

IPA-Bericht

Code-Generator



Autor Luan Caduff

Klasse ISO-20

Datum 03. Mai 2024

Firma Technogroup IT-Service AG / Evernex IT Services Switzerland AG

Table of Contents

1	Teil 1	4
1.1	Dokumenteninformationen	4
1.1.1	Historie	4
1.1.2	Eigenschaften	4
1.2	Aufgabenstellung	5
1.3	Projektorganisation	7
1.4	Projektmethode	7
1.5	Deklaration der Vorkenntnisse	8
1.6	Deklaration der Vorarbeiten	8
1.7	Deklaration der benutzen Firmenstandards	8
1.8	Organisation der Arbeitsergebnisse	9
1.9	Zeitplan	10
1.10	Arbeitsjournal	11
1.10.1	Tag 1, 18. April 2024	11
	Tag 2, 19. April 2024	12
1.10.2	Tag 3, 23. April 2024	12
1.10.3	Tag 4, 24. April 2024	12
1.10.4	Tag 5, 25. April 2024	12
1.10.5	Tag 6, 26. April 2024	12
1.10.6	Tag 7, 30. April 2024	12
1.10.7	Tag 8, 01. Mai 2024	12
1.10.8	Tag 9, 02. Mai 2024	12
1.10.9	Tag 10, 03. Mai 2024	13
1.10.11	Zusammenfassung Zeitplan	14
2	Teil 2	15
2.1	Management Summary	15
2.1.1	Ausgangslage	15
2.1.2	Vorgehen	15
2.1.3	Ergebnis	15
2.2	Informieren	16
2.2.1	Aufgabestellung	16
2.2.2	Technologien	16
2.2.3	Use-Cases	17
2.3	Planen	19
2.3.1	GUI MockUps	19
2.4	Entscheiden	22

2.5	Realisieren.....	23
2.6	Kontrollieren.....	24
2.6.1	Tests und Testprotokolle	24
2.7	Auswerten	25
2.8	Verzeichnisse.....	26
2.8.1	Abbildungsverzeichnis.....	26
2.8.2	Tabellenverzeichnis.....	26
2.8.3	Links.....	26
2.8.4	Glossar / Abkürzungen	26
2.9	Anhang.....	27

1 Teil 1

1.1 Dokumenteninformationen

1.1.1 Historie

Version	Gültig ab	Dokumentenhistorie / Änderungshinweis	Autor
0.1.0.1	17.04.2024	Erstellung der Grundstruktur	Luan Caduff
0.2.0.2	18.04.2024	Zeitplan, Informieren und Planen	Luan Caduff
0.3.0.3	19.04.2024	Informieren, Entscheiden	Luan Caduff
1.0.0.11	03.05.2024	Version bei der Abgabe	Luan Caduff

Tabelle 1 - Dokumentenhistorie

Versionierung

A.B.C.D

A = Eine Veröffentlichung / bereit zum Druck / Abgabe

B = Inhaltliche Änderungen am Dokument

C = Korrekturen (keine inhaltlichen Änderungen)

D = Laufnummer (wird bei jeder Änderung erhöht)

1.1.2 Eigenschaften

Bezeichnung	Detailinformationen
Status	In Arbeit
Autor	Luan Caduff
Ausbildung zum	Eidg. Dipl. Informatiker EFZ
Fachrichtung	Applikationsentwickler
Version	1.0.0.XX
Versionsdatum	03.05.2024
Seiten	27

Tabelle 2 - Dokumenteneigenschaften

1.2 Aufgabenstellung

Die Firma S + O AG ist Partnerfirma der Evernex IT Services Switzerland AG und zuständig für den Betrieb und die Weiterentwicklung deren Web-Applikationen. Die bestehende Applikation SWO (Simple Web Office) soll abgelöst werden. In einem ersten Schritt ist geplant, auf der bestehenden MariaDB Datenbank-Struktur neue Web-Views (CRUDs) für die Administration zu erstellen. Um diese Arbeit zu erleichtern und zu beschleunigen soll ein Code-Generator erstellt werden, welcher anhand der Struktur einer Datenbank-Tabelle Code-Snippets erstellt.

