

Projekt: *SpoMan*

Klasse: 2TPIF2

Schüler-Name: _____

Einführung

Deine Aufgabe während des Projektes ist es eine Webseiten-Applikation zu entwickeln, deren Zweck es ist **Spenden für einen Guten Zweck** zu sammeln.

In Bezug auf z.B. die Ukraine-Krise möchten **mehrere Klassen/Schulen** zusammen **eine oder mehrere Spendenaktionen** (auch noch **Aktivität** genannt) organisieren. Für solche Spendenaktionen werben die **Schüler** in ihren **Familien** und in ihrem **Bekanntenkreis** während **ein paar Tagen/Wochen um Sponsoren**. **Die Aktivität wird dann geplant(!) von den Schülern durchgeführt.** Dabei werden die Schüler ihr Bestes geben bzw. versuchen bestmögliche Resultate zu erzeugen, um so, soviel Geld wie möglich von den Sponsoren für den jeweiligen **Guten Zweck** zu sammeln.

Mittels deiner **WebApp zu verwalten** sind folglich:

- **Schulen***
- **Klassen**
- **Schüler**
- **Aktivitäten**
- **Sponsoren**
- **Versprechen¹**
- **Resultate²**
- **...***

Auf der Webseite müssen, **in Echtzeit, der bereits verdiente Geld-Betrag pro Schüler, Klassen und Schulen*** miteinander verglichen werden können. Dies soll zur positiven **Konkurrenz** führen und somit die Motivation der Schüler wie auch die der Sponsoren steigern um noch Spenden zu generieren.

* : optional

¹ Versprechen: zu spenden versprochener Geldbetrag pro Resultat

² Resultat: *Runden* bzw. [km] bei einer Aktivität vom Typ Laufen / Radfahren,
Punkte bei einer Aktivität vom Typ Lernen, *Schwimmbad-Längen* bzw. [m] beim Schwimmen, ...

Ablauf des Projektes

→ Dieses Dokument ist zuhause auszudrucken und in jedem WSERS-Kurs dabeizuhaben!

→ Das Projekt startet am **18. Mai**.

→ Die Abgabe des Projektes ist am **30. Juni**.

Das Projekt wird in sechs Phasen eingeteilt, deren Durchlauf in deiner Verantwortung liegt:

1. Während der Planungsphase musst du dich **mit dem Projekt vertraut machen** und einen Zeitplan für die verschiedenen Projekt-Etappen aufstellen.
 - ☐ Skizzen zum besseren Verständnis
 - ☐ Dem Lehrer Fragen zum Projekt stellen
 - ☐ Excel-Datei mit Milestones erstellen und auf eduMoodle hochladen.
2. Du musst eine ansprechende **Webseite planen** die den gewünschten Kriterien entspricht (unterschätze diesen Zeitaufwand nicht!!)
 - ☐ Skizzen auf Papier (verschiedene Layouts/Funktionalitäten)
 - ☐ Wireframes / Mockups zuerst auf Papier, danach digital
 - ☐ Benötigte Datenbank-attribute erkennen
3. Du musst eine **Datenbank planen, erstellen** und mit validen **Test-Daten** füllen.
 - ☐ MCD (KDM) zuerst auf Papier, danach digital
 - ☐ MLD (LDM) zuerst auf Papier, danach digital
 - ☐ SQL-Script* / phpMyAdmin zum Anlegen der DB – mysql-Server
 - ☐ MLD (LDM) mit Test-Daten validieren, gegebenenfalls Verbesserungen vornehmen
4. Du musst eine ansprechende **Webseite implementieren** (html, css, js, php und sql)
 - ☐ Denke an den Aufbau komplexerer Websites (Templates / Projekt vom WSERS1-Kurs)
 - ☐ Modulare Programmierung (Datenstrukturen, Funktionen, Dokumentieren sowie Auslagerung von Code, ...)
 - ☐ Benutze JavaScript um evt. irrtümlich vom Benutzer gestartete Aktionen, doch noch abbrechen zu können. (vor allem beim Löschen!!)
 - ☐ Deine Abfragen (queries) sollten Injection-safe sein
5. Während der **Test- und Debugging-Phase** musst du deine Webseite auf Fehler überprüfen und gegebenenfalls verbessern.
 - ☐ Nutze die Developer-Tools im Browser
 - ☐ Erstelle dir Test-Szenarien, zuerst auf Papier, danach mithilfe deiner Web-App
6. In der Woche vom 04.07.2022 musst du dein **Projekt vorstellen**.
 - ☐ Bereite deinen Pitch zuhause vor!

