



**HAMBURG-BUSTARDS *RELOADED***

Konzept von

Florian Stammerjohann

WDDp 1108



## **Inhaltsverzeichnis**

<b>1 Projektbeschreibung.....</b>	<b>3</b>
<b>2 Zielsetzung &amp; Zielgruppendefinition.....</b>	<b>3</b>
2.1 Zielgruppe.....	3
2.2 Features.....	4
2.2.1 Basis.....	4
2.2.2 Erweiterungen.....	5
2.3 Zeitplan.....	5
<b>3 Design.....</b>	<b>6</b>
3.1 Typographie & Farbgebung.....	6
3.2 Logo.....	6
3.3 Layout der Internetpräsenz.....	7
3.3.1 Head- und Footer-Bereich.....	7
3.3.2 Team-Ansicht.....	8
3.3.3 Spielerprofil-Ansicht.....	9
3.3.4 News-, Kontakt-, Supporter- und Impressums-Ansicht.....	9
3.3.5 Administration.....	10
3.4 Design des MiniPBProfile-Signatur-Mashups.....	10
<b>4 Technik / Programmierung.....</b>	<b>12</b>
4.1 Selbstentwickeltes CMS mit CakePHP.....	12
4.1.1 Model-View-Controller (MVC).....	12
4.1.2 Sicherheit.....	13
4.1.3 Schnittstellenprogrammierung.....	13
4.2 Informationsbeschaffung.....	13
4.2.1 Das CakePHP-Cookbook.....	13
4.2.2 jQuery.....	13
4.2.3 Flickr-API.....	14
4.3 Datenbankdiagramm.....	14



## 1 Projektbeschreibung

Das Projekt *Hamburg-Bustards Reloaded* befasst sich mit der Neugestaltung der Internetpräsenz Hamburg-Bustards.de, einer Website eines Hamburger Paintballteams. Die Neugestaltung umfasst eine Aktualisierung des Designs, sowie einer kompletten CMS-Neuentwicklung.

## 2 Zielsetzung & Zielgruppendefinition

Eine klare Definition von Basis-Features und möglichen Erweiterungen, hilft der Zeit- und Kosten/Nutzen-Einschätzung. Die Zielgruppendefinition vor allem bei der Erstellung des Layouts und damit der Usability.

### 2.1 Zielgruppe

Die Internetpräsenz richtet sich vorrangig an Paintballspieler. Die Zielgruppe könnte jedoch durch zusätzliche Features\* wie das „Paintballinfopaket“ vergrößert werden. Diese Erweiterung enthält z.B. eine Einführung bzw. eine Auflistung verschiedenen Spielarten, einem Einsteigerguide und Videos.

---

\* Erweiterungen sind mit zusätzlichen Kosten verbunden



## 2.2 Features

Wie eingangs erwähnt, bietet das CMS einige Basis-Features und Erweiterungsmöglichkeiten. Alle Basis-Features werden bis zur Deadline fertiggestellt, sind also Bestandteil des Projekts und somit in den Kosten enthalten.

### 2.2.1 Basis

Die folgenden Features sind in Auftrag gegeben worden:

- News-Seite inkl. Modul zur Verwaltung der einzelnen Beiträge
- Kontakt-Seite inkl. Modul zur Verarbeitung der Kontaktnachrichten
- Team/Events-Seite inkl. Administrationsmöglichkeiten
- Spielerprofil-Seite mit umfangreichen Informationen (z.B. Kontaktmöglichkeiten, gespielte Events und Bilder) inkl. Administrationsmöglichkeiten
- Spielerprofil-Schnittstelle (um extern Spielerdaten auszulesen)
- Modul zur Anzeige der Supporter und Sponsoren
- MiniPBProfile-Signatur-Mashup: Für das vorhandene HH-Bustards-Forum<sup>1</sup> (Forensoftware phpBB<sup>2</sup>) etc. (zeigt Spielerdaten und Events (Spielerprofil-Schnittstelle), sowie Fotos (werden dynamisch aus der HH-Bustards-Gruppe von Flickr geladen<sup>3</sup>).
- SEO: Erreicht wird dies, durch wohlformatierte statische Adressen (mittels mod\_rewrite), Einhaltung der (x)HTML-Standards und strikter Trennung von Inhalt und Design (CSS).
- Sicherheit: Durch automatische Filterfunktion von Formulardaten und das Absichern von Passwörter mittels *Salting*<sup>4</sup>, wird ein hoher Grad an Sicherheit erreicht.

