Projektname

Fachbereich: Informatik, Applikationsentwicklung

Projektleitung: Jonas Metzener

Co-Leitung: Fabian Jäiser

Co-Leitung: Jonas Metzener

Betreuer: Jakob Wanner  
Datum: 04.05.2015

Firma: JoFa AG

# Abstract (Kurzbeschreibung)

Unsere Website soll eine Online Darstellung von Stundenplänen sein. Die Stundepläne aller Klassen sind öffentlich ersichtlich. Die Lehrer können sich einloggen und Stunden erfassen. Der Superuser kann Benutzer und Klassen erfassen, bearbeiten und löschen.

Inhaltsverzeichnis

[1 Abstract (Kurzbeschreibung) 1](#_Toc419701768)

[2 Lastenheft (Aufgabenstellung) 6](#_Toc419701769)

[2.1 Einführung 6](#_Toc419701770)

[2.2 Funktionale Anforderungen 6](#_Toc419701771)

[2.2.1 Umfang 6](#_Toc419701772)

[2.2.2 GUI 6](#_Toc419701773)

[2.2.3 Mitgliederbereich 6](#_Toc419701774)

[2.2.4 Formulare 6](#_Toc419701775)

[2.2.5 MySQL-Datenbank 6](#_Toc419701776)

[2.2.6 CSS 6](#_Toc419701777)

[2.3 Nichtfunktionale Anforderungen 6](#_Toc419701778)

[2.3.1 Technik 7](#_Toc419701779)

[2.3.2 Rechtliches 7](#_Toc419701780)

[2.3.3 Benutzbarkeit 7](#_Toc419701781)

[2.3.4 Effizienz 7](#_Toc419701782)

[2.3.5 Wartbarkeit 8](#_Toc419701783)

[2.3.6 Verantwortlichkeiten 8](#_Toc419701784)

[2.4 Lieferumfang 8](#_Toc419701785)

[2.4.1 Webseite 8](#_Toc419701786)

[2.4.2 Dokumentation 8](#_Toc419701787)

[2.4.3 Präsentation 8](#_Toc419701788)

[2.5 Abnahmekriterien 9](#_Toc419701789)

[3 Projektorganisation 10](#_Toc419701790)

[3.1 Organigramm 10](#_Toc419701791)

[3.2 Zuständigkeiten der Teilprojektleiter 10](#_Toc419701792)

[3.2.1 Backend-Entwickler 10](#_Toc419701793)

[3.2.2 Teilprojektleiter B 10](#_Toc419701794)

[4 Pflichtenheft 11](#_Toc419701795)

[4.1 Anforderungsanalyse 11](#_Toc419701796)

[4.1.1 Zielgruppe 11](#_Toc419701797)

[4.1.2 Ziele 11](#_Toc419701798)

[ Die Benutzer haben die Möglichkeit neue Stunden zu erstellen, bearbeiten und zu löschen. 11](#_Toc419701799)

[ Der Superuser muss neue Benutzer und neue Klassen erstellen, bearbeiten und löschen können. 11](#_Toc419701800)

[ Die Lehrer haben die Möglichkeit ihr vorgegebenes Passwort zu ändern. 11](#_Toc419701801)

[ Die Stundenpläne sind öffentlich in einer Übersicht ersichtlich. 11](#_Toc419701802)

[ Eine funktionale Login-Seite ist vorhanden. 11](#_Toc419701803)

[ Auf der Startseite werden die Klassen aufgelistet, welche dann angeklickt werden können um den Stundenplan der Klasse anzusehen. 11](#_Toc419701804)

[4.2 Betriebsumgebung 11](#_Toc419701805)

[4.3 Lösungsvarianten 12](#_Toc419701806)

[4.3.1 Variante A 12](#_Toc419701807)

[4.3.2 Variante B 12](#_Toc419701808)

[4.3.3 Variante C 13](#_Toc419701809)

[4.4 Lösungsbewertung 13](#_Toc419701810)

[4.5 Risikoanalyse 13](#_Toc419701811)

[4.5.1 Szenario/Situation 1 – Die Entwicklungsumgebung funktioniert nicht 14](#_Toc419701812)

[4.5.2 Szenario/Situation 2 – Berechtigungssystem funktioniert nicht 14](#_Toc419701813)

[4.5.3 Szenario/Situation 3 – Erstellung der Stunden macht Schwierigkeiten 15](#_Toc419701814)

[4.5.4 Szenario/Situation 4 – Datenbankverbindung funktioniert nicht 15](#_Toc419701815)

[5 Planung 17](#_Toc419701816)

[5.1 Projektstrukturplan (PSP) 18](#_Toc419701817)

[5.2 Arbeitspaket A – Design 19](#_Toc419701818)

[5.3 Arbeitspaket B – Datenbank 20](#_Toc419701819)

[5.4 Arbeitspaket C – Models 21](#_Toc419701820)

[5.5 Arbeitspaket D – Mappers & Views 22](#_Toc419701821)

[5.6 Arbeitspaket E – Testing 23](#_Toc419701822)

[5.7 Zeitplanung 24](#_Toc419701823)

[5.7.1 GANTT 24](#_Toc419701824)

[6 Technische Dokumentation (Produktdokumentation) 25](#_Toc419701825)

[6.1 Tests 25](#_Toc419701826)

[7 Benutzerhandbuch 26](#_Toc419701827)

[7.1 Anmelden, Login, Logout oder z.B. Inbetriebnahme bei einem Gerät 26](#_Toc419701828)

[7.2 Funktionen 27](#_Toc419701829)

[8 Weiterentwicklungsmöglichkeiten 28](#_Toc419701830)

[9 Fazit 28](#_Toc419701831)

[10 Abkürzungen / Begriffserklärung 28](#_Toc419701832)

[A Anhang 29](#_Toc419701833)

[A.1 Schema 29](#_Toc419701834)

