ulm university universität

uulm

Universität Ulm | 89069 Ulm | Germany

**Faculty of
Engineering, Computer
Science and Psychology**
Databases and Information
Systems Department

Structured classification and evaluation of current Web UI Frameworks
Bachelor's thesis at Universität Ulm

Submitted by:
Nadine Harsch
nadine.harsch@uni-ulm.de

Reviewer:
Prof. Dr. Manfred Reichert

Supervisor:
Johannes Schobel

2019

Version from June 3, 2019

Composition: PDF-LaTeX 2_ε

Abstract

In a development and technology dependent world, digitalization plays a major role in our everyday life, so the significance of Web UI frameworks has become more and more important. A general concept is to simplify computer-based applications and especially the use of internet websites. This is where Web UI frameworks come into notice. These make web design easier with design elements and tools. Because the choice is so huge, choosing the right web UI framework is a big challenge and the question is which fits best in a given scenario.

The aim of this thesis is to give an overview of the currently existing Web UI frameworks and to support the decision-making process in the selection of a specific framework. First, state of the art Web UI frameworks are collected and listed. These frameworks are then compared against previously defined easy-to-evaluate criteria and reduced to four Web UI frameworks. The latter frameworks are then evaluated using a decision-making method, based on the considered scenario. The results of these thesis indicate that it is generally not possible to determine the most appropriate framework. The main reason for this is that the individual frameworks have different functions and work differently depending on the application. This work can be used as a guide to decide which framework suits best in a given scenario.

Acknowledgment

At this point I would like to thank my family and friends who helped me to write this bachelor thesis by correcting my work and giving feedback. My thanks also go to my parents who made my studies possible. I would also like to thank my supervisor Johannes Schobel, who helped me with useful tips.

Contents

1

Introduction

The digitalisation of the world continues and Web UI frameworks are becoming more and more important. In recent years, the choice of Web UI frameworks has become more and more extensive and confusing, so that the overview is quickly lost. Web UI frameworks offer ready-made design elements and functions that make work easier for the user. For example, the frameworks differ in their functionality. Not every framework is suitable for every use case. Therefore, it is important to check carefully what is best suited for the specific use case before deciding on a framework. Such a decision is a challenging task because of the many factors to consider.

Two main objectives are pursued in this thesis. First, an overview of the currently available Web UI frameworks is shown. For this purpose, a list of modern Web UI frameworks is presented, which are preselected according to easily evaluable criteria on four frameworks. In addition, these four remaining frameworks will be subjected to a structured comparison in order to determine which of the frameworks is best suited for our specific use case. First, criteria for the structured comparison of the frameworks are defined. The final evaluation within the framework of the respective application is carried out with the help of the Analytical Hierarchy Process.

Chapter 2 explains the evaluation technique used to evaluate the Web UI frameworks and other basic techniques required to understand the frameworks presented. Chapter 3 presents a current selection of frameworks, the criteria by which they were preselected, and the criteria used to compare frameworks. In addition, this chapter introduces the four frameworks that will be compared later. In Chapter 4, the frameworks are evaluated and compared in three steps. Chapter 5 discusses related work, followed by a summary of the thesis and possible future work in Chapter 6.

2

Background and Fundamentals

In this chapter will be explained the decision-making process in general and present the decision method used. It will also be explained what Web UI Frameworks are and when to use them.

2.1 Decision-making process

In general, decision-making processes follow eight decision-making steps, which are shown in Figure 2.1 [1]. The first and last step are marked with an ellipse, the remaining steps with rectangles. The first step is to identify the problem or opportunity. The next step is to define the requirements and collect information so that a decision can be made based on facts and data. This involves value analysis to identify what information is important for the decision and how it can be obtained [2]. For the third step, the objectives must be defined, this requires a positive evaluation of the objectives. The fourth step defines the alternatives. Usually there are different ways to achieve the goal. All possible and desired alternatives are listed. Often none of the alternatives fits the objective exactly, but the one that fits best has to be selected. The alternatives are evaluated according to a number of criteria. In the fifth step, the evaluation criteria are defined [1]. For example, Baker et al. [3] described some specifications for criteria. The next step is to choose the decision-making tool that best suits your own project. Then find out how to use the tool and apply it. The last step is to check the answer of the tool and the alternatives should be evaluated against criteria to select the most appropriate one [1].
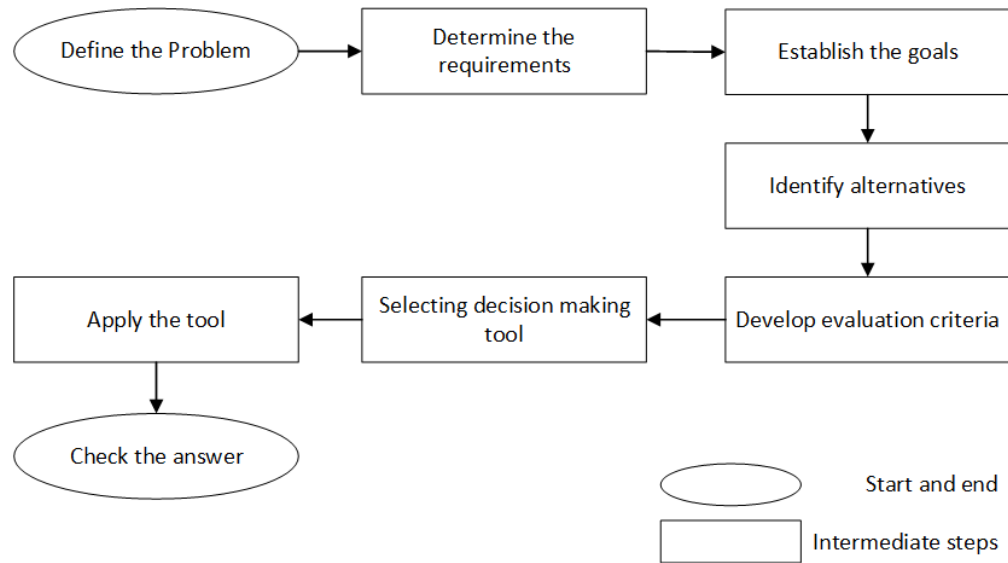
Figure 2.1: General Decision-Making Process as on [1]

2.2 AHP

The mathematician Thomas L. Saaty has developed the Analytic Hierarchy Process (AHP) to support decision-making processes. The goals of the method are to support decisions, to find a solution and to minimize the time required, to make the decision-making comprehensible, to detect possible inconsistencies in decision-making. The AHP serves to review and complement subjective decisions, as well as to elaborate qualitative weighting decisions based on comparative decisions and for the structured and hierarchical representation of a final decision by applying a decision tree [4].

In short, the AHP method is a pairwise comparison of individual weighted criteria. The advantages of AHP are the hierarchical structuring of decision problems and the combination of multiple entries from several persons to a consolidated result. The disadvantages of the method are that the weighting of each criterion has a significant influence on the final evaluation and that the weighting is based on the decision of the decision maker, so that the accuracy of this method can differ strongly because it is subjectively dependent [1].

To support decisions and prioritize them in an organized way, the decision-making process is broken down into the following steps:

1. Determine problem and possible criteria for a solution.

2. Organize the decision hierarchy from top to bottom with the goal of the decision, and then list the objectives from a wider viewpoint, through the intermediate stages to the lowest one. In the intermediate stages there are criteria on which the following elements depend. At the lowest level, a number of alternatives are usually listed.

3. Now build a set of paired comparison matrices. All elements of a higher level are compared with the elements of the level directly below. How the paired comparisons look like in general, can be seen in Figure 2.2.

4. The priorities derived from the comparisons are used to determine the following priorities. This is done for each element. Next, for each element in the level below, its weighted values are added to obtain the global priority. This continues until all priorities are set.

In order to be able to compare the criteria with each other, a scale of numbers is needed that indicates how much more important an element is to another element in relation to the criterion to which it is compared. Table 2.1 shows the scale. In the transposition position is always the reciprocal value [5].

Figure 2.2: Screenshot of Paired Comparison from [6]

Table 2.1: The fundamental scale of absolute numbers as on [5]

Intensity of Importance	Definition	Explanation
1	Equal Importance	Two activities contribute equally to the objective
2	Weak or slight	
3	Moderate Importance	Experience and judgement slightly favour one activity over another
4	Moderate plus	
5	Strong Importance	Experience and judgement strongly favour one activity over another
6	Strong plus	
7	Very strong or demonstrated Importance	An activity is favoured very strongly over another; its dominance demonstrated in practice
8	Very, very strong	
9	Extreme Importance	The evidence favouring one activity over another is of the highest possible order of affirmation
Reciprocals of above	If activity i has one of the above non-zero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i	A reasonable assumption

2.3 Web UI Frameworks

A Web UI Framework is a software framework that uses Cascading Style Sheets, Hypertext Markup Language, and JavaScript to enable simpler, standards-compliant web design. Most current frameworks are based on grid design and basic design elements such as buttons, menus, and input fields. Some frameworks offer more functionality and additional JavaScript-based features. Larger frameworks use a CSS preprocessor such as Less or Sass [7]. Web UI frameworks are designed to enhance the look and feel of web pages, web applications and everything that has a user interface. The Web UI frameworks can be installed using NPM. The next subsections will explain what Hypertext Markup Language, Cascading Style Sheets and JavaScript as well as Web pages, Web applications and NPM are.

- Hypertext Markup Language [1]: shortened HTML, is used to structure web pages and web applications and is a text-based markup language. HTML documents are displayed by web browsers. HTML is often used together with CSS and JavaScript.

- Cascading Style Sheets [2]: shortened CSS, is used to design HTML documents and is a style sheet language. CSS is used to define the presentation of content (e.g. layout, colors and typography).

- JavaScript [3]: shortened JS, is a high-level, interpreted programming language. JS allows to interact with the website and gives e.g. buttons a function. It is also an important part of web applications.

