

# An App to Detect and Minimize Stress

Easten Wember

# Description + General Requirements

Design an application that encourages us to be good to ourselves by providing stress-relieving tips and/or exercises and that can monitor stress triggers.

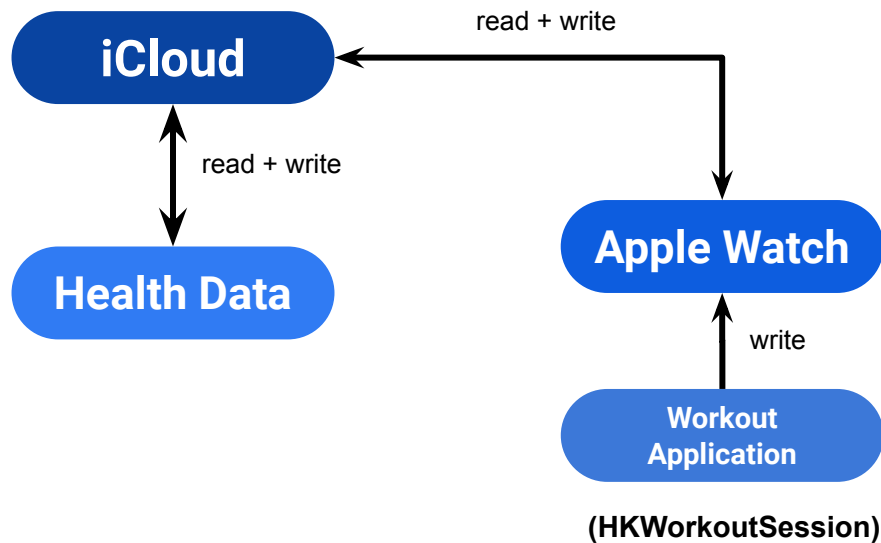
---

1. Interface the application with sensor data from an Apple Watch.
2. Investigate the reasons for stress and anxiety on college campuses.
3. Identify triggers or indicators that might be indicators of stress.
4. Gather health data during workouts and times of rest.
5. Provide historical data to help identify patterns and/or significant changes.
6. Inform user of times of high stress or significant change in health data.

# Methods: Experimenting with Apple Watch Development

- Workout Application using Swift/SwiftUI
- HealthStore → privacy and authorization

```
healthStore.requestAuthorization(toShare: typesToShare, read: typesToRead) { (success, error) in }
```



# Methods

- HKWorkoutSession
  - HKLiveWorkoutBuilder
  - Update statistics → dispatch queue
- HealthKit → accessing data through HKStatistics

```
guard let statistics = statistics else { return }  
//unwraps statistics  
let stepCount = HKUnit.meter()  
self.stepCount =  
statistics.sumQuantity()?.doubleValue(for:  
stepCount) ?? 0
```

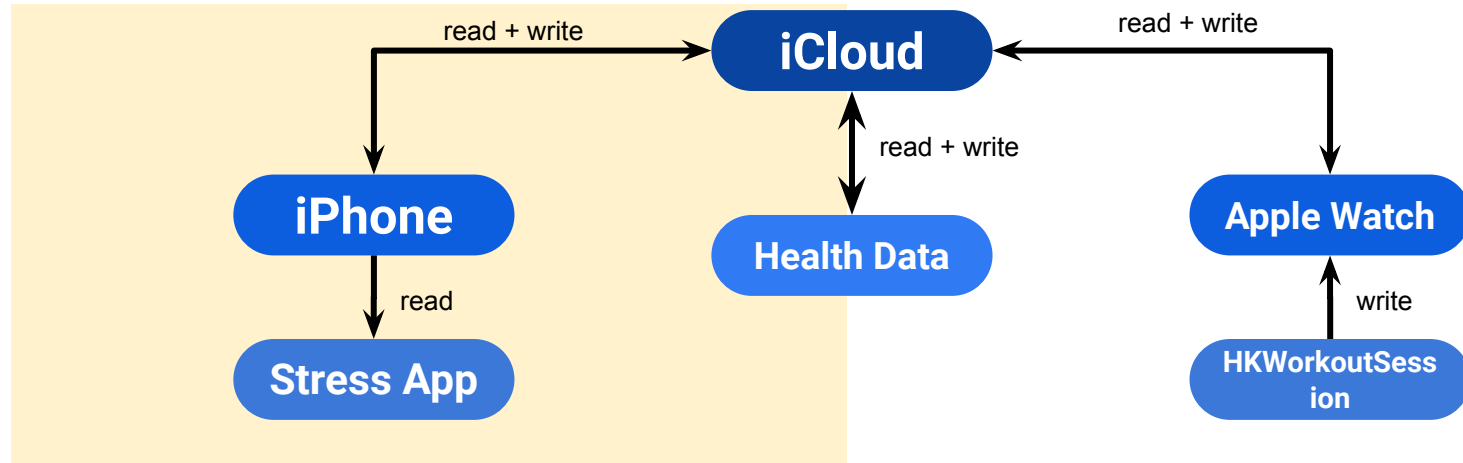


# Limitations and Experiments

- Implementing requirements on the watch application
  - Limited screen space to interpret/display data
  - Unable to push notifications locally
  - Application not for long term/background use
- Two-way communication using Watch Connectivity
- Switch to iOS application
  - Unable to use HKWorkoutSession

