

Entwicklung einer nativen iOS-App

Einen Rad- und Lauf-Tracker erstellen

Willem L. Middelkoop

May 11, 2024



Als kleines Nebenprojekt, zwischen meine normale Arbeit gequetscht, habe ich an etwas gearbeitet, das mich persönlich interessiert: eine native Workout-Tracking-App für iOS. Ich wollte meine Smartwatch überflüssig machen und stattdessen mein Telefon zum Tracken von Workouts verwenden. Wie schwer konnte es sein, detaillierte Sensordaten mit nativen Swift-APIs zu sammeln?

Hintergrund

Wenn Sie meinem Blog folgen, ist dies vielleicht keine Überraschung, aber ich liebe es, Fahrrad zu [fahren](#) und habe vor kurzem angefangen zu [laufen](#). Ich habe einiges an Erfahrung mit dem Tracking meiner Workouts, mit Garmin, [Strava](#), [Omata](#), Wahoo, [Biostrap](#), [Whoop](#) und verschiedenen [Apps](#) und [Sensoren](#). Ich finde Gesundheits- und

Fitnessdaten nützlich, um etwas über die eigene [körperliche Leistung](#) zu lernen und eine gewisse Selbstverantwortung zu haben (z. B. die Motivation, die Workouts fortzusetzen).



Nichts gibt einem so ein Gefühl von Lebendigkeit, wie rauszugehen und sich körperlich zu betätigen - ich liebe es zu trainieren!

Der Nerd in mir liebt den großen Datensatz, die vielen verschiedenen verfügbaren Sensortypen sind ein Fest für meine Neugier - doch es kollidiert mit meinem Wunsch nach [Minimalismus](#) und Einfachheit. [Braucht man wirklich alle Trainingsdaten?](#) Reicht es nicht, [einfach auf seinen Körper zu hören?](#) Ich weiß es nicht, aber ich schätze die Fähigkeit einer Smartwatch, Workouts (fast) mühelos und detailliert zu tracken. Mir gefällt, wie die [Apple Watch zum Tracking von Trainingseinheiten verwendet werden kann](#), neben ihren Fähigkeiten als [moderne Toolwatch](#) und [Smartphone-Ersatz](#). Obwohl ich verstehe, dass die meisten Leute ihre Smartwatch tragen und es dabei belassen würden, mag ich [altmodische mechanische Uhren](#) wie eine [Rolex](#), [Tudor](#), [Grand Seiko](#) oder sogar [etwas individuell Angefertigtes](#). Obwohl [ich beide schätze](#), tendiere ich zu [mechanischen Uhren](#), wenn ich zwischen Smartwatches und mechanischen Uhren wählen muss.

Idee für die App: Gran Fondo

Was wäre, wenn es eine Smartphone-App gäbe, die alle gewünschten Trainingsdaten sammelt, ohne Aufwand und Unordnung? Ich stellte mir vor, sie wäre:

- **Einfach:** Sie muss sich "aus dem Weg räumen", so konzipiert sein, dass sie "einfach funktioniert", ohne dass man mit endlosen Einstellungen oder Optionen herumfummeln muss. Einfach "Start" drücken und mit dem Training beginnen.
- **Flexibel:** Die App auf verschiedene Arten verwenden: Montieren oder tragen Sie Ihr Telefon gut sichtbar für Echtzeitdaten oder stecken Sie Ihr Telefon in die Tasche und lassen Sie es im Hintergrund arbeiten.

- **Leistungsstark:** Verbindung mit Herzfrequenz-, Trittfrequenz-, Leistungs- und Geschwindigkeitssensoren über Bluetooth; Unterstützung mehrerer Sensoren, ideal, wenn Sie verschiedene Fahrräder verwenden.
- **Apple Health-freundlich:** Speichern Sie alle Daten in Apple Health mit maximalen Details, einschließlich Trainingsrouten und individuellen Leistungs-, Trittfrequenz- und Geschwindigkeitsmessungen. So können Sie Aufzeichnungen in jeder App analysieren, die Apple Health unterstützt, wie Strava, WHOOP, HealthFit und viele andere.
- **Ehrlich:** Keine Werbung oder Datenverkauf. Keine Konten oder Registrierung. Ihre Daten sollten privat, sicher und auf Ihrem Gerät bleiben.

