Final Report

Login-Free Message Board (MeBo)

1. Introduction

Summary

The Login-Free Message Board (MeBo) is a web application which enables users to create their own message board, share a link with other users and write message in a collaborative way. Every user can post new messages or edit existing messages - although if they were written by other users.

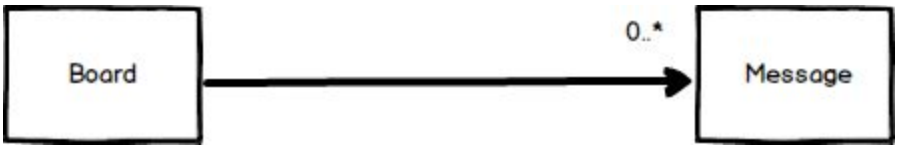
Features

- Users can create boards (without registration or login)
- Boards can be shared by their unique link
- Users can write messages to the board
- Every user can edit, delete or vote on every message
- Messages can be searched and ordered

2. Design and Implementation

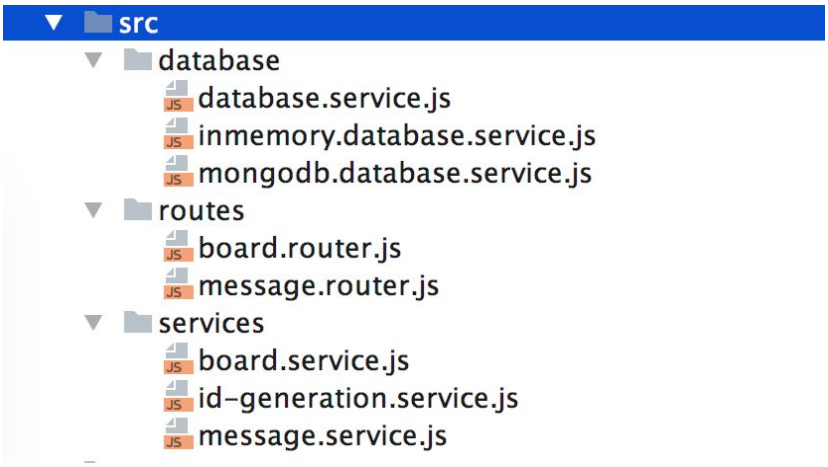
System Design

The system is design around the data model of boards and messages. The system contains multiple boards which can be created by the users. Each board contains multiple messages. A message can be edited or deleted by anyone who has access to the board.