Der Code-Generator soll mit PHP sowie HTML, CSS und JavaScript umgesetzt werden. Es ist eine neue, stand-alone Applikation, welche entsprechend unabhängig läuft, ohne Einbettung in ein bestehendes Umfeld.

Da diese Applikation lediglich lokal von uns verwendet werden soll, benötigt es kein Authentifizierungssystem.

Der Benutzer dieses Code-Generators soll in einem Web-GUI den Datenbank-Server aus einer Selectbox auswählen können. Für die Auswahl des Datenbank-Servers sollen die zum Verbindungsaufbau nötigen Informationen aus einer Konfigurations-Datei gelesen werden (CSV, eine Zeile pro Datenbank-Server mit Strichpunkt getrennte Informationen wie Host, Login, Passwort).

Nach Auswahl des Datenbank-Servers werden dessen Datenbanken in einer weiteren Selectbox zur Auswahl angezeigt. Nachdem eine Datenbank gewählt wurde, werden dessen Tabellen ebenso zur Auswahl angezeigt. Nachdem eine Tabelle ausgewählt wurde, kann mit dem jeweiligen Button ein Code-Snippet erstellt werden.

Das generierte Code-Snippet soll im Web-GUI angezeigt werden und mittels eines Copy-Buttons in die Zwischenablage kopiert werden können.

Das Ziel dieser Web-Applikation ist es also ein sauber formatiertes (übliche Einrückungen, Zeilen-Abstände zur besseren Lesbarkeit) Code-Snippet in der Zwischenablage zur weiteren Verarbeitung bereit zu stellen.

Nachfolgende Code-Snippets sollen generiert werden können:

Snippet Art	Zweck
PHP-Model-Klasse	Abbild der Tabelle (Attribute), Konstruktor, Getter/Setter, JSON-Serialize, Objekt-Erstellung aus Daten
PHP-Gateway-Klasse	Hinzufügen bzw. Anpassen eines neuen Datensatzes anhand des Model-Objektes
ExtJS-Model	Laden bzw. mappen via JSON übertragener Datensätze
ExtJS-Grid-List	Liste der geladenen Datensätze darstellen
ExtJS-Create-Dialog	Eingabe-Formular für einen neuen Datensatz
ExtJS-Edit-Dialog	Eingabe-Formular für die Anpassung eines bestehenden Datensatzes
ExtJS-Info-Dialog	Darstellen aller Informationen (nicht editierbare Datenfelder) eines bestehenden Datensatzes

Tabelle 3 - Code-Snippet Arten

Als Vorlage für die Code-Snippets dient das vorhandene Test-CRUD der Tabelle «erp_article_service», an dem ich in den letzten Monaten gearbeitet habe. Diese Vorlagen werden aber hier nicht mitgeliefert bzw. hochgeladen, da es sich dabei um die gesamte Test-Applikation handelt und daraus allgemeine (bzw. für die gewählte Tabelle), sinnvolle Code-Snippets (nach obiger Auflistung) erstellt werden sollen. Die Auswahl für «sinnvoll» ist Bestandteil der IPA und soll auch entsprechend begründet werden.

Die Architektur soll MVC nach Firmenusanz abbilden: Klassen mit entsprechenden Funktionalitäten in entsprechenden Verzeichnissen. Dazu existieren keine dokumentierten Firmenstandards.

Die technische Dokumentation zum Aufbau der Applikation soll mittels Use Case und eines Klassen-Diagramms erstellt werden.

Das Code-Styling soll wie folgt sein: Die Namensgebung ist einfach gut gewählt. Die Struktur des Codes ist ebenfalls einfach übersichtlich gemacht. Es ist eine gewisse Einheit zu sehen in der Art und Weise, wie der Code strukturiert ist (d.h. es ist überall etwa gleich gemacht).

