Keelyn McNamara
5/05/2023

RISE Advanced Fitness Data
Final Report

Abstract
The industry for health and fitness technology has exploded, making way for new expensive applications and devices. These applications track advanced fitness metrics like recovery from sleep or the difficulty of a workout. RISE is an iOS application on the Apple App Store that provides this advanced fitness for free. RISE utilizes personalized Apple Watch data to track a user's Recovery, Workout Intensity and Daily Strain. Using Apple's HealthKit API, RISE takes a user's heart rate and workouts to provide more than just individual metrics. RISE not only provides a daily Recovery percentage, Workout Intensities and Strain score but also displays this weekly data using engaging data visualizations. RISE is an easy to use, low-cost way for Apple Watch users to get more out of their device. RISE allows users to deepen their understanding of their health and get closer to their fitness goals.

Introduction
Health based technology has advanced rapidly over the last decade allowing anyone who can purchase these devices the ability to track their individual fitness and health metrics. RISE pushes Apple's health data services farther, past individual metric tracking, and into more advanced fitness analysis. While RISE is not the first application to provide this type of fitness data, it will be the first to do it free of charge or subscriptions. Fitness trackers and devices like WHOOP and Peloton are expensive with subscriptions ranging from $30 to $44 a month, not even including equipment. While these companies target those who already consider themselves athletes, RISE can be used by anyone with an Apple Watch. RISE will track a user's daily Recovery, Workout Intensities and Strain using health data from the user's Apple Watch, completely free of charge. Using HealthKit, Apple's health database, RISE will retrieve a user's daily workouts, activities, and heart rates to calculate how intense their workouts are and how much cardiovascular strain they're putting on their bodies as well as how many hours they're recovering at night. Each metric calculation is custom to the user and their own data, providing personalized analysis. By showing more than just individual statistics, RISE allows anyone, whether they consider themselves to be an athlete or not, to better understand their body's needs. Users no longer must pay high prices to achieve their health and fitness goals.

Related Work
There are over 18,000 health applications on the Apple App Store, many of these apps calculate similar metrics to RISE. The most popular among these health and fitness apps are WHOOP and Peloton. The WHOOP fitness tracker is very similar to the Apple Watch except it is solely a heart rate monitor; no screen or time is shown. WHOOP and Apple Watch utilize the same photodiode technology to measure a user's heart rate.[1] Heart rates and other associated metrics are tracked on these devices using the light emitted from the photodiodes that touch the user's

[1] "Establishing Optimal Signal Quality for Monitoring Heart Rate." 2023. WHOOP. March 22, 2023. https://support.whoop.com/s/article/Establishing-Optimal-Signal-Quality-for-Monitoring-Heart-Rate.

skin as the device is worn[2]. These photodiodes can measure the volumetric variations of blood circulation[2]. This methodology of using light to calculate changes in heart rate is known as photoplethysmography (PPG).[2] According to Alaa Khushal and his colleagues, "The measurement of heart rate during acute exercise is one of the most common and pragmatic methods for estimating exercise intensity and prescribing exercise training thresholds as well as measuring the heart rate recovery after intense work.[2]" The data that WHOOP provides is daily strain, workout intensities, sleep performance and recovery analysis. Because WHOOP's wearable heart rate tracker is only a measuring device it can be worn for long periods of time without charging. This allows data to be collected the entire night. WHOOP bases their strain calculation off the Borg Scale of perceived exertion and displays the metric from 6-21.[3] It is important to note WHOOP's strain score does not decrease throughout the day and the more strain you accumulate the harder it is for you to continue to increase your strain[3]. Strain is believed to be logarithmic rather than linear as, Director of Analytics at WHOOP, Emily Capodilupo, states "The longer you go, the harder you go, the harder it becomes to build strain.[3]" WHOOP is able to provide users with incredibly accurate and personalized data because users can wear the watch constantly. Unfortunately, WHOOP is only marketed to those who already consider themselves serious athletes. They also require a monthly base subscription of $30 which is a significant financial burden over time. RISE delivers very similar metrics but for free, allowing not just professional or collegiate athletes access this advanced data, but anyone with an Apple Watch.

Peloton has also revolutionized health and fitness tracking with their equipment and programs. When using their machinery, users can have their Workout Intensity tracked for them as long as they have a compatible heart rate monitor. Peloton markets their "Strive Score" as "a personal, noncompetitive metric based on your heart rate.[4]" It is calculated based on how much time a user spends in certain heart rate zones[4]. Points are awarded based on the amount of time spent in each zone. The harder the user works the higher the Strive Score becomes[4]. There is no set scale for this Strive Score which makes it harder for users to understand overall intensity.

Overall, these two applications are based on sound research and utilize verifiable metrics like heart rate for their data analysis. WHOOP and Peloton's downfall is their price tag, as they exclude the public from being able to access this analyzed fitness data. Most people have already bought a device (like the Apple Watch) that can calculate advanced fitness metrics. They must pay an additional large sum to actually see their metrics. RISE provides very similar data analysis as these popular apps but free of charge for anyone to use.

