

Regeln für das Erstellen von Web-Applikationen

Autor: Michael Abplanalp
Version: 1.0
Datum: 6. Oktober 2016
Module: 133 und 151

Inhaltsverzeichnis

1	Einleitung.....	3
2	Allgemeine Regeln.....	3
2.1	Navigation in der Applikation.....	3
2.2	Zugang zu Funktionen	3
2.3	Sehen, wo man befindet	3
2.4	Formulare	3
2.4.1	Einzeilige Textfelder	4
2.4.2	Mehrzeilige Textfelder	4
2.5	Keine vordefinierten Abläufe und Abbrechen von Aktionen	4
2.6	Navigation nach erfolgreicher Aktion	4
3	Spezifische Regeln	5
3.1	Login- und Member-Bereich	5
3.2	Registration	5
3.3	Login	6
3.4	Bearbeiten von Elementen.....	6
3.4.1	Einfügen eines Elementes	6
3.4.2	Bearbeiten eines Elementes.....	6
3.4.3	Löschen eines Elementes	6
4	Datenbanken	6

1 Einleitung

Funktionen und Abläufe von verschiedenen Web-Applikationen wiederholen sich immer wieder. In diesem Dokument werden grundsätzliche Anforderungen festgehalten, die für alle Web-Applikationen gelten. Wenn im Arbeitsauftrag von Web-Projekten und Web-Applikationen nichts Gegenteiliges vermerkt ist, dann müssen diese Anforderungen umgesetzt werden. Jedes Nicht-Einhalten führt zu einem Punkteabzug.

Zur Veranschaulichung der Regeln wird eine Blog-Applikation verwendet.

2 Allgemeine Regeln

2.1 Navigation in der Applikation

Die Navigation in der Applikation soll durch Menüs, Links und Buttons sichergestellt werden. Die Back- und Forward-Funktionen des Browsers sind tabu.

2.2 Zugang zu Funktionen

Alle Funktionen sollen nach Möglichkeit via Haupt- und Untermenüs zugänglich sein. Dies ist jedoch nicht immer möglich bzw. sinnvoll. Wenn der Zugang zu gewissen Funktionen im Content-Bereich erfolgt (via Links bzw. Buttons), dann muss die Navigation sichergestellt werden (z.B. mit Zurück-Buttons).

Beispiel: In einer Übersicht werden alle Blog-Beiträge angezeigt. Direkt neben jedem Beitrag befindet sich ein Symbol zum Bearbeiten und Löschen des Beitrags.

2.3 Sehen, wo man befindet

Der Benutzer soll sehen, wo innerhalb der Website er sich befindet. Dies kann wie folgt realisiert werden:

- Der Menüpunkt, den er gewählt hat, ist markiert (d.h. hebt sich von den anderen Rubriken ab).
- Dasselbe gilt für das Untermenü bzw. die Untermenüs. Das bedeutet, dass pro Menüebene ein Eintrag markiert ist.
- Wenn Funktionen nicht via Menü, sondern im Content-Bereich ausgewählt werden können, dann soll dies sichtbar sein, z.B. im Titel der Seite.
- Zusätzlich oder als Alternative kann die Brotkrümelnavigation (breadcrumbs navigation) verwendet werden: Der Benutzer sieht oben im Content-Bereich, wo er sich gerade befindet. In der Regel sind das alle Menü- und Untermenüpunkte, die er gewählt hat.

Beispiel: Im Online-Shop *brack.ch* will der Benutzer einen USB-Stick kaufen. Er wählt nacheinander die Menüpunkte *IT-Hardware*, *PC-Peripherie* und *USB-Sticks*. Zwischen dem Menü und dem Content-Bereich sieht er seine Navigation:

[IT-Hardware](#) → [PC-Peripherie](#) → [USB-Sticks](#). Zudem sind die einzelnen Menüpunkte als Link hinterlegt, der Benutzer kann also ganz einfach in der Navigation eine oder mehrere Stufen zurückgehen.

2.4 Formulare

- Die Reihenfolge der Formular-Elemente (Input-Felder und Buttons) ist von oben nach unten (und wenn mehrspaltig von links nach rechts).
- Beim Öffnen der Maske liegt der Fokus auf dem ersten Feld bzw. Button.
- Es soll klar ersichtlich sein, was in die Felder eingegeben werden muss. Bsp.: Wenn beim Benutzername eine Mail-Adresse erwartet wird, muss dies ersichtlich sein.

- Wenn bei einem Feld ein spezifisches Format verlangt wird, muss dieses Format bereits bei der Eingabe ersichtlich sein (und nicht erst in einer Fehlermeldung nach Klicken auf den Senden-Button). Bsp.: Format des Passwortes = 1 Gross- und Kleinbuchstabe, 1 Ziffer, 1 Sonderzeichen und Minimallänge von 8 Zeichen.

