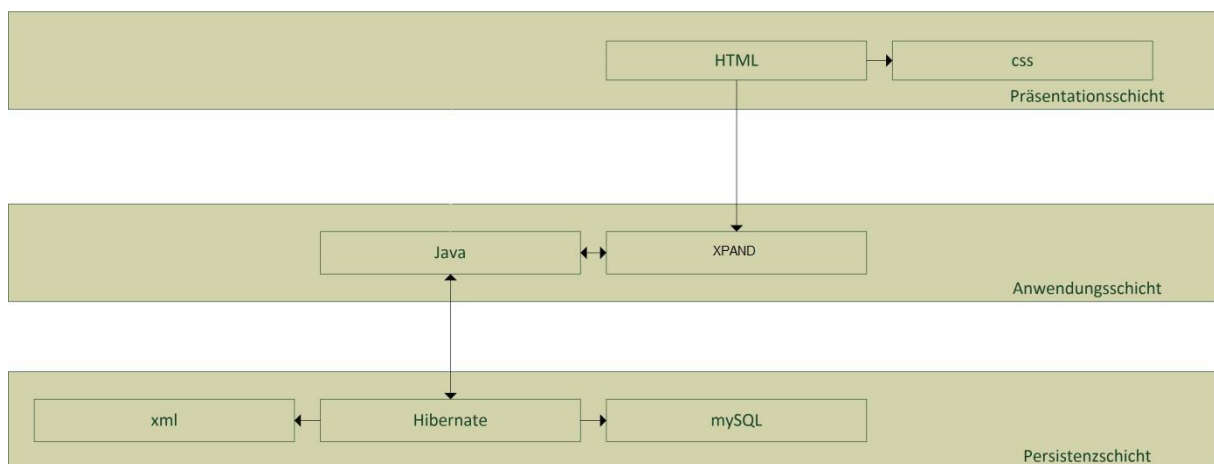


## Entwurfsbeschreibung der Softwarestudie

1. Allgemeines
2. Produktübersicht
  - 2.1 Codegenerator
  - 2.2 Datenbank
  - 2.3 Weboberfläche
3. Grundsätzliche Struktur- und Entwurfsprinzipien
4. Struktur- und Entwurfsprinzipien einzelner Pakete
  - 4.1 Der Generator
  - 4.2 MySQL
  - 4.3 Weboberfläche
5. Datenmodell
6. Testkonzept
7. Glossar
8. Anhang

## 1. Allgemeines

Dieser Prototyp stellt eine Softwareumgebung, mit welcher es möglich ist eine Datenbank zu erstellen. Diese Datenbank kann dann mittels einer einfach gehaltenen Weboberfläche verwaltet werden, das heißt, dass man über diese Oberfläche (Benutzerschnittstelle) Einträge in der Datenbank ausgeben und erstellen kann. Die Funktionen Löschen und Ändern werden bis zum Endrelease implementiert worden sein. Die Softwareumgebung verwendet UML2-Diagramme als Vorlage und erstellt dann mithilfe dieser Grundlage die oben genannte Benutzerschnittstelle und Datenbank. Zu beachten ist hierbei, dass diese beiden Produkte auf dem ausführenden Rechner erstellt und gespeichert werden, es sollte also ein Speicherplatz zur Verfügung stehen, der der Komplexität der Datenbank gerecht wird. In dieser Grundversion des Programmes wird außerdem ein Browser benötigt. Eine Internetverbindung ist nicht notwendig (dies würde sich aber bei einer Auslagerung der Datenbank unter Umständen ändern)



## 2. Produktübersicht

### 2.1 Codegenerator

Der Codegenerator soll aus einem UML-Modell, welches dem UML2-Standard entspricht und vollständig ist, dessen Klassen, Instanzen, Attribute und Relationen auslesen und daraus einen funktionierenden, kompilierbaren Java-Quellcode erstellen. Dieser Java-Quellcode wird in darauf folgenden Schritten zur Erstellung der funktionstüchtigen Web-Oberfläche und die Klassen, Instanzen, etc. zur Realisierung der Datenbank bzw. dessen Schemas benötigt. Dadurch erhält der Generator eine eminent wichtige Stellung im Gesamtkonzept und kann auch als Basis des Produktes gesehen werden.

### 2.2 Datenbank

Um alle, für die Anwendung nötigen Strukturen des UML-Diagramms dauerhaft zu speichern wird eine Datenbank erstellt. Als Datenbankschema wird bei dieser Anwendung MySQL verwendet, da dies einfach zu installieren ist und eine übersichtliche Anwendung ermöglicht.

Alle Klassen und Relationen werden mit Hilfe dieses Schemas erstellt und sorgen damit für die nötige Persistenz und Konsistenz. Damit werden auch mögliche Eingabefehler des Anwenders / der Anwender minimiert.

## 2.3 Weboberfläche

Auf der Basis des erstellten Quellcodes wird ebenfalls eine Webanwendung zur Verfügung gestellt.

