

Auftrag 7: Projekt *Bilder-Datenbank*

1. Einleitung

Fotoalben im Internet erfreuen sich grosser Beliebtheit, die bekanntesten Vertreter sind *Flickr* und *Google Fotos*. In einer Einzelarbeit erstellen Sie eine Web-Applikation für eine einfache Bilderverwaltung und -anzeige.

2. Arbeitsjournal

Führen Sie ein Arbeitsjournal, in dem Sie die geplanten Arbeiten und Ergebnisse (Soll / Ist) und aufgetretene Probleme mit Datum und Zeitangaben eintragen. Es hat den Hauptzweck, Ihre Methodenkompetenz zu verbessern. Und es dient der Lehrperson dazu, Sie während der Arbeit individuell und besser begleiten zu können. Das Arbeitsjournal wird zur Bewertung der Leistungsbereitschaft und zur Beurteilung der Eigenständigkeit der Arbeit herangezogen.

Das Journal wird pro Arbeitstag (Arbeitstag an der Schule und zu Hause) geführt und sieht wie folgt aus:

Tätigkeiten	Aufwand geplant (Std)	Aufwand effektiv (Std)
Beschreibung		
Beschreibung		
Tages Ablauf		
Beschreibung von Erfolgen und Misserfolgen (Problemen)		
Hilfestellungen		
Wer hat in welcher Form Hilfestellung erbracht		
Reflexion		
Gedanken über das Vorgehen. Was ist gut was würde ich besser machen		
Nächste Schritte		
Wie weiter, nächste Schritte...		

Hintergrundinformationen der IPA

Gemäss Art. 5 Absatz 2 der Wegleitung über die individuelle praktische Arbeit (IPA) an Lehrabschlussprüfungen des BBT vom 27. August 2001 gilt:

Die zu prüfende Person führt ein Arbeitsjournal. Sie dokumentiert darin täglich das Vorgehen, den Stand der Prüfungsarbeit, sämtliche fremde Hilfestellungen und besondere Vorkommnisse wie z.B. Änderungen der Aufgabenstellung, Arbeitsunterbrüche, organisatorische Probleme, Abweichungen von der Soll-Planung.“

Das Arbeitsjournal zur IPA ist zwingend zu führen und den Experten und Fachvorgesetzten vorzulegen. Das Arbeitsjournal ist täglich sinngemäss und korrekt auszufüllen.

Das Arbeitsjournal dient der Nachvollziehbarkeit der von den Lernenden ausgeführten Arbeiten und wird als Teil der IPA in die Bewertung mit einbezogen.

3. Rahmenbedingungen

Diese Bedingungen legen den Rahmen des Projektes fest. Es handelt sich zum einen um grundsätzliche Anforderungen, die für alle Web-Applikationen gelten. Zum andern handelt es sich um einen fixen Rahmen, damit bei allen Projekten dieselben Voraussetzungen gelten und diese fair bewertet werden können.

- Das Dokument „Regeln für das Erstellen von Web-Applikationen“ (Dateiname *M133-151Regeln_fuer_Web-Applikationen.docx*) ist die Grundlage für alle Web-Applikationen, die in Modul 133 und 151 erstellt werden. Es ist verbindlich für dieses Web-Projekt.
- Bis auf wenige Ausnahmen dürfen keine Hilfsmittel, Bibliotheken oder Frameworks (Open Source, kommerzielle, etc.) eingesetzt werden. Erlaubte Frameworks bzw. Hilfsmittel sind:
 - Das MVC-Framework, das im Rahmen des Moduls 151 zur Verfügung gestellt wird.
 - Das Open Source Framework vom Berufsbildungszentrum Bern (BBC-MVC).
 - jQuery und Bootstrap
 - Weitere Hilfsmittel können von der Lehrperson zugelassen werden, wenn diese ausschliesslich das Design betreffen. Eine Absprache mit der Lehrperson ist zwingend!
- Die Applikation muss nach dem MVC-Muster (Model View Controller) strukturiert werden, siehe Kapitel 6 *Nicht-Funktionale Anforderungen*.
- Überflüssiger Code, überflüssige Funktionen, Klassen und Methoden führen zu einem Punkteabzug.
- Der Code muss selbstständig entwickelt worden sein. Fremdcode - sogenannte Snippets - müssen referenziert sein. Sie müssen jede Zeile ihres Codes kennen, auch den Code der Snippets!
- Es werden keine Einstellungen in Konfigurationsdateien vorgenommen, es wird mit der Grundkonfiguration von XAMPP (V 7.0.13) getestet. Einzige Ausnahme: Erstellen eines DNS-Namens, was z.B. für das BBC-MVC notwendig ist. Betroffene Dateien: *hosts* im Windows-Verzeichnis des Servers und *httpd-vhosts.conf* im Apache-Verzeichnis des Servers. Siehe auch nächstes Kapitel.
- Die *Weisungen für Leistungsbeurteilungen InformatikerInnen EFZ* regeln die verbindlichen Aspekte zu Leistungsbeurteilungen. Diese Weisungen kommen bei allen Leistungsbeurteilungen zum Einsatz. U.a. werden die Sanktionen für Plagiate und zu spätes Abgeben aufgeführt.

