

Projektarbeit 2016/2017

Quiz-Webseite

Verfasser: Benedikt Ross
Lukas Keller

Betreuer: Andre Maier
Jürgen Mayr

Elektronikschule – Oberhofer Straße 25 – 88069 Tettnang

Inhaltsverzeichnis

Inhaltsverzeichnis

Eidesstattliche Erklärung.....	1
Kurzbeschreibung.....	2
Abstract.....	2
Motivation und Ziel der Projektarbeit.....	3
Übersicht über das Zusammenarbeiten der Technologien.....	4
Gründe für die Auswahl der Technologien.....	5
Benutzeroberfläche und Benutzerhandbuch.....	6
Technische Beschreibung der Funktionen.....	11
Login-Skript.....	11
Registrierungs-Skript.....	12
Index-Skript.....	12
Profilübersichts-Skript.....	13
Ranglisten-Skript.....	13
Spiel-Starten-Skript.....	17
Spiel-Skript.....	19
Spielauswertungs-Skript.....	25
Spiel-Rundenübersicht-Skript.....	29
Adminpanel-Skript.....	31
Fragen-hinzufügen-Skript.....	32
Logout-Skript.....	32
Beschreibung der Datenbank.....	33
Aufgabenteilung und Arbeitsprozessbericht.....	34
Eigene Bewertung von Ablauf und Ergebnis.....	35
Hinweis zur Weiterentwicklung des Produktes.....	36
Abbildverzeichnis.....	37
Quellenverzeichnis für die Dokumentenation.....	37
Quellenverzeichnis für das Projekt.....	37

Eidesstattliche Erklärung

Wir versichern, dass die Projektarbeit von uns selbstständig angefertigt und nur die angegebenen Hilfsmittel benutzt wurden.

Alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, haben wir durch Angabe der Quellen kenntlich gemacht.

Datum

Unterschrift

Datum

Unterschrift

Kurzbeschreibung

Das Ziel der vorliegenden Projektarbeit war es, eine Plattform zu schaffen, auf welcher Nutzer ihr Wissen testen und sich mit anderen messen können. Diese Plattform ist webbasiert und über das Internet erreichbar. Das Quiz besteht aus 13 verschiedenen Themenbereichen. Zuerst muss der Nutzer sich registrieren, mit diesem erstellten Benutzerkonto werden dann die Spielstände gespeichert. Um das Spiel zu starten wählt der Nutzer ein Themenbereich aus und ihm werden Fragen mit je vier Antwortmöglichkeiten gestellt. Nach einer Runde wird dem Nutzer angezeigt, wie viele Fragen er richtig und wie viele Fragen er falsch beantwortet hat. Auf einer separaten Übersicht sieht er seine erreichten Punkte in den verschiedenen Themenbereichen. Auf einer Rangliste kann er seinen Platz unter den Nutzern einsehen. Außerdem hat der Nutzer die Möglichkeit Fragen einzureichen, welche dann von einem Administrator bearbeitet, hinzugefügt oder gelöscht werden können.

Abstract

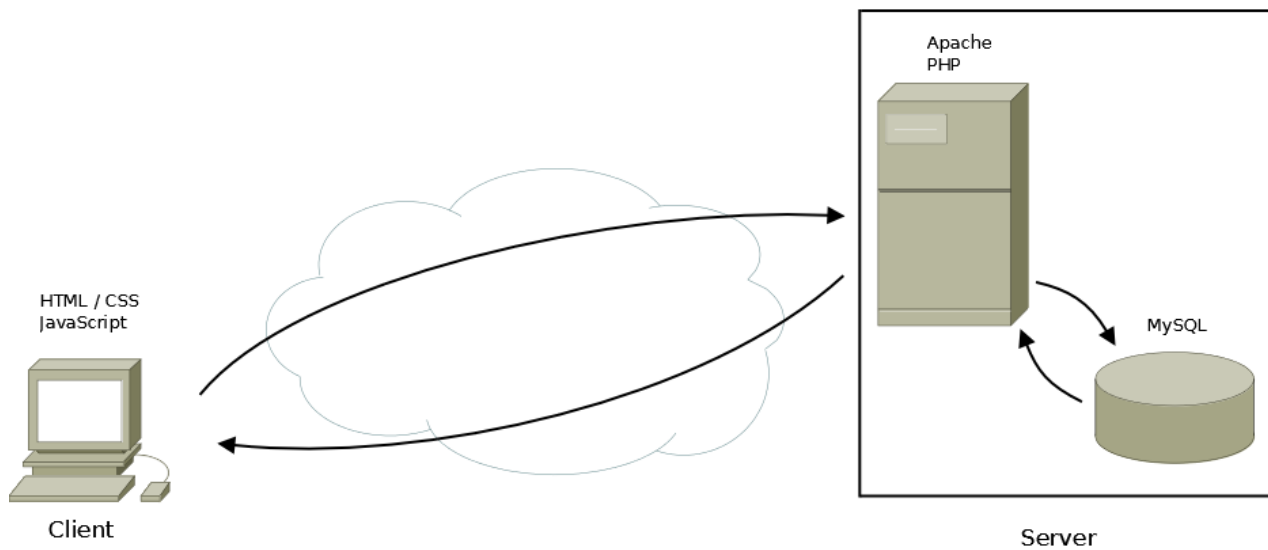
The goal of this project was to create a platform where user can test their knowledge and measure them which other users. The platform is web-based and can be accessed through the internet. There are 13 different topics in this quiz. First the user have to register an account, with this useraccount his scores will be saved. To start the game the user chooses a topic. A question will appear with four answers to choose. After the round the user is shown the number of questions he has answered right and the number of question he have answered wrong. A separate Page shows the user the reached points of the topics. On a ranglist he can look on which place he is amoung the users. A user also can submit questions to an administrator. The administrator has the option to edit, add or delete the question.

Motivation und Ziel der Projektarbeit

Unsere Motivation für die Projektarbeit entstand so, dass wir Tag täglich das Internet und somit auch Webseiten benutzen. Ob für Spiele oder auch für die Beschaffung von Informationen, wie z.B. Wikipedia. Daher haben wir es uns zum Ziel gesetzt, eine eigene kleine Webseite zu kreieren, welche nicht nur dem vergnügen gilt sondern auch einen positiven Effekt auf das Allgemeinwissen der Nutzer ausübt. Also haben wir eine Plattform entwickelt, auf welcher sich die User gegenseitig testen können. Somit können alle User eigene Quiz-Fragen hinzufügen um das Wissen der anderen zu testen.

Das eigentliche Ziel unseres Projektes soll nicht nur daraus bestehen eine Plattform für Wissensbegeisterte User zu sein, sondern uns auch einen tieferen Einblick in das Internet und das erstellen von eigenen funktionsfähigen Webseiten zu geben.

Übersicht über das Zusammenarbeiten der Technologien



Der Client ruft die Webseite auf. Das Skript wird vom Server ausgeführt. Eventuell werden auch Daten aus der Datenbank geholt. Zuletzt schickt der Server die Webseite den Client zurück.

Gründe für die Auswahl der Technologien

Wir haben uns für HTML, PHP, MySQL und JavaScript entschieden, da es für die Umsetzung einfacher war, da wir bereits umfangreiche HTML Kenntnisse hatten und da wir das Projekt ursprünglich als APP geplant hatten aber dies um einiges mehr an Einarbeitung benötigt hätte.

Grund für die Auswahl der jeweiligen Technologie:

HTML

- Für die Beschreibung der Webseite und das hervorsagende Zusammenspiel mit PHP

PHP

- Für die Funktionsfähigkeit der Webseite und da sie zum anderen eine sehr leichte Programmiersprache ist

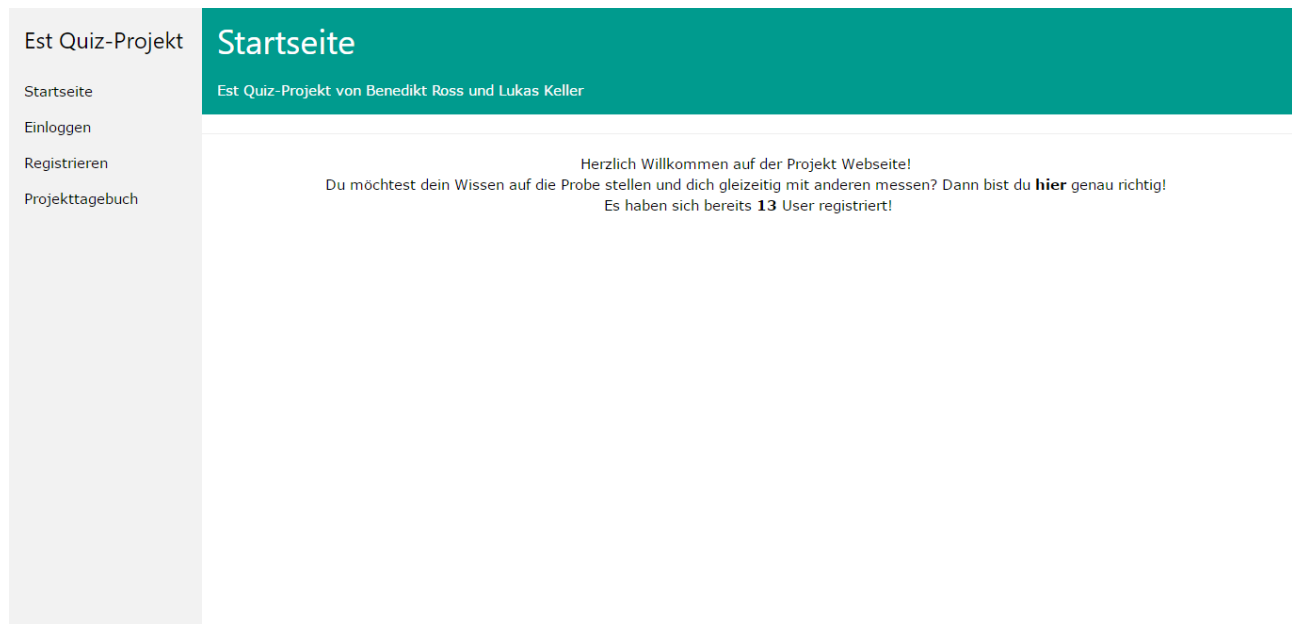
JavaScript

- Hierbei haben wir JavaScript nur verwendet, um für ein kleines Problem eine Lösung zu haben (index.php im eingeloggt Zustand, die Diashow)

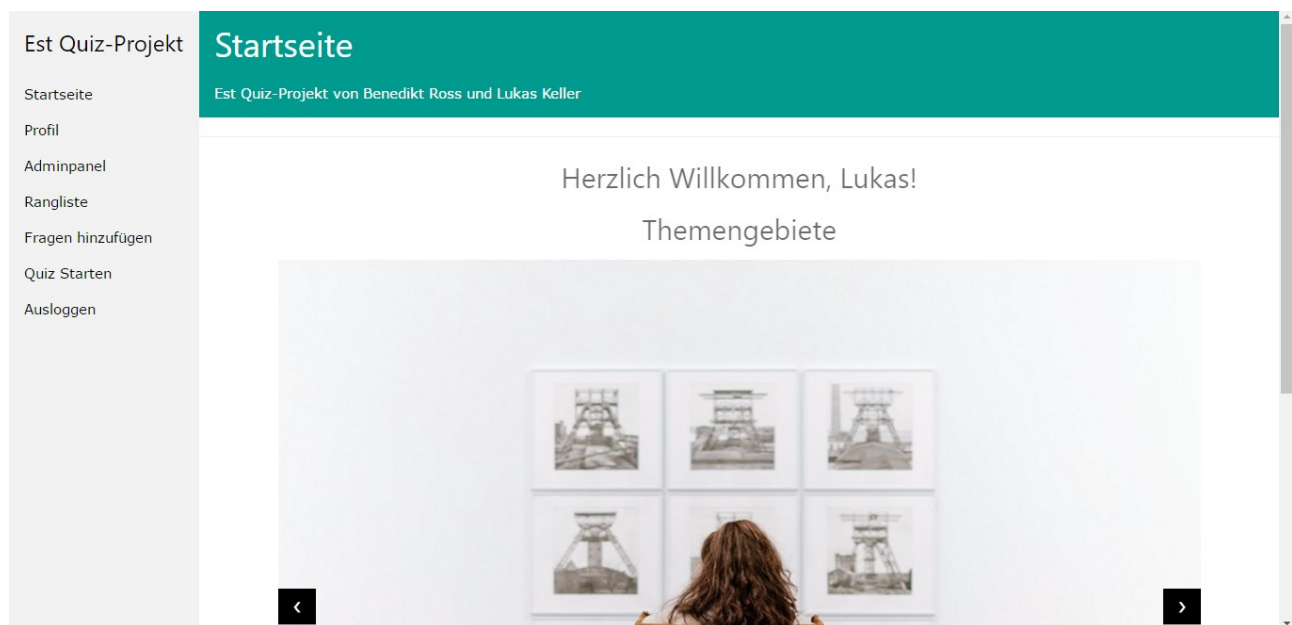
MySQL

- Für unsere Datenbank haben wir MySQL verwendet, da es sehr schnell und einfach zu verstehen ist, aber dennoch sehr effektiv

Benutzeroberfläche und Benutzerhandbuch



Die Startseite (index.php) wenn man nicht eingeloggt ist. Dort befindet sich ein kleiner Überblick, wie viele User bereits registriert sind und ein kleiner Text zur Begrüßung.



