Fakultät Medien

Alexander Rothe

Entwicklung eines Programms zum Erstellen von Kleindrucksachen im PDF-Format in Browsern

Bachelorarbeit

Hochschule Mittweida (FH) – University of Applied Sciences

Leipzig 2010

Fakultät Medien

Alexander Rothe

Entwicklung eines Programms zum Erstellen von Kleindrucksachen im PDF-Format in Browsern

eingereicht als Bachelorarbeit

Hochschule Mittweida (FH) – University of Applied Sciences

Erstprüfer: Prof. Dr. phil. Ludwig Hilmer
Zweitprüfer: Dipl.-Ing. (FH) Greim

Leipzig 2010

Bibliographische Beschreibung und Referat

Alexander Rothe

Entwicklung eines Programms zum Erstellen von Kleindrucksachen im PDF-Format in Browsern.

Mittweida, Hochschule Mittweida (FH),
Fakultät Medien, Bachelorarbeit

2010 – 71 Seiten

Die hier vorliegende Bachelorarbeit beschäftigt sich mit der Entwicklung eines Programms zum Erstellen von Kleindrucksachen im PDF-Format im Browsern. Am Anfang wird festgelegt, welchen Funktionsumfang das Programm enthalten soll. Daraufhin wird die beste Technik zur Umsetzung des Programms gesucht und die Probleme betrachtet die bei der Umsetzung entstehen können. Anschließend wir die praktische Umsetzung des Programms beschrieben.

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

Ziel dieser Bachelorarbeit ist es, ein Programm zum Gestalten und Bestellen von Visitenkarten und Stempeln im Internet zu erstellen. Oberstes Ziel dabei ist es, dem Benutzer die Möglichkeit zu geben, das Endergebnis so genau wie möglich abzuschätzen. Während der Erstellung des Produktes soll die Vorschau so exakt wie Möglich dem späteren Endergebnis gleichen.

Dies ist nicht in allen Bereichen ohne Einschränkungen möglich. Eine exakte Übereinstimmung der angezeigten Vorschau am Monitor und dem später gedruckten Produkt ist nicht realisierbar. Zahlreiche Faktoren sind nicht vorhersehbar oder zu beeinflussen. Jeder Monitor weist eine unterschiedliche Charakteristik in der Farbwiedergabe und Helligkeit auf. Da sich das Programm nicht an Profis wendet, ist davon auszugehen, dass die Benutzer nicht über einen kalibrierten Monitor zur exakten Farbwiedergabe verfügen. Detailliert auf die erwarteten Probleme wird im Kapitel 3 eingegangen.

Der Benutzer soll die Möglichkeit bekommen, den Text frei zu formatieren und eigene Schriftarten verwenden zu können. Zusätzlich soll es möglich sein, einen eigenen Hintergrund einfügen zu können. Die Wahl der Produktgröße ist frei bzw. unterliegt den Möglichkeiten, die der Druckanbieter zur Verfügung stellt. Die genauen Funktionen und ihre Umsetzung werden im Kapitel 3 beschrieben.

Im Kapitel 3.4 und 3.5 werden verschiedene Techniken verglichen mit denen das Programm realisiert werden kann. Die größte Herausforderung an die verwendete Technik ist, dass sie in jedem aktuellen Browser funktioniert und gleiche Ergebnisse liefert. Da es sich um Druckdaten handelt, müssen Vektordaten erstellt werden. Vektordaten sind im Browser nur sehr schwer darzustellen da die HTML Spezifikation nur Pixelbilder zulässt. Eine Möglichkeit um Vektordaten anzuzeigen, sind SVG Dateien. Die SVG Unterstützung ist in Browsern

noch nicht weit verbreitet[1], so dass nicht jeder Internetbenutzer SVG mit seinem verwendeten Browser anzeigen kann oder die Darstellung abweicht. Es existieren allerdings Javascript Bibliotheken, die Vektorgrafiken mit Javascript Syntax beschreiben und dann während des Renderns in ein browserkompatibles Format wie SVG oder VML umwandeln[2]. Eine mögliche Alternative wäre eine Umsetzung mittels Flash.

1 http://www.webmasterpro.de/portal/webanalyse-technologien.html ,30.10.2009

2 http://raphaeljs.com/, 30.10.2009

1 Funktionsumfang

Mit Hilfe des Programms soll es möglich sein, im Browser eine Visitenkarte oder ein ähnliches Kleindruckobjekt zu gestalten. Nach dem Abschluss des Gestaltungsvorgangs wird das erzeugte Dokument in einer Datenbank abgespeichert, um später eine weitere Bearbeitung oder Umwandlung in ein anderes Format als PDF zu ermöglichen.

Während des Gestaltens kann jeder Zeit eine PDF als Vorschau erzeugt werden. Der gesamte Gestaltungsprozess soll so ablaufen, dass der Benutzer ständig sicher sein kann, dass die Anzeige auf dem Bildschirm auch der später Druckausgabe entspricht. Die Gestaltung folgt also dem Prinzip des WYSIWYG. Dafür sollen seitens des Anwenders keine besonderen Kenntnisse erforderlich sein. Weitere Erklärungen zu dem Prinzip des WYSIWYG folgen im Kapitel 4.2.

Zu den Gestaltungsmöglichkeiten gehören das Einfügen eines Hintergrundbildes, das Hinzufügen beliebiger Textfelder und einer eigenen Grafik. Der eingefügte Text kann mit den aus Textverarbeitungsprogrammen bekannten Werkzeugen formatiert werden. Die Platzierung der Textfelder soll frei mit der Maus erfolgen können.

Die Anwendung soll in allen verbreiteten Browser lauffähig sein und in jedem Browser dieselben Ergebnisse liefern. Auf die Installation zusätzlicher Plugins soll wenn möglich verzichtet werden, um eine möglichst große Anzahl von Nutzern erreichen zu können.

Der Benutzer soll über keine speziellen Vorkenntnisse verfügen müssen, um das Programm bedienen zu können. Es wird versucht, die Bedienung so leicht verständlich wie möglich zu gestalten.

Für den Betreiber soll es leicht möglich sein, das Programm an die jeweiligen Einsatzbedingungen anzupassen. Es sollte, wann immer möglich, freie Software zum Einsatz kommen, so dass weder beim Einsatz noch beim Bearbeiten dieser Lizenzgebühren zu entrichten sind.

2 Aufbau und Funktionsweise

Das Programm wird in drei Teile aufgeteilt:

A (Laden) – Einstellen der Dokumenteigenschaften und das Laden von bereits vorhandenen Daten.

B (Editor) – Gestalten des Dokumentes (der eigentliche Editor)

C (Vorschau und Abspeichern) – Generieren der PDF und Abspeichern der Daten in die Datenbank

Diese drei Teile sollen möglichst unabhängig voneinander laufen, so dass es möglich ist, sie an die jeweilige Einsatzmöglichkeit anzupassen oder auch wegzulassen wenn sie nicht benötigt werden. Dadurch kann das gesamte System klein und an die jeweiligen Einsatzzwecke anpassbar gehalten werden.

2.1 Teil A

Im ersten Programmteil gibt der Benutzer die Eigenschaften des Dokumentes wie Höhe und Breite an. Danach kann automatisch einen Vorschlag erteilt werden, wie die Textfelder auf der Visitenkarte angeordnet werden können. Diese Anordnung kann vom Benutzer übernommen oder wieder verworfen werden.

Im Teil A kann vom Anbieter festgelegt werden, welche Größen verfügbar sind und ob er schon Vorlagen für die Benutzer bereitstellen will.

Die Größe und wenn gewünscht die Positionen der vorgeschlagenen Textboxen wird an den Programmteil B übergeben.

2.2 Teil B

Der Programmteil B ist der aufwändigste Teil des Programms. In diesem findet die eigentliche Gestaltungsarbeit des Nutzers statt. Um den Anspruch WYSIWYG gerecht zu werden erfolgt die Darstellung des Dokumentes durch Vektoren. Die Darstellung muss in Echtzeit erfolgen können, so dass der Benutzer ein Element mit der Maus anfassen und sofort verschieben kann, ohne dass Wartezeiten auftreten.

Ein weiterer wichtiger Punkt ist, dass jeder Benutzer das Programm benutzen kann, ohne zusätzliche Software, Plugins, oder einen anderen Browser verwenden zu müssen. Es muss also darauf geachtet werden, dass die verwendeten Techniken weit verbreitet sind.

Der Programm Teil B wird auf dem Rechner des Benutzers ausgeführt, um die Geschwindigkeit zu erhöhen. Dies vermeidet ebenfalls Probleme, wenn zu viele Benutzer gleichzeitig das Programm verwenden. Da das Programm nicht auf den Servern des Anbieters, sondern lokal auf dem Rechner des Benutzers läuft, wird eine hohe Serverlast vermieden.

Programmteil B wird nur auf dem Client ausgeführt. Es ist dadurch leicht möglich, eine Anwendung daraus zu entwickeln, die ohne Internetverbindung funktioniert. Einzig die Speicherung und Übergabe der Daten müsste in eine Datei erfolgen, die dann erst später über ein Webinterface zum Anbieter hochgeladen wird oder durch das Programm übertragen wird, sobald eine Internetverbindung besteht.

2.3 Teil C

Programmteil C erzeugt eine Vorschau PDF und speichert die Daten in einer Datenbank ab.

Der Benutzer kann zu jeder Zeit eine Vorschau PDF erzeugen lassen. Nach dem Beenden des Bearbeitens werden die Daten in der Datenbank abgespeichert und die fertige PDF auf dem Server des Anbieters abgelegt.

3 Vorbetrachtungen

3.1 Webstandards

Das World Wide Web Consortium (W3C) ist ein Gremium zur Standardisierung der das World Wide Web betreffenden Techniken. Bei das W3C handelt es sich allerdings nicht um eine staatlich anerkannte Organisation. Dadurch ist das W3C nicht berechtigt, ISO-Normen festzulegen. [3]

Allerdings hat das W3C viele de-facto Standards hervorgebracht. Zum Beispiel HTML, XHTML, XML, CSS, SVG, und RSS. Das W3C nennt seine Standards W3C-Recommendations, also W3C-Empfehlungen, um dem nicht offiziellen Status zu entsprechen.[4]

Die heute verwendeten Standards zur Strukturierung von Webseiten HTML 4.01 und XHTML 1.0 sind unter http://www.w3.org/TR/html401/ und http://www.w3.org/TR/xhtml1/ spezifiziert.

Zur optischen Gestaltung und Trennung von Inhalt und Gestaltung wird CSS verwendet. CSS liegt aktuell in der Version 2.1 vor und ist unter http://www.w3.org/TR/CSS21/ spezifiziert.[5]

Die korrekte Interpretation dieser W3C Standards durch aktuelle Browser lässt sich durch den vom Web Standards Project entwickelten Acid Test überprüfen. Der aktuelle Acid3 Test prüft die Browser vor allem auf ihre Interaktivität und testen dementsprechend primär DOM Level 2 und EMCAScript (JavaScript). Weiterhin prüft der Test auch die SVG und XML Unterstützung der Browser. [6]

3 http://www.w3c.de/about/overview.html, 04.11.2009
4 http://de.wikipedia.org/wiki/World_Wide_Web_Consortium, 04.11.2009
5 http://de.wikipedia.org/wiki/Cascading_Style_Sheets, 04.11.2009
6 http://www.webstandards.org/action/acid3, 04.11.2009

Zum derzeitigen Zeitpunkt (04.11.2009) absolvieren den Acid3 nur drei Browser fehlerfrei. Nur zwei davon sind schon stabile und veröffentlichte Versionen. Der Opera Brower absolvierte den Test mit der am 01.09.2009 erschienen Version 10.0 fehlerfrei, der Safari Browser mit der am 08.06.2009 erschienen Version 4.0. Die Vorabversion 4.0.206.0 des Google Chrome Browsers absolvierte den Test ebenfalls erfolgreich. Alle anderen Browser stellen den Acid3 Test nicht standardkornform dar. [7]

Diese drei Browser haben zusammen nur einen Marktanteil von 3,9% (Stand 04.11.2009)[8]. Durch die geringe Verbreitung der Browser, die die Webstandards so interpretieren wie sie von der W3C vorgesehen sind, kann nicht davon ausgegangen werden, dass die erstellte Webseite oder das im Rahmen dieser Arbeit entstehende Programm bei jedem Benutzer gleich wiedergeben wird. Um trotzdem zu gewährleisten, dass das Prinzip des WYSIWYG erfolgreich umgesetzt werden kann, müssen Vorbereitungen getroffen werden, die die Fehler der verschiedenen Browser ausgleichen können. Es muss also eine Technik gefunden werden, die trotz der Unzulänglichkeiten in der Implementierung der Webstandards dafür sorgt, dass jeder Benutzer, egal welchen Browser er verwendet, dasselbe Ergebnis vor sich sieht.

7 http://de.wikipedia.org/wiki/Acid_%28Browsertests%29#Kompatibilit.C3.A4t_von_Anwendungen_2, 04.11.2009

8 http://www.w3schools.com/browsers/browsers_opera.asp Verbreitung Opera nach Versionsnummer, http://www.w3schools.com/browsers/browsers_safari.asp Verbreitung Safari nach Versionsnummer, http://www.w3schools.com/browsers/browsers_chrome.asp Verbreitung Chrome nach Versionummerr alle abgerufen am 04.11.2009

3.2 WYSIWYG

Größtes Augenmerk soll beim Programmieren darauf gelegt werden, dass das WYSIWYG Prinzip bestmöglich umgesetzt wird. WYSIWYG ist ein Akronym welches ausgeschrieben "**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et" oder auf deutsch („*Was du siehst, ist [das,] was du bekommst.*“) bedeutet.[9]

Echtes WYSIWYG bedeutet, dass ein Dokument während der Bearbeitung am Bildschirm genauso angezeigt wird, wie es bei der Ausgabe über ein anderes Gerät aussieht. [10]

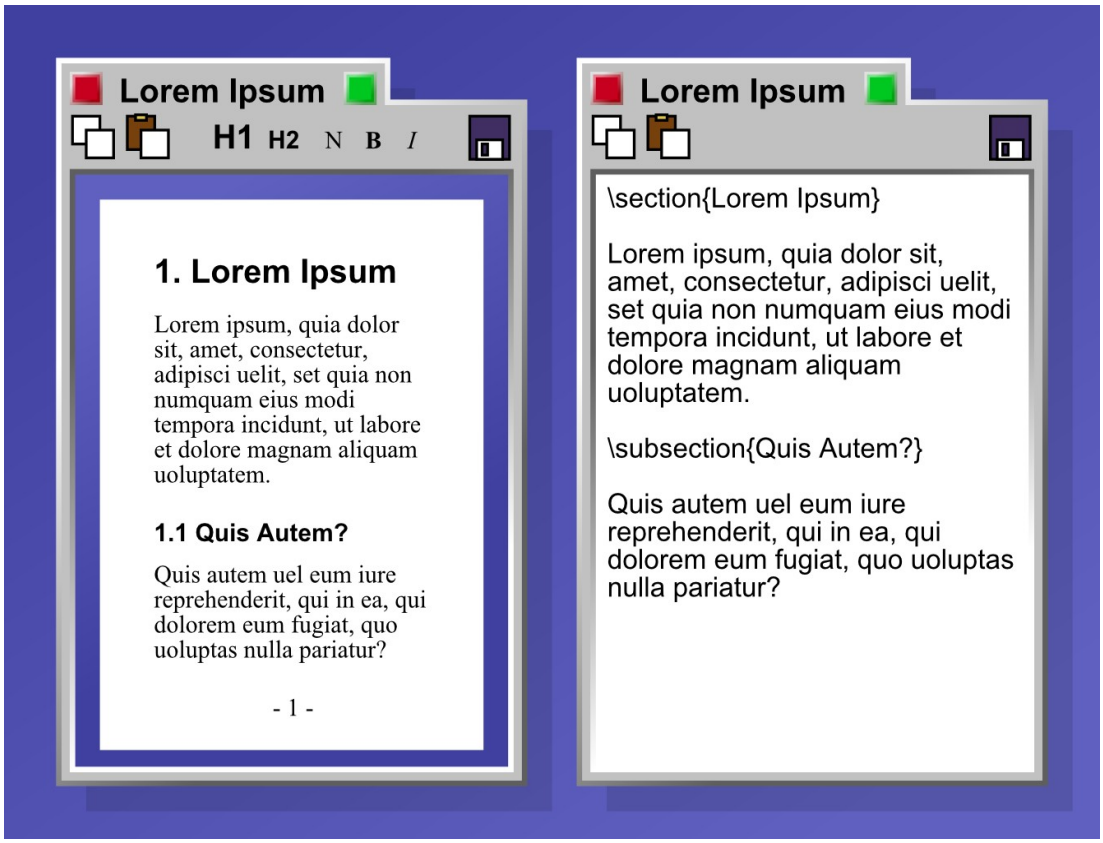


Abbildung 1: Vergleich eines WYSIWYG Programms mit einem nicht WYSIWYG Programm

9 http://www.askoxford.com/concise_oed/wysiwyg?view=uk 04.11.2009
10 http://www.askoxford.com/concise_oed/wysiwyg?view=uk 04.11.2009

Der bekannteste Vertreter der WYSIWYG Programme ist das Textverarbeitungsprogramm Microsoft Word. Der Benutzer schreibt einen Text und formatiert ihn. Der Ausdruck entspricht dann dem am Computer angezeigten Dokument. Der Benutzer kann sich dadurch sicher sein, dass was er geschrieben und formatiert hat, auch so gedruckt, wird wie es angezeigt wird.

