**Entwicklung standortbezogener**

**Funktionen für eine Webanwendung**

**Individuelle Praktische Arbeit (IPA)**

**Severin Peter**

**Hauptexperte**

**Alexandre Moeri**

**Fachexperte**

**Max Meisterhans**

**ZHAW Institut für Wirtschaftsinformatik, Winterthur**

**19.Mai 2017**

Inhaltsverzeichnis

[1. Umfeld und Ablauf 4](#_Toc451105624)

[1.1 Aufgabenstellung 4](#_Toc451105625)

[1.1.1 Ausgangslage 4](#_Toc451105626)

[1.1.2 Ziele 4](#_Toc451105627)

[1.1.3 Anwendungsfälle 4](#_Toc451105628)

[1.1.4 Systemschnittstellen 5](#_Toc451105629)

[1.1.5 Nicht-funktionale Anforderungen 6](#_Toc451105630)

[1.1.6 Rahmenbedingungen 6](#_Toc451105631)

[1.1.7 Artefakte 6](#_Toc451105632)

[1.2 Projektorganisation 7](#_Toc451105633)

[1.3 Vorkenntnisse 8](#_Toc451105634)

[1.4 Vorarbeiten 9](#_Toc451105635)

[1.5 Zeitplan 10](#_Toc451105636)

[1.6 Arbeitsjournal 11](#_Toc451105637)

[2. Projekt 21](#_Toc451105638)

[2.1 Kurzfassung 21](#_Toc451105639)

[2.1.1 Ausgangslage 21](#_Toc451105640)

[2.1.2 Umsetzung 21](#_Toc451105641)

[2.1.3 Ergebnis 21](#_Toc451105642)

[2.2 Projektmanagement 22](#_Toc451105643)

[2.3 Aufgabenstellung 23](#_Toc451105644)

[2.3.1 Verfeinerung der Aufgabenstellung 23](#_Toc451105645)

[2.4 Konzept 25](#_Toc451105646)

[2.4.1 Use-Case Diagramm 25](#_Toc451105647)

[2.4.2 Evaluation 25](#_Toc451105648)

[2.4.3 Coding Style 29](#_Toc451105649)

[2.4.4 Gegebene XML Strukturen 29](#_Toc451105650)

[2.4.5 Programm Ablauf 34](#_Toc451105651)

[2.5 Realisierung 36](#_Toc451105652)

[2.5.1 Versionsverwaltung 36](#_Toc451105653)

[2.5.2 Entwicklungsumgebung Einrichten 36](#_Toc451105654)

[2.5.3 Programm Gerüst 36](#_Toc451105655)

[2.5.4 Main 37](#_Toc451105656)

[2.5.5 Model 37](#_Toc451105657)

[2.5.6 GUI 37](#_Toc451105658)

[2.5.7 Controller 37](#_Toc451105659)

[2.5.8 XMLReader 38](#_Toc451105660)

[2.5.9 XMLWriter 38](#_Toc451105661)

[2.5.10 Pom.xml 38](#_Toc451105662)

[2.6 Testen 39](#_Toc451105663)

[2.6.1 Testkonzept 39](#_Toc451105664)

[2.6.2 Blackbox Tests 39](#_Toc451105665)

[2.6.3 Testprotokoll 42](#_Toc451105666)

[2.7 Ergebnis 45](#_Toc451105667)

[2.7.1 Konvertieren mit dem GUI 45](#_Toc451105668)

[2.7.2 Konvertieren mit der Konsole 47](#_Toc451105669)

[2.8 Schlusswort 48](#_Toc451105670)

[2.9 Glossar 49](#_Toc451105671)

[2.10 Quellen 50](#_Toc451105672)

[3. Anhang 1 51](#_Toc451105673)

[4. Anhang 2 52](#_Toc451105674)

[4.1 Benutzeranleitung 52](#_Toc451105675)

[4.1.1 Einleitung 53](#_Toc451105676)

[4.1.2 Konvertieren über das GUI 53](#_Toc451105677)

[4.1.3 Konvertieren über die Konsole 55](#_Toc451105678)

[4.1.4 Benutzer Nachrichten 55](#_Toc451105679)

[5. Anhang 3 58](#_Toc451105680)

[5.1 Protokolle 58](#_Toc451105681)

# Umfeld und Ablauf

## Aufgabenstellung

### Ausgangslage

Im Verlaufe der letzten Monate habe Ich einen Prototyp für eine Webapplikation erstellt, welcher es einem Benutzer erlaubt eine sich einen Artikel aus einem beliebigen Laden zu bestellen. Dazu gibt er den Artikel, den Laden, den Preis, den Preis den er willig ist zu bezahlen und seine Adresse an. Danach Wird der Auftrag auf der Website angezeigt, sodass ihn alle anderen Benutzer ansehen können. Wenn man einen Auftrag sieht, der einem gefällt, wählt man diesen aus und bestätigt, dass man diesen Auftrag annimmt.

### Ziele

Das Ziel dieser Arbeit ist die Integrierung einer Map welche die Location des Auftraggebers und des Ladens anzeigt, wodurch man den Weg ermitteln kann den man als Dienstleister zurücklegen müsste, es werden auch die Wegkosten Berechnet. Ausserdem soll man die gelisteten Aufträge nach Distanz sortieren können.

## Anwendungsfälle

### **Erstellen**

Der Anwendungsfall „Auftrag Erstellen“ lauft wie folgt ab:

1. Der Anwender kommt auf die Webseite.
2. Der Auftragsgeber wählt im GUI den Button „Neuer Auftrag“.
3. Auf der Erscheinenden Maske gibt er die notwendigen Daten ein.
4. Er sendet den Auftrag ab.
5. Der Auftrag wird von einem Dienstleister erfüllt.

### **Annehmen**

Der Anwendungsfall „Auftrag Annehmen“ lauft wie folgt ab:

1. Der Anwender kommt auf die Webseite.
2. Er sucht sich einen passenden Auftrag aus.
3. Nach anschauen der genauen Details des Auftrags gibt er an, dass er den Auftrag annimmt.
4. Er erfüllt den Auftrag

## Projektorganisation

|  |  |
| --- | --- |
| Name | Peter Severin |
| Funktion | IPA Kandidat/ Projektleiter |
| Telefon | 076 428 97 79 |
| Mail | sevi.peter@gmail.com |
| Geschäftsadresse | ZHAW |
|  | Stadthausstrasse 14 |
|  | 8400, Winterthur |

|  |  |
| --- | --- |
| Name | Meisterhans Max |
| Funktion | Fachvorgesetzter/ Auftragsgeber |
| Telefon | 058 934 66 75 |
| Mail | meix@zhaw.ch |
| Geschäftsadresse | ZHAW |
|  | Stadthausstrasse 14 |
|  | 8400, Winterthur |

|  |  |
| --- | --- |
| Name | Moeri Alexandre |
| Funktion | Experte |
| Telefon | 043 500 39 80 |
| Adresse | Grünaustrasse 9 |
|  | 8064, Zürich |

## Vorkenntnisse

* Webentwicklung mit Angular 2
* Versionsverwaltung mit Git(GitHub)

