

# **Tradingfun**

## **Webapplikation als Plattform für den Tausch von Stickern und Sammelkarten**

**Softwarepraktikum WS13/14 - SS14**

**Dozent: Prof. Dr. Lütticke**

Daniel Miguel Baltes Amado, Felix Rogge

26.06.2014

# Inhaltsverzeichnis

<b>1 Benutzerhandbuch</b>	<b>1</b>
1.1 Einleitung	1
1.2 Auf einen Blick	2
1.3 Registrierung	2
1.4 An- und Abmeldung	3
1.5 Die Oberfläche	4
1.6 Das Benutzerprofil	4
1.7 Bestände anlegen und verändern	5
1.8 Der Marktplatz	5
1.9 Versendete und erhaltene Angebote	7
1.10 Einen Benutzer nach dem Tausch bewerten	9
1.11 Die Autorenoberfläche	10
<b>2 Technische Dokumentation</b>	<b>13</b>
2.1 Einleitung	13
2.2 Spezifikationen	13
2.2.1 Hardware	13
2.2.2 Software	13
2.3 Architektur	14
2.3.1 Frontendserver: nginx	14
2.3.2 Datenbank: MySQL	16
2.3.3 Application Server: Node.JS	17
2.3.4 Templatesystem: Jade	18
2.3.5 Frontend: HTML5 / JavaScript	18
2.4 Performance / Load Test	19

# **1 Benutzerhandbuch**

## **1.1 Einleitung**

Tradingfun ist eine Webplattform die ihren Benutzern die Möglichkeit liefert Bestände von Sammelkarten und Stickern zu verwalten und einen Tausch einzelner und mehrerer Karten zu realisieren. Die angelegten Sammelkarten werden kategorisch in Alben aufgeführt. Ein Tausch kann, sowohl von Album zu Album, als auch Alben übergreifend vollzogen werden. Bei einem Tausch muss nicht zwingend dieselbe Anzahl von Karten getauscht werden. Nach Abschluss eines erfolgreichen Tauschs können Benutzer sich zudem gegenseitig bewerten. Die ergonomische und intuitive Benutzeroberfläche gewährleistet eine einfache Handhabung. Es folgen Beschreibungen über die Bedienungsabläufe von Tradingfun.

## 1.2 Auf einen Blick



**Abbildung 1**

1. "Admin": Autorenbereich zum Anlegen und Löschen von Alben und Karten und für eine Übersicht über angelegte Benutzer
2. "Mein Profil" zur Übersicht des eigenen Benutzerprofils, sowie der Bewertungen
3. "Meine Sammlung" zur Verwaltung der Kartenbestände
4. "Angebote" zur Übersicht von versendeten und erhaltenen Angeboten
5. "Marktplatz" zur Findung von geeigneten Tauschpartnern
6. "Abmelden" zur Abmeldung aus Tradingfun

## 1.3 Registrierung

Für die Nutzung von Tradingfun ist ein Benutzerprofil zwingend notwendig. Über die Schaltfläche „Registrierung“ gelangt man zur Registrierungsmaske, die es dem Benutzer ermöglicht ein Benutzerprofil unter Angabe eines Benutzernamens, einer gültigen Emailadresse und eines Passwortes zu erstellen.

**Registrierung**

Die Registrierung und Nutzung dieser Plattform ist kostenlos. Nachdem Sie das unten stehende Formular ausgefüllt und abgesendet haben, erhalten Sie eine Bestätigung per E-Mail.

Benutzername

E-Mail-Adresse

Passwort

Passwort bestätigen

Registrieren

**Abbildung 2**

Ist der ausgewählte Benutzername noch verfügbar und die angegebene Emailadresse gültig, so erhält man nach abgeschlossener Registrierung eine Bestätigungsmail, die besagt, dass der angelegte Account nun aktiv ist. Einzelne Emailprovider könnten unter Umständen die Bestätigungsmail als Spam kennzeichnen. Eine Überprüfung des Spamordners ist ggf. notwendig.

## 1.4 An- und Abmeldung

Sofern der Benutzer über einen gültigen Tradingfun Account verfügt (siehe Kapitel 3), hat er die Möglichkeit sich über die Schaltfläche „Anmelden“, durch Angabe seines Benutzernamens und des dazugehörigen Passwortes, anzumelden.

Ist der Benutzer am System angemeldet, so kann er durch Betätigung der Schaltfläche „Abmelden“ jederzeit eine Abmeldung aus Tradingfun durchführen.

**Anmeldung**

felix

.....

Anmelden

**Abbildung 3**

## 1.5 Die Oberfläche

Hat der Benutzer sich erfolgreich angemeldet, so erwartet ihn zunächst die Startseite. Hier erscheint ein Begrüßungstext mit einer kurzen erläuternden Beschreibung der Funktionalitäten von Tradingfun. Des Weiteren erhält der angemeldete Benutzer im internen Bereich Zugang zur Navigationsleiste, über die er die angebotenen Funktionalitäten erreichen kann.



Abbildung 4

## 1.6 Das Benutzerprofil

Über den Menüeintrag „Mein Profil“ gelangt der Bediener auf seine persönliche Profilseite. Hier erwartet ihn, neben seinem persönlichen Profilbild, das Datum seiner Registrierung und Bewertungen, die er durch Tauschaktionen mit anderen Benutzern erhalten hat (siehe Kapitel 10). Es wird sowohl die Anzahl, als auch eine prozentuale Angabe über den Ausfall der Bewertungen angegeben. Des Weiteren werden Kommentare anderer Be-



Abbildung 5

nutzer bezüglich ihrer abgegebenen Bewertungen auf der Profilseite aufgeführt.

