

HandHeld – mobiler Editor für Printprodukte



Diplomarbeit

Eingereicht an der:

HTL Perg für Informatik

Fachgebiet:

Programmieren

Betreuung:

Dipl. – Ing. Christian Aberger

Eingereicht von:

Samantha Plank

Christoph Mayrbäurl





Inhaltsverzeichnis

1. Impressum	8
2. Danksagung	10
3. Eidesstaatliche Erklärung.....	12
4. Einleitung	14
4.1 Kurzbeschreibung.....	14
4.2 Abstract.....	15
5. Projektdefinition.....	16
5.1 Auftraggeber	16
5.1.1 Persönliche Daten	16
5.2 Projektteam.....	16
5.2.1 Samantha Plank.....	17
5.2.2 Christoph Mayrbäurl	18
5.3 Projektumfeld.....	19
5.3.1 Schule	19
5.3.2 Betreuungslehrer	19
5.3.3 Ansprechpartner	19
5.3.4 Hierarchie.....	20
6. Ausgangslage und Zielsetzungen	21
6.1 Thema und Aufgabenstellung	21
6.1.1 Ist-Zustand	21
6.1.2 Problembereich	21
6.2 Zielsetzung	21
6.3 Funktionsumfang.....	22
6.3.1 Überblick Gesamtsystem.....	22
6.3.2 Funktionalitäten	23
7. Planung.....	24



7.1	Zeitplanung.....	24
7.2	Auflistung der Meilensteine.....	26
7.3	Projektstrukturplan.....	27
7.4	Konfigurationsplan.....	28
7.4.1	User.....	29
7.4.2	Mobilgerät.....	29
7.4.3	Internet.....	29
7.4.4	Abergerger Software GmbH.....	29
7.4.5	Versandunternehmen.....	29
8.	Verwendete Technologien.....	30
8.1	HTML.....	30
8.2	CSS.....	31
8.3	JavaScript.....	32
8.4	JQuery.....	32
8.5	AJAX.....	33
8.6	Fabric.js.....	34
8.7	Node.js.....	35
8.8	PHP.....	36
8.9	Apache HTTP Server.....	37
8.10	Dropbox.....	38
8.11	Eclipse.....	38
8.12	Phonegap.....	39
8.12.1	PhoneGap Kompatibilität:.....	40
8.13	Apache Cordova.....	41
8.14	Android.....	42
8.14.1	Android Architektur:.....	42
8.14.2	Android Versionen.....	44
9.	Realisierung.....	45
9.1	Installation.....	45



9.2	Darstellung	49
9.2.1	Produktübersicht	49
9.2.2	Informationsseite	50
.....	50
9.2.3	Produktdetail	51
9.2.4	Editor.....	52
9.2.5	Vorschau.....	54
9.2.6	Bestelldaten.....	55
9.3	Risiken	56
9.3.1	Neue Programmiersprachen.....	56
9.3.2	Neue Technologie Phonegap.....	56
9.3.3	Qualitätsverlust.....	57
9.3.4	Funktionsumfang zu groß.....	57
9.4	Verwendete Ressourcen	58
9.4.1	Software.....	58
9.4.2	Hardware.....	58
9.4.3	Orgware	58
9.5	Code.....	59
9.5.1	snapPicture() und getPicture()	59
9.5.2	list()	61
9.5.3	preview() und restore().....	62
9.5.4	data().....	63
9.5.5	clear()	64
9.5.6	Text hinzufügen.....	65
9.5.7	getText().....	66
9.5.8	getTextColor()	66
9.5.9	getBackgroundColor()	66
9.5.10	getFont().....	67
9.5.11	chkFormular()	67



9.5.12	postSVG.php	69
9.5.13	Cross Domain Requests	70
10.	Probleme und Lösungen	71
10.1	Allgemeine Probleme	71
10.1.1	Allgemeiner Problemfall 1 - Aufwandschätzung	72
10.2	Technische Probleme	73
10.2.1	Technischer Problemfall 1 - Cross Domain Request sind standardmäßig nicht erlaubt	73
10.2.2	Technischer Problemfall 2 - Android Versionen	74
11.	Qualitätssicherung	75
11.1	Probleme	75
11.2	Qualitätsmerkmale	75
12.	Testen	76
12.1	Allgemein	76
12.2	Übersicht Testfälle	77
12.3	Testfälle	78
12.3.1	Testfall 1: Aufruf der Übersicht	78
	78
12.3.2	Testfall 2: Infos anzeigen	79
12.3.3	Testfall 3: Editor aufrufen	80
12.3.4	Testfall 4: Bild und Text hinzufügen	81
12.3.5	Testfall 5: SVG erstellen	82
12.3.6	Testfall 6: Benutzerdaten eingeben	83
12.3.7	Testfall 7: Senden der Daten an den Server	84
13.	Glossar	85
13.1	SVG	85
13.2	XML	85
13.3	SMIL	85
14.	Resümee – Samantha Plank	86



15.	Resümee – Christoph Mayrbäurl	87
16.	Verzeichnisse	88
16.1	Abbildungsverzeichnis	88
16.2	Tabellenverzeichnis.....	89
17.	Quellen.....	90





1. Impressum

Schule

HTBLA Perg für EDV & Organisation

Machlandstraße 48

4320 Perg

Schuljahr

2014-2015

Klasse

5AHIF

Projekttitlel

Handheld - mobiler Editor für Printprodukte

Projektteam

Samantha Plank

Christoph Mayrbäurl

Betreuungslehrer

Dipl.-Ing. Christian Aberger

Auftraggeber

Aberger Software GmbH





2. Danksagung

Wir möchten uns bei allen Personen bedanken, welche uns während der Abwicklung dieser Diplomarbeit geholfen und unterstützt haben!

Als erstes möchten wir Herrn Professor Dipl.-Ing. Christian Aberger danken. Durch ihn wurde das Projekt erst möglich und er stand und stets mit Tipps und Ratschlägen zur Seite.

Auch den Professoren der HTL Perg wollen wir danken. Durch Sie konnten wir zusätzlich sehr viele Informationen sammeln und diese auch verwenden.

Herzlichen Dank!





3. Eidesstaatliche Erklärung

Perg, am

Im Zuge dieser Erklärung versichern wir, Samantha Plank und Christoph Mayrbäurl, dass die hier vorliegende Diplomarbeit, ohne Zuhilfenahme nicht angeführter Quellen und Hilfsmittel, selbstständig verfasst zu haben. Alle aus fremden Quellen übernommenen Gedanken sind als solche gekennzeichnet. Ein vollständiges Quellen- und Literaturverzeichnis ist am Ende des Dokumentes zu finden.

Die Arbeit wurde bisher weder in gleicher sowie in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Wir sind uns der Tatsache bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Samantha Plank

Christoph Mayrbäurl





4. Einleitung

4.1 Kurzbeschreibung

Oft schießen einem Ideen für Geschenke durch den Kopf, doch dummerweise ist man genau dann nicht zu Hause um sie zu notieren oder gleich am PC zu bestellen. Dieses Problem gehört nun dank der Applikation Handheld Print Product Editor der Vergangenheit an, denn nun ist es möglich, dass man egal wo und egal wann über sein Smartphone Fototassen, T-Shirts, Visitenkarten, etc. zu designen und bestellen kann, sofern eine aktive Internetverbindung besteht.

Unsere Diplomarbeit befasst sich mit der Programmierung von einer App, die es ermöglicht schnell einfach von überall Printprodukte zu gestalten und zu bestellen.

Bei unserer Lösung wird zuerst das Produkt ausgewählt, danach bekommt man eine kleine Übersicht zu dem gewünschten Artikel. Überzeugt einen das Produkt, so kann man dies mit dem Editor gestalten. Möglichkeiten sind: Fotos von seinem Smartphone zu wählen oder ein neues Bild aufzunehmen oder einen gewünschten Text zuzufügen. Im nächsten Schritt bekommt man eine Vorschau vom gestalteten Produkt, wenn das Ergebnis noch nicht den Wünschen entspricht kann der Artikel noch weiter verschönert werden. Entspricht nun das designte Produkt ihren Vorstellungen so wird dies abschließend an die Aberger Software GmbH gesendet. Dort wird die Bestellung umgesetzt und danach per Post an den Kunden gesendet.



4.2 Abstract

Often it happens, that the idea for a gift appears in your mind, therefore it is a pity, that you are usually not at home to note it or eventually order it immediately when this is the case. But thank goddess, this problem is now a thing of the past because our Application "Handheld Print Product Editor" enables us to design and order gift ideas via Smartphone, no matter where you are. All you need is a Smartphone and an active Internet Connection.

Our degree dissertation covers the programming of an Application which enables us to design and order Print Products independently of the place you are.

With our solution the first thing you do is to select the desired Product, then you are navigated to a simple overview of the article you chose. If the product satisfies your needs you can step forward to the editor, where you are able to create a design. The editor enables you to add text and pictures, either from your storage or you may as well take a new picture directly in the application. The pictures and your chosen text is freely resizable and replaceable. The next step you are led to is the preview, where you get a final overview of your design. If the design comes up to your expectations you send it to the "Aberger Software GmbH" which will produce the selected product with your individual design and ship it to you via delivery service. If you are not pleased with your design you can head back to the editor and improve it.



5. Projektdefinition

5.1 Auftraggeber

Unser Auftraggeber ist die Aberger Software GmbH, die unter der Leitung von Herrn Dipl. –Ing. Christian Aberger steht. Zudem ist Herr Professor Aberger unser Programmierlehrer seit der vierten Klassen und steht als Betreuungslehrer zur Seite.

Die Aberger Software GmbH beschäftigt sich in erster Linie mit der Entwicklung von Software. Produkte von ihnen sind zum Beispiel ein Fotobestellsystem sowie Print Softwarelösungen.

5.1.1 Persönliche Daten

Aberger Software GmbH	
Adresse	Softwarepark 37 4232 Hagenberg
E-Mail	office@abergger.at
Telefon	+ 43 720 348 450
Homepage	www.abergger.at

Tabelle 1: Kontaktdaten Aberger Software GmbH

5.2 Projektteam

Diese Diplomarbeit wurde im Rahmen der Matura an der HTL in Perg im Schuljahr 2014/2015 von den folgenden Diplomanden abgeschlossen:



5.2.1 Samantha Plank

Samantha Plank	
Staatsangehörigkeit	Österreich
Geburtsdatum	02.02.1996
Adresse	Riederbachstraße 4 4310 Langenstein
Telefon	0650 994 41 96
E-Mail	samantha.plank@hotmail.com
Ausbildung	2010 – 2015 HTL Perg 2006 – 2010 HS Mauthausen 2002 – 2006 VS Langenstein
Sprachen	<ul style="list-style-type: none">• Deutsch• Englisch
EDV-Technologien	<ul style="list-style-type: none">• C• C#• Java• SQL• PL-SQL• APEX• Oracle-Datenbanken• MySQL• Netzwerktechnik

Tabelle 2: Daten Samantha Plank



5.2.2 Christoph Mayrbäurl

Christoph Mayrbäurl	
Staatsangehörigkeit	Österreich
Geburtsdatum	03.01.1994
Adresse	Brausmühlstraße 2 4481 Asten
Telefon	0660 532 46 06
E-Mail	cchristoph.m@gmail.com
Ausbildung	2008 – 2015 HTL Perg 2004 – 2008 Gymnasium Enns 2000 – 2004 VS Asten
Sprachen	<ul style="list-style-type: none">• Deutsch• Englisch
EDV-Technologien	<ul style="list-style-type: none">• C• C#• Java• SQL• PL-SQL• APEX• Oracle-Datenbanken• MySQL• Netzwerktechnik

Tabelle 3: Daten Christoph Mayrbäurl



5.3 Projektumfeld

5.3.1 Schule

Diese Diplomarbeit wurde im Rahmen der Matura an der HTL in Perg, Schwerpunkt Informatik, im Schuljahr 2014/2015 durchgeführt.

5.3.2 Betreuungslehrer

Betreut wurde diese Diplomarbeit von unserem Lehrer DI Christian Aberger. Er war der Ansprechpartner bei organisatorischen Dingen und hatte eine beratende Funktion inne. Wie schon erwähnt ist Professor Aberger auch der Inhaber der Aberger Software GmbH.

5.3.2.1 Persönliche Daten

DI Christian Aberger	
Adresse	Softwarepark 37 4232 Hagenberg
E-Mail	christian@abergger.at

Tabelle 4: Kontaktdaten DI Christian Aberger

5.3.3 Ansprechpartner

Unser Ansprechpartner bei der Aberger Software GmbH ist Martin Brandstätter.



