

Abschlussprüfung Winter 2015/2016

Fachinformatiker für Anwendungsentwicklung  
Dokumentation zur betrieblichen Projektarbeit

# **Entwicklung einer Webapplikation**

## **Geburtstagsmailing**

Abgabedatum: Hamburg, den 09.12.2015

**Prüfungsbewerber:**

J V  
Straße 1  
2 Hamburg



t8y.com gmbh

**Praktikumsbetrieb:**

t8y.com GmbH  
Bernhard-Nocht-Straße 113  
20359 Hamburg

## Inhaltsverzeichnis

<b>1 Einleitung.....</b>	<b>1</b>
1.1 Projektbeschreibung.....	1
1.2 Projektziel .....	1
1.3 Projektumfeld .....	2
1.4 Projektbegründung.....	2
1.5 Projektschnittstellen .....	2
1.6 Projektabgrenzung .....	2
<b>2 Projektplanung .....</b>	<b>2</b>
2.1 Projektphasen .....	2
2.2 Ressourcenplanung .....	3
2.3 Entwicklungsprozess.....	3
<b>3 Analysephase .....</b>	<b>4</b>
3.1 Ist-Analyse .....	4
3.2 Wirtschaftlichkeitsanalyse .....	4
3.2.1 Projektkosten .....	4
3.2.2 Amortisationsdauer .....	5
3.3 Nicht-monetäre Vorteile.....	5
3.4 Anwendungsfälle.....	5
3.5 Lastenheft .....	6
<b>4 Entwurfsphase.....</b>	<b>6</b>
4.1 Zielplattform .....	6
4.2 Architekturdesign .....	7
4.3 Benutzeroberfläche .....	7
4.4 Animationslogik .....	9
4.5 Maßnahmen zur Qualitätssicherung .....	9
4.6 Pflichtenheft .....	10
<b>5 Implementierungsphase .....</b>	<b>10</b>

---

5.1	Iterationsplanung.....	10
5.2	Implementierung der Benutzeroberfläche/Animation .....	10
5.2.1	Handlebars-Code erzeugt HTML-Template.....	11
5.2.2	JSON-Datei enthält Inhalt der Webseite .....	11
5.2.3	Less-Code für Formatierung und Gestaltung.....	12
5.2.4	JavaScript überwacht Click-Events .....	13
5.2.5	JavaScript steuert das Verhalten.....	14
<b>6</b>	<b>Abnahme.....</b>	<b>15</b>
6.1	Abnahme durch die Abteilung der Qualitätssicherung .....	16
6.2	Abnahme durch die Geschäftsführung .....	16
6.3	Deployment.....	16
<b>7</b>	<b>Dokumentation .....</b>	<b>16</b>
7.1	Benutzerhandbuch .....	16
7.2	Entwicklerdokumentation.....	16
<b>8</b>	<b>Fazit.....</b>	<b>16</b>
8.1	Soll-Ist-Vergleich .....	17
8.2	Ausblick.....	17
<b>9</b>	<b>Anhang.....</b>	<b>18</b>
<b>10</b>	<b>Literaturverzeichnis/Quellenangabe .....</b>	<b>36</b>
<b>11</b>	<b>Glossar.....</b>	<b>37</b>

## 1 Einleitung

In der folgenden Dokumentation wird der Ablauf des Handelskammer-Abschlussprojektes (HK) dargestellt, das der Autor im Rahmen seiner Umschulung zum Fachinformatiker Fachrichtung Anwendungsentwicklung durchgeführt hat. Praktikumsbetrieb ist die t8y.com GmbH (t8y), ein Tochterunternehmen der Omnicom Group, die als technischer Dienstleister individuelle Lösungen im Bereich der digitalen Werbung entwickelt. Dabei werden individuelle Lösungen für Web-, Mobile- und Email-Marketing-Projekte umgesetzt. t8y entwickelt Newsletter, Microsites, Corporate Sites und, auf Kundenwunsch hin, spezialisierte Content Management Systeme. Zurzeit beschäftigt die t8y ca. 35 Mitarbeiter und Zeitweise bis zu 5 Freelancer.

### 1.1 Projektbeschreibung

Bei dem Projekt handelt es sich um eine Full Responsive Webapplikation. Responsivität bedeutet, dass die Applikation auf allen mobilen Endgeräten, sowie den gebräuchlichsten Browsern sowohl dargestellt werden kann, als auch ein der entsprechenden Viewport Größe angepasstes Aussehen erhält. Der Autor realisiert das Frontend. Dies erfordert den Einsatz aktueller Frontendtechniken, wie z.B. HTML5, CSS3 und JavaScript.

Bei dem Geburtstagsmailing handelt es sich um eine Animation, die soweit möglich mit CSS3 umgesetzt werden soll.

Der Nutzer erhält zu seinem Geburtstag von einem Projektmanager von t8y eine E-Mail, auf welcher sich ein link zur Webapplikation befindet. Beim Klick auf den link, öffnet sich die Webseite. Daraufhin kann der Kunde mit der vom Autor erstellten Webapplikation interagieren. Dies bedeutet, dass sich auf der Seite klickbare Elemente befinden bzw. dass sie mittels touch-Geste auslösbar sind. Wenn diese betätigt werden, sie eine Kette von Folgeanimationen auslösen.

Die Animation soll mit einem „Happy Birthday Dear“ und dem entsprechenden Namen des Kunden enden, der über eine Datenbank ausgelesen und übermittelt wird. Die Programmierung zum Auslesen des Namens aus der Datenbank wird vom Backend realisiert.

### 1.2 Projektziel

Ziel des Projektes ist, die langfristige und stabile Zusammenarbeit mit den Kunden zu festigen und intensivieren zu können. Aus diesem Grund soll eine speziell für t8y konzipierte Full Responsive Geburtstagsmailing in Form eines Newsletters und einer entsprechenden Landingpage (Microsite) entwickelt werden. Es soll sowohl personalisierbar sein, als auch die einzelnen Gewerke wie Newsletter, Frontend-Entwicklung, Projekt Management und ihre Stärken repräsentieren.

### 1.3 Projektumfeld

Bei der Webapplikation handelt es sich um eine in sich abgeschlossene Aufgabe, die in der Agentur t8y durchgeführt wird. Hieran sind sowohl das Backend als auch das Frontend beteiligt. Diese Dokumentation befasst sich ausschließlich mit der Umsetzung des Frontend, da der Autor nur für diesen Bereich verantwortlich ist.

Auftraggeber des Projektes ist die Geschäftsführung (GF) von t8y.

Um den Anforderungen der GF gerecht zu werden, ist eine regelmäßige Kommunikation und Rücksprache erforderlich. Diese findet regelmäßig mit dem Head of Webdevelopment statt, welcher dabei als Stellvertreter für die GF agiert.

### 1.4 Projektbegründung

Da aufgrund der Arbeitsabläufe, kein ständiger Kundenkontakt besteht, bietet dieses Geburtstagsmailing eine gute Möglichkeit, sich positiv beim Kunden in Erinnerung zu bringen.

### 1.5 Projektschnittstellen

Damit der Name des Endanwenders auf dem Bildschirm angezeigt werden kann, muss sichergestellt sein, dass 1. sein Name als get-Parameter an den Uniform Resource Locator (URL) mit übergeben wird, und 2. dass dieser aus der URL ausgelesen werden kann. Bei der Programmierung der Webapplikation wird der 2. Punkt sicherstellt.

Zum anderen muss die Webapplikation sicherstellen, dass eine „Kommunikation“ mit der Animation möglich ist. Dies ist durch die klickbaren Elemente bzw. mittels der touch-Gesten gewährleistet.

### 1.6 Projektabgrenzung

Da der Projektumfang begrenzt ist, wird der Zugriff auf die Datenbank nicht berücksichtigt, die Webapplikation gewährleistet nur das Auslesen des Namens des Endanwenders über den URL, und ist somit kein Bestandteil des Abschlussprojektes.

## 2 Projektplanung

In der Projektplanung werden sowohl die benötigte Zeit, die notwendigen Ressourcen sowie der Ablauf der Durchführung des Projektes berücksichtigt.

### 2.1 Projektphasen

Für die Umsetzung des Projektes standen dem Autor 70 Stunden zur Verfügung. Sie wurden vor Projektbeginn auf folgende Phasen verteilt und zeitlich festgelegt. Die einzelnen Phasen und ihre zeitliche Zuordnung sind schematisch aus der nachstehenden Tabelle, Tabelle 1:

---

Grobe Zeitplanung, ersichtlich. Eine detaillierte Darstellung befindet sich im Anhang 1 A: Detaillierte Zeitplanung auf Seite 18.

Projektphasen	Geplante Zeit in Stunden
Anforderungsanalyse	7,00
Konzeptionierung/Designphase	5,00
Entwicklung	42,00
Testumgebung	6,00
Dokumentation	8,00
Abnahme	2,00
<b>Summe:</b>	<b>70,00</b>

Tabelle 1: Grobe Zeitplanung

## 2.2 Ressourcenplanung

Im Anhang 2 A: Verwendete Ressourcen auf Seite 19 sind die verwendeten Ressourcen aufgelistet, die im Projekt verwendet wurden. Dabei handelt es sich sowohl um Hard- und Softwareressourcen als auch um Personal. Bei der Auswahl der verwendeten Software handelte es sich um Open Source Produkte oder um solche, für die t8y bereits Lizenzen besitzt. Dadurch wurden anfallende Kosten gering gehalten.

## 2.3 Entwicklungsprozess

Für die Realisierung des Projektes entschied sich der Autor für eine agile Softwareentwicklung.

Im Gegensatz zur klassischen Vorgehensweise, bei der im Vorhinein das fertige Layout und das Verhalten feststehen, geht die agile Softwareentwicklung schrittweise voran. Bei den einzelnen Entwicklungsschritten findet jeweils eine Rücksprache mit dem Kunden statt.

Die Webapplikation wurde entsprechend der agilen Softwareentwicklung in einzelnen Iterationsphasen erstellt, wobei in jeder einzelnen Entwicklungsstufe Rücksprache mit dem Head of Development stattfand. So konnte entweder unmittelbar eine Änderung bzw. Code Refactoring vorgenommen, oder die nächste Iterationsphase begonnen werden.

