

Projektspezifikation

PictureCloud



I. Allgemein

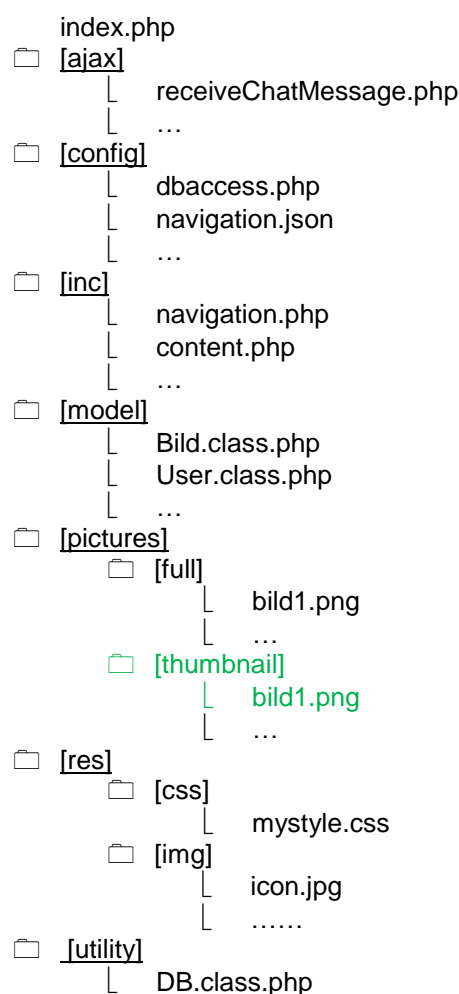
- a. Zum Abschluss der LV Web Engineering muss ein Web-Projekt umgesetzt werden. Die Gewichtung der einzelnen Aufgaben und die Punkteverteilung finden Sie in der zugehörigen Bewertungsmatrix.
- b. Verwenden Sie für die Umsetzung Teile Ihrer bereits abgegeben (und verbesserten) Übungen wieder.
- c. Die Bearbeitung erfolgt einzeln oder in 2er Teams.
 - i. **Blau markierte Aufgaben sind nur von 2er Teams zu bearbeiten, Einzelarbeiten lassen diese aus!**
- d. Das Abnahmegespräch erfolgt dann ebenfalls einzeln oder im Team, jedenfalls muss jedes Teammitglied das Projekt (Funktionsumfang und Code) kennen und Fragen dazu beantworten können (ähnlich den Übungsabnahmen)
- e. **Für manche in grün markierte Tasks, gibt es Extrapunkte zu erreichen. Die Aufgaben sind optional.**
- f. Die Angabe lässt gewisse Gestaltungs- und Implementierungsfreiheiten zu. Folgende **Basis-Features** müssen abgebildet werden:
 - i. Registrationsmöglichkeit und Profilbearbeitung für User
 - ii. Userverwaltung durch den Administrator
 - iii. Hochladen eigener Fotos (mit Geo-Informationen)
 - iv. Verwaltung der Bilder in einer Datenbank und übersichtliche Darstellung im Browser
 - v. Positionierung der Bilder auf einer Karte entsprechend ihrer Geo-Informationen (sofern beim Bild gespeichert)
 - vi. Tagging (Beschlagwortung) von Bildern inkl. zugehöriger Verwaltungsfunktionen
 - vii. Chat-Funktion
- g. Die Umsetzung des Projekts wird **sämtliche LV-Inhalte** abdecken:
 - i. Frontend: HTML5, CSS3, Bootstrap, JavaScript & jQuery (AJAX)
 - ii. PHP & OOP
 - iii. Fileupload
 - iv. MySQLi + Prepared Statements

- v. Authentifizierung
- vi. Kommunikation via RESTful-Web Services (JSON, PHP & AJAX mit jQuery)
- vii. Verwendung von HTML/CSS/JavaScript-Frameworks und PHP-Libraries
- h. Achten Sie auf eine **gute User Experience** innerhalb der Applikation (keine Whitepages, keine Fehlerzustände, guter Gesamteindruck, ...)