Process and Methodology

Design

RISE's four main deliverables are a Recovery percentage, Intensity score, Strain score and Evaluation of weekly metrics. The first three are calculated metrics that rely on heart rate data. The Evaluation deliverable relies on the successful calculation of the other three metrics.

---

[2] Khushhal, Alaa, Simon Nichols, Will Evans, Damien Gleadall-Siddall, Richard Page, Alasdair O'Doherty, Sean Carroll, LeeIngle, and Grant Abt. "Validity and Reliability of the Apple Watch for Measuring Heart Rate During Exercise." Sports Medicine International Open 1, no. 06 (2017): E206–11. https://doi.org/10.1055/s-0043-120195

[3] Ahmed, Will, Kristen Holmes, and Emily Capodilupo. "Understanding Strain | Cardiovascular Exertion | WHOOP Podcast." WHOOP. Accessed September 15, 2022. https://www.whoop.com/thelocker/podcast-26-understanding-strain/.

[4] Peloton. "Peloton's Strive Score Is the Latest Way to Track Your Performance." PELOTON | The Output, April 30, 2021. https://blog.onepeloton.com/strive-score/.

Recovery- The Recovery metric tracks time in bed using the sleep analysis built into Apple's HealthKit. Unfortunately, Apple Watches are not typically worn at night as they are charging. This means that heart rate data cannot be collected. Sleep on the Health App tracks roughly the time a user goes to bed by analyzing phone usage and haptic sound around the phone. Sleep on the Health App also tracks the time the user wakes up using their alarm data[5]. Using these bed and wake times, Sleep can calculate the time the user spends in bed (See Figure-1 for Sleep UI representation). While this is not always fully accurate, RISE has a built-in function that allows the user to set the hours that they were in bed as well as their sleep quality. This is in case the Health App's Sleep calculated time in bed is incorrect. The sleep quality function lets the user pick the percentage of the night they were asleep. The number of hours spent in bed is multiplied by sleep quality and is compared to the recommended 8 hours a night by the CDC[6]. This fraction is then multiplied by 100 to give the final Recovery percentage. See Figure-2 for full Recovery equation. This metric, while not incredibly personalized, does allow the user to track their sleep and overall Recovery.

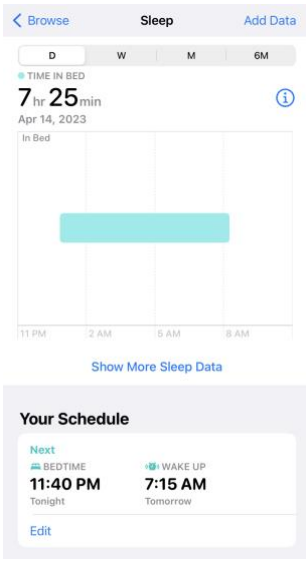

Figure-1
Sleep tracker UI on Apple's Health app tracking time in bed.

$$\frac{\text{(Awake Time - Bed Time) X Sleep Quality Percentage}}{\text{8 Hour Sleep Standard}} = \text{Percent Recovered}$$

Figure-2
Recovery percentage equation uses the time in bed multiplied by the sleep quality percentage and compares that to 8-hour sleep standard. Recovery is expressed as a percentage.

Intensity -

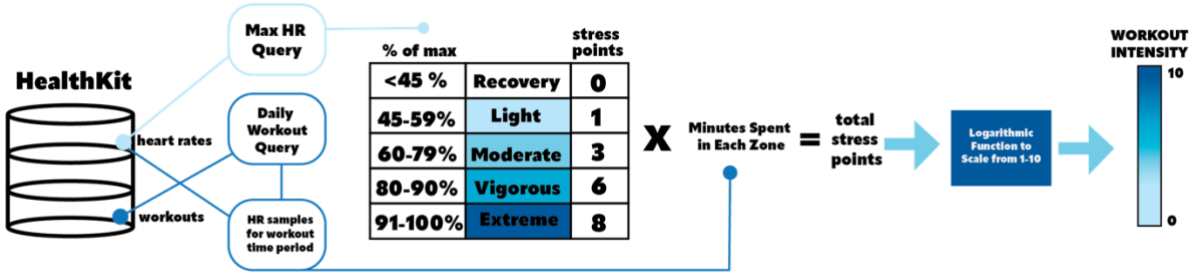

Figure-3
Visual representation of Workout Intensity calculation.

The Workout Intensity metric is defined by how much cardiovascular effort you are putting into your workout. When a user tracks a workout on their Apple Watch, it is recording heart rate samples every 1 to 5 seconds. This heart rate data is saved in the user's HealthKit database each

[5] "Track Your Sleep on Apple Watch and Use Sleep on IPhone." 2022. Apple Support. September 12, 2022. https://support.apple.com/en-us/HT211685.

