

23. Oktober 2020



## **Technische Dokumentation der entwickelten Anwendung – webbasierter Strick-Konfigurator**

### **Der Projektumsetzer:**

Institut für Innovations- und Informationsmanagement GmbH (ifii)  
Magdeburger Straße 50  
14770 Brandenburg an der Havel  
Web: [www.drei-i-m.de](http://www.drei-i-m.de)  
Kontakt: [kontakt@drei-i-m.de](mailto:kontakt@drei-i-m.de)

## Inhaltsverzeichnis

|  |    |
|--|----|
| Abbildungsverzeichnis.....                             | 2  |
| 1. Zusammenfassung .....                               | 3  |
| 2. Einleitung.....                                     | 3  |
| 3. Eingesetzte Technologien .....                      | 3  |
| 3.1. JavaScript.....                                   | 4  |
| 3.2. PHP - Hypertext Preprocessor .....                | 4  |
| 3.3. Canvas-Element (HTML).....                        | 5  |
| 3.4. FabricJS .....                                    | 6  |
| 3.5. JQuery .....                                      | 7  |
| 3.6. SpectrumJS.....                                   | 7  |
| 3.7. FileSaverJS .....                                 | 9  |
| 4. Beschreibung der wichtigsten Funktionalitäten ..... | 10 |
| 4.1. Farben auswählen .....                            | 10 |
| 4.2. Hintergrund/Medium auswählen.....                 | 11 |
| 4.3. Bild hochladen und umwandeln .....                | 11 |
| 4.4. Objekt bewegen.....                               | 11 |
| 4.5. Objekt bearbeiten .....                           | 12 |
| 4.6. Objekt löschen .....                              | 12 |
| 4.7. Text hinzufügen .....                             | 12 |
| 4.8. Text bearbeiten.....                              | 13 |
| 4.9. Bild abspeichern .....                            | 13 |
| 5. Design & Produkt.....                               | 14 |
| 6. Abschluss & Ausblick .....                          | 16 |

## Abbildungsverzeichnis

|  |    |
|--|----|
| Abbildung 1: Architektur Konfigurator .....                            | 5  |
| Abbildung 2: Screenshot - Color-Picker Konfigurator von Spectrum ..... | 7  |
| Abbildung 3: Farben vererben.....                                      | 8  |
| Abbildung 4: Bilder werden bei Farbanpassung umgewandelt.....          | 9  |
| Abbildung 5: Oberfläche Strick-Konfigurator .....                      | 15 |
| Abbildung 6: Icons der Werkzeuge .....                                 | 16 |
| Abbildung 7: Icons der Produkte .....                                  | 16 |

## **1. Zusammenfassung**

Die Herausforderung bestand darin, ein Produkt-Konfigurator zu entwickeln, welcher auf Strickprodukte ausgerichtet ist.

Ein Konfigurator ist im Wesentlichen eine Möglichkeit für Kunden bzw. Nutzer das Design eines Produktes selbst zu bestimmen.

Das Besondere bei Strickwaren ist, dass Textelemente und Bilder direkt im Strickerzeugnis zu integrieren sind und nicht aufgedruckt werden, wie zum Beispiel bei T-Shirts. Dabei verarbeiten Strickmaschinen nur eine begrenzte Anzahl von Farben gleichzeitig. Lädt beispielshalber ein Nutzer ein Foto hoch, um dieses auf seinen Kopfkissenbezug stricken zu lassen, so muss das Foto auf 4 ausgewählte Farben reduziert werden.

## **2. Einleitung**

Im Rahmen eines „Gemeinsam digital“ Projektes wurde vom Institut für Innovations- und Informationsmanagement (ifii) folgender Anwendungsfall erarbeitet:

### **Problemstellung:**

- Hoch individuelle Strickprodukte sollen im Rahmen einer Anwendung von Designern und Selbständigen frei gestaltet und für die Herstellung auf einer Strickmaschine vorbereitet werden.
- Dabei können die Strickmaschinen nicht wie in gewöhnlichen Druckverfahren, auf alle Farben innerhalb eines CMYK-Farbraums, zurückgreifen.
- Strickmaschinen sind in der Auswahl der Farben auf eine bestimmte, meist geringe Anzahl an Farben, eingeschränkt.