Abbildung 1 stellt die unterschiedliche Arbeitsweise eines WYSIWYG Programms (links) und LaTex (rechts), welches ein sogenanntes logisches Markup (Auszeichnung) verwendet, gegenüber. Es ist zu erkennen, dass auf der linken Seite das Endergebnis schon während des Erstellens sichtbar ist. Die Überschrift wird formatiert, indem man den Text markiert und auf die Schaltfläche H1 klickt.

Auf der rechten Seite ist die Vorgehensweise anders. Hier wird kein WYSIWYG verwendet. Die Formatierung der Überschrift erfolgt, indem man um die die Überschrift den Befehl \section{} schreibt. Das Ergebnisse wird nicht sofort sichtbar, sondern erst wenn das Dokument als PDF gerendert wird.

Es wird deutlich, dass die WYSIWYG Variante während der Bearbeitung schon den eingestellten Rand, die gewählte Schriftart und die Seitenzahl anzeigt.

Eine Einschränkung besteht allerdings bei der Größe des Dokumentes. Da viele Monitore eine unterschiedliche Auflösung besitzen, ist es nicht möglich die exakte Schriftgröße abzuschätzen. Dies ist nur möglich, wenn der Benutzer durch vergrößern oder verkleinern der Ansicht diese Ansicht so anpasst, dass das erstellte Dokument auf dem Monitor so groß angezeigt wird, wie es später gedruckt wird.

WYSIWYG findet auch bei der Gestaltung von Internetseiten Verwendung. Bekannte WYSIWYG Editoren für Internetseiten sind: Dreamweaver, GoLive, NetObjects Fusion. Durch die unterschiedliche Interpretation der Webstandards, durch die verschiedenen am Markt erhältlichen Browser, ist es für die WYSIWG Editoren sehr schwer, ein Ergebnis zu erzielen das auf allen Browsern

gleich wiedergegeben wird. Der erzeugte HTML Code muss also manuell nachgearbeitet werden, um auf alle Eigenheiten der Browser eingehen zu können und so ein einheitliches Erscheinungsbild zu erreichen.

3.3 Existierende Editoren für Visitenkarten

3.3.1 Einordnung in die Automatenklassen nach Zipper

Bernd Zipper, Geschäftsführer der ZIPCON Consulting GmbH, hat eine sehr aussagekräftige Unterteilung in drei Gruppen vorgenommen:

- Klasse A: Automat[11]
- Klasse B: Layout-Automat[12]
- Klasse C: Kampagne-Automat[13]

Klasse A beschreibt Zipper als „...Web-to-Print System zur Realisierung von einfachen Druckvorlagen (Visitenkarten etc.). [Eventuelle] Möglichkeit zum Austausch von Bildern und einzelnen Layoutelementen. [Eine] Vorbereitung zur Anbindung des Systems an ein gegebenes Shopsystem [kann vorhanden sein] – ggf. einfaches Shopsystem im Lieferumfang [enthalten.]“ [14]

Bei einem Klasse B automat handelt es sich laut Zipper um ein „Hochindividuelles Web-to-Print-System zur Realisierung von einfachen bis hin zu komplexen Druckvorlagen. [Die] Möglichkeit zum Austausch von Bildern und einzelnen Layoutelementen [ist gegeben], ggf. [kann eine] eigene Bild- und Layoutdaten-

11 Zipper 2006, 7 für diese Arbeit entnommen aus Förster 2007, 12
12 Zipper 2006, 7 für diese Arbeit entnommen aus Förster 2007, 12
13 Zipper 2006, 7 für diese Arbeit entnommen aus Förster 2007, 12
14 Zipper 2006, 7 für diese Arbeit entnommen aus Förster 2007, 12

bank [integriert sein]. [Es besteht die] Möglichkeit zur Freigestaltung von Layoutelementen. [Ein] Shopsystem/Schnittstellen [für ein Shopsystem sind] im Lieferumfang [enthalten.]“[15]

Einen Klasse C Automat beschreibt Zipper als „Hochindividuelles Web-to-Print-System zur Realisierung von einfachen bis hin zu komplexen Druckvorlagen auf Basis einer Layoutengine (Remote-Anwendung von InDesign/QuarkDDS/VivaIP). [Ein Klasse-C-Automat besitzt die] Möglichkeit zum Austausch von Bildern und einzelnen Layoutelementen, ggf. [ist eine] eigene Bild- und Layoutdatenbank [integriert]. [Es besteht die] Möglichkeit zur Freigestaltung von Layoutelementen. [Ein] Shopsystem/Schnittstellen [dafür sind] im Lieferumfang [enthalten] – [Die] Steuerung des Kunden [erfolgt] über [ein] eigenes CRM[16]. [Die Möglichkeit der] Freigestaltung von individuellen Workflows [ist gegeben].“[17]

Das innerhalb dieser Bachelorarbeit entstandene Programm gehört der Klasse A an. Für eine Zuordnung der Klasse B fehlt die Möglichkeit, eine Bild oder Layout Datenbank anzubinden. Die Automaten der Klasse C basieren auf einer Layoutengine. Diese ist innerhalb des im Rahmen dieser Arbeit entstandenen Programms nicht vorhanden.

15 Zipper 2006, 7 für diese Arbeit entnommen aus Förster 2007, 12

16 CRM – Customer Relationship Management (dt. Kundenbeziehungsverwaltung oder Kundenpflege)

17 Zipper 2006, 7 für diese Arbeit entnommen aus Förster 2007, 12

3.3.2 Vergleich mit anderen Automaten der Klasse A

Es gibt am Markt eine Vielzahl von verschiedenen Editoren, die in die Klasse A nach Zipper fallen. Bei diesen handelt es sich um Eigenproduktionen. Sie unterscheiden sich sehr stark in Bedienung und Funktionsumfang. Nach Recherchen des Autors war es nicht möglich, einen Hersteller für diese Editoren zu finden. Dies stützt die Vermutung, dass es noch keine Standardanwendung oder Hersteller gibt, sondern alle zur Zeit verwendeten Editoren speziell produzierte Anwendungen sind.

Die vorhandenen Editoren lassen sich im Wesentlichen in drei Gruppen einteilen:

- statische Editoren, die über ein HTML Formular funktionieren
- einfache JavaScript Editoren
- umfangreiche JavaScript/Java Editoren

Alle gemeinsam haben die Editoren, dass die Schriftart, -größe, -farbe und die Größe des Dokumentes bearbeitet werden können.

Stellvertretend für die Gruppe der statischen Editoren ist der Editor von www.visitenkarten.net [18].

18 http://www.visitenkarten.net/index.php?root=0032&imgqw=0h00009&imgqwid=60, 22.12.2009

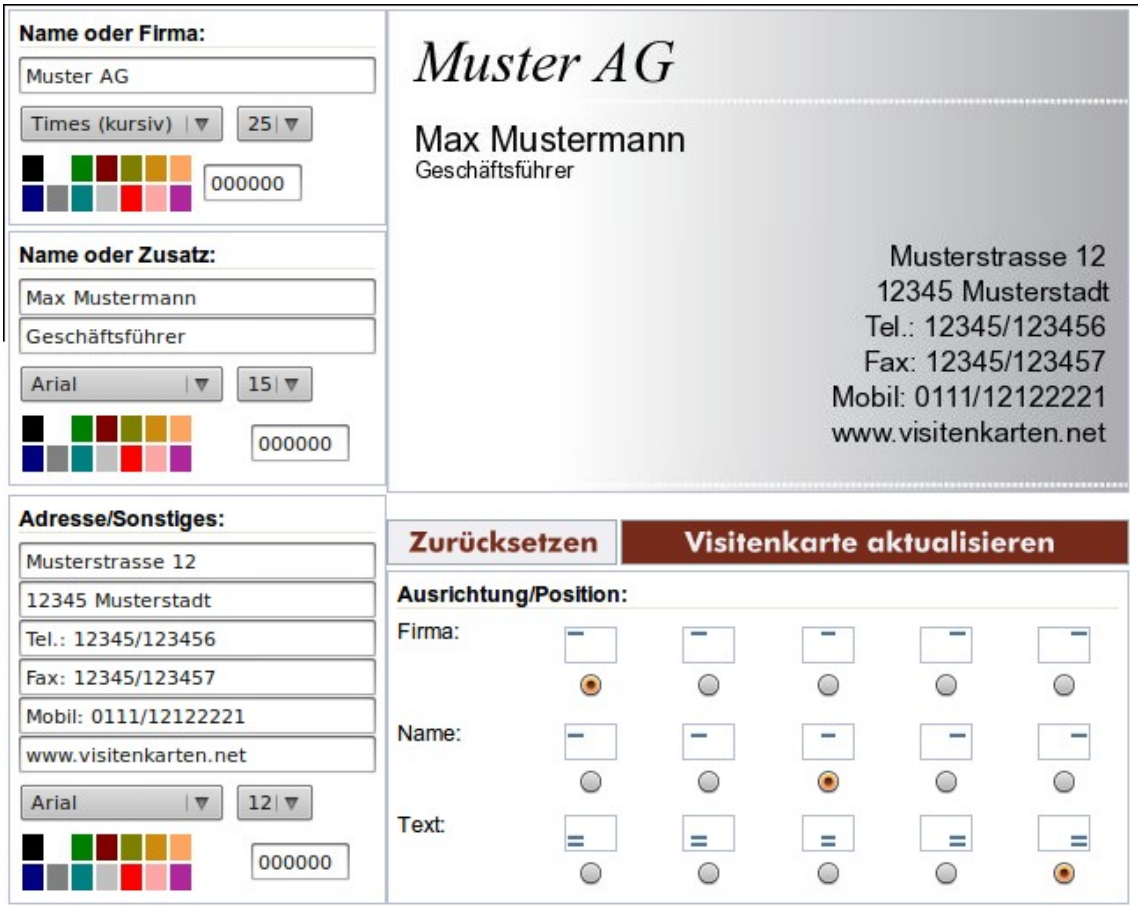


Abbildung 2: Editor von www.visitenkarten.net

Diese Art von Editoren ist sehr eingeschränkt, da die Anzahl und Position der Textfelder nicht variabel ist. Es besteht nur die Möglichkeit, die Ausrichtung des Textes zu ändern. Die Anzeige erfolgt durch ein Bild, das immer nach dem Aktualisieren neu erzeugt und angezeigt wird. Der Benutzer hat keine Möglichkeit, eine PDF Vorschau zu erzeugen. Allerdings ist davon auszugehen, dass die Vorschau durch das Bild auch dem späteren Ergebnis entspricht, da ein Bild in jedem Browser gleich dargestellt wird und der Anbieter die volle Kontrolle während des Erzeugens über das Endprodukt besitzt. Da kein Zugriff auf den Quellcode oder einer druckfertigen PDF möglich ist, kann dies allerdings nicht überprüft werden. Da die Vorschau über ein Pixelbild realisiert wird, ist es nicht möglich zu zoomen, um genauer arbeiten zu können. Der Hintergrund ist in diesem Editor frei wählbar.

Ein Beispiel für die Gruppe der einfachen JavaScript Editoren ist der Editor von www.visitenkartenonline.de [19].

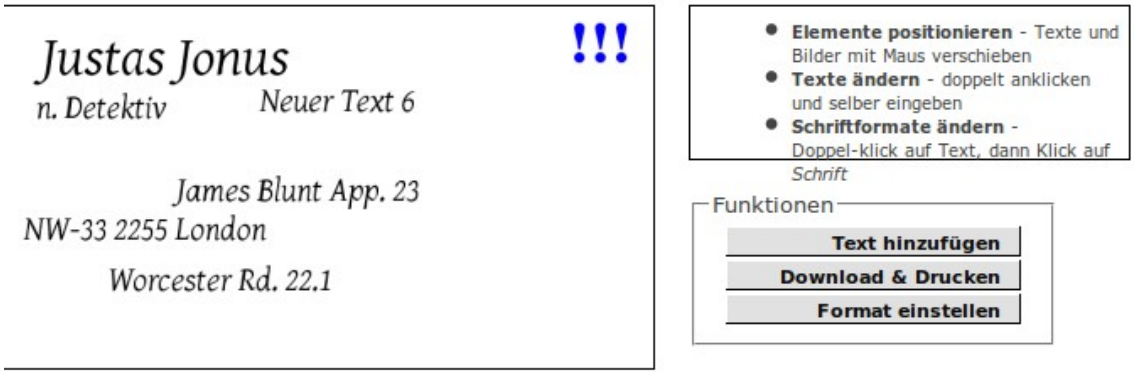


Abbildung 3: Editor von www.visitenkartenonline.de

In diesem Editor ist es möglich, die Anzahl der Textfelder frei zu bestimmen. Die Position der Textfelder kann durch einfaches Drag and Drop verschoben werden. Die Anzeige der Texte erfolgt als Bild. In den meisten Editoren ist es möglich, ein Hintergrundbild hinzuzufügen, bei diesem Beispiel ist es allerdings nicht möglich. Die PDF Ausgabe stimmt exakt mit der Vorschau im Browser überein.

Da auch hier teilweise Pixelbilder verwendet werden, um die Vorschau zu realisieren, ist Zoomen nicht möglich.

