**BWL2 Praktikum Nr. 1 SS 2015**

**Teammitglieder:**

Saeed Shanidar

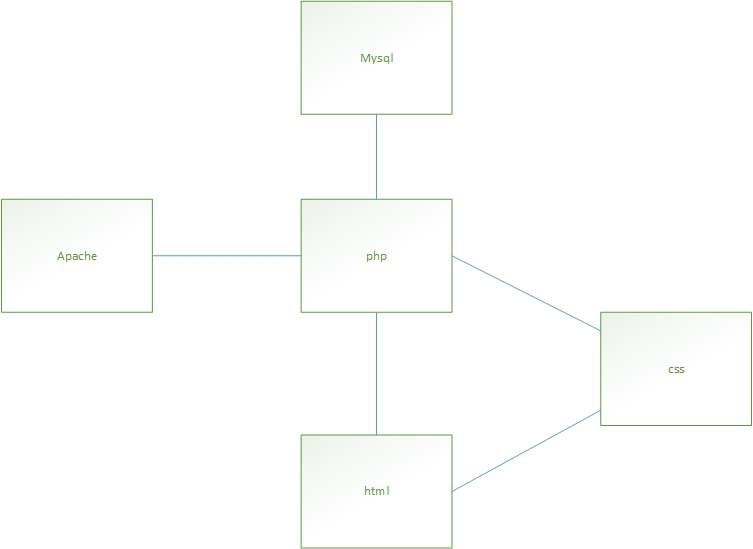
Dimitri Meier

Hossein Asgari

**Abgabe (zu Beginn des nächsten Praktikums):**

1. Architekturübersicht der Anwendung (welche Software-Komponenten werden benötigt?)

* html, php, css
* Mysql
* Apache
* Browser



-Browser

* Der Großteil der Benutzeroberfläche wird in der Regel zur  
   Anzeige von Inhalten genutzt.
* Diese können durch Eingabe in der Adressleiste erreicht werden.
* Daneben verfügen Browser über Schaltflächen, mit denen der Benutzer zu zuvor   
   besuchte Seite sowie zur Startseite navigieren kann.
* Die URL der Website kann als Lesezeichen gespeichert werden, um weitere   
   Besuche der Webseite zu vereinfachen.
* Unterstützt HTML und CSS

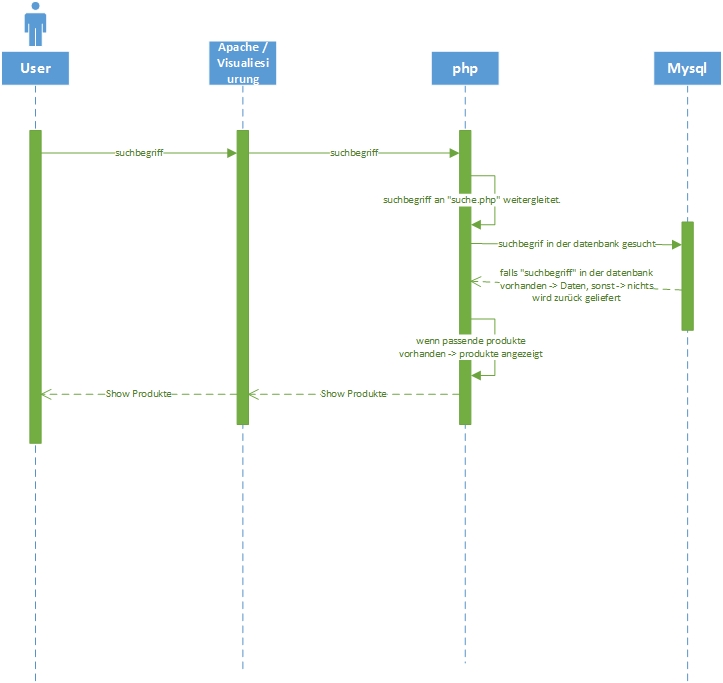
-PHP Server

* Der Webserver lädt Daten von Datenbank
* übergibt das PHP-Script an den PHP-Interpreter.
* Dieser führt das Script aus.
* Der verarbeitete Inhalt wird ausgegeben.

-MySQL Datenbank

* Datenbanken dienen der strukturierten Ablage von Daten und erlauben damit einen   
   schnellen Zugriff auf ihre Inhalte.
* Dies macht sie besonders geeignet, um im Zusammenspiel mit eigenen Skripten für   
   Dynamik auf der Webseite zu sorgen.
* Bei der Webseite ändern sich Inhalte einer Seite häufig – entweder automatisch   
   oder bei Eingabe bestimmter Parameter.
* Die dabei generierten Daten können entweder in Datenbank abgelegt werden oder   
   aber effizienter, d. h. geordnet und strukturiert, in einer Datenbank.