Die gesamte Entwicklung findet auf dem t8y eigenem Boilerplate statt. Dies gibt gewisse Strukturen und Frameworks vor.

### 3 Analysephase

#### 3.1 Ist-Analyse

Wie bereits im Abschnitt 1.1 (Projektbeschreibung) erwähnt wurde, handelt es sich bei dem Projekt, um eine Full Responsive Webapplikation, mit der das Ziel verfolgt wird, die langfristige und stabile Zusammenarbeit mit dem Kunden zu festigen und zu stärken.

Aufgrund des technischen Ablaufs einer Webapplikation, wie im Abschnitt 1.4 (Projektbegründung) erwähnt, kommt es häufig vor, dass der direkte Kundenkontakt bei den Werbeagenturen liegt. Da t8y als technischer Dienstleister, Designs und Layouts in lauffähige Webapplikationen bzw. Newsletters umsetzt, liegt dabei der Hauptkundenverkehr bei den Werbeagenturen, die das Layouten und Designen umsetzen. Diese haben in deren Entwicklungsphase immer wieder Rücksprache mit den Kunden, um so bestmögliche Arbeit gewährleisten zu können. Wenn dann t8y, die Umsetzung vornimmt, liegt der Hauptkontakt vorwiegend mit den Werbeagenturen vor und nicht mit dem Kunden selbst.

Um dennoch nicht bei den Kunden in Vergessenheit zu geraten, ist es sinnvoll mit einer Art „Aufmerksamkeit“, sich immer wieder ins Gedächtnis der Kunden zu rufen. So kann eine langfristige Zusammenarbeit erhöht und die Bindung zum Kunden untermauert werden. All dies kann mit der Webapplikation realisiert werden. Aus diesem Grund ist die Umsetzung des Projektes mehr als gerechtfertigt und vielversprechend.

Inwieweit die Webapplikation Früchte tragen wird, kann nur in den folgenden Jahren beurteilt und bewertet werden.

#### 3.2 Wirtschaftlichkeitsanalyse

Aufgrund des nicht voll ausgeschöpften Kundenkontakte, welcher in den Abschnitten 1.4 (Projektbegründung) und 3.1 (Ist-Analyse) geschildert und erläutert wurde, ist die Umsetzung des Projektes sehr sinnvoll. Ob die Realisierung aus wirtschaftlichen Gesichtspunkten und nicht-monetären Vorteilen gerechtfertigt ist, soll in den folgenden Abschnitten geklärt werden.

Da bei der Webapplikation kein direkt messbarer finanzieller Nutzen entsteht, kann nur ein qualitativer Nutzen benannt werden.

##### 3.2.1 Projektkosten

Die anfallenden Projektkosten werden im Folgenden kalkuliert. Hierbei sind Personalkosten ebenso zu berücksichtigen wie die Kosten für Ressourcen. Aus Datenschutzgründen dürfen die genauen Stundensätze nicht veröffentlicht werden. Aus diesem Grund wird mit einem pauschalen Stundensatz gerechnet, der von der Personalabteilung der Doyle Dane Bern-

bach Group GmbH (DDB) genannt wurde. Hiernach liegt der Stundensatz eines Umschülers bei 5,00 €, der eines Mitarbeiters bei 27,50 €. Die Ressourcennutzung wurde pauschal mit einem Stundensatz von 18,00 € veranschlagt.

Die Kostenaufstellung für das Projekt, findet sich in der Tabelle 2: Kostenaufstellung (Zeit wird in Stunden berechnet).

Vorgang	Mitarbeiter	Zeit	Personal	Ressourcen	Gesamt
Entwicklungskosten	1 x Umschüler	70 h	350,00 €	1.260,00 €	1.610,00 €
Fachgespräch	2 x Mitarbeiter	1 h	55,00 €	36,00 €	91,00 €
Code-Review	3 x Mitarbeiter	1 h	82,50 €	54,00 €	136,50 €
Abnahme	1 x Mitarbeiter	1 h	27,00 €	18,00 €	45,00 €
<b>Projektkosten gesamt 1.883,00 €</b>					

Tabelle 2: Kostenaufstellung

### 3.2.2 Amortisationsdauer

Die Amortisationsdauer wird berechnet, indem man die Anschaffungskosten durch die laufende Kostenersparnis dividiert.

Da es in diesem Fall keine Kostenersparnis gibt, ist es bei diesem Projekt nicht möglich, zu ermitteln, ab welchem Zeitpunkt sich die Entwicklung der Anwendung amortisiert hat, weil dieses Projekt mit keiner Kostenersparnis verglichen werden kann.

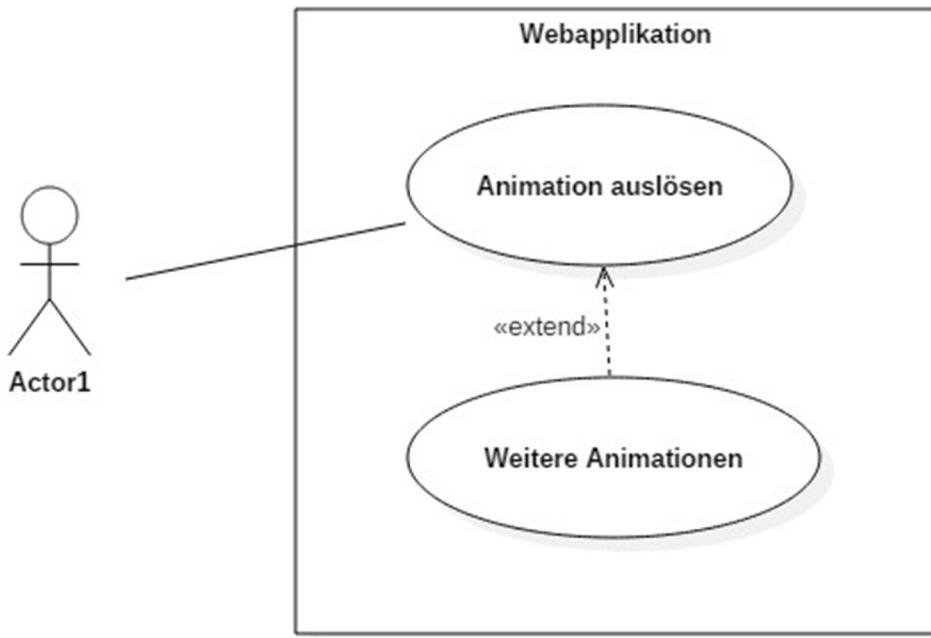
### 3.3 Nicht-monetäre Vorteile

Da es keine direkt messbaren Ergebnisse aus der monetären Sicht gibt, soll an dieser Stelle auf eine Analyse der nicht-monetären Vorteile eingegangen werden.

Für t8y ist es von größter Bedeutung, sich immer wieder positiv bei Ihren Kunden in Erinnerung zu rufen. Des Weiteren ist es wichtig, Kundenkontakte aufrecht zu halten und zu pflegen. All dies wird mit dem Projekt realisiert. Daher ist die Entwicklung des Projektes sinnvoll, auch wenn kein expliziter monetärer Vorteil erzielt werden kann.

### 3.4 Anwendungsfälle

Um eine grobe Übersicht, über den Anwendungsfall zu erhalten, der von dem umzusetzenden Programm abgedeckt werden soll, hat der Autor im Laufe der Analysephase folgendes Use-Case-Diagramm erstellt.



### 3.5 Lastenheft

Im Anschluss an die Analysephase erstellte der Autor in Zusammenarbeit mit dem Head of Webdevelopment das Lastenheft. Hierin sind alle Anforderungen des Auftragsgebers (GF) enthalten die an die zu erstellende Webapplikation gestellt werden. Der Anhang 3 A: Lastenheft (Auszug) auf Seite 20 gewährt einen Einblick.

## 4 Entwurfsphase

Vor der eigentlichen Implementierung des Projektes wird eine Entwurfsphase durchgeführt. Hierbei wurde ein grober Entwurf erstellt, wie die Anwendung später aussehen sollte und wie der Ablauf der Animation sein sollte.

Als Resultat der Entwurfsphase entstand ein Pflichtenheft, welches den Auftraggebern des Projektes vorgelegt wurde.

### 4.1 Zielpлатform

Die Auswahl der Programmiersprachen, mit der das Projekt umgesetzt werden soll, ist vorgegeben, da das Projekt mit dem firmeneigenen Boilerplate umgesetzt werden soll.

Bei den Programmiersprachen handelt es sich einmal um HTML, wobei die einzelnen Templates mit Handlebars.js realisiert werden. Des Weiteren wird CSS3, welches mit Less geschrieben wird, eingesetzt. Bei Less handelt es sich um einen Präprozessor für CSS. Außerdem wird auch Bootstrap angewendet, welches ein CSS-Framework darstellt. Als Program-

---

miersprache wird JavaScript verwendet und jQuery, welches ein Framework für JavaScript ist.

Im Boilerplate selbst, wird für JavaScript das Framework Backbone.js verwendet sowie Require.js zum Einbinden der einzelnen js-Dateien. An dieser Stelle sei kurz erwähnt, dass man z.B. auch Angular.js statt Backbone oder SASS anstatt Less hätte verwenden können. Bei der Vielzahl an Frameworks die es auf dem Markt gibt, stehen die eben genannten stellvertretend und sind ohne Bewertung zu betrachten.

## 4.2 Architekturdesign

Da das Projekt im firmeneigenen Boilerplate, wie im Abschnitt 4.1 (Zielplattform) beschrieben, entwickelt werden soll, ist neben den Programmiersprachen auch die Architektur vorgegeben. Im Boilerplate ist ein abgewandeltes Model View Controller (MVC) integriert, welches auf das Framework Backbone.js aufsetzt. Dabei wird das View, durch HTML (Handlebars) realisiert. Der Controller besteht aus einzelnen js-Dateien, welche Events des View entgegennehmen und am Graphical User Interface (GUI) updates vornehmen. Das Model besteht ebenfalls aus js-Dateien. Es dient hier, um Daten zwischen zu lagern, zu sharen und zwischen mehreren Controllern auszutauschen.

