

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

2.5	Realisieren.....	19
2.6	Kontrollieren.....	20
2.6.1	Tests und Testprotokolle	20
2.7	Auswerten	21
2.8	Verzeichnisse.....	22
2.8.1	Abbildungsverzeichnis.....	22
2.8.2	Tabellenverzeichnis.....	22
2.8.3	Links.....	22
2.8.4	Glossar / Abkürzungen	22
2.9	Anhang.....	23

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
1.0.0.X	03.04.2024	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	1.0.0.11
Versionsdatum	03.05.2024
Seiten	23

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 dieser 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 dann 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 zu weiterer 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 vom Lernenden 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

→ Organigramm

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 eingerichtet und die Dokumentenstruktur erstellt.

1.7 Deklaration der benutzten Firmenstandards

Es existieren keine Firmenstandards.
Der Code wird nach Firmenusanz aufgebaut.

1.8 Organisation der Arbeitsergebnisse

Die Arbeitsergebnisse werden täglich in das GIT-Repository gepusht.
Das Dokument wird ebenfalls im GIT-Repository in ein dafür vorgesehenes Verzeichnis kopiert.

1.9 Zeitplan

Code Generator Zeitplan

Gesamte geplante Zeit: 80

Geplante Arbeit:

Geplante Doku:

Firmenname
Projektstart

Evernex IT Services Switzerland AG
Do, 18.4.2024

Gesamte aufgewendete Zeit: 8

Effektive Arbeit:

Effektive Doku:

		Do, 18.4.2024		–		Fr, 19.4.2024		Di, 23.4.2024		–		Fr, 26.4.2024		Di, 30.4.2024		–		Fr, 3.5.2024			
		Tag 1		Tag 2		Tag 3		Tag 4		Tag 5		Tag 6		Tag 7		Tag 8		Tag 9		Tag 10	
AUFGABE	SOLL/IST	Do, 18.4.2024		Fr, 19.4.2024		Di, 23.4.2024		Mi, 24.4.2024		Do, 25.4.2024		Fr, 26.4.2024		Di, 30.4.2024		Mi, 1.5.2024		Do, 2.5.2024		Fr, 3.5.2024	
Dokumentieren	28																				
	3																				
Informieren																					
Use Case Diagramm	2																				
	2																				
Planen																					
GUI Mockups	2																				
	3																				
Entscheiden																					
Entscheid	2																				
	0																				
Realisieren																					
Tabellen-Auswahl (Frontend)	6																				
	0																				
Snippet-Auswahl (Frontend)	4																				
	0																				
DB Verbindungen (Backend)	2																				
	0																				
Tabellen-Auswahl (Backend)	6																				
	0																				
Snippet-Generierung (Backenend)	10																				
	0																				
Kontrollieren																					
Testfälle	8																				
	0																				
Auswerten																					
Auswerten	4																				
	0																				
Diverses																					
Besuche/Gespräche	2																				
Reserve	4																				
Meilensteine																					
Bereit zur Realisierung																					
Bereit für Testing																					
Bereit für Abgabe																					

1.10 Arbeitsjournal

1.10.1 Tag 1, 18. April 2024

Zeitplan erstellt

IPA Dokument Teil 1 abgefüllt.

IPA Dokument Teil 2:

- Use Case Diagramm erstellt
- 2 GUI MockUp Varianten erstellt.

MockUps haben mehr Zeit in Anspruch genommen als geplant.

Deshalb fehlen noch die Beschreibungen der Use Cases.

Das erledige ich morgen als erstes, im geplanten Dokumentieren-Block.

1.10.2 Tag 2, 19. April 2024

1.10.3 Tag 3, 23. April 2024

1.10.4 Tag 4, 24. April 2024

1.10.5 Tag 5, 25. April 2024

1.10.6 Tag 6, 26. April 2024

1.10.7 Tag 7, 30. April 2024

1.10.8 Tag 8, 01. Mai 2024

1.10.9 Tag 9, 02. Mai 2024

1.10.10 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	8	
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

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.