## 1.7 Bestände anlegen und verändern

Das Anlegen und die Bearbeitung der eigenen Bestände erfolgt durch die Schaltfläche „Meine Sammlung“. Hier hat der Benutzer die Möglichkeit die in seinem Besitz stehenden Karten und Alben in seinen virtuellen Bestand, unter Angabe der Anzahl der Exemplare, aufzunehmen. Über den Button „Bearbeiten“ aktiviert der Benutzer den Editiermodus um Karten und Alben dem Bestand hinzuzufügen. Um die Menge der Exemplare anzugeben, ist die Verwendung der Schaltflächen „+/-“ vorgesehen.

Meine Sammlung		Bearbeiten
Road to FIFA World Cup Brazil 2014		
1	Sergio Romero-Argentina	- +
2	Pablo Zabaleta-Argentina	- +
0	Ezequiel Garay-Argentina	- +
1	Nicolas Burdisso-Argentina	- +
0	Javier Mascherano-Argentina	- +
0	Angel Di Maria-Argentina	- +
2	Fernando Gago-Argentina	- +

### Abbildung 6

Im Besitz stehende Karten werden im Bearbeitungsmodus schwarz, fehlende Karten hellgrau dargestellt.

## 1.8 Der Marktplatz

Der Marktplatz dient zum Finden von potentiellen Tauschpartnern. Es wird dem Bediener eine Auflistung der in Frage kommenden Tauschpartner kategorisch aufgelistet. Ein Tauschpartner kommt in Frage, sofern sich Karten in seinem Bestand befinden die dem Bediener fehlen.

Marktplatz	
Benutzer	Karten
daniel	7

### Abbildung 7

Entscheidet man sich dafür eine Tauschanfrage an einen Benutzer zu versenden, so wählt man diesen aus und gelangt zur Übersicht der verfügbaren Karten des ausgewählten Tauschpartners.

Sammlung von daniel		
Road to FIFA World Cup Brazil 2014		
7	Sergio Romero-Argentina	Angebot
0	Pablo Zabaleta-Argentina	
0	Ezequiel Garay-Argentina	
1	Nicolas Burdisso-Argentina	Angebot

### Abbildung 8

In dieser Übersicht hat man nun die Gelegenheit ein Angebot zu der gewünschten Karte zu versenden. Hierfür wählt man die Karte aus und betätigt die Schaltfläche „Angebot“. Der Bediener kann nun eine Auswahl treffen welche Karten er seinem Tauschpartner im Gegenzug anbieten möchte.



Angebot an daniel

daniel	Ich
Road to FIFA World Cup Brazil 2014	Road to FIFA World Cup Brazil 2014
Tausch / Bestand	Tausch / Bestand
0 / 7 Sergio Romero-Argentina	0 / 2 Pablo Zabaleta-Argentina
0 / 1 Nicolas Burdisso-Argentina	0 / 1 Nicolas Burdisso-Argentina
	0 / 4 Javier Mascherano-Argentina

Angebot senden

**Abbildung 9**

Um zu definieren welche Karten er dem Tauschpartner anbieten möchte, muss der Benutzer den Mauscursor in die Textbox der entsprechenden Karte bewegen. Es erscheinen Pfeile, über die die Anzahl der Exemplare für die ausgewählte Karte, bestimmt werden kann. Dieser Vorgang lässt sich für beliebig viele, im Besitz stehende Karten, wiederholen. Ist die Auswahl komplett, so wird das Angebot über die Schaltfläche „Angebot versenden“ abgesendet.



**Abbildung 10**

## 1.9 Versendete und erhaltene Angebote

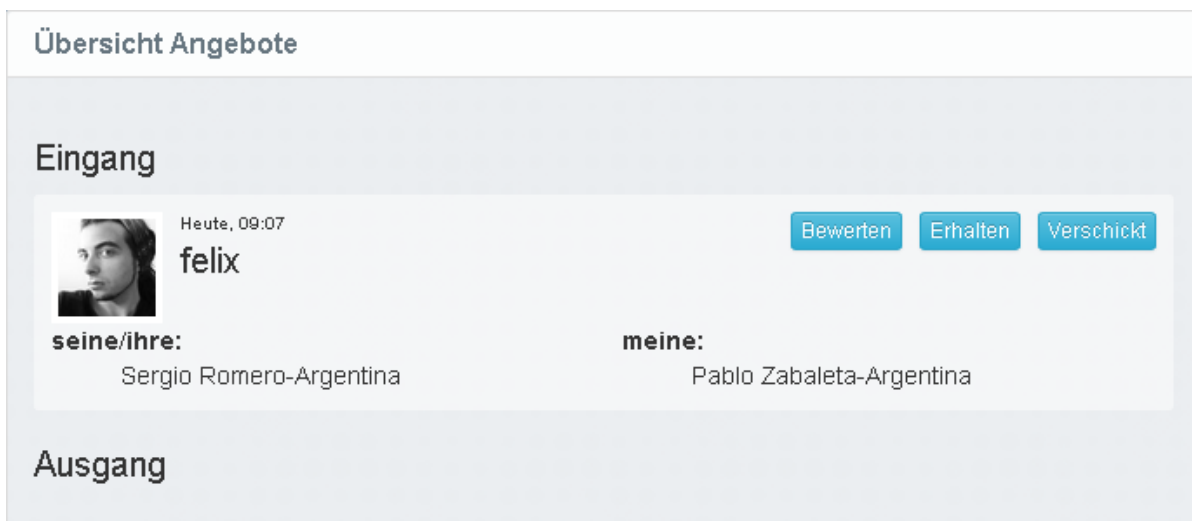
Über die Kategorie Angebote erhält der Bediener eine Übersicht über erhaltene und versendete Angebote. Wurde ein Angebot an einen anderen Benutzer versendet, so befindet sich das erhaltene Angebot mit Angaben zu den angefragten und angebotenen Karten in der Sektion „Eingang“. Versendet der Bediener selbst ein Angebot an einen potentiellen Tauschpartner, so befindet sich seine Anfrage in der Sektion „Ausgang“. Über die Schaltflächen „Ablehnen“ und „Akzeptieren“ kann das Angebot nun bestätigt oder abgelehnt werden