4. Vorgaben für die Abgabe und das Testen der Arbeit

- Der Lehrer testet das Endergebnis auf seiner Arbeitsumgebung. D.h. die Anwendungen von allen Schülern (und mehreren Klassen) werden parallel auf einem System installiert. Deshalb ist es unerlässlich, dass Sie sich an die nachfolgenden Vorgaben halten!
- Datenbankbenutzer = *root*, Passwort = *gibbiX12345*
- Name Datenbank: *bilderdb_[Klassenname 2 Zeichen]_[Nachname ohne Umlaute]*
Z.B. *bilderdb_4e_mueller*
Bitte alle Buchstaben klein schreiben, das Zeichen Underscore "_" und keine Umlaute verwenden (ä = ae, ö = oe, ü = ue).
- Der Zeichensatz der Datenbank muss *utf8* sein, die Kollation *utf8_german2_ci*. Die Datenbanken von allen Lernenden werden genauso auf dem System der Lehrperson erstellt.
- Damit die Lehrperson die Applikation testen kann, muss in der Datenbank vorhanden sein:
 - Benutzer mit der Mailadresse *test@gibb.ch* und dem Passwort *gibbiX12345*. Falls ein Benutzername verwendet wird: Benutzername = *test*.
 - Eine Galerie mit beliebigem Namen.
 - Mind. 5 Bilder und 1-2 Tags pro Bild, falls die Tags implementiert werden.
- Erstellen Sie ein Skript, das die komplette Datenbank erstellt, inklusive der notwendigen Daten gemäss obigem Punkt. Dazu können Sie in *phpMyAdmin* die Funktion *Exportieren* verwenden (verwenden Sie die Standardeinstellungen, ohne Erstellen der Datenbank).

- Grundsätzlich sollte die Applikation in jedem Verzeichnis unterhalb des Web-Roots laufen. Denken Sie daran, dass viele Anwendungen parallel installiert werden! Falls Sie das Installationsverzeichnis auf dem Lehrersystem wissen wollen:
C:\xampp\htdocs\m151\bilderdb\[Klassenname 2 Zeichen]\[Nachname ohne Umlaute],
z.B. C:\xampp\htdocs\m151\bilderdb\4e\mueller
- Wie oben beschreiben ist die einzig erlaubte Konfiguration die Angabe eines DNS-Namens. Dieser muss wie folgt aussehen: [Nachname ohne Umlaute].local, z.B. mueller.local. Die notwendigen Anpassungen in den Dateien *host* und *httpd-vhosts.conf* müssen in einer Read-Me-Datei dokumentiert sein
- Alle Dateien des Projektes werden gepackt (gezippt), Sie müssen genau eine Datei abgeben. Die Zip-Datei können Sie mit einem Passwort belegen, damit Klassenkameraden keinen Zugriff auf Ihre Applikation haben. Das Passwort können Sie via Mail mitteilen.
- **Abgabe des Projektes ist der Samstag der Kalenderwoche 18 (6. Mai 2017) um 24:00.**
- Die Zip-Datei muss auf dem Klassenshare unter *K:\inf2015.4x-151\Bilder-DB\Abgabe Projekt* abgelegt werden.
Achtung: Die ftp-Verbindung zum Klassenshare muss von zu Hause aus funktionieren, Anleitung dazu unter www.iet-gibb.ch.

5. Funktionale Anforderungen

Die beiden Funktionen *Registration* und *Login* befinden sich im öffentlichen Bereich. Alle übrigen Funktionen sind nur im Member-Bereich zugänglich.

Registration

Neue Benutzer können registriert werden. Als Benutzername wird eine E-Mail-Adresse verwendet, diese muss einmalig sein.

