Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Telecommunications and Media Informatics

Designing and Implementing an Educational Support System

Bachelor's Thesis

Author
Nóra Szepes

Advisors
Dr. Sándor Gajdos
Bence Golda

2015

Contents

Kivonat **i**

Abstract **ii**

1 Introduction **1**
1.1 The Old Administration Portal 1
1.2 Purpose of the Thesis 1
1.3 ? 1

2 Comparing JavaScript Frameworks **2**
2.1 React 2
2.2 AngularJS 3
2.3 Mithril 3
2.4 Conclusion 3

3 specification **4**

4 conceptual **5**

5 design **6**

6 implementation **7**

7 test **8**

8 deployment **9**

9 conclusion **10**

Acknowledgements **iii**

List of Figures **iv**

List of Tables **v**

Bibliography **vi**

Appendix **vii**

HALLGATÓI NYILATKOZAT

Alulírott *Szepes Nóra*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2015. október 18.

Szepes Nóra
hallgató

Kivonat

Abstract

Chapter 1

Introduction

During the summer of 2015 my teacher, Sándor Gajdos contacted me to give him a review about his subject, Software Laboratory 5. I told him what I thought was good and bad in the subject, not only about the tasks, but also about the administration portal. It really bothered me that the portal didn't have e-mail notification, so I told Sándor, that I'd like to develop it into the current portal. All I knew that it was written in php. I told him about my ideas and he contacted the creator of the old portal, Bence Golda to ask for some information about the old portal's code and József Márton to create a "noreply" e-mail address for the notification module. József gave us an idea to creating a new portal and other team members, Bence, Gábor Szárnyas and I agreed with the idea.

In the beginning of August we had our first meeting. Before that I decided to look up all the different homework portals I've ever worked with during my student years. I asked for an account to Zoltán Czirkos's InfoC [12], because that website started after I've finished the subject Basics of Programming 1. After some research I made a small specification for an ideal homework portal and some ideas how could we use one portal for more than one subject's administration.

During the meeting we talked about this, and what others expect from a new portal. It started as a department project but József asked for some ideas about what students want from a portal. I wanted to participate but I said that besides my thesis I won't have that much time to work on the portal. At the end Sándor offered me that this could be my thesis topic and he would be my advisor. Bence liked this idea and since I didn't have any thesis topic, I accepted the idea.

1.1 The Old Administration Portal

*****TODO***: ask Bence about details**

1.2 Purpose of the Thesis

mit tudok, mit fogok végrehajtani, mi az elérendő cél

1.3 ?

hogy álltam neki a fejlesztésnek - melyik fejezetben mit fogok bemutatni

Chapter 2

Comparing JavaScript Frameworks

For the project I wanted to choose a JavaScript framework for faster developement than using plain JavaScript with jQuery. I chose the TodoMVC [11] website to find the most popular frameworks.

I tried these frameworks to see how fast and easily can I build a basic website, how can I access a server with AJAX requests and how routing and data binding works.

In JavaScript with AJAX requests we can send requests to a server asynchronously without reloading a page. In a single-page application we want to make the browser think it is always on the same page. When the user clicks on a new link, the browser won't reload the whole page, it will just simply load the new view into the old frame. Everything happens in the background so the application won't force the user to wait while it sends data to a server. If the application is retrieving data, then when it arrives, the application can process it and show the result to the user.

There are two types of routings. Routing can be either a way to manipulate browser's URL and the part of a web application what decides which controller will handle the requests. I was looking for a solution for URL manipulation.

The classic data binding model is when the view template and the data from the model are merged together to create the to be displayed view. Any data changes in the view won't automatically sync into the model. The developer has to write the controller what syncs the changes between the model and the view. [3]

React is basically only for building views and I should use it combined with other frameworks to have full MVC experience. I really liked the fast way of creating UIs but I was looking for a framework that supports everything what I need.

Although Angular contains everything what I need, I didn't like the extended HTML vocabulary. I liked the fast and easy way of creating websites in React. Mithril provides a similar way of doing it.

In this project I'll use Mithril. It is a simpler framework than Angular, because it doesn't implement features what make it feel like a new programming language and not a JavaScript framework.

Chapter 3

specification

Chapter 4

conceptual

Chapter 5

design

Chapter 6

implementation

Chapter 7

test

Chapter 8

deployment

Chapter 9

conclusion

Acknowledgements

List of Figures

List of Tables

Bibliography

[1] AngularJS. https://www.angularjs.org/.

[2] AngularJS, API Reference, http. https://docs.angularjs.org/api/ng/service/$http.

[3] AngularJS, Developer Guide, Data Binding. https://docs.angularjs.org/guide/databinding. Accessed: 2015-10-18.

[4] JSX Specification. https://facebook.github.io/jsx/.

[5] Mithril. http://mithril.js.org/.

[6] Mithril, m.request. http://mithril.js.org/mithril.request.html.

[7] Mithril, routing. http://mithril.js.org/routing.html.

[8] MSX. https://github.com/insin/msx.

[9] React. https://facebook.github.io/react/.

[10] React/router. https://github.com/rackt/react-router.

[11] TodoMVC. http://todomvc.com/. Accessed: 2015-10-18.

[12] Dr. Zoltán Czirkos. Infoc. https://infoc.eet.bme.hu. Accessed: 2015-10-18.

Appendix