|  |
| --- |
|  |
| Winterprüfung 2018 |
|  |
| Fachinformatiker Anwendungsentwicklung |
| Dokumentation zur praktischen Projektarbeit |
|  |
| Dokumentation |
| Antrags- und Formularerstellungstool |
| Werkzeug zur modularen Antrags- und Formularerstellung mit Prozessabwicklung |
|  |
| **Abgabetermin: 07.12.2018** |
|  |
| **Kunde:**  ATIW Berufskolleg Paderborn  Riemkestraße 160  33106 Paderborn |
|  |
| **Ausbildungsbetrieb:**  Atos IT Dienstleistung und Beratung GmbH  Bruchstraße 5  45883 Gelsenkirchen |
|  |
|  |
|  |
|  |
|  |

Inhalt

[Einführung 1](#_Toc531858441)

[Kundenauftrag 1](#_Toc531858442)

[Ist-Zustand 1](#_Toc531858443)

[Soll-Zustand 1](#_Toc531858444)

[Analyse- und Planungsphase 3](#_Toc531858445)

[Allgemeiner Ablauf der Phase 3](#_Toc531858446)

[Terminplan 3](#_Toc531858447)

[Gantt-Diagramm 3](#_Toc531858448)

[Kostenplanung 4](#_Toc531858449)

[Sachplan 5](#_Toc531858450)

[Durchführungsphase 6](#_Toc531858451)

[Allgemeiner Ablauf der Phase 6](#_Toc531858452)

[Ablauf der Implementation 6](#_Toc531858453)

[Ergebnis der Implementation 6](#_Toc531858454)

[Ablauf der Abnahme 6](#_Toc531858455)

[Projekt-Tagebuch 6](#_Toc531858456)

[Testphase 8](#_Toc531858457)

[Allgemeiner Ablauf der Phase 8](#_Toc531858458)

[Ablauf des Testens 8](#_Toc531858459)

[Bewertung 9](#_Toc531858460)

[Allgemeiner Ablauf des Projekts 9](#_Toc531858461)

[Abweichung in der Zeitplanung 9](#_Toc531858462)

[Abweichungen in den Muss-, Soll- und Kann-Ziele 10](#_Toc531858463)

[Abweichungen von Ergebnissen aus der Analyse- und Planungsphase 11](#_Toc531858464)

[Abweichungen in der Qualitätskontrolle 11](#_Toc531858465)

[Abweichungen in der Kostenplanung 11](#_Toc531858466)

[Mehrwert geschaffen? 12](#_Toc531858467)

[Benutzerhandbuch 12](#_Toc531858468)

[Anhang 12](#_Toc531858469)

[Pflichtenheft 12](#_Toc531858470)

[Graphische Darstellungen (Struktogramm, UML, ER-Diagramme) 12](#_Toc531858471)

[Testprotokolle 12](#_Toc531858472)

# Einführung

## Kundenauftrag

Das Projekt des Antrags- und Formularerstellungstool wurde uns am 13.11.2018 vorgeschlagen. Der Kunde ist das Kollegium des ATIW Berufskollegs gGmbH Paderborn, Ansprechpartner ist Herr Böhmer. Es handelt sich um ein Tool, das die internen Prozesse in der Berufsschule ATIW in Paderborn vereinfachen soll, da aktuell grundlegende Formulare noch in Papierform angefertigt und genehmigt werden. Es ist datiert auf den 07.12.2018.

### Ist-Zustand

Das Problem des Kollegiums ist das interne Antrags- und Formularwesen. Derzeit müssen Anträge über verschiedenste Word- bzw. PDF Vorlagen ausgefüllt, ausgedruckt und unterschrieben werden, bevor Sie zu Fuß an die nächst höhere Instanz weitergeleitet werden. Das bestehende System ist folglich sehr fehleranfällig. Dokumente gehen verloren, Fristen werden nicht eingehalten und falsche Angaben werden gemacht. Anträge in Papierform erzeugen darüber hinaus unnötigen bürokratischen Aufwand, der sich durch digitale Abläufe reduzieren ließe. Dementsprechend beauftragt uns das Berufskolleg eine Lösung für Ihre Antrags- und Formularbearbeitung zu entwickeln.

### Soll-Zustand

Kundenseitig bestehen keine konkreten Anforderungen an die Lösung. Es wurde jedoch darum gebeten, das System zu vereinheitlichen und zu vereinfachen. Ob das zum Beispiel durch eine eigene Cloud oder Datenbank geregelt werden soll, wurde nicht erwähnt.

Das Projektteam schlägt eine Stand-Alone-Lösung vor. Diese verwaltet alle Prozesse und Formularvorlagen an einem zentralen Ort. Prozesse sind konkrete Abläufe die beschreiben, wie ein Antrag ausgefüllt, weitergeleitet und bestätigt werden soll. Die Anträge und Prozesse sollen über ein XML-File beschrieben und digitalisiert werden. XML-Files werden vorgeschlagen, da XML (Extended Markup Language) einfach und schnell erlernbar ist, sodass auch eine einfache Hilfskraft Prozesse schreiben und anpassen kann. Außerdem dient es der Beschreibung des Gesamtprozesses mit all seinen benötigten Daten und Verantwortlichen und gibt gleichzeitig Namenskonventionen für die Datenbank und Vorlagenerstellung vor.

