Projektdokumentation

Encuesta – Die einfach starke Umfrage

Internet-Technologien Prof. Dr. Knut Barghorn, Christian Kuka

Marvin Friedemann, Lars Stuhr, Stefan Willenborg, Björn Kohnen

Inhaltsverzeichnis

[Programmkenndaten Lars 2](#_Toc289010756)

[Programmidentifizierung 2](#_Toc289010757)

[Programmkurzbeschreibung 2](#_Toc289010758)

[Aufgabe des Programms 2](#_Toc289010759)

[Programmbedarf 2](#_Toc289010760)

[Programmiersprachen / Compiler 2](#_Toc289010761)

[Datenorganisation 2](#_Toc289010762)

[Programmfunktion Stefan 3](#_Toc289010763)

[Aufgabenbeschreibung 3](#_Toc289010764)

[Aufgabenlösung 3](#_Toc289010765)

[Funktionshierarchie 3](#_Toc289010766)

[Methoden/Algorithmen 3](#_Toc289010767)

[Fehlerbehandlung 3](#_Toc289010768)

[Programmaufbau 3](#_Toc289010769)

[Programmbausteine 3](#_Toc289010770)

[Programmablauf 6](#_Toc289010771)

[Programmablaufbeschreibung 6](#_Toc289010772)

[Dialogfolgeplan 7](#_Toc289010773)

[Dialogablaufplan 9](#_Toc289010774)

[Dialogbeschreibung Lars 19](#_Toc289010775)

[Datenorganisation 20](#_Toc289010776)

[Eingabedaten 20](#_Toc289010777)

[Ausgabedaten 21](#_Toc289010778)

[Datenbankanwendung 21](#_Toc289010779)

[Datenbankschema 21](#_Toc289010780)

[Nutzerverwaltung 21](#_Toc289010781)

[Umfrageverwaltung 22](#_Toc289010782)

[Speicherung der Umfrageergebnisse 22](#_Toc289010783)

[Anwendungsbeispiel Stefan 22](#_Toc289010784)

# Formulierung des Ziel / der Aufgabenstellung

# Beschreibung der Herangehensweise

# Darstellung des Ergebnisses

# Fazit (Ziel/Ziele erreicht?)

# Ausblick (Was muss noch bearbeitet / umgesetzt werden, wie könnte eine Fortführung des Projektes aussehen?)

# Sie sollten mit ca. 20 Seiten (weniger kann auch mehr sein) auskommen. Quellcode und eventuelle weitere detaillierte Darstellungen gehören in den Anhang (der nicht zu den Seitenzahlen zählt).

# Programmkenndaten Lars

Im Rahmen der Veranstaltung Internet Technologien war es unsere Aufgabe als Gruppe ein Umfragesystem zu entwickeln. Eine der Anforderungen war die Möglichkeit, zwischen Single/Multiple Choice-Antworten auswählen zu können. Außerdem sollte es möglich sein diese Fragen und Antworten in einer Datenbank zu speichern. Auf diesen Grundlagen kann dann eine Auswertung der Daten erfolgen, die auch eine Auswertung von verschiedenen Fragetypen beinhaltet.

Das Umfragesystem Encuesta unterstützt all diese Anforderungen und unterstützt den Benutzer dabei eine Umfrage zu erstellen, zu bearbeiten, frei zu geben und zu löschen.

Zur Ideensammlung wurde die Systeme „Doodle“ und „Askailo“ betrachtet und durch diesen Vergleich entstand das Programm Encuesta.

Programmidentifizierung

Das Umfragesystem trägt den Namen „Encuesta“. Dieser Name stammt aus dem spanischen Raum und bedeutet übersetzt so viel wie Umfrage. Da es unsere Aufgabe war, ein Umfragetool zu programmieren, fanden wir diesen Namen als sehr passend. Momentan steht das Programm in der Version 0.183 zu Verfügung.

Programmkurzbeschreibung

Aufgabe des Programms ist es, anhand eines registrierten Benutzers Umfragen anzulegen, zu bearbeiten, freizugeben, auswerten zu können und sie auch wieder zu löschen. Damit Teilnehmer an den erstellten Umfragen teilnehmen können und eine Auswertung überhaupt möglich ist, generiert das Programm einen Link, womit sich die Umfrage verschicken lässt. Alle Daten wie Benutzer, Umfragen und Auswertungen werden in einer Datenbank zur Datenkonservierung gespeichert.

Programmbedarf

Programmiersprachen / Compiler

Angaben zu Ruby on Rails, CSS usw. kurz beschreiben

Genaue Angaben zur Sprach- und Compilerversion.

Datenorganisation

Bilder hier, Datenbank da…

Auflistung aller verwendeten Dateien (Name,

Zweck, Zugriffsart, Speicherbedarf, Anwendungsgrenzen).

Angaben zur Verwendung eines

Datenbankbetriebssystems.

Programmfunktion Stefan  
  
Aufgabenstellung

Aufgabenbeschreibung

Detaillierte Beschreibung der zu lösenden Aufgabe

Überblick, Zusammenhänge, Was soll gelöst

werden?)

Aufgabenlösung

Funktionshierarchie

Hierarchische Struktur von Aufgaben und Unteraufgaben

(Bezug zum Programmaufbau).

Methoden/Algorithmen

Exakte Beschreibung der Lösungsverfahren mit

Anwendungsgrenzen (Wie wird die Aufgabe

gelöst?). Beschreibung von Plausibilitätskontrollen.

