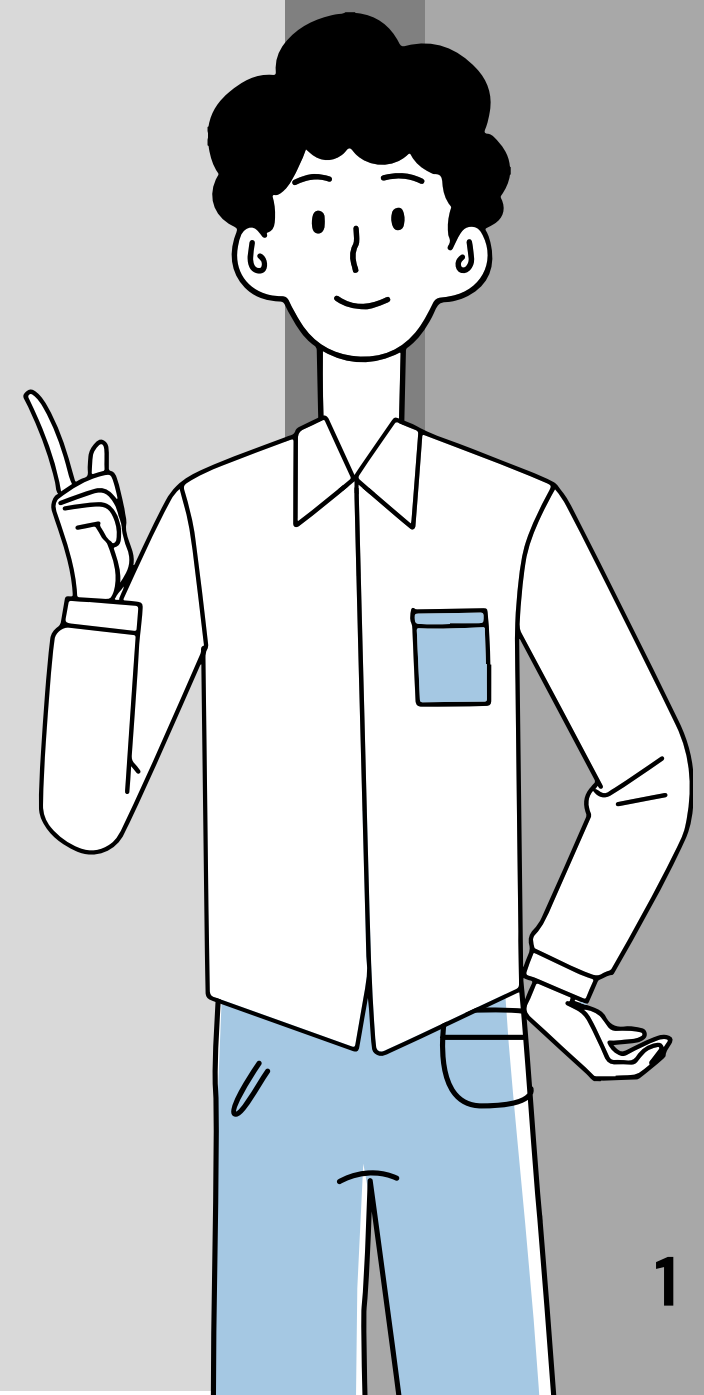


IOS APP wearable advices connection with heart rate

2023/07/21 (Sat)

第四組：林劭芸 廖苡恩 朱芷鈴 吳宜芊



Contents

- 1 UI development
- 2 ChatGPT connection
- 3 Graph insert
- 4 Flask connection
- 5 Database connection
- 6 Academic article discussion

UI development



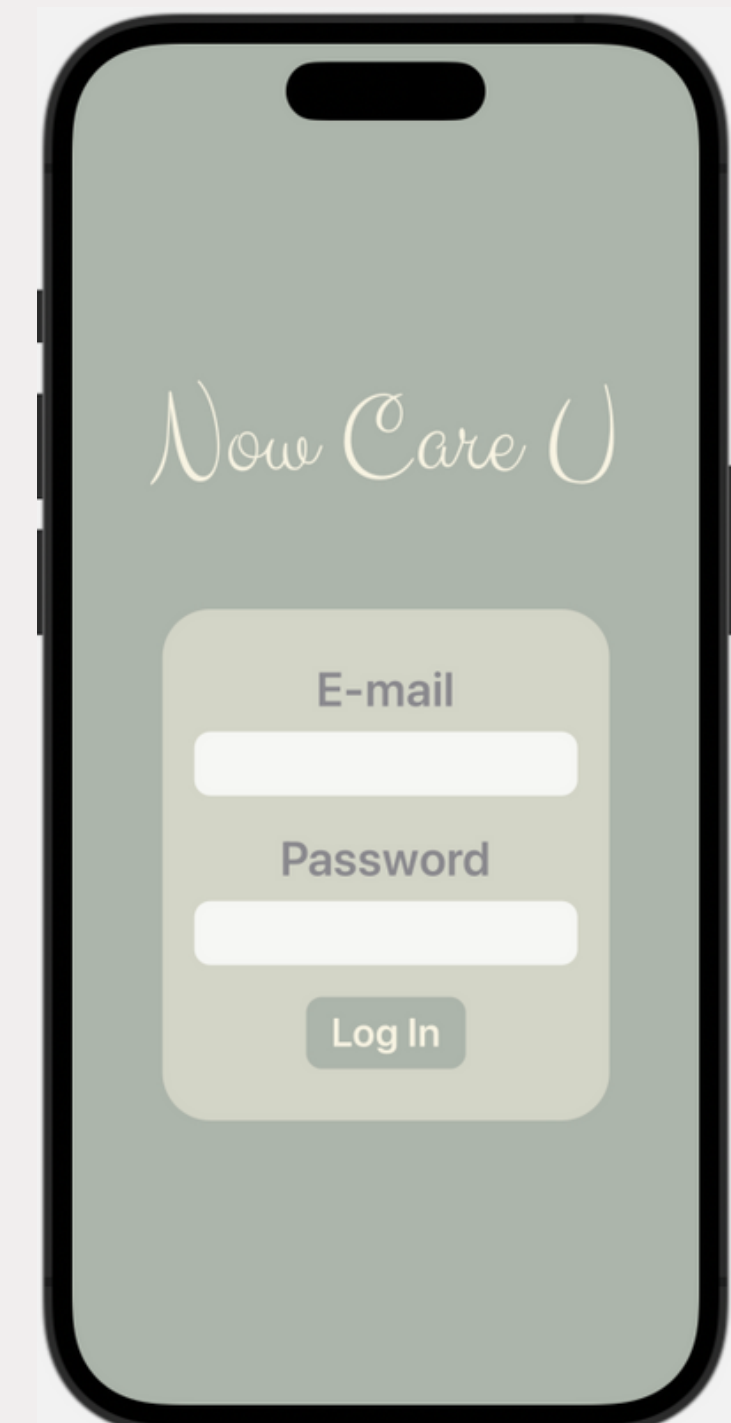
UI development 🔍

Log In Page

```
if email==" " || password==" "{
    isNavigation=false
}
else{
    sendData()
    if results == "email wrong"{
        email=""
        isNavigation=false
    }
    else if results == "password wrong"{
        password=""
        isNavigation=false
    }
    else if !(results==""){
        let r=results.split(separator: ",")
        isNavigation=true
    }
}
```

```
func sendData(){
    let jsonString = "{ \"email\" : \""+email+"\", \"password\" : \""+password+"\"}"
    guard let jsonData = jsonString.data(using: .utf8) else { return }
    // Send request
    guard let url = URL(string: "https://1eac-140-115-227-57.ngrok-free.app/login") else
    {return}

    var request = URLRequest(url: url)
    request.httpMethod = "POST"
    request.httpBody = jsonData
    request.setValue("application/json", forHTTPHeaderField: "Content-Type")
    let task = URLSession.shared.dataTask(with: request)
    { (data, response, error) in
        if let error = error {
            print(error)
            // Handle HTTP request error
        } else if let data = data {
            results=String(decoding: data, as: UTF8.self)
        } else {
        }
    }
    task.resume()
}
```



