

Dokumentation

„Ultimate Vong Memes“

Markus Gatz, Jorma Bork, Björn Hoppe-Kossak

Dokumentation

„Ultimate Vong Memes“

Markus Gatz, Jorma Bork, Björn Hoppe-Kossak

[illegible]

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
1 Grundlagen.....	3
2 Beschreibung der Idee	4
3 Relevante Klassen der Umsetzung	6
3.1 dbconn.php	6
3.2 index.php	6
3.3 image.php	7
3.4 meme.php.....	8
3.5 uploadMeme.php.....	8
3.6 register.php.....	9
3.7 package.json.....	9
3.8 login.php	9
3.9 logout.php.....	9

1 Grundlagen

Im Internet auftretende Hypes um Bilder, Videos und Musikdateien werden immer populärer. Vor Allem die Verbreitung von Bildern ist schon seit mehreren Jahren ein wichtiger Bestandteil von Internetphänomenen. Eine Unterform dieser Internetphänomene sind die sogenannten Memes. Memes sind meistens Bilder, deren Bedeutung durch einen Text verdeutlicht wird. Dieser befindet sich im oberen und unteren Bereich des Bildes. Entweder verdeutlicht er die Bedeutung des Bildes, fügt dem Bild eine Bedeutung hinzu oder verändert diese vollständig. Häufig entstammen diese Bilder Serien, Filmen, Cartoon-Neuschöpfungen oder sind einfach im Internet besonders populär. Es gibt bereits Seiten, auf denen Memes erstellt werden können. Diese bestehen aus einem simplen Editor, in dem durch den Nutzer hochgeladene Bilder bearbeitet werden können.

2 Beschreibung der Idee

Unsere Idee ist es, an bereits bestehenden Seiten zur Erstellung von Memes anzuknüpfen. Hier können eigene Bilder hochgeladen und bearbeitet werden. Diese können auch anderen Nutzern zur Verfügung gestellt werden. Die Idee wird als eine responsive Browseranwendung umgesetzt. Dabei wird zwischen zwei Bereichen unterschieden, einem frei zugänglichen und einem geschützten. Der frei zugängliche Bereich lässt den Nutzer Beispiele ansehen, sich registrieren oder, falls bereits registriert, ein Login durchführen. Für die Registrierung müssen ein Passwort und eine Emailadresse angegeben werden. Diese sind bei einem Login einzugeben, wie in folgendem Use-Case dargestellt.

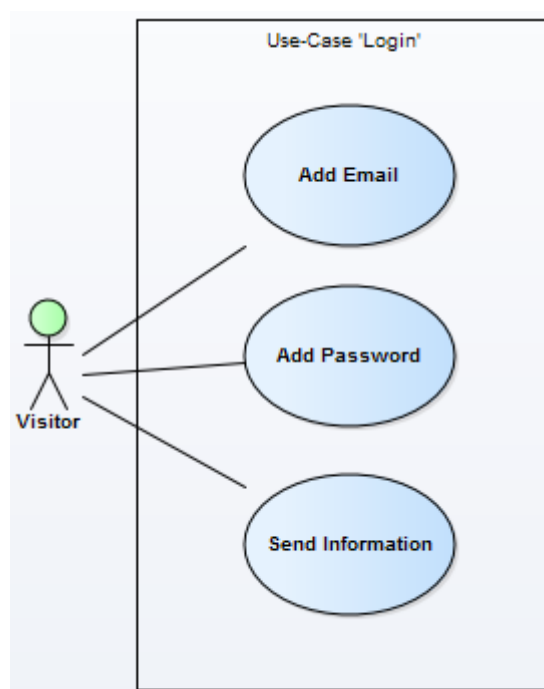


Abbildung 1 - Use-Case Login

Wird ein Login durchgeführt, wechselt der Nutzer in den geschützten Bereich. In diesem ergibt sich die Möglichkeit, erstellte Memes hochzuladen und diese in einer eigenen Sammlung abzuspeichern.

