Entwicklung und Implementierung von UI-Widgets zur Erweiterung eines bestehenden Formular-Editors

Individuelle Praktische Arbeit (IPA)

Kandidat:

Jakov Ratkovic Startdatum:

Bleikenstrasse 87 03.03.2020

9630 Wattwil

0774379449 Enddatum:

Jakov.ratkovic18@gmail.com 16.03.2020

Fachvorgesetzter: Betrieb:

Kay Stenschke gyselroth GmbH

stenschke@gyselroth.com Albisriederstrasse 226

8047 Zürich, Schweiz

Hauptexperte: +41 44 404 22 44

Wilfried Krug mail@gyselroth.com

[Wilfried@krug.cc](mailto:Wilfried@krug.cc)

Inhaltsverzeichnis

[1 Dokumenteninformationen 4](#_Toc34131955)

[1.1 Versionen 4](#_Toc34131956)

[1.2 Typografische Richtlinien 4](#_Toc34131957)

[1.3 Allgemeine Informationen 4](#_Toc34131958)

[2 IPA-Bericht Teil 1 Umfeld und Ablauf 5](#_Toc34131959)

[2.1 Aufgabenstellung 5](#_Toc34131960)

[2.2 Projektaufbauforganisation 10](#_Toc34131961)

[2.3 Deklarationen 10](#_Toc34131962)

[2.3.1 Vorkenntnisse 10](#_Toc34131963)

[2.3.2 Vorarbeiten 10](#_Toc34131964)

[2.3.3 Firmenstandards 10](#_Toc34131965)

[2.3 Deklarationen 11](#_Toc34131966)

[2.3.1 Vorkenntnisse 11](#_Toc34131967)

[2.3.2 Vorarbeiten 11](#_Toc34131968)

[2.3.3 Firmenstandards 11](#_Toc34131969)

[2.4 Zeitplan 12](#_Toc34131970)

[2.5 Arbeitjournal 13](#_Toc34131971)

[2.5.1 Dienstag 03.03.20 13](#_Toc34131972)

[2.5.2 Mittwoch 04.03.20 14](#_Toc34131973)

[2.5.3 Donnerstag 05.03.20 15](#_Toc34131974)

[2.5.4 Freitag 06.03.20 16](#_Toc34131975)

[2.5.5 Montag 09.03.2020 17](#_Toc34131976)

[2.5.6 Dienstag 10.03.20 18](#_Toc34131977)

[2.5.7 Mittwoch 11.03.20 19](#_Toc34131978)

[2.5.8 Donnerstag 12.03.20 20](#_Toc34131979)

[2.5.9 Freitag 13.03.20 21](#_Toc34131980)

[2.5.10 Montag 16.03.20 22](#_Toc34131981)

[2.6 Organisation der Arbeitsergebnisse 23](#_Toc34131982)

[2.6.1 Versionierung 23](#_Toc34131983)

[2.6.2 Aufbau des Arbeitsplatzes 23](#_Toc34131984)

[2.7 Projekmanagementmethode 24](#_Toc34131985)

[2.8 3 Zeiten-Methode 25](#_Toc34131986)

[3 IPA-Bericht Teil 2: Projekt 26](#_Toc34131987)

[3.1 Kurzfassung 26](#_Toc34131988)

[3.1.1 Kurze Ausgangssituation 26](#_Toc34131989)

[3.1.2 Umsetzung 26](#_Toc34131990)

[3.1.3 Ergebnis 26](#_Toc34131991)

[3.2 Informieren 27](#_Toc34131992)

[3.2.1 System und Umfeld 27](#_Toc34131993)

[3.2.2 Umgebung 27](#_Toc34131994)

[3.2.3 Aufgabenstellung und Kriterienkatalog analysieren 29](#_Toc34131995)

[3.2.3 Mit Teil.1 der IPA-Dokumentation beginnen 30](#_Toc34131996)

# 1 Dokumenteninformationen

## Versionen

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Bearbeitet am | Bearbeitet von | Bemerkung |
| 0.1 | 03.03.2020 | Jakov ratkovic | Dokument erstellt |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabelle 1 Versionen des Dokuments

## Typografische Richtlinien

|  |  |  |  |
| --- | --- | --- | --- |
| Formatvorlagen | Schriftart | Schriftschnitt | Grösse |
| Überschrift 1 | Kings Caslon Display Typo | Normal | 18 |
| Überschrift 2 | Kings Caslon Display Typo | Normal | 16 |
| Überschrift 3 | Kings Caslon Display Typo | Normal | 14 |
| Fliesstext | Locator Light | Normal | 11 |