Fehlerbehandlung

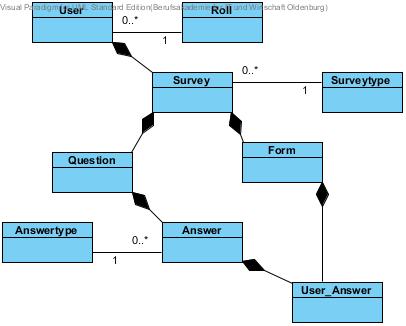
Angabe der vorgesehenen Fehlermeldungen und

der daraus resultierenden Maßnahmen.

Programmaufbau

Programmbausteine

Die Programme des Umfrage-Systems wurden vollständig mit Ruby on Rails geschrieben und gliedern sich in eine Klassenstruktur nach dem Model-View-Controller-Prinzip auf. Die Models spiegeln dabei die Tabellen der Datenbank wieder und sind über Fremdschlüsselrelationen miteinander verknüpft. Es gibt jeweils für jeden Bestandteil der Umfrage ein Model.



Liste aller Models:

|  |  |
| --- | --- |
| Model | Beschreibung der Ruby-Klasse |
| user.rb | Klasse für Nutzerobjekte, hat eine Rolle und kann mehrere Umfragen besitzen |
| survey.rb | Klasse für die Umfragen, gehört zu einem Nutzer und hat eine bis mehrere Fragen |
| surveytype.rb | Einer von zwei Umfrage-Typen wird einer Umfrage zugeordnet |
| question.rb | Klasse für Frageobjekte, ist einer Umfrage zugeordnet und enthält eine bis mehrere Antworten |
| answer.rb | Klasse für Antwortobjekte, ist einer Frage zugeordnet und hat einen von vier Antworttypen |
| answertype.rb | Ein Antworttype wird jeder Antwort zugeordnet um deren Art zu bestimmen. |
| form.rb | Klasse für Formular-Objekte. Formulare werden anhand einer fertigen Umfrage mit Fragen und Antworten erstellt und werden von einem Teilnehmer(nicht der Benutzer) ausgefüllt. Das Formular ist der Quell-Umfrage zugeordnet. Anhand der Angaben werden User\_Antworten dem Formular zugeordnet. |
| user\_answer.rb | Klasse für User\_Antworten. Werden einem Formular zugeordnet. |
| roll.rb | Klasse für Rollen von Nutzern (wurde nicht implementiert, ist aber vorhanden) |

Für jedes Model existiert ein Controller und mindestens ein View. Die Views bestehen aus fertigem HTML und HTML-generierenden Ruby-Code. Über die Views werden sowohl Informationen und Designs angezeigt als auch über Formulare Anweisungen an die Datenbank gegeben. Die Anweisungen dabei sind meist für die Erstellung und Zuweisung eines Objektes zuständig, werden aber auch für die Abfrage von Objekten aus der Datenbank verwendet.

Liste aller Controller und Views zu einem Model

|  |  |  |
| --- | --- | --- |
| Model | Controller | Views |
| user.rb | users\_controller.rb | new.html.erb  edit.html.erb  show.html.erb |
| survey.rb | surveys\_controller.rb | new.html.erb  edit.html.erb  show.html.erb  \_survey.html.erb |
| surveytype.rb | surveytypes\_controller.rb | (keine Views) |
| question.rb | questions\_controller.rb | new.html.erb  edit.html.erb |
| answer.rb | answers\_controller.rb | new.html.erb  edit.html.erb |
| answertype.rb | answertypes\_controller.rb | (keine Views) |
| form.rb | forms\_controller.rb | new.html.erb |
| user\_answer.rb | user\_answers\_controller.rb | (keine Views) |
|  | pages\_controller.rb | attend.html.erb  contact.html.erb  forgottenpw.html.erb  help.html.erb  home.html.erb  impressum.html.erb |
| evaluation\_controller.rb | Show.html.erb zur Darstellung einer Auswertung zur Umfrage |
| application\_controller.rb | application.html.erb |
| sessions\_controller.rb | new.html.erb |
|  | Shared-Views:  \_error\_messages.html.erb  \_survey\_form.html.erb  \_user\_info.html |

Die Controller-Methoden zu den Models sind grundsätzlich nach dem CRUD-Prinzip(„Create“, “Read“, „Update“ und „Destroy“) gegliedert. Ausnahmen dazu sind einige privaten Methoden des users\_controller, die die Authentifizierung des Nutzers ermöglichen sollen und der include des sessions\_helper in den application\_controller. Der surveys\_controller verfügt zudem über eine lock-Methode, über die ein Wert in der Datenbank geändert wird und daran in der \_survey.html.erb einigen Links an- bzw. nicht anwählbar gemacht werden. (Bsp. Lock = true; „Bearbeiten“ nicht mehr möglich; „Teilnahme“ und „Auswerten“ möglich)

Survey- und Answertypes wurden während der Implemetierung manuell in die Datenbank geschrieben. Daher sind Controller dafür nicht nötig, da eine Erstellung von Objekten dieser Klassen nicht im laufenden Betrieb erdacht waren. User\_Answers werden während der Erstellung des Formulars (forms) erstellt und diesem zugeordnet. Daher ist auch hier kein Controller von Nutzen und bleibt ohne Funktionen. Gleiches gilt für die jeweiligen Views. Entweder werden die Objekte nur in anderen Views abgefragt um eine Zuordnung zu ermöglichen (Survey- und Anwertypes) oder werden über ein anderes View indirekt erstellt (User\_Answers über Forms).