**Abbildung 11**

Wurde das Angebot akzeptiert, wird dies mit einer Erfolgsmeldung, dass die Tauschanfrage angenommen wurde, quittiert. Analog dazu verhält sich die Betätigung der Schaltfläche „Ablehnen“.

Nach dem Akzeptieren der Anfrage erhält der Bediener neue Schaltflächen die Angaben zum Status des Tausches geben sollen. Hierfür hat er die Schaltfläche „Erhalten“, die Auskunft darüber geben soll ob er die gewünschte Karte von seinem Tauschpartner bereits erhalten hat sowie die Schaltfläche „Verschickt“, die dem Tauschpartner signalisiert, dass seine zu tauschende Karte sich bereits auf dem Weg zum Tauschpartner befindet.



**Abbildung 12**

Die Betätigung dieser Schaltflächen wird ebenfalls mit entsprechenden Erfolgsmeldungen quittiert.

Hat der Benutzer seine Karten verschickt und die Karten seines Tauschpartners erhalten, so kann er über die Schaltfläche „Bewerten“ eine Rezension über seinen Tauschpartner abgeben um zukünftigen Interessenten einen Eindruck über die Se-

riosität des Benutzers zu geben (siehe 1.10).

## 1.10 Einen Benutzer nach dem Tausch bewerten

Sofern der Benutzer den Tausch bestätigt hat und die Angabe über den Erhalt der Karte betätigt hat (siehe Kapitel 9), hat er nun Gelegenheit den Tauschpartner zu bewerten. Durch Betätigung der Schaltfläche „Bewerten“, öffnet sich ein Popup, das ein Formular beinhaltet über das er Bewertungsangaben tätigen kann.



The screenshot shows a web form titled "Ihre Bewertung für felix" with a close button (X) in the top right corner. Below the title, there are three buttons: "Negativ" (red), "Neutral" (grey), and "Positiv" (green). Underneath these buttons is a large text input field with the placeholder text "Ein paar Worte zum Tausch mit felix...". At the bottom right of the form, there are two buttons: "Später" (grey) and "Bewertung abgeben" (blue).

### Abbildung 13

Zunächst einmal kann eine Angabe darüber gemacht werden, ob der Benutzer den Tausch als „Negativ“, „Neutral“ oder „Positiv“ empfunden hat. Kriterien wie ein schneller Versand oder Zustand der Karte können hier Kriterien für die Bewertung sein. Des Weiteren ist es möglich den Grund der Bewertung näher zu spezifizieren. Hierfür trägt der Bediener ein paar erklärende Worte zum Tausch in das dafür vorgesehene Textfeld ein. Ist die Bewertung komplett, so wird durch Betätigung der Schaltfläche „Bewertung abgeben“ die Bewertung des Tauschpartners abgeschickt. Eine abgegebene Bewertung wird daraufhin im Benutzerprofil des Tauschpartners aufgeführt (siehe Kapitel 6). Möchte der Benutzer mit der Bewertung noch etwas warten, so kann er durch Nutzung der Schaltfläche „Später“, die Bewertung zu einem späteren Zeitpunkt durchführen.

## 1.11 Die Autorenoberfläche

Zweck der Autorenoberfläche ist die Realisierung des Hinzufügens von Alben und Karten über einen gesonderten Autorenaccount. Hat ein Benutzer besagte Autorenrechte so sieht er, anders als ein normaler Benutzer, in der Navigationsleiste zusätzlich die Option „Admin“, über die sich ein Dropdownmenü öffnet aus dem er die Bereiche „Bestände“ und „Benutzer“ auswählen kann. Wählt der Autor die Kategorie „Bestände“ aus, so gelangt er in eine Übersicht, aus der er zunächst ein Album, aus der Combobox, auswählen kann.