Tabelle 2 Typografische Richtlinien

## Allgemeine Informationen

|  |  |
| --- | --- |
| Dokumenttitel | Entwicklung und Implementierung von UI-Widgets zur Erweiterung eines bestehenden Formular-Editors |
| Author | Jakov Ratkovic |
| Dateiname | IPA\_2020\_Bericht\_Jakov\_Ratkovic.docx |
| Ablageort | Um meine Daten abzusichern werden die Dokumente mit der 3-2-1 Backup Regel abgesichert unter folgenden Orten  Onedrive  Lokal auf Laptop  USB-Stick |

Tabelle 3 Allgemeine Informationen

# IPA-Bericht Teil 1 Umfeld und Ablauf

## Aufgabenstellung

(Originaler Text aus pkorg)

Titel der Arbeit

Entwicklung und Implementierung eines UI-Widgets und Formular-Elementes zur Erweiterung eines bestehenden Formular-Editors

Ausgangslage

Die Firma gyselroth entwickelt und supported die Applikation Intranet Sek II, eine zentrale Plattform zur Datenverwaltung an Mittel- und Berufsfachschulen im Kanton Zürich. In der Applikation können Benutzer verschiedener Berechtigungsgruppen sich einloggen, Daten erfassen, bearbeiten und Ansehen.

Teil der Applikation ist ein Formular-Editor: Redaktoren können einfache Formulare Erstellen und den Anwendern der Plattform zur Verfügung stellen. Formulare werden z.B. im Rahmen von Umfragen oder zur Ermöglichung von vorgefertigten Anfragen in der Applikation genutzt.

Bislang können Formulare ausschliesslich aus einfachen HTML-Form Elementen bestehen.

Ziel dieser Aufgabe ist die Implementierung eines erweiterten Kalender-Widgets:

Zur Steigerung der Benutzerfreundlichkeit sollen zusätzlich zur eigentlichen Datums- und Zeitraum-Wähler Funktion des Formular-Elementes, gewählte personen-spezifische Informationen am betreffenden Zeitpunkt dynamisch dargestellt werden.

Detaillierte Aufgabenstellung

Von Redaktoren erstellte Formulare sollen neu ein erweitertes Kalender-Element beinhalten können:

Zur Steigerung der Benutzerfreundlichkeit sollen in diesem Element alleinstehend oder zusätzlich (d.h. zur Nutzung als Formular- oder informatives Element) zur eigentlichen Datums- und Zeitraum-Wähler Funktion des Formular-Elementes, gewählte personen-spezifische Informationen am betreffenden Zeitpunkt dynamisch darstellt werden.

Die Datenquelle der anzuzeigenden Zusatz-Information soll vom Redaktor bei Erstellung/Bearbeitung des Formulars festlegbar sein (Quellen: Feiertage und Schulanlässe, Personen/Klassen-Lektionen, Raum-Belegungen).

Das Widget soll in der veröffentlichten Darstellung zusätzlich zur Basis-Funktionalität (Kalender mit Option zum Wählen eines Datums oder Zeitraums) auch personen-spezifische Datums-basierte Daten des eingelogged'ten Endanwenders darstellen.

Ein geplanter Einsatzzweck des neuen Elementes ist die Ermöglichung von Urlaubsanträgen auf Basis eines vorgefertigten Formulars. Das Element soll potentiell auch in anderen Formularen, z.B. zur Erfassung von Abmeldungen in Krankheitsfällen u.a. nutzbar sein.

Die angezeigten Zusatz-Informationen sollen beim Ausfüllen des Formulars optional\* interaktiv nutzbar sein: bei Anwahl der dargestellten Daten, sollen diese markiert werden und die betreffenden Felder im Formular entsprechend befüllt werden.

\*optional = Nicht-interaktiv kann das Element als Referenz/Information eingesetzt werden, z.B. um mit der Belegung anderer Elemente der selben Datenquelle (Personen/Räume/etc.) zum selben Zeitpunkt/Zeitraum vergleichen zu können.