Bei der Entwicklung der Webapplikation wird auf das Model verzichtet, da in dieser Anwendung keine Daten für einen längeren Zeitraum gespeichert werden müssen. Der Name des Kunden, welcher als get-Parameter an die URL übergeben wird, wird nur temporär gespeichert.

## 4.3 Benutzeroberfläche

Damit die Benutzeroberfläche nach Fertigstellung der Webapplikation den Anforderungen und Vorstellungen der GF entspricht, wurde zunächst ein Prototyp entwickelt.

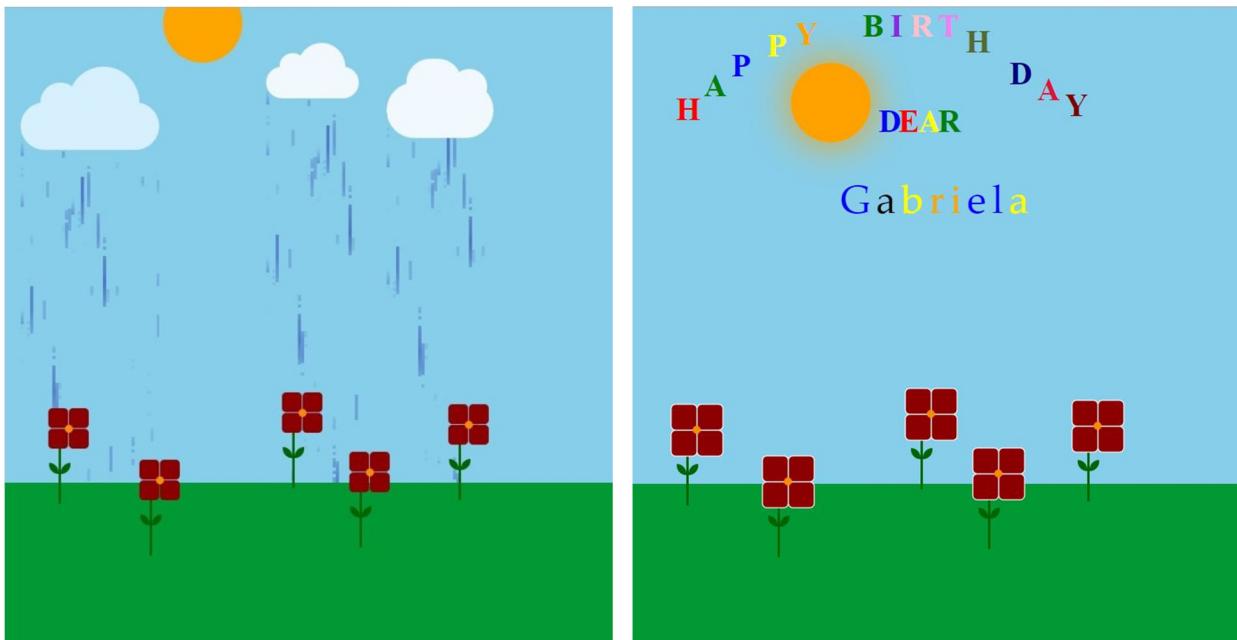


Abbildung 1: Screenshot vom Prototyp der Webapplikation

Dieser war zunächst lediglich für eine Desktopanwendung ausgelegt. Anhand dieses Prototyps wurde dann die eigentliche Webapplikation entwickelt.

Die Benutzeroberfläche soll in vier verschiedene Viewport-Größen unterteilt werden. Dabei soll es zum einen ein Design für Mobiltelefone geben zum anderen eines für Tablets. Für die Desktopanwendung soll es zwei verschiedene Designs geben.

Die Schwierigkeit dabei ist, dass die Webapplikation nicht zu leicht aber auch nicht zu schwer sein darf. Der User soll herausfinden können, welches die klickbaren Elemente sind, bzw. wie er über eine touch-Geste mit der Animation interagieren kann. Deshalb muss die Benutzeroberfläche so gestaltet sein, dass der User zwar in der Lage ist, zu erkennen, wie er mit der Animation interagieren kann, aber auch nicht sofort sichtbar ist, was als nächstes bedient werden muss.

Bei der Webapplikation bekam der Autor die Möglichkeit, einen eigenen Ablauf für die Animation zu entwickeln. Es gab eine Einschränkung, die darin bestand, dass keine Luftballons und Kerzen in der Animation vorkommen sollten. Der vom Autor entwickelte Ablauf wird ausführlich im Abschnitt 7.1 (Benutzerhandbuch) geschildert.

Für die Animation selbst, hat sich der Autor für ein Zusammenspiel aus Blumen, Wolken und Sonne entschieden. Diese stellen auch die klickbaren bzw. über touch-Geste erreichbaren Elemente dar.

## 4.4 Animationslogik

Die Animationslogik ist unmittelbar mit der Benutzeroberfläche verknüpft. Wenn ein klickbares Element entweder durch touch-Geste oder per Mausklick berührt wird, löst dieses in der Animation eine Reaktion aus. Die einzelnen Abläufe innerhalb der Animation sind in unterschiedliche Module bzw. Dateien aufgeteilt.

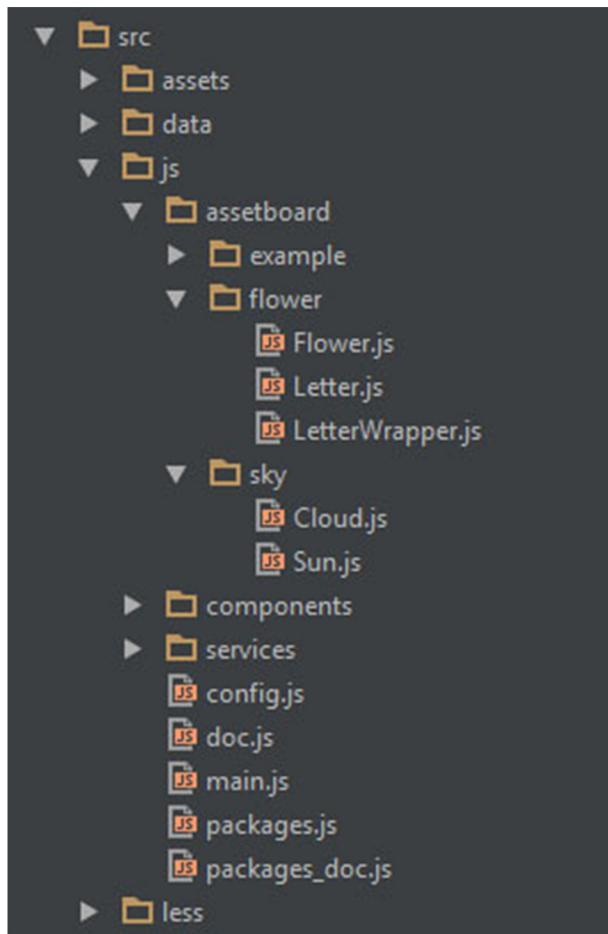
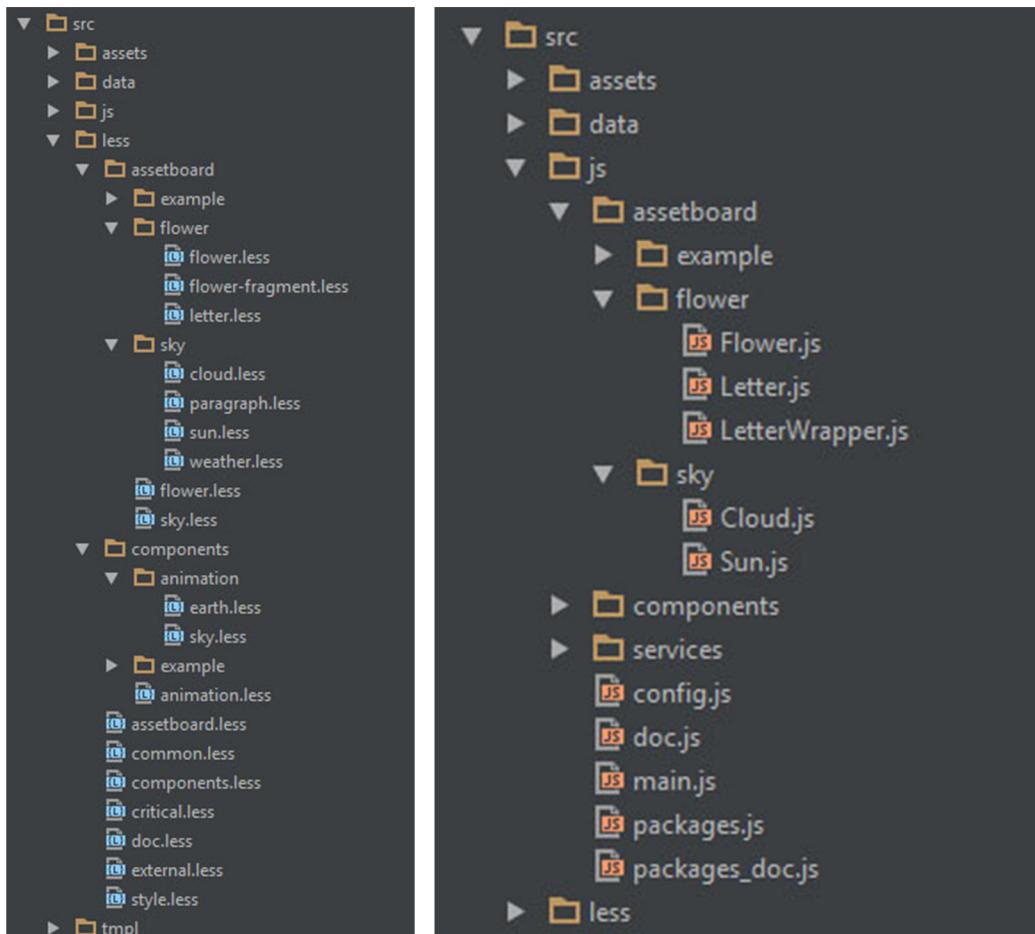


Abbildung 2: Datenstruktur von den Less-Dateien und den js-Dateien

Die ausgelöste Reaktion wird ausgeführt und die Veränderung ist auf dem Display sichtbar. Danach stehen dem Anwender weitere klickbare Elemente zur Verfügung, wie bereits im Abschnitt 4.3 (Benutzeroberfläche) geschildert.