Keine der bestehenden Sport- und Fitness-Apps für Smartphones erfüllt alle oben genannten Punkte. Die meisten von ihnen sind unnötig komplex oder irgendwie darauf ausgelegt, Ihre Daten zu verwenden, um dem App-Entwickler oder seiner Plattform zu dienen. Könnte ich es besser machen?

Entwicklung der App

Da ich wusste, warum und was ich wollte, machte ich mich daran, meine bevorzugte Methode für die Entwicklung der App zu wählen.

Native vs. Hybrid-/Web-Apps

Für die meisten meiner Arbeiten bevorzuge ich offene und freie (im Sinne von libre) Webtechnologien wie HTML, JavaScript und CSS - oder Serversoftware, die unter einer [freien Softwarelizenz](#) veröffentlicht wurde, wie z. B. GPL. Für diesen speziellen Anwendungsfall musste ich extrem energieeffizient sein und in der Lage sein, mich tief in das Betriebssystem meines Smartphones zu integrieren. Der beste Weg, dies für ein iPhone zu tun, ist, das Software Development Kit von Apple voll zu nutzen: die App nativ in Swift mit SwiftUI zu erstellen und die [technischen](#) und [Design- Richtlinien](#) so genau wie möglich zu befolgen.

KI-gestützte Programmierung

Obwohl [ich meine Bedenken habe](#), hat die Verwendung von KI für die Programmierung in letzter Zeit alle Aufmerksamkeit auf sich gezogen. Manche behaupten sogar, dass Technologien wie Copilot oder ChatGPT Programmierer überflüssig machen würden. Ich kenne mich mit KI aus, da ich sie zum [Generieren von Code](#) verwendet habe und [Leuten beibringe, wie man sie benutzt](#), aber ich habe sie nie verwendet, um eine *komplette* iOS-App zu erstellen. Ich dachte mir, dies wäre eine perfekte Gelegenheit, um zu sehen, wie gut KI für etwas Komplexeres als ein einfaches Code-Snippet geeignet ist. Lassen Sie uns herausfinden, wie obsolet ich als Programmierer in diesem modernen KI-Zeitalter bin...

GPS und Bluetooth Low Energy Sensoren

Nachdem ich meine Xcode-Umgebung eingerichtet hatte, bat ich OpenAI's ChatGPT, mir Code zum Lesen von GPS-Standortdaten und zum Generieren der notwendigen Swift-Klassen zum Erkennen und Verbinden mit Bluetooth-Sensoren zu geben. Junge, bin ich da in den sprichwörtlichen Kaninchenbau geraten!



Mein Fahrrad mit Bluetooth-Sensoren, verbunden mit meinem MacBook, auf dem Xcode läuft

Die Sache mit der KI-gestützten Programmierung ist die, dass sie, **JA**, funktioniert; aber nur **TEILWEISE**. Sie erhalten funktionierende Teile, aber sie passen nicht perfekt zusammen und funktionieren nicht unbedingt zusammen. Den generierten Code zu sehen ist wie das Hören von Echos eines großartigen Liedes; man erkennt, dass es etwas ähnelt - aber es ist nicht ganz dasselbe, wie die Band live zu hören; geschweige denn, es selbst zu singen!

Ich nahm die Teile von der KI und machte mich daran, die relevante Dokumentation selbst zu lesen, was sich als viel effektiver erwies, da es mir ermöglichte, tatsächlich zu verstehen, was der Mechanismus tun sollte. Die KI half mir, die Dinge zu finden, die ich lernen musste. Das Verfolgen von GPS-Standortdaten war mit [Apple's CoreLocation](#) überraschend einfach. Die Verbindung des Bluetooth-Sensors mit der App erwies sich als viel schwieriger, da sie viele verschiedene Schritte umfasst:

