ADR

**Title:**

Relational or Non-relational Database

**Context + Problem Statement:**

We need to decide whether the database should be relational or non-relational, making sure the architecture and application are appropriate to the requirements of the project.

**Decision Driver:**

* To ensure the architecture is consistent throughout the documentation.
* To ensure the best stack is selected for the application and that it allows the features to be implemented.

**Option:**

Relational database. Non-relational database.

**Considered Options:**

* Relational database: A relational database is organized using a structured format of rows and columns within a defined schema, relying on SQL (Structured Query Language) for its operations. It maintains data accuracy and consistency while establishing predefined relationships between tables (insight & software, 2023).
* Non-relation: A non-relational database or NoSQL database which is designed to handle scalable applications and allows to handle large amounts of data. It is also beneficial for the database when storing the data that doesn’t have a predefined schema (insight & software, 2023).

**Decision Outcomes:**

Chosen option: non-relational database.

This is mainly because we want to use Firebase, which is a non-relational database, due to it having services that would help with certain features, such as realtime database and authentication.

It's also a valid choice due to the time constraints and the data for this application is not too dependent on data integration.

**Consequences:**

Good, because operations tend to be faster, such as reads/writes/

Good, because it's suitable for storing diverse data formats, such as images, videos, documents, JSON files, etc.

Good, because this application does not need to focus on data integration.

Good, because we are thinking about using Firebase due to specific services.

Good, because it’s flexible and can change to requirements easily.

Bad, because it lacks advanced querying capabilities like joins and complex filtering found in SQL.

Bad, because a lack of strict structure may lead to inconsistent data.

Bad, because it may underperform outside specific use cases.

Bad, because it is not designed to scale out the applications by itself and requires using the NoSQL management system such as BigTable or MongoDB replica sets. (Adservio, 2021).

**Confirmation:**

Before deciding on the stack and architecture design.

**Pros and Cons:**

Pros:

* Faster and handles diverse data formats.
* Doesn’t need to focus on data integration.
* Can use Firebase.
* Flexible and can adjust to changes in requirements.

Cons:

* Lacks advanced querying capabilities.
* Can lead to inconsistent data.
* May underperform outside specific use cases.
* Cannot scale out independently and relies on NoSQL management system.

# References

insight, & software . (2023, May 15). *What’s the Difference? Relational vs Non-Relational*

*Databases*. Insightsoftware.

<https://insightsoftware.com/blog/whats-the-difference-relational-vs-non-relational-databases/>

Adservio. (2021, March 15). *What are the Pros and Cons of NoSQL*. Www.adservio.fr.

<https://www.adservio.fr/post/what-are-the-pros-and-cons-of-nosql>