

# **SWE alternative Prüfungsleistung**

## **Beispiel: Haushaltsbuch-App**

### **1 Vorbereitung**

#### **Bilden Sie Gruppen mit 5 Mitgliedern!**

- a) Bestimmen Sie eine Kontaktperson/Teamleiter!
- b) Sie tragen die Gruppe mit Teamleiter und Teilnehmern in das Projekt Wiki ein.

#### **1.1 Ziel und Abgrenzung des Praktikums**

Das Praktikum hat zum Ziel, den Studierenden anhand eines fiktiven Szenarios einen Einblick in ein Softwareentwicklungsprojekt zu geben. Dabei steht die Arbeit mit dem UML - Standard sowie der Umgang mit aktuellen Entwicklungswerkzeugen im Vordergrund.

Das Praktikum ist aufgeteilt in Vorbereitung, Durchführung und Nachbereitung. Mit der Lösung der vorbereitenden Aufgaben erarbeiten Sie sich die Voraussetzungen für den Durchführungsteil, welcher in einem Computerpool an der HfTL gemeinsam bearbeitet wird. Als Software kommt das Programm Enterprise Architect von Sparx Systems zum Einsatz [2]. Zusammen mit Antworten auf die Fragen der Nachbereitung bilden die erfolgreiche Bearbeitung aller Teile die Prüfungsvorleistung.

#### **1.2 Voraussetzungen**

Grundlegende Kenntnisse des Konzeptes der Objektorientierung und einer objektorientierten Programmiersprache sind Voraussetzungen für die Teilnahme am Praktikum. Aus den Vorlesungen müssen die Begriffe Objekt, Klasse, Methode, Attribut, Akteur und Anwendungsfall bekannt sein.

#### **1.3 Literaturangaben**

[1] <http://www.omg.org/technology/documents/formal/uml.htm> - Unified Modeling Language, Superstructure, v2.2, Seite 603 ff.

[2] <http://www.sparxsystems.at/> - Enterprise Architect 7.5 UML Modeling Software

#### **1.3 Inhalt der Prüfungsleistung**

Jede Projektgruppe erstellt das Software System und *eine* Dokumentation. Folgen Sie bei der Erstellung der Dokumentation den „Richtlinien zur Anfertigung schriftlicher Arbeiten der HfTL“. Die Dokumentation beginnt mit einem Deckblatt, das die Autoren, das Thema, den Bearbeitungszeitraum und die Semestergemeinschaft nennt.

Von den 13 UML Diagrammen wurden 5 für die Dokumentation ausgewählt: Use-Case, Activity, Class, Sequence Diagram und State Machine. Diese 5 Diagramme sind mit angemessener Sorgfalt und Detaillierung zu entwickeln, siehe Durchführung.

Der Report aus dem Enterprise Architect kann für die Dokumentation genutzt werden, muss aber überarbeitet werden. Besonders leere Seiten und Tabellen müssen gelöscht werden.

Die Dokumentation soll auch die Auswertung, siehe 3. enthalten.

**Die Dokumentation wird in Ilias als pdf eingereicht. Das Software System wird als alternative Prüfungsleistung vorgeführt. Details zum Terminplan und der Bewertung der alternativen Prüfungsleistung entnehmen Sie der Datei apl.pdf.**

### **1.4 Szenariobeschreibung**

Als Mitarbeiter eines Softwarehauses haben Sie den Auftrag erhalten, eine Softwarelösung für einen **Haushaltsbuch-App** zu erarbeiten. Der Auftraggeber hat bereits Vorstellungen davon, was die Software unbedingt leisten soll. Leider sind diese Vorstellungen unvollständig.

Der Auftraggeber erwartet, dass Sie ihn beratend bei diesem Projekt unterstützen. Bevor der Entwicklungsauftrag endgültig vergeben wird, ist dem Auftraggeber ein verbindliches Angebot über den Funktionsumfang der Lösung zu unterbreiten. Gleichzeitig bildet dieses Angebot inkl. des Pflichtenheftes die rechtliche Grundlage für den Entwicklungsauftrag. Der Auftraggeber beschreibt die Funktionen der Software auszugsweise wie folgt:

„Die App soll in Echtzeit die monetären Ausgaben einer Person speichern, sowie ausgewertet wiedergeben. Hierfür soll es möglich sein:

- bei einem Einkauf Informationen über einen Artikel von einem Etikett einzulesen, beziehungsweise bei bestehender EAN Nummer aus einer Datenbank zu laden und aus diesen Argumenten einen Einkauf zu erstellen
- sonstige Kosten aufzunehmen, die nicht mit einem EAN Code in Verbindung gebracht werden können.
- von aktiven Einkäufen unabhängige, regelmäßige Kosten zu erfassen.
  1. Es soll möglich sein für jemand anderen oder eine Gruppe (z.B. WG) etwas einzukaufen.
  2. Die Daten werden Zentral in einer, über das Internet erreichbare, Datenbank gespeichert.
  3. Die App soll primär ein einfaches Front End bereitstellen, um Informationen zu sammeln und zu organisieren
  4. Eine Web Site ist momentan out of scope, wäre aber eine sinnvolle Ergänzung für die Ausgabe von Statistiken und die Benutzerverwaltung.
  5. Die Möglichkeiten der Auswertung sind vielfältig und können in Listen oder Diagrammen dargestellt werden.