[A.2 Fertigungsdokumente 29](#_Toc419701835)

[A.3 Beschreibung der Ordnerstruktur 29](#_Toc419701836)

[A.4 Testprotokolle 29](#_Toc419701837)

[A.5 JavaDoc 29](#_Toc419701838)

[A.6 Mindmaps 29](#_Toc419701839)

[A.7 … 29](#_Toc419701840)

[A.8 Ressourcen 30](#_Toc419701841)

[A.8.1 Bilder 30](#_Toc419701842)

[A.9 IPERKA 31](#_Toc419701843)

[A.10 Abnahmeformular Pflichtenheft 32](#_Toc419701844)

[A.10.1 Projektmitglieder 32](#_Toc419701845)

[A.11 Usw. 33](#_Toc419701846)

[11 Beurteilungskriterien (gemäss MP-307-414) 34](#_Toc419701847)

Abbildungsverzeichnis

[Abbildung 1: Organigramm der Projektgruppe 6](#_Toc368916081)

[Abbildung 2: Bewertung der Risiken 10](#_Toc368916082)

[Abbildung 3: IPERKA 23](#_Toc368916083)

Tabellenverzeichnis

[Tabelle 1: Test Case 17](#_Toc368916084)

[Tabelle 2: Testresultate 18](#_Toc368916085)

# Lastenheft (Aufgabenstellung)

## Einführung

Für das Modul Webentwicklung ist eine Webseite zu erstellen. Dabei darf das Thema frei gewählt werden. Während der gesamten Dauer des Moduls wird im Rahmen des Projektmanagements diese Dokumentation vervollständigt. Die Bewertungskriterien können am Ende des Anhangs nachgelesen werden. Wichtig ist, dass am Ende des Moduls eine voll funktionsfähige Webseite präsentiert werden kann welche folgende Anforderungen erfüllt:

## Funktionale Anforderungen

### Umfang

Die Webseite muss aus mindestens 5 Seiten bestehen.

### GUI

Für den Benutzer muss eine einfache Benutzerführung erstellt werden. Dazu müssen alle Webseiten des Projekts sinnvoll gegliedert werden, damit eine klar ersichtliche und benutzerfreundliche Navigation erstellt werden kann. Im Minimum müssen 2 Hauptkategorien erstellt werden, wobei ein Bereich z.B. Themen wie Hilfe, Support und Service behandeln könnte. Damit sich der Benutzer schnell auf der Webseite zurechtfindet, müssen sinnvolle Namen für die Haupt- und Unterkategorien gefunden werden.

### Mitgliederbereich

Die Webseite muss einen Mitgliederbereich zur Verfügung stellen. Dabei ist es wichtig, dass gewisse Informationen nicht öffentlich sind, sondern erst nach einer Registrierung mit mindestens dem Benutzername und E-Mail sichtbar werden. Im Minimum muss zwischen Gast (öffentlich) und angemeldetem Benutzer unterschieden werden.

### Formulare

Für alle Formulare muss eine Plausibilitätsprüfung durchgeführt werden. Diese muss mit Javascript und PHP durchgeführt werden.

### MySQL-Datenbank

Die Webseite arbeitet mit einer MySQL-Datenbank im Hintergrund. Dabei sind sinnvolle Anwendungsmöglichkeiten zu finden wo z.B. Benutzerdaten und Inhalte der Webseite in der Datenbank abgelegt werden.

### CSS

Die Webseite sollte möglichst vollständig mit CSS formatiert werden. Dabei müssen alle Elemente in externen CSS Dateien ausgelagert werden.

## Nichtfunktionale Anforderungen

### Technik

Die Webseite muss folgende Anforderungen erfüllen:

* Min. XTHML 1.0 Transitional (muss valid sein)
* CSS 2.0 (CSS 3.0 nur kompatible Attribute)
* Kompatibilität mit Firefox 11.0, Chrome 18.0.1025 und Internet Explorer 9.0

### Rechtliches

Wer Internetseiten erstellt, muss die rechtlichen Rahmenbedingungen kennen. Das wichtigste Recht dabei ist das Urheberrecht. Es schütz das geistige Eigentum eines Urhebers. Wer seine Webseite veröffentlicht (z.B. im Internet) muss daher seine Rechten und Pflichten kennen. Zum Beispiel ist es nicht erlaubt Bilder die auf Google gefunden wurden auf der eigenen Webseite zu verwenden.

Darum bietet zum Beispiel Google die Möglichkeit unter „Erweiterte Suche“ > „Nutzungsrechte“ > „kostenlos zu nutzen oder weiterzugeben – auch für kommerzielle Zwecke“ Bilder zu filtern.

Im Projekt dürfen nur Inhalte verwendet werden, die kostenlos weitergegeben werden können. Im Internet gibt es dazu verschiedene Fotogalerien in denen Fotographen ihre Bilder unter gewissen Umständen freigeben:

<http://www.flickr.com/creativecommons/>

Meistens wird eine Namensnennung vorgeschrieben. Darum muss jedes Webprojekt eine Impressumseite enthalten auf der die Urheber genannt werden. Auf dieser Seite werden vor allem Urheber der Bilder genannt, wo der Urheber nicht unter das Bild geschrieben werden kann. Das ist zum Beispiel bei CSS Designs der Fall.

Weitere Ressourcen (Seiten mit Bildern) sind im Anhang zu finden.

Die Einhaltung der Litzenrechte liegt bei der jeweiligen Gruppe.

### Benutzbarkeit

Die fertige Webseite muss einfach zu bedienen sein und hat ein ansprechendes Design. Durch einen Abschlusstest müssen alle Seiten geprüft werden um sicherzustellen, dass keine Fehler während der Verwendung auftreten.