II. Projektstruktur und Grundlayout

- a. Erstellen Sie eine Seite **index.php**, welches das **Grundlayout** enthält und die Hauptbereiche der Bildverwaltung abdeckt sowie etwaige CSS und JS-Libraries einbindet. Die Inhalte aller Bereiche werden mittels include eingebunden. (Header, die Navigation, der Hauptinhaltsbereich ..).
- b. Die Projektstruktur der Bildverwaltung soll **Komponenten** klar **unterteilen** (siehe Beispielstruktur).

Beispielstruktur:



- c. Die Implementierung der Bildverwaltung ist mittels **OOP-Konzept** umzusetzen. Erstellen Sie Klassen für alle Entitäten, die im System gebraucht werden (bspw. User, Bilder, ...) und legen Sie eine DB-Service-Klasse an, welche **zentral** die Zugriffe auf die DB implementiert.
- d. Achten Sie beim Implementieren darauf, dass **Sicherheitslücken** möglichst **geschlossen** werden.
- e. Eine **Datenbank**, welche die Bildverwaltung abbildet, ist anzulegen.

III. Frameworks und Libraries

- a. Es bleibt Ihnen überlassen, ob Sie für die Implementierung diverse Frameworks oder Bibliotheken verwenden, wie z.B. Bootstrap, jQuery, jQueryUI, dropzone, leaflet etc.
- b. Die Verwendung von Fertiglösungen (wie CMS o.ä.) sind nicht erlaubt

IV. Abgabemodalitäten

- a. Das Projekt ist **vollständig** als zip/rar File in Moodle hochzuladen (maximal Dateigröße 20MB), inklusive einer Kopie der MySQL-Datenbank und einem Hinweis auf die Teammitglieder im Dateinamen, bspw:
 - i. **WEB-Projekt-2020_Nachname_Nachname.zip**
 - ii. Verkleinern Sie ggf. einige Beispielbilder, sollte das File für den Upload im Moodle zu groß sein oder splitten Sie das zip/rar-Archiv
- b. **Kommentieren** Sie komplexere Abläufe/Funktionen im Code
- c. Falls weitere Konfigurationsschritte zur Inbetriebnahme d. Applikation nötig sind, legen Sie eine **readme.txt**-Datei bei (z.B. Zugangsdaten zur Datenbank).

Grundidee

User der Website haben die Möglichkeit eigene Bilder hochzuladen, zu verwalten, zu taggen und Bilder mit unterschiedlichen Rechten freizugeben. Freigegebene Bilder anderer User können in der eigenen Galerie und auch auf einer Karte (bzw. in Google Maps) angezeigt werden. Schwerpunkte sollen sein:

- **User:** Userverwaltung (2-3 Zustände: anonym, (registrierter) User, [Admin](#)). Admins können auch User verwalten.
- **Datenbank:** Grundlegende Verwaltung aller Daten in einer Datenbank (User, Bilder, Tags, ...) → Aufbau einer geeigneten Datenbankstruktur, Zugriff mittels entsprechender DB-Zugriffs-Klassen
- **Upload / eigene Bilder:** User können eigene Bilder hochladen, diese verwalten (bearbeiten, löschen) und mit Tags versehen
- **Suche und Tagging:** Nach Bildern (eigene, fremde) kann anhand deren Tags und anderer Kriterien (z.B. Datum, Geolocations, ...) gesucht werden. Die geographische Zuordnung von Bildern soll anhand (optional) gespeicherter Geolocations auf einer Karte dargestellt werden.
- **Chat:** User haben die Möglichkeit sich mittels Chat-Funktion zu unterhalten (Senden/Empfangen von Nachrichten über kleines Chatfenster an der Oberfläche, siehe Facebook)
- weitere Bereiche der Webseite:
 - **Hilfe** (Benutzeranleitung)
 - **Impressum** (Standard-Impressum), **Namen und Bilder der Ersteller** der Website

Die gesamte Website muss **responsive** sein und somit für Smartphones, Tablets und Desktop-Rechner verwendbar sein.

Für eine außergewöhnlich gute **Usability**, Grundzüge der **Accessibility** und **Suchmaschinenoptimierung** werden Zusatzpunkte gewährt.