### **Aufgabenstellung:**

- Entwicklung einer Anwendung, welche es dem Nutzer erlaubt ein Basisprodukt (Schal, Kissen, Decke, MNS-Maske) zu wählen.
- Darauf die entsprechenden Garn-Farben auszuwählen und ein Bild in den Konfigurator hochzuladen, welches sich automatisch an die 4 ausgewählten Garnfarben anpasst.
- Das Design des Nutzers soll darauf abgespeichert und dem Partnerunternehmen in die Produktionsvorstufe überreicht werden.

## **3. Eingesetzte Technologien**

In diesem Kapitel werden die eingesetzten Technologien beschrieben, die für die Erstellung des Konfigurators ausgewählt wurden. Ziel ist es am Beispiel des der Integration eines Bilds zu vermitteln, wie die verschiedenen Komponenten zusammenspielen. Die Technologien werden zum Beginn kurz beschrieben und darauffolgend der eigene Anwendungsfall aufgeführt.

Seite 3 von 16

### 3.1. JavaScript

JavaScript ist eine Programmiersprache, die es ermöglicht Webseiten und Anwendungen in einem Internetbrowser dynamisch zu gestalten. In erster Linie, um Interaktionen des Benutzers aufzufangen und auszuwerten, spezielle Inhalte zu verändern, Inhalte bei Bedarf nachzuladen oder neu zu generieren. So wird die Bandbreite der Möglichkeiten von HTML (Strukturgerüst einer Webseite) und CSS (das Design der Webseite) erweitert.

JavaScript ist als eine etablierte Programmiersprache Ziel von unabhängigen Entwicklern, großen Organisationen und Konsortien geworden: zum einen um die Sprache immer weiter zu standardisieren, Erweiterungen und Vereinfachungen in neuen Versionen für den produktiven Einsatz zu schaffen und um explizit Bibliotheken und Frameworks für gezieltere Einsätze der Programmiersprache zu ermöglichen. Ganz nach dem Sinnbild – „warum das Rad neu erfinden?“

Beispiele für einen vereinfachten Ansatz von JavaScript werden in den Kapiteln 3.4. und 3.5. aufgeführt.

JavaScript wurde im Konfigurator vor allem dafür genutzt, um die Benutzerschnittstelle der Anwendung zu erstellen. Alle Eingaben und Funktionen für den Nutzer werden über diese Programmiersprache gesteuert und verarbeitet.

### 3.2. PHP - Hypertext Preprocessor

PHP ist neben JavaScript eine weitere sehr beliebte und längst etablierte Programmiersprache. Im Gegensatz zu JavaScript arbeitet die Programmiersprache, allerdings primär serverseitig und erstellt von dort dynamischen Webseiten. Sobald die Webseiten geladen wurden, kann PHP selbst keinen Einfluss mehr auf die Inhalte nehmen. PHP und JavaScript ergänzen sich daher in ihren Anwendungsgebieten und bilden keine Konkurrenz<sup>1</sup>.

PHP wurde vor allem dazu genutzt, um die Umwandlung der Farben durch den Nutzer auf dem Server durchzuführen und entsprechend abzuspeichern. Die neuen Bilder werden mittels eines Links zurück an die Nutzer gesendet und für die weitere Bearbeitung des Designs angezeigt.

---

<sup>1</sup> JavaScript als auch PHP verfügen mittlerweile über Frameworks und Bibliotheken, welche in die Bereiche der jeweiligen Schwerpunkte greifen (PHP – serverseitig & JS – clientseitig). Inhärent bieten die jeweiligen Programmiersprachen jedoch nicht die Möglichkeiten.