Schritte zur Umsetzung:

Planung/Konzeption:

• Erstellen eines Flow-Diagramms als UML, um Operationen und benötigte Funktionalitäten, sowie zugehörige Privilegien zu identifizieren

• Ermitteln der notwendigen Parameter welche im Formular-Editor gesetzt werden können müssen

• UI-Konzeption des Widget-Elements im Formular-Editor (Verwendung durch Redaktoren)

• UI-Konzeption des Form-Elementes (Verwendung durch Endanwender)

• Analyse der bestehenden Datenbank-Struktur und Planung allfällig notwendiger Erweiterungen/Anpassungen

Programmierung:

• Programmierung (MVC) des Widgets im Formular-Editor zur Verwendung durch berechtigte Nutzer (Administratoren)

• Programmierung (MVC) der Anzeige/Funktion des Formular-Elementes zur Verwendung durch Endanwender

-> Es wird erwartet dass der Programm-Code den Code-Conventions (Kommentierung, Namensgebung, Formatierung, Separation of Concerns (siehe Dokumentenpool)) der Firma entspricht

-> Es wird Planung und Implementierung von Fehlerbehandlung erwartet (siehe Dokumentenpool)

• Konfiguration von ACL-Privilegien und Implementierung der zugehörigen Prüfungen

-> Sicherheit: Es wird erwartet dass die serverseitigen Komponenten erhaltene Formulardaten vor Weiterverarbeitung auf erlaubte Datentypen und/oder -Werte prüfen bzw. umwandeln/filtern, und bei nicht verarbeitbaren Werten die Operation zurückweisen

-> Sicherheit: Es wird erwartet dass Requests welche SQL-Injections enthalten, von Controllern nicht ungefiltert an Models übergeben werden

-> Sicherheit: Es wird erwartet dass Requests welche XSS-Payloads enthalten nicht ungefiltert an Models übergeben werden

-> Sicherheit: Es wird erwartet dass Daten aus der Datenbank welche XSS-Payloads enthalten nicht unescaped in Views ausgegeben werden

Testing:

• Manuelles Testen des Elementes im Formular-Editor, sowie des Elementes im Formular

• Prüfen der Daten-Speicherung durch Endanwender und der an den Redaktor/das System zurückgelieferten verarbeiteten Daten

• Sicherstellen dass über das Endanwender-Formular übermittelte Daten im Backend den Administratoren korrekt angezeigt werden

• Sicherstellung von ACL zur Verwendung des Widgets im Formular-Editor, sowie im veröffentlichten Formular

Technische Dokumentation:

• Erfassen eines Test-Protokolls zur Qualitätssicherung des Ergebnisses, sowie für die zukünftige Wahrung durch manuelles Testing nach Anpassungen am Formular-Editor oder verwendeter Core-Komponenten, negative (Quer-) Effekte auf dieses Element vor Releases ausschliessen zu können.

Versionierung und erwarteter Stand des technischen Ergebnisses

• Es wird erwartet, dass alle Arbeiten der Programmierung im VCS (SVN) der Firma gyselroth, entsprechend der Firmenstandards (siehe Dokumentenpool) versioniert sind

• Der Feature-Branch muss bei Abschluss der Arbeit zusätzlich zum Programmcode, sämtliche zur Inbetriebnahme notwendigen Daten (Konfigurationen, ACL-Privilegien und Datenbank-Delta Files) beinhalten

Firmenstandards zu Formatierung, Kommentierung, Namensgebung, Separation of Concerns, VCS bei Entwicklung von Features Fehlerbehandlung - Siehe "gyselroth\_standards.pdf" im Dokumentenpool

Mittel und Methoden

Hardware: Laptop mit Linux Mint

Planungs-Tools: Wireframing Tool z.B. Balsamiq

Entwicklungs-Tools: PhpStorm IDE, DataGrip (Database IDE), Subversion Versionskontrolle, Commandline/Bash

Verwendete Programmiersprachen/Libraries und Systeme: PHP 7.4, Zend-Framework 1, ggf. Dependency-/Autoloading-Management via Composer, MariaDb Datenbank, TypeScript/JavaScript, KendoUI Framework, SASS/CSS, HTML

Software zum Betrieb/Testing: Chromium und Firefox Web-Browser, Enthaltene Entwickler-Tools/Konsolen, ggf. CLI/cURL

Vorkenntnisse

In den letzten 6 Monaten hat Jakov im Rahmen des betrieblichen Praktikums unter Anleitung Arbeiten ausgeführt, welche den Inhalten dieser IP-Arbeit ähneln.

Die Aufgabe enthält keine dem Kandidaten gänzlich neuen Tätigkeiten/Produkte/Techniken.