Userverwaltung

Es gibt 3 verschiedene Usertypen: **anonym**, (registrierter) **User**, **Admin**

Alle Usertypen sehen

- Hilfe und Impressum
- Suche und recherchieren in verfügbaren (freigegebenen) Bilder

Was kann der **anonyme User**:

- Registrierung und Login sind verfügbar
 - Hinweis: eine Registration über das Webinterface speichert immer normale User, Administratoren müssen direkt in der Datenbank als solche gekennzeichnet werden
- KEINE eigenen Bilder oder Uploadmöglichkeit
- Es werden aber alle fremden Bilder angezeigt (Bilder die irgendjemand anders hochgeladen UND öffentlich freigegeben hat). Weiters ist Suchen und Filtern unter diesen Bildern möglich, ebenso die Darstellung der Position auf einer Karte.

Was kann der **registrierte User** zusätzlich zum anonymen User:

- statt Registrierung und Login kann sich ein eingeloggter User wieder ausloggen
- Bearbeitung des eigenen Profils
- eigene Bilder verwalten (Bildinformationen, Upload, Tagging, Freigabe, Löschen)
- eine Tag Cloud (Häufigkeit der verwendeten Schlagwörter) anzeigen

Was kann der **Administrator** zusätzlich zum registrierten User:

- Verwaltung ALLER Bilder (also nicht nur eigener): Upload, Tagging, Löschen
- Userverwaltung (Übersicht bestehender und deaktivieren einzelner User)
- eine Tag Cloud (Häufigkeit aller verwendeten Schlagwörter aller User) anzeigen

Aufbau der Datenbank

Datenbankmodell:

Überlegen Sie sich ein **geeignetes Datenbankmodell**

Hinweise: Sie müssen User und ihre Rolle speichern, Bilder verwalten (die Bilder selbst liegen im Filesystem des Webserver, die Datenbank verwaltet nur Referenzen auf die Bilder), welche Tags gibt es und welche Tags sind welchen Bildern zugeordnet, wer ist der Besitzer eines Bildes, Freigabestatus der Bilder. Weiters wird empfohlen wichtige Kenndaten zu den Bildern ebenfalls in der Datenbank abzulegen, wie z.B. Bildname, Geoinformationen, Erstellungsdatum, Uploaddatum, ...), User sollen auch miteinander chatten können - somit ist auch eine zentrale Ablage von gesendeten/gelesenen Nachrichten nötig

Datenbankzugriff mittels PHP:

Der Datenbankzugriff hat über **mysqli** zu erfolgen (alternativ auch PDO möglich)

Die Zugangsdaten sollen in einer **zentralen config-Datei** gespeichert werden, damit sie an zentraler Stelle leicht zu ändern sind.

Die Kommunikation zwischen Datenbank und PHP muss über **PHP-Klassen** (objektorientierte Programmierung) realisiert werden.

Explizite Vorkehrungen **gegen SQL Injections** werden mit Extrapunkten belohnt.

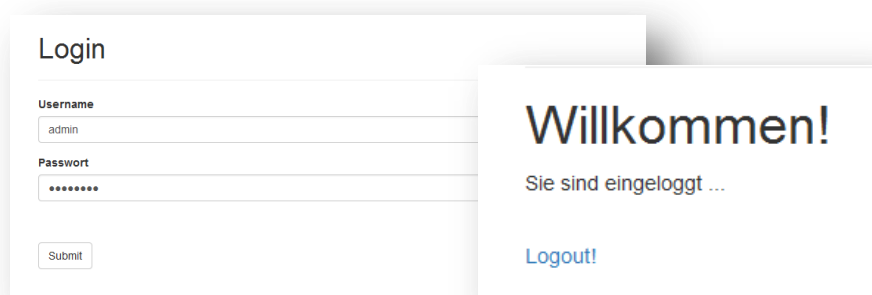
Beschreibung der einzelnen Komponenten

1) User-Registrierung

- a. Beim Erfassen eines Users sind (zumindest) folgende Daten anzugeben:
 - i. Anrede
 - ii. Vorname
 - iii. Nachname
 - iv. Adresse
 - v. PLZ
 - vi. Ort
 - vii. Emailadresse
 - viii. Benutzername
 - ix. Passwort
- b. Überprüfen Sie alle eingegebenen Daten sowohl clientseitig als auch serverseitig auf Vollständigkeit und Richtigkeit. Das Passwort muss immer 2x angegeben werden und die Werte werden auf Übereinstimmung geprüft. Der Username muss eindeutig sein (auch in der Datenbank). Erst wenn alle Daten validiert wurden, wird der neue User in der DB angelegt. Das Passwort wird verschlüsselt in der DB abgelegt.
- c. Legen Sie manuell in der Datenbank einen eigenen User an, der als Administrator gekennzeichnet wird. Die Rechte des Administrators unterscheiden sich von herkömmlichen Usern.

