Department of Computing

Bachelor of Science (Hons) in Software Development

PaceRunner – Adaptive Runner's Training App

Design Document 2023 – 2024

Student name: Sebastian Firsaev

Student Number: C00263348

Supervisor: Dr Chris Meudec

Date: 30/10/2023

Table of Contents

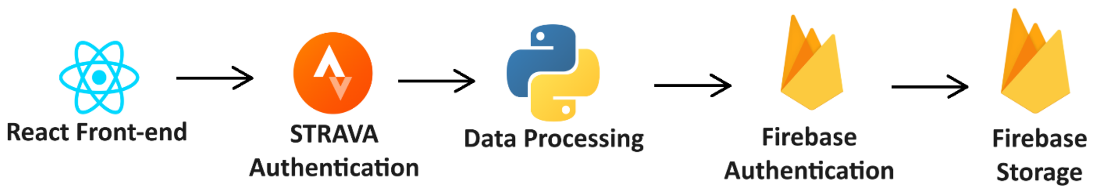
Abstract

This document provides a detailed insight into the design of PaceRunner, a Progressive Web App (PWA) made to enhance running training. It explains how the app works and is used, offering clarity through visual aids such as Entity Relationships, sequence, and high-level design diagrams. These diagrams illustrate how runners interact with the app's various features. This design document will also explain the system architecture, Data model, and database in detail.

Introduction

The PaceRunner app is a dynamic fitness application designed to cater to the needs of runners, specifically improvers that want to maximise their marathon performance. This versatile web application serves as a platform for runners to access customised training programs, monitor their progress, and stay motivated on their marathon journey. Runners can create and update their profiles, track their running statistics, and receive personalised training plans that adapt to their performance. The app's runner-friendly interface, responsive design, and cross-platform compatibility ensure that runners can access it seamlessly from mobile devices. Leveraging the latest web technologies, such as React framework, HTML, CSS, and JavaScript, the PaceRunner app aims to improve the running training experience.

System Architecture



Sequence Diagrams

Data

Integration with STRAVA

Harnessing data from Strava to influence the training program offers a host of possibilities to enhance a runner's fitness journey. PaceRunner's approach is to analyse a runner's historical data, including metrics like pace, heart rate, distance, and training consistency. By examining trends and performance patterns, the training program can be tailored to address specific weaknesses and capitalise on strengths. For instance, if a runner consistently struggles with pacing, the application can reduce the pace for the next run and adjust the rest of the training plan accordingly. Heart rate data can be used to establish optimal training zones, ensuring workouts are both challenging and safe. Additionally, tracking adherence to the training plan will trigger suggestions for improvement such as "reduce pace!", motivational messages and adjustments to keep runners engaged and committed. leveraging Strava data empowers the training program to evolve dynamically, addressing individual needs and goals, ultimately leading to more effective and satisfying training experiences.

Database

PaceRunner uses firebase as it's real time database. Firebase is a powerful platform that revolutionises the way data is handled in applications. The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronised in realtime to every connected client [1].What makes Firebase truly exceptional is its capacity to function seamlessly offline, thanks to the data persistence mechanism within the Firebase Realtime Database SDK. In situations where connectivity is temporarily lost, the client device retains and synchronises any changes upon reconnection, effortlessly merging any conflicts that may arise. Furthermore, Firebase offers accessibility directly from client devices or web browsers, eliminating the necessity for a dedicated application server. Security and data validation are achieved through the innovative Firebase Realtime Database Security Rules, which govern data access based on expression-based criteria [1]. This distinctive NoSQL database offers immense optimization and functionality, making it an ideal choice for delivering real-time, high-performance applications that prioritise runner experience.

Data extraction

Runner Authentication: PaceRunner will implement Strava OAuth 2.0 authentication in its React frontend to allow runners to connect their Strava accounts securely.

Access Tokens: PaceRunner will request and obtain access tokens from Strava's API for authorised access to runner data.

API Endpoint Calls: PaceRunner's Python backend, will use these access tokens to make authorised API calls to Strava's endpoints. Retrieve data such as activities, heart rate, pace, and distance.

Data Transformation: PaceRunner will process the retrieved Strava data in Python to ensure it aligns with the data structure expected by Firebase. Convert and structure the data accordingly.