Vorarbeiten

Kennenlernen des bestehenden Formular-Editors und dessen Funktionalität zur Veröffentlichung, Nutzung und Auswertung von Formularen. Review und allfällig Erweiterung der bestehenden Tests des Formular-Editors, um Impediments (d.h. Hindernisse, welche nicht zum eigentlichen Inhalt dieser IPA gehören) ausschliessen zu Helfen, sowie die zu erweiternden Funktionalitäten vor der Arbeit ausreichend zu kennen.

Neue Lerninhalte

Die planerischen- und Anforderungen bzgl. Programmierung und Testing/Qualitätssicherung der Aufgabe sind inhaltlich bekannt. Neu ist die grösstenteils (bisauf allfällig notwendige Absprache bei Unklarheiten) selbständige Planung und Umsetzung.

Arbeiten in den letzten 6 Monaten

-Angeleitetes/Kontrolliertes Bugfixing (PHP, Type/JavaScript, KendoUI, SCSS) im Rahmen der Weiterentwicklung und des Kundensupports der Applikation "Intranet Sek II" (diverse Module)

-Durchführung von Modernisierungs/Homogenisierungs-Massnahmen am Legacy-Code der Applikation "Intranet Sek II" nach Vorgaben zur Qualitätssicherung (primär PHP, TypeScript/KendoUI)

-Angeleitete/kontrollierte Umsetzung von Verbesserungen und neuen Features am bestehenden UI der Applikation "Intranet Sek II", unter Verwendung von TypeScript, JavaScript, KendoUI, SCSS

-Testing und Pflege von firmen-eigenen PHP Vendor-Packages (PHP) welche in der Applikation verwendet werden

-Regelmässiges manuelles und z.T. Programmierung automatisierten Testings der "Applikation Intranet Sek II" (v.A. in Staging-Phasen)

-Implementierung von PhpCs Code-Sniffs zur Qualitätssicherung

-Vereinzelte Präsentationen zu verschiedenen Technologie-Standards im Rahmen der weekly Meetings im Developer-Team

## 2.2 Projektaufbauforganisation

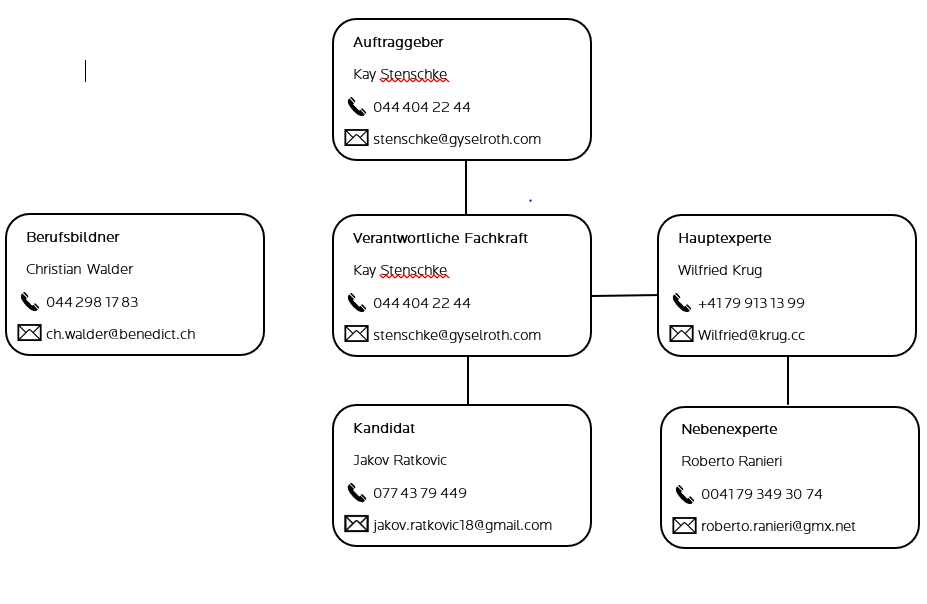


Abbildung 1 Projektaufbauorganisation

|  |  |  |
| --- | --- | --- |
| Personen | Rollen | Aufgaben/Verantwortung |
| Kay Stenschke | Auftraggeber /  Verantwortliche Fachkraft | Erstellung der PA-Planung  Erstellung der detaillierten Aufgabenstellung  Prüfung des Berichts auf fachliche Richtigkeit  Kanditat bei Fragen unterstützen |
| Wilfried Krug | Hauptexperte | Festlegung der Besuchstermine  Führung des Fachgesprächs  Bewertung der Arbeit  Prüfung der korrekten Durchführung des Projekts |
| Roberto Ranieri | Nebenexperte | Protokollierung des Fachgesprächs  Unterstützung des Hauptexperten bei der Bewertung |
| Jakov Ratkovic | Kandidat | Durchführung der IPA  Einhaltung der Rahmenbedingungen  Korrekte Ausführung des Projekts innerhalb des vorgegebenen Zeitrahmens |
| Christian Walder | Berufsbildner | Fachkompetente Ausbildung der Schüler |

