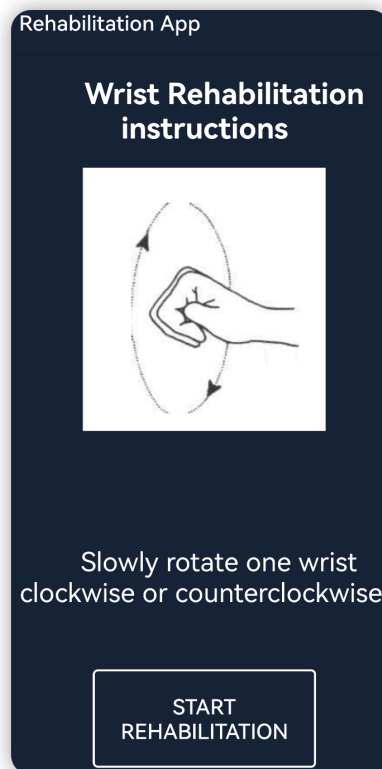


Figure 11. Wrist Rehabilitation Scheme Instructions.

There are three different body sites can be selected to do the rehabilitation training, each of them have individual training instructions. The user can follow the instructions to start Training.

4.5 Linear Acceleration line chart

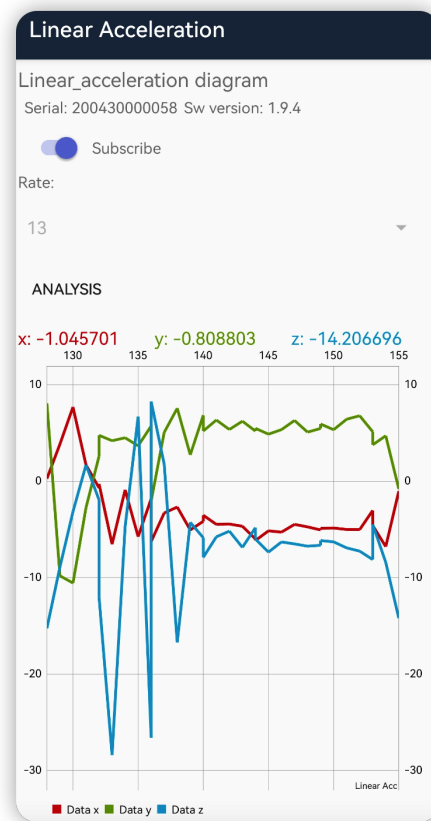


Figure 12. Movesense sensor measurement data analysis screen.

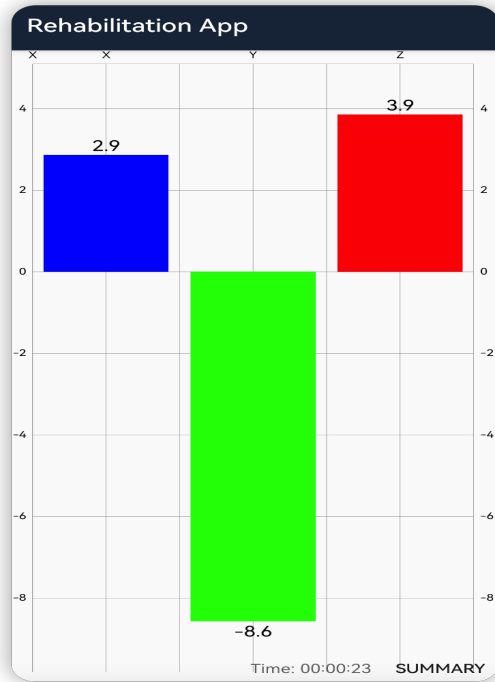


Figure 13. Movesense sensor measurement data Mean Value analysis bar Chart.