Unser Tool erlaubt es jederzeit weitere XML Files einzulesen. Das bedeutet, dass im Nachhinein immer neue Anträge und Prozesse ins Tool eingebunden werden können.

Zur Papierform kann durch explizite Anweisung in der XML-Datei in Form eines Belegs jederzeit zurückgekehrt werden.

Bestmöglich sollte ein Tool erstellt werden, welches das händische Ausfüllen der Vorlagen ersetzt und die zu prüfenden Daten in einer Datenbank abbilden kann. Dies sollte den Zeit- und Papier- bzw. Laufaufwand deutlich senken.

Der Mehrwert des Kunden entsteht über ein einheitliches System über das die vielen Dokumentenvorlagen verwaltet werden können. Außerdem wird der bürokratische Aufwand maßgeblich gemindert, da zum Beispiel Anträge nicht mehr verloren gehen können.

#### Ziele und Funktionen

Das Hauptziel ist es eine funktionierende Software zu entwickeln, die die oben genannten Ziele erfüllt, ohne dabei in seiner Komplexität den Rahmen zu sprengen. Die Muss-Funktionen legen in diesem Sinne dar, welche Ziele unter allen Umständen zu erreichen sind.

XML-Prozessdateien werden automatisch auf Syntax und Struktur geprüft, ausgelesen und erzeugen daraus das User Interface, das als direkter Ersatz für die ursprünglichen Formulare dient. In diesem erscheinen zum Beispiel die Datumsfelder etc. eines Urlaubsantrags.

Das User Interface ist somit Teil des beschriebenen Prozesses. Der Prozess soll ermöglichen Genehmigungen einzuholen, sowie Zielpersonen für Genehmigungen auszuwählen und wenn Belege von Nöten sind, diesen automatisch auszufüllen und zu speichern.

Eine ausführliche Auflistung der Ziele, einschließlich der Soll- und Kann-Zielen ist im Anhang unter dem Reiter Pflichtenheft einsehbar.

#### Grobe Zeitplanung

Die Zeitplanung die wir im Projektantrag erfasst haben, dient als Struktur für das ganze Projekt. Auf der Abbildung 1 - Zeitplanung basiert auch das im späteren Verlauf folgende Gantt-Diagramm.

|  |  |  |
| --- | --- | --- |
| Phase | Tätigkeit | Aufwand in Stunden |
| **Anforderungsanalyse** |  | **11** |
|  | Kundengespräche | 2 |
|  | Erstellung von der Projektstruktur | 6 |
|  | Erstellung des Projektantrages | 3 |
| **Planung** |  | **20** |
|  | Informieren über Frameworks | 3 |
|  | Erstellung von UML-Diagrammen | 8 |
|  | Erstellung von Basisklassen | 2 |
|  | GUI planen | 7 |
| **Umsetzung** |  | **23** |
|  | XML-Interpreter schreiben | 10 |
|  | UI Handler schreiben | 13 |
| **Testen** |  | **9** |
| **Dokumentation** |  | **6** |
| **Übergabe** |  | **1** |
|  |  | **70** |

Abbildung 1 - Zeitplanung

# Analyse- und Planungsphase

## Allgemeiner Ablauf der Phase

Die Phase wurde erfolgreich in der angegebenen Zeit umgesetzt. Es gab wenig bis kaum Probleme in der Aufstellung der Konzepte. Die Kundengespräche verliefen produktiv.

Allerdings wurde in der Analyse- und Planungsphase ein Problem übersehen, welches uns in der Durchführungsphase unter nicht unerheblichen Druck setzte. Der Aufwand für die netzwerkseitige Implementation von Sicherungsmechanismen für die personenbezogenen Daten wurde unterschätzt. Dieser Ungenauigkeit lässt sich auf die fehlende Erfahrung des Teams in Bezug auf Projekte mit personenbezogenen Daten zurückführen.

## Terminplan

### Gantt-Diagramm

In der unteren Abbildung 2 - Gantt - Zeitplanung sehen wir die zeitliche Terminierung unseres Projektes bezogen auf die einzelnen Vorgänge der Phasen. Zu beachten ist, dass wir wegen der Größe des Projektes während der Durchführungsphase parallel arbeiteten. Das ermöglichte uns die Teilaufgaben schneller und effizienter umzusetzen und nur bei komplexen Problemen die Kräfte zu bündeln.