UI development 🔍

Home Page

```
init() {
    UIPageControl.appearance().currentPageIndicatorTintColor =
        UIColor(red:248/255,green:244/255,blue:227/255, alpha: 1)
    UIPageControl.appearance().pageIndicatorTintColor =
        UIColor(red:154/255,green:153/255,blue:140/255, alpha: 1)
    UIPageControl.appearance().backgroundStyle = .minimal
}
```

```
TabView{
    ForEach(0..<2){index in
        ZStack{
            Rectangle()
                .fill(Color
                    (red:173/255,green:181/255,blue:171/255))
                .frame(width: 400,height: 480)
            Text("心律圖")
                .font(.system(size: 20))
                .offset(y:-190)
                .bold()
            Rectangle()
                .fill(Color
                    (red:154/255,green:153/255,blue:140/255))
                .frame(width: 160,height: 25)
                .cornerRadius(15).offset(y:-150)
```



UI development 🔍

```
Button(action:{
  if(bottleIsPressed[index]){bottleIsPressed[index] = false}
  else{bottleIsPressed[index] = true}
})){
  ZStack{
    if(bottleIsPressed[index]){
      Image("waterBottlePressed")
        .offset(x:CGFloat(bottleCount*5)-15,y:-50)
      Image(systemName:
        "checkmark")
        .offset(x:CGFloat(bottleCount*5)-15,y:-50)
        .font(.system(size:20))
      .foregroundColor(Color
        (red:159/255,green:113/255,blue:105/255))
    }
    else{
      Image("waterBottle")
        .offset(x:CGFloat(bottleCount*5)-15,y:-50)
    }
  }
}
```



UI development 🔍

```
@State private var sportList = [["figure.run", "", ""]]
@State private var sportIcons =
    ["figure.run", "figure.open.water.swim", "figure.outdoor.cycle", "figure
    .strengthtraining.traditional", "basketball.fill", "volleyball.fill", "baseball
    .fill", "tennis.racket"]
```

```
Button(action:{
    let i:Int? = sportIcons.firstIndex(of: sportList[index][0])
    if i!+1==sportIcons.count{
        sportList[index][0]=sportIcons[0]
    }else{
        sportList[index][0]=sportIcons[i!+1]
    }
}){
    Image(systemName:sportList[index]
        [0])
        .foregroundColor(Color
            (red:159/255,green:113/255,blue:105/255))
        .font(.system(size: 30))
}.offset(x:-130)
```

今日運動量



跑步

30 分鐘



游泳

? 分鐘



```
HStack{
    Button(action:{
        if !(sportList[sportList.count-1][1]=="") &&
            !(sportList[sportList.count-1][1]==""){
            sportList.append(["figure.run", "", ""])
        }
    }){
        Image(systemName:"plus.circle.fill")
            .foregroundColor(Color
                (red:159/255,green:113/255,blue:105/255))
            .font(.system(size: 20))
    }
    Button(action:{
        sportList.removeLast()
        print(sportList)
    }){
        Image(systemName:"minus.circle.fill")
            .foregroundColor(Color
                (red:159/255,green:113/255,blue:105/255))
            .font(.system(size: 20))
    }
}.offset(x:120,y:CGFloat(sportList.count)*10)
```

