|  |  |  |  |
| --- | --- | --- | --- |
|  | 2/4/2012 |  | |
| |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  | | |  | |
| Tippspiel EM2012  *Seminararbeit Web-Programmierung* | | | |
|  |  | | |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
|  |  | | Bouchier, Anas | Friedhoff, Patrick | Münster, Gregor |

Tippspiel EM2012

Seminararbeit Web-Programmierung

Inhalt

[Planung 2](#_Toc316486948)

[Mockup 2](#_Toc316486949)

[PivotalTracker 2](#_Toc316486950)

[Datenkonzept 2](#_Toc316486951)

[MySQL 3](#_Toc316486952)

[Datenbankdiagramm 3](#_Toc316486953)

[Versionsverwaltung 4](#_Toc316486954)

[Implementierung 4](#_Toc316486955)

[PHP 4](#_Toc316486956)

[Sessions 5](#_Toc316486957)

[Cookies 5](#_Toc316486958)

[JavaScript 5](#_Toc316486959)

[jQuery 5](#_Toc316486960)

[AJAX 5](#_Toc316486961)

[Smarty 5](#_Toc316486962)

[Stand 1.0 5](#_Toc316486963)

[Beschreibung der Seiten und Spezialitäten = ALL 5](#_Toc316486964)

[Tippeingabe 5](#_Toc316486965)

[Statistik 8](#_Toc316486966)

[Follow-Ups 9](#_Toc316486967)

[Import von Spieldaten 9](#_Toc316486968)

[SOAP 9](#_Toc316486969)

# Planung

## Mockup

Das Ziel der Projektarbeit ist ein dynamisch erweiterbares und einfach nutzbares Tippspiel für die Europameisterschaft 2012. Da bei Programmierarbeiten in einem Team häufig zu Missverständnissen bzw. zu verschiedenen Ansichten des fertigen Produktes kommt, wurde zuerst ein Grobdesign im Team erarbeitet.

Hierfür bietet sich das Tool Balsamiq Mockups an, da hier in kurzer Zeit ein Designentwurf entstehen kann, ohne viel Zeit in Detailfragen investieren zu müssen. Ebenfalls sind die Entwürfe jederzeit über die integrierte XML-Schnittstelle exportierbar und können so über das Web jederzeit aufgerufen bzw. angepasst werden.

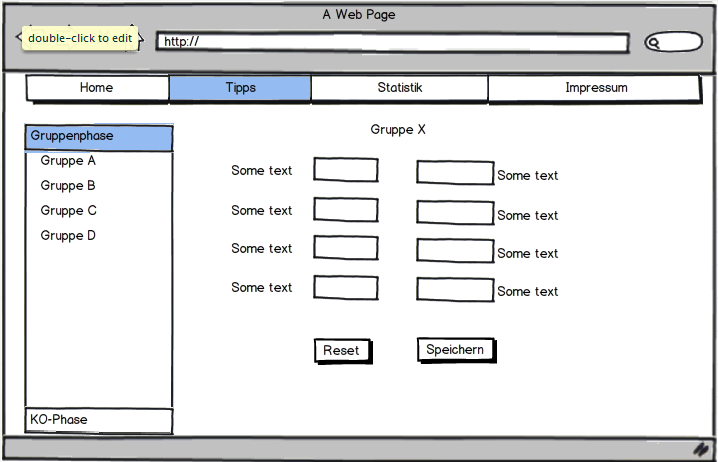


Abbildung - Feriges Mockup

## PivotalTracker

Pivotal Tracker ist ein agiles Projektmanagement Tool, das uns bei der Planung und Bewältigung der Entwicklungsaufgaben geholfen hat. Wir standen vor der Herausforderung, selbst ein Projekt zu planen, User Stories und Spezifikationen zu dokumentieren, Zuständigkeiten für die anstehenden Aufgaben im Backlog festzulegen und zu tracken. Außerdem sollten Bugs und Releases verwaltet werden.

Alles sollte jederzeit und jedem im Team zugänglich sein. Dies konnten wir mit Hilfe von Pivotal Tracker bewerkstelligen. Pivotal Tracker ist ein Webbasiertes und kostenloses Tool das SCRUM abbildet. Die Arbeit mit dem Tool sah wie folgt aus:

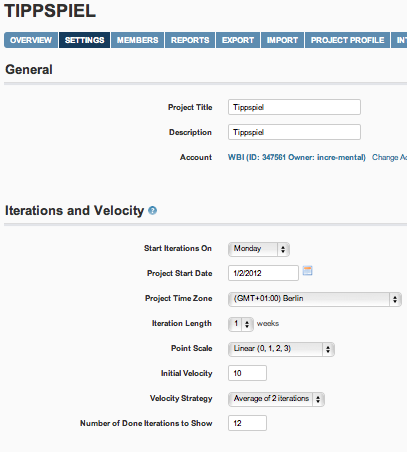
* Wir haben ein Team gebildet und ihm das Tippspiel Projekt zugewiesen. Dann wichtige Einstellungen zum Projekt vorgenommen wie z.B. die Velocity, die Dauer der Iterationen, die Punkte Skala etc.)