1. UML-Sequenzdiagramm (mit den Komponenten von 1. für eine Artikelsuche)



1. Begründung für die gewählten Technologien, insbesondere die Programmiersprache.

Erfahrung mit den Programmiersprachen

SQL:

* Daten Persistent Speichern

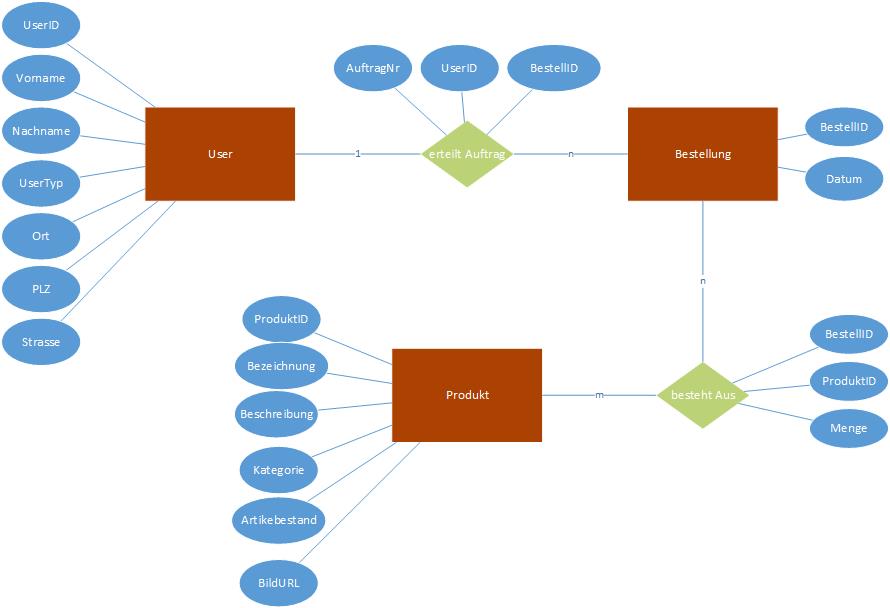
PHP:

* um Logik zu programmieren

HTML:

* Als Oberfläche. Diese wird von allen gängigen Browser unterstütz

1. ERM der Datenbank, SQL-Tabellendefinition und Glossar der Attribute



**Glossar**

**USER**

|  |  |
| --- | --- |
| ***Fachbegriff*** | ***Erklärung*** |
| \*ID | Ist eine fortlaufende Nummer die eindeutig für die jeweils zugehörige Tabelle geführt wird. |
| Vorname Nachname Ort | Vorname, Nachname und Ort des Kunden sind in reiner Textform gespeichert |
| UserTyp | Es gibt drei verschiedene Benutzertypen:   1. Admin 2. registrierter Kunde 3. nicht registrierter Kunde (Gast) |
| PLZ | Postleitzahl als Nummer |
| Strasse | Die Strasse besteht aus einer textuelle Beschreibung und einer eindeutigen Nummer |

**AUFTRAG**

|  |  |
| --- | --- |
| ***Fachbegriff*** | ***Erklärung*** |
| Auftragsnummer | Diese Nummer ist eine Zusammensetzung von \*ID’s aus den Tabellen USER und BESTELLUNG. |

**BESTELLUNG**

|  |  |
| --- | --- |
| ***Fachbegriff*** | ***Erklärung*** |
| Datum | Das Attribut Datum wird in reiner Textform gepflegt |

**PRODUKT**

|  |  |
| --- | --- |
| ***Fachbegriff*** | ***Erklärung*** |
| Bezeichnung | Ist eine Textuelle Bezeichnung des Produktes (Titel des Produktes) |
| Beschreibung | Ist eine Textuelle Beschreibung des Produktes  (Werbetext des Produktes) |
| Kategorie | Ist eine Textuelle Bezeichnung was zur Unterteilung des Produktes dient |
| Artikelbestand | Eine nummerische Bezeichnung die die Verfügbarkeit des Produktes darstellt |
| BildURL | Ist ein Link, der den Speicherort des Bildes beschreibt. |

**BESTEHTAUS**

|  |  |
| --- | --- |
| ***Fachbegriff*** | ***Erklärung*** |
| Menge | Dient als Information über die Menge des gekauften Artikels |

**SQL-Datenbank**

--

-- Datenbank: `online\_shop`

--

CREATE DATABASE IF NOT EXISTS `online\_shop` DEFAULT CHARACTER SET utf8 COLLATE utf8\_bin;

USE `online\_shop`;

-- --------------------------------------------------------

--

-- Tabellenstruktur für Tabelle `auftrag`

--

DROP TABLE IF EXISTS `auftrag`;

CREATE TABLE IF NOT EXISTS `auftrag` (

`AuftragNr` int(4) NOT NULL,

`UserID` int(4) NOT NULL,

`BestellID` int(4) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- RELATIONEN DER TABELLE `auftrag`:

-- `UserID`

-- `user` -> `UserID`

--

-- --------------------------------------------------------

--

-- Tabellenstruktur für Tabelle `besteht\_aus`

--

DROP TABLE IF EXISTS `besteht\_aus`;

CREATE TABLE IF NOT EXISTS `besteht\_aus` (

`BestellID` int(4) NOT NULL,

`ProduktID` int(4) NOT NULL,

`Menge` varchar(3) COLLATE utf8\_bin NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- RELATIONEN DER TABELLE `besteht\_aus`:

-- `BestellID`

-- `bestellung` -> `BestellID`

-- `ProduktID`

-- `produkt` -> `ProduktID`

--

-- --------------------------------------------------------

--

-- Tabellenstruktur für Tabelle `bestellung`

--

DROP TABLE IF EXISTS `bestellung`;

CREATE TABLE IF NOT EXISTS `bestellung` (

`BestellID` int(4) NOT NULL,

`Datum` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

-- --------------------------------------------------------

--

-- Tabellenstruktur für Tabelle `produkt`

--

DROP TABLE IF EXISTS `produkt`;

CREATE TABLE IF NOT EXISTS `produkt` (

`ProduktID` int(4) NOT NULL,

`Bezeichnung` text COLLATE utf8\_bin NOT NULL,

`ProduktInfo` text COLLATE utf8\_bin NOT NULL,

`Kategorie` text COLLATE utf8\_bin NOT NULL,

`Artikelbestand` varchar(3) COLLATE utf8\_bin NOT NULL,

`Dateipfad` text COLLATE utf8\_bin NOT NULL,

`Preis` varchar(6) COLLATE utf8\_bin NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=2001 DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- Daten für Tabelle `produkt`

--

INSERT INTO `produkt` (`ProduktID`, `Bezeichnung`, `ProduktInfo`, `Kategorie`, `Artikelbestand`, `Dateipfad`, `Preis`) VALUES

(1000, 'Super geniales Bett', 'Variante 90/200 cm\r\nMaße ca.: 100/148/188\r\nMassive Nordliche Kiefer', 'betten', '5', '../pics/betten/Futonbett\_1tlg.png', '300'),

(1001, 'Super weiches Bett', 'Variante 180/200 cm\r\nMaße ca.: 200/148/220\r\nMassive Nordliche Kiefer', 'betten', '6', '../pics/betten/Futonbett\_2tlg.png', '500'),

(2000, '»Frottee KS, Deluxe & Premium Cool Plus«', 'Ca. 25 cm hoch mit 7 Liegezonen \r\nIn den Härtegraden 2 -5', 'Matratze', '10', '../pics/matratzen/canvas.png', '200');

-- --------------------------------------------------------

--

-- Tabellenstruktur für Tabelle `user`

--

DROP TABLE IF EXISTS `user`;

CREATE TABLE IF NOT EXISTS `user` (

`UserID` int(4) NOT NULL,

`Vorname` text COLLATE utf8\_bin NOT NULL,

`Nachname` text COLLATE utf8\_bin NOT NULL,

`UserTyp` text COLLATE utf8\_bin NOT NULL,

`Ort` varchar(20) COLLATE utf8\_bin NOT NULL,

`PLZ` varchar(5) COLLATE utf8\_bin NOT NULL,

`Strasse` text COLLATE utf8\_bin NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

--

-- Indizes der exportierten Tabellen

--

--

-- Indizes für die Tabelle `auftrag`

--

ALTER TABLE `auftrag`

ADD PRIMARY KEY (`AuftragNr`), ADD UNIQUE KEY `User\_ID` (`UserID`) COMMENT 'user', ADD UNIQUE KEY `Bestell\_ID` (`BestellID`) COMMENT 'Bestell';

--

-- Indizes für die Tabelle `besteht\_aus`

--

ALTER TABLE `besteht\_aus`

ADD PRIMARY KEY (`BestellID`,`ProduktID`), ADD KEY `ProduktID` (`ProduktID`);

--

-- Indizes für die Tabelle `bestellung`

--

ALTER TABLE `bestellung`

ADD PRIMARY KEY (`BestellID`);

--

-- Indizes für die Tabelle `produkt`

--

ALTER TABLE `produkt`

ADD PRIMARY KEY (`ProduktID`);

--

-- Indizes für die Tabelle `user`

--

ALTER TABLE `user`

ADD PRIMARY KEY (`UserID`);

--

-- AUTO\_INCREMENT für exportierte Tabellen

--

--

-- AUTO\_INCREMENT für Tabelle `auftrag`

--

ALTER TABLE `auftrag`

MODIFY `AuftragNr` int(4) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT für Tabelle `bestellung`

--

ALTER TABLE `bestellung`

MODIFY `BestellID` int(4) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT für Tabelle `produkt`

--

ALTER TABLE `produkt`

MODIFY `ProduktID` int(4) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=2001;

--

-- AUTO\_INCREMENT für Tabelle `user`

--

ALTER TABLE `user`

MODIFY `UserID` int(4) NOT NULL AUTO\_INCREMENT;

--

-- Constraints der exportierten Tabellen

--

--

-- Constraints der Tabelle `auftrag`

--

ALTER TABLE `auftrag`

ADD CONSTRAINT `UserID` FOREIGN KEY (`UserID`) REFERENCES `user` (`UserID`);

--

-- Constraints der Tabelle `besteht\_aus`

--

ALTER TABLE `besteht\_aus`

ADD CONSTRAINT `BestellID` FOREIGN KEY (`BestellID`) REFERENCES `bestellung` (`BestellID`),

ADD CONSTRAINT `ProduktID` FOREIGN KEY (`ProduktID`) REFERENCES `produkt` (`ProduktID`);