EDAF90

Web programming

Doonutz - Project Report

Hanna Bayerlein, dic14hba
Fredrik Lütz, dic14flu
Fredrik Magnusson, fr2231ma-s
Axel Holmqvist, tfr14ah2
Johannes Persson, bas15jpe

March 15, 2019

1 Introduction

1.1 Purpose

To build a single page web application in the JavaScript framework Angular and the state manager ngrx, a state management tool similar to Redux for React. The web application will contain five to six pages and will be an online web shop, including browsing, ordering, changing the order and paying for the order. The different pages will communicate and get the vital information via their states, with the help from @ngrx/store state management. The styling of the application will be done with the ng-bootstrap framework. The design will be simple and easy to understand, even if the customer is new to the web site, but also modern and inviting.

1.2 Packages

- The npm package Angular CLI 7 is used to create a workspace and initial application. This package includes the server.
- @ngrx/store is a RxJS (Reactive Extensions library for Javascript) powered state management for Angular applications, similar to Redux, and will be used as the state management for the web shop.
- Ng-Bootstrap will be used for the design. This version only uses Bootstrap 4 CSS and the widgets and APIs are designed for the Angular framework. This means no dependencies on 3rd party JavaScript is necessary. (Later changed to Angular Material, which is discussed later in the report)

2 Method

2.1 Contribution statements

The project was divided into four different main areas, design, architecture, code and the report. All of the areas were discussed within the project group, and all the members were involved in some aspect of every area. But the work load was divided between the members and different students had different main assignments to focus on. The division looked like this;

- Design: Two students
- Code: All students
- Architecture: One student
- Report: Two students

2.2 Time Log

- Start-up meeting where the theme of the online shop was discussed, the right packages were downloaded and the architecture of the code was set. 4 hours per student.

- Self-study time, including Angular CLI, ng-bootstrap and ngrx. 4 hours per student.
- Code meeting. Started with the base set-up for the web shop, the graphic design and the architecture with the help of an UML diagram. 4 hours per student.
- Code meeting. Implemented the ngrx framework and the outline of the shop. The change from ng-bootstrap to material was decided. 4 hours per student.
- Own work, including design, coding, architecture and reading. 10 hours per student.
- Report writing. 6 hours.

3 Result

3.1 Main obstacles

Since the team members are in different years, it was sometimes difficult to setup a meeting with everyone. However, with the help of GitHub and Messenger, it was not much of a problem.

Another obstacle was the Angular import procedure. There were often different errors popping up that had something to do with a missing import, which was resolved when all the right imports were in place. This was also the case when we noticed that our code was missing the Angular Routing package from the start.

In the middle of the third day of coding we noticed that the ng-bootsrap framework did not include the kind of visual design we were aiming for. After struggling for some time the project group took the decision to switch from ng-bootstrap to angular material. Material is component based design which suited our architecture and design choices better than ng-bootsrap.

3.2 Success ratio and lessons learned

It is difficult to actually implement an idea meticulously. There will always be obstacles that may change the application over time making the website look and behave slightly different from the actual idea. For example, we made our mockup in Adobe Illustrator, and to make the exact same layout in Angular and Material would turn out to be difficult. So, one lesson that we learned was that it is okay to not completely try to replicate an idea, at least visually. Another thing that we learned after a while is that we prefer working in React. The project was however in Angular, so there was not much we could do about that. One final thing that we learned about Angular was that much of the logic is placed in the HTML components, for example when using the *ngFor* or the *ngIf* statement for conditional rendering.

In the end, we are pleased with the result. There were some minor changes during the development phase. For example, we switched from using Bootstrap to Material since Material had better components for our design choices. We also took away the option of increasing the amount of donuts in the cart since it felt unnecessary and time-consuming. The success ratio is in our opinion very high. We have a working donut shop that is built in Angular and that utilises NGRX, just as we visualized in the planning phase.