Abbildung 2 - Projekteinstellungen

* User Stories für die Version 1.0 wurden angelegt. Jede User Story wird zunächst in die „Icebox“ gelegt und die User Stories sind noch nicht priorisiert.
* Die User Stories lassen sich per Drag & Drop ins Backlog verschieben. Die Stories werden vorher vom Team bezüglich des Schwierigkeitsgrads geschätzt. Pivotal Tracker versucht dann automatisch die ausgewählten Stories den Menüs „Backlog“ und „Current“ anhand der für das Projekt eingestellten Velocity zuzuweisen.
* Wenn das Backlog fertiggestellt wurde, kann das Team mit der Bearbeitung anfangen.
* Ist das Team mit der Implementierung fertig, so bekommen die Stories den Status „Deliver“, damit der Product Owner akzeptieren oder ablehnen kann.

In der begrenzten, zur Verfügung stehenden zeit haben wir uns bei der Planung auf die Version 1.0 beschränkt. Die Liste der geplanten Features für die Version 2.0 steht schon bereit.

**Release Planung**

* + V 1.0
    - User-Registrierung / -Anmeldung
    - Sessions
    - Tippeingabe
    - Statistik
  + V 2.0
    - Projektportierung in ein MVC Framework
    - Userverwaltung (Admin / User)
    - Userberechtigung
    - Automatischer Ergebnisimport (Web Serices, X-Path)
    - Graphische Darstellung der Statistiken

Folgende Vorteile konnten wir bei der Nutzung von Pivotal tracker feststellen:

* Es hat eine aufgeräumte und einfach gestaltete Benutzeroberfläche, die das wesentliche anzeigt, was im Rahmen der agilen Softwareentwicklung benötigt wird.
* Der Projektstatus und der Backlog sind stets verfügbar.
* Der User muss nicht ständig Iterationen planen.

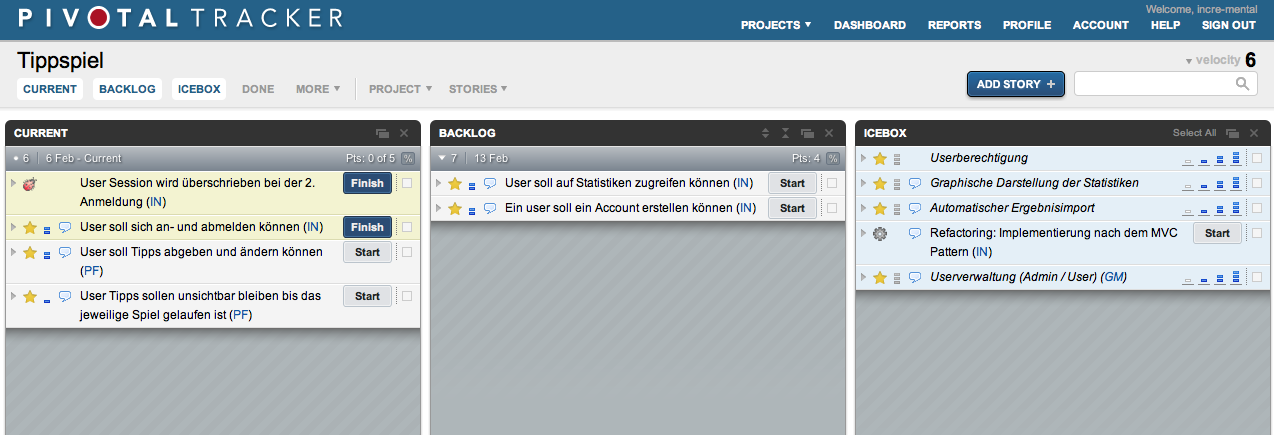
Wir konnten aus Zeitgründen weitere Funktionen des Tools nicht testen wie z.B. die Report Generierung oder die User Stories Import Funktion. Die Einfachheit der Benutzeroberfläche ermöglichte uns dafür einen schnellen Einstieg.

Abbildung 3 - Pivotal Tracker Projektübersicht

# Datenkonzept

## MySQL

Bei einem Tippspiel mit mehreren Benutzern fallen sehr viele verknüpfte Datensätze an. Um die se Daten verwalten zu können, ist eine performancestarke Datenbank nötig.