2) User-Login

- a. Erstellen Sie ein Login-Formular mit Eingaben für Usernamen und Passwort
- b. Überprüfen Sie die Werte gegen die Datenbank. Nur wenn es genau einen User in der DB mit den Daten gibt, gilt der Login als ok und der Username wird am Bildschirm angezeigt. Der Login-Status ist permanent auf der Seite sichtbar. Eingeloggte User sehen einen Logout-Button.
- c. Auf Basis des eingeloggten Users werden Features der Webseite freigeschaltet oder deaktiviert. (Als Beispiel folgender Screenshot)



The screenshot shows a web interface with a login form and a welcome message overlay. The login form, titled "Login", has two input fields: "Username" with the value "admin" and "Passwort" with masked characters "*****". A "Submit" button is at the bottom. Overlaid on the right is a white box with the text "Willkommen!" in large blue font, followed by "Sie sind eingeloggt ..." in smaller grey font, and a blue "Logout!" link at the bottom.

3) Profilverwaltung

- a. Eingeloggte User können eigene Registrationsdaten einsehen und bearbeiten (Stammdaten bearbeiten, Passwort neu setzen, ...)
- b. Achten Sie darauf, dass sensible Informationen (Passwort) nicht vollständig angezeigt werden und beim Ändern des Passwortes auch das alte Passwort eingegeben werden muss.

4) Hilfe / Impressum

- a. Erstellen Sie eine Hilfe-Seite, auf der erklärt wird, wie die Website zu verwenden ist (kurze Benutzeranleitung)
- b. Fügen Sie ein Impressum ein (→ recherchieren Sie, was aktuell ein Impressum enthalten muss).
- c. Das Impressum enthält zusätzlich Bilder und Namen der am Projekt beteiligten Personen
- d. Sowohl Hilfe als auch Impressum müssen jederzeit einfach erreichbar sein

5) Bildverwaltung generell

Die Bilderverwaltung besteht aus folgenden Funktionen:

- a. **Bildanzeige:** Anzeige der Bilder im Browser als Galerie (Thumbnails und zum Blättern) nach flexiblen Auswahl-Kriterien (zb Filter Tag, Name, Datum, Ort, ...)
- b. **Kartendarstellung:** Darstellung der Geo-Informationen der Bilder auf einer Karte (Google, OSM, ...)
- c. **FileUpload:** Hochladen eigener Bilder und Erfassen von Meta-Informationen (Name, Geo-Informationen, Tags, ...)
- d. **Freigabe von Bildern:** Bilder können für andere User oder öffentlich freigegeben werden (User muss bereits existieren)
- e. **Tagging & Suche:** Beschlagwortung von Bildern für flexible Suche- Filter- und Sortiermöglichkeiten