19 http://www.visitenkartenonline.de/online-editor.html, 22.12.2009

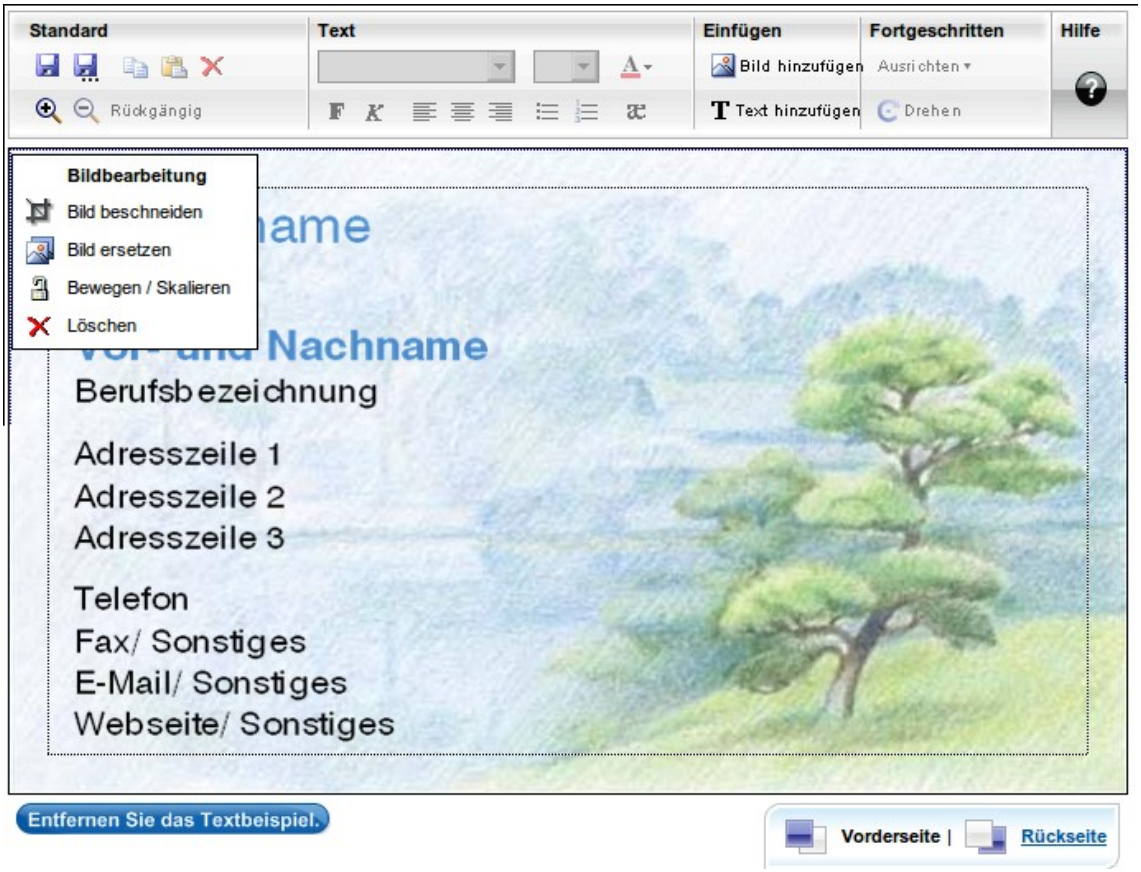


Abbildung 4: Editor von www.vistaprint.ch

Die erweiterten Editoren bieten noch umfangreichere Bearbeitungsmöglichkeiten an. So ist es möglich, den Text beliebig zu drehen oder zusätzlich zu einem Hintergrundbild noch weitere Grafiken einzufügen. Zusätzlicher Bedienkomfort ergibt sich aus der Möglichkeit, die Vorschau zu vergrößern oder Arbeitsschritte über eine Rückgängigfunktion zurück zu nehmen. Ein Beispiel für solch einen Editor ist der Editor von vistaprint.ch [20]

Die Auswahl der Editoren ist nicht vollständig und stellt nur einen sehr kleinen Querschnitt der vorhandenen Lösungen dar.

20 http://www.vistaprint.ch/vp/ns/studio3.aspx?pf_id=064&combo_id=4298&category_id=1&referer=http%3a%2f%2fwww.vistaprint.ch%2fvp%2fwelcome.aspx%3fxnav%3dwelcomeback%26rd%3d2&rd=2 dann klick auf Erweitere Bearbeitung, 22.12.2009

Es existiert keine freie Anwendung die an die eigenen Bedürfnisse angepasst werden kann. Ein Zugriff auf den kompletten Quellcode ist nicht möglich. Alle JavaScript Editoren setzten auf ein JavaScript Framework. Dies wird aus den zugänglichen Quellcodeteilen ersichtlich. Durch die Verwendung eines Frameworks werden die Anwendungen sehr groß und sind von den Funktionen die das Framework bietet, abhängig. Die Einarbeitung in ein solches Framework ist sehr aufwändig und die Möglichkeit eigene Änderungen vorzunehmen leidet darunter. Die Lizenzbedingungen müssen beim Einsatz eines Frameworks beachtet werden.

Das Programm welches im Rahmen dieser Bachelorarbeit entsteht, schließt diese Lücke. Es kommt kein JavaScript Framework zum Einsatz. Der Aufbau ist modular und einfach gehalten, sodass eine Anpassung ohne großen Aufwand möglich ist.

3.4 Auswahl der technischen Umsetzung des Teil B

Durch die beschriebenen Probleme bei WYSIWYG Umsetzungen in Browsern muss vor der Umsetzung geprüft werden, ob die verwendete Technik dazu in der Lage ist, auf allen Plattformen ein einheitliches Ergebnis zu liefern.

Da der Editor eine Client Lösung sein soll, um die Serverlast so gering wie möglich halten, kommen nur JavaScript oder Flash als Lösung in Frage. Nur diese haben einen so hohe Verbreitung, dass ihr Einsatz sinnvoll möglich ist.

Für Javascript spricht, dass dies 98%[21] der Webbenutzer aktiviert haben und die Unterstützung schon im Browser vorhanden ist, ohne dass Plugins oder Erweiterungen installiert werden müssen.

Es soll für jeden leicht möglich sein, das Programm weiter zu entwickeln, oder an die eigenen Bedürfnisse anzupassen. Dafür ist es von Vorteil, wenn der Quellcode einfach einzusehen und zu ändern ist. Funktionierende Flash Editoren

21 http://www.webmasterpro.de/portal/webanalyse-technologien.html, 23.12.2009

sind zur Zeit nur von Adobe gegen Entrichten einer Lizenzgebühr zu erhalten. Die Preise für die Lizenzen liegen bei ca. 800€[22] für die aktuelle Version. Da es keine andere funktionierende Möglichkeit gibt, Flashdateien zu erzeugen oder zu bearbeiten, würde dies die Anpassbarkeit des Programms stark einschränken und von der Verwendung bestimmter Adobe Software abhängig machen.

Javascript hingegen, lässt sich mit jedem beliebigen Texteditor bearbeiten und der Quellcode liegt offen vor. Das Programm in JavaScript ist dadurch für eine viel größere Gruppe von Personen leicht zu bearbeiten. Durch die Raphaël JavaScript Library ist es möglich, in JavaScript durch SVG und VML beliebige Schriftarten in Vektoren darzustellen und anzuzeigen. Zusätzlich ist durch die Verwendung der Raphaël Bibliothek gewährleistet, dass auch die nicht SVG-fähigen Browserversionen des Internet Explorers die Schriften darstellen können, indem die Grafiken automatisch als VML gerendert werden, welches dem Internet Explorer kompatibel sind.

Da die Raphaël Bibliothek Vektorgrafiken erzeugt, ist ein Zoomen über die in jedem modernen Browser vorhandene Zoomfunktion möglich.

Aufgrund der einfachen nachträglichen Bearbeitung und der in JavaScript durch die Raphaël Bibliothek vorhandenen Fähigkeit, Vektorgrafiken zu erzeugen und Schriftarten darzustellen, erfolgt die Umsetzung des Programmteils B mittels Javascript.

22 https://store5.adobe.com/cfusion/store/index.cfm?&store=OLS-DE&view=ols_prod&category=/Applications/FlashP&distributionMethod=FULL&nr=0#loc=de_de&store=OLS-DE&view=ols_prod&category=/Applications/FlashP 30.11.2009

3.5 Auswahl der technischen Umsetzung der Teile A und C

3.5.1 Teil A

Teil A ist für das Laden eines schon gespeicherten Dokumentes oder das Erzeugen eines neuen Dokumentes zuständig. Da beim Ladevorgang eine Menge DOM Operationen ausgeführt werden müssen, um das Dokument entsprechend vorbereiten zu können, kommt dafür nur JavaScript in Frage.

Beim Laden eines Dokumentes müssen die Daten aus einer MySQL Tabelle geladen werden. Die Datenbank Abfrage ist am einfachsten durch PHP zu realisieren. Da PHP auf fast jedem Webspace vorhanden ist, welcher Zugriff auf eine MySQL Tabelle liefert, bietet sich die Verwendung an.

Das Erzeugen eines neuen Dokumentes besteht ebenfalls aus dem Anpassen der vorhanden DOM Elemente an die ausgewählte Dokumentengröße und das Übergeben des Namens des Autors. Dies ist alles am einfachsten mit JavaScript umzusetzen.

3.5.2 Teil C

Teil C erzeugt die Vorschau-PDF und legt diese auf dem Server ab, wenn der Benutzer den Auftrag abschickt. Da keine Interaktion mit dem Benutzer stattfindet ist, und alle Arbeitsschritte auf dem Server ausgeführt werden, wird für Teil C ausschließlich PHP verwendet.

Zum Erzeugen der PDF wird FPDF verwendet. Dieses steht unter freier Lizenz und ist dadurch kostenfrei einsetz- und anpassbar. Es ermöglicht das Erzeugen von PDF Dateien, ohne dass die PDFlib Bibliothek auf dem Webserver installiert seinen muss. Außerdem vereinfacht es das Erzeugen einer PDF, da FPDF viele Arbeitsschritte zusammenfasst, die dann durch eine FPDF Funktion aufgerufen werden können.

4 Umsetzung

4.1 Raphaël

Raphaël ist eine JavaScript Bibliothek, welche unter der MIT License steht und kann ohne Lizenzgebühren frei verwendet werden[23]. Sie verwendet SVG und VML als Grundlage für das Erstellen der Vektorgrafiken. Raphaël unterstützt folgende Browser: Firefox 3.0+, Safari 3.0+, Opera 9.5+ und Internet Explorer 6.0+. Damit werden alle aktuell am Markt relevanten Browser unterstützt (siehe Kapitel 3.1).

Jedes durch Raphaël erzeugte Element ist ein DOM Objekt und kann so durch JavaScript nach dem Erzeugen noch weiter verändert werden.

Mit Raphaël ist es möglich, beliebige Schriftarten zu verwenden, diese müssen lediglich als JavaScript Datei vorliegen. Dabei wird auf Cufón zurückgegriffen. Cufón besteht aus zwei Teilen, einem Schriftarten Generator und einer Render Engine in JavaScript. Die Render Engine ist in Raphaël bereits integriert, sodass es nicht nötig ist, ein Zusatzprogramm zu verwenden. Zum Generieren kann auf den Cufón Generator unter http://cufon.shoqolate.com/generate/ zurückgegriffen werden.

Der Generator kann eine beliebige Font Datei umwandeln. In mehreren Schritten wird erst ein SVG Font erzeugt; dieser wird in SVG Pfade umgewandelt, die als Grundlage für VML Pfade dienen. Dieser Aufwand ist nötig, da der Internet Explorer nur auf die VML Pfade zurückgreifen kann. Das Endergebnis wird dann als ein JSON (JavaScript Object Notation) abgespeichert und kann als Schriftart für Raphaël registriert werden.

23 http://raphaeljs.com/ 03.12.2009

4.2 Allgemeine Erklärung

Wie oben beschrieben, erfolgt die Umsetzung des Editors mittels JavaScript und HTML. An einigen Stellen musste auf PHP als Hilfsmittel zurückgegriffen werden, da Dateioperationen oder Datenbankabfragen auf dem Server ausgeführt werden mussten.

Das Erzeugen der PDF und das Ablegen des Dokumentes in der Datenbank wird durch ein PHP Script realisiert.

Folgender Ablauf findet statt, wenn die Datei Editor.html geladen wird:

1. Alle für Teil A und B benötigten JavaScript Teile werden eingebunden.
2. Alle vorhanden Schriftarten werden für die Verwendung mit Raphaël registriert.
3. Der DIV Container mit allen HTML Elementen die für Programmteil B benötigt werden, wird geladen aber mittels der CSS Eigenschaft „display:none“ noch nicht für den Benutzer sichtbar gemacht.
4. Der DIV Container mit allen HTML Elementen für Teil A wird geladen und dem Benutzer angezeigt.

Der Programmteil A, der zum Laden eines Dokuments dient, ist in die Datei editor.html integriert. Es handelt sich dabei um ein HTML Formular, um die benötigte Eingaben des Benutzers zu erfassen und um einige JavaScript Scripten, die das Dokument laden und den Editor vorbereiten. Dieser Teil kann einfach deaktiviert werden, so dass der Editor auch ohne Internetverbindung benutzt werden könnte. Es müsste nur eine Lösung für die lokale Zwischenspeicherung der Daten gefunden werden, bis wieder eine Internetverbindung besteht und der Auftrag endgültig abgeschickt wird. Dafür müsste nur der Teil C an die neue Speicherart angepasst werden. Eine PDF Vorschau ist dann allerdings nicht mehr möglich, da dafür PHP zur Verfügung stehen muss.

4.3 Teil A

Direkt nach dem Start des Programms hat der Benutzer die Möglichkeit, ein neues Dokument zu erstellen oder ein schon vorhandenes zu laden und die Bearbeitung fortzusetzen.

Die Programmteile A und B sind sehr eng miteinander verknüpft. Sie benutzen ein gemeinsames HTML-Gerüst (editor.html) und werden gemeinsam nach dem Aufruf des Programms geladen. Dies ist nötig, da beim Laden eines Dokumentes die entsprechenden Funktionen aus Programmteil B aufgerufen werden müssen.

Direkt nach dem Start sind alle Bedienelemente, die zu Teil B gehören, ausgeblendet und nur die Programmteile A sind sichtbar.

Wenn der Benutzer ein schon vorhandenes Dokument laden möchte, wird er dazu aufgefordert seinen Vor-, Nachnamen und die entsprechende Auftragsnummer einzugeben. Anhand dieser Daten wird der entsprechende Auftrag gefunden. Nach dem Klicken auf den Button „weiter“ wird durch das JavaScript geprüft, ob alle geforderten Eingaben vorhanden sind und die Abfrage an die MySQL Datenbank abgeschickt.

Da mit JavaScript keine MySQL Datenbankabfragen möglich sind, muss die Abfrage an ein PHP Skript weitergeleitet werden, welches die Daten aus der Datenbank abfragt und das Ergebnis an das JavaScript Script zurück gibt. Da PHP eine serverseitige Sprache ist, muss die Seite neu geladen werden um die Anfrage abzuschicken und die Daten zurückzuerhalten. Da es sich bei der editor.html Datei aber um eine HTML Datei handelt, kann kein PHP Code integriert werden und es ist nicht gewünscht, dass die Seite neu geladen werden muss, weil dann alle Änderungen die bereits am DOM vorgenommen wurden erneut ausgeführt werden müssten.