[6] Center for Disease Control and Prevention. "Target Heart Rate and Estimated Maximum Heart Rate." Centers for Disease Control and Prevention, June 3, 2022. https://www.cdc.gov/physicalactivity/basics/measuring/heartrate.htm.

day. When a user wants to see their Workout Intensities, RISE queries the heart rate samples that were recorded during the user's workout. RISE also pulls the user's max heart rate over the last 6 months. Custom heart rate zones are calculated based on certain percentages of that queried max heart rate. These heart rate zones correspond to the effort at which you are working (See Figure-3 for visual representation of the heart rate zones.) RISE utilizes a custom max heart rate rather than one based on the user's age. This allows for more accuracy within the heart rate intensity zones. Heart rate zones are used by Peloton and WHOOP in their calculations for workout intensity[7]. According to the CDC, "...moderate-intensity physical activity, your target heart rate should be between 64% and 76% of your maximum heart rate" and "For vigorous-intensity. physical activity, your target heart rate should be between 77% and 93%[6]." RISE zones are altered slightly to be more understandable to users and to have larger ranges in the moderate category as it is the targeted safe range for users[6].

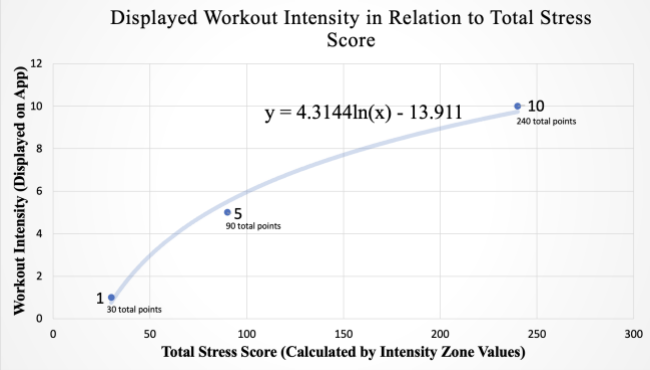

Figure-4

This graph shows the logarithmic function used to map stress points to a 0-10 scale. Utilized low, medium and high intensity workouts for 30 minutes to create RISE's Workout Intensity scale.

Each zone is given a specific number of stress points based on the effort needed to stay within that zone. This is the same methodology that Peloton uses to calculate their Strive Score[8]. Peloton's zone points begin at 1 then go to 2, 4 and 8, vigorous and extreme zones both being 8[8]. RISE's zone points are based on how hard it is to go from one zone to another, for example to go from a light zone to an extreme zone you would have to put in 8 times more effort. For every minute that the heart rate samples stay in a zone the user is awarded that number of stress points. The more intense the workout, the higher the total stress points are. These total points are then applied to a logarithmic function that puts them on a 0-10 scale (See function in Figure-4). The function was calculated by utilizing the total stress points that correspond to thirty minutes of light, moderate and extreme effort. These 3 points mapped the minimum of the Workout Intensity scale being a 1, then the middle being 5, and finally the maximum at 10. This function converts the total amount of stress points into an understandable Workout Intensity value from 0-10. A logarithmic function was used based on WHOOP's finding that "Strain is Logarithmic, Not Linear.[3]" The more effort a user puts in the harder it is for them to maintain that effort or increase it. Not only does the logarithmic scale more accurately represent Workout Intensity and effort but it allows users to better interpret their results on a scale that is familiar.

[7] Ahmed, Will, Kristen Holmes, and Emily Capodilupo. "Understanding Strain | Cardiovascular Exertion | WHOOP Podcast." WHOOP. Accessed September 15, 2022. https://www.whoop.com/thelocker/podcast-26-understanding-strain/.

[8] Peloton. "Peloton's Strive Score Is the Latest Way to Track Your Performance." PELOTON | The Output, April 30, 2021. https://blog.onepeloton.com/strive-score/.

Strain- The Strain metric continuously tracks a user's cardiovascular strain from their daily activities. This metric changes throughout the day depending on hours that the user is inactive or active. To calculate Strain, the resting heart rate for the user is queried each day. Every heart rate sample recorded in the day is compared to that resting heart rate. If the heart rate falls anywhere between the resting heart rate and 45% of the max heart rate, then the user is said to be inactive or in a recovery state. The time the user spends in other heart rate zones during the day is applied to the Workout Intensity calculation. A user is given a Workout Intensity for the times where their heart rate was elevated throughout the day but not actually tracking a workout. The heart rate samples taken during actual workouts are not applied to this daily elevated heart rate Workout Intensity. Elevated heart rate Intensity is summed with Workout Intensities to add to the Strain score. Inactive time subtracts away from Strain as the heart rate is in a recovery phase. To allow for the metric to be on 0-10 scale, RISE has a custom weighted equation (See Figure-5). Workout/elevated heart rate Intensities are summed and divided by 2, while hours inactive are divided by 4 and subtracted from the overall Intensity sums. WHOOP tracks a user's strain using similar metrics like elevated heart rates and Workout Intensities but does not factor in recovery time[7]. RISE's original Strain calculation utilizes the same metrics as top competitors but simplifies it for users to understand.

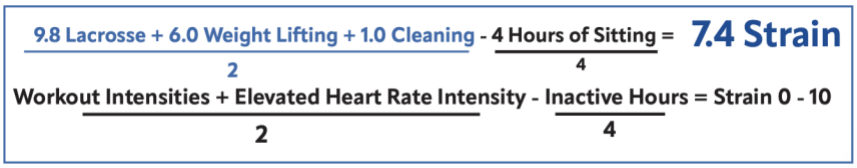

Figure – 5
The top color-coded equation is an example using real life data. The bottom equation shows the overall Strain equation and the weights applied to each component.