Die Startseite (index.php) wenn man eingeloggt ist. Zum einen wird man mit seinem gewählten Usernamen begrüßt und erhält, weiter unten, einen kleinen bildlichen Überblick, welche Themen man für das jeweilige Quiz nehmen kann.

Est Quiz-Projekt

Startseite

Einloggen

Registrieren

Projekttagbuch

Registrieren

Est Quiz-Projekt von Benedikt Ross und Lukas Keller

Username

Passwort

Passwort wiederholen

E-Mail-Adresse

Registrieren

Die Registrierungsseite (register.php), falls man noch keinen Account registriert hat, kann man sich dort registrieren. Hierbei soll man das Formular sinngemäß ausfüllen, damit man sich einen Account erstellen kann. Bei falschen Angaben wird eine Fehlermeldung ausgegeben.

Est Quiz-Projekt

Startseite

Einloggen

Registrieren

Projekttagbuch

Einloggen

Est Quiz-Projekt von Benedikt Ross und Lukas Keller

Username

Passwort

Einloggen

Hat man bereits einen Account registriert, kann man sich dort mit seinen Accountdaten einloggen und wird dann zur Startseite weitergeleitet, sofern die Accountdaten mit den Daten in der Datenbank übereinstimmen.

Est Quiz-Projekt
Startseite
Profil
Adminpanel
Rangliste
Fragen hinzufügen
Quiz Starten
Ausloggen

Profil

Est Quiz-Projekt von Benedikt Ross und Lukas Keller

Persönliches

Registrierungsdatum	Identifikationsnummer	Status
2017-04-07 08:56:50	1	Admin

Globales

Globaler Score	Fragen hinzugefügt	Fragen beantwortet	Fragen richtig beantwortet	Fragen falsch beantwortet
34	23	54	34	20

Themenstatistiken

Thema	Fragen beantwortet	Prozentual erledigt
Kunst	2	40% von 100%
Bibel	10	66.666666666667% von 100%
Essen	0	0% von 100%

Das Profil (profil.php) beinhaltet sämtliche Statistiken, welche du während dem Spielen erspielt hast, auch werden das Registrierungsdatum, sowie die Identifikationsnummer und der derzeitige Status (Rang) angegeben.

Est Quiz-Projekt
Startseite
Profil
Adminpanel
Rangliste
Fragen hinzufügen
Quiz Starten
Ausloggen

Rangliste

Est Quiz-Projekt von Benedikt Ross und Lukas Keller

Platz	Username	Punkte
1	Lukas [Admin]	34
2	benedikt [Admin]	15
3	samu [User]	9
4	April [User]	8
5	Username [User]	8
6	Floqueboque [User]	6
7	haha [User]	4
8	donald [User]	3
9	Anonym [User]	1
10	Jochen [User]	1

Die Rangliste (rank.php) beinhaltet die besten 10 Spieler. Hierbei wird zuerst der Platz ausgegeben, dann der Name (Grün = man selbst), dahinter der Rang und zuletzt die Anzahl der Punkte, welche man sich erspielt hat.

Est Quiz-Projekt
Startseite
Profil
Adminpanel
Rangliste
Fragen hinzufügen
Quiz Starten
Ausloggen

Fragen hinzufügen

Est Quiz-Projekt von Benedikt Ross und Lukas Keller

Themenauswahl
Kunst

Quiz Frage

Richtige Antwort

1. Falsche Antwort

2. Falsche Antwort

3. Falsche Antwort

Die Seite (addquestion.php) um eigene Quiz-Fragen hinzuzufügen (falls man Admin ist) oder sie einzureichen (kein Admin). Hierbei wählt man zuerst das Themengebiet aus, dann die Quiz-Fragen und zuletzt die jeweiligen Antworten, wobei die erste Antwort die Richtige sein muss. Sobald alles ausgefüllt wurde, wird man zu einer Übersicht weitergeleitet, in welcher man seine Angaben sieht und diese falls nötig noch ändern kann oder sie dann einfach einfügt bzw. Einreicht.

Est Quiz-Projekt
Startseite
Profil
Adminpanel
Rangliste
Fragen hinzufügen
Quiz Starten
Ausloggen

Spiel starten

Est Quiz-Projekt von Benedikt Ross und Lukas Keller

Wähle aus einer der Themengebiete das Thema für dein Quiz

Kunst

Weiter

Das Spielprinzip (play.php), dort erhält man nun die Möglichkeit sein gewünschtes Themengebiet auszuwählen und dieses dann zu spielen. Falls in dem jeweiligen Themengebiet keine Fragen mehr zuverfügung stehen (alle bereits beantwortet) wird eine kleine Fehlermeldung ausgegeben.

Est Quiz-Projekt

Startseite

Profil

Rangliste

Fragen hinzufügen

Quiz Starten

Ausloggen

Spiel

Est Quiz-Projekt von Benedikt Ross und Lukas Keller

Wie viele verschieden Arten von Tieren und Pflanzen gibt es auf der Welt?

☐ 2,45 Millionen

☐ 1,75 Millionen

☐ 1,98 Millionen

☐ 1,32 Millionen

Weiter

Hat man nun ein Themengebiet ausgewählt, geht es auch schon mit der ersten Quiz-Fragen los, nach 10 Fragen ist eine Runde beendet.

Est Quiz-Projekt

Startseite

Profil

Rangliste

Fragen hinzufügen

Quiz Starten

Ausloggen

Rundenübersicht

Est Quiz-Projekt von Benedikt Ross und Lukas Keller

Ergebnis der 1. Runde

Richtige Antworten **7**

Falsche Antworten **3**

Startseite

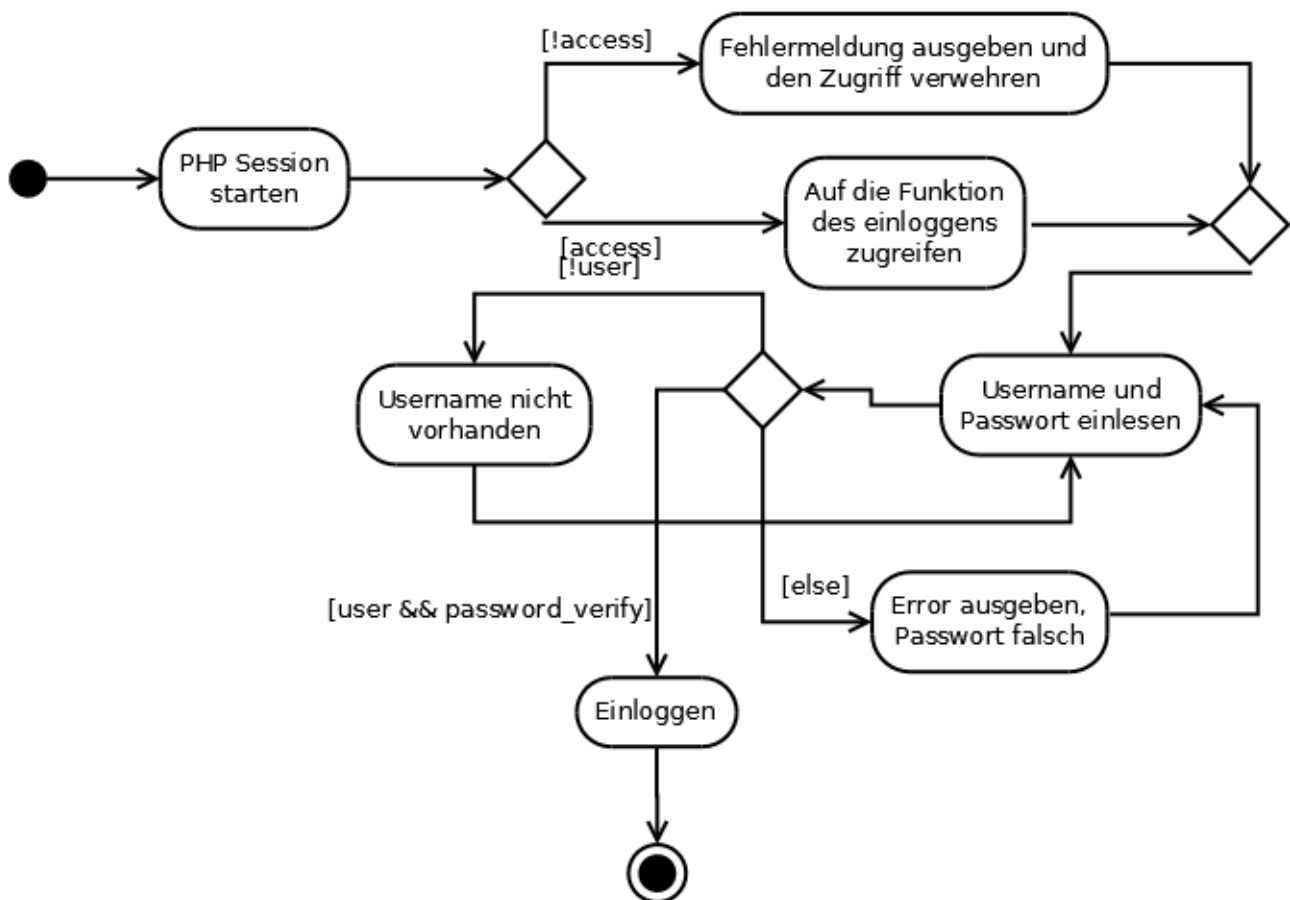
Weiter

Hat man nun 10 Quiz-Fragen beantwortet, wird man auf die Spielübersicht (game_overview.php) weitergeleitet. Dort erhält man einen kleinen Überblick in welcher Spielrunde man gerade ist, wie viele Fragen man richtig beantwortet hat und wie viele man falsch beantwortet hat. Daraufhin kann man entweder zurück zur Startseite oder noch einmal eine neue Runde spielen.

Technische Beschreibung der Funktionen

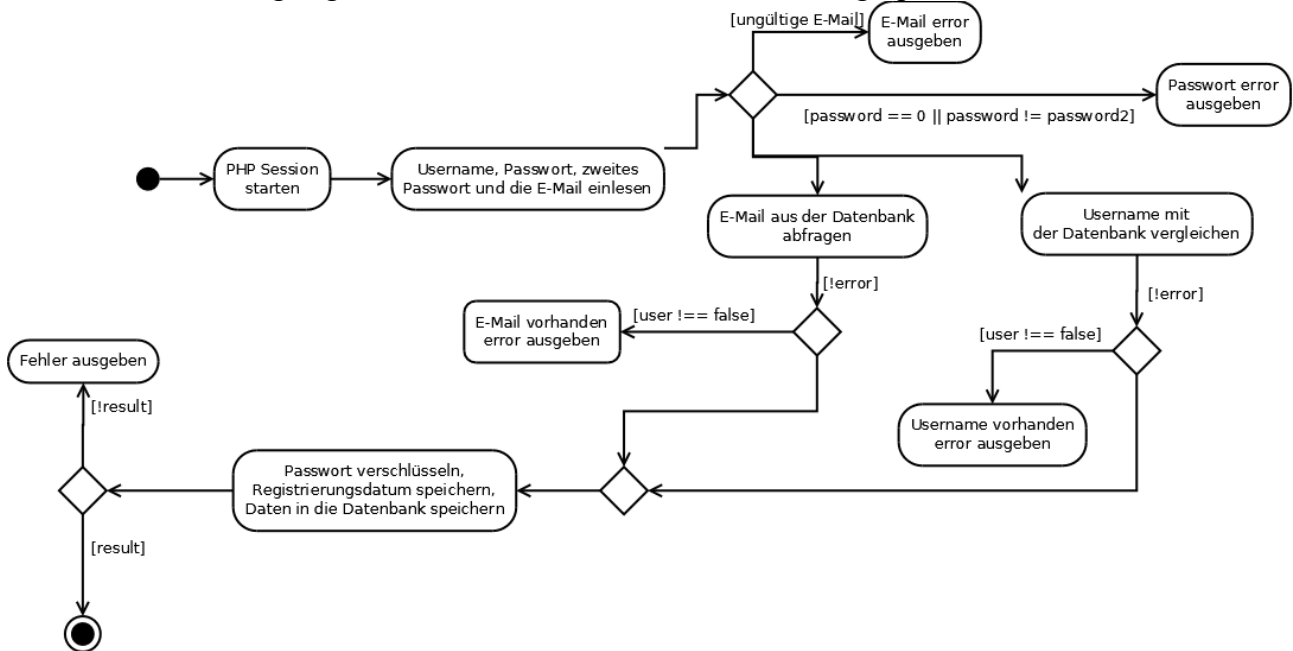
Login-Skript

Zunächst wird geprüft, ob man Zugang hat (ob man bereits eingeloggt ist), danach wird der Username und das Passwort eingelesen. Ist der Username nicht vorhanden, wird eine Fehlermeldung ausgegeben. Sollte der Username richtig sein, wird das Passwort mit dem aus der Datenbank verglichen, ist dieses falsch wird eine Fehlermeldung ausgegeben. Falls der Username und das Passwort richtig ist, wird der User eingeloggt.



