Inhalt

[Aufgabenstellung 2](#_Toc472117970)

[Situationsanalyse 2](#_Toc472117971)

[Ziele 3](#_Toc472117972)

[Mussziele 3](#_Toc472117973)

[Wunschziele 4](#_Toc472117974)

[Use Case Diagrams 4](#_Toc472117975)

[Lösungsvariante 4](#_Toc472117976)

[Twitter Bootstrap 4](#_Toc472117977)

[Programmiersprache 4](#_Toc472117978)

[Framework 4](#_Toc472117979)

[Datenbank 4](#_Toc472117980)

[Versionsverwaltungsprogramm 4](#_Toc472117981)

[Integrated Development Environment 4](#_Toc472117982)

[Meilensteine 5](#_Toc472117983)

[Projektplanung 6](#_Toc472117984)

[Architektur & Komponenten bestimmen 6](#_Toc472117985)

[GIT Hub Projekt einrichten 7](#_Toc472117986)

[IDE einrichten 7](#_Toc472117987)

[DB Design 7](#_Toc472117988)

[Realisierungsphase 1 8](#_Toc472117989)

[Realisierungsphase 2 8](#_Toc472117990)

[Realisierungsphase 3 9](#_Toc472117991)

[Realisierungsphase 4 9](#_Toc472117992)

[Realisierungsphase 5 10](#_Toc472117993)

[Testen 10](#_Toc472117994)

[Bugfixing 10](#_Toc472117995)

[Zwischenstand 11](#_Toc472117996)

[PowerPoint-Präsentation erstellen 11](#_Toc472117997)

[Dokumentation 11](#_Toc472117998)

[Abgabe & Präsentation 12](#_Toc472117999)

# Aufgabenstellung

## Situationsanalyse

## Ziele

### Mussziele

|  |  |  |
| --- | --- | --- |
| ID | Prio | Beschreibung |
| FA\_1.1 | 1 | Im Web-Shop sollen Artikel über das Internet angeboten und verkauft werden. Die Aspekte der Bezahlung und der Auslieferung werden der Einfachheit halber nicht berücksichtigt. |
| FA\_1.2 | 1 | Die Artikel im Web-Shop sind in Produktkategorien in 2 Levels strukturiert |
| FA\_1.3 | 1 | Die Produktkategorien und die Zuordnung der Produkte werden in der DB konfiguriert, dadurch ist der Web-Shop generisch für die verschiedensten Produkte verwendbar |
| FA\_1.4 | 1 | Produkte werden in der DB konfiguriert. Sie haben minimal die folgenden Attribute Produkt-ID (eindeutig), Bezeichnung (max. 200 Zeichen), Beschreibung (max. 2000 Zeichen), 0-5 Bilder, Status (active, hidden) |
| FA\_1.5 | 2 | Zusätzlich werden zu jedem Produkt noch die folgenden Angaben verwaltet: Lagerbestand, Nachbestellung (Lieferdatum und Bestellmenge) |
| FA\_1.6 | 1 | Mittels geeigneter Navigation kann der Kunde durch Produktkategorien und Produkte navigieren |
| FA\_1.7 | 1 | Der Kunde kann Produkte seinem Warenkorb hinzufügen. |
| FA\_1.8 | 1 | Der Kunde kann seinen Warenkorb anzeigen lassen (Auflistung der Produkte inkl. Preis und Menge) und editieren (Menge ändern, Produkte herausnehmen) |
| FA\_1.9 | 2 | Sobald ein Kunde Produkte im Warenkorb hat, kann er "zur Kasse gehen". Dort wird ihm die Bestellung nochmals angezeigt, er muss seine Adresse und Email eingeben und kann die Bestellung auslösen. |
| FA\_1.10 | 2 | Alle Angaben einer Bestellung (Warenkorb, Kundendaten, Bestelldatum) werden in der DB gespeichert |
| FA\_1.11 | 2 | Anzeigen einer Liste aller Produkte. Durch Anklicken des Produktes gelangt man in die Detailansicht welche alle spezifizierten Attr. des ausgewählten Produktes darstellt (Admin Funktion) |
| FA\_1.12 | 2 | Anzeigen einer Liste aller Bestellung. Durch Anklicken einer Bestellung gelangt man in die Detailansicht (Admin Funktion) |
| FA\_2.1 | 1 | Die Administration von Produkten und Produktkategorien (anlegen, editieren, löschen) soll auf einer entsprechenden Admin Seite möglich sein. |
| FA\_2.2 | 1 | Ein Kunde kann sich im Web-Shop registrieren. Nach dem Login mit der Email und Passwort muss er bei einer Bestellung seine Adresse nicht mehr eingeben. |
| FA\_2.3 | 2 | Das Passwort darf nicht als Klartext in der DB abgelegt werden. |
| FA\_2.4 | 2 | Ein registrierter Kunde kann seine Daten einsehen und editieren. |
| FA\_3.1 | 1 | Der Shop wird in der Basiswährung CHF betrieben. Die Shop Besucher sollen auf Wunsch (Popup oder ähnliches) eine andere Anzeigewährung auswählen können. (US$, € und JPY) |
| FA\_3.2 | 1 | Der Kurs soll jeweils (mindestens) Tagesaktuell von der Webseite <http://www.ecb.europa.eu/stats/eurofxref/eurofxref-daily.xml> sein. Alternativ ist auch ein anderer Onlinekurslieferant erlaubt. |
| FA\_3.3 | 2 | Die Bestellung wird in Basswährung ausgeführt, es soll aber auch die Ausgewählte Währung des Kunden, sowie der Kurs zum Zeitpunkt der Bestellung ersichtlich sein. |
| NFA\_1.1 | 1 | Ein User mit minimalen Internetkenntnissen muss den Web-Shop intuitiv bedienen können. (Benutzbarkeit) |
| NFA\_1.2 | 2 | Der Web-Shop soll unter Linux und Windows lauffähig sein (Übertragbarkeit) |
| NFA\_1.3 | 2 | Die Antwortzeit des Web-Shops soll unter 3 Sek. liegen (Zeitverhalten) |
| NFA\_1.4 | 2 | Der Shop muss gegen SQL Injections immun sein |
| NFA\_1.5 | 3 | Der Shop muss auch gegen XSS Angriffe abgesichert sein |