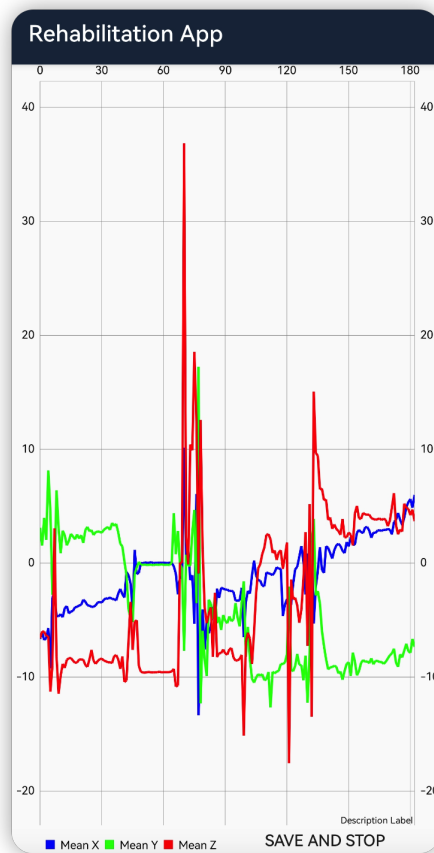


Figure 14. Movesense sensor measurement data overall Mean Value Summary line Chart.

The Movesense sensor has in-built measurement elements, for example, movement sensor: acceleration, gyroscope, and magnetometer.

The user can track their movement data with friendly readable charts and summaries.

4.6 Remote diagnosis

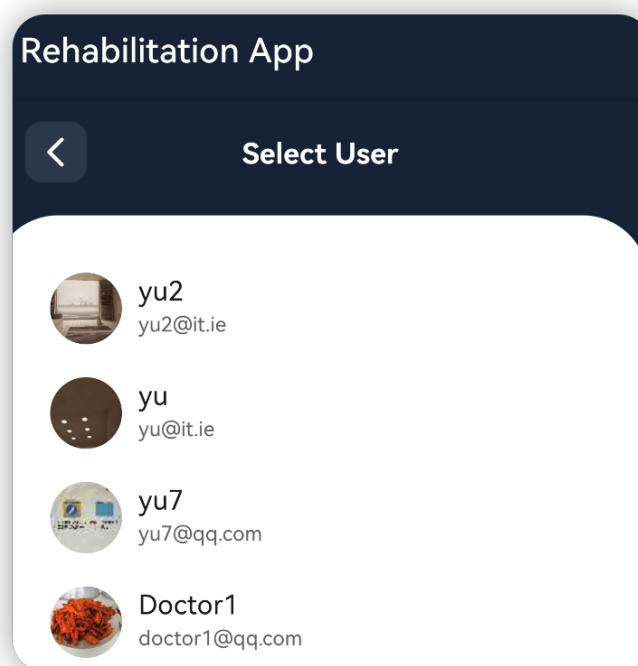


Figure 15. Remote diagnosis user selection screen.

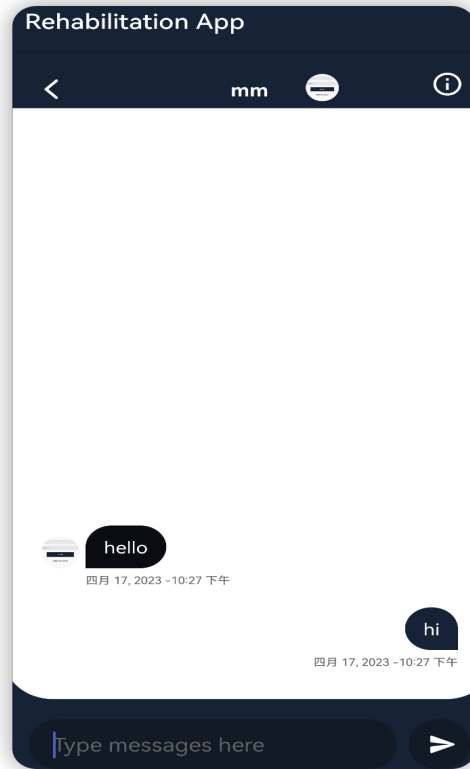


Figure 16. Remote diagnosis user message conversation screen.

The user can have a remote diagnosis service with medical care experts. the user's personal rehabilitation status and movement data analysis data can be shared with rehabilitation medical care experts for diagnosis usage.

4.7 User Personal Details



The screenshot shows a mobile application interface titled "Rehabilitation App". It contains a form with the following fields and values:

- Header: Rehabilitation App
- Instruction: Please enter below information
- Age: 26
- Gender: women
- Illness History: here is the user illnesses history
- Rehabilitation Plan: here is the rehabilitation plan details

Figure 17. User personal details screen.

