Time for other Stuff

Kanban Board

FS2022

Dominik Ehrle
Marco Agostini
Mathias Lenz

Computer Science
University of Applied Sciences of Eastern Switzerland

Contents

1 Vision

We envision to create a real time Kanban board to empower people to collaborate efficiently. The user can create Kanban boards and collaborate with invited people in real time on the board.
Our board differentiates itself with the additional analytical features the user can access for free.

1.1 Technology Stack

We build our application with the following technologies: The frontend will be built with React and the backend will be implemented in ExpressJS. We will use MongoDB as the database to guarantee persistent storage for our application.

2 Requirements

2.1 Functional

- The application should allow anyone to create Kanban boards
- It should allow to customize the board and the buckets
- It should allow to create items
- The items should be movable between buckets
- It should synchronize instantly to enable collaboration with other people
- It should reconnect automatically in case of a server disconnection

2.2 Non-Functional

ID	Category	Priority	Requirement	Measure
NFR-1	Time-behaviour	Medium	The system reacts in appropriate time	The system reacts to interaction in less than 5 seconds
NFR-2	Availability	High	The system is distributed and implements a Load Balancer	The system is stable against server failure
NFR-3	Data Integrity	High	Changes to data on a board are shared almost instant	The system shows changes in under a second
NFR-4	Capacity	High	The system can handle multiple simultaneous users	The system is able to handle average load of at least 10 simultaneous users
NFR-5	Usability	Medium	Creation of a board or item in reasonable time	Creation of board or item in less than 5 seconds

3 Analysis

3.1 Domain Model

The domain model is relatively simple:

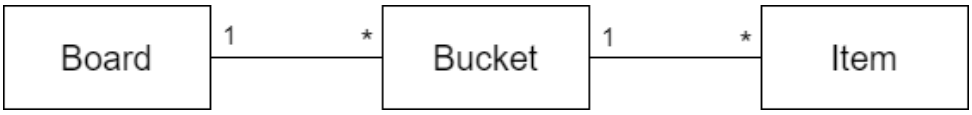

Figure 1: Domain Model

3.2 Use Cases

The use cases define the minimum viable product (MVP) of the application.

Use Case List

- User can create a new Kanban board
- User can give the board a name
- User can see an overview of the Kanban board
- User can create buckets on the board
- User can change the name of a bucket
- User can delete buckets
- User can share the link for their board with other people
- User can create new items inside a bucket
- User can drag and drop items between buckets
- User can reorder items in a bucket via drag and drop
- User can edit items
- User can delete items
- User can immediately see the changes of other users on the board

Possible Expansions

- User can create tags for the board
- User can edit tags of the board
- User can assign tags to items
- User can filter based on tags
- User can assign tasks to a specific user
- User can view metrics of the Kanban board
- User can close/delete a board
- User can create a read-only, shareable link to the board
- User can password protect a board
- User can view board collection

4 Architecture

4.1 Technologies

- **Frontend:** React - A JavaScript library for building user interfaces
- **Backend:** Node.js with Express.js as webframework
- **Websocket:** ws - a Node.js WebSocket library
- **Load Balancer:** Nginx
- **Database:** MongoDB (NoSQL) with Mongoose as a MongoDB object modeling tool
- **Dockerized**

4.2 Application Architecture

The application is based on a client server system with a client-side rendered frontend and websocket as a means of synchronization with the backend server.

Figure 2: Application Architecutre

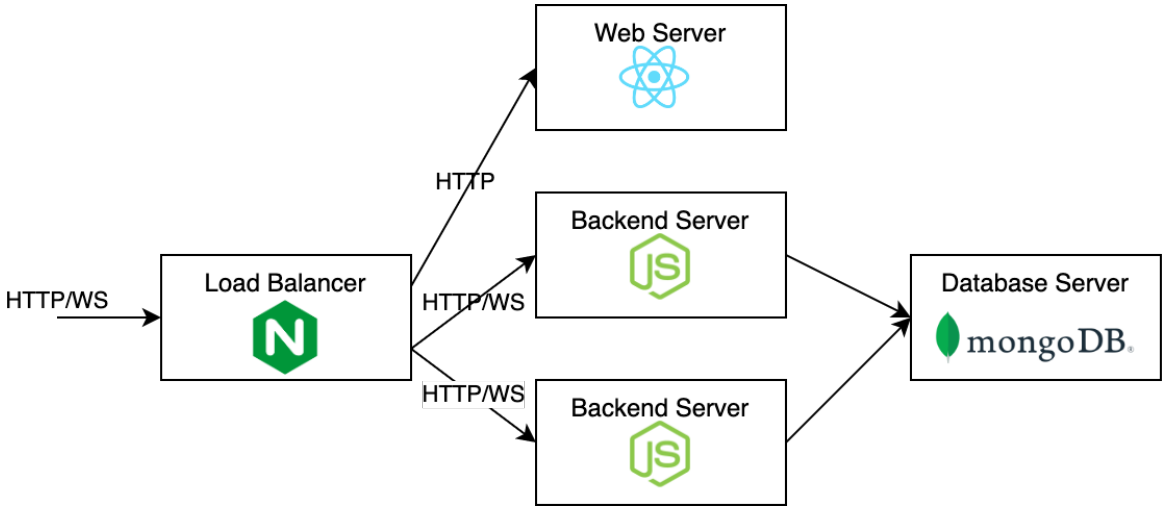

Figure 3: Deployment Architecture

4.3 Sequence Diagram

To synchronize the board across several users, we developed our own protocol for websocket. It features a get message to get the initial state of the board and an update message that allows to change the board.