### Effizienz

Während der gesamten Projektdauer ist auf Effizienz zu achten (der Aufwand muss in gutem Verhältnis zum Nutzen sein). Dabei muss der Projektleiter darauf achten, dass alle die am Projekt beteiligt sind, optimal arbeiten können. Zudem soll darauf geachtet werden, dass kein Code (PHP, JavaScript, CSS oder HTML) doppelt erstellt wird.

### Wartbarkeit

Damit die fertige Webseite auch später weiterentwickelt werden kann, sollte der Code möglichst Wartungsfreundlich gestaltet werden. Dazu muss der Code mit Kommentar beschrieben werden. Folgende Code-Elemente sind zu beschreiben:

PHP: Oben in der Datei (z.B. jede Datei/Klasse) und jeweils jede Funktion

JavaScript: Wie bei PHP oder mindestens jedes Skript kommentieren das verwendet wird.

CSS: Kurze Beschreibung der Formatierung und Name des Autors oder Quelle.

Bei jeder Funktion muss der Autor erwähnt werden, um bei einem späteren Zeitpunkt nachfragen zu können.

### Verantwortlichkeiten

Während des Projekts werden die zu erledigenden Aufgaben auf die Teammitglieder (Projektleiter und Teilprojektleiter) aufgeteilt. Dabei erhält jedes Teammitglied einen Bereich (z.B. HTML, CSS, DB, usw.) für das er oder sie zuständig ist. Der Teilprojektleiter ist dabei zuständig, dass jedes Mitglied für seinen Teilbereich auch Code schreibt. Zum Beispiel muss der Teilprojektleiter für CSS schauen, dass jedes Teammitglied mindestens einen HTML-Tag formatiert hat. Da jede z.B. CSS Definition kommentiert wird, ist ersichtlich, wer welchen Code geschrieben hat. Am Ende muss jedes Teammitglied in jedem Bereich aktiv Code geschrieben haben.

## Lieferumfang

Am Ende des Projekts müssen folgende Produkte abgegeben werden:

### Webseite

Es muss eine voll funktionsfähige Webseite abgegeben werden, die aus mindestens 5 Seiten besteht. Alle Formulare, Skripts funktionieren und die Inhalte (Texte, Bilder, usw.) werden korrekt dargestellt.

### Dokumentation

Die Dokumentation (dieses Dokument) ist vollständig ergänzt worden. Alle geforderten Inhalte sind genügend genau beschrieben so, dass eine andere Gruppe das Projekt später weiterführen könnte.

### Präsentation

Abschluss des Projekts bildet eine Produktpräsentation von max. 15 Minuten. Diese beinhaltet folgende Themen:

ca.5 Minuten: Vorstellen des Projekts (Idee) und des Projektteams

ca.5 Minuten: Vorstellung der Webseite mit allen Funktionen

ca.5 Minuten: Besprechung des Projekts

## Abnahmekriterien

Eine Abnahme kann nur erfolgen, wenn alle oben genannten Anforderungen erfüllt sind und das Pflichtenheft durch den Coach (Kurt oder Mike) abgenommen wurde. Das Formular im Anhang A.10 Abnahmeformular Pflichtenheft muss dazu vollständig ausgefüllt und vom Coach unterschrieben sein.

# Projektorganisation

In diesem Kapitel werden die Zuständigkeiten im Projekt definiert.

## Organigramm



Abbildung 1: Organigramm der Projektgruppe

## Zuständigkeiten der Teilprojektleiter

### Backend-Entwickler

Der Backend-Entwickler ist für die Datenbank verantwortlich. Die Struktur und Verbindung wird von ihm betreut.

### Teilprojektleiter B

Teilprojektprojekt B ist für die Verteilung der Ressourcen verantwortlich.

# Pflichtenheft

## Anforderungsanalyse

### Zielgruppe

Das Produkt soll es den Schülern erleichtern, ihren Stundenplan einzusehen.

### Ziele

Folgende Punkte müssen erfüllt sein, damit das Projekt als abgeschlossen gilt:

### Die Benutzer haben die Möglichkeit neue Stunden zu erstellen, bearbeiten und zu löschen.

### Der Superuser muss neue Benutzer und neue Klassen erstellen, bearbeiten und löschen können.

### Die Lehrer haben die Möglichkeit ihr vorgegebenes Passwort zu ändern.

### Die Stundenpläne sind öffentlich in einer Übersicht ersichtlich.

### Eine funktionale Login-Seite ist vorhanden.

### Auf der Startseite werden die Klassen aufgelistet, welche dann angeklickt werden können um den Stundenplan der Klasse anzusehen.

#### Muss-Ziele

Erstellen von Benutzern, Klassen und Stunden. Eine funktionierende Login-Seite ist zwingend. Eine Übersicht der Stundenpläne muss ebenfalls vorhanden sein.

#### Optionale Ziele

Die Lehrer sollen die Möglichkeit haben ihr vorgegebenes Passwort zu ändern.

## Betriebsumgebung

Als Entwicklungsumgebung benutzen wir XAMPP mit MySQL, PHP und Nginx. Für den produktiven Einsatz wird ein Linux-Server gebraucht.

## Lösungsvarianten

### Variante A

#### Beschreibung

Das Design wird mit dem CSS-Framework Bootstrap umgesetzt. Damit hätten wir bereits ein Gridsystem und essenzielle Design-Bausteine. Da die Mitarbeiter bereits mit dem Framework bekannt sind, wird der Aufwand noch mehr minimiert.

#### Realisierungsaufwand

3 Tage

#### Vorteile

* Gridsystem
* Einfach
* Geringer Zeitaufwand

#### Nachteile

* keine

### Variante B

#### Beschreibung

Das Design setzten wir mit AngularJS um, damit wir eine One-Page Website haben. Dafür müsste man jedoch noch AngularJS lernen, damit es umgesetzt werden kann.