The screenshot shows the 'Administration Bestände' interface. At the top, there's a section for 'Alben' with a dropdown menu currently showing 'Road to FIFA World Cup Brazil 2014'. To the right of the dropdown are three buttons: '+ Neues Album' (blue), '=> Album umbenennen' (grey), and 'Album löschen' (red). Below this is a section for 'Sticker / Karten' with a '+ Neuer Sticker / Karte' button. The main part of the interface is a table with four columns: 'id', 'Title', and two columns of action buttons. The table contains four rows of data.

id	Title	Speichern	Reset	Löschen
103	Adel Rami-France	<input type="button" value="Speichern"/>	<input type="button" value="Reset"/>	<input type="button" value="Löschen"/>
127	Aiden McGeady-Republ	<input type="button" value="Speichern"/>	<input type="button" value="Reset"/>	<input type="button" value="Löschen"/>
212	Aiden McGeady-Republ	<input type="button" value="Speichern"/>	<input type="button" value="Reset"/>	<input type="button" value="Löschen"/>
175	Alan Dzagoev-Rossija	<input type="button" value="Speichern"/>	<input type="button" value="Reset"/>	<input type="button" value="Löschen"/>

**Abbildung 14**

Hier hat der Benutzer nun die Möglichkeit neue Alben zu erstellen, Alben umzubenennen oder Alben komplett zu löschen. Für die genannten Aktionen gibt es die vorgesehenen Schaltflächen „Neues Album“, „Album umbenennen“ und „Album löschen“. Betätigt man eine dieser Schaltflächen so öffnet sich ein Popup, in dem die gewünschte Funktion vollzogen werden kann.

A screenshot of a web application dialog box. The title bar at the top reads "Road to FIFA World Cup Brazil 2014 umbenennen in" and has a close button (X) on the right. Below the title bar is a large, empty text input field with a blue border. At the bottom right of the dialog are two buttons: a grey "Cancel" button and a blue "OK" button.

### Abbildung 15

Die Angabe des Albumnamens ist für die Erstellung und Umbenennung von Alben zwingend notwendig. Im Falle einer Löschung ist lediglich eine Bestätigung erforderlich.

Möchte ein Autor nun ein Album mit Karten befüllen, Karten löschen oder Karten umbenennen, so wählt er dies ebenfalls über die die Combobox aus und verwendet für die gewünschte Bearbeitung die Schaltflächen unterhalb der Rubrik „Sticker / Karten“. Zum Hinzufügen einer neuen Karte in ein Album betätigt der Autor die Schaltfläche „Neuer Sticker / Karte“, woraufhin ein Popupfenster im Vordergrund erscheint, in das der Autor den Kartennamen eintragen und die Erstellung mit „Ok“ bestätigen kann. Ein Abbruch kann mit der Schaltfläche „Abbruch“ voll zogen werden. Für die Umbenennung einer Karte, muss der Autor das zugehörige Textfeld der Karte und die Änderung durchführen. Ändert der Autor den Text, so verfärbt sich die Schaltfläche „Speichern“ grün, um zu signalisieren, dass Änderungen, die eine Speicherung benötigen, vorhanden sind. Die Änderungen am Namen der ausgewählten Karte können, sofern sie noch nicht gespeichert wurden, durch Betätigung der Schaltfläche „Reset“ zurückgesetzt werden. Eine Löschung der Karte kann durch die Schaltfläche „Löschen“ getätigt werden. Ein Popupfenster, das den Autor um eine Bestätigung der Löschung bittet, muss zur endgültigen Löschung bestätigt werden.

Des Weiteren hat der Autor noch die Möglichkeit über das Dropdownmenü in der Navigationsleiste im Administrationsbereich die Kategorie „Benutzer“ auszuwählen. Hierdurch gelangt er auf eine Auflistung, die ihm Auskunft über alle Benutzer von Tradingfun mit Angabe ihrer hinterlegten Emailadressen liefert.

#### Administration Benutzer

id	Benutzername	E-Mail	Registriert seit	
1	daniel	phunkei@gmail.com	12.10.2013	Admin
2	felix	felix-rogge@gmail.com	12.02.2014	Admin
3	Jeff	jeff@tradinx.de	12.10.2013	
5	Larry	phunkeristic@gmail.com	24.06.2014	
6	Peter	felix.rogge@gmail.com	25.06.2014	

**Abbildung 16**

## 2 Technische Dokumentation

### 2.1 Einleitung

Die primäre Zielsetzung des Projektes ist eine alternative Tauschplattform für Sticker und Sammelkarten im Web anzubieten. Darüber hinaus, soll die Applikation hochskalierbar und kostengünstig zu betreiben sein. Eine der häufigsten Architekturen im Web ist der "Architekturstack" LAMP<sup>1</sup>. Diese bringt viele funktionale Werkzeuge und Frameworks mit und ist umfangreich dokumentiert, ist aber bei horizontaler Skalierung aufwändig im Betrieb. Eine Implementierung in Java erfordert große Serverkapazitäten und benötigt für Enterpriseskalierungen oftmals teure Lizenzen. Eine Möglichkeit diese Nachteile zu umgehen, bieten neue Architekturen wie Node.JS mit entsprechenden komplementären Open-Source-Produkten.

### 2.2 Spezifikationen

#### 2.2.1 Hardware

Für den Betrieb der Software ist ein X86 oder ARM kompatibles System erforderlich. Das hier beschriebene Produktivsystem verwendet einen X86 Server mit 1,5 GHz Prozessorleistung und 2 GB RAM Arbeitsspeicher. Außerdem verfügt das System über 100 GB Festplattenspeicher (Für die Applikation selbst werden weniger als 1 GB benötigt).

#### 2.2.2 Software

Sowohl der Application Server Node.JS<sup>2</sup>, sowie das RDMS MySQL sind für zahlreiche Plattformen verfügbar und können im Zweifelsfall lokal kompiliert werden. Das hier beschriebene Produktivsystem läuft unter Ubuntu 10.04.4 LTS. Als Webserver (Reverse Proxy) kommt nginx<sup>3</sup> zum Einsatz. Alternative Webserver wie Apache sind

---

<sup>1</sup>Linux, Apache, MySQL und PHP

<sup>2</sup><http://nodejs.org/download/>

<sup>3</sup><http://nginx.org/>

ebenfalls verfügbar, sollten aber aufgrund der Architektur vernachlässigt werden (siehe hierzu 2.3.1).

