

Übungsblatt 13 (19.01.2016)

Use Case, Architektur

In dieser Übung:

- ✓ Erstellen Sie einen Use Case für den *Movie Manager* Webclient.
- ✓ Bewerten Sie verschiedene Architekturmuster für die *Movie Manager* Anwendung.
- ✓ Leiten Sie aus Ihren Erfahrungen Funktionalitäts-Vorschläge für UNICASE ab.

Aufgabe 13.1: Use Case für den Movie Manager Webclient

Präsenz: Ja	Punkte: 12	Team: Ja(4)	Projekt: Ja	
-------------	------------	-------------	-------------	--

In dieser Aufgabe erstellen Sie anhand Ihrer Implementierung einen Use Case für den *Movie Manager* Webclient. Beachten Sie bei der Bearbeitung auch den Stilratgeber für Use Cases in Moodle.

Erstellen Sie für den *Movie Manager* Webclient einen neuen Use Case **Loan Items** in Ihrem UNICASE Projekt, der den typischen Anwendungsfall zum Ausleihen und Zurückgeben von Filmen, Serien, Staffeln und Episoden im *Movie Manager* Webclient widerspiegelt. Sollte die Ausleih- und Rückgabefunktionalität auf zwei Arbeitsbereiche verteilt sein, so beschreiben Sie das in zwei getrennten Use Cases. Geben Sie die Arbeitsbereiche und Systemfunktionen entsprechend Ihres UNICASE Projekts an und ergänzen Sie Regeln und mögliche Ausnahme im Use Case. Achten Sie bei der Erstellung des Anwendungsfalls darauf, dass auch die Funktionen zum Filtern und Sortieren enthalten sind. Speichern Sie Ihren Use Case in dem Abschnitt „Use Cases“ im „ReqSpec Document“ Ihres UNICASE Projekts.

Sie können die Lösung im Team abgeben, wir empfehlen aber, dass Sie die Erstellung einzeln üben.

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server!

Bitte speichern Sie eine .zip-Datei bis **Montag 25.01.2016 um 10.00 Uhr** in Moodle bestehend aus:

- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)

Aufgabe 13.2: Architekturmuster für Movie Manager Anwendung

Präsenz: Nein	Punkte: 12	Team: Ja(4)	Projekt: Nein	
---------------	------------	-------------	---------------	--

Bewerten Sie die untenstehenden Architekturvorschläge für die *Movie Manager* Anwendung. Prüfen Sie die Vorschläge auf Unterstützung für die von den KundInnen gestellten Architekturkriterien.

- Teilen Sie die Architekturvorschläge in Ihrem Team auf. Bewerten Sie jeden Architekturvorschlag gegen die nachfolgend gegebenen Architekturkriterien.
 - Verwenden Sie die Rationale-Funktionalität von UNICASE um die Architekturvorschläge gegen die Architekturkriterien zu bewerten: Erstellen Sie ein Element vom Typ „Issue“, in welchem Sie die Vorschläge anhand der Architekturkriterien bewerten und sich für eine Lösung entscheiden. Speichern Sie die Rationale-Elemente im „Design Document“ -> „Architecture“.

- Begründen Sie für jeden Architekturvorschlag und jedes Kriterium Ihre Bewertung im Beschreibungstext des Architekturvorschlages, d.h. des dafür von Ihnen erstellten „Issues“, insbesondere die Einschätzung auf der Skala im Vergleich zu den anderen Vorschlägen.
2. Erarbeiten Sie in Ihrem Team aus allen vorgeschlagenen und bewerteten Architekturvorschlägen einen neuen Vorschlag, der Ihrer Meinung nach die Architekturkriterien am besten erfüllt. Verwenden Sie die Rationale-Funktionalität von UNICASE, um Ihren Vorschlag gegen die Architekturkriterien zu bewerten und begründen Sie Ihre Bewertung. Benutzen Sie die folgende Skala:

Skala	Bedeutung
2	Eigenschaft wird optimal erfüllt
1	Eigenschaft wird gut erfüllt
0	Eigenschaft wird ausreichend erfüllt
-1	Eigenschaft wird nicht ausreichend erfüllt
-2	Eigenschaft kann gar nicht unterstützt werden

Ergebnis:

Bitte speichern Sie Ihre Änderungen am UNICASE Projekt auf dem Server.