Challenges

Methodology challenges were the biggest hurdles to overcome for this project. The main compute of this application relies on the querying and handling of massive data samples. Each metric required several different queries and comprehensive algorithms to generate a final score or percentage. The main challenge of the methodology was understanding the HealthKit API and querying it for data. Lots of research had to be done to understand what fitness data was available and how it was packaged. The first struggle came from a misunderstanding on HealthKit's workout objects and what data was stored in each workout. The HealthKit stores each workout with several pieces of metadata attached like name of workout, start and end time[9]. I assumed that it also stored heart rate samples, but the workout object is not big enough to store the whole collection of samples. Instead of querying just the workout objects for the samples, I had to write a new workout query to get the start and end times of the workouts. Then use those times in a heart rate sample query to get the heart rates that occurred only during those specific workout times. Once the data was queried, challenges arose when trying to store and do calculations on the heart rate samples. The collection of heart rates for each workout were far too big to be stored in a simple array. This meant that as they were queried, they had to be analyzed and put through the Intensity calculation. While this required a lot of complex function calls and arithmetic, I was successfully able to process each heart rate collection and output a final Workout Intensity. This is a great example of the learning curves that come with a new API. A

[9] "About the HealthKit Framework." Apple Developer Documentation, https://developer.apple.com/documentation/healthkit/about_the_healthkit_framework. Accessed 3 May 2023.

lot of my initial design was built from research and how I assumed the HealthKit worked. My software design needed to be constantly tweaked as I learned more about how the HealthKit actually worked.

Another huge hurdle with learning and using the HealthKit for the first time was dealing with the asynchronous queries. I had never dealt with multithreading, so it was difficult at first to trace my code and find the source of errors. After researching and meeting with my advisor, I was able to use timers/delays to align my code with the asynchronous queries. Unfortunately, these queries are time extensive due to all the data they are pulling. For example, a heart rate query for a 2-hour workout will pull over a thousand heart rate samples, this takes extensive compute time so the code must wait until this data is available to proceed with its calculations. While a timer is not the most secure and efficient methodology it was what allowed me to successfully run the calculations. As a beginner programmer in swift and multithreading some of my code is clunky and inefficient but it does work. By my last metric of Strain, I began to utilize completion handlers which wait to execute functions until other functions are complete. Concurrency with the data was difficult to handle but it was a learning process that allowed me to grow and utilize more efficient strategies over time.

I found that most of my mistakes came from a lack of knowledge and understanding. It took a lot of trial, error and research to fully grasp and understand how Swift's APIs worked. In the beginning, I assumed that Apple would allow app developers to query and access a user's phone activity and phone usage. This was how I was going to track time in bed and therefore calculate Recovery. For security and privacy purposes Apple does not allow outside applications to look at a user's phone usage. Only select apps can be monitored by other apps and this feature is still limited. This was a huge hurdle as I no longer had a reliable way of tracking a user's bedtime. Fortunately, after further research I found that Apple's Health Sleep feature will track bedtime and wake time on its own. These metrics are stored also in the HealthKit and can be queried like any other health data. This was a great work-around as Apple handled the bedtime analysis and RISE simply had to query the data. Overall, I was able to find workable solutions to the problems that arose. These challenges pushed me to deepen my understanding on software design and look for creative solutions.

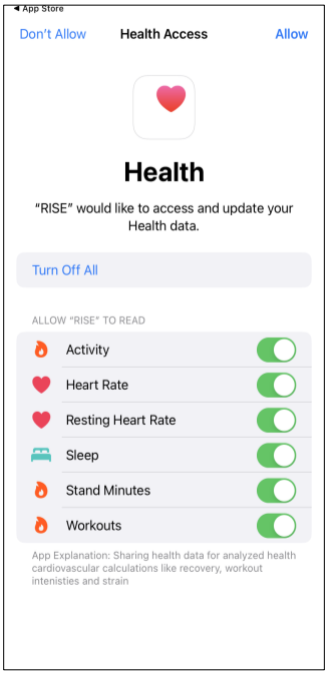

Figure-6
Pop up menu that Apple generates to let users select which data RISE is allowed to utilize.

Results – Features

All features that were initially proposed were successfully implemented. RISE has four main pages, Recovery, Intensity, Strain and Evaluation. A menu at the bottom of the screen allows users to toggle between pages. The first feature that was implemented was allowing RISE to have access to the user's HealthKit and alerting the user of the data that RISE is viewing. This feature is a Swift UI generated pop-up menu that asks the user for permission for RISE access and view specific health data. The user can allow specific health data to be viewed and used. See Figure-6 for a visual of this feature.

Recovery

Recovery is the first metric that is displayed. This screen shows the number of hours the user was in bed, their sleep quality, their Recovery

percentage, and a simple bar graph that visualizes the percentage. This Recovery page is interactive as the sleep quality and time slept can be edited. By clicking the grey “Sleep Quality” button, a menu will pop up that asks the user to rank their sleep quality on a 1-10 scale (See Figure-7). This correlates to how much of the night they were asleep. Once they select a specific number, the change will be reflected in the Recovery percentage and graph. The grey “Time Slept” button functions the same way as users can adjust the hours they were sleep if the calculated time is inaccurate (See Figure-8). The overall displayed percentage controls the height of the graph. When the Recovery percentage updates, so will the visualization. To update the Recovery data the blue “Refresh” button can be used. This will refresh the data if a new day has occurred or if the user wants to go back to the original queried hours. Figure-9 displays entire Recovery UI.