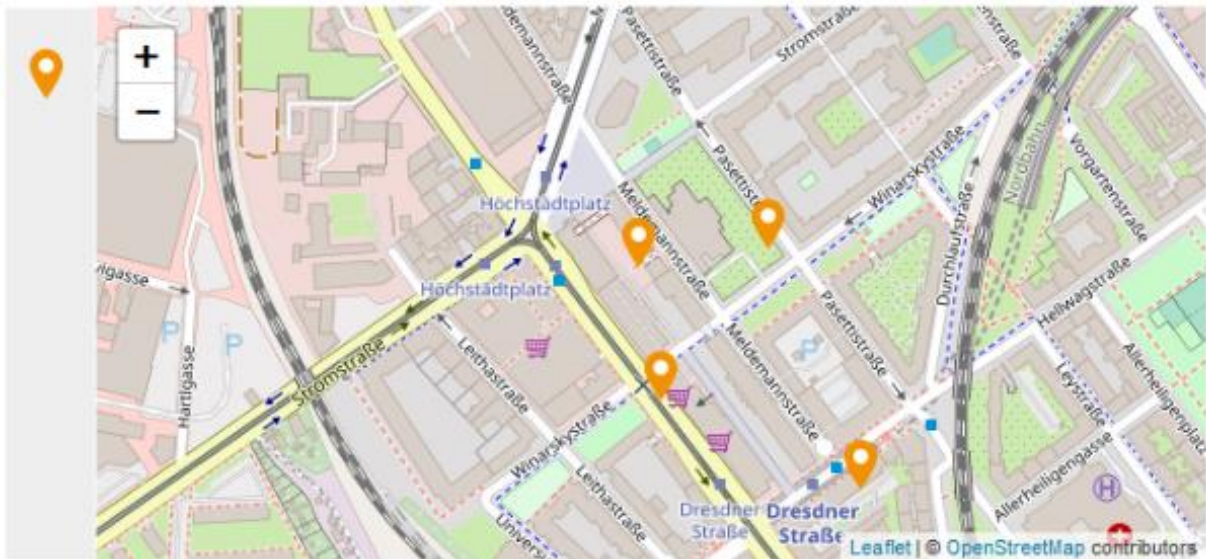
6) Bildverwaltung / Bildanzeige

Die Bildanzeige ist ein zentraler Bestandteil der Bildverwaltung. Die Darstellung soll sehr flexibel sein und Benutzern vielfältige Einstellmöglichkeiten bieten

- a. Bilder werden **als Thumbnails** konstanter Größe dargestellt. Achten Sie darauf, dass die Bilder, dort wo sie als Thumbnails benötigt werden, tatsächlich nur in Thumbnail-Größe vom Server zum Browser übertragen werden. Es empfiehlt sich die Thumbnails als fertige Bilder am Server in einem eigenen Verzeichnis vorzubereiten (automatische Verkleinerung mittels PHP).
- b. Die Darstellung soll übersichtlich sein, mehrere Bilder je Zeile.
- c. Es soll eine Auswahlmöglichkeit geben, welche Bilder angezeigt werden (eigene, freigegebene Bilder von unterschiedlichen Usern bzw. eine Kombination daraus). Es soll bei der Anzeige klar ersichtlich sein, zu welcher Kategorie (eigenes Bild, andere User, ...) ein Bild gehört.
- d. **Zusatzinformationen** zu Bildern werden angezeigt (permanent oder zum Ein-/Ausblenden), das sind z.B.
 - i. Bildname
 - ii. Position (Latitude, Longitude)
 - iii. Aufnahmedatum/Änderungsdatum
 - iv. Freigabestatus
- e. **Sortierung nach unterschiedlichen Kriterien entsprechend der Zusatzinformationen**
- f. (nur) **eigene Bilder können** auch wieder **gelöscht werden**.
- g. **Fancybox/Lightbox**: durch Klick auf ein Bild wird dieses im Großformat in einer Fancybox/Lightbox dargestellt. Es kann weiter zum nächsten Bild (laut Sortierung) oder zum Vorhergehenden gewechselt werden.

7) Bildverwaltung / Kartendarstellung

Binden Sie eine Kartendarstellung ein (z.B. Google oder Open Street Map).



Bilder aus der Bildanzeige werden automatisch positioniert, abhängig von ihren Geo-Informationen (= Marker werden gesetzt). Geo-Informationen können beim Bilder-Upload erfasst bzw. nachträglich bearbeitet werden. Für die Darstellung der Karte im Browser sowie deren Funktionalität werden JavaScript-Libraries empfohlen, wie z.B. Leaflet.

- a. **Bilder in der Bildanzeige werden optisch markiert**, wenn sie Geo-Informationen enthalten (damit der User sofort sieht, welches Bild Geoinformationen hat).
- b. **Zuordnung Bild -Position**: sehen Sie eine klare Zuordnung Bild – Kartenposition vor (bspw.: wird die Karte automatisch auf einen Marker zentriert, sobald ein Bild angeklickt wurde oder der Marker wird hervorgehoben ...)
- c. Alternativ kann ein Mini-Thumbnail des Bildes an der Kartenposition angezeigt werden.
- d. Die Kartendarstellung kann bei Bedarf ein- und ausgeblendet werden.

