**Projektbeschreibung**

# Stundenplan-App

für Smartphones, Tablets und Desktop-PCs

**Projekt von**

Alexander Zilch, Dominik Jahn, Felix Hollmann und Sebastian Weiß




# Inhaltsverzeichnis

# Projektbeschreibung

Das Ziel der Projektarbeit ist die Erstellung einer HTML-basierten Applikation zur Darstellung des Stunden- und Vertretungsplans für die Schüler und Lehrer der EDV-Schulen Wiesau. Die Benutzer können nach der erfolgten Anmeldung (die Anmeldedaten entsprechen denen im Schulnetzwerk) den Stundenplan für jede Kalenderwoche sowie detailiert für einzelne Tage einsehen.

Die Erstellung der Applikation erfolgt in zwei Schritten: die Erstellung einer API, welche die Daten aus einer MySQL-Datenbank ausließt und als XML- oder JSON-Daten bereitstellt, sowie eine Benutzeroberfläche, die diese Daten ließt und in einer für den Benutzer leicht lesbaren Form darstellt. Die Implementierung der Datenschnittstelle ist nicht Teil dieses Projekts und wird von einem anderen Entwicklungsteam durchgeführt. Primäres Ziel ist die Programmierung einer intuitiven Benutzeroberfläche.

Diese Oberfläche soll sowohl auf Smartphones und Tablets sowie im Browser eines Desktop-PCs funktionieren. Smartphones und Tablets mit einer Bildschirmdiagonalen unter 10” sollen dabei eine speziell für Mobilgeräte optimierte Seite anzeigen, während große Tablets und PCs eine alternative Ansicht erhalten.

# Umsetzung

Im folgenden wird beschrieben, wie das Projekt realisiert werden soll.

## Technologien

### HTML 5

Da es sich um eine Webapplikation handelt, die unter verschiedenen Betriebssystemen und unterschiedlichen Gerätetypen funktionieren soll, muss für die Oberflächenbeschreibung HTML eingesetzt werden. HTML 5 enthält dabei viele neue Änderungen, die besonders für Webanwendungen interessant sind, ganz besonders die Möglichkeit, den Browser-Cache zu kontrollieren und damit die Anwendung auch dann Verfügbar zu machen, wenn das eingesetzte Gerät gerade nicht mit dem Internet verbunden ist.

### jQuery und jQuery Mobile

Um Ladezeiten zu verkürzen und eine flüssige Benutzerführung zu ermöglichen muss JavaScript eingesetzt werden. Bedingt durch den Einsatz in verschiedenen Browsern (Chrome und Firefox Mobile auf Android, Safari auf iOS, etliche verschiedene bei Desktop-PCs) muss dabei eine JavaScript-Bibliothek wie jQuery eingesetzt werden, welche viele Funktionalitäten, die sonst browserspezifisch sind, vereinheitlichen und somit viel Entwicklungs- und Testaufwand sparen.

Für eine einheitliche Oberfläche auf Smartphones und kleinformatigen Tablets wird jQuery Mobile eingesetzt. Dadurch erhält der Benutzer eine für berührungsempfindliche Bildschirme optimierte Oberfläche, welche durch Bewegungsgesten die Navigation vereinfacht.

### AngularJS

AngularJS ist eine weitere JavaScript-Bibliothek, welche in diesem Fall für die Benutzeroberfläche auf großen Tablets und Desktop-PCs eingesetzt wird.

### PHP oder Java Server Pages

API-Anfragen werden aus Gründen der Sicherheit nicht direkt per AJAX an den Schnittstellenserver übertragen sondern intern getunnelt – dies erfolgt entweder über ein PHP- oder JSP-Programm. Des weiteren wird ein Zwischenspeicher eingerichtet, welcher den Schnittstellenserver entlasten soll und eingreift, wenn der Browser neue Daten anfordert und der Server nicht erreichbar ist.

### SQLite

Der lokale Zwischenspeicher benutzt SQLite zur logischen Speicherung der Daten. Es wird dabei bewusst auf MySQL oder andere Datenbankserver verzichtet, da ein Cache nur zur temporären Speicherung gedacht ist und auch nur aktuelle Daten enthalten soll.

# Benutzeroberfläche

Beim Aufruf der Anwendung wird anhand der Browsersignatur bestimmt, ob es sich um ein mobiles Endgerät handelt oder nicht. Je nach Ergebnis wird dabei eine mit jQuery Mobile unterstützte Variante der Benutzeroberfläche oder eine Desktopversion angezeigt.