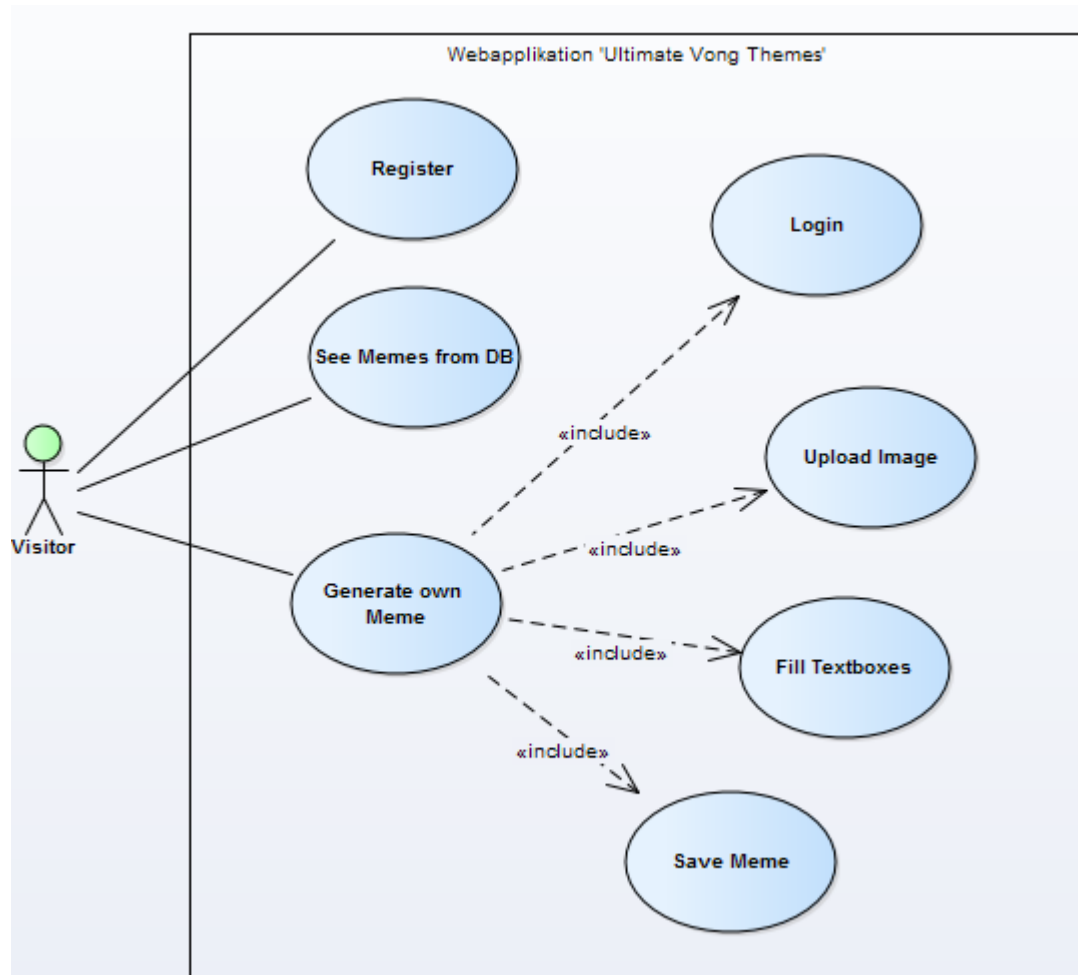


Abbildung 2 - Webapplikation "Ultimate Vong Memes"

3 Relevante Klassen der Umsetzung

Die folgenden Klassen stellen das Gerüst für die Funktionalitäten zur Verfügung.

3.1 dbconn.php

Essentiell um Datenbankabfragen in MySQL zu starten. Zuerst findet eine Initialisierung und Deklaration von notwendigen Variablen statt. Für ein besseres Fehlerhandling ist der eigentlichen Verbindung mit der Datenbank ein TryCatch-Block vorangestellt. Es wird eine PDO-Verbindung mit der Datenbank hergestellt.

3.2 index.php

Diese Klasse beinhaltet den Start der Session. Was bedeutet, dass an dieser Stelle eine Methode ausgeführt wird. Sie greift auf die dbconn.php zu, um die nötige Datenbankverbindung herzustellen. Über einen if-Block wird festgestellt, ob es sich um einen bereits registrierten, angemeldeten Nutzer handelt. Die Anzeige eines Logins/Logouts in der Section „Login“ beeinflusst. Es wird innerhalb der Klasse der CSS für ein Bootstrap-Template geladen, der für die Darstellung der Seite verantwortlich ist. Daraufhin werden individuelle Schriftarten für das Template eingestellt. Ein weiterer CSS wird verwendet, um einen sogenannten SlickSlider und Sweet-Alert-Notifications bereitzustellen. Darauf folgen individuelle Einstellungen für den SlickSlider, welche Aussehen und Verhalten der einzelnen Bilder und Richtungspfeile betreffen. Der folgende Funktionsblock gilt für die Navigation. Hier werden die Login/Logout Elemente verwendet, die sich je nach

