Technology Choice Document

Movey – Sports Community app

By Rareș Nagy

Table of Contents

[Introduction 2](#_Toc210220807)

[Method 2](#_Toc210220808)

[Frontend 3](#_Toc210220809)

[JavaScript Framework 3](#_Toc210220810)

[Backend 3](#_Toc210220811)

[Database 3](#_Toc210220812)

# Introduction

This is a documentation of the technology choice for the second project. It shows my thought process and reasons for choosing the application’s frontend, backend, database. These choices were made with a mixture of clear and measurable indicators, such as performance and ease of use, and personal preference and experience.

# Method

I’ve used decision matrices for these choices, with some exceptions, where the choice didn’t make much of a difference, or I just had enough experience with it that it just made more sense (e.g. Backend). This a clear and simple way to make a technology choice (or any choice really). A **decision matrix** is a list of values in rows and columns that allows an analyst to systematically identify, analyze, and rate the performance of relationships between sets of values and information. Elements of a decision matrix show decisions based on certain decision criteria. The matrix is useful for looking at large masses of decision factors and assessing each factor's relative significance by weighting them by importance.[[](https://en.wikipedia.org/wiki/Decision_matrix" \l "cite_note-1)(Wikipedia Authors, n.d.)

# Frontend

This is the only part that the client will ever see, so the choices made for it are bound to have a big impact. For this I need to choose a JavaScript framework (or decide not to use any framework) and a way to style the webpage.

## JavaScript Framework

The market of Frontend JS frameworks is vast, it feels like there’s a new one every day. Since my project will need to integrate the frontend with the backend, the choice of what JavaScript framework will be used (if any) is an important one. I used [this video by Fireship](https://youtu.be/cuHDQhDhvPE) and my own personal experience to create a decision matrix and find out the answer.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Criteria/Framework | Familiarity | Ease of use | Declarative | Overkill | State management | Popularity | Total |
| Vanilla | 7 | 3 | 2 | 0 | 2 | 9 | 238 |
| React | 3 | 2 | 6 | 3 | 8 | 10 | 218 |
| Svelte | 8 | 9 | 10 | 2 | 10 | 5 | 380 |
| Vue | 7 | 8 | 10 | 2 | 9 | 7 | 365 |
| Importance Multiplier | 8 | 9 | 10 | 10 | 4 | 3 | N/A |

I chose Svelte, not just because it scored the best in the table, but also because I had a bias in choosing it. I just like using it.

## Styling

Since I don’t plan

# Backend

Because of the applications relatively small scope, the backend choice is not that impactful. Speed is not a huge worry either, so it doesn’t need to be fast necessarily. I chose the **Go** programming language, as I’ve used it to build backends before, and it has great support for REST APIs. Its great performance is more of a bonus than an advantage for this project.

# Database

A database is essential for this project because it securely stores user profiles, posts, and live interactions, ensuring everything loads fast and stays consistent across millions of users in real time.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Weight (1–5)** | **PostgreSQL** | **MySQL** | **MariaDB** | **SQLite** |
| **Scalability (High Read/Write Load, Sharding/Clustering Support)** | 5 | 3 | 3 | 3 | 1 |
| **Real-Time Feed Support (Pub/Sub, Triggers, Notifications)** | 5 | 2 (LISTEN/NOTIFY) | 2 | 2 | 1 |
| **Full-Text Search & Indexing Performance** | 4 | 5 | 3 | 4 | 1 |
| **Consistency & Transaction Reliability (Followers, Likes, Wallets)** | 4 | 5 | 4 | 4 | 3 |
| **Ease of Horizontal Scaling / Replication Options** | 4 | 3 | 3 | 4 (Galera Cluster) | 1 |
| **Setup & Operational Cost / Complexity** | 3 | 3 | 3 | 3 | 5 |
| **Developer Familiarity & Ecosystem Support** | 3 | 5 | 5 | 4 | 4 |

Final Scores

|  |  |
| --- | --- |
| **Database** | **Total Score** |
| **PostgreSQL** | **116** |
| **MariaDB** | **106** |
| **MySQL** | **98** |
| **SQLite** | **60** |

As such, Postgres will be used for this project