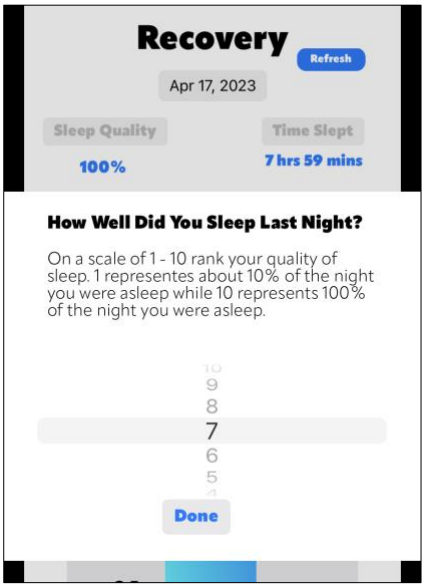

Figure-7
Pop-up menu for setting sleep quality. This UI utilizes a scrollable menu for the user to select a number.

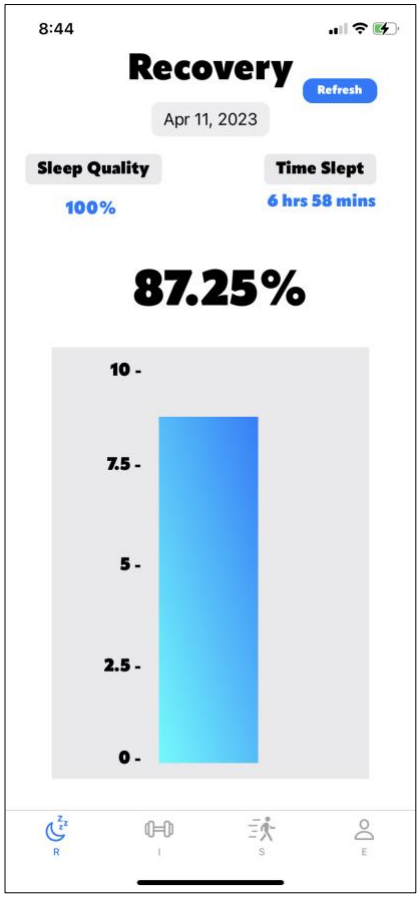

Figure-9
Recovery tab of UI that displays Recovery percentage, visualization, sleep quality and time slept.

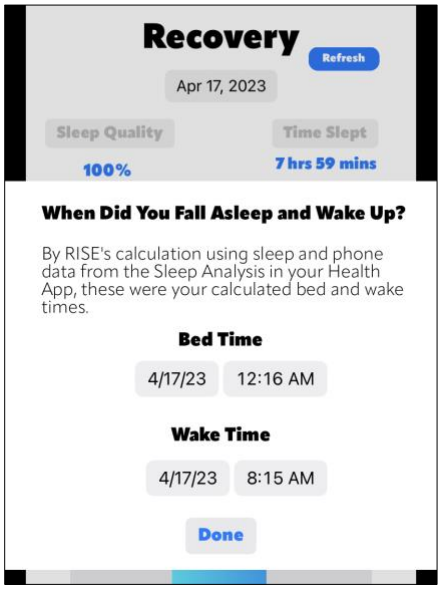

Figure-8
Pop-up menu for setting time slept hours. Times displayed are bed and wake times.

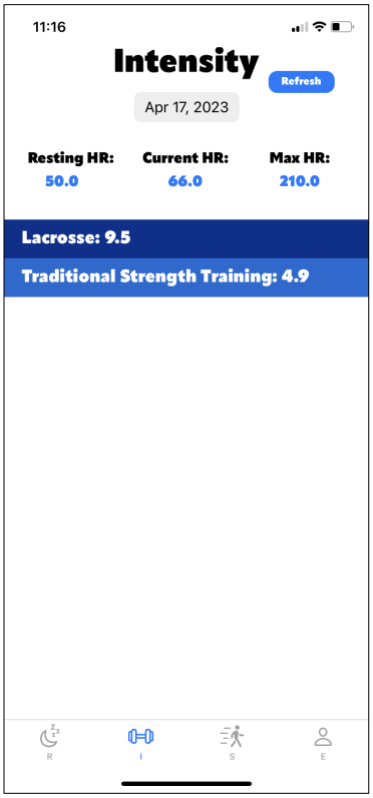

Figure-10
The Intensity page will display the names of the workouts as well as the intensity in a color-coded list.