5.3.4 Hierarchie

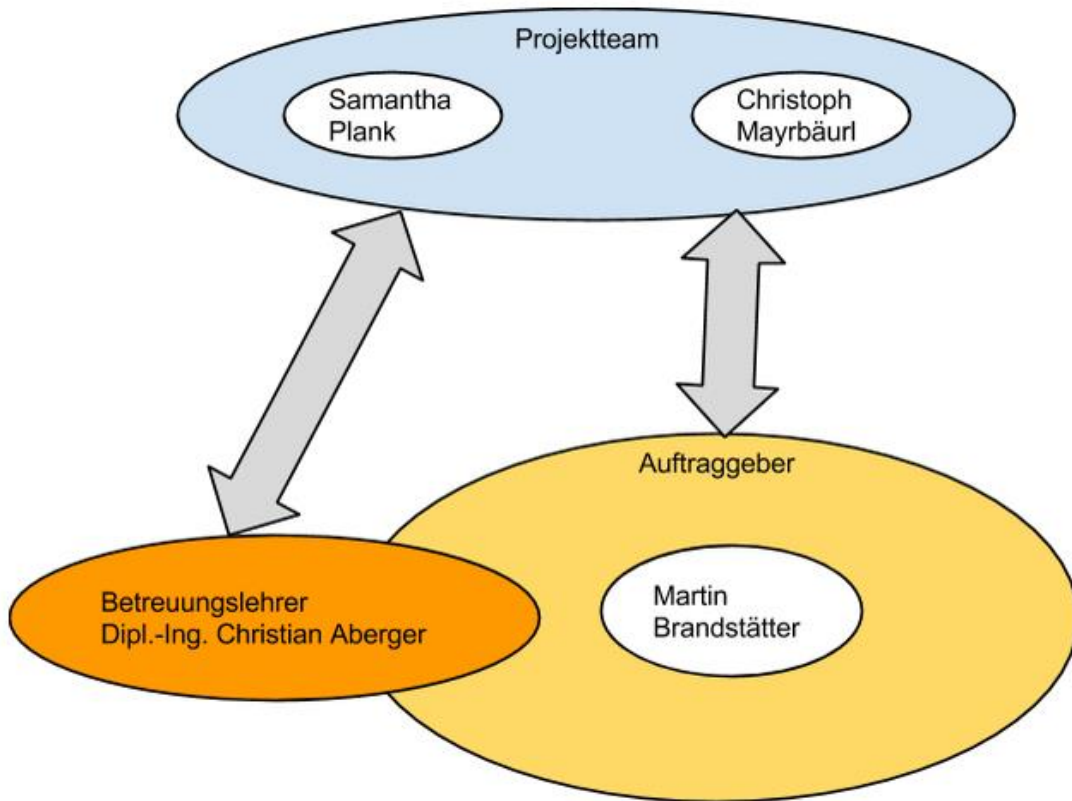


Abbildung 1: Umfeld Hierarchie



6. Ausgangslage und Zielsetzungen

6.1 Thema und Aufgabenstellung

6.1.1 Ist-Zustand

Oft fällt einem unterwegs ein, dass man noch ein Geschenk benötigt, aber die Zeit zu knapp ist, um zu warten bis man zu Hause ist, um dieses Mitbringsel zu gestalten und zu bestellen. Oft vergisst man auch seine tolle Idee wieder bis man Daheim ist.

6.1.2 Problembereich

Dies ist sehr umständlich, da man extra nach Hause fahren muss um die Bestellung am Computer abzuwickeln.

6.2 Zielsetzung

- App muss intuitiv bedienbar sein
- Ablauf soll einfach sein
- Upload der Bilder muss direkt vom Mobilgerät erfolgen
- Der Text muss frei positionierbar sein
- Das Foto muss frei positionierbar sein
- Es muss eine Produktauswahl vorhanden sein
- Bsp.: Tassen, T-Shirts, Keilrahmenbilder, Fotobücher, etc.
- Der Upload des fertigen Produkts muss als SVG erfolgen



6.3 Funktionsumfang

6.3.1 Überblick Gesamtsystem

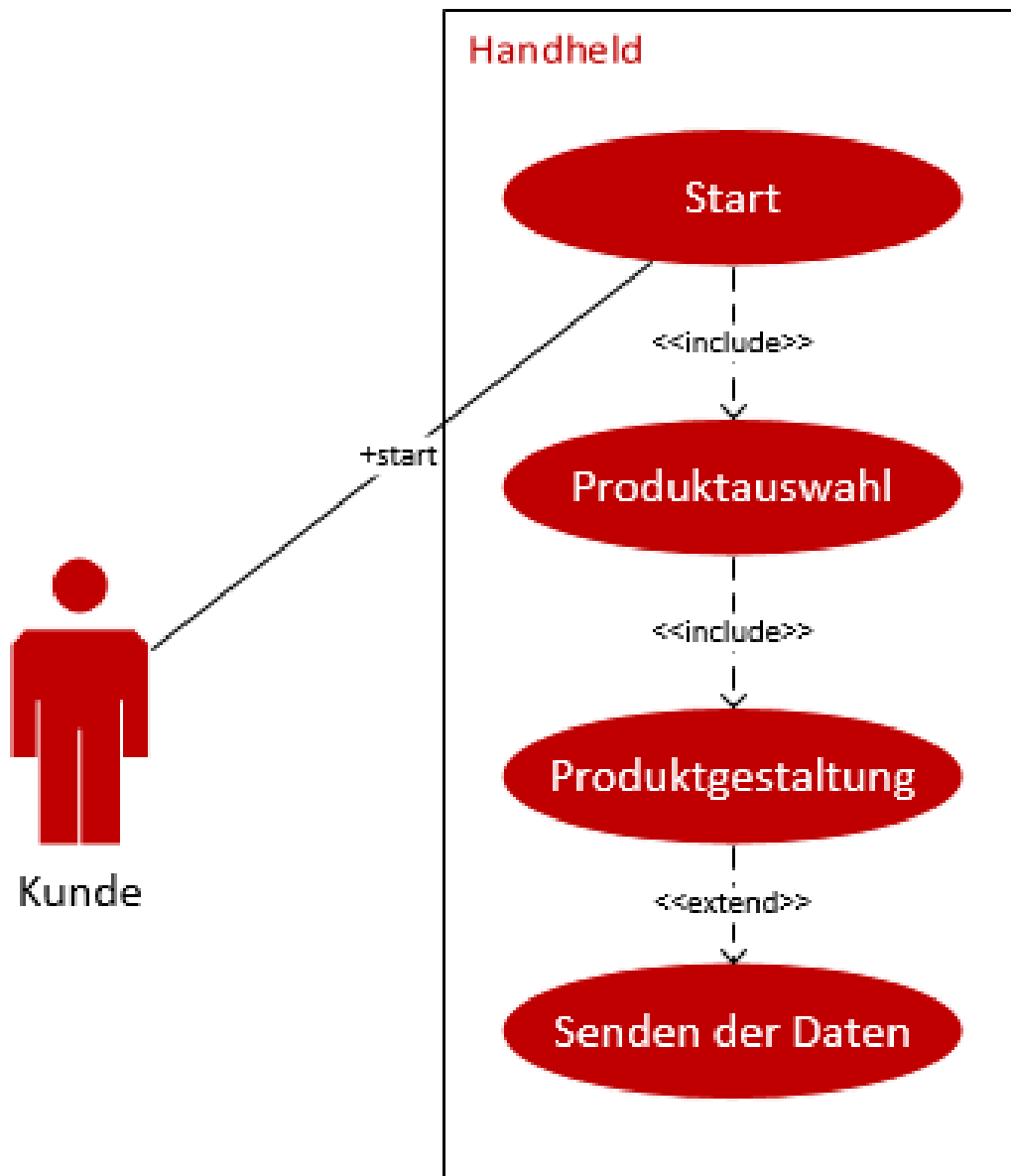


Abbildung 2: Überblick Gesamtsystem - USE CASE



6.3.2 Funktionalitäten

6.3.2.1 Produktauswahl

Als erstes wird eine Liste von Produkten aufgelistet. Der User kann nun ein Produkt auswählen und mehr Informationen über den Artikel erhalten.

6.3.2.2 Produktgestaltung

Im nächsten Schritt kann der Benutzer das Produkt gestalten, indem er Bilder aus der Galerie wählt, Fotos aufnimmt, oder einen Text hinzufügt. All diese Objekte sind durch Fabric.js frei positionierbar und veränderbar. Die Auswahl und die Aufnahme von Fotos erfolgt mit Hilfe von Cordova zur Verfügung gestellten Methoden.

6.3.2.3 Senden der Daten an den Server

Danach erscheint eine Vorschau des gestalteten Produkts. Als nächstes erscheint ein Eingabeformular für die Kaufdaten. Nach dem ausfüllen wird das Produkt mittels SVG, und die Benutzerdaten per JQuery und AJAX an den Sever versendet.



7. Planung

7.1 Zeitplanung

Die Planung ist eines der wichtigsten Instrumente in der Projektentwicklung und sollte deshalb stets sorgfältig und genau durchgeführt werden. Technische Hilfe in der Zeitplanung lieferte uns das Programm GanttProject, welches uns bereits seit Beginn des Gegenstandes Projektentwicklung begleitete.

Diese Grafik zeigt die Meilensteine und die von uns zu tätigen Aufgaben, um das Ziel für die Fertigstellung unserer Diplomarbeit zu erreichen. Die Verantwortlichkeiten wurden von uns je nach Vereinbarung ausgewählt, es hat aber jeder immer an jedem Meilenstein mitgearbeitet, d.h. jeder wurde in jedes Thema und in jeden Teil unserer Diplomarbeit involviert. Ein Vorgang im Projekt unterscheidet sich jedoch von einem Meilenstein, da dieser keinen Anfangs- und Endzeitpunkt hat. Ein Meilenstein ist ein Ereignis und hat somit einen Zeitpunkt. In der Praxis hängt ein Meilenstein oft mit einem finanziellen Zuschuss zusammen, welcher die Weiterentwicklung eines Projektes ermöglicht.

In der nachstehenden Grafik sind die Meilensteine mit Rauten gekennzeichnet.



Abbildung 3: Zeitplan



7.2 Auflistung der Meilensteine



Abbildung 4: Auflistung Meilensteine



7.3 Projektstrukturplan

Ein weiterer wichtiger Punkt in der Planung eines Projektes ist der Projektstrukturplan, welcher das Projekt in mehrere kleine Teilaufgaben gliedert und somit für eine bessere Übersicht auf das Gesamtprojekt sorgt.