- Webpage [4]: refers to an HTML document that can be found on the World Wide Web and can be accessed with a browser using the Uniform Resource Locator (URL). Web pages are offered by web servers.

---

[1]`https://en.wikipedia.org/w/index.php?title=HTML&oldid=899304260`, last visited: May 30, 2019

[2]`https://en.wikipedia.org/w/index.php?title=Cascading_Style_Sheets&oldid=897993741`, last visited: May 30, 2019

[3]`https://en.wikipedia.org/w/index.php?title=JavaScript&direction=prev&oldid=899841445`, last visited: May 30, 2019

[4]`https://de.wikipedia.org/w/index.php?title=Webseite&direction=prev&oldid=188786386`, last visited: May 06, 2019

- Web application [5]: is an application based on the client-server model. A web application is mostly used via a web browser. For the data displayed in the Web browser, a Web UI framework can be used.

- NPM [6]: is a software registry. Open source developers use NPM to share software. Many organizations also use NPM to manage private development. The use of NPM is free. The most Web UI frameworks can be installed using NPM.

---

[5] `https://de.wikipedia.org/w/index.php?title=Webanwendung&direction=prev&oldid=189166547`, last visited: May 06, 2019
[6] `https://www.npmjs.com/`, last visited: May 06, 2019

3

Web UI Frameworks

3.1 Long List

In the Internet was looked for Web UI frameworks and then they were put in a long list. The long list consisted of the following Web UI frameworks:

- Bootstrap
- Semantic-UI
- Foundation
- Skeleton
- UIKit
- YAML
- Susy
- Materialize

- Material UI
- Pure
- Milligram
- Spectre.css
- Furtive
- Basscss
- Tailwind

- Polymer
- Bulma
- Mini.css
- Kube
- Blaze
- Purgecss
- Suit css

3.2 Criteria for Eliminating Web UI Frameworks

In order to reduce the long list to a short list, the following easy to judge criteria have taken, so that in the end four frameworks remain, which are considered closer:

1. GitHub Stars

2. Last Commit

3. CSS-Preprocessor

4. First Commit

These criteria were selected so that the remaining frameworks are not too old are or are no longer being actively developed, that the frameworks are also accepted and supported (GitHub Stars) and that the frameworks are not too young and that the use of CSS preprocessors (less or sass) is supported.

Table 3.1 shows which criteria has sorted which framework out of the long list to reduce them to four frameworks, that will be evaluated. The green ones are the ones, that will be evaluated in this work. The red ones are the ones that will eliminated by the red marked criteria. The information, which are used in Table 3.1 are from the GitHub sites of the different frameworks.

3.2.1 GitHub Stars

On the GitHub [1] website, repositories of other users can be marked with a star. GitHub stars are easy to grasp and shows how popular an open source project is. For this criterion the framework should have at least 30.000 GitHub Stars Figure 3.1 shows where the GitHub stars can be found. As an example, a screenshot of the bootstrap repository [2] was taken. The red ellipse marks the spot where the GitHub stars are listed.
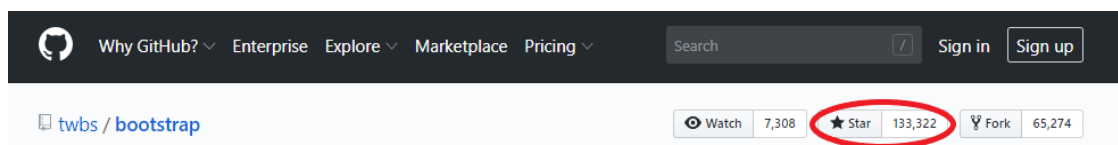

Figure 3.1: GitHub Stars

---

[1] https://github.com/, last visited on May 17, 2019
[2] https://github.com/twbs/bootstrap, last visited on May 17, 2019

3.2.2 Last Commit

GitHub was used to look up when the last commit for a framework was. The last commit should be after September 9, 2018, so that not more than six months have passed on the day of review. Figure 3.2 shows when the last commit was. As an example, a screenshot of the Materialize repository [3] was taken. The red ellipse marks the spot where the date of the last commit is listed.
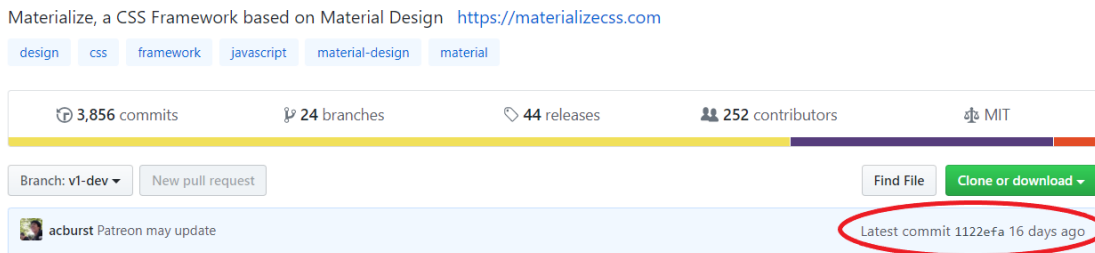
Figure 3.2: Last Commit

3.2.3 CSS-Preprocessor

Whether the Web UI Framework uses a CSS preprocessor was also checked in GitHub. A CSS preprocessor [4] is a program that can be used to generate CSS from the preprocessors syntax. There are many CSS preprocessor, in this work, the use of Sass [5] and Less [6] were considered, as these are the best known.

3.2.4 First Commit

GitHub was consulted when the first commit for a framework was. The readme file was used because it is usually one of the first files created. The time of the oldest commit should be before March 7, 2017, that the framework exists for at least two years and

---

[3] `https://github.com/dogfalo/materialize/`, last visited on May 17, 2019
[4] `https://developer.mozilla.org/en-US/docs/Glossary/CSS_preprocessor`, last visited on May 17, 2019
[5] `https://sass-lang.com/`, last visited on May 17, 2019
[6] `http://lesscss.org/`, last visited on May 17, 2019

is not that new. To find the really first commit, click "Older" until the button is grayed out, then the entry directly above the Older button will be used. Figure 3.3 shows when the first commit was. One of the red ellipses marks the spot where the date of the first commit is listed and the other red ellipse marks the older button. As an example, a screenshot of the Bulma repository [7] was taken.
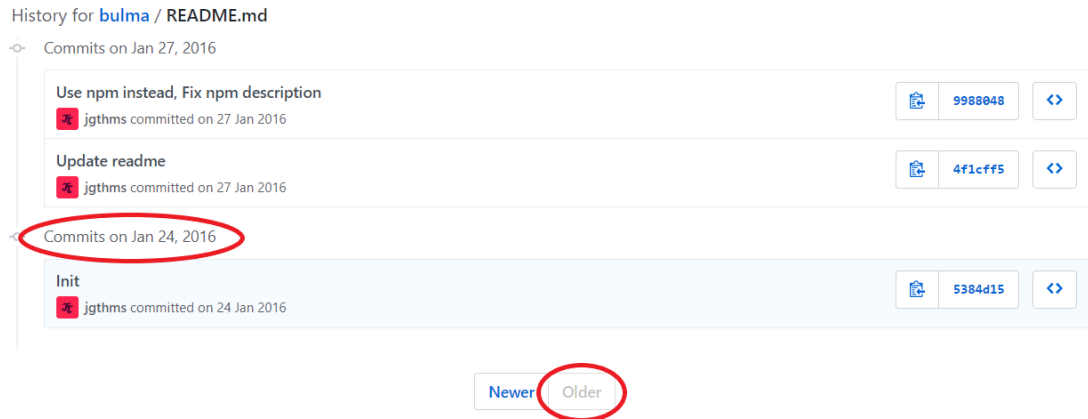

Figure 3.3: First Commit

---

[7] https://github.com/jgthms/bulma, last visited on May 17, 2019

Table 3.1: Eliminating Web UI Frameworks, last status: 2019-03-21

Framework / Criterion	Bootstrap	Semantic-UI	Foundation	Skeleton	UIKit	YAML	Susy	Materialize	Material UI	Pure	Milligram
Github Stars min. 30.000	131.740	44.847	28.038	16.528	14.139	271	3.936	35.318	45.354	19.692	7.959
Last commit after 2018-09-07	2019-03-21	2018-10-21	2019-01-28	2014-12-29	2019-03-21	2014-02-17	2018-08-28	2019-02-12	2019-03-21	2019-02-18	2019-01-30
CSS-Preprocessor Sass or Less	Less/Sass	Less	Sass	Less/Sass	Less/Sass	Sass	Sass	Sass	CSS-in-JS	-	Less
First Commit (Readme) before 2017-03-07	2011-05-04	2013-04-09	2012-06-26	2014-12-01	2015-11-13	2012-10-26	2012-07-13	2014-11-05	2014-08-18	2013-04-22	2015-12-10

Framework / Criterion	Spectre.css	Furtive	Basscss	Tailwind	Polymer	Bulma	Mini.css	Kube	Blaze	Purgecss	Suit.css
Github Stars min. 30.000	9.129	623	5.145	9.645	20.787	33.691	2.593	105	1.374	44	3.574
Last commit after 2018-09-07	2018-12-23	2018-04-07	2018-09-11	2019-03-01	2019-03-02	2019-03-20	2018-11-15	2018-12-05	2019-03-18	2019-01-26	2019-01-16
CSS-Preprocessor Sass or Less	Sass	-	Sass	Less	-	Sass	Sass	Sass	Sass	Sass/Less	PostCSS
First Commit (Readme) before 2017-03-07	2016-03-09	2014-04-09	2013-10-19	2017-07-20	2017-06-29	2016-01-24	2016-08-22	2018-07-20	2015-11-13	2016-07-04	2012-10-03

eliminated frameworks
selected frameworks for evaluation

3.3 Short List

Due to the limited time, the list of frameworks has been shortened to four. Before they are evaluated, they will be presented in this subsection. The order is for this purpose irrelevant.

