

# Winterprüfung 20xx

## **Thema der Projektarbeit**

Automatisierter Mechanismus zum Laden/Verifizieren von  
Softwarekomponenten

---

# 1 Thema der Projektarbeit

Automatisierter Mechanismus zum Laden/Verifizieren von Softwarekomponenten

## 2 Geplanter Bearbeitungszeitraum

Beginn: 01.10.20xx

Ende: 31.10.20xx

## 3 Projektbeschreibung

Im Rahmen der Erstentwicklung einer Platform-as-a-Service (PaaS) ERP Software (folgend Trägersoftware) soll zur generellen Wartbarkeit und zur Pflege des Bestandscodes ein System für das Laden und Verifizieren modularer Softwarekomponenten entwickelt werden. Ein Modul ist eine Softwareeinheit, die einem sehr speziellen, klar abgegrenzten Zweck dient.

Um den automatisierten Mechanismus zur Validierung und Verifizierung zu ladender Softwarekomponenten (folgend Moduleloader) zu implementieren, müssen diverse Kernkomponenten der Trägersoftware angepasst werden.

Das System grenzt sich dabei von folgenden Kernkomponenten ab: SiteManager, PageController, ThemeLoader und Themes. Ein Beispiel-Modul wird soweit implementiert, dass die Funktionsweise des Moduleloaders getestet werden kann.

### 3.1 Zielsetzung

- Lizenzierte Module sollen ins System geladen werden
- Das Integrieren dieser Module soll möglichst einfach sein
- Lizenzierung soll pro Modul gewährleistet werden
- Wenn die Lizenz abgelaufen ist, soll sich das Modul deaktivieren

### 3.2 Aufgaben

- Eine Modulrelation mit den Informationen der einzelnen Module
- Grundlegendes Benutzerinterface zum Anlegen/Lesen/Aktualisieren und Löschen von Modul-Listen
- Datenbankrelation, welche Kunde, Modul und Lizenzablaufdatum enthält
- Datenbankrelation, welche die Abhängigkeit von Modulen und Themes enthält
- Eine Modulklassse, welche diverse Informationen für das Modul enthält
- Beim Seitenaufbau soll die Software prüfen, ob der Kunde aktivierte Module besitzt
- Ebenfalls soll bei aktivierten Modulen geprüft werden, ob diese noch lizenziert sind
- Wenn Module aktiviert sind, müssen diese mit dem PageController geladen werden
- Templates müssen einen Verweis auf das Modul erhalten

- 
- Droht die Modullizensierung abzulaufen (Frühwarnzeitraum einstellbar), wird eine Benachrichtigung für eine spätere Verwendung gespeichert
  - Wenn die Lizenzkontrolle negativ ausfällt, wird das Modul nicht geladen

### 3.3 Technische Ressourcen

Verwendet wird PHP und HTML um Logik und Struktur der Software zu erstellen. Als Editor wird der ATOM Editor verwendet, als Versionskontrolle GitHub. Der serverseitigen Programmiersprache PHP liegt eine Installation eines lokalen Webserver (XAMPP) zugrunde.

## 4 Projektumfeld

Auftraggeber ist eine Werbeagentur mit Schwerpunkt Onlinepräsenz. Zum Kerngeschäft gehört das Erstellen, Optimieren und Warten von Webseiten, das Entwickeln von Forensoftware sowie das Programmieren von Logistik- und Verwaltungslösungen. Zurzeit arbeitet die Agentur an der Diversifikation der Angebotspalette. Hierzu gehört auch ein Framework zum Einbinden von ERP-Systemen, welches 20xx released wird. Die Agentur beschäftigt noch einen weiteren Praktikanten, dessen Projekt sich klar von meinem abgrenzt.

## 5 Projektphasen mit Zeitplanung

### 1. Analyse (Gesamtzeit 1h)

#### 1.1 Durchführen der Ist-Analyse 1h

### 2. Projektplanung (Gesamtzeit 11h)

#### 2.1 Entwerfen eines Komponentendiagramms (Gesamtzeit 3h)

#### 2.2 Entwerfen einer Datenbankstruktur inkl. ER-Modell 4h

#### 2.3 Ableiten des Tabellen-, und Domänenmodells aus dem ER-Modell inkl. Klassendiagramm 4h

### 3. Implementierung (Gesamtzeit 40h)

#### 3.1 Anlegen der nötigen Projektdateien 2h

#### 3.2 Implementierung der Relationen des Klassendiagramms 4h

#### 3.3 Anlegen eines einfachen Beispielsmoduls 2h

#### 3.4 Implementieren der CRUD-Methoden (CRUD = Create, Read, Update, Delete) im ModuleController der Module-Liste 4h

#### 3.5 Erstellen der MySQL-Abfragen für die CRUD-Methoden 4h

#### 3.6 Anlegen von einfachen Templates, um die Methoden aus 3.6 anzuwenden 3h

#### 3.7 Test der CRUD-Methoden 3h

#### 3.8 Implementieren des Moduleloaders 5h

#### 3.9 Implementieren der Lizenkontrolle 5h

#### 3.10 Modul-Verfügbarkeit für Templates implementieren 3h

3.11 Verfügbares Modul in Template anzeigen 1h

4. Test des Systems (Gesamtzeit 10h)

4.1 Planung der Tests 1 h

4.2 Test des Moduleloaders 3 h

4.3 Test der Modul-Verfügbarkeit 2 h

4.4 Test der Lizenzkontrolle 3 h

4.5 Korrekturen 1 h

5. Abschlussphase (Gesamtzeit 4h)

5.1 Code-Review und Abnahme durch Projektleiter 2h

5.2 Erstellen des Soll-Ist-Vergleichs 2h

6. Dokumentation (Gesamtzeit 14h)

6.1 Erstellen der Projektdokumentation 8h

6.2 Erstellen von Diagrammen und Schaubildern 6h

(Gesamtzeit des Projektes 80h)

## **6 Präsentationsmittel**

Beamer, Notebook, 15 Minuten Rüstzeit