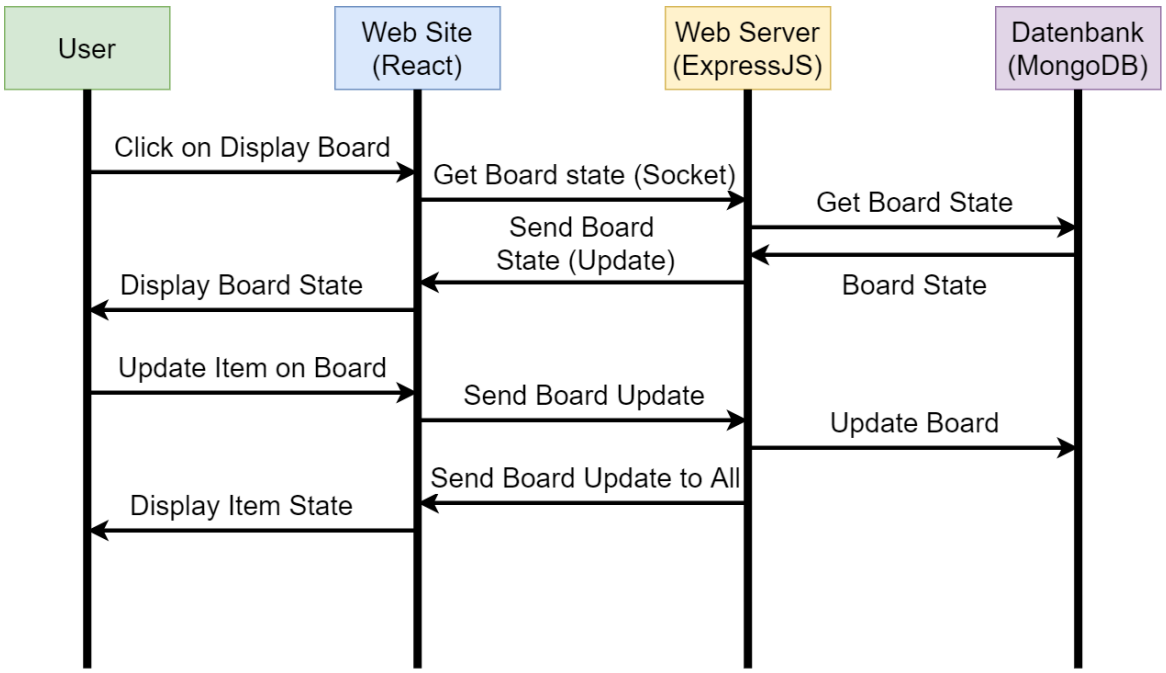

Figure 4: Sequence Diagram

5 Conclusion

While we didn't meet our initial vision completely, we are happy with the result. We were able to create the MVP of our application based on the use case list and works seamlessly even in case of a server outage - and it also looks aesthetically very satisfactory.
The main hurdles for us were the load balancer and the automatic reconnect of the frontend.
We think the application has a lot of potential and there are a lot of features that could be added to the board to increase its usefulness.

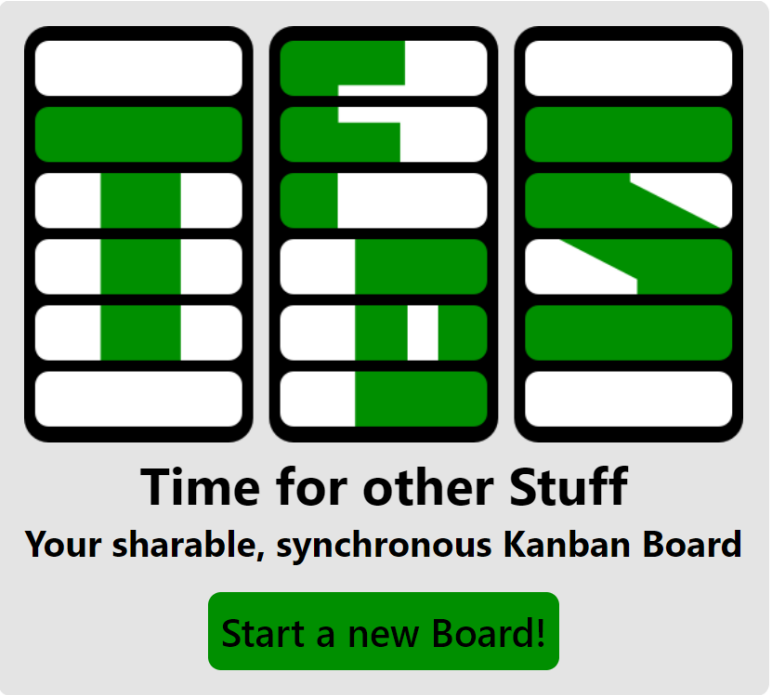

Figure 5: Application Index Page