Segment Effort Scoring System – Documentation

# 📌 Purpose

This system is designed to normalize and evaluate running efforts across segments, enabling objective comparison between them, even when the segments differ vastly in:

- Distance (e.g. 100m vs 5km),  
- Elevation gradient (e.g. uphill vs downhill),  
- Athlete performance (actual times).

By computing a normalized performance score, this model identifies which segments the athlete can improve with the least effort or where they have already performed at a high level.

# 📊 How It Works

Each segment is described by:

- `nonFlatDistanceMeters` — actual distance run  
- `averageGradePercent` — average gradient (%)  
- `userPersonalRecordSeconds` — athlete's best time on the segment

# ✅ Step 1: Grade Adjustment

To remove the bias caused by uphill or downhill effort, the model computes a grade-adjusted flat equivalent distance using empirically-derived effort factors for gradients between -25% and +25%.

This is based on research such as:

Minetti et al. (2002), "Energy cost of running" – showing how running efficiency changes on slopes.

The adjustment ensures a fairer representation of effort for all segments, flattening them out mathematically.

# ✅ Step 2: Normalize Effort to a Common Reference (300m flat)

The adjusted effort is now scaled to what the equivalent pace (in min/km) would be if run as a 300m flat segment.

This step uses a distance-performance model based on a reference table of elite paces:

| Distance | Reference Pace (min/km) |  
|----------|--------------------------|  
| 50m | 1.7 |  
| 100m | 2.0 |  
| 300m | 2.4 |  
| 1km | 3.0 |  
| 5km | 3.5 |  
| 42km | 4.5 |

These paces are interpolated linearly and represent equivalent levels of performance at each distance. This mirrors how athletes' pace declines with increasing distance due to endurance limits.

# ✅ Step 3: Score Calculation

The score is scaled so that:  
- A perfect performance (3:00/km pace on 300m flat) → Score = 100  
- Slower efforts yield proportionally lower scores  
- Very poor efforts (e.g., walking) may score < 30

The formula:

SCORE = constant / normalizedPace

…ensures that better pace results in a higher score.

# 🧠 Design Rationale

Based On:  
- Real-world physiology: Effort vs gradient from treadmill studies  
- Empirical pacing data: Elite athlete performance across standard race distances  
- Normalization: Inspired by concepts in cycling such as "Normalized Power" and in running like Strava's GAP (Grade Adjusted Pace)

# 📐 Differences vs Similar Systems

| Feature | This System | Strava GAP / Fitness Score |  
|----------------------------------|----------------------------------|--------------------------------|  
| Grade adjustment | ✔ Empirical factors | ✔ Similar |  
| Distance normalization | ✔ Via elite pacing model | ❌ Not normalized |  
| Comparison across segment types | ✔ Direct | ❌ Difficult |  
| Score intuitive scale (0–100+) | ✔ Yes | ❌ Varies by context |  
| Open & inspectable | ✔ Fully transparent | ❌ Proprietary logic |

# 🔎 Significance

This system allows athletes to:  
- Compare short vs long segments fairly  
- Identify underperforming segments even if they look fast on paper  
- Track effort-based progression rather than time-based alone  
- Plan training or segment targeting intelligently

# 🧾 Credits

This model is inspired by and adapts ideas from:  
- Scientific research on human running metabolism and slope energy costs (e.g., Minetti et al., 2002)  
- Elite pacing benchmarks from world record data  
- Normalization and comparison ideas from:  
 - Cycling: Functional Threshold Power (FTP), Normalized Power  
 - Running: Strava GAP, WKO Running Power Models

# 📦 Class Summary

Score.java  
- getScore(SegmentDTO) — main entry point  
- Computes score based on grade-adjusted, distance-normalized pace

DistancePaceNormalizer.java  
- Uses elite pace data to estimate relative effort per distance  
- Supports normalization of pace across any distances

GradeAdjustmentModel.java  
- Provides adjustment factors for running on inclines or declines  
- Interpolates between known values for gradient-effort relationships