2.4.1 Einzeilige Textfelder

Einzeilige Textfelder sind geeignet für Benutzernamen, Mail-Adressen, Passwörter, Titel usw. Zu beachten:

- Beim Speichern in eine Datenbank müssen die Eingaben maskiert werden. Bsp. MySQL und PHP:
 - Maskierung mit `mysqli_real_escape_string()`
 - Falsch ist eine HTML-Codierung wie `htmlspecialchars()`, denn die Daten sollen ausgabeneutral abgespeichert werden
- Beim Anzeigen müssen die Daten HTML-codiert werden, z.B. mit der PHP-Funktion `htmlspecialchars()`

2.4.2 Mehrzeilige Textfelder

Oft werden relativ viele Daten in ein Feld eingegeben, z.B. bei Forums- und Blog-Beiträgen. In diesen Fällen sind zwingend mehrzeilige Textfelder zu verwenden. Die Anfangsgrösse sollte zudem angemessen gross gewählt werden, auch wenn diese zur Laufzeit vergrössert werden können. Zu beachten:

- Für das Speichern und Anzeigen gelten dieselben Regeln wie bei einzeiligen Textfeldern
- Zeilenumbrüche: Zeilenumbrüche, die der Benutzer eingegeben hat, sollen bei der Ausgabe entsprechend angezeigt werden.
 - Zeilenumbrüche werden beim Speichern nicht speziell behandelt. Sie sollten auf keinen Fall durch "
" ersetzt werden.
 - Bei der Ausgabe erfolgt nach der HTML-Codierung eine Umsetzung der Zeilenumbrüche in "
" mit der PHP-Funktion `nl2br()`.

2.5 Keine vordefinierten Abläufe und Abbrechen von Aktionen

Es sollen keine vordefinierten Abläufe programmiert werden, wo man erst nach mehreren Schritten wieder auf der Ursprungsmaske landet. Die Regel lautet: Jede Aktion muss abgebrochen werden können. Dies kann auf zwei Arten geschehen:

- Via Abbrechen-Button gelangt man zurück auf die Ursprungsmaske.
- Via Menü gelangt man zurück auf die Ursprungs- oder auf eine andere Maske.

Beispiel: Auf der Maske zum Hinzufügen eines Blog-Beitrags soll sich ein Abbrechen-Button befinden. Damit wird nicht die Maske geleert, sondern der Benutzer kehrt auf die Ursprungsmaske zurück.

2.6 Navigation nach erfolgreicher Aktion

Wenn eine Aktion erfolgreich abgeschlossen worden ist, kann folgendes geschehen:

- Der Benutzer sieht eine Erfolgsmeldung und bleibt auf derselben Maske stehen.
 - Beim Hinzufügen eines Elementes wird das Formular geleert.
 - Beim Ändern eines Elementes bleiben die Daten im Formular stehen.

Beispiel: Nach dem Einfügen eines Blog-Beitrags erscheint eine Erfolgsmeldung, die Maske wird geleert und ist bereit für die Eingabe des nächsten Kontaktes.

- Der Benutzer kehrt auf die Ursprungsmaske zurück.

Beispiel: Bei der Übersicht aller Blog-Beiträge klickt der Benutzer auf *Neuer Beitrag*. Nach dem Einfügen kehrt er auf die Übersicht zurück, wo der neue Beitrag nun ersichtlich ist.

- Eine Übersichtsmaske mit allen Elementen wird angezeigt.

Beispiel: Ein Beitrag wird angezeigt und der Benutzer löscht diesen. Danach wird eine Übersicht aller verbleibenden Beiträge angezeigt.

3 Spezifische Regeln

3.1 Login- und Member-Bereich

Viele Web-Anwendungen haben zwei separate Bereiche mit unterschiedlichem Menü und unterschiedlichen Funktionen:

- Ein öffentlicher Bereich, der für alle (auch nicht-registrierten) Benutzer zugänglich ist.
- Ein Member-Bereich, der für registrierte Benutzer reserviert ist.

Der öffentliche Bereich besteht normalerweise aus folgenden Funktionen:

- Registration: Neue Benutzer können sich registrieren
- Login: Registrierte Benutzer können sich anmelden
- Weitere Funktionen wie z.B. das Auswählen und Anzeigen von Blogs

Der Member-Bereich kann je nach Anwendung völlig unterschiedlich aussehen. Was immer erfüllt bzw. vorhanden sein muss:

- Es wird eine Sitzung (*Session*) aufgebaut. Anhand der Session-ID wird der Benutzer vom Server identifiziert.
- Es sollte immer eine Logout-Funktion vorhanden sein. Damit meldet sich der Benutzer ab und die Session wird zerstört.

