Kniffel-Projekt

PHP-Projektdokumentation

**Hochschule Weserbergland**

Studiengang: Wirtschaftsinformatik

Studiengruppe: WI 44/14

Modul: Praktikum Datenbankprogrammierung

Betreuender Dozent: Till Glöggler

**Hendrik Schulze Brandhoff**

Am Westruper Bach 19

59348 Lüdinghausen

**Kevin Hesmer**

Nordring 68a

48348 Ostbevern

**Robin Haarlammert**

Moorstraße 53

49549 Ladbergen

**Yannick Retzlaff**

Im Nattkamp 20

59302 Oelde

S.

**Ausbildungsbetrieb:**

Fiducia & GAD IT AG

GAD-Straße 2-6

48163 Münster

1. Inhaltsverzeichnis

[I Inhaltsverzeichnis 1](#_Toc451719072)

[1 Vorgehen und Designentscheidungen 2](#_Toc451719073)

[1.1 Anforderungen 2](#_Toc451719074)

[1.2 Navigationskonzept 2](#_Toc451719075)

[1.3 Allgemeine Designentscheidungen 4](#_Toc451719076)

[2 Installationsanleitung 4](#_Toc451719077)

[3 Beschreibung der Datenbank 5](#_Toc451719078)

[3.1 Datenbankschema 5](#_Toc451719079)

[3.2 Normalisierung 7](#_Toc451719080)

[4 Beschreibung der Anwendung 7](#_Toc451719081)

[4.1 Der Controller index.php 7](#_Toc451719082)

[4.2 Klassenmodell 8](#_Toc451719083)

[4.3 Weitere Klassen 9](#_Toc451719084)

# Vorgehen und Designentscheidungen

## Anforderungen

Zunächst wurden die groben Anforderungen an das Kniffel-Spiel analysiert.

Laut Vorgaben des Dozenten muss das Spiel eine Nutzerverwaltung, eine Funktion zur Anzeige von Highscores und das Fortsetzen von begonnenen Spielen unterstützen. Darüber hinaus muss das Spiel mehrbenutzerfähig sein und somit einen Login unterstützen. Es sollen sich bis zu 4 Spieler an einem Gerät einloggen können, um gegeneinander zu kniffeln. Darüber hinaus sind grundlegende Sicherheitsmechanismen zu implementieren, die die Anwendung vor Session-Hijacking, SQL-Injection und XSS schützen.

## Navigationskonzept

Nach der Definition der Anforderungen wurden Pläne für den Programmablauf entwickelt. Die Startseite besteht lediglich aus einem Button zum Start eines neuen Spiels, einem Button zur Fortsetzung eines begonnenen Spiels und einem Button um zur Benutzerverwaltung zu gelangen. Zusätzlich wird auf der Startseite bereits ein Ranking angezeigt und es gibt einen Button, um zu einer Seite mit weiteren Spielstatistiken zu gelangen.

Hat der Nutzer den Button zum Start eines neuen Spiels gedrückt, gelangt er auf die Login-Seite, auf der sich bis zu vier Spieler für ein Spiel einloggen können.

Entscheidet sich der Nutzer dazu, ein begonnenes Spiel fortzusetzen, bekommt er eine Liste mit sämtlichen nicht beendeten Spielen, von denen er sich ein Spiel mithilfe von Radiobuttons auswählen kann. Um ein Spiel besser wiederfinden zu können, kann er in dieser Ansicht die angezeigten Spiele nach den vorhandenen Attributen filtern. Hat der Nutzer ein Spiel ausgewählt, gelangt er auf die Login-Seite, auf der sich alle Spieler dieses angefangenen Spiels nacheinander anmelden müssen.