#### Realisierungsaufwand

8 Tage

#### Vorteile

* Schöner Aufbau

#### Nachteile

* Grosser Zeitaufwand
* AngularJS muss noch erlernt werden

### Variante C

#### Beschreibung

Wir erstellen das Design selber, ohne irgendwelche Frameworks. So könnten wir das ganze Design komplett selber bestimmen und wären sehr flexibel. Jedoch braucht dies bedeutend mehr Zeit.

#### Realisierungsaufwand

5 Tage

#### Vorteile

* Hohe Flexibilität

#### Nachteile

* Grosser Zeitaufwand

## Lösungsbewertung

Die Variante A ist zu bevorzugen, da sie den kleinsten Aufwand braucht. Die Mitarbeiter kennen das Framework bereits und können das Frontend dadurch sehr schnell und effizient umsetzen.

## Risikoanalyse

In der Folgenden Analyse werden kritische Bestandteile gesucht, die das Projekt gefährden können. Die Schadenshöhe entspricht dabei dem Schaden das etwas verursacht, wenn es nicht funktioniert. Die Wahrscheinlichkeit definiert, mit welcher Wahrscheinlichkeit, dass dieses Szenario während des Projekts vorkommen wird.



Abbildung 2: Bewertung der Risiken

### Szenario/Situation 1 – Die Entwicklungsumgebung funktioniert nicht

Das Projekt kann in dieser Zeit nicht weitergeführt werden. Dies hätte Auswirkungen auf das Zeitmanagement, da man viel Zeit benötigt um das Problem zu beheben.

#### Mögliche Schadensbegrenzung

* Die Entwicklungsumgebung möglichst simpel halten

### Szenario/Situation 2 – Berechtigungssystem funktioniert nicht

Die Schadenhöhe durch dieses Szenario wäre ziemlich hoch, da jeder Lehrer neue Benutzer und neue Klassen erfassen, bearbeiten und löschen könnte. Die Wahrscheinlichkeit, dass dies nicht funktioniert ist mittel.

#### Mögliche Schadensbegrenzung

* Von Anfang an integrieren
* Stark priorisieren

### Szenario/Situation 3 – Erstellung der Stunden macht Schwierigkeiten

Wenn ein Lehrer neue Stunden erfasst und dabei die Zeitangaben nicht übereinstimmten, stimmt der Stundeplan nicht mehr. Die Schadenhöhe wäre hoch, da die Stundenpläne nicht mehr stimmen und dadurch die Schüler nicht mehr wissen, wann und wo Sie Unterricht haben.

#### Mögliche Schadensbegrenzung

* Darauf achten, dass die Zeitangaben korrekt in der Datenbank hinterlegt werden

### Szenario/Situation 4 – Datenbankverbindung funktioniert nicht

Falls die Datenbankverbindung nicht funktioniert, kann der Superuser keine neuen Benutzer und Klassen erstellen. Dadurch beinhaltet die Website ebenfalls keine Stundenpläne. Die Schadenhöhe wäre ziemlich hoch, da die Website nicht verwendet werden könnte. Die Wahrscheinlichkeit, dass dieses Szenario auftritt, ist allerdings niedrig.

#### Mögliche Schadensbegrenzung

* Datenbankverbindungen testen
* Zugriff auf die Datenbank so simpel wie möglich machen

Abgabe Pflichtenheft!

An dieser Stelle muss das Pflichtenheft dem Auftraggeber zur Annahme abgegeben werden. Mach mit ihm einen Besprechungstermin ab und schicke ihm vorgängig das Dokument per Mail. Nimm das Formular im Kapitel A.10 Abnahmeformular Pflichtenheft (**nur diese eine A4 Seite**) ausgedruckt zur Unterschrift mit an die Besprechung.

Nun wird vom Auftraggeber entschieden ob das Projekt so durchgeführt werden kann. In diesem Schritt kann es auch Anpassungen geben.

Warnung:

**Wer hier ohne Rücksprache mit dem Auftraggeber weiterfährt, gefährdet das Projekt.** Der Auftraggeber könnte die Annahme des Projekts am Schluss verweigern.

**In unserem Fall würde das einer Note 1 entsprechen.**

# Planung

In diesem Schritt werden die Arbeitspakete definiert. Normalerweise wird ein Arbeitspaket nur von einer Person abgearbeitet. Jedoch macht das während der Basisausbildung nicht überall Sinn. Vor allem wenn verlangt wird, dass alle Projektarbeiter alles kennen lernen.

Folgende Kriterien muss ein Arbeitspaket erfüllen:

1. Es muss von einem (nicht mehreren) Projektmitarbeiter erledigt werden können.
2. Der Mitarbeiter kann (muss nicht) das Paket in Teilschritte unterteilen => IPERKA.  
   (siehe Kapitel 11.6 IPERKA)

Das bedeutet, der Umfang der Arbeit die das Paket hat, muss sinnvoll gewählt werden. Zum Beispiel könnte ein Paket sein, die Dokumentation erstellen. Das gibt aber viel zu tun für eine Person, darum kann das Pakete in kleinere Pakete unterteilt werden. Zum Beispiel unter anderem in Risikoanalyse und Arbeitspakete definieren.

Mögliche Arbeitspakete sind:

* Windows Server installieren, Verzeichnisdienst aufsetzen.
* HTML Grundgerüst erstellen
* Firewall konfigurieren
* Benutzerdokumentation
* Testfälle erstellen
* Testfälle durchführen
* Testfälle dokumentieren

Die Arbeitspakete werden im folgenden Kapitel zur Erstellung der Zeitplanung (GANTT) verwendet.