Die new.html.erb-Views werden für die Erstellung von Objekten verwendet. In ihnen können nötige Daten in HTML-generierte Formulare eingeben werden. Aus diesen wird mit Absenden des Formars die Create-Methode der jeweiligen Controller aufgerufen und das neue Objekt in die Datenbank geschrieben. Die View „\_survey\_form.html.erb“ gehört ebenfalls dazu. Durch sie ist es bereits in der Home-Seite möglich eine neue Umfrage zu erstellen.

Die edit.html.erb-Views gehen nach dem gleichen Prinzip vor. Hier werden jedoch nicht neue Objekte erstellt, sondern bereits vorhandene ausgelesen und die Daten im Formular verändert.

Die show.html.erb-Views bei Usern, Surveys und Evaluations dienen nur dem Zweck, eine Sicht auf die Daten in der Datenbank zu erhalten. Somit werden bei Usern nur deren Daten angezeigt, bei Surveys ein Umfragebogen ohne „Submit“-Button generiert und bei Evaluationen das Ergebniss der Auswertung ausgegeben.“\_survey.html.erb“ ist ebenfalls eine „Show-View“. Jedoch wird hier nur der Name der Umfrage ausgeben und Links für weitere Schritte („Bearbeiten“, „Freigeben“, „Auswerten“, „Löschen“ und „Teilnehmen“) sowie die Anzahl der bereits durchgeführten Teilnahmen angezeigt.

Views, die zu den Pages gehören enthalten hauptsächlich nur Informationen bzw. in HTML geschriebene Seiten. Sie selbst haben mit den Funktionen im Umfrage-System nichts zu tun und dienen lediglich der Information des Nutzers und der Darstellung dieser.

Über Redirects in den Controller-Methoden wird auf die jeweiligen Views für den nächsten Schritt im Programmablauf gewiesen.

Die application.html.erb bildet die Hauptstruktur der Seite und ermöglicht neben der Navigation und dem Banner auch noch die Login-Maske sowie die Einbindung von anderen Pages. Sie bildet damit damit den Bereich, der grundsätzlich mit jeder Seite gleich bleibt.

Programmablauf

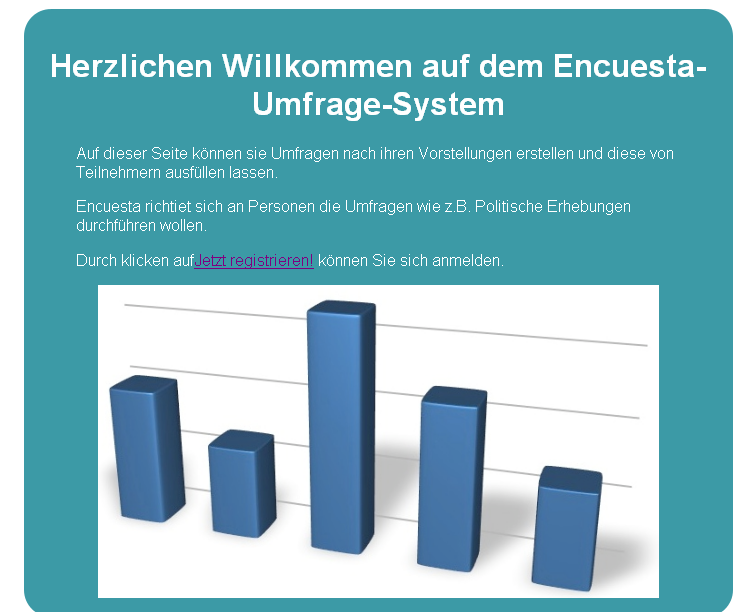
Programmablaufbeschreibung

### Dialogfolgeplan

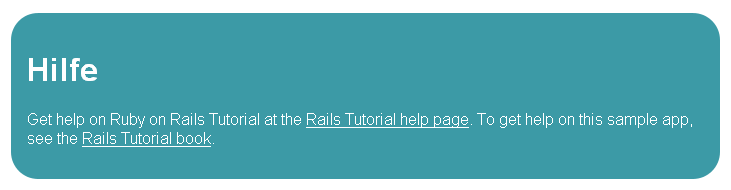
#### Menübottons:



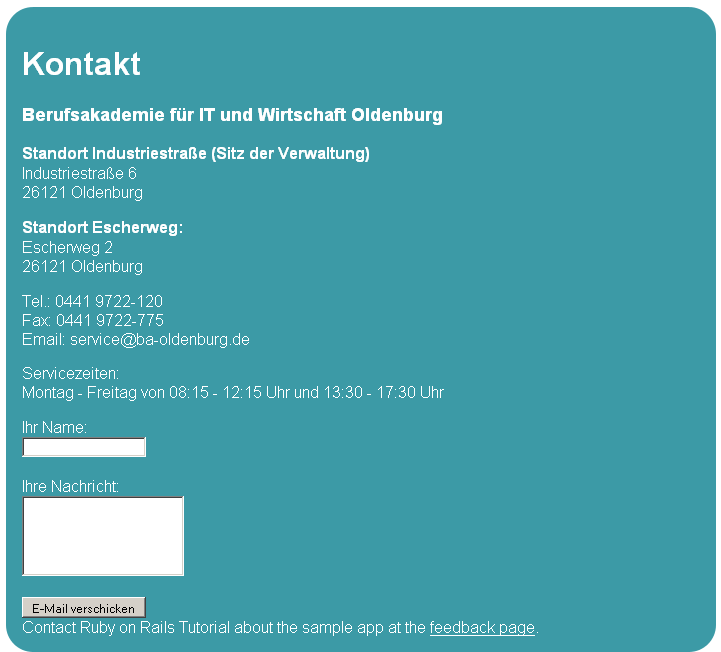
* Wenn man auf den Button klickt, gelangt man immer wieder zur Startseite zurück.