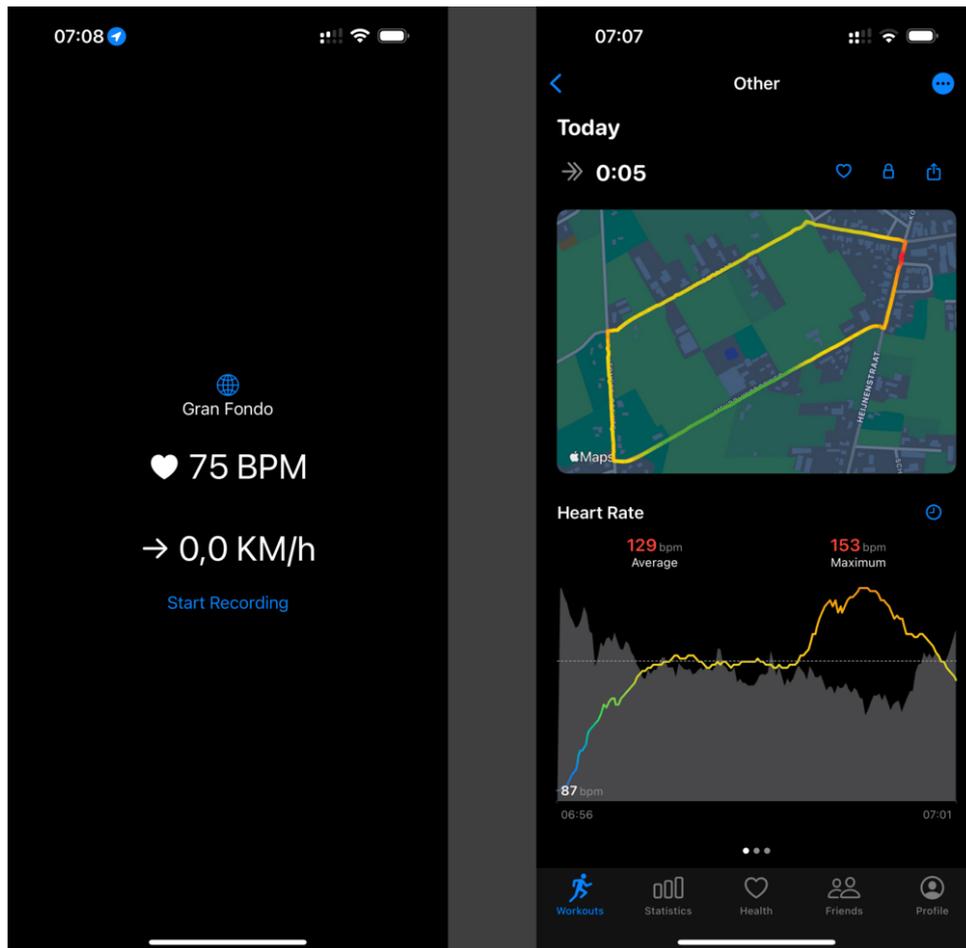
- **Erkennen von Bluetooth-Geräten:** Scannen der Umgebung nach verfügbaren und sendenden Geräten
- **Verbinden mit einem Bluetooth-Gerät:** Herstellen *und Aufrechterhalten* einer Verbindung zwischen der App und dem Gerät
- **Ermitteln der Eigenschaften von Bluetooth-Geräten:** Es gibt kein einfaches Bluetooth-Gerät, man muss die Fähigkeiten und Merkmale eines Geräts selbst "entdecken"; viele Geräte senden mehrere verschiedene Signale für Sensordaten, Steuermechanismen und Betriebsparameter wie den Batteriestand.
- **Abonnieren von Merkmalen:** Letztendlich abonnieren Sie einen Datenfeed, indem Sie die Bluetooth-Funkgeräte auf ein bestimmtes Bluetooth-Merkmal einstellen.
- **Dekodierung von Daten:** Aufgrund seiner Energieeffizienz erhalten Sie minimale Daten in Rohbits; Sie müssen die Datenpakete selbst dekodieren; der schwierige Teil

ist, dass jeder Sensor von jedem Anbieter seine eigene Art der Datenkodierung hat.

Nehmen wir zum Beispiel einen relativ "einfachen" Sensor wie einen Trittfrequenzsensor, den Sie am Pedal Ihres Fahrrads anbringen, mit dem Sie die Umdrehungen pro Minute bestimmen können, die Sie während einer Fahrradeinheit machen. Wie schwer kann das sein? Ich dachte, die [offizielle Bluetooth Spezifikation](#) würde mir alle notwendigen Details liefern... NEIN! Ein einfacher Trittfrequenzsensor ist überraschend komplex:

- **Kombiniertes Bluetooth-Profil für Fahrradgeschwindigkeit und Trittfrequenz:** Jeder Trittfrequenzsensor identifiziert sich **immer** als *sowohl* Geschwindigkeits- *als auch* Trittfrequenzsensor, Sie müssen seine Fähigkeiten "on the fly" entdecken.
- **Nicht ähnliche Datenpakete:** Der Inhalt der tatsächlichen Daten hängt von den aktuellen Fähigkeiten des Sensors ab (z. B. hat er etwas erkannt), Sie müssen das Datenpaket Bit für Bit interpretieren, indem Sie nach möglichen Flags suchen, die anzeigen, was das nächste Datenbit darstellt.
- **Nicht triviale Daten:** Erwarten Sie keinen eindeutigen Wert für "Trittfrequenz", der Sensor liefert Ihnen eine kumulative Anzahl der Umdrehungen **und** einen rudimentären Zeitstempel der letzten erkannten Änderung; Sie müssen diese Werte verwenden, um die Metrik zu berechnen, die Sie messen möchten.