Das PHP Script wird deshalb durch einen unsichtbaren iFrame für den Benutzer unsichtbar eingebunden. Das Script ist innerhalb des iFrames unabhängig von der übergeordneten Seite editor.html und kann daher neu geladen werden, ohne dass die komplette Seite neu aufgebaut werden muss.

Das Ablaufdiagramm zu Teil A ist als Anlage 1 im Anhang zu finden.

4.3.1 Ablauf:

Die Eingabe des Benutzers erfolgt innerhalb eines HTML Formulars. Der Submit Button des Formulars ist allerdings für den Benutzer nicht sichtbar. Der Weiter Button, den der Benutzer sieht, schickt nicht sofort das Formular ab. Nach dem Drücken des weiter Buttons wird die Funktion weiter1 ausgeführt. Die Funktion prüft, ob der Benutzer ein Dokument laden oder neu erstellen will.

4.3.1.1 Laden

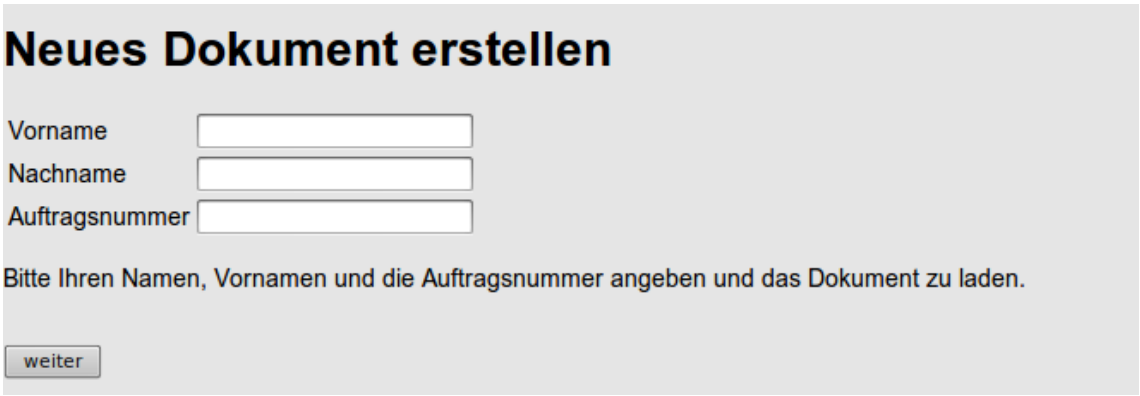


Abbildung 5: Eingabemaske zum Laden eines Dokuments

Im Falle des Ladens eines Dokumentes werden alle benötigten JS Arrays durch die Funktion erzeugt, in die die Daten aus der Datenbankabfrage geschrieben werden. Erst danach wird das Abschicken des Formulars ausgelöst.

Das PHP Skript bekommt die vom Benutzer eingebenen Daten (Vorname, Nachname und Auftragsnummer) über eine POST Weiterleitung übergeben.

Die Datenbankabfrage erfolgt, die gefunden Daten werden in die vorbereiteten JavaScript Arrays geschrieben und die JavaScript-Funktion „laden_finish“ wird mit dem Paramter „found“ gestartet. Wenn keine Ergebnisse gefunden wurden, wird die Funktion mit dem Parameter „nothing“ gestartet. Damit ist die Arbeit des PHP Scripts beendet.

Die vom PHP Script gestartete JavaScript-Funktion laden_finish unterscheidet, ob etwas gefunden wurde (Paramter „found“ wurde übergeben) oder nichts gefunden wurde (Parameter „nothing“ wurde übergeben). Wenn nichts gefunden wurde, wird ein Popup Fenster mit einer Fehlermeldung angezeigt.

Falls Einträge gefunden werden, beginnt die Funktion zuerst, alle geladenen Textfelder anzulegen und die geladenen Eigenschaften zu übertragen. Dies geschieht auf demselben Weg, wie es auch der Benutzer machen würde. Der geladene Textinhalt wird in das Eingabefeld geschrieben, die Farbwerte werden in das dazugehörige Feld geschrieben etc. Nachdem alle Eigenschaften übertragen wurden, wird die Aktualisierung des Textfeldes ausgelöst. Dies entspricht einem Klick des Benutzers auf die Schaltfläche Aktualisieren. Welche Mechanismen dabei genau ablaufen wird, in Teil B erklärt.

Da dieselbe Funktion zum Erstellen der Daten durch den Benutzer und zum Laden verwendet wird, kann diese Funktion beliebig geändert werden, ohne dass dies Einflüsse auf den Ladevorgang hat. Es müssen nur eventuell neu hinzugekommene Datenfelder in die Funktion eingetragen werden, die die Daten aus der Datenbank überträgt.

Damit ist der Vorgang des Ladens abgeschlossen und der Benutzer kann weiterarbeiten.

4.3.1.2 Neu erstellen

Das Erstellen eines neuen Dokumentes läuft viel einfacher ab als das Laden. Nach dem Klicken auf weiter wird die Funktion weiter1 gestartet. Diese erkennt anhand einer nicht gesetzten Variable, dass ein neues Dokument erstellt werden soll und überträgt die gewählten Abmessungen des Dokumentes in die dafür vorgesehen Felder. Danach wird die Funktion resize1 ausgeführt, die das DIV der Zeichenfläche auf die gewählte Größe eingestellt.

Abbildung 6: Eingabemaske zum Erstellen eines Dokument

Danach werden alle Bedienelemente, die zu Teil A gehören, aus dem DOM gelöscht. Der Editor wird angezeigt und der Benutzer kann mit dem Bearbeiten beginnen.

4.4 Teil B

Durch die Umsetzung mit JavaScript wird versucht, so wenig Last wie möglicht auf den Server des Anbieters zu erzeugen. Der Server des Anbieters wird nur belastet, wenn der Benutzer ein eigenes Hintergrund Bild einbindet oder die PDF Vorschau durch Teil C generieren lässt. Teil B ist der umfangreichste der drei Programmteile.

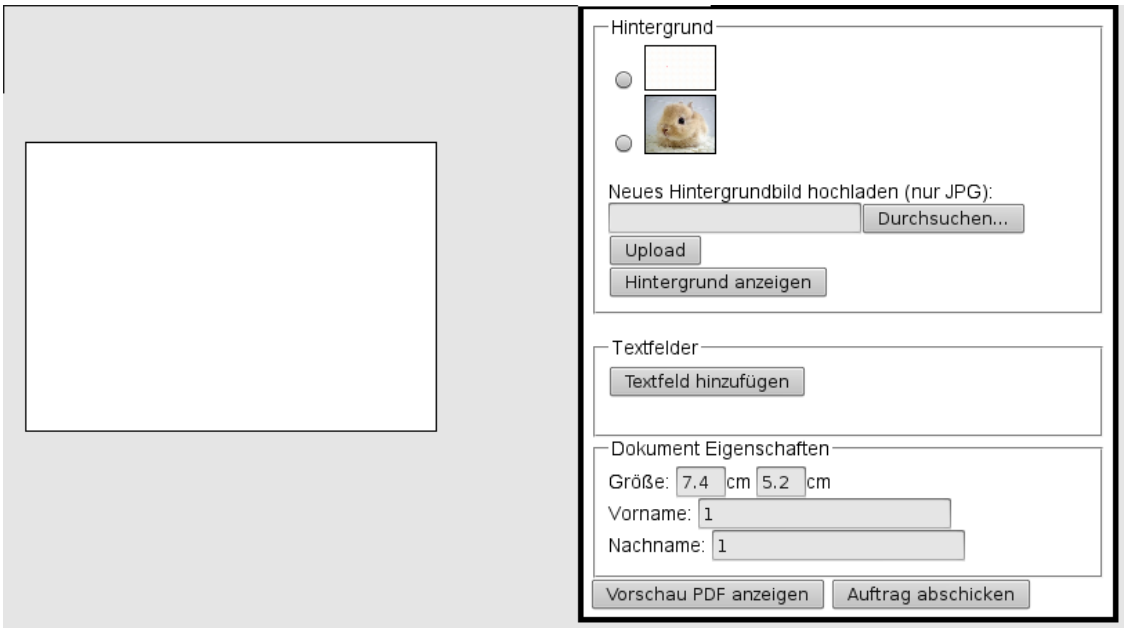


Abbildung 7: Programmteil B nach dem Start

Der Programmteil B umfasst folgende Funktionen:

- Einbinden eines schon vorhanden Hintergrundbildes
- Hinzufügen und Einbinden eines eigenen Hintergrundbildes
- Hinzufügen von Textfeldern
- Formatieren des Textes in Schriftart, Schriftgröße und Schriftfarbe
- Freies Positionieren der Textfelder durch Anfassen und Verschieben mit der Maus

Das Grundgerüst bildet ein HTML Formular, in dem alle Daten abgelegt und an Teil C zum Erzeugen der PDF übergeben werden. Wenn ein neues Textfeld angelegt wird, werden durch JavaScript dem DOM des Formulars weitere Felder hinzugefügt, um die zusätzlichen Daten aufnehmen zu können. Die Textfelder und die dazugehörigen Eigenschaftenfeder werden durch eine laufende Nummer gekennzeichnet.

Es werden folgend Daten an Programmteil C zum Erzeugen der PDF übergeben:

- Pfad zum Hintergrundbild
- Abmessungen des Dokumentes
- Name und Vorname des Verfassers
- Inhalt, Position, Farbe und Größe der Textboxen

Die Zeichenfläche besteht aus einem DIV-Container mit einer ein-Pixel-breiten, schwarzen Outline. Dieser DIV-Container wird immer an die ausgewählte Größe des Dokumentes angepasst.

Ein Klick auf den Button „Textfeld hinzufügen“ fügt ein neues Textfeld mit einer Standardformatierung und die dazugehörigen Einstellungsoptionen hinzu.

Abbildung 8: Textoptionen

Das neu erstellte Textfeld erscheint links oberhalb der Zeichenfläche und kann mit der Maus angeklickt und verschoben werden. Nach dem Beenden des Verschiebevorgangs wird die aktuelle Position des Textfeldes in für den Benutzer nicht sichtbare Eingabefelder im Formular gespeichert.

Das Textfeld besteht aus einem umschließendem DIV Container, in dem die SVG Datei erzeugt wird. Da der DIV-Container verschoben werden kann, verschiebt sich auch die darin enthaltene SVG Datei.

Das Rendern des Textes erfolgt über die schon beschriebene Raphaël Bibliothek.

Die Größe der SVG Zeichenfläche muss immer angegeben werden. Es ist zwar möglich, die Größe des SVG automatisch anpassen zu lassen; allerdings passiert es ab einer gewissen Schriftgröße, dass die Ober- und Unterlängen der Buchstaben abgeschnitten werden. Die Größe des SVG wird daher abhängig von der Schriftgröße und der Anzahl der Buchstaben errechnet. Das umschließende DIV passt sich automatisch an die Größe der darin enthaltenen SVG an.

Der Farbwähler ist ein freies Projekt namens jscolor. Dabei handelt es sich um einen freien und leicht zu implementierender Farbwähler auf JavaScript Basis. Für die Anzeige im Editor werden die hexadezimalen Farbwerte verwendet. Diese werden später während des Erzeugens der PDF in RGB-Farbwerte umgerechnet.

4.4.1 Ausgleichen der Schriftlinie

Eine Schwierigkeit bestand darin, dass die SVG immer an der linken oberen Ecke des umschließenden DIV Containers ausgerichtet ist. Das heißt, die Schriftlinie verschiebt sich immer mit der Größe der Schrift nach unten. Um für den Benutzer das Arbeiten zu erleichtern und eine exakte Wiedergabe in der PDF zu ermöglichen, darf sich die Schriftlinie beim Ändern der Schriftgröße nicht verschieben.

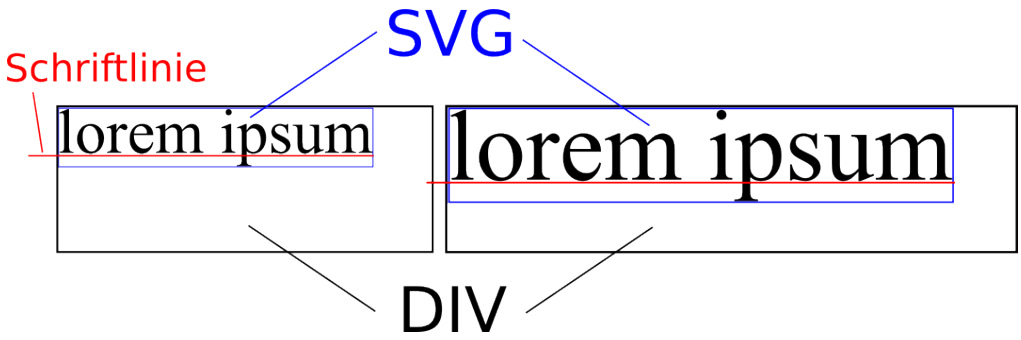


Abbildung 9: Ohne Ausgleich

Dieses lässt sich durch das Verschieben des DIV-Containers nach oben durch die CSS Funktion Margin erreichen. Mittels Margin lässt sich der Außenabstand eines DIV-Containers verändern. Durch einen negativen oberen Margin ist es möglich, den DIV-Container nach oben zu verschieben. Das Verschieben des Containers mittels Margin hat keinen Einfluss auf die angebene Position. Wenn zum Beispiel die Koordinaten X=100 und Y=100 betragen und der Container durch einen negativen oberen Außenabstand um 10 Pixel (Margin-Top:-10px) nach oben verschoben wird, bleiben die Koordinaten des Containers X=100 und Y=100. Dargestellt wird der Container allerdings so, als würde er die Koordinaten X=100 und Y=90 besitzen.

Der Container muss also immer abhängig von der Schriftgröße nach oben verschoben werden um die unterschiedlichen Schriftgrößen konstant auf einer gemeinsamen Schriftlinie zu halten.

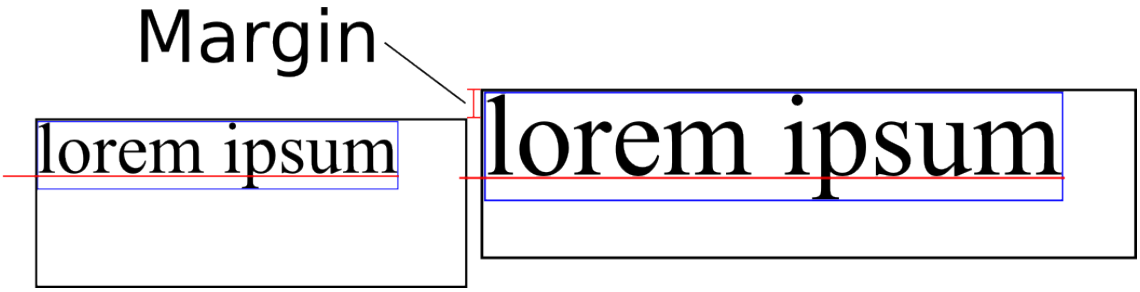


Abbildung 10: Mit Ausgleich

Der benötigte negative obere Außenabstand beträgt immer Schriftgröße geteilt durch -2.14.