## 2.3 Architektur

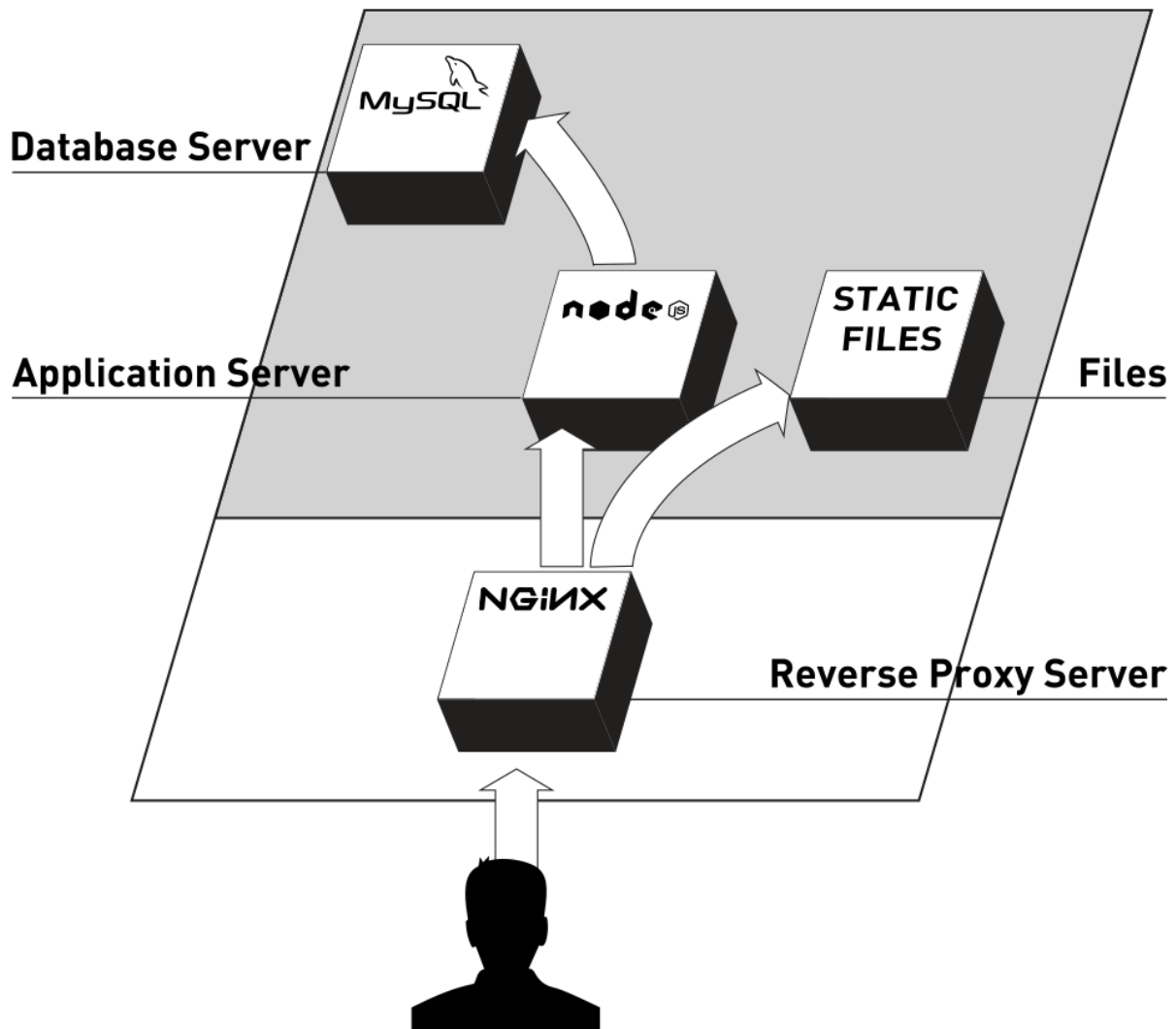


Abbildung 17

### 2.3.1 Frontendserver: nginx

Nginx ist ein Webserver, der verschiedene Techniken wie Lastverteilung, Reverse Proxying und Websockets unterstützt. Ursprünglich für eine russische Suchmaschine entwickelt, erreicht nginx mittlerweile eine Verbreitung von 22,4% unter den Top