## Vorarbeit

* Entwicklung Prototyp mit Angular 2

## C:\Users\petv\Desktop\Unbenannt.JPG Zeitplan

## Arbeitsjournal

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Montag, 08.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | | | | | | | | | | | | | | | | |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | Erstellen Zeitplan | | | | | | ✓ | | Layout brauchte Anpassung, damit es im Word gut angezeigt werden kann. | | | | | | | |
| 2 | Definieren Vorkenntnisse | | | | | | ✓ | |  | | | | | | | |
| 3 | Schreiben Kurzfassung | | | | | | ✓ | |  | | | | | | | |
| 4 | Beginn mit Verfeinerung der Aufgabenstellung | | | | | | 🗶 | | Sollte am Tag 2 erledigt sein | | | | | | | |
| Aufgetretene Probleme | | | | | Layout Probleme mit dem Zeitplan | | | | | | | | | | | |
| Wissensbeschaffung | | | | | <https://www.pkorg.ch/> (IPA Detaildaten Eugster Alex) | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Im Zeitplan. Etwa 10 min zurück | | | | | | | | | | | |
| Reflexion | | | | | Anfang der IPA: Da ich seit längerem nicht mehr so ausführlich dokumentieren musste und ich mich nicht gewohnt bin, mit einem Zeitplan zu arbeiten, bin ich nicht so schnell vorangekommen, wie ich angenommen hatte. Ich habe momentan zwar viel Text, bin mir aber bewusst, dass dieser noch einiges an Überarbeitung braucht, damit ich mit diesem zufrieden bin.  Aufgrund eines Kickoff Meetings heute Morgen bin ich dem Zeitplan zusätzlich noch etwas im Hintertreffen, hoffe aber, dass ich dies im Laufe der Woche wieder gut machen kann. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 7:00 h | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | - | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dienstag, 09.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | Fertigstellen Aufgabenstellung und Verfeinerung | | | | | | ✓ | |  | | | | | | | |
| 2 | Zeichnen Anwendungsfalldiagramm | | | | | | ✓ | |  | | | | | | | |
| 3 | Zeichnen Aktivitätsdiagramm | | | | | | ✓ | |  | | | | | | | |
| 4 | Zeichnen Klassendiagramm | | | | | | ✓ | |  | | | | | | | |
| 5 | Zeichnen GUI Design | | | | | | ✓ | |  | | | | | | | |
| 6 | Evaluation | | | | | | ✓ | |  | | | | | | | |
| Aufgetretene Probleme | | | | | - | | | | | | | | | | | |
| Wissensbeschaffung | | | | | <https://de.wikipedia.org/wiki/Anwendungsfalldiagramm>  <https://de.wikipedia.org/wiki/Klassendiagramm>  <https://de.wikipedia.org/wiki/Aktivit%C3%A4tsdiagramm> | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Aufgeholt | | | | | | | | | | | |
| Reflexion | | | | | Wie erwartet, hatte ich verhältnismässig lange am Zeichnen der verschiedenen Diagramme. Im Betrieb hatten wir wohl mit Diagramen gearbeitet, trotzdem fehlte mir die entsprechende Übung, um solche effizient zeichnen zu können. Ich musste nochmals einige Beispiele im Netz anschauen, um zu überprüfen, ob alle meine Pfeile in die korrekte Richtung zeigen und dass auch alle übrigen Konventionen eingehalten wurden. Da ich das GUI Design mit Photoshop erstellen wollte, musste ich dies von zuhause aus machen. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 7:40 h | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | 0:30 h | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mittwoch, 10.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | | | | | | | | | | | | | | | | |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | Erstellen Testkonzept | | | | | | ✓ | |  | | | | | | | |
| 2 | Expertenbesuch | | | | | | ✓ | |  | | | | | | | |
| 3 | Erstellen Blackbox Tests | | | | | | ✓ | |  | | | | | | | |
| 4 | Erstellen Grundgerüst Programm | | | | | | ✓ | |  | | | | | | | |
| 5 | Überarbeitung von Dokumentation | | | | | | ✓ | |  | | | | | | | |
| 6 | Beginn Programmieren | | | | | | 🗶 | |  | | | | | | | |
| Aufgetretene Probleme | | | | | In Absprache mit meinem Fachexperten betreffend mein Design vom GUI, habe ich ein paar Änderungen vorgenommen, die die Funktionalitäten des Programms betreffen. Da ich bereits mit den Blackbox Tests begonnen hatte, musste ich auch diese nochmals überarbeiten. | | | | | | | | | | | |
| Wissensbeschaffung | | | | | https://de.wikipedia.org/wiki/Black-Box-Test | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Etwa 0:30h im Rückstand | | | | | | | | | | | |
| Reflexion | | | | | Trotz der Mehrarbeit infolge der vorgenommenen Anpassungen, war die heutige Arbeit effizient und lieferte mehrere Dokumentationsseiten. Ich musste einige grössere funktionale Änderungen durchführen und bin so zeitlich etwas in Rückstand geraten. Obwohl ich ein paar Diagramme und teils auch an Dokumentation kleine Änderungen vornehmen musste, bin ich mit der IPA jetzt aber wieder auf Vordermann, sodass ich am Tag 4 mit dem Schreiben von Code beginnen kann. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 8:00 h | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | - | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Freitag, 11.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | | | | | | | | | | | | | | | | |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | Programmieren zum Einlesen von XML | | | | | | ✓ | |  | | | | | | | |
| 2 | Einbinden von einer Property Datei | | | | | | ✓ | |  | | | | | | | |
| 3 | Eingelesene Daten säubern | | | | | | ✓ | |  | | | | | | | |
| Aufgetretene Probleme | | | | | - | | | | | | | | | | | |
| Wissensbeschaffung | | | | | Stack Overflow, Dokumentation JDOM | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Etwa 1:00h im Voraus | | | | | | | | | | | |
| Reflexion | | | | | Es stellte sich heraus, dass ich für die Programmierungsteile der IPA zu viel Zeit eingeplant hatte. Probleme, mit welchen ich ursprünglich gerechnet hatte, sind teilweise nicht aufgetreten oder konnten viel schneller als erwartet gelöst und behoben werden. Beispielsweise hatte ich beim Einbinden der Properties Datei mit Schwierigkeiten gerechnet, da ich dies vorher noch nie gemacht hatte. Erfreulicherweise konnte ich bereits nach der Hälfte der ursprünglich eingeplanten Zeit, mit der nächsten Aufgabe weiterfahren. Dies ermöglichte es mir, nicht nur den Rückstand aufzuholen, welchen ich gegenüber dem Zeitplan hatte, sondern sogar schon mit der Arbeit für Tag 5 beginnen zu können. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 8:00 | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | - | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |
|  | |  |  | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Montag, 12.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | | | | | | | | | | | | | | | | |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | XML schreiben | | | | | | ✓ | |  | | | | | | | |
| 2 | Base64Converter | | | | | | 🗶 | |  | | | | | | | |
| Aufgetretene Probleme | | | | | - | | | | | | | | | | | |
| Wissensbeschaffung | | | | | <http://stackoverflow.com/questions/23979842/convert-base64-string-to-image> | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Etwa 0:30h voraus | | | | | | | | | | | |
| Reflexion | | | | | Das Programmieren verlief erfreulich gut und mit nur wenigen Problemen. Ich kam so gut voran, dass ich bereits mit den Arbeiten von Tag 6 beginnen konnte. Die gewonnene Zeit investierte ich hauptsächlich in das Überarbeiten der Dokumentation, mit welcher ich meines Erachtens noch etwas in Verzug bin. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 8:15 | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | - | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dienstag, 15.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | | | | | | | | | | | | | | | | |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | Fertigstellen Base64Converter | | | | | | ✓ | |  | | | | | | | |
| 2 | Error Handling | | | | | | 🗶 | | Das Error-Handling muss nach dem Fertigstellen des GUIs nochmals aufgegriffen und abgeschlossen werden. | | | | | | | |
| 3 | Kreieren Grundlegendes GUI | | | | | | ✓ | |  | | | | | | | |
| 4 | Überarbeiten Diagramme | | | | | | ✓ | |  | | | | | | | |
| Aufgetretene Probleme | | | | | Durch einen Fehler in der Planung konnte das für heute geplante Error Handling noch nicht ganz abgeschlossen werden. Dies kann erst nach dem Programmieren des GUI vollständig durchgeführt werden, weshalb ich diesen Punkt erst morgen abschliessen kann. | | | | | | | | | | | |
| Wissensbeschaffung | | | | | Buch Design-Pattern und Buch UML2.0 Zertifizierung.  Stack Overflow | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Etwa 1:30h im Voraus | | | | | | | | | | | |
| Reflexion | | | | | Erneut ging die Arbeit erfreulich und gut voran. Wie die Tage zuvor hatte ich nur wenige Probleme mit dem Programmieren und konnte die so gewonnene Zeit in die Überarbeitung der Dokumentation investieren. Insbesondere nutzte ich die Zeit, die Diagramme in der Dokumentation zu überarbeiten, da ich diese beim ersten Versuch etwas übereifrig und nicht ganz zufriedenstellend gezeichnet hatte. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 8:10 h | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | - | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mittwoch, 16.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | | | | | | | | | | | | | | | | |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | 2. Besuch vom Experten | | | | | | ✓ | |  | | | | | | | |
| 2 | Funktionalitäten GUI implementieren | | | | | | ✓ | |  | | | | | | | |
| 3 | Abschliessen Error Handling | | | | | | ✓ | |  | | | | | | | |
| Aufgetretene Probleme | | | | | - | | | | | | | | | | | |
| Wissensbeschaffung | | | | | Stack Overflow | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Etwa 1:00h im Voraus. | | | | | | | | | | | |
| Reflexion | | | | | Der zweite Besuch des Experten verlief positiv. Kleinere Fragen, welche zwischen den zwei Gesprächen aufgekommen waren, konnte ich klären.  Nach dem erfolgreichen Implementieren des GUIs konnte ich auch endlich den Rest des Error Handlings implementieren. Das Error Handling hat mir auch gezeigt, welche Methoden ich zusätzlich implementieren sollte, um das Programm noch einfacher und benutzerfreundlicher zu gestalten. Ich habe zusätzlich begonnen, die Applikation konsolenfähig zu machen. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 8:10 | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | - | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Donnerstag, 17.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | | | | | | | | | | | | | | | | |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | Applikation konsolenfähig machen | | | | | | ✓ | |  | | | | | | | |
| 2 | Durchführung Blackbox Tests | | | | | | ✓ | |  | | | | | | | |
| 3 | Refactoring | | | | | | ✓ | | 1. grosser Refactoring Block | | | | | | | |
| Aufgetretene Probleme | | | | | Ich hatte Probleme, die Applikation konsolenfähig zu machen. Da ich mit Maven noch wenig Erfahrung habe und das Tool für das Erstellen der JAR-Datei zuständig ist, hatte ich bei der Umsetzung einiges länger als erwartet. | | | | | | | | | | | |
| Wissensbeschaffung | | | | | Stack Overflow | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Ziemlich genau im Zeitplan | | | | | | | | | | | |
| Reflexion | | | | | Es stellte sich heraus, dass es nicht so einfach war, die Applikation konsolenfähig zu machen. Zusätzlich musste ich noch einen grösseren Refactoring Block einbinden, da der Code durch die verschiedenen Implementationen etwas undurchsichtig geworden war. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 8:10 | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | - | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Freitag, 18.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | | | | | | | | | | | | | | | | |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | Kleine Nachbesserungen | | | | | | ✓ | |  | | | | | | | |
| 2 | Schreiben Benutzeranleitung | | | | | | ✓ | |  | | | | | | | |
| 3 | Schreiben Ergebnis | | | | | | ✓ | |  | | | | | | | |
| Aufgetretene Probleme | | | | | - | | | | | | | | | | | |
| Wissensbeschaffung | | | | | - | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Im Zeitplan 1:00h voraus. Ich befürchte aber, dass ich die Zeit für die Überarbeitung der Dokumentation zu knapp eingeplant habe. | | | | | | | | | | | |
| Reflexion | | | | | Bei den Nachbesserungen, welche ich noch einbinden musste, handelte es sich um kleinere Flüchtigkeitsfehler, die sich eingeschlichen hatten. So waren beispielsweise gewisse Ausgaben auf Englisch, auch Grossschreibefehler musste ich korrigieren. Die Benutzeranleitung bereitete mir wenige Probleme, auch das Schreiben des Ergebnisses lief problemlos. Ich denke aber, dass die Dokumentation noch einiges an Überarbeitung braucht. Dies hatte ich bei der Planung offensichtlich etwas unterschätzt. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 8:05 | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | - | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dienstag, 19.05.2017** | | | | | | | | | | | | | | | | |
|  | | | | IPA-Tag | | 1 | 2 | 3 | | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | | | | | | | | | | | | | | | | |
| Nr. | Ausgeführte Arbeiten | | | | | | Status | | Bemerkungen | | | | | | | |
| 1 | In Ordnung bringen der Dokumentation | | | | | | ✓ | |  | | | | | | | |
| 2 | Hochladen der Dokumentation auf PKORG | | | | | | ✓ | |  | | | | | | | |
| Aufgetretene Probleme | | | | | - | | | | | | | | | | | |
| Wissensbeschaffung | | | | | - | | | | | | | | | | | |
| Vergleich mit dem Zeitplan | | | | | Im Zeitplan. | | | | | | | | | | | |
| Reflexion | | | | | Am letzten Tag befasste ich mich insbesondere mit dem Überarbeiten der Dokumentation. Um sicher zu stellen, dass ich das Projekt rechtzeitig abschliessen und abgeben kann, habe ich am Wochenende ca. 3:00 Stunden an der Dokumentation weiter gearbeitet und diese im Detail angeschaut, durchgelesen und umformuliert sowie auf Grammatikfehler hin überprüft. | | | | | | | | | | | |
| Arbeitsdauer | | | | | 5:00h | | | | | | | | | | | |
| Beanspruchte Hilfeleistung | | | | | - | | | | | | | | | | | |
| Nacht- Wochenendarbeit | | | | | 3:00h | | | | | | | | | | | |
| ✓ | | = | Erfolgreich | | | | | | | | | | | | | |
| 🗶 | | = | Nicht beendet, fehlgeschlagen | | | | | | | | | | | | | |

# Projekt

## Kurzfassung

### Ausgangslage

In dieser Diplomarbeit „XML-Dateikonverter für HSRvote“ wird eine Java Applikation entwickelt mit dem primären Ziel, Fragebogen, welche im online Tool Moodle erstellt wurden, in ein internes Programm namens HSRvote einzulesen zu können. Es gilt die Problemstellung zu lösen, dass der von Moodle erstellte XML Output nicht kompatibel ist mit dem XML Input, welches HSRvote einliest. Es muss also ein Dateikonverter erstellt werden.