## 2.3 Deklarationen

### 2.3.1 Vorkenntnisse

Die folgenden Vorkenntnisse habe ich mir während 2 Jahren in der Schule sowie im Praktikum angeeignet

### 2.3.2 Vorarbeiten

### 2.3.3 Firmenstandards

Die Firmenstandards werde ich hier nicht aufführen, sie können in einem separaten Dokument gyselroth\_standards.pdf, welches sich im Dokumentenpool von pkorg befindet, nachgelesen werden.

Tabelle 4 Projektaufbauorganisation

## 2.3 Deklarationen

### 2.3.1 Vorkenntnisse

Die folgenden Vorkenntnisse habe ich mir während 2 Jahren in der Schule und während des betrieblichen Praktikums angeeignet.

|  |  |  |  |
| --- | --- | --- | --- |
| Technologie | Kenntnisse | Kenntnisse angeeignet | Bemerkung |
| PHP | sehr Gut | Schule, Praktikum | Bugfixes, und Erweiterung des Intranets |
| Javascript | Gut | Praktikum | E2E Tests erweitern |
| Typescript | Gut | Praktikum | Erweiterung und Anpassung der In2 Frpntends |
| MySQL | Grundlagen | Schule | Übungsaufgaben |
| SVN | Gut | Praktikum | Versionierung des In2 |
| Git | Grundlagen | Praktikum | Versionierung der E2E Tests sowie die PHP-Helper Denpendency |
| PHPStorm | Gut | Praktikum | Tägliche Anwendung |
| ZendFramework | Grundlagen | Praktikum | Anwendung einer Komponenten wie z.B ZendView |
| Docker | Grundlagen | Praktikum | Tägliche Anwendung um Projekt lokal zu starten |
| HTML | Grundlagen | Schule, Praktikum | Anpassungen im Intranet sowie kleinere statischen Websiten |
| CSS | Grundlagen | Schule, Praktikum | Styling von Websiten |
| Security | Grundlagen | Schule, Praktikum | Sicherstellung der Integrität der Daten und Fremdzugriff absichern. |

### 2.3.2 Vorarbeiten

|  |  |
| --- | --- |
| Bereich | Bemerkung |
| Codestandard PSR-2 | Über Codestandard informiert |
| MVC | Wissen über die einzelnen Schichten erweitert/aufgefrischt |
| ACL | Implementierung von Rechten angeeignet |
| Kendo UI | Eingelesen in die API-Doc von Kendo UI |
| MongoDB | Einlesen in NO-SQL Datenbank |

### 2.3.3 Firmenstandards

In meiner Arbeit werde ich mich nach den Standards im Dokument gyselroth DEV Standards richten, welches im Dokumentenpool von pkorg zu finden ist.

## 2.4 Zeitplan

## 2.5 Arbeitjournal

### 2.5.1 Dienstag 03.03.20

|  |
| --- |
| Tag 1 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

### 2.5.2 Mittwoch 04.03.20

|  |
| --- |
| Tag 2 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

### 2.5.3 Donnerstag 05.03.20

|  |
| --- |
| Tag 3 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

### 2.5.4 Freitag 06.03.20

|  |
| --- |
| Tag 4 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

### 2.5.5 Montag 09.03.2020

|  |
| --- |
| Tag 5 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

### 2.5.6 Dienstag 10.03.20

|  |
| --- |
| Tag 6 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

### 2.5.7 Mittwoch 11.03.20

|  |
| --- |
| Tag 7 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

### 2.5.8 Donnerstag 12.03.20

|  |
| --- |
| Tag 8 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

### 2.5.9 Freitag 13.03.20

|  |
| --- |
| Tag 9 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

### 2.5.10 Montag 16.03.20

|  |
| --- |
| Tag 10 |
| Ausgeführte Arbeiten |
|  |
| Aufgetretene Probleme |
|  |
| Erfolgreiche oder erfolglose Tests |
|  |
| Beanspruchte Hilfestellung |
|  |
| Nacht- und Wochenendarbeiten |
|  |
| Vergleich mit Zeitplan |
|  |
| Reflexion |
|  |

## 2.6 Organisation der Arbeitsergebnisse

### 2.6.1 Versionierung

Damit alle Versionen der Arbeitsergebnisse wiederhergestellt werden können und auf sie zugegriffen werden kann werden die Daten folgendermassen abgesichert:

Dokumentation:

Bei der Dokumentation werde ich die 3-2-1 Backup Regel anwenden, das bedeutet ich habe 3 Datenkopien 2 Medien und 1 externes Backup.