Die Logik der Animation wird soweit es geht mit CSS3 Transition und der CSS3 Animation umgesetzt. Was nicht mit CSS3 realisierbar war, wurde mit JavaScript und dem Framework jQuery programmiert.

## 4.5 Maßnahmen zur Qualitätssicherung

Während der Entwicklung der Webapplikation nimmt der Autor selbstständig Tests vor. Das heißt, dass nach jedem logischen Abschnitt, welcher entwickelt wurde, Tests vorgenommen

werden. Diese Tests finden erst einmal auf den Mobiletelefonen statt, danach auf unterschiedlichen Tablets und zuletzt werden die Desktop-Versionen überprüft. Dabei ist es wichtig, zu überprüfen, ob das bisher programmierte noch einwandfrei funktioniert. Dann wird das neu implementierte überprüft. So kann sichergestellt werden, dass die gesamte programmierte Anwendung etwa durch einen neu hinzugefügten Code nicht mehr funktioniert oder fehlerhaftes Verhalten bzw. Aussehen aufweist.

Nach Fertigstellung der Webapplikation, wird die Anwendung in der internen Qualitätssicherungsabteilung (QS) noch einem finalen Testdurchlauf unterzogen.

## 4.6 Pflichtenheft

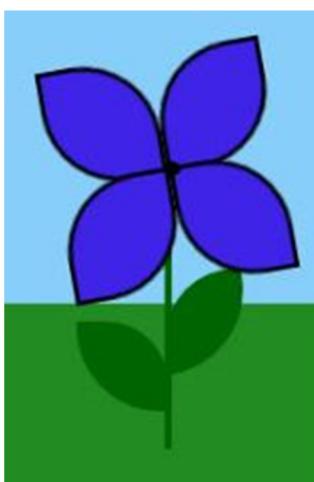
Im Anschluss an die Entwurfsphase wurde vom Autor ein Pflichtenheft erstellt, das auf das Lastenheft (vgl. dazu Abschnitt 3.4 (Anwendungsfälle)) Bezug nimmt. Hierin wird beschrieben, wie und womit die Webapplikation umgesetzt werden kann. Es stellt also den Leitfaden für die Realisierung des Projektes dar. Ein Auszug aus dem Pflichtenheft ist im Anhang 4 A: Pflichtenheft (Auszug) auf Seite 21 dargestellt.

# 5 Implementierungsphase

## 5.1 Iterationsplanung

Bevor die eigentliche Implementierung begonnen wurde, hat der Autor einen Iterationsplan erstellt. In diesem werden die einzelnen Iterationsschritte dargestellt und geben dem Entwickler somit eine Orientierung bei der Programmierung an die Hand. Er befindet sich im Anhang 5 A: Iterationsplan auf Seite 22.

## 5.2 Implementierung der Benutzeroberfläche/Animation



Zuerst wurde der Hintergrund mit HTML bzw. mit Handlebars.js erstellt. Danach wurden die einzelnen Elemente, wie Sonne, Wolken und Blumen implementiert, ebenfalls mit HTML und Handlebars.js. So ist es zum Beispiel möglich, nur eine einzige Blume zu entwickeln, die dann über eine JSON-Datei mehrfach eingelesen werden kann.

Abbildung 3: Blume aus der Webapplikation

### 5.2.1 Handlebars-Code erzeugt HTML-Template

```
<div class="assetboard {{class}}" data-assetboard="flower/flower">
  <div class="fragments">
    {{#each fragments-letters}}
      {{{partial "assetboard/flower/flower-fragment" flower-
      fragment}}}
    {{/each}}
  </div>
  {{#each fragments-steam}}
    {{{partial "assetboard/flower/flower-fragment" flower-
      fragment}}}
  {{/each}}
</div>
```

Abbildung 4: Code-Beispiel einer Handlebar-Datei. Hier flower.handlebars

Bei diesem Codebeispiel ist zu sehen, dass mit dem Framework Handlebars.js ein HTML Template für die Blume erstellt wurde. Dabei werden zwei Schleifen durchlaufen, bei der ersten Schleife werden die Blüten und der Stempel (fragments-letters) erzeugt. Beim Durchlauf der zweiten Schleife, werden der Stamm und die Blätter (fragments-steam) gebildet.

### 5.2.2 JSON-Datei enthält Inhalt der Webseite

```
. . . "earth": {
  "class": "flower earth",
  "flowers": [
    {
      "class": "flower1 flower",
      "fragments-letters": [
        {
          "class": "topleft petal"
        },
        {
          "class": "topright petal"
        },
        {
          "class": "bottomleft petal"
        },
        {
          "class": "bottomright petal"
        },
        {
          "class": "center"
        }
      ]
    }
  ]
}
```

```
        ],
        "fragments-steam": [
            {
                "class": "leftleaf leaf"
            },
            {
                "class": "steam"
            },
            {
                "class": "rightleaf leaf"
            }
        ]
    },
}
```

Abbildung 5: Code-Beispiel (Ausschnitt) einer JSON-Datei. Hier animation.json

Der vollständige Code der Datei befindet sich im Anhang 6 A: Animation.json auf Seite 23.

Bei diesem Codebeispiel ist zu sehen, dass sich in der JSON-Datei ein Array namens „flowers“ befindet, welches die ganzen Blumen enthält. Hier ist exemplarisch flower1 abgebildet. Innerhalb dieses Arrays befinden sich zwei weitere Arrays, dabei handelt es sich um die zwei angesprochenen Arrays, „fragments-letters“ und „fragments-steam“.

Nach diesem Vorgehen, wurden die einzelnen Elemente entwickelt, um sie danach mit CSS positionieren und stylen zu können. Dieses wurde mit dem Präprozessor Less verwirklicht.

### 5.2.3 Less-Code für Formatierung und Gestaltung

```
.assetboard[data-assetboard="flower/flower"] {
    position: absolute;
    z-index: 3;

    &.flower1 {
        width: 11%;
        padding-top: 21%;
        left: 28%;
        bottom: 15%;
    }
    &.flower2 {...}
    &.flower3, &.flower4, &.flower5 {
        display: none;
    }
    @media screen and (min-width: @screen-sm-min) {
        &.flower {
            display: block;
        }
    }
}
```

```
    }
    &.flower1, &.flower2 {
        display: none;
    }
    &.flower3 {
        width: 12%;
        padding-top: 25%;
        left: 15%;
        bottom: 30%;
    }
    &.flower4 { ... }
    &.flower5 { ... }
}
@media screen and (min-width: @screen-lg-min) {
    ...
}
```

Abbildung 6: Code-Beispiel (Ausschnitt) einer Less-Datei. Hier flower.less  
Der vollständige Code der Datei befindet sich im Anhang 7 A: Flower.less auf Seite 27.

Bei diesem Codebeispiel sind die Positionierung sowie die Größe der Blumen zu sehen, wobei es auch hier exemplarisch an den Blumen „flower1“ und „flower3“ gezeigt wird.

#### 5.2.4 JavaScript überwacht Click-Events

Die Animation wurde, soweit es geht, mit den neuen Techniken von CSS3 realisiert. Dazu gehört unter anderem Transition als auch Transform. Was nicht mit CSS3 zu realisieren ging, wurde mit JavaScript umgesetzt. Dazu gehört z.B., auf Click-Events zu hören.

```
function onClick(e) {
    var $currentTarget = $(e.currentTarget);

    if ($currentTarget.hasClass('click')) {
        $currentTarget.removeClass('click');
        $('.rain').addClass('rain-disappear').find('.cloud').
            addClass('cloud-disappear');

        setTimeout(function () {
            $('.sun').addClass('set-sun-shine');
        }, 7000);
    } ...
}
```

Abbildung 7: Code-Beispiel (Ausschnitt) einer js-Datei. Hier flower.js  
Der vollständige Code der Datei befindet sich im Anhang 8 Flower.js auf Seite 29.

Bei diesem Codeabschnitt ist zusehen, dass auf das Click Event gehört wird. Beim Klicken eines klickbaren Elements, werden z.B. Klassen hinzugefügt sowie entfernt

### 5.2.5 JavaScript steuert das Verhalten

Bei der Anordnung der Buchstaben vom „Happy Birthday Dear“ gab es Probleme, dies mit CSS3 zu realisieren. Es kam zu Problemen beim Zusammenspiel von unterschiedlichen Zustandspositionen, der Elemente. Die Buchstaben hätten „fixed“ positioniert werden müssen, um so das responsive Verhalten zu gewährleisten, wenn sie sichtbar sind und den Bogen darstellen. Zuvor hätten sie jedoch „absolute“ positioniert sein müssen, damit sie innerhalb einer Blume unsichtbar bleiben. Daher hat sich der Autor an dieser Stelle für eine Lösung mit JavaScript entschieden. So konnten die Buchstaben problemlos „absolute“ positioniert werden, und die Anordnung des Bogens führt JavaScript aus. Hierfür wird die Formel der Parabel verwendet.

```
function parabel(a, x, c, topPadding) {
    return a * (x * x - c) + topPadding;
}

function animateHappyBirthday(letters) {
    var i,
        lettersLength = letters.length,
        top,
        left,
        distance = 100 / (lettersLength + 2),
        offset = distance / 2,
        c = 50 - offset,
        topPadding = 3;

    for (i = lettersLength / -2; i < lettersLength / 2; i += 1)
    {
        left = distance * i + offset + c;
        top = parabel(a, left - c, c, topPadding);
        $('.letter' + (i + lettersLength / 2)).
            data({position: {left: left + '%', top: top + '%'}});
    }
}
```

Abbildung 8: Code-Beispiel (Ausschnitt) einer js-Datei. Hier letterWrapper.js  
Der vollständige Code der Datei befindet sich im Anhang 9 LetterWrapper.js auf Seite 30.