### Wunschziele

## Use Case Diagrams

# Lösungsvariante

## Twitter Bootstrap

Eine Weblösung kommt heut zu Tage ohne Bootstrap CSS nicht mehr aus, die meisten Firmen verwenden den Bootstrap. Da auf das Design der Seite nicht viel Wert gesetzt wird, werden wir Bootstrap Plain einsetzen ohne eigene Customization vorzunehmen.

## Programmiersprache

Für die Umsetzung des Auftrags haben wir drei Entwicklungswerkzeuge PHP, JAVA, SCALA näher in Betracht gezogen und miteinander verglichen.

|  |  |  |  |
| --- | --- | --- | --- |
|  | PHP | JAVA | SCALA |
| Einfache Programmiersprache | 5 | 1 | 0 |
| Einfache Webserver Konfiguration | 2 | 3 | 3 |
| Einfacher Debugging-Support | 2 | 5 | 5 |
| OOP-Support | 3 | 5 | 5 |
| Type-Safe | 0 | 5 | 5 |
| IDEA Compiler-Support | 3 | 5 | 5 |
| Saubere Trennung zwischen GUI und Logik | 5 | 5 | 5 |
| Robuste Frameworks vorhanden | 5 | 5 | 5 |
| Potenzielle Entwickler auf dem Markt | 5 | 3 | 0 |
| Geringe Serverauslastung | 5 | 2 | 2 |
| Geringe Hosting Kosten | 5 | 1 | 1 |
| Total | **40** | **40** | **36** |

Für die Implementierung werden wir PHP verwenden, weil die meisten Entwickler sich mit PHP mehr oder weniger auskennen. Zudem gibt es viele Web Hoster welche PHP Out of the Box unterstützen.

## Framework

|  |  |  |  |
| --- | --- | --- | --- |
|  | Symfony | Laravel | CodeIgniter |
| Ausführliche Dokumentation | 5 | 2 | 3 |
| Bereite Community | 5 | 3 | 2 |
| Vorkenntnisse im Team vorhanden | 3 | 0 | 0 |
| IDE Support | 4 | 3 | 2 |
| Total | **17** | **8** | **7** |

Da für die Umsetzung die Sprache PHP gewählt wurde kommen nicht mehr viele Frameworks zu Frage. Symfony ist eines der bekanntesten und hat eine bereite und ausgereifte Community was sehr wichtig ist für neu Einsteiger.

Direkter Link Symfony Framework: <https://symfony.com/>

## Datenbank

|  |  |  |  |
| --- | --- | --- | --- |
|  | MS-SQL | MySQL | Postgre-SQL |
| Opensource | 0 | 5 | 5 |
| User-Defined Data-Ttypes | 1 | 2 | 5 |
| User-Defined Domains | 2 | 0 | 3 |
| Plattform unabhängig | 0 | 3 | 5 |
| Geringe Aufwand für Initialisierung | 1 | 5 | 1 |
| Vorkenntnisse im Team vorhanden | 2 | 5 | 1 |
| GUI Oberfläche von Haus aus unterstützt | 0 | 5 | 0 |
| Total | **6** | **25** | **20** |

Als Datenbank werden wir MySQL verwenden mit PHP-My-Adamin.

## Versionsverwaltungsprogramm

|  |  |  |  |
| --- | --- | --- | --- |
|  | GIT | TSF | SVN |
| Intelligentes System | 5 | 5 | 2 |
| Verbreitet | 5 | 2 | 5 |
| Bereite Community | 5 | 3 | 3 |
| Plattform unabhängig | 5 | 3 | 5 |
| Vorkenntnisse im Team vorhanden | 5 | 0 | 3 |
| Total | **25** | **13** | **18** |

Für die Realisierung des Projektes werden wir GIT für ein verteiltes Version Verwaltung einsetzen. Als GIT Repository werden wir GitHub verwenden, das Projekt wird unter dem Namen **ibz-it5-casestudy-webshop** veröffentlicht.

