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

Tippspiel EM2012

Seminararbeit Web-Programmierung

Inhalt

[Planung 1](#_Toc315788076)

[Mockup (->Designplanung) = GM 1](#_Toc315788077)

[PivotalTracker (->Planung, Zuweisung von Aufgaben) = AB 1](#_Toc315788078)

[Versionsübersicht = GM 1](#_Toc315788079)

[Datenkonzept 1](#_Toc315788080)

[MySQL = GM 1](#_Toc315788081)

[Datenbankdiagramm = PF 1](#_Toc315788082)

[Versionsverwaltung 1](#_Toc315788083)

[GIT = PF 1](#_Toc315788084)

[Implementierung 1](#_Toc315788085)

[PHP = ALL 1](#_Toc315788086)

[Sessions = AB 1](#_Toc315788087)

[Cookies = PF 1](#_Toc315788088)

[JavaScript = AB 1](#_Toc315788089)

[jQuery = AB 1](#_Toc315788090)

[AJAX = PF 1](#_Toc315788091)

[Smarty = PF 1](#_Toc315788092)

[Stand 1.0 3](#_Toc315788093)

[Beschreibung der Seiten und Spezialitäten = ALL 3](#_Toc315788094)

[Follow-Ups 3](#_Toc315788095)

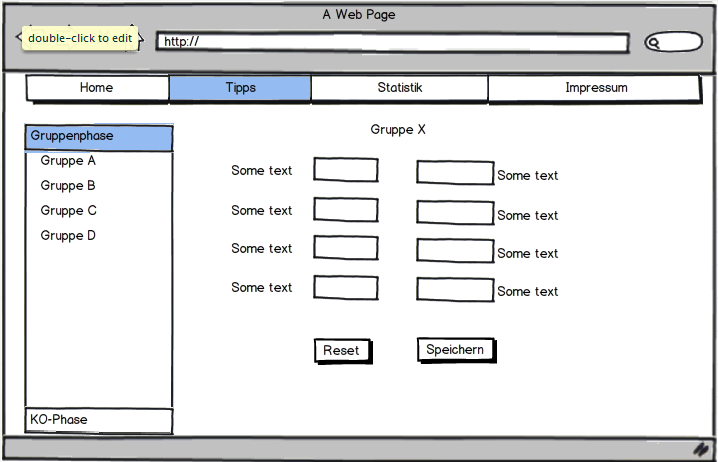
[Import von Ergebnissen/Teams/Begegnungen/Phasen = GM 3](#_Toc315788096)

# Planung

## Mockup (->Designplanung) = GM

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.



## PivotalTracker (->Planung, Zuweisung von Aufgaben) = AB

## Versionsübersicht = GM

# Datenkonzept

## MySQL = GM

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

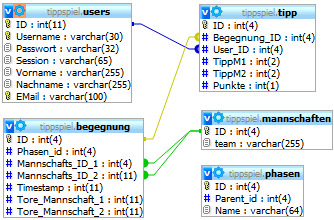
Als Mittel der Wahl hat sich für das Tippspiel eine MySQL-Datenbank herausgestellt, da diese auf fast allen Webspaces zur Verfügung gestellt wird. Ebenso ist die Performance für 50 erwartete User des Tippspiels mehr als ausreichend, sodass die Antwortzeiten der Anwendung auf ein Minimum reduziert sind.

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 durch eine SOAP-Schnittstelle automatisiert geschehen.