3.3.1 Bootstrap

Bootstrap (version 4.3.1) is an open source frontend-framework developed by Twitter with HTML, CSS and JavaScript for developing responsive projects on the Web [8]. Bootstrap is not only statically integrated, it can be extended and adapt. There are already a number of predefined themes that can be easily integrated. Further, it is possible to customize the latter to match the design criteria and style guides of the web application (e.g. change colors or customize behaviors) [9].

Install Bootstrap via NPM, Composer or Meteor. If only bootstraps compiled CSS or JS are needed, BootstrapCDN can be used [8]. BootstrapCDN [8] is a content delivery network that immediately loads CSS and JavaScript files from their servers.

Listing 3.1 should be integrated in the Code if Bootstrap is not installed on the Computer before and using CDN is preferred.

```
1  <link rel="stylesheet"
2  href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
3  integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
4  crossorigin="anonymous">
5  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
6  integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
7  crossorigin="anonymous"></script>
```

Listing 3.1: CSS and JS import as on [8]

In Bootstrap there are a couple of options for including all compiled CSS or only the ones needed [8]. Table 3.2 and Table 3.3 shows which file include which functionality when it will be integrated in the code.

---

[8]`https://en.wikipedia.org/w/index.php?title=BootstrapCDN&oldid=894237584`, last visited on May 13, 2019

Table 3.2: Functionalities that the files contain as on [8]

CSS files	Layout	Content	Components	Utilities
bootstrap.css bootstrap.min.css	Included	Included	Included	Included
bootstrap-grid.css bootstrap-grid.min.css	Only grid system	Not included	Not included	Only flex utilities
bootstrap-reboot.css bootstrap-reboot.min.css	Not included	Only Reboot	Not included	Not included

Table 3.3: Functionalities that the files contain as on [8]

JS files	Popper	jQuery
bootstrap.bundle.js bootstrap.bundle.min.js	Included	Not included
bootstrap.js bootstrap.min.js	Not included	Not included

The code in Listing 3.2 should be included in your code if Bootstrap is installed on the computer. The href part will be adjusted depending on which functions are to be used and which CSS file and JS file is needed.

```
1  <link rel="stylesheet" type="text/css" href="/bootstrap/css/bootstrap.min.css" />
2  <script type="text/javascript" type="/bootstrap/js/bootstrap.min.js"></script>
```

Listing 3.2: Bootstrap integration in own code as on [9]

Bootstrap supports the latest versions of most popular browsers including Chrome, Firefox, and Microsoft Edge. Bootstrap 4 lets create your own theme using Sass variables, Sass maps, and custom CSS, and enable the built-in theme, for example, to add gradients and shadows. Furthermore, Bootstrap can integrate JavaScript plugins and there are data and programming interface options. Layouting your Bootstrap project is possible with wrapping container, a grid system, media queries and responsive utility classes [8]. For example, the Utility classes allow to set or hide certain topics only on smartphones, or on tablets, or on desktops. Bootstrap comes with two CSS files when downloading, one normal and one for Responsive Design. For the responsive

design CSS file, the column width of the grid system and the width of the spacing match automatically depending on the screen resolution.

If using the standard grid system, containers are needed and there are 12 columns to use. It can be chosen from a responsive, fixed-width container or fluid-width. It uses media queries to create meaningful breakpoints for layouts and interfaces because bootstrap was originally designed for mobile devices. With these breakpoints, which are based on the minimum width of the viewports, elements can be automatically scaled when the viewport changes. Since Sass is supported, media queries are also available through Sass mixins. There are also different options for typography for example global settings, headings, body text, lists. Bootstrap provides the ability to write code directly into the HTML document. Bootstrap offers many different components that can be used for own purpose and to design the web page or web application [8].

3.3.2 Semantic UI

Semantic version (2.4.2) is a UI framework using HTML, CSS and JavaScript. It is designed to scale a website across multiple devices and works with frameworks like React, Angular, Meteor and Ember. This means that it can be integrated into any of these frameworks to position the user interface layer next to the application logic. [10].

```
1   <link rel="stylesheet" type="text/css" href="semantic/dist/semantic.min.css">
2   <script
3   src="https://code.jquery.com/jquery-3.1.1.min.js"
4   integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8="
5   crossorigin="anonymous"></script>
6   <script src="semantic/dist/semantic.min.js"></script>
```

Listing 3.3: Semantic UI integration as on [11]

Using the NPM package, which contains special install scripts, is the easiest way to install Semantic UI, but it can also be used if the code in Listing 3.4 is integrated, if Semantic UI is not installed on the Computer [11]. After downloading, integrate the CSS and JavaScript files in the code as in Listing 3.3.

```
1   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/semantic-ui@2.4.2/dist/
        semantic.min.css">
2   <script src="https://cdn.jsdelivr.net/npm/semantic-ui@2.4.2/dist/semantic.min.js"></script>
```

Listing 3.4: CSS an JS import as on [11]

Semantic support Firefox, Chrome for Windows and Android, Safari for Mac and iOS, Internet Explorer and Microsoft Edge [10].

By default, the standard design is used, however the theme can be changed for individual components. Less can be used to specify which theme each component should use. In Semantic different themes, responsive elements, grids, attached content, login forms, sticky or fixed menus can be used to layout objects. Responsive tables, grids, menus and items are offered. Semantic uses 16 columns for their grid system. Semantic offers different functionalities [11].

3.3.3 Materialize

Materialize (version 1.0.0) is a Web UI Framework from Google using HTML, CSS and JavaScript, based on Material Design. Material Design is a design language that unites the fundamental principles of design with innovation and technology. It is possible to select between the standard version with minified and unminified CSS and JavaScript or the Sass version with source SCSS files [12].

```
1   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/
        css/materialize.min.css">
2   <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/
        materialize.min.js"></script>
```

Listing 3.5: CSS and JS import as on [12]

Installing Materialize is possible by using NPM, Bower or downloading the files direct on the web page [9], but it can also be used if the code in Listing 3.5 is integrated by using CDN.

---

[9]https://materializecss.com/getting-started.html, last visited on May 26, 2019

```
1    <link type="text/css" rel="stylesheet" href="css/materialize.min.css"  media="screen,
         projection"/>
2    <script type="text/javascript" src="js/materialize.min.js"></script>
```

Listing 3.6: Materialize integration as on [12]

For Materialize the CSS and JavaScript files should be integrated as on Listing 3.6. Materialize support following browsers: Chrome, Firefox, Safari, Opera, Edge and Internet Explorer. Sass and media queries can be used in Materialize [12]. There are a few community plugins that extend Materialize [13]. Materialize offers two different ready to use templates for downloading. It has an embedded responsive design, so that the website created with Materialize is redesigned according to the size of the device. Materialize is fully compatible with PC, tablets and mobile devices. In addition, various CSS functions and components are provided which can be used [12]. Materialize is by design very minimal, because it is much easier to add new CSS rules than to overwrite the existing ones [14]. It also shows what can be done with JavaScript and how to use Forms in different ways. Materialize offers a 12 column responsive grid system. Materialize shows on their web page what the mobile version looks like. It is also possible to view pages designed with Materialize. Furthermore, additional themes can also be purchased there. [12].

3.3.4 Bulma

Bulma (version 0.7.4) is a free, open source, mobile-first CSS framework based on Flexbox [15]. Flexbox [10] is an easy way to create responsive and flexible layouts without having to use fixed size values. To use Bulma only one CSS file is needed. Because Bulma only includes CSS classes, the written HTML code does not affect the styling of your page [15].

It has 3 ways to install Bulma. The first way is to install it over NPM, which is recommended, the second way is to use CDN, and the third way is to install Bulma over the GitHub repository [15].

---

[10]https://css-tricks.com/snippets/css/a-guide-to-flexbox/, last visited on May 28, 2019

If the chosen way is to use CDN, Bulma will be integrated into the own code as shown in Listing 3.7.

The recent browser versions of Chrome, Edge, Firefox, Opera and Safari are supported. Internet Explorer is only partially supported [16].

```
1  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.7.4/css/
       bulma.min.css">
2  <script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></script>
```

Listing 3.7: CSS and JS import as on [15]

In Bulma there are 39 sass files that can be imported individually. Furthermore, there are 415 Sass variables on four different levels. The levels are initial variables, derived variables, generic variables and element/component variables. Since Bulma is a mobile-first framework, the elements are also mobile-first and optimized for vertical reading. Bulma has five breakpoints for mobile, tablet, desktop, widescreen and fullhd and uses nine responsive mixins. Bulma has a grid system with 12 columns and these are named to specify how many columns of 12 are used. By default, the columns are only next to each other from the tablet size. With the mobile size the columns are deactivated, this can be changed by is-mobile. Then the columns are adapted to the display already at the mobile size. Bulma uses containers to center the content. Many elements and components are offered by Bulma [15].

4

Evaluation

In this section, the four remaining Web UI Frameworks are evaluated according to the following criteria.

4.1 Criteria

First the criteria are presented and explained what is to be examined with them.

- License and costs: describe under which license the frameworks run and how much the use of the frameworks costs.

- CSS Preprocessors: This criterion is used to see if the frameworks support CSS preprocessors and if so, which ones they support and in what way.

- Responsive Design: Display have different sizes on different devices. Therefore, it is examined whether the frameworks support responsive design, i.e. whether the website can also adapt to the display size.

- Performance and Efficiency: describe how efficiently one can work with this framework.

- Browser Version: There are many different browsers and this criterion looks at which browsers and which versions of the browsers are supported by the framework.

- Functionality: This criterion is used to examine how many functions the framework offers the user.

- Learning Curve: With learning curve is meant, how long the user needs to get along with the framework and how quickly one can get familiar with the framework.