### Umsetzung

Bei der Arbeit soll die Java Library DOM (JDOM) verwendet werden, um XML in die Applikation einzulesen, beziehungsweise wieder auszuschreiben. Ebenfalls soll die Applikation umfassend getestet sein. Dies umfasst sowohl Blackbox- als auch JUnit Tests.

Die Applikation beinhaltet ein kleines GUI, welches mit dem Netbeans GUI-Builder erstellt wurde. Es erleichtert dem User den Umgang mit der Applikation.

Im HSRvote ist es möglich, Bilder zu speichern. Diese werden jedoch im Moodle als Base64 Zeichenkette ausgegeben. Es muss also neben dem Konvertieren vom XML, auch noch ein Bild erstellt werden können.

### Ergebnis

Eine voll funktionstüchtige Java Applikation, welche alle beschriebenen Anforderungen erfüllt. Es soll möglich sein, die Applikation sowohl als Konsolenapplikation, als auch als JAR Datei auszuführen.

## Projektmanagement

Die Planung für die IPA ist sehr konkret und IPERKA lässt keinen Spielraum für nachträgliche Anpassungen. Diese Voraussetzungen sind für diese Applikation unvorteilhaft, da kleinere Änderungen im Konzept nicht auszuschliessen sind. Aus diesem Grund wird das Projektmanagement in einem mini-Projektmanagement (mPM) durchgeführt.

Geplant sind zehn Arbeitstage an je acht Stunden. Es ist davon auszugehen, dass etwa 40%, also rund 32 Stunden, in die Dokumentation der Arbeit gehen und rund 60%, sprich 48 Stunden, in die tatsächliche Realisierung des Projektes laufen werden. Die Planungsphase beginnt mit Tag 1 mit dem Erstellen des Zeitplans. In dieser Phase sollen Unklarheiten und Probleme erkannt und falls nötig mit dem Fachvorgesetzten abgesprochen werden. Zusätzlich sollen Lösungsmöglichkeiten evaluiert und realisiert werden.

Es sollen aus der Planung vier klare Meilensteine hervorgehen: Das Ende der Planung, das Erfüllen der grundsätzlichen Funktionen des Programms, das Ende des Refactoring und Testing Prozesses, sowie die Fertigstellung der Dokumentation und somit des Projekts.

Bevor die Realisierungsphase beginnt, müssen Diagramme und GUI-Entwürfe fertig gezeichnet sein. Sobald ein Testprotokoll erfasst worden ist, kann damit begonnen werden, das Programm zu entwickeln und zu testen. Während der Realisierung wird laufend dokumentiert. Die Realisierungsphase wird mit GIT überwacht.

## Aufgabenstellung

### Verfeinerung der Aufgabenstellung

Die Kernaufgabe dieser Diplomarbeit ist die Entwicklung eines Dateikonverters, welcher es ermöglicht, in Moodle erstellte Fragebögen in HSRvote einzulesen. Der Dateikonverter wird als fertig angesehen, wenn folgende Kriterien erfüllt sind:

1. Konvertierung

* Es ist möglich, eine XML Datei von der Moodle Plattform in eine XML Datei umzuwandeln, welche HSRvote einlesen kann.
* Die Konvertierung filtert die HTML Tags, welche von Moodle mitgegeben werden.
* Es gibt Werte, welche das HSRvote benötigt, aber nicht aus der XML Datei von Moodle herausgelesen werden können. Diese Werte werden in einer Property Datei gespeichert.
* Die Konvertierung kann auch über die Konsole direkt ausgeführt werden.

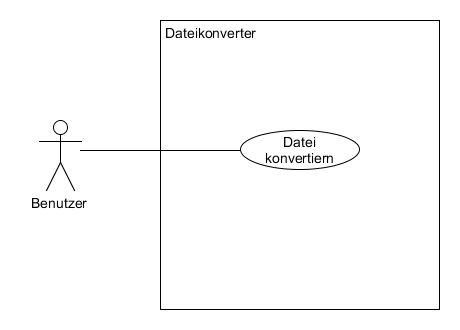
1. Das GUI

* Der Dateikonverter hat ein GUI.
* Im GUI kann eine Moodle XML Datei zum Umwandeln und ein Name für die Ausgabedatei angegeben werden.
* Das GUI kann jederzeit geschlossen werden.
* Ist eine Moodle XML Datei erfolgreich konvertiert worden, wird dies vom GUI auch signalisiert.
* Liegt ein Fehler vor, so wird man vom GUI über ein Dialogfeld darauf aufmerksam gemacht.
* Nach der Konvertierung einer Datei können auch noch weitere XML Dateien von Moodle konvertiert werden.
* Es ist möglich, eine Angabe für eine XML Datei zu überarbeiten und eine andere Datei anzugeben. Dasselbe gilt natürlich auch für den Namen der Ausgabedatei.

1. Bild Transformation
   * Eine Bilddatei, welche als Base64 Zeichenkette gespeichert wird, kann ausgelesen werden und es kann daraus eine Bilddatei erstellt werden.
   * Das neu erstellte Bild wird im korrekten Ordner erstellt.
   * Das neu erstellte Bild behält seinen Namen über die ganze Konvertierung.
2. Benutzeranleitung
   * Es liegt eine ausführliche Benutzeranleitung zur Applikation vor, welche die Kernfunktionen und Fehlermeldungen klar beschreibt.

## Konzept

### Anwendungsfalldiagramm



In einem Anwendungsfalldiagramm können Anwendungsfälle des Dateikonverters gut dargestellt werden. Der einzige Anwendungsfall, welcher ein Benutzer ausführen kann, ist das Konvertieren.

### Evaluation

Aufgrund der genauen Auflistung der Anwendungsfälle ist klar definiert, welche Probleme beim Programmieren zu lösen sind. Man kann also im Vorherein abschätzen, mit welchen Mitteln diese Probleme am einfachsten zu lösen sind, indem man verschiedene Methoden miteinander vergleicht.

#### Evaluation JDOM Library

Das Kernproblem dieser Diplomarbeit ist das Ein- und Auslesen von XML Daten. Wie in der Aufgabenstellung klar definiert, soll dies mit JDOM gelöst werden. JDOM ist eine leichte Open Source Library, welche sich genau darauf spezialisiert hat. Es ist für Java die wahrscheinlich einfachste Möglichkeit für das Ein- und Auslesen sowie das Bearbeiten von XML Daten.[[1]](#footnote-1) [[2]](#footnote-2)

#### Evaluation HTML Filter

Ein weiteres Problem ist das Parsen von HTML Tags aus den Fragetexten und aus den Antworten. Moodle ermöglicht die Formatierung von Antworten und Fragen. So können gewisse Wörter kursiv, fett oder unterstrichen angezeigt werden. Dies macht Moodle über HTML Tags. Da HSRvote solche Modifizierungen nicht unterstützt, müssen diese Tags nach dem Einlesen der XML Datei entfernt werden. Grundsätzlich gibt es hierzu zwei Optionen.

##### JSoup

JSoup ist eine externe Library, die auf das Lesen und Schreiben von HTML Tags spezialisiert ist. Sie offeriert eine API für das Extrahieren, Bearbeiten und Filtern von Daten, basierend auf DOM, CSS und JQuerry-ähnlichen Methoden.

Die Library unterstützt HTML5 und ist OpenSource unter der MIT Lizenz. Die Software kann also für dieses Projekt ohne Bedenken verwendet werden.[[3]](#footnote-3) [[4]](#footnote-4)

##### HTMLEditorKit

Das von Java Swing integrierte HTMLEditorKit ist die offizielle Schnittstelle der HTML Sprache. Diese Klasse unterstützt HTML bis auf die Version 3.2. Dies ist grundsätzlich genug weit, um die Tags von Moodle herauszufiltern.[[5]](#footnote-5)

##### Übersicht Kriterien

|  |  |  |
| --- | --- | --- |
|  | JSoup | HTMLEditorKit |
| Benutzerfreundlichkeit | Einfach zu nutzen, viele Beispiele. Welche Tags im String vorhanden sind, kann ignoriert werden. | Komplex zu nutzen, wenige gute Beispiele. Die Klasse erwartet auch eine genauere Angabe, über welche Tags sie filtern soll. |
| Verhaltung bei zusätzlicher Komplexität | JSoup hat auch bei verschachtelten HTML Tags keine Probleme, den Text noch zu lesen. | HTMLEditorKit kann Verschachtelungen erkennen und filtern. Dies ist aber komplex und braucht viel Code. |
| Import | Muss zusätzlich importiert werden | Ist bereits vorhanden. Muss nicht importiert werden. |
| Aktualität | HTML5 Die Version wird laufend weiterentwickelt. | HTML 3.2  Es ist nicht sicher wann die nächste Version kommt. Es kann sein, dass neue Tags missinterpretiert werden. |

##### Bewertungsmatrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Gewichtung  (%) | JSoup | | HTMLEditorKit | |
| **Wert** | **Gew. Wert** | **Wert** | **Gew. Wert** |
| Einfache Benutzung | 30 | 9 | 270 | 5 | 150 |
| Verhalten bei zusätzlicher Komplexität | 30 | 10 | 300 | 5 | 150 |
| Import | 20 | 7 | 140 | 10 | 200 |
| Aktualität | 20 | 9 | 180 | 4 | 80 |
| Total | 100 |  | 890 |  | 580 |

##### Fazit

Die JSoup Library geht für mich als klare Gewinnerin in dieser Evaluation hervor. Es war mir wichtig, dass die Benutzung der Library einfach ist und dass auch ineinander verschachtelte HTML Tags gefiltert werden können. Mit Moodle ist es möglich, Fragen oder Antworten als Liste, kursiv, mit oder ohne Zeilenumbrüchen anzugeben. Um HTML Tags aus Moodle herauszufiltern, ist der Code mit JSoup um ein Vielfaches simpler, als der Code, welcher bei der Verwendung des HTMLEditorKit verwendet werden müsste. Zusätzlich wird JSoup laufend auf dem neusten Stand gehalten. Dies war für dieses Projekt nicht sehr wichtig. Sollte Moodle dennoch neue Optionen anbieten, um Text zu formatieren, müsste der Source Code nicht umgeschrieben werden.