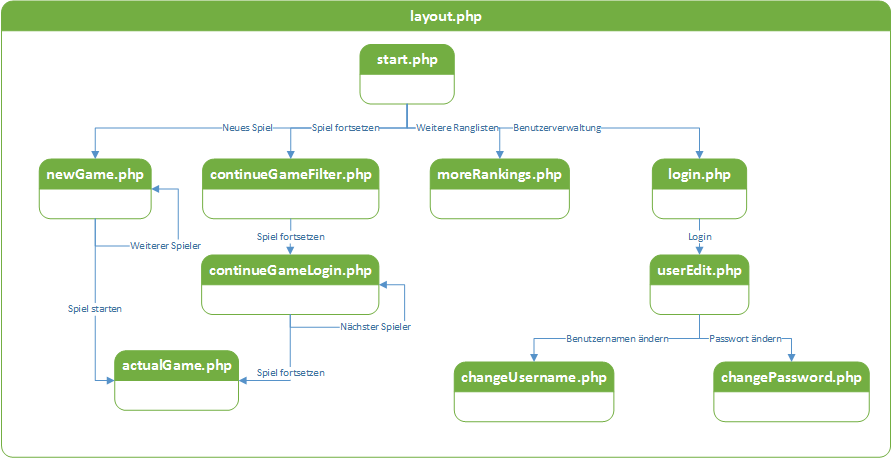
Möchte der Spieler die Nutzerverwaltung öffnen, öffnet sich zunächst eine Login-Seite, auf der er sich authentifizieren muss.

Auf der Login-Seite für die Anmeldung zum Spiel kann man über einen entsprechenden Button einen weiteren Spieler hinzufügen, das Spiel starten, oder Abbrechen und zurück zur Startseite gelangen. Außerdem kann man die Check-Box "Account erstellen" auswählen, um einen neuen Nutzer in der Datenbank anzulegen. Ein neuer Benutzer ist automatisch für das Spiel angemeldet.

Ist der Spieler von der Spiel-Fortsetzen-Seite zur Login-Seite gelangt, sind die Namen der Anzumeldenden Spieler bereits in den entsprechenden Feldern vorausgewählt. Außerdem kann man hier keine neuen Nutzer erstellen.

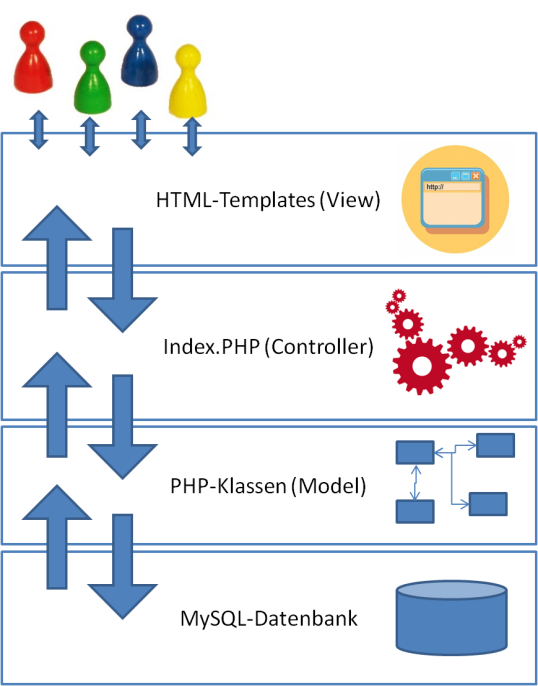
Mit dem Klick auf "Spiel starten" gelangt der Spieler auf die Seite um das Spiel zu spielen.

Die Spielseite besteht grundlegend aus 2 Bereichen. Die linke Hälfte der Seite zeigt die aus dem Originalspiel bekannte Kniffel-Tabelle. Entsprechende Buttons sorgen hier dafür, dass der Spieler auswählen kann, in welches Feld er seine Punkte eintragen möchte. Die rechte Hälfte der Seite soll später zur Darstellung der Würfelvorgänge dienen und entsprechende Eingabeelemente (voraussichtlich Buttons) zur Verfügung stellen.

Auf allen Seiten sind Buttons implementiert, mit dem man eine gestartete Spielsession beenden, und auf die Startseite zurück gelangen kann.

Die Abbildung beschreibt die Navigation und Aufrufreihenfolge der Templates. Beim rendern eines Templates wird immer das Template "layout.php" gerendert, und das aufzurufende Template inkludiert.

## Allgemeine Designentscheidungen

Es wurde entschieden, dass für das Kniffel-Projekt das Model-View-Controller-Entwurfsmuster (MVC) umgesetzt werden soll.

Nach der Entwicklung der groben Oberflächenstruktur erfolgte die Entwicklung des Datenbankentwurfs und die Umsetzung der Datenbank mittels MySQL. Genauere Merkmale des Datenbankdesigns finden Sie im Kapitel "Beschreibung der Datenbank. Außerdem wurden einzelne Templates erstellt, die zur Anzeige der Anwendungsinhalte dienen.

