M151 Projekt Dokumentation

Inhalt

[Ziel der Website 2](#_Toc106621519)

[Meilensteine 2](#_Toc106621520)

[Technologien 2](#_Toc106621521)

[Thema Sicherheit 2](#_Toc106621522)

[Man in the middle (Man-in-the-Browser) 2](#_Toc106621523)

[Denial of Service, Distributed Denial of Service 2](#_Toc106621524)

[Cross-Site-Request-Forgery 2](#_Toc106621525)

[SQL-Injection 2](#_Toc106621526)

[Cross Site Scripting 2](#_Toc106621527)

[Session Hijacking 3](#_Toc106621528)

[4-Tier Architektur 3](#_Toc106621529)

[Presentation Layer 3](#_Toc106621530)

[Data Service Layer 4](#_Toc106621531)

[Business Logic Layer 4](#_Toc106621532)

[Data Access Layer 4](#_Toc106621533)

[CRUD 4](#_Toc106621534)

[Sessions / Transaktionen / SSL / TLS 4](#_Toc106621535)

[Session 4](#_Toc106621536)

[Transaktionen 4](#_Toc106621537)

[SSL/TLS 4](#_Toc106621538)

[Arbeitsjournal 5](#_Toc106621539)

[23.5.2022 5](#_Toc106621540)

[30.5.2022 5](#_Toc106621541)

[13.6.2022 5](#_Toc106621542)

[20.6.2022 5](#_Toc106621543)

# Ziel der Website

Das Ziel unsere Website ist es Bilder in eine Datenbank hochzuladen und dort zu speichern. Zudem sollte jeder User sein Persönliches Login haben. Die User können die Bilder, welche sie hochgeladen haben, veröffentlichen und andere veröffentlichte Bilder liken. Die meistgelikten Bilder können aufgelistet werden.

# Meilensteine

* Registration
* Login
* Bilder können hochgeladen und angezeigt werden
* Bilder können geliket werden

# Technologien

Wir wollen für unsere Website HTML&CSS verwenden. Für die Logik unserer Seite nutzen wir PHP, da wir damit schon andere Webseiten gemacht haben und es die einzige Webtechnologie ist, die wir in der Schule vertieft angeschaut haben. Als DBMS brauchen wir PhpMyAdmin mit MySQL.

# Thema Sicherheit

### Man in the Middle (Man-in-the-Browser)

Um die Benutzer unserer Website gegen «Man in the Middle» zu schützen, hat jeder Benutzer sein eigenes Login. Um ganz sicher zu gehen kann man eine Zwei-Faktor-Authentifizierung verwenden. Noch sicherer wird es, wenn der Benutzer alle drei Monate das Passwort ändern muss. Da unsere Website keine sehr privaten Daten enthalten verzichten wir auf die letzten 2 Punkte. Abgesehen vom Passwort verwenden wir https und nicht http damit alles verschlüsselt ist.

### Denial of Service, Distributed Denial of Service

Damit unser System nicht durch ein (Distributed) Denial of Service Angriff lahmgelegt wird, müssen wir unsere Website bei jemandem Hosten der eine Firewall zur Verfügung stellt und genug Leistung besitz damit kleinere Angriff keinen schaden anrichten.

### Cross-Site-Request-Forgery

Um sich gegen Cross-Site-Request-Forgery zu schützen, benutzen wir ein Session-Token. Eine Zufalls Zahl mit mindestens 4-Stellen. Diese Zahl wird bei jedem http-Request aus dem Frontend z.B. über ein verstecktes Feld mitgegeben und kann dann überprüft werden, ob es immer noch die gleiche ist.

### SQL-Injection

Um einen solchen Angriff zu verhindern, sollen Sonderzeichen rausgefiltert werden. Zudem sollte man nur Anfragen erlauben, die auch den gewünschten Datentyp liefern und nicht irgendeinen. Zusätzlich sollte man auf sogenannte Prepared-Statments (vereinfachts SQL für häufig gebrauchts SQL) nicht benutze da sie eine Sicherheitslücke sind. Auf Seiten des Datenbankmanagementsystem muss man schauen das es möglichst wenig Benutzerkonten hat und mit nur so viel rechte wie nötig. Das System muss man immer schön Updaten und nicht benutzte Dienste ausschalten.

