Sport für alle

Factsheet XML Blockwoche

**Autoren**

Julien Grüter

Joshua Heller

Maurizio Hostettler

Stefan Stofer

**Semester**

HSLU.BSCI.HS19.XML

**Webseite**

<http://xml.joshuart.ch/index.php>

# <Auftrag />

**Idee**

Kundensegment im Behindertensport erweitern, Serviceangebote vergrössern im Bereich Sport und Erholung. Die Erreichbarkeit durch einen Webauftritt gewährleisten

**Vorteile**

Die Webseite bietet einen Barrierefreier Zugang, sie bietet direkt Online ein Buchungssystem inklusive Belegungsdiagramm. Eine automatisierte Bestätigung liefert direkt ein PDF Dokument.

**Vision**

Ein neues Zentrum für Behindertensport etablieren, die Akzeptanz für Behindertensport fördern und dank dem erweiterten Angebot können Meisterschaften im Ort durchgeführt werden.

# <Architektur />

Im Web Browser werden die Technologien (X)HTML, CSS und JavaScript genutzt. Die Kommunikation zwischen dem Client und Webserver verläuft über PHP. Die Verarbeitung, Darstellung und Quelle der Daten werden XML Technologien genutzt.

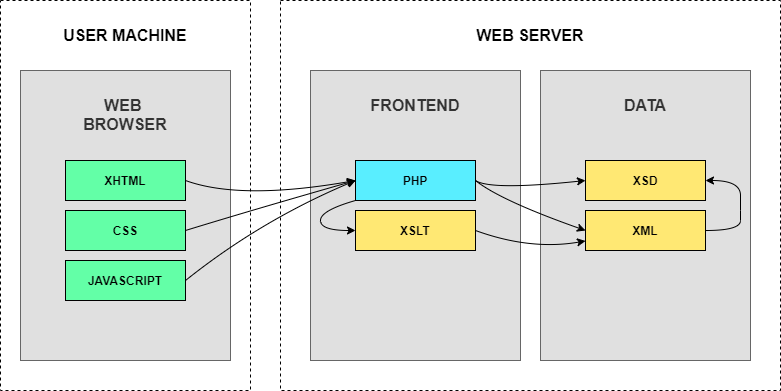


Abbildung 1 Technologienübersicht

Die Darstellung zeigt alle definierten Use Cases. Bei http GET wird ein XML Dokument über ein XSL zu einer XHTML Seite transformiert. Bei einer http POST Methode wird immer eine serverseitige Validierung vorgenommen. Dafür werden die Werte gegen ein XML Schema geprüft.