Bei dem dargestellten Code bestimmt die Funktion parabel die Form der Parabel, sprich, sie übernimmt die Anordnung der Buchstaben. Der Betrag von a bestimmt dabei, wie schmal bzw. wie breit die Parabel ist. Der Parameter c gibt die x-Koordinate des Scheitelpunkts an.

Dabei kommt es auf das Vorzeichen an. Hier ist es ein „-“ und verschiebt somit die Parabel nach rechts. „topPadding“ verschiebt die Parabel auf der y-Achse, hier nach unten.

Die Funktion `animateHappyBirthday` erhält vom Typ String ein Argument „Happy Birthday Dear“. Innerhalb der Funktion werden zunächst die benötigten Variablen deklariert und z.T. mit Werten versehen.

Beim Schleifendurchlauf werden für jeden einzelnen Buchstaben dessen Position errechnet und in den Variablen „left“ und „top“ gespeichert. Diese Werte werden dann dem Element mit der Klasse „letter“ angehängt. Bei dem Aufruf wird das Framework jQuery verwendet. Die einzelnen Positionsdaten von „left“ und „top“ werden als JSON-Objekt dem Framework internen data-Objekt übergeben. Dies ist ein Objekt von einem Controller und übergibt hier neue Werte an das View.

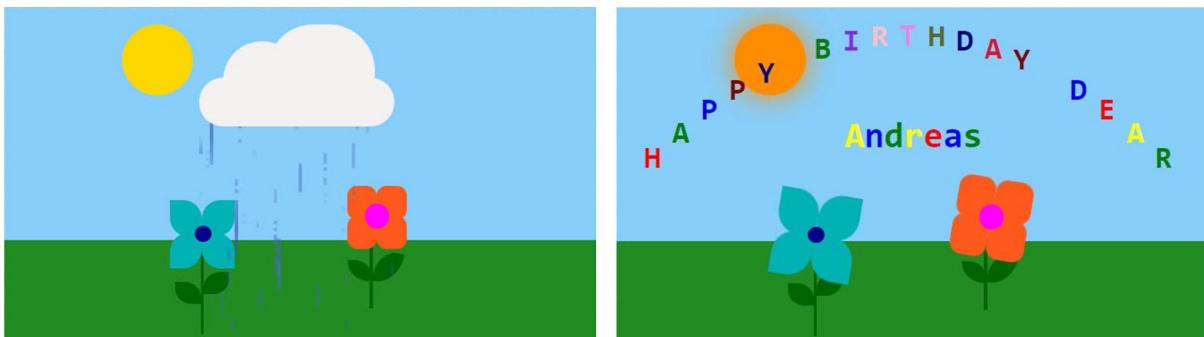


Abbildung 9: Screenshots von der Webapplikation

Bei den beiden Bildern handelt es sich um Screenshots der Webapplikation, in diesem Fall der mobilen Version (Landscape). Den kompletten Bildverlauf der Animation der mobilen Version ist im Anhang 10 Animationsablauf für Mobile (Ausschnitt) auf Seite 32 abgebildet und für Mobile-Portrait im Anhang 11 Animationsablauf für Mobile-Portrait (Ausschnitt) auf Seite 33. Des Weiteren sind einzelne Ausschnitte der Tablet Version im Anhang 12 Animationsablauf für Tablet (Ausschnitt) auf Seite 34 und der Desktop Version im Anhang 13 Animationsablauf für Desktop (Ausschnitt) auf Seite 35 abgebildet.

## 6 Abnahme

Für die endgültige Abnahme und deren Testung der Anwendung, wurde Sie auf einen Testserver abgelegt, um so wahrheitsgetreu testen zu können. Bevor die Anwendung zur Qualitätssicherung übergeben wurde, führte der Autor eigenständig Tests vom World Wide Web Consortium (W3C) zur Validierung fürs HTML und CSS durch. Des Weiteren führte der Autor einen Pagespeed Test von Google Developers durch.

## 6.1 Abnahme durch die Abteilung der Qualitätssicherung

Nachdem der Autor die Webapplikation fertig gestellt hatte, wurde sie intensive in der firmeneigenen Abteilung für Qualitätssicherung (QS) getestet. Auf diese Weise wurde sicher gestellt, dass die Webapplikation fehlerfrei funktioniert und allen Anforderungen der GF nachkommt. Die QS stellte keine Mängel fest.

## 6.2 Abnahme durch die Geschäftsführung

Nachdem die Anwendung der QS unterzogen wurden war, konnte sie nun der GF zur Abnahme übergeben werden. Diese war mit dem Endprodukt zufrieden. So konnte die Anwendung zum deployen frei gegeben werden.

## 6.3 Deployment

Damit der Zugriff auf die Webapplikation für alle Projektmanager möglichst einfach zu realisieren ist, wurde die Anwendung auf ein bereits bestehendes Netzlaufwerk bereitgestellt. Der Vorteil bestand darin, dass jeder schon einen existierenden Zugriff besaß. Die Anwendung wurde als lauffähiges Programm abgelegt.

# 7 Dokumentation

Zum Schluss des Projekts soll noch auf das Benutzerhandbuch und die Entwicklerdokumentation eingegangen werden.

## 7.1 Benutzerhandbuch

Da es sich bei der Webapplikation um ein Geburtstagsmailing handelt, deren zentralen Funktionsweise es ist, dass der Endanwender/Kunde selber heraus bekommen soll, wie der Ablauf der Anwendung funktioniert, wurde auf ein detailliertes Benutzerhandbuch verzichtet.

## 7.2 Entwicklerdokumentation

Auf eine Entwicklerdokumentation wurde ebenfalls verzichtet. An den Autor und somit an den Programmierer der Anwendung wurde die Herausforderung gestellt, selbst erklärenden Code zu schreiben, so dass eine zusätzliche Entwicklerdokumentation nicht notwendig ist. Der Code durfte und sollte gegebenenfalls mit zusätzlichen Kommentaren versehen werden. Als Vorlage solch einen Code zu schreiben, stand der Standpunkt von Robert C. Martin, im Vordergrund.

# 8 Fazit

Zum Schluss soll noch ein Fazit vom Autor gezogen werden.

## 8.1 Soll-Ist-Vergleich

Nun soll das Projekt rückblickend betrachtet werden. Dabei kann man feststellen, dass es doch zu recht vielen Abweichungen der zuvor erstellten Zeitplanung gekommen ist. Siehe Tabelle 3: Soll-Ist-Vergleich (Zeitangaben sind in Stunden zu interpretieren).

Projektphasen	Soll	Ist	Differenz
Anforderungsanalyse	7,00	6,00	- 1,00
Konzeptionierung / Designphase	5,00	6,00	+ 1,00
Entwicklung	42,00	39,00	- 3,00
Testumgebung	6,00	7,00	+ 1,00
Dokumentation	8,00	10,00	+ 2,00
Abnahme	2,00	2,00	+/- 0,00
<b>Summe:</b>	<b>70,00</b>	<b>70,00</b>	<b>0 h</b>

Tabelle 3: Soll-Ist-Vergleich

Insgesamt konnte die geforderte Zeit der HK von einem festgelegten Zeitraum von 70 Stunden jedoch eingehalten werden. Hinzuzufügen ist noch, dass alle an dem Autor geforderten Anforderungen an das Projekt gemäß dem Pflichtenheft umgesetzt wurden.

Durch die Umsetzung des Projektes, konnte der Autor viel Erfahrung sammeln in Bezug auf Planung und Umsetzung eines Projektes. Dabei wurde besonders deutlich, wie komplex und facettenreich die Webentwicklung ist. Ebenso wurde sehr deutlich, wie wichtig die Kommunikation untereinander und zwischen Auftraggeber und Auftragnehmer ist. Des Weiteren konnte der Autor feststellen, wie wichtig Lasten- und Pflichtenhefte sind, da sie die Richtlinien zur Programmierung vorgeben. Zum Schluss sei noch erwähnt, dass die Realisierung des Projekts eine große Bereicherung und einen großen Lernfaktor für den Autor darstellte.

## 8.2 Ausblick

Ob das Projekt den Anforderungen gerecht werden kann, ist zu diesem Zeitpunkt schwer zu beurteilen. Im Gegensatz zu vielen anderen Projekten, ist es in diesem Fall schwer zu beurteilen, ob die Webapplikation den Anforderungen nachkommen kann. Da es z.B. keine erfassten Zahlen, Zeiten oder ähnliches gibt, welche miteinander verglichen werden können. Es hängt viel von der Einschätzung des Projektmanagements ab, ob und in wie weit die Webapplikation zur Kundenbindung beitragen kann. Eine eventuelle Faktorenanalyse, wie sie z.B. auch in der Psychologie eingesetzt wird, könnte mehr Aufschluss geben. Ob sich der Aufwand dafür lohnt, möchte der Autor anderen überlassen.