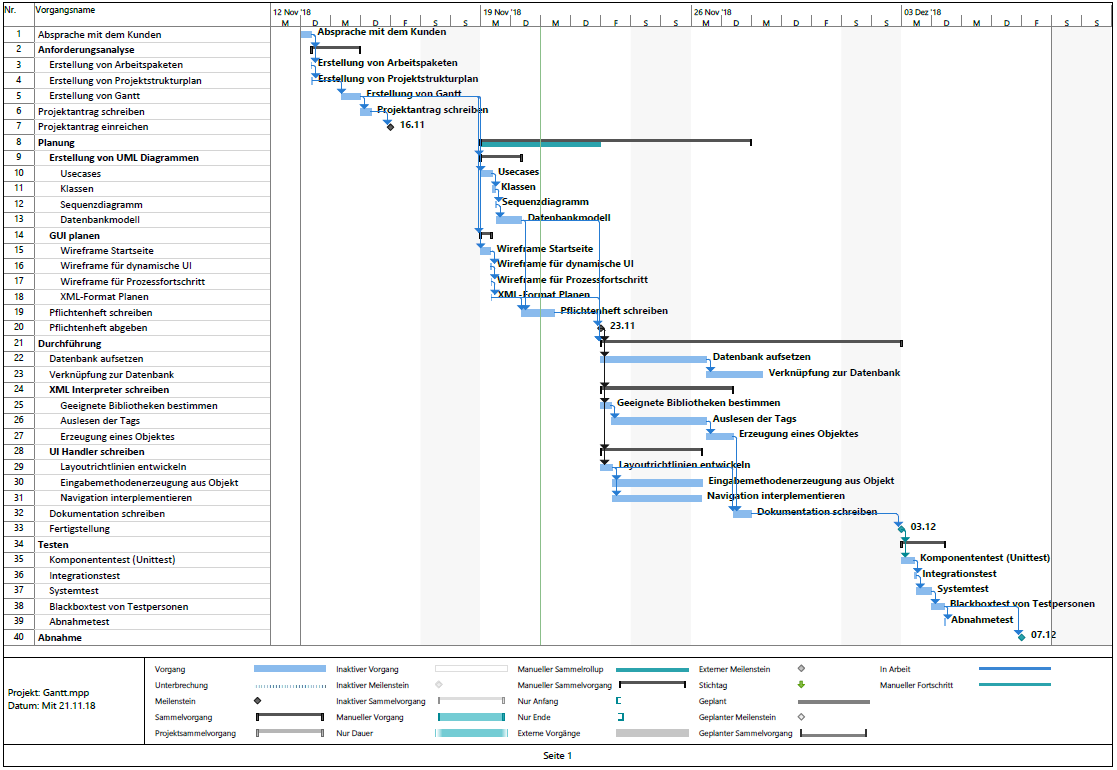


Abbildung 2 - Gantt – Zeitplanung

## Kostenplanung

In der unteren Abbildung 3 - Kostenplan sehen wir die Aufstellung der Kosten für unser Projekt. Die Personalkosten der beiden Mitarbeiter (Bauer & Nordhues) belaufen sich auf knapp 60% der Gesamt-kosten. Die restlichen 40% wurden in Betriebsmittel investiert. Diese sind in einer Gemeinkostenpauschale abgedeckt, die die Nutzung von PCs, Räumen und Lehrkräften einschließt. Als gesonderte Position ist unter den Betriebsmitteln Lizenzen für die Entwicklerumgebung Rider und die Office Produkte ausgewiesen.

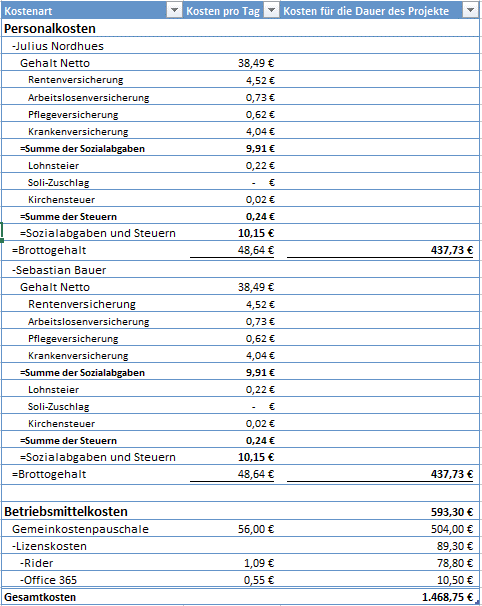


Abbildung 3 – Kostenplan

## Sachplan

In der Abbildung 4 - Sachplan wird gezeigt und erläutert, warum wir welche Techniken oder Tools genutzt haben. Außerdem wird kurz eingegangen, warum wir keine Alternativen genutzt haben.