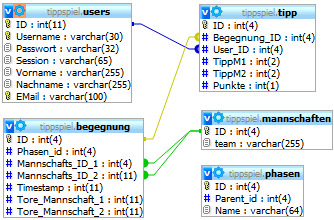
Als Mittel der Wahl hat sich für das Tippspiel eine MySQL-Datenbank herausgestellt, da diese auf fast allen Webspaces zur Verfügung gestellt wird. Ebenso ist die Performance für 50 erwartete User des Tippspiels mehr als ausreichend, sodass die Antwortzeiten der Anwendung auf ein Minimum reduziert sind.

Ein wichtiger Faktor ist hierbei die Normalisierung der Datenbank, also das Vermeiden von redundanten Datensätzen innerhalb einer Tabelle.  
Aus diesem Grund gibt es in dem Tippspiel insgesamt fünf Tabellen, die die verschiedenen Daten bereithalten bzw. die Usereingaben speichern. In der Tabelle „users“ werden alle relevanten Daten der Benutzer gespeichert, sowie die jeweilige Session-ID abgelegt. Diese Session-ID wird auf allen Unterseiten benötigt, um zum einen die Authentifizierung des Benutzers zu prüfen, zum anderen aber auch um die Tippeingaben dem richtigen Benutzer zuzuordnen.

In der Tabelle „tipp“ werden die jeweiligen Eingaben der Benutzer zu den Begegnungen gespeichert. Diese Tabelle wird nach Ende eines Spiels durch einen Cronjob, also einen automatisierten Prozess auf dem Webserver, um die erreichten Punktzahlen ergänzt, sodass das Ranking der Benutzer stets auf dem aktuellen Stand gehalten werden kann und hierfür keine Interaktion durch einen Administrator erforderlich ist.

Die Tabelle „begegnung“ wird in der Version 1 des Tippspiels noch manuell mit den Spieldaten befüllt. Dies soll in der Version 2 über eine SOAP-Schnittstelle automatisiert geschehen.

## Datenbankdiagramm



# Versionsverwaltung

Eine Source-Code Versionsverwaltung dient der Verwaltung allen Source-Codes, die einem Projekt zugehörig sind. Gerade in Teamarbeit ist es unerlässlich, dass Änderungen zwischen den Teammitgliedern einfach und fehlerfrei synchronisiert werden können. Außerdem bietet die Versionsverwaltung die Vorteile der vollständigen Verfolgung der Entwicklung, sowie der Möglichkeit Änderungen leicht rückgängig machen zu können.

Für die Versionsverwaltung des Tippspiels wird „Git“ verwendet. Hierbei handelt es sich um ein verteiltes Versionsverwaltungstool, bei dem alle Teammitglieder eine lokale Kopie des Repositories besitzen und selbst entscheiden können, wann die Änderungen den anderen Teammitgliedern zur Verfügung gestellt werden. Dieses ist ein Vorteil gegenüber zentraler Versionsverwaltungstools, wie beispielsweise „SVN“. Um im diesen Projekt den Quellcode synchronisieren zu können, wurde ein Repository bei „Github“ eingerichtet. Die Tatsache, dass für eine kostenlose Nutzung von „Github“ das Repository öffentlich ist, stellt für dieses Projekt kein Problem dar.

# Implementierung

## PHP

## Sessions

In der Login Klasse werden mit Hilfe der Datenbank- und der Session-Klasse die Anmeldedaten verarbeitet. Die Login Klasse beinhaltet das Login-Formular. Die Methode showLogin gibt den Anmeldedialog aus, mit dem ein Benutzer sich anmelden kann.  


Abbildung 4 - Login Klasse, Login Fomular

Jede Instanz der Login Klasse erwartet im Konstruktor die aktuelle Session -Id.

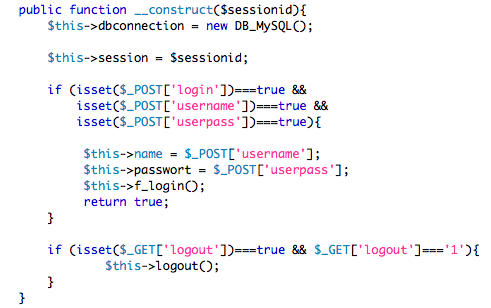


Abbildung 5 - Login Klasse, Konstruktor

Außerdem werden Login Login- oder Logout- Informationen geprüft und der Konstruktor führt das Login oder das Logout durch. Die Methoden f\_login und logout sind privat und nur innerhalb der Klasse Login erreichbar.  


Abbildung 6 - Login Klasse, f\_login()

Über die Methode logged\_in prüft die Klasse die authentifizierung des Benutzers.  
Dazu wird in der Datenbank nach einem Benutzer gesucht, der die gleiche Session-Id hat. Das Pendant dazu ist die Methode showLogout. Durch den Aufruf dieser Methode wird ein Link ausgegeben, welcher zur Abmeldung dient.