Externes Backup: OneDrive

Medium 1: Lokal auf Laptop

Medium 2: USB-Stick

Source Code:

Der Source Code wird mittels Subversion (SVN) abgesichert was den Zugriff auf alle Versionen sicherstellt. Zusätzlich wird der gesamte Projektordner lokal auf dem Firmenlaptop abgespeichert welcher tagesaktuell ist.

### 2.6.2 Aufbau des Arbeitsplatzes

Der Aufbau meines Arbeitsplatzes besteht aus dem Arbeits-Laptop und einem zusätzlichen externen 27-Zoll-Monitor. Zudem werde ich noch meinen privaten Laptop für die Dokumentation verwenden da ich mit Microsoft Word arbeiten möchte.

Firmen Laptop:

Macbook Pro

Hardware:

Intel Core i7-3615QM CPU @ 2.3GHzx 4

Betriebssystem :

Linux Mint 19.3 Cinnamon

## 2.7 Projekmanagementmethode

Damit ich mein Projekt sauber planen und durchführen kann werde ich die IPERKA-Methode verwenden.

Die Methode eignet sich sehr gut für dieses Projekt da sie durch sechs logisch aufgebaute und leicht einprägbare Schritte der Projektplanung gekennzeichnet ist, und somit ein strukturiertes Vorgehen ermöglicht. Zudem wird der Planungsphase im Vergleich zur Realisierungsphase mehr Gewicht verliehen, da diese in Projekten oft zu wenig Beachtung bekommt.

Informieren:

Im ersten Schritt geht es darum, sich ein Bild über den Auftrag zu machen, Unklarheiten abzuklären und alle benötigten Informationen zu beschaffen, die für das Projekt notwendig sind.

Planen:

Bei der Planung werden verschiedene Lösungswege erarbeitet, die zur Erreichung des Ziels führen, einen Zeitplan mit den auszuführenden Tätigkeiten erstellt, sowie abgeklärt, welche Ressourcen zur Verfügung stehen.

Entscheiden:

In dieser Phase werden die verschiedenen Lösungsvarianten, die bei der Planung erarbeitet wurden, verglichen und die beste Lösung ausgewählt.

Realisieren:

In der Realisierungsphase werden die erstellten Entwürfe und der ausgewählte Lösungsweg unter Einhaltung des Zeitplanes in die Tat umgesetzt.

Kontrollieren:

Funktionalität der Applikation wird anhand der in der Planung erstellten Testfälle getestet und das Ergebnis dokumentiert. Zudem soll kontrolliert werden, ob alle Prüfkriterien erfüllt worden sind und die Umsetzung mit der Planung verglichen wurde.

Auswerten:

In der letzten Phase wird die Projektarbeit reflektiert. Es werden Erfolge, Misserfolge sowie mögliche Verbesserungsvorschläge dokumentiert.

## 2.8 3 Zeiten-Methode

Damit die Festlegung der Zeit von Arbeitspaketen nicht nach Bauchgefühl gemacht werden muss, werde ich die 3 Zeiten-Methode verwenden. Die Methode wird mir helfen mit meiner begrenzten Erfahrung in der Software-Entwicklung eine realistische Zeitschätzung zu machen.



# IPA-Bericht Teil 2: Projekt

## Kurzfassung

### Kurze Ausgangssituation

Die Firma Gyselroth entwickelt und supported die Applikation Intranet Sek II, eine zentrale Plattform zur Datenverwaltung an Mittelschulen und Berufsschulen im Kanton Zürich. In der Applikation Intranet Sek II können unterschiedliche Benutzer verschiedener Berechtigungsgruppen sich einloggen, Daten erfassen, bearbeiten und ansehen.

Ein Teil des Intranet Sek II ist das Form2 Modul. In diesem können berechtigte Personengruppen über ein GUI ein Formular nach ihren wünschen erstellen, veröffentlichen und die Ergebnisse analysieren. Teil der Aufgabe ist den Formulartypen Datepicker zu erweitern so dass beim Endanwender in dem Datepicker Fenster dynamisch Daten wie z.B Belegung der Räume oder Personen/Klassen Lektionen angezeigt werden. Die Datenquelle der Daten die dem Endanwender angezeigt werden können vom Administrator im Editor Bereich definiert werden.

