NTNU – Trondheim
Norwegian University of
Science and Technology

IT2810 Webutvikling

Arbeidsoppgave 3.2: Innlevering uke 41

Kristian Skog
Cornelius Grieg Dahling
Christoffer Andre Nilsen
Joakim Lindgren
Martin Dørum
Eirik Fosse

October 16, 2016

1 Prototype

1.1 Code documentation

The code for both the React application and the Angular application is included in the `zip` file. The React application is running on `http://it2810-08.idi.ntnu.no/`, while the Angular application is running on `http://it2810-08.idi.ntnu.no:3000/`.

React In our `index.js`, we have set up a router for the React application. Each route has a path (URL) and maps to a component. When the user visits a URL, the component matching the path will be rendered. Each of the "main pages" is located in separate folders in `src/*folder-name*`. Components that are reused are placed in `src/components`.

User logins will be handled by Facebook login. We have integrated Facebook login in our application, but the functionality handling users is not yet complete. Both the login page and the logout button is therefore rendered in the navbar, so that all the components are visible in the prototype.

The `Overview` component will be listing the users portfolio of equities. Return on investment, change last day, total value etc. will be listed here. When clicking on a row of an equity, more info are presented in a popup.

The profile page is showing info about the user, and it will also be possible to change privacy settings here (whether to make a users portfolio visible for other users). We also have a search page, where it is possible to add new funds to your portfolio. The data

for both the overview and the search page is currently hard coded in a `json` file, but the equity data will be fetched from an external API (Oslo Børs) while the stockholding data will be fetched from a local database.

Angular The Angular application is based on what we made in the React application. The structure of the application is therefore pretty much the same, although there are some deviations. The main pages are located in separate folders along with associated components. We have implemented most of the functionality we did with React, except the search function.

Similary as in React, we will use Facebook Login for user logins, display the users portfolio of equities with Overview, with a more detailed page/popup if an equity is clicked, and show information about the user in Profile. The Search page will display all the equities, and the users can search/filter through the data and add any funds/stocks to their Overview page. We also have a Home page, which will work as the default page, showing information about the latest stock changes and such.

1.2 Distribution of work

Martin was on leave this week, and did therefore not participate in this assignment. He will be back next week. If a person had made one component in React, he had to make another one in Angular to get more experience. We did not come as far on our Angular prototype as our React prototype. Much of the tasks were peer programmed.

- React
  - Routing and skeleton: Eirik
  - Login: Joakim
  - Home page: Kristian
  - Overview: Eirik and Cornelius
  - Profile: Christoffer
  - Search: Eirik and Kristian

- Angular
  - Routing and skeleton: Cornelius
  - Login: Eirik
  - Home page: Kristian
  - Overview: Joakim
  - Profile: Christoffer
  - Search: Joakim and Cornelius

1.3 Choice of framework: React

All of the team members felt that they got a better grip of React. We also think that we get the functionality that we need from React. Since Angular is a more comprehensive framework, we believe it is unnecessary to use this. During the development of the prototype in Angular, the group also encountered some unexpected problems with git. More functionality was also implemented in the React prototype, and we think we have a better basis there.

React's compile-time error versus Angular's run-time error, having React display more of a comprehensive error-log helping us the developers solve issues faster. React will provide both line number, and missing unclosed tags making finding issues quicker.

Most of Angular's documentaions is written in TypeScript, while this can be converted to, or replaced with JavaScript, our team have more experience with JavaScrip, and feel it will be easier to apply our knowledge with React.