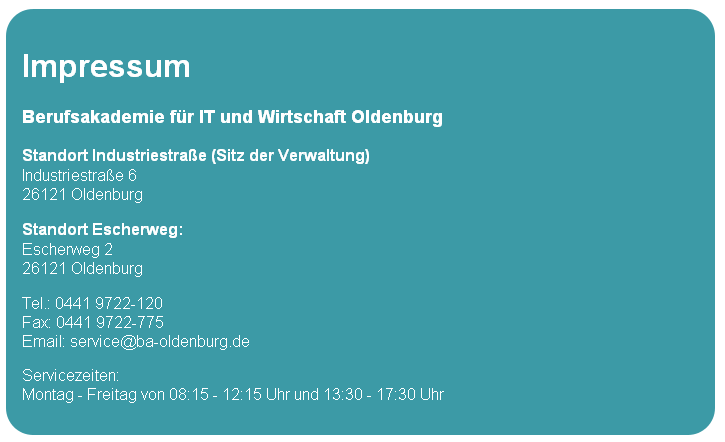
* Der  Button schafft Abhilfe bei Problemen, z.B. wenn Sie nicht wissen wie sie mit dem System interagieren sollen. Dann finden sie unter diesem Punkt Hilfe.



* Über den  Button können sie zu uns Kontakt aufnehmen.

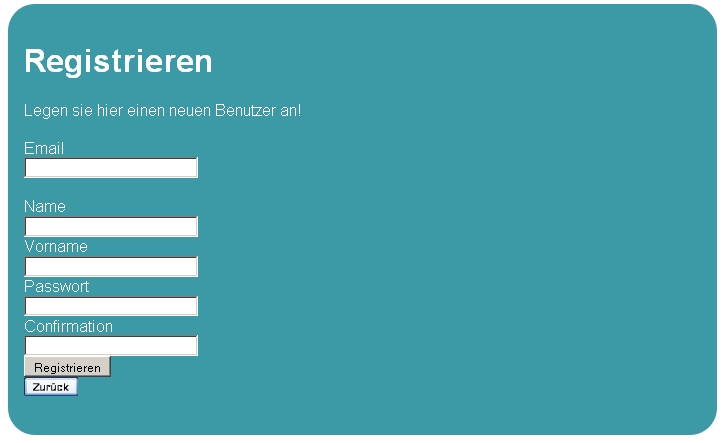


* Der  Button zeigt das Impressum der Seite an.



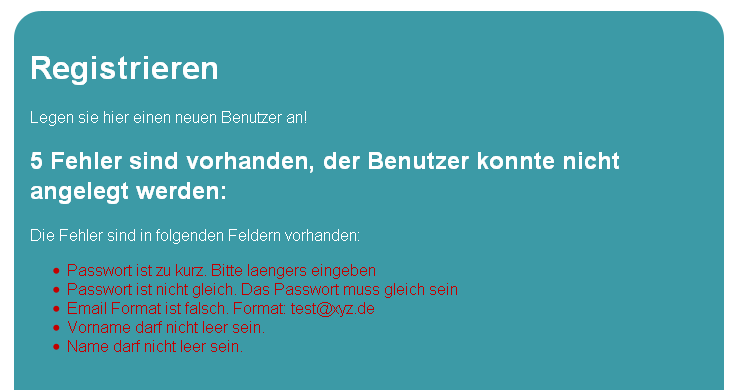
#### Die Registrierung:

* Durch den „Jetzt Registrieren “ Link oder durch den Registrierungsbutton im rechten Login Bereich erscheint das Registrierungsformular.