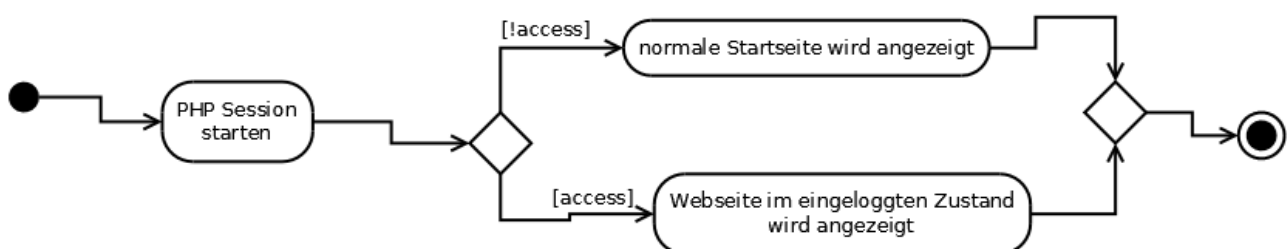
Registrierungs-Skript

Zuerst wird die PHP Session gestartet, dann der Username, die E-Mail und dann das Passwort eingelesen. Weiter wird dann die E-Mail überprüft, ob es eine gültige ist. Dann das Passwort mit dem zweiten Passwort verglichen um sicherzustellen, dass der User sich nicht vertippt hat. Als nächstes wird dann überprüft, ob es den Usernamen bereits gibt. Sollte das alles richtig sein, wird ein User-Account angelegt und sämtliche Daten in die Datenbank gespeichert.

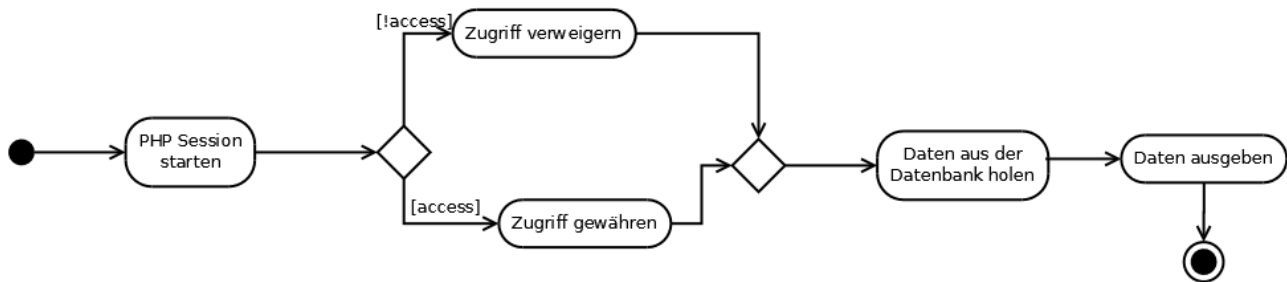


Index-Skript

Es gibt zwei verschiedene Versionen der Startseite. Bei der ersten kommt man nur drauf, wenn man nicht eingeloggt ist, bei der anderen kommt man nur drauf wenn man eingeloggt ist.



Profilübersichts-Skript



Bei dem Profil-Skript wird zuerst eine PHP Session gestartet, dann überprüft ob man Zugang hat und dann die Daten, des jeweiligen Users, aus der Datenbank geholt und ausgegeben.

Ranglisten-Skript

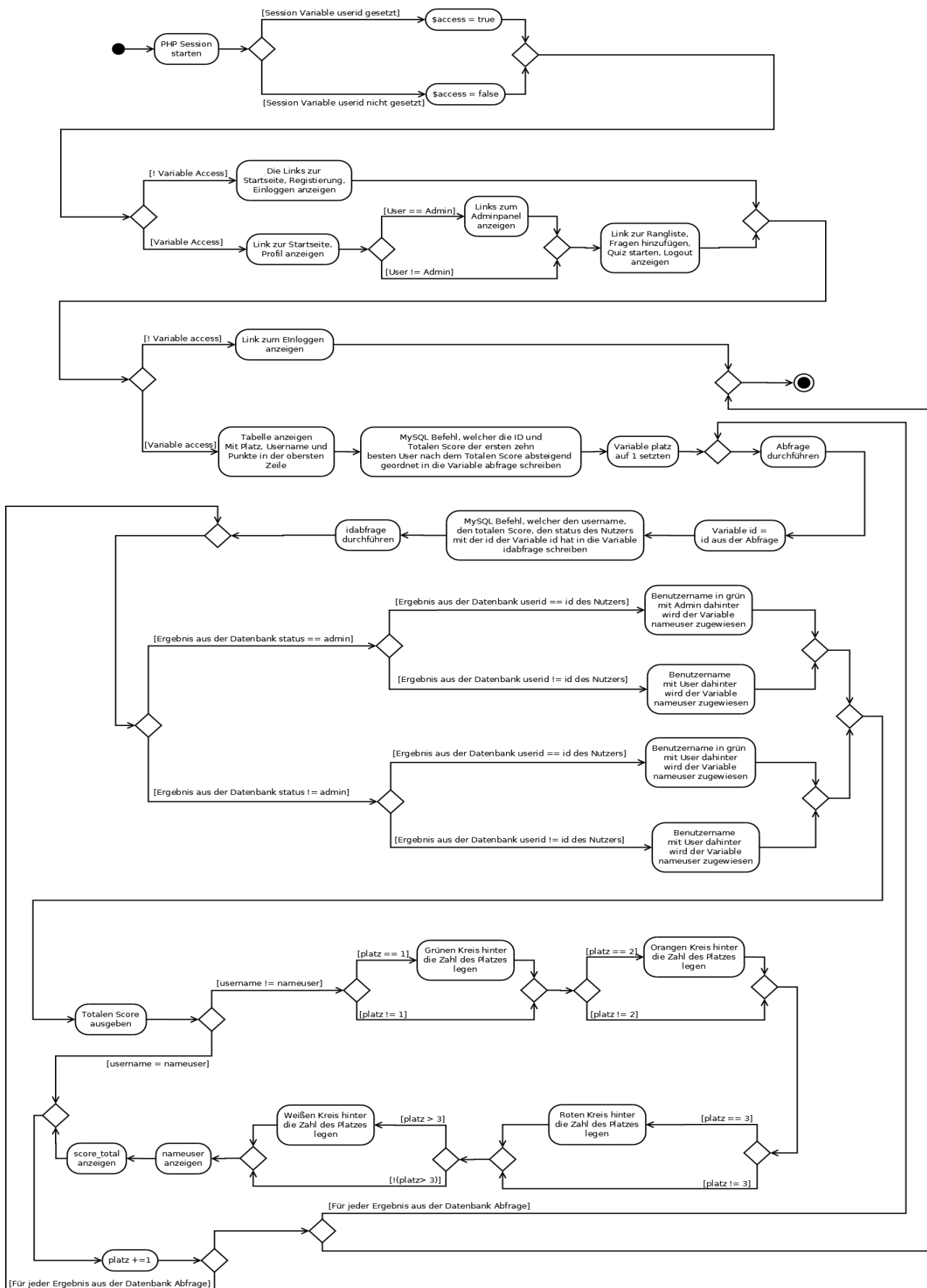
Zuerst wird die PHP Session gestartet und das Skript `function.php`, welches das PHP Data Object instanziiert. Ist die Session-Variable `userid` gesetzt wird der Variable `access` der Boolean Wert `true` zugewiesen, ansonsten der Boolean Wert `false`. Als nächstes bekommt die Variable `id`, den Wert der Session-Variable `userid` zugewiesen. Über eine Datenbank werden alle Daten des Nutzer aus der Datenbank geholt. Nun folgt der HTML Doctype und der HTML und Head Tag wird geöffnet. Im Head Tag werden die Metadaten, der Titel und das CSS-Stylesheet definiert. Jetzt wird der Head geschlossen und der Body Tag geöffnet. In einer `div`-Box wird nun die grau hinterlegte Seitenleiste festgelegt. Zuerst kommt die Überschrift „Est Quiz-Projekt“. Ist der Inhalt der Variable `access` `false`, wird der Link zur Startseite, Einloggen, zum Registrieren und zu dem Projekttagbuch auf Github angezeigt. Ist dies nicht der Fall, wird der Link zur Startseite, zum Profil, falls der Administrator ist, wird noch der Link zum Adminpanel, der Link zur Bestenliste, der Link zur Seite zum Fragen hinzufügen, der Link zum Spiel starten und der Link zum Ausloggen angezeigt. Nun wird die `div`-Box geschlossen und eine neuer `div`-Tag wird geöffnet, mit welchen die ganze Seite um 15 % nach rechts verschoben wird. In einem weiteren `div`-Tag wird die grünliche Box, die oberhalb der Seite angezeigt wird definiert. Die Überschrift „Rangliste“ mit dem Paragraph „Est Quiz-Projekt von Benedikt Ross und Lukas Keller“ werden angezeigt. Nach der Schließung des `div`-Tags folgt eine horizontale Linie. Hat der User jetzt Zugriff, d.h. ist der Inhalt der Variable `access` `false`, wird der Paragraph „Bitte zuerst einloggen“ mit dem Link zum Einloggen angezeigt. Andernfalls wird eine Tabelle angezeigt, welche eine Weite von 90% hat und links und rechts ein Rand von 5%. Die oberste Zeile wird hell grau hinterlegt und zeigt in der ersten Spalte „Platz“, in der zweiten Spalte „Username“ und in der dritten Spalte „Punkt“ an. Die Variable `platz` wird auf

eins gelegt. Über eine MySQL Datenbank Abfrage werden die ID und totale Score von den ersten zehn Nutzern, mit den höchsten Scores, in absteigender Reihenfolge geholt. Für jedes Ergebnis aus der Abfrage wird das Ergebnis der id in die Variable id geschrieben. Als nächsten wird der username, der totale score ,der Status, die id des Nutzer der die id der Variable id hat. Für jedes Ergebnis aus der Datenbank Abfrage wird zuerst der HTML Tag tr ausgegeben um den Anfang einer Zeile zu definieren. Wenn der Nutzer Administrator ist, der Wert der Session-Variable userid ist und das Ergebnis der id aus der zweiten Datenbank Abfrage den gleichen Wert hat, wird der Variable nameuser der HTML Code, zuerst bekommt der Nutzernamen die Farbe grün und das Wort „Admin“ in Klammern rot, zugewiesen. Ist der Inhalt der Variablen nicht gleich wird der Variable nameuser der HTML Code, um den username, aus der Datenbank Abfrage und das Wort „Admin“ in Klammern rot gefärbt.

Ist der Nutzer kein Admin, aber der Inhalt der Session-Variable userid und der Inhalt der Variable id aus der Datenbank Abfrage gleich, wird der Variable nameuser der HTML Code zugewiesen, um den Benutzernamen aus der Datenbank Abfrage in grün anzuzeigen und User in eckigen Klammern und in blauer Farbe dahinter. Ist der Inhalt der Variablen nicht gleich so wird der Variable username der HTML Code zugewiesen, um den Username aus der Datenbank Abfrage und der Wort „User“ in eckigen Klammern und in blau anzuzeigen. Hier enden die if-Verzweigungen.

Der Variable score_total wird der HTML Code einer Spalte mit den totale_score aus der Datenbank zugewiesen.

Ist nun der Wert der Variable username nicht gleich wie der Inhalt der Variable nameuser, werden die Plätze mit verschiedenen Farben hinterlegt. Ist so der Wert der Variable eins, so wird der HTML Code, um den Inhalt der Variable platz mit einem grünen Kreis zu hinterlegen ausgegeben. Ist so der Wert der Variable zwei, so wird der HTML Code, um den Inhalt der Variable platz mit einem orangenen Kreis zu hinterlegen ausgegeben. Ist aber der Wert der Variable drei, so wird der HTML Code, um den Inhalt der Variable platz mit einem roten Kreis hinterlegt ausgegeben. Und ist so der Wert der Variable größer als drei, so wird der HTML Code, um den Inhalt der Variable platz mit einem weißen Kreis hinterlegt ausgegeben. Nun wird der Inhalt der Variable nameuser und der Variable score zurückgegeben. Hier endet die if-Verzweigung. Jetzt wird der Inhalt der Variable platz um eins erhöht und das Ende der Spalte ausgegeben. Hier ist dann das Ende der inneren foreach-Schleife und auch das Ende der äußeren foreach-Schleife. Dann wird noch der Tabellen Tag, der div Tag, der Body Tag, der HTML Tag geschlossen.