Wie bereits oben beschrieben, ist diese Oberfläche in der Lage Einträge der Datenbank auszugeben und zu erstellen. Dabei wird aber die Stabilität der Datenbank nicht angegriffen.

## 3. Grundsätzliche Struktur- und Entwurfsprinzipien

Als Hauptanwendung für die Software wurde JAVA-Eclipse Juno verwendet. Um den Quellcode korrekt zu kompilieren wird in jedem Fall mindestens das JAVA-JDK1.6 verwendet. Dieses ist Standard und auf den meisten Rechnersystemen vorhanden. Möchte man den Java-Quellcode über Eclipse kompilieren, sollte Eclipse v3.5 oder höher verwendet werden um Probleme auszuschließen. Allerdings werden für die Anwendung keinerlei Programmierkenntnisse vorausgesetzt. Das Eclipse Plugin Xpand muss installiert sein, da dies den Grundbaustein für den Generator darstellt.

Zur Codegenerierung nutzen wir Xpand Templates, welche Java Dateien und andere Dateien (z.B. properties, .tml), die für Tapestry und Hibernate benötigt werden, generieren.

Für die Datenbank wird Hibernate mit JDBC benutzt, das heißt es werden in die erstellten Java-Klassen JDBC-Befehle eingebunden. In der Datenbank können Tabellen und records erstellt und gelöscht werden.

Des Weiteren wird es einen Server (mit MySQL) geben, welcher die Datenkommunikation zwischen der Datenbank und Website leitet. Der Datenbankserver läuft auf der Homepage.

Für die Datenkommunikation werden außerdem noch Apache tomcat und Tapestry benutzt.

Die Weboberfläche wird mittels Tapestry erstellt und pflegt nun die Daten. Die Oberfläche an sich ist sehr einfach gehalten. Sie kann aber natürlich durch größere UML-Modelle auch komplexer erscheinen. Über eine Verwendung von php wird nachgedacht.

Grundsätzlich sind Templates für die Webanwendung zu entwickeln, d.h. dynamische Dateien mit Platzhaltern für Klassennamen oder Ähnlichem.

Die Webanwendung selbst wird in einer 3-Schichten-Architektur umgesetzt, gegliedert in Präsentations-, Anwendungs- und Persistenz Schicht. Dabei wird für die Präsentationsschicht auf php, html und das Tapestry-Framework zurückgegriffen. Durch Stylesheets (css) können evtl. noch Änderungen am äußeren Erscheinungsbild spezifiziert werden (optionales Kriterium). Die Hauptfunktionalität wird in Java realisiert und eingebunden. Die persistente Speicherung der Daten wird über das JDBC realisiert. Dabei werden Mapping-Dateien im xml Format und ein Datenbankschema zum objektrelationalen Mapping (ORM) benötigt.

## 4. Struktur- und Entwurfsprinzipien einzelner Pakete

### 4.1 Der Generator

Mitgeliefert in unserem Programm ist eine Eclipse Version in der Topcased vorinstalliert ist. Topcased ist eine Erweiterung um sich UML-Dateien ansehen/erstellen/bearbeiten zu können.

Eine JAVA-Klasse startet 2 View Klassen, die Eine ist eine Begrüßungsseite und die Andere leitet den Benutzer durch die Installation. Es ist sehr einfach und kleinschrittig gehalten, um eine größere Benutzerfreundlichkeit zu gewährleisten.

Es werden mit Xpand workflow-Dateien erstellt, welche dann aufgeführt werden.

### 4.2 MySQL

Mithilfe von Xpand wird eine XML-Startdatei für Hibernate erstellt und alle Klassen werden mit den nötigen Kommentaren für Hibernate versehen.

Der MySQL Server läuft über die Homepage. Auf der die Datenbank gespeichert wird. Die Dateien von Hibernate liegen der Softwarestudie ebenfalls bei und sind einsehbar.

### 4.3 Weboberfläche

Ansicht:

Die Präsentationsschicht wird mittels HTML/AJAX und JSF-Dateien realisiert. Die Website wird seitenorientiert gehalten. Dabei werden zur Modellierung der einzelnen Seiten und deren Funktionalität vor allem Zustandsdiagramme benutzt. Deren Zustände stellen grundsätzlich Seiten und Zustandsübergänge zur realisierenden Funktionalität dar. Enthalten sind eine Menüleiste mit den Attributen: "Erstellen", "Ansicht" und "Bearbeiten/löschen" und ein Fenster, in dem die Auswahl des Menüs dargestellt wird.