#### Evaluation Property Datei

Die XML Datei von HSRvote erwartet einen Input, welcher von der Moodle XML Datei nicht mitgegeben wird. Es handelt sich um die Definition der gewünschten Zeitspanne, welche dem Benutzer zur Beantwortung einer Frage zur Verfügung steht. Damit HSRvote die vom Programm erstellte Datei lesen kann, ist eine Property Datei zu verwenden. Diese ist einfach zu konfigurieren und hat einen geringen Datenverbrauch, was der Grund ist, dass Property Dateien in Java Applikationen der Standard für konfigurierbare Parameter sind.[[6]](#footnote-6)

Wäre nicht in der Aufgabenstellung bestimmt, dass für die Arbeit eine Property Datei verwendet werden muss, gäbe es auch noch die Möglichkeit, diese als Variable im Code zu speichern und bei der Eingabe festzulegen. Diese Option ist aber meist umständlicher für den Benutzer als einmal die Property Datei zu konfigurieren.

#### Evaluation Maven

Mit JSoup, JDOM und den zugehörigen JDOM Imports hat dieses Projekt mehr als genug Libraries, um Maven als Library Verwaltung in Betracht zu ziehen. Maven hat den Vorteil, anstelle vom Herunterladen und Einbinden von Libraries in ein Projekt, diese in einer Project Object Model (POM) Datei anzugeben, woraus Maven wiederum das Projekt eigenständig erstellt. Dies spart einerseits Zeit, andererseits ist die Verwaltung der Libraries, welche im Projekt gebraucht werden, viel einfacher.[[7]](#footnote-7)

#### Evaluation Test Driven Development

Test Driven Development (TDD) ist ein Softwareentwicklungsprozess, bei welchem das Testen mit Micro-Tests im Vordergrund steht. Programmiert wird dabei in kleinen Etappen, bei welchen zuerst ein kleiner Testfall geschrieben wird, welcher alsdann im Code implementiert wird. Durch die Verwendung von TDD wird der Refactoring Prozess nicht zu einem grossem Schritt, welcher nach dem Fertigstellen des Programmes angegangen werden muss, sondern zu vielen kleinen Schritten, in welchen der Code fortlaufend überarbeitet wird. Das Ziel von TDD ist sauberer Code aufgrund der Einteilung in einzelne Überarbeitungsschritte.

### Coding Style

Für den Code gelten die Regeln der Oracle Coding Konvention vom 20. April 1999. Alle Informationen dazu auf der Webseite:

<http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>

|  |  |
| --- | --- |
| javadoc | Es werden nur public und protected Member (Methoden und Attribute), Klassen und Interfaces dokumentiert. |
| Inline-doc | Es wird nur bei komplexen Algorithmen angewendet, oder bei einem Verhalten einer Bibliothek, das nicht zu erwarten war. |
| Klassendiagramm | Das Klassendiagramm wird nach dem Buch Design Pattern (Gang of Four) gezeichnet. |

### Gegebene XML Strukturen

Um Daten zu konvertieren, muss vorab definiert werden, welche Daten für die HSRvote Datei relevant sind und wo diese aus der Moodle Datei ausgelesen werden können. Glücklicherweise sind die XML Strukturen, welche von HSRvote eingelesen werden, sowie die Strukturen, welche von Moodle ausgegeben werden, genau definiert. So muss also nur nach bestimmten Elementen in der Moodle Datei gesucht werden, um diese herauszulesen und in den richtigen Elementen vom HSRvote abzuspeichern.

#### XML Aufbau von Moodle

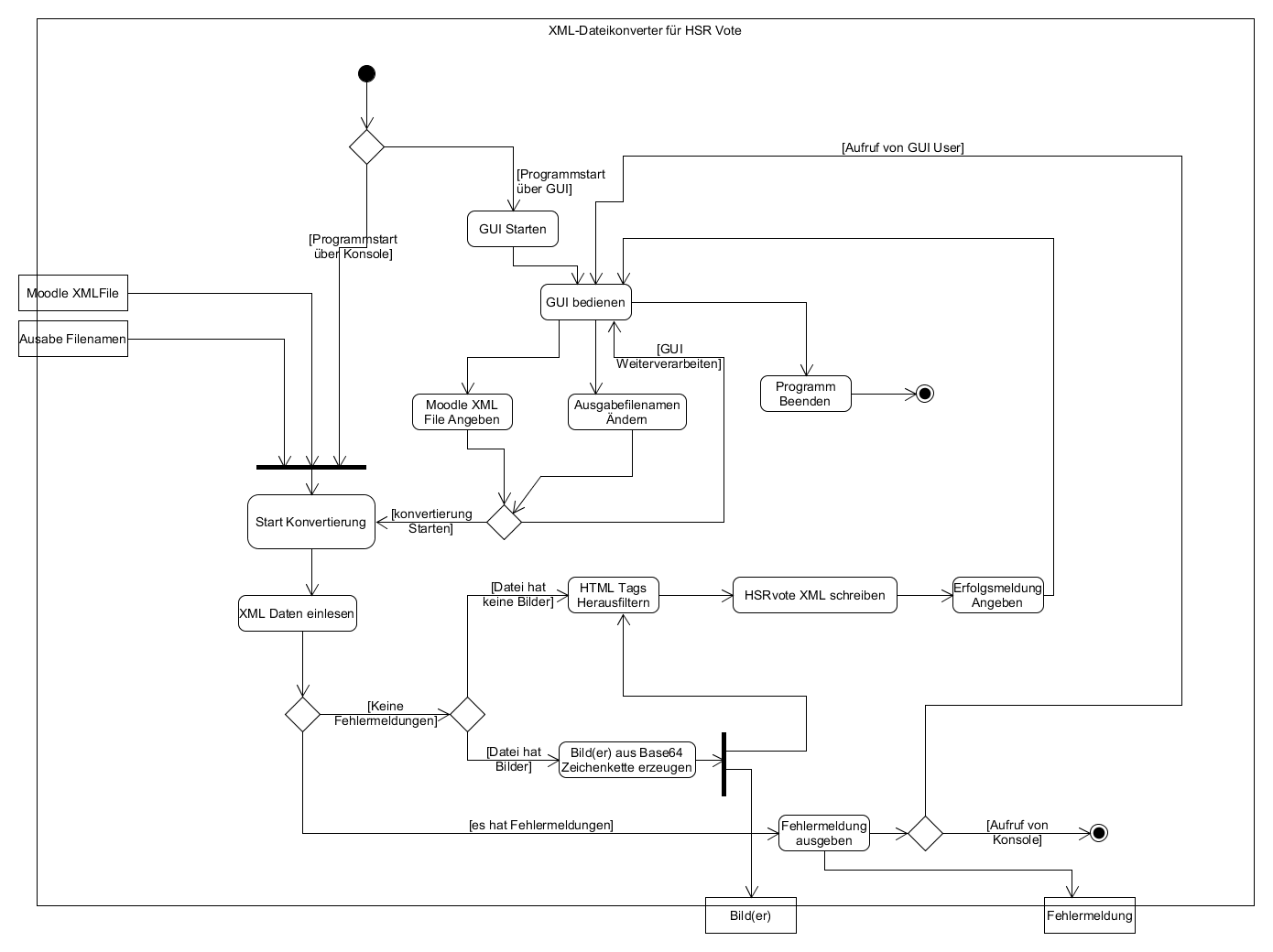
|  |  |
| --- | --- |
| <XML> | Dieses Element ist zuständig für die XML Deklaration.  Attributes:  version: Auskunft über die verwendete XML Version.  encoding: Auskunft über die erlaubten Zeichen im Dokument. |
| <quiz> | Dies ist das Root Element.  Children:  <question> |
| <question> | Dieses Element speichert die Frage und ihre Attribute. Zusätzlich bestimmt es den Typ der Frage.  Attributes:  type: Der Typ der Frage. Relevant für die Applikation sind nur Fragen mit dem Typ „multichoice“.  Children:  <name>  <questiontext>  <generalfeedback>  <defaultgrade>  <penalty>  <hidden>  <single>  <shuffleanswers>  <answernumbering>  <correctfeedback>  <partiallycorrectfeedback>  <incorrectfeedback>  <shownumcorrect>  <answer> |
| <name> | Dieses Element speichert den Namen der Frage.  Childeren:  <text> (Element um den Namentext zu speichern.) |
| <questiontext> | Dieses Element speichert den Fragetext.  Attributes:  format: Auskunft über das Textformat.  Childeren:  <text> (Element um den Fragetext zu speichern.)  Optional:  <file> (Element, welches verwendet wird, falls die Antwort ein Bild ist.) |
| <generalfeedback> | Dieses Element speichert den Feedbacktext.  Attributes:  format: Auskunft über das Textformat  Childeren:  <text> (Element um den Feedbacktext zu speichern.) |
| <defaultgrade> | Dieses Element speichert die Anzahl Punkte, welche bei einer korrekten Antwort gutgeschrieben werden. |
| <penalty> | Dieses Element speichert die Anzahl Punkte, welche bei einer falschen Antwort abgezogen werden. |
| <hidden> | Dieses Element gibt Auskunft darüber, ob die Frage angezeigt werden soll oder nicht. |
| <single> | Dieses Element gibt Auskunft ob die Frage einzeln angezeigt werden soll. |
| <shuffleanswers> | Dieses Element gibt Auskunft über die Reihenfolge der Antworten. (Ob diese gemischt werden dürfen, oder ob sie immer dieselbe Reihenfolge aufweisen müssen) |
| <answernumbering> | Dieses Element gibt Auskunft, wie die Antworten nummeriert sind (alphabetisch, mit Zahlen, etc.). |
| <correctfeedback> | Dieses Attribut speichert die Ausgabe für eine korrekte Antwort.  Attributes:  format: Auskunft über das Textformat  Childeren:  <text> (Element um den Feedbacktext zu speichern. |
| <partiallycorrect  feedback> | Dieses Attribut speichert die Ausgabe für eine teilweise korrekte Antwort.  Attributes:  format: Auskunft über das Textformat  Childeren:  <text> (Element um den Feedbacktext zu speichern.) |
| <incorrectfeedback> | Dieses Attribut speichert die Ausgabe für eine falsche Antwort.  Attributes:  format: Auskunft über das Textformat  Childeren:  <text> (Element um den Feedbacktext zu speichern.) |
| <shownumcorrect> | Dieses Element gibt an, ob nach dem Antworten die korrekte Antwort angezeigt werden soll. |
| <answer> | Dieses Element beinhaltet eine Antwort. Für jede Antwort hat es ein <answer> Element.  Attributes:  fraction: Prozentuale Angabe der Richtigkeit.  format: Auskunft über das Textformat.  Childeren:  <text> (Element um den Antworttext zu speichern.)  <feedback> (Element um den Feedbacktext zu speichern.) |
| <file> | Optional: (Element welches verwendet wird, falls die Antwort ein Bild ist.)  Dieses Element ist zuständig zum Speichern eines Bildes.  Attributes:  name (Name des Bildes)  path (optional: Pfad des Bildes)  encoding (Auskunft über den Encoding-Style des Bildes) |