Firebase Integration: PaceRunner will use Firebase SDK in your React frontend to store the processed Strava data in the Firebase Realtime Database to ensure that data is synchronised in real-time.

Data Utilisation for smart training adjustments

To utilise the data and perform smart adjustments runners will have the option to provide their age and if known their resting heart rate (HRrest) and max heart rate (MHR). If the runner doesn't conduct specific tests to determine their MHR or undertake any analysis for HRrest, the app will use historical data from previous runs and age to create approximate heart rate and pace zones.

Estimating Target Heart Rate:

Proxy for MHR:

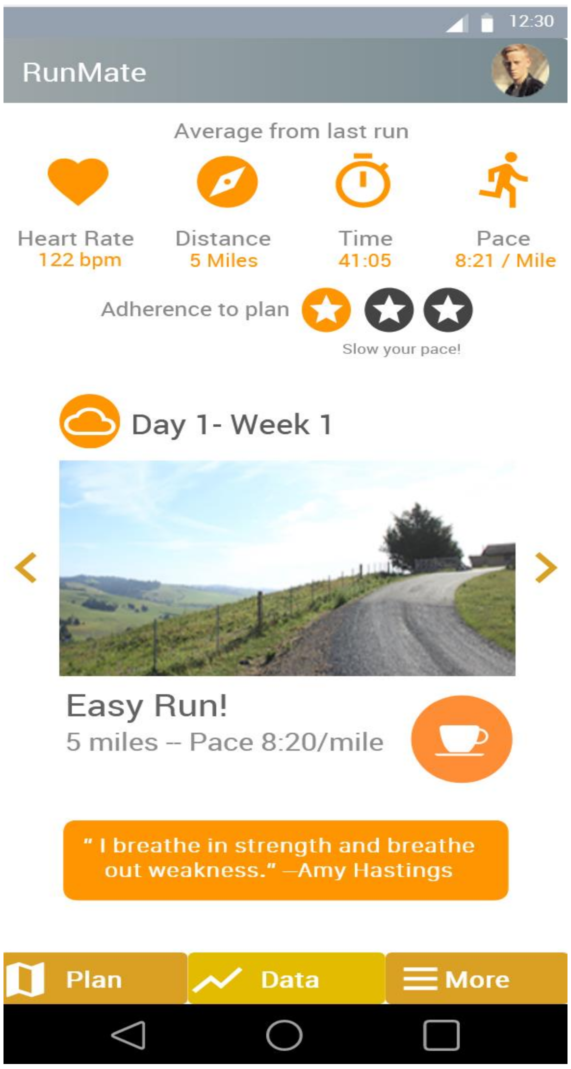
Without specific data the app will use an age-based estimate: The Karvonen Formula. This method of calculating your target training zone is based on your maximal heart rate and resting pulse [2]. The formula is Target Heart Rate (THR)=((MHR−HRrest) x Intensity %)+HRrest. Example, taking a 50-year-old runner, for MHR, subtract their age from 220, resulting in 170 beats per minute (bpm). To get the upper limit of exercise intensity, around 75 percent of MHR, multiply the MHR (170 bpm) by 0.75. This yields an approximate heart rate of 128 bpm.

Estimating target Pace:

Recent run times for various distances will be used to estimate pace zones for different types of runs (easy, tempo, long runs). For training adjustments PaceRunner can estimate the runner's VO2 max and create personalized pace recommendations. For example, VO2 Max can be roughly estimated from a runner's heart rate (HR) using the formula VO2 max = 15 x (HRmax ÷ HRrest)[3]. Once the estimated VO2 max is determined, the data will be used to adjust T-pace (Threshold) running. The proper pace for T-pace running is about 83 to 88 percent of VO2 Max, or 88 to 92 percent of vVO2 Max or maximum heart rate [4].