|  |  |  |
| --- | --- | --- |
| Ergebnis | Alternative | Entscheidung |
| Rider | Visual Studio | Bessere Code Analyse, Code Editing und Refactoring vereinfachen den Arbeitsprozess deutlich. Ausserdem ist die GIT Integration etwas gelungener als bei Visual Studio. |
| C# | Java | Team hat größere Erfahrungen in .NET als in JAVA und kann damit komplexe Probleme schneller lösen. |
| GitHub | SourceForge | GitHub hat sich über Jahre hinweg als effiziente und sichere Lösung erwiesen, während die Alternative zuletzt mit Malwareproblemen zu kämpfen hatte. |
| Eigene Datenbank | ATIW-Datenbank | Durch den Aufbau einer eigenen Datenbank konnten wir die Integrität der bestehenden Daten in der Entwicklung sicherstellen und reduzierten die Einarbeitungszeit in die bestehende Struktur. |
| MySQL | mssql | Die Einrichtung im Netzwerk gestaltete sich als deutlich einfacher. |
| WPF | Windows Forms | WPF bietet eine neuere Bibliothek mit viel reichhaltigeren Visualisierungsmöglichkeiten und höherer Anpassbarkeit bei aufgeräumterer Code-Basis. |
| PlantUML | Microsoft Visio | Wegen der Erstellung von UML Diagrammen in einer eigenen DSL (Domain Specific Language). Damit sind UML Diagramme leicht über Versionsverwaltungen austauschbar und der Formatierung wird automatisch Sorge getragen. |

Abbildung 4 – Sachplan

# Durchführungsphase

## Allgemeiner Ablauf der Phase

Die Durchführungsphase war intensiv, aufgrund der, im Reiter Allgemeiner Ablauf der Phase erwähnten, unterschätzten Sicherheitsvorkehrungen. Im Laufe dieser Phase wurde zudem das Projekt bezüglich der Klassenstruktur deutlich komplexer als angenommen. Wir konnten den Zeitplan dennoch gut einhalten.

## Ablauf der Implementation

Die Implementation verlief Großteil nach Plan. Es wurde parallel gearbeitet und der XML-Interpreter, sowie der UI Handler wurden früh fertiggestellt.

Im ersten Schritt wurden ein XML-Interpreter, Objekte zur Abbildung der XML-Datei im Programm sowie ein UI-Handler entwickelt, der anhand der Ergebnisse des Interpreters eine WPF Formular-Benutzeroberfläche erzeugen kann. Neben den gelegentlichen Änderungen und Anpassungen folgte im nächsten Schritt die Entwicklung einer grundlegenden REST-Schnittstelle, die uns mit einigen Problemen konfrontierte, die aber durch Zeiteinsparungen bei der UI Erzeugung nicht schwer ins Gewicht fielen. Dem Grundsystem folgte die Entwicklung einer Datenbank und der Schnittstelle zu dieser, sowie die einhergehende Erweiterung der REST-Schnittstelle.

Nachdem sich das erste REST-Konzept aber als zu komplex herausstellte, musste der Ansatz verworfen und neu entwickelt werden. Das Ergebnis war im Gegenzug simpler und weniger fehleranfällig als sein Vorgänger.

Zuletzt wurde die provisorische UI einer ausführlichen Neugestaltung unterzogen, um sie für die alltägliche Nutzung vorzubereiten.

## Ergebnis der Implementation

Es wurde ein lauffähiges und funktionierendes Programm erstellt, das den Muss-Zielen/Funktionen entspricht. Es ist nun möglich, alle Anträge und Formulare in einem zentralen Tool zu verwalten und auch zu nutzen. Die Erstellung von Prozessen wird in dem Benutzerhandbuch ausführlich erklärt, damit Hilfskräfte neue Anträge einpflegen können.

## Ablauf der Abnahme

Die Abnahme war auf den 06.12.2018 datiert. Der Abnehmer sollte der Kunde, in unserem Fall vertreten von Herrn Böhmer, sein. Da dieser jedoch krankheitsbedingt ausfiel, wurde er durch die Kollegin Frau Wibbe vertreten. Im Verlauf ab es kleine technische Schwierigkeiten, die allerdings schnell behoben wurden.

Das Projekt wurde mit der Anmerkung abgenommen, dass die Funktionen zur Berechnung von Urlaubstagen überarbeitet werden sollte.

## Projekt-Tagebuch

Das Projekttagebuch wurde aufgrund der Größe gesondert in den Anhang gehängt. Die unter dem Punkt Abweichung in der Zeitplanung aufgeführten Daten, stammen aus dem Projekttagebuch und dem zugrundeliegendem Gantt-Diagramm.

In der unteren Abbildung sehen wir, wie die Tagebucheinträge, welche die Projektablauf-dokumentation darstellen, aussehen.

|  |  |  |
| --- | --- | --- |
| Projekt-Tagebuch (26.11.2018) | | |
| Ressource: | Sebastian Bauer | Julius Nordhues |
| Aufgabenbeschreibung | * Vererbungskonzept erstellt und umgesetzt * Dynamischen UI-Handler schreiben | * XML-Interpreter schreiben * Objekte mit XML-Interpreter verknüpfen |
| Anmerkung |  |  |

Abbildung 5 - Tagebucheintrag vom 26.11.2018

# Testphase

## Allgemeiner Ablauf der Phase

Testfälle wurde parallel zur Implementierung durch einen anderen Entwickler erstellt. Die Testfälle hatten folgende Schwerpunkte:

* Berechtigungssystem
* Datenbankzugriffe
* XML-Interpretation und Fehler Prävention
* UI Aufbau
* Prozessablauf
* User Komfort und Freundlichkeit

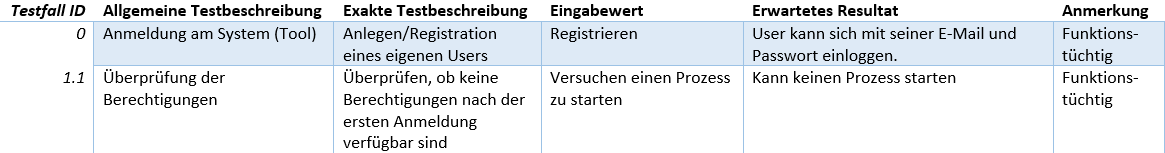
Die ausgeführten Testfälle werden im Anhang aufgeführt. Beispielsweise wird unten ein Testfall aufgeführt.