The user details contain the Age, Gender, illness History and the Rehabilitation Plan.

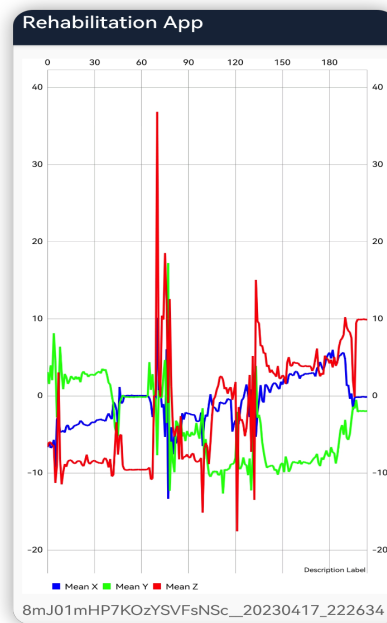


Figure 18. User personal rehabilitation summary line chart screen.

The user personal rehabilitation summary line chart screen, with the trend of mean value linear acceleration data analysis.

5. Sequence diagram

The following sequence diagrams illustrate the project application's different function's operation in detail.

5.1 Login sequence diagram

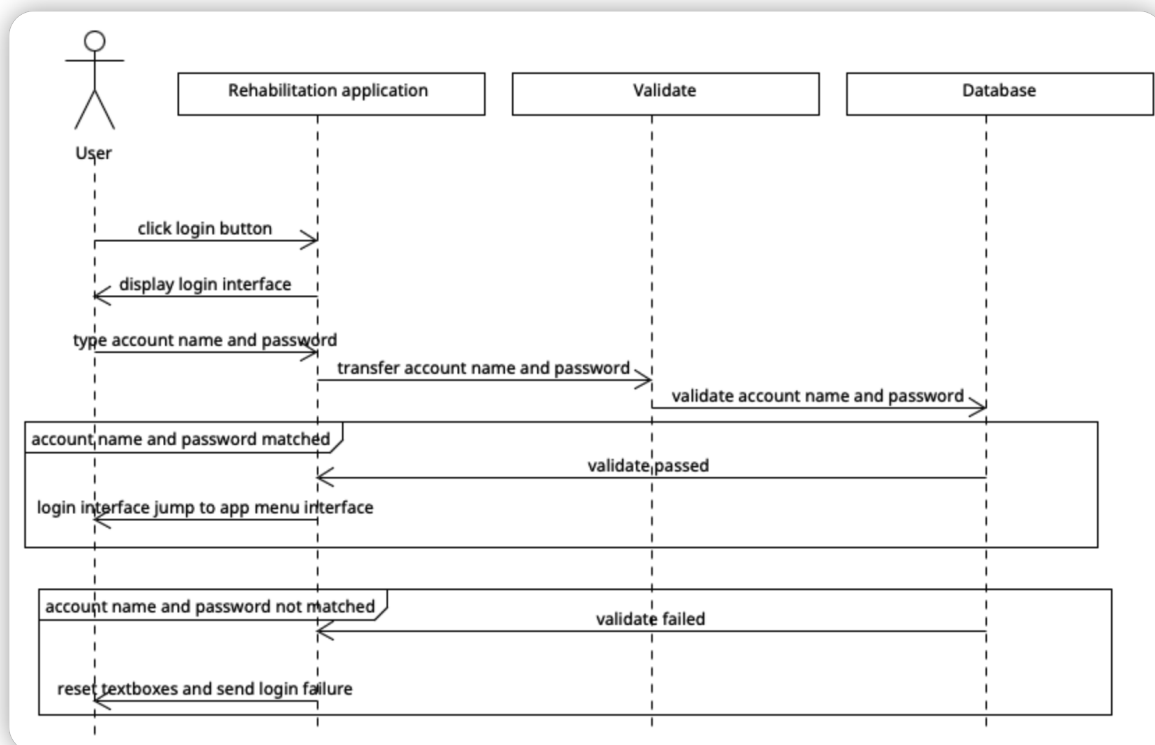


Figure 19. Login sequence diagram screenshot.