## Datenbankdiagramm = PF



# 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 = ALL

## Sessions = AB

## Cookies = PF

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 = AB

## jQuery = AB

## AJAX = PF

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 = PF

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 = ALL

### Tippeingabe

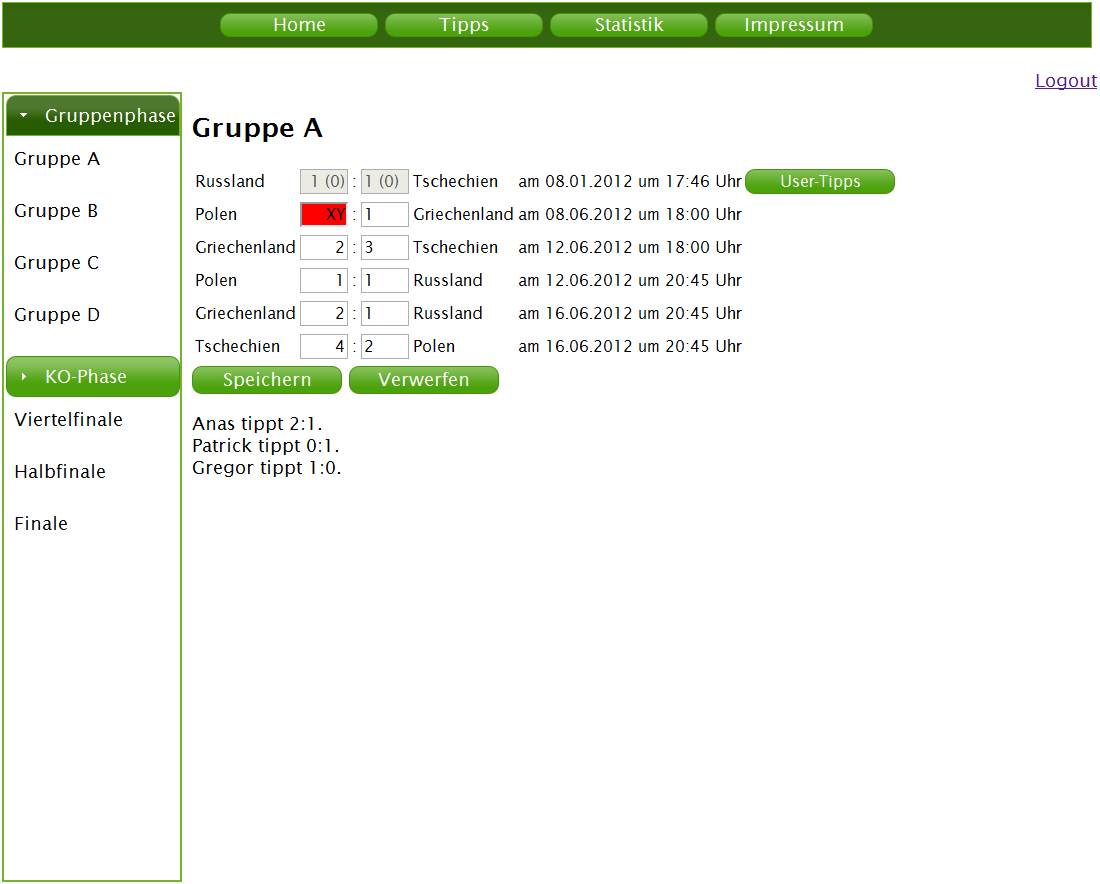
 Im oberen Bereich befindet sich die Menüleiste des gesamten Tippspiels, das linke Menü zeigt die Spielphasen, die während des Turniers durchlaufen werden. Klickt der Benutzer auf eine übergeordnete Spielphase, beispielsweise Gruppenhase, werden die verfügbaren Gruppen unterhalb der „Gruppenphase“ angezeigt. 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 1 - Seite "Tipps"

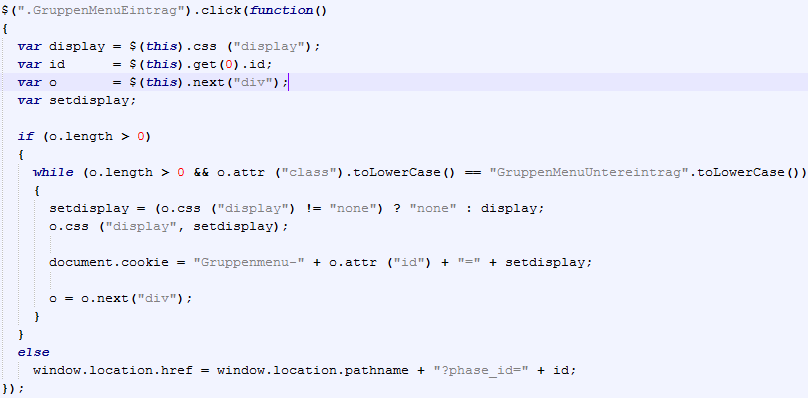
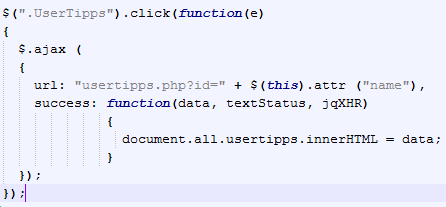


Abbildung 2 - Setzen von Cookies

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.

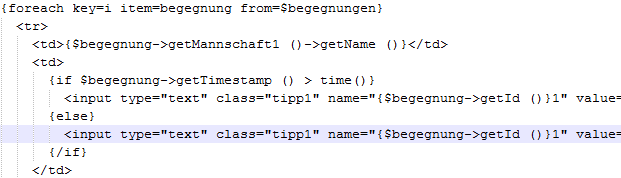
Tipps können nur solange abgegeben werden, solange das Spiel noch nicht gestartet ist. Wenn ein Spiel gestartet ist, werden die Eingabefelder disabled. Dieses dient der Usability, da die Benutzer keine unerlaubten Tipps abgeben können. Da dieses aber nur die Clientseite betrifft, wird auch bei jedem 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 3 - 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 4 - Codebeispiel Klasse "Tipp" | Abbildung 5 - 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 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.

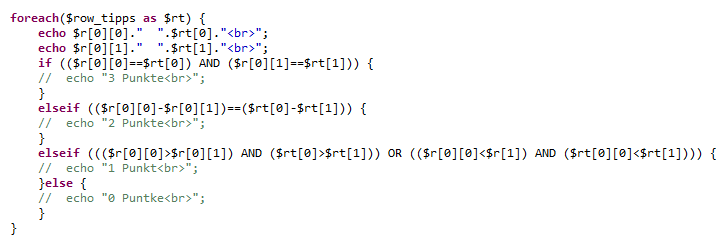


Abbildung 6 - Funktion zur Ermittlung der Punktzahl

# Follow-Ups

## Import von Ergebnissen/Teams/Begegnungen/Phasen = GM

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.