Projektstrukturpläne können hierarchisch beliebig weit unterteilt werden, der nachfolgende Plan unserer Diplomarbeit zeigt nur eine grobe Einteilung in die Hauptfunktionalitäten, sowie einen kleinen Teil der Dokumentationen, die in unserem Projekt anfielen

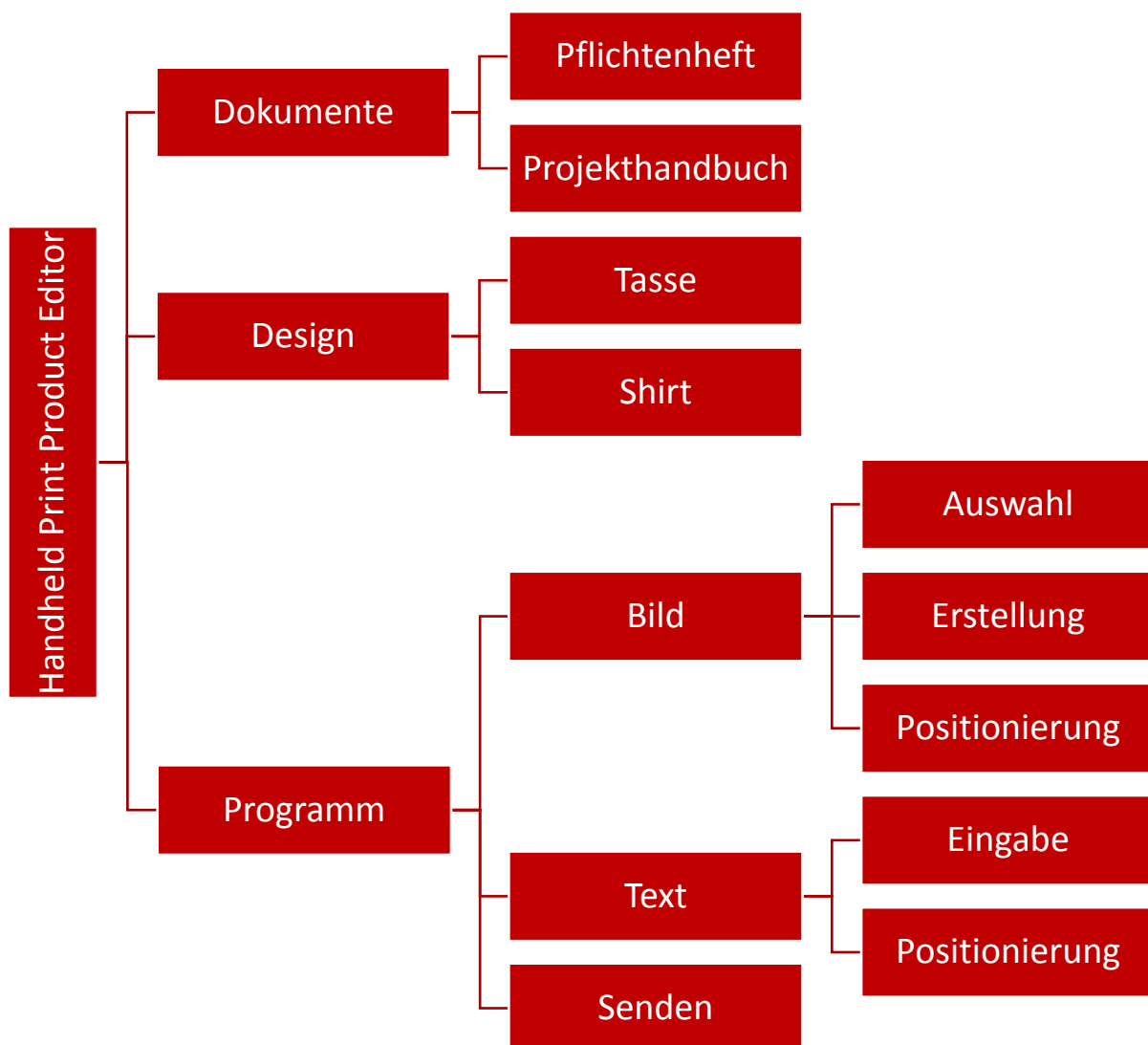


Abbildung 5: Projektstrukturplan



7.4 Konfigurationsplan

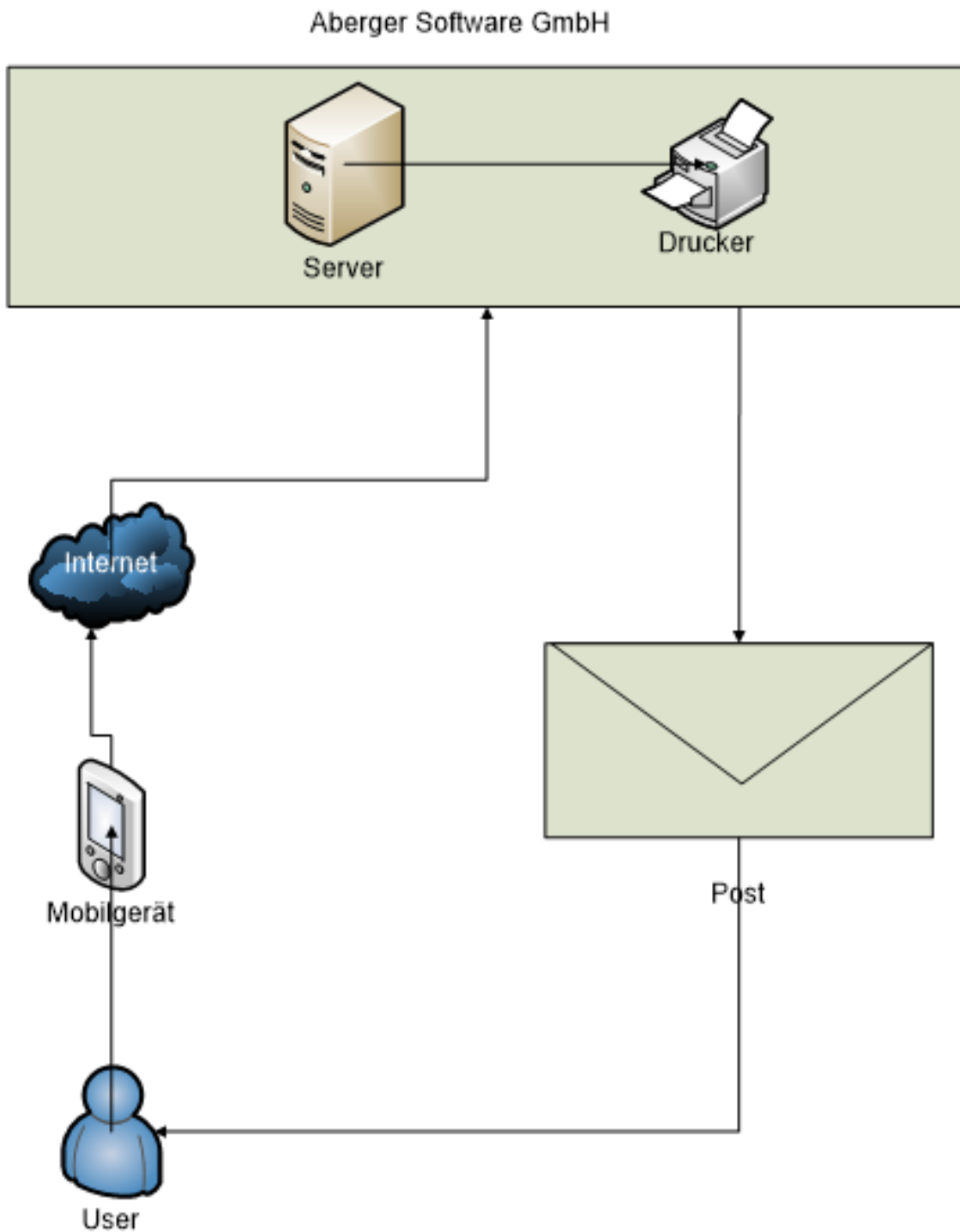


Abbildung 6: Konfigurationsplan



7.4.1 User

Der User ist eine Privatperson welche den Wunsch verspürt, sich über ihr Smartphone oder Tablet ein Druckprodukt zu gestalten und zu bestellen. Er ist auch derjenige welcher später bezahlt.

7.4.2 Mobilgerät

Das Mobilgerät ist ein Internetfähiges Smartphone oder Tablet, welches die Applikation Handheld Print Product Editor installiert hat. Auf diesem Gerät wird das Produkt ausgewählt, designed und bestellt. Es dient des Weiteren als Medienspeicher für die zur Verfügung stehenden Bilder beim Erstellen des Druckproduktes.

7.4.3 Internet

Das Internet dient als Schnittstelle vom Anwender zum Anbieter. Es wird benötigt um den Produktentwurf an den Anbieter zu senden, die Zahlung zu tätigen und die Informationen des Kunden an den Anbieter zu übermitteln.

7.4.4 Aberger Software GmbH

Die Aberger Software GmbH fungiert, wie bereits oben erwähnt, als unser Auftraggeber.

Die eingehenden Produkthanfragen werden von der Aberger Software GmbH entgegen genommen und bearbeitet.

7.4.4.1 Server:

Der Server nimmt die Anfrage des Kunden entgegen. Er liest die Daten des Kunden aus, überprüft den Zahlungsvorgang und sendet den gewünschten Entwurf zur Verarbeitung an die Drucker der Aberger Software GmbH weiter.

7.4.4.2 Drucker:

Hier wird der erstellte Entwurf auf das ausgewählte Produkt gedruckt.

7.4.5 Versandunternehmen

Liefert das vom Drucker fertiggestellte Produkt von der Aberger Software GmbH zum Kunden.



8. Verwendete Technologien

8.1 HTML

HTML ist die Abkürzung von dem englischen „Hypertext Markup Language“. Mit HTML wird eine Struktur aufgebaut, um digitale Inhalte darzustellen. Inhalte können als Beispiel Bilder oder Texte sein. Aufgrund dessen ist HTML eine textbasierte Auszeichnungssprache.

Mit HTML werden die Überschriften oder Aufzählungen definiert. Mit den Tags wird die Struktur einer Webseite

vordefiniert. Der Browser liest diese Struktur und zeigt die Inhalte dem entsprechen an.

Zudem kann man weitere Informationen einer Webseite speichern. Diese Informationen bezeichnet man als Meta-Informationen, diese dienen dazu weitere spezielle Daten zu speichern. Daten können zum Beispiel der Name des Autors sein. Auch der zusammengefasste Inhalt der Seite kann so gesichert werden.



Abbildung 7: HTML Logo



8.2 CSS

CSS heißt ausgeschrieben „Cascading Style Sheets“. Wie der Name schon vermuten lässt geht es hierbei um das „Stylen“, also das Gestalten von Webseiten.

CSS wird verwendet um Programmiercode zu vereinfachen und die Arbeit zu erleichtern. Früher musste man für jeden einzelnen Satz immer wieder die Farbe oder die Position bestimmen. Dies ist händisch ein sehr großer Aufwand.



Abbildung 8: CSS Logo

Zudem wird dadurch auch die Datenmenge der Webseite, wenn alles in der HTML-Datei geschrieben wird, was das Laden der Webseite verlangsamt.

Durch CSS kann man eigene Klassen oder ID's für bestimmte HTML-Elemente anlegen und diese immer wieder verwenden. So erspart man sich gleiche Formatierungen immer wieder zu definieren.

Hier die Vorteile von CSS in der Übersicht:

- die Arbeit für die Gestaltung wird vereinfacht
- der Code wird übersichtlicher
- ein erstelltes Design kann schnell immer wieder verwendet werden auch für andere HTML-Lösungen
- Positionierung von Elementen (Bilder) wird vereinfacht
- Webseiten werden schneller geladen, durch externe Dateien
- Fehler im Design sind leichter zu entdecken



8.3 JavaScript

JavaScript ist eine clientseitige Script-Sprache. Sie wurde ursprünglich fürs dynamische HTML entwickelt. JavaScript ermöglicht Benutzerinteraktionen auszuwerten. Das heißt Inhalte zu verändern, nachzuladen oder zu generieren. Also HTML und CSS auszubauen. Es

wird vom Browser interpretiert und ausgewertet.



Abbildung 9: JavaScript Logo

8.4 JQuery

JQuery ist ein plattformunabhängiges Framework für JavaScript und steht als Open-Source-Software zur Verfügung.



JQuery ist eine umfassende Klassenbibliothek, durch die der Zugriff auf das Document Object Model (DOM) realisiert wird.

Durch die bereit gestellten Funktionen wird die Manipulation, Behandlung von Events, die Interaktion mit Ajax und die Animation von HTML-Seiten erheblich vereinfacht.

Abbildung 10: jQuery Logo

Da JQuery die Abfragen von HTML-Elementen mit Selektoren erlaubt. Durch diese Selektoren kann man verschiedene Befehle auf die Elemente anwenden.

Die Verwendung von jQuery hat folgende Vorteile:

- Code und Zeit werden reduziert
- das Warten auf das vollständigen laden eines Dokuments entfällt
- JavaScript wird um zahlreichen Funktionen erweitert



8.5 AJAX

AJAX ist die Abkürzung für „Asynchronous JavaScript and XML“ und wird verwendet um Daten mit einem Server auszutauschen und ist

daher keine Programmiersprache.

Zudem kann man Teile einer

Webseite zu aktualisieren ohne alles neu zu laden.



Abbildung 11: AJAX Logo



8.6 Fabric.js

Fabric.js ist eine JavaScript Library, die den Umgang mit dem HTML5-Element `<canvas>` sehr vereinfacht.



Abbildung 12: Fabric.js Logo

Fabric ist ein Open-Source-Projekt was unter der MIT-Lizenz bereitgestellt wird. Sie stellt ein SVG- und JSON-Parser und ein Objektmodell zur Verfügung.

Weiters unterstützt es Node.js und die Entwicklung mobilen Endgeräten.

Mit Fabric kann man neue Objekte im Canvas erstellen und anordnen. Zudem kann man einen Text hinzufügen und bearbeiten. Das heißt die Schriftart, die Farbe und Größe ändern.

Außerdem ist es möglich die Canvas-Elemente zu skalieren und zu drehen. Auch Animationen und Effekte können den Objekten zugeordnet werden.