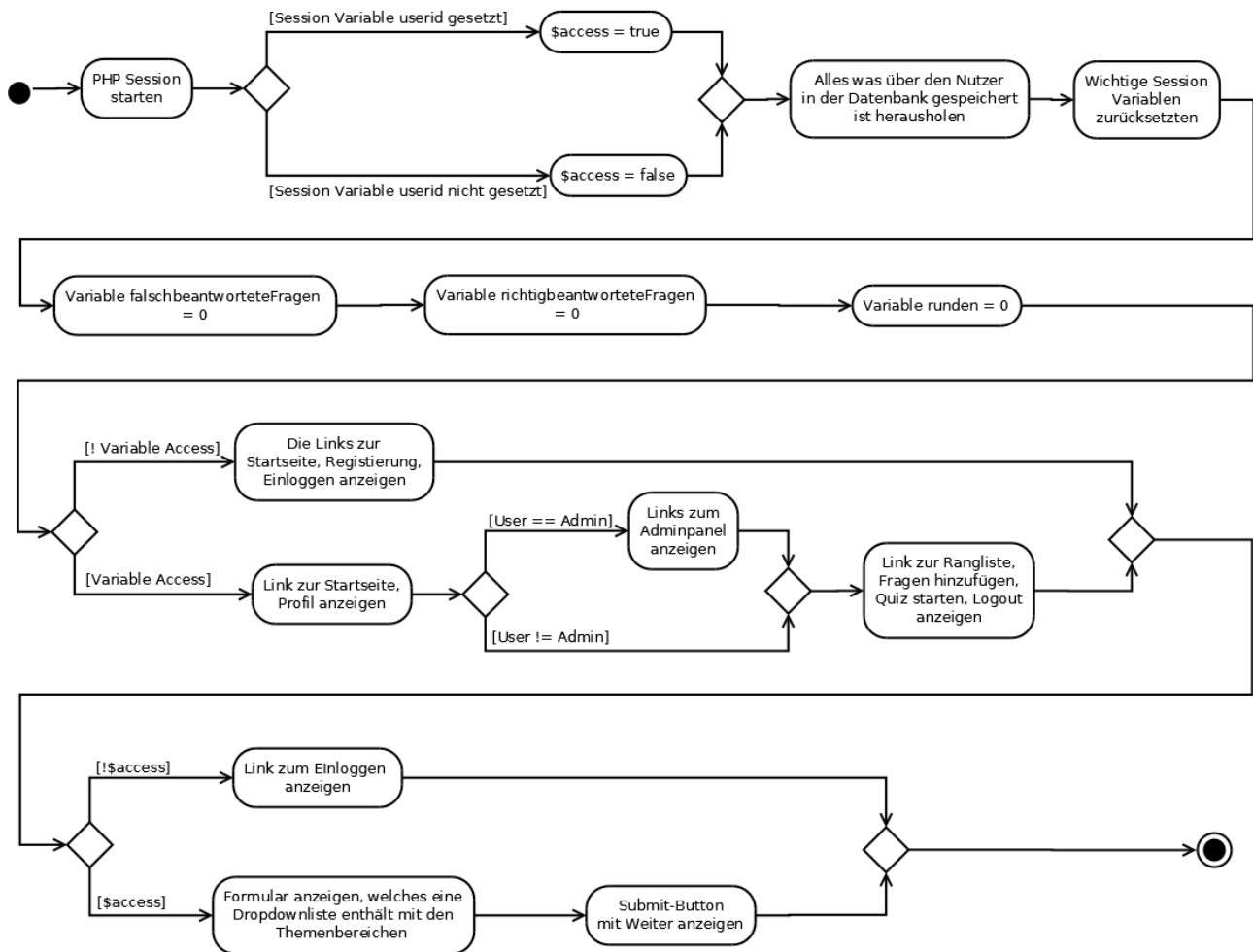


Spiel-Starten-Skript

Das Skript „Spiel starten“ gibt den Nutzer die Möglichkeit ein Themenbereich auszuwählen, über dem ihm in dem darauffolgenden Spiel Fragen gestellt werden. Zuerst wird, in dem PHP Teil, die PHP Session gestartet. Darauf wird das function Skript eingebunden, welches das PHP Data Object instanziert um eine Verbindung zur Datenbank aufzubauen. Die Userid die in einer Session-Variable ist, wird der Variable userid zugewiesen. Über eine Datenbank Abfrage werden nun alle Einträge der Nutzer mit der userid aus der Datenbank geholt. Der username aus der Datenbank Abfrage wird der Variable username zugewiesen. Wichtige Session-Variablen wie themaQuestion und themaScore, welche das Thema für Datenbank Abfragen enthalten, die „ersteMal“ Variable, welche angibt ob es das erste Mal eines Spieldurchlaufes ist, die „IDaktuelleFrage“, welche die ID der aktuellen Frage enthält, und die die Variable „durchlauf“, welche die Anzahl der Durchläufe enthält, zurücksetzen über die PHP Funktion „unset()“.

Die Session-Variable „falschBeantworteteFragen“ enthält die Anzahl an Fragen die der Benutzer in der Runde falsch beantwortet hat, wird auf null gesetzt. Genauso auch die Session-Variable „richtigBeantworteteFrage“, welche die Anzahl an Fragen enthält die richtig beantwortet wurden. Die Session-Variable „runden“ zählt die Runden an Fragen und wird hier auf 0 gesetzt. Darauf folgt der HTML Doctype und dann im Head-Bereich die Metadaten, den Titel und die Einbindung des Stylesheet. Nun kommt der HTML-Body, mit zuerst der grauen Seitenleiste, welche sich in einer div-Box befindet. Die Überschrift ist „Est Quiz-Projekt“. Hat der Nutzer kein Zugriff, d.h. Ist die Variable access false, werden ihm noch die Links zur Startseite, zum Einloggen, zum Registrieren und zu den Projekttagbuch angezeigt. Wenn der Nutzer Zugriff hat, werden ihm die Link zur Startseite, zum Profil, der Link zum Adminpanel, falls er Administrator ist, der Link zur Rangliste, zur der Seite, wo er Fragen hinzufügen kann, zum Spiel starten_Seite und der Link zum Ausloggen angezeigt. Hier endet die Div Box. Die div-Box die daraufkommt, wird 15% durch ein CSS Style nach rechts verschoben. Nun kommt die grünliche Box, welche die Überschrift Profil mit dem Paragraph „Est Quiz-Projekt von Benedikt Ross und Lukas Keller“ ist. Unter der Box ist eine horizontale Linie durch den HTM-Tag <hr>. Der Paragraph „Bitte zuerst einloggen“ mit dem Link zum Login, wird angezeigt wenn der Nutzer kein Zugriff hat. Ansonst wird in einer div-Box zentriert, ein Paragraph mit dem Text „Wähle aus einer der Themengebiete das Thema für dein Quiz“. Darauf kommt ein Zeilenumbruch und ein Formular, welches die Daten an das game.php Skript sendet. In dem Formular befindet sich der HTML-Tag fieldset. In diesem befindet sich eine

Drop-Down Auswahlliste, mit den Optionen Kunst, Bibel, Essen, Sport, Kulturen, Geschichte, Filme, Musik, Politik, Wissenschaft, TV Serien, Technologien. Nun folgt noch ein Submit Button mit dem Ende des fieldsets, des Formulars und der div-Box. Jetzt wird noch die andere div-Box, der Body und der HTML-Tag geschlossen.



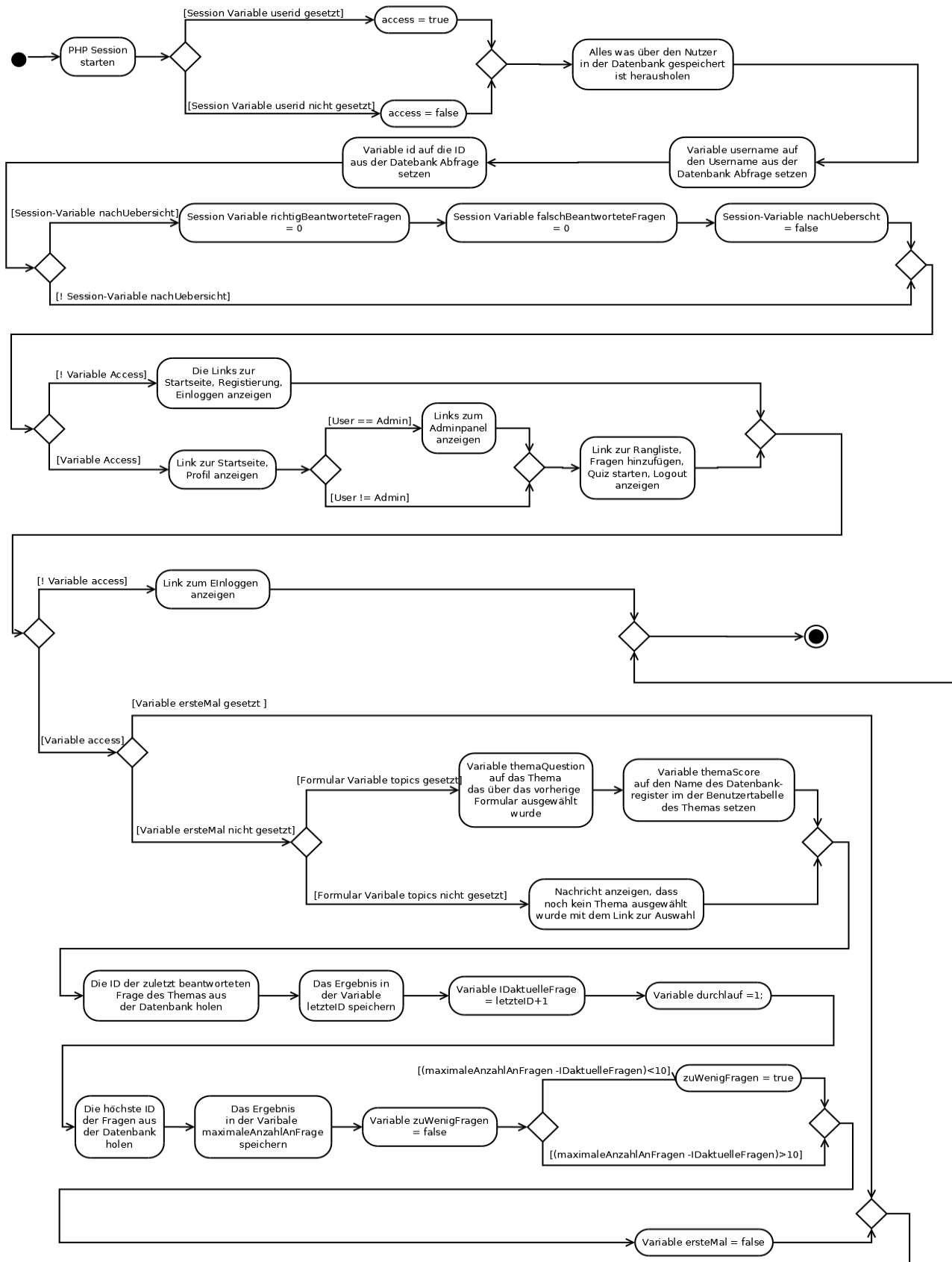
Spiel-Skript

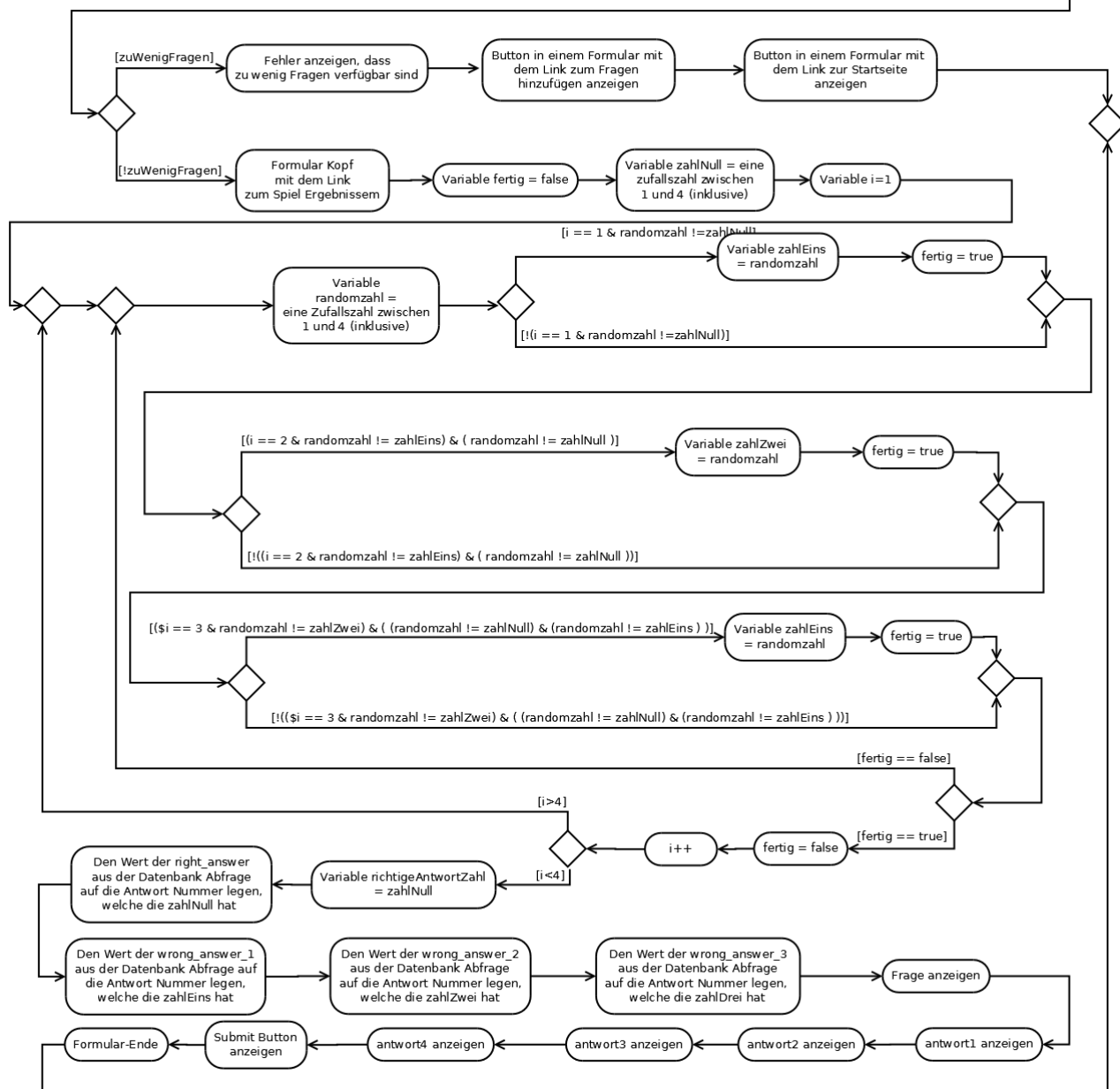
Als allererstes wird die PHP-Session gestartet. Dann wird das Skript functions.php eingebunden. Ist nun die Session-Variable userid gesetzt, wird die Variable access auf true gesetzt, ansonsten auf false. Darauf wird der Wert der Session-Variable userid der Variable userid zugewiesen. Aus der Datenbank werden alle Daten des Benutzers mit dieser ID geholt. Die Variable username bekommt den Wert des username aus der Datenbank Abfrage zugewiesen. Der Variable id wird die id aus der Datenbank Abfrage zugewiesen.