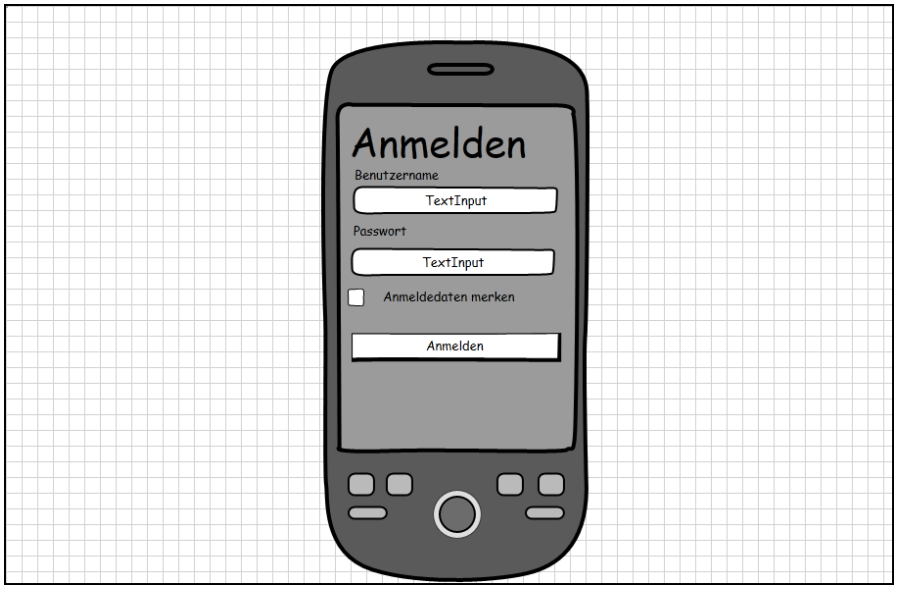
Die Benutzeroberfläche besteht aus den folgenden Komponenten:

## Anmeldebildschirm

Aus Datenschutz- und Sicherheitsgründen kann die Anwendung nur für authentifizierte Benutzer freigegeben werden. Dazu ist eine einmalige Anmeldung notwendig. Die Anmeldedaten können dabei für einen gewissen Zeitraum (zum Beispiel drei Monate) auf dem Gerät gespeichert werden.

Auf der Desktopoberfläche wird dabei ein vertikal und horizontal zentriertes Formular angezeigt, auf mobilen Geräten wird diese den gesamten Bildschirm in Anspruch nehmen.

Das Formular hat ein Eingabefeld für den Benutzernamen und das Passwort sowie ein Ankreuzfeld zur Speicherung der Anmeldedaten. Eine “Passwort vergessen”-Funktion ist allein schon aus technischen Gründen nicht vorgesehen.



## Startbildschirm / Tagesansicht

Direkt nach der Anmeldung erscheint der Stundenplan für den aktuellen Tag. Dabei wird unterschieden, ob es sich bei dem aktuellen Benutzer um einen Schüler oder Lehrer handelt.

### Schüler

Im oberen Bereich der Seite befindet sich eine Auflistung aller Fächer, welche der angemeldete Schüler hat. Für Blockstunden wird dabei das Feld entsprechend größer dargestellt als für Einzelstunden. Wie aus der bereits bekannten Digikabu-Desktopanwendung wird auch hier angezeigt, welcher Lehrer das Fach unterrichtet und in welchem Raum dies stattfindet.

Am unteren Bildschirmrand befindet sich eine Auswahlliste, welche auch die anderen Klassen beinhaltet, um auch den Stundenplan anderer Klassen anzusehen.

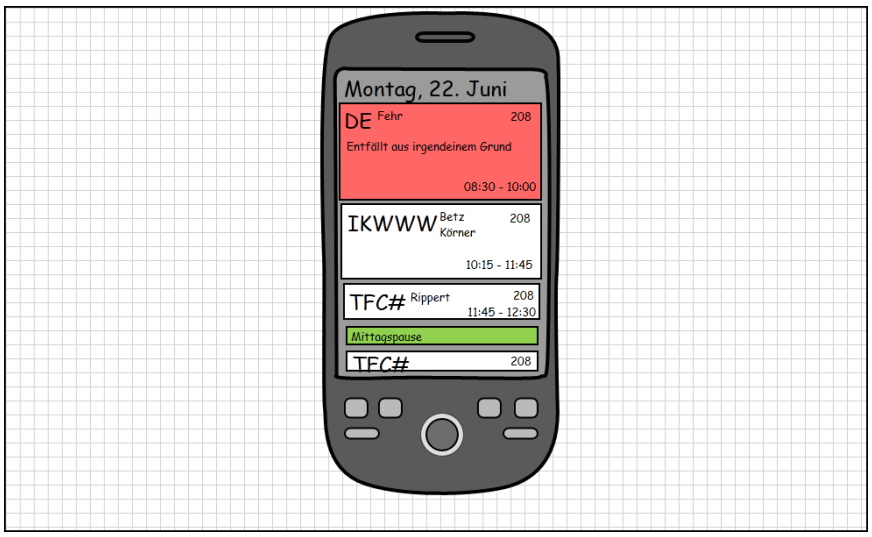
### Lehrer

Im oberen Bereich befindet sich auch hier wieder eine Auflistung der Fächer, welcher der angemeldete Lehrer heute unterrichtet. Der einzige Unterschied zur Schüleransicht ist der, dass statt des eigenen Namens die betroffene Klasse angezeigt wird sowie ein anderer Lehrer, der ebenfalls am Unterricht beteiligt ist, sofern vorhanden.