UI development 🔍

Settings -
User's information
Settings
Application Introduction
Privacy



UI development

Editing self information



Now Care U
編輯個人資料

名稱：
輸入名稱

生日：
輸入生日

性別：
輸入性別

Email：
輸入電子郵件

身高：
輸入身高



輸入名稱

生日：
輸入生日

性別：
輸入性別

Email：
輸入電子郵件

身高：
輸入身高

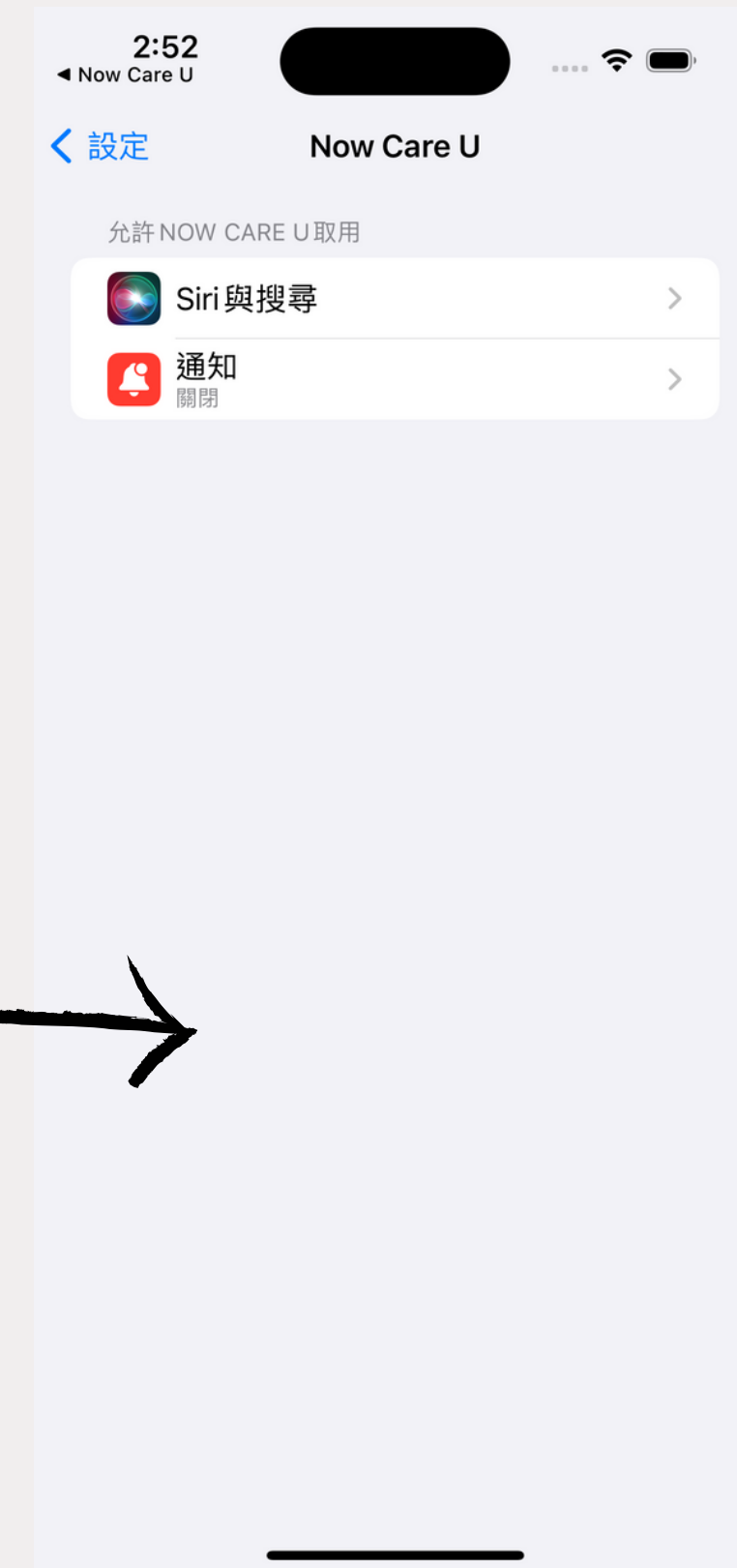
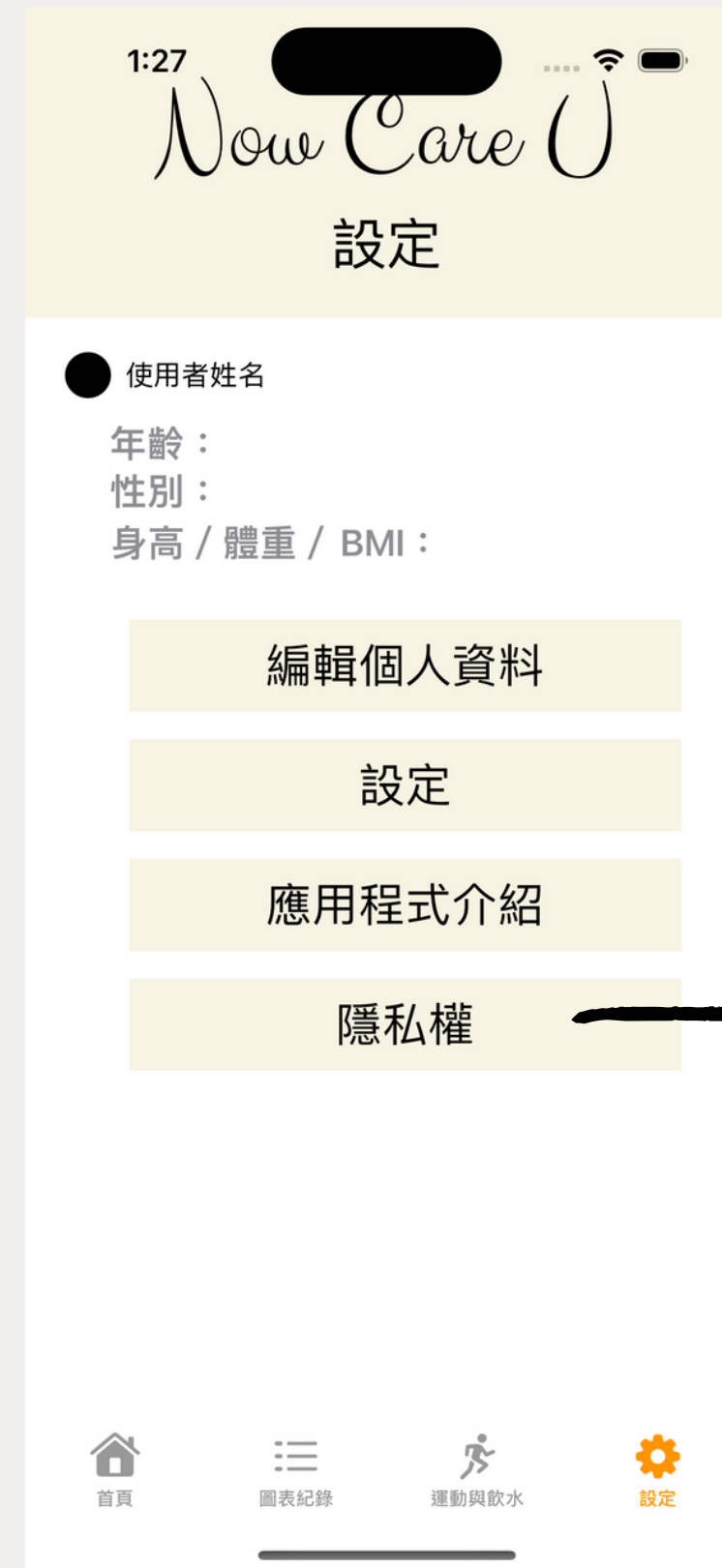
體重：
輸入體重