4 Evaluation

- Support: Here it is checked whether support is provided.

  – Theme: This criterion is used to check whether the frameworks offer the user the possibility to create his own themes and whether there are already themes and how easily they can be applied by the user.

  – Community: This criterion examines whether the community supports the development of the framework and establish forks of the framework and if the community helps each other if problems occur.

  – Developer: This criterion is used to check whether and to what extent the framework is being further developed and to what extent the developers support the community in case of problems.

- Skill Level: When selecting a frontend framework, it is important to consider the user's own skills. Beginners can use a more robust framework with many useful pre-installed widgets, which requires minimal programming knowledge. With more experience, a simpler framework with plenty of space for customization can be used [17].

- Template Generator: This criterion is used to check if templates can be created and if possible, how the creation of the templates works.

- Expandability: This criterion is used to check if one can extend the framework.

- Configure / Add Modules: This criterion is to check if it is possible to configure something yourself and add modules.

- Ease of Installation: This criterion is used to check how easy the installation is and whether there are more difficulties with the installation. Furthermore, it is checked if the documentation is sufficient to install the framework, or if still have to inquire on the internet, what one has to do to use the framework.

22

- Ease of Integration: This criterion should be checked how easy it is to integrate the framework everywhere.

- Maintainability: This criterion is used to check how well the framework is maintained.
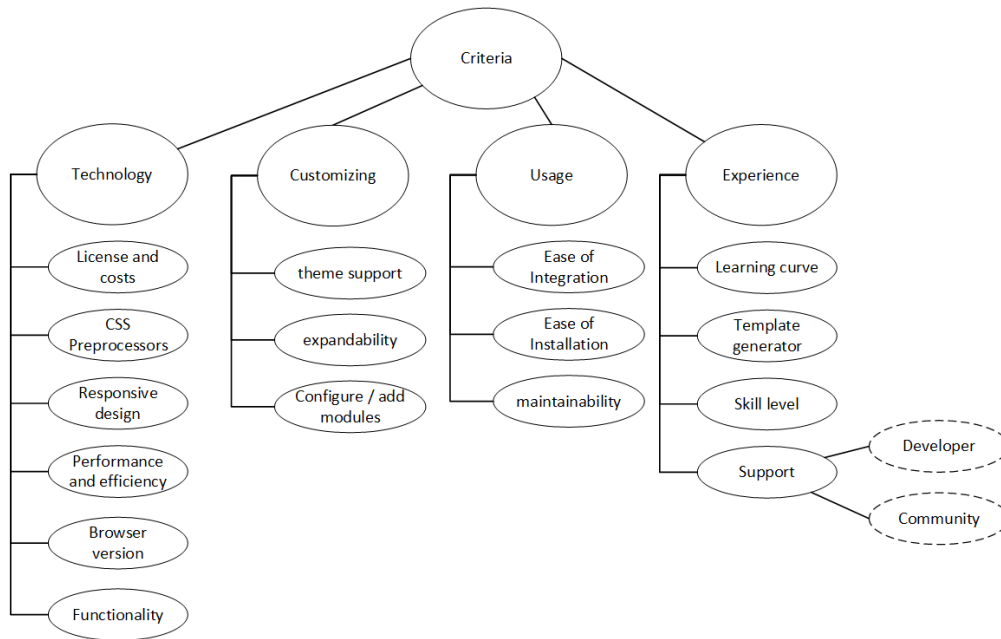

Figure 4.1: Criteria for Evaluating Frameworks

4.2 Framework Evaluation

In the next subsections, the four remaining frameworks are compared and the similarities and differences between the frameworks are shown. These are Bootstrap [1], Semantic UI [2], Materialize [3] and Bulma [4]. The most information in relation to the evaluation criteria are from the Documentation of the frameworks, linked in the footnote. The evaluation with the criteria is my own assessment and can of course be different if the priorities are set differently. The ranking tables under the individual criteria are from the online AHP evaluation tool[5].

4.2.1 License and costs

The information about the license and costs are provided by the GitHub repository of the different frameworks. With this criterion, the frameworks perform the same way, since they are all freeware and run under an MIT license [6]. With this license all rights for commercial use, modification, distribution and private use are reserved. There are limitations to liability and warranty. The Condition is a license and copyright notice.

Table 4.1: Ranking of Frameworks for License and Costs

	Priority	Rank
Bootstrap	25.0%	1
Semantic UI	25.0%	1
Materialize	25.0%	1
Bulma	25.0%	1

---

[1]`https://getbootstrap.com/`, last visited: May 28, 2019
[2]`https://semantic-ui.com/`, last visited: May 28, 2019
[3]`https://materializecss.com/`, last visited: May 28, 2019
[4]`https://bulma.io/`, last visited: May 28, 2019
[5]`https://bpmsg.com/academic`, last visited on May 31, 2019
[6]`https://opensource.org/licenses/MIT`, last visited on May 28, 2019

4.2.2 CSS Preprocessors

The usage of the CSS preprocessors Less and Sass will be evaluated. Less is based on JavaScript and Sass on Ruby.

Bootstrap has up to version 3 the possibility to use Less and from version 3 Sass. Thus, both can be used in version 3. Version 4 does not support Less anymore but Sass [8].

Semantic UI officially uses only Less, but there are versions of Sean Deng [7] and NiftyCo [8] Semantic UI that are converted to Sass.

Materialize offers Sass as CSS preprocessor. Some users have requested a Less version, but the GitHub issue [9] was closed and it was announced in February 2017 that no Less version will be provided and maintained.

Bulma provides Sass as CSS preprocessor. No information was found for a Less version.

Table 4.2: Ranking of Frameworks for CSS Preprocessors

	Priority	Rank
Bootstrap	59.5%	1
Semantic UI	26.5%	2
Materialize	7.0%	3
Bulma	7.0%	3

4.2.3 Responsive Design

All of the 4 Frameworks support responsive design.

Bootstrap is fully responsive with five sizes and based on Flexbox. The five different sizes are *extra small* `.col-` until 575px, *small* `.col-sm-` from 576px, *medium* `.col-md-` from 768px, *large* `.col-lg-` from 992px and *extra large* `.col-xl-` from 1200px [8].

---

[7]`https://github.com/doabit/semantic-ui-sass`, last visited: May 28, 2019
[8]`https://github.com/aniftyco/semantic-ui-sass`, last visited: May 28, 2019
[9]`https://github.com/Dogfalo/materialize/issues/313`, last visited: May 28, 2019

Semantic UI has a responsive grid, but there is no further information in the documentation about breakpoints or different sizes.

Materialize has a fluid responsive grid with four different sizes. These are *mobile* `.s` until 600px, *tablet* `.m` from 601px, *desktop* `.l` from 993px and *large desktop* `.xl` from 1201px [12].

Bulma has in addition to the five breakpoints nine mixins and two sizes that are not included in the previous ones. The five sizes are *mobile* `is-mobile` until 768px, *tablet* `is-tablet` from 769px, *desktop* `is-desktop` from 1024px, *widescreen* `is-widescreen` from 1216px and *fullhd* `is-fullhd` from 1408px. The nine mixins has the same five sizes than the breakpoints and additionally four new sizes, these are *tablet-only* from 769px until 1023px, *touch* until 1023px, *desktop-only* from 1024px until 1215px, *widescreen-only* from 1216px until 1407px. The two additional sizes are *until-widescreen* until 1215px and *until-fullhd* until 1408px [15].

Table 4.3: Ranking of Frameworks for Responsive Design

	Priority	Rank
Bootstrap	26.0%	2
Semantic UI	6.2%	4
Materialize	17.2%	3
Bulma	50.6%	1

4.2.4 Performance and Efficiency

For performance, the calendar with the individual frameworks was used. It was examined in Google Chrome how long the files of the frameworks take to load the content. The content was loaded five times and then the average value was calculated. The loading time is very various, because of the server connection. When loading for the first time there are differences, but the user does not really notice them, because it is in the millisecond range. Furthermore, this will not matter later, because the files will be loaded from the cache.

Bootstrap has a bootstrap.min.css which size is 22,9 kilobyte (kB) and needs 30,91 milliseconds (ms) to load content. The jquery.min.js has 30,3 kB and needs 18,56 ms. The bootstrap.min.js has 15,2 kB and needs 14,15 ms.

Semantic UI has a semantic.min.css which size is 101 kB and needs 81,75 ms to load content. The jquery-3.1.1.min.js has 29,8 kB and needs 26,78 ms. The semantic.js has 719 kB and needs 9,04 ms.

Materialize has a materialize.min.css which size is 19,8 kB and needs 16,98 ms to load content. The jquery.min.js has 30,1 kB and needs 17,92 ms. The materialize.min.js has 40 kB and needs 18,97 ms.

Bulma has a bulma.min.css which size is 21,2 kB and needs 22,49 ms to load content. The jquery-3.1.1.min.js has 29,8 kB and needs 32,55 ms. The all.js has 401 kB and needs 134,99 ms.

Table 4.4: Ranking of Frameworks for Performance and Efficiency

	Priority	Rank
Bootstrap	25.0%	1
Semantic UI	25.0%	1
Materialize	25.0%	1
Bulma	25.0%	1

4.2.5 Browser Version

Bootstrap supports the latest versions of Chrome and Firefox for Android, iOS, Mac and Windows. Microsoft Edge is supported for Android, iOS, Mac, Windows as well as Windows 10 mobile. Android also supports the Android web browser version 5.0 and higher. Safari is also supported for iOS and Mac. For Mac and Windows additionally, Opera is supported and Windows further supports Internet Explorer from version 10 [18]. Officially, Bootstrap supports these browser versions, but most older versions still work.

Semantic UI supports the latest two versions of Chrome and Firefox, as well as Safari for Mac. Microsoft Edge is supported from version 12. Android also supports Chrome for Android version 44 and higher starting with Android version 4.4. For iOS Safari from

version 7 is also supported. Windows also supports Internet Explorer version 10 or higher [10].