#### XML Aufbau HSRvote

|  |  |
| --- | --- |
| <XML> | Dieses Element ist zuständig für die XML Deklaration.  Attributes:  version: Auskunft über die verwendete XML Version.  encoding: Auskunft über die erlaubten Zeichen im Dokument  standalone: Auskunft über die Validierung des XMLs |
| <ArrayOfQuestion> | Dies ist das Root Element.  Children:  <Question> |
| <Question> | Dieses Element speichert die Frage mit ihren Attributen.  Children:  <QuestionText>  <Answer1>  <Answer2>  <Answer3>  <Answer4>  <PictureQuestionText>  <PictureAnswer1>  <PictureAnswer2>  <PictureAnswer3>  <PictureAnswer4>  <Solution>  <Time>  <Points> |
| <QuestionText> | Dieses Element speichert die Frage als Text |
| <Answer1> | Dieses Element speichert die erste Antwort als Text |
| <Answer2> | Dieses Element speichert die zweite Antwort als Text |
| <Answer3> | Dieses Element speichert die dritte Antwort als Text |
| <Answer4> | Dieses Element speichert die vierte Antwort als Text |
| <PictureQuestionText> | Dieses Element speichert die Frage als Bild |
| <PictureAnswer1> | Dieses Element speichert die erste Antwort als Bild |
| <PictureAnswer2> | Dieses Element speichert die zweite Antwort als Bild |
| <PictureAnswer3> | Dieses Element speichert die dritte Antwort als Bild |
| <PictureAnswer4> | Dieses Element speichert die vierte Antwort als Bild |
| <AnswerDelay> | Dieses Element speichert die Anzahl Sekunden, bevor die Antworten der Frage erscheinen. |
| <Solution> | Dieses Element gibt die korrekte Lösung zur Antwort an. |
| <Time> | Dieses Element speichert Anzahl Sekunden, welche man hat, um die Frage zu beenden. |
| <Points> | Dieses Element speichert die Anzahl Punkte für das korrekte Beantworten der Frage. |

Zu beachten bei dieser Struktur ist, dass jeweils nur entweder der Tag Answer oder der Tag PictureAnswer einen Wert haben darf. Ansonsten wird die XML Datei nicht in HSRvote eingelesen.

### Programm Ablauf

Die Applikation gibt dem Benutzer zwei Wege, um Dateien zu konvertieren. Entweder kann das Programm über das GUI oder direkt über die Konsole mittels Angabe zweier Parameter gestartet werden. Das GUI erlaubt dem Benutzer in einfacher Weise eine Datei auszuwählen, welches er konvertieren möchte, sowie den Namen der Ausgabedatei zu bestimmen. Das GUI kann während der Benutzung laufend geschlossen werden, was über den „Abbrechen“ Button ausgeführt werden kann. Ist der Benutzer zufrieden mit seinen Einstellungen, kann er versuchen, eine Konvertierung zu starten. War die Konvertierung erfolgreich, bekommt der Benutzer eine Bestätigung und kann abermals das GUI bedienen. Wurde die Applikation von der Konsole aufgerufen, so wird im Falle einer erfolgreichen Konvertierung eine Bestätigung auf der Konsole angezeigt.

## Realisierung

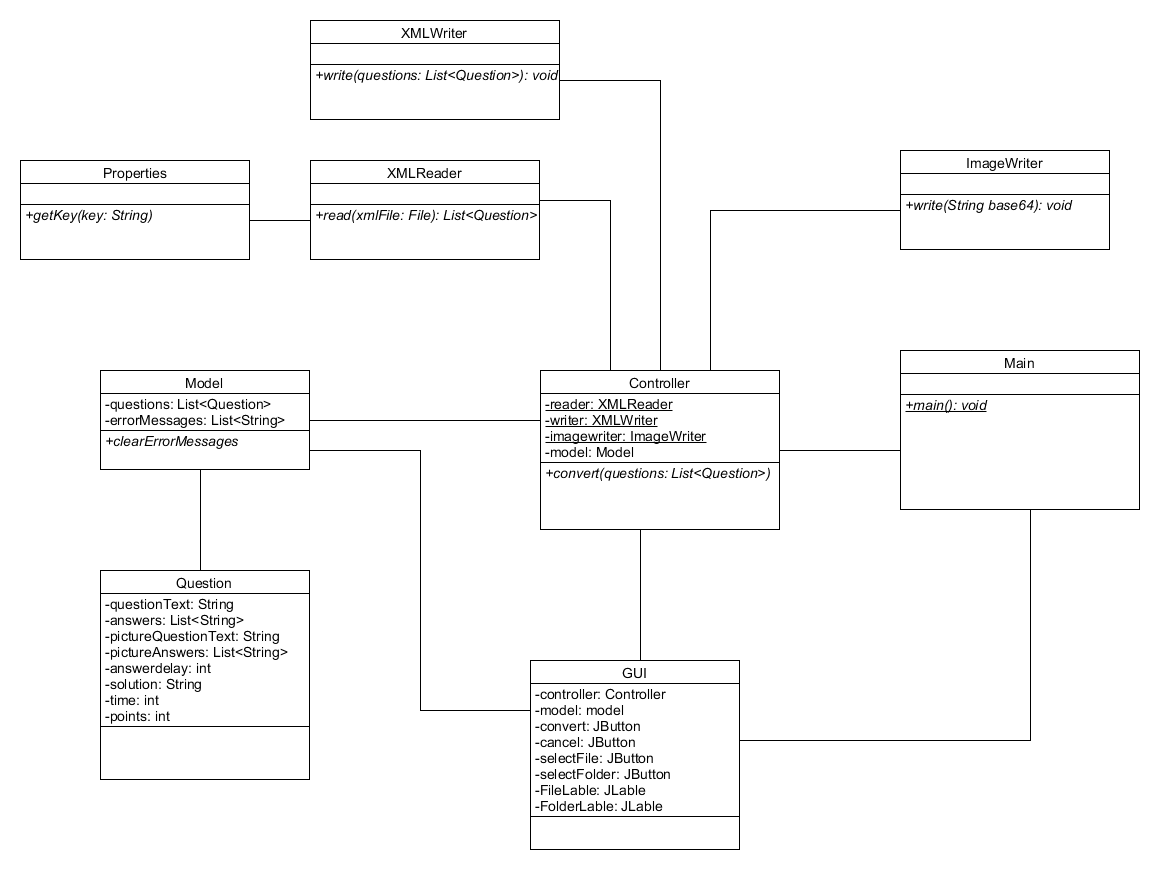
### Versionsverwaltung

Für die Versionsverwaltung wird sowohl GIT verwendet, als auch das vorgeschriebene Ordnersystem, in welchem alle Daten täglich in einen neuen Ordner gezogen werden. Das GIT wird über die Entwicklungsumgebung initialisiert und verwaltet.

### Entwicklungsumgebung einrichten

Die Applikation wird auf Netbeans 8.0.2 entwickelt. Netbeans ist kostenlos und ist als Entwicklungsumgebung bestens geeignet, um diese Applikation umzusetzen. Netbeans beinhaltet auch einen GUI-Builder, welcher beim Entwickeln des GUIs verwendet wird.

### Programm Gerüst



Das Programm wird nach MVC entwickelt. Das Model – View –Controller Prinzip hat sich bewährt und ermöglicht das Schreiben von sauberem und übersichtlichem Code. Die View wird dabei von einem GUI dargestellt.

Um XML Daten einzulesen beziehungsweise auszuschreiben und um Bilder aus einem base64Code zu erstellen, werden jeweils eigene Klassen angelegt. Damit die XML Daten gut gehandhabt werden können, werden sie in eine Question Klasse verpackt.