Nun wird geprüft ob die Session-Variable nachUebersicht den Wert true hat, wenn dies der Fall ist werden die Session-Variablen „richtigBeantwortetenFragen“ und „falschBeantwortetenFragen“ auf null zurückgesetzt. Dazu wird die Session-Variable „nachUebersicht“ auf false gesetzt. Dann folgt der HTML Doctype mit dem Öffnen des HTML-Tags. Im Headbereich befinden sich die Metadaten, der Titel und die Einbindung des Stylesheets. Im Bodyteil wird eine div-Box für die graue Box an der linken Seite geöffnet. Dann folgt die Überschrift „Est Quiz-Projekt“. Ist der Inhalt der Variable access false, wird den Nutzer der Link zur Startseite, den Link zum Einloggen, zur Registrieren und zum Projekttagebuch angezeigt. Ist der Inhalt true wird der Link zur Startseite, der Link zum Profil, falls der Nutzer Administrator ist, der Link zum Adminpanel, der Link zur Bestenliste, der Link zum Fragen hinzufügen und der Link zum Ausloggen angezeigt. Hier wird der Div-Tag wieder geschlossen.

Darauf folgt eine div-Box welche den Inhalt enthält geöffnet. Nun kommt eine weitere Box, welche das grünliche Feld mit der Überschrift „Spiel“ und dem Paragraph „Est Quiz-Projekt von Benedikt Ross und Lukas Keller“ enthält. Zur Trennung folgt eine horizontale Linie. Hat der Nutzer kein Zugriff, wenn der Inhalt der Variable false ist, wird eine Nachricht „Bitte zuerst Einloggen“ mit dem Link zum Einloggen angezeigt. Andernfalls wird eine Div-Tag geöffnet, dessen Inhalt zentriert angezeigt wird. Ist nun die Session-Variable ersteMal nicht gesetzt, und die Session-Variable topics auch nicht gesetzt wird ein Paragraph mit dem Text „Noch kein Thema ausgewählt. Zurück zur Auswahl“ mit dem Link zur Themenauswahl. Ansonsten wird in einem Switch-case die Formular Variable topics, welche über „post“ übermittelt wurde, überprüft welchen Wert sie enthält. So werden, je nach Thema das ausgewählt wurde, den Session-Variable themaQuestion und themaScore Werte zugewiesen. Der Session-Variable themaQuestion wird der Tabellenname des ausgewählten Themas in der Datenbank zugewiesen, der Session-Variable themaScore der Spaltenname des Thema in der Datenbank, welches die ID der zuletzt beantworteten Frage speichert. Hier endet die Else Klammer.

Als nächstes wird der MySQL Befehl, um die ID der zuletzt beantworteten Frage des Nutzer aus der Datenbank zu holen, in der Variable `abfrageLetzteId` gespeichert. Das Statement wird durch eine `foreach` Schleife ausgeführt und das Ergebnis der Abfrage in der Variable `letzteID` gespeichert.





Die Session-Variable `IDaktuelleFrage` speichert die Zahl der aktuellen Frage, ihr wird der Wert der Variable `letzteID+1` zugewiesen. Die Session-Variable `durchlauf`, speichert die Anzahl an Durchläufen, d.h. eine Frage und Antwort, wird eins zugewiesen. Darauf wird über eine Datenbank Anfrage herausgefunden welchen Wert die höchste ID der Fragen des Themas hat. Das Ergebnis wird der Session-Variable `maximaleAnzahlAnFragen` zugewiesen. Der Session-Variable `zuWenigFragen`, gibt an ob noch Fragen verfügbar sind, wird false zugewiesen. Wenn die

Subtraktion des Werts der Session-Variable `maximaleAnzahlAnFragen` und des Wertes der Session-Variable `IDaktuelleFrage` kleiner als zehn ist, wird die Session-Variable `zuWenigFragen` auf `true` gesetzt. Zum Schluss wird die Session-Variable `ersteMal` auf `false` gesetzt. Hier endet die `if`-Verzweigung.

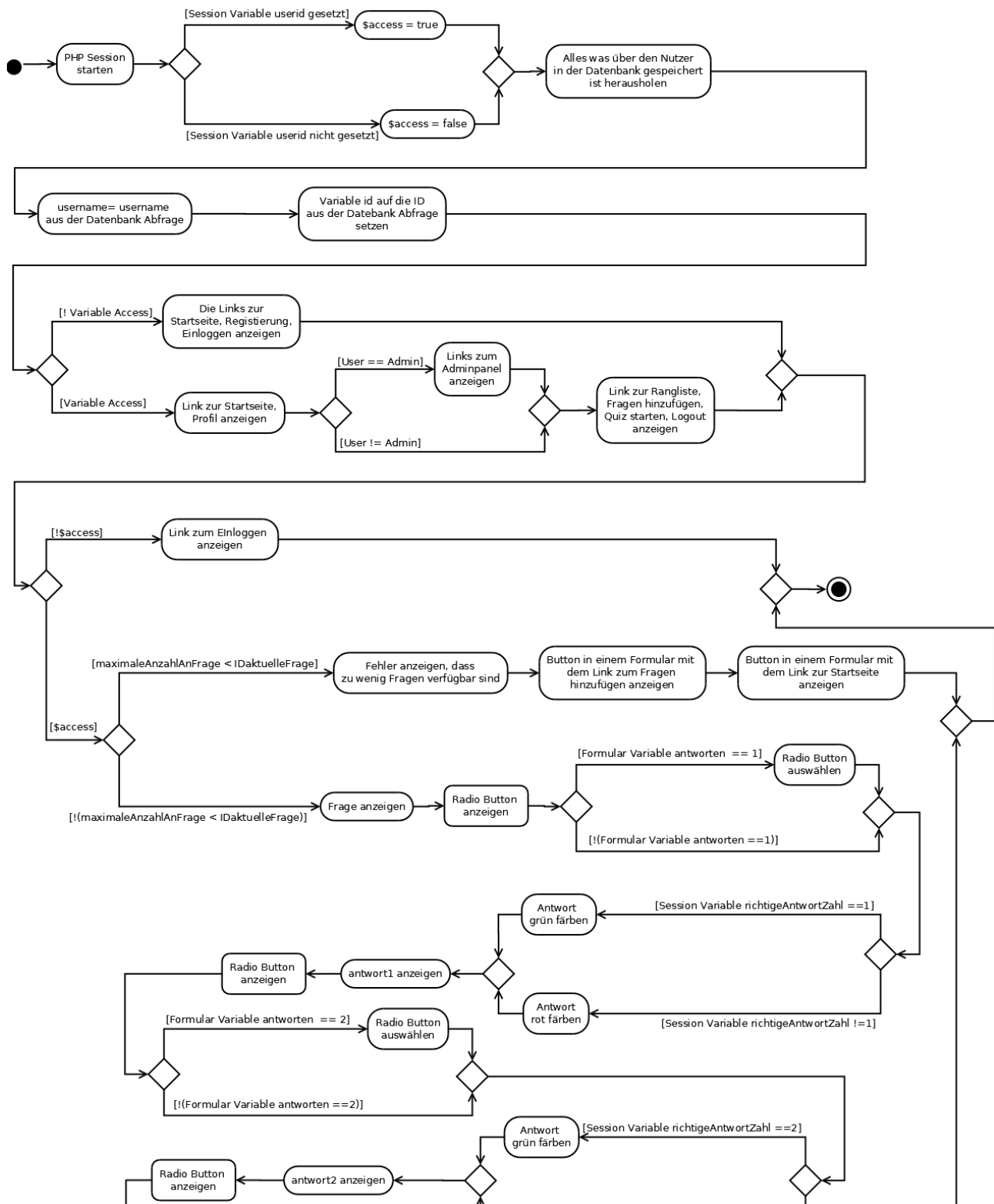
Nun wird geprüft ob die Session-Variable `zuWenigFragen` `true` ist, falls wird ein Fehler ausgegeben, welcher die Überschrift „Fehler!“ mit dem Paragraph „Es sind nicht genug Fragen in diesem Themenbereich verfügbar um eine neue Runde zu starten. Füge weitere hinzu oder spiele in einem anderen Themenbereich weiter.“ enthält. Darauf folgt ein Formular mit einem Submit Button, welcher die Aufschrift „Fragen hinzufügen“ hat und zu dieser Seite führt. Nun wird auch noch ein Formular zur Startseite mit einem Submit-Button, welcher die Aufschrift Startseite hat. Sind noch Fragen verfügbar, wird über eine MySQL Statement, welches die ID der aktuellen Frage hat und das Thema, die Frage und die Antwortmöglichkeiten aus der Datenbank holt. Die Frage wird in die Session-Variable `aktuelleFrage` geschrieben. Nun beginnt der Formular-Kopf, der zu den Spielergebnis Skript führt, über die Methode `post`. Als nächstes sollen vier Zahlen zufällig erzeugt werden, von den keine den gleichen Wert wie die andere hat. Die Zahlen sollen sich in dem Zahlenraum 1 (inklusive) bis 4 (inklusive) befinden. Nach diesen Zahlen werden dann die Antworten zugeordnet, d.h. z.B. hat die Zahl Null den Wert 2 so wird die richtige Antwort der zweiten Antwort zugewiesen. Die Antwortmöglichkeiten sind in der Reihenfolge richtige Antwort, erste falsche Antwort, zweite falsche Antwort, dritte falsche Antwort.

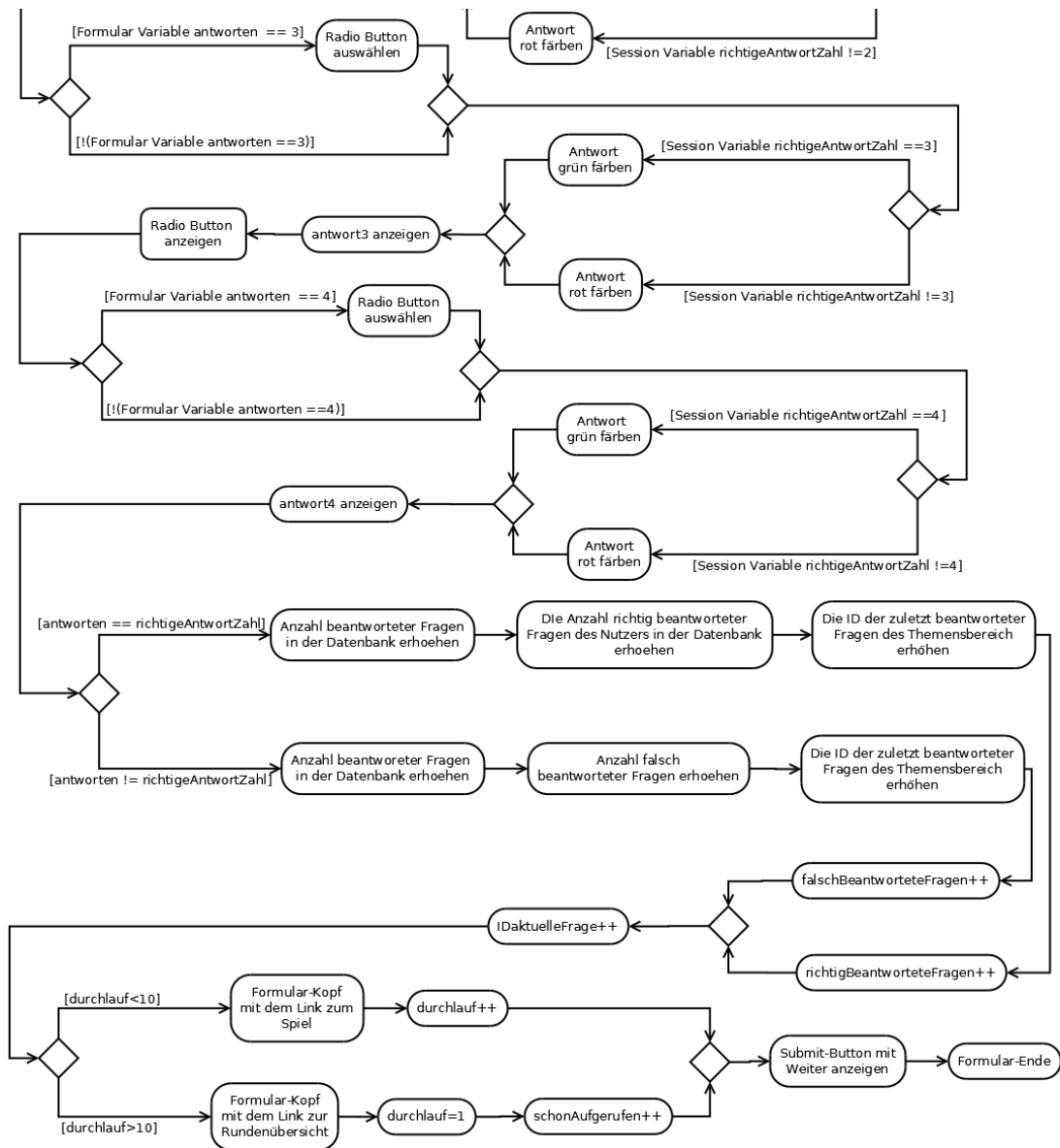
So wird als erstes der Variable fertig `false` zugewiesen. Die Variable `zahlNull` wird eine zufällige Zahl zwischen 1 (ink.) und 4 (ink.) zugewiesen, über die PHP Funktion `rand()`. Nun folgt eine `for`-Schleife, welche einen Zähler `i` hat, der zu Beginn eins ist und jede Runde um eins erhöht wird, bis `i` größer als vier ist. Darin ist eine `while`-Schleife, welche so lange durchlaufen wird bis die Variable `true` ist. Nun wird eine zufällige Zahl zwischen 1 (ink.) und 4 (ink.) durch die `rand()` Funktion erzeugt, diese wird der Variable `randomzahl` zugewiesen. Wenn die Variable `i` den Wert zwei hat und die `randomzahl` nicht den gleichen Wert der Variable `zahlNull` hat, wird der Variable `zahlEins` der Wert der Random Zahl zugewiesen und die Variable fertig auf `true` gesetzt. Ist die Variable `i` zwei und die `randomzahl` ungleich der `zahlEins`, und die `randomzahl` hat einen Wert haben der nicht der `zahlNull` entspricht, dann wird der Wert der `randomzahl` der `zahlZwei` zugewiesen. Und die Variable fertig auf `true` gesetzt. Zuletzt wird geprüft ob Variable `i` gleich drei ist und die Variable `randomzahl` ungleich der Variable `zahlZwei`, und ob der Wert der Randomzahl nicht den gleichen Wert der Variable `zahlNull` und der Wert der `randomzahl` ungleich den Wert der Variable `zahlEins` ist, wird