---

1 Link: <http://forum.hamburg-bustards.de>

2 Link: <http://www.phpbb.com> bzw. <https://www.phpbb.de>

3 Link: <http://www.flickr.com/groups/hamburg-bustards>

4 Link: [http://de.wikipedia.org/wiki/Salt\\_%28Kryptologie%29](http://de.wikipedia.org/wiki/Salt_%28Kryptologie%29)



## 2.2.2 Erweiterungen

Folgend werden alle Erweiterungen aufgelistet die einen Mehrwert der Internetpräsenz darstellen könnten, jedoch nicht in diesem Projekt mit inbegriffen sind.

- Paintballinfopaket inkl. Administrationsmöglichkeiten für die Verwaltung eines Einsteigerguides und Videos
- Terminkalender (mit An- und Abmeldemöglichkeiten zu Trainingseinheiten und Turnieren)
- Teamkasse inkl. Administrationsmodul (Kassen-Admin)
- Dashboard für jeden Benutzer mit vielen verschiedenen Widgets (u.a. News von Lieblingsteams via Twitter, den letzten Forenbeiträgen, Kassenübersicht für Spieler und Einladungen zu anstehenden Events).

## 2.3 Zeitplan

Die Umsetzungstermine richten sich nach dem Arbeitsaufwand und sind dem Zeitplan zu entnehmen.

	Juli				August				September				Oktober			
<b>Kalenderwoche</b>	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Projektbesprechung																
Konzepterstellung																
<b>Design:</b>																
- Haupt-Layouts (grob)																
- Designentwicklung																
- Fertigstellung Design																
<b>Technik / Programmierung:</b>																
- Datenbank (Prototyp & Final)																
- MVC-Prototyp																
- PMiniProfile-Feature-Entwicklung																
- Entwicklung -> Models																
- Entwicklung -> Controller																
- Entwicklung -> Views																
Testphase & Debugging																
Präsentation																



## 3 Design

### 3.1 Typographie & Farbgebung

Die Farbgebung wird sich an der aktuellen Internetpräsenz orientieren (bedeutet Schwarz, Weiß, Grautöne und Grün) und eventuell, jedoch nur nach Absprache mit dem Kunden, eine weitere Farbe.

Eine gute Lesbarkeit am Bildschirm wird durch Verwendung einer serifenlosen Schriftart, wie Verdana, Arial und/oder Helvetica erreicht. Dabei wird eine einheitliche Schriftgröße von 12px verwendet, Headlines werden dementsprechend formatiert bis zu einer max. Schriftgröße von 18px\*.

### 3.2 Logo

Einer Überarbeitung oder Neugestaltung des Teamlogos ist nicht gewünscht, kann jedoch als Zusatzleistung im Nachhinein in Auftrag gegeben werden.

### 3.3 Layout der Internetpräsenz

Das Layout hat eine Breite von 950px und einer variablen Höhe, die sich dem Inhalt anpasst, wobei die Seite im Browser zentriert wird und somit auch bei höheren Bildschirmauflösungen ein harmonisches Erscheinungsbild gibt. Dieses lässt sich in drei Teile zerlegen: **Header**, **Content** und **Footer**. Die folgend aufgeführten Layouts der Hauptelemente könnten leichte Abweichungen zur Finalen Version aufweisen, die in der 33 KW dem Kunden vorgelegt und anschließend finalisiert wird\*\*.