Im Anschluss daran wurde das Kern-Klassenmodell erstellt, mit dem es möglich ist, ein Kniffel-Spiel abzuwickeln. Details hierzu sind im Kapitel "Beschreibung der Anwendung" zu finden.

Da bei diesem Spiel mehrere Nutzer gleichzeitig an einem Endgerät gegeneinander spielen, unterscheidet sich die Anwendung von einer klassischen Sitzungsverwaltung. Eine Sitzung wird nicht durch die Authentifizierung eines Spielers, sondern durch die Authentifizierung mehrerer Spieler gestartet. Eine Sitzung ist also ein gestartetes Spiel mit mehreren authentifizierten Spielern auf einem Gerät.

# Installationsanleitung

Zur Installation der Anwendung werden eine lauffähige Webumgebung (bspw. XAMPP) und ein MySQL-Datenbankserver benötigt. Um auf die verwendete Datenbank zugreifen zu können muss in der PHP-File "config.php" in der Variable "$dbh" die Verbindung zur Datenbank definiert werden. Die Erstellung der Datenbanktabellen ist in der beiliegenden Datei "createDatabase.sql" dokumentiert. Diese enthält zusätzlich einige Standardeinträge, welche in die Datenbank geschrieben werden. Die SQL-Statements aus der Datei müssen ausgeführt werden.

Darüber hinaus muss in den PHP-Einstellungen die Option der "short\_open\_tags" aktiviert werden, bevor losgespielt werden kann.

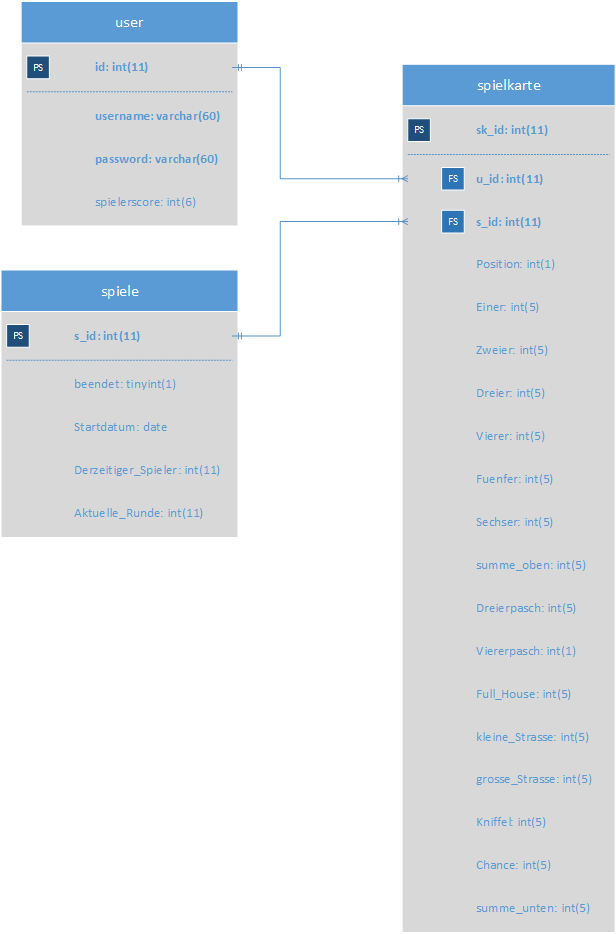
Das Spiel wird über die "index.php" gestartet. Die Anwendung ist kompatibel mit den Browsern Chrome und Firefox, wobei im Firefox beim Suchen nach gestarteten Spielen keine Datumsauswahl über einen Date-Picker möglich ist und keine Scrolleiste zur Verfügung steht.

Bevor das Spielen begonnen werden kann, muss mindestens ein eigener Spieler angelegt werden. Da dieser ein individuelles Passwort benötigt, das als Hashwert gespeichert wird, ist es uns nicht möglich diesen per SQL-Insert vorzudefinieren. Aus diesem Grund gibt es auch keine weiteren Testdaten. Diese müssen also erst erspielt werden.

# Beschreibung der Datenbank

## Datenbankschema

Die Folgende Darstellung zeigt den strukturierten Aufbau der Datenbank.



