

# FishSticks

## Architectural Report

**Team:** Minciunescu Tudor-Andrei

Podariu Maria Flavia

Sofei Andrei Adrian

Mirea Oana Gabriela

### ➤ Identify the main quality attributes of your application

The purpose of the app is to make group decisions in a fun way. In order to accomplish this, the app needs the following:

**Ease of use** – only lobby owners need to create an account, everybody else just needs a lobby code, to keep the app intuitive and fun to use

**Variety** – lots of games to choose from

**Scalability** – the app is created in a way that makes it very easy to add new games

### ➤ Reason why those are the most important attributes

The attributes above are very important for the app, because of its purpose.

Because such an app is well suited for making quick decisions, it has to be very easy to use, otherwise, the users might feel like this is not an efficient way of solving their problem.

In a group of people there are usually lots of personalities/ preferences, so more games to choose from will make the experience better for more users.

### ➤ Design Patterns used in the application -> + Benefits and Tradeoffs

**REST architecture:**

**-Benefits:** allows client and server to be implemented independently, stateless

- Tradeoffs: harder to implement (compared to using a backend as a service like Firebase)

### Functional Components

- Benefits: easier to write, re-use
- Tradeoffs – harder to maintain

### Conditional Rendering

- Benefits: easy to write,
- Tradeoffs – hard to read when complexity grows

### React Hooks

- Benefits: easier to write, decoupling logic from UI
- Tradeoffs: can introduce unexpected behaviors

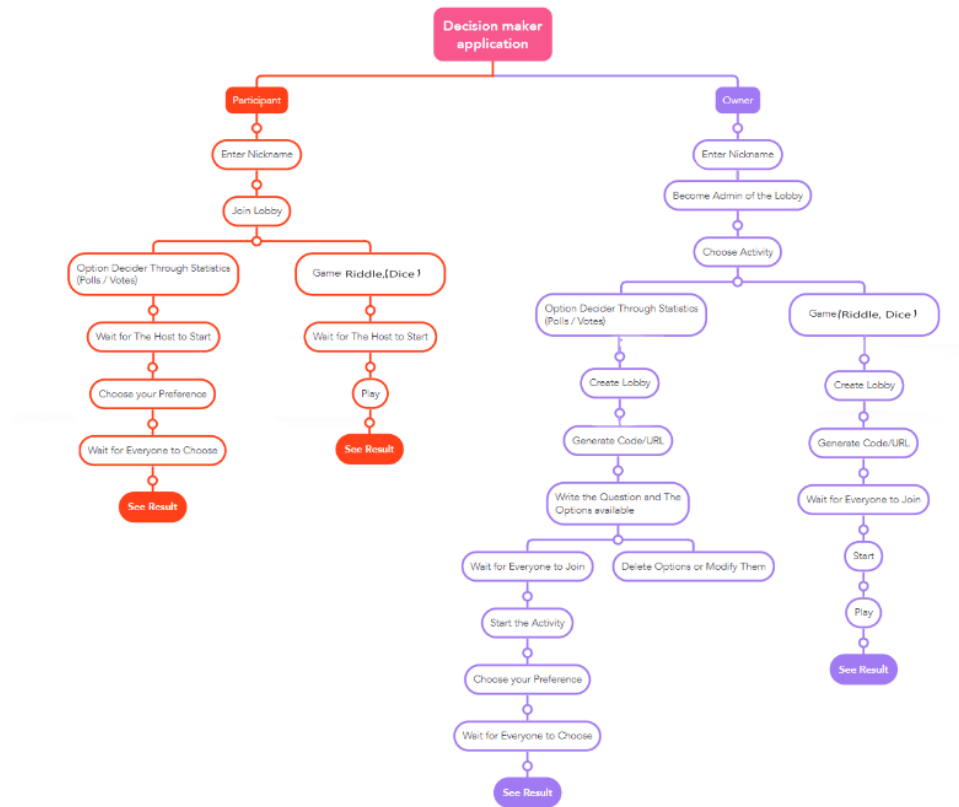
### Next.js routing system:

- Benefits: easy to create new routes and switch between them
- Tradeoffs: hard to maintain as complexity grows

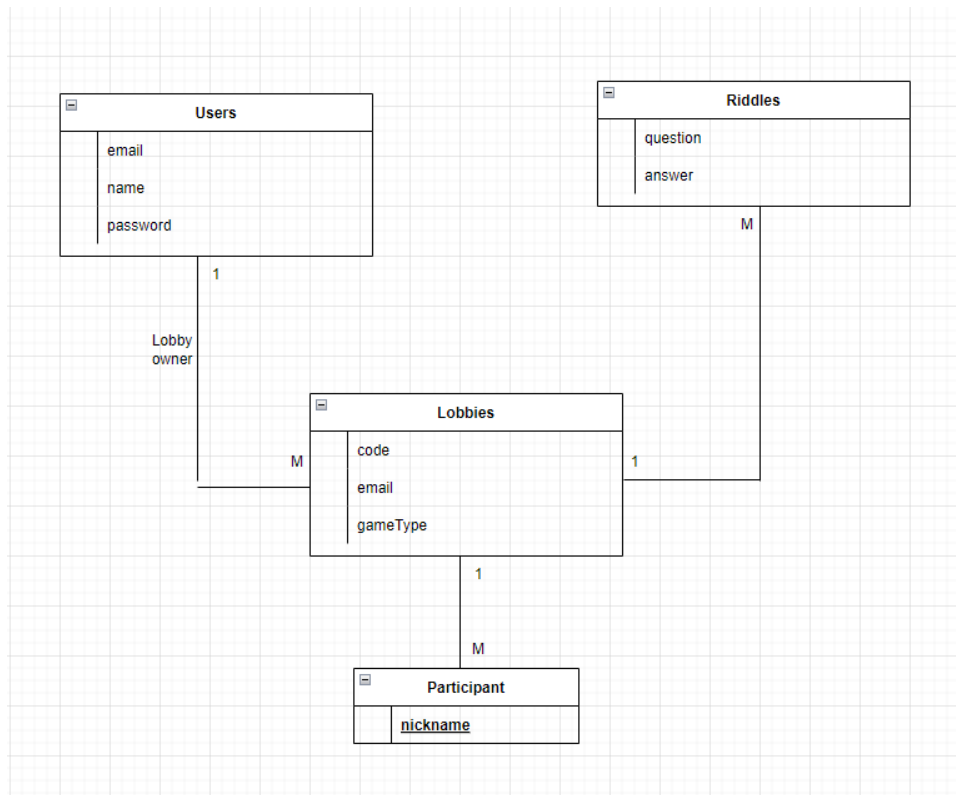
### ➤ Used technologies

- React
- NextJS
- Python
- Flask
- Socket
- Firestore

### ➤ Functional decomposition diagram



## ➤ Tables diagram



## ➤ Activity diagrams