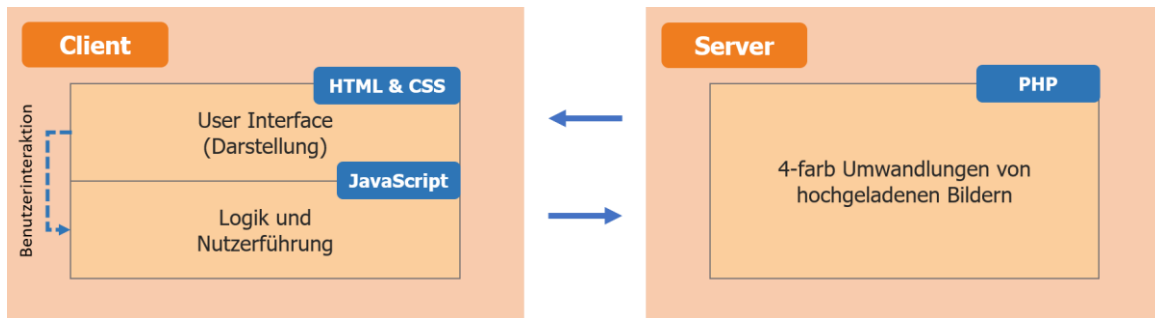


Abbildung 1: Architektur Konfigurator

### 3.3. Canvas-Element (HTML)

Canvas (deutsch: Leinwand) ist ein HTML-Element<sup>2</sup>, welches über ein HTML-Tag direkt eingefügt werden kann. Den Canvas auf einer Webseite sinnvoll zu nutzen ist nicht zu schwierig, setzt aber ein einfaches Verständnis von HTML und JavaScript voraus.

Einige ältere Browser unterstützen das Canvas-Element nicht. Die Standardgröße des Canvas beträgt 300 x 150 Pixel (Breite x Höhe). Diese Größe lässt sich über die Attribute „height“ und „width“ ändern. Um auf dem Canvas-Element zu zeichnen, nutzen Entwickler unterschiedliche Bibliotheken. Es kommt hier ganz darauf an, was für Aufgaben ein Canvas-Element erfüllen soll. Grundsätzlich lassen sich allerdings alle Formen, Farben, Videos und Animationen und sogar Spiele umsetzen.

Im Rahmen des Projektes hat sich das Canvas Element für den Einsatz von Konfiguratoren bewährt. Ein umfassendes Tutorial bietet w3schools.com über folgenden Link an [https://www.w3schools.com/tags/ref\\_canvas.asp](https://www.w3schools.com/tags/ref_canvas.asp)