Eine weitere Funktion wird gewährleistet, das freie zeichnen im Canvas. Hier kann man auch die Farbe und Dicke bestimmen.



8.7 Node.js

Node.js ist ein Programmierframework und

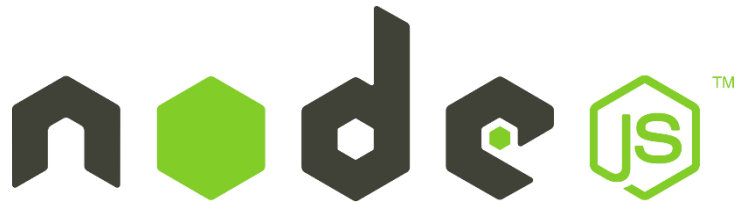


Abbildung 13: Node.js Logo

basiert auf JavaScript. Das

Hauptanwendungsgebiet ist

die Webserver-Programmierung. Node wurde 2009 von Ryan Dahl entwickelt.

Zudem basiert Node.js auf Googles V8 Engine, welche bei Google Chrome zum Einsatz kommt.

Node ermöglicht ganze Web-Apps zu bauen. Auch ist es ressourcenschonend da Node.js Event-basiert ist. Zudem bietet es auch einen eigenen Webserver.

Das heißt es kann das ganze Routing übernommen werden. Dies ist für Single-Page-Apps sehr praktisch.



8.8 PHP

PHP ist eine Sprache mit der ein Webserver gesteuert werden kann. Sie ist eine Open-Source-Software und wird unter der PHP-Lizenz verbreitet. PHP steht für Hypertext Preprocessor. Die Syntax ist an C und Perl angelehnt.

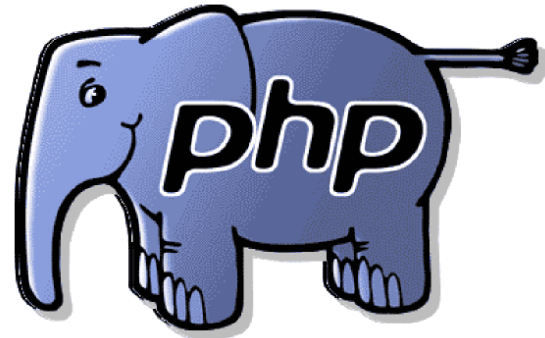


Abbildung 14: Php Logo

Der Code wird zum Zeitpunkt des

Aufrufst vom Webserver in Maschinencode, den der Server verwendet, umgerechnet. PHP-Programme sind plattformunabhängig und können auf unterschiedlichen Systemen ausgeführt werden.

Im Großen und Ganzen wird PHP für die Entwicklung dynamischer Webseiten oder Webanwendungen verwendet.

Positive Aspekte von PHP sind die Datenbankunterstützung, die Internet-Protokolleinbindung und die vielen Funktionsbibliotheken.



8.9 Apache HTTP Server

Der Apache HTTP Server ist ein freies Produkt der Apache Software Foundation und ist laut Netcraft.com der meistbenutzte Webserver im Internet.

Die Servertechnologie ist modular aufgebaut, was eine sehr spezielle Konfiguration ermöglicht. Die häufig auf dem Apache HTTP Server angewandte Skriptsprache PHP muss per Modul eingebaut oder über eine CGI-Schnittstelle integriert werden.



Abbildung 15: Apache HTTP Server Logo



8.10 Dropbox

Dropbox ist ein Filehosting-Dienst von dem Unternehmen Dropbox Inc. Durch Dropbox wird ermöglicht Dateien online zu speichern und von jedem ans Internet angeschlossenen Computer diese Dateien abzurufen.

So wird der Datenaustausch zwischen mehreren Personen gewährleistet.



Abbildung 16: Dropbox Logo

8.11 Eclipse

Eclipse ist ein Programmierwerkzeug zur Entwicklung von Software. Früher diente Eclipse als Entwicklungsumgebung für die Programmiersprache Java. Heute wird es für



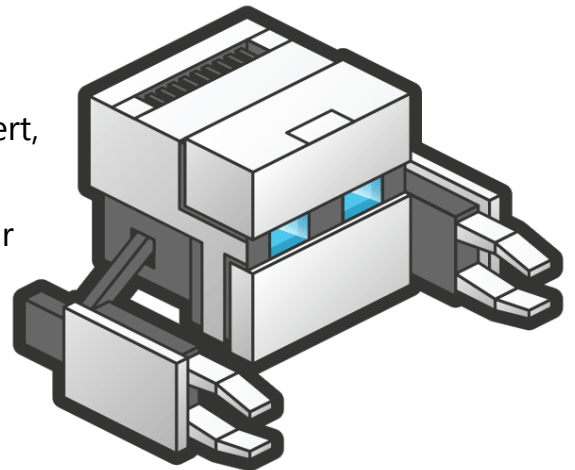
Abbildung 17: Eclipse Logo

verschiedene Entwicklungsaufgaben eingesetzt.



8.12 Phonegap

Phonegap ist ein Framework zur Erstellung von Applikationen für mobile Endgeräte. PhoneGap wurde ursprünglich von der Firma Nitobi produziert, welche 2011 von Adobe Systems gekauft wurde. Phonegap ermöglicht es, Anwendungssoftware für mobile Endgeräte mit JavaScript, HTML5 und CSS 3 zu schreiben, anstelle von gerätespezifischen Programmiersprachen wie



Objective-C (iOS) oder Java (Android).¹

Abbildung 18: Phonegap Logo

Die dabei entstehenden Applikationen sind hybride Applikationen. Das bedeutet, sie sind weder native Applikationen, weil das Layout mittels Webtechnologie und nicht mit nativen User Interface Frameworks gemacht wird, noch sind sie webbasierte Applikationen, da sie als Apps über die Verkaufsportale der Betriebssystem Hersteller verteilt werden können und auf die Programmierschnittstellen der Betriebssysteme der Endgeräte zugreifen können.

¹ (Wikipedia, 2015)



8.12.1 PhoneGap Kompatibilität:

In der folgenden Grafik ist ersichtlich, auf welche Funktionen der mobilen Endgeräte via PhoneGap zugegriffen werden kann.

	 iPhone / iPhone 3G	 iPhone 3GS and newer		 OS 4.6-7	 OS 5.x	 OS 6.0+			 SYMBIAN
Geschwindigkeitsmesser	✓	✓	✓	✗	✓	✓	✓	✓	✓
Kamera	✓	✓	✓	✗	✓	✓	✗	✗	✓
Kompass	✗	✓	✓	✗	✗	✗	✗	✗	✗
Kontakte	✓	✓	△	✗	✓	✓	✗	✓	✓
Dateien	✗	✗	✓	✗	✓	✓	△	✗	✗
GPS	✓	✓	✓	✓	✓	✓	✓	✓	✓
Medien (Audioaufnahme)	△	△	✓	✗	✗	✗	✗	△	✗
Benachrichtigung (Ton)	✓	✓	✓	✓	✓	✓	✓	✓	✗
Benachrichtigung (Vibration)	✓	✓	✓	✓	✓	✓	✗	✓	✓
Speicher	✓	✓	△	✗	△	✓	✓	✗	✗

Abbildung 19: Phonegap Kompatibilität



8.13 Apache Cordova

Apache Cordova ist eine Ansammlung von Geräte APIs. Es erlaubt dem Entwickler via Javascript auf die nativen Funktionen von mobilen Endgeräten zuzugreifen. Beispiele hierfür wären, wie in unserer Diplomarbeit die Kamera, jedoch können auch Funktionen für den Geschwindigkeitsmesser, Speicheroperationen und vieles mehr aufgerufen werden. Der große Vorteil bei der Verwendung von Apache Cordova ist, dass man alles in Javascript realisiert, man benötigt keine Kenntnisse in den, dem Gerät zugrundeliegende, Programmiersprache (Java, Objective-C, ...).



Abbildung 20: Apache Cordova Logo

Durch die Konsistenz der Javascript APIs, muss man für die verschiedenen Systeme nur einen Sourcecode schreiben, dieser ist dann (mit minimalen Änderungen) mit mehreren Systemen kompatibel.

Obwohl bei Cordova die Applikationen ohne nativen Code geschrieben werden, so sind sie doch echte Apps, und können somit auch in den App Stores der einzelnen Plattformen publiziert werden.



8.14 Android

Android ist sowohl ein Betriebssystem als auch eine Software-Plattform für mobile Geräte wie Smartphones, Netbooks und Tablet Computer.

Android ist im Bereich der Smartphone-Betriebssysteme klarer Marktführer (71% stand Jänner 2015). Und ist daher klarerweise eine Plattform auf der unsere Applikation laufen muss.



Abbildung 21: Android Logo

Des Weiteren können Android-Applikationen mit PhoneGap programmiert werden.

Um nun für Android Smartphones Programme entwickeln zu können, mussten wir nur mehr das Android Software Development Kit (Android SDK) runterladen. In diesem SDK ist sowohl ein Android-Emulator als auch ein modifiziertes Eclipse enthalten.

8.14.1 Android Architektur:

Android basiert auf dem Linux-Kernel 2.6 (bei neueren Versionen aktuell auf 3.x) und wurde von der Open Handset Alliance (Hauptmitglied: Google) entwickelt. Android ist eine freie Software, die quelloffen entwickelt wird. Der Linux-Kernel ist für die Speicherverwaltung und Prozessregelung zuständig und stellt die Schnittstelle zum Abspielen von Multimedia Inhalten und der Netzwerkkommunikation dar. Ein weiterer Baustein ist die von Sun Microsystems entwickelte virtuelle Maschine Dalvik und die dazugehörige Java-Klassenbibliothek. In der Regel ist die Android-Bibliothek in Java geschrieben, jedoch greift diese bei geschwindigkeitskritischen Bereichen auf zahlreiche in C und C++ geschriebene native Bibliotheken zu.



Diese Abbildung zeigt noch einmal die 4 Schichten, auf welchen das Betriebssystem aufbaut. Jeder Teil und seine Arbeit sind strikt voneinander getrennt. Der Linux-Kernel ist das Zentralelement. Hier sind alle Treiber installiert, damit das System auch mit den verbauten Komponenten kommunizieren und arbeiten kann. Eine Stufe darüber sind die installierten Bibliotheken, welche Grundfunktionen für das Betriebssystem darstellen. Darin enthalten ist auch die Android Runtime. Eine Stufe darüber befindet sich nun das Framework für die Applications. In diesem Teilbereich sind alle Management-Funktionen implementiert, welche sich schlussendlich mit den Anwendungen im vierten Teilbereich befassen.