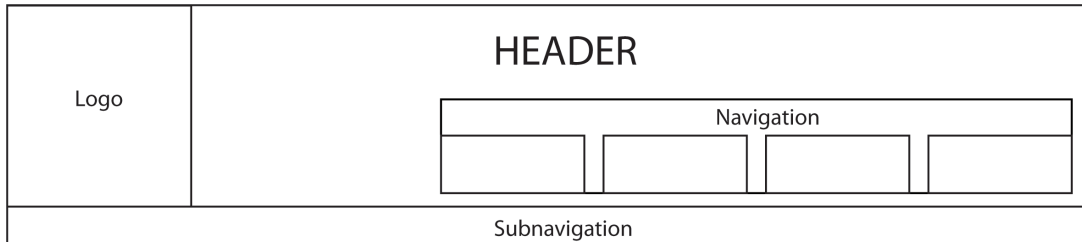
---

\* Änderungen dieser Werte sind bis einschließlich KW 32 möglich

\*\* Bis zu der genannten Kalenderwoche können weitere Vorschläge des Kunden, in den Designprozess aufgenommen werden



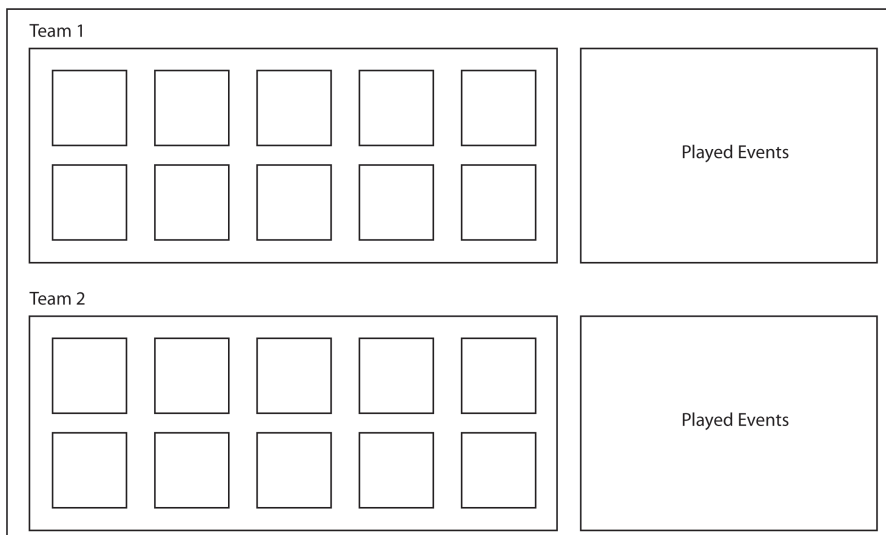
### 3.3.1 Head- und Footer-Bereich



Der **Header** beinhaltet drei bzw. vier Elemente. Auf der linken Seite wird das **Teamlogo** positioniert, im danebenliegenden Hauptbereich die **Navigation**, das **Login-Modul** und darunter die **Subnavigation**. *Über eine grobe Positionierung des Login-Moduls, muss bis einschließlich KW 31 eine Einigung erreicht werden.*

Der **Footer** befindet sich untern dem **Content-Bereich** und beinhaltet das **Copyright** und ein Link zum **Impressum**.