5.2 Manage Rehabilitation Site Selection

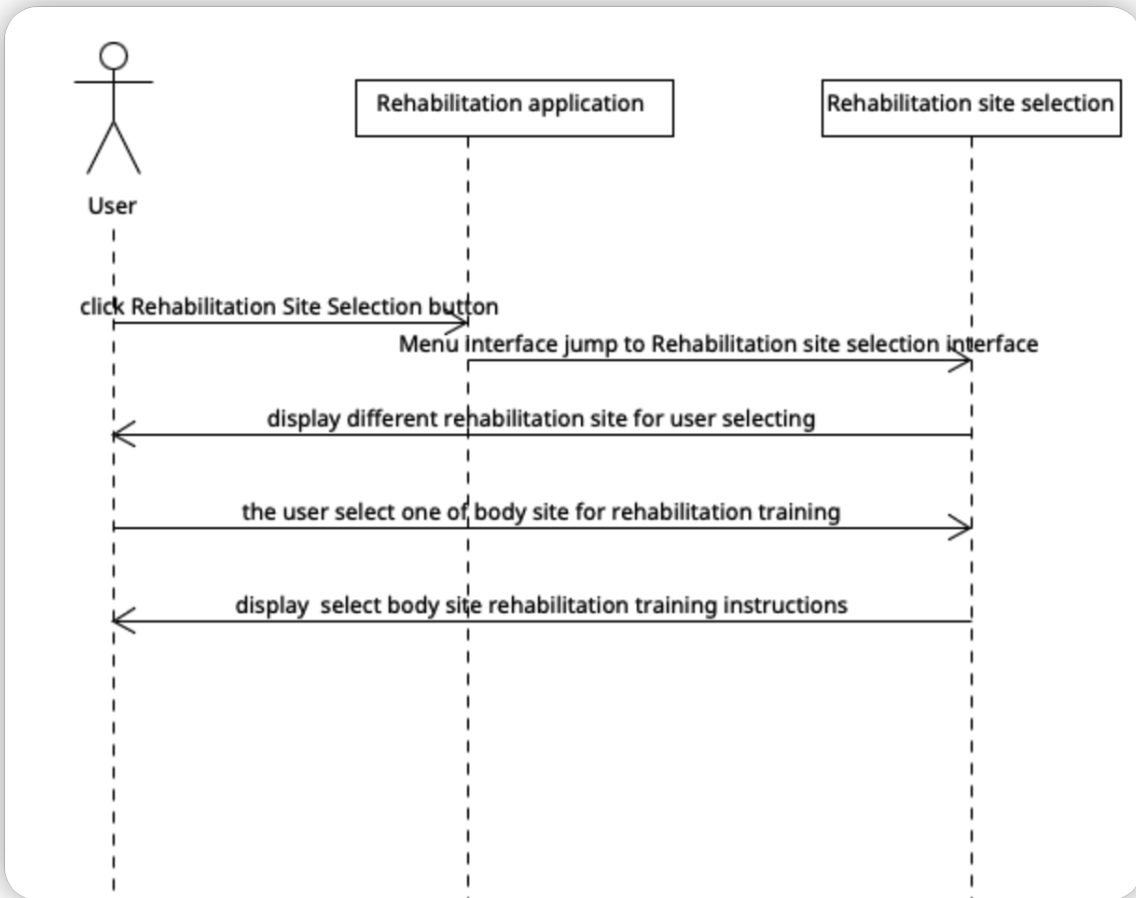


Figure 20. Rehabilitation Site Selection sequence diagram screen.