Zusatzfunktionen

- Die Benutzer können ihre Daten ändern.
- Die Benutzer können ihren Account löschen, inklusive aller Daten und Bilder.

Login

Die Benutzer können sich einloggen und gelangen in den Member-Bereich.

Logout

Die Benutzer können sich ausloggen und kehren in den öffentlichen Bereich zurück.

Fotogalerien

Ein Benutzer kann eine oder mehrere Fotogalerien anlegen. Jede Galerie besteht aus 0-n Bildern. Er kann nur seine eigene Galerie anschauen und bearbeiten.

Zusatzfunktionen

- Der Benutzer kann die Attribute einer Galerie (z.B. den Namen) ändern.
- Der Benutzer kann eine Galerie löschen.
- Die Benutzer können ihre Galerien für andere Benutzer zur Ansicht freigeben.

Fotogalerie

- Via Web-Interface können auf dem lokalen Computer Bilder ausgewählt und zu der Bildergalerie hinzugefügt werden. Dabei passiert folgendes:
 - Das ausgewählte Bild wird hochgeladen und in einem Verzeichnis auf dem Server oder in der Datenbank abgelegt.
 - Es soll möglich sein, Bilder bis zu einer Grösse von 4MB hochzuladen. Es ist erlaubt, grosse Bilder zu verkleinern.

- Achtung vor doppelten Bildernamen (d.h. der Dateiname sollte nicht einfach 1:1 vom Client übernommen werden).
- Vom ausgewählten Bild wird ein Thumbnail erstellt und ebenfalls in einem Verzeichnis auf dem Server oder in der Datenbank abgelegt.
- Das Seitenverhältnis bei den Thumbnails muss identisch sein mit dem Original, d.h. die Bilder dürfen nicht verzerrt werden.
- Für die Bildbearbeitung kann die PHP-GD-Library verwendet werden, die standardmässig mit XAMPP installiert wird.
- Die Bilder können mit einem Namen versehen werden.
- In einer Übersicht werden alle Thumbnails der Fotogalerie angezeigt.
- Beim Klicken auf einen Thumbnail wird das Originalbild angezeigt (max. Grösse = Bildschirm).
- Bilder können aus der Fotogalerie gelöscht werden.

Zusatzfunktionen

- Zu jedem Bild werden Informationen wie Datum, Dimension, Dateigrösse, etc. angezeigt.
- Die Bilder können nach verschiedenen Kriterien sortiert werden.
- Bei vielen Bildern werden die Thumbnails auf mehrere Seiten verteilt.
- Jedes Bild kann mit 0-n Tags (= Stichworte) versehen werden.
- Filterfunktion: Über ein Suchfeld können Begriffe eingegeben werden. Anschliessend werden nur noch diejenigen Bilder angezeigt, die mit diesen markiert (*getagged*) worden sind.
- Anstatt einem Textfeld für die Suche wird eine Auswahl mit allen vorhandenen Tags angezeigt.

Weitere Zusatzfunktionen

- Administrator mit dem Recht, alle Benutzer, Galerien und Bilder zu bearbeiten und zu löschen.
- Weitere Zusatzfunktionen nur in Absprache mit der Lehrperson!

6. Nicht-Funktionale Anforderungen

Speicherung der Bilder

Grundsätzlich gibt es zwei Möglichkeiten zur Speicherung von Bildern und Thumbnails: Entweder direkt in der Datenbank (Datentyp *Blob*) oder im Dateisystem des Servers. Erarbeiten Sie in einem ersten Schritt die Vor- und Nachteile der beiden Varianten, sowohl aus Sicht des Datenbankadministrators als auch aus Sicht des Entwicklers. Treffen Sie anschliessend die Wahl für Ihre Applikation.

MVC-Muster

Die Applikation muss nach dem MVC-Muster strukturiert werden. Es wird das einfache MVC-IET-gibb Framework zur Verfügung gestellt und empfohlen. Als Alternative steht das Open Source BBC-MVC zur Verfügung, oder die MVC-Struktur kann auch selber erstellt werden. Dabei ist es erlaubt, die Struktur eines eigenen, früheren Projektes zu übernehmen. Wird nicht nach dem MVC-Muster vorgegangen, werden bei der entsprechenden Bewertung keine Punkte vergeben.

Das MVC-IET-gibb Grundgerüst (ohne Funktionen) steht Ihnen auf dem Modulshare unter *AB151-07_Zusatzdokumente* zur Verfügung. Eine Installationsanleitung, eine Beispielapplikation und weitere Informationen zum MVC-IET-gibb finden Sie auf dem Modulshare, Arbeitsauftrag 5 *AB151-05MVC-IET-gibb_Einfuehrung.docx*.