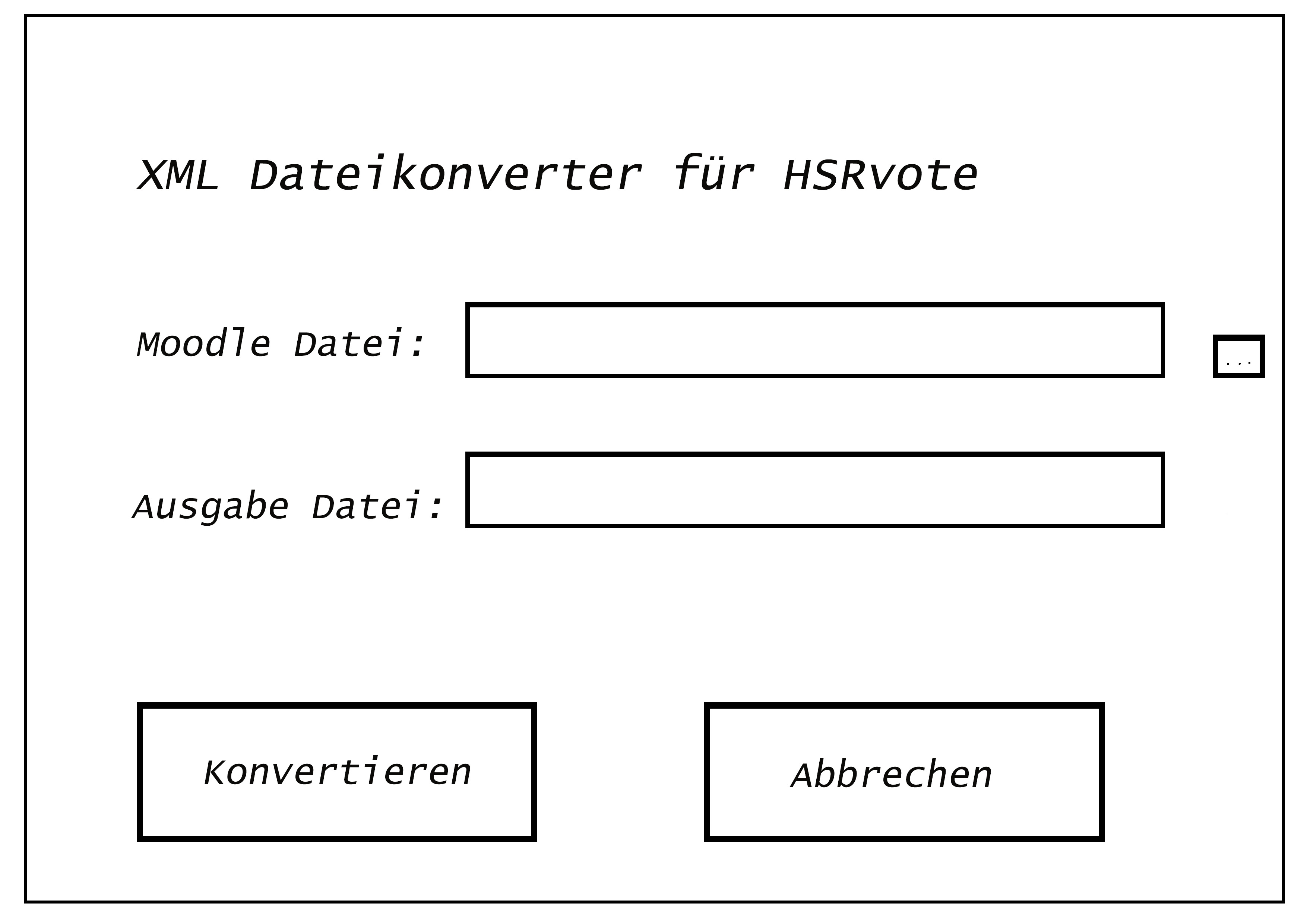
### Main

Die Main-Klasse ist für den Start der Applikation zuständig. Da die Applikation sowohl über eine Konsole als auch über ein GUI gestartet werden kann, sind die Argumente, welche dem Programm beim Start mitgegeben werden, essenziell. Beim Starten der Applikation ohne Argumente wird ein GUI gestartet, bei einem Aufruf mit Argumenten aber nicht.

### Model

Das Model ist sehr schlank aufgebaut. Wie es sich gehört, ist das Model bei MVC für die Daten zuständig, mit welchen das Programm arbeitet. In dieser Applikation speichert es die Fragen einer Datei in einer Liste von Question Objekten.

### GUI



Das GUI übernimmt in der Applikation mit der Konsole die Rolle der View und ist für das Darstellen der Daten zuständig. Es hat ein einfaches Design, welches aus zwei Teilen besteht. Im oberen Teil hat der User die Option, über einen Button eine Datei auszuwählen, welche er konvertieren möchte. Der Name der Ausgabedatei wird automatisch generiert und setzt sich aus dem Namen der eingelesenen Datei mit der Endung „–vote“ zusammen. Im Unteren Teil des GUIs hat es zwei Buttons: Einer zum Starten der Konvertierung und einer zum Abbrechen derselben, also zum Beenden der Applikation.

### Controller

Die Controller-Klasse führt Aktionen aus, welche es vom GUI erhalten hat. Es besitzt zwei statische Variable, welche jeweils eine Instanz der XML-Reader und XML-Writer Klassen sind. Die wichtigste Methode in dieser Klasse ist „convert“. Sie gibt der Reader und der Writer Klasse den Befehl, Dateien zu lesen beziehungsweise zu schreiben.

Zusätzlich ist der Controller zuständig für das Behandeln der Fehlermeldungen innerhalb der Reader und Writer Klassen. Über die Funktion „handleErrorMessages“ wird die Nachricht der Fehlermeldung für den Benutzer festgelegt, sowie das GUI informiert, sofern dieses existiert und darauf reagieren könnte. Ist dies nicht der Fall, wird die Nachricht auf der Konsole ausgegeben.

### XMLReader

Die XMLReader Klasse ist primär zuständig für das Einlesen von Daten. Mit der Funktion „fetchQuestions“ werden die Fragen aus der mitgegebenen XML Datei ausgelesen und in Question Objekten gespeichert. Im XMLReader werden die Daten während des Einlesens bereits aufbereitet. HTML Tags, welche in Frage- oder Antworttexten sind, werden gefiltert und aus Bildern, welche in der XML Datei als Base64Code gespeichert sind, werden mit der ImageWriter Klasse zu Bilddateien konvertiert.

### XMLWriter

Die XMLWriter Klasse schreibt eine XML Datei aus einem Objekt. Dies geschieht mit JDOM. Da der Header des Programmes im Normalfall vom Dokument abgeschnitten wird und von JDOM ein Standardheader geschrieben wird, muss dieser angepasst werden. Dazu wird die Methode „printDeclaration“ JDOM der Klasse „AbstractXMLOutputProcessor“ überschrieben.

### Pom.xml

Die Datei pom.xml ist eine von Maven generierte Datei, welche dazu dient, die Libraries für das Projekt zu konfigurieren. Dies wird mit der GroupID, ArtifaktID und der aktuellen Version der Library gemacht. Mit diesen Informationen kann Maven innerhalb eines dependancy Elements das Projekt und die JAR Datei erstellen.

## Testen

### Testkonzept

Die Applikation wird mit JUnit- und mit Blackbox-Tests getestet. JUnit Tests werden verwendet, da die Applikation mit TDD (Test Driven Developement) entwickelt wird. Für den Softwareentwicklungsprozess sind JUnit Tests somit wichtig, da sie das Fundament für die Applikation bilden. Die JUnit Tests werden auf der Version 4.0 programmiert.

Trotz der breiten Testabdeckung durch TDD werden auch noch Blackbox Tests benötigt. Erfolgreiche Blackbox Tests garantieren, dass das Programm so läuft, wie es geplant wurde. JUnit Tests hingegen garantieren lediglich, dass die programmierten Funktionen erfolgreich durchgeführt werden können.

### Blackbox Tests

| Test Nr. | Testfall | Erwartetes Resultat |
| --- | --- | --- |
| 1. | Das GUI wird vom User gestartet. | Das GUI erscheint. |
| 2. | GUI kann vom User mit den „Abbrechen“ Button geschlossen. | Das GUI schliesst sich. |
| 3. | Datei auswählen, welche konvertiert werden soll. | Der Dateipfad der ausgewählten Datei wird in der Anzeige für den Dateipfad angezeigt. |
| 4. | Namen der Ausgabedatei auswählen | Der angegebene Name wird im GUI ausgewählt. |
| 5. | Nach dem Angeben einer beliebigen XML Datei, wird diese über den Button geändert, welcher zum Auswählen der XML Datei zuständig ist. | Der Dateipfad der neu ausgewählten Datei erscheint im entsprechenden Dateitextfeld. |
| 6. | Der Benutzer wählt die gewünschte Moodle Datei, woraufhin der Ausgabedateiname automatisch generiert wird, sofern dieser vorab noch nicht bestimmt wurde. | Es wird automatisch ein Name für die Ausgabedatei erstellt. Der autogenerierte Name ist derselbe, wie der von der Name der gewählten XML Datei, mit der zusätzlichen Endung „–vote“. |
| 7. | Zuerst wird ein Name für eine Ausgabedatei gesetzt, danach wird eine neue XML Datei eingelesen. | Der bereits angegebene Ausgabedateiname wird nicht überschrieben. |
| 8. | Nach einer erfolgreichen Konvertierung wird die XML Datei über den entsprechenden Button geändert, welcher zum Auswählen der XML Datei zuständig ist. | Der Dateipfad der neu ausgewählten Datei wird in der Anzeige für den Datei Pfad angezeigt. |
| 9. | Nach einer erfolgreichen Konvertierung, wird der Ausgabedateiname geändert. | Die Anzeige für den Ausgabedateinamen zeigt den neuen Namen an. |
| 10. | Das GUI wird geschlossen und wieder geöffnet, um zu prüfen, dass die Anzeige für die ausgewählte XML Datei leer ist. | Die Anzeige für die ausgewählte XML Datei ist leer. |
| 11. | Das GUI wird geschlossen und wieder geöffnet, um zu prüfen, ob die Anzeige für den gewählten Ausgabedateinamen leer ist. | Die Anzeige für den gewählten Ausgabedateinamen ist leer. |
| 12. | Eine valide XML Datei, welche von Moodle generiert wurde, wird über das GUI zu einer HSRvote XML Datei konvertiert. | Eine HSRvote XML Datei wird erzeugt und kann mit HSRvote eingelesen werden. |
| 13. | Eine XML Datei, welche nicht von Moodle generiert wurde, wird vom Benutzer konvertiert. | Das GUI signalisiert dem Benutzer, dass die gewählte Datei nicht konvertiert werden kann. |
| 14. | Beim Konvertieren befindet sich die Ausgabe des Programms im gleichen Ordner wie die Eingabedatei. | Der Output befindet sich im gleichen Ordner wie die Eingabedatei. |
| 15. | Es wird eine XML Datei konvertiert, welche Bilder speichert. | Erfolgreiche Konvertierung, bei welcher Bilder als eigene Dateien erstellt werden. |
| 16. | Bilder welche bei der Konvertierung zu HSRvote erstellt wurden, behalten denselben Namen bei, welchen sie hatten, als sie in eine XML Datei von Moodle konvertiert wurden. | Neu erstellte Bilder haben denselben Namen, welchen sie vor der Konvertierung hatten. |
| 17. | Bilder welche bei der Konvertierung mit dem GUI entstanden sind, erscheinen im gleichen Ordner wie die Ein- und Ausgabedatei. | Die Bilder werden im gleichen Ordner erstellt, wie die Ein- und Ausgabedatei. |
| 18. | Der Applikation wurde eine XML Datei angegeben, welche nicht geparst werden kann. | Es erscheint eine Fehlermeldung, dass die Datei nicht geparst werden kann. |
| 19. | Der Applikation wurde eine XML Datei angegeben, welche nicht konvertiert werden kann | Es erscheint eine Fehlermeldung, dass die Datei nicht konvertiert werden kann. |
| 20. | Für die Ausgabedatei wird keine Datei Extension angegeben und dennoch konvertiert. | Die Datei Extension für die Ausgabedatei wird automatisch generiert. |
| 21. | Es wird für die Ausgabedatei eine Datei Extension angegeben, welche nicht mit „.xml“ endet. | Die angegebene Datei Extension wird entfernt und mit der Extension „.xml“ ersetzt. |

