**Pflichtenheft**

**QuickQuizzer**

**-**

**Multiplayer Browser-Quiz**

**Projektmitwirkende:**

Norman Bestfleisch, Alexander Papenfuß, Marcel Wesberg, Enrico Nohl

Mittwoch, 31. Oktober 2012

**Version 1.0**

Inhaltsverzeichnis

[1. Zielbestimmung 3](#_Toc339479080)

[1.1 Musskriterien 3](#_Toc339479081)

[1.2 Wunschkriterien 3](#_Toc339479082)

[1.3 Abgrenzung 4](#_Toc339479083)

[2. Produkteinsatz 4](#_Toc339479084)

[2.1 Anwendungsbereich 4](#_Toc339479085)

[2.2 Zielgruppe 4](#_Toc339479086)

[2.3 Betriebsbedingungen 4](#_Toc339479087)

[3. Technische Produktumgebung 5](#_Toc339479088)

[3.1 Software 5](#_Toc339479089)

[3.2 Hardware 5](#_Toc339479090)

[3.3 Orgware 5](#_Toc339479091)

[4. Produktfunktionen 6](#_Toc339479092)

[5. Produktdaten 7](#_Toc339479093)

[5.1 Datenbankmodellentwurf 8](#_Toc339479094)

[6. Produktleistungen 9](#_Toc339479095)

[7. Benutzeroberfläche 10](#_Toc339479096)

[8. Qualitätsanforderungen 11](#_Toc339479097)

[8.1 Zusammenfassung 11](#_Toc339479098)

[8.2 Benutzerfreundlichkeit 11](#_Toc339479099)

[8.3 Effizienz 12](#_Toc339479100)

[8.4 Zuverlässigkeit 12](#_Toc339479101)

[9. Globale Testszenarien 13](#_Toc339479102)

[10. Glossar 14](#_Toc339479103)

# 1. Zielbestimmung

Ziel dieses Projekts ist die Erstellung eines Spiels mit dem Namen QuickQuizzer. QuickQuizzer ist ein „Echtzeit“-Browserspiel, welches Mehrspielerunterstützung bietet.

Der Spieler soll die Möglichkeit besitzen in „Echtzeit“ (siehe Kap. 10 „Glossar“) mit anderen Spielern in Wettstreit zu treten, auf Basis eines klassischen Frage-Antwort Prinzips, bei dem es darum geht, als erster die richtige Antwort zu geben.

## Musskriterien

Die im Folgenden genannten Kriterien muss die Software erfüllen:

* Der Nutzer muss sich mit einem sogenannten Nickname am Spiel anmelden können.
* Die Nutzer müssen gleichzeitig und in „Echtzeit“ am Spiel teilnehmen können.
* Der Nutzer muss die Möglichkeit besitzen die Fragen aus einer bestimmten Kategorie wählen zu können.
* Die Fragen müssen sich, nach einem Zufallsprinzip in Ihren Schwierigkeitsgrad verändern.
* Der Nutzer muss die Möglichkeit besitzen, einen anderen Spieler aufzufordern die Quizfrage zu beantworten.
* Der Nutzer muss seinen Spielstand (Punktzahl) einsehen können.

## Wunschkriterien

QuickQuizzer soll in kommenden Versionen um nachfolgende Features erweitert werden:

* Eigenständiges Login mit Benutzername und Passwort,
* browserübergreifendes Layout,
* administrative Oberfläche zur Konfiguration,
* Wiedergabe von Sprachsequenzen,
* virtuelle Spielräume, um mehrere Spielergruppen voneinander zu trennen.

## 1.3 Abgrenzung

Das Spiel soll kein auf Gewinn orientiertes Produkt werden.

# 2. Produkteinsatz

Das Browserspiel QuickQuizzer soll im Internet über einen Browser dauerhaft verfügbar sein.

## 2.1 Anwendungsbereich

Der Anwendungsbereich beschränkt sich auf das Spielen dieses Spiels. Nutzer können dieses Spiel zur allgemeinen Unterhaltung verwenden.

## 2.2 Zielgruppe

Die Zielgruppe beschränkt sich auf einen Personenkreis im Alter von 10 bis 65 Jahren. Weitere Kriterien müssen nicht erfüllt werden.

## 2.3 Betriebsbedingungen

Für das Betreiben dieses Spiels werden mindestens ein Server und ein Clients als Voraussetzung definiert.