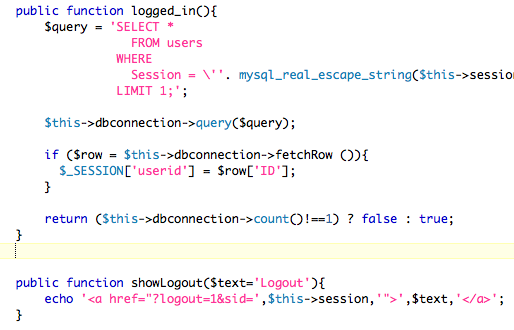
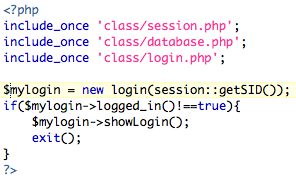


Abbildung 8 - Login Klasse, Logged\_in() und showLogout()

Abbildung 7 - logintemplate.php



Die Datei logintemplate.php ermöglich den Zugang zu geschützten Dateien.   
Diese Datei wird im Header eingebunden und ermöglicht berechtigten Nutzern den Zugang zur Seite oder aber leitet sie zur Startseite um.

Session-Informationen von einer Seite zur nächsten übergeben. Diese Sessioninformationen werden über den Link und nicht über Cookies weitergegeben. Jeder Benutzer bekommt zunächst einen Anmelde-Dialog zu sehen. Werden die Anmeldedaten korrekt eingegeben, erfolgt die automatische Weiterleitung zur angeforderten Seite.

Abbildung 9 - Session Klasse, Konstruktor

Die Session-Klasse überprüft die übergebenen Session-Informationen auf Gültigkeit oder stellt diese bereit. Sie prüft ob eine Session-ID übergeben wurde (POST und GET) oder erzeugt eine neue Session-Id.   
 Zum Schutz der Session-ID gegen Manipulation wird sie mit einem Passwort verschlüsselt und an die SID angehängt. Der Konstruktor erzeugt automatisch eine neue SID, wenn die übergebenen Daten keine gültige SID enthalten. Jeder Hyperlink von einer auf eine andere Seite muss die aktuelle Session ID beinhalten, da ansonsten die Anmeldeinformationen verloren gehen.

## Cookies

Cookies sind spezielle Zusatzinformationen, die vom Server gesetzt und auf dem Client gespeichert werden. Ein Cookie besteht immer aus einem Name-Wert-Paar und Metainformationen wie die Dauer der Gültigkeit und Domain. Vor Ausführung eines Requests prüft der Client, ob gültige Cookies für diesen Request vorliegen, und überträgt diese dann an den Webserver, der die Informationen wiederum auswerten kann. Im Tippspiel werden Cookies dafür verwendet, den Status des Gruppenmenüs, d.h. welche Gruppen sind ein- und welche ausgeblendet, auf dem Client zu speichern. Diese Informationen sind Clientspezifisch und müssen somit nicht auf dem Webserver hinterlegt werden.

## JavaScript

Bei Webanwendungen mit serverseitiger Datenanbindung erfolgt die Interaktion mit dem Anwender in der Regel über Formulare in HTML. So wird beispielsweise ein Suchvorgang gestartet oder eingegebene Daten werden in eine Datenbank geschrieben.

Auch wenn ein Formular so gestaltet ist, dass ein Anwender weiß, wie er es korrekt und sinnvoll ausfüllt, so kommt es doch immer wieder vor, dass Anwender bewusst oder unbewusst Unsinn in ein Formular schreiben und es absenden versuchen.

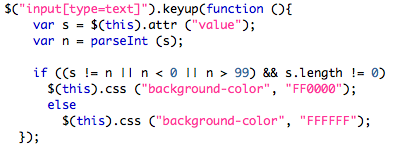
Zusätzlich zum Skript, das auf dem Server die Formulardaten empfängt und die Daten verarbeitet  wurde in unserem Projekt eine Formularüberprüfung mit JavaScript implementiert, welche zusammen mit dem Formular an den Browser übertragen wird. Das Script wird dann aufgerufen, wenn der Anwender das Formular absenden will. Sind alle Eingaben in Ordnung, werden die Daten an den Server gesendet. Stellt das Script Fehler fest, kann der Anwender diese gleich korrigieren. Der Vorteil dieser Lösung ist also, dass der User sofort weiß, ob seine Angaben valide sind oder nicht. Der Nachteil ist, dass bei deaktiviertem JavaScript keine Überprüfung stattfindet und eine Serverseitige Überprüfung erfolgen soll, bevor der User ein Feedback bekommt.