## 9 Anhang

### 1. A: Detaillierte Zeitplanung

Projektphasen	Zeitplanung in Stunden
<b>Anforderungsanalyse</b>	<b>7,00</b>
• Ist-Analyse	1,00
• Soll-Konzept	2,00
• Kosten-Nutzen-Betrachtung	1,00
• Erstellen eines Pflichtenheftes	3,00
<b>Konzeptionierung / Designphase</b>	<b>5,00</b>
• Entwurf des Designs und Layouts	2,00
• Entwurf des Programmkonzeptes	2,00
• Ermittlung der Programmiersprachen	1,00
<b>Entwicklung</b>	<b>42,00</b>
• Einrichten des Projektes	1,00
• Einrichten des firmeneigenen Boilerplates	3,00
• Programmierung	38,00
<b>Testumgebung</b>	<b>6,00</b>
• Qualitätssicherung der Funktionalität	1,00
• Testdurchführung auf diversen Geräten und Browsern	4,50
• Sicherstellung eines Fallbacks	0.50
<b>Dokumentation</b>	<b>8,00</b>
• Projektdokumentation	6,00
• Anwenderdokumentation	1,00
• Ist-Soll-Vergleich	1,00
<b>Abnahme</b>	<b>2,00</b>
• Abnahme des Projektes	1,00
• Deployment	1,00
<b>Summe</b>	<b>70,00</b>

---

## 2. A: Verwendete Ressourcen

### Hardware:

- Büroarbeitsplatz

### Software:

- Windows 7 Enterprise – Betriebssystem
- IntelliJ IDEA 13.16 – Entwicklungsumgebung
- StarUML
- Photoshop
- Firmen internes Boilerplate
- Git – Verteilte Versionsverwaltung
- Word 2010
- Excel

### Personal:

- Mitarbeiter des Frontendteams / Head of Webdevelopment - Festlegung der Anforderungen und Abnahme des Projektes
- Umschüler – Umsetzung des Projektes

### 3. A: Lastenheft (Auszug)

Der nachstehende Auszug aus dem Lastenheft definiert die Anforderungen, die an die zu entwickelnde Webapplikation gestellt werden.

#### Anforderungen

Von der Webapplikation müssen folgende Anforderungen erfüllt werden:

- Die Webapplikation muss in allen modernen Web-Browsern lauffähig sein, sowie in allen modernen mobilen-Geräten.
- Die Webapplikation muss sich auf allen Viewport-Größen sämtlich erdenklicher Maße anpassen können.
- Die Webapplikation muss für Browser sowie für mobile-Geräte, welche nicht die modernsten Techniken unterstützen, eine alternative zur Verfügung stellen.
- Die Webapplikation sollte bei Veränderungen auf der Webseite, darauf so schnell wie es die Technologien zu lassen, reagieren können.
- Die Webapplikation muss Werte aus der URL auslesen können.
- Die Webapplikation muss dem Endanwender Möglichkeiten der Interaktivität zur Verfügung stellen.

[...]

---

#### 4. A: Pflichtenheft (Auszug)

Der nachstehende Auszug aus dem Pflichtenheft beschreibt wie die Umsetzung der im Lastenheft definierten Anforderungen auszusehen soll.

#### Umsetzung der Anforderungen

- Die Webapplikation wird mit den modernsten Technologien der Webentwicklung wie HTML5, CSS3 und JavaScript umgesetzt.
- Die Webapplikation wird als Full Responsive Webapplikation, unter Einbindung von Media Queries, umgesetzt.
- Die Webapplikation wird eine Default Seite für Browser und Mobile Geräte bereitstellen, die nicht die modernsten Technologien interpretieren können.
- Die Webapplikation (Animation) wird soweit es geht mit CSS3 umgesetzt, um so das schnellst mögliche Rendern zu gewährleisten.
- Das Auslesen von übergebenen Werten als GET-Parametern in der URL, wird mit der Programmiersprache JavaScript realisiert.
- Die Webapplikation wird klickbare bzw. mittels touch-Geste auslösbar Elemente dem Endanwender zur Verfügung stellen, damit dieser mit der Webapplikation (Animation) interagieren kann.

[...]

---

## 5. A: Iterationsplan

- Erstellung der grafischen Oberfläche für die mobile Version
  - Erstellung der einzelnen Module (Himmel, Erde, Blumen, Sonne)
- Implementierung der klick-Elemente
  - Blumen, Wolken, Sonne
- Integration des Regens
- Implementierung der Animationslogik
  - Wolken
  - Sonne
  - Blumen
- Erstellung der grafischen Oberfläche für die Tablet Version sowie der Ergänzung weitere Module, klick Elemente usw.
- Erstellung der grafischen Oberfläche für die 2 Desktop Versionen sowie der Ergänzung weiterer Module, klick Elemente usw.

## 6. A: Animation.json

```
{
  "page": {
    "basePath": "",
    "title": "Happy Birthday",
    "head": {
      "meta": {
        "description": "",
        "keywords": ""
      },
      "socialmedia": {
        "title": "",
        "description": "",
        "image": {
          "src": "",
          "width": 200,
          "height": 200
        }
      }
    },
    "body": {
      "sky": {
        "class": "sky",
        "weather-condition": [
          {
            "class": "weather weather1",
            "cloud": {
              "class": "cloud cloud1"
            }
          },
          {
            "class": "weather weather2",
            "cloud": {
              "class": "cloud cloud2"
            }
          },
          {
            "class": "weather weather3",
            "cloud": {
              "class": "cloud cloud3"
            }
          }
        ],
        "sun": {
          "class": "sun click"
        },
        "paragraph": {
          "class": "name",
          "name": ""
        }
      },
      "earth": {
        "class": "flower earth",
        "flowers": [

```

```
{
  {
    "class": "flower1 flower",
    "fragments-letters": [
      {
        "class": "topleft petal"
      },
      {
        "class": "topright petal"
      },
      {
        "class": "bottomleft petal"
      },
      {
        "class": "bottomright petal"
      },
      {
        "class": "center"
      }
    ],
    "fragments-steam": [
      {
        "class": "leftleaf leaf"
      },
      {
        "class": "steam"
      },
      {
        "class": "rightleaf leaf"
      }
    ]
  },
  {
    "class": "flower2 flower",
    "fragments-letters": [
      {
        "class": "topleft petal"
      },
      {
        "class": "topright petal"
      },
      {
        "class": "bottomleft petal"
      },
      {
        "class": "bottomright petal"
      },
      {
        "class": "center"
      }
    ],
    "fragments-steam": [
      {
        "class": "leftleaf leaf"
      },
      {
        "class": "steam"
      },
      {
        "class": "rightleaf leaf"
      }
    ]
  }
}
```

```
        "class": "rightleaf leaf"
    }
]
},
{
    "class": "flower3 flower",
    "fragments-letters": [
        {
            "class": "topleft petal"
        },
        {
            "class": "topright petal"
        },
        {
            "class": "bottomleft petal"
        },
        {
            "class": "bottomright petal"
        },
        {
            "class": "center"
        }
],
    "fragments-steam": [
        {
            "class": "leftleaf leaf"
        },
        {
            "class": "steam"
        },
        {
            "class": "rightleaf leaf"
        }
]
},
{
    "class": "flower4 flower",
    "fragments-letters": [
        {
            "class": "topleft petal"
        },
        {
            "class": "topright petal"
        },
        {
            "class": "bottomleft petal"
        },
        {
            "class": "bottomright petal"
        },
        {
            "class": "center"
        }
],
    "fragments-steam": [
        {
            "class": "leftleaf leaf"
        },
        {
            "class": "steam"
        },
        {
            "class": "rightleaf leaf"
        }
]
}
```

```
        {
            "class": "steam"
        },
        {
            "class": "rightleaf leaf"
        }
    ]
},
{
    "class": "flower5 flower",
    "fragments-letters": [
        {
            "class": "topleft petal"
        },
        {
            "class": "topright petal"
        },
        {
            "class": "bottomleft petal"
        },
        {
            "class": "bottomright petal"
        },
        {
            "class": "center"
        }
    ],
    "fragments-steam": [
        {
            "class": "leftleaf leaf"
        },
        {
            "class": "steam"
        },
        {
            "class": "rightleaf leaf"
        }
    ]
}
]
```

## 7. A: Flower.less

```
.assetboard[data-assetboard="flower/flower"] {

  /* FLOWERS
  ****
  position: absolute;
  z-index: 3;

  ///////////////////////////////////////////////////
  //
  // DEFAULT  DEFAULT  DEFAULT  DEFAULT
  //
  ///////////////////////////////////////////////////
  /* FLOWER 1, 2
  ****
  &.flower1 {
    width: 11%;
    padding-top: 21%;
    left: 28%;
    bottom: 15%;
  }
  &.flower2 {
    width: 10%;
    padding-top: 20%;
    left: 58%;
    bottom: 35%;
  }
  &.flower3,
  &.flower4,
  &.flower5 {
    display: none;
  }

  ///////////////////////////////////////////////////
  //
  // TABLET  TABLET  TABLET  TABLET
  //
  ///////////////////////////////////////////////////
  @media screen and (min-width: @screen-sm-min) {

    /* FLOWER 3, 4, 5
    ****
    &.flower {
      display: block;
    }
    &.flower1,
    &.flower2 {
      display: none;
    }
    &.flower3 {
      width: 12%;
      padding-top: 25%;
      left: 15%;
      bottom: 30%;
    }
    &.flower4 {
      width: 13%;
      padding-top: 27%;
```

```
        left: 55%;  
        bottom: 60%;  
    }  
    &.flower5 {  
        width: 11%;  
        padding-top: 24%;  
        left: 75%;  
        bottom: 45%;  
    }  
}  
  
// DESKTOP DESKTOP DESKTOP  
//  
/* FLOWER 1, 2, 3, 4, 5 */  
&.flower1,  
&.flower2 {  
    display: block;  
}  
&.flower1 {  
    left: 5%;  
    bottom: 10%;  
}  
&.flower2 {  
    left: 20%;  
    bottom: 35%;  
}  
&.flower3 {  
    left: 45%;  
    bottom: 3%;  
}  
&.flower4 {  
    left: 58%;  
    bottom: 25%;  
}  
&.flower5 {  
    left: 82%;  
    bottom: 12%;  
}  
}  
  
/* FLOWER 3 */  
&.flower3 {  
    left: 95%;  
    bottom: 50%;  
}  
}
```

## 8. Flower.js

```
define(['jquery', 't8y/base/Controller'], function ($, Controller) {  
  
    return Controller.extend({  
  
        initialize: function (options) {  
            Controller.prototype.initialize.apply(this, arguments);  
            this.$el.on('click', function(e) {  
                onClick(e);  
            })  
        }  
    });  
  
    function onClick(e) {  
  
        var $currentTarget = $(e.currentTarget);  
  
        if ($currentTarget.hasClass('set-flower-animation')) {  
            $('.letter').addClass('start');  
            $('main').data('interface').model.trigger('fly');  
            $('.name').removeClass('name').find('span').addClass('show-  
name');  
        }  
    }  
});
```