10000 Webseiten<sup>4</sup>. Eine Schlüsselfunktion für das Projekt, ist das Reverse Proxying. Da eine Webapplikation meist aus dynamischen wie statischen Inhalten besteht, ist eine getrennte Übertragung aus Kostengründen sinnvoll. Statische Dateien brauchen bei entsprechender Konfiguration nicht erst den jeweiligen Applikationsserver durchlaufen und sparen Ressourcen. Zu diesem Zweck wurde der Webserver so konfiguriert, dass ein regulärer Ausdruck die URL auf bekannte Dateipfade prüft. Im Erfolgsfall wird umgehend die passende Datei übertragen. Die physikalische Position der statischen Dateien wurde dabei so gewählt, dass kein Zugriff auf ausführbare Dateien oder sensible Daten möglich ist. Die Applikation verfügt über einen Unterverzeichnis "public", in welchem alle Dateien liegen, die für die Clients benötigt werden (Stylesheets, JavaScript, Medien). Dabei horcht der Webserver auf Port 80, gemäß dem HTTP<sup>5</sup>. Anfragen, die nicht mit einer statischen Datei aufgelöst werden, werden an einen internen Stream auf Port 3000 durchgereicht. Auf diesem Port arbeitet der Applikationsserver und bearbeitet alle definierten Anfragen. Der Applikationsserver ist extern nur über den Frontendserver erreichbar. Im Gegensatz zum Apache, wird für die Dauer der Anfrage kein stetig aktiver Thread benötigt, sondern ein Event in einem Hauptprozess emittiert. Bei Beendigung der Aufgabe, übergibt der Prozess das Ergebnis an den Output-Stream. Dieser grundlegende Unterschied kommt besonders bei großer konkurrierender Last zum Tragen, da eine große Anzahl an Threads enorme Mengen an Speicher belegen würde<sup>6</sup>. Für eine horizontale Skalierung bietet nginx außerdem eine Lastverteilung an. Hier können beliebig viele Serverinstanzen gekapselt werden, wodurch sowohl Redundanzen geschaffen als auch Lastspitzen abgefangen werden können.

---

<sup>4</sup><http://de.wikipedia.org/wiki/Nginx>

<sup>5</sup>Hypertext Transfer Protocol

<sup>6</sup><http://readystate4.com/2012/07/08/nginx-the-non-blocking-model-and-why-apache-sucks/>

### 2.3.2 Datenbank: MySQL

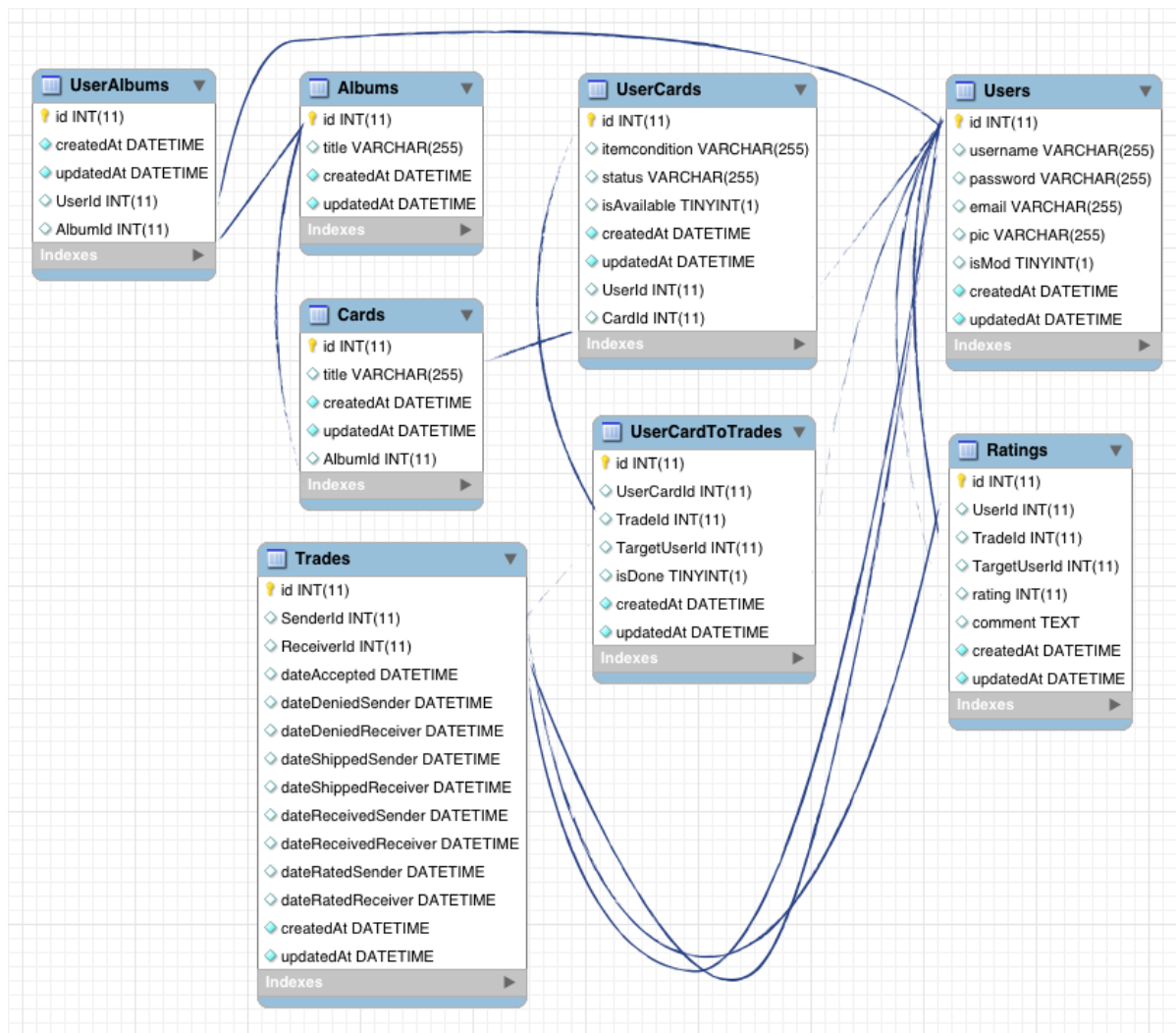


Abbildung 18

Die Datensätze werden in einer MySQL-Datenbank mit der Engine InnoDB verwaltet. Das System erstreckt sich über 8 Tabellen in Normalform. Die Tabellen werden auf sogenannte “Models” abstrahiert, um einen objektorientierten Umgang innerhalb der Applikation zu erlauben. Die Models enthalten alle Eigenschaften, die sich auch in der zugehörigen Tabelle als Spalte finden. Die Synchronisation des Schemas erfolgt ebenfalls über die Objekte. Sobald eine Änderung an der Definition vorgenommen wird, überträgt sich die Änderung automatisch auf die Datenbank bei einem Neustart

der Applikation im “Debug Modus”<sup>7</sup>. Die o.g. Models, befinden sich als Schnittmenge von Datenbank und Applikation im Verzeichnis “models”. Diese werden automatisch eingelesen und sind in der gesamten Applikation verfügbar.