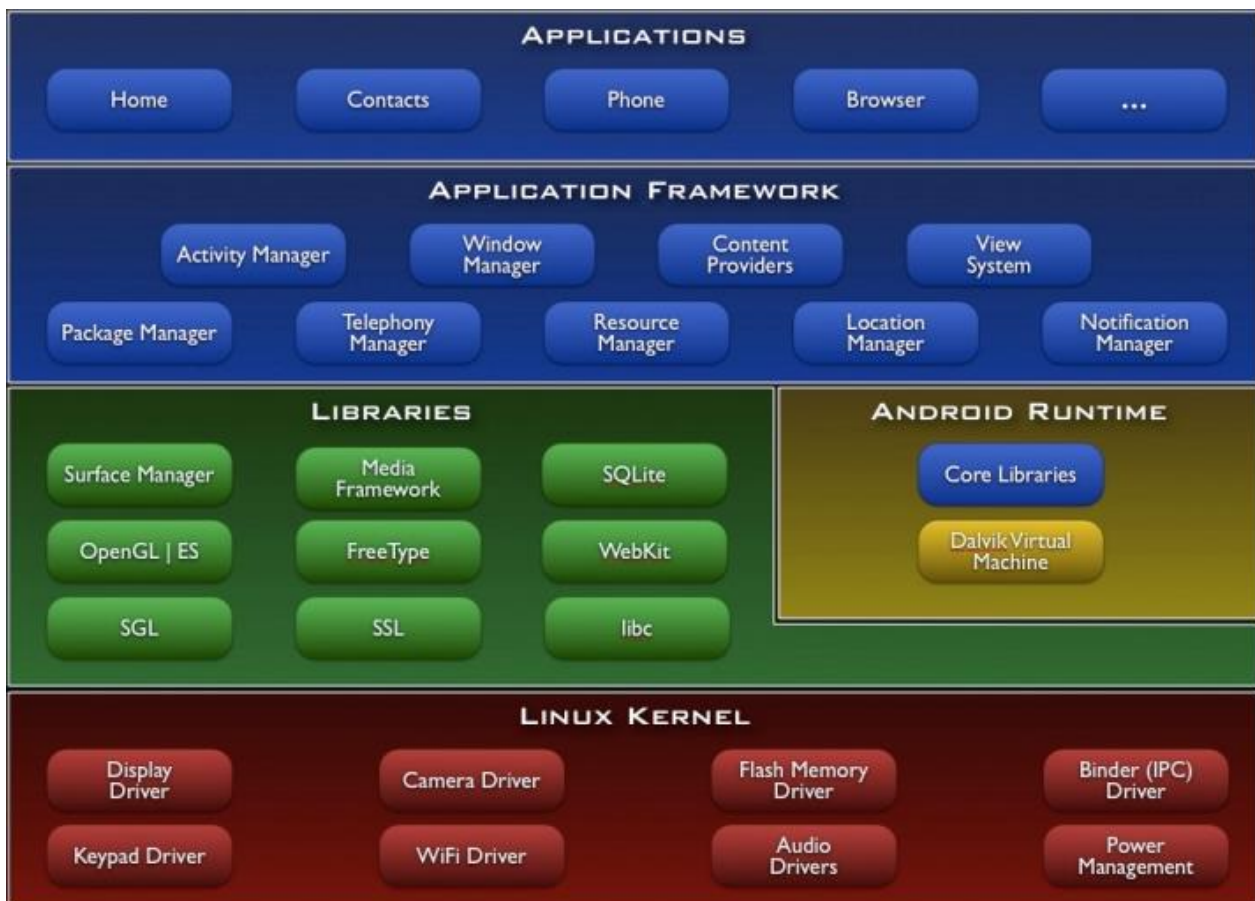


Abbildung 22: Android Architektur



8.14.2 Android Versionen

Die Android Versionen sind, wie es in der Linux Branche üblich ist, mit Themenabhängigen Namen versehen. Im Falle von Android sind es Süßwaren.

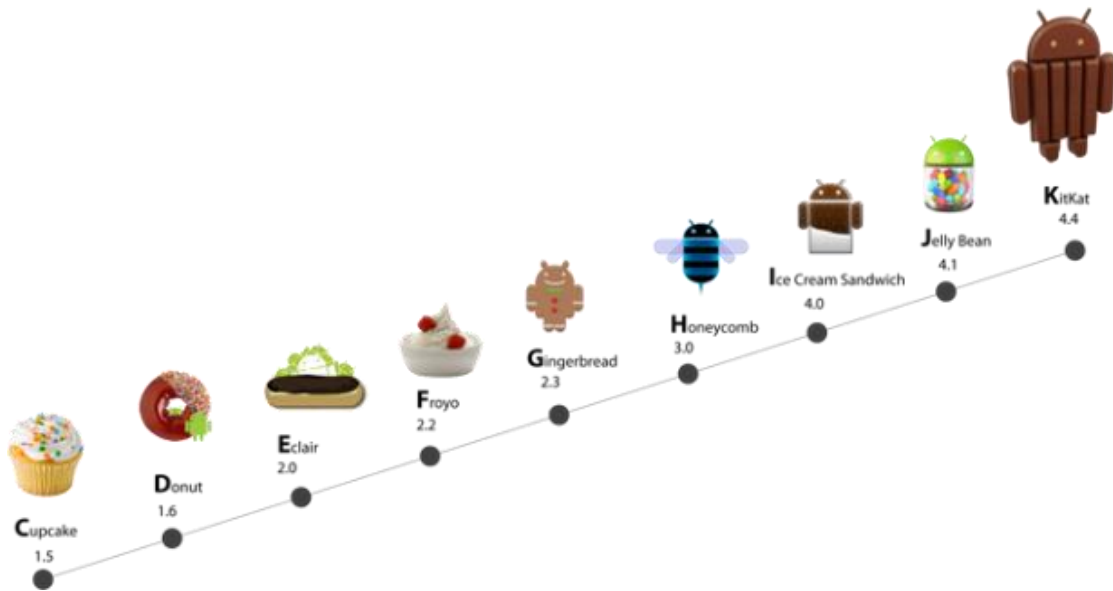


Abbildung 23: Android Versionen



9. Realisierung

9.1 Installation

Zuerst muss man das Java Development Kit (JDK) installieren.

Unter diesem Link findet man die aktuellste Version:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Als nächsten Schritt braucht man das Android Software Development Kit.

Das findet man unter diesem Link:

<http://developer.android.com/sdk/index.html>

Danach muss man android-sdk/tools für die User PATH Variable setzen.

Zudem benötigt man Eclipse unter diesen Link findet kann man Eclipse downloaden.

<http://www.eclipse.org/downloads/>

Nachdem installieren von Eclipse benötigt man das ADT Plugin für Eclipse. ADT (Android Development tools) ist ein Plugin von Eclipse, welches ein komplettes IDE für das entwickeln einer Adroid Application bereitstellt. Dadurch kann man neue Android Projekte erstellen und Android Projekte von existierenden Quellen erstellen. Auch kann man dadurch ein Android Projekt debuggen.

Um das ADT Plugin zu installieren, klickt man auf „install new software“ in Eclipse unter dem Register Help und fügt die folgende Seite ein: <http://dl-ssl.google.com/android/eclipse/> . Danach folgt man den Anweisungen.

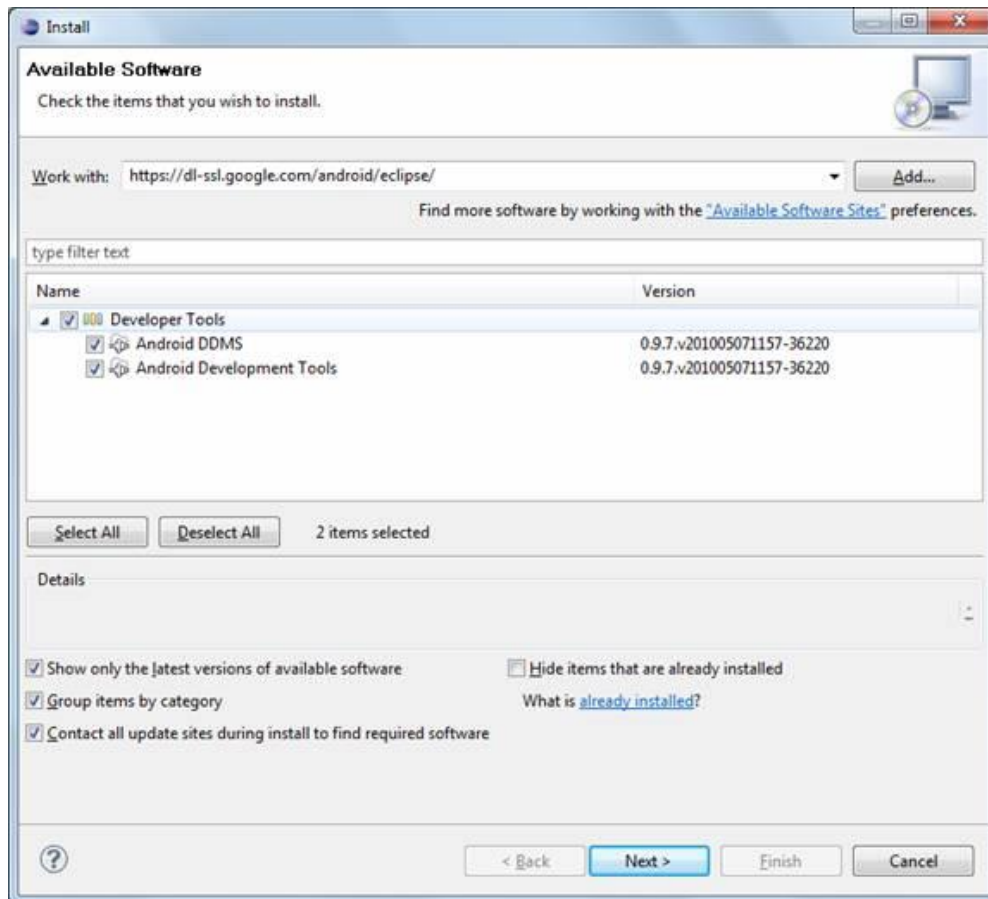


Abbildung 24: Installation ADT Plugin

Wenn man das ADT Plugin installiert hat, muss man die Android Platform und andere Komponenten installieren. Dies geschieht unter dem Register Option -> Android DK and AVD manager und wählt die neueste Android API.

Danach benötigt man Apache Ant. Apache Ant findet man unter diesem Link: <http://ant.apache.org/bindownload.cgi>. Als nächstes entpackt man das gedownloadete Zip file und setzt wieder den bin Ordern in der PATH Variable.

Natürlich braucht man auch das PhoneGap Framework. Dieses findet man hier: <http://phonegap.com/install/>



Weiters sollte man auch diese Pfade in der PATH Variable setzen:

- C:\adt-bundle-windows-x86_64-20140702\sdk\tools;
- C:\adt-bundle-windows-x86_64-20140702\sdk\platform-tools
- C:\Program Files\Java\jdk1-8.0_25\bin;
- C:\Program Files (x86)\Git\bin
- C:\Program Files (x86)\Git\bin;C:\apache-ant-1.9.4\bin;
- C:\Users\Benutzername\AppData\Roaming\npm
- C:\Program Files\nodejs\

Auch muss man diese Variablen setzen:

- JAVA_HOME – path of your JDK directory
- ANT_HOME – path of you apache-ant directory
- ANDROID_HOME – path to your android SDK directory.

Im nächsten Schritte gibt man diese Befehle in der Kommandozeile ein:

- npm install -g phonegap
- npm install -g cordova

So erstellt man ein neues Projekt:

- cordova create hello com.example.hello HelloWorld
- cd hello
- cordova platform add android
- cordova build



Danach muss man einen Emulator erstellen. Dazu geht man unter das Register Window/Android Virtual Device Manager und klickt auf den Button Create und fügt beispielweise diese Daten ein:

AVD Name: MyPhone

Device: Nexus S (4.0\", 480 x 800: hdpi)

Target: Android 4.4.2 - API Level 19

CPU/ABI: Intel Atom (x86)

Keyboard: Hardware keyboard present

Skin: HVGA

Front Camera: None

Back Camera: Webcam0

Memory Options: RAM: 343 VM Heap: 32

Internal Storage: 200 MiB

SD Card:

Size: MiB

File: Browse...

Emulation Options: Snapshot Use Host GPU

OK Cancel

Abbildung 25: Emulator erstellen



9.2 Darstellung

9.2.1 Produktübersicht

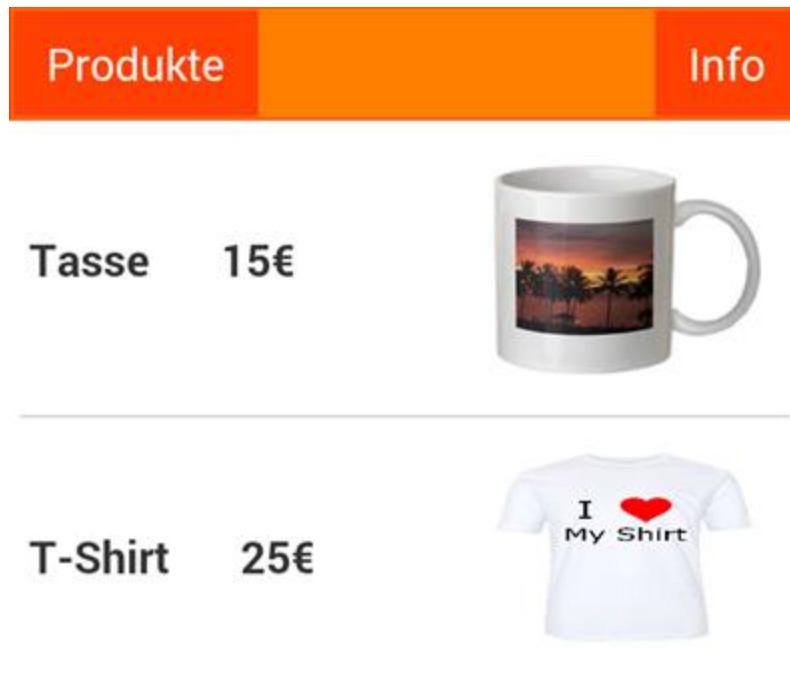


Abbildung 26: Produktübersicht

Wie der Name schon vermuten lässt, findet man hier die Auflistung aller Produkte.