Datenbank

Mit deiner Datenbank sollen Schulen*, Klassen, Schüler und Sponsoren verwaltet werden. Die einzelnen Spenden sollen ebenfalls abgespeichert werden. Deine Datenbank soll die Funktionalitäten deiner Web-App unterstützen.

Mache dir zuerst klar, was die Funktionen deiner Seite sein sollen und führe wie beim miniProjekt eine Liste der benötigten Attribute und deren Datentypen.

Modelliere zuerst die Datenbank mithilfe eines MCD³ und zeige es deinem Lehrer, bevor du weiter machst. Erstelle nun aus dem MCD ein MLD⁴ und anschließend deine MySQL-Datenbank, die du entweder mittels MySQL-Skript oder phpMyAdmin erstellst.

Kriterien für deine Datenbank:

- Must:

- ☐ Erstelle eine Liste der benötigten Einträge der Datenbank
- ☐ Erstelle ein MCD entsprechend dieser Angabe und deiner selbst definierten Web-App-Funktionalitäten
- ☐ Übersetze das MCD in das MLD
- ☐ Erstelle deine MySQL Datenbank

- Should:

- Could:

- ☐ Benutze ein MySQL-Skript, um die Datenbank zu erstellen
- ☐ Schreibe ein weiteres MySQL-Skript, welches die Datenbank mit Daten füllt

³ de: KDM – Konzeptionelles Datenmodell

⁴ de: LDM – Logisches Datenmodell

Webseite

Auf deiner Webseite soll es möglich sein sich entweder als **Administrator**, als **Schüler** oder als **Sponsor*** mithilfe jeweiliger Passwörter einzuloggen.

Ein **Administrator** kann **Schulen***, Klassen, Schüler und Sponsoren verwalten.

Wenn sich ein **Schüler** einloggt, dann soll er **alle** seine **Sponsoren** sehen, seine **erzielten Resultate** und seinen **bereits gesammelten Spendenbetrag pro durchgeführter Aktivität**. Er muss die Möglichkeit haben seine Sponsoren in die Datenbank einzutragen.

[Ein **Sponsor** muss sich ein Konto erstellen, falls er noch keines besitzt. Nach dem Einloggen kann er einen oder mehrere Schüler unterstützen und sich seine Spenden/Versprechen anzeigen lassen. Wegen Datenschutzgründen muss er den Untis-Code des Schülers kennen. Wird diese Funktionalität eingebunden, so muss dies der Schüler natürlich nicht mehr tun]*

Das Web-Interface des Administrators muss den folgenden Kriterien entsprechen:

- Must:
 - ☐ Es ist möglich, sich im Web-Interface per Benutzername/Passwort einzuloggen.
 - ☐ Der Administrator kann sein Passwort/Profil* ändern.
 - ☐ Der Administrator kann Klassen verwalten
 - ☐ Der Administrator kann die Schüler einer Klasse verwalten.
 - ☐ Die Änderungen werden jeweils korrekt in der Datenbank abgespeichert.
- Should:
 - ☐ Die Webseite ist gegen SQL-Injektions abgesichert.
 - ☐ Das Passwort wird sicher in der Datenbank abgespeichert.
 - ☐ Code entspricht unserem Qualitätsstandart.
 - ☐ Die Webseite hat ein ansprechendes Design. (CSS)

- Could:
 - ☐ Der Administrator kann Schulen verwalten.
 - ☐ Der Administrator kann eine Schule aus einer Liste auswählen um z.B. Infos zur Schule anzuzeigen (Adresse, Webseite, ... aber auch evt. bereits unterstützte Aktivitäten und deren Resultate/Geld-Beträge).