## Ablauf des Testens

Das Testen wurde mit der Testtabelle aus dem Anhang ausgeführt. In dem Feld Anmerkung werden die Ergebnisse (Funktionstüchtig, Fehlerhaft) festgehalten oder weitere Verbesserungsvorschläge niedergeschrieben.

Unten sehen wir eine Abbildung zu der Auswertung eines Testfalles.



Die weiteren Testfälle und Auswertungen werden in den Anhang gehängt.

# Grundlegende technische Konzepte:

## Der Webserver:

Der Webserver wurde in Form eines einfachen http-Servers implementiert, der seine Dienste über eine REST-Schnittstelle bereitstellt. Bei REST-Schnittstellen werden alle für eine Anfrage benötigten Informationen wie Nutzernamen und Passwörter bei der Übertragung mitgesendet, um auf die Verwaltung aktiver User-Sessions verzichten zu können.

Dabei arbeitet die REST-Schnittstelle mit sogenannten DTOs (Data Transfer Objects). Dabei handelt es sich um eine Development-Pattern bei der für die Übertragung von Daten ein Objekt, speziell zugeschnitten auf seinen Anwendungszweck erstellt wird. Jede Anfrage an den und Antwort vom Server hat also ein eigenes DTO, das alle benötigten Daten beinhaltet.

Die REST Schnittstelle stellt dabei folgende Einstiegspunkte bereit:

|  |  |  |  |
| --- | --- | --- | --- |
| Url | Beschreibung | Methode | Permission |
| **/ProcessTemplate/{message}** | Gibt den XML-Text einer Prozessvorlage, nach Angabe der Id zurück. | GET | 1 |
| **/ProcessTemplate** | Sendet eine Prozessvorlage an den Server. | POST | 2 |
| **/ProcessTemplateList** | Gibt eine Liste aller aktuell bestehenden Prozessvorlagen mit Id und Description zurück (ohne den XML-Text). | GET | 1 |
| **/DocumentTemplate/{message}** | Gibt eine DocX-Vorlage nach Angabe ihrer ID zurück. | GET | 1 |
| **/DocumentTemplate** | Sendet eine DocX-Vorlage an den Server. | POST | 2 |
| **/DocumentTemplateList** | Gibt eine Liste der Namen aller aktuell bestehenden DocX-Vorlagen zurück. | GET | 1 |
| **/Entries/{message}** | Gibt eine Liste aller Einträge zur Id einer bestimmten Prozessinstanz zurück. | GET | 1 |
| **/Entry** | Sendet einen Eintrag für ein bestimmtes Feld einer bestimmten Prozessinstanz an den Server. | POST | 1 |
| **/EntryUpdate** | Aktualisiert einen bestehenden Eintrag anhand seiner Id. | POST | 1 |
| **/ProcessCreate** | Sendet Anfrage zur Erstellung einer neuen Prozessinstanz. | POST | 1 |
| **/ProcessInstance/{message}** | Gibt eine Prozessinstanz anhand ihrer Id zurück. | GET | 1 |
| **/ProcessInstanceList** | Gibt eine Liste aller bestehenden Prozessinstanzen des anfragenden Nutzers zurück. | GET | 1 |
| **/ProcessUpdate** | Aktualisiert den Status einer Prozessinstanz. | POST | 1 |
| **/ResponsibilityList** | Gibt eine Liste von Einträgen zurück, die ein User noch zu prüfen hat. | GET | 1 |
| **/UserList** | Gibt eine Liste aller Nutzer und ihrer Berechtigungslevel zurück. | GET | 3 |
| **/Level** | Sendet eine Aktualisierung für den Berechtigungslevel eines bestimmten Users. | POST | 3 |
| **/PermissionLevel/{message}** | Holt den Berechtigungslevel eines bestimmten Users, wird kein User angegeben, wird der Level des fragenden Users zurückgegeben | GET | 1 / 3 |
| **/User** | Sendet Registrierungsanfrage für einen neuen User | POST | 0 |
| **/AuthCheck** | Prüft ob ein Satz von Zugangsdaten gültig ist. | GET | 0 |

Gesichert wird das System über mehrere Stufen. Einerseits prüft der Client die Gültigkeit aller Eingaben und zeigt nur Optionen an, zu denen ein Nutzer berechtigt ist, in zweiter Instanz prüft aber auch der Server alle Anfragen, um unberechtigte Zugriffe über Browser oder Posting-Tools zu vermeiden.

