

Hochschule Worms  
Fachbereich Informatik  
Studiengang Angewandte Informatik - dual (B.Sc)

Abschlussarbeit

# Bachelorarbeit

Wie kann die Anmeldung und Zustellung von Weinen für  
Weinproben des Regionalverbunds für Weine in der  
Weinregion Mosel effizient und profitabel durch eine  
TYPO3-Erweiterung realisiert werden?

Version 1.3

Vorgelegt von

Leon Etienne, 676838  
inf3642@hs-worms.de  
mit Medienagenten Stange & Ziegler oHG  
Im Sommersemester 2023  
Bad Dürkheim

bei Volker Schwarzer  
schwarzer@hs-worms.de  
Worms

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist noch nicht in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung vorgelegt worden.

Bad Dürkheim, 22. Dezember 2022

Leon Etienne \_\_\_\_\_

---

# Inhaltsverzeichnis

|   |            |
|---|------------|
| <b>Abbildungsverzeichnis</b>  | <b>I</b>   |
| <b>Tabellenverzeichnis</b>  | <b>II</b>  |
| <b>Abkürzungsverzeichnis</b>  | <b>III</b> |
| <b>Glossar</b>  | <b>IV</b>  |
| <b>1. Einleitung</b>  | <b>1</b>   |
| 1.1. Problemstellung . . . . .  | 1          |
| 1.2. Zielsetzung . . . . .  | 2          |
| <b>2. Stand der Forschung</b>   | <b>3</b>   |
| 2.1. Modell nach Parviainen et al. . . . .                              | 3          |
| 2.2. Modell nach Verhoef et al. . . . .                                 | 6          |
| 2.3. Abwägung in Bezug auf die Problemstellung . . . . .                | 6          |
| 2.3.1. Parviainen et al. . . . .  | 7          |
| 2.3.2. Verhoef et al. . . . .   | 7          |
| 2.3.3. Fazit . . . . .  | 7          |
| <b>3. Stand der Technik</b>   | <b>9</b>   |
| 3.1. Die bestehende Webseite . . . . .                                  | 9          |
| 3.2. QR-Code-Bibliotheken . . . . .                                     | 9          |
| 3.2.1. Javascript-Implementationen . . . . .                            | 10         |
| 3.2.2. PHP-Implementationen . . . . .                                   | 11         |
| 3.2.3. Subjektiver Vergleich im Bezug auf die Problemstellung . . . . . | 13         |
| 3.2.4. Fazit . . . . .  | 17         |
| 3.3. PDF-Generator . . . . .  | 17         |
| <b>4. Anforderungserfassung</b>   | <b>19</b>  |
| 4.1. Interview mit Product Owner . . . . .                              | 20         |

---

|  |           |
|--|-----------|
| 4.2. Online-Fragebögen für Stakeholder . . . . . | 20        |
| 4.3. Ergebnisse . . . . .                        | 21        |
| <b>5. Umsetzung</b>                              | <b>24</b> |
| 5.1. Setup einer TYPO3-Extension . . . . .       | 24        |
| 5.2. Digitization . . . . .                      | 24        |
| 5.3. Digitalization . . . . .                    | 28        |
| 5.3.1. Teilnehmerregistrierung . . . . .         | 28        |
| 5.3.2. Weinregistrierung . . . . .               | 33        |
| 5.3.3. PDF- und QR-Code-Generierung . . . . .    | 39        |
| <b>6. Ergebnisse</b>                             | <b>41</b> |
| <b>7. Ausblick</b>                               | <b>42</b> |
| <b>Literaturverzeichnis</b>                      | <b>43</b> |
| <b>Anhang</b>                                    | <b>49</b> |
| <b>A. Anmeldeforumular Jahresauswahlprobe</b>    | <b>50</b> |
| <b>B. Interview-Fragebogen</b>                   | <b>53</b> |
| <b>C. Online-Fragebogen Weingüter</b>            | <b>55</b> |
| <b>D. Interview-Protokoll</b>                    | <b>58</b> |
| <b>E. Pflichtenheft</b>                          | <b>61</b> |
| <b>F. Klassendiagramm</b>                        | <b>64</b> |

---

# Abbildungsverzeichnis

|  |    |
|--|----|
| 2.1. Model for tackling digital transformation . . . . .   | 4  |
| 2.2. Plan-Do-Check-Act (PDCA) . . . . .  | 5  |
| 2.3. Umsetzungsplanung . . . . .   | 8  |
| 5.1. Objektrelationen . . . . .  | 26 |
| 5.2. UX-Flow: Registrierung . . . . .  | 30 |
| 5.3. UX-Flow: Weinanmeldung . . . . .  | 34 |
| 5.4. Stichprobenartige Laufzeitanalyse des Kategorie-Renderers,<br>gegenüber einer quadratischen Kurve . . . . . | 37 |

# Tabellenverzeichnis

|   |    |
|---|----|
| 3.1. Subjektive Evaluation: KJua . . . . .  | 14 |
| 3.2. Subjektive Evaluation: Soldair/node-qrcode . . . . .                           | 15 |
| 3.3. Subjektive Evaluation: Chillerlan/php-qrcode . . . . .                         | 15 |
| 3.4. Subjektive Evaluation: Kreativekorp/barcode . . . . .                          | 16 |
| 3.5. Subjektive Evaluation: Bacon/BaconQRCode . . . . .                             | 17 |
| 3.6. Gesamtübersicht: Subjektive Evaluation der QR-Code Bi-<br>bliotheken . . . . . | 17 |
| 4.1. Initial bekannte Anforderungen . . . . .                                       | 19 |