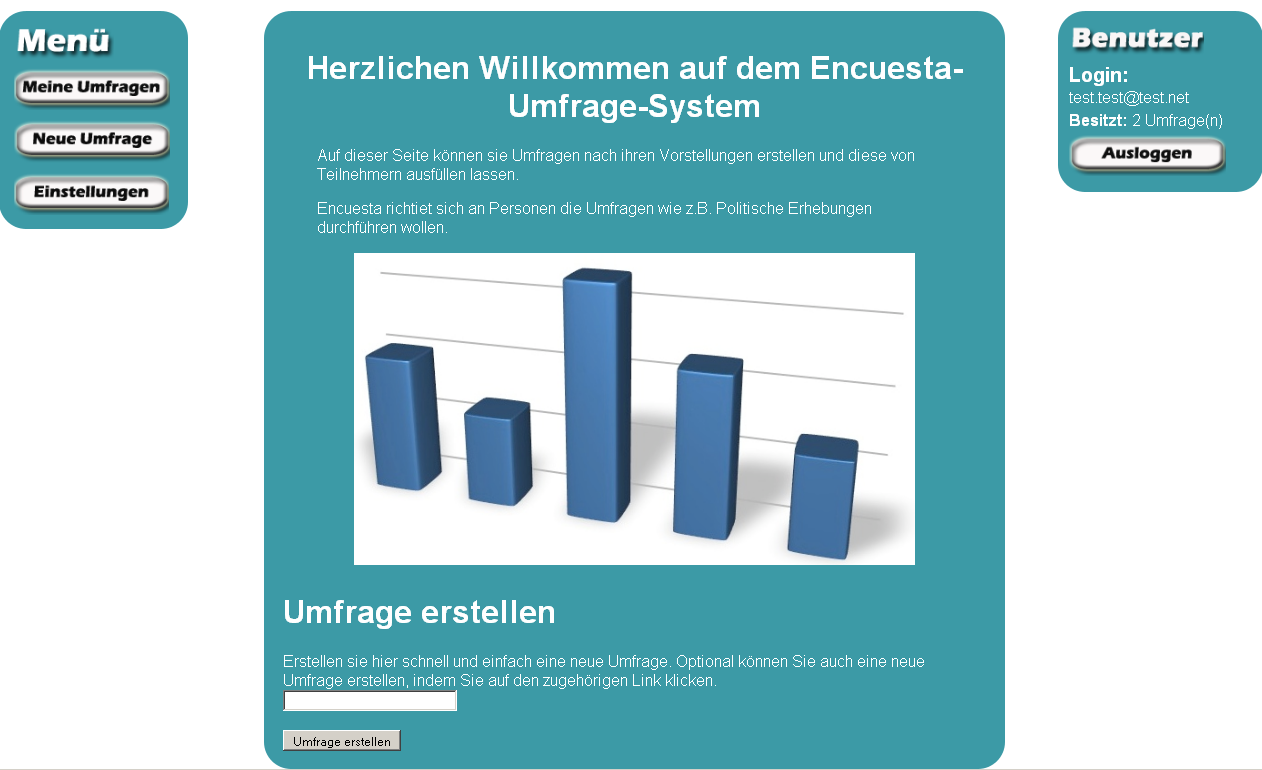


Dialogablaufplan

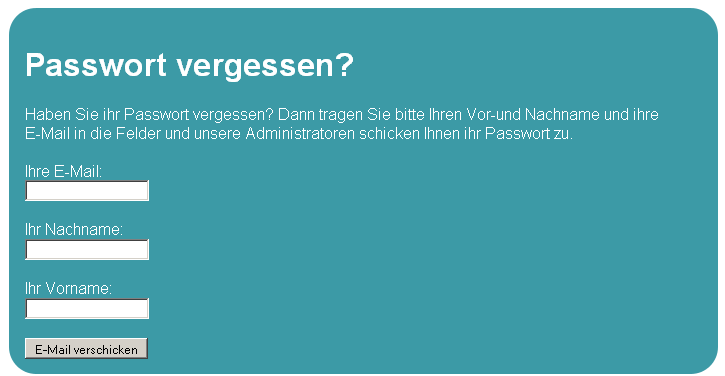
* Durch Fehleingabe in der Registrierungsmaske, wie z.B. Passworteingabe zu kurz oder falsche E-Mail Adresse, kommt es zu einer Error-Message.



* Sind alle Fehler beseitigt und man konnte sich erfolgreich registrieren kommt man zum Benutzer Layout.



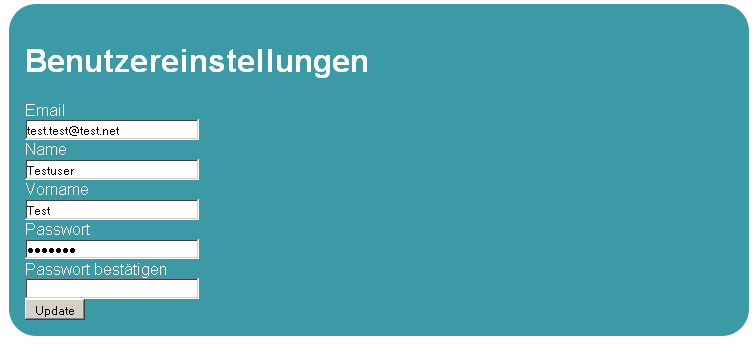
* Besitzt man schon einen Account, hat aber das Passwort vergessen, lässt sich über den „Passwort vergessen“ Link eine Anfrage zur Zurücksetzung des Passworts ermöglichen. Dies geschieht durch Ausfüllen der Eingabefelder und das Klicken des Buttons „E-Mail verschicken“. Dadurch öffnet sich Microsoft Office, falls auf dem Computer vorhanden, und schreibt diese Parameter in die E-Mail. Zum erfolgreichen Versenden muss dann nur noch im Mailprogramm auf „Senden“ gedrückt werden.



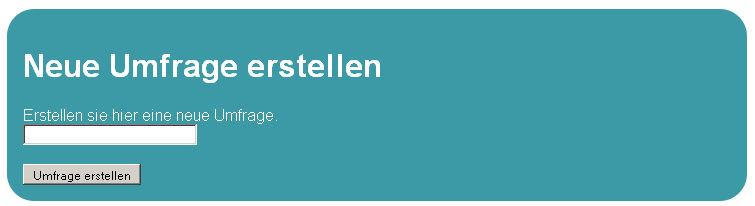
#### Menü Leiste:



* Über den Button  gelangt man zu seinen Benutzereinstellungen. Dort kann man seine persönlichen Einstellungen ändern.



* Mit Hilfe des  Buttons gelangt man zur Umfrage Erstellung.



* Dies ist allerdings auch schon auf der User Home Seite möglich.



* Gibt man jetzt in das Textfeld einen Umfragenamen ein, wie z.B. „Getränke“ und klickt auf „Umfrage erstellen“, fertigt das System einen Fragebogen, bei dem es möglich ist, eigene Fragen hinzuzufügen. Außerdem ist es möglich auszuwählen welche Typ Art die Frage darstellen soll. Möglich sind Radio-,Checkbox-, Textfeld- und Nummernfeld-Frage.