Abbildung 10 - Client-seitige Prüfung der Tippeingaben

## jQuery

Für die Gestaltung der Benutzeroberfläche haben wir jQuery UI verwendet.  
jQuery UI ist eine Erweiterung des Javascript frameworks jQuery und kommt mit vorbereiteten Features, die einfach in das bestehende Projekt integriert werden konnten. Ganz bequem geht das mit dem ThemeRoller, das eine beträchtliche Anzahl an vorkonfigurierten Themes mitbringt, die der Entwickler nach belieben bearbeiten kann.

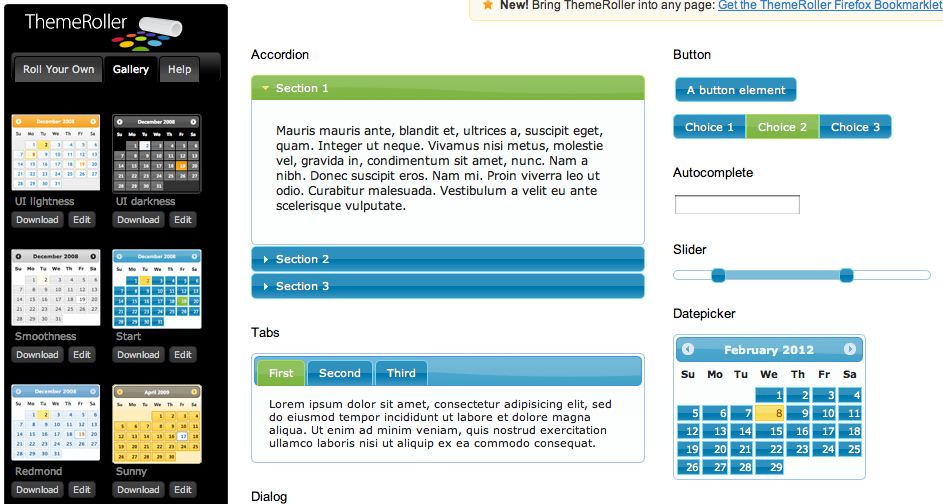
****

Abbildung 11 - jQuery UI, ThemeRoller

## AJAX

AJAX bedeutet „Asynchronous JavaScript and XML“. Der Vorteil von Ajax liegt darin, dass Teile von Webseiten nachgeladen und zur Anzeige gebracht werden können, ohne die gesamte Seite neu zu laden. Dadurch werden nur relevante Teile der Webseite aktualisiert, wodurch weniger Daten zwischen Server und Client übertragen werden müssen und somit die Geschwindigkeit erhöht wird.

Ajax wird im Tippspiel für die Anzeige der Tipps anderer User benutzt, da diese auf derselben Seite wie die Tippeingabe und erst nach einer Benutzeraktion angezeigt werden.

## Smarty

Bei Smarty handelt es sich um eine PHP Template-Engine. Diese dient vornehmlich dazu, dass der PHP-Code von der reinen Anzeige/Ausgabe von HTML Elementen getrennt wird. Für eine Seite wird dazu ein sogenanntes Template angelegt, bei der die Nutzung von Platzhaltern, die aus PHP gefüllt werden, möglich ist. Dieses wurde in diesem Tippspiel gewählt, um eine Trennung zwischen Logik und reiner Anzeige herstellen zu können. Der geringfügige Mehraufwand dieser einfach zu erlernenden Engine wurde dabei in Kauf genommen.