Die KI lieferte mir Beispielcode, der sich jedoch als nicht funktionsfähig mit meinem speziellen Sensor erwies. Mit einer Bluetooth-Debugging-App, [Bluefy](#), konnte ich ein besseres Verständnis der vom Sensor übertragenen Daten gewinnen. Ich fand ein Stück Java-Code im [offiziellen Bluetooth Blog](#), das mir einige Hinweise zur Interpretation der Daten von den Sensoren gab. Ich passte diesen Code an verschiedene Sensorprofile an und übersetzte ihn in Swift-Code. Dies ergab einen frühzeitig funktionierenden Prototyp.



Früher Prototyp der App, der mit einem Bluetooth-Herzfrequenzsensor arbeitet und dessen frühe Datensätze in einer anderen App (HealthFit) angezeigt werden, die hübsche Grafiken erstellt

UI/UX-Design

Einer der Hauptgründe, warum ich die Apple Watch Workout-App mag, ist, wie ihre Benutzeroberfläche (UI) gestaltet ist, um eine angenehme Benutzererfahrung (UX) zu ermöglichen. Die App auf der Uhr hat ziemlich große Schaltflächen, die man drückt, um eine Aufzeichnung zu starten - *und das war's dann auch schon*. Die App selbst hat wenige Optionen, ihr Design ist eigenwillig, aber gut durchdacht. Sie steht in starkem Kontrast zu Geräten wie Garmin, die eine Menge Optionen zur Anpassung bieten. Der Trick besteht darin, eine gute Balance zwischen Flexibilität bei den Anwendungsfällen *und* einem Standard-Setup zu finden, das gut funktioniert.

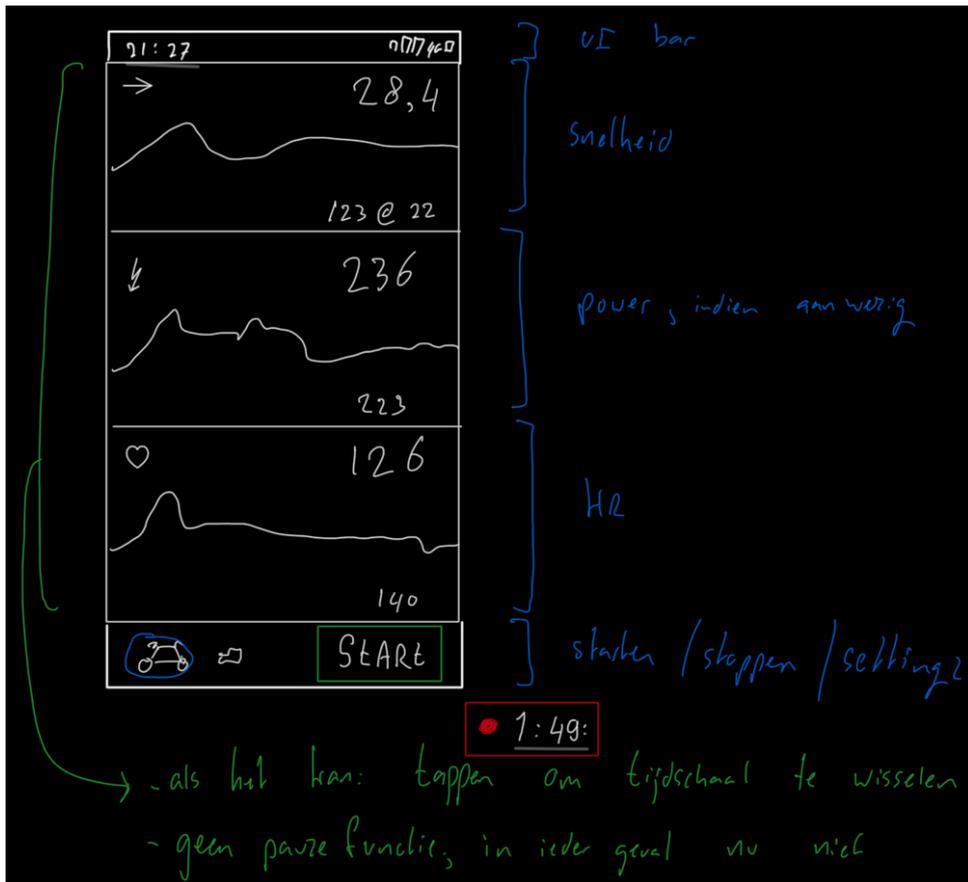


Die Apple Watch Workout-App zeigt große Schaltflächen zum Starten eines Trainings - keine zusätzliche Konfiguration erforderlich

