Development of a Chrome Extension for Personalized Filter Suggestions

Abschlussarbeit

zur Erlangung des akademischen Grades:

Bachelor of Science (B.Sc.)

an der

Hochschule für Technik und Wirtschaft (HTW) Berlin
Fachbereich 4: Informatik, Kommunikation und Wirtschaft
Studiengang *Angewandte Informatik*

1. Gutachter_in: Prof. Dr. Elena Schüler
2. Gutachter_in: Birol Aksu

Eingereicht von Sunan Regi Maunakea 566144

Datum

Danksagung

[Text der Danksagung]

Zusammenfassung

[Text der Zusammenfassung]

Abstract

[Summary of the thesis]

Contents

List of Figures

1 Introduction

1.1. Background

It is a common belief in modern society that the more choices, the better–that the human ability to manage, and the human desire for, choice is unlimited. One study showed that the existence of choice increases motivation and enhances performance on doing tasks [5]. However, another study has shown that although having more choices might appear desirable, it may sometimes have negative effects on human motivation [3]. In our digital and website-driven era, these studies can be applied on e-commerce and one of the solutions to this problem is a search-and-filter functionality. Amazon is one of the corporate giants for e-commerce, that implemented this feature in their platform. Another example would be marta.

Marta as a business is currently best described as a marketplace between caregivers and families requiring 24-hour care. 24-hour care can be defined as living in a household with the person in need of care for a certain period of time. This means that caregivers are primarily responsible for basic care and household chores. In addition, they support the person in care's relatives in need of assistance in carrying out the activities they wish to do. Marta as a marketplace connecting families with caregivers is competing against more traditional agencies, where it can take several days or even weeks to find a family for a newly signed up caregiver or the other way around.

1.2. Problem Statement and Objectives

In this rapidly growing world, human has access to almost everything. On the other hand, having too many options may cause negative effects on the human motivation. Imagine browsing for a cheap, small vacuum cleaner on Amazon to replace your trusty 5-years-old broom. You'd find a thousand cheap small vacuum cleaners from different brands, even in different colors. You'd end up spending most of your time comparing which vacuum cleaner would deliver the best performance, of any sort, for its price, even though all of those vacuum cleaners are exactly what you wanted, cheap and small. As a multinational technology company, Amazon aims to allow users to complete a transaction as quickly as possible. In order to achieve this goal, Amazon introduced a search-and-filter bar in their website. It sounds like a fair idea at first, but if users searched for a similar item over and over again, they would need to type in the same letters or words over and over again.

Similar to Amazon, marta would need to enhance their product, to compete in this expeditiously developing business. One way marta can provide a superior experience for both caregivers and families is to speed up the matching process. The caregiver and family inquiry forms are designed to record as much information as possible which can be used during the matching phase. The matches are created by the teams in Berlin and

Romania, but to make their job easier, the technical team in marta compute a "matching score" by which possible matches are sorted. As a result, a number of caregiver profiles with high matching scores will be shown to the family. To quicken the search, filter functionality is also provided. This includes caregiver's earliest starting date, German skills, experience with diseases, etc.

Marta needs to introduce a continually improved, user-friendly filtering functionality to adjust to the needs of families. An example of a user-friendly filter would be to provide quick filter suggestions which the user can click once and the desired results will be shown, instead of letting users select the same filter manually over and over again. In order to determine which filter suggestions are the most beneficial, frequently used filters need to be identified.

1.3. Structure of the Thesis

The bachelor thesis is structured as follows: Besides the introduction in chapter 1, browser extensions are discussed in more detail in Chapter 2 along with the reasons why they are a suitable solution to this issue. Additionally, it describes the extension's architecture and implementation. Based on this, Chapter 3 describes the experimental methodology used for studies conducted with the Chrome extension. The extension's implementation and use in a practical situation are discussed in some detail in Chapter 4 along with some lessons learned. Subsequently, the extension is evaluated and analyzed. Finally, problems are identified and an outlook for possible improvements of the extension is given.

2 Theoretical Basis

2.1. Browser Extension Background

2.1.1. Definition

Browser extensions or addons are third party programs, that can extend the functionality of browsers and improve users' browsing experience [4]. A browser extension, as opposed to a standard web page, is created specifically for a given browser and uses that browser's extension API. It was necessary to choose a browser as a result. There are frameworks that try to make it feasible to create an extension for several different browsers at once. Although the caliber of these frameworks was unclear, it was decided that the expense of potential problems and additional time spent debugging in many browsers outweighed the benefits.