Portabilität

Die Applikation muss den Ansprüchen der Plattformunabhängigkeit genügen. Am Ende des Projektes wird sie in die DMZ-Umgebung *vmLS2* portiert, was in die Bewertung einfliesst.

Unterschiedliche Bereiche

Es gibt einen öffentlichen Bereich, der für alle zugänglich ist: Der Bereich, wo man sich registrieren und einloggen kann. Daneben gibt es den Member-Bereich für angemeldete Benutzer. Die zwei Bereiche haben unterschiedliche Menüs.

Sicherheit

Die Sicherheit von Web-Applikationen wird erst nach dem Projekt behandelt. Die Anwendung soll trotzdem einigen (minimalen) Sicherheitsanforderungen genügen:

- Wie bereits im Modul 133 arbeiten wir mit Sessions:
 - Der Benutzer kann sich ein- und ausloggen. Dabei wird eine Session aufgebaut bzw. zerstört.
 - Member-Funktionen können ohne gültige Session nicht ausgeführt werden.
- Sichere Passwörter:
 - Formatvorgabe für Passwörter (mind. 8 Zeichen, mind. 1 Gross- und Kleinbuchstabe, 1 Ziffer, 1 Sonderzeichen)
 - Passwörter mit sicherem Hash versehen, inkl. Salt-Bestandteil (z.B. `password_hash()`)
- Daten sollen zum Speichern in die Datenbank maskiert werden (`mysqli_real_escape_string()`), um SQL-Injection zu verhindern.
- Daten sollen zur Ausgabe in HTML codiert werden (`htmlspecialchars()` bzw. `htmlspecialchars()`).

Zusätzliche Sicherheit

Zusätzlich eingebaute Elemente, die der Sicherheit dienen, werden mit Zusatzpunkten belohnt:

- Es soll verhindert werden, dass ein Benutzer Bilder in fremde Galerien hochladen und fremde Bilder bearbeiten oder löschen kann.
- Es soll verhindert werden, dass ein Benutzer die Bilder eines fremden Benutzers anzeigen kann (bei Ablage der Bilder im Dateisystem). Zwei mögliche Lösungen:
 - Via `.htaccess`-Datei wird die Berechtigung auf die Bilderverzeichnisse entzogen. Der Zugriff auf die Bilder erfolgt via PHP. Alternative: Die Bilder liegen ausserhalb des Webroots.
 - Die Dateinamen werden auf eine bestimmte Grösse gehasht, so dass die diese unmöglich erraten werden können. Zusätzlich muss verhindert werden, dass die Bilderverzeichnisse ausgelesen werden können.
- Die wohl beste Massnahme gegen diverse Bedrohungen ist die Validierung von sämtlichen Benutzereingaben (`$_POST` und `$_GET`).
- Einsatz von Prepared Statements.
- Die Session-ID wird nach dem Login und bei jeder Aktion geändert.

Design

Layout und Design werden in der Regel nicht vom Entwickler erstellt und deshalb nicht bewertet in diesem Projekt. Mit einem schönen Design können jedoch Zusatzpunkte erworben werden. Dabei ist der Einsatz von Bootstrap erlaubt.

Benutzerfreundlichkeit

Im Gegensatz zum Design liegt es mindestens teilweise in der Hand der Entwickler, die Applikation benutzerfreundlich und ergonomisch zu gestalten. Die einzelnen Anforderungen zum Erreichen dieses Zieles sind für alle Web-Applikationen gleich oder zumindest ähnlich, weshalb sie (mind. teilweise) im allgemeinen Dokument *M133-151Ergonomie_Web-Applikationen.pdf* aufgeführt sind. Das Nicht-Einhalten dieser Richtlinien führt bei den einzelnen Funktionen zu einem Punktabzug.

7. Dokumentation

Ein Dokument ist von allen zu erstellen, mit folgenden Bestandteilen:

- Arbeitsjournal, gemäss Kapitel 2 *Arbeitsjournal*.
- Vor- und Nachteile vom Speichern der Bilder in der Datenbank bzw. im Dateisystem.

- ER-Diagramm zum Abbilden der Datenbankstruktur.

Die Dokumentation wird ebenfalls bewertet, fehlende Dokumente = fehlende Punkte.