Mit einem Befestigungssystem wie QuadLock können Sie Ihr Smartphone fest am Lenker Ihres Fahrrads befestigen. Das bedeutet, dass das iPhone Ihr Garmin als Fahrradcomputer ersetzen kann. Moderne iPhones verfügen über ein Always-on-Display, das Informationen energieeffizient anzeigen kann, vorausgesetzt, Sie verwenden das Apple Software Development Kit, das verschiedene Optimierungen enthält. Nach vielen, [vielen](#), Kilometern mit meinem Garmin-Fahrradcomputer habe ich mich für eine Ansicht mit Echtzeit-Datenfeldern für Geschwindigkeit, Leistung und Herzfrequenz entschieden. Was ich an den fortschrittlicheren Garmin Edge-Geräten mag, ist, dass sie ein Diagramm darstellen können. So können Sie die Augen auf der Straße behalten, wenn es darauf ankommt, und die historischen Daten zu einem (etwas) späteren Zeitpunkt betrachten.



Ich habe mein Garmin Edge Datenlayout als Inspiration für das Layout meiner App verwendet, wenn sie als montierter Fahrradcomputer verwendet wird

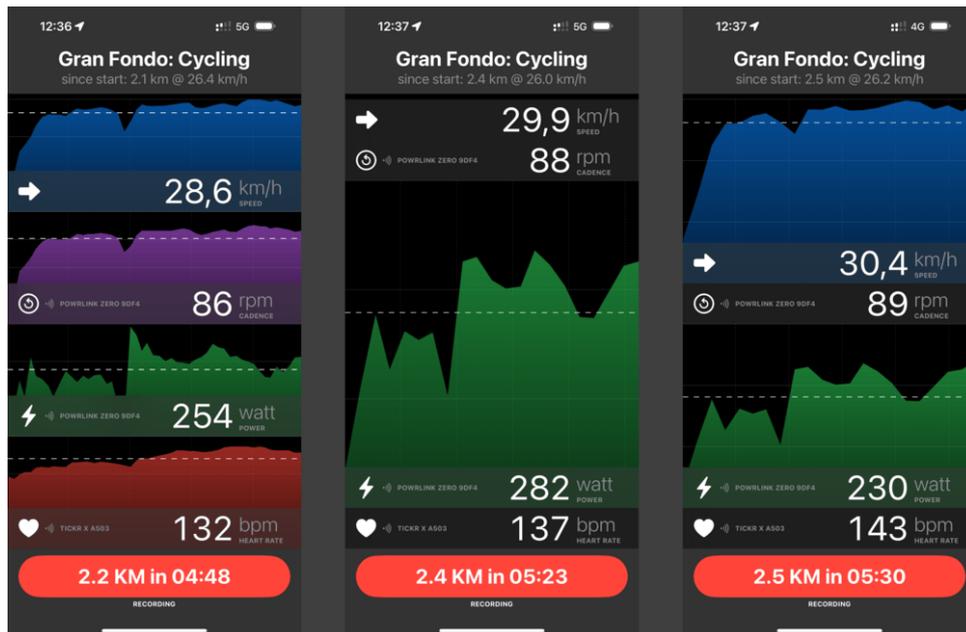


Designskizze, die das gewünschte Layout beschreibt



Testen der App auf meinem Fahrrad mit einem QuadLock Case und einer Halterung

Um die Energieeffizienz zu maximieren, habe ich mich entschieden, die Apple-eigenen Bibliotheken für Diagramme zu verwenden: [Swift Charts](#). Ein Problem, auf das ich bei der Entwicklung stieß, ist, dass Messungen von Sensoren in unterschiedlichen Intervallen und Zeiten eingehen. Man muss eine Art Synchronisationsmechanismus erstellen, damit die Datenpunkte mit gemeinsamen Zeitstempeln übereinstimmen, um Werte vergleichen und in Beziehung setzen zu können. Ich habe diesen Mechanismus so konzipiert, dass auch UI-Updates synchronisiert werden: So können alle Diagramme und Datenpunkte in einer einzigen Aktualisierung aktualisiert werden, wodurch die CPU- und Grafikleistung (und damit der Akkuverbrauch) optimiert wird. Als Bonus habe ich die Unterstützung für verschiedene Zeitbereiche eingebaut, so dass die Diagramme Daten für die letzte Minute, 5 Minuten, 15 Minuten, Stunde oder das gesamte Workout anzeigen können.