# Abkürzungsverzeichnis

**WM** “Weinland Mosel” e.V.

**PDCA** Plan-Do-Check-Act

**SWOT** Strength-Weakness-Opportunity-Threat

**FA** Funktionale Anforderung

**NFA** nichtfunktionale Anforderung

**ELN** Einlieferungsnummer

**JAP** Jahresauswahlprobe

**TCA** Table configuration array

**PO** Product Owner

**UID** Unique Identifier

**PID** Parent Identifier

# Glossar

## Frontend

*“Das Frontend einer Webseite ist der Teil der Webseite, der Endnutzern präsentiert wird. Ein Frontend besteht aus HTML und Cascading Style Sheets” [Udjaja, 2018].*

## TYPO3-Backend

*“The backend’s main role is to enable users to create and publish content for their site. The backend is also used to configure a TYPO3 installation. Domains, languages and other information that determine how a site behaves are managed via the backend. Tasks such as adding backend users and managing third-party extensions also take place in the backend. ” [TYPO3 Contributors, 2023g].*

## TYPO3-Extension

*“Developed by the community, extensions provide a range of solutions that help extend TYPO3” [TYPO3 Contributors, 2023g].*

## TYPO3-Plugin

*“Plugins are a specific type of content elements [...] Plugins often use additional database tables which contain records which are dynamically displayed via the plugin - often in a list view, a single view, optionally with pagination and search functionality. An extension may provide several plugins, each with a dedicated function, such as the list view.” [TYPO3 Contributors, 2023l]*

# 1. Einleitung

Der Regionalverband für Weine “Weinland Mosel” e.V. (WM) lässt Weine in organisierten Weinproben, sog. Jahresauswahlproben (JAPs), von Juroren bewerten. Teilnehmende Weingüter registrieren ihre Weine in verschiedenen Kategorien und schicken diese auf dem Postweg ein. Dieser Prozess bildet sich in Form ausgedruckter Formulare, die von Hand ausgefüllt und von Hand in eine Excel-Tabelle übertragen werden, ab.

## 1.1. Problemstellung

Die Teilnehmenden Weingüter schicken ihre Weine zusammen mit Formularen über den Postweg zu WM. Es ist der Normalfall, dass ein teilnehmendes Weingut *mehrere* Weine zur Bewertung anbringt. In diesem Fall ist für jeden anzumeldenden Wein ein solches Formular erneut auszufüllen. Hierbei werden sämtliche auf das Weingut bezogene Daten redundant ausgefüllt. Diese Daten sind redundant, da sie keine Eigenschaften der Weine, sondern die des Weingutes selbst sind. Da sich das Weingut zwischen den Weinen nicht ändert, ändern sich die darauf bezogenen Daten auch nicht. Sie müssen aber für jeden Wein erneut ausgefüllt werden. Abgesehen davon, dass solche Redundanzen auf Weinguts- und Verbandsseite die hedonische Qualität schädigen, bietet so ein Workflow Freiraum für Fehler und Inkonsistenzen. Dieser Workflow mit den zuvor genannten Nachteilen wird auf Verbandsseiten, nach Zustellung der Weine, weiter fortgeführt: WM erfährt erstmalig mit der Zustellung eines Weines von dessen Teilnahme. Das erschwert das Planen der Logistik, da im Voraus keine konkrete Zahl der zu erwartenden Flaschen bekannt ist. Geht eine Flasche auf dem Postweg verloren, könnte das unbemerkt bleiben, da der Prozess für das Weingut mit dem Versand endet und der Prozess für WM erst mit dem Erhalt des der Flasche beiliegendem Formulars beginnt. Der Postweg stellt somit eine Lücke zwischen diesen Prozessen dar. Kommt ein teilnehmender Wein bei WM an, wird

das beiliegende Formular von Hand in eine Excel-Tabelle übertragen. Diese Schnittstelle ist besonders ressourcenaufwändig und fehleranfällig, da es oft vorkommt, dass die teils dysgraphisch verfassten Formulare nur schwer, mehrdeutig, oder gar nicht dechiffriert werden können. In diesem Prozess wird der Teilnahme des Weines eine inkrementell aufsteigende Einlieferungsnummer (ELN) zugewiesen, die anschließend in Form eines Aufklebers an der Flasche befestigt wird. Anschließend wird die Flasche im Lager verstaut.

### 1.2. Zielsetzung

Ziel dieser Arbeit ist es, in Erfahrung zu bringen, wie der zuvor genannte Prozess bestmöglichst, im Rahmen bestimmter Constraints und funktionalen- sowie nichtfunktionalen Anforderungen, digitalisiert werden kann. Während die Constraints bereits bekannt sind, werden detaillierte Anforderungen im Rahmen der Anforderungstechnik ausgearbeitet [Heinemann, 2021]. Die Constraints besagen, dass der Anmeldeprozess in die existierende Internetpräsenz des Weinverbandes integriert werden muss. Bei dieser Internetpräsenz handelt es sich um ein TYPO3-Redaktionssystem. Sämtliche Interaktionen zwischen Akteuren, die nicht WM oder dem System zugehörig sind, müssen im Frontend der Webseite stattfinden. Oberflächen für Mitarbeiter von WM dürfen in der TYPO3-