### Cross Site Scripting

Wir sollten sicher darauf achten das nur Bilder in der Datenbank gespeicherte werden können und keine manipulierten Dateien, die unsere Datenbank kaputt machen. Das gleiche gilt für die Registration, hier müssen wir schauen das der Benutzername oder Password nicht schädlicher JavaScript Code oder SQL enthält. Um solche Angriffe zu verhindern, müssen wir Sonderzeichen bei Inputfelder verbieten oder wegschneiden zum Beispiel Semikolons zum Schluss oder ‘</>’ wegschneiden. Zusätzlich können wir Libaries verwenden, welche den Text noch genauer auf schädlichen Code überprüft. Das Bild sollten wir auf jeden Fall auf den Dateityp überprüfen und nur zulassen, wenn das Dateiformat der Bilder JPEG oder PNG ist, damit kein schädliches Skript abgespeichert werden kann.

### Session Hijacking

Unsere Website verwendet https damit das Cookie verschlüsselt wird und für den Hacker schwere erreichbar ist. Ein **Intrusion Detection System ist ein System, das jede Veränderung im Netzwerk war, nimmt und dich bei Bedrohung wart. Jedoch wird das System nur von grossen Firmen verwendet und wäre zu viel des Guten für unser Projekt.**

Abgesehen von den verschiedenen Angriffsmöglichkeiten und deren Präventationsmöglichkeiten wollen wir die öffentlichen Daten, in unserem Fall die Bilder von den privaten Logindaten trennen. Damit im Fall eines Angriffs die privaten Daten nicht erreichbar sind.

# 4-Tier Architektur

Die vier Schichten der 4-Tier Architektur sind Presentation Layer, Data Service Layer, Business Logic Layer und Data Access Layer.

Diagram

Description automatically generated

Wie aus dem Diagramm hervorgeht, lassen sich die Schichten in zwei Typen unterteilen. Die eine ist die Data Access Layer, die Business Logic Layer und die, Data Service Layer, die sich auf der Serverseite befindet, sowie das Geschäftsentitätsmodell und die generische Klassenbibliothek. Die andere ist die Presentation Layer, die sich auf dem Client befindet.

## Presentation Layer

Die Presentation Layer sorgt für Anzeige von Daten, welche es vom Server bekommt, interpretiert Benutzer Befehle und sendet Daten an den Server. Es ist die Schnittstelle zum User und bietet eine graphische Benutzeroberfläche.

## Data Service Layer

Die Data Service Layer ist der Botschafter zwischen der Presentation Layer und Business Logic Layer. Sie sorgt für Schutz gegen Einsicht in die Business Logic Layer und konvertiert Daten, die in eine der zwei Richtungen gehen, in die erforderliche Form um.

## Business Logic Layer

Die Business Logic Layer ist für Geschäftsprozesse verantwortlich und verarbeitet Daten. Sie ist wie das Gehirn der Applikation.

## Data Access Layer

Die Data Access Layer ist für den Zugriff auf die Datenbank verantwortlich. Sie ändert, löscht, schreibt und sucht Daten auf der Datenbank.

# CRUD

Der Begriff CRUD steht in engem Zusammenhang mit der Verwaltung digitaler Daten. Genauer gesagt ist CRUD ein Akronym für die Namen von vier grundlegenden Operationen, die von persistenten Datenbankanwendungen abgeleitet werden:

* **C**reate (Datensatz anlegen)
* **R**ead bzw. **R**etrieve (Datensatz lesen)
* **U**pdate (Datensatz aktualisieren)
* **D**elete bzw. **D**estroy (Datensatz löschen)

Einfach ausgedrückt fasst der Begriff CRUD die Funktionalität zusammen, die Benutzer zum Erstellen und Verwalten von Daten benötigen. Verschiedene Datenverwaltungsprozesse basieren auf CRUD, sodass der Betrieb speziell auf die Anforderungen des Systems und des Benutzers zugeschnitten ist – egal, ob es sich um die Verwaltung einer Datenbank oder die Nutzung einer Anwendung handelt. Operationen sind ein typisches und unverzichtbares Zugriffstool, mit dem Experten beispielsweise auf Datenbankprobleme prüfen. CRUD bedeutet, dass Benutzer jederzeit Konten erstellen und Zeiten verwenden, anpassen oder löschen können. Außerdem werden CRUD-Operationen je nach Datenbanktechnologie sehr unterschiedlich ausgeführt.