Verschiedene Konfigurationen von Echtzeitdaten

Die App zeigt automatisch die verfügbaren Diagramme basierend auf den verbundenen Sensoren an, so dass keine unterschiedlichen Einstellungen erforderlich sind, wenn Sie mehr als ein Fahrrad haben. Wenn Sie auf ein Diagramm tippen, wird es maximiert oder minimiert. Wenn Sie auf den Datenbereich tippen (z. B. "letzte 5 Minuten"), können Sie zwischen den Bereichen wechseln. Die App merkt sich Ihre Konfiguration und richtet Sie automatisch für das nächste Workout korrekt ein. Da alle Touch-Targets und Schaltflächen ziemlich groß sind, funktionieren sie gut, wenn Ihre Hände verschwitzt sind oder Sie Handschuhe tragen.

Taschentauglichkeit: Optimierung für Hintergrundnutzung und Energieverbrauch

Eine alternative Möglichkeit, die App zu verwenden, die ich mir vorgestellt habe, war die Verwendung aus der Tasche heraus, als Hintergrund-App. Dies würde eine unauffällige Datenerfassung ermöglichen, bei der Sie die App starten und Ihr Telefon weglegen. Ich liebe die Idee, einen "vollständigen Datensatz" für *alle* meine Fahrten und Läufe zu haben, ohne dass die App *die ganze Zeit* im Mittelpunkt stehen muss. Viele Trainingsläufe und -fahrten sind nicht *so* interessant.

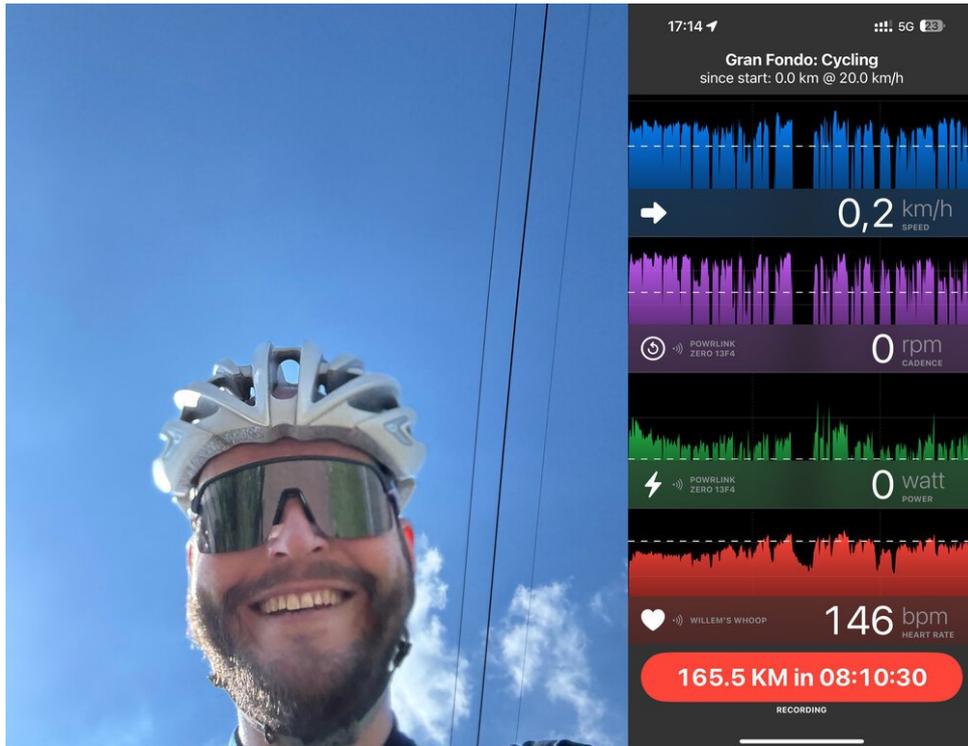


Ich habe Magie in meiner Tasche: Die App trackt Workouts im Hintergrund - während der Aufzeichnung wird nur ein Widget auf dem Homescreen angezeigt