藥物使用：
簡述藥物使用

日常運動量：
簡述日常運動量

UI development

User's privacy



UI development 🔍

Upload photo to APP



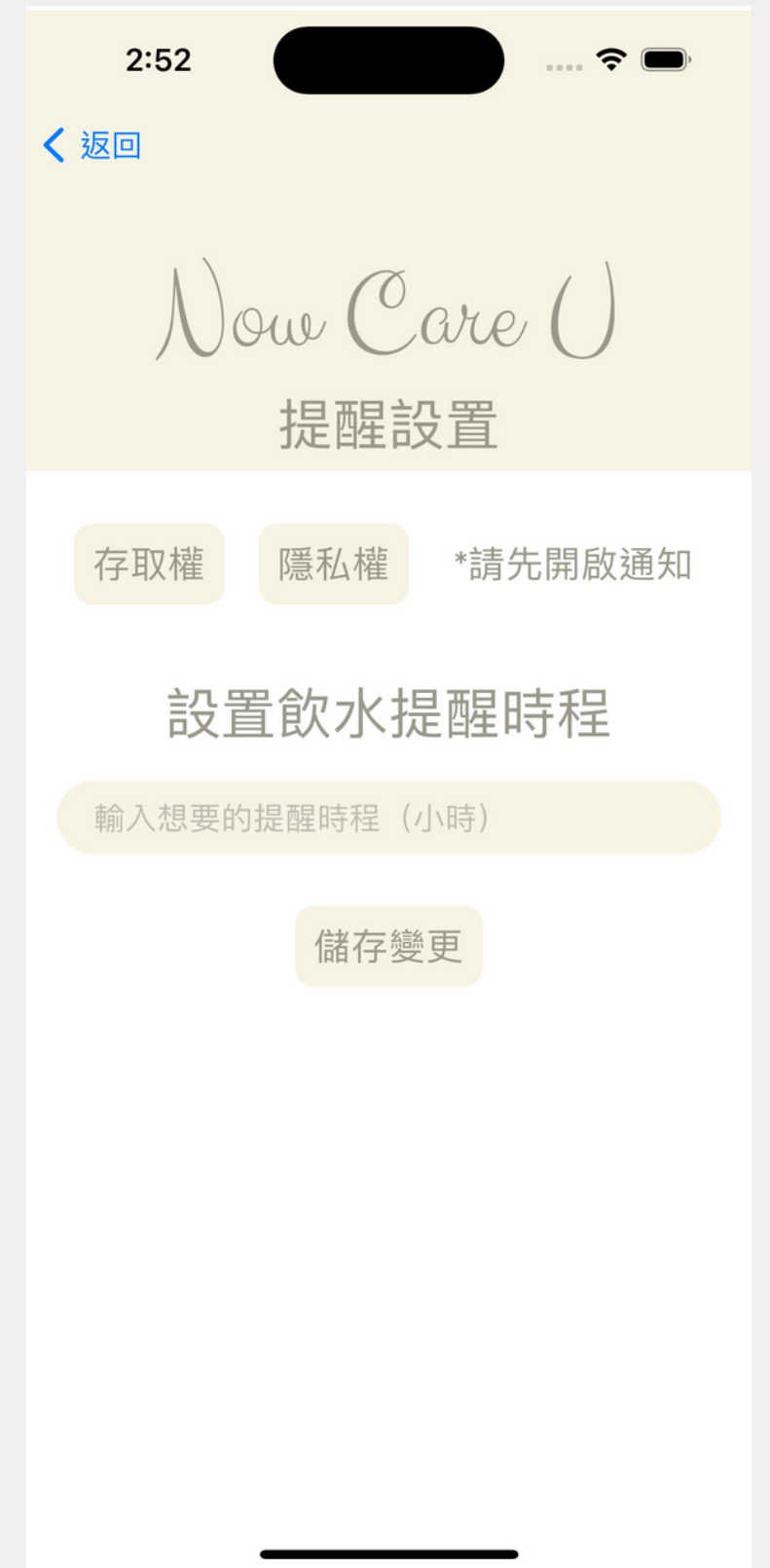
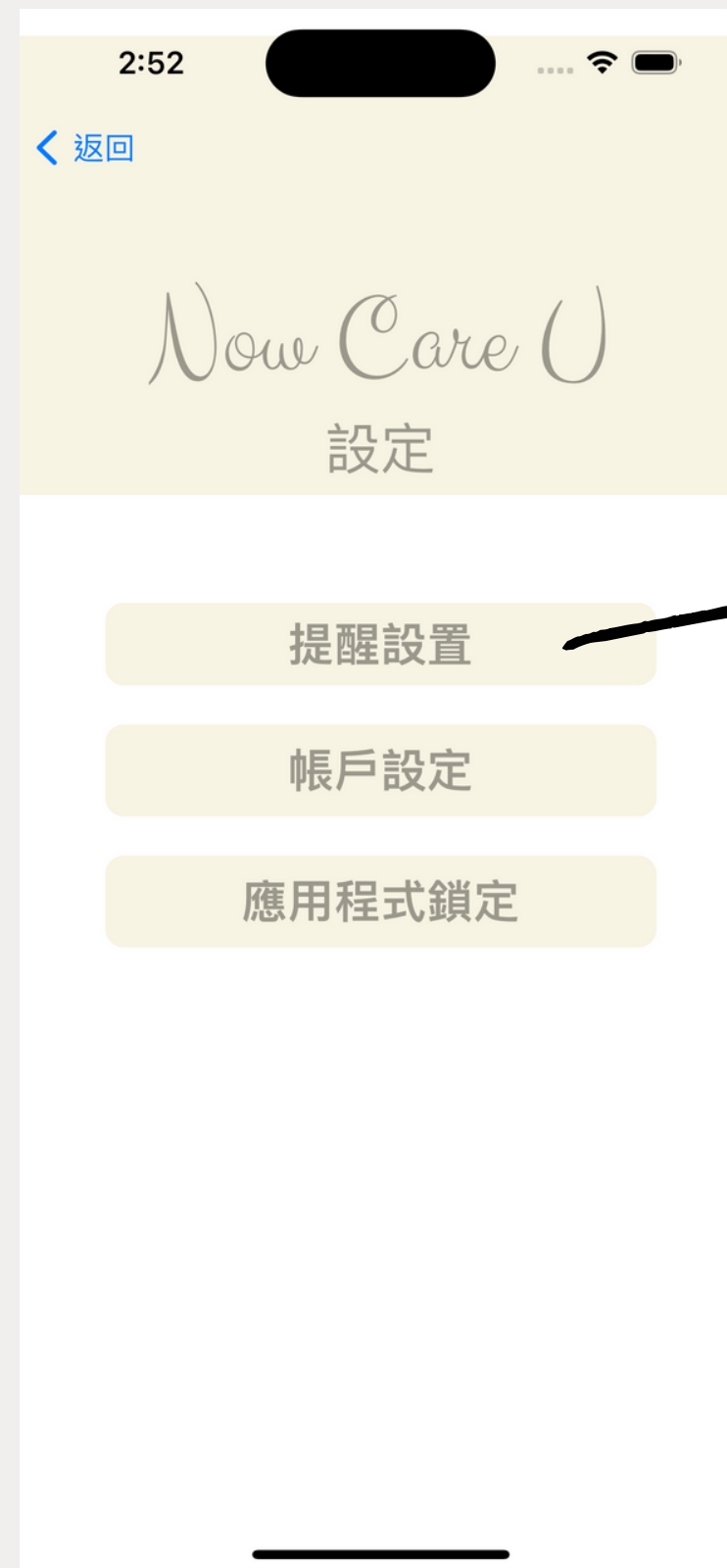
UI development 🔍

Settings - Notifications, Password and Account



UI development

Drinking water notifications

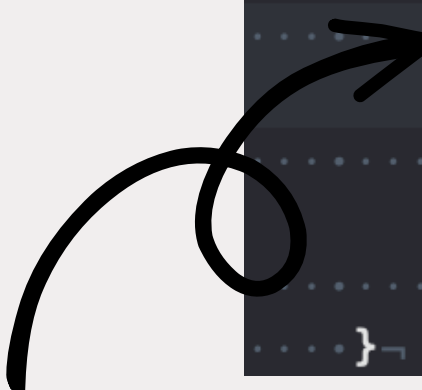


UI development 🔍

UNUserNotificationCenter requestAuthorization

```
.....Button("存取權"){  
.....UNUserNotificationCenter.current().requestAuthorization(options: [.alert,  
.....    .badge, .sound]){ success, error in  
.....    if success{  
.....        print("Notification Permission Set")  
.....    } else if let error = error{  
.....        print(error.localizedDescription)  
.....    }  
.....}
```

```
.....Button("儲存變更"){  
.....    let content = UNMutableNotificationContent()  
.....    content.title = "Drink Time!"  
.....    content.subtitle = "該攝取些水分囉!"  
.....    content.sound = UNNotificationSound.default  
.....  
.....    let trigger = UNTimeIntervalNotificationTrigger(timeInterval: 60,  
.....        repeats: true)  
.....    let request = UNNotificationRequest(identifier: UUID().uuidString,  
.....        content: content, trigger: trigger)  
.....    UNUserNotificationCenter.current().add(request)  
.....}
```



UNTimeIntervalNotificationTrigger

60 seconds cycle (repeat)