### 2.3.3 Application Server: Node.JS

Der zentrale Punkt des Systems ist eine Applikation die in der Node.JS-Umgebung arbeitet. Es handelt sich hier um eine Plattform die in JavaScript implementiert ist und innerhalb der Google V8 Engine interpretiert wird. Das System ist “nicht-blockierend”. Anfragen an eine Node.JS-Instanz werden nicht in einzelnen Threads abgearbeitet, sondern asynchron in einem Prozess (Es besteht die Möglichkeit für jeden verfügbaren Rechenkern einen Prozess zu starten, was dieses Projekt bereits implementiert hat und automatisch vornimmt). Der Vorteil liegt wie bereits bei nginx im ressourcenfreundlichen Betrieb bei hohem Lastaufkommen. Das Design ist durchgehend modular aufgebaut. So wird für jede Art von Anfrage eine Route erstellt (reguläre Ausdrücke sind möglich), die eine entsprechende Funktion mitbringt. Diese Funktion übernimmt das Steuern der Models und Templates, sowie jegliche Logik, die nicht in den Bereich dieser beiden fällt. Diese Routen entsprechend weitestgehend den Controllern des MVC Entwurfsmusters. Eine Route bekommt ein Request, sowie ein Response Objekt übergeben, auf welchen wie auf üblichen Streams gearbeitet werden kann. Über das Request-Objekt werden Daten über die Anfrage und den Client eingeholt und am Ende des Prozesses wird über das Response-Objekt geantwortet. Es wird streng zwischen der Art der HTTP-Anfragen unterschieden, so dass beispielsweise für GET und POST unterschiedliche Mechanismen implementiert werden können und auch sollten (GET für lesende und POST für schreibende Zugriffe). Alle Routen befinden sich im Verzeichnis “routes” und werden bei Systemstart automatisiert eingelesen. Die Arbeitsweise innerhalb der Routen ist durchgehend asynchron, deshalb wird vermehrt mit “Callback-Funktionen” und “Closures” gearbeitet. Die Node.JS-Applikation wird durch einen zentralen Prozess initialisiert und gesteuert. Dieser Prozess ist in der Datei app.js beschrieben. Zu Beginn wird das System auf verfügbare Rechenkerne überprüft, um anschließend für jeden Kern einen Thread zu starten. Diese “Worker” genannten Threads arbeiten in einem Cluster. Sobald ein Thread endet

---

<sup>7</sup>Der Debugmodus ist ein Programmaufruf der Applikation mit zusätzlichen Parametern für den Entwicklungsbetrieb. Dieser wurde eigens für dieses Projekt entwickelt bietet Möglichkeiten für Debugging und Monitoring. Im Produktiveinsatz sollte dieser nicht verwendet werden, da zusätzlicher Overhead erzeugt und Funktionen wie Caching von kompilierten Templates deaktiviert sind.

(z.B. bei einem kritischen Fehler), startet das Cluster einen Nachfolger. Die Module innerhalb von Node.JS werden designbedingt stark gekapselt. So liegt der oberste Knotenpunkt in einem Objekt der Klasse "Express"<sup>8</sup> und kapselt wiederum weitere Module, Funktionen und Variablen, die beispielsweise in o.g. Routen verwendet werden können. Aufgrund der hierarchischen Struktur, sollte bereits zu Beginn eine strukturierte Form eingehalten werden, da eine vertikale Refaktorisierung unter Umständen sehr aufwändig werden kann. Ferner bietet Node.JS die Möglichkeit, beliebige C Programme als Module zu betreiben.

### **2.3.4 Templatesystem: Jade**

Der Benutzer oder Client bekommt als Antwort auf eine Serveranfrage HTML übergeben. Dieses HTML wird aus vorgefertigten Templates generiert, mithilfe der Jade-Engine. Die Template-Engine Jade bietet unter anderem die Möglichkeit Iterationen und Konditionen in Templates umzusetzen. Die Sprache ist minimalistischer gehalten als HTML und verzichtet auf schließende Tags. Die Position der Elemente im DOM<sup>9</sup>, wird nur durch Einrückungen im Code spezifiziert. Da Jade ebenfalls auch JavaScript basiert, kann in den Templates mit JSON-Objekten<sup>10</sup> gearbeitet werden. Neben JSON beherrscht Jade auch ungekapselte primitive Datentypen. Die benötigten Templates werden zur Laufzeit kompiliert und falls möglich gecached. Wiederkehrendes Markup wird durch Inklusion und Funktionen (sogenannte Mixins) gehandhabt und kann somit an zentraler Stelle angepasst werden.