Beispiel:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid;"></canvas>
```

Ergebnis:



Von nun an muss JavaScript angewendet werden, um die Funktionen des Canvas zu definieren und zu nutzen.

<sup>2</sup> Grundlagen in HTML werden vorausgesetzt.

### 3.4. FabricJS

Fabric.js ist eine leistungsstarke JavaScript HTML5-Canvas-Bibliothek und bietet ein interaktives Objektmodell über dem Canvas-Element. Fabric.js verfügt außerdem über einen Parser für SVG-zu-Canvas (und Canvas-zu-SVG).

Mit Fabric.js können außerdem mittels JavaScript Formen wie Rechtecke, Kreise, Dreiecke und andere Polygone oder komplexere Formen aus vielen Pfaden, als auch Bilder auf dem HTML-Element <canvas> auf einer Webseite erstellt werden. Mit Fabric.js kann anschließend die Größe, Position und Drehung dieser Objekte mit einer Maus verändert werden. Es ist auch möglich, einige der Attribute dieser Objekte wie Farbe, Transparenz, Tiefenposition auf der Webseite zu ändern oder Gruppen dieser Objekte mithilfe der Bibliothek auszuwählen. Mit Fabric.js können eine Vielzahl an Bildern in JavaScript-Daten konvertiert werden, mit denen es in das Canvas-Element eingefügt werden kann. Fabric.js bietet neben der Vielzahl an nutzerfreundlicher Funktionen - welche sofort und inhärent nach dem Einbinden der Bibliothek erfolgt ist, auch sehr nützliche Funktionen, um gewisse „Ereignisse“ aus dem Canvas Element abzufangen. Sinnvolle Funktionen für das Abfangen von nutzergetriebenen Ereignissen des Konfigurators sind:

- Klicks innerhalb des Canvas-Elements
- Aktives (ausgewähltes) Objekt
- Objekt wechseln
- Objekt bewegt
- Objekt verändert
- Objekt erstellt

All diese exemplarischen Ereignissen auf der Nutzerseite können Funktionen auslösen. Diese gilt es jedoch selbstständig in JavaScript zu implementieren.

Fabric.js erleichtert nur den Weg der Entwicklung der angestrebten Anwendung und lässt den Entwicklern einen komplett offenen Gestaltungsspielraum im Gegensatz zu fertigen Lösungen. Nach einer kurzen Einarbeitung in der vorhandenen Dokumentation<sup>3</sup> und anhand der exemplarischen Beispiele<sup>4</sup>, kann hier sehr schnell mit der produktiven Entwicklung gestartet werden. Fortschritte werden sehr schnell erkennbar, jedoch benötigt der Feinschliff der finalen Anwendung erfahrungsgemäß mehr Zeit.

**Fabric.js auf Github** - <https://github.com/fabricjs/fabric.js>

- Stars: 17.000
- Forks: 2.600
- Contributors: 236

<sup>3</sup> Dokumentation Fabric.js - <http://fabricjs.com/docs/>

<sup>4</sup> Beispiele Fabric.js: <http://fabricjs.com/demos/>

### 3.5. JQuery

JQuery ist im wesentlichen vereinfachtes JavaScript. JQuery wurde entwickelt, um effizienter in JavaScript zu programmieren und bietet wesentliche Algorithmen bereits als fertige Funktionen in einer Bibliothek. Durch JQuery ist es möglich HTML Code besser von JavaScript Code zu trennen. Im Rahmen des Projektes wird JQuery benutzt um mit wenigen Anweisungen auf häufig benutzte Funktionen von Effekten wie Hide und Show zuzugreifen.

Auch ist es notwendig JQuery zu nutzen, um das gesamte Farbmanagement abbilden zu können, welches über eine weitere Bibliothek komplett in JQuery angeboten wird.

#### JQuery auf Github - <https://github.com/jquery/jquery>


- Stars: 54.100
- Forks: 19.600
- Contributors:301

### 3.6. SpectrumJS

Spectrum 2.0 ist eine sehr leichtgewichtige Bibliothek, um Farben auf einer Webseite auswählen zu können. Die Erstellung eines „Color-Picker“ kann beispielhaft über einen kleinen Konfigurator direkt auf der Webseite - <https://seballot.github.io/spectrum/>, von Spectrum vorgeschlagen werden. Der Code verändert sich je nach Auswahl und kann anschließend benutzt werden.

#### Configurator

| Type                                   | Palette   | Picker  |
|--|---|---|
| <input checked="" type="radio"/> Color | <input type="checkbox"/> Show Palette                 | <input type="checkbox"/> Show Input                 |
| <input type="radio"/> Text             | <input checked="" type="checkbox"/> Show Palette Only | <input type="checkbox"/> Show Initial Color         |
| <input type="radio"/> Text + Color     | <input type="checkbox"/> Toggle Picker Button         | <input checked="" type="checkbox"/> Transparency    |
| <input type="radio"/> Flat             | <input type="checkbox"/> Hide After Palette Select    | <input type="checkbox"/> Show Cancel/Submit Buttons |
|  |   | <input type="checkbox"/> Clear Button               |



```
<input id="color-picker" value='#276cb8' />
```

```
$('#color-picker').spectrum({  
  type: "color",  
  showPalette: "false",  
  showPaletteOnly: "true",  
  showButtons: "false",  
  allowEmpty: "false"  
});
```

Abbildung 2: Screenshot - Color-Picker Konfigurator von Spectrum

Die individuelle Farbpalette der Color-Picker kann leicht angepasst werden. Im Rahmen des Projektes sind insgesamt 16 Farben vorgegeben worden, die wie im folgenden Beispiel umgesetzt worden sind.