UI Design

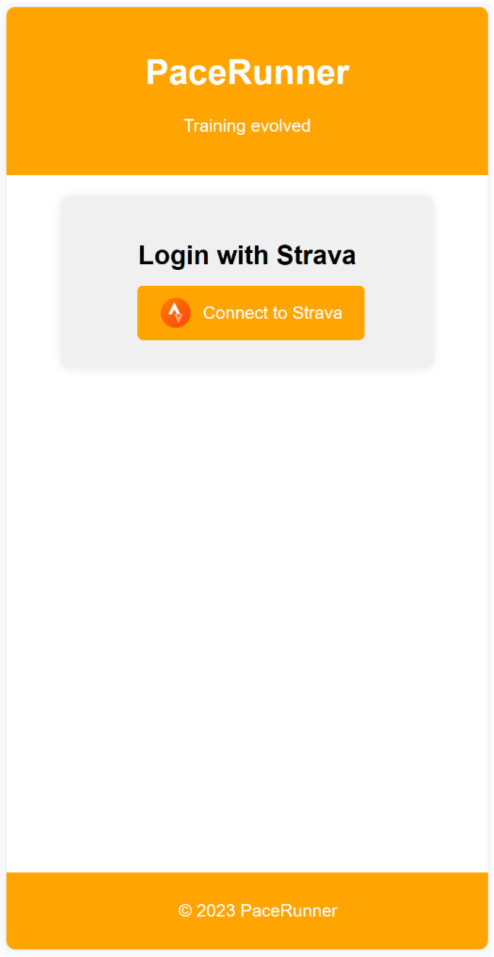
The following are sample UI designs for PaceRunner, showcasing what the app will look like on a mobile device.



PaceRunner's home page. This is the first page the runner will be after they login to the app. It displays the most important information the runner will care about from their previous run, including heart rate, distance, time, and pace. The runner will also be able to see if they are adhering to their training plan using a three-star system. For example, the sample screenshot shows one star, indicating that the runner didn't follow the plan as they ran too fast.

From this page the runner is also able to see their next training session, they can check the weather via the cloud icon and force a break in the training via the coffee cup icon. The runner can also read an inspiring message that changes with every login.

PaceRunner's login screen uses a minimalist and simple design to allow super quick access and set up. Runners will only need to press one button "connect to Strava" to start.



Displayed in mobile form using Microsoft Edge's developer tools.

Technologies

PaceRunner uses React for its frontend. React is a JavaScript library for building runner interfaces, developed and maintained by Facebook. It enables the creation of interactive and dynamic web applications by allowing developers to efficiently manage and update the runner interface as data changes, ensuring a seamless and responsive runner experience.

Python is a versatile programming language widely employed for data processing and analysis. Its extensive ecosystem of libraries, such as NumPy, Pandas, and Matplotlib, simplifies data manipulation, statistical analysis, and visualization. Python's readability and ease of use make it an excellent choice PaceRunner's data processing needs, facilitating the extraction of insights from complex datasets and the development of data-driven custom training programs.

Flask is a lightweight and versatile web framework for Python. It simplifies the process of building web applications by providing the essential tools and components, allowing developers to create web-based solutions with minimal overhead. Flask is known for its simplicity, making it an excellent choice for medium web projects and rapid development.

Conclusion

In conclusion, this design document provides an extensive overview of PaceRunner, an innovative Progressive Web App (PWA) engineered to elevate running training. The document clarifies the app's functionality, offering lucidity through visual aids, including Entity Relationships, sequence diagrams, and high-level design schematics, which illustrate runner interactions with various features. PaceRunner, a dynamic fitness application, caters to runners' diverse needs, enabling them to access customised training programs, track progress, and maintain motivation on their marathon journey. Its responsive design and cross-platform compatibility ensure easy access from mobile devices. By leveraging cutting-edge web technologies like React, HTML, CSS, and JavaScript, the PaceRunner app aspires to enhance the running training experience.

References

[1] Google. (2023). Firebase Realtime Database Documentation. [Online]. Available at: https://firebase.google.com/docs/database (Accessed: 07/10/23).

[2] Das, P. What is the Karvonen Formula? [Online]. Available at: https://www.physiotherapy-treatment.com/karvonen-formula.html (Accessed: 01/12/23)

[3] MACKENZIE, B. (2001) VO2 max [Online] Available at: https://www.brianmac.co.uk/VO2max.htm (Accessed 17/11/2023)

[4] Daniels, J., Ph.D. (2022). What You Need to Know About Threshold Training. [Online]. Available at: https://www.runnersworld.com/advanced/a20807282/threshold-training/ (Accessed: 01/12/23).

[5] Herbert, D. (2023). React.js: What It Is and How It Works. HubSpot Blog. Available at: https://blog.hubspot.com/website/react-js (Accessed:15/11/23).