Die Applikation soll manuell, anhand von sinnvollen Testfällen getestet werden. Die Definition der Testfälle ist Bestandteil der IPA und die einzelnen Tests müssen dann auch entsprechend protokolliert werden.

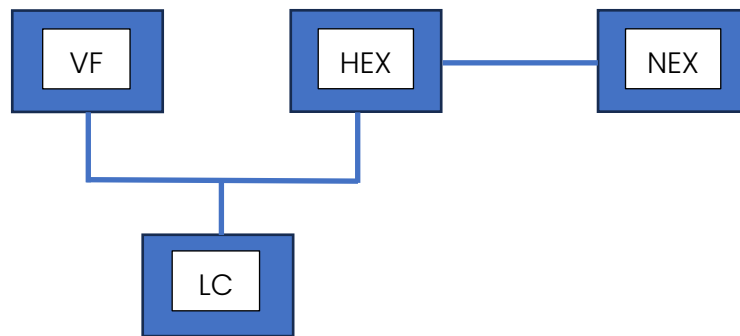
1.3 Projektorganisation

VF – Roman Born (Verantwortliche Fachkraft)

HEX – Dennis Werkes (Hauptexperte)

NEX – Michelle Schär (Nebenexperte)

LC – Luan Caduff (Kandidat)



1.4 Projektmethode

Dieses Projekt wird mit IPERKA durchgeführt. Dies ist eine simple Projektmanagement-Methode zur strukturierten Planung und Umsetzung eines Projektes.

IPERKA ist ein Akronym und bedeutet folgendes:

I – Informieren – Relevante Informationen sammeln

P – Planen – Damit einen Plan erstellen

E – Entscheiden – Sich für eine spezifische Vorgehensweise entscheiden

R – Realisieren – Das Projekt durchführen

K – Kontrollieren – Endprodukt testen und überprüfen

A – Auswerten – Über den Arbeitsprozess und das Resultat reflektieren

Weitere Informationen zur IPERKA-Methode finden Sie unter https://www.ict-berufsbildung-bern.ch/resources/Iperka_OdA_200617.pdf

1.5 Deklaration der Vorkenntnisse

Alle geplanten Tätigkeiten/Produkte/Techniken sind bekannt und wurden während der gesamten Praktikumszeit eingesetzt.

Technologie	Erfahrung
PHP	Sehr gute Kenntnisse Seit 2.5 Jahren aktiv genutzt.
HTML	Sehr gute Kenntnisse Seit 3.5 Jahren aktiv genutzt.
JavaScript + jQuery	Ziemlich gute Kenntnisse Seit 2.5 Jahren öfters genutzt.
CSS + Bootstrap	Ziemlich gute Kenntnisse Seit 3.5 Jahren öfters genutzt.
MariaDB	Sehr gute Kenntnisse Seit 3.5 Jahren aktiv genutzt.

1.6 Deklaration der Vorarbeiten

In den letzten Monaten habe ich ein Test-CRUD der Tabelle «erp_article_service» aufgebaut. Dieses CRUD dient als Grundlage für die Code-Snippets, die der Generator erstellen soll.

In direktem Zusammenhang mit dieser Arbeit habe ich ein GIT-Repository auf GitHub eingerichtet.

Des Weiteren habe ich die Dokumentenvorlage inklusive Verzeichnisstruktur am Vortag zum IPA-Start erstellt.

1.7 Deklaration der benutzten Firmenstandards

Es existieren keine dokumentierten Firmenstandards.

Der Code wird nach Firmenusanz aufgebaut, das heisst nach gängigen Konventionen in Bezug auf die Einrückung, Gross- Kleinschreibung etc.