Intensity

The second metric that is displayed is the Workout Intensity. This page is the second option in the bottom menu. The Intensity tab first will display the users resting heart rate, current heart rate and maximum heart rate for that day. If the user has logged any workouts, they will be displayed in a list on the screen. The type of workout will be listed as well as the Workout Intensity. The background color will also change based on the intensity of the workout. The higher the Workout Intensity is, the darker blue the background color is. These UI features are shown in Figure-10. The Intensity tab will display up to 10 daily Workout Intensities. The user does not track the workouts using RISE. The user simply tracks the workout on their Apple Watch and RISE will out put an Intensity. If the user records several workouts throughout the day, they can click the blue refresh button and have their new workouts displayed. If the user does not have any workouts for the day, only the heart rates will be displayed. This Workout Intensity calculation is time consuming and requires several intensive heart rate sample queries. This is why the Workout Intensity list takes several seconds to populate.

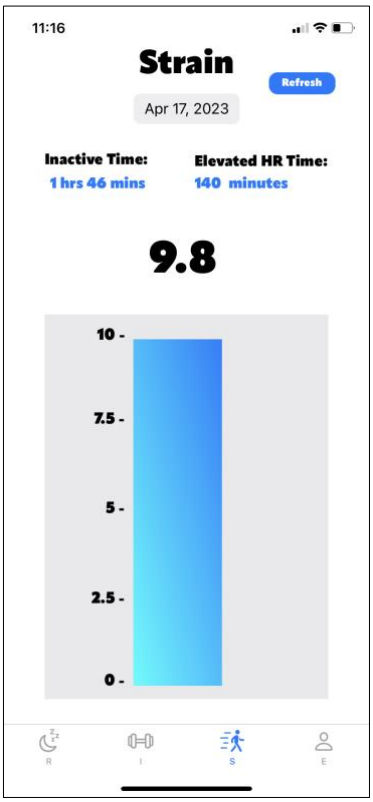

Figure-11
The Strain page displays the calculated daily Strain for the user while also providing total inactive and elevated heart rate time.

Strain

The third tab on the bottom navigation menu is the user's daily Strain. The UI is similar in appearance to the Recovery display. The same graph visualization is used to show daily Strain on a 0-10 scale. The graph is animated to move up and down depending on how the Strain changes throughout the day. The Strain is displayed at the top with the user's active and inactive hours. The daily Strain is also a very time expensive calculation as an entire day's worth of heart rates are being analyzed on the backend of this display. This is the most time intensive calculation, so the UI has the longest load time. After 8-10 seconds the active hours, elevated heart rate hours and Strain will be updated to the screen. The daily Strain will always start at 0.0 and be updated once Intensities are added. A user can refresh the data using the blue refresh button. This will set all the data points back to zero and recalculate each of the displayed data points. These UI features are shown in Figure-11.

Evaluation

The last page on the UI is the Evaluation tab which visualizes a week's worth of Recovery, Intensity and Strain data. The data is saved in a SQLite database by day then queried to populate each graph. The Recover and Strain graphs utilize a bar chart where and the Workout Intensity graph utilizes a line chart. Images of all graphs can be seen in Figure-12. The weekly averages for Strain and Recovery are also displayed on each chart.

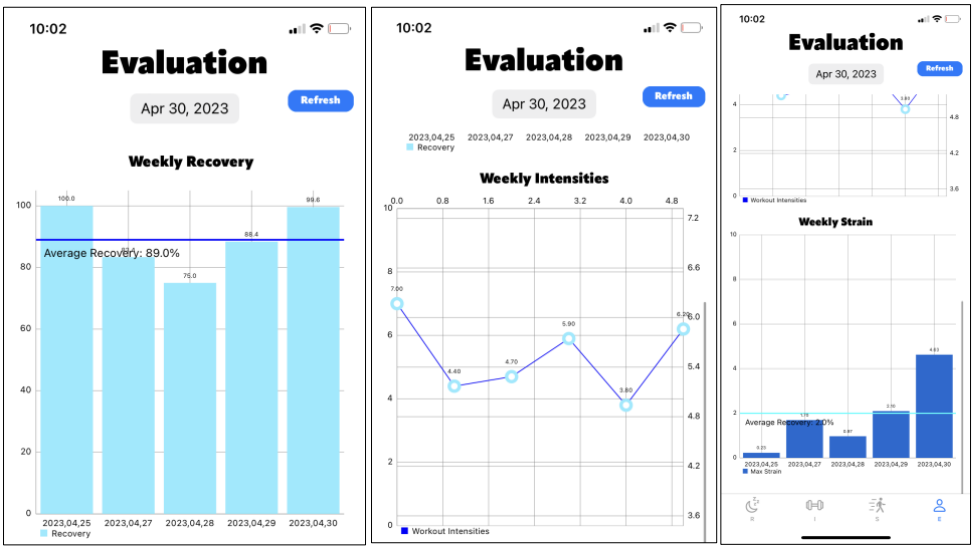

Figure-12

The Evaluation page is scrollable allowing the user to see a weekly record of all three of their metrics.

The user can scroll down on the Evaluation page to see each graph. The page can also be refreshed as the metrics throughout the day change. Once a day has past, edits cannot be made to the metrics tracked on that particular day.

Results – User Experience

One of the biggest accomplishments of RISE was being accepted into Apple's App store. This allows anyone with an iPhone to download RISE and test it out. Being on the App store allows for fast and easy distribution to clients and it is free of charge. The accessibility in both price and software was a huge motivation for the project. RISE successfully accomplished its goal of not only calculating and visualizing advanced health data but also providing it for free. RISE is not just an application on my computer or my personal device. It is now a global software that anyone with an Apple Watch can use.