## Die Formularerstellung:

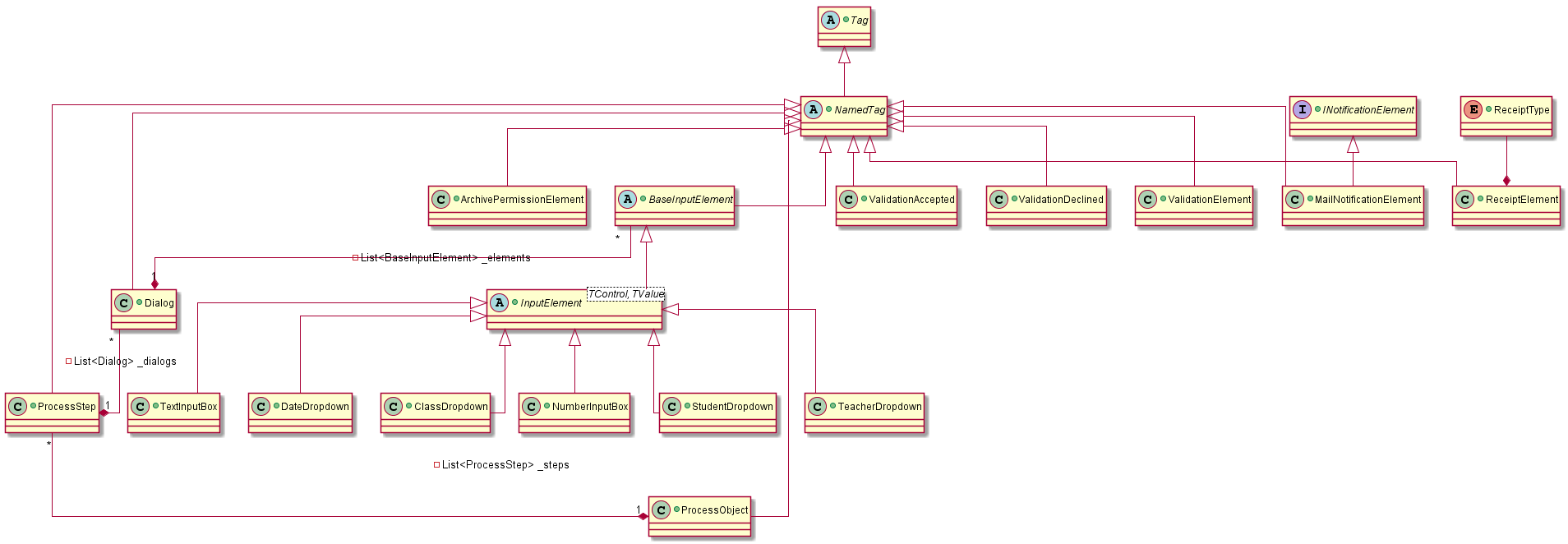
Wie bereits zuvor beschrieben, werden Prozesse im XML-Format formuliert. Die Interpretation der Dateien erfolgt über eine „XmlHelper“ Klasse, die diese auch auf ihre Gültigkeit prüfen kann. Diese Prüfung erfolgt jedoch nur oberflächlich und kann nicht alle Strukturfehler aufdecken.

Aus der Xml-Datei entstehen in Folge Objekte, die einerseits ihre Struktur abbilden und andererseits Hilfsmethoden zur Interaktion mit einzelnen Elementen dieser Struktur anbieten. Zum Beispiel speichert ein Dialog Objekt nicht nur seine Eingabe Objekte, sondern bietet auch Zugriffsmethoden auf deren Berechnungsfunktionen an.

Die Struktur der Klassen zur Abbildung der XML-Datei kann dem auf der nächsten Seite folgenden UML Diagramm entnommen werden.

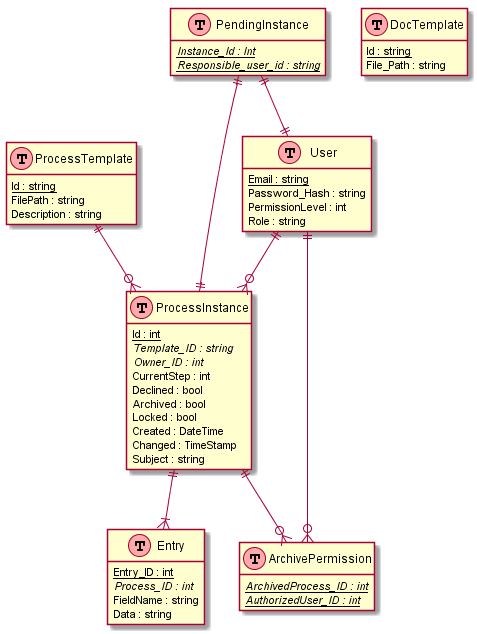
Aus den im Diagramm ersichtlichen Eingabeelementen die von BaseInputElement erben, wird die Formularansicht für den Benutzer erzeugt und seine Eingaben verarbeitet.