2.1.2. Choice of Implementation Browser

The selection of a browser was based on a number of factors. The first was the browser's usage rate. This is significant since anyone who doesn't already have the necessary browser will need to download and install it. Drop-outs due to the installation process or being unfamiliar with a new browser can significantly raise the cost of conducting the survey. The ease of use of the API and simplicity of implementation were additional crucial criteria. The extension programming process should ideally only need a basic understanding of the extension API. The capabilities of the API offered by the browser was another factor considered. The extension needs to:

1. Store a big amount of data on the client side
2. Read URL - so query parameters can be passed to the extension
3. Modify URL - so frequently used query parameters can be utilized

Internet Explorer was the most challenging in terms of implementation simplicity. The features appeared to be restricted, and it needed knowledge of the Component Object Model (COM). [8] After reviewing the documentation, it was still unclear how tasks like making HTTP queries or changing the DOM would be carried out. The instructions appears to be primarily concerned with optional features like adding menu items and explorer bars. Thus, IE was eliminated from the list, leaving Firefox and Chrome as the only options.

Both the Firefox and Chrome extension APIs had similar features. The fact that Firefox required knowing XUL (XML User Interface Language), Mozilla's XML-based language for creating application user interfaces, was one drawback. [9] Extensions for

the Google Chrome browser can be created entirely in HTML, CSS, and JavaScript. The team members' familiarity with the languages and the simplicity of the solution were appealing. Additionally, it was discovered that the Chrome documentation was easy to grasp and was divided up into sections. Firefox's market share was roughly double that of Chrome. As a result, it was decided to develop a Google Chrome browser extension.

2.2. Chrome Extension Architecture

2.2.1. Chrome Extension Basics

A Chrome extension is only a bundled collection of files (HTML, JavaScript, etc.) that enhance the browser's capabilities. [5] They also have access to the APIs that browsers provide for tasks like XMLHttpRequests and HTML5 features on web sites. The following files can be found in an extension:

1. A manifest.json file
2. One or more HTML files
3. Any other files such as CSS or JavaScript needed by the extension to run

The majority of extensions have a background page that contains their primary logic and state. They frequently also contain content scripts that can communicate with websites. Asynchronous message passing is used to communicate between the background page and the content scripts. Additionally, extensions can save data via localStorage and other HTML5 storage APIs.

2.2.2. Manifest Files

A manifest.json file is required for each extension. It includes crucial information about the extension, such its name, version, scripts used for its content, minimum Chrome version, and permissions. Each field is described in full at http://developer.chrome.com/extensions/manifest. The content-scripts field was the most crucial one for this expansion. Each study and content-related webpage need its own content script. Each one was defined in the scripts column, which also mapped each one to the appropriate URLs.

2.2.3. Content Scripts

JavaScript files called content scripts are used on websites to add new functionality. They have full control to modify the entire web page because they can directly access the Document Object Model (DOM) of these web sites. They do, however, have some restrictions. The following list of restrictions was taken from the documentation website [4]:

1. Use chrome.* APIs
2. Use variables or functions defined by their extension's pages

3. Use variables or functions defined by web pages or by other content scripts

The good news is that you can get around some of these restrictions, such the inability to use variables defined by their extension's pages, by sending messages to the background page of their parent extension.

In order to locate and replace the proper HTML element, the content scripts are launched after the DOM has been loaded. Despite several drawbacks, this approach is effective. One is that the previous advertisement may be visible for a long period of time before it is replaced since the script must wait for the DOM to load. Users may notice a flicker as a result of this.

2.2.4. Background page

The last component of the extension is the background page.

2.3. React.js

React is a product of Facebook's engineering team, which is a JavaScript framework for creating user interfaces [1]. Because of its simplicity and straightforward but efficient development process, React is quite well-liked in the developer communities. Interactive user interfaces are simpler to develop with React. It effectively updates by accurately drawing each state's view's constituent parts, and it updates the application's data [2].

3 Methodology

3.1. Requirements Elicitation and Analysis

3.2. Design Concept

4 Project Implementation

References

[1] Cory Gackenheimer. "Introducing flux: An application architecture for react". In: *Introduction to React*. Springer, 2015, pp. 87–106.

[2] Naimul Islam Naim. "ReactJS: An Open Source JavaScript Library for Front-end Development". In: (2017).

[3] Sheena S Iyengar and Mark R Lepper. "When choice is demotivating: Can one desire too much of a good thing?" In: *Journal of personality and social psychology* 79.6 (2000), p. 995.

[4] Dolière Francis Somé. "Empoweb: empowering web applications with browser extensions". In: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2019, pp. 227–245.

[5] Miron Zuckerman et al. "On the importance of self-determination for intrinsically-motivated behavior". In: *Personality and social psychology bulletin* 4.3 (1978), pp. 443–446.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt durch meine Unterschrift, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe, mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Die Arbeit hat in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

_____

Datum, Ort, Unterschrift