Zu jedem Produkt stehen ein Name, der Preis des Produktes sowie ein Beispielbild zur Verfügung. Mithilfe des Buttons Info kann man auf die Infoseite der Applikation navigieren.



9.2.2 Informationsseite

Produkte

Handheld Print Product Editor

Inhalt der Application soll es sein,
Online, durch Fotografieren und
Hochladen von Bildern, Printprodukte zu
bestellen. Diese Produkte sind z.B.:
Tassen, T-Shirts, etc.

Abbildung 27: Informationsübersicht

Auf dieser Seite ist die Information über das Programm ersichtlich

Mit dem Button "Produkte" in der oberen Linken Ecke ist es möglich wieder auf die Seite Produkte zu wechseln.



9.2.3 Produktdetail



Tasse



Ob zu Weihnachten, Ostern oder zum Geburtstag ist eine Fototasse immer ein schönes Geschenk.

Preis: 15€

Abbildung 28: Produktdetail

Nach der Auswahl eines der zur Verfügung stehenden Produkte kommt man auf eine Übersichtsseite des jeweiligen Produktes.

Hier sieht man wieder den Namen, ein Beispielbild und den Preis.

des Weiteren wird hier ein Info Text zum gewählten Produkt angezeigt.

Von hier kommt man über den Button Gestalten in den Editor.



9.2.4 Editor



Abbildung 29: Editor allgemein

Im Editor können in dem schwarz umrandeten Feld Bilder aus dem Album, neu aufgenommene Bilder oder auch Text hinzugefügt und positioniert werden.

Über den Button "Vorschau" gelangt man zu einer Übersicht, man sieht das fertige Produkt.

Der Entwurf kann auch wieder gelöscht werden.



Abbildung 30: Editor - Produkt gestaltet

Hier sieht man, dass verschiedene Schriftarten zur Verfügung stehen. Ist das Textfeld markiert kann man es verschieben, drehen, oder seine Größe ändern. Dieselben Eigenschaften gelten auch für die eingefügten Bilder.



9.2.5 Vorschau



Abbildung 31: Vorschau

Dieses Bild gibt einen Einblick in die Vorschau. Man sieht wie das erstellte Layout auf dem Produkt aussehen wird. Über den Button "Bestellen" wird man zu eine Bestellformular weiter geleitet, wo die relevanten Daten für den Kauf eingegeben werden.



9.2.6 Bestelldaten

ZurückBezahlen

Daten für den Kauf

Anrede

Herr Frau

Nachname

Vorname

Adresse

Mailadresse



Abbildung 32: Bestelldaten

Hier sieht man das Bestellformular. Hat man alles ausgefüllt, so wird das Produkt mittels SVG und die Kaufdaten an den Server gesendet. Die Bezahlung ist nicht Teil der Diplomarbeit.



9.3 Risiken

Risiko	Risikohöhe	Eintrittswahrscheinlichkeit
neue Programmiersprachen	5	5
neue Technologie Phonegap	5	5
Qualitätsverlust	4	3
Funktionsumfang zu groß	4	3

Tabelle 5: Risiken

9.3.1 Neue Programmiersprachen

9.3.1.1 Risikobeschreibung

Die Programmierumsetzung der Applikation kann nicht stattfinden, da sich niemand mit den Sprachen etc. auskennt.

9.3.1.2 Lösungsansatz

Herr Professor Aberger wird kontaktiert um Fragen und Möglichkeiten zu klären. Außerdem wird ein Selbststudium erfolgen.

9.3.1.3 Gewünschte Auswirkungen

Das nötige Wissen wird zusammengetragen und die Programmierung erfolgt erfolgreich.

9.3.1.4 Unerwünschte Auswirkungen

Das Verständnis der Materie wird nicht erreicht und das Projekt kann nicht umgesetzt werden.

9.3.2 Neue Technologie Phonegap

9.3.2.1 Risikobeschreibung

Die Programmierumsetzung der Applikation kann nicht stattfinden, da sich niemand mit den Phonegap auskennt.

9.3.2.2 Lösungsansatz

Herr Professor Aberger wird kontaktiert um Fragen und Möglichkeiten zu klären. Außerdem wird ein Selbststudium erfolgen.

9.3.2.3 Gewünschte Auswirkungen

Das nötige Wissen wird zusammengetragen und die Verwendung von Phonegap erfolgt erfolgreich.



9.3.2.4 Unerwünschte Auswirkungen

Das Verständnis der Materie wird nicht erreicht und das Projekt kann nicht umgesetzt werden.

9.3.3 Qualitätsverlust

9.3.3.1 Risikobeschreibung

Es wird erkannt, dass das Endprodukt nicht in der vereinbarten Qualität erstellt werden kann.

9.3.3.2 Lösungsansatz

- Schulung der Projektmitarbeiter
- Reduzierung der Qualitätsanforderungen

9.3.3.3 Gewünschte Auswirkungen

- Qualität verbessert sich
- Das neue Qualitätsziel kann erreicht werden

9.3.3.4 Unerwünschte Auswirkungen

- Terminziel ist gefährdet
- Der Auftragnehmer hat aufgrund der mangelhaften Produktqualität einen schlechten Ruf beim Kunden

9.3.4 Funktionsumfang zu groß

9.3.4.1 Risikobeschreibung

Es wird erkannt, dass der geforderte Funktionsumfang nicht im geplanten Zeitrahmen realisierbar ist.

9.3.4.2 Lösungsansatz

- Reduzierung des Funktionsumfangs
- Überstunden

9.3.4.3 Gewünschte Auswirkungen

- Die Terminalsituation entspannt sich
- Das Terminziel kann erreicht werden

9.3.4.4 Unerwünschte Auswirkungen

- Der Kunde ist unzufrieden
- Die Qualität leidet



9.4 Verwendete Ressourcen

9.4.1 Software

- Windows 8
- Android
- iOS
- Dropbox
- Eclipse
- PhoneGap
- Node.js
- Microsoft Office

9.4.2 Hardware

- Smartphone iOS
- Smartphone Android
- 2 Laptops
- Server der Aberger Software GmbH

9.4.3 Orgware

- Mobiles Internet oder WLAN



9.5 Code

9.5.1 snapPicture() und getPicture()

snapPicture() und getPicture() sind Funktionen die von Cordova zur Verfügung gestellte Methoden verwenden um mit der eigenen Handykamera Fotos aufzunehmen oder Bilder aus der Galerie auszuwählen und der Applikation bereit zu stellen.

Mit **navigator.camera.getPicture** wird auf dem jeweiligen Gerät die native Bilderfassung gestartet.

quality: hiermit kann man angeben wie stark ein Bild komprimiert wird. Dort wird ein Wert zwischen 0 und 100 erwartet.

destinationType: Mit dieser Operation kann man festlegen, ob man den Pfad des Bildes als Rückgabewert will oder eine Repräsentation als base64-codierten String.

correctOrientation: ermöglicht die Drehung des Bildes während der Aufnahme. (Boolean)

saveToPhotoAlbum: ermöglicht das Speichern des aufgenommenen Bildes in der Galerie des Smartphones.

sourceType: hier gibt man an welche Quelle für das Bild zu nutzen ist. So kann man auf die Fotogalerie oder Alben Auswahl zugreifen. Dadurch kann man mit dem getPicture-Aufruf die Bildauswahl nutzen

```
function snapPicture () {
    navigator.camera.getPicture (onSuccess, onFail,
        { quality: 50,
          destinationType: Camera.DestinationType.FILE_URI,
          correctOrientation: true,
          saveToPhotoAlbum: true});
}
```

Abbildung 33: snapPicture



```
function getPhoto () {
  navigator.camera.getPicture(onSuccess, onFail,
  { quality: 50,
  destinationType: Camera.DestinationType.FILE_URI,
  sourceType: navigator.camera.PictureSourceType.SAVEDPHOTOALBUM,
  correctOrientation: true });
}
```

Abbildung 34: getPhoto

Es gibt 2 Fälle bei der Bildaufnahme oder Bildauswahl. Entweder es gelingt oder es gelingt nicht. Bei success wird die Methode onSuccess() aufgerufen. Dort wird das Bild richtig skaliert und danach den Canvas hinzugefügt. Bei einem Fehler wird die Methode onFail() aufgerufen. Dort wird die Fehlermeldung ausgegeben.

```
function onSuccess (imageURI) {
  new fabric.Image.fromURL(imageURI, function(img) {
    img.scale(0.1).set({
      left: 90,
      top: 150,
      cornerSize: 20
    });
    canvas.add(img);
  });
}

function onFail (message) {
  alert('An error occurred: ' + message);
}
```

Abbildung 35: onSuccess onFail



9.5.2 list()

Mit der Funktion list() wird auf das Array mit den Produkten zugegriffen und die Informationen auf der Startseite aufgelistet.

```
function list(){
  for (var i = 0; i < products.length; i++) {
    document.write('<span onclick="delivery('+products[i]["Id"]+')"><li>');
    document.write('<div id="content-right">'+products[i]["Bild"]+</div>');
    document.write('<div id="content-left"> <label>'+products[i]["Name"]+<label>');
    document.write('<label id="preis-content">'+products[i]["Preis"]+</label></div>');
    document.write('</li></span>');
  }
}
```

Abbildung 36: list



9.5.3 preview() und restore()

Die preview() und restore() Funktionen dienen zur Verwirklichung der Vorschau. In der preview() wird der ganze Editor ausgeblendet, das Produktbild als Hintergrund eingestellt und der mit der Methode toSVG() den Canvas in ein SVG umgewandelt. Diese SVG wird nun aus Hintergrundbild angezeigt.

Bei der restore() wird das SVG und das Hintergrundbild wieder ausgeblendet und der Editor wieder eingeblendet. Nun kann man das Gestalten des Artikels fortsetzen. Auch die Bestelldaten werden ausgeblendet, wenn man wieder

```
function preview(){
    json = JSON.stringify(canvas);
    $("#style").css('display','none');
    $("#preview").html('<a ><button onclick="data()" class="btn btn-warning navbar-brand">Bestellen</button></a>');
    $("#back").html('<a href="#top"><button onclick="restore()" class="btn btn-warning navbar-brand">Zurück</button></a>');
    id = parseInt(helpId);
    $("#previewsvg").css('background-image', 'url(' + products[id]["Bild2"] + ')');
    $("#previewsvg").addClass("svg"+id+"");
    $("#previewsvg").css('display','inherit');

    svg = canvas.toSVG();
    $("#svg").html(svg);
}
```

zurück will.

```
function restore(){
    $("#previewsvg").removeClass("svg"+id+"");
    $("#style").css('display','inherit');
    $("#previewsvg").css('display','none');
    $("#data").css('display','none');
    canvas.loadFromJSON(json, function () {
        canvas.renderAll();
    });
    $("#preview").html('<a id="preview" ><button onclick="preview()" class="btn btn-warning navbar-brand ">Vorschau</button></a>');
    $("#back").html('<a href="index.html" id="back"><button class="btn btn-warning navbar-brand"> Zurück</button></a>');
}
```

Abbildung 38: preview



9.5.4 data()

Hier wird das SVG ausgeblendet und das Formular für die Bestelldaten eingebledet.

```
function data(){
  $("#previewsvg").removeClass("svg+id+");
  $("#previewsvg").css('display','none');
  $("#data").css('display','inherit');
  $("#preview").html('<a id="preview" ><button onclick="chkFormular()" class="btn btn-warning navbar-brand ">Bezahlen</button></a>');
  $("#back").html('<a href="#top" id="back"><button onclick="restore()" class="btn btn-warning navbar-brand">Zurück</button></a>');
}
```

Abbildung 39: data



9.5.5 clear()

Mit der clear() Funktion kann man einzelne oder mehrere ausgewählte Objekte löschen.

```
function clear(){  
  
    if(canvas.getActiveGroup() != null){  
        canvas.getActiveGroup().forEachObject(function(o){ canvas.remove(o) }); canvas.discardActiveGroup().renderAll();  
  
    }else if(canvas.getActiveObject() != null){  
        var object = canvas.getActiveObject();  
        canvas.remove(object);  
        canvas.renderAll();  
  
    }else{  
        canvas.clear();  
        canvas.renderAll();  
    }  
}
```