Im Falle eines Blogs könnten folgende Funktionen im Member-Bereich vorhanden sein:

- Auswahl des Blogs, falls das nicht im öffentlichen Bereich möglich ist
- Anzeige der Blog-Beiträge
- Erstellen von eigenen Blog-Beiträgen
- Ändern von eigenen Blog-Beiträgen
- Löschen von eigenen Blog-Beiträgen

3.2 Registration

- Die Registration besteht mind. aus den Attributen *Benutzername* und *Passwort*.
- Der Benutzername identifiziert den Benutzer: Entweder ist es ein fiktiver Benutzername oder eine Mail-Adresse. Der Wert muss einzig sein (Merkmal *unique*).
- Das Passwort soll ein minimales Format erfüllen: Mind. 1 Gross- und Kleinbuchstabe, 1 Ziffer, 1 Sonderzeichen, Minimallänge = 8 Zeichen.
- Das Passwort ist nicht in Klartext sichtbar (input type="password").
- Das Passwort soll immer wiederholt werden, um Fehleingaben zu vermeiden.
- Überflüssige Attribute sind zu vermeiden, nur so viele wie nötig verwenden.
- Alle notwendigen Attribute sind zu überprüfen. Wenn es sich um eine Client-Server Applikation handelt, dann muss die Validation auf Serverseite erfolgen. Bei reinen Client-Applikationen erfolgt die Validation clientseitig mit HTML5, JavaScript und/oder jQuery.
- Nach erfolgreicher Registration sind drei Szenarien möglich:
 - Der Benutzer bleibt auf der (leeren) Registrationsmaske und sieht eine Erfolgsmeldung.

- Die Login-Seite wird angezeigt.
- Der Benutzer ist automatisch angemeldet, er befindet sich im Member-Bereich.

3.3 Login

Die Login-Maske besteht aus zwei Attributen, der Benutzeridentifikation (Benutzername bzw. Mail-Adresse) und dem Passwort:

- Keines der Attribute wird auf gültiges Format überprüft, es wird einfach in der Datenbank nach gültigen Credentials gesucht.
- Das Passwort ist nicht in Klartext sichtbar (input type="password").
- Registration und Login sind aus Benutzersicht zu trennen (es kann aber dasselbe Formular verwendet werden mit ein- und ausblenden von Attributen).

3.4 Bearbeiten von Elementen

3.4.1 Einfügen eines Elementes

Das Einfügen eines Elementes erfolgt nach Klicken auf Speichern oder Senden, eine Bestätigung ist nicht notwendig. Nach erfolgreichem Speichern wird entweder das Formular geleert oder eine Übersicht mit allen Elementen (inkl. dem Neuen) angezeigt. In jedem Fall erhält der Benutzer eine Erfolgsmeldung.

3.4.2 Bearbeiten eines Elementes

Die Änderungen erfolgen nach Klicken auf Speichern oder Senden, eine Bestätigung ist nicht notwendig. Nach erfolgreichem Speichern bleibt der Benutzer entweder auf dem Formular stehen oder es wird eine Übersicht mit allen Elementen angezeigt. In jedem Fall erhält er eine Erfolgsmeldung.

3.4.3 Löschen eines Elementes

Bevor ein Element gelöscht wird, sollte immer eine Löschbestätigung erfolgen („Wollen Sie dieses Element wirklich löschen?“). Eine einfache Variante mit einem JavaScript-Alert ist dazu völlig ausreichend.

4 Datenbanken

Viele Web-Anwendungen funktionieren mit Datenbanken, um Daten persistent zu speichern. In der Folge sind die wichtigsten Regeln zur Erstellung einer Datenbank aufgeführt:

- Zur Dokumentation gehört in jedem Fall ein Datenbankschema. Dieses wird vor dem Anlegen der Datenbank erstellt.
- Die Daten befinden sich in der 3. Normalform.
- Jede Tabelle hat einen ein einzelnes Attribut als Primärschlüssel (ID), die Werte sind ohne Aussagekraft. Das Attribut ist entsprechend zu indexieren.

Ausnahme: In einer n:m-Beziehung gibt es eine Zwischentabelle, um die Beziehungen abzubilden. Diese Zwischentabelle hat keine ID als Primärschlüssel, der Primärschlüssel setzt sich zusammen aus den beiden Fremdschlüsseln.

- Die Beziehung Primärschlüssel-Fremdschlüssel ist in der Datenbank immer abzubilden (entsprechende Indices setzen).
- Attributwerte, die nur einmal vorkommen dürfen, sind entsprechend zu indexieren (Merkmal *unique*). Bsp.: Mail-Adresse.
- Passwörter sind immer verschlüsselt zu speichern. Wenn nichts anderes festgelegt ist, soll ein Hash-Algorithmus mit Salt verwendet werden (PHP: `password_hash()`).

5 Rolle der Lehrperson

Ihre Lehrperson hat zwei Rollen in Softwareprojekten.

5.1 Rolle als Lehrperson und Coach

Die Lehrperson definiert das Projekt und legt die Regeln und Bedingungen fest. Sie macht die Einführung, gibt immer wieder Inputs und steht Ihnen während dem Projekt als Coach mit Rat und Tat zur Seite.

5.2 Rolle als Kunde und Auftraggeber

Betrachten Sie die Lehrperson auch als Kunden und Auftraggeber und verhalten Sie sich entsprechend:

- Jede Abweichung zu den Projektauflagen muss begründet und mit der Lehrperson abgesprochen werden.
- Wenn Sie nicht sicher sind, wie die Anforderungen zu verstehen sind, fragen Sie auf jeden Fall nach.
- Wenn Sie einen Termin (Zwischentermine, Abgabetermin, etc.) nicht einhalten können, informieren Sie die Lehrperson. Der Kunde muss wissen, wenn er mit einer Verzögerung zu rechnen hat!