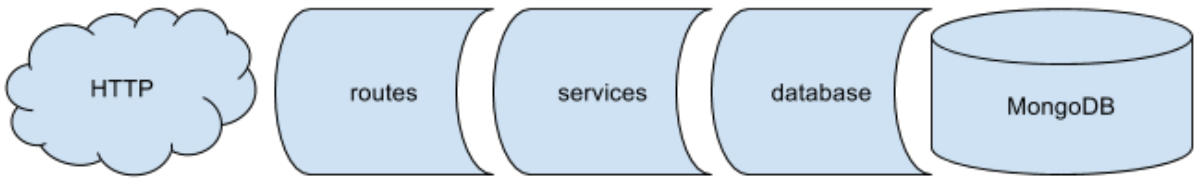
Implementation

Backend

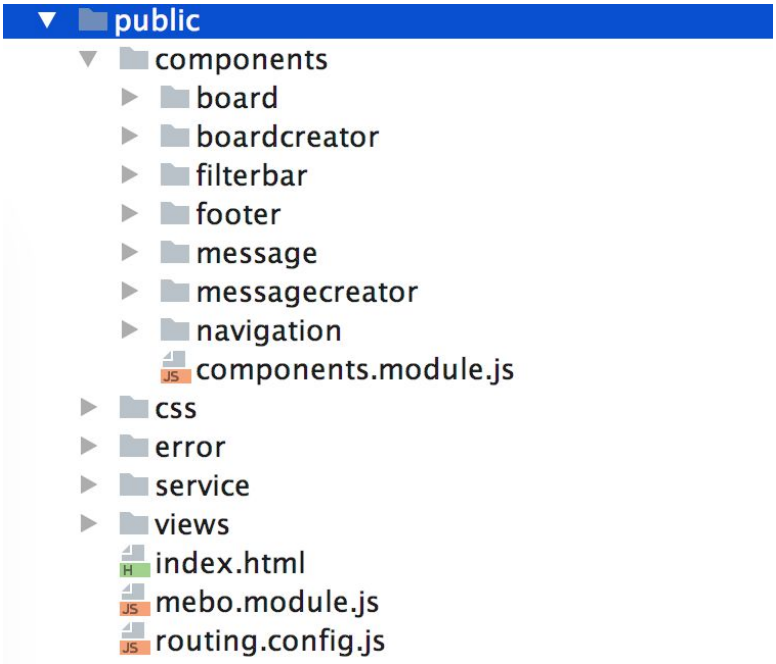


The backend is organized around 3 main packages:

- "**database**" contains the storage layer, especially all interactions with MongoDB. Only this package know how the data is persisted. It encapsulates all storage operations. It also contains a mock implementation which runs without MongoDB but stores all data in-memory.
- "**routes**" contains all REST endpoints. Those endpoints define the URLs and HTTP methods to interact with the backend. There's a separate router for each data model.
- "**services**" contains helper services and business logic. This package stands between the REST endpoints ("routes") and the database layer ("database").



Frontend



The frontend is based on components which have been introduced in AngularJS 1.5. A component concludes controller and HTML partial to a new HTML element. This is useful to encapsulate behaviour in single independent modules.

Read more at: https://scotch.io/tutorials/how-to-use-angular-1-5s-component-method