### Umsetzung

Damit das Projekt über eine geeignete Struktur verfügt, wird die IPERKA-Projektmanagement Methode eingesetzt. Die Planung der Zeit werde ich mit Hilfe der 3 Zeiten Methode festlegen. Die Realisierung des Projekts wird auf der Entwicklungsumgebung PHPStorm in einem extra dafür erstellten Feature-Branch umgesetzt. Das Projekt wird mittels der Technologien PHP, JavaScript/TypeScript, CSS und HTML umgesetzt.

### Ergebnis

Das Ergebnis ist die erfolgreiche Erweiterung des Kalender Widgets im Form2 Modul des Intranet Sek II. Folgende Punkte müssen erfüllt sein damit die Erweiterung als erfolgreich zählt:

Formular Administration:

Der Administrator hat die Möglichkeit beim Kalender Widget optional eine Datenquelle auszuwählen.

Formular Endanwender:

Der Endanwender kann die festgelegten Daten im Kalender Widget sehen.

Dies ist eine grobe Beschreibung und es werden nicht detailliert alle Features aufgelistet. Genauere Informationen werden im Testkonzept ersichtlich sein.

## 3.2 Informieren

Ich begann den Tag indem ich zuerst nochmal die Kriterien für die Dokumentation studierte und danach mit dem Teil 1 der Dokumentation startete. Zudem legte ich mir eine klare Ordnerstruktur für die Dokumentation an in dem alle Versionen klar ersichtlich sind. Dadurch war der Grundstein für ein strukturiertes Vorgehen gelegt.

Zudem musste ich kurzzeitig von der IPERKA Methode abweichen, Grund dafür war das die Erstellung eines Zeitplans erst in der planungs-Phase realisiert wird. Daher habe ich den Zeitplan für die erste Phase vorweg schon realisiert.

### 3.2.1 System und Umfeld

|  |  |
| --- | --- |
| Bereich | Version |
| Intranet Sek II (Dev) | 1.1.1 |
| PHP | 7.4.2 |
| PHPStorm | 2019.3.1 |
| Datagrip | 2019.3.3 |
| Zendframework | 1.13 |
| Docker | 19.03 |
| ZF1-Future | 1.16.2 |
| Typescript | 3.8.2 |
| KendoUI | 2019.3.1 |

Die Informationen zu den Versionen wurden aus den jeweiligen Programmen und der Entwicklungsumgebung (Dev) zusammengetragen.

### 3.2.2 Umgebung

Intranet Sek II

Die zentrale Datenverwaltungsplattform Intranet Sek II kurz auch IN2 genannt, in welcher die Erweiterung implementiert wird, wird an Mittel und -Berufsschulen im Kanton Zürich eingesetzt. Die Applikation wurde mithilfe des Zend-Frameworks 1 aufgebaut. Das Projekt läuft auf 3 verschiedenen Serverumgebungen und ist klar in Ihre Verwendungen aufgeteilt.

**Live-Umgebung:**

Produktive Umgebung, welche von den Schülern, Lehrer und weiteren zugriffsberechtigten genutzt wird. Hier darf auf keinen Fall Code deployt werden der nicht getestet und funktionsfähig ist.

**Stage:**

Testumgebung, auf welcher die neu entwickelten Features und Änderungen getestet und bei einem Release auf Live gepatcht werden.

**Development:**

Entwicklungsumgebung, in welcher neue Features oder allfällige Änderungen vorgenommen werden.

**KendoUI:**

KendoUi ist ein Framework, welches über 70 verschiedene Komponenten wie z.B Tabellen, Diagramme oder Kalender Widgets zur Verfügung stellt und eine schnelle Realisierung einer professionellen Website ermöglicht. Ein besonderer Vorteil liegt darin, dass für alle Komponenten eine ausführliche Dokumentation der Optionen und Funktionen vorliegt. Zudem stehen auch zahlreiche Demos zur Verfügung, welche direkt vom User bearbeitet werden können.

**Typescript:**

Dies ist eine Programmiersprache die von Microsoft entwickelt wurde, um eine typisierte Javascript Programmierung zu ermöglichen. TypeScript ist ein Superset von JavaScript, dass bedeutet das es eine Erweiterung von JavaScript ist. Schlussendlich gibt es keinen Unterschied zwischen den beiden Sprachen da TypeScript zu JavaScript kompiliert wird.