4.4.2 Hintergrundbild

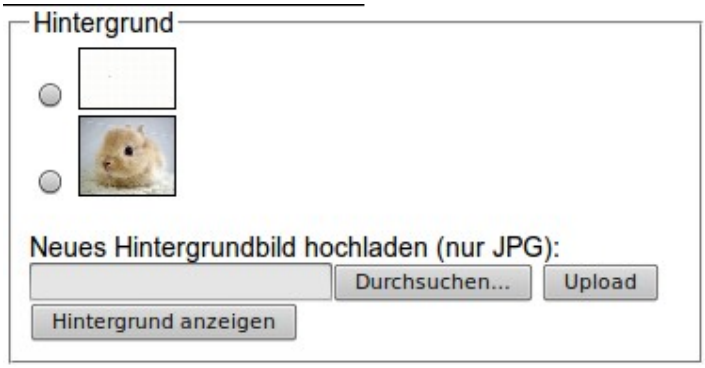


Abbildung 11: Dialog zum Hinzufügen einen Hintergrundbildes

Der Benutzer hat die Möglichkeit, einen Hintergrund einzubinden. Es kann eine Auswahl an Bildern zur Verfügung gestellt werden, zu der der Benutzer noch ein frei wählbares Bild hinzufügen kann.

Das Bild wird als Raphaël Image Objekt eingebunden, welches innerhalb des DIV-Containers erzeugt wird, der die Zeichenfläche darstellt. Dadurch ist es automatisch an der linken obereren Ecke ausgerichtet. Die Größe bestimmt sich aus den Abmessungen des Dokumentes.

Für das Hinzufügen eines eigenen Hintergrundes muss wieder ein Umweg über ein PHP Script gegangenen werden. Das Sicherheitskonzept von JavaScript erlaubt keinerlei Dateioperationen auf dem Client oder Server. Es ist also nicht möglich, ausschließlich mit JavaScript das selbst gewählte Hintergrundbild auf den Server zu übertragen.

Es kommt dieselbe Vorgehensweise wie beim Laden der Daten aus der MySQL Datenbank zum Einsatz. Für das Hintergrundbild wird ein eigenes Formular verwendet, welches nicht in Verbindung mit dem Formular steht, in dem die Daten für das Erzeugen für die PDF gespeichert werden.

Das PHP Script kopiert das ausgewählte Bild auf den Server und setzt die Zugriffsrechte so, dass der Webserver auf das Bild zugreifen kann.

Das Auswahlfeld für das eigene Hintergrundbild ist im DOM schon vorhanden, allerdings für den Benutzer noch nicht sichtbar. Nachdem das PHP Script den Upload beendet hat, wird eine JavaScript Funktion aufgerufen, die den Pfad des Vorschaubildes anpasst und das Auswahlfeld inkl. des Vorschaubild für den Benutzer sichtbar macht.

Der Dateiname des gewählten Hintergrundes wird nach dem Anzeigen in ein verstecktes Input Feld innerhalb des Formulars zum Speichern der Daten geschrieben.

4.4.3 Vorgehen beim Aktualisieren der Textfelder

Nachdem der Benutzer auf die Schaltfläche „Aktualisieren“ geklickt hat, wird als Erstes das dazugehörige Textfeld und der DIV-Container, welcher die Position bestimmt, aus dem DOM gelöscht. Dies ist nötig, da es nicht möglich ist, den angezeigten Text nochmals durch Raphaël zu ändern. Es wurde allerdings die Position des DIV-Containers gespeichert und dieser wird an exakt derselben Stelle wieder erstellt und mit dem neuen SVG gefüllt.

Das neue SVG wird jetzt anhand der in den Input Feldern gespeicherten Daten neu erzeugt. Dies geschieht so schnell, dass der Benutzer keine Verzögerung wahrnimmt. Dadurch wird die Änderung sofort sichtbar.

Das Ablaufdiagramm zum Aktualisieren der Textfelder ist als Anlage 2 im Anhang zu finden.

4.5 Teil C

4.5.1 Anzeigen der PDF

Für das Generieren der PDF wird, wie oben bereits erwähnt, FPDF verwendet. An Teil C werden alle Daten in Arrays aus dem Formular und die letzte laufende Nummer des Textfelds übergeben. Es gibt je ein Array pro Eigenschaft. Die laufende Nummer des Textfeldes steht dabei für den Index des Arrays in dem die dazugehörige Eigenschaft abgespeichert wurde. Da durch das Löschen von Textfeldern in der Nummerierung Lücken entstehen können (zum Beispiel wenn Textfeld Nummer 3 gelöscht wird, werden Textfelder 1, 2, 4 und 5 übergegeben). Es ist nötig, die Arrays auf Inhalt zu überprüfen und in ein neues Array ohne Lücken zu schreiben.

Im nächsten Schritt wird das PDF Dokument mit den richtigen Maßen angelegt. Es ist wichtig, den Mindestabstand auf einen hohen negativen Wert zu setzten, um ein Umbrechen der Textzeilen zu vermeiden. Zusätzlich wird noch das automatische Erzeugen neuer Seiten deaktiviert, da sonst eine zweite Seite angelegt werden würde, sofern ein Textfeld zu nah am unteren Rand positioniert wurde.

Falls ein Hintergrundbild eingebunden wurde, wird dies als nächstes angezeigt. Die Größe des Hintergrundbildes wird auf die Größe des Dokumentes gesetzt.

Der nächste Schritt besteht darin, die Textfelder und deren Inhalt anzulegen. Die Schleife wird so oft ausgeführt, bis alle Textfelder abgearbeitet sind. Da die PDF mit der Maßeinheit Millimeter arbeitet, müssen zuerst die in Pixel vorliegenden Koordinaten in Millimeter umgerechnet werden. Die Schriftgrößenangabe von Raphaël unterscheidet sich von FPDF. Die Schriftgröße wird linear durch das Multiplizieren mit dem Faktor 0,71 angeglichen.

FPDF verwendet ein Koordinatensystem dessen Koordinatenursprung in der linken oberen Ecke liegt. Es ergibt sich dieselbe Problematik wie bereits in Kapitel 5.4.1 beschreiben. Das Ausgleichern der Schriftlinie der PDF ist jedoch einfacher, da die endgültigen Koordinaten der Textfelder ohne Bedeutung sind. Um die Schriftlinien konstant zu halten, wird die Schriftgröße durch 10 geteilt und dieser Wert von der Y-Koordinate abgezogen.

Der Außenabstand des Textes zur umgebenden Textbox, weicht von dem im Editor angezeigten Wert ab. Auch dieser Abstand muss also ausgeglichen werden. In X Richtung um den Wert +2.5 und in Y Richtung um den Wert -4.1.

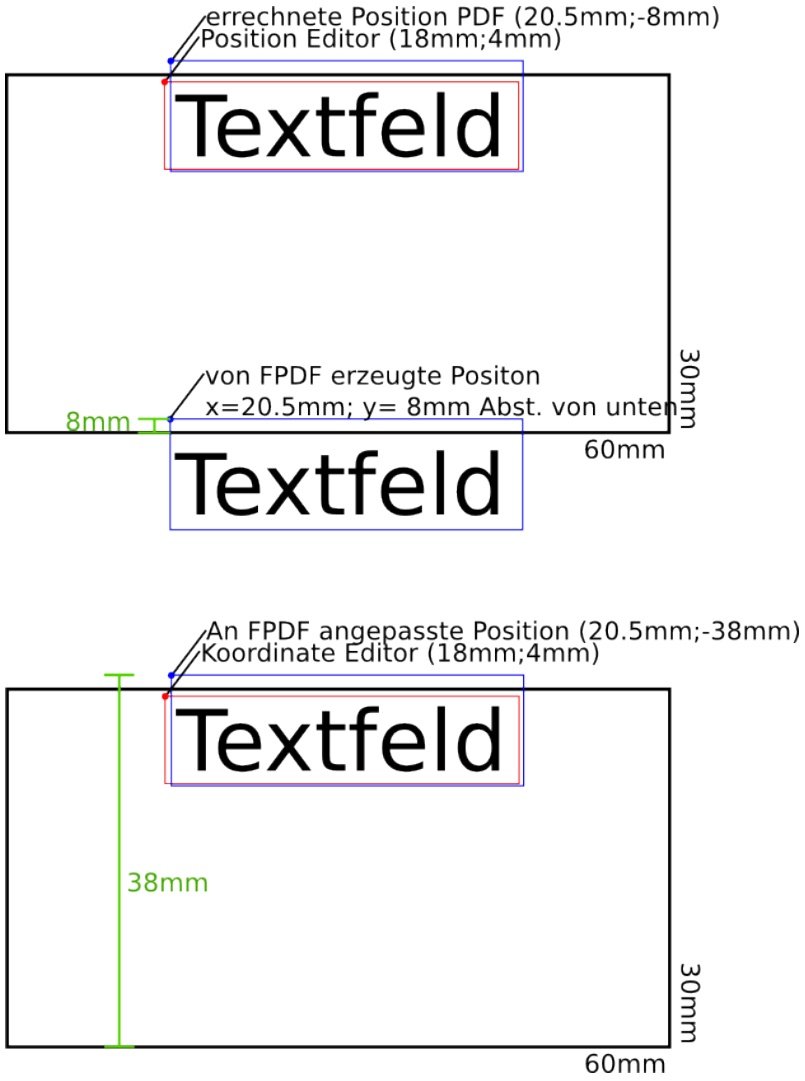


Abbildung 12: Umgang mit negativen Koordinaten

Wenn ein Textfeld zu nah am Rand des Dokumentes platziert wird, kann es durch das Ausgleichen zu negativen Koordinaten kommen. FPDF geht mit negativen Koordinaten allerdings anders um. Sobald eine Koordinate ein negatives Vorzeichen besitzt, ändert sich für diese der Koordinatenursprung nach unten rechts und die Koordinate wird so behandelt, als wäre sie positiv. Der Punkt wird dadurch an der falschen Stelle erzeugt. Um dies zu vermeiden muss jede Koordinate < 0 mit der negativen Dokumentengröße in der entsprechenden Richtung addiert werden. Als letzter Schritt muss nur noch die Farbe und Schriftart gesetzt und das Textfeld ausgegeben werden.

Nachdem alle Textfelder abgearbeitet wurden, ist das Erzeugen der PDF abgeschlossen und die fertige PDF wird angezeigt.

Das Ablaufdiagramm zum Erstellen der PF ist als Anlage 3 im Anhang zu finden.

4.5.2 Speichern der Daten

Wenn der Nutzer den Auftrag nicht abschickt, werden vor dem Erzeugen der PDF sämtliche übergebene Daten unverändert in eine MySQL Tabelle geschrieben. Alle Anpassungen die an Schriftgröße, Koordinaten etc. vorgenommen werden, um die PDF Ausgabe an den Editor anzupassen, werden nicht abgespeichert. Dies ist nötig, da sonst die Daten im Editor falsch dargestellt werden.

Zusätzlich kann die Art und Weise des PDF Erstellens angepasst werden, ohne dass die schon gespeicherten Daten nutzlos werden.

Die PDF wird zu Abschluss angezeigt und zusätzlich noch auf dem Server abgelegt. Der Dateiname entspricht der Auftragsnummer.

5 Ausblick

Der Editor wird unter der GPL 3 Lizenz veröffentlicht und ist damit eine freie Software [24]. Vorrangig ist der Editor für das Erstellen von Visitenkarten gedacht. Durch die Beschränkung auf nur eine Farbe und das Anpassen der Mindestabstände an die Gegebenheiten von Stempeln, ließe sich der Editor auch zum Erstellen von Stempeln verwenden.

Durch die Möglichkeit, Dokumente zu speichern und wieder zu laden, lassen sich leicht Vorlagen erstellen, die der Benutzer verwenden kann. Diese müssen hierzu nur angelegt und gespeichert werden. Dann könnte beim Erstellen eines neuen Dokumentes eine Auswahl der Vorlagen angezeigt werden. Da das Erzeugen der Elemente durch ein Skript gesteuert werden kann ist es möglich, dynamische Vorschläge zu erstellen. Der Benutzer kann dann nicht nur auf vorher angelegte Vorlagen zurückgreifen, sondern es wäre auch möglich, anhand der gewünschten Dokumentenmaße eine optimale Verteilung der Textfelder vorzuschlagen und auszuführen.

Visitenkarten und Stempel sind allerdings nicht die einzigen Einsatzmöglichkeiten. Durch Erweiterungen des Codes wäre es möglich, mit dem Editor beliebige Kleindrucksachen zu erstellen. Es wäre denkbar, dass ein kleiner Verlag mit diesem Programm dem Kunden eine Möglichkeit zur Verfügung stellt, um Anzeigen zu entwerfen und einzureichen. Der Vorteil des Systems ist, dass keine spezielle Software benötigt wird. PHP und MySQL ist auf jedem Webserver vorhanden, JavaScript in jedem Browser. Falls eine andere Datenbank benötigt wird, lässt sich das Programm leicht anpassen, da nur der Teil, welcher die Datenbankabfrage steuert, angepasst werden muss. Durch den modularen Aufbau ist es möglich, den Editor in ein CMS wie zum Beispiel Typo3 oder in ein Sho-

24 http://www.gnu.org/licenses/gpl.html, 14.12.2009

psystem wie Magento über ein Plugin zu intrigieren. Dadurch wird ein kompletter Bestellablauf möglich, der in ein schon vorhandenes Shopsystem eingegliedert werden kann.

6 Quellcode

6.1.1 Editor.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Drag & Drop</title>
<link rel="stylesheet" type="text/css" href="style.css" />
<script src="bibliotheken/raphael.js" type="text/javascript"></script>
<script type="text/javascript"
src="bibliotheken/jscolor/jscolor.js"></script>

<script src="schrifarten/schriftart.js"
type="text/javascript"></script>
<script src="schrifarten/Times_New_Roman_400.font.js"
type="text/javascript"></script>
<script src="schrifarten/Times_kursiv_italic_400.font.js"
type="text/javascript"></script>
<script type="text/javascript" src="js/dragdrop.js"></script>
<script type="text/javascript" src="js/text.js"></script>
<script type="text/javascript" src="js/hintergrund.js"></script>
<script type="text/javascript" src="js/load.js"></script>
<script type="text/javascript">
function getInternetExplorerVersion() {
    var rv = -1; // Return value assumes failure.
    if (navigator.appName == 'Microsoft Internet Explorer') {
        var ua = navigator.userAgent;
       var re = new RegExp("MSIE ([0-9]{1,}[\.0-9]{0,})");
       if (re.exec(ua) != null)
           rv = parseFloat(RegExp.$1);
    }
    return rv;
}

window.onload = function () {
      //IE Erkennen
      IE = getInternetExplorerVersion();
      if (IE < 8) {
            IE='\v'=='v';
      }
      counttext = 0;
      paper = new Object();
      schrift();
      schrift2();
      schrift3();
      draginit();
      bild = null;
      text_font_i = null;
```

```
}
</script>

</head>

<body id="bd">

<div id="editor" class="hidden">
      <div id="textboxen">

            </div>



      <div id="karte">

      </div>
      <div id="notepad">

      </div>
      <div id="menu" >

            <form name="uploadform" action="upload.php"
target="hidden_upload" method="post" enctype="multipart/form-data">
                  <fieldset>
                     <legend>Hintergrund</legend>
                        <label><input type="radio"
name="Hintergrund_pfad" id="Hintergrund_pfad" value="punkt.jpg"> <img
src="bilder1/punkt.jpg" alt="" width="50" border="1"/></label><br/>
                        <label><input type="radio"
name="Hintergrund_pfad" id="Hintergrund_pfad" value="bunny.jpg"> <img
src="bilder1/bunny.jpg" alt="" width="50" border="1"/></label><br/>
                        <label><input id="eigen" class="hidden"
type="radio" name="Hintergrund_pfad" id="Hintergrund_pfad"
value="eigenes"> <img src="bilder/bunny.jpg" id="eigen1" alt=""
width="50" border="1" class="hidden"/></label><br/>
                        Neues Hintergrundbild hochladen (nur JPG):<br/>
                        <input type="file" name="upload" />
                        <input type="submit" value="Upload"
onclick='aktiv =
window.setInterval("upload_finish(document.uploadform.upload.value)",
1000);'/><br/>
                        <input onclick="hintergrund1(0)" type="button"
name="hintergrund" id="hintergrund" value="Hintergrund
anzeigen"></input>
                        <input onclick="hintergrund_remove1()"
type="button" name="hintergrund_remove" id="hintergrund_remove"
value="Hintergrund enfernen" class="hidden"></input>
                  </fieldset>
            </form>