```

1. $("#primaerfarbe").spectrum({
2.   preferredFormat: "hex",
3.   showPaletteOnly: true,
4.   showPalette:true,
5.   color: '#b4b339',
6.   palette: [
7.     ['#000000', '#f9f9ef', '#d4d6d9', '#505558'],
8.     ['#c6a15c', '#3c2e22', '#b22b38', '#ce3a2c'],
9.     ['#ec5528', '#f2b743', '#b4b339', '#0a2343'],
10.    ['#143b93', '#8ab4e4', '#db93a2', '#ebacfb'],
11.  ]
12. });

```

In Zeile 6 unter der „palette:“ bis 11 sind die auswählbaren Farben zu definieren. Hierzu ist es nicht wichtig in welchem Format die Farben angegeben werden. Im Rahmen des Konfigurators haben wir uns für Hexwerte entschieden, da hier die Implementierung der 4-farb Umwandlung von Bildern auf Grundlage von Hexwerten bereits implementiert wurde. Denkbar wären allerdings auch die folgenden anderen Formate wie RGB, HSV oder einfach die jeweiligen englische Bezeichnungen (in CSS) der Farb-Namen.

Im Rahmen des Farbmanagements des Konfigurators, wurden noch weitere Anpassungen weiterer Farbpaletten übernommen. Innerhalb der Anwendung sollen die Nutzer insgesamt ihren Pool an 4 Farben über jeweils einen Color-Picker bestimmen.

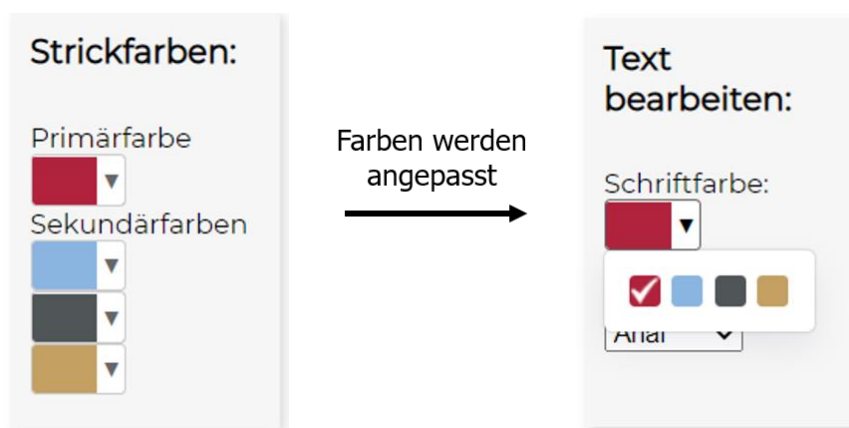


Abbildung 3: Farben vererben



Hierbei wird von nun an die Farbauswahl der „Strickfarben“ das zentrale Steuerungselement für die Nutzer der Anwendung. Bei der Veränderung einer Farbe wird zum einen die ausgewählte Farbpalette an die Farbpalette der Textbearbeitung vererbt, als auch das neue Laden eines aktiven Bildes über den Server angestoßen.



Abbildung 4: Bilder werden bei Farbanpassung umgewandelt

**Spectrum auf Github** - <https://github.com/seballot/spectrum>

- Stars: 49
- Forks: 555
- Contributors: -

### 3.7. FileSaverJS

FileSaver.js ist eine leichtgewichtige Lösung zum Speichern von Dateien auf der Clientseite und eignet sich für Webanwendungen, die Dateien und Bilder auf dem Client generieren. Wenn die Datei jedoch vom Server stammen soll, wird diese Bibliothek nicht funktionieren.

**Ab speichern der Inhalte eines Canvas:**

1. `var canvas = document.getElementById("my-canvas");`
2. `canvas.toBlob(function(blob) {`
3. `saveAs(blob, "name.png");`
4. `});`

Sollten sehr große Dateien abgespeichert werden, sollte **StreamSaver.js** in Betracht gezogen werden.

FileSaver.js auf Github - <https://github.com/eligrey/FileSaver.js/>

- Stars: 15.200
- Forks: 3.600
- Contributors: 56

## 4. Beschreibung der wichtigsten Funktionalitäten

In diesem Kapitel werden die wichtigsten Funktionalitäten in einer Art Pseudocode bzw. in der Beschreibungssprache **Gherkin**<sup>5</sup> beschrieben. Gherkin wird in verschiedenen Behavior-Driven-Development-Implementierungen (BDD – deutsch: verhaltensgetriebene Software-entwicklung) verwendet und ist dafür geschaffen, um Stakeholder und auch nicht-technische Teammitglieder in die Softwaretests mit einzubeziehen. Diese Sprache gibt es sowohl mit englischen Schlüsselwörtern (Given, When, Then, And, ...), deutschen (Gegeben, Wenn, Dann, Und, ...) und in weiteren Sprachen.