der zahlDrei der Wert der randomzahl zugewiesen und der Variable fertig true. Ist dies alles geprüft, wird die while Schleife wieder geprüft. Jetzt wird die Variable fertig auf false gesetzt und zurück zur for-Schleife gesprungen.

Jetzt wird die Session-Variable richtigeAntwortZahl der Wert der Variable zahlNull zugewiesen. Diese Variable enthält die Zahl bzw. die Stelle an der die richtige Antwort erscheint. Als nächstes werden die Antwortmöglichkeiten je nach dem Zahlen durch eine Funktion den Antworten antwort1, antwort2, antwort3 und antwort4 zugeordnet. Zuerst wird die richtige Antwort an die Stelle der zahlNull, dann die erste falsche Antwort an die Stelle der zahlEins, usw. zugeordnet. Nun wird die Frage in einem h1 Tag angezeigt. Die Antwortmöglichkeiten sind in einer div-Box, die in der Mitte der Seite ist, jedoch werden die Fragen linkszentriert angezeigt. Nun folgen die Antworten. Zuerst der Radio-Button, dann ein Leerzeichen, die Antwort, ein Zeilenumbruch und zum Schluss noch eine horizontale Linie. Dies geschieht für die erste Antwort, die zweite Antwort und die dritte Antwort. Bei der letzten wird keine horizontale Linie mehr angezeigt. Hier endet die div-Box und noch Submit-Button folgt noch. Dann endet das Fieldset, das Formular und die div-Boxen werden noch geschlossen. Dazu auch der Body-Tag und der HTML-Tag.

Spielauswertungs-Skript





Zuerst wird bei dem `game_result.php` die PHP Session gestartet und das Skript `function.php` eingebunden. Ist die Session-Variable `userid` gesetzt, bekommt die Variable `access` den Wert `true` zugewiesen, ansonsten `false`. Die soll feststellen ob der Nutzer eingeloggt ist. Darauf wird der Variable `userid` der Wert der Session-Variable `userid` zugewiesen. Dann wird über eine Datenbank Abfrage alles geholt, was über den Nutzer mit der ID der Variable `userid` in der Datenbank gespeichert ist. Die Variable `username` bekommt den Wert `username` des Ergebnis der Datenbank Abfrage zugewiesen. Die Variable `id` den Wert der `id` aus der Datenbank Abfrage.

Nun folgt der HTML Doctype und der HTML Tag wird geöffnet. Im darauf folgenden body Tag, der geöffnet wird, befinden sich die Metadaten, der Titel und der Link zum CSS-Stylesheet. Hier wird der Head Tag wieder geschlossen und der body tag geöffnet.

Zuerst wird die graue Seitenleiste an der linken Seite in einer div-Box definiert, sie ist 15% der Seite breit. Jetzt folgt die Überschrift „Est Quiz-Projekt“. Hat der Nutzer keinen Zugriff, werden die Links zur Startseite, zum Einloggen, zum Registrieren und zu dem Projekttagebuch. Ansonsten der Link zur Startseite, zum Profil, zur Bestenliste, zum Fragen hinzufügen, zum Spiel starten und zum Ausloggen. Ist der Nutzer Administrator wird ihm noch der Link zum Adminpanel zwischen dem Profillink und dem Bestenlistenlink angezeigt.

Nun folgt der Hauptinhalt in einer div-Box, die um 15% nach rechts verschoben ist. Zuerst wird die grünliche Box, welche oberhalb angezeigt wird definiert, in einer div-Box. Diese enthält eine Überschrift „Spiel“ mit einem Paragraph „Est-Quiz-Project von Benedikt Ross und Lukas Keller“. Die Box wird wieder geschlossen und eine horizontale Linie folgt. Hat der User jetzt kein Zugriff wird ihm die Nachricht „Bitte zuerst Einloggen“ mit dem Link zum Einloggen in einem Paragraph zentriert angezeigt. Ansonsten wird geprüft ob der Wert der Session-Variable `maximaleAnzahlAnFragen` kleiner ist als der Wert des Session-Variable `IDaktuelleFrage`. Ist es so wird eine Fehler angezeigt. Dieser Fehler ist in einer div-Box, welche rot hinterlegt ist. Der Inhalt der Box ist eine Überschrift mit dem Text „Fehler“ und ein Paragraph mit dem Text „Es sind keine weiteren Fragen in diesem Themenbereich verfügbar. Füge weiter hinzu oder spiele in einem andern Themenbereich weiter.“ Nun folgt ein Button mit der Aufschrift „Fragen hinzufügen“ in einem Fieldset, welches in einem Formular ist, das zur Seite zum Fragen hinzufügen führt. Und ein weiterer Button mit der Aufschrift „Startseite“ in einem Fieldset, welches in einem Formular ist, das zur Startseite führt.

Andernfalls wird die Frage in einem h1 Tag ausgegeben. Diese befindet sich in der Session-Variable `aktuelleFrage`. Jetzt folgt in einer div-Box die Antwortmöglichkeiten. Die Box ist der Mitte der Seite, der Inhalt aber linkszentriert. Jetzt werden der Radio-Button deaktiviert angezeigt. Hat die Formular-Variable `antworten` den Inhalt eins, wird der Radio-Button noch ausgewählt angezeigt. Diese Formular-Variable enthält die Nummer der Antwort die der Nutzer in dem `game.php` Skript ausgewählt hat. Nun folgt ein Leerzeichen und der Text aus der Session-Variable `antwort1` in einem span Tag. Ist der Wert der Session-Variable `richtigeAntwortZahl` gleich eins wird der Text durch CSS grün gefärbt, ansonsten rot. Das gleiche geschieht auch für die Antworten zwei, drei und vier. Zwischen den Antworten ist noch jeweils eine horizontale Linie.

--

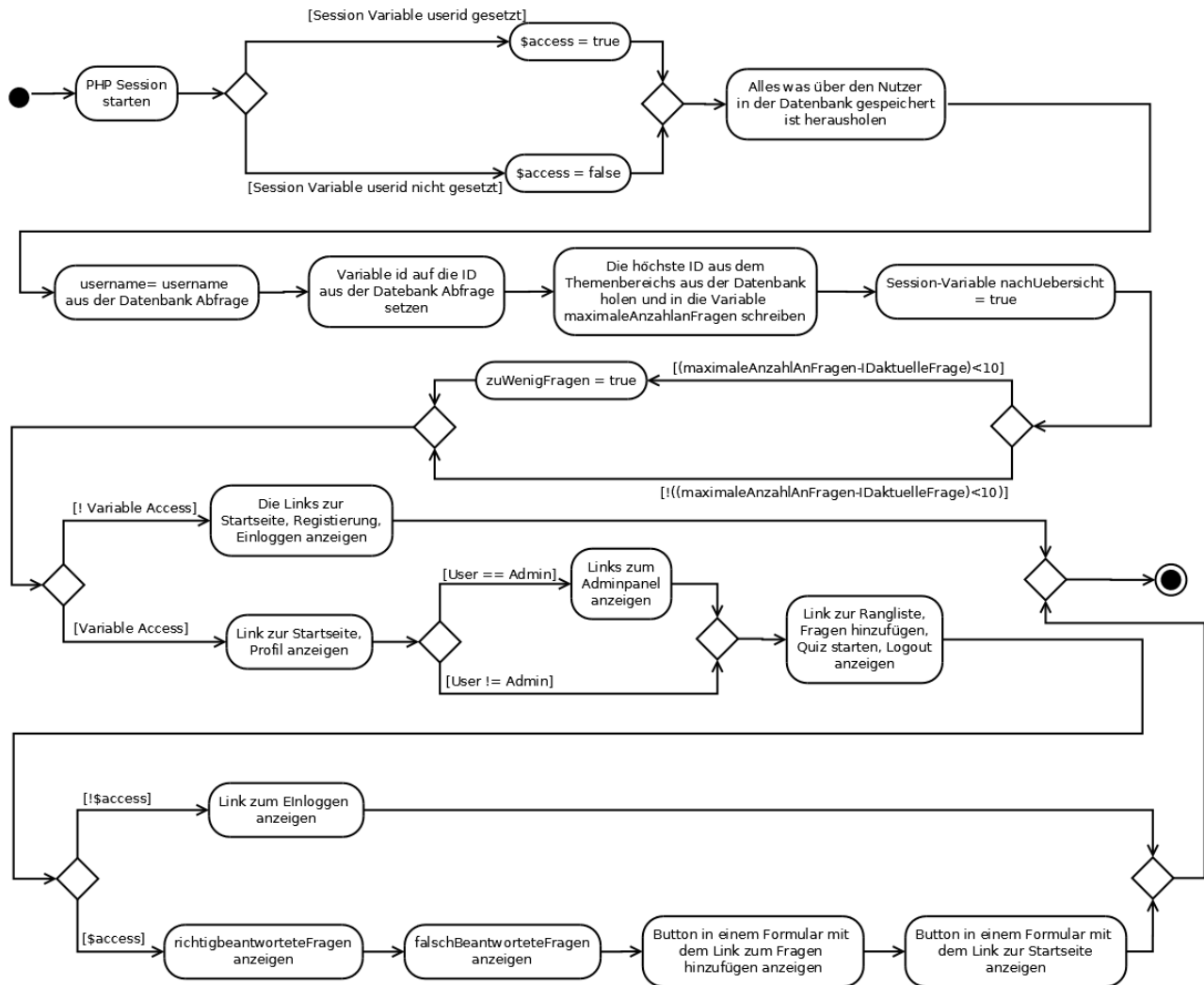
Ist der Inhalt der Formular-Variable `antworten` der gleiche wie der der Session-Variable `richtigeAntwortZahl` wird der totale Score, die Anzahl der richtig beantworteten Fragen und die