Die Datenbank ermöglicht eine Speicherung der Spieler mit ihrem Namen und dem Passwort in der Tabelle "user". Hier wird zudem ein spielübergreifender, langfristiger Spielerscore festgehalten. In der Tabelle "spiel" werden für ein Spiel folgende Informationen gespeichert: Information über den Status des Spiels (beendet oder nicht); das Startdatum des Spiels um das Spiel als Nutzer einfacher wiederfinden zu können; der Spieler der bei einem unbeendeten Spiel aktuell an der Reihe ist mit Würfeln; und zuletzt die aktuelle Spielrunde, um das Spiel an dieser Stelle fortsetzen zu können. Über die Spielkarte können User dem Spiel zugeordnet werden und umgekehrt. Ein User kann mehrere Spielkarten haben, und so an mehreren Spielen beteiligt sein. Einem Spiel können mehrere Spielkarten (und somit auch Spieler) zugehören. Eine Spielkarte ist aber immer eindeutig einem Spieler und einem Spiel zugeordnet. Der Fall, dass ein Spiel aus maximal 4 Spielern bestehen kann, ist nicht durch die Datenbank-, sondern durch die Anwendungslogik geregelt, um dafür zu sorgen.

## Normalisierung

Der Datenbankentwurf wurde zunächst in der dritten Normalform entwickelt. Bei der Einbindung der Datenbank in das Programm ist schließlich aufgefallen, dass gewisse Abfragen sehr Umfangreich und Aufwendig sind. Aus diesem Grund wurden einige Datenbankfelder eingefügt, die gegen die dritte Normalform verstoßen. So verstoßen die Felder "summe\_oben" und "summe\_unten", "spielerscore", "beendet", "derzeitiger\_Spieler" und "aktuelle\_Runde" formell gegen das Verbot von transitiven Abhängigkeiten, da diese Attribute sich aus anderen Nichtschlüsselattributen (oder aus dem Kontext Mehrerer) ableiten lassen.

# Beschreibung der Anwendung

## Der Controller index.php

Für die Ausführung der Anwendung ist der Controller verantwortlich. Dieser Controller wird durch die "index.php" repräsentiert. Beim Aufruf dieses Controllers wird die "config.php" aufgerufen. Für jede unterschiedliche Aktion in der Anwendung (Button-Klicks) ist in der "index.php" ein Programmablauf beschrieben, der bei einer Aktion ausgeführt wird. Zur Seitenanzeige wird in diesen Programmabläufen die statische Funktion "render($template\_path, $data)" der Klasse "Template" aufgerufen, die die übergebene Seite rendert und mit den übergebenen Daten versorgt. Die "index.php" hat ein Array "$template\_data", das dieser Methode immer übergeben wird. Vor dem aufruf der Render-Funktion müssen der "$template\_data" folglich alle für die Anzeige benötigten Daten hinzugefügt werden.

Um ein Spiel während der Gesamten Sitzung persistent halten zu können, wird es in der globalen Variable "$\_SESSION" gespeichert. Allerdings wird dieses Spiel-Objekt nicht etwa erst bei Klick auf "Spiel starten" erzeugt, sondern bereits bei Start der Anwendung. Dies ist notwendig, da unter anderem der Login von bis zu 4 Spielern vor dem eigentlichen Spielstart erfolgt, und die eingeloggten Spieler der Spiel-Instanz hinzugefügt werden müssen.