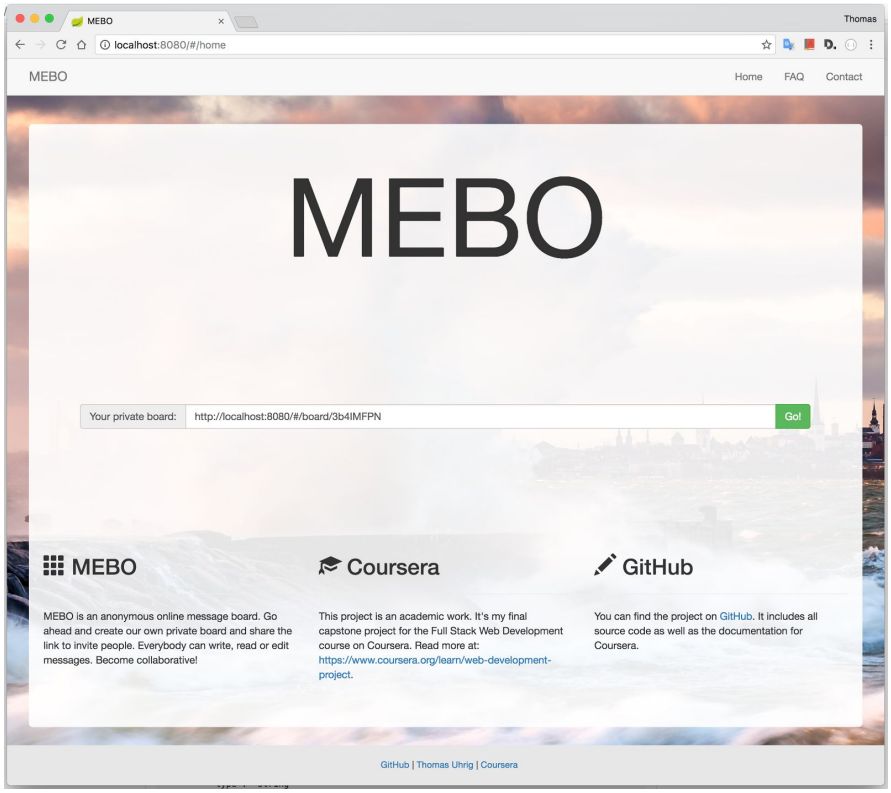
Problems

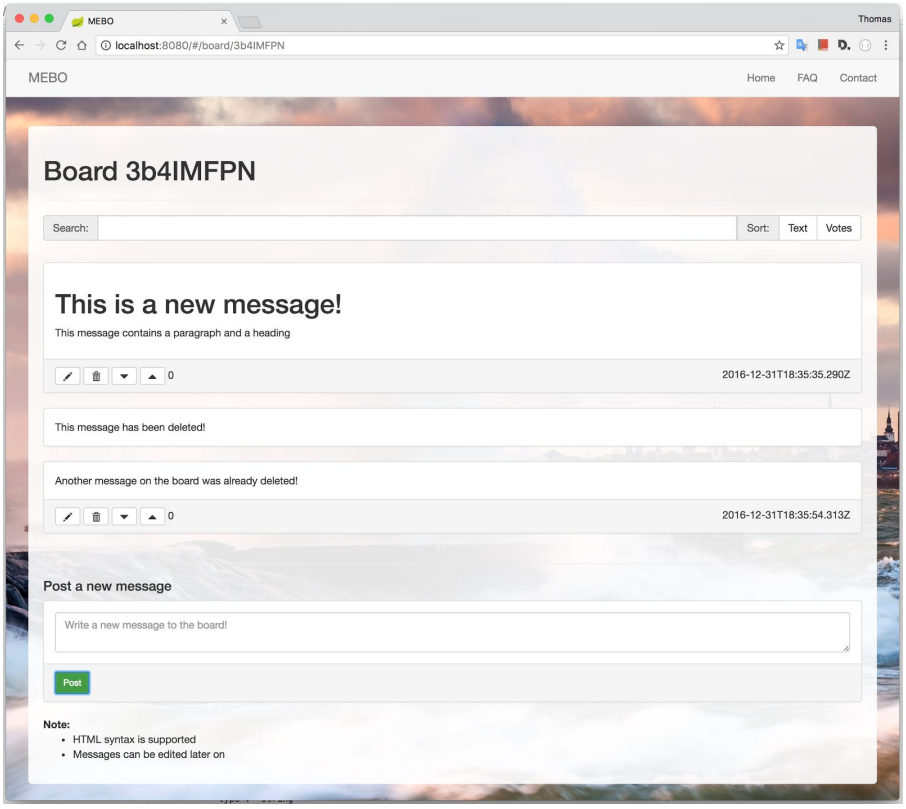
- IBM Bluemix uses an old version of MongoDB (namely 2.4). I ran into a problem with this old version ("***Mod on _id not allowed***") which cost me hours to fix, while the project was running perfectly fine on my local machine. This forced me to change some functions which saved data in MongoDB (from **mongodb.database.service.j**s).