Standardmäßig schränkt Apples iOS-Betriebssystem die Möglichkeiten von Apps, im Hintergrund zu laufen, stark ein. Dies maximiert die Akkulaufzeit und den Datenschutz der Benutzer. Wenn Sie die Richtlinien von Apple befolgen und den Benutzer korrekt um die erforderlichen Berechtigungen bitten, können Sie Ihre App so gestalten, dass sie perfekt im Hintergrund funktioniert. Ich habe die Bemühungen von Apple, den gesamten Stack zu optimieren, wirklich zu schätzen gelernt, da sie es Ihrer App ermöglichen, energieeffizienter zu sein. Einfach ausgedrückt: *"Rufen Sie uns nicht an, wir rufen Sie an"* ist das, was das Betriebssystem tut: Es übernimmt die Kontrolle darüber, *wann* Ihre App ihre Aufgaben erledigt - so kann das Betriebssystem verschiedene Aufgaben auf den verfügbaren (energieeffizienten) Kernen der Apple Silicon-Chips synchronisieren, kombinieren und stapelweise verarbeiten.

Während eines meiner Tests mit einer Prototypversion der App konnte ich über 127 Tausend (!!)

Datenpunkte während einer 8-stündigen Langstreckenfahrt mit einer einzigen iPhone-Akkuladung sammeln, wobei der Bildschirm *die ganze Zeit* eingeschaltet war. Wenn man nach Apples Regeln spielt, hat man wirklich *die Macht* (sozusagen). Die Entwicklung dieser App hat mich wirklich weitergebracht, ha!



”Die Extrameile gehen”: 8 Stunden auf dem Fahrrad, um den Akkuverbrauch zu testen (Anmerkung: es sind noch 23% Akku übrig!)

Wahre Künstler liefern ab

Der Weg zur Hölle ist mit guten Vorsätzen gepflastert, sagt man... Aus diesem Grund beende ich meine Projekte wirklich gerne. Für eine App bedeutet das, dass ich sie im App-Store veröffentlichen sollte. Sie für Apples Prüfung vorzubereiten und von den AppStore-Prüfern genehmigen zu lassen, ist ein zusätzlicher Meilenstein. Er ist wichtig, weil er einen zum Nachdenken über das Onboarding (neuer Benutzer) und das Ertragsmodell anregt.

Onboarding

Eine besonders herausfordernde Aufgabe bei der Entwicklung einer App ist es, Leute, die neu in Ihrer App sind, in der Bedienung anzuleiten. Eine großartige Benutzeroberfläche kann helfen, aber angesichts der tiefen Integration der App mit Apple Health, Standortdaten und Bluetooth-Sensoren ist es zusätzlich erforderlich, höflich um Berechtigungen zu bitten.