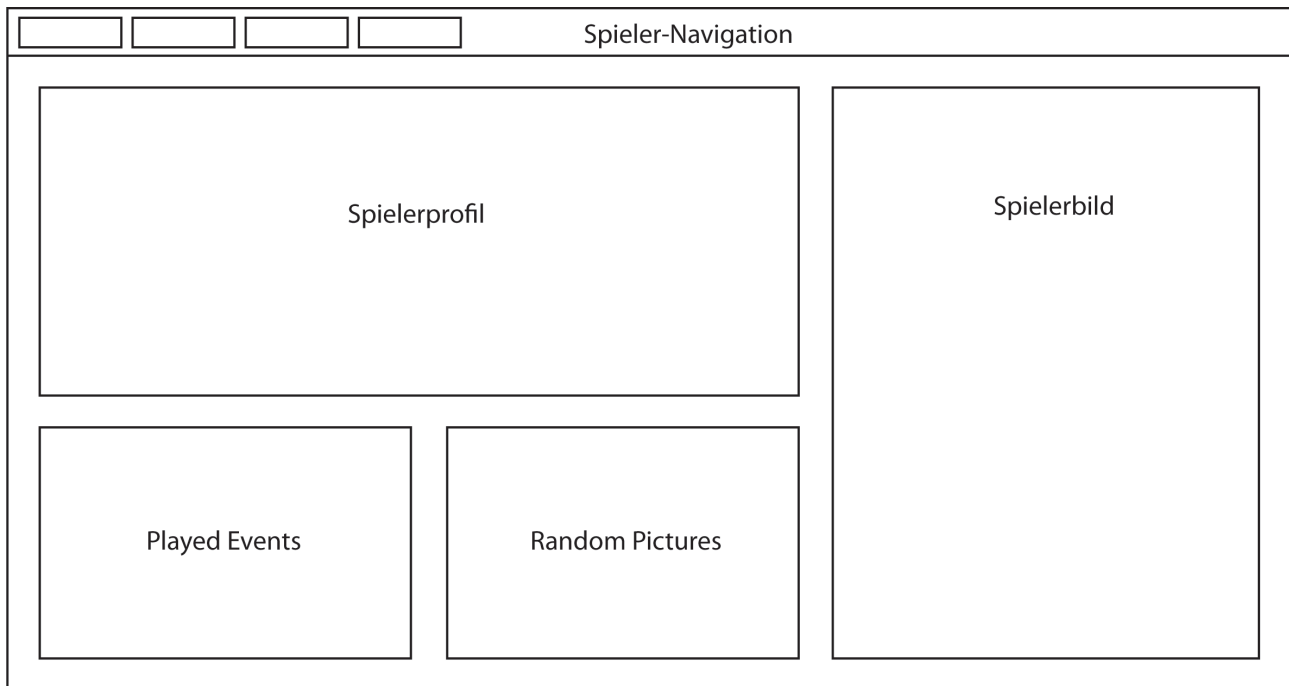
### 3.3.2 Team-Ansicht



Die Team-Ansicht listet Fun- und Ligateam der Hamburg-Bustards mit deren gespielten Events (nach Datum sortiert) auf. Jeder Spieler wird seinem Team zugeordnet, mit Player-Avatar und Namen angezeigt. Die Administration erfolgt auf einer Seite im Backend.



### 3.3.3 Spielerprofil-Ansicht



Die umfangreichste Unterseite des Projekts ist die Spielerprofil-Ansicht. Sie enthält im oberen Teil eine Navigation, mit der man schnell jeden Spieler, egal ob aus dem Fun- oder Ligateam, erreichen kann. Der größte Teil des Spielerprofils listet kurz und knapp, Fakten und Kontaktmöglichkeiten des Spielers auf. Zusätzlich wird ein großes Bild auf der rechten Seite angezeigt, sowie die zuletzt gespielten Events und einige Bilder. Letztere werden genau wie bei dem MiniPBProfile-Signatur-Mashup-Feature, dynamisch aus der HH-Bustards-Gruppe von Flickr geladen. Aus diesem Grund entfällt eine Verwaltung der Bilder, da diese automatisch von den Benutzern der Flickr-Gruppe gemacht werden. Einzelheiten zur Technik finden sich unter Punkt 4, *Technik / Programmierung*.

### 3.3.4 News-, Kontakt-, Supporter- und Impressums-Ansicht

Diese vier Unterseiten werden alle ein Mehrspaltiges Layout enthalten und direkt im Frontend bearbeitet werden können, näheres dazu im Unterpunkt *Administration*.