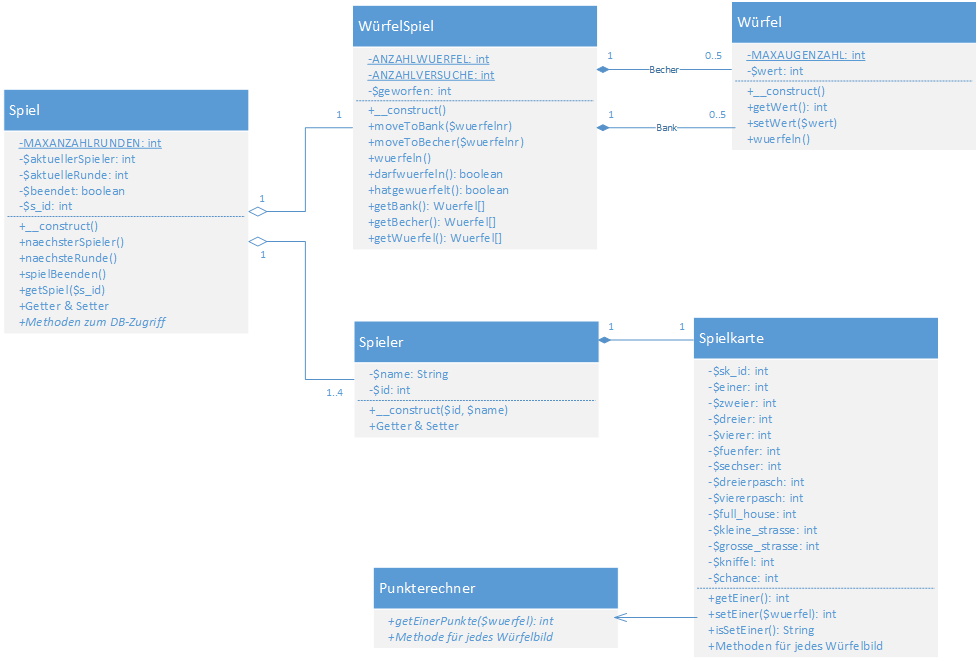
## Klassenmodell

Um die Kernfunktionalität unabhängig von Oberfläche und Datenbank nach dem MVC-Konzept zu implementieren, wurde ein Klassenkonzept entworfen, mit dem es theoretisch möglich ist, ein Spiel abzuwickeln bzw. zu simulieren.

Die Hauptklasse dieses Kerns ist die Klasse "Spiel". Von hier aus wird das aktuelle Würfelspiel und die Spielkarten der Spieler verwaltet. Ein Spiel stellt für jeden Würfeldurchgang eine Instanz der Klasse "WürfelSpiel" bereit. Ein Würfelspiel besteht aus 5 Würfeln, die entweder im Array "$becher" oder im Array "$bank" liegen können. Ein Würfel hat lediglich einen "$wert" und stellt eine Funktion zum setzen eines zufälligen Wertes bereit. Wird die Funktion "wuerfeln()" der Klasse "WürfelSpiel" aufgerufen, werden für alle Würfel, die im Array "$becher" liegen neue Werte ermittelt. In einer Instanz von "WürfelSpiel" kann maximal 3 mal (bzw. Wert der Konstante "ANZAHLVERSUCHE") gewürfelt werden. Soll die Punktzahl eines Würfelbildes in der Spielkarte eines Spielers eingetragen werden, ist beim Spielkarten-Objekt des aktuellen Spielers (Attribut "$aktuellerSpieler" in "Spiel") die entsprechende Funktion (zum Beispiel "setEiner($wuerfel)") aufzurufen. Dieser Methode wird eine Array mit Würfeln (Becher + Bank) übergeben, die die Funktion "getWuerfel()" der Klasse "WürfelSpiel" zurückliefert.

Da die Ermittlung der Punktzahl aus einem Würfelbild mitunter sehr kompliziert ist, sind entsprechende Funktionen statisch in die Klasse Punkterechner ausgelagert. Für komplizierte Würfelbilder (zum Beispiel Straße oder Full House) wird hier auch eine Prüfung vorgenommen, ob das Würfelbild überhaupt gewürfelt wurde.

Neben der Kernfunktionalitäten der Klassen beinhalten diese auch Funktionen, um die Daten auch in der Datenbank zu persistieren oder aus der Datenbank zu lesen. Diese sind im nachfolgenden Klassendiagramm aus Gründen der Übersichtlichkeit, genau wie Getter- und Setter-Funktionen, nicht aufgeführt.



## Weitere Klassen

Neben dem eigentlichen Spiel bietet die Anwendung weitere Funktionalitäten, zu denen die Benutzerverwaltung und ein Spielerranking zählen.

Für diese Funktionalitäten werden weitere Klassen ("Benutzer" und "Ranking") für den Zugriff auf die Datenbank benötigt, da die Funktionalitäten mit der eigentlichen Hauptfunktionalität, dem Spielen, nicht direkt etwas zu tun haben.

Diese Klassen sind von dem Spielkern entkoppelt und müssen nicht instanziiert werden und stellen deshalb nur statische Funktionen zur Verfügung.