### **2.3.5 Frontend: HTML5 / JavaScript**

Das Frontend der Applikation basiert auf HTML5, JavaScript und CSS3. Um bei der Bedienung die Ergonomie einer nativen Desktopanwendung zu erreichen, werden viele Informationen in Echtzeit ausgetauscht und Interaktionen benötigen nicht immer ein Neuladen der Seite. Durch asynchrone Requests und die Applikation werden beispielsweise die Kartenbestände abgeglichen. Die Oberfläche arbeitet eventbasiert.

---

<sup>8</sup>Express stellt eine Middleware zur HTTP-Funktionalität von Node.JS dar. Durch den bereits abstrahierten Code, kann direkter am Produktsystem gearbeitet werden, da einige Grundlagen bereits abgedeckt sind.

<sup>9</sup>Document Object Model

<sup>10</sup>JavaScript Object Notation

Einzelne DOM-Elemente (z.B. Buttons, Textfelder) emittieren bei bestimmten Handlungen festgelegte Ereignisse in der JavaScript-Umgebung, die eine Aktion auslösen. Beim Erstellen eines Angebotes wird beim editieren der Mengenangaben, der Inhalt des Dialogs mit einem nicht sichtbaren JavaScript-Objekt abgeglichen, welches beim Bestätigen an die Applikation geschickt wird. Dieses JSON-Objekt kann aufgrund der Architektur nativ ausgelesen und ausgewertet werden. Die Formatierung des Layouts ist durch CSS3 bestimmt. Durch Media-Queries und flexible Layouts, ist das Frontend auch in mobilen Endgeräten einsetzbar.

## 2.4 Performance / Load Test

Die Applikation ist sehr belastbar, trotz der leistungsschwachen Hardware. In klassischen Architekturen steigt die Rechenauslastung exponentiell zum Lastaufkommen. Die nicht-blockierende Architektur dieses Projektes zeigt hingegen einen linearen Anstieg, der geringer ist als der Anstieg des Lastaufkommens.

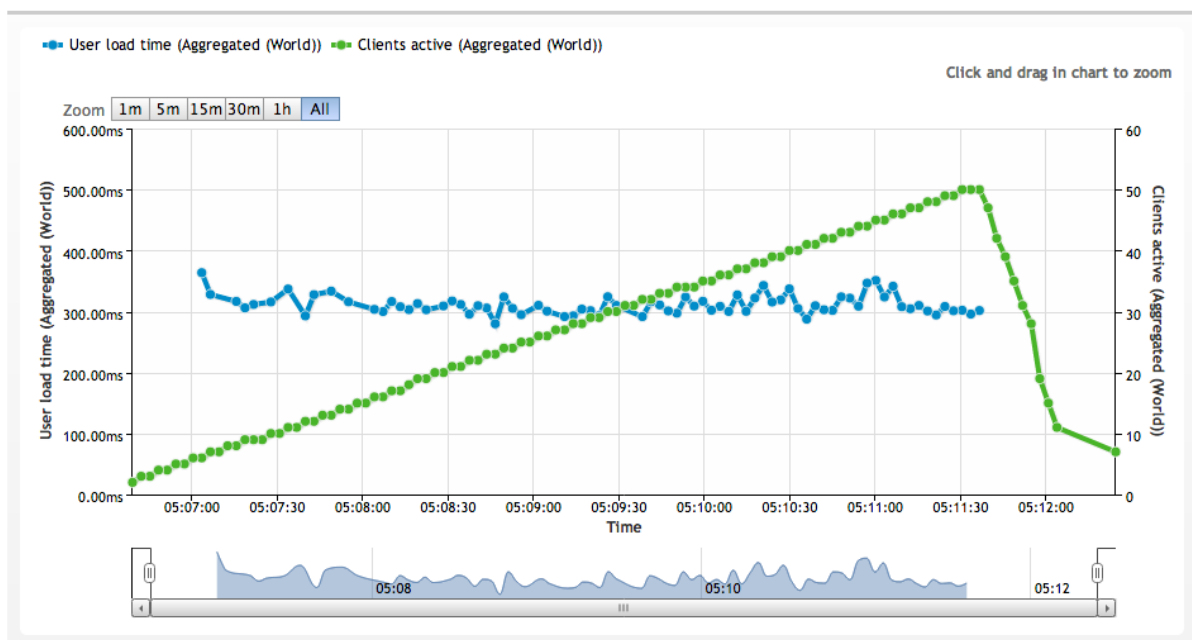


Abbildung 19

# LOAD TEST REPORT

DATE: 6/25/2014

TEST FROM : IRELAND

Query URL: http://83.169.39.87:80

Started at: Wed Jun 25 2014, 05:22:53 +02:00

Finished at: Wed Jun 25 2014, 05:22:53 +02:00

## ANALYSIS

This run generated **12,446** successful hits in **120 seconds** and we transferred **11.48 MB** of data in and out of your app. The average hit rate of **104/second** translates to about **8,961,120** hits/day.

The average response time was **189 ms**.

RESPONSE TIMES	TEST CONFIGURATION	OTHER STATS
FASTEST: <b>47 ms</b>	REGION: <b>IRELAND</b>	AVG. HITS: <b>104/sec</b>
SLOWEST: <b>634 ms</b>	DURATION: <b>120 seconds</b>	DATA TRANSFERED: <b>11.48 MB</b>
AVERAGE: <b>189 ms</b>	LOAD: <b>1-1000 users</b>	



HITS **36.27%** (12446)  
ERRORS **0.00%** (0)  
TIMEOUTS **63.73%** (21870)

Abbildung 20