UI development

setting notifications



UI development 🔍

Graph Record

- SwiftUI Charts

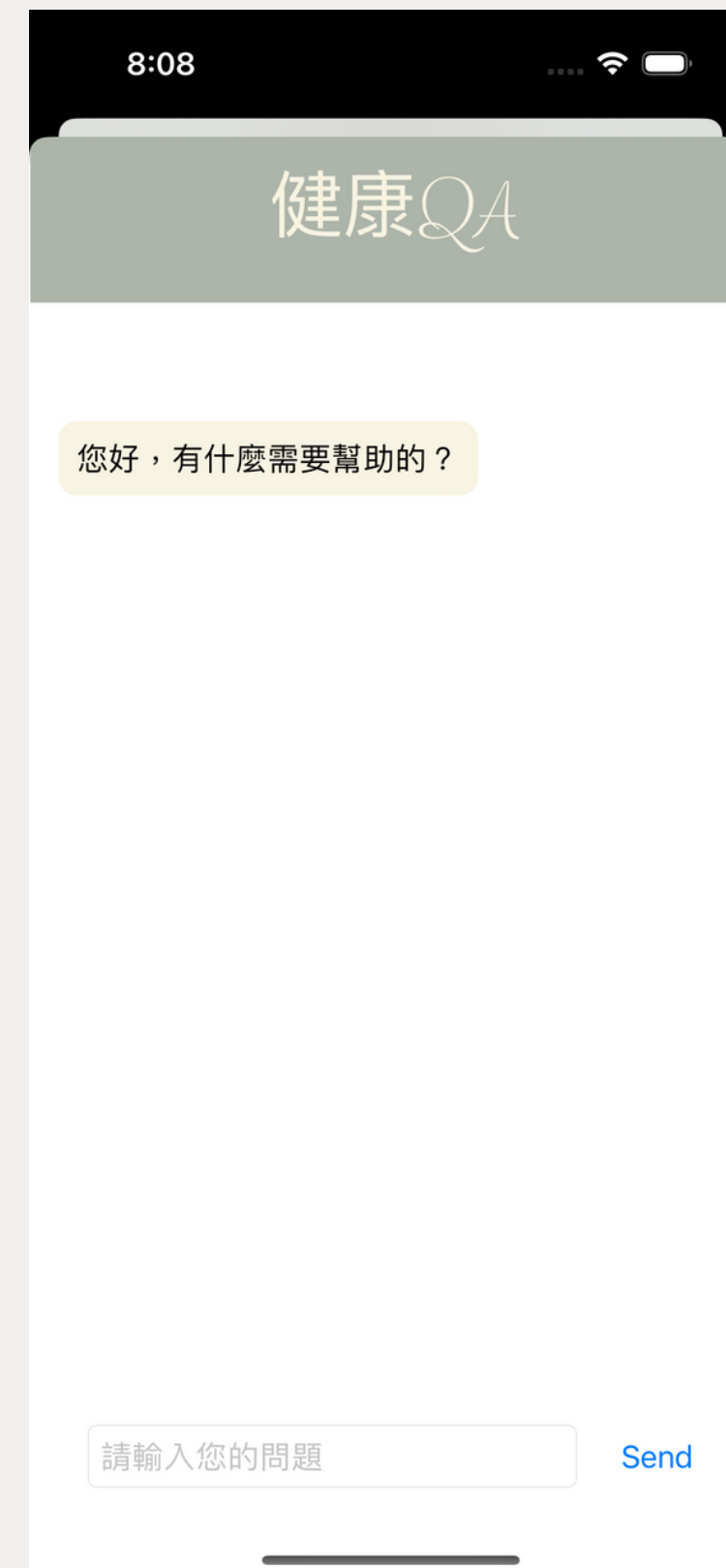
```
.....Chart {  
.....ForEach(0..  
.....BarMark(  
.....x: .value("Day", "\n(formattedDate(dates[index]))"),  
.....y: .value("Steps", stepData[index])  
.....)  
.....}  
.....|  
.....}  
.....Chart {  
.....ForEach(0..  
.....BarMark(  
.....x: .value("Day", "\n(formattedDate(dates[index]))"),  
.....y: .value("Heart Rate", heartRateData[index])  
.....)  
.....}  
.....}
```



UI development 🔍

ChatGPT Health QA

```
10 struct ChatRoomView: View {
11     @State private var inputMessage = ""
12     @State private var chatHistory: [ChatMessage] = [
13         ChatMessage(content: "您好，有什麼需要幫助的？", isUser: false),
14     ]
15     @State private var showTypingAnimation = false // 控制动画的状态
16
17     var body: some View {
18         ZStack(alignment: .top) {
19             VStack {
20                 ZStack(alignment: .leading) {
21                     Rectangle()
22                     .fill(Color(red: 173/255, green: 181/255, blue: 171/255))
23                     .frame(width: 391.5, height: 120)
24                     .offset(x: 0, y: -30)
25                     Text("健康QA")
26                     .foregroundColor(Color(red: 248/255, green: 244/255, blue: 227/255))
27                     .font(.custom("ClickerScript-Regular", size: 40))
28                     .offset(x: 130, y: -20)
29                 }
30                 ScrollView {
31                     LazyVStack {
32                         ForEach(chatHistory) { message in
33                             ChatBubble(content: message.content, isUser: message.isUser)
34                         }
35                         if showTypingAnimation {
36                             ChatBubble(content: "回答中...請稍候", isUser: false) // 显示动画效果
37                         }
38                     }
39                 }
40             }
41         }
42     }
43 }
```



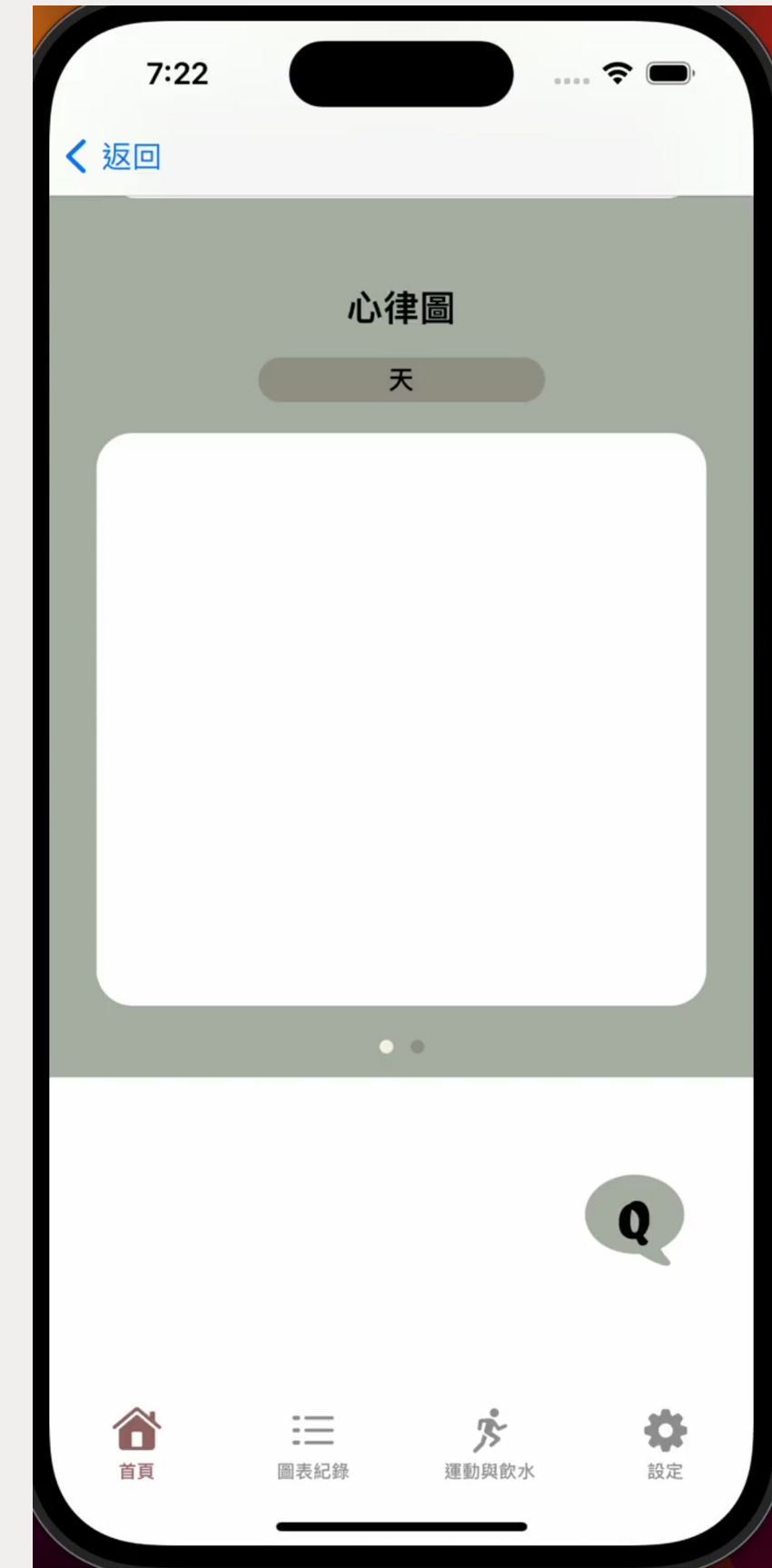
C h a t G P T
c o n n e c t i o n



ChatGPT connection

ChatGPT connection result

- Complete chatroom interface
- Answer health related questions only
- "answering the question, please hold on." message in the chat box.