## Projektstrukturplan (PSP)

Der Projektstrukturplan hilft eine bessere Übersicht über das Projekt zu erhalten. Im Unterschied zu den Mitarbeitern im Organigramm, enthält der PSP die Namen der Arbeitspakete (blau hinterlegt). Damit überhaupt eine Übersicht entsteht, macht es Sinn die Pakete zu verschiedenen Bereichen zuzuordnen. Für unser Projekt unterscheiden wir drei Pfade. Die Projektführung, die Entwicklung und die Integration. Für euer Projekt müsst ihr nur die gelben Arbeitspakete ändern. Diese könnt und sollt ihr nach Bedarf erweitern. Die blau hinterlegten Arbeitspakete kommen in jedem Projekt vor. Es mach darum sinn diese zu behalten zu beschreiben und in das GANTT Diagramm aufzunehmen.



## Arbeitspaket A – Design

|  |  |  |
| --- | --- | --- |
| **Design der Website erstellen** | | |
| Ziele (Was für Ziele werden gesetzt)   * Das Design der Website mit Hilfe des CSS-Frameworks Bootstrap erstellen | | |
| Ergebnisse (was muss am Schluss abgeschlossen sein)   * Das Design der Website fertigstellen | | |
| Schnittstellen / Abhängigkeiten (von welchen Arbeitspaketen ist dieses AP abhängig)   * - | | |
| Aktivitäten in diesem AP   * Bootstrap herunterladen / integrieren * CSS Code Validieren | | Termine [Anz. Tage, Was]  [mindestens Fertigstellungstermin]  ………………………………………………….  In 3 Tagen muss alles fertig sein |
| Aufwand (in Personentagen)  2PT – 3PT (Personentage)  Planung und Design [I,P,E] 1PT, Realisieren [R] 1.5PT, Test [K,A] 0.5PT | | |
| Terminkontrolle (Wie wird garantiert, dass die Termine eingehalten werden?, Risiken) | Qualitätskontrolle (Qualitätskriterien, Prüfmethoden und –Zeitpunkte) | Abschlussarbeiten |
| Soll / Ist Vergleich Kontrolle Terminplan | Schlusskontrolle: Prüfen ob die Ziele des Arbeitspakets erfüllt wurden | Dokumentation nachtragen  Meldung an Projektleiter |

## Arbeitspaket B – Datenbank

|  |  |  |
| --- | --- | --- |
| **Datenbank erstellen** | | |
| Ziele (Was für Ziele werden gesetzt)   * XAMPP installieren * Datenbank für User, Klassen und Stunden einrichten | | |
| Ergebnisse (was muss am Schluss abgeschlossen sein)   * XAMPP installiert * Datenbank eingerichtet und getestet | | |
| Schnittstellen / Abhängigkeiten (von welchen Arbeitspaketen ist dieses AP abhängig)   * - | | |
| Aktivitäten in diesem AP   * XAMPP installieren und konfigurieren * Datenbank für User, Klassen und Stunden erstellen | | Termine [Anz. Tage, Was]  [mindestens Fertigstellungstermin]  ………………………………………………….  In einem Tag muss alles fertig sein |
| Aufwand (in Personentagen)  0.5PT (Personentage)  Planung und Design [I,P,E] 0.2PT, Realisieren [R] 0.2PT, Test [K,A] 0.1PT | | |
| Terminkontrolle (Wie wird garantiert, dass die Termine eingehalten werden?, Risiken) | Qualitätskontrolle (Qualitätskriterien, Prüfmethoden und –Zeitpunkte) | Abschlussarbeiten |
| Soll / Ist Vergleich Kontrolle Terminplan | Schlusskontrolle: Prüfen ob die Ziele des Arbeitspakets erfüllt wurden | Dokumentation nachtragen  Meldung an Projektleiter |

## Arbeitspaket C – Models

|  |  |  |
| --- | --- | --- |
| **Models** | | |
| Ziele (Was für Ziele werden gesetzt)   * Models erstellen | | |
| Ergebnisse (was muss am Schluss abgeschlossen sein)   * Die Models sind fertiggestellt | | |
| Schnittstellen / Abhängigkeiten (von welchen Arbeitspaketen ist dieses AP abhängig)   * Datenbank | | |
| Aktivitäten in diesem AP   * Models erstellen | | Termine [Anz. Tage, Was]  [mindestens Fertigstellungstermin]  ………………………………………………….  In einem Tag muss alles fertig sein |
| Aufwand (in Personentagen)  1PT (Personentage)  Planung und Design [I,P,E] 0.1PT, Realisieren [R] 0.75PT, Test [K,A] 0.15PT | | |
| Terminkontrolle (Wie wird garantiert, dass die Termine eingehalten werden?, Risiken) | Qualitätskontrolle (Qualitätskriterien, Prüfmethoden und –Zeitpunkte) | Abschlussarbeiten |
| Soll / Ist Vergleich Kontrolle Terminplan | Schlusskontrolle: Prüfen ob die Ziele des Arbeitspakets erfüllt wurden | Dokumentation nachtragen  Meldung an Projektleiter |

## Arbeitspaket D – Mappers & Views

|  |  |  |
| --- | --- | --- |
| **Mappers & Views** | | |
| Ziele (Was für Ziele werden gesetzt)   * Die Mappers und die Views erstellen | | |
| Ergebnisse (was muss am Schluss abgeschlossen sein)   * Die Mappers und die Views sind fertiggestellt | | |
| Schnittstellen / Abhängigkeiten (von welchen Arbeitspaketen ist dieses AP abhängig)   * Datenbank | | |
| Aktivitäten in diesem AP   * Die einzelnen Mappers und Views erstellen | | Termine [Anz. Tage, Was]  [mindestens Fertigstellungstermin]  ………………………………………………….  In 4 Tagen muss alles fertig sein |
| Aufwand (in Personentagen)  3.5 (Personentage)  Planung und Design [I,P,E] 0.5PT, Realisieren [R] 2.5PT, Test [K,A] 0.5PT | | |
| Terminkontrolle (Wie wird garantiert, dass die Termine eingehalten werden?, Risiken) | Qualitätskontrolle (Qualitätskriterien, Prüfmethoden und –Zeitpunkte) | Abschlussarbeiten |
| Soll / Ist Vergleich Kontrolle Terminplan | Schlusskontrolle: Prüfen ob die Ziele des Arbeitspakets erfüllt wurden | Dokumentation nachtragen  Meldung an Projektleiter |