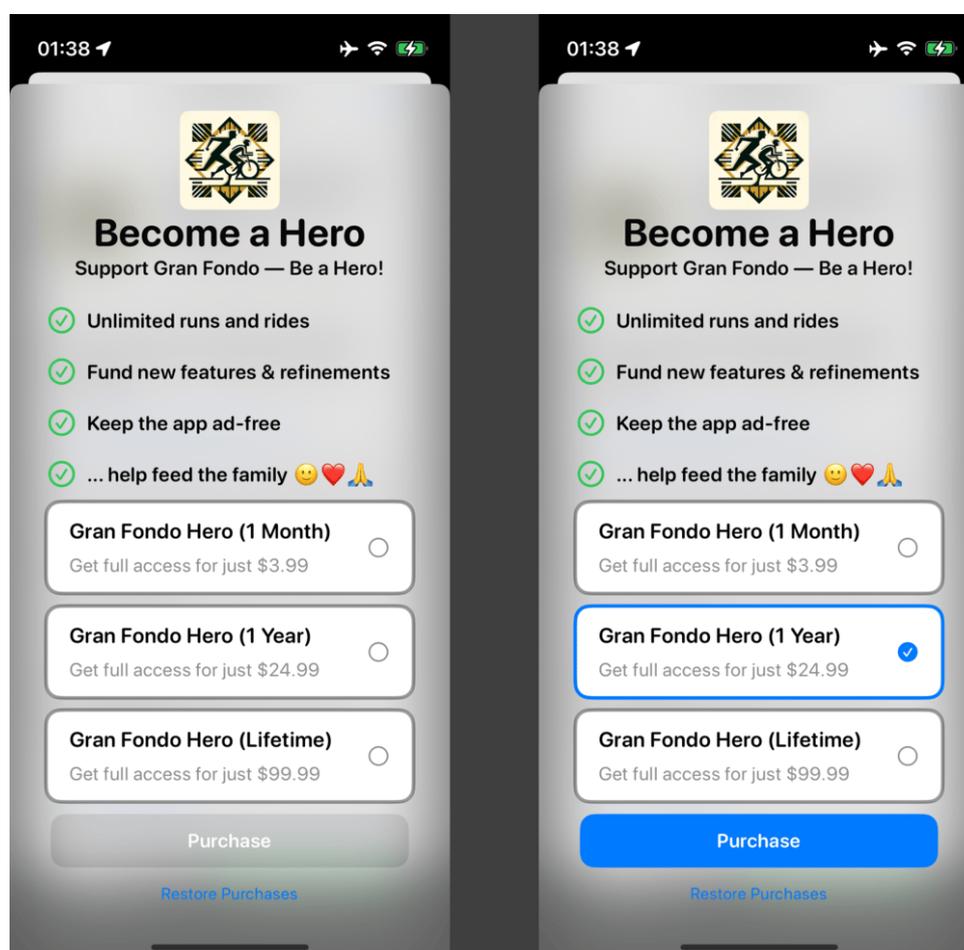


Screenshots des Onboarding-Prozesses der App: Einholen von Berechtigungen und Anleiten neuer Benutzer

Sie müssen den Prozess so gestalten, dass er nicht fehlschlägt, wenn Benutzer bestimmte Berechtigungen ablehnen. Apple ist hier sehr streng: Die Leute haben das Recht, abzulehnen. Es ist Ihre Aufgabe, die App mit keinen oder teilweisen Berechtigungen funktionsfähig zu machen. Das ist eine zusätzliche Herausforderung.

Ertragsmodell zur Finanzierung der Entwicklung

Obwohl ich die Bedeutung von [freier Software](#), im Sinne von libre, nicht hoch genug einschätzen kann, kann ich nicht ignorieren, dass ich eine Familie zu ernähren habe. Aber darüber hinaus kann die Suche nach Leuten, die bereit sind, für Ihr Produkt zu bezahlen, als zusätzliche Validierung Ihrer Idee dienen. Seien Sie offen und ehrlich über Ihr Ertragsmodell, und es wird Ihnen helfen, Ihr eigenes Projekt zu finanzieren; verlassen Sie sich nicht auf Investoren oder Risikokapital, bootstrappen Sie Ihre eigene Entwicklung!



Entwerfen einer Paywall für meine App: nicht meine Lieblingsaufgabe im Umgang mit Benutzern, aber dennoch eine sehr wichtige.

Sie müssen nicht bezahlen, um die App auszuprobieren. Der Zahlungsbildschirm erscheint erst, wenn die Leute die App tatsächlich benutzen (und er kann jederzeit ausgeblendet werden). Warum probieren Sie es nicht selbst aus, [Sie finden die App hier](#).

Zukunft

”Eat your own dogfood”, sagt man - ich werde die App weiterentwickeln, indem ich sie tatsächlich selbst benutze. Ich erwarte nicht, dass dieses Projekt ein schneller (finanzieller) Gewinn sein wird, aber wenn es für mich selbst von praktischem Nutzen sein kann, besteht die Chance, dass auch andere Personen es nützlich finden.



Keine durchschnittliche Debugging-Session: Fietselfstedentocht 2024

Eine Sache, die ich an diesem speziellen Projekt mag, ist, dass es wenig Aufwand erfordert, es am Laufen zu halten, da ich gerne trainiere. Das Debuggen neuer Versionen dieser App beinhaltet oft einen Lauf- oder Radausflug. **Wer hat jemals gesagt, Softwareentwicklung sei ein sitzender Beruf?!**

Fazit

Viele Leute haben ”tolle Ideen” - wenige setzen sie tatsächlich um. Das unterscheidet die Macher vom Rest. Ich mag es, Dinge zu machen, da der Prozess des Machens erfordert, dass man sich wirklich mit der Materie auseinandersetzt. Man lernt eine Menge und sammelt Erfahrungen, die andere nicht haben. Verdammt, bauen Sie es selbst, machen Sie es zu Ihrem eigenen!