ChatGPT connection

chatroom interface

- isUser -> distinguish question or answer
- showTypingAnimation

```
struct ChatRoomView: View {  
    @State private var inputMessage = ""  
    @State private var chatHistory: [ChatMessage] = [  
        ChatMessage(content: "您好，有什麼需要幫助的？", isUser: false),  
    ]  
    @State private var showTypingAnimation = false // 控制動畫的狀態
```

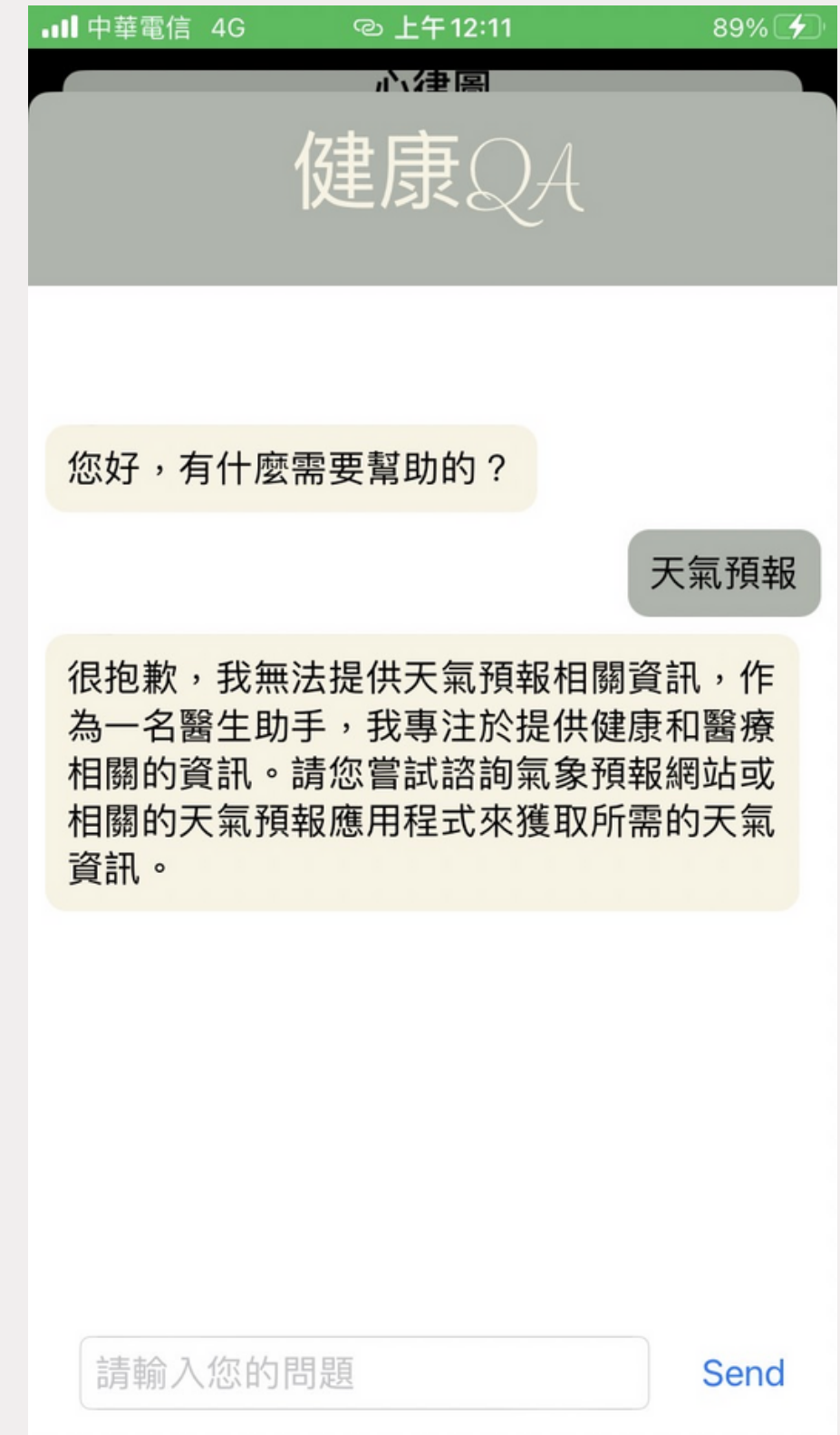


ChatGPT connection

chat function

- ChatGPTKit -> read API Key
- doctorMessage -> set the character for GPT

```
func chat() {  
    guard !inputMessage.isEmpty else {  
        return  
    }  
  
    chatHistory.append(ChatMessage(content: inputMessage, isUser: true))  
    showTypingAnimation = true // 显示动画效果  
  
    Task.detached {  
        let chattyGPT = ChatGPTKit(apiKey: "sk-U2YX0yzF1ZtN7KOGv0QwT3B1bkFJ5s5e40BbZQJmMHzwu4Y7")  
        var history = [Message]()  
        let doctorMessage = Message(role: .assistant, content:  
            "你現在是一名醫生，只能回覆健康相關資訊，如果無關健康醫療相關問題，一律無法回答")  
        history.append(doctorMessage)
```



ChatGPT connection

chat function

- do-catch
- chatHistory
- UIPasteboard

```
do {  
    let result = try await chattyGPT.performCompletions(messages: history)  
    switch result {  
    case .success(let response):  
        if let firstResponse = response.choices?.first {  
            chatHistory.append(ChatMessage(content: firstResponse.message.content, isUser: false))  
            UIPasteboard.general.string = firstResponse.message.content  
            print(firstResponse.message.content)  
        }  
    case .failure(let error):  
        print(error)  
    }  
} catch {  
    print("Error: \(error)")  
}  
  
showTypingAnimation = false  
}  
inputMessage = ""  
}
```

G r a p h i n s e r t



Graph insert



Graph insert result

- request authorization success
- fetch step count data success
- fetch heart rate data fail

Graph insert



Request authorization

- HealthKit manager

```
// 健康數據管理器
class HealthKitManager: ObservableObject {
    let healthStore = HKHealthStore()

    func requestAuthorization() {
        let typesToRead: Set<HKSampleType> = [
            HKObjectType.quantityType(forIdentifier: .stepCount)!,
            HKObjectType.quantityType(forIdentifier: .heartRate)!
        ]

        healthStore.requestAuthorization(toShare: nil, read: typesToRead) { (success,
            error) in
            if success {
                // 授權成功
                print("授權成功")
            } else {
                // 授權失敗或被用戶拒絕
                if let error = error {
                    print("授權失敗，錯誤訊息：\(error.localizedDescription)")
                } else {
                    print("用戶拒絕了授權請求")
                }
            }
        }
    }
}
```