Bitte speichern Sie eine .zip-Datei bis **Montag 25.01.2016 um 10.00 Uhr** in Moodle bestehend aus:

- Exportiertes UNICASE-Projekt als .ecp (damit Stand des Projekts bewertet werden kann)
- Begründung für Bewertung als .pdf

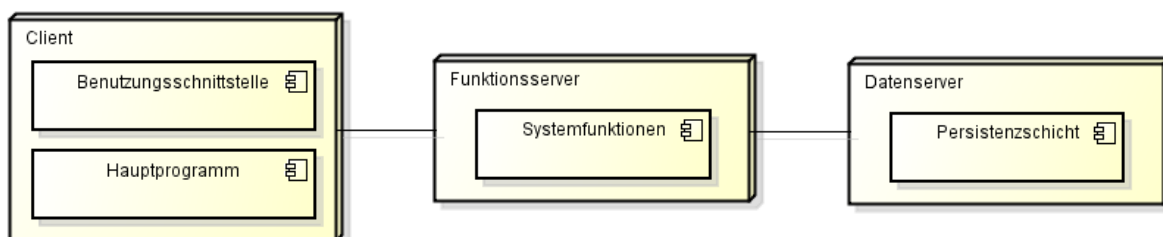
Architekturkriterien für Aufgabe 13.2

Architekturkriterien:

- „Keine zusätzliche Hardware erforderlich, außer dem Rechner auf dem die Movie Manager Anwendung bereits ausgeführt wird.“ (Architectural Constraint)
- „Die Gesamtmenge von Daten, die über Netzwerke ausgetauscht werden, soll klein sein“ (Architectural Constraint)
- „Sichere Verbindung zu den Dokumenten“ (Architectural Constraint)
- „Weltweite Verfügbarkeit“ (Global Functional Constraint)
- „Leicht anpassbar für verschiedene Betriebssysteme“ (Design Goal)
- „Gute Erweiterbarkeit um zusätzliche Funktionalität“ (Design Goal)

Architekturvorschlag 1:

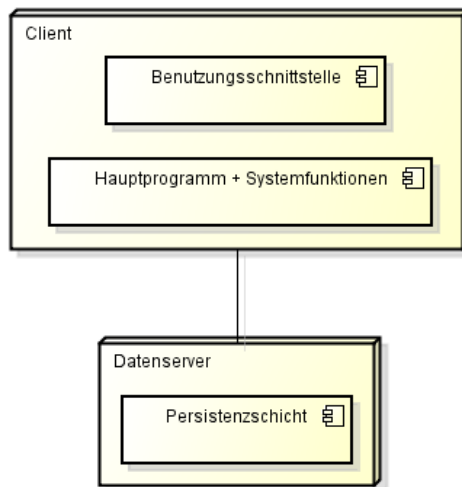
Es wird vorgeschlagen, die *Movie Manger* Anwendung mit einer 4-Schichten Referenzarchitektur aufzubauen, wobei die einzelnen Schichten auf verschiedene Rechner verteilt werden.



Für die Verteilung wird eine Three-Tier-Client-Server Architektur vorgeschlagen. Auf dem Client liegen die Benutzungsschnittstelle und das Hauptprogramm. Auf einem Funktionsserver liegen die Funktionselemente, die vom Hauptprogramm auf dem Client aufgerufen werden können („Call-Return“). Der Funktionsserver speichert die von den NutzerInnen eingegebenen Daten auf einem zusätzlichen Datenserver.

Architekturvorschlag 2:

Für die *Movie Manager* Anwendung wird eine 3-Schichten Referenzarchitektur vorgeschlagen, umgesetzt in einem Client-Server Modell.