Materialize supports Chrome from version 35, Firefox from version 31, Safari from version 9, Opera, Edge and Internet Explorer from version 11 [13].

Bulma supports the recent versions of Chrome, Firefox, Safari, Opera and Edge. Internet Explorer version 10 or higher is only partially supported [16].

Table 4.5: Ranking of Frameworks for Browser Version

	Priority	Rank
Bootstrap	24.4%	2
Semantic UI	21.9%	3
Materialize	43.8%	1
Bulma	9.9%	4

4.2.6 Functionality

The functionalities are all taken from the documentation of the frameworks.

Bootstrap offers 24 components for example buttons, button groups, media objects and tooltips. In addition, there are 19 utilities offered, for example borders, sizing and shadows.

Semantic UI offers 16 elements, for example Buttons, Header and Icons. In addition, six collection as example forms and tables as well as 15 modules such as checkbox, dropdown and modal.

Materialize offers 11 components, for example Buttons and Icons. In addition, there are 15 JavaScript functionalities like modals and tooltips as well as nine forms, for example checkboxes and radio buttons.

Bulma offers seven forms such as checkboxes and text areas, as well as 11 elements such as buttons and icons. Furthermore 10 components like dropdowns and modals are offered.

Table 4.6: Ranking of Frameworks for Functionality

	Priority	Rank
Bootstrap	51.7%	1
Semantic UI	23.8%	2
Materialize	16.8%	3
Bulma	7.7%	4

4.2.7 Learning Curve

Bootstrap has a lot of tutorials and also the documentation is very detailed and helpful to quickly get along with it. Especially the tutorials from w3schools [10] are very useful. The search bar on the documentation page is also very helpful if something is not found right away.

Semantic UI has also many tutorials a helpful documentation. A disadvantage of the documentation is that there is no search function. This makes it harder to find the functions being searched for, so it is a bit more difficult to quickly get along with the framework. Udemy [11] offers a good, helpful and free tutorial for beginners.

Materialize also offers a helpful documentation, which also has a search function, so it is easier to find the function that is being searched for. Also, for Materialize there are some helpful tutorials, which make it easier to get along with the framework, an example for this would be the tutorial of Tutorialspoint [12].

Bulma offers just like the three previous frameworks, also a helpful documentation, which unfortunately again does not offer a search function. In the internet some tutorials like that of Tutorialspoint [13] can be found, which make it easier to start working with the framework.

---

[10]`https://www.w3schools.com/bootstrap4/`, last visited on May 30, 2019

[11]`https://www.udemy.com/semantic-ui-for-beginners/`, last visited on May 30, 2019

[12]`https://www.tutorialspoint.com/materialize/`, last visited on May 30, 2019

[13]`https://www.tutorialspoint.com/bulma/index.htm`, last visited on May 30, 2019

Table 4.7: Ranking of Frameworks for Learning Curve

	Priority	Rank
Bootstrap	46.7%	1
Semantic UI	16.0%	3
Materialize	27.7%	2
Bulma	9.5%	4

4.2.8 Theme support

With each of the four frameworks it is possible to use themes and create own themes.
Bootstrap offers an own page [14] to buy themes and templates and there is also a documentation [15] on how to use it. Additionally, there is the possibility to sell own Bootstrap Themes on their page. Beside the official bootstrap theme page there are many other pages where themes can be downloaded.
Semantic UI explains in the official documentation how to use themes and how to manage own themes through theme.config.
Materialize also offers like Bootstrap on their official page [16] themes for sale and there is an active support for every purchased theme. For this the questions and concerns should be sent by mail. Additionally, to the official page, there are a lot of more pages where themes can be downloaded.
Bulma also offers themes [17] and templates [18], which can be downloaded and edited for free. And, here there are further pages where themes can be downloaded.

---

[14]`https://themes.getbootstrap.com/`, last visited on May 30, 2019
[15]`https://bootstrap-themes.github.io/dashboard/docs/`, last visited on May 30, 2019
[16]`https://materializecss.com/themes.html`, last visited on May 30, 2019
[17]`https://www.bulmathemes.com/`, last visited on May 30, 2019
[18]`https://github.com/BulmaTemplates/bulma-templates/blob/master/templates/blog.html`, last visited on May 30, 2019

Table 4.8: Ranking of Frameworks for Theme Support

	Priority	Rank
Bootstrap	50.2%	1
Semantic UI	9.0%	4
Materialize	27.0%	2
Bulma	13.8%	3

4.2.9 Community support

All four frameworks offer Twitter [19] and GitHub to exchange information with the community. Furthermore, on Stack Overflow for all four frameworks help can be found. For every framework a fork can be build.

Bootstrap also offers the possibility to chat via Internet Relay Chat [20] on the server `irc.freenode.net`. A channel on Slack [21] is also offered, but it doesn't work at the moment. According to GitHub there are currently 65,470 forks from Bootstrap. An example would be Jasny Bootstrap [22].

Semantic UI also offers Gitter [23] and a user forum [24], which is not available at the moment. According to GitHub Semantic UI has 4,780 Forks. One of the best-known fork is Fomantic UI [25].

Materialize offers as well as Semantic UI Gitter. According to GitHub, there are currently 4780 forks.

Bulma does not offer any other ways to interact with the community other than those mentioned above. There exist currently 2797 forks.

---

[19] `https://twitter.com/`, last visited on May 28, 2019
[20] `https://en.wikipedia.org/w/index.php?title=Internet_Relay_Chat&oldid=899204389`, last visited on May 29, 2019
[21] `https://slack.com/intl/de-de/`, last visited on May 29, 2019
[22] `https://www.jasny.net/bootstrap/`, last visited on May 29, 2019
[23] `https://gitter.im/`, last visited on May 29, 2019
[24] `http://forums.semantic-ui.com/`, last visited on May 29, 2019
[25] `https://fomantic-ui.com/`, last visited on May 29, 2019

Table 4.9: Ranking of Frameworks for Community Support

	Priority	Rank
Bootstrap	20.9%	3
Semantic UI	34.6%	1
Materialize	34.6%	1
Bulma	9.8%	4

4.2.10 Developer support

The commits in GitHub show that Bootstrap and Bulma were recently updated. The last commit on Materialize was on first May and Semantic UI hasn't done anything since October in 2018. According to official documentation, all developers offer to contact them via Twitter or GitHub if there are any questions.

Bootstrap offers an additional blog from the developer side, which should support in case of problems.

Semantic UI offers email and Gitter for developer support.

Materialize also offers Gitter.

Bulma offers a book [26] with step by step instructions, which costs $15 in the regular version and $20 in the deluxe version.

Table 4.10: Ranking of Frameworks for Developer Support

	Priority	Rank
Bootstrap	39.1%	1
Semantic UI	13.8%	4
Materialize	27.6%	2
Bulma	19.5%	3

---

[26]`https://jgthms.com/css-in-44-minutes-ebook/`, last visited on May 30, 2019

4.2.11 Skill Level

The skill levels are evaluated based on the offered functionalities, since it is easier for beginners if all the functionalities are already available and it is not necessary to create them yourself.

Bootstrap has with 43 offered functionalities the most, so this framework is a good choice for Beginner.

Semantic UI has 37 offered functionalities, so its better for a bit more advanced programmer.

Materialize has 35 offered functionalities, so it is for advanced programmers.

Bulma has only 28 functionalities and does not offer JavaScript functions, so it is rather for experts who can write the required JavaScript functions themselves.

Table 4.11: Ranking of Frameworks for Skill level

	Priority	Rank
Bootstrap	42.8%	1
Semantic UI	28.4%	2
Materialize	20.0%	3
Bulma	8.7%	4

4.2.12 Template Generator

For **Bootstrap** there is a template generator [27] with which the desired functions can easily be dragged and dropped to the place where they should be. It will directly be displayed what it looks like and either only the HTML or a zip with directory structure, CSS, less and JS can be downloaded.

For **Semantic UI** could no template generator be found.

For **Materialize** there is a template generator [28] with which the desired functions can be selected as HTML code quite simply by clicking, depending on what is needed. Clicking on Run displays a preview. The created code can then be copied and pasted at the right

---

[27]`https://www.layoutit.com/build`, last visited on May 30, 2019
[28]`http://jamespetney.com/portfolio/materializecss.html`, last visited on May 30, 2019

place.

For **Bulma** also no template generator could be found.

Table 4.12: Ranking of Frameworks for Template Generator

	Priority	Rank
Bootstrap	50.4%	1
Semantic UI	8.9%	3
Materialize	31.8%	2
Bulma	8.9%	3

4.2.13 Expandability

All four frameworks are expandable. These frameworks are all open source and therefore everyone has the possibility to create a fork and to extend and modify it according to his own wishes.

Table 4.13: Ranking of Frameworks for Expandability

	Priority	Rank
Bootstrap	25.0%	1
Semantic UI	25.0%	1
Materialize	25.0%	1
Bulma	25.0%	1

4.2.14 Configure / Add Modules

Since all four frameworks are open source, the existing framework can be taken and configured according to your wishes and preferences. As an example, it is possible to create a Less version if only Sass is officially offered. Furthermore, it is also possible to add own modules which are required.

Table 4.14: Ranking of Frameworks for Configure / Add Modules

	Priority	Rank
Bootstrap	25.0%	1
Semantic UI	25.0%	1
Materialize	25.0%	1
Bulma	25.0%	1

4.2.15 Ease of Installation

The installation of the frameworks for **Bootstrap**, **Materialize** and **Bulma** worked well and without problems. At **Semantic UI** my laptop as well as the desktop computer got stuck, but after a while this framework could be installed as well.