### Testprotokoll

| Test Nr. | Resultat | Bemerkung | Erfolg\* |
| --- | --- | --- | --- |
| 1. | Das GUI erscheint. |  |  |
| 2. | Das GUI schliesst sich. |  |  |
| 3. | Der Dateipfad der ausgewählten Datei wird in der Anzeige für den Dateipfad angezeigt. |  |  |
| 4. | Der angegebene Name wird im GUI ausgewählt. |  |  |
| 5. | Der Dateipfad der neu ausgewählten Datei erscheint im entsprechenden Dateitextfeld. |  |  |
| 6. | Es wird automatisch ein Name für die Ausgabedatei erstellt. Der autogenerierte Name ist derselbe, wie der von der Name der gewählten XML Datei, mit der zusätzlichen Endung „–vote“. |  |  |
| 7. | Der bereits angegebene Ausgabedateiname wird nicht überschrieben. |  |  |
| 8. | Der Dateipfad der neu ausgewählten Datei wird in der Anzeige für den Datei Pfad angezeigt. |  |  |
| 9. | Die Anzeige für den Ausgabedateinamen zeigt den neuen Namen an. |  |  |
| 10. | Die Anzeige für die ausgewählte XML Datei ist leer. |  |  |
| 11. | Die Anzeige für den gewählten Ausgabedateinamen ist leer. |  |  |
| 12. | Eine HSRvote XML Datei wird erzeugt und kann mit HSRvote eingelesen werden. |  |  |
| 13 | Das GUI signalisiert dem Benutzer, dass die gewählte Datei nicht konvertiert werden kann. |  |  |
| 14. | Der Output befindet sich im gleichen Ordner wie die Eingabedatei. |  |  |
| 15. | Erfolgreiche Konvertierung, bei welcher Bilder als eigene Dateien erstellt werden. |  |  |
| 16. | Neu erstellte Bilder haben denselben Namen, welchen sie vor der Konvertierung hatten. |  |  |
| 17. | Die Bilder werden im gleichen Ordner erstellt, wie die Ein- und Ausgabedatei. |  |  |
| 18. | Es erscheint eine Fehlermeldung, dass die Datei nicht geparst werden kann. |  |  |
| 19. | Es erscheint eine Fehlermeldung, dass die Datei nicht konvertiert werden kann. |  |  |
| 20. | Die Datei Extension für die Ausgabedatei wird automatisch generiert. |  |  |
| 21. | Die angegebene Datei Extension wird entfernt und mit der Extension „.xml“ ersetzt. |  |  |

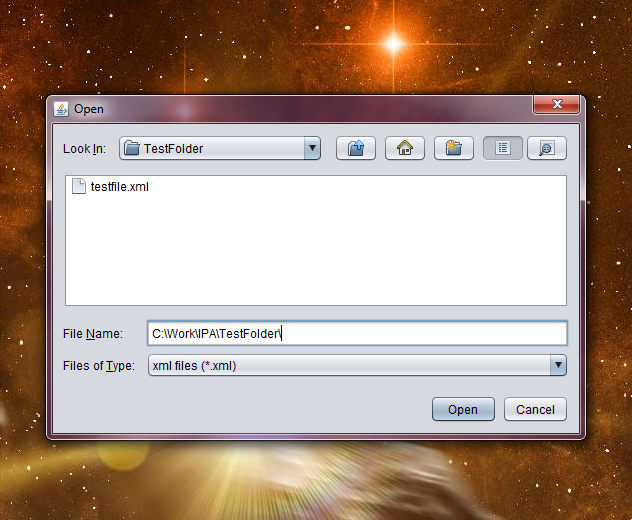
## Ergebnis

Die Applikation erfüllt alle definierten Erwartungen.

### Konvertieren mit dem GUI



Über das Ausführen der JAR Datei wird das GUI geöffnet. Es können Dateien mittels des Buttons neben dem ersten Textfeld ausgewählt werden, welche konvertiert werden sollen. Zur ausgewählten Datei wird auch automatisch ein passender Name generiert, welcher aber noch geändert werden kann.

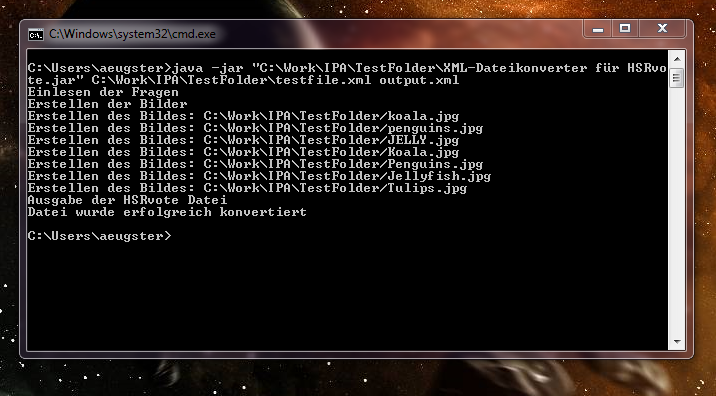


Über ein Dateiauswahlfenster kann eine Datei zum Konvertieren ausgewählt werden.



Mittels Pop-Up Fenster werden dem Benutzer Mitteilungen angezeigt, wie beispielsweise eine erfolgreiche Konvertierung oder Fehlermeldungen.

### Konvertieren mit der Konsole



Die Applikation kann über die Konsole aufgerufen werden und so Dateien konvertieren. Über die Konsole wird dem Benutzer mittgeteilt, woran das Programm gerade arbeitet. Gibt es Fehlermeldungen, werden diese dem Benutzer angezeigt.

## Schlusswort

Die Individuelle Praktische Arbeit wurde innerhalb der vorgegebenen Frist von zehn Tagen erfolgreich gelöst und umgesetzt. Alle Punkte der Aufgabenstellung wurden berücksichtigt und erfüllt. Die Abgabe der IPA beinhaltet den Dateikonverter, welcher detailliert geplant und ausführlich getestet wurde, sowie eine umfassende Dokumentation mit Benutzeranleitung.

Das Projekt stellte sich als aufwändiger heraus, als ich ursprünglich angenommen hatte, war aber auch sehr lehrreich und motivierend. Das detaillierte Dokumentieren war mir fremd und der Entscheid, was genau und wie ich gewisse Teile dokumentieren soll, war nicht immer einfach. Zudem fiel mir das verständliche Formulieren nicht immer auf Anhieb leicht. Eine weitere Herausforderung war das Zeichnen von Diagrammen. Es fehlte mir die Übung und so musste ich die meisten Diagramme mehrfach überarbeiteten, bis diese zu meiner Zufriedenheit waren. Dies kostete mich viel Zeit, welche ich sicherlich optimaler hätte nutzen können. Generell war der Faktor Zeit neu für mich. Ich hatte bis anhin noch nie nach einem so klar definierten Zeitplan gearbeitet, geschweige denn, je einen solchen erstellt.

Rückblickend denke ich aber, dass der Zeitplan, welchen ich für dieses Projekt entwickelt und nach welchem ich gearbeitet habe, zu einem guten Resultat geführt und mir die Sicherheit vermittelt hat, zeitlich auf Kurs zu sein und alle Faktoren berücksichtigt zu haben. Während der Realisierungsphase des Projektes stellte ich fest, dass sich das Arbeiten mit JDOM für mich als weniger komplex erwies, wie ich dies anfangs erwartet hatte. Durch die gewonnene Zeit beim Programmieren konnte ich meinen Rückstand im Dokumentieren wettmachen und stand so während der Arbeit selten unter Zeitdruck.

Rückblickend darf ich festhalten, dass die Arbeit gut durchdacht war und sich auch vom Umfang her ideal für eine IPA eignete. Es machte mir Spass, ein Projekt von Anfang bis zum Ende realisieren zu können und ich bin stolz auf das Ergebnis. Die Nutzbarkeit des Projektes sehe ich als wertvoll an und hoffe, dass durch die Optimierung HSRvote nun häufiger genutzt werden kann.

Ich möchte mich an dieser Stelle bei meinem Fachvorgesetzten Herr Peter Bühler bedanken, der mich während der IPA betreut hat und die Idee für diese spannende IPA hatte.

## Glossar

GIT: Software zur Versionsverwaltung

GUI: Graphical User Interface ist ein Anzeigefenster für den Benutzer, um mit dem Programm zu interagieren.