# Sessions / Transaktionen / SSL / TLS

## Session

Wir starten unsere Session mit der PHP-Funktion session\_start(). Danach können wir mit $\_SESSION[name] auf Seiten übergreifende Informationen zugreifen. Die Session beginnt mit dem Login und wird mit dem Logout wieder beendet.

## Transaktionen

Wir überprüfen immer, ob eine Transaktion vollständig durchgeführt wurde. Wenn nicht wird sie wieder Rückgängig gemacht und nicht durchgeführt.

## SSL/TLS

PHP hat verschiedene eigene Funktionen um SSL/TLS zu konfigurieren. Mit local\_cert() kann man zum Beispiel den Dateipfad zu der lokalen Datei eingeben.

# Arbeitsjournal

## 23.5.2022

Linus: Heute habe ich mich über die Anforderungen informiert und die Struktur für die Doku erstellt. Das heisst ich habe alle Überschriften gemacht für die Themen, die in die Doku sollen, sowie Kopfzeile und Inhaltsverzeichnis. Danach hatte ich noch Zeit, um mich mit Sven über das Ziel und die Technologie auszutauschen. Wir hatten ein bisschen Probleme uns auf die Technologie zu entscheiden, da wir beide keine Technologie haben, bei der wir uns 100% wohl fühlen. Danach fing ich noch an mit der Dokumentation von Ziel und Technologie.

Sven: Ich habe mich mit Linus über die Anforderungen informiert. Danach habe ich mich mit Linus über die Technologie ausgetauscht und schon angefangen die Grundstruktur der Applikation zu erstellen.

## 30.5.2022

Linus: Am Anfang der Lektionen habe ich noch das Dokumentieren der Technologie und Ziel fertig gemacht. Es war nicht ganz einfach an alles Besprochene von letzter Woche zu erinnern. Als ich damit fertig war widmete ich mich den Meilensteinen. Nachdem ich diesen Teil auch fertig gemacht habe, fing ich an mit dem Programmieren, und zwar mit dem Funktionen.php um schon die ersten MySQL Anfragen zu machen. Danach habe ich auch noch das Thema Sicherheit ausgefüllt, soweit ich es ohne Recherche ausfüllen konnte.

Sven: Ich habe die Login Page und die Funktionen dazu gemacht und habe noch mit der Homepage angefangen. Ich kam am Anfang nur langsam voran, da ich mich nicht mehr so gut mit PHP ausgekannt habe, aber später hatte ich es wieder im Griff. Ich habe noch Probleme mit dem Git, ich kann aus irgendeinem Grund nichts pushen, es kommt immer irgendeine Fehlermeldung.

## 13.6.2022

Linus: Nachdem ich das letzte Mal mit dem Thema Sicherheit begonnen habe, habe ich heute die Recherche gemacht und das Thema abgeschlossen. Es war erstaunlich einfach die Informationen zu finden, doch manchmal sind die Schutzmöglichkeiten ziemlich kompliziert und sind oft zu krass und gross für unsere kleine Website. Danach ergänzte ich noch die Funktionen.php File mit ein paar neuen Funktionen während ich noch das most\_liked.php machte, dieses ist notwendig, um nach den meist gelikten Bilder zu filtern. Zum Glück konnte ich dann sogar noch mit dem edit.php beginnen, in diesem sollte man sein eigenes Profil bearbeiten. Dieser Tag war ziemlich stressig da wir heute möglichst weit kommen wollte, um nächste Woche das Projekt gut abschliessen zu können.

Sven: Ich habe heute meine Recherchen zu den Technologien gemacht und die Doku mit den Informationen ergänzt. Das hat länger gedauert als geplant, aber ich konnte dann noch die Homepage und die Registrierung fertig machen. Ich hatte am Anfang noch Probleme mit der Registrierung aber am Schluss hat es funktioniert. Ausserdem habe ich geschafft mein Code ins Repository zu pushen.

## 20.6.2022

Linus: Heute letzter Tag habe ich begonnen mit einer Absprache mit Sven was es noch zu tun gibt. Danach mussten wir zuerst ein kleines Problem bei Git beheben. Danach haben wir noch die Fehlenden Punkte für die Doku aufgeteilt und gelöst.

Sven: Am letzten Tag habe ich noch die Upload Funktion eingebaut und die SQL-Datenbank mit initialisierten Daten ins Repository hochgeladen. Abschliessend habe ich noch die Recherchen zu den Technologien in der Doku ergänzt.