5.3 Remote diagnosis sequence diagram

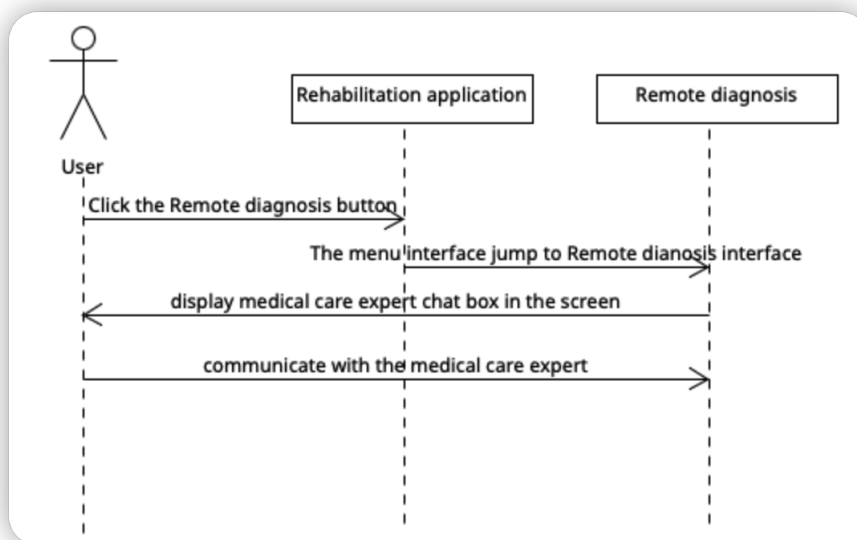


Figure 21. Remote diagnosis sequence diagram

5.4 Measurement data analysis

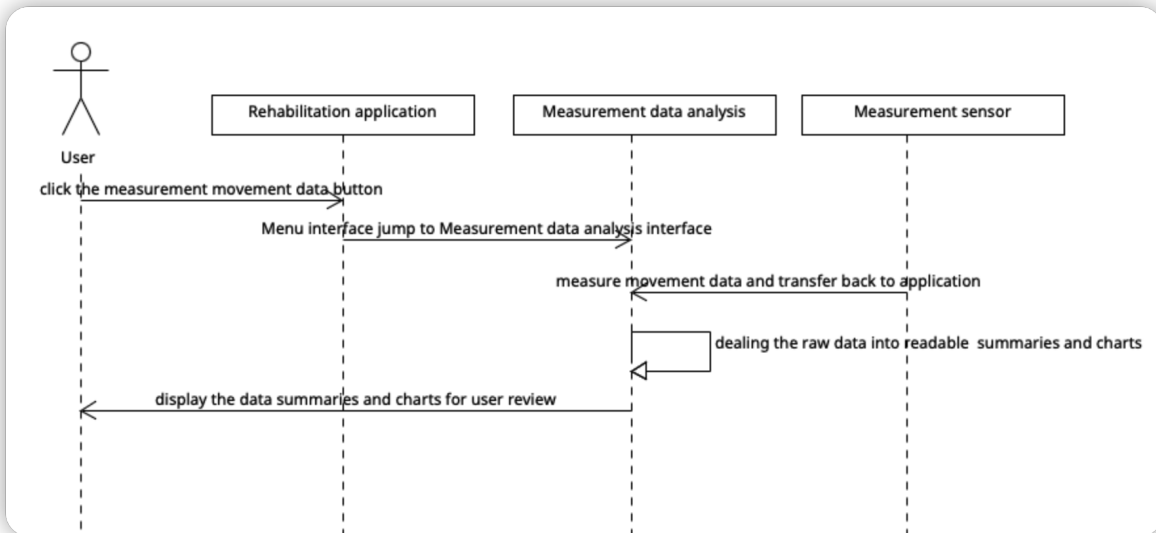


Figure 22. Measurement data analysis sequence diagram.