            <form id="form" name="form" action="pdf_anzeigen.php"
method="post">
            <br/>
            <input type="hidden" name="hintergrund_datei"
```

```
id="hintergrund_datei" value="false" ></input>
            <fieldset>
                  <legend>Textfelder</legend>
                  <input onclick="add1()" type="button" name="add"
id="add" value="Textfeld hinzuf&uuml;gen"></input>
                  <br/><br/>
                  <div id="textfelder">
                  </div>
                  </fieldset>

                  <fieldset>
                        <legend>Dokument Eigenschaften</legend>
                        Gr&ouml;&szlig;e:
                        <input type="text" readonly="readonly"
name="sizeX" id="sizeX" class="size" value="8"></input>cm
                        <input type="text" readonly="readonly"
name="sizeY" id="sizeY" class="size" value="5"></input>cm
                        <input onclick="resize()"  type="button"
name="sizebutton" id="sizebutton" value="Gr&ouml;&szlig;e &auml;ndern"
class="hiddens"></input><br/>
                        Vorname: <input readonly="readonly" type="text"
id="vorname" name="vorname"></input><br/>
                        Nachname: <input readonly="readonly"
type="text" id="nachname" name="nachname"></input><br/>
                        <div id="auftrag_nr_div" class="hidden">Auftrag
Nr.: <input readonly="readonly" type="text" id="auftrag_nr"
name="auftrag_nr" value="0"></input></div>
                  </fieldset>

                  <input type="submit" name="submit" id="submit"
value="Vorschau PDF anzeigen"></input>
                  <input type="submit" name="auftrag" id="auftrag"
value="Auftrag abschicken"></input>
                  <input type="hidden" name="count" id="count"></input>
                  <input type="hidden" name="save" id="save"></input>
            </form>

            <div class="hidden">
                  <iframe src="upload.php" name="hidden_upload"
class="hidden"></iframe>
            </div>

            <input id="statusmsg" value="tesst" type="hidden"></input>
      </div>
</div>
<div id="teila">
      <h1>Neues Dokument erstellen</h1>
      <form name="erstellen" action="laden.php" target="hidden_laden"
method="post">
            <table>
                  <tr>
                        <td>Vorname</td>
                        <td><input type="text"
name="vorname"></input></td>
                  </tr>
                  <tr>
```

```
                        <td>Nachname</td>
                        <td><input type="text"
name="nachname"></input></td>
                  </tr>
                  <tr id="remove1">
                        <td>Gr&ouml;&szlig;e</td>
                        <td>
                              <select size="1" name="size" id="size">
                                    <option>74x52</option>
                                    <option>81x57</option>
                                    <option>85x55</option>
                                    <option>88.9x50.8</option>
                              </select> mm
                        </td>
                  </tr>
                  <tr class="hidden" id="auftrag_nr_input">
                        <td>Auftragsnummer</td>
                        <td><input type="text"
name="auftrags_nr"></input></td>
                        </tr>
            </table>
                  <input type="text" name="error" id="error"
value="standby" class="hidden"></input>
            <p class="hidden" id="auftrag_text">
                  Bitte Ihren Namen, Vornamen und die Auftragsnummer
angeben und das Dokument zu laden.
            </p>
            <a onclick="laden1()" href="#" id="remove2">oder
gespeichertes Dokmment laden<br/></a>
            <br/><input onclick="weiter1()"type="button"
value="weiter"></input>
                  <input type="submit" name="submit" id="submit"
class="hidden"></input>
      </form>


            <iframe src="laden.php" name="hidden_laden"
class="hidden"></iframe>

</div>
</body>
</body>
</html>
```

6.1.2 Text.js

```
function text_akt(nummer){
      var textboxen = document.getElementById("textboxen");
      var div = document.getElementById("textbox-"+nummer);
      //alte postion des div fÃür das  svg speichern
      var posX = div.style.left;
      var posY = div.style.top;
      //altes div inkl svg lÃöschen
      textboxen.removeChild(div);
```

```
	//neues div vorbereiten
	var div = document.createElement("div");
	div.setAttribute("id","textbox-"+nummer);
	//div.setAttribute("onmousedown","dragstart(this)");
	if (IE == true){
		div.setAttribute("className","textbox");
	} else {
		div.setAttribute("class","textbox");
	}
	//alte position wiedrherstellen
	div.style.left = posX;
	div.style.top = posY;
	//div erzeugen
	textboxen.appendChild(div);
	div.onmousedown = function(){dragstart(this);};
	//Negatives margin errechen, damit der text auch bei
schriftgrÃöÃßenÃänderungen
	var textboxen2 = document.getElementById("textbox-"+nummer);
	if ((document.getElementById("fontsize"+nummer).value > 65) ||
(document.getElementById("fontsize"+nummer).value < 1)){
		alert('SchriftgrÃöÃße muss zwischen 1 und 65 liegen');
		document.getElementById("fontsize"+nummer).value = 20;
	}
	var faktor = (document.getElementById("fontsize"+nummer).value/-
2.14);
	faktor = faktor.toString(10);
	textboxen2.style.marginTop = faktor+"px";
	var schrift = document.getElementById("fontface"+nummer);
	//benÃötigte HÃöhe errechnenaj
	var test = document.getElementById("fontsize"+nummer).value;
	test = parseInt(test);
	var svg_height = (test+31.82)/1.8;
	var svg_width =
(document.getElementById("text"+nummer).value.length*(test + 10))/1.5;
	//neune zeichenflÃäche erzeugen
	paper[nummer] =
Raphael(document.getElementById("textbox-"+nummer),svg_width,svg_heigh
t);
	//text einfÃügen
	if (text_font_i != null){
		var schrift1 = text_font_i;
	} else		{
		var schrift1 = schrift[schrift.selectedIndex].text;
	}
	document.getElementById("fontface_trans"+nummer).value =
schrift1;
	var c = paper[nummer].print(15,15,
document.getElementById("text"+nummer).value,
paper[nummer].getFont(schrift1, 800),
document.getElementById("fontsize"+nummer).value).attr({fill:
'#'+document.getElementById("color"+nummer).value});
}

function resize(){
	var karte = document.getElementById("karte");
	var sizeX = document.form.sizeX.value;
```

```
        var sizeY = document.form.sizeY.value;
        sizeX = sizeX * 40;
        sizeY = sizeY * 40;
        // GrÃöÃße des Hintergrundbildes anpassen
        if (bild != null){
              paper_bild.setSize(sizeX,sizeY)
              bild.attr({width: sizeX, height:sizeY});
        }
        sizeX = sizeX.toString(10);
        sizeY = sizeY.toString(10);
        karte.style.width = sizeX+"px";
        karte.style.height = sizeY+"px";
}

function add1(){
        //Neue Eigenschaftenfelder anlegen
        //--------------------------------
        var textfelder = document.getElementById("textfelder");
        //Wrapper fÃür Eigenschaften
        var append = document.createElement("div");
        append.setAttribute("id","text-w"+counttext);
              if (IE == true){
              append.setAttribute("className","text-style");
        } else {
              append.setAttribute("class","text-style");
        }
        textfelder.appendChild(append);
        //Box fÃür Inhalt
        var textfelder = document.getElementById("text-w"+counttext);
        var append = document.createElement("input");
        append.setAttribute("id","text"+counttext);
        append.setAttribute("name","text"+counttext);
        append.setAttribute("type","text");
        append.setAttribute("value","neues textfeld");
        textfelder.appendChild(append);
        //Feld fÃür Schriftart
        append = document.createElement("select");
        append.setAttribute("id","fontface"+counttext);
        append.setAttribute("name","fontface"+counttext);
        append.setAttribute("size","0");
        textfelder.appendChild(append);
        var x3  = document.getElementById("fontface"+counttext);
        var y3 = document.createElement("option");
        var z3 = document.createTextNode("Times");
        y3.appendChild(z3);
        x3.appendChild(y3);
        var y3 = document.createElement("option");
        var z3 = document.createTextNode("Superexpanded");
        y3.appendChild(z3);
        x3.appendChild(y3);
        var y3 = document.createElement("option");
        var z3 = document.createTextNode("Times kursiv");
        y3.appendChild(z3);
        x3.appendChild(y3);
        //Zeilenumbruch
        append = document.createElement("br");
```

```
        textfelder.appendChild(append);
        //FarbwÃ¤hler
        append = document.createElement("input");
        append.setAttribute("id","color"+counttext);
        append.setAttribute("name","color"+counttext);
        append.style.width = '5em';
        var col = new jscolor.color(append);
        col.fromRGB(0, 0, 0);
        textfelder.appendChild(append);
        //Feld fÃ¼r SchriftgrÃ¶Ãße
        append = document.createElement("input");
        append.setAttribute("id","fontsize"+counttext);
        append.setAttribute("name","fontsize"+counttext);
        append.setAttribute("type","input");
        append.setAttribute("value","20");
        append.style.width = '2em';
        textfelder.appendChild(append);
              //Zeilenumbruch
        append = document.createElement("br");
        textfelder.appendChild(append);
              //Button Aktuallisieren
        append = document.createElement("input");
        append.setAttribute("id","akt"+counttext);
        append.setAttribute("name","akt"+counttext);
        append.setAttribute("type","button");
//      append.setAttribute("onmousedown","text_akt("+counttext+")");
        append.setAttribute("value","Aktualisieren");
        textfelder.appendChild(append);
        var tmp = counttext
        append.onmousedown = function(){text_akt(tmp);};
        //LÃ¶schen
        append = document.createElement("input");
        append.setAttribute("id","del"+counttext);
        append.setAttribute("name","del"+counttext);
        append.setAttribute("type","button");
        append.setAttribute("value","entfernen");
        //append.setAttribute("onmousedown","del("+counttext+")");
        textfelder.appendChild(append);
        var tmp = counttext
        append.onmousedown = function(){del(tmp);};
        //Feld fÃ¼r Koordiante
        append = document.createElement("input");
        append.setAttribute("id","posx"+counttext);
        append.setAttribute("name","posx"+counttext);
        append.setAttribute("type","hidden");
        textfelder.appendChild(append);
        //Feld fÃ¼r Koordiante
        append = document.createElement("input");
        append.setAttribute("id","posy"+counttext);
        append.setAttribute("name","posy"+counttext);
        append.setAttribute("type","hidden");
        textfelder.appendChild(append);
        //Feld fÃ¼r Schriftart Ãübergaeb
        append = document.createElement("input");
        append.setAttribute("id","fontface_trans"+counttext);
        append.setAttribute("name","fontface_trans"+counttext);
```

```
        append.setAttribute("type","hidden");
        append.setAttribute("value","Times");
        textfelder.appendChild(append);
        //DIV fÃür ZeichenflÃäche anlegen
        var container =  document.getElementById("textboxen");
        var append = document.createElement("div");
        append.setAttribute("id","textbox-"+counttext);
        //append.setAttribute("onmousedown","dragstart(this)");
        if (IE == true){
              append.setAttribute("className","textbox");
        } else {
              append.setAttribute("class","textbox");
        }
        container.appendChild(append);
        append.onmousedown = function(){dragstart(this);};
        document.getElementById("textbox-"+counttext).style.marginTop =
"-9.34579px";
        //ZeichenflÃäche fÃür den SVG text anlegen
        var button = document.getElementById("akt"+counttext)
        button.click();
        paper[counttext] =
Raphael(document.getElementById("textbox-"+counttext), 200, 25);
        paper[counttext].print(15,15, "neues textfeld",
paper[counttext].getFont("Times", 800), 20);
        //Neue DIV Box anlegen um darin die SVG genereiren zu kÃönnen
        counttext++;
        document.form.count.value = counttext;
}

function del(nummer){
        var textfelder = document.getElementById("textfelder");
        var remove = document.getElementById("text-w"+nummer);
        textfelder.removeChild(remove);
        var container = document.getElementById("textboxen");
        remove = document.getElementById("textbox-"+nummer);
        container.removeChild(remove);
}
```

6.1.3 Dragdrop.js

```
//Das Objekt, das gerade bewegt wird.
var dragobjekt = null;

// Position, an der das Objekt angeklickt wurde.
var dragx = 0;
var dragy = 0;

// Mausposition
var posx = 0;
var posy = 0;


function draginit() {
 // Initialisierung der ÃÜberwachung der Events
```

```
  document.onmousemove = drag;
  document.onmouseup = dragstop;
}


function dragstart(element) {
   //Wird aufgerufen, wenn ein Objekt bewegt werden soll.
  dragobjekt = element;
  dragx = posx - dragobjekt.offsetLeft;
  dragy = posy - dragobjekt.offsetTop;
}


function dragstop() {
  //Wird aufgerufen, wenn ein Objekt nicht mehr bewegt werden soll.
  if (dragobjekt!=null){
            if (dragobjekt.id != 'menu'){
                  var posx1 = dragobjekt.style.left;
                  var posy1 = dragobjekt.style.top;
                  var id = dragobjekt.id.substr(8);
                  var posx = document.getElementById("posx"+id);
                  var posy = document.getElementById("posy"+id);
                  posx.value  = posx1;
                  posy.value  = posy1;
            }
        dragobjekt=null;
  }
}


function drag(ereignis) {
      //Aktuelle GrÃ¶ÃŸe des Dokumentes ermitteln
      var width =
parseInt(document.getElementById("karte").style.width.substr(0,documen
t.getElementById("karte").style.width.length-2));
      var height =
parseInt(document.getElementById("karte").style.height.substr(0,docume
nt.getElementById("karte").style.height.length-2));
  //Wird aufgerufen, wenn die Maus bewegt wird und bewegt bei Bedarf
das Objekt
    posx = document.all ? window.event.clientX : ereignis.pageX;
       posy = document.all ? window.event.clientY : ereignis.pageY;
  if(dragobjekt != null) {
       var margin = dragobjekt.style.marginTop;
       margin = parseInt(margin);
       //Bewegung in X Richtung begrenzen
    if((posx - dragx)>= width+100){
            dragobjekt.style.left = width+100+"px";
       } else {
      if((posx - dragx)<= 60){
                  dragobjekt.style.left = "60px";
            } else {
            dragobjekt.style.left = (posx - dragx)+ "px";
      }
    }
```

```
      //Bewegung in Y Richtung begrenzen
    if((posy - dragy)>= height+100){
            dragobjekt.style.top = height-margin+100+"px";
       } else {
      if((posy - dragy)<= 60){
                  dragobjekt.style.top = 60-margin+"px";
            } else {
            dragobjekt.style.top = (posy - dragy)-margin+ "px";
      }
    }
  }
}
//-->
```

6.1.4 Hintergrund.js