## 9. LetterWrapper.js

```
define(['jquery', 't8y/base/Controller', 't8y/base/DomModel', 't8y/services/js'], function ($, Controller, DomModel, parser) {

    return Controller.extend({
        model: DomModel,

        initialize: function (options) {
            Controller.prototype.initialize.apply(this, arguments);
            this.model.on('fly', onFly.bind(this));
            onResize();
            $(window).resize(onResize);
        }
    });

    function getCustomerName() {
        var customerName = "Customer";
        if (getUrlVar('name') !== undefined) {
            customerName = getUrlVar('name');
        }
        return customerName;
    }

    function onFly() {
        this.$('[data-assetboard="flower/letter"]').each(function (i, node) {
            var $node = $(node);
            $node.css($node.data('position'));
        });
    }

    function getUrlVars() {
        var vars = [],
            hash,
            hashes = window.location.href.slice(window.location.href.indexOf('?') + 1).split('&'),
            i;
        for (i = 0; i < hashes.length; i += 1) {
            hash = hashes[i].split('=');
            vars.push(hash[0]);
            vars[hash[0]] = hash[1];
        }
        return vars;
    }

    function getUrlVar(name) {
        return getUrlVars()[name];
    }

    function setColorsToName() {
        var colors = ['red', 'green', 'blue', 'yellow', 'orange'],
            colorsLength = colors.length - 1,
            colorName,
            i,
            customerName = getCustomerName(),
            nameLength = customerName.length,
            randomNumber;
```

```
for (i = 0; i < nameLength; i += 1) {

    randomNumber = Math.round(Math.random() * colorsLength);
    colorName = colors[randomNumber];
    $('.name').append('<span class="assetboard controller ' + colorName + '" ' +
        'data-assetboard="sky/paragraph">' +
        customerName[i] + '</span>');
}

function setHappyBirthday(letters) {
    var i,
        lettersLength = letters.length;

    for (i = lettersLength / -2; i < lettersLength / 2; i += 1) {
        $('main').append('<span class="assetboard controller letter' +
            letter + (i + lettersLength / 2) + '" ' +
            'data-assetboard="flower/letter" ' +
            'data-controller="assetboard/flower/letter">' + letters[i + lettersLength / 2] + '</span>');
    }
    setColorsToName();
}

function parabel(a, x, c, topPadding) {
    return a * (x * x - c) + topPadding;
}

function animateHappyBirthday(a, letters) {
    var i,
        lettersLength = letters.length,
        top,
        left,
        distance = 100 / (lettersLength + 2),
        offset = distance / 2,
        c = 50 - offset,
        topPadding = 3;

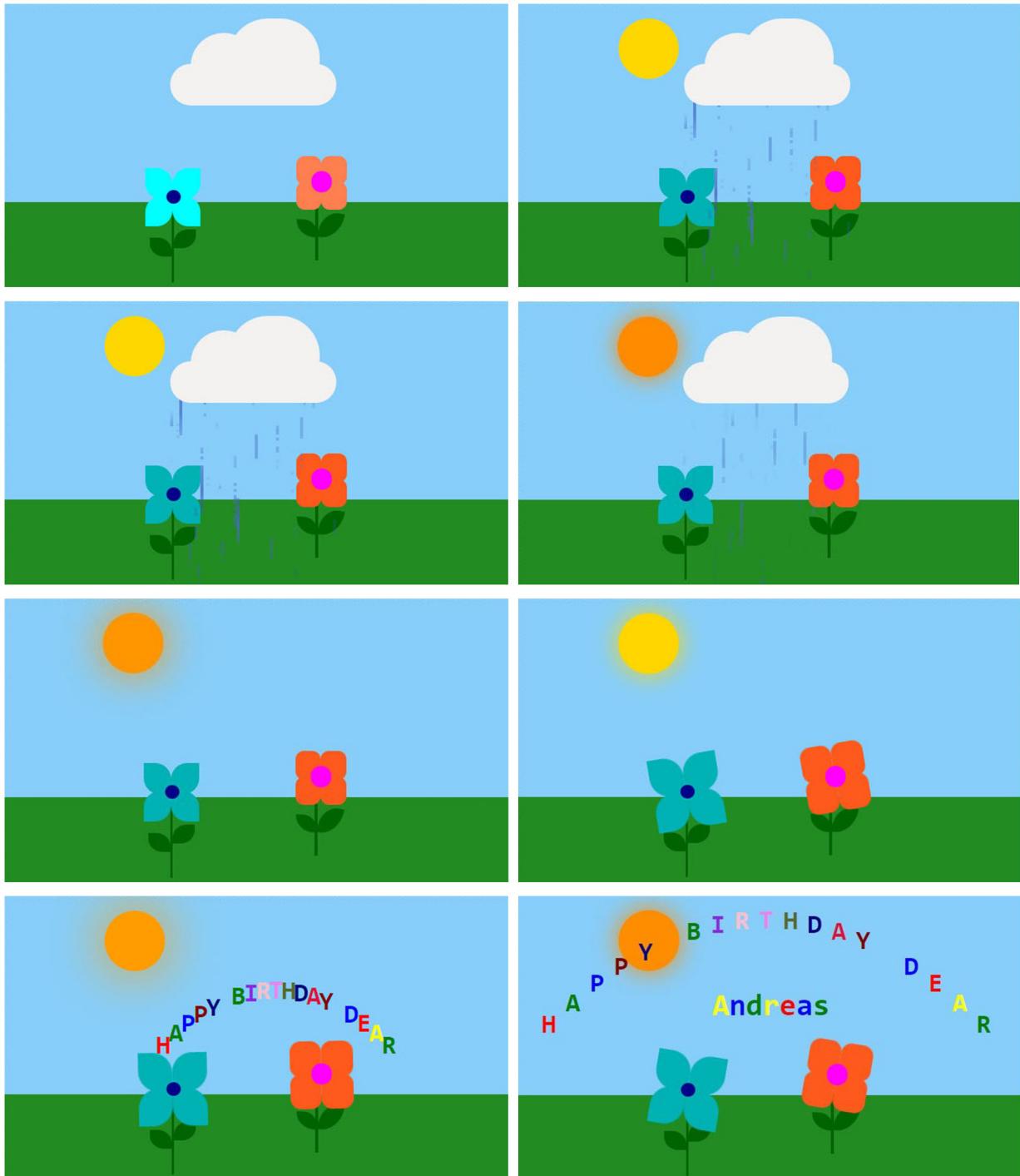
    for (i = lettersLength / -2; i < lettersLength / 2; i += 1) {
        left = distance * i + offset + c;
        top = parabel(a, left - c, c, topPadding);
        $('.letter' + (i + lettersLength / 2)).data({position: {left: left + '%', top: top + '%'}});
    }
}

function onResize() {
    var letters = ['H', 'A', 'P', 'P', 'Y', ' ', ' ',
        'B', 'I', 'R', 'T', 'H', 'D', 'A', 'Y', ' ', ' ',
        'D', 'E', 'A', 'R'];

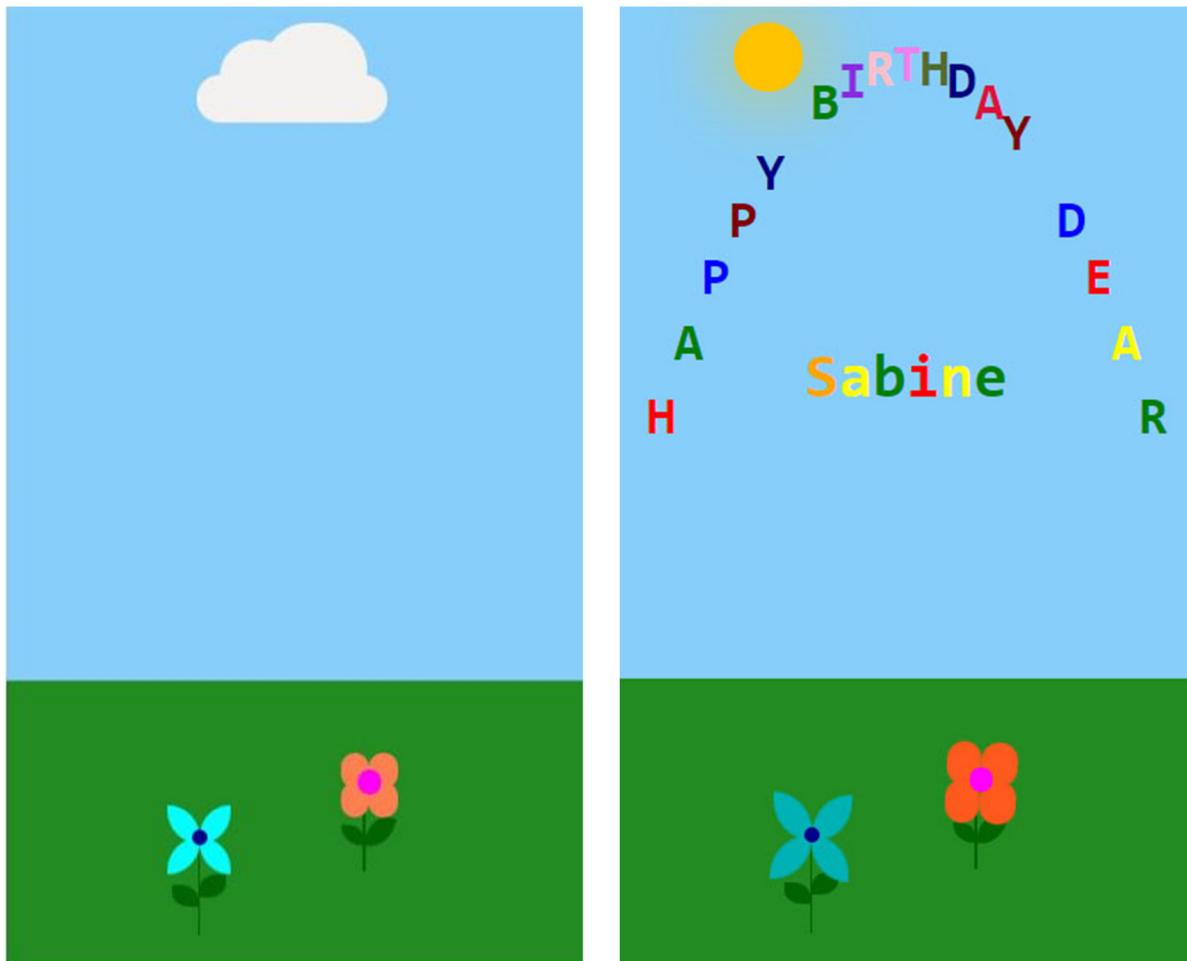
    setHappyBirthday(letters);
    parser.parse($('main'));
    animateHappyBirthday(0.02, letters);
}

});
```