6. Movesense sensor raw data transfer screenshot

```
BluetoothGatt: onNotify() - Device=BC:8C:8C:8C:8C:8C handle=25
BleManager: Received data: WB_DATAMSG_CLIENT_ON_NOTIFY, type: 1, direct msg, msgLen: 41, reqId: 0, size: 47
MdsOperationHandler: SdsInternalCallback with call type: 2 header: {"Content-Type": "application/json", "Uri": "MDS/EventListener/34", "Content-Length": 180, "Reason": "LinearAccelerationTestActivity: onSuccess()"} {"Body": {"Timestamp": 5429430, "ArrayAcc": [{"x": 0.17946811020374298, "y": -0.074180148541927338, "z": -9.851602554321289}]}
BluetoothGatt: onNotify() - Device=BC:8C:8C:8C:8C:8C handle=25
BleManager: Received data: WB_DATAMSG_CLIENT_ON_NOTIFY, type: 1, direct msg, msgLen: 41, reqId: 0, size: 47
MdsOperationHandler: SdsInternalCallback with call type: 2 header: {"Content-Type": "application/json", "Uri": "MDS/EventListener/34", "Content-Length": 180, "Reason": "LinearAccelerationTestActivity: onSuccess()"} {"Body": {"Timestamp": 5429509, "ArrayAcc": [{"x": 0.17228938639163971, "y": -0.074180148541927338, "z": -9.765458106994628}]}
BluetoothGatt: onNotify() - Device=BC:8C:8C:8C:8C:8C handle=25
BleManager: Received data: WB_DATAMSG_CLIENT_ON_NOTIFY, type: 1, direct msg, msgLen: 41, reqId: 0, size: 47
MdsOperationHandler: SdsInternalCallback with call type: 2 header: {"Content-Type": "application/json", "Uri": "MDS/EventListener/34", "Content-Length": 179, "Reason": "LinearAccelerationTestActivity: onSuccess()"} {"Body": {"Timestamp": 5429588, "ArrayAcc": [{"x": 0.14357449114322662, "y": -0.11007377505302429, "z": -9.8492097854614258}]}
BluetoothGatt: onNotify() - Device=BC:8C:8C:8C:8C:8C handle=25
BleManager: Received data: WB_DATAMSG_CLIENT_ON_NOTIFY, type: 1, direct msg, msgLen: 41, reqId: 0, size: 47
```

Figure 23. Movesense sensor raw data transfer screenshot.

As above Movesense sensor raw data is transferred to the application screenshot, which shows the Movesense sensor movement track that is represented as X, Y, Z three dimensions (3D) direction data.

7. Firebase firestore of the application data structure

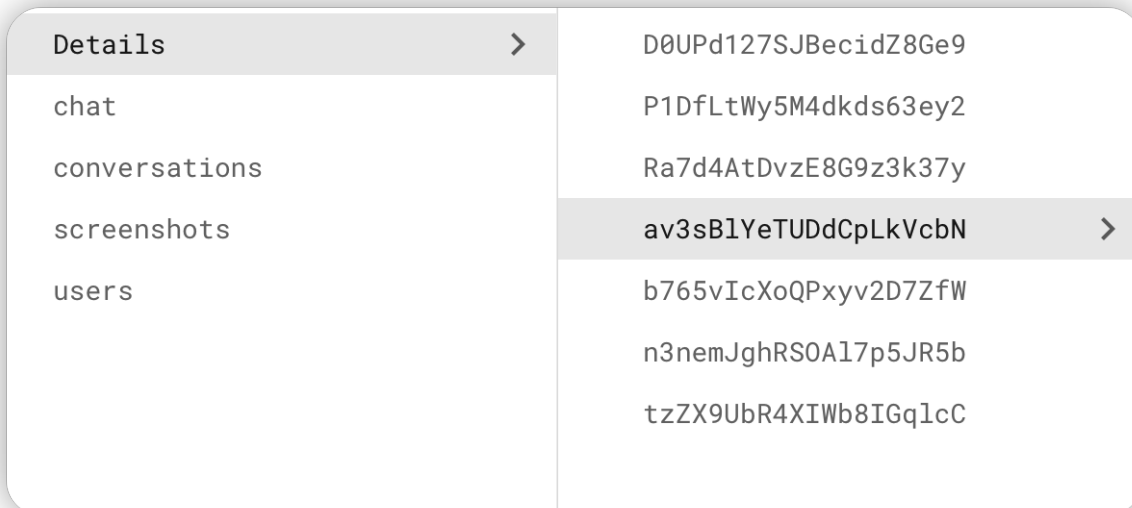


Figure 24. Firebase Firesotre application variable data collections screenshot.