Table 4.15: Ranking of Frameworks for Ease of Installation

	Priority	Rank
Bootstrap	30.0%	1
Semantic UI	10.0%	4
Materialize	30.0%	1
Bulma	30.0%	1

4.2.16 Ease of Integration

Bootstrap shows in the documentation how CSS and JS can be integrated in two different ways. One way is the integration via BootstrapCDN, and the other way is how it has to be integrated if the files are available locally. The explanation is easy to understand and can be easily imitated.

Semantic UI also shows in the documentation two different ways how the CSS and JS can be integrated. The ways are the same as for Bootstrap and the explanation is also easy to understand and the embedding is easy to imitate.

With **Materialize** it is just as easy and understandable as with the two frameworks before it. Materialize also offers the same two types as the other two.

With **Bulma** the whole thing is a bit more difficult, it is only explained how to integrate the CSS via CDN. Otherwise there is one explanation how to include the files if using NPM and this explanation is not as easy to understand as the others.

Table 4.16: Ranking of Frameworks for Ease of Integration

	Priority	Rank
Bootstrap	30.0%	1
Semantic UI	30.0%	1
Materialize	30.0%	1
Bulma	10.0%	4

4.2.17 Maintainability

In GitHub it is possible to look when they released the last version of the framework. **Bootstrap** releases new versions and fixes at irregular intervals. Furthermore, alpha and beta versions are released from time to time, where users can test new features. **Semantic UI** also offer new versions and fixes at irregular intervals. However, here is the last release from October 2018. No alpha or beta versions were found. **Materialize** offers alpha and beta versions, which are also listed in the releases in GitHub. Here, as well, new versions and fixes are released at irregular intervals. **Bulma** also releases new versions and fixes at irregular intervals. As with semantic ui, no alpha and beta versions could be found.

Table 4.17: Ranking of Frameworks for Maintainability

	Priority	Rank
Bootstrap	38.6%	1
Semantic UI	6.3%	4
Materialize	38.6%	1
Bulma	16.5%	3

4.3 AHP

In this section are the results of the evaluation with AHP, based on the framework evaluation with the different criteria. 4.2 shows the weighting of the four frameworks after the evaluation. With my chosen priorities of the evaluation criteria section 4.2, the following ranking comes out after the evaluation by the online calculator [29]. Bootstrap is on rank 1, followed by Materialize, then Semantic UI and finally Bulma as shown in Table 4.18. Figure 4.3 illustrates the weighting of the criteria. The most important sections of the generated `.csv` file from the online AHP evaluation can be found in Table A.1 and Table A.2.
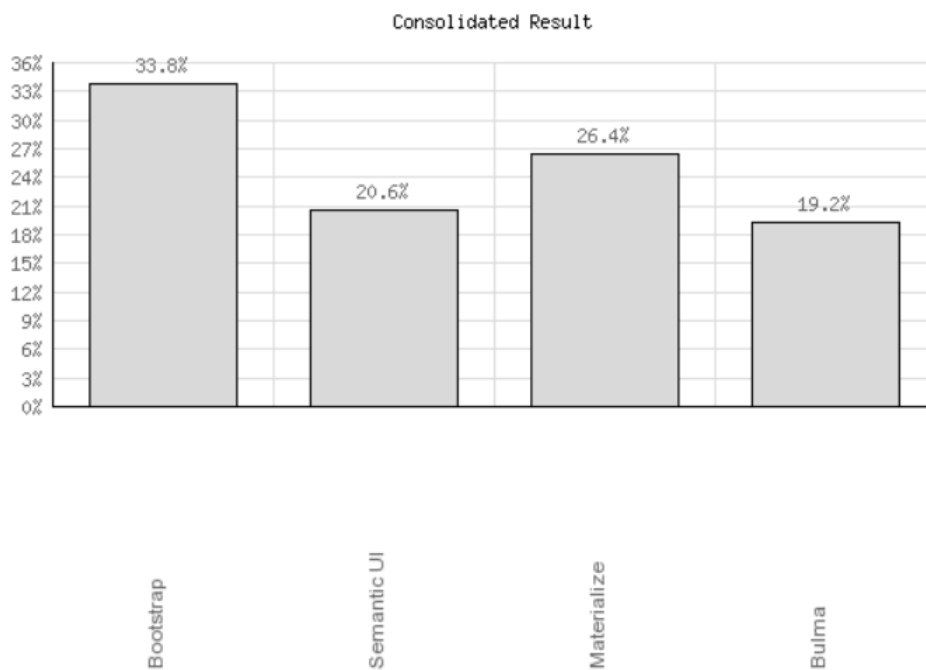
Figure 4.2: Consolidated Weights of Frameworks

---

[29]`https://bpmsg.com/academic`, last visited on May 31, 2019

Table 4.18: Ranking of Frameworks

Bootstrap	33.8%	1
Materialize	26.4%	2
Semantic UI	20.6%	3
Bulma	19.2%	4

Figure 4.3: AHP Decision Hierarchy

4.4 Calendar

This section presents the scenario on which the individual frameworks were applied. For this a HTML calendar was created, which also has some JavaScript functions. As a basis for the creation of the HTML calendar, the page from MrKnowing [30] was used. The calendar displays a whole month and if the month does not start on a Monday, the last days of the previous month are also displayed. The first days of the following month are also always displayed. It can be changed between the individual months. If the mouse is moved over days of the selected month, the day on which the mouse hovers is displayed in blue. Below the days is an input field and next to it a save button. By clicking this button, a modal opens which asks if the user is sure to save the event. By clicking on 'Save changes' or 'No' the modal closes itself. All four frameworks offered the desired functions for the design of the calendar. Because of this no statement can made about which framework is best suited in this case. That depends very much on what the user wants.

Figure 4.4 shows the calendar without the use of any framework.

«	**May 2019**					»
Mo	**Di**	**Mi**	**Do**	**Fr**	**Sa**	**So**
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Figure 4.4: Calendar without Framework

---

[30] http://www.mrknowing.com/2013/07/25/eigenen-html-kalender-erstellen-html-und-javascript/, last visited on May 15, 2019

As example for the Calendar with a Framework is Bootstrap chosen. Figure 4.5 shows the calendar designed with Bootstrap. Figure 4.6 shows the modal. The additional pictures of the calendar with Semantic UI, Materialize and Bulma are in Appendix A.

«	May 2019					»
Mo	**Di**	**Mi**	**Do**	**Fr**	**Sa**	**So**
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Save

Figure 4.5: Calendar with Bootstrap
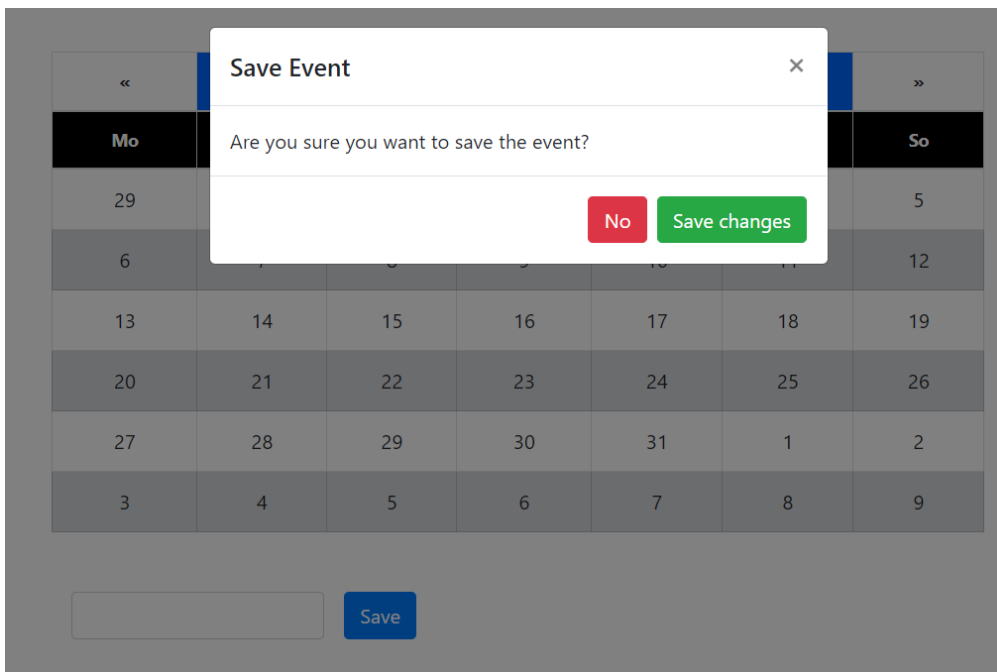
Figure 4.6: Bootstrap Modal

5

Related Work

Since no closely related work on the comparison of Web UI frameworks was found, related work from the more general area was taken, which compare frameworks with each other.

[19] compares current frameworks for the development of mobile applications. In the thesis, different development approaches were presented and three suitable frameworks were selected, each of them realising one of these development approaches. The aim of the work was to give an overview of the currently available technologies with these frameworks and to identify the strengths and weaknesses of the frameworks. Soft criteria are used for the comparison and hard criteria for the final evaluation. The hard criteria result in a ranking for the frameworks, which differs depending on the application. The evaluation for the question which framework is best suited for which of the three use cases takes place with the help of AHP. As the paper shows, no general statement can be made about which framework is best, as this depends on many different factors.

[20] analyzes frameworks for the development of web applications and classifies and evaluates them according to various distinguishing characteristics. Five classification criteria were defined. Wende has selected nine frameworks which are widely used or at least offer an interesting concept. After all frameworks were classified in the classification scheme, three basic types were defined for the frameworks. With the same or very similar classification, the frameworks were assigned to the same type. For each type a framework was selected and with these a small web application was implemented. This web application shows that the examined approaches differ widely in terms of implementation complexity, flexibility and abstraction. In conclusion, it can be said that it depends very much on the usage which framework is best suited with which approach.