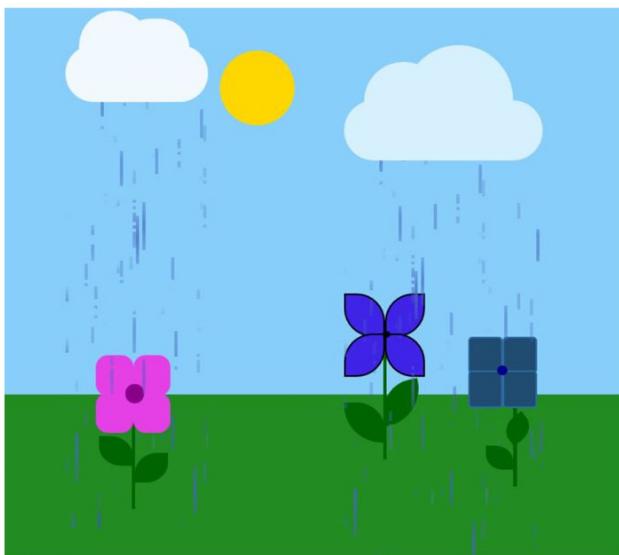
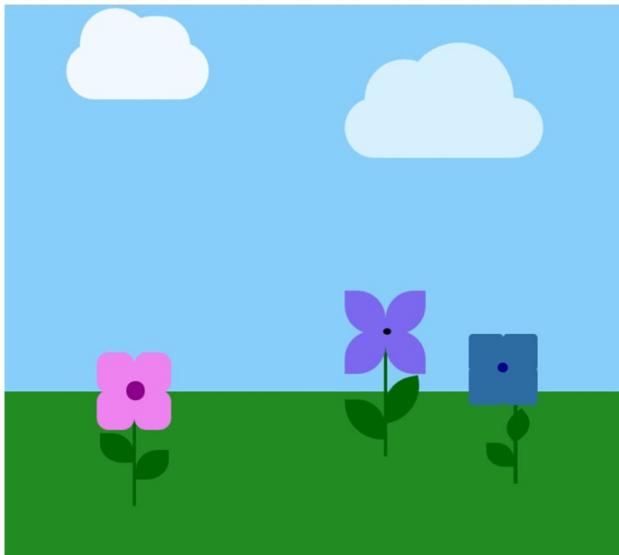
## 10. Animationsablauf für Mobile (Ausschnitt)



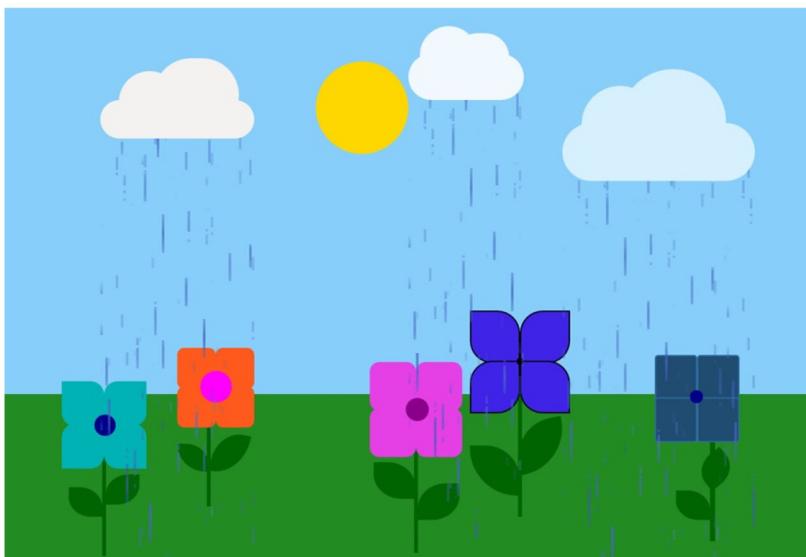
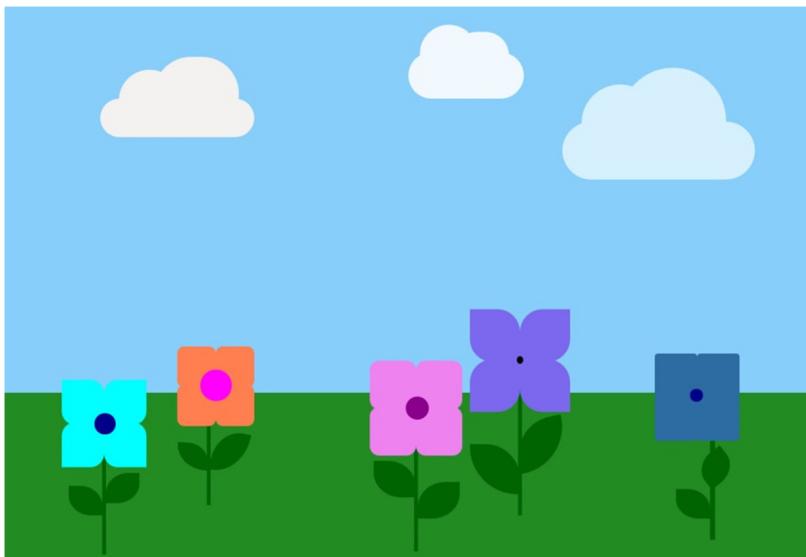
## 11. Animationsablauf für Mobile-Portrait (Ausschnitt)



## 12. Animationsablauf für Tablet (Ausschnitt)



### 13. Animationsablauf für Desktop (Ausschnitt)



## 10 Literaturverzeichnis/Quellenangabe

A product of The Eighth, L. (2014). <http://www.cssnewbie.com/>. Abgerufen am 07. 12 2015 von <http://www.cssnewbie.com/css-art-the-flower/#.VmUvXrjhDIW>

Basicuse. (2015). <http://basicuse.net/>. Abgerufen am 07. 12 2015 von [http://basicuse.net/articles/pl/textile/html\\_css/how\\_to\\_create\\_flashing\\_glowing\\_button\\_using\\_animation\\_in\\_css3](http://basicuse.net/articles/pl/textile/html_css/how_to_create_flashing_glowing_button_using_animation_in_css3)

Crockford, D. (2015). *Douglas Crockford's Wrrld Wide Web*. Abgerufen am 07. 12 2015 von <http://www.crockford.com/>

e.V., G. f. (kein Datum). <https://de.serlo.org/>. Abgerufen am 07. 12 2015 von <https://de.serlo.org/mathe/funktionen/wichtige-funktionstypen-ihre-eigenschaften/quadratische-funktionen-parabeln/parabeln/parabel>

Foundation, T. j. (2015). *jQuery*. Abgerufen am 07. 12 2015 von <https://jquery.com/>

Freemann, E. F. (2006). *HTML mit CSS & XHTML von Kopf bis Fuss*. Köln: O'Reilly.

<http://handlebarsjs.com/>. (2015). *handlebars*. Abgerufen am 07. 12 2015 von <http://handlebarsjs.com/>

Inc., W. F. (2015). <https://de.wikipedia.org/wiki/Wikipedia:Hauptseite>. Abgerufen am 07. 12 2015 von [https://de.wikipedia.org/wiki/Parabel\\_\(Mathematik\)](https://de.wikipedia.org/wiki/Parabel_(Mathematik))

Lehmann, D. (2015). *Quirks-Modus*. Abgerufen am 07. 12 2015 von <http://www.quirks-modus.de/>: <http://www.quirks-modus.de/archives/341>

Martin, R. C. (2009). *Clean Code*. z.B. Hamburg: mitp.

MKLab, C. (2014-2015). *StarUML2*. Abgerufen am 07. 12 2015 von <http://staruml.io/>

Robson, E. F. (2015). *JavaScript Programmierung von Kopf bis Fuss*. Köln: O'Reilly.

team, c. (2015). *Bootstrap*. Abgerufen am 07. 12 2015 von <http://getbootstrap.com/>

team, t. c. (2009). *Less*. Abgerufen am 07. 12 2015 von <http://lesscss.org/>

TheCodePlayer. (2014). *The Code Player*. Abgerufen am 06. 12 2015 von Make a simple Cloud in CSS: <http://thecodeplayer.com/walkthrough/make-a-simple-cloud-in-css3>

w3c. (2015). *w3c*. Abgerufen am 07. 12 2015 von <http://www.w3.org/>

[www.cssscript.com/](http://www.cssscript.com/). (01. 12 2013). <http://www.cssscript.com/>. Abgerufen am 06. 12 2015 von <http://www.cssscript.com/demo/create-rain-snow-falling-effects-using-css3-weather-css/#>

## 11 Glossar

<b>HK</b>	Handelskammer
<b>t8y</b>	t8y.com GmbH
<b>HTML/HTML5</b>	Hypertext Markup Language
<b>CSS/CSS3</b>	Cascading Style Sheets
<b>GF</b>	Geschäftsführung
<b>URL</b>	Uniform Resource Locator
<b>DDB</b>	Doyle Dane Bernbach Group GmbH
<b>Less</b>	Dynamische Stylesheet Sprache
<b>SASS</b>	Syntactically Awesome Stylesheets
<b>MVC</b>	Model View Controller
<b>GUI</b>	Graphical User Interface
<b>JSON</b>	JavaScript Object Notation
<b>W3C</b>	World Wide Web Consortium
<b>QS</b>	Qualitätssicherung