Gherkin kann einerseits wie eine Checkliste auf Papier abgearbeitet werden oder auch durch kollaborative Softwareunterstützung, die Gherkin kompatibel sind, wie zum Beispiel Cucumber. Die Features helfen den Entwicklern bei der Umsetzung der Software, helfen den Testern in den Softwaretests die Anforderungen zu überprüfen und schränken die Fehlerquellen während des Debuggings ein.

Die folgenden Unterkapitel beinhalten jeweils die Beschreibungen der umgesetzten Kernfeatures in dazu passender Gherkin-Syntax.

### 4.1. Farben auswählen

**Funktionalität:** Farbauswahl

Um die gewünschten vier Farben des Strick Erzeugnisses zu Bestimmen, müssen im Vorfeld die Farben vom Nutzer gewählt werden. Diese Funktion muss durch eine Farbauswahl an vorgegebenen Farben erfolgen.

**Szenario:** Auswahl der Primärfarbe

**Angenommen** ich möchte eine rote Farbe auswählen  
**Wenn** ich auf die Farbauswahl der "**Primärfarbe**" klicke  
**Dann** sehe ich eine Farbpalette von insgesamt 16 Farben  
**Wenn** ich auf die rote Farbe drücke  
**Dann** sollte sich das Farbfeld rot einfärben

<sup>5</sup> <https://docs.behat.org/en/v2.5/guides/1.gherkin.html>

#### 4.2. Hintergrund/Medium auswählen

**Funktionalität:** Hintergrund/Medium des Canvas verändern  
Der Hintergrund kann so angepasst werden, dass das Designen von Objekten direkt auf dem Strickerzeugnis geschehen kann.

**Szenario:** Auswahl eines beliebigen Produktes

**Gegeben sei** eine ausgewählte Farbe als "**Primärfarbe**"  
**Wenn** ich auf ein Produkt klicke  
**Dann** sehe ich das gewählte Produkt in der Farbe der "**Primärfarbe**"

#### 4.3. Bild hochladen und umwandeln

**Funktionalität:** Nutzer können ein Bild (PNG) in der Anwendung uploaden, welches dann in die ausgewählten vier Farben umgewandelt und für den Nutzer im Canvas angezeigt wird

**Szenario:** Bild hochladen

**Angenommen** ich möchte ein bestimmtes Bild als PNG uploaden  
**Gegeben sei** dass alle vier Farben ausgewählt sind  
**Wenn** ich auf "**Bild hinzufügen**" klicke  
**Dann** habe ich die Möglichkeit ein Bild auszuwählen  
**Dann** wird das ausgewählte Bild in den Konfigurator geladen  
**Und** das Bild wurde in die ausgewählten vier Farben umgewandelt  
**Aber** wird eine Farbe geändert  
**Dann** wird das Bild erneut in die 4 Farben umgewandelt  
**Und** an die alte Position wieder eingefügt

#### 4.4. Objekt bewegen

**Funktionalität:** Nutzer können alle Objekte im Konfigurator beliebig verschieben

**Szenario:** Objekt verschieben

**Gegeben sei** dass ein Objekt im Konfigurator ist  
**Und** ich halte das Objekt mit meinem Mauszeiger gedrückt  
**Dann** habe ich die Möglichkeit das Objekt per Mausbewegung zu verschieben  
**Wenn** ich den Mausklick beende  
**Dann** lässt sich das Objekt nicht mehr verschieben  
**Aber** das Objekt ist noch aktiv



## 4.5. Objekt bearbeiten

**Funktionalität:** Objekte können durch die Nutzer bearbeitet werden

**Szenario:** Objekt bearbeiten

**Gegeben sei** dass ein Objekt aktiv ist

**Wenn** ich mit der Maus auf "**Spiegeln**" klicke

**Dann** hat sich das Objekt gespiegelt

**Wenn** ich mit der Maus auf "**Drehen**" klicke

**Dann** hat sich das Objekt um 90 Grad gedreht

**Wenn** ich mit der Maus auf "**Kopieren**" klicke