## Arbeitspaket E – Testing

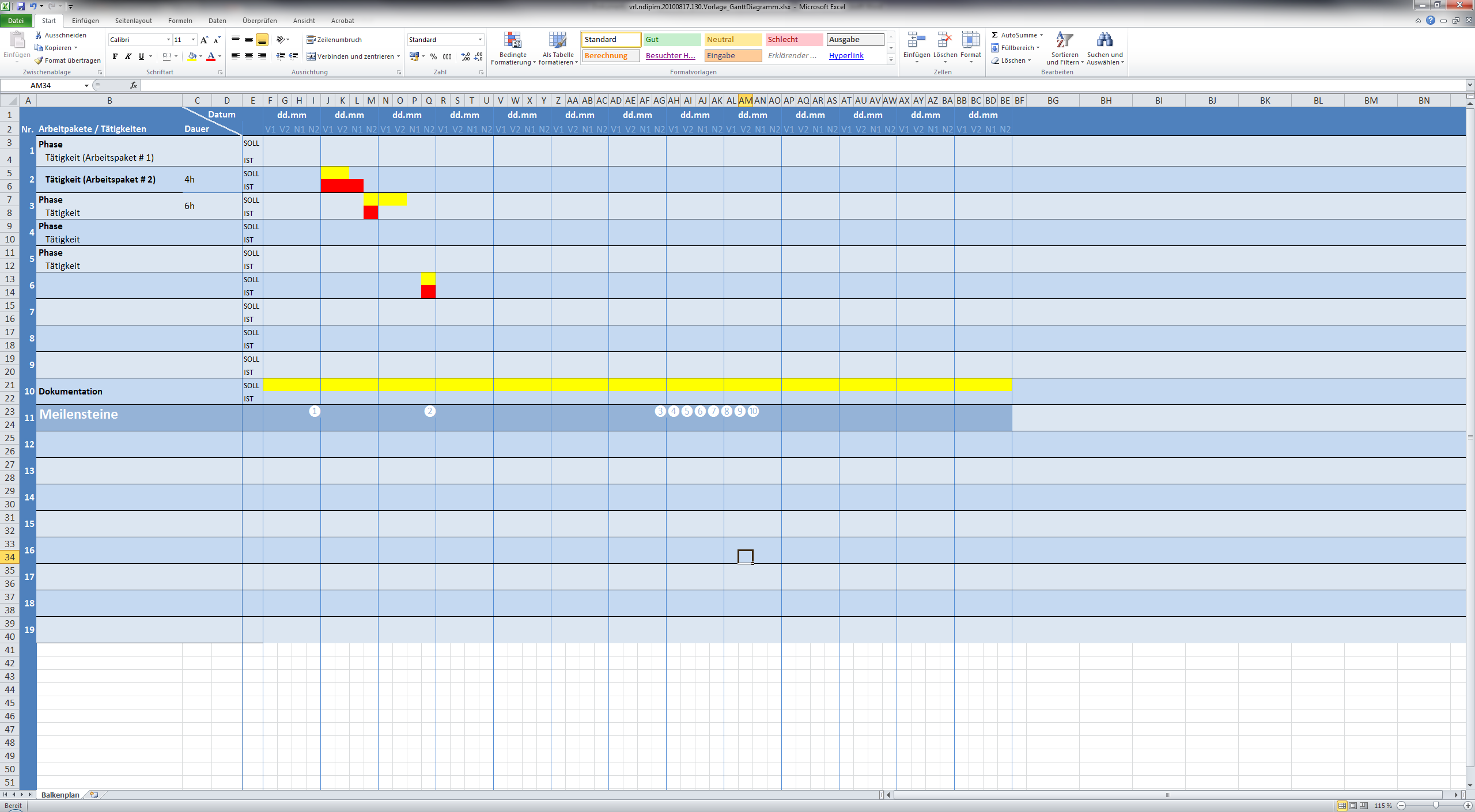
|  |  |  |
| --- | --- | --- |
| **Testing** | | |
| Ziele (Was für Ziele werden gesetzt)   * Die Website testen | | |
| Ergebnisse (was muss am Schluss abgeschlossen sein)   * Alle Funktionen auf der Website müssen funktionieren | | |
| Schnittstellen / Abhängigkeiten (von welchen Arbeitspaketen ist dieses AP abhängig)   * Design * Datenbank * Models * Mappers & Views | | |
| Aktivitäten in diesem AP   * Die einzelnen Funktionen der Website testen | | Termine [Anz. Tage, Was]  [mindestens Fertigstellungstermin]  ………………………………………………….  In einem Tag muss alles fertig sein |
| Aufwand (in Personentagen)  1PT (Personentage)  Planung und Design [I,P,E] 1PT, Realisieren [R] 1.5PT, Test [K,A] 0.5PT | | |
| Terminkontrolle (Wie wird garantiert, dass die Termine eingehalten werden?, Risiken) | Qualitätskontrolle (Qualitätskriterien, Prüfmethoden und –Zeitpunkte) | Abschlussarbeiten |
| Soll / Ist Vergleich Kontrolle Terminplan | Schlusskontrolle: Prüfen ob die Ziele des Arbeitspakets erfüllt wurden | Dokumentation nachtragen  Meldung an Projektleiter |

## Zeitplanung

### GANTT

Hier Excel Tabelle GANTT einfügen (auch als Bild möglich) Die Excel-Vorlage befindet sich im selben Ordner wie diese Dokumentationsvorlage.

Das Arbeitspaket „Dokumentation“ darf hier in der Planung nicht vergessen werden: Z.B. zu unterst, parallel während der ganzen Dauer einplanen. Mehr Informationen zum GANTT-Diagramm gibt es unter (Wikipedia GANTT-Diagramm).