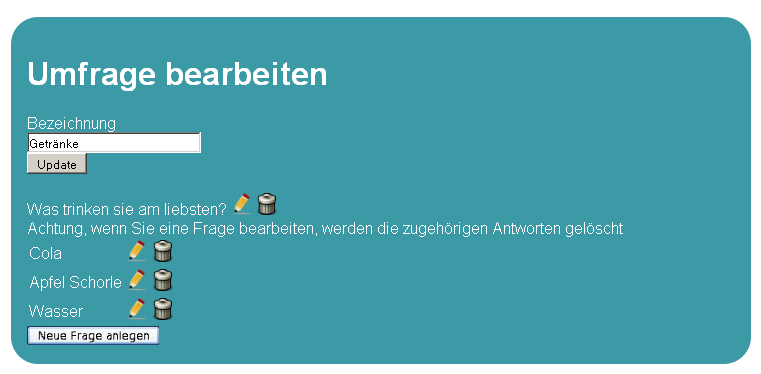
* Ist die Frage erstellt, gelangt zur nächsten Maske, in der die Antworten hinzugefügt werden. In diesem Szenario wurde jetzt die Frage „Was trinken sie am liebsten?“ hinzugefügt.



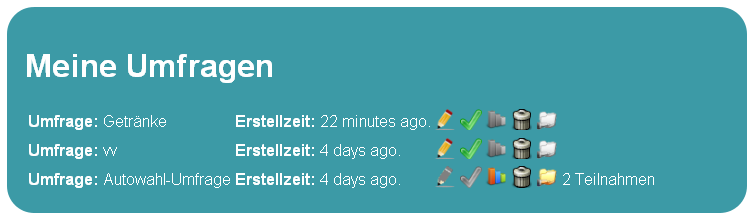
* Eine Frage muss mindestens eine Antwortmöglichkeit besitzen, wenn man diese hinzufügt, wird sie sofort angezeigt, um doppelte Antworten auszuschließen.



* Mit dem Button gelangt man wieder zum Frage Erstell-Formular.
* Durch  kommt man zu einer kompletten Übersicht der Umfrage. Hier wäre es nochmals möglich Fragen oder Antworten zu bearbeiten oder zu löschen.



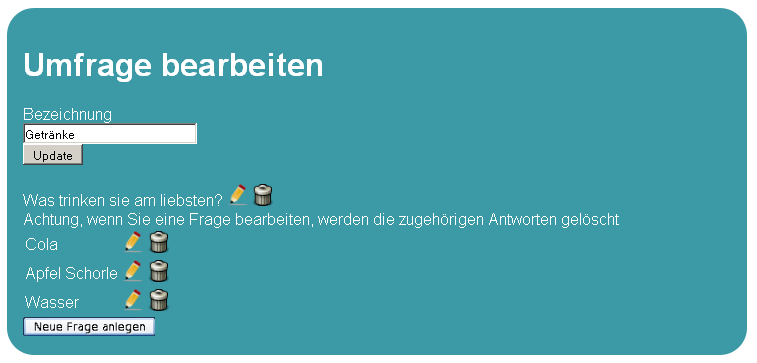
* Unter dem letzten Menüpunkt  sieht man nun alle eigenen erstellten Umfragen und kann mit ihnen verschiedene Funktionen ausführen.



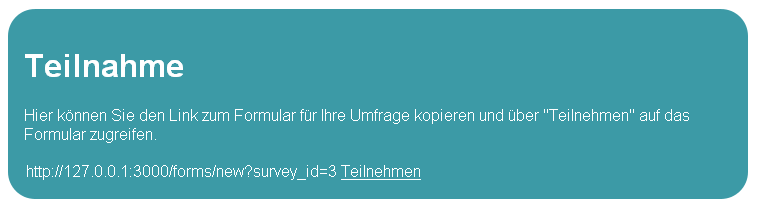
#### Meine Umfragen Funktionen:

#### 

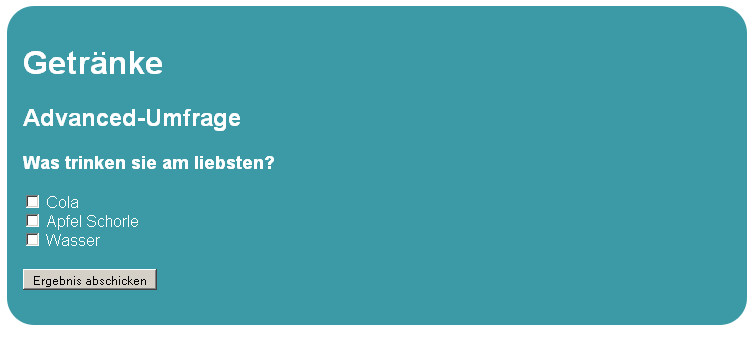
* Das Symbol  „Umfrage bearbeiten“ lässt den Benutzer in den Bearbeitungsmodus der Umfrage zurückkehren.



Mit dem grünen Harken  schaltet man die Umfrage frei. Diese Funktion steht einmalig zur Verfügung, der Harken wir dann aus gegraut . Außerdem ist ab diesem Zeitpunkt eine Bearbeitung der Umfrage nicht mehr möglich.



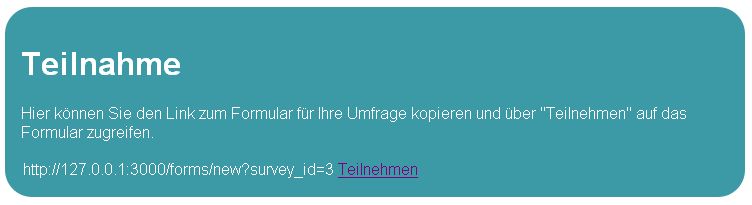
* Durch die Freischaltung erhält man einen Teilnahmelink, der an die Teilnehmer versendet wird, damit diese an der Umfrage teilnehmen können.