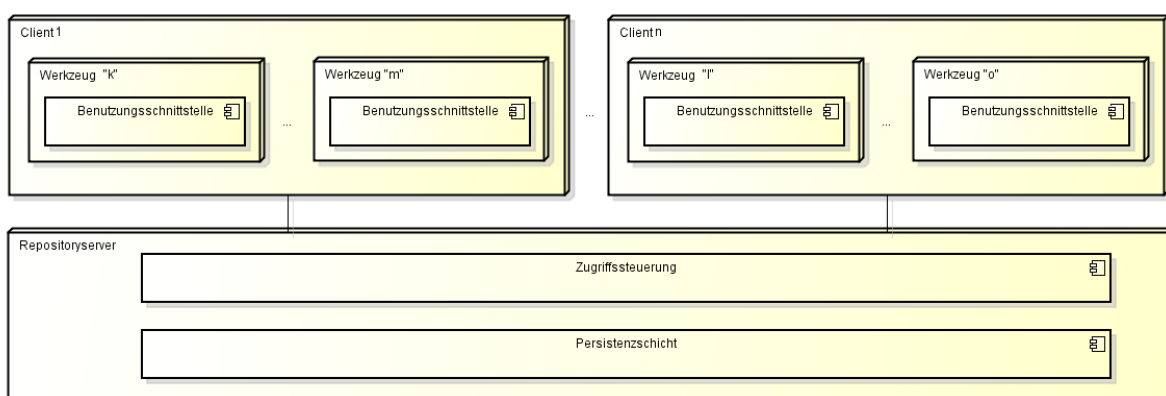


Die Verteilung der Schichten soll nach folgendem Muster geschehen: Auf dem Client soll die Benutzungsschnittstelle und die Funktionalität (Hauptprogramm + Systemfunktionen) liegen und auf dem Server die Daten, mit denen gearbeitet werden soll. Die Aufrufe der einzelnen Methoden auf dem Client erfolgen nach dem Call-Return-Prinzip.

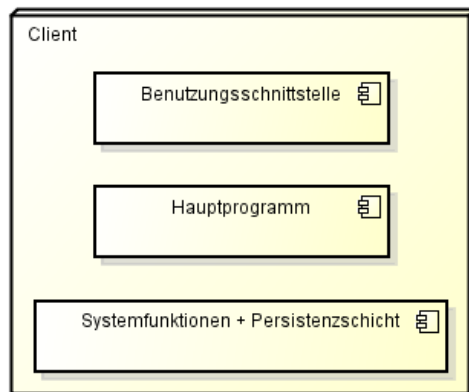
Architekturvorschlag 3:

Für die Architektur der *Movie Manager* Anwendung wird das Muster des Repository vorgeschlagen. Das Repository stellt die Daten und den Datenzugriff zur Verfügung und legt diese permanent ab. Die einzelnen Werkzeuge stellen das Hauptprogramm und die verschiedenen Systemfunktionen für die Erzeugung, Veränderung, Löschung der Daten, sowie die Benutzungsschnittstelle zur Verfügung. Die Aufrufe erfolgen dabei nach dem Master-Slave-Prinzip.

Die einzelnen Teile des Programms sind wie folgt verteilt: Auf einem Server liegt die Datenhaltung und die Zugriffssteuerung. Die Werkzeuge mit der Benutzungsschnittstelle liegen auf dem Client.



Architekturvorschlag 4:



Die *Movie Manager* Anwendung wird nach einer 3-Schichten Architektur aufgebaut, wobei alle 3 Schichten auf einer Maschine liegen und der Zugriff nur über den Computer erfolgen kann, auf dem auch das Programm läuft. Der Ablauf des Programms ist nach dem Call-Return-Prinzip organisiert. Eine Trennung von Daten und Funktionalität ist nicht vorgesehen. Aber bei Bedarf stellt das Programm eine Exportfunktion für die Daten zur Verfügung, sodass diese Daten auf verschiedenen Wegen transportiert und anschließend auf anderen Computer in dort laufende Programme eingebunden werden können.

Aufgabe 13.3: Vorschlag neuer Funktionalität – UNICASE

Präsenz: Nein

Punkte: 6

Team: Ja(4)

Projekt: Nein

Überlegen Sie sich, basierend auf Ihren Erfahrungen beim Arbeiten mit dem CASE-Tool UNICASE, um welche neue Funktionalität UNICASE für das nächste Jahr sinnvoll ergänzt werden sollte, bzw. was Sie an bestehenden Funktionen ändern würden. Beschreiben Sie sowohl neu vorgeschlagene Funktionalität als auch Änderungen an bestehender Funktionalität in textueller Form und begründen Sie diese. Gehen Sie dabei ein auf:

- die dafür benötigten Systemfunktionen
- die zum Umsetzen der Systemfunktionen notwendigen Daten
- die Änderungen an der Benutzungsschnittstelle.

Ergebnis:

Speichern Sie bitte Ihre Ergebnisse als PDF bis **Montag 25.01.2016 um 10.00 Uhr** in Moodle.