Graph insert



fetch step count data

- fetchStepCount

```
func fetchStepCount(forLastWeek: Bool, completion: @escaping ([Double], [Double], [Date]) -> Void) {  
    let stepCountType = HKObjectType.quantityType(forIdentifier: .stepCount)!  
    let heartRateType = HKObjectType.quantityType(forIdentifier: .heartRate)!  
    let calendar = Calendar.current  
  
    var startDate: Date  
    var endDate: Date  
  
    if forLastWeek {  
        let today = calendar.startOfDay(for: Date())  
        let dayOfWeek = calendar.component(.weekday, from: today)  
        let daysToAdd = (dayOfWeek - 1) * -1  
  
        startDate = calendar.date(byAdding: .day, value: daysToAdd - 7, to: today)!  
        endDate = calendar.date(byAdding: .day, value: daysToAdd, to: today)!  
    } else {  
        // 設定你想要的日期範圍，例如：  
        let dateFormatter = DateFormatter()  
        dateFormatter.dateFormat = "yyyy/MM/dd"  
        startDate = dateFormatter.date(from: "2023/7/19")!  
        endDate = Date()  
    }  
  
    let predicate = HKQuery.predicateForSamples(withStart: startDate, end: endDate, options: .strictStartDate)  
  
    let query = HKSampleQuery(sampleType: stepCountType, predicate: predicate, limit: HKObjectQueryNoLimit, sortDescriptors: nil) { (query, samples, error) in  
        guard let samples = samples as? [HKQuantitySample] else {  
            completion([], [], [])  
            return  
        }  
  
        var stepCounts: [Double] = []  
        var stepDates: [Date] = []  
        // var heartRates: [Double] = []  
    }
```

Graph insert



fetch step count data

- date formatter

```
.....// 格式化日期顯示
.....private func formattedDate(_ date: Date) -> String {
.....    let formatter = DateFormatter()
.....    formatter.dateFormat = "MM/dd"
.....    return formatter.string(from: date)
.....}
}
```

Graph insert

fetch step count data

- date range

7月						
1 十四						
2 十五	3 十六	4 十七	5 十八	6 十九	7 二十	8 廿一
9 廿二 ●	10 廿三 ●	11 廿四	12 廿五	13 廿六	14 廿七	15 廿八
16 廿九	17 三十	18 六月	19 初二	20 初三	21 初四	22 初五
23 初六	24 初七	25 初八	26 初九	27 初十	28 十一	29 十二
30 十三	31 十四					

```
.....if forLastWeek {  
.....    let today = calendar.startOfDay(for: Date())  
.....    let dayOfWeek = calendar.component(.weekday, from: today)  
.....    let daysToAdd = (dayOfWeek - 1) * -1  
.....  
.....    startDate = calendar.date(byAdding: .day, value: daysToAdd - 7, to: today)!  
.....    endDate = calendar.date(byAdding: .day, value: daysToAdd, to: today)!  
.....}
```

Graph insert



fetch step count data

- HKSampleQuery

```
.....let query = HKSampleQuery(sampleType: stepCountType, predicate: predicate, limit:
    HKObjectQueryNoLimit, sortDescriptors: nil) { (query, samples, error) in
.....guard let samples = samples as? [HKQuantitySample] else {
.....completion([], [], [])
.....return
.....}
.....
.....var stepCounts: [Double] = []
.....var stepDates: [Date] = []
```

Graph insert



fetch heart rate data

Thread 1: Fatal error: Index out of range

- insufficient data?

F l a s k c o n n e c t i o n



Flask connection 🔍

```
app = Flask(__name__)

@app.route('/login', methods=['POST'])
def test_send_page():
    if request.method=='POST':
        data=request.get_json()
        email=data.get("email")
        password=data.get("password")
        sql = "select * from ios_database.ios_data where `email`='"+email+"';"
        res=json.dumps(mysql_get_one(sql),ensure_ascii=False)
        if res=="null":
            return 'email wrong'
        else:
            if json.loads(res)[3]==password:
                return json.loads(res)
            else:
                return 'password wrong'

if __name__=="__main__":
    app.run()
```

**D a t a b a s e
c o n n e c t i o n**



Database connection



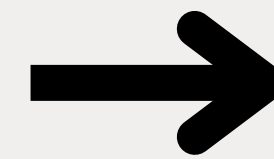
DB connection options

- Xcode directly connect to mysql
- use flask relative packages
- use python relative packages

Database connection



1 . Xcode directly connect to mysql



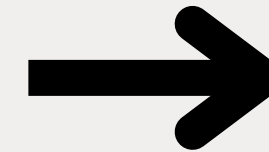
Procedure:

1. Through the terminal, create a new folder in the Xcode project, download and copy the MySQL connection file into this newly created folder.
2. In Xcode, go to the "Frameworks, Libraries, and Embedded Content" section, click the "+" button, and select "Add Files" to add the copied MySQL connection file to the project.

Database connection



1 . Xcode directly connect to mysql



Disadvantages:

- 1.Security Risks
- 2.Platform Limitations
- 3.Increased Workload
- 4.Code Maintenance

Database connection



2. Use flask relative packages

```
shaoyunlin@shaoyundeMac-mini local % export PATH=$PATH:/usr/local/mysql/bin
shaoyunlin@shaoyundeMac-mini local % pip3 install flask-mysqldb
Collecting flask-mysqldb
  Using cached Flask-MySQLdb-1.0.1.tar.gz (4.3 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: Flask>=0.12.4 in /opt/homebrew/lib/python3.11/site-packages
Collecting mysqlclient>=1.3.7 (from flask-mysqldb)
  Using cached mysqlclient-2.2.0.tar.gz (89 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... error
error: subprocess-exited-with-error

× Getting requirements to build wheel did not run successfully.
  exit code: 1
  [27 lines of output]
```

- `import Flask_mysqldb`

Database connection



2. Use flask relative packages

```
1  # main.py
2
3  from flask_sqlalchemy import SQLAlchemy
4  from flask import Flask
5
6  db = SQLAlchemy()
7
8  app = Flask(__name__)
9
10 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
11 app.config['SQLALCHEMY_DATABASE_URI'] = "mysql+pymysql://myname:root1234@IP:3306/testdb"
12
13 db.init_app(app)
14
15 @app.route('/')
16 def index():
17
18     sql_cmd = """
19         select *
20         from product
21         """
22
23     query_data = db.engine.execute(sql_cmd)
24     print(query_data)
25     return 'ok'
26
27
28 if __name__ == "__main__":
29     app.run()
```

- `import Flask_alchemy`

Database connection



3. Use python relative packages

```
from flask import Flask, request, g
import pymysql, json

def mysql_put(sql):
    connect_db = pymysql.connect(host='localhost', port=3306, user='root', passwd='root1234', charset='utf8', db='ios_database')
    with connect_db.cursor() as cursor:
        cursor.execute(sql)
        connect_db.commit()
    connect_db.close()

def mysql_get_all(sql):
    connect_db = pymysql.connect(host='localhost', port=3306, user='root', passwd='root1234', charset='utf8', db='ios_database')
    with connect_db.cursor() as cursor:
        cursor.execute(sql)
        data = cursor.fetchall()
    connect_db.close()
    return data

def mysql_get_one(sql):
    connect_db = pymysql.connect(host='localhost', port=3306, user='root', passwd='root1234', charset='utf8', db='ios_database')
    with connect_db.cursor() as cursor:
        cursor.execute(sql)
        data = cursor.fetchone()
    connect_db.close()
    return data
```