Das Web-Interface des Schülers muss den folgenden Kriterien entsprechen:

- Must:
 - ☐ Es ist möglich, sich im Web-Interface per Benutzername/Passwort einzuloggen.
 - ☐ Der Schüler kann sein Passwort/Profil* ändern.
 - ☐ Der Schüler kann sich seine Sponsoren verwalten/nur anschauen*.
 - ☐ Der Schüler erkennt die Herausforderungen, die ihm/der Klasse/der Schule gestellt werden.
- Should:
 - ☐ Die Webseite ist gegen SQL-Injektions abgesichert.
 - ☐ Code entspricht unserem Qualitätsstandart.
 - ☐ Die Webseite hat ein ansprechendes Design. (CSS)

- Could:
 -
 -
 -
 -

Das Web-Interface des **Sponsors*** muss den folgenden Kriterien entsprechen:

- **Must:**
 - ☐ Der Sponsor kann sich ein Konto erstellen.
 - ☐ Falls bereits ein Konto existiert, kann der Sponsor sich einloggen.
 - ☐ Der Sponsor kann sein Passwort/Profil ändern.
 - ☐ Mithilfe des Untis-Codes, kann der Sponsor einen Schüler unterstützen.
- **Should:**
 - ☐ Die Webseite ist gegen SQL-Injektions abgesichert.
 - ☐ Der Sponsor kann sich alle Schüler anzeigen lassen, denen er Geld zu spenden hat bzw. bereits gespendet hat.
 - ☐ Der Sponsor kann eine fehlerhafte Spende rückgängig machen – sofern die Aktivität noch nicht gestartet wurde.
 - ☐ Code entspricht unserem Qualitätsstandart.
 - ☐ Die Webseite hat ein ansprechendes Design. (CSS)

- **Could:**
 - ☐ Der Sponsor wird via email um das Einhalten seines Versprechens gebeten (Geldüberweisung) sobald die Aktivität beendet ist.
 - ☐ Der Sponsor wird via email um eine Bestätigung gebeten, bevor sein Konto erstellt bzw. aktiviert wird (Bestätigung durch Link-Aktivierung).

Die aktuellen Statistiken der Webseite sollen öffentlich zugänglich sein. Diese ermöglichen das Vergleichen der Spenden der einzelnen Schulen*.

Wenn man eine Schule auswählt, werden die Spenden der einzelnen Klassen miteinander verglichen*.

Falls man zusätzlich als Schüler bereits eingeloggt ist, dann kann man die Spenden innerhalb seiner Klasse betrachten. Beachte, dass die Namen der Schüler wegen Datenschutzgründen nicht öffentlich preisgegeben werden können.

Die öffentlich zugängliche Webseite soll den folgenden Anforderungen entsprechen:

- Must:
 - ☐ Man kann die gesammelten Spenden der Schulen* miteinander vergleichen.
 - ☐ Man kann die gesammelten Spenden/Resultate verschiedenen Klassen einer Schule miteinander vergleichen.
 - ☐ Falls man als Schüler eingeloggt ist kann man die Spenden innerhalb seiner Klasse vergleichen.
- Should:
 - ☐ Die Webseite ist gegen SQL-Injektions abgesichert.
 - ☐ Code entspricht unserem Qualitätsstandart.
 - ☐ Die Webseite hat ein ansprechendes Design. (CSS)

- Could:
 - ☐ Die Webseite ist „responsive“, d.h. sie passt sich der Bildschirmgröße (PC, Tablett, Smartphone, ...) an um sauber und elegant dargestellt zu werden – dies erhöht die *Lesbarkeit* und somit die *User Experience* enorm.

Woche	Geplante Aufgaben	Check (done)	Check (tested)	Probleme	Check (gelöst)
W1					
W2					
W3					
W4					
W5					
W6					

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.