**Dann** hat sich nichts verändert

**Wenn** ich mit der Maus auf "**Einfügen**" klicke

**Dann** sehe ich das Objekt ein zweites Mal im Konfigurator

**Aber** es sieht genauso aus wie ich es vorher bearbeitet habe

## 4.6. Objekt löschen

**Funktionalität:** Objekte können durch die Nutzer wieder entfernt werden

**Szenario:** Objekt löschen

**Angenommen** ich möchte ein Objekt löschen

**Gegeben sei** dass ein Objekt im Konfigurator ist

**Wenn** ich mit der Maus auf das Objekt klicke

**Dann** sehe ich eine Umrandung des Objektes

**Wenn** ich auf den Button "**Löschen**" klicke

**Oder** die Taste "**Entf**" drücke

**Dann** ist das Objekt nicht mehr im Konfigurator

## 4.7. Text hinzufügen

**Funktionalität:** Nutzer können einen Text hinzufügen und aus einer der vier vorher angepassten Farben auswählen

**Szenario:** Text hinzufügen

**Angenommen** ich möchte ein Text hinzufügen

**Gegeben sei** dass alle vier Farben ausgewählt worden sind

**Wenn** ich mit der Maus auf "**Text hinzufügen**" klicke

**Dann** öffnet sich ein Fenster

**Und** darin kann ich meine Anweisung bestätigen

**Und** ich sehe anschließend ein Textfeld in der "**Primärfarbe**"

**Dann** klicke ich per "**Doppelklick**" auf das Textfeld

**Und** kann das Textfeld bearbeiten

#### 4.8. Text bearbeiten

**Funktionalität:** Nutzer können die Schriftart ändern und die Farbe aus den vorher definierten vier Farben anpassen.

**Szenario:** Textfarbe anpassen

**Angenommen** ich möchte meine Textfarbe anpassen  
**Gegeben sei** dass auf der linken Seite nur der Bereich **"Text bearbeiten"** angezeigt wird  
**Gegeben sei** dass das Textfeld aktiv ist  
**Wenn** ich mit der Maus auf die **"Farbauswahl"** klicke  
**Dann** öffnet sich eine Farbpalette  
**Und** zeigt mir nur meine vorher definierten vier Farben zur Auswahl an  
**Und** darin kann ich meine gewünschte Farbe auswählen  
**Dann** verändert sich die Farbe des aktiven Textfeldes

**Szenario:** Schriftart anpassen

**Angenommen** ich möchte meine Schriftart anpassen  
**Gegeben sei** dass auf der linken Seite nur der Bereich **"Text bearbeiten"** angezeigt wird  
**Gegeben sei** dass das Textfeld aktiv ist  
**Wenn** ich mit der Maus auf die **"Schriftart"** klicke  
**Dann** öffnet sich ein Dropdown mit den Schriftarten  
**Und** darin kann ich meine gewünschte Schriftart auswählen  
**Dann** verändert sich die Schriftart des aktiven Textfeldes

#### 4.9. Bild abspeichern

**Funktionalität:** Nutzer können das Design abspeichern und finden es als Bild anschließend Lokal auf der Festplatte.

**Szenario:** Bild abspeichern

**Angenommen** ich habe mein Design fertig gestellt  
**Gegeben sei** dass die Objekte bereits auf dem Konfigurator platziert sind  
**Gegeben sei** dass ein Hintergrund ausgewählt wurde  
**Wenn** ich auf **"Design speichern"** klicke  
**Dann** öffnet sich ein Fenster wo ich das Design benennen kann  
**Dann** kann ich das Design abspeichern  
**Und** ich sehe je nach Browser im unteren Bereich ein Download  
**Dann** wechsele ich in das Downloadverzeichnis  
**Und** kann das Bild öffnen und mein Design anschauen  
**Aber** der Hintergrund wurde nicht mit abgespeichert

## 5. Design & Produkt

Es wurden Bildbearbeitungstools und andere Konfigurationstools im Web hingehend zur Nutzerfreundlichkeit evaluiert und eine erste ausschließlich grafische Visualisierung aufgesetzt. Diese wurde dann vor der Umsetzung getestet, reviewt und fortlaufend optimiert. Dafür wurden verschiedene User **Interface Elemente** und Icons aus bestehenden **Design-Frameworks** evaluiert.