**Docker:**

Docker ist eine Software, welche die Container-Virtualisierung von Anwendungen ermöglicht. Im Vergleich zu virtuellen Maschinen beinhalten die Container kein komplettes Betriebssystem, sondern beinhalten lediglich die Daten, die für die darauf installierte Anwendung notwendig sind. Ein Container kann zudem sehr leicht zwischen versch. Umgebungen hin und her geschoben werden.

**Zend-Framework 1:**

Ein Framework für PHP das einem sehr viel Arbeit abnimmt. Funktionalitäten sind z.B die Zend-View mit der man Daten serverseitig zum Client schicken kann oder die Möglichkeit SQL-Querys zusammenzustellen.

**ZF1-Future:**

Da der Support für Zend-Framework 1 seit dem 28.September 2016 eingestellt wurde, verwenden wir die Dependency ZF1-Future welche uns ermöglicht das Framework weiterhin zu nutzen mit der aktuellsten PHP-Versionen.

### 3.2.3 Aufgabenstellung und Kriterienkatalog analysieren

Damit ich alle für das Projekt benötigten Informationen sammeln konnte, druckte ich mir die Aufgabenstellung und den Kriterienkatalog aus und markierte mir die beim durchlesen wichtigen Stellen.

Analyse der Hauptanforderungen:

Neues Kalender-Widget im Formular Editor:

Es muss bei der Auswahl im Formular Editor im Dropdown bei den Formulartypen ein neuer Eintrag definiert werden.

Den neuen Formulartyp als Kalender festlegen:

Der neue Formulartyp muss ein Kalenderelement sein.

Die Datenquelle der zu anzuzeigenden Informationen soll vom Administrator festlegbar sein:

Bei der Editierung des Kalenderelements soll es möglich sein zwischen verschiedenen Datenquellen zu wählen. Die möglichen Datenquellen sind Feiertage, Schulanlässe, Personen/Klassen-Lektionen und Raum-Belegungen.

Das Widget soll in der veröffentlichten Darstellung zusätzlich zu Basisfunktionalität personen-spezifische Daten des eingeloggten Users darstellen.

Der Endanwender soll im auszufüllenden Formular seine eigenen Daten sehen, wann er eine Lektion oder Freistunde hat.

Das Kalenderelement soll alleinstehend oder zusätzlich zur Datums und Zeitraum-Wähler Funktion dargestellt werden.

Der Administrator kann festlegen ob der Endanwender das Kalenderelement als informatives oder interaktives Element nutzen kann. Sprich der Endanwender sieht die zusätzlichen Informationen im Kalender Widget kann aber kein Datum wählen und abschicken, oder eben er kann ein gewünschtes Datum aus dem Kalender Widget wählen.

### 3.2.3 Mit Teil.1 der IPA-Dokumentation beginnen

Die Grundstruktur de Dokumentation hatte ich bereits vorbereitet (das heisst Titelblatt, Kopf- und Fusszeile und alle Übertitel) was den Einstieg recht beschleunigt hat. Zu allererst habe ich den Originaltext der Aufgabenstellung in die Dokumentation eingefügt und zudem die Projektaufbauorganisation erstellt. Weiter deklarierte ich die Vorkenntnisse, Vorarbeiten und erstellte den Zeitplan mit der 3 Zeiten-Methode.

## 3.3 Planen

### 3.3.1 Zeitplan erstellen

Aufbau:

Der Zeitplan gliedert sich in die jeweils täglich geplanten Tätigkeiten sowie der zur Verfügung stehenden Zeitspanne der IPA, wobei die Aufteilung der einzelnen Tage in fünf zwei Stunden-Raster aufgeteilt wurde.

Meilensteine:

Für jeden der sechs teilschritte der IPERKA-Methode habe ich mir einen Meilenstein für den Abschluss der Phase definiert.

Informieren:

Benötigte Informationen konnten beschafft und die Phase erfolgreich abgeschlossen werden.

Planen:

Die Arbeit konnte komplett geplant werden, sowie der zugehörige Zeitplan.

Entscheiden:

Alle noch zu treffenden Entscheidungen konnten analysiert und getroffen werden.

Realisieren:

Die Realisierung des Auftrags war erfolgreich und der Zeitplan konnte eingehalten werden.

Kontrollieren:

Das Testen der implementierten Aufgabe ergab ein positives Ergebnis und allfällige Bugs konnten behoben werden.

Auswerten:

Die Auswertung der Tests wurde durchgeführt und der IPA-bericht wurde zeitgerecht hochgeladen.