1.8 Organisation der Arbeitsergebnisse

Der erarbeitete Code wird in das lokale GIT-Repository committed.
Das IPA-Dokument wird ebenfalls – jeweils vor Arbeitsende – im lokalen GIT-Repository in das dafür vorgesehene Verzeichnis (000_Dokumentation) sowohl als Word-Dokument als auch als PDF kopiert und committed.

Das Repository wird auf GitHub gepusht:

<https://github.com/xivia/ipa-luan---code-generator>

1.9 Zeitplan

Code Generator Zeitplan

Gesamte geplante Zeit: 80

Geplante Arbeit:

Geplante Doku:

Firmenname	Evernex IT Services Switzerland AG
Projektstart	Do, 18.4.2024

Gesamte aufgewendete Zeit: 14

Effektive Arbeit:

Effektive Doku: 

[illegible]

1.10 Arbeitsjournal

1.10.1 Tag 1, 18. April 2024

Arbeiten	Dokumentation <ul style="list-style-type: none"> ▪ Zeitplan erstellt ▪ Im IPA Dokument Teil 1 abgefüllt Informieren <ul style="list-style-type: none"> ▪ Use Case Diagramm erstellt Planen <ul style="list-style-type: none"> ▪ MockUps erstellt
Probleme	Ich habe die MockUps erstellt, jedoch wurden diese nicht richtig gespeichert und sind verloren gegangen. Aus diesem Grund habe ich sie dann erneut erstellen müssen.
Hilfestellungen	Zeitplan mit VF kurz besprochen.
Überzeiten	Aufgrund meines Problems habe ich für die MockUps eine Stunde mehr als geplant aufgewendet.
Ungeplante Arbeiten	Keine ungeplanten Arbeiten.
Erfolge	Alle Arbeiten wurden zufriedenstellend abgeschlossen.
Misserfolge	Keine Misserfolge.
Reflektion	<p>Ich habe die MockUps wie bereits erwähnt mit demselben Web-Tool erstellt, wie zuvor die Use Cases. Mit diesem Tool konnte ich die MockUps effizient zusammensetzen. Leider habe ich aus Versehen meinen Browser geschlossen, bevor das Tool die Änderungen speichern konnte. Zukünftig werde ich besser darauf Acht geben, Dateien öfters abzuspeichern.</p> <p>Da ich nun etwas in Verzug bin, da die Use Case Beschreibungen noch nicht dokumentiert wurden, muss ich morgen im geplanten Dokumentieren-Block als erstes diese noch verfassen, um so wieder auf Kurs zu kommen.</p>

Tag 2, 19. April 2024

Arbeiten	Erstes Gespräch mit HEX Dokumentation <ul style="list-style-type: none"> ▪ Generell Dokumentation verschönert Informieren <ul style="list-style-type: none"> ▪ Use Case Beschreibungen verfasst Entscheiden <ul style="list-style-type: none"> ▪ GUI-Varianten Entscheid gefällt
Probleme	Keine Probleme.
Hilfestellungen	Vorbereitung mit VF auf HEX-Gespräch.
Überzeiten	Keine Überzeiten.
Ungeplante Arbeiten	Keine ungeplanten Arbeiten.
Erfolge	Alle Arbeiten wurden zufriedenstellend abgeschlossen.
Misserfolge	Keine Misserfolge.
Reflektion	Dieser Schritt war schwerer als gedacht, da die Vergleichskriterien schwierig zu finden sind bzw. nicht immer objektiv sind.

1.10.2 Tag 3, 23. April 2024

1.10.3 Tag 4, 24. April 2024

1.10.4 Tag 5, 25. April 2024

1.10.5 Tag 6, 26. April 2024

1.10.6 Tag 7, 30. April 2024

1.10.7 Tag 8, 01. Mai 2024

1.10.8 Tag 9, 02. Mai 2024

1.10.9 Tag 10, 03. Mai 2024

1.10.11 Zusammenfassung Zeitplan

Hier zählen wir alle Stunden pro Tag zusammen um einen guten, übersichtlichen Soll-/Ist- Vergleich zu erhalten.