Login-Status des Nutzers verändern. Inhalte der Frontpage werden festgelegt, darunter ein Einstiegstext, worum es sich bei Memes handelt und Bilder die als Beispiele für Memes verwendet werden. Ein weiterer Funktionsblock befasst sich mit dem Registrierungsformular, mit welchem sich ein Nutzer für die Seite registrieren kann. Weiter werden Javascript-Dateien aufgerufen, die für den JQuery, den Slider, den Tether nötig sind sowie ein Bootstrap-JS für Funktionen des Bootstrap Moduls. Die weiteren aufgerufenen JS-Dateien sorgen für die Validierung der Login und Registrierungsdaten und AJAX um die Übermittlung der Login und Registrierungsdaten aus den Formularen sicherzustellen. Die Einbettung von jquery-easing, scrollreveal und magnific-popup sind teilweise für die Funktion des „Creative Bootstrap Templates“ notwendig. Zusatzfeatures werden gegeben, etwa SmoothScrolling. Außerdem bietet die SweetAlert2.min.js optisch schönere Meldungen. Ein großer Funktionsblock beinhaltet individuelle Parameter für die Einstellungen des SlickSliders. Die Bilder-Liste wird durch AJAX zur Verfügung gestellt und dem Slider übergeben. Der letzte Funktionsblock dient dem Ein- und Ausblenden der Elemente des Logins und der Registrierung.

3.3 image.php

Diese Klasse ist für die Löschung und Übermittlung von Informationen der Memes verantwortlich. Es erfolgt eine Prüfung, ob der passende Dateiname der angefragten ID des Memes in der Datenbank existiert und gibt diesen zurück. Wird ein Dateiname ausgegeben, wird die physisch vorhandene Datei gelöscht und der Eintrag aus der Datenbank entfernt. Rückgabe über Erfolg durch JSON. Weiterer Funktionsblock zur Abfrage der Informationen aller Memes, also der Bilder, aus der Datenbank und Rückgabe durch JSON.

3.4 `meme.php`

Zuerst erfolgt eine Prüfung, ob der Nutzer eingeloggt ist. Es werden individuelle Schriftarten für das Template und den MemeGenerator bereitgestellt. Nach grundsätzlichen Einstellungen, die auch in der `index.php` vorhanden sind, wird der eigentliche MemeGenerator bereitgestellt. Der Funktionsblock wird bei Click auf den Upload-Button gestartet. Dadurch wird das aktuelle Canvas-Element eingelesen. Eine `dataURI` wird zurückgegeben, zur Repräsentation des Bildes. Der eingegebene Text wird an die Klasse `uploadMeme.php` weitergegeben. Die Funktion `readMemes` wird ausgeführt. Eine Funktion liest die eingegebene Überschrift und den Untertitel. Danach wird das Meme gezeichnet. Die Bildskalierung wird eingestellt und Position der Buchstaben auf dem Canvas und die Schriftgröße festgelegt. Das Bild kann durch das `FileSaver.js` gespeichert werden.

3.5 `uploadMeme.php`

Diese Klasse sorgt für das Hochladen der Memes. Dazu muss über die `dbconn.php` bestehen. Es muss eine Session bestehen. Über den Upload-Button der `meme.php` ist ein Datenarray übermittelt worden, in welchem sich Image und Memetext befinden. Die Image-Datei wird zunächst auf dem Server gespeichert. Das Bild wird im Ordner „images/“ abgelegt. Der Dateiname setzt sich aus einer Kombination aus Jahr, Monat, Tag, Stunden, Minuten und Sekunden zusammen, dadurch werden doppelte Einträge weitestgehend ausgeschlossen. Filename und SessionID werden gespeichert, da diese Informationen benötigt werden, um den Dateinamen und den jeweiligen User in der Tabelle Memes zu hinterlegen. Diese Hinterlegung wird durch einen TryCatch-Block bei Fehlern aufgefangen.

3.6 register.php

Übermittelte Daten werden hier empfangen und es wird geprüft, ob die angegebene Email-Adresse bereits vorhanden ist. Falls nicht kann die Registrierung durchgeführt werden. Andernfalls wird eine Fehlermeldung ausgegeben, je nachdem, ob die Emailadresse bereits existiert oder ein anderer Fehler aufgetreten ist. Dies sind zwei verschiedene Fehlermeldungen.

3.7 package.json

Beinhaltet Informationen zu Repositories und Dependencies.

3.8 login.php

Empfängt Daten bei Login und überprüft diese. Zuerst wird überprüft, ob die Emailadresse existiert. Falls ja wird das Passwort überprüft. Sind eingegebene Daten inkorrekt, wird ein Fehler über nicht existierende Daten geworfen.

3.9 logout.php

Mit der Ausführung wird der angemeldete Nutzer abgemeldet. Die Session wird gelöscht und alle damit in Verbindung stehenden Daten verworfen. Der Nutzer wird auf die Startseite zurückgeleitet.