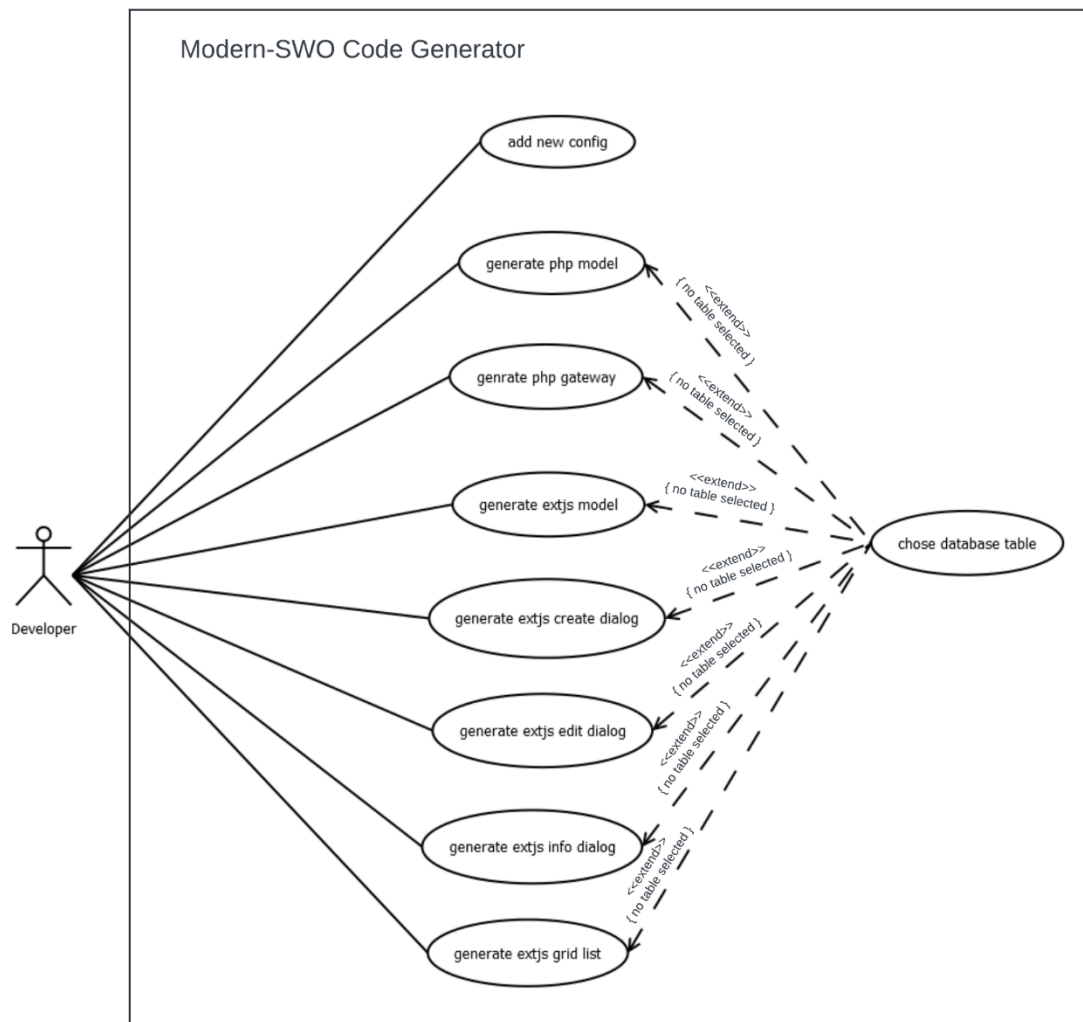
2.2.2 Technologien

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/>



Use Cases beschreiben

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

Modern-SWO Code Generator

Config Label
<selectbox>

DB Label
<selectbox>

Table Label
<selectbox>

new

PHP
gen gen

ExtJS
gen gen gen gen gen

Output:

copy

output box

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).

2.4 Entscheiden

2.5 Realisieren

2.6 Kontrollieren

2.6.1 Tests und Testprotokolle

2.7 Auswerten

2.8 Verzeichnisse

2.8.1 Abbildungsverzeichnis

Abbildung 1 - MockUp 1.....	16
Abbildung 2 - MockUp 2.....	17

2.8.2 Tabellenverzeichnis

Tabelle 1 - Dokumentenhistorie.....	4
Tabelle 2 - Dokumenteneigenschaften.....	4
Tabelle 3 - Code-Snippet Arten.....	6
Tabelle 4 - Zeitplan Zusammenfassung.....	12
Tabelle 5 - Glossar.....	22

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