• `import pymysql`

Database connection



3. Use python relative packages

```
[mysql> insert into ios_database.ios_data(`name`,`id`,`email`,`password`) values ('user',109303000,'email','pwd');
Query OK, 1 row affected (0.01 sec)

[mysql> select * from ios_data;
+-----+-----+-----+-----+
| name      | id        | email                      | password          |
+-----+-----+-----+-----+
| 林劭芸    | 109303501 | 109303501@cc.ncu.edu.tw   | 109303501ncuedu  |
| user      | 109303000 | email                      | pwd               |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

["user", 109303000, "email", "pwd"]

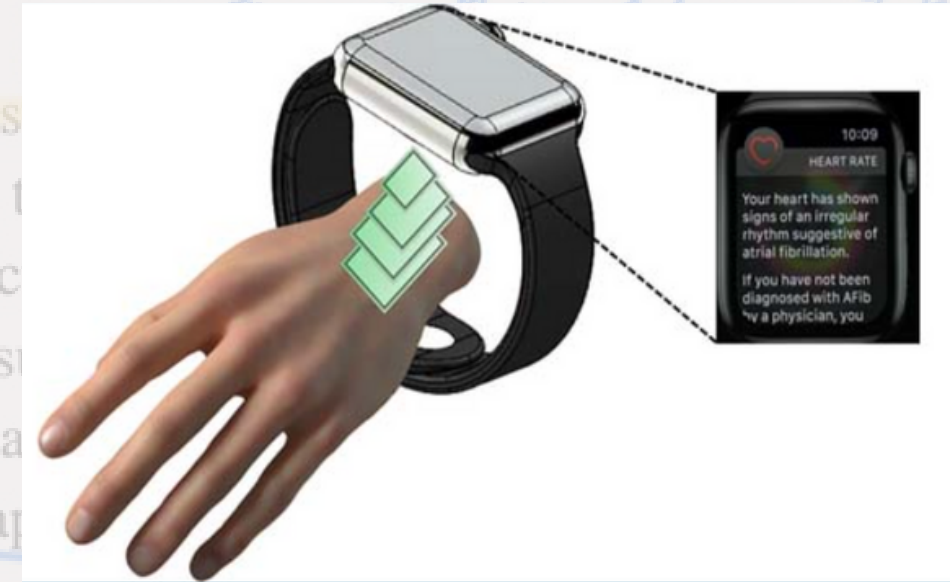
Academic article discussion



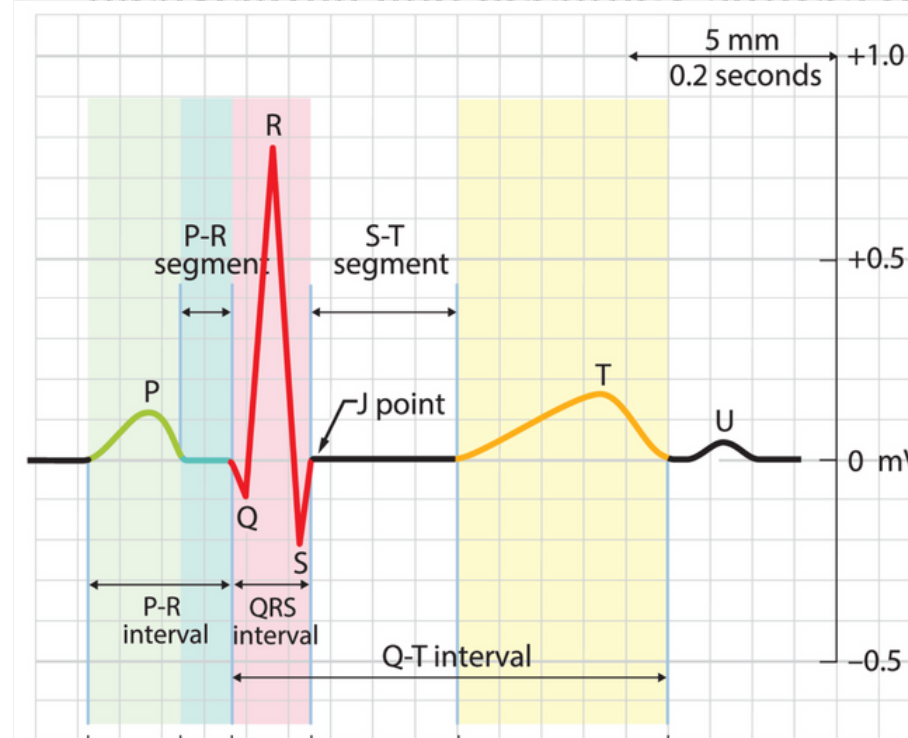
1. Apple Watch, Wearables, and Heart Rhythm: where do we stand?



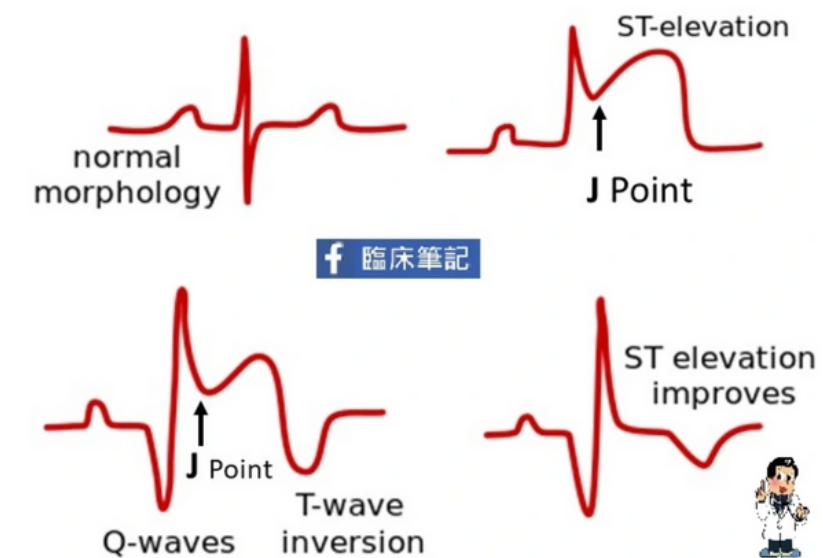
1. What is PPG?
2. What are the applications about PPG?
3. What is the relationship between PPG and Apple Watch?



• PPG



急性心肌缺氧的ECG變化



• Healthy ECG

• AF ECG

2 . The Clinical Value of Heart Rate Monitoring : Using an Apple Watch



• Fitbit charge

1. Summary
2. Apple watch & Fitbit charge
3. The clinical experience



• Apple watch

**What we learn during
our discovery?**

What we learn during our discovery?

廖苡恩

- Swift language
- Figma design
- app UI development
- How to fetch HealthKit data and present as a chart
- How to connect ChatGPT into iOS app

What we learn during our discovery?

吳宜芊

- How to use Figma?
- Swift language
- How to build environment for Mysql database?
- How to insert and read data in Mysql?
- How to summary the Academic articles?
- How to think from different views?

What we learn during our discovery?

朱芷鈴

- How to use Swift to create an App.
- Using my expertise and promote myself.
- The ability of solving problem.
- Reading paper more efficiently.
- Research something that I didn't really know.
- Being more carefully.

What we learn during our discovery?



林劭芸

- Swift language
- app UI development
- Swift connect Flask
- Flask connect Mysql

Thank You