# Solution

- HKHealthStore to request authorization
- getLatestHR() function
  - HKSampleQuery → returns all samples currently saved in HealthKit store
  - Takes: quantityType, predicate (current date), sort descriptor (most recent values first)



- HKSampleQuery returns data once
  - Timer executes getLatestHR() every 30 seconds

```
let timer = Timer.publish(every: 30.0, on: .main, in: .common).autoconnect() //timer goes off every 30 second
```

```
.onReceive(timer, perform: { _ in //when landingView call received, timer is called fcn to perform is timerHR  
    print("in onReceive")  
    timerHR() //fcn that calls getLatestHR()
```

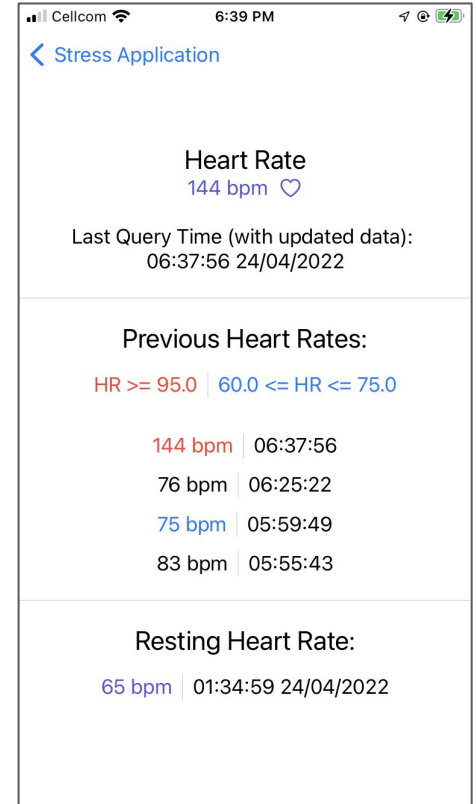
- Query does **not** force new data to be generated, just asks for the most recent values stored in the HealthKit
- New samples stored in HealthStore generated in a range of 1-20+ minutes

# Demonstration



# Application Design

- Show the query time of current heart rate
- Previous heart rate array displayed with their query times
  - Blue for calm heart rates (60.0 - 70.0)
  - Red for high heart rates (95.0+)
- Display resting heart rate
  - `getRestHR()`
  - `HKSampleQuery` for resting heart rate



# Development Process and Sources

- CSCI 350
  - Background on mobile development
- Apple WWDC tutorials
  - Video tutorials + explanation of Swift/SwiftUI
- Trial and error projects
- Professors

# Extensions

- Usage of different queries
  - HKAnchoredObjectQuery → returns most recent data and monitors for changes
  - HKActivitySummaryQuery → runs in background and monitors for changes
- heartRateVariabilitySDNN quantity sample
  - Measures standard deviation of heart rate intervals
- Implement two-way communication
  - Apple watch application with iOS companion for data communication
- Graphs based on data
- Breathing animation exercises

Questions?

```

guard let sampleType = HKObjectType.quantityType(forIdentifier: .heartRate) else{
    return
}

let startDate = Calendar.current.date(byAdding: .month, value: -1, to: Date())

//let endDate = Date()

let predicate = HKQuery.predicateForSamples(withStart: startDate, end: Date(), options: HKQueryOptions.strictEndDate)

let sortDescriptor = NSSortDescriptor(key: HKSampleSortIdentifierStartDate, ascending: false)

//result handler to health store executuin query
let query = HKSampleQuery(sampleType: sampleType, predicate: predicate, limit: Int(HKObjectQueryNoLimit), sortDescriptors: [sortDescriptor]) { (sample, result, error) in
    guard error == nil else{
        print("query issue")
        return
    }//end error guard

    let data = result![0] as! HKQuantitySample //grab the data in pos 0..should be HKQuantSamp HR
    let unit = HKUnit(from: "count/min")
    let latestHR = data.quantity.doubleValue(for: unit)
    print("Latest HR: \(latestHR) BPM")
    print("count: \(self.count)")

    self.theHR = latestHR //give theHR whatever updated HR is...put in theHR bc we know the type of var for sure

    //used to be let(s)..tried as vars bc let = immutable..need var instead of let! else doesnt update?
    let dateFormatter = DateFormatter()
    dateFormatter.dateFormat = "hh:mm:ss dd/MM/yyyy"
    let startDate = dateFormatter.string(from: data.startDate)
    let endDate = dateFormatter.string(from: data.endDate)
    print("StartDate: \(startDate) ... EndDate: \(endDate)")

    let timeFormatter = DateFormatter() //used to print just time next to previous heart rates
    timeFormatter.dateFormat = "hh:mm:ss"
    self.timeDate = timeFormatter.string(from: data.startDate) //hold time of queries
    print("Time date: \(self.timeDate)")

    self.theDateString = startDate //give theDateString the start date of curr query
}

healthStore.execute(query)

```



HealthData

Capstone\_iOSAppApp



Capstone\_iOSApp



LandingPage



Info



Capstone\_iOSApp > Info > No Selection

Key	Type	Value
Information Property List	Dictionary	(5 items)
> NSExtension	Dictionary	(2 items)
> Required background modes (Watch)	Array	(1 item)
Privacy - Health Share Usage Description	String	Allow this app to share your Health data to doctors and clinicians.
Privacy - Health Records Usage Description	String	Access data to provide summaries and notifications
Privacy - Health Update Usage Description	String	Allow this app to write health data provided from your doctor or clinician.