Einzig die News-Ansicht wird sich über zwei Unterseiten erstrecken. Auf der Eingangsseite werden auf der linken Seite nur die Headlines der letzten fünf News und die ersten zwei Zeilen des dazugehörigen Beitrags (mit einem Link zu dem vollständigen Beitrag) angezeigt. Auf der rechten Seite werden Headlines von Artikel (inkl. Verweise) aus dem Paintball-Portal<sup>5</sup> aufgelistet, für eine engere Bindung an die Paintball-Szene. Die vollständigen Beiträge, sowie eine Auflistung von älteren Beiträgen, werden auf einer weiteren Unterseite aufgeführt.

### **3.3.5 Administration**

Die Verwaltung der meisten Seiten wird im Frontend vorgenommen. Dabei wird zuerst geprüft, ob der Benutzer über die erforderlichen Administrations-Rechte verfügt und ihm, im Falle einer positiven Überprüfung, eine leicht modifizierte Seite ausgegeben. Die Modifikation enthält kleine „Editier-Icons“, über die ein Eingabeformular aufgerufen wird und das beim Betätigen des Senden-Buttons den Inhalt (per AJAX also ohne Page-Reload) an den Server schicken soll.

Nur die Verwaltung der Benutzer, Teams und Events, werden in einem kleinen Backend-Bereich administriert.

Durch die vorrangige Verwaltung im Frontend, wird die Ansicht der Administration vereinfacht und somit die Usability gesteigert.

### **3.4 Design des MiniPBProfile-Signatur-Mashups**

Das Mashup soll Informationen und Bilder von einem Spieler des Paintballteams anzeigen. Informationen wie Name, Trikotnummer, gespielte Events und ein Foto des Spielers, sollen über die Schnittstelle des Hauptprojekt bezogen werden. Fotos z.B. von Turnieren o.Ä., auf dem der Spieler zu sehen ist, sind wie bei der „Spielerprofil-Ansicht“ von Flickr zu beziehen. Hinzu kommt, dass die Anwendung als Signatur in einem Forum eingebunden

---

<sup>5</sup> Link: <http://www.pbportal.de>



wird. Sie muss also klein genug sein, um den Leser des Forums nicht von dem eigentlichen Inhalt abzulenken. Zusätzlich dürfen keine Fehler auftreten, falls der Forenbesucher JavaScript ausgeschaltet- oder gar nicht hat. Für diesen Fall ist auch davon abzuraten dem Client unnützes HTML zu übermitteln.

Die folgende Bilder zeigen Layout und Funktion, des zu erstellenden Mashups.

**Profil von stammflo**

- Name: FLORIAN
- Alter: 22
- Team: LIGA
- Position: MID, FRONT
- Beruf: WEB APPLICATION DEVELOPER
- Hobby: WEBDESIGN, POKER

Nr.: #42

Last 6 Events Total: 9

Random Pictures

**Profil von stammflo**

Last 6 Events Total: 9

30.06.10	XPSL Region Nord 2. Spieltag
26.05.10	XPSL Region Nord 1. Spieltag
25.02.10	ProSchool mit den Ducks
20.02.10	Winter-Fun-Liga Nord 4. Spieltag
08.01.10	Winter-Fun-Liga Nord 3. Spieltag
15.12.09	Winter-Fun-Liga Nord 2. Spieltag

Nr.: #42

Random Pictures

**Profil von stammflo**

- Nam
- Alter
- Team
- Posit
- Beru
- Hobl

Nr.: #42

Last 6 Events

Random Pictures



## 4 Technik / Programmierung

Für die Umsetzung der Internetpräsenz wird ein CMS entwickelt, das so perfekt an die Bedürfnisse des Kunden angepasst werden kann. Durch die strikte Einhaltung von (x)HTML-Standards und die Verwendung von CSS, wird die Internetpräsenz auf allen aktuellen Browsern gleich aussehen\* und ein gutes Ergebnis in Suchmaschinen erzielen können.