Direkter Link: <https://github.com/AruPersia/ibz-it5-casestudy-webshop>

## Integrated Development Environment

Als IDE werden wir IntelliJ als Entwicklungsumgebung verwenden welche eine gute Basis für PHP und Symfony Framework anbietet.

Direkter Link IntelliJ: <https://www.jetbrains.com/idea/>

# Meilensteine



### Projektplanung

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| * Vorgehensmodell * Definition der Arbeitspakete (ID, Beschreibung, Zeitaufwand, Abhängigkeiten) * Zeitplanung, es soll für jede Woche ersichtlich sein, wer wie lange an welchen Arbeitspaketen arbeitet * Zu erreichende Milestones | AAF | 8 | 6 | 2 |  |

### Architektur & Komponenten bestimmen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| * Beschreibung der SW Architektur des Web-Shops * Welche Technologien und Komponenten eingesetzt werden (Datenbank, Programmiersprache, Webserver, CMS ...) inkl. Version | AAF | 4 |  |  |  |

### GIT Hub Projekt einrichten

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| * GIT Hub Projekt eröffnen | AAF | 2 |  |  |  |

### IDE einrichten

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| * IntelliJ einrichten, GIT Hub Zugriff testen | AAF | 4 |  |  |  |
| * IntelliJ einrichten, GIT Hub Zugriff testen | ASC | 4 |  |  |  |
| * IntelliJ einrichten, GIT Hub Zugriff testen | OWA | 4 |  |  |  |

### DB Design

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| Datenbank Schema erstellen | AAF | 8 |  |  |  |

### Realisierungsphase 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| DTOs erstellen   * FA\_1.4 * FA\_1.5 * FA\_3.1 (berücksichtigen) * FA\_3.2 (berücksichtigen) | AAF | 8 |  |  |  |
| DAOs erstellen | AAF | 8 |  |  |  |
| Forms erstellen | AAF | 8 |  |  |  |

### Realisierungsphase 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| Routings definieren | AAF | 4 |  |  |  |
| Controllers definieren (Vorbereitung) | AAF | 8 |  |  |  |
| Produkt erfassen (Vorbereitung)   * FA\_1.2 * FA\_1.3 | AAF | 8 |  |  |  |
| Registration & Login   * FA\_2.2 * FA\_2.3 | AAF | 8 |  |  |  |

### Realisierungsphase 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| Navigation erstellen   * FA\_1.6 |  | 4 |  |  |  |
| Warenkorb implementieren   * FA\_1.7 * FA\_1.8 |  | 8 |  |  |  |
| Zahlungsabschluss (Kasse/Bestellung) implementieren   * FA\_1.9 * FA\_1.10 |  | 8 |  |  |  |

### Realisierungsphase 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| Registrierter Kunde kann sein Profil bearbeiten   * FA\_2.4 |  | 8 |  |  |  |
| Admin Funktion: Produkten (anlegen, editieren, löschen)   * FA\_2.1 |  | 8 |  |  |  |

### Realisierungsphase 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| Preis in andere Währung anzeigen   * FA\_3.1 * FA\_3.2 * FA\_3.3 |  | 8 |  |  |  |
| Admin Funktion: Anzeigen alle Produkte und Bestellungen   * FA\_1.11 * FA\_1.12 |  | 8 |  |  |  |

### Testen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| Webshop testen und prüfen ob alle Ziele erreicht wurden |  | 4 |  |  |  |

### Bugfixing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Gefundene Bugs | Wer | Geplant | Geleistet | Delta | Erklärung |
|  |  | 80 |  |  |  |

### Zwischenstand

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| * Aufzeigen des Soll/Ist Vergleiches von Zeitplanung und Milestones |  | 2 |  |  |  |

### PowerPoint-Präsentation erstellen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| * Vorbereitung einer PowerPoint-Präsentation * Planung & Einteilung wer präsentiert welche Folien |  | 4 |  |  |  |

### Dokumentation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| * Projektplanung (siehe oben) inkl. Soll/Ist Vergleiche von Zeitplanung und Milestones * Zusammenfassung der erreichten Resultate * SW Design (UML Diagramme, Beschreibung der wichtigsten Funktionen) * Datenbank Design (ER - Diagramme) * Positive und negative Aspekte der Arbeit * Lessons Learned |  | 24 |  |  |  |

### Abgabe & Präsentation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aktivität | Wer | Geplant | Geleistet | Delta | Erklärung |
| * Applikation im Quelltext, inklusive Buildframework als ZIP Datei * Binärversion + Datenbankdump der Applikation * URL zum lauffähigen Projekt, so dass der Shop getestet werden kann. * Der Shop muss mindestens noch für eine Woche nach der Präsentation in Betrieb sein |  | 4 |  |  |  |