ProcessObject hält mehrere Prozess-Schritte die ihrerseits wieder Dialoge, Bestätigungstags etc beinhalten.



## Datenhaltung:

Die Datenhaltung wird über eine MariaDB (MySQL Datenbank) umgesetzt, deren Struktur im folgenden Diagramm zu sehen ist.



# Bewertung

## Allgemeiner Ablauf des Projekts

Das Projekt verlief gut. Das Ergebnis ist ein lauffähiges Programm das vom Kunden abgenommen wurde. Trotz kleiner Fehlplanungen in der Analysephase bezogen auf den Aufwand der Sicherheitsvorkehrungen und der Vererbungsstruktur, wurde das Projekt im Projektzeitraum bearbeitet und fertig gestellt. Weitere Anpassungen sind, bei erneuter Beauftragung durch den Kunden, möglich und werden vom Entwicklerteam begrüßt.

## Abweichung in der Zeitplanung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vorgangsname | Datum | | Ver-zug | Grund |
| Plan | Ist |
| Anforderungsanalyse |  |  |  |  |
| Erstellung von Arbeitspaketen | 13.11.18 | 13.11.18 |  |  |
| Erstellung von Projektstrukturplan | 13.11.18 |  |  |  |
| Erstellung von Gantt | 14.11.18 | 14.11.18 |  |  |
| Projektantrag schreiben | 15.11.18 | 15.11.18 |  |  |
| **Projektantrag einreichen** | 16.11.18 | 16.11.18 |  |  |
| Planung |  |  |  |  |
| Erstellung von UML Diagrammen |  |  |  |  |
| Usecases | 19.11.18 | 19.11.18 |  |  |
| Klassen | 19.11.18 | - |  | Es wurde sich gegen ein Klassendiagramm wegen einer zu hohen Komplexität entschieden |
| Sequenzdiagramm | 19.11.18 | 19.11.18 |  | Es wurde anstelle eines Sequenzdiagramms sich für ein Ablaufdiagramm entschieden, da es eine bessere Darstellung ermöglicht |
| Datenbankmodell | 20.11.18 | 23.11.18 |  |  |
| GUI planen |  |  |  |  |
| Wireframe Startseite | 19.11.18 |  |  |  |
| Wireframe für dynamische UI | 19.11.18 |  |  |  |
| WiIreframe für Prozessfortschritt | 19.11.18 |  |  |  |
| XML-Format Planen | 19.11.18 | 20.11.18 |  |  |
| Pflichtenheft schreiben | 21.11.18 | 21.11.18 |  |  |
| **Pflichtenheft einreichen** | 23.11.18 | 23.11.18 |  |  |
| Durchführung |  |  |  |  |
| Datenbank aufsetzen | 26.11.18 |  |  |  |
| Verknüpfung zur Datenbank | 28.11.18 | 26.11.18 |  |  |
| XML-Interpreter schreiben |  |  |  |  |
| Geeignete Bibliothek bestimmen | 23.11.18 | 23.11.18 |  |  |
| Auslesen der Tags | 26.11.18 | 26.11.18 |  |  |
| Erzeugung eines Objektes | 27.11.18 | 26.11.18 |  |  |
| UI Handler schreiben |  |  |  |  |
| Layoutrichtlinien entwickeln | 23.11.18 |  |  |  |
| Eingabemethodenerzeugung aus Objekt | 26.11.18 |  |  |  |
| Navigation interpretieren | 26.11.18 |  |  |  |
| Dokumentation schreiben | 28.11.18 |  |  |  |
| **Fertigstellung** | 03.12.18 |  |  |  |
| Testen |  |  |  |  |
| Komponententest | 03.12.18 |  |  |  |
| Integrationstest | 03.11.18 |  |  |  |
| Systemtest | 04.12.18 |  |  |  |
| Blackboxtest von Testpersonen | 04.12.18 |  |  |  |
| Abnahmetest | 04.12.18 |  |  |  |
| **Abnahme** | 07.12.18 |  |  |  |

Abbildung 6 - Zeitplan Ist-Soll-Vergleich

|  |  |
| --- | --- |
| Meilenstein | Datum |
| Projektantrag einreichen | 16.11.2018 |
| Pflichtenheft einreichen | 23.11.2018 |
| Fertigstellung eines lauffähigen Programmes | 03.12.2018 |
| Abnahme | 07.12.2018 |

Die Meilensteine wurden alle eingehalten und es kam bezüglich dieser nicht zu Verzögerungen.

## Abweichungen in den Muss-, Soll- und Kann-Ziele

Die Muss-Ziele wurden fast vollständig umgesetzt. Es wurde dennoch in der Abnahme vorgeschlagen, die Urlaubstageberechnung zu verbessern, was vermerkt wurde. Diese Funktion ist in der Entwicklung und könnte innerhalb von ein bis zwei Werktagen nachgeliefert werden. Außerdem haben wir noch keine Lösung für einen Mailserver gefunden, womit wir noch keine E-Mail-Benachrichtigung garantieren können. Dies wird weiter verfolgt, wenn der Kunde wieder im Hause ist.