Auswertungsbeispiele:

- a) Durchschnittliche Tageskosten eines Zeitraumes (z.B. Woche oder Monat)
- b) Maximal oder Minimalpreis innerhalb eines Zeitraumes (z.B. Woche oder Monat)
- c) Eine Grafik, die den Ausgabenverlauf innerhalb eines Zeitraumes angibt
- d) Eine Extrapolation regelmäßig gekaufter Artikel (Ersatz des „Einkaufzettels“)
- e) Das persönliche Tracking der allgemeinen Ausgaben
- f) weitere Möglichkeiten...

### 1.5 Vorbereitenden Aufgaben

1. Wer arbeitet mit dem Softwaresystem?
2. Welcher Benutzer benötigt welche Funktionen?
3. Welche Informationen müssen zu einer Person/Benutzer gespeichert werden, um einen Geschäftsprozess, z. B. das **für WG einkaufen**, mit dem System abzuwickeln?
4. Welche im Szenario nicht genannten Funktionen werden von dem Softwaresystem benötigt, um heutigen Anforderungen zu entsprechen? Nennen Sie beispielhaft fünf Funktionen!
5. Was ist ein Anwendungsfall und welche Beziehungen zwischen Anwendungsfällen beschreibt der Standard [1]?
6. Beschreiben Sie die Anwendungsfälle „**Einkauf einlesen**“ und „**Ausgabenverlauf anzeigen**“ nach dem folgenden Muster!

**use case** *Teilnehmer anmelden*

**actors**

*Liste der Akteure*

**precondition**

*Voraussetzungen*

**main flow**

*Beschreibung des Anwendungsfalls für den einfachsten bzw. normalen Verlauf*

**alternative flow** *Bezeichnung (Substantiv Verb)*

*Beschreibung des alternativen Ablaufs des Anwendungsfalls*

**postcondition**

*Ergebnis des Anwendungsfalls*

**exceptional flow** *Ausnahme 1*

*Beschreibung der Ausnahme 1*

**exceptional flow** *Ausnahme 2*

*Beschreibung der Ausnahme 2*

**postcondition**

*Ergebnis der Ausnahmesituationen*

**end** *Teilnehmer anmelden*

## **2 Durchführung**

### **2.1 Hinweis**

Starten Sie den PC mit Windows und öffnen Sie das Programm Enterprise Architect. Importieren (Add Key) Sie einen EA Academic Schlüssel vom Rechner \\10.1.100.236\Sparx! Legen Sie ein neues Projekt auf dem Netzlaufwerk Z:\ an, indem Sie einen aussagekräftigen Namen wählen.

### **2.2 Anwendungsfalldiagramm**

Erstellen Sie ein allgemeines Anwendungsfalldiagramm zum Softwaresystem!

Erstellen Sie ein Anwendungsfalldiagramm für eine konkrete Situation, z.B. „**Einkauf einlesen**“ und „**Ausgabenverlauf anzeigen**“.

### **2.3 Aktivitätsdiagramm**

Erstellen Sie ein Aktivitätsdiagramm für den konkreten Anwendungsfall von 2.2!

### **2.4 Klassendiagramm**

Erstellen Sie ein Entitäts-Klassendiagramm zum Szenario! Das Klassendiagramm enthält mindestens 5 Klassen, die in Beziehung zueinander stehen. Die Attribute und Methoden jeder Klassen werden definiert. Die Beziehungen sind vollständig definiert und enthalten die notwendigen Kardinalitäten.

### **2.5 Sequenzdiagramm**

Erstellen Sie ein Sequenzdiagramm für den konkreten Anwendungsfall von 2.2!

### **2.6 Zustandsdiagramm**

Erstellen Sie ein Zustandsdiagramm zur Entitätsklasse „**Einkaufszettel**“!

## **3 Auswertung**

1. Stellen Sie die Eigenschaften der Diagramme gegenüber! Gehen Sie dabei darauf ein, was in den Diagrammen dargestellt wird und welche Aspekte in Bezug auf die anderen Diagramme nicht ersichtlich sind! Nutzen Sie dazu eine Tabelle!

2. Welches Diagramm muss innerhalb des Designprozesses der Software konkretisiert werden?

3. Wie unterstützt der EA bei der Erstellung der UML Diagramme? Nennen Sie mindestens drei Beispiele, die Ihnen besonders aufgefallen sind!

*Wissenschaft ist die systematisch veranstaltete, professionell betriebene und arbeitsteilig organisierte Befriedigung von Neugier. Georg Franck*