*  führt zu Beendigung und Abschicken der Umfrage.



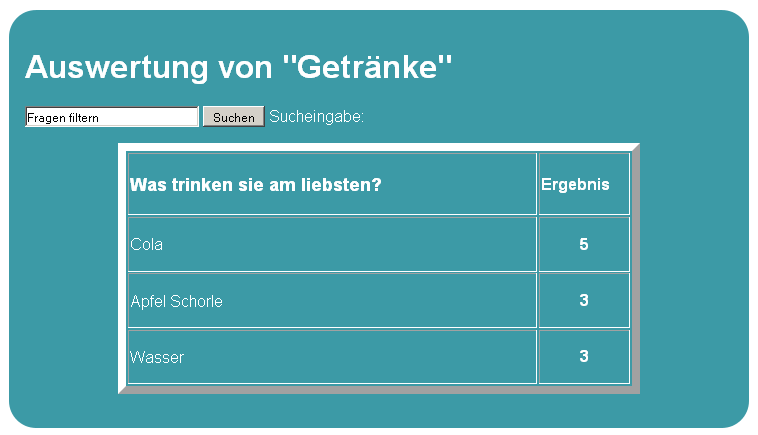
* Umfrage und Link „Ansehen“  war vor der Freischaltung ausgegraut. Nach der Freischaltung aber  ist es möglich, über dieses Symbol sich den Teilnahmelink nochmals angeben zu lassen.



* Der Mülleimer  steht für „die Umfrage löschen“. Es erscheint ein Popup mit der Anfrage, ob die Umfrage wirklich gelöscht werden soll. Wird mit OK bestätigt, ist die Umfrage gelöscht.
* Hinter den Symbolen steht die Anzahl der Teilnehmer, die die Umfrage ausgefühlt haben.



* Nachdem Teilnehmer die Umfrage ausgefüllt haben, ist es möglich eine Auswertung vor zunehmen. Bei einer Umfrage mit vielen Fragen besteht die Option über die Suchfunktion nach bestimmten Schlagwörtern zu suchen.



* Der letzte Button der eine Interaktion mit dem System ausführt, ist der  Button auf der rechten Seite der Benutzer Box. Dieser Button loggt den Benutzer aus und lässt ihn auf die Startseite des Umfragesystems zurückkehren.



Dialogbeschreibung Lars

Wir haben uns für ein „Vier Bereich Design“ entschieden, das bedeutet, wir verwenden die linke und rechte Seite, den Kopfanteil und die Mitte. Die rechte Seite verändert ihren Aufbau und die linke Seite erscheint beim Anmelden eines Benutzers. Die Textuellen Teile werden jeweils im Mittelblock ausgetauscht.



In diesen Browserausschnitt sieht man die Startseite von Encuesta, unserem Umfragesystem. Im oberen Bildschirmbereich befindet sich eine Menüleiste mit den wichtigsten Informationen auf unserer Seite. Direkt darunter unserer Logo mit unserem Werbeslogan „Die einfach starke Umfrage“. Mit diesem Spruch wollten wir anzeigen, dass es sich um ein einfach bedienbares Umfragesystem handelt, welches aber trotzdem stark in seiner Auswertung ist. In der Mitte befindet sich unser zentraler Block, der sich durchs ganze Programm zieht. Hier werden auf der Startseite der Willkommenstext und der Eye catcher platziert. Außerdem findet sich ein Registrierungslink „Jetzt registrieren“ in der Mitte der Seite. Auf dem rechten Teil befindet sich die Login Maske, mit der man sich als Benutzer anmelden kann und, falls man sein Passwort vergessen hat, ein „Passwort vergessen?“ Link. Der Loginteil beisitzt außerdem ein Registrierungsbutton.



Dies ist das Layout, wenn sich der Benutzer mit dem System angemeldet hat. Dort wo vorher der Login Bereich war, ist jetzt der Benutzer-Bereich. Hier stehen die Informationen, mit welchen sich der Nutzer angemeldet hat und die Anzahl der Umfragen, die bis jetzt erstellt wurden. Außerdem kann man sich durch betätigen des „Ausloggen“ Buttons aus dem System abmelden.

Auf der linken Seite, die vorher leer war, ist jetzt das Benutzermenü. Durch dieses Menü kann man auf verschiedene Seiten gelangen, je nachdem welchen Button man betätigt. Der „Meine Umfragen“ führt zu den gespeicherten Umfragen, „Neue Umfrage“ führt zum Erstellen einer neuen Umfrage und unter „Einstellungen“ lassen sich die persönlichen Daten verändern.

Datenorganisation

Eingabedaten

Die Eingabedaten die das System entgegen nimmt teilen sich auf in:

* Nutzerdaten bei der Registrierung
  + Name
  + Vorname
  + E-Mailadresse
  + Passwort
* Anmeldedaten bei der Anmeldung
  + E-Mailadresse
  + Passwort
* Eingaben bei der Erstellung von Umfragen
  + Name der Umfrage
  + Fragetexte
  + Antwortete
  + Auswahl der Fragetypen
* Antworten der Teilnehmer
  + Checkboxen oder Radiobutton
  + Texteingaben
  + Zahleneingaben

Das System nimmt die Daten je nach Eingabefeld über Tastatur oder Mausauswahl entgegen.