Upon first release several bugs were found by test users. The Evaluation charts did not load or store data and if Apple Watch data was not present then the app would force quit. These were all bugs that did not come up in my personal device testing. With continuous testing and the addition of SQLite frameworks being added to my overall Swift project file, I was able to fix the charts problem and properly visualize the weekly metrics. Unfortunately, due to time restraints, users without Apple Watch data may still experience shutdowns of the app as no calculations can actually be made. These are unusual cases as users with Apple Watch data are RISE's primary clients.

Load time is another factor of RISE's user experience. RISE currently has faster load time than competitors. WHOOP may take several minutes to load the user's data while RISE only takes a few seconds. As of now the load time for RISE is around 8 seconds which is not instant. In talking with some of the first RISE testers, the load time was confusing, as the user did not wait to see the data populate the screen. Once it did, they realized the data had been calculated. After continued usage the user can better understand the load time and when their data will appear. While faster than competitors, it would be beneficial to continue to increase efficiency within the application and decrease the load time for almost instant data display.

Limitations

While all intended features were implemented, there are a few limitations to note that occurred due to time constraints as well as lack of knowledge. Most of the limitations occurred within the metric calculations as they became incredibly complex. For example, the Recovery metric was supposed to be dependent on the previous day's Strain. This would have required more SQLite data base querying and error handling for when the user did not have data for the prior day. I did not have much experience with SQLite and already was having challenges just storing and displaying the saved data. I did not add this extra facet to the Recovery calculation as I did not have the time to add more complexity to the calculation. The Strain data was also supposed to be carried over from day to day. This became difficult to configure as the Strain is changing constantly. The bedtime Strain may be a lot lower than the highest Strain recorded for that day. Only the highest Strain is logged in the user's personal database, so it was difficult to design this feature into the already working Strain calculation. I was on a strict schedule for completing calculations and designing the UI, so I only had the time to get the calculations working at their individualized form. These added complexities within the calculations could provide for more accuracy but it was not worth breaking the established and successful code.

Another limitation is the timing of the calculations and when they are being performed. RISE only calculates data when the app is opened. If a user forgets to open RISE for an entire day, their data for that day will be not be recorded. The user must click on each page to have each metric calculated. Background calculations was a feature that I would have liked to install if provided more time. I did not have enough knowledge of Swift or concurrency to program an app that is constantly running. As of right now, RISE requires the user to open the app throughout the day and visit all pages for the three metrics to be calculated and recorded. This means the actual highest Strain of the day may not be recorded if the user does not open or refresh the app after a workout or time of elevated heart rate. If the user only opens the app at the end of the day, then their highest Strain for that day may not be incredibly accurate. This is because the entire day's inactive time is being subtracted from the total Strain. If the user refreshes the app several times a day, then the Strain will constantly be updated and the highest Strain recorded may be more accurate. None of the other metrics are affected by time of opening the application as the Recovery and Intensity metrics can be refreshed and updated using the UI.

One of the biggest limitations of RISE is handling cases where Apple Watch data is not present or accessible. Unfortunately, due to lack of understanding and limited time, RISE's error handling for null values is not full comprehensible. If a user with limited to no data tries to use RISE, the app force quits. This is because RISE's calculations are dependent on the existence of data like heart rate samples. If RISE cannot find them then none of the calculations can be executed. Instead of force quitting, I would like RISE to send the user an error message that notifies them they need an Apple Watch to use this application. This would help improve user experience as it would inform the user of the error rather than just shutting down. This lack of proper error handling resulted from another limitation of narrow environment testing. Before launching RISE, I had only been testing it on my personal iPhone which has years worth of health data from my Apple Watch. This provided the perfect testing environment which may not have been effective for exposing possible bugs or testing one-off cases. Limitations in time as well as access to devices stopped me from trying out other testing environments until after releasing on the App Store. Overall RISE functions well in the environment it is supposed to function in but struggles when it faces unexpected cases.

Conclusion

RISE was ultimately a very successful project in which all the initial goals were accomplished. The final product is something I am very proud of. It calculates the desired metrics, is free of charge and I use it in my daily life. While the RISE application is the final deliverable to come out of this year long project, I gained a lot more than just a fitness app. I learned a completely new software and API as well as advanced programming principals in a hands-on way.

This project was built entirely from the ground up. I had never programmed in Swift before, built a UI or even programmed an entire application. To execute this idea of analyzing health data, I had to first understand the basics of the Swift and HealthKit API. I realized that I didn't have time to learn everything and then begin the project. I had to learn and program as I went, breaking up my methodology into small pieces. By compartmentalizing each metric, I was able to take the necessary steps to learn and then apply that knowledge to progress in the project. I started very basic with learning how to create a four-page UI menu and how to get text to appear on the screen. I then delved into the HealthKit API and did lots of testing with queries and snippets of code. I began just testing chunks of code, understanding what they were doing and then manipulating that test code into what I needed. I found tutorials and online code resources incredibly vital to this learning process. The online resources provided the foundation for this project, but I had to analyze and understand how the code was functioning to tailor it my own project. I really enjoyed the process of using what I knew to figure out what I didn't. This project was unlike any other assignment as I had to teach myself Swift, the other APIs and frameworks. This self-motivated work was incredibly rewarding as the success of this project relied solely on my ability to problem solve, adapt, and learn something completely new.