# Stand 1.0

## Beschreibung der Seiten und Spezialitäten

### Tippeingabe

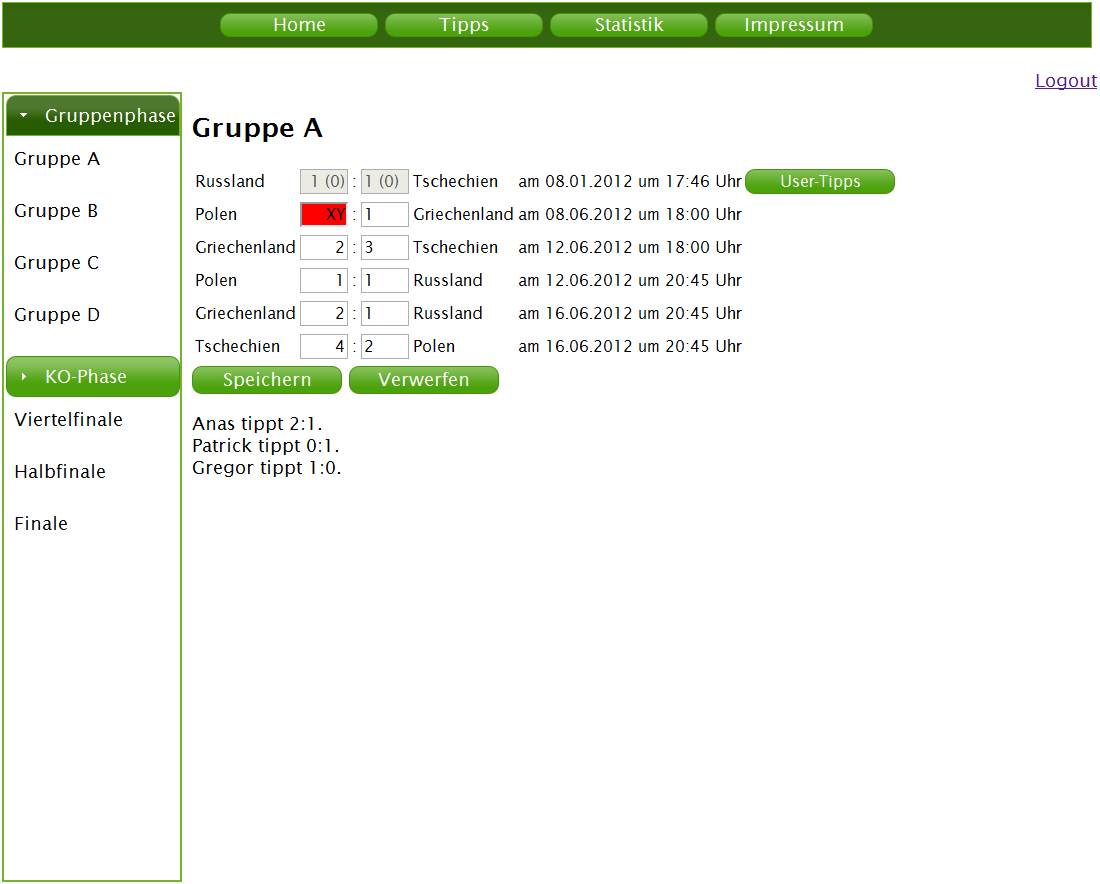
 Im oberen Bereich befindet sich die Menüleiste des gesamten Tippspiels, das linke Menü zeigt die Spielphasen, die während des Turniers durchlaufen werden. Klickt der Benutzer auf eine übergeordnete Spielphase, beispielsweise Gruppenhase, werden die verfügbaren Gruppen unterhalb der „Gruppenphase“ angezeigt. Realisiert ist dieses mit Javascript und dem OnClick Ereignis der Divs, aus dem das Menü aufgebaut ist. Eine Schwierigkeit bestand darin, beim Neuladen der Seite die angezeigten Untermenüpunkte wiederherzustellen. Dieses wird durch Setzen von Cookies beim Klick auf eine übergeordnete Spielphase und durch Auslesen derselben beim Laden der Seite realisiert:

Abbildung - Seite "Tipps"

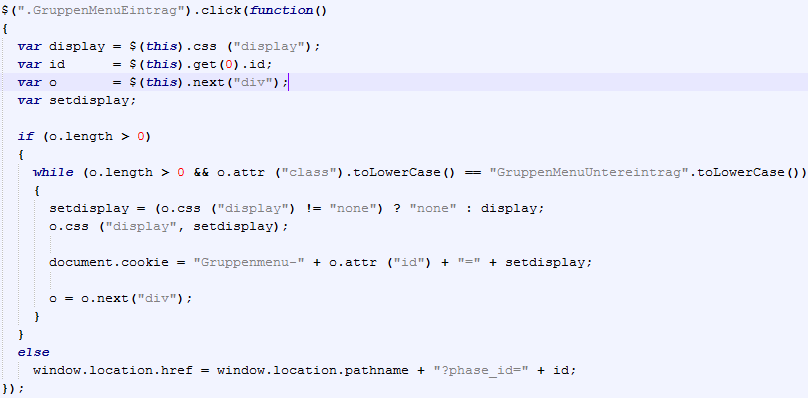
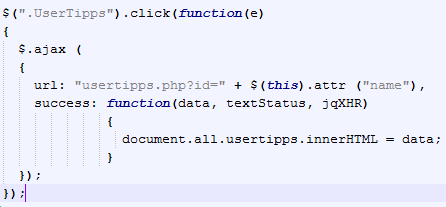


Abbildung - Setzen von Cookies

Im Content-Bereich besteht für die Benutzer die Möglichkeit eigene Tipps einzugeben, zu ändern oder zu löschen. In der Überschrift wird die im linken Menü ausgewählte Spielphase angezeigt. Darunter befindet sich eine Tabelle, in der die Begegnungen (mit Tippeingabe und Spieldatum) dargestellt werden.