### 4.1 Selbstentwickeltes CMS mit CakePHP

Durch die Entwicklung mit **CakePHP**<sup>6</sup>, einem OpenSource-Framework für PHP, wird die Entwicklungszeit verkürzt und dadurch die Kosten niedrig gehalten, ohne dabei an Sicherheit oder Komfort einzubüßen. Wie der Name schon verrät, wird hauptsächlich PHP als Programmiersprache verwendet, allerdings kommt bei der AJAX-Kommunikation das JavaScript-Framework **jQuery**<sup>7</sup> zum Einsatz.

#### 4.1.1 Model-View-Controller (MVC)

Beim Modell-View-Controller handelt es sich um ein Set von Entwurfsmustern, die eingesetzt werden, um die einzelnen Anwendungsschichten voneinander zu trennen. Die Schichten werden wie folgt voneinander getrennt:

- Der *View* stellt die Daten des *Models* dar. Er kann die Daten beim *Model* abfragen.
- Das *Model* speichert die Anwendungsdaten und kann den *View* über Änderungen der Daten informieren.
- Der *Controller* nimmt die Aktionen des Benutzers an und fordert daraufhin *Model* oder *View* auf, den Zustand zu ändern.

Das CakePHP-Framework benutzt zusätzlich noch einen sog. Dispatcher, der die angeforderte URL überprüft und an den zuständigen Controller weiterleitet.

---

\* Minimale Pixelverschiebungen sind jedoch nicht 100% auszuschließen

<sup>6</sup> Link: <http://cakephp.org>

<sup>7</sup> Link: <http://jquery.com>



### 4.1.2 Sicherheit

Jedes Passwort wird verschlüsselt durch einen sog. Hash in der Datenbank abgelegt. Eine zusätzliche Sicherheitsstufe wird erreicht, durch das hinzufügen einer beliebigen Zeichenkette an das Passwort, vor dem Verschlüsselungsvorgang.

Daten die über ein Formular an den Server geschickt werden, werden einer automatischen Validierung unterzogen, um „SQL-Injections“, „cross-site scripting“ u.ä. zu verhindern.

E-Mail-, Cookie-, Session- und Request Handling- Komponenten runden das Paket ab.

### 4.1.3 Schnittstellenprogrammierung

Es soll eine Schnittstelle geschaffen werden, um alle Spielerdaten auszulesen. Eine vollständige Schnittstelle mit Lese- und Schreibrechten ist nicht gefordert. Die Antwort des Servers soll JSON-Daten zurückliefern, was am einfachsten mit der RequestHandler-Komponente<sup>8</sup> von CakePHP zu realisieren ist.

## 4.2 Informationsbeschaffung

Alle Informationen zu den verwendeten Frameworks bzw. APIs die für die Umsetzung dieses Projekts nötig sind, können den jeweiligen offiziellen Projektseiten entnommen werden.

### 4.2.1 Das CakePHP-Cookbook

Das Cookbook für CakePHP ist unter dem Link <http://book.cakephp.org/de/toc/876/The-Manual> zu erreichen. Es enthält u.a. auch ein paar gute Tutorials.

### 4.2.2 jQuery

Die komplette API-Referenz mit vielen nützlichen offiziellen- und Community basierten Beispielen finden sich unter [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page)

<sup>8</sup> Link: <http://book.cakephp.org/de/view/1295/Responding-To-Requests>



### 4.2.3 Flickr-API

Die Übersicht über alle Methoden sind unter <http://www.flickr.com/services/api/> aufgeführt. Die Methode zur Datengewinnung der Bilder aus der Flickr Gruppe des Teams, gibt es unter folgendem Link <http://www.flickr.com/services/api/flickr.groups.pools.getPhotos.html>

### 4.3 Datenbankdiagramm

Alle Daten der Website werden in einer MySQL-Datenbank abgelegt. Um die Generierung der Datenbankstruktur zu erleichtern, wurde der folgende Prototyp erstellt.