8) Bildverwaltung / Fileupload

Es können neue (eigene) Bilder aus dem Filesystem zum Server hochgeladen werden (per Drag&Drop via bspw. der JavaScript-Library dropzone). Dabei ist zu beachten, dass nur Bilddateien für den Upload erlaubt sein sollen.

Serverseitig müssen hochgeladene Bilder weiterbearbeitet werden:

- a. Hochgeladene Bilder werden immer dem gerade eingeloggten User als **Besitzer** zugeordnet
- b. **Speichern der Bilder** erfolgt in einem eigenen Bilderverzeichnis.
- c. Gleichzeitig wird die **Erstellung von Thumbnails** in einem eigenen Verzeichnis, für die spätere Verwendung in der Bildanzeige empfohlen.
- d. Die **Speicherung von häufig benötigten Informationen muss in der Datenbank erfolgen** (Bildname, Bildverzeichnisname, Geo-Informationen, Aufnahmedatum, ...)

9) Bildverwaltung / Tagging & Suche

Bilder im Anzeigebereich (nur eigene Bilder) können mit Tags versehen werden. Daraus ergeben sich weitere Möglichkeiten für die Bildanzeige:

- a. Tags können **beliebige Schlagwörter** sein (z.B. „Wien“, „Garten“, ...).
- b. In der Bildanzeige gibt es die Möglichkeit **nach Tags zu filtern** Es werden dann nur solche Bilder angezeigt.
- c. Man kann auch nach mehreren Tags gleichzeitig suchen.
- d. Tags können auch **wieder entfernt** werden.
- e. Es soll auf der Website eine **Auflistung aller bereits in der Datenbank erfassten Tags** geben, mittels **Drag&Drop** (z.B. mittels jQueryUI) können Tags einem Bild zugeordnet werden.
- f. Weiters gibt es ein **Suchfeld**. Eingegebene Begriffe werden in den Metadaten der Bilder gesucht (Bildname, Beschreibung, ...).
- g. **Continuous Search Filter**: Tippt der User bei der Suche einen Buchstaben ein, beginnt bereits die Suche. Basierend auf der Eingabe werden Bilder aus der Datenbank geholt und die Ergebnisse angezeigt, ohne einen Reload der Seite zu erwirken.

10) Bilder freigeben

- a. Bei jedem Bild gibt es die Option ein Bild für bestimmte User, oder öffentlich (für alle) freizugeben
- b. Bei der Freigabe für einen User, muss es diesen bereits in der Datenbank geben

- c. Bilder, die freigegeben wurden, werden entsprechend gekennzeichnet (mit den Usernamen oder einem Symbol für "öffentlich")
- d. Die Freigabe kann jederzeit wieder geändert werden (Freigabe für einzelne User oder alle löschen)
- e. Die Bildanzeige kann nach privaten und freigegebenen (anderer User) Bildern gefiltert werden.

11) Userverwaltung für Administratoren

Nur Administratoren haben die Möglichkeit User zu verwalten. Dazu gehört:

- a. Anzeige aller User und Auswahlmöglichkeit
- b. Zu jedem User eine Liste aller Bilder und der Freigaben
- c. Verändern des Userstatus: aktiv, inaktiv
- d. Inaktive User können sich nicht mehr einloggen, bleiben aber in der Datenbank erhalten.
- e. Reset des Passwortes eines bestimmten Users und Zusenden des neuen Passwortes an die eingetragene Emailadresse (Achtung: Email-Versand bei XAMPP erfordert mitunter Anpassung der PHP-Settings).

12) Chat Funktion

- a. Ergänzen Sie die Website um ein Chat-Fenster, das man ein- und ausblenden kann
- b. User können miteinander Kontakt aufnehmen und sich Nachrichten senden (ähnlich zu FB-Chat)
- c. Auf der Oberfläche wird eine Liste aller User angezeigt. Aktuell "aktive" User werden markiert.
- d. An gerade eingeloggte User können Nachrichten versendet werden
- e. Nachrichten werden via AJAX versendet und empfangen, sodass eine Unterhaltung in Echtzeit möglich ist, ohne Reload der Seite.
- f. Gelesene/Ungelesene Nachrichten sollen entsprechend markiert werden.
- g. Nachrichten werden im JSON-Format vom und zum Server übertragen.