Tipps können nur solange abgegeben werden, solange das Spiel noch nicht gestartet ist. Wenn ein Spiel gestartet ist, werden die Eingabefelder disabled. Dieses dient der Usability, da die Benutzer keine unerlaubten Tipps abgeben können. Da dieses aber nur die Clientseite betrifft, wird auch bei jedem Speichern eines Tipps serverseitig geprüft, ob das Spiel bereits gestartet ist. Wenn dieses der Fall ist, werden die Eingaben verworfen. Eine weitere Einschränkung der Eingabe betrifft die Anzahl der einzugebenden Daten. Diese ist auf 2 Zeichen begrenzt. Falscheingaben werden dem Benutzer dadurch signalisiert, dass beim Loslassen eines Keys die Eingabe geprüft wird. Bei nichtvalider Eingabe (Eingabe von Text) wird das Eingabefeld rot hinterlegt. Klickt der Benutzer dennoch auf die Speichern Schaltfläche wird die Eingabe verworfen und die bisherige gespeicherte Eingabe angezeigt.

Die Tippeingabe dient auch gleichzeitig dazu, das Spielergebnis anzuzeigen. Dieses wird in runden Klammern hinter dem eigenen Tipp angezeigt, um einen schnellen Vergleich für die Benutzer zu ermöglichen.

Sobald ein Spiel gestartet ist, wird eine Schaltfläche „User-Tipps“ neben der Begegnung angezeigt. Dadurch können alle Tipps der anderen Benutzer geladen und dargestellt werden. Dieses wird durch einen AJAX-Request gelöst, da nur weitere Informationen auf derselben Seite angezeigt werden sollen, ohne die komplette Seite neu zu laden. Der Request wird an die „usertipps.php“ mit dem Post-Parameter der Begegnungs-Id geschickt. Als Ergebnis kommt lediglich eine bereits formatierte Liste der Tipps anderer Benutzer zurück, welche in dem Div „usertipps“ angezeigt wird.

Wie bereits weiter oben beschrieben, wird die Template Engine Smarty verwendet. Hier ist ein Codebeispiel zu finden, welches die erste Tabellenspalte der Tippeingabe aufbaut.

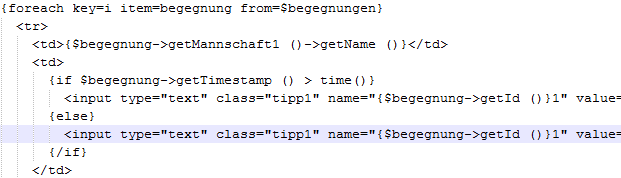


Abbildung - AJAX-Request

Sowohl das Gruppenmenü, als auch das Auslesen und Speichern der Tipps wird in PHP auf Basis von Objektorientierung gelöst. Das bedeutet, dass für jede Datenbankentität je eine Klasse für das Abbilden der Entität und eine Klasse für das Auslesen und Speichern aus, bzw. in die Datenbank erstellt wird. Dieses hat den Vorteil, dass die Logik einer Klasse von dem Lesen und Schreiben in die Datenbank getrennt wird und dadurch eine bessere Übersichtlichkeit und Wartbarkeit erreicht wird:

|  |  |
| --- | --- |
|  |  |
| Abbildung - Codebeispiel Klasse "Tipp" | Abbildung - Codebeispiel Klasse "DB\_Tipp" |

Im obrigen Beispiel sind Teile der Klassen Tipp und DB\_Tipp abgebildet. Die Klasse Superclass enthält die Standardattribute id und db, welche für jede Datenbankklasse verwendet wird. Die Klasse DB\_MySQL ist eine Klasse, welche die Verbindung zur Datenbank aufbaut und Queries absetzt.

### Statistik

Um den direkten Vergleich der Benutzer untereinander ziehen zu können, wurde eine Statistikfunktion implementiert.  
Hierbei gibt es die Möglichkeit entweder das Ranking aller Benutzer über alle bisherigen Spiele oder der einzelnen Phasen auszuwählen.   
In der aktuellen Verison 1.0 wird nur eine Tabelle sortiert nach Punkten ausgegeben, die mit der Template-Engine Smarty generiert wird.

Die Pflege der Ergebnisse der Spiele muss in der aktuellen Version noch durch einen Admin manuell eingepflegt werden. Anschließend kommt ein separates Skript zum Einsatz, welches über einen Cronjob direkt auf dem Webserver ausgeführt wird. Hierbei werden alle Spiele abgefragt, deren Timestamp bereits vergangen ist und die Felder „Tore“ in der Tabelle „Begegnung“ leer sind.

Alle zutreffenden Tipps werden durchiteriert und anhand der hinterlegten Logikfunktion werden die erreichten Punkte ermittelt und in dem Tipp des Users gespeichert. Bei kompletter Übereinstimmung des Tipps mit dem Ergebnis werden dem Benutzer 3 Punkte gutgeschrieben. Wenn das Ergebnis nicht übereinstimmt, aber die Tordifferenz korrekt ist, werden 2 Punkte vergeben. Hat der Benutzer den Sieg korrekt getippt, bekommt er noch 1 Punkt.