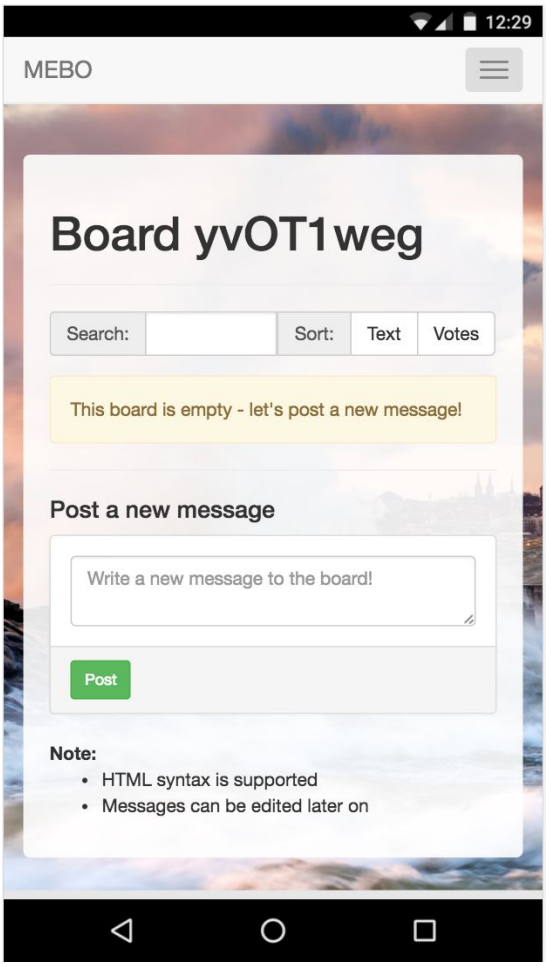
Libraries

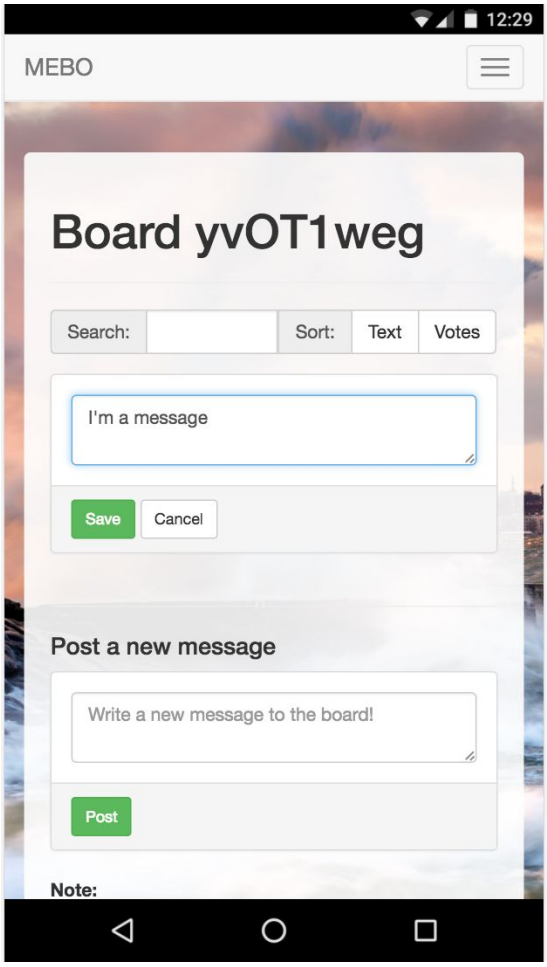
- **AngularJS** as frontend library to implement the single page application
- **Bootstrap CSS** to design the application
- **LodashJS** for operation on arrays and objects (map, filter, find, reduce)
- **Mocha**, **Chai** and **Sinon** to write unit test for the backend
- **Log4js** to do logging

Screenshots






12:29
MEBO
Board yvOT1weg
Search:
Sort:
Text
Votes
This board is empty - let's post a new message!
Post a new message
Write a new message to the board!
Post
Note:
HTML syntax is supported
Messages can be edited later on


12:29
MEBO
Board yvOT1weg
Search:
Sort:
Text
Votes
I'm a message
Save
Cancel
Post a new message
Write a new message to the board!
Post
Note:

3. Conclusions

- I was able to implement the main features of the project. Users can create boards, write messages and edit them later on.

- The project setup took a lot of time. I tried different folder structures, made an account on Travis CI and evaluated libraries to use. All of that did not add any value to the final app but was important in order to start the actual implementation.

- The deployment on IBM Bluemix was smooth and went well overall. Just the old version of MongoDB made some problems (see section "***Problems***" above).

- The most difficult part to implement was the database layer. MongoDB was new to me, so implementing the database schema with the corresponding read and write methods took some time.

4. References

- Tool used to draw the UI mock-ups:
  https://balsamiq.com/products/mockups

- Tool used to draw UML activity diagrams:
  https://www.websequencediagrams.com

- NodeJS demo application for IBM Bluemix:
  https://github.com/IBM-Bluemix/node-helloworld

- AngularJS 1.5 components tutorial:
  https://scotch.io/tutorials/how-to-use-angular-1-5s-component-method