Interface Elemente sind im wesentlichen nutzergesteuerte Eingabe- und Interaktionsmöglichkeiten, um die Anwendung zu bedienen. Buttons, Hervorhebungen, klickbare Bilder und Veränderungen der Oberfläche können dazu dienen dem Nutzer die Oberfläche zu erklären und entsprechend zu bedienen.

Im Bereich der nutzerfreundlichen Gestaltung von Software, gibt es eine Vielzahl an wiederverwendbaren Elementen aus bewährten Open-Source Projekten. Dabei arbeiten sehr viele Entwickler als auch Designer zusammen, um die gesammelten Erfahrungen in sogenannte Design-Frameworks aufzunehmen und verfügbar zu machen. Die Design-Frameworks beinhalten bereits vorgefertigte Designs, Icons und Buttons, welche Entwickler lediglich in die Anwendung integrieren müssen.

Design Frameworks werden von einer Vielzahl großer renommierter Plattformen eingesetzt und gewährleisten einen Wiedererkennungswert in der Oberflächenbedienung.

**Beispielhafte Design-Frameworks sind:**

- Bootstrap
- UI Kit
- Material Design
- Kube UI

Die Oberfläche des Strick-Konfigurators wurde mittels eigens erstellter Design Elemente in Anlehnung des Design-Frameworks **UI-Kit** entwickelt.



Abbildung 5: Oberfläche Strick-Konfigurator

Erläuterungen zur Abbildung 5:

1. Grafischer Designer
2. Farbmanagement
3. Werkzeuge für weitere Bildbearbeitungsmöglichkeiten
4. Mögliche Produkte / Strickerzeugnisse
5. Objekte (Bilder und Texte) hinzufügen
6. Design abspeichern
7. Rückgängig / Wiederholen

Die Icons innerhalb des Konfigurators wurden ergänzend designed und dem speziellen ifii UI-Kit hinzugefügt.

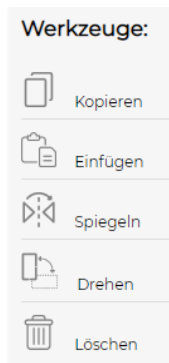


Abbildung 6: Icons der Werkzeuge

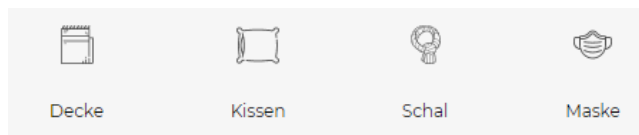


Abbildung 7: Icons der Produkte

Nach mehrmaligen Testen wurde zu jedem Icon auf der Oberfläche ein Wort zur Beschreibung der jeweiligen Schaltflächen hinzugefügt, um die Verständlichkeit für die Nutzer zu optimieren.

## 6. Abschluss

In der Dokumentation wurden vor allem die technischen Aspekte des webbasierten Konfigurators betrachtet und die verschiedenen Technologien aufgeführt, welche für die Umsetzung herangezogen worden sind. Die zu erfüllenden Anforderungen der wichtigsten Features sollen den Entwicklern als auch weiteren Stakeholdern eine Orientierung bieten, welchen Umfang eine mögliche eigene Entwicklung der Anwendung hat. Auch können sich Designer ein Bild von einer möglichen Oberfläche anhand dieser technischen Dokumentation machen.

Bei Interesse an der Lösung oder bei Rückfragen melden Sie sich gerne beim Institut für Innovations- und Informationsmanagement GmbH (ifii):

Tel.: 03381/355843  
E-Mail: [kontakt@drei-i-m.de](mailto:kontakt@drei-i-m.de)