

Einsendeaufgabe 3, Thema: PHP-Programmierung mit OOP, DB und Pattern.

Ziel dieser Aufgabe:

Wenden Sie bei der Lösung dieser Aufgabe unter anderem PHP-Programmierung, OOP, und Zugriff auf eine Datenbank bei nicht ganz trivialer Navigation an. Nutzen Sie AJAX, wenn dies sinnvoll ist.

Wichtiger Hinweis; abzugeben sind:

- Die Lösung von a),
- Bei b), c) und d) kommt es darauf an wie weit Sie gekommen. Die Lösung von soll in c) enthalten sein und die Lösung von c) (und somit transitiv, die von b)) in d). Somit geben sie von den drei Aufgabenteilen den höchstwertigen ab, je nachdem, wie weit Sie gekommen sind.
- die Lösung von e), welche alle vorigen Teillösungen enthält.
- die Lösung von f), welche alle vorigen Teillösungen enthält.

gesammelt in derselben zip-Datei.

Im Verzeichnis darüber sei eine HTML-Seite, welche auf die drei Aufgabenteile verlinkt.

Zur Aufgabe:

Konkretes Lernziel:

Eine kleine Website zu bauen, die als Basis für weitere Entwicklungen dienen kann. Sie enthalte unter anderem Validierung, Datenbankzugriff und Pattern.

Kontext:

Wir bleiben im Kontext unserer Projektkontakte.

Aufgabenstellung

a) Die clientseitige Validierung soll fehlende Eingaben anmerken.

Die **server**seitige Validierung soll

- I. syntaktische Fehler verhindern.
- II. sicherstellen, wenn die Vorwahl mit auf ein Mobiltelefon hindeutet, dass die Ortswahl mindestens 6 Ziffern hat.
- III. sicherstellen, dass die Domain-Kennung der E-Mail-Adresse eine von zwei fest einprogrammierten ist, die erlaubt sind.

Überlegen Sie bei welcher der Anforderungen i, ii oder iii der Einsatz von regular expressions Sinn macht und verwenden Sie diese dann.

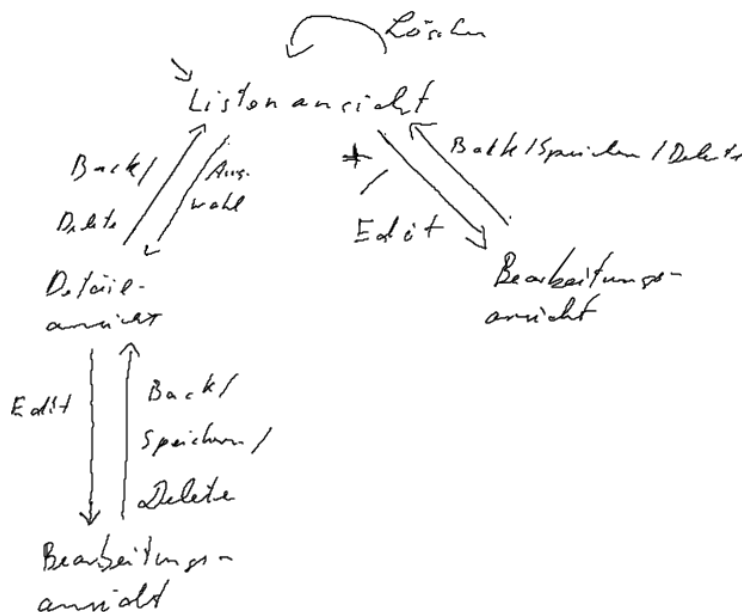
- b) Die Architektur der Anwendung wird nun verändert. Es gebe ein Interface IntPKontakt:

Dies wird implementiert durch Projektkontakt und TeamProjektkontakt. Beide implementierende Klassen haben neu das Attribut "Rolle". Die toString()-Funktion der Objekte soll alle relevante Daten in einem Satz ausgeben. Macht es Sinn, das Interface anzupassen?

Bei Projektkontakt wird als Rolle "keine" zurückgegeben und ist unveränderlich. (Gemeint sind bei Projektkontakt externe Projektkontakte.)

Bei TeamProjektkontakt kann eine sinnvolle Zeichenkette zurückgegeben werden. Die Rolle des TeamProjektkontaktes kann nur bei der Erzeugung des TeamProjektkontaktes hinzugefügt werden, später ist Sie nicht mehr änderbar. (Teamprojektkontakt meint: Projektkontakte aus dem Entwicklungsteam.)

- c) Alle CRUD Operationen sollen mittels MariaDB realisiert sein
- d) Es wird nun noch die Navigation verändert. Dabei ist die Listenansicht beim Start zu aktualisieren oder wenn gelöscht oder gespeichert wurde, nicht jedoch bei einfachem Zurückkommen zur Listenansicht. Hier eine Übersicht der Navigationswege. Bitte dran denken: Dies ist immer noch eine serverseitige Applikation.



Erläuterungen:

- Wenn man in der Liste ein Element löscht, bleibt man in der Listenansicht.
- Wenn man von der Listenansicht in die Detailansicht geht, kommt man nach Back oder Delete zurück zur Listenansicht. Bei Delete ist die Listenansicht zu aktualisieren. Bei Back nur dann, wenn das Objekt in der Bearbeitungsansicht gespeichert oder gelöscht wurde.

- Bei der Bearbeitungsansicht sind die Navigationsmöglichkeiten kontextsensitiv.
- Mittels Back kommt man in die Ansicht von der die Bearbeitungsansicht aufgerufen wurde. Es braucht nichts aktualisiert zu werden.
- Mittels Speichern kommt man in die Ansicht von der die Bearbeitungsansicht aufgerufen wurde. Die vorige Ansicht wird aktualisiert. Ist die vorige Ansicht die Detailansicht, so muss nach dem Back die Listenansicht aktualisiert werden.
- Mittels Löschen kommt man in die Ansicht von der die Bearbeitungsansicht aufgerufen wurde. Die vorige Ansicht wird aktualisiert. Ist die vorige Ansicht die Detailansicht, so wird diese nicht an den Client ausgeliefert, sondern ein weiteres Back erfolgt.

Die Persistenz mittels MariaDB bleibe erhalten.

- e) Weiterhin ist das Decorator-Pattern zu nutzen. Unsere bisherigen Objekte können diese Dekorierer haben:

- wieder erreichbar ab <Datum als Text>
- inaktiv:

oder unsere Objekte haben keinen Dekorierer.

Die toString()-funktion der dekorierten Objekte soll die Dekoriererbezeichner nach den Daten im Rückgabewert enthalten.

Hinweise:

Die erforderlichen Umstellungen sind schwierig.

Daher ist Persistenz mittels MariaDB hier nicht erforderlich.

- f) Optional und zum Knobeln:
Die Persistenz mittels MariaDB funktioniert auch noch nach Einbau des Decorator – Pattern.

Schlussbemerkung:

Was Sie bei Ihrer Programmierarbeit „auf dem Schirm“ haben sollten, ist, dass die gesamte Funktionalität der Anwendung serverseitig stattfindet. Auf der Clientseite sollen nur zurückgelieferter HTML-Code eingebunden werden oder zurückgelieferte Datenobjekte in Formularen angezeigt werden.