Servervoraussetzungen:

* Node.js
* MySQL DBMS
* Internetverbindung

Clientvoraussetzungen:

* Internetverbindung
* ein Internetbrowser(z.B. Firefox, Chrome)
* JavaScript

# 3. Technische Produktumgebung

## 3.1 Software

Client:

* Internetbrowser mit aktiviertem JavaScript:
  + Mozilla Firefox
  + Google Chrome

Server:

* Betriebssystem: Ubuntu Server 12.04 LTS
* Node.js
  + Modul: Socket.io
* MySQL DBMS

## 3.2 Hardware

Client:

* internetfähiger Standardcomputer
* Computermaus oder vergleichbare Eingabegerät

Server:

* internetfähiger Server
* Standardcomputer, der die Ansprüche der o.g. Server-Software erfüllt
* Rechenkapazität:
  + CPU: mind. 1,6 GHz
  + RAM: mind. 2 GB
* Festplattenkapazität > 20 GB

## 3.3 Orgware

* Gewährleistung einer permanenten Internetanbindung

# 4. Produktfunktionen

**/F010/ - Nickname wählen**

Zu Beginn des Spiels muss der Benutzer einen individuellen Nutzernamen angeben, mit dem er während des Spiels identifiziert und ihm seine Punktzahl zugeordnet werden kann.

**/F020/ - Fragenkategorie auswählen**

Während des Spiels können zufällig ausgewählte Spieler die Kategorie bestimmen, aus der die nächste Frage stammt.

**/F030/ - Frage beantworten**

Der Spieler kann aus vier gegebenen Antworten eine auswählen, um die Frage zu beantworten.

**/F040/ - Gegenspieler „festnageln“**

Ein Spieler kann einmal pro Spielrunde einen anderen Mitspieler auffordern, eine Frage zu beantworten.

**/F050/ - Anzeigen der Punktzahl**

Währenddessen und nach dem Spiel wird dem Spieler seine erreichte Punktzahl, die sich nach der richtigen Beantwortung der Quizfragen richtet, angezeigt.

**/F060/ - Anzeigen des Highscores**

Nach einem Spiel wird allen Spielern eine Rangliste mit den zehnbesten Spielern und ihren erreichten Punkten angezeigt.

# 5. Produktdaten

**/D010/ - Spielername**

* eindeutige ID
* Nickname

**/D020/ - Highscore**

* eindeutige ID
* Spieler-ID
* Punktzahl

**/D030/ - Quizfragen**

* eindeutige ID
* Fragentext
* Kategorie-ID
* Schwierigkeitsgrad-ID
* zugehörige Antworten
* Index der korrekten Antwort

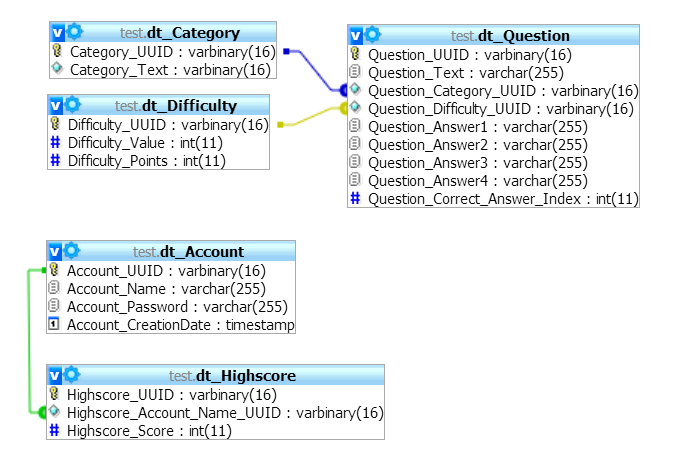
**/D040/ - Fragenkategorien**

* eindeutige ID
* Name der Kategorie

**/D050/ - Fragenschwierigkeitsgrad**

* eindeutige ID
* Schwierigkeitsgrad
* Punkte pro Schwierigkeitsgrad

## 5.1 Datenbankmodellentwurf

Abb. 1: Entwurf eines MySQL-Datenbankmodells (Änderungen vorbehalten)

# 6. Produktleistungen

**/L010/**

Für die Funktion /F010/ muss überprüft werden, ob ein gleichnamiger Spieler bereits im aktuellen Spiel vorhanden ist. Ist dies nicht der Fall, wird der Name im Spiel registriert.

**/L020/**

Für die Funktion /F030/ erfolgt eine Überprüfung auf Korrektheit der Antwort eines Spielers.

**/L030/**

Die Daten der Nutzer werden dauerhaft gespeichert, wenn aufgrund der erreichten Punktzahl ein Eintrag in den Highscore erfolgt.

**/L040/**

Es muss gewährleistet sein, dass die Kommunikation zwischen Client und Server in „Echtzeit“ stattfindet.

**/L050/**

Die Ausfallzeit des Systems muss so gering wie möglich gehalten werden.

# 7. Benutzeroberfläche

**/B010/**

Design und Layout der Weboberfläche werden mit HTML und CSS realisiert. Der Inhalt wird mit Hilfe von JavaScript aktualisiert.

**/B020/**

Die Benutzeroberfläche ist auf Mausbedienung auszulegen.

**/B030/**

Die Benutzeroberfläche sollte möglichst intuitiv und ohne nennenswerte Einarbeitungszeit verwendbar sein.