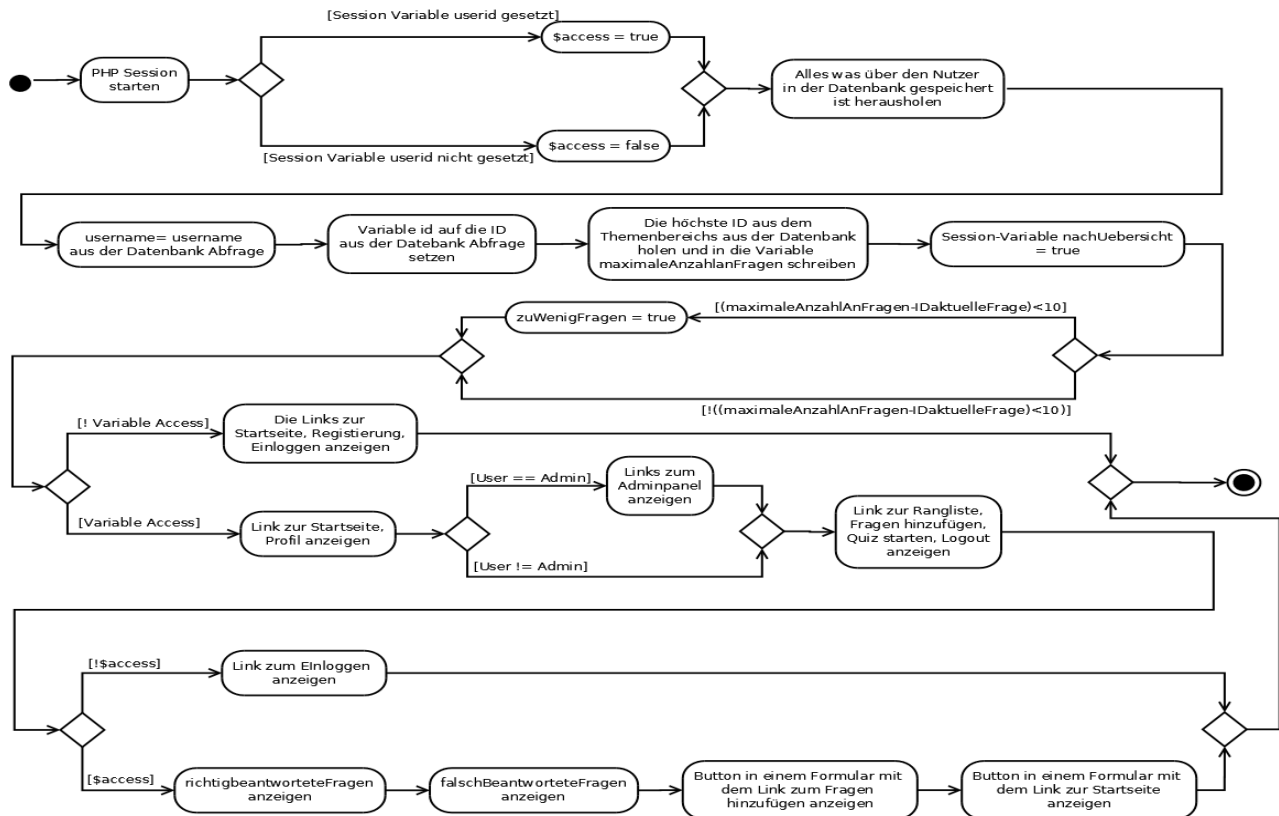
Zahl welche die zuletzt beantwortet Frage angibt in der Datenbank unter der ID des Nutzer um eins erhöht. Dazu wird die Session-Variable `richtigBeantworteteFragen` um eins erhöht. Entspricht der Inhalt der einen Session-Variable nicht der anderen, wird die Anzahl an beantworteten Fragen, die Anzahl der falsch beantworteten Fragen und die Zahl welche die zuletzt beantworteten Frage angibt in der Datenbank unter der ID des Nutzers um eins erhöht. Dazu wird die Session-Variable `falschBeantworteteFrage` um eins erhöht. Für beide wird dann wieder die Session-Variable `IDaktuelleFrage` um eins erhöht.

Ist nun der Wert der Session-Variable `durchlauf` kleiner als zehn, wird der Formular-Kopf mit dem Link zum Spiel zurückgegeben, dazu wird die Session-Variable `durchlauf` um eins erhöht.

Andernfalls wird der Formular-Kopf der zur Rundenübersicht führt zurückgegeben, dazu wird die Session-Variable `durchlauf` auf eins gesetzt. Jetzt kommt noch ein Submit Button in einem `fieldset` Tag und das Ende des Formulars. Nun werden noch die `div`-Tags, der `Body`-Tag und der `HTML`-Tag geschlossen.

Spiel-Rundenübersicht-Skript





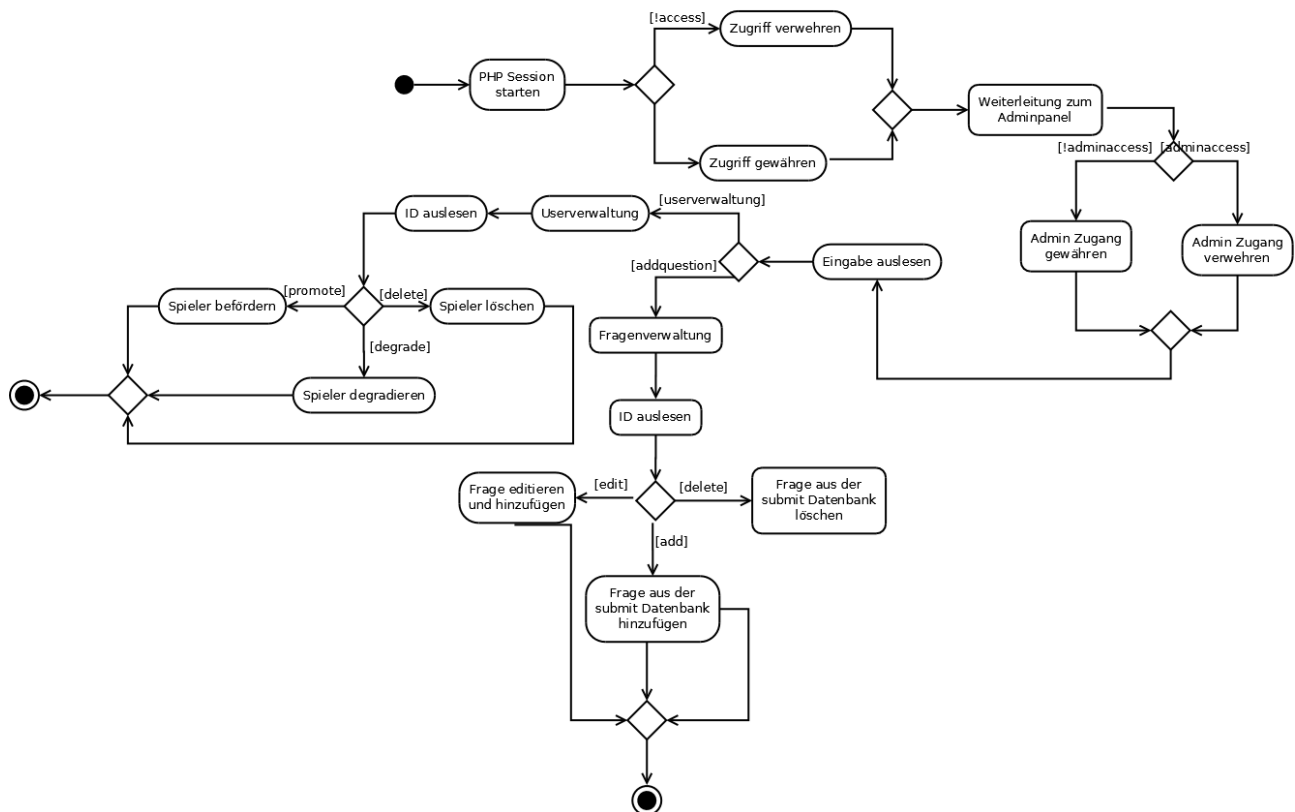
Als erstens wird wieder die PHP Session gestartet. Das Skript functions.php wird eingebunden, in welchen das PHP Data Obejct instanziiert wird. Hat der Nutzer Zugriff auf die Seite, d.h. Ist die Session-Variable userid gesetzt, wird der Variable access true zugewiesen, ansonsten false. Der Variable userid wird der Wert der Session-Variable userid zugewiesen. Nun wird alles aus der Datenbank geholt, was über den Nutzer mit der ID gespeichert ist. Dem username wird der username aus der Datenbank Abfrage zugewiesen, genauso wird der Variable id der Wert der id aus der Datenbank Abfrage zugeordnet. Nun wird die höchste ID der Fragen des Themas, welches in der Session-Variable themaQuestions gespeichert ist, aus der Datenbank geholt. Das Ergebnis wird in der Session-Variable maximaleAnzahlAnFrage gespeichert. Jetzt wird geprüft ob die Subtraktion der Session-Variable maximaleAnzahlAnFragen und die Session-Variable IDaktuelleFrage kleiner als zehn ist wird der Session-Variable zuWenigFragen der Wert true zugewiesen. Der Session-Variable nachUebersicht wird der Wert true zugewiesen.

Nach dem HTML Doctype wird der HTML Tag geöffnet. In dem darauf folgenden Head Tag befinden sich die Metadaten, den Titel und die Einbindung des CSS-Stylesheets. Jetzt wird der Head Tag geschlossen und der HTML Tag geöffnet. Zuerst wird die Sidebar definiert in einer div-Box. Diese zeigt zuerst die Überschrift „Est Quiz-Projekt“. Hat der User Zugriff, d.h. die Variable access hat der Wert false, so wird der Link zur Startseite, der Link zum Einloggen, der Link zum

Registrieren und der Link zum Projekttagebuch angezeigt. Ansonsten sieht der Nutzer die Links zur Startseite, zum Profil, wenn er Administrator ist, auch den Link zum Adminpanel, den Link zur Bestenliste, den Link zum Fragen hinzufügen, den Link zum Quiz starten, den Link zum Ausloggen. Hier wird der Div-Tag wieder geschlossen. Nun folgt eine div-Box, welche die rechte Seite, inklusive des grünlichen Feldes, enthält. Zuerst wird diese grünliche Box in einem Div-Tag definiert. Die Überschrift „Rundenübersicht“ und ein Paragraph mit dem Inhalt „Est Quiz-Projekt von Benedikt Ross und Lukas Keller“ erscheint. Nun wird der div-Box wieder geschlossen und eine horizontale Linie folgt. Wenn die Variable access den Wert false hat dann wird ein Paragraph mit dem Text „Bitte zuerst einloggen“ und dem Link zum Einloggen angezeigt. Ansonsten wird in einem blauen Feld, welches eine div-Box ist, die durch „w3-panel w3-blue“ gestylt wurde, eine Überschrift mit dem Text „Ergebnis der [Wert der Session-Variable runden] Runde“ angezeigt. Dann wird die div-Box geschlossen. Nun folgt ein Paragraph mit dem Text „Richtige Antworten“, die Zahl die nun angezeigt wird ist der Wert der Session-Variable richtigBeantwortetFragen, als Hintergrund dient ein grün gefüllter Kreis. Dann kommt noch ein Paragraph mit dem Text „Falsche Antworten“ mit dem Wert der Session-Variable falschBeantwortetFragen mit einem roten gefüllten Kreis als Hintergrund.

Der erste Button hat die Aufschrift Startseite und ist einem Fieldset tag, welches in einem Formular, das zur Startseite weiterleitet. Der zweite Button, hat die Aufschrift Weiter und ist auch in einem fieldset, welches in einem Formular, das zum Spiel-Skript weiterleitet, ist. Zum Schluss werden die Tags div, div, body, html geschlossen.

Adminpanel-Skript

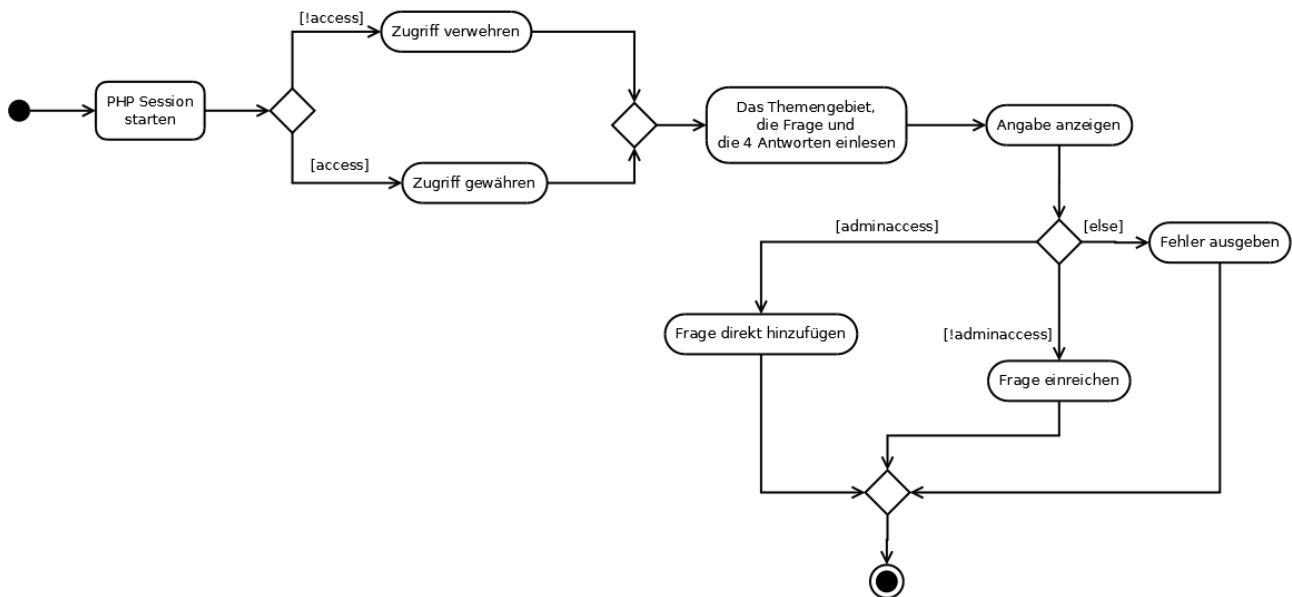


Zuerst wird eine PHP Session gestartet, dann überprüft ob man Zugang hat und dann wird noch einmal überprüft ob man als Admin eingeloggt ist. Ist dies der Fall, kann man entweder zur Userverwaltung oder zur Quiz-Fragen-Verwaltung gehen.