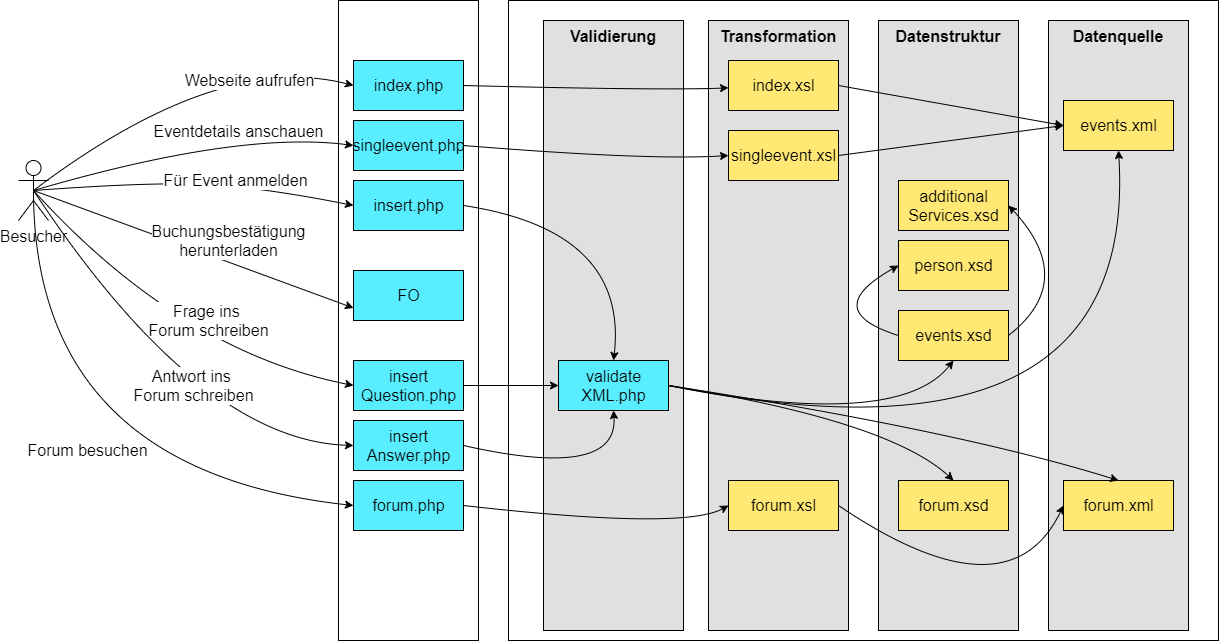


Abbildung 2 Detailierte Use-Case Übersicht

# <Verwendete Frameworks />

**Bootstrap**

Das Framework wird für Design (unter anderem auch für Responsive Design) genutzt.

**Fontawesome**

Es werden die Icons für Öffnungszeiten, Standort und Kontakt genutzt.

# <Technische Stolpersteine />

**XSLT in XML Quelle ersetzen**

Das XML «events.xml» wird für mehrere Funktionen (Übersicht, Anmeldeseite, Buchungsbestätigung) der Webseite als Datenbasis genutzt. Dabei soll für jede Funktion ein anderes XSLT verwendet werden. Das Problem wird mit der PHP-Klasse «XSLTProcessor» gelöst. Damit wird beim Aufruf von einem der genannten Funktionen das XSL Dokument ausgetauscht.

**Docker Read-only**

**Zu Beginn haben wir ein Dockerfile genutzt, um den PHP Webserver, mit der Webseite als Inhalt, zu bauen. Die ersten Probleme sind beim Anmeldeformular aufgetreten. Die Anmeldedaten können nicht ins XML geschrieben werden, da Docker Container read-only sind. Darauf hin haben wir uns für XAMPP entschieden.**

**XAMPP und XSTL**

**Der XAMPP hat keine XSL Transformationen ausgeführt. Es musste im php.ini den XSLT-Support manuell eingeschaltet werden.**

# <Einsatz von nicht XML-Technologien />

**PHP**

* Pro Seite existiert ein PHP Script. Beim Aufruf der PHP Scripts wird das XSLT Sheet im XML Dokument ersetzt, damit das korrekte XHTML Dokument geladen und angezeigt wird. Wir haben keine andere Variante gefunden, um in einem XML Dokument mehrere XSLT anwenden zu können.

**CSS**

* Da Design und Responsive Design nicht im Fokus dieser Arbeit sind, jedoch die Bedienbarkeit und Darstellung verbessern, haben wir uns für das Bootstrap Framework entschieden. Dabei wird das CSS von Bootstrap genutzt.

**JAVASCRIPT**

* Da Design und Responsive Design nicht im Fokus dieser Arbeit sind, jedoch die Bedienbarkeit und Darstellung verbessern, haben wir uns für das Bootstrap Framework entschieden. Dabei wird JavaScript von Bootstrap genutzt.
* Client-seitige Validierung wer Forms wird mit JavaScript durchgeführt. Dies dient er Benutzerfreundlichkeit und ermöglicht den User auf falsche Inputs hinzuweisen. Server-seitig wird jedoch mit dem Input gegen das XML Schema validiert.

# <Fazit />