```
//Nach dem Erfolgreichen Upload das Bild in der Auswaahl einblenden
function upload_finish(datei){
      if (document.getElementById("statusmsg").value == '1'){
            document.getElementById("eigen1").style.display = "inline";
            document.getElementById("eigen1").src     =
"bilder1/"+datei;
            document.getElementById("eigen").value = datei;
            document.getElementById("eigen").style.display = "inline";
            window.clearInterval(aktiv);
      }

}

function hintergrund1(geladen){
      if (bild==null){

      } else {
            hintergrund_remove1();
      }
            if (geladen != 0){
                  hg = background[0];
            } else {
                  var r = document.uploadform.Hintergrund_pfad;
                  for (i = 0;i < r.length;i++){
                        if (r[i].checked){ var hg = r[i].value;}
                  }
                  hg = 'bilder1/'+hg
            }
            // GrÃöÃße der Karte ermitteln und GrÃöÃße des Bildes
anpassen
            var karte = document.getElementById("karte");
            var sizeX = document.form.sizeX.value;
            var sizeY = document.form.sizeY.value;
            sizeX = sizeX * 40;
            sizeY = sizeY * 40;
            paper_bild =
Raphael(document.getElementById("karte"),sizeX,sizeY);
            bild = paper_bild.image(hg, 0, 0, sizeX,sizeY);
```

```
            var i = document.getElementById("hintergrund_remove");
            i.style.display = "inline";
            var i = document.getElementById("hintergrund_datei");
            i.value = hg;
}

function hintergrund_remove1(){
            bild.remove();
            bild = null;
            var i = document.getElementById("hintergrund_remove");
            i.style.display = "none";
            var i = document.getElementById("hintergrund_datei");
            i.value = "false";
      }
```

6.1.5 load.js

```
//Function wird aufgerufen, wenn der Benutzer auf "gespeichertes
Dokmment laden" klickt
function laden1 (){
      //Eingabe Formular wird um Feld fÃür Auftragsnummer erweitert
      //document.getElementById("auftrag_nr_input").style.display =
"table-row";
      document.getElementById("auftrag_nr_input").className = "";
      document.getElementById("auftrag_text").style.display = "block";
      //Auswahl der GrÃöÃße und Button um zur laden Ansicht zu
wechseln wird entfernt
      document.getElementById("remove1").style.display = "none";
      document.getElementById("remove2").style.display = "none";
      document.form.save.value = "true";
}

//Funktion die aufgerufen wird, wenn der Benutzer auf weiter klickt
function weiter1(){
      //Wenn Dokument geladen werden soll
      if (document.form.save.value == 'true'){
            //Abfrage ob alle Felder ausgefÃüllt wurden
            if (document.erstellen.vorname.value != '' &&
document.erstellen.nachname.value != '' &&
document.erstellen.auftrags_nr.value != ''){
                  //Anlegen aller JS Arrays in die das PHP Script die
Daten aus der Tabelle schreibt
                  text_string = new Array();
                  text_x = new Array();
                  text_y = new Array();
                  text_font = new Array();
                  text_size = new Array();
                  text_color = new Array();
                  text_nr = new Array();
                  paper_y = new Array();
                  paper_x = new Array();
                  background = new Array();
                  nachname = new Array();
                  vorname = new Array();
```

```
                    auftrag = new Array();
                    count1 = null;
                    //Das Formular wird an laden.php im iFrame
abgeschickt. Der Button ist fÃür den Benutzer unsichtbar
                    document.erstellen.submit.click();
              } else {
                    alert('Bitte Name und Auftragsnummer angeben');
              }
        //Wenn ein neues Dokument erstellt werden soll
        } else {
              //Abfrage ob alle Felder ausgefÃüllt wurden
              if (document.erstellen.vorname.value != '' &&
document.erstellen.nachname.value != ''){
                    document.form.vorname.value =
document.erstellen.vorname.value;
                    document.form.nachname.value =
document.erstellen.nachname.value;
                    var size1 = document.getElementById("size");
                    size1 = size1[size1.selectedIndex].text.split("x");
                    document.form.sizeX.value = size1[0]/10;
                    document.form.sizeY.value = size1[1]/10;
                    resize();
                    document.getElementById("teila").style.display =
"none";
                    document.getElementById("editor").style.display =
"inline";
                    var body = document.getElementById("bd");
                    var teila = document.getElementById("teila");
                    body.removeChild(teila);
              } else {
                    alert ('Bitte Namen angeben');
              }
        }
}

//Diese Funktion wird von laden.php aufgerufen wenn das
Datenbankabfrage beendet ist
function laden_finish(ergebnis){
        //Fall das ein Ergebnis gefunden wurden
        if (ergebnis == 'found'){
              //lÃänge das Text String Arrays wird ermittelt, dies
entspricht der anzahl der vorhanden Textfelder
              var count1 = text_size.length;
              var i = 0;
              //Auftragsnummer wird Ãübergeben
              document.getElementById("auftrag_nr").value = auftrag[0]
              //FÃür Alle Textfelder wird diese Schleife ausgefÃührt
              while (i < count1){
                    //Funktion add1 wird ausgefÃührt (entspricht klick
auf "Textfeld hinzufÃügen")
                    add1()
                    //Daten aus der Datenbank werden in die
Entsprechenden Felder eingetragen
                    document.getElementById("text"+i).value =
text_string[i];
                    document.getElementById("color"+i).value =
```

```
text_color[i];
                    document.getElementById("fontsize"+i).value =
text_size[i];
                    document.getElementById("posx"+i).value = text_x[i]
+"px";
                    document.getElementById("posy"+i).value = text_y[i]
+"px";
                    document.getElementById("textbox-"+i).style.left =
text_x[i]+"px";
                    document.getElementById("textbox-"+i).style.top =
text_y[i]+"px";
                    document.getElementById("nachname").value =
nachname[0];
                    document.getElementById("vorname").value =
vorname[0];
                    // da eine vor selection der Schriftart nur sehr
kompliziert mÃöglich ist wird darauf verzichtet und die Schriftart nur
in das ÃÜbergabefeld geschrieben ohne in der select liste den
richtigen Eintrag zu makieren
                    text_font_i = text_font[i];
                    //Funtion text_akt wird ausgefÃührt entspricht dem
klicken auf aktualisieren durch den Benutzer
                    text_akt(i);
                    i++;
              }
              document.form.sizeX.value = paper_x[0];
              document.form.sizeY.value = paper_y[0];
              resize();
              //falls ein Hintergrund vorhanden war wird er hier geladen
              if (background[0] != 'false'){
                    //Funktion hintergrund wird ausgefÃührt entspricht
dem klicken von Hintergrund anzeigen, Der Parameter 1 zeigt an das die
Funktion nach dem laden gestartet wurde
                    hintergrund1(1);
              }
              //Eintragen des Namen und der Auftragsnummer in die
entsprechenden Felder
              document.getElementById("editor").style.display = "inline";
              document.getElementById("teila").style.display = "none";
              document.getElementById("auftrag_nr_div").style.display =
"inline";
              //lÃöschen aller zum laden angelegten arrays
              text_font_i = null;
              delete text_string;
              delete text_x;
              delete text_y;
              delete text_font;
              delete text_size;
              delete text_color;
              delete text_nr;
              delete paper_y;
              delete paper_x;
              delete background;
              delete nachname;
              delete vorname;
              delete auftrag;
```

```
        }
        //Kein Ergebnis gefunden
        if (ergebnis == 'nothing'){
              alert('Auftrag wurde nicht gefunden')
              clearInterval(aktiv);
        }

}
```

6.1.6 pdf_anzeigen.php

```
<?php
$posL = $_POST[bild1];
$posT = $_POST[bild2];

require('pdf/fpdf.php');
define('FPDF_FONTPATH','pdf/font/');

#print_r($_POST);

//Funktion um HEX Farbwerte in RGB zu wandeln
function html2rgb($color)
{
    if ($color[0] == '#')
        $color = substr($color, 1);

    if (strlen($color) == 6)
        list($r, $g, $b) = array($color[0].$color[1],
                                 $color[2].$color[3],
                                 $color[4].$color[5]);
    elseif (strlen($color) == 3)
        list($r, $g, $b) = array($color[0].$color[0], $color[1].
$color[1], $color[2].$color[2]);
    else
        return false;

    $r = hexdec($r); $g = hexdec($g); $b = hexdec($b);

    return array($r, $g, $b);
}
//GrÃöÃße des Dokumentes
$size[1] = $_POST[sizeY];
$size[0] = $_POST[sizeX];
$size[1] = $size[1]*10;
$size[0] = $size[0]*10;

//ÃÜberprÃüfen welche Elemente alle vorhanden sind
$count = $_POST[count];
// Arrays fÃür Textboxen anlegen
$text = array();
$posx = array();
$posy = array();
$color = array();
//ÃÜbergeben posy Variable auf Inhalt ÃüberprÃüfen um die wieder
```

```
gelÃöschten Objekte zu umgehen
//Wenn das Obejkt vorhanden ist werden die Eigenschaften in die
jeweiligen Arrays geschrieben
for($i = 0; $i < $count; $i++ ) {
      if (isset($_POST['posy'.$i])){
            $posy[] = $_POST['posy'.$i];
            $text[] = $_POST['text'.$i];
            $posx[] = $_POST['posx'.$i];
            $color[] = $_POST['color'.$i];
            $fontface[] = $_POST['fontface_trans'.$i];
            $fontsize[] = $_POST['fontsize'.$i];
      }
}
$count = count($text);

//datenbankverbindugn aufbauen und Daten abspeichern
if (isset($_POST[auftrag])){
      $db = @new mysqli('localhost', 'user', 'pwd', 'db');
      if (mysqli_connect_errno()) {
      die ('Konnte keine Verbindung zur Datenbank aufbauen:
'.mysqli_connect_error().'('.mysqli_connect_errno().')');
      }
      $sql = 'SELECT auftrag FROM auftrag ORDER BY auftrag DESC LIMIT
1';
      $result = $db->query($sql);
      if (!$result) {
          die ('Etwas stimmte mit dem Query nicht: '.$db->error);
      }
      while($row = $result->fetch_assoc()) {
                  $auftrags_nr = $row[auftrag];
      }
      $auftrags_nr++;
      $posy[$i] = $db->real_escape_string($posy[$i]);
      $text[$i] = $db->real_escape_string($text[$i]);
      $posx[$i] = $db->real_escape_string($posx[$i]);
      $color[$i] = $db->real_escape_string($color[$i]);
      $fontface[$i] = $db->real_escape_string($fontface[$i]);
      $fontsize[$i] = $db->real_escape_string($fontsize[$i]);
      for ($i=0;$i<$count;$i++){
            $sql = "INSERT INTO auftrag
(text_x,text_y,text_font,text_size,text_color,text_nr,text_string,Auft
rag,paper_y,paper_x,background,name,vorname)
                        value('".$posx[$i]."','".$posy[$i]."','".
$fontface[$i]."','".$fontsize[$i]."','".$color[$i]."','".$i."','".
$text[$i]."','".$auftrags_nr."','".$_POST[sizeY]."','".
$_POST[sizeX]."','".
$_POST[hintergrund_datei]."','".trim($_POST[nachname])."','".trim($_PO
ST[vorname])."')";
            $result = $db->query($sql);
            if (!$result) {
                die ('Etwas stimmte mit dem Query nicht: '.$db->error);
            }
      }
}
```

```
$pdf=new FPDF('P','mm',$size);
define('FPDF_FONTPATH','pdf/font/');
require('pdf/font/makefont/makefont.php');

//MArgin recht so weit weg um zeilenumbruch zu verhindern
$pdf->SetMargins(-1000,-1000,-1000);
$pdf->SetAutoPageBreak(false);
$pdf->SetCompression(false);
$pdf->AddPage();
//Hintergrundbild
if ($_POST[hintergrund_datei] != 'false'){
      $pdf->Image($_POST['hintergrund_datei'],0,0,$size[0],$size[1]);
}
//MakeFont('arial.ttf','arial.afm','cp1252');
$pdf->AddFont('Superexpanded','','superexpanded.php');
//Textboxen erstellen
for ($i = 0; $i < $count; $i++){
      //Koordinaten umrechnen
      $posX = ((substr($posx[$i], 0, -2)-100)/40)*10;
      $posY = ((substr($posy[$i], 0, -2)-100)/40)*10;
      //SchriftgrÃöÃße Ausgleichen da sie im Browser anders als in der
PF ist
      $fontsize[$i]     = $fontsize[$i]*0.71;
      //Richtige Y Koordinate ermitteln
      $posY = $posY - $fontsize[$i]/10;
      $posX = $posX + 2.5;
      $posY = $posY - 4.1;
      //Negative Variablen umrechnen, da bei negatien Koordinaten von
unten bzw. recht gemessen wird
      if ($posX < 0){
            $posX =  $posX + ($size[0] * -1);
      }
      if ($posY < 0){
            $posY =  $posY + ($size[1] * -1);
      }
      //Zu Koordianten springen
      $pdf->SetXY($posX,$posY);
      //Farbwerte umwandeln
      $color1 = html2rgb($color[$i]);
      //SchriftgrÃöÃße umrechnen
      //Schriftart udn GrÃöÃße festlegen
      $style = '';
      if (eregi('kursiv',$fontface[$i])==true){
            $style = 'I';
            $fontface[$i] = explode(" ",$fontface[$i]);
            $fontface[$i] = $fontface[$i][0];
      }
      $pdf->SetFont($fontface[$i],$style,$fontsize[$i]);
      //Textfarbe einstellen
      $pdf->SetTextColor($color1[0],$color1[1],$color1[2]);
      //Text ausgeben
      $pdf->Write(15,$text[$i]);
      //$pdf->Cell(10,10,$text[$i],1,1,'L');

}
if (isset($_POST[auftrag])){
```

```
      echo 'Ihr Auftrage wurde erfolgreich entgegengenommen <br/>
                  <br/>Ihre Auftragsnummer lautet: '.$auftrags_nr.'
                  <br/>Vorname: '.$_POST[vorname].'
                  <br/>Nachname: '.$_POST[nachname].'
                  <br/><br/> Bitte notieren Sie sich diese Angaben um
das Dokument sp&auml;ter nochmals laden und bearbeiten zu k&ouml;nnen!
                  ';
} else {
      $pdf->Output();
}


?>
```

6.1.7 laden.php

```
<?php
if ($_POST[submit]){
      $db = new mysqli('localhost', 'user', 'pwd', 'db');
      if (mysqli_connect_errno()) {
            die ('Konnte keine Verbindung zur Datenbank aufbauen:
'.mysqli_connect_error().'('.mysqli_connect_errno().')');
      }
      //Datenbank abfrage
      $_POST[auftrags_nr] = $db-
>real_escape_string($_POST[auftrags_nr]);
      $_POST[nachname] = $db->real_escape_string($_POST[nachname]);
      $_POST[vorname] = $db->real_escape_string($_POST[vorname]);
      echo $_POST[vorname];
      $sql = ("SELECT * FROM auftrag WHERE Auftrag = '".
$_POST[auftrags_nr]."' AND name = '".$_POST[nachname]."' AND vorname =
'".$_POST[vorname]."'");
      $result = $db->query($sql);
      if (!$result) {
          die ('Etwas stimmte mit dem Query nicht: '.$db->error);
      }
      echo '<script type="text/javascript">';
      $i=0;
      while($row = $result->fetch_assoc()) {
                  //Ergebniss wird in die vorbereiteten JS Arrays
geschrieben
                  echo 'parent.top.text_string['.$i.'] = "'.
$row[text_string].'";';
                  echo 'parent.top.text_x['.$i.'] = "'.
$row[text_x].'";';
                  echo 'parent.top.text_y['.$i.'] = "'.
$row[text_y].'";';
                  echo 'parent.top.text_font['.$i.'] = "'.
$row[text_font].'";';
                  echo 'parent.top.text_size['.$i.'] = "'.
$row[text_size].'";';
                  echo 'parent.top.text_color['.$i.'] = "'.
$row[text_color].'";';
                  echo 'parent.top.paper_y['.$i.'] = "'.
```