8. Zeitplan

Die Arbeit wird in mehreren Etappen durchgeführt. Das Einhalten des Zeitplans ist obligatorisch!

Wochen 1-3

- Arbeiten Sie sich in das MVC-IET-gibb bzw. BBC-MVC ein.
- Alternative: Erstellen Sie eine eigene MVC-Struktur.
- Treffen Sie die Entscheidung, ob die Bilder in der DB oder im Dateisystem gespeichert werden.
- Erstellen Sie das Datenbankmodell.
- Realisieren Sie die Registration und das Login.
- Testen Sie die beiden Funktionen ausführlich.
Ab Woche 4 müssen die Registration und das Login für den Test durch die Lehrperson bereit sein.
- Beginnen Sie mit der Realisierung der Bilderdatenbank.

Woche 4

- Review (Test) von Registration und Login durch die Lehrperson.
- Realisieren Sie die Bilderdatenbank.

Wochen 5-6

- Realisieren Sie die Bilderdatenbank.
- Testen Sie die Funktionen der Bilderdatenbank ausführlich.
- Portieren Sie die Applikation in die DMZ-Umgebung *vmLS2*.

Woche 7

In der Woche nach der Projektabgabe wird die Applikation kurz in der DMZ-Umgebung getestet.

9. Bewertung

Die Bewertung erfolgt gemäss dem beigelegten Bewertungsschema und setzt sich wie folgt zusammen:

- Obligatorische Funktionen = 60 Punkte
- Nichtfunktionale Anforderungen (ebenfalls obligatorisch) = 40 Punkte
- Zusätzliche Funktionen = 40 Punkte
- Von den zusätzlichen Funktionen können beliebige ausgewählt werden, jedoch sind max. 20 Punkte damit zu erreichen
- Notwendige Punktzahl für Note 6.0 = 110 Punkte

Daraus folgt:

- Mit den obligatorischen Funktionen und den nichtfunktionalen Anforderungen kann maximal die Note 5.5 erreicht werden.
- Es braucht 10 Punkte von den Zusatzfunktionen, um maximal die Note 6.0 zu erreichen.

Berechnung der Note

$$((\text{Pte. Funktionen} * \text{Faktor} / 6 + \text{Pte. weitere Anforderungen}) / \text{Pt.zahl für Note 6} * 5) + 1$$

Faktor

Mit jedem Lernenden wird eine Schlussbefragung durchgeführt. Bei dieser Befragung (2 Fragen) können 0-6 Punkte erreicht werden. Diese Punkte dienen als Faktor für die Berechnung der Punktzahl für die obligatorischen und zusätzlichen Funktionen.

Die Punktevergabe pro Frage (bei 2 Fragen) erfolgt nach den folgenden Kriterien:

3 = Beantwortung behandelt alle Aspekte professionell, fachlich kompetent und korrekt

2 = Beantwortung lässt nebensächliche(n) Aspekt(e) offen. Antworten meist korrekt

1 = Beantwortung lässt zentralen Aspekt ausser Acht. Antworten häufig unpräzise

0 = Kandidat/in kann Fragestellung nicht korrekt beantworten

Beispiel 1 Berechnung

- Obligatorische Funktionen: 58 Pte.
- Zusätzliche Funktionen: 14 Pte.
- Befragung: 4 Pte.
- Nichtfunktionale Anforderungen: 35 Pte.
- Punktzahl für Note 6: 110
- Note: $((72 * 4 / 6 + 35) / 110 * 5) + 1 = ((48 + 35) / 110 * 5) + 1 = (76.33 / 110 * 5) + 1 = 4.8$
- Wegen der Befragung bleiben von den 72 Punkten nur gerade 48 Punkte übrig, was die Note ziemlich hinunterdrückt

Beispiel 2 Berechnung

- Obligatorische Funktionen: 50 Pte.
- Zusätzliche Funktionen: 6 Pte.
- Befragung: 6 Pte.
- Nichtfunktionale Anforderungen: 35 Pte.
- Punktzahl für Note 6: 110
- Note: $((56 * 6 / 6 + 35) / 110 * 5) + 1 = ((56 + 35) / 110 * 5) + 1 = (91 / 110 * 5) + 1 = 5.1$
- Trotz der schlechteren Ausgangslage als bei Bsp. 1 erreicht dieser Lernende die bessere Note. Ganz einfach, weil er bei der Befragung keine Schwierigkeiten hatte, Auskunft über seinen Code zu geben. Deshalb fließen die 56 Punkte für die Funktionen voll in die Bewertung ein.