Funktionen:

Die Struktur der Seite wird durch einen mitgegebenen AJAX/Javascript/HTML-Quellcode dynamisch erstellt, welcher nur folgendes beinhaltet:

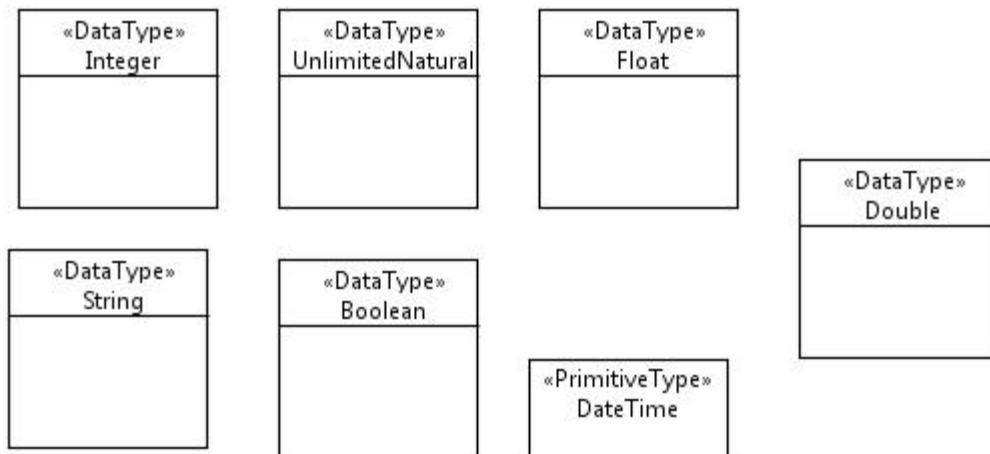
- Erstellung eines leeren Fensters
- Anlegen des Menüs

Das Anlegen des Fensters erfolgt der Einfachheit halber durch ein <div> via html. Das Menü wird durch JavaScript unterstützt und setzt Referenzen auf das zentrale <div>, in dem später die abgerufenen Informationen erscheinen sollen, und mittels AJAX wird eine Funktion erstellt, welche es ermöglicht neue html-Seiten in dem Fenster aufzurufen und damit eine gewisse Objektorientierung zu erreichen und leichtere Bearbeitbarkeit der einzelnen Menüleisten, da diese nun separat erstellt werden. Dies besitzt ebenfalls den Vorteil, dass das Laden der Seite und Abrufen von der Datenbank kürzer/kleiner und damit schneller sind.

Umsetzung:

Die Umsetzung erfolgt über PHP mit Zugriff auf die Datenbank. Dabei werden für jedes Menüfenster alle erforderlichen Attribute und Relationen mittels SQL-Befehlen abgefragt und dann mittels HTML/Javascript die notwendigen Buttons und Textfelder erstellt und mit php deren Funktionen festgelegt. Dies geht von einfachen Ansichtsfenstern ("Labels"), über Textfelder("TextBox") bis hin zu Buttons ("Input").

## 5. Datenmodell



Die Datentypen müssen im UML-Diagramm angegeben werden. Die Daten können sich von UML zu UML unterscheiden. In der Webanwendung müssen die Daten ihren Typen entsprechend eingegeben werden.

Die Klassennamen werden von einer Funktion abgespeichert in einem Array.

Dokumente werden berücksichtigt, das heißt sie können hoch- und heruntergeladen werden. Enumerationen werden eingehalten.

## 6. Testkonzept

Das Testkonzept wird in die drei Teilbereiche: Generator, Homepage und Datenbank gegliedert werden.

Zum Testen des Generators werden UML-Modelle angefertigt, die mit steigender Komplexität und verschiedenen Kombinationen der Klassenbeziehungen die Richtigkeit der Übersetzung prüfen sollen.

Daran anknüpfend wird auch die Datenbank mit steigender Schwierigkeit geprüft. Zunächst sind Testeinträge in der Datenbank vorgesehen, welche über das CMD geschehen sollen. Des Weiteren wird es Java Klassen geben, die Methoden unserer erstellten Klassen aufruft und untersucht ob der Datenbankzugriff störungsfrei ablaufen kann.

Zuletzt wird die Website dahingehend geprüft, ob auch Dateien hoch- und heruntergeladen werden können.

### 7. Glossar

<a href="#">Apache Tapestry</a>	Webframework für die Programmiersprache Java
Xpand	Xpand ist eine <a href="#">statisch typisierte Templatesprache</a> mit speziellen, für die Codegenerierung wichtigen Features.
TOPCASED	(Toolkit in Open Source for Critical Applications & Systems Development; deutsche Quelloffene Werkzeugsammlung für die Entwicklung kritischer Anwendungen und Systeme) ist ein <a href="#">CAE-Werkzeug</a> .
Hibernate	<a href="#">englisch</a> für Winterschlaf halten) ist ein <a href="#">Open-Source-Persistenz-</a> und <a href="#">ORM-Framework</a> für <a href="#">Java</a> .
PHP	ist eine <a href="#">Skriptsprache</a> mit einer an <a href="#">C</a> und <a href="#">Perl</a> angelehnten <a href="#">Syntax</a> , die hauptsächlich zur Erstellung <a href="#">dynamischer Webseiten</a> oder <a href="#">Webanwendungen</a> verwendet wird.
<a href="#">Eclipse (IDE)</a>	Software-Plattform und Integrierte Entwicklungsumgebung
Unified Modeling Language	(Vereinheitlichte Modellierungssprache), kurz UML, ist eine graphische <a href="#">Modellierungssprache</a> zur Spezifikation, Konstruktion und Dokumentation von Software-Teilen und anderen Systemen

### 8. Anhang