The hands-on approach not only helped me learn new programming languages but also different design tactics. Before RISE, I had never dealt with concurrency or multi-threading. I was forced into learning these principals as the HealthKit utilizes asynchronous queries. It was challenging trying to understand race conditions and controlling when processes would start and end. This was a great lesson into more advanced software design and helped me to see concurrency processes and issues firsthand. While it was a frustrating and slow process, I did deepen my knowledge in software design. I was forced to confront and understand common concurrency issues so that I could find a working solution. Each phase of the project produced new problems that had to be creatively solved. This helped me think outside the box, utilize my resources, and optimize my unit testing. I had to rely on the skills developed from my previous computer science courses to work through my lack of experience and the issues that came with that.

Over the course of the year, I gained an abundance of knowledge in iOS app development and software design. By overcoming challenges and frustrations, I improved upon my problem-solving skills and creative thinking. While I am incredibly proud of my product and having RISE on the app store, I am more satisfied with the soft skills I gained. I was able to build an app that I had conceptualized completely from scratch with very little help from professors or professionals. This senior project not only taught me about programming concepts and software design but also about myself. Being an interdisciplinary major, I did not think I knew everything I needed to be successful with this project. I have learned I am a lot more capable than I give myself credit for. A deep personal connection to this project and the motivation to learn was all I needed to accomplish my goals.

References

Apple. “IOS - Health.” Accessed September 22, 2022. https://www.apple.com/ios/health/.

“About the HealthKit Framework.” Apple Developer Documentation, https://developer.apple.com/documentation/healthkit/about_the_healthkit_framework.

Ahmed, Will, Kristen Holmes, and Emily Capodilupo. “Understanding Strain | Cardiovascular Exertion | WHOOP Podcast.” WHOOP. Accessed September 15, 2022. https://www.whoop.com/thelocker/podcast-26-understanding-strain/.

Boly, Jake. “What Is Heart Rate Variability? How to Use and Monitor HRV (Brought to You by WHOOP).” *BarBend* (blog), July 25, 2019. https://barbend.com/what-is-hrv/.

Borg, Gunnar. *Borg’s Perceived Exertion and Pain Scales*. Borg’s Perceived Exertion and Pain Scales. Champaign, IL, US: Human Kinetics, 1998.

Center for Disease Control and Prevention. “Target Heart Rate and Estimated Maximum Heart Rate.” Centers for Disease Control and Prevention, June 3, 2022. https://www.cdc.gov/physicalactivity/basics/measuring/heartrate.htm.

“Establishing Optimal Signal Quality for Monitoring Heart Rate.” 2023. WHOOP. March 22, 2023. https://support.whoop.com/s/article/Establishing-Optimal-Signal-Quality-for-Monitoring-Heart-Rate.

Deborah Riebe, Jonathan K Ehrman, Gary Liguori, Meir Magal. Chapter 6 General Principles of Exercise Prescription. In: ACSM’s Guidelines for Exercise Testing and Prescription. 10th Ed. Wolters Kluwer/Lippincott Williams & Wilkins, Philadelphia, PA: 2018, 143-179.

Jasmine Shaikh. “What Heart Rate Is Too High? Is 200 Bpm Bad?” MedicineNet, March 2, 2022. https://www.medicinenet.com/what_heart_rate_is_too_high/article.htm.

Khushhal, Alaa, Simon Nichols, Will Evans, Damien Gleadall-Siddall, Richard Page, Alasdair O’Doherty, Sean Carroll, LeeIngle, and Grant Abt. “Validity and Reliability of the Apple Watch for Measuring Heart Rate During Exercise.” Sports Medicine International Open 1, no. 06 (2017): E206–11. https://doi.org/10.1055/s-0043-120195

Labs, D. I. “What Is Exercise Intensity and How Do I Measure It?” WHOOP. Accessed September 15, 2022. https://www.whoop.com/thelocker/what-is-exercise-intensity-and-how-do-i-measure-it/.

Laskowski, Edward. “How Much Exercise Do You Really Need?” Mayo Clinic. Accessed

September 17, 2022. https://www.mayoclinic.org/healthy-lifestyle/fitness/expert-answers/exercise/faq-20057916.

Moran, Daniel S., Avraham Shitzer, and Kent B. Pandolf. "A Physiological Strain Index to Evaluate Heat Stress." *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology* 275, no. 1 (July 1998): R129–34. https://doi.org/10.1152/ajpregu.1998.275.1.R129.

Peloton. "Peloton's Strive Score Is the Latest Way to Track Your Performance." PELOTON | The Output, April 30, 2021. https://blog.onepeloton.com/strive-score/.

"Track Your Sleep on Apple Watch and Use Sleep on IPhone." 2022. Apple Support. September 12, 2022. https://support.apple.com/en-us/HT211685.