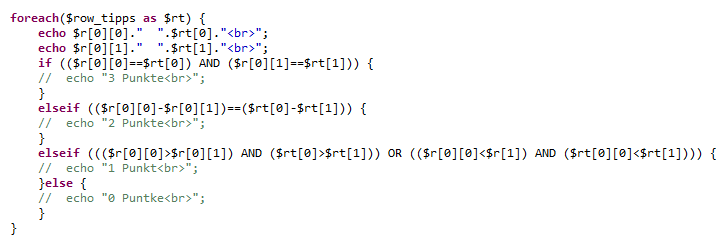


Abbildung - Funktion zur Ermittlung der Punktzahl

Die Tabelle der Benutzer wird bei jedem Aufruf des Rankings neu erstellt und mit Hilfe eines Smarty-Templates ausgegeben. Beim Aufruf der Funktion aufbau\_ranking() werden das zugewiesene Smarty-Objekt für die Ausgabe und das User-Objekt übergeben, sodass eine Markierung des jeweiligen Benutzers in der Tabelle erfolgen kann. Diese Funktion ist auch für die spätere Erweiterung vorbereitet, sodass bei einer hohen Teilnehmerzahl auch nur ein Ausschnitt der Tabelle ausgegeben werden kann, der die umliegenden Platzierungen anzeigt.

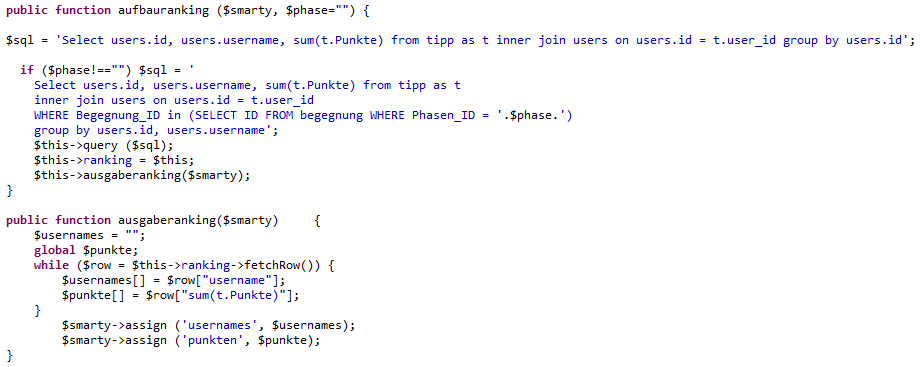


Abbildung - Funktion zur Ausgabe des Rankings

# Follow-Ups

## Import von Spieldaten

Am Anfang des Projektes fand ein Brainstorming des Teams statt, in dem Ideen für mögliche Features gesammelt wurden. Schnell stellte sich heraus, dass in der Kürze der Zeit nicht alle Ideen in die finale Version einfließen konnten.

Aus diesem Grund wurden verschiedene Versionsstände erarbeitet, die stetig weiterentwickelt werden sollen.

In der Version 1 stehen alle grundlegenden Funktionalitäten zur Verfügung.  
Hierzu gehören die Benutzerregistrierung, Login und Logout, die Eingabe der Tipps und das Ranking der Benutzer.

Um den manuellen Pflegeaufwand zu minimieren bzw. die Wiederverwendbarkeit in anderen Turnieren zu erhöhen, wird ab der Version 2 über eine SOAP-Schnittstelle direkt auf die Daten der openligadb zugegriffen. Dies war bereits ein geplantes Feature der Version 1, ließ sich jedoch nicht implementieren, da die Spieldaten noch nicht vollständig verfügbar sind.

Anschließend sind noch kleinere Implementierungen geplant. Diese umfassen eine Push-Notification an die Smartphones der Benutzer nach Ende eines Spiels mit der Übermittlung der erzielten Punkte und des aktuellen Platzes in der Rangliste.

Ebenfalls soll die Möglichkeit geschaffen werden, dass für die Begegnungen Gewinnquoten ermittelt werden und die Benutzer virtuelles Guthaben setzen können anstatt für jedes Spiel 0-3 Punkte zu erhalten.

### SOAP

Soap ist ein xml-basierendes Netzwerkprotokoll, mit dem Daten zwischen zwei Systemen ausgetauscht werden können.  
Durch die standardisierte Formatierung der Abfrage und der Ergebnisse lassen sich die so übermittelten Daten sehr gut weiterverarbeiten ohne die Daten noch extra parsen zu müssen.

Dadurch, dass die übertragenen Daten vollkommen flexibel kombiniert übertragen werden können, lassen sich alle Daten einer Begegnung in einem Objekt übermitteln. Dieser Aspekt hebt auch den Nachteil des großen Overheads bei der Übermittlung via XML auf, da mit nur einer Anfrage alle notwendigen Daten übertragen werden können.



Abbildung - Abfrage des Spiels mit der ID 626