Abbildung 40: clear



9.5.6 Text hinzufügen

Wird der Button für das Hinzufügen eines Textes angeklickt. So wird dies ausgeführt:

```
$("#btnAddCustomText").click(function () {  
    var activeObject = canvas.getActiveObject();  
  
    if(activeObject!=null && activeObject.type === 'custom-text'){  
        activeObject.set("text", getText());  
        activeObject.set("fill", getTextColor());  
        activeObject.set("textBackgroundColor", getBackgroundColor());  
        activeObject.set("fontFamily", getFont());  
  
        canvas.renderAll();  
    }else{  
        var text = new fabric.CustomText(getText(), {  
            fontFamily: getFont(),  
            cornerSize: 20,  
            fill: getTextColor(),  
            textBackgroundColor: getBackgroundColor();  
        });  
        canvas.add(text);  
        canvas.renderAll();  
    }  
});
```

Abbildung 41: Text hinzufügen

Ist ein Text-Objekt ausgewählt so wird dieses verändert. Ist kein Text-Objekt ausgewählt so wird ein neues Text-Objekt mit den ausgewählten Eigenschaften erstellt und den Canvas hinzugefügt. Dies geschieht mit den folgenden Funktionen:



9.5.7 getText()

Hier holt man sich aus dem Inputfeld mit der ID „enterText“ den gewünschten Text und übergibt diesen.

```
function getText() {  
    var text = "";  
    text=document.querySelector('#enterText').value;  
    return text;  
}
```

Abbildung 42: getText

9.5.8 getTextColor()

Hier wird die ausgewählte Farbe aus dem Input mit der ID „textColor“ geholt und zurückgegeben.

```
function getTextColor() {  
    var stextColor = document.getElementById('textColor');  
    return stextColor.value;  
}
```

Abbildung 43: getTextColor

9.5.9 getBackgroundColor()

Hier wird die ausgewählte Farbe aus dem Input mit der ID „textBgColor“ geholt und zurückgegeben.

```
function getBackgroundColor() {  
    var stextBgColor = document.getElementById('textBgColor');  
    return stextBgColor.value;  
}
```

Abbildung 44: getBackgroundColor



9.5.10 getFont()

Hier wird die Schriftart vom select mit der „Id“ „selectedFontFamily“ geholt und zurückgegeben.

```
function getFont () {
    var myFont = document.getElementById("selectFontFamily");
    return myFont.options[myFont.selectedIndex].value;
}
```

Abbildung 45: getFont

9.5.11 chkFormular()

Bei dieser Funktion chkFormular werden die eingegebenen Daten geprüft. Als erstes wird überprüft, ob auch wirklich alle Felder ausgefüllt sind und im nächsten Schritt wird noch geprüft, ob eine E-Mail Adresse eingegeben wurde.

```
function chkFormular () {
    var mail = $("#mail").val();
    if ($("#nachname").val() == "") {
        alert("Bitte Ihren Nachnamen eingeben!");
    } else if ($("#vorname").val() == "") {
        alert("Bitte Ihren Vornamen eingeben!");
    } else if ($("#ort").val() == "") {
        alert("Bitte Ihren Wohnort eingeben!");
    } else if ($("#mail").val() == "") {
        alert("Bitte Ihre E-Mail-Adresse eingeben!");
    } else if (mail.indexOf("@") == -1) {
        alert("Keine E-Mail-Adresse!");
    } else {
        var anrede;
        for (var i=0; i < document.form.anrede.length; i++)
        {
            if (document.form.anrede[i].checked)
            {
                anrede = document.form.anrede[i].value;
            }
        }
    }
}
```

Abbildung 46: chkFormular1



Danach werden die Daten in Variablen gespeichert und zu einen String zusammengefügt. Dieser Datenstring wird mit Hilfe von AJAX an die Datei postSVG.php, die am Server liegt, gesendet.

```
var nachname = $("#nachname").val();
var vorname = $("#vorname").val();
var ort = $("#ort").val();
var mail = $("#mail").val();
var dataString = 'nachname='+ nachname + '&vorname='+ vorname + '&mail='+ mail
                + '&ort='+ ort+ '&anrede='+ anrede +'&svg=' +svg;
var formData = {anrede: anrede, nachname: nachname, vorname: vorname, ort: ort, mail: mail, svg: svg};
makeCorsRequest(dataString);
$.ajax({
  url : 'http://localhost:81/postSVG.php',
  type: "POST",
  data : dataString,
  success: function(data, textStatus, jqXHR)
  {
    console.log('Die Daten konnten an den Server gesendet werden!');
  },
  error: function (jqXHR, textStatus, errorThrown)
  {
    console.log('Die Daten konnten leider nicht an den Server gesendet werden!');
  }
});
}
```

Abbildung 47:chkFormular2



9.5.12 postSVG.php

```
<?php

$anrede = "";
$nachname = "";
$vorname = "";
$mail = "";
$ort = "";
$svg = "";

if(!empty($_POST['anrede']) && !empty($_POST['nachname']) && !empty($_POST['vorname'])
    && !empty($_POST['mail']) && !empty($_POST['ort']) && !empty($_POST['svg'])){
    $anrede=$_POST['anrede'];
    $nachname=$_POST['nachname'];
    $vorname=$_POST['vorname'];
    $mail=$_POST['mail'];
    $ort=$_POST['ort'];
    $svg=$_POST['svg'];
}

?>
```

In dieser Datei werden die empfangenden Daten am Sever gespeichert.

Abbildung 48: postSVG



9.5.13 Cross Domain Requests

Hier wird ein Request erzeugt, welcher es ermöglicht Daten von einer Domain in eine andere zu senden, da dies Standardmäßig nicht erlaubt ist, jedoch für unsere Diplomarbeit erforderlich ist.

```
function createCORSRequest(method, url) {
  var xhr = new XMLHttpRequest();
  if ("withCredentials" in xhr) {
    xhr.open(method, url, true);
  } else if (typeof XDomainRequest != "undefined") {
    xhr = new XDomainRequest();
    xhr.open(method, url);
  } else {
    xhr = null;
  }
  return xhr;
}
```

Abbildung 49: Cross Domain Request1

```
function makeCorsRequest(inhalt) {
  var url = 'http://localhost:81/postSVG.php';

  var xhr = createCORSRequest('POST', url);
  if (!xhr) {
    alert('CORS not supported');
    return;
  }
  xhr.onload = function() {
    var text = xhr.responseText;
  };
  xhr.onerror = function() {
    alert('Woops, there was an error making the request.');
```

Abbildung 50: Cross Domain Request2

Hier wird der Request auf den Server „localhost:81“ abgesetzt. Die verwendete Methode ist „POST“. Der Wert der Variable „inhalt“ wird gesendet



10. Probleme und Lösungen

Im Laufe der Entwicklung stießen wir sowohl auf technische, als auch auf allgemeine Probleme mit denen wir uns fortan auseinandersetzen mussten. Bei Problemfällen ist es wichtig, sich sofort mit diesen auseinander zu setzen, da grundsätzlich nicht mit einem fehlerhaften System gearbeitet werden soll. Daher hatte der Umgang mit Problemen in unserer Diplomarbeit eine hohe Priorität.

10.1 Allgemeine Probleme

Die Allgemeinen Probleme waren oftmals insofern schwierig zu beheben, da es sich nicht um rein technische Probleme handelte, für welche man eventuell schon ein gewisses Muster zur Lösung dieser Entwickelt hat. Diese Situation machte es oftmals schwierig auf einen grünen Zweig zu kommen.



10.1.1 Allgemeiner Problemfall 1 - Aufwandschätzung

Aufgrund der fehlenden Kenntnisse bei den verwendeten Technologien schien es anfangs, als seien die Aufgaben der Diplomarbeit im Handumdrehen zu lösen. Jedoch stellte sich schnell heraus, dass die Anforderungen weit höher waren als gedacht. Nun rächte sich die anfängliche Entspannung mit der wir die Arbeit angegangen sind.

Lösung:

Durch eine schnelle Änderung des Zeitplanes und der Hilfe von Herrn Professor Aberger war es uns möglich die Diplomarbeit im gewünschten Zeitrahmen abzuwickeln.



10.2 Technische Probleme

10.2.1 Technischer Problemfall 1 - Cross Domain Request sind standardmäßig nicht erlaubt

Als die Applikation bis auf den Teil mit dem Senden fertig war, dachten wir uns, dass wir eigentlich nichts mehr zu tun haben, jedoch tauchte schnell das Problem auf, dass Cross Domain Requests nicht erlaubt sind.

Anfangs konnten wir mit dieser Fehlermeldung nicht viel anfangen, da wir dachten, dass beim normalen Senden keinerlei Probleme mehr auftauchen.

Lösung:

Zum Glück konnte auf der Seite Stack Overflow schnell eine Lösung gefunden werden: CORS (Cross Origin Resource Sharing)!

Wir benutzen nun CORS um via XMLHttpRequests unseren Entwurf an den Server zu senden.



10.2.2 Technischer Problemfall 2 - Android Versionen

Aufgrund der vielen Android Versionen stießen wir auf Probleme mit der Kompatibilität unseres Codes. Leider sind die Kompatibilitätsprobleme von Phonegap mit Android Versionen nur sehr schlecht bis gar nicht dokumentiert, was es uns sehr erschwerte Lösungen für solche Probleme zu finden.

Lösung:

Wir lösten den Großteil unserer Probleme mit der Kompatibilität von Android Versionen mit den gewählten Funktionen indem wir den Baustein durch eigenen Code ersetzen, was zwar für erhöhten Aufwand und nicht ganz so hohe Effizienz sorgt, jedoch eine effektive Lösung ist.



11. Qualitätssicherung

Um eine einwandfreie Benutzung des Produktes gewährleisten zu können, ist es wichtig, Funktionalitäten zu jedem Änderungszeitpunkt genau zu testen. Die Tests sollen ausführlich und genau durchgeführt werden, dies soll einer langwierigen Fehlersuche, im späteren Verlauf der Entwicklung, vorbeugen.

11.1 Probleme

Durch das laufende Testen der Funktionen, hatten wir im Verlauf der Programmierung kaum größere Probleme. Nur das Senden der Daten bereitete uns aufgrund unvorhergesehener Restriktionen kleinere Schwierigkeiten

11.2 Qualitätsmerkmale

- Da unser Produkt auf der aktuellsten Androidversion basiert, und natürlich auch abwärtskompatibel ist, ist natürlich eine gewisse Langlebigkeit gegeben.
- Durch unser einfaches und übersichtliches Design ist unser Programm einfach und intuitiv zu bedienen.
- Durch die Tests ist die korrekte Funktionsweise gewährleistet



12. Testen

12.1 Allgemein

Das Testen des Produktes ist ein sehr wichtiger Bestandteil in der Projektentwicklung und ist daher nicht aus dieser wegzudenken. Aus diesem Grund haben wir die App, also das Produkt unserer Diplomarbeit, bereits im Laufe der Entwicklungsphase ständig überprüft.

Durch diverse Tests in den Funktionsbereichen werden bereits vor der Veröffentlichung des Produktes möglichst alle Fehler ausgemerzt und somit eine hohe Qualität des Endproduktes gewährleistet.