```
$row[paper_y].'";';
                    echo 'parent.top.paper_x['.$i.'] = "'.
$row[paper_x].'";';
                    echo 'parent.top.background['.$i.'] = "'.
$row[background].'";';
                    echo 'parent.top.nachname['.$i.'] = "'.
$row[name].'";';
                    echo 'parent.top.vorname['.$i.'] = "'.
$row[vorname].'";';
                    echo 'parent.top.auftrag['.$i.'] = "'.
$row[Auftrag].'";';
                    $i++;
      }
      echo '</script>';
      //Wenn kein gefundenes ergebniss vor liegt wird die Funktion
laden_finish mit dem Wert nothing geladen
      if ($i < 1){
            echo '<script type="text/javascript">';
            echo "parent.laden_finish('nothing');";
            echo '</script>';
      //Wenn ein ergebniss vor liegt wird die Funktion laden_finish
mit dem wert found gestartet
      } else {
            echo '<script type="text/javascript">';
            echo "parent.laden_finish('found');";
            echo '</script>';
      }
}
?>
```

6.1.8 upload.php

```
<?php
if(isset($_FILES['upload'])) {
      move_uploaded_file($_FILES['upload']['tmp_name'], 'bilder1/'.
$_FILES['upload']['name']);
      chmod('bilder1/'.$_FILES['upload']['name'], 0755);
      echo "File upload successfull";
      echo '<script type="text/javascript">';
      echo "parent.top.document.getElementById('statusmsg').value =
'1';";
      echo '</script>';
}
?>
```

Literaturverzeichnis

Bücher

Heidinger, Albrecht: Was kann Web-to-Print wirklich? Software-Lösungen für Printbuyer und Dienstleister im Test. Böblingen 2008

Hochschulschriften

Förster, Thomas: Web-to-Print – Eine Analyse verschiedener Softwarelösungen aus Anwender- und Nutzersicht im Hinblick auf kleine und mittelständische Unternehmen.
Diplomarbeit, Hochschule Mittweida (FH), Mittweida 2007

Internetseiten

Webanalyse: Verbreitung von Webtechnologien und Plugins
http://www.webmasterpro.de/portal/webanalyse-technologien.html, 30.10.2009

Raphaël Webseite und kurze Beschreibung des Projektes
http://raphaeljs.com/, 30.10.2009

AskOxford.com English Dictioaries
http://www.askoxford.com/concise_oed/wysiwyg?view=uk, 04.11.2009

Über das World Wide Web Consortium (W3C) von Ian Jacobs am 05.11.2007
http://www.w3c.de/about/overview.html, 04.11.2009

Wikipedia: World Wide Web Consortium
http://de.wikipedia.org/wiki/World_Wide_Web_Consortium, 04.11.2009

Wikipedia: Cascading Style Sheets
http://de.wikipedia.org/wiki/Cascading_Style_Sheets, 04.11.2009

The Web Standards Project: Acid3 Browser Test
http://www.webstandards.org/action/acid3. 04.11.2009

Wikipedia: Acid3 Kompatibilität von Anwendungen
http://de.wikipedia.org/wiki/Acid_

%28Browsertests29#Kompatibilit.C3.A4t_von_Anwendungen_2, 04.11.2009

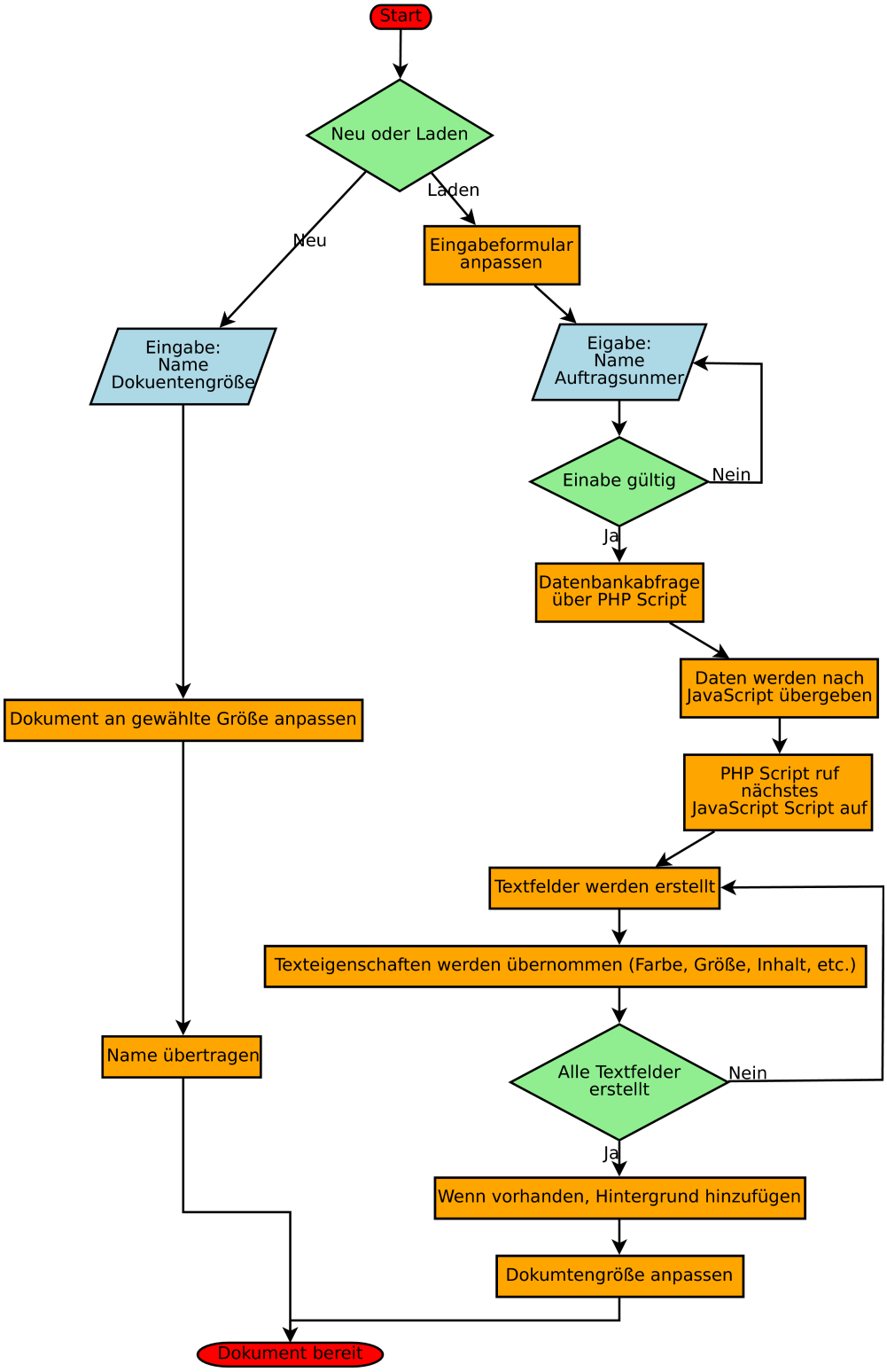
Adobe.com: Adobe Online Shop Deutschland
https://store5.adobe.com/cfusion/store/index.cfm?&store=OLS-DE&view=ols_prod&category=/Applications/FlashP&distributionMethod=FULL&nr=0#loc=de_de&store=OLS-DE&view=ols_prod&category=/Applications/FlashP, 30.11.2009

GNU: General Public License
http://www.gnu.org/licenses/gpl.html, 14.12.2009

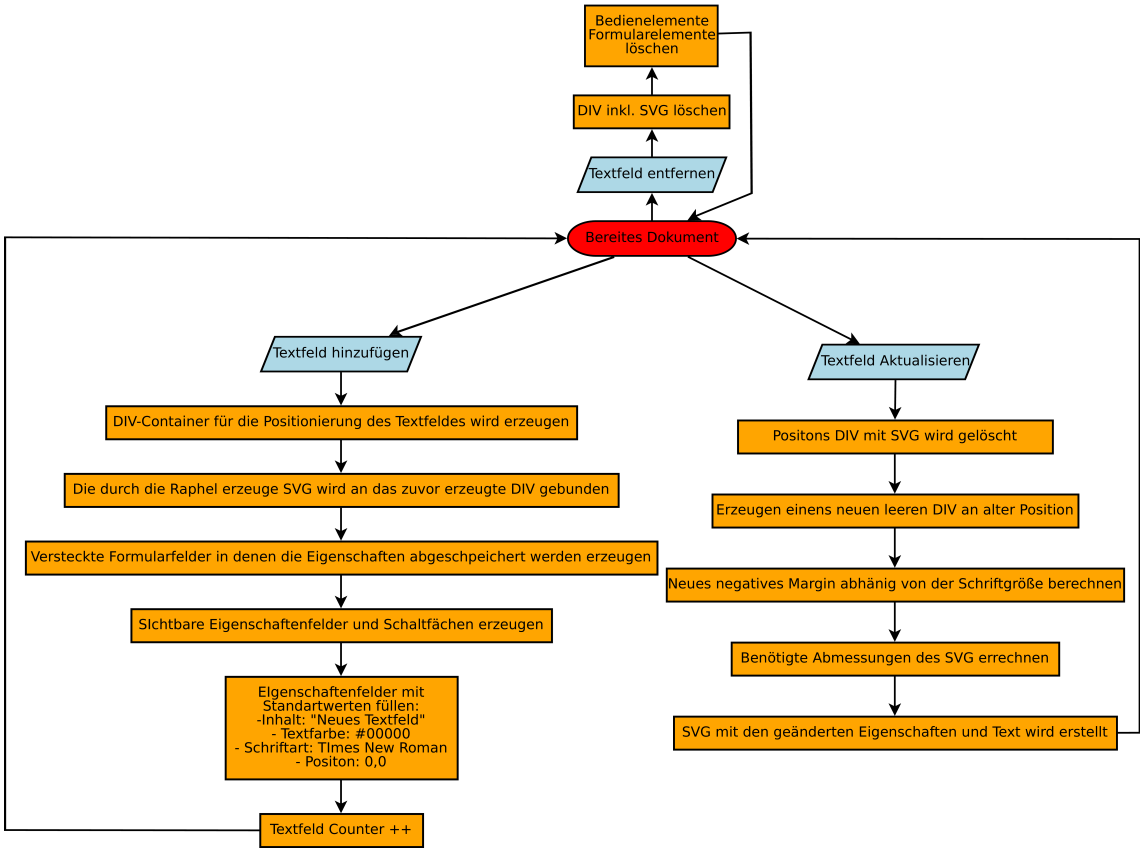
Anlagen

Anlagenverzeichnis

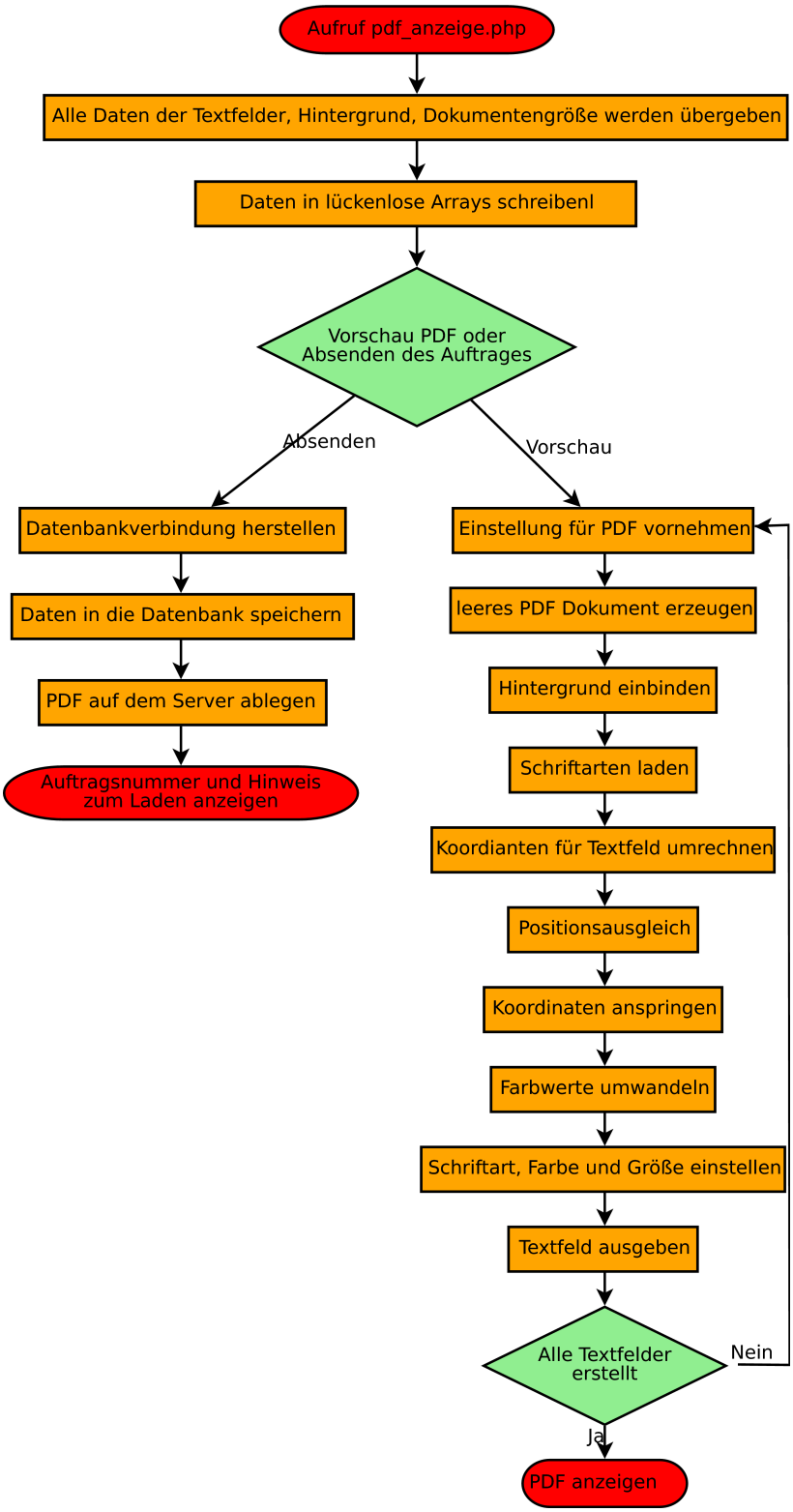
Anlage 1: Ablaufdiagramm Teil A:



Anlage 2 : Ablaufdiagramm Textfelder



Anlage 3 : Erstellen der PDF



Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Alle Teile, die wörtlich oder sinngemäß einer Veröffentlichung entstammen, sind als solche kenntlich gemacht. Die Arbeit wurde noch nicht veröffentlicht oder einer anderen Prüfungsbehörde vorgelegt.

Alexander Rothe

Leipzig, 25.01.2010