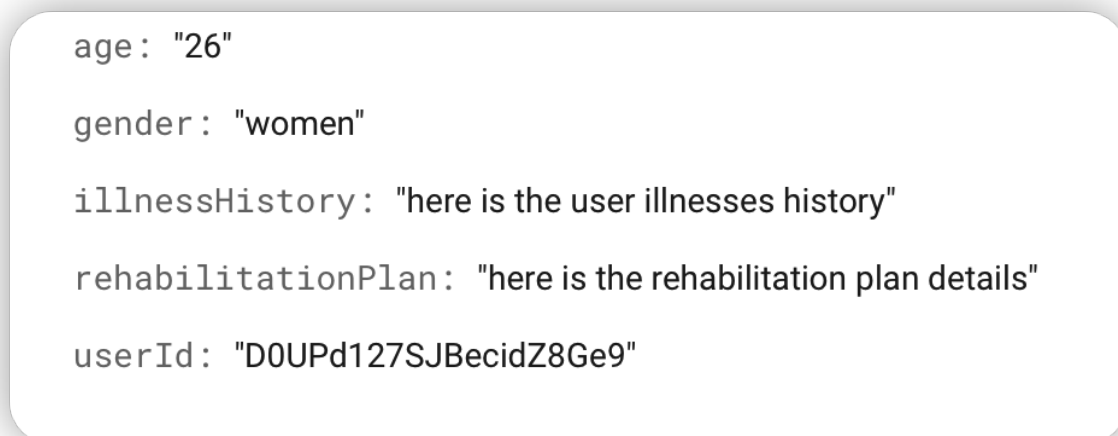


Figure 25. Firebase Firesotre application user details screenshot.

message: "uuuuuu"

receiverId: "D0UPd127SJBecidZ8Ge9"

senderId: "b765vIcXoQPxyv2D7ZfW"

Figure 26. Firebase Firesotre application user mesasge screenshot.

createDate: "20230417_222634"

description: "8mJ01mHP7KOzYSVFsNSc__20230417_222634"

fileName: "MeanValueSummary_20230417_222633.png"

image: "iVBORw0KGgoAAAANSUgAABUAAAAMSCAIAAACeMrI8AAAA/
MwuNgAAAIABJREFUeJzs3X9onPe96Pmniw6MIX+MwQsecKFTUs
YLk90CgpuEoLuXK6NFbzR2OnbCr3srmSA03UssmVAwIWoDIS4GSt
NFDDPBCB9o9RR2N5JDtOrZnPzOv119OZZ9wner7z4z3PM9/nG6ur
HgAAAIQ8BulK2my0uqN4I76GvX80hm9n4HSNM0TdPbfKwB0Ln
DWgXix8vTr0+Nf2P0+XFcpIkhfsK/f+h//Dg4cl9hVZvGn8BaZrOvj7+
gf/twOOPPv4sFm/6cMOjkyxdWTr565015aEnhm66E+396NI0nX53e
9w8cHOjd33vThxsAgUy/Mz1/YT5JkuEfDufvzt90/blzc9NvT8+9N7d.
1v9a4TuF4R8Pb3avAdCGvuorwPy5+el3pmf/39nyYjldSQv3FPoe6hs
PZkb/ziZTGb81Hirt46va+FF4r7Nq30Yz87Vq1Wt3i44dFhhp4aqu3B
//BTw8vLy1s83ACIYmFhof/R/vo00j9/fuv1L//x8sBjA5sOjB8PL//Jwl
DWz2cAOg3XzVAbD8xXL9w0CTV4Aj3hpoawJ+dfzUeP2JI81m+x/u
5u/O9z/cv+HoyvCR4c0ebnh0mLNzZ+vvshcXL269sr0f3dk/nN349H
ACF8/sXnl8+ObPgYvfXH94ufXGw8Opf/Vn7gsYH+h/ozmfVP5IOHBl
8eRmNgt4A6Ct3MYAqFar/Q+t136xWBx+anj4yHDjt70DBweqXzY/un

Figure 27. Firebase Firesotre application rehabilitation analysis summary screenshot.

8. Bibliography

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

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

“The Movesense-API.” *The Movesense-API*, available on <https://bitbucket.org/movesense/movesense-device-lib/src/master/MovesenseCoreLib/reources/movesense-api/>.

Java 2017, Java Wikipedia overview, available on [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

Mockflow 2023, official website, available on <https://support.mockflow.com/category/92-video-tutorials>

Official Website, Android Studio. “Download Android Studio & App Tools.” *Android Developers*, available on <https://developer.android.com/studio>.