HSR: Hochschule für Technik Rapperswil

HSRvote: Ist ein didaktisches Hilfsmittel, mit welchem Multiple-Choice Fragen für den Unterricht durchgeführt werden können. Über ein kreditkartengrosses Gerät können bei einer Abstimmung eine von vier Antworten ausgewählt werden, welches dann von HSR-vote registriert wird.

JDOM: Java Library für das Ein- und Auslesen von XML Dateien

Moodle: Eine Software um im Internet Lernkurse durchzuführen und zu entwickeln.

UI: User Interface (siehe GUI)

XML: Extensible Markup Language ist ein Dateiformat, welches verwendet wird um Daten zu speichern in einer Sprache, welche von Mensch und Maschine gelesen werden kann.

## Quellen

<http://vote.hsr.ch>

Buch „XML IN A NUTSHELL“ vom O'Reilly Verlag GmbH & Co. KG (ISBN: 3897213397)

<https://www.pkorg.ch/> (IPA Detaildaten Eugster Alex)

<https://de.wikipedia.org/wiki/Anwendungsfalldiagramm>

<https://de.wikipedia.org/wiki/Klassendiagramm> <https://de.wikipedia.org/wiki/Aktivit%C3%A4tsdiagramm>

<https://de.wikipedia.org/wiki/Black-Box-Test>

Buch UML 2 - Zertifizierung: Fundamental, Intermediate und Advanced vom dpunkt.verlag. (ISBN: 3-89864-294-1)

<http://www.jdom.org/docs/faq.html#a0000>

<https://en.wikipedia.org/wiki/JDOM>

<https://jsoup.org/>

<https://jsoup.org/license/>

<https://docs.oracle.com/javase/7/docs/api/javax/swing/text/html/HTMLEditorKit.html>

<https://en.wikipedia.org/wiki/.properties>

<https://maven.apache.org/what-is-maven.html>

<http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>

Buch Design Pattern vom Prentice Hall Verlag  (ISBN-13: 978-0201633610)

<http://stackoverflow.com/questions/23979842/convert-base64-string-to-image>

# Anhang 1

[Code]

# Anhang 2

## Benutzeranleitung

Contents

[Einleitung 53](#_Toc450915792)

[Konvertieren über das GUI 53](#_Toc450915793)

[General User Interface (GUI) 53](#_Toc450915794)

[Dateiauswahl 54](#_Toc450915795)

[Konvertieren über die Konsole 55](#_Toc450915796)

[Benutzer Nachrichten 55](#_Toc450915797)

### Einleitung

Der Dateikonverter für HSRvote ist dazu zuständig, Multiple-Choice Fragen, welche auf der Lernplattform Moodle entwickelt wurden, für das Tool HSRvote lesbar zu machen.

Die Applikation kann auf zwei Arten gestartet werden. Der benutzerfreundlichere Weg ist über das Öffnen der JAR Datei. Es erscheint ein GUI, in welchem die Moodle Datei und der Ausgabedateiname angegeben werden kann. Alternativ kann die Applikation über die Konsole gestartet werden. In diesem Fall muss der Benutzer die Datei und den Ausgabedateinamen als Parameter mitgegeben.

### Konvertieren über das GUI

#### General User Interface (GUI)

Durch das Öffnen der JAR Datei öffnet sich das GUI. (Bild1)



Bild1

Im Fenster hat es zwei Eingabefelder. Im ersten Eingabefeld soll die Datei angegeben werden, welches von Moodle generiert wurde und konvertiert werden soll.

Entweder gibt der Benutzer den Dateipfad direkt in das Textfeld ein, oder er wählt eine Datei aus, indem er auf den Button drückt, der sich neben dem Textfeld befindet. Dieser Button öffnet die Dateiauswahl Ansicht. (Bild2)

Nachdem eine Datei über die Dateiauswahl gewählt wurde, wird automatisch ein Ausgabe Dateiname festgelegt. Dieser wird im zweiten Textfeld angezeigt, sofern noch kein anderer Name bestimmt wurde. Ist der Name unpassend, kann der Benutzer diesen noch anpassen.

Auf dem Fenster werden zwei weitere Buttons angezeigt:

* Der Button „Konvertieren“ startet die Konvertierung.
* Der Button „Abbrechen“ beendet die Applikation.

Wird Konvertierung durchgeführt, erscheint ein Dialogfenster. Dieses bestätigt die erfolgreiche Konvertierung oder zeigt eine Fehlermeldung an. Will der Benutzer seine Eingabe korrigieren, kann er über den „OK“ Button zurück zum Ursprungsfenster gelangen. Das Programm kann an dieser Stelle auch mittels des entsprechenden Buttons beendet werden. (Bild4)

#### Dateiauswahl

In der Dateiauswahl Ansicht lassen sich XML Dateien auswählen, welche man konvertieren möchte. Hierzu kann man entweder durch das User Interface zur Datei navigieren, oder im Eingabefeld „File Name“ den Pfad direkt angeben.

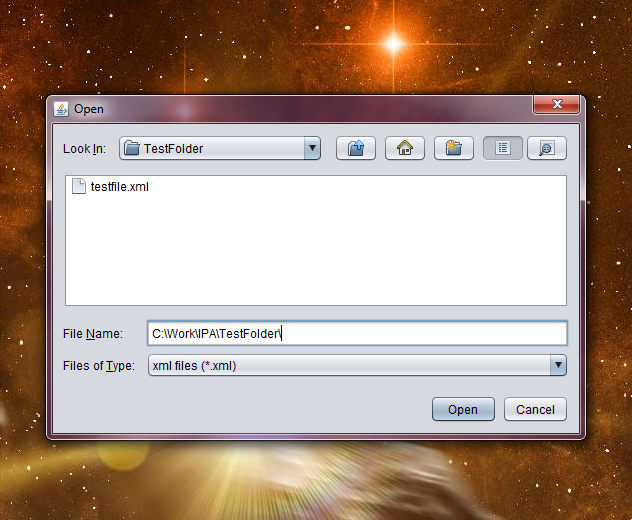


Bild2

### Konvertieren über die Konsole

Die Applikation kann auch über die Konsole gestartet werden. Der Befehl für eine Konvertierung sieht wie folgt aus (Bild3):

Java –jar arg1 arg2 arg3

Arg1: Pfad zum Dateikonverter JAR

Arg2: Pfad zur XML Datei

Arg3: Ausgabe Name der Datei

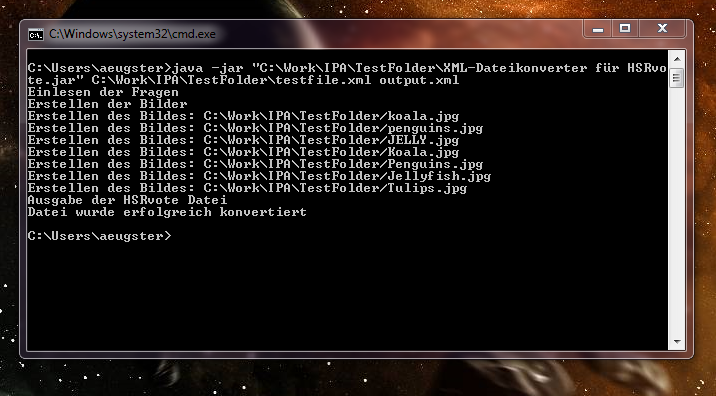


Bild3

### Benutzer Nachrichten

Um dem Benutzer zu signalisieren, dass die Konvertierung abgeschlossen ist, wird eine Nachricht angezeigt.(Bild4)

Bild4

Nachricht bei einer erfolgreichen Konvertierung:

|  |  |
| --- | --- |
| Nachricht | Beschreibung |
| Datei wurde erfolgreich konvertiert. | Bestätigung, dass eine Konvertierung erfolgreich durchgeführt werden konnte und alle Bilder erstellt wurden. |

Nachricht bei einer fehlerhaften Konvertierung:

|  |  |
| --- | --- |
| Nachricht | Beschreibung |
| Die Eingabedatei ist ungültig | Die Eingabedatei konnte nicht konvertiert werden. Dies kann der Fall sein, wenn die Datei sich in einem Ordner befindet, auf welchen die Applikation keine Schreibrechte hat, oder nicht existiert. |
| Die Datei hat keine .xml Extension | Die angegebene Datei hat eine Datei Extension, welche nicht gelesen werden kann. Die Applikation liest nur Dateien ein, welche auf .xml enden. |
| Es gab ein Problem mit dem Erstellen der Bilder | Die Bilder konnten nicht erstellt werden. Dies kann passieren, wenn der Ordner, in welchem Bilder erstellt werden sollten, schreibgeschützt ist. Auch führt es zu Problemen, wenn die Bilder ein Format haben, welches nicht zu einem Bild konvertiert werden kann. |
| Die eingegebene Datei wurde nicht von Moodle generiert | Die angegebene Datei konnte nicht zu HSRvote konvertiert werden, da das XML Format nicht von Moodle generiert wurde. |
| Es gab ein Problem mit dem Erstellen der Ausgabedatei | Es konnte keine Ausgabedatei erstellt werden. Dies kann passieren, wenn der Ordner, in welchem die Datei erstellt werden sollte, schreibgeschützt ist. |

# Anhang 3

## Protokolle

**Datum: 04.05.2016**  
Anwesende: Peter Bühler, Alex Eugster

Bei der Besprechung wurde festgelegt, dass:

* Der Benutzer soll nicht ein Zielverzeichnis angeben können, sondern dafür den Namen der Ausgabedatei.
* Es soll automatisch ein Ausgabedateiname generiert werden, wenn eine Datei ausgewählt wurde. Der automatisch generierte Name soll derselbe sein, wie der Name der Eingabedatei, jedoch soll die Endung „-vote“ dem Dateinamen angehängt werden.

1. http://www.jdom.org/docs/faq.html#a0000 [↑](#footnote-ref-1)
2. https://en.wikipedia.org/wiki/JDOM [↑](#footnote-ref-2)
3. <https://jsoup.org/> [↑](#footnote-ref-3)
4. <https://jsoup.org/license/> [↑](#footnote-ref-4)
5. <https://docs.oracle.com/javase/7/docs/api/javax/swing/text/html/HTMLEditorKit.html> [↑](#footnote-ref-5)
6. https://en.wikipedia.org/wiki/.properties [↑](#footnote-ref-6)
7. https://maven.apache.org/what-is-maven.html [↑](#footnote-ref-7)