Bei Textfeldern ist eine maximale Länge von 255 Zeichen vorgesehen. Das Problem von Textzeicheneingaben bei Nummernfeldern ist bekannt aber die Behebung dieses Problems ist für eine spätere Programmversion vorgesehen.

Ausgabedaten

Das System informiert den Ersteller zu jederzeit über seine Eingaben. Als wichtigste Ausgabe lässt sich die Auswertung nennen. Es werden je nach Antwortyp Count, Durchschnitte oder die Teilnehmerantworten als solches ausgegeben.

Datenbankanwendung

Als Datenbanksystem dient eine Sqlite Datenbank in der Version 3.7.4. Die Datenbank deckt alle Funktionen des Encuesta Umfragesystems ab. Sie übernimmt die Verwaltung der Nutzerdaten, der erstellten Umfragen, sowie die Speicherung der Ergebnisse.

## Datenbankschema

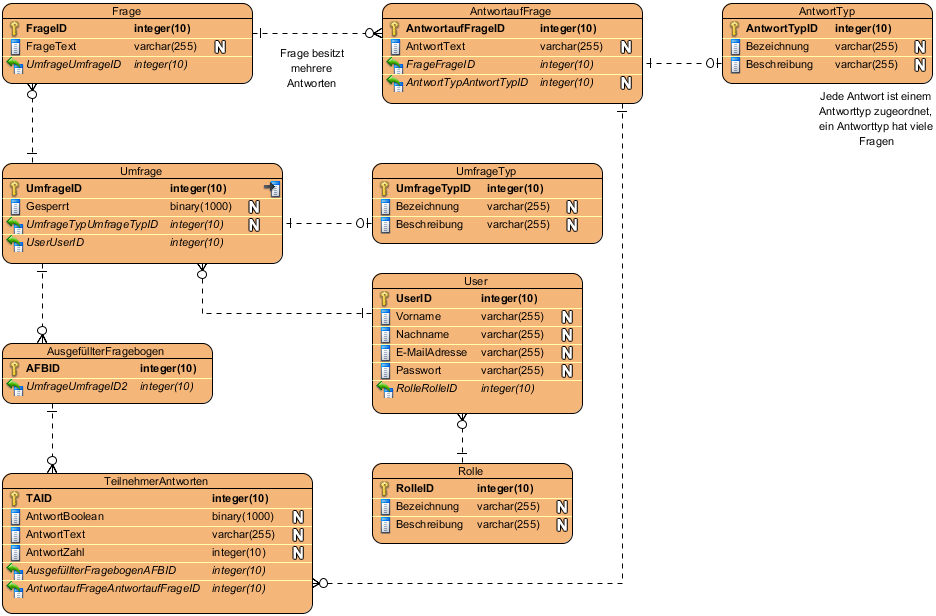


Abbildung 1 Datenbankschema

### Nutzerverwaltung

Die Nutzer werden in zwei Tabellen verwaltet. Zum einen der Benutzer selbst (Tabelle „User“) und zum anderen die Art des Benutzers („Rolle“). Auf diese Art ist das System für einen zukünftigen Ausbau offen. Über das Anlegen einer neuen Rolle können beispielsweise Administratoren in das System eingefügt werden. Das Passwort des Benutzers wird als MD5 Hash abgelegt und mittels „salt“ bestmöglich gesichert. Selbst bei einem möglichen Diebstahl der Hashes können diese nicht erraten werden, da sie mit dem „salt“ zusätzlich abgesichert sind.

### Umfrageverwaltung

Eine Umfrage setzt sich aus der Umfrage selbst, ihrem Typ, den Fragen sowie den Antworten und deren Typen zusammen. Auch hier wurde auf einfach Erweiterbarkeit wert gelegt. Durch die Tabelle *AntwortTyp* (answertypes) lassen sich leicht neue Antworttypen integrieren. Die Umfrage als solches besitzt viele Fragen. Diese Fragen besitzen wiederum mehrere Antworten welche einen bestimmten Typ zugeordnet sind. Im Auslieferungszustand sind vier Fragetypen hinterlegt.

|  |  |
| --- | --- |
| Bezeichnung | Beschreibung |
| Radio-Frage | Einfachauswahl |
| Checkbox-Frage | Mehrfachauswahl |
| Textfeld-Frage |  |
| Nummernfeld-Frage |  |

Über den Umfragetypen lassen sich unterschiedliche Umfragetypen definieren. So ist es möglich unterschiedliche Fragebögen zu kreieren. Im derzeitigen Zustand wird nur ein Umfragetyp unterstützt, die Advanced-Umfrage zum Aufbau eines Fragebogens mit vielen Fragen.

### Speicherung der Umfrageergebnisse

Die Speicherung der Ergebnisse wird mittels der Tabellen *AusgefüllterFragebogen* („forms“) und *TeilnehmerAntworten* (user\_answers) realisiert. Es gibt viele *AusgefüllterFragebogen* welche jeweils zu einer *Umfrage* (surveys) gehören. Für die *AusgefüllterFragebogen* gibt es entsprechned viele *TeilnehmerAntworten*. Um in der Auswertung eine Nachvollziehbarkeit zu erhalten welche *TeilnehmerAntworten* zu welcher Frage gehört, sind diese jeweils mit einer dazugehörigen *AntwortaufFrage* verknüpft*.*

Anwendungsbeispiel Stefan

Erläuterung eines Anwendungsfalls mit Beschreibung

sämtlicher Programmeingaben/ausgaben.