[21] analyzes, compares and evaluates web frameworks using a test environment. The frameworks are analyzed based on selected criteria. A scenario was created with the different frameworks to ensure that the same conditions prevail. Three different technologies were chosen, each represented by a framework. The goal of the work was the analysis and evaluation of how the individual frameworks with different architectural patterns in the developed test environment have been completed. The Google Web Toolkit was the winner for this test case.

[22] provides an overview of the current concepts of modular programming and compares MV* frameworks of the script language JavaScript for this purpose. These frameworks are used to write a simple client-side application and are evaluated according to various criteria. The result of this work is that due to the rapid development of technologies it is also necessary to adapt the application developments to it. Furthermore, it was shown that for small client-side applications the frameworks AngularJS or VueJS are recommended and for larger applications BackboneJS.

[23] analyses and compares the WebGL frameworks three.js, Processing.js and SceneJS under certain criteria using a demo application that was implemented with all three frameworks. The result of the work is that three.js performs best and that the development of WebGL applications can be facilitated using frameworks. Furthermore, the strengths and weaknesses of the three frameworks were pointed out. In addition, it was explained how improvements could be achieved. It was noted that the frameworks are still at the beginning of their development.

6

Summary

In this thesis different Web UI frameworks were listed, which were found during the research on the topic. As a limitation it must be mentioned that not all frameworks can be evaluated due to the limited time. In order to reduce the number of frameworks, easy-to-evaluate criteria were used, which reduced the number of frameworks to be evaluated to four. In practice it is unlikely that a selection will be made according to the chosen easy to-evaluate criteria, since only those frameworks will be considered that are suitable for the user. After the research on decision-making processes it was decided to use AHP for the evaluation in this thesis, but this could also be done with another decision-making process. AHP was chosen because it structures the problem effectively and makes the evaluation easy to reconstruct. Furthermore, AHP is also suitable if several people are involved in the decision. The following evaluation with the criteria chosen in this thesis would also be different depending on the scenario for the use of the framework. Depending on the priorities and wishes for the respective scenario, other criteria are more important than the prioritized ones. Depending on this, the inputs into the AHP tool should also be modified. For example, if a web application and a mobile app should be designed in the same way, different frameworks and criteria will be selected than if only one web page is designed. Another limitation of this work is that due to the criteria considered and the limited time, only one scenario was implemented. In the case of several different scenarios with different criteria, the result would probably have been different than in this case. Thus, it can be said that it is impossible to make a general and always accurate statement which framework is best suited. The choice of the frameworks to be examined as well as the choice of criteria and the prioritization of these have an impact on the evaluation result.

6.1 Future Work

Further to this work, the remaining frameworks could be evaluated with the same criteria and then all frameworks could be compared with each other. Another approach would be to select fewer criteria and evaluate more frameworks based on the selected criteria. Furthermore, various elaborate scenarios with differently prioritized criteria could serve as a basis for the evaluation.

Bibliography

[1] Sabaei, D., Erkoyuncu, J., Roy, R.: A review of multi-criteria decision making methods for enhanced maintenance delivery. Procedia CIRP **37** (2015) 30 − 35 CIRPe 2015 - Understanding the life cycle implications of manufacturing.

[2] Hussung, T.: Steps of the decision making process. `https://online.csp.edu/blog/business/decision-making-process` (2019) Last viewed on May 28, 2019.

[3] Baker, D., Bridges, D., Hunter, R., Johnson, G., Krupa, J., Murphy, J., Sorenson, K.: Guidebook to decision-making methods. Technical report, WSRC-IM-2002-00002, Department of Energy, USA (2002)

[4] Wikipedia: Analytic hierarchy process. `https://de.wikipedia.org/w/index.php?title=Analytic_Hierarchy_Process&oldid=184371075` (2019) Last viewed on April 29, 2019.

[5] Saaty, T.L.: Decision making with the analytic hierarchy process. International journal of services sciences **1** (2008) 83–98

[6] Goepel, K.D.: Ahp priority calculator. `https://bpmsg.com/academic/ahp_calc.php` (2019) Last viewed on June 02, 2019.

[7] Wikipedia: Css framework. `https://en.wikipedia.org/w/index.php?title=CSS_framework&direction=prev&oldid=897633844` (2019) Last viewed on May 06, 2019.

[8] Otto, M., Thornton, J., Rebert, C., XhmikosR, Lauke, P.H., Mazovetskiy, G., Johann-S, Galante, A., Cuppens, M.: Bootstrap. `https://getbootstrap.com/` (2019) Last viewed on March 06, 2019.

[9] Leirich, W.: Bootstrap. `https://www.work-paper.de/twitter-bootstrap-css-framework/` (2019) Last viewed on May 11, 2019.

[10] Semantic-Organisation: Semantic ui github. `https://github.com/semantic-org/semantic-ui/` (2019) Last viewed on May 16, 2019.

[11] Semantic-Organisation: Semantic-ui. `https://semantic-ui.com/` (2019) Last viewed on May 14, 2019.

[12] Wang, A., Chang, A., Mark, A., Louie, K.: Materialize. `https://materializecss.com/` (2019) Last viewed on May 14, 2019.

[13] Wang, A., Chang, A., Mark, A., Louie, K.: Materialize github. `https://github.com/Dogfalo/materialize` (2019) Last viewed on May 25, 2019.

[14] Javatpoint: Materialize tutorial. `https://www.javatpoint.com/materialize-css-tutorial` (2019) Last viewed on May 28, 2019.

[15] Thomas, J.: Bulma. `https://bulma.io/` (2019) Last viewed on May 28, 2019.

[16] Thomas, J.: Bulma github. `https://github.com/jgthms/bulma` (2019) Last viewed on May 28, 2019.

[17] Arsenault, C.: Top 10 front-end frameworks in 2018. `https://www.keycdn.com/blog/front-end-frameworks` (2018) Last viewed on May 28, 2019.

[18] Otto, M.: Bootstrap github. `https://github.com/twbs/bootstrap` (2019) Last viewed on May 28, 2019.

[19] Schwer, D.: Strukturierter vergleich aktueller frameworks zur entwicklung mobiler anwendungen. Bachelor's thesis, Universität Ulm, `http://dbis.eprints.uni-ulm.de/1586/` (2017) Last viewed on May 28, 2019.

[20] Wende, A.: Klassifikation und bewertung von frameworks für die entwicklung von web-anwendungen. Bachelor's thesis, Universität Leipzig, `http://nbn-resolving.de/urn:nbn:de:bsz:15-qucosa2-168088` (2005) Last viewed on May 28, 2019.

[21] With, N.: Testbett für eine kriteriengestützte analyse von webframeworks. Bachelor's thesis, Hochschule für angewandte Wissenschaften Hamburg, `http://edoc.sub.uni-hamburg.de/haw/volltexte/2010/1074/index.html` (2010) Last viewed on May 28, 2019.

[22] Keller, R.: Vergleich von mv* frameworks der skriptsprache javascript. Bachelor's thesis, Hochschule Mittweida, `https://monami.hs-mittweida.de/frontdoor/deliver/index/docId/8480/file/Bachelorarbeit_Normal_Farbe_RobertKeller_M31567.pdf` (2016) Last viewed on May 28, 2019.

[23] Kornher: Analyse und vergleich von webgl-frameworks. Bachelor's thesis, Eberhard Karls Universität Tübingen, `https://moritzkornher.de/files/kornher_analyse_und_vergleich_von_webgl-frameworks.pdf` (2013) Last viewed on May 28, 2019.

A
Sources

In this chapter are the screenshots of the remaining calendars designed with the frameworks and the snippets of the `.csv` file from the project of the online tool [1].

«	May 2019					»
Mo	**Di**	**Mi**	**Do**	**Fr**	**Sa**	**So**
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Save