12.2 Übersicht Testfälle

Nummer	Testfall	Erwartetes Ergebnis	Erfolgreich?
1	Aufruf der Übersicht	Es soll eine Liste mit den zur Verfügung stehenden Produkten angezeigt werden	Ja
2	Infos anzeigen	Es sollen sowohl die Info über unsere App, sowie auch die der Produkte angezeigt werden	Ja
3	Editor	Bei ausgewähltem Produkt soll der Editor geöffnet werde, in diesem soll es möglich sein Text und Bild zu positionieren	Ja
4	Bild und Text hinzufügen	Bild und Text sollen auf den Editor hinzugefügt werden	Ja
5	SVG erstellen	Ein korrektes SVG soll erstellt werden und in der Vorschau angezeigt werden	Ja
6	Benutzerdaten eingeben	Aufruf der Benutzerdatenmaske und Prüfung ob alle Daten eingegeben wurden	Ja
7	Bestellung absenden	Wird die Bestellung in korrekter Form an den Testserver gesendet	Ja

Tabelle 6: Testfälle



12.3 Testfälle

12.3.1 Testfall 1: Aufruf der Übersicht

- I. Starten der Applikation „Handheld Print Product Editor“
- II. Erwartetes Ergebnis: Laden der Liste mit den Angebotenen Produkten inklusive der Preise.
- III. Ergebnis: Erfolgreiches Laden der Liste

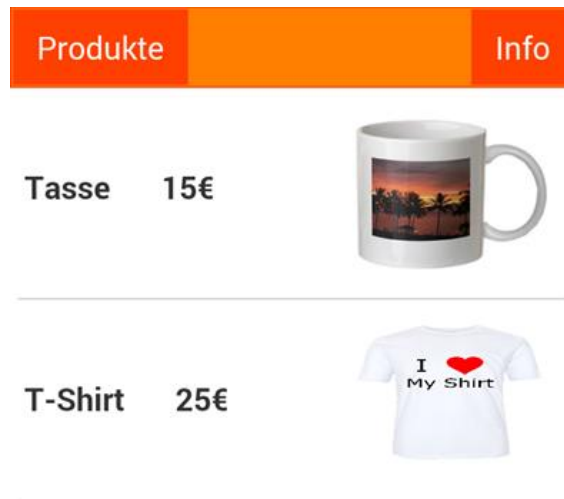


Abbildung 51: Produkte



12.3.2 Testfall 2: Infos anzeigen

- I. Starten der Applikation „Handheld Print Product Editor“
- II. Auswahl des Buttons Info
- III. Erwartetes Ergebnis: Anzeige der Info Seite
- IV. Ergebnis: Infoseite wird Angezeigt



Abbildung 52: Information

- V. Auswahl des Buttons „Produkte“
- VI. Auswahl eines Produkts
- VII. Erwartetes Ergebnis: Infoseite des Produktes soll angezeigt werden
- VIII. Ergebnis: Infoseite des Produktes wird Angezeigt



Abbildung 53: Produktdetail



12.3.3 Testfall 3: Editor aufrufen

- I. Starten der Applikation „Handheld Print Product Editor“
- II. Auswahl eines Produktes
- III. Auswahl des Buttons „Gestalten“
- IV. Erwartetes Ergebnis: Editor soll aufgerufen werden
- V. Ergebnis: Editor wird aufgerufen



Abbildung 54: Editor



12.3.4 Testfall 4: Bild und Text hinzufügen

- I. Starten der Applikation „Handheld Print Product Editor“
- II. Auswahl eines Produktes
- III. Auswahl des Buttons „Gestalten“
- IV. Auswahl des Buttons „Bild“, „Kamera“ oder „Text“
- V. Erwartetes Ergebnis: Eingabemaske öffnet sich und der gewünschte Text oder das gewünschte Bild wird hinzugefügt
- VI. Ergebnis: Bild oder Text wird hinzugefügt



Abbildung 55: Editor Bilder



12.3.5 Testfall 5: SVG erstellen

- I. Starten der Applikation „Handheld Print Product Editor“
- II. Auswahl eines Produktes
- III. Auswahl des Buttons „Gestalten“
- IV. Produkt gestalten
- V. Auswahl des Buttons Vorschau
- VI. Erwartetes Ergebnis: Das ausgewählte Produkt wird mit dem selbst erstellten Design angezeigt
- VII. Ergebnis: Das gewählte Produkt wird mit dem gewünschten Design angezeigt



Abbildung 57: Vorschau



12.3.6 Testfall 6: Benutzerdaten eingeben

- I. Starten der Applikation „Handheld Print Product Editor“
- II. Auswahl eines Produktes
- III. Auswahl des Buttons „Gestalten“
- IV. Produkt gestalten
- V. Auswahl des Buttons Vorschau
- VI. Auswahl des Buttons Bestellen
- VII. Eingabe der Daten in die dafür vorgesehenen Felder
- VIII. Erwartetes Ergebnis: Benutzerdaten werden auf ihre Vollständigkeit überprüft
- IX. Ergebnis: Bei vergessenen Feldern kommt eine Aufforderung die restlichen Felder zu ergänzen

The screenshot shows a mobile application interface with a dark blue header containing 'Zurück' and 'Bezahlen' buttons. The main title is 'Daten für den Kauf'. Below the title, there are radio buttons for 'Anrede' (Herr/Frau) and a text field for 'Nachname'. An alert dialog box is displayed over the form, with the text 'Alert' and 'Bitte Ihren Nachnamen eingeben!' and an 'OK' button. The email field below contains 'max.mustermann@gmail.com'.

Abbildung 59: Fehlermeldung Nachname

The screenshot shows the same mobile application interface. The 'Nachname' field is filled with 'Mustermann'. The alert dialog box now displays the text 'Alert' and 'Keine E-Mail-Adresse!' with an 'OK' button. The email field below contains 'max.mustermann'.

Abbildung 58: Fehlermeldung leeres Feld



12.3.7 Testfall 7: Senden der Daten an den Server

- I. Starten eines Emulators
- II. Starten der Applikation „Handheld Print Product Editor“
- III. Auswahl eines Produktes
- IV. Auswahl des Buttons „Gestalten“
- V. Produkt gestalten
- VI. Auswahl des Buttons Vorschau
- VII. Auswahl des Buttons Bestellen
- VIII. Eingabe der Daten in die dafür vorgesehenen Felder
- IX. Auswahl des Buttons „Bezahlen“
- X. Erwartetes Ergebnis: Daten werden an den Server gesendet
- XI. Ergebnis: Der Server erhält die gesendeten Daten



13. Glossar

13.1 SVG

SVG ist die Abkürzung für „Scalable Vector Graphics“. Es ist eine Beschreibung von zweidimensionaler Vektorgrafiken von World Wide Web Consortium(W3C). SVG basiert auf XML und wurde erstmals 2001 veröffentlicht.

Mittels SMIL werden Animationen unterstützt. Die Manipulation des SVG-DOM ist mit Hilfe von den Funktionen von Scriptsprachen möglich.

13.2 XML

XML steht für „Extensible Markup Language“ und dient zur Darstellung von hierarchischen strukturierter Daten in Form von Textdateien. Daher wird XML als Auszeichnungssprache bezeichnet.

Einsetzungsgebiet ist der plattform- und implementationsunabhängige Austausch von Daten zwischen Computersystemen, meist über das Internet.

Auch XML wurde vom World Wide Web Consortium (W3C) herausgegeben. Eine XML-Datei enthält Textzeichen und ist in den meisten Fällen mit einer ASCII-Kodierung kodiert und damit auch lesbar.

13.3 SMIL

SMIL bedeutet „Synchronized Multimedia Integration Language“ und basiert auf XML Standard einer Ausgangssprache für zeitsynchronisierte multimediale Inhalte.



14. Resümee – Samantha Plank

Durch die Diplomarbeit konnte ich mein Wissen im Bereich der Webentwicklung und im Bereich der Applikationsentwicklung erweitern. Bei verschiedensten Problemen kamen Christoph und ich durch unserer Zusammenarbeit schnell auf die Lösung der Problemfälle.

Zudem erweiterte ich meine Fähigkeiten in der Projektentwicklung. Das wichtigste in einen Projekt ist der Zeitplan und die Kommunikation im Team. Am Beginn der Diplomarbeit kostet es mir viel Zeit sich in die Technologien wie Phonegap und JavaScript einzulesen und so kam es schnell zu einen Terminverzug.

Je mehr ich begriff wie einfach und schnell man eine App mit Phonegap entwickeln kann, desto mehr begeisterte ich mich für die Programmierung der Applikation. Auch die vielen Möglichkeiten der Webprogrammierung weckten mein Interesse.

Ich würde mich jederzeit wieder für eine Diplomarbeit entscheiden, da man durch eine Diplomarbeit das selbständige Arbeiten und das Verantwortungsbewusstsein stärkt und dies im späteren Berufsleben benötigt.



15. Resümee – Christoph Mayrbäurl

Die Diplomarbeit erlaubte es mir, mein Wissen in den Bereichen der Projektentwicklung, der Webentwicklung sowie auch der mobilen Applikationsentwicklung zu erweitern. Dank der Zusammenarbeit mit meiner Partnerin bei dieser Diplomarbeit war es uns möglich die aufgetretenen Probleme schnell zu lösen.

Zu meinem Leidwesen wurde mir auch wieder klar, wie viel Aufwand es bedeuten kann, sich neue Technologien anzuwenden, da wie dies unterschätzt hatten gerieten wir dadurch Anfangs schnell in Verzug.

Umso mehr ich mich jedoch mit dem Thema beschäftigte umso mehr Spaß machte die Arbeit mit Phonegap, da man damit schnell und einfach nützliche Applikationen erstellen kann.

Bekäme ich noch einmal die Chance für eine Diplomarbeit würde ich diese jederzeit wieder wahrnehmen, da man dadurch einen Sinn für Verantwortung bei selbstständiger Arbeit erhält und auch schnell merkt wie man sich selbst motivieren muss.



16. Verzeichnisse

16.1 Abbildungsverzeichnis

Abbildung 1: Umfeld Hierarchie	20
Abbildung 2: Überblick Gesamtsystem - USE CASE	22
Abbildung 3: Zeitplan	25
Abbildung 4: Auflistung Meilensteine	26
Abbildung 5: Projektstrukturplan	27
Abbildung 6: Konfigurationsplan.....	28
Abbildung 7: HTML Logo.....	30
Abbildung 8: CSS Logo	31
Abbildung 9: JavaScript Logo.....	32
Abbildung 10: jQuery Logo.....	32
Abbildung 11: AJAX Logo	33
Abbildung 12: Fabric.js Logo	34
Abbildung 13: Node.js Logo	35
Abbildung 14: Php Logo.....	36
Abbildung 15: Apache HTTP Server Logo.....	37
Abbildung 16: Dropbox Logo	38
Abbildung 17: Eclipse Logo.....	38
Abbildung 18: Phonegap Logo.....	39
Abbildung 19: Phonegap Kompatibilität.....	40
Abbildung 20: Apache Cordova Logo	41
Abbildung 21: Android Logo	42
Abbildung 22: Android Achitektur	43
Abbildung 23: Android Versionen.....	44
Abbildung 24: Installation ADT Plugin	46
Abbildung 25: Emulator erstellen	48
Abbildung 26: Produktübersicht.....	49
Abbildung 27: Informationsübersicht.....	50
Abbildung 28: Produktdetail.....	51
Abbildung 29: Editor allgemein.....	52
Abbildung 30: Editor - Produkt gestaltet.....	53
Abbildung 31: Vorschau	54
Abbildung 32: Bestelldaten.....	55



Abbildung 33: snapPicture	59
Abbildung 34: getPhoto	60
Abbildung 35: onSuccess onFail	60
Abbildung 36: list.....	61
Abbildung 38: restore	62
Abbildung 37: preview	62
Abbildung 39: data	63
Abbildung 40: clear.....	64
Abbildung 41: Text hinzufügen	65
Abbildung 42: getText.....	66
Abbildung 43: getTextColor.....	66
Abbildung 44: getBackgroundColor.....	66
Abbildung 45: getFont	67
Abbildung 46: chkFormular1	67
Abbildung 47:chkFormular2	68
Abbildung 48: postSVG.....	69
Abbildung 49: Cross Domain Request1	70
Abbildung 50: Cross Domain Request2	70
Abbildung 51: Produkte.....	78
Abbildung 52: Information	79
Abbildung 53: Produktdetail.....	79
Abbildung 54: Editor	80
Abbildung 55: Editor Bilder	81
Abbildung 56: Text hinzufügen	81
Abbildung 57: Vorschau	82
Abbildung 58: Fehlermeldung leeres Feld	83
Abbildung 59: Fehlermeldung Nachname.....	83

16.2 Tabellenverzeichnis

Tabelle 1: Kontaktdaten Aberger Software GmbH.....	16
Tabelle 2: Daten Samantha Plank	17
Tabelle 3: Daten Christoph Mayrbäurl.....	18
Tabelle 4: Kontaktdaten DI Christian Aberger.....	19
Tabelle 5: Risiken	56
Tabelle 6: Testfälle.....	77



17. Quellen

<http://media2.giga.de/2013/01/InnovationM-Android-Release-History.png>

<http://elinux.org/images/c/c2/Android-system-architecture.jpg>

http://winfwiki.wi-fom.de/images/5/55/Phoneygap_support_matrix.png

<http://de.statista.com/statistik/daten/studie/256790/umfrage/marktanteil-von-android-und-ios-am-smartphone-absatz-in-deutschland/>

<https://cordova.apache.org/>

<http://www.html5rocks.com/en/tutorials/cors/>

<http://stackoverflow.com/>

<http://de.wikipedia.org/wiki/Wikipedia:Hauptseite>

https://www.dpunkt.de/leseproben/3787/7_Die%20Kamera%20%28Kapitel%20auszug%29.pdf

<http://code.tutsplus.com/tutorials/creating-an-android-hello-world-application-with-phoneygap--mobile-2532>