Statt einer Klassenauswahlliste können die Lehrer sich den Stundenplan anderer Lehrer ansehen, erhalten aber ebenfalls Einsicht in die Stundenpläne aller Klassen.

### Mobile Geräte

Auf mobilen Geräten kann per Schiebebewegung zwischen den einzelnen Tagen hin- und hergeblättert werden.



## Wochenansicht

Die Wochenansicht zeigt den gesamten Stundenplan der Woche an. Auch hier ist wieder die eigene Klasse vorausgewählt, es kann jedoch jederzeit zwischen den Klassen gewechselt werden. Es bestehen wieder die gleichen Unterschiede zwischen Lehrern und Schülern wie in der Tagesansicht.

### Mobile Geräte

Da auf Smartphones meist kein Platz für eine für Stundenpläne übliche tabellarische Darstellung ist, kann dieser hier nicht angezeigt werden. Erst ab einer bestimmten (bisher unbekannten) mindesten Bildschirmbreite wird die Wochenansicht angezeigt, etwa bei 7”-Tablets (zum Beispiel iPad Mini, Nexus 7 oder Galaxy Tab 7).

# Verbindung zum Schnittstellenserver

Da die Entwicklung des Schnittstellenservers durch ein anderes Programmierteam erfolgt kann hier derzeit noch nicht auf technische Details eingegangen werden. Deswegen finden sich hier nur oberflächliche Beschreibungen und eventuelle Anforderungen an die API-Struktur.

## API-Struktur

Wünschenswert wäre eine RESTful-API, da sich diese meist aus semantischen Gründen am leichtesten Implementieren lassen.

Folgende Anfragen wären dabei denkbar:

```
GET /stundenplan
```

Ruft den aktuellen Stundenplan für die eigene Klasse ab

```
GET /stundenplan/IT11a
```

Ruft den aktuellen Stundenplan für die angegebene Klasse ab

```
GET /stundenplan/IT11a/2014-06-06
```

Ruft den Stundenplan für das angegebene Datum und Klasse ab

```
GET /status last_synchronization=2014-06-06 17:05:48
```

Fragt ab, ob seit der letzten Synchronisierung angegebenen Datum Änderungen angefallen sind

### Authentifizierung

Die Authentifizierung sollte über die HTTP-üblichen und dafür vorgesehenen Header erfolgen.

## Verbindungsaufbau

Die Benutzeroberfläche darf direkt keine Anfragen an den Schnittstellenserver stellen. Stattdessen soll ein AJAX-Request an den lokalen Server gerichtet werden, welcher wiederum die Anfrage an den API-Server weiterleitet.

Dadurch wird verhindert, dass vom Benutzer aus Daten wie die Anmelde daten übertragen werden.

Das PHP- oder JSP-Programm wertet dabei die eingehenden Daten aus, speichert diese aufbereitet in der SQLite-Datenbank ab und gibt sie 1:1 an die Benutzeroberfläche zurück.

Ein großer Vorteil der Zwischenspeicherung auf dem lokalen Server entsteht durch die Aufbereitung der Schnittstellenantwort: wenn ein Schüler bereits die Daten heruntergeladen hat, so können diese Daten auch für alle weiteren Benutzer verwendet werden und eine weitere Anfrage ist nicht mehr notwendig. Dadurch entsteht zwar eine sehr redundante Datenhaltung (auf dem Schnittstellenserver, auf dem lokalen Server sowie auf dem Endgerät), allerdings wird hierdurch eine hohe Ausfallsicherheit erzeugt und die Netzwerkbelastung minimiert.

## Cache-Optimierung auf Endgeräten

Um den Schnittstellenserver zu entlasten sowie die Benutzung der Anwendung auch ohne Internetverbindung zu ermöglichen wird die in HTML5 eingeführte Technologie “ApplicationCache” eingesetzt. Dadurch kann der Zwischenspeicher des Browser kontrolliert werden, um eine nahezu dauerhafte Speicherung der Daten zu ermöglichen.

Dabei sollte eine XML-Datei gespeichert werden, welche alle erforderlichen Daten enthält - beispielsweise den aktuellen Wochenstundenplan für die eigene Klasse – wodurch es dann möglich ist, in der Anwendung zu navigieren als wäre man noch mit dem Internet verbunden. Lediglich das Abrufen anderer Daten würde eine Verbindung voraussetzen, weshalb auch angezeigt werden sollte, dass man sich im “Offline-Modus” befindet.