Datum	Soll [h]	Ist [h]
Tag 1, 18 April 2024	8	8.50
Tag 2, 19 April 2024	8	8.50
Tag 3	8	
Tag 4	8	
Tag 5	8	
Tag 6	8	
Tag 7	8	
Tag 8	8	
Tag 9	8	
Tag 10	8	
Total	80	

Tabelle 4 - Zeitplan Zusammenfassung

Begründung

2 Teil 2

2.1 Management Summary

2.1.1 Ausgangslage

Text

2.1.2 Vorgehen

Text

2.1.3 Ergebnis

Text

2.2 Informieren

2.2.1 Aufgabestellung

Siehe Kapitel 1.2 Aufgabenstellung

Nach Analyse dieser Aufgabenstellung habe ich bemerkt, dass nicht definiert ist, wie diese Config-Datei abgefüllt werden soll. Nach Rücksprache kann sie manuell bearbeitet werden. Ich habe mich dann aber dazu entschlossen ein kleines Formular zum Hinzufügen eines Eintrages umzusetzen, da dies kein grosser Zusatzaufwand ist aber einen Mehrwert bietet.

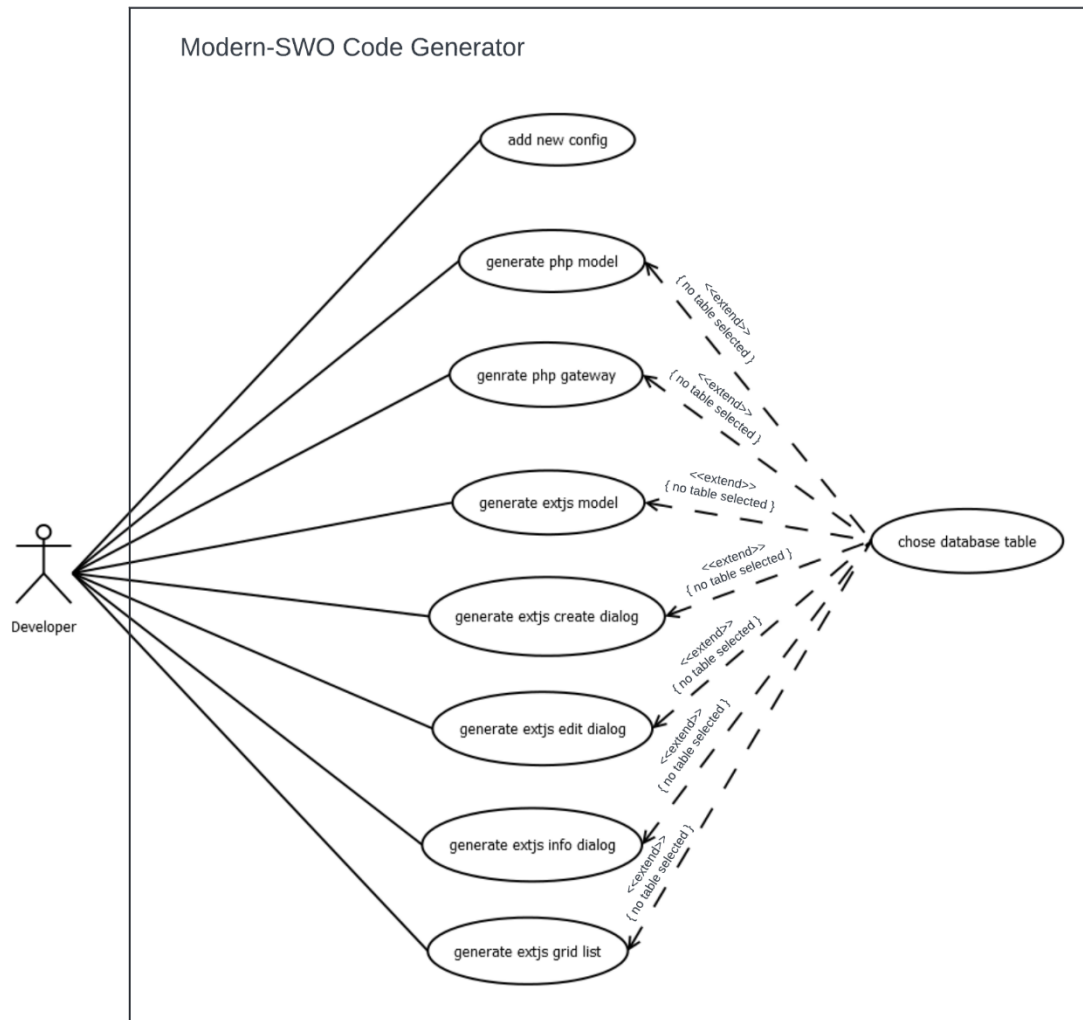
2.2.2 Technologien

Als Vorgabe habe ich folgende Technologien zu benützen:

- ➔ PHP – Personal Home Page Hypertext Preprocessor (rekursives Akronym)
 - Verwendet wird die Version 8.2
- ➔ HTML – Hypertext Markup Language
 - Verwendet wird HTML5
- ➔ JavaScript mit der jQuery Library
 - Hier kommen ECMAScript 2023 und jQuery Version 3.7.1 zum Einsatz
- ➔ CSS mit dem Bootstrap Framework – Cascading Style Sheets
 - Hier verwenden wir auch die aktuellsten Versionen der jeweiligen Technologien; CSS 3 & Bootstrap 5.3.2

2.2.3 Use-Cases

Mithilfe des Web-Tools «Lucidchart» habe ich das Use-Case-Diagramm erstellt.
Mehr zum Tool finden Sie unter <https://www.lucidchart.com/pages/>



Developer – add new config

Durch Klick auf einen Button öffnet sich ein Formular, in welchem eine neue Datenbankverbindung hinterlegt werden kann. Es müssen Host, User, Passwort und der Port angegeben werden. Danach wird diese in der config-datei gespeichert. Ob die Verbindung hergestellt werden kann wird in diesem Prozess nicht geprüft.