Von den Soll-Zielen wurde das Rollensystem implementiert, sowie das Uploaden und Aktualisieren von XML-Prozessdateien. Der Prozessfortschritt wurde in der DB hinterlegt und mit dem Frontend verknüpft.

Die restlichen Soll-und Kann-Funktionen wurden noch nicht umgesetzt.

## Abweichungen von Ergebnissen aus der Analyse- und Planungsphase

Das relationale Datenbankschema wurde durch zusätzliche Tabellen erweitert und teilweise umstrukturiert. Außerdem wurde kein Sequenzdiagramm erstellt, wie man aus der Abbildung 6 - Zeitplan Ist-Soll-Vergleich nehmen kann. Stattdessen wurde auf ein Flussdiagramm zurückgegriffen, weil dies den Prozess besser abbilden kann. Es wurde auch kein Klassendiagramm aufgeführt, weil unser Konzept, durch mehrstufige Vererbung und aufwendige Netzwerkprotokolle schwierig darzustellen war und keinen Mehrwert für den Kunden gehabt hätte.

Die deutlichsten Änderungen ergaben sich an der allgemeinen Struktur des Projektes. So war in der Planungsphase davon ausgegangen worden, zwei getrennte Applikationen zu entwickeln die jeweils unabhängige Ressourcen nutzen.

Im abschließenden Ergebnis bestehen jedoch drei Projekte. Eines für die Clientseite, eines für die Serverseite, sowie eine Klassenbibliothek die Klassen für beide Projekte zur Verfügung stellt. Das gilt zum Beispiel für die Klassen zur Abbildung der Datenbankobjekte oder Hilfsklassen wie den XML-Interpreter.

Auch die REST-Schnittstelle musste deutlich überarbeitet werden. In der Planungsphase war angedacht, wenige Schnittstellen bereitzustellen, die aber breite Funktionen abdecken können. Z.B. eine Schnittstelle unter „../fileupload“ die Dateien entgegen nimmt und erst bei deren Empfang entscheidet was bei welchem Dateityp zu tun ist.

Diese Funktionen wurden nun in einzelne Schnittstellen getrennt, so dass es für jede Anfrage eine Schnittstelle gibt, sowie ein Request- und Reply-Objekt, die den Aufbau der Anfragen an und Antworten vom Server bestimmen.

## Abweichungen in der Qualitätskontrolle

Die Allgemeine Komplexität des Auftrags gestaltete sich deutlich größer, als erwartet wurde, speziell die Ziele der einfachen Erweiterbarkeit aber auch der einfachen Antragserstellung trugen maßgeblich dazu bei. Das führte zu deutlichen Umstrukturierungen der grundlegenden Konzepte wie der Vererbungsstruktur der Eingabeelemente und einer komplexen Code-Basis, die nicht unbedacht angefasst werden sollte.

Zur Lösung dieses Problems haben wir schließlich die Code-Basis durch Vererbung in mehrere Ebenen aufgeteilt, von der die oberste weitreichende und verständliche Schnittstellen bereitstellt, so dass das Programm in Teilen einfach erweiterbar bleibt. Im Gegenzug sind die Klassenzusammenhänge der tieferen Ebenen sehr komplex.

Die Erweiterbarkeit des Programmes ist somit gegeben, wird aktuell aber wegen fehlender Dokumentation erheblich erschwert, die in diesem Sinne nachgereicht wird.

Durch ausführliche Tests ist die Korrektheit des Programmes weitgehend gesichert. Die Benutzerfreundlichkeit wurde laufend während der Implementierung verbessert. Z.B. wird sichergestellt, dass der Nutzer möglichst selten in Mitteilungssackgassen läuft oder mit Exception-Meldungen beworfen wird. Stattdessen wird er klar über Fehler aufgeklärt und sollte das nicht möglich sein, kann ein Administrator, da die Fehler nahezu alle Serverseitig auftreten, diese mit Uhrzeit im Serverlog nachverfolgen. Dies ermöglicht uns eine gute Benutzerfreundlichkeit zu schaffen, weil der Nutzer selber weiß, wo der Fehler ist und ihn auch verbessern kann.

## Abweichungen in der Kostenplanung

Es gibt keine Abweichungen in der Kostenplanung, da das Projekt zum Stichtag erfolgreich abgenommen wurde und keine zusätzlichen Ressourcen bezüglich zusätzlicher Arbeitskräfte, -stunden oder Lizenzen benötigt wurden.

## Mehrwert geschaffen?

Frau Wibbe hat bei der Abnahme deutlich gemacht, dass Sie mit den Resultaten zufrieden ist. Von dieser Aussage leiten wir ab, dass wir den Kunden zufrieden gemacht haben. Dieses Tool bietet die Möglichkeit den Alltag von Lehrpersonal deutlich zu vereinfachen, was einen klaren Mehrwert für das Lehrpersonal und das Betriebsklima schafft.

# Benutzerhandbuch



# Anhang

## Pflichtenheft



## Projekttagebuch



## Testprotokolle