Ein Feld entspricht 2h (wir rechnen mit einem 8h Tag)

# Technische Dokumentation (Produktdokumentation)

Hier seid ihr frei wie ihr euer Projekt beschreiben möchtet. Die folgenden Kapitel könnt ihr nach Belieben ändern, löschen oder ergänzen. Wichtig ist, dass aber alle Programmteile gut getestet werden, damit es während der Produktepräsentation nicht zu peinlichen Situationen kommt.

Mögliche Themen Elektronik: Mögliche Themen Informatik:

Hardwarebeschreibung Benutzerführung GUI  
 Elektronik Datenbank Design (ERM)  
 Mechanik Hilfsmittel (z.B. verwendetet APIs / Frameworks)

Softwarebeschreibung Verantwortlichkeiten (Tests, die ihr macht)   
 Softwaredesign Verwendete Software  
 Beschreibung der Module  
 Hilfsmittel (Compiler, IDEs usw.)

## Codestyle

### Namenskonventionen

#### Camel Case

Bei Camel Case wird jedes neue Wort mit einem Grossbuchstaben begonnen und keine Lücken oder Underscores benutzt. Der erste Buchstabe eines Wortes wird aber klein geschrieben.

Beispiele: *getFooBar, ichBinNurEinBeispiel, hansPeter*

#### Pascal Case

Bei Pascal Case wird jedes neue Wort mit einem Grossbuchstaben begonnen und keine Lücken oder Underscores benutzt.

Beispiele: *LessonClass, IchBinNurEinBeispiel, HansPeter*

#### Upper Case mit Underscore

Bei Upper Case mit Underscore wird jeder Buchstabe gross geschrieben und die Wörter mit Underscores getrennt.

Beispiele: *KONSTANTE\_FOO, ICH\_BIN\_NUR\_EIN\_BEISPIEL, HANS\_PETER*

### PHP

#### Curly Braces

Öffnende Curly Braces werden prinzipiell am Ende der Zeile geöffnet und haben keinen Code mehr dahinter. Die Schliessenden werden immer auf einer neuen Zeile gemacht.

#### Klassen

Dateien, die eine Klasse enthalten, dürfen ausserhalb der Klasse nur andere Dateien einbinden oder Namespaces setzen. Eine Datei mit einer Klasse darf nur eine Klasse enthalten und hat das folgende Namenschema: *<Klassenname>.class.php*

#### Namenskonventionen

|  |  |  |
| --- | --- | --- |
| Typ | Nameskonvention | Beispiel |
| Klasse | Pascal Case | class FooBar {} |
| Funktion | Camel Case | function fooBar() {} |
| Variable | Camel Case | $fooBar = “example“ |
| Konstante | Upper Case mit Underscore | $PROJECT\_PATH = “/usr/“ |

### Javascript

#### Semikolon

In Javascript sind die Semikolon überflüssig, daher werden sie nicht gebraucht.

#### Strikt-Modus

Am Anfang jeder JS-Datei muss ein „use strict“ sein, damit der Strikt-Modus benutzt wird

#### Namenskonventionen

|  |  |  |
| --- | --- | --- |
| Typ | Nameskonvention | Beispiel |
| Funktion | Camel Case | function fooBar() {} |
| Variable | Camel Case | var fooBar = true |
| Variable (jQuery Objekt) | Camel Case mit „$“ Prefix | var $fooBar = $(‚body‘) |

## Tests

In diesem Kapitel definiert ihr die Tests die Ihr macht.   
Es müssen minimal 5 Tests gemacht werden.

|  |  |
| --- | --- |
| **Testnummer** | **TF01** |
| Testfall | Login |
| Beschreibung | Das Login der Website testen. |
| Voraussetzung | Das Login muss funktionieren / das Arbeitspaket B (Datenbank) ist fertiggestellt, wurde getestet und funktioniert. |
| Testobjekt | Das Login der Website |
| Konsequenzen bei Nichterfüllung | Die betroffenen Dateien überprüfen und die Fehler beheben. |
| Schritte  (was muss gemacht werden) | Überprüfen, ob die Fehlermeldungen beim Einloggen funktionieren. Also Benutzernamen und Passwort. Das Einloggen ohne Passwort darf ebenfalls nicht funktionieren. Anschliessend testen ob man sich mit Benutzernamen und Passwort einloggen kann. |
| Zu erwartendes Ergebnis | Das alles problemlos funktioniert. |
| Ergebnis: | OK |
| Datum | 11.05.2015 |

|  |  |
| --- | --- |
| **Testnummer** | **TF02** |
| Testfall | Benutzerverwaltung |
| Beschreibung | Das Erstellen, Bearbeiten und Löschen von Benutzern testen. |
| Voraussetzung | Das Login muss funktionieren / das Arbeitspaket B (Datenbank) ist fertiggestellt, wurde getestet und funktioniert. |
| Testobjekt | Die Erstellung und Bearbeitung der Benutzer testen. |
| Konsequenzen bei Nichterfüllung | Die betroffenen Dateien überprüfen und die Fehler beheben. |
| Schritte  (was muss gemacht werden) | Testen ob man einen bereits vorhandenen Benutzern bearbeiten oder löschen kann. Dieses Recht hat allerdings nur der Superuser. Die restlichen Benutzer sollen nur die Möglichkeit haben ihr eigenes Passwort zu ändern. Zudem ist der Superuser in der Lage einen neuen Benutzer zu erstellen. |
| Zu erwartendes Ergebnis | Das alles problemlos funktioniert. |
| Ergebnis: | OK |
| Datum | 11.05.2015 |

# Benutzerhandbuch

An dieser Stelle kommt ein kleines Benutzerhandbuch. Beschreibt euer Produkt so, dass es jedermann mit der Bedienungsanleitung verwenden kann.