Developer – chose database table

Es kann eine Datenbankverbindung gewählt werden, woraufhin versucht wird, diese Verbindung herzustellen. Bei Erfolg kann eine darin enthaltene Datenbank ausgewählt werden. Danach kann die Tabelle der Datenbank ausgewählt werden. Kann die Verbindung nicht hergestellt werden, erscheint eine Fehlermeldung.

Developer – generate php model**Developer – generate php gateway****Developer – generate extjs model****Developer – generate extjs create dialog****Developer – generate extjs edit dialog****Developer – generate extjs info dialog****Developer – generate extjs grid list**

Vorbedingung: Es muss eine Datenbank-Tabelle ausgewählt sein.

Durch Klick auf den entsprechenden Button wird das Code-Snippet generiert und dargestellt. Jetzt kann das Code-Snippet in die Zwischenablage kopiert werden.

Wenn in der Tabelle ein unbekannter Datentyp vorkommt wird eine Fehlermeldung ausgegeben.

2.3 Planen

2.3.1 GUI MockUps

Ich möchte zwei Versionen eines möglichen Frontend-Aufbaus dieser Applikation entwerfen. Dazu verwende ich dasselbe Web-Tool wie bereits für die Use-Cases.

Entwurf 1

The mockup shows a web application titled "Modern-SWO Code Generator". On the left is a sidebar with three labels: "Config Label", "DB Label", and "Table Label", each followed by a "<selectbox>" placeholder. Below these is a "new" button. The main area on the right has a title "Modern-SWO Code Generator". Under "PHP", there are two "gen" buttons. Under "ExtJS", there are five "gen" buttons. Below these is an "Output:" label and a large dark gray box labeled "output box" with a "copy" button in its top right corner.

