ADR – Authentication for Profile Management

**(written independently by Kim Nyaguse)**

**Title:**

Authentication/Validation for Profile Management

**Context + Problem Statement:**

The application users need to be able to update their profile information. The system should validate this data and show this through Firebase, and they should be able to change this information multiple times which is why security and real-time updates are essential

**Decision Driver:**

* Validation and authentication mechanisms need to be reliable and secure for users.
* Real-time updates need to be able to be reflected immediately.
* Integration with the preexisting Firebase backend is made a lot simpler and more efficient.

**Option:**

* Firebase Authentication with Node.js
* Custom Authentication with Node.js
* Third-party OAuth providers (Facebook, Google)

**Considered Options:**

* Firebase Authentication with Node.js
* Custom Authentication with Node.js

**Decision Outcomes:**

Chosen Option: “Firebase authentication” because its built-in features are easy to use and there are many resources available. The Firebase Authentication can also be easily integrated with the Firebase Database due to it being built in.

**Consequences:**

* Good because Firebase Authentication allows for users to be secure, and it is a consistent experience.
* Good because the real-time database allows users to have immediate feedback and updates.
* Bad because it has very limited flexibility. Authentication cannot be easily customised due to protections in place by the creators.

**Confirmation:**

Unit tests will be able to verify the validation of unit input. Integration tests can be done to ensure that profile changes are reflected correctly in the Firebase database. Security audits can validate the data protection measures in place.

**Pros and Cons:**

Custom Authentication with Node.js

* Good because the developer has full control over authentication mechanisms, increasing their autonomy.
* Neutral because there is the flexibility to customise the authentication based on the developers' specific requirements.
* Bad because the development time is drastically increased as well as the complexity.
* Bad because there is an increased responsibility for ensuring security and complience with GDPR guidelines.

Template used:

<https://adr.github.io/madr/decisions/adr-template.html>