## Anmelden, Login, Logout oder z.B. Inbetriebnahme bei einem Gerät

## Funktionen

# Weiterentwicklungsmöglichkeiten

Beschreibt hier wie es mit eurem Projekt weitergeht nachdem es fertig ist. Zum Beispiel:

* Was könnte man erweitern?
* Was könnte man optimieren?
* Gibt es optionale Arbeitspakete die das Produkt ideal ergänzen?
* Usw.

# Fazit

Hier kommt eure Reflexion zum Projekt.

* Was lief gut/schlecht?
* Wie seid ihr mit dem Endergebnis zufrieden?
* Was habt ihr gelernt?
* War alles vorhanden oder was fehlte noch?
* Usw.

# Abkürzungen / Begriffserklärung

Hier werden alphabetisch alle Fachbegriffe und Abkürzungen erklärt:

HTML Hypertext Markup Language

MySQL Rationales Datenbankverwaltungssystem

PCB Printed Circuit Board

I2C Inter-Integrated Circuit (serieller Datenbus)

1. Anhang
   1. Schema
   2. Fertigungsdokumente
   3. Beschreibung der Ordnerstruktur
   4. Testprotokolle
   5. JavaDoc
   6. Mindmaps
   7. …
   8. Ressourcen
      1. Bilder

In folgenden Galerien sind Bilder verfügbar, die unter gewissen Bedingungen für das Projekt verwendet werden können. Die jeweiligen Nutzungsrechte können für jedes Bild verschieden sein und müssen entsprechend geprüft werden.

[search.creativecommons.org](http://search.creativecommons.org/)  
Allows you to search google and flickr for creative commons images, and several other search providers for non-image based content (try attaching "image of" or "stock photo" to your search terms.

[images.google.com](http://images.google.com/)  
Click on "advanced image search" and then select for the "Usage Rights" dropdown, ALWAYS confirm that the images you find are actually licensed as you selected.

[commons.wikimedia.org](http://commons.wikimedia.org/)  
more than 10 million categorized, photos, illustrations, footage, sound bytes etc.

[flickr.com/creativecommons](http://www.flickr.com/creativecommons/)  
More than 100 million CC licensed images from the popular photo sharing site (15 million with the most flexible "attribution only" license).

[geograph.org.uk](http://www.geograph.org.uk/)  
contributors submit images covering locations all over the uk sorted on ordiancesurvey map references.

[everystockphoto.com](http://everystockphoto.com/)  
Search a selection of free images sites, the license selector in advanced search allows you to filter by creative commons and other free license types.

[creativity103.com](http://creativity103.com/)  
unusual selection of textures, backdrops and abstract photos and illustration

[animalphotos.info](http://animalphotos.info/)  
Specialist collection of animal photos, well categorized.

[carpictures.cc](http://carpictures.cc/)  
A rev-heads wet dream, browse car photos organized by marque and model, sourced from flickr

* 1. IPERKA



Abbildung 3: IPERKA

* 1. Abnahmeformular Pflichtenheft

Projektname:

Kurze Beschreibung:

* + 1. Projektmitglieder

Projektleiter Name: Vorname: Hauptaufgabe:

Teilprojektleiter Name: Vorname: Hauptaufgabe:

Teilprojektleiter Name: Vorname: Hauptaufgabe:

Durch den Coach auszufüllen:

Notizen:

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

Datum: …………………………………….. Unterschrift Coach: ……………………………………

* 1. Usw.

# Beurteilungskriterien (gemäss MP-307-414)

|  |  |
| --- | --- |
| **Gestaltung der Oberfläche (5 Punkte)** | **Punkte** |
| Ist die Navigation klar ersichtlich und benutzerfreundlich? |  |
| Ist die Webseite, unter Berücksichtigung der Zielgruppe, selbsterklärend? |  |
| Sind die Inhaltselemente einzelner Seiten klar strukturiert und einheitlich? |  |
| Wurden gezielt grafische- und/oder Bildelemente eingesetzt? |  |
| Ist ein Farbkonzept sichtbar und einheitlich umgesetzt? |  |
| Total: |  |
| Gewichtetes Total | x1 = |
| Zielerreichung (8 Punkte) | *Punkte* |
| 100% aller Mussziele wurden erreicht |  |
| Es wurden sinnvolle Kann-Ziele definiert und evtl. sogar umgesetzt |  |
| Der Umfang der Arbeit ist angemessen bezüglich der Projektdauer |  |
| Alle Projektmietglieder wurden gleichmässig beschäftigt |  |
| Total: |  |
| Gewichtetes Total | x2 = |
| Technische Realisierung (14 Punkte) | *Punkte* |
| Es wird bei Formularen im Browser eine Plausibilitätsprüfung durchgeführt |  |
| Es wird bei Formularen serverseitig eine Plausibilitätsprüfung durchgeführt |  |
| Der Code ist übersichtlich Formatiert und entspricht den Vorgaben. |  |
| Der Code ist gut kommentiert so, dass jemand anderes weiterfahren kann. |  |
| Es gibt praktisch keine Redundanzen im Code (z.B. keine Funktionen doppelt) |  |
| Ist die Webseite immun gegen Cross Site Scripting? |  |
| Die Datenbank Abfragen sind sicher z.B. gegen SQL-Injections |  |
| Total: |  |
| Gewichtetes Total | x2 = |

|  |  |  |
| --- | --- | --- |
| Dokumentation (10 Punkte) | | *Punkte* |
| Ist die Dokumentation vollständig? | | |
| Planung (Pflichtenheft, Terminplanung, Kann/Muss Ziele) |  | |
| Technische Dokumentation |  | |
| Benutzerdokumentation |  | |
| Die Dokumentation erfüllt die formalen Anforderungen |  | |
| Die Dokumentation ist ausführlich und seriös nachgeführt worden |  | |
| Total: |  | |
| Gewichtetes Total | x2 = | |