In der Userverwaltung hat man die Möglichkeit einen bestimmten User zu löschen, zu degradieren (zum User) oder zu befördern (zum Admin).

In der Quiz-Fragen-Verwaltung kann man eingereichte Fragen löschen, hinzufügen oder editieren und dann hinzufügen.

Fragen-hinzufügen-Skript



Zuerst wird eine PHP Session gestartet, dann überprüft ob man Zugang hat. Sollte dies der Fall sein, kann man ein Themengebiet wählen, eine Frage auswählen bzw. reinschreiben und dazu dann 4 Antwort-Möglichkeiten, wobei die erste Antwort die richtige Antwort sein muss. Hat man dies getan, wird man zur Übersicht weitergeleitet, in welcher man seine Frage, Themengebiet und Antworten sieht. Sollten alle Angaben richtig sein, kann man dies einreichen (als User) bzw. hinzufügen (als Admin), ist dies nicht der Fall hat man die Möglichkeit zurück zu gehen und seine Angaben zu ändern.

Logout-Skript



Bei diesem Skript wird die komplette Session gelöscht und somit loggt man sich aus. Das heißt man kann nicht einfach per „zurück“ Button in den eingeloggt Status wechseln.

Beschreibung der Datenbank

est-quiz

art	bible	eating	freetime	user
*id int(11)	*id int(11)	*id int(11)	*id int(11)	*id int(11)
*question longtext	*question longtext	*question longtext	*question longtext	username varchar(255)
*r_answer varchar(255)	*r_answer varchar(255)	*r_answer varchar(255)	*r_answer varchar(255)	*password varchar(255)
*f_answer_1 varchar(255)	*f_answer_1 varchar(255)	*f_answer_1 varchar(255)	*f_answer_1 varchar(255)	*email varchar(255)
*f_answer_2 varchar(255)	*f_answer_2 varchar(255)	*f_answer_2 varchar(255)	*f_answer_2 varchar(255)	*total_score int(255)
*f_answer_3 varchar(255)	*f_answer_3 varchar(255)	*f_answer_3 varchar(255)	*f_answer_3 varchar(255)	*registerdate datetime
				*score_art int(255)
				*score_bible int(255)
				*score_eating int(255)
				*score_freetime int(255)
				*score_geography int(255)
				*score_history int(255)
				*score_movies int(255)
				*score_music int(255)
				*score_nature int(255)
				*score_politics int(255)
				*score_science int(255)
				*score_technology int(255)
				*score_series int(255)
				*counter_answers int(255)
				*counter_wrong_answers int(255)
				*counter_add_questions int(255)
				*status varchar(255)

geography	history	movies	music
*id int(11)	*id int(11)	*id int(11)	*id int(11)
*question longtext	*question longtext	*question longtext	*question longtext
*r_answer varchar(255)	*r_answer varchar(255)	*r_answer varchar(255)	*r_answer varchar(255)
*f_answer_1 varchar(255)	*f_answer_1 varchar(255)	*f_answer_1 varchar(255)	*f_answer_1 varchar(255)
*f_answer_2 varchar(255)	*f_answer_2 varchar(255)	*f_answer_2 varchar(255)	*f_answer_2 varchar(255)
*f_answer_3 varchar(255)	*f_answer_3 varchar(255)	*f_answer_3 varchar(255)	*f_answer_3 varchar(255)

politics	nature	science	series
*id int(11)	*id int(11)	*id int(11)	*id int(11)
*question longtext	*question longtext	*question longtext	*question longtext
*r_answer varchar(255)	*r_answer varchar(255)	*r_answer varchar(255)	*r_answer varchar(255)
*f_answer_1 varchar(255)	*f_answer_1 varchar(255)	*f_answer_1 varchar(255)	*f_answer_1 varchar(255)
*f_answer_2 varchar(255)	*f_answer_2 varchar(255)	*f_answer_2 varchar(255)	*f_answer_2 varchar(255)
*f_answer_3 varchar(255)	*f_answer_3 varchar(255)	*f_answer_3 varchar(255)	*f_answer_3 varchar(255)

submit	technology
*id int(11)	*id int(11)
*question longtext	*question longtext
*r_answer varchar(255)	*r_answer varchar(255)
*f_answer_1 varchar(255)	*f_answer_1 varchar(255)
*f_answer_2 varchar(255)	*f_answer_2 varchar(255)
*f_answer_3 varchar(255)	*f_answer_3 varchar(255)
*topic varchar(255)	
*eingereichtid int(255)	

Die Datenbank heißt est-quiz und hat 15 Tabellen. 13 davon enthalten die Fragen mit den Antwortmöglichkeiten. Sie sind nach dem gleichen Schema aufgebaut. Zuerst kommt die id, welche auto_increment ist, d.h. sie erhöht sich automatisch sobald ein neuer Eintrag in die Tabelle geschrieben wird. Die id ist von Datentyp int und auf 11 Stellen beschränkt. Darauf folgt question, diese Spalte enthält die Frage. Question ist vom Datentyp ein Longtext. Nun folgt die right_answer, welche von Datentyp varchar ist und auf eine Länge von 255 Zeichen beschränkt ist. Nun folgen noch wrong_answer_1, wrong_answer_2, wrong_answer_3 welche nach dem gleichen Schema aufgebaut sind wie right_answer. Die verschiedenen Tabellennamen der Themen sind art, bible, eating, freetime, geography, history, movies, music, nature, politics, science, series und technology. Dann gibt es noch die Tabelle submit, sie enthält die Fragen die ein Benutzer eingereicht hat, welche jedoch noch nicht von einem Administrator in die Datenbank eingefügt wurden. Die Spalten id, question, right_answer, wrong_answer_1, wrong_answer_2, wrong_answer_3 sind gleich aufgebaut, wie der Tabellen der Themen. Noch folgen die Spalten topic und eingereichtid. Die Spalte topic ist von Datentyp varchar und auf 255 Zeichen begrenzt und die Spalte eingereichtid ist int und auf 255 Stellen begrenzt.

Zuletzt folgt noch die user Tabelle, hier werden alle Nutzerdaten gespeichert. Zuerst gibt es die Spalte id, diese ist auto_increment, der Datentyp int und die Länge auf 11 Zeichen beschränkt. Nun folgen die Spalten username, password, email diese sind alle von Datentyp varchar und auf die

Länge auf 255 Zeichen beschränkt.

Die nächste Spalte `total_score` enthält die Anzahl an Fragen die der Nutzer richtig beantwortet hat, ist von Datentyp `int` und auf die Länge von 255 Zeichen beschränkt. Die Spalte `registerdate` enthält das Datum und die Zeit, wann der Eintrag in der Datenbank erstellt wurde und hat den Datentyp `datetime`. Die folgenden Spalten `score_art`, `score_bibel`, `score_eating`, `score_freetime`, `score_geography`, `score_history`, `score_movies`, `score_music`, `score_nature`, `score_politics`, `score_science`, `score_technology`, `score_series` enthalten alle die ID der zuletzt beantworteten Frage des jeweiligen Themenbereichs. Sie haben den Datentyp `int` und sind auf die Länge von 255 beschränkt. `Counter_right_answer` enthält die Anzahl an richtig beantworteten Fragen. `Counter_answers` enthält die Anzahl an beantworteten Fragen. `Counter_wrong_answer` enthält die Anzahl an falsch beantworteten Fragen. Und `counter_add_questions` speichert die Anzahl an Fragen die der Nutzer zum Quiz hinzugefügt hat. Die Spalten `counter_right_answer`, `counter_answer`, `counter_wrong_answer`, `counter_add_questions` haben alle den Datentyp `int` und sind auf eine Länge von 255 beschränkt. Die letzte Spalte `status` ist ein `varchar` mit einer maximalen Länge von 255 Zeichen. Die Spalte hat entweder „user“ oder „admin“ gespeichert. Sie gibt den Status des Benutzers an.

Aufgabenteilung und Arbeitsprozessbericht

Die Aufgaben waren so aufgeteilt, dass Lukas die PHP Skripte schreibt und Benedikt die Datenbank erstellt. Dies stellte sich aber nach einiger Zeit als nicht vorteilhaft heraus, da der hauptsächliche Teil in PHP und HTML geschrieben werden musste. So teilten wir die Aufgaben anders auf, sodass jeder von beidem etwas machte.

Kamen neue Aufgaben, so teilten wir sie auf.

Um gemeinsam an Skripten zu arbeiten, wie z.B. den Einloggen Skript, trafen wir uns bei Benedikt Zuhause oder in der Schule.

Im späteren Verlauf kümmerte Lukas sich um das Einloggen und das Registrieren. Er erstellte zudem das Skript um Fragen hinzuzufügen, die Profilübersicht und erstellte das Adminpanel. Benedikt kümmerte sich um den Spielablauf, darunter sind die Skript Spiel starten, Spiel, die Spielauswertung und die Rundenübersicht. Wir haben lange überlegt und uns viele Gedanken gemacht, wie wir unsere Projektarbeit anfangen sollen. Die Entscheidung fiel auf eine tiefere

Einarbeitung in die Technologien HTML und PHP. Durch den Unterricht wurden uns schon Grundlagen vermittelt, diese reichten jedoch nicht immer aus. In die Datenbanken mussten wir uns von Grund auf einarbeiten. Hilfreich waren dabei Tutorials und Apps. Unser Startpunkt war die Erstellung einer Datenbank mit einem Login Skript. Dieses funktionierte dann auch nach einigem probieren und längerer Zeit. Weil wir oft nicht wussten wo das Problem lag. Um das Registrierungs-, Login-, Logout-Skript bauten wir weiter. Es folgten die Index Skripte. Nun planten wir einen Ablaufplan eines Spielvorganges. Zu Beginn hatten wir zwei Datenbanken, die erste enthielt die Fragen und die zweite die Benutzerdaten. Unser erster Designentwurf folgte nun. Zu diesem Zeitpunkt erstellten wir die Bilder Show. Unseren ersten Entwurf konnten wir nicht wie geplant umsetzen, da uns das Wissen über die Technologie JavaScript fehlte. So entschieden wir uns ein neues Design zu wählen. Dieses war insgesamt dunkler. Die nächste Neuerung waren die Bestenliste. Darauf entschieden wir die verschiedenen Kategorien einfach, mittel, schwer wegzulassen, da das Feststellen der Schwierigkeit einer Frage schwierig ist. Zuerst war unser Entwurf ein Skript zu machen, welchen den ganzen Spielablauf enthält. Jedoch änderten wir die Idee später, da es mit mehreren Skripten einfacher zu programmieren ist. Nun fügten wir den die Profilübersicht und das Adminpanel hinzu. Darauf kam die letzte große Änderung. Dies war die Änderung des Designs und Datenbank. Das Design wurde auf ein übersichtliches und schöneres Design geändert. Die Datenbanken wurden zusammengelegt, um die Koordinierung zu erleichtern. Die Skripte mit dem Spielprinzip wurden zu der Zeit fertigstellen und an das neue Design angepasst. Genau wie die Profilübersicht. Das Adminpanel wurde fertigstellen und war nun funktionstüchtig. Zu den Spielprinzip Skripten wurde nun noch die Fehlermeldung , die angezeigt wird falls zu wenig Fragen verfügbar sind, hinzugefügt. Die Skripte Bestenliste und Rundenübersicht wurden grafisch noch verbessert. Zum Schluss wurden noch kleine Fehler behoben.

Eigene Bewertung von Ablauf und Ergebnis

Der Verlauf der Arbeit lief sehr gut ab. Es gab keine Probleme in der Zusammenarbeit. Der Grund dafür war, dass jeder seine Aufgaben gewissenhaft erfüllt hat. So gab es keine Verzögerung in der Weiterarbeit des Projekts

Bei weiterem bearbeiten, ist uns aufgefallen, dass man das Spielprinzip hätte besser umsetzen können. Man könnte die Skripte Spiel starten, Spiel, Spiel Ergebnis und die Rundenübersicht in ein Skript schreiben. So könnte man mögliches schummeln verhindern.

Hinweis zur Weiterentwicklung des Produktes

Profilübersicht

- Die Profilübersicht könnte man noch so abändern, das man auch auf die Profile der anderen zugreifen kann

Mobile Version

- Anpassung der Webseite an Mobile-Geräte

User-Control-Panel

- Hinzufügung eine Control-Panels für die normalen Nutzer, in welchem sie ihren Namen, ihr Passwort, ihre E-Mail und ihren Account zurücksetzen können oder auch löschen

Chat

- Einbauen eines Chats, in welchem sich die User unterhalten können

Verwendung der E-Mail

- Einbauen einer Funktion, welche eine E-Mail an den jeweiligen Account schickt um ihn z.B. zu bestätigen oder zu benachrichtigen das er/sie z.B. alle Fragen in einem Themenbereich abgeschlossen hat

Abbildverzeichnis

Sämtliche Bilder, welche sich in dieser Dokumentation befinden, sind von uns selbst angefertigt worden bzw. Sind Screenshots der Webseite.

Quellenverzeichnis für die Dokumentation

Für die Dokumentation wurden keine externen Quellen benötigt bzw. Benutzt.

Quellenverzeichnis für das Projekt

Login sowie das Registrierungsscript: <http://www.php-einfach.de/experte/php-codebeispiele/loginscript/>

Aufgerufen am 26.04.2017 um 14:20 Uhr