Figure A.1: Calendar with Semantic UI

---

[1] `https://bpmsg.com/academic`, last visited on May 31, 2019
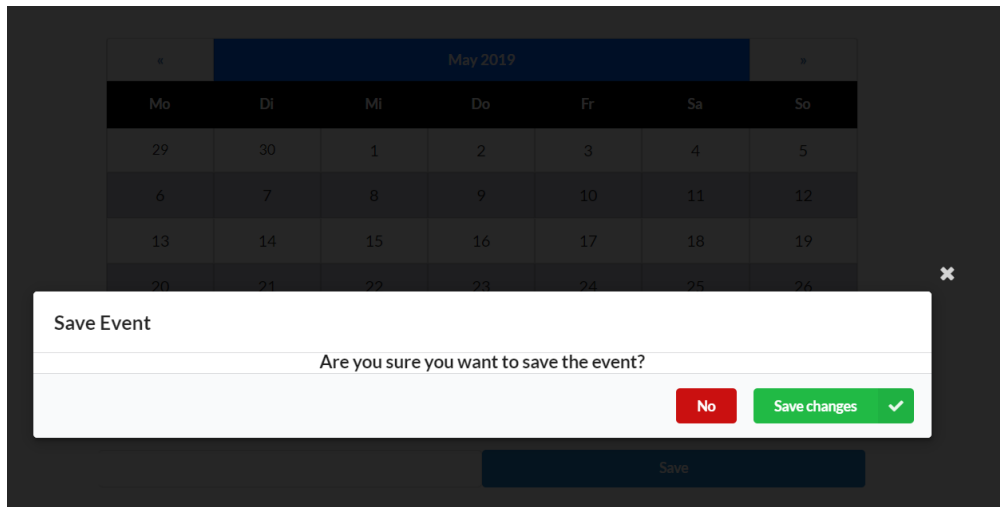
Figure A.2: Semantic UI Modal
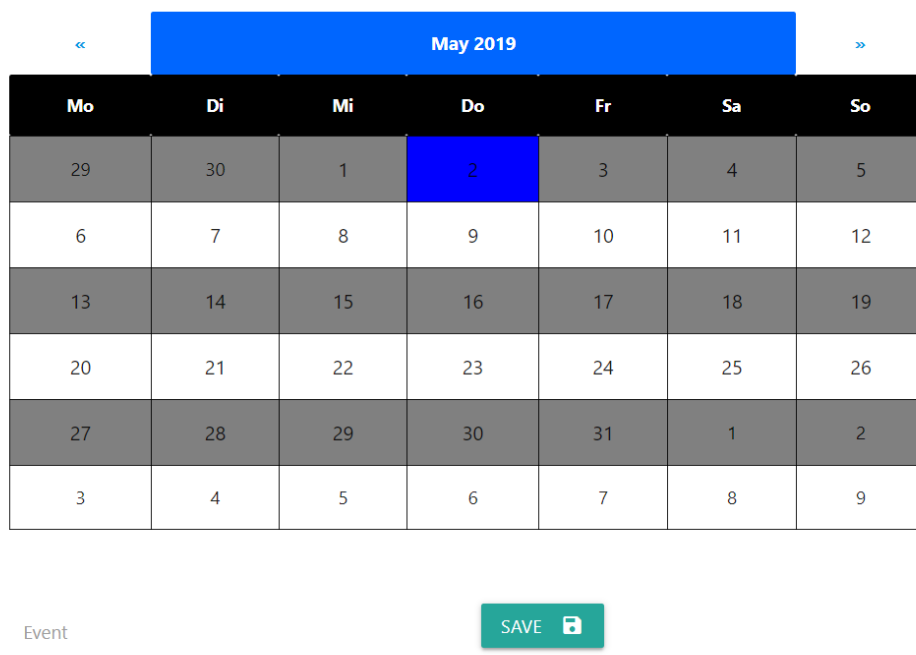
Figure A.3: Calendar with Materialize
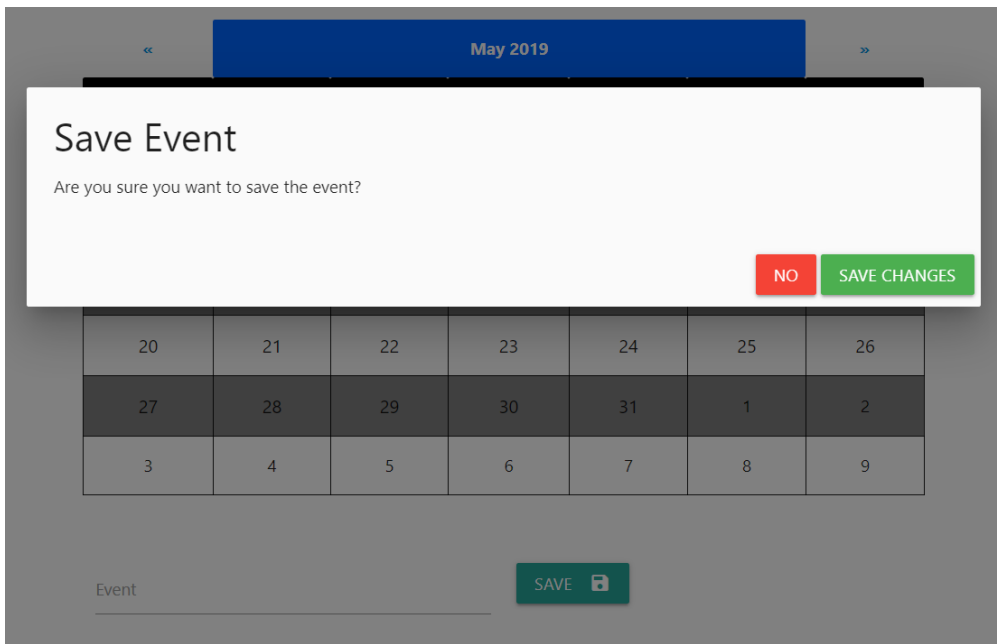
Figure A.4: Materialize Modal
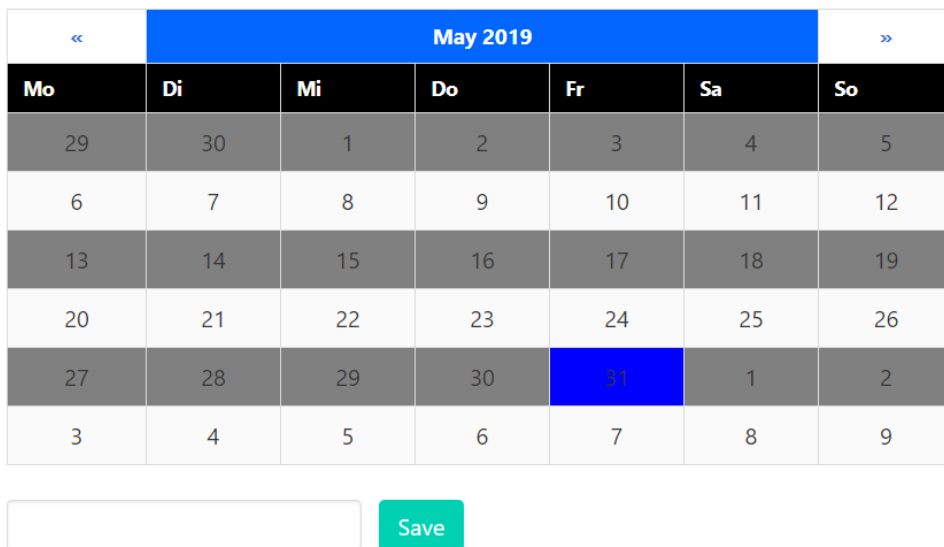
Figure A.5: Calendar with Bulma
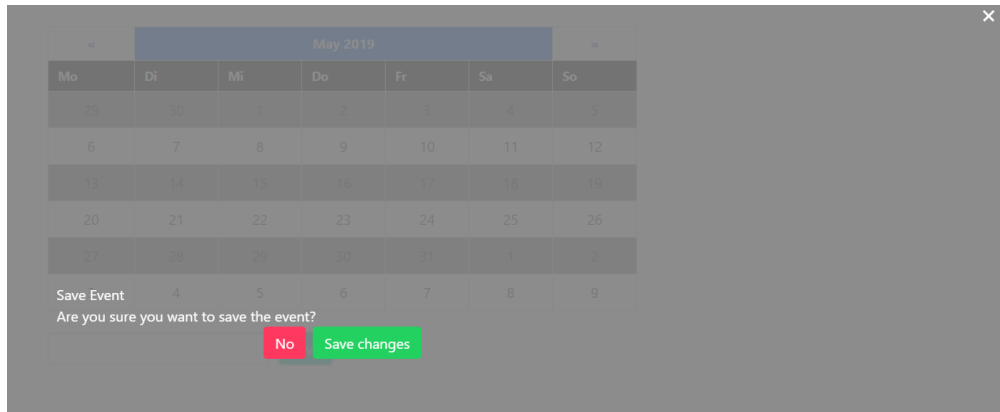
Figure A.6: Bulma Modal

```
 1   Framework Selection:
 2     Technology=0.17613157, Customizing=0.48268921,
 3     Usage=0.09393513, Experience=0.24724409;
 4   Technology: License and costs=0.09658759,
 5     CSS Preprocessors=0.0899404,
 6     Responsive Design=0.12345633,
 7     Performance and Effciency=0.23797043,
 8     Browser version=0.05322835, Functionality=0.39881691;
 9   Customizing: theme support=0.12600678,
10     expandability=0.4579335, configure/add modules=0.41605973;
11   Usage: Ease of integration=0.2, Ease of Installation=0.2,
12     Maintainability=0.6;
13   Experience: Learning curve=0.12903186,
14     Template generator=0.35323309, Skill level=0.1021585,
15     Support=0.41557655;
16   Support: Developer=0.5, Community=0.5;
```

Listing A.1: Hierarchy Definition Text

Table A.1: Alternatives with local weights (Weighted sum method)

1. Alternatives with local weights (Weighted sum method)

Crit/Alt	pGlb	Bootstrap	Semantic UI	Materialize	Bulma
License and costs	0.017012	0.250000	0.250000	0.250000	0.250000
CSS Preprocessors	0.015841	0.595381	0.265308	0.069655	0.069655
Responsive Design	0.021745	0.259546	0.061901	0.172272	0.506281
Performance and Effciency	0.041914	0.250000	0.250000	0.250000	0.250000
Browser version	0.009375	0.243694	0.218973	0.437946	0.099388
Functionality	0.070244	0.516648	0.238211	0.168213	0.076928
theme support	0.060822	0.501674	0.090064	0.269861	0.138401
expandability	0.221040	0.250000	0.250000	0.250000	0.250000
configure/add modules	0.200828	0.250000	0.250000	0.250000	0.250000
Ease of integration	0.018787	0.300000	0.300000	0.300000	0.100000
Ease of Installation	0.018787	0.300000	0.100000	0.300000	0.300000
Maintainability	0.056361	0.386085	0.063148	0.386085	0.164682
Learning curve	0.031902	0.467295	0.160089	0.277181	0.095435
Template generator	0.087335	0.503925	0.088935	0.318206	0.088935
Skill level	0.025258	0.427903	0.284441	0.200479	0.087176
Developer	0.051374	0.390504	0.138066	0.276155	0.195275
Community	0.051374	0.209387	0.346201	0.346201	0.098212
Group Result		0.337940	0.206022	0.264086	0.191952

Table A.2: Alternatives by participant

2. Alternatives by participant

Name	Bootstrap	Semantic UI	Materialize	Bulma	CR max
Group	0.337940	0.206022	0.264086	0.191952	0.068318
Author	0.337940	0.206022	0.264086	0.191952	0.068318

List of Figures

List of Tables

Name: Nadine Harsch   Matriculation number: 879030

Honesty disclaimer

I hereby affirm that I wrote this thesis independently and that I did not use any other sources or tools than the ones specified.

Ulm, ...

Nadine Harsch