Abbildung 1 – MockUp 1

Entwurf 2

Modern-SWO Code Generator

new

Config Label

<selectbox>

DB Label

<selectbox>

Table Label

<selectbox>

PHP

gen

gen

ExtJS

gen

gen

gen

gen

gen

Output:

copy

output box

Abbildung 2 – MockUp 2

Der Button «new» sollte den Dialog zum Erfassen einer neuen Datenbankverbindung öffnen. Entweder als Pop-Up (MockUp 1), oder als versteckte Formfelder, die dann auftauchen (MockUp 2).

Die Selectboxen gewährleisten jeweils die Auswahl des Datenbank-Servers, der Datenbanken und der Tabelle.

Die Buttons, die mit «gen» gekennzeichnet sind, werden die Code-Snippets generieren und in der output box darstellen.

Um das Kopieren in die Zwischenablage zu erleichtern habe ich noch einen Copy-Button in die output box integriert.

2.4 Entscheiden

Wir müssen uns für eine der beiden GUI-Varianten entscheiden.

Variante 1 oder Variante 2

Modern-SWO Code Generator

Config Label
<selectbox>

DB Label
<selectbox>

Table Label
<selectbox>

new

PHP
gen gen

ExtJS
gen gen gen gen

Output:

output box

Modern-SWO Code Generator

new

Config Label
<selectbox>

DB Label
<selectbox>

Table Label
<selectbox>

PHP
gen gen

ExtJS
gen gen gen gen

Output:

output box

Variante 1 hat ein horizontales Layout, wo die Selectboxes links ziemlich eng zusammen sind, und unterhalb dieser der «new» Button mit einem Trennstrich isoliert ist. Dies liegt daran, dass der Platz unter dem «new» Button dafür vorgesehen ist, ein Formular anzuzeigen, wenn der Button gedrückt wird.

Bei der Variante 2 wäre solch ein Formular vom Platz her nicht umsetzbar, und man müsste ein modales Pop-Up verwenden.

Bei der Variante 1 gibt es vertikal mehr Platz für das Code-Snippet als bei Variante 2 und grundsätzlich wird das Code-Snippet eher länger als breiter.

Der new-Button impliziert bei Variante 2, dass zuerst eine Config erstellt werden muss, was nicht der Fall ist. Und wenn der new-Button unter den Selectboxes ist muss man diesen jeweils „überspringen“.

Der «Arbeits-Prozess»: Wähle Config -> DB -> Table -> Snippet -> Copy geht bei Variante 1 ein wenig „lockerer von der Hand“: von links oben nach rechts versus links-rechts-links

Entscheid: Es wird Variante 1 umgesetzt.

2.5 Realisieren

2.6 Kontrollieren

Text

2.6.1 Tests und Testprotokolle

2.7 Auswerten

Text

Bezug auf die ganze Arbeit

Persönliches Fazit

2.8 Verzeichnisse

2.8.1 Abbildungsverzeichnis

Abbildung 1 - MockUp 1.....	19
Abbildung 2 - MockUp 2.....	20

2.8.2 Tabellenverzeichnis

Tabelle 1 - Dokumentenhistorie.....	4
Tabelle 2 - Dokumenteneigenschaften.....	4
Tabelle 3 - Code-Snippet Arten.....	6
Tabelle 4 - Zeitplan Zusammenfassung.....	14
Tabelle 5 - Glossar.....	26

2.8.3 Links

https://www.ict-berufsbildung-bern.ch/resources/lperka_OdA_200617.pdf

Website der Berufsbildung Bern...

Heruntergeladen am: 18.04.2024

<https://www.lucidchart.com/pages/>

Web-Tool für MockUps

Erstellt am: 18.04.2024

2.8.4 Glossar / Abkürzungen

CRUD	Create, Read, Update, Delete
SWO	Simple Web Office

Tabelle 5 – Glossar

2.9 Anhang

Besprechungsprotokolle

Quellcode