****

Abb. 2: Entwurf der Client-Benutzeroberfläche (Veränderungen sind im Laufe des Projektes nicht

auszuschließen)

# 8. Qualitätsanforderungen

## 8.1 Zusammenfassung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Produktqualität** | Sehr gut | Gut | Normal | Nicht relevant |
| **Funktionalität** |  |  |  |  |
| Angemessenheit |  |  | **x** |  |
| Korrektheit |  | **x** |  |  |
| Interoperabilität | **x** |  |  |  |
| Ordnungsmäßigkeit |  |  | **x** |  |
| Sicherheit |  |  | **x** |  |
| **Zuverlässigkeit** |  |  |  |  |
| Reife |  |  |  | **x** |
| Fehlertoleranz |  | **x** |  |  |
| Wiederherstellbarkeit |  |  |  | **x** |
| **Benutzerfreundlichkeit** |  |  |  |  |
| Verständlichkeit |  | **x** |  |  |
| Erlernbarkeit |  | **x** |  |  |
| Bedienbarkeit |  | **x** |  |  |
| **Effizienz** |  |  |  |  |
| Zeitverhalten | **x** |  |  |  |
| Verbrauchsverhalten |  |  | **x** |  |
| **Änderbarkeit** |  |  |  |  |
| Analysierbarkeit |  |  |  | **x** |
| Modifizierbarkeit |  |  | **x** |  |
| Stabilität |  | **x** |  |  |
| Prüfbarkeit |  |  | **x** |  |
| **Übertragbarkeit** |  |  |  |  |
| Anpassbarkeit |  |  |  | **x** |
| Installierbarkeit |  |  |  | **x** |
| Konformität |  |  |  | **x** |
| Austauschbarkeit |  |  |  | **x** |

Tab. 1: Zusammenfassung Qualitätsanforderungen

## 8.2 Benutzerfreundlichkeit

Es wird hoher Stellenwert auf eine simple und intuitive Bedienung der Weboberfläche beigemessen. Es soll für Benutzer ersichtlich sein, wie das Spiel zu bedienen ist. Auch die Spielregeln sollen möglichst einfach gehalten werden und schnell zu erlernen sein.

## 8.3 Effizienz

Das Zeitverhalten von QuickQuizzer ist von beträchtlicher Bedeutung. Das Spielerlebnis soll in Echtzeit stattfinden, d.h. alle Spieler sollen zur selben Zeit Fragen beantworten können und Aktionen anderer Spieler sollen umgehend zu sehen sein. Größere Verzögerungen sind zu vermeiden.

## 8.4 Zuverlässigkeit

Es muss sicher und ohne Fehler festgestellt werden, welcher Spieler als erster die richtige Antwort abgegeben hat. Diese Vorgabe dient zur korrekten Vergabe der Punkte.

# 9. Globale Testszenarien

Die Testszenarien beschränken sich auf das Sicherstellen der geforderten Funktionalität durch die Entwickler. Auf Fehlerfreiheit und Robustheit des Quellcodes kann aufgrund der knappen Zeitvorgabe nicht eingegangen werden.

**Testfall 1:**

**/T010/ - Anmeldung**

Der Benutzer meldet sich beim Spiel an.

**/T020/ - Fragenkategorie auswählen**

Der Benutzer wählt eine Fragenkategorie aus.

**/T030/ - Frage beantworten**

Der Benutzer beantwortet eine Quizfrage.

**/T040/ - Gegenspieler „festnageln“**

Ein Spieler kann einmal pro Spielrunde einen anderen Mitspieler auffordern, eine Frage zu beantworten.

**/T050/ - Punktzahl anzeigen**

Währenddessen und nach dem Spiel wird dem Spieler seine erreichte Punktzahl, die sich nach der richtigen Beantwortung der Quizfragen richtet, angezeigt.

**/T060/ - Highscores anzeigen**

Nach einem Spiel wird allen Spielern eine Rangliste mit den zehnbesten Spielern und ihren erreichten Punkten angezeigt.

# 10. Glossar

**Echtzeit** Unter dem Begriff Echtzeit wird verstanden, dass bei Veränderung eines Zustands eine automatische Aktualisierung der grafischen Benutzeroberfläche des Clients stattfindet, ohne dass der Client erfragen muss, ob es einen neuen Zustand gibt.

**Nickname** Kurzer Name, den ein Benutzer selbst auswählt und zu dessen Identifizierung verwendet wird.

**Node.js** Node.js ist ein serverseitiges Framework zum Erstellen von serverseitigen Webanwendungen mit JavaScript, das sich besonders gut für skalierbare, hochperformante und echtzeitfähige Webanwendungen eignet.

**Socket.io** Ein Node.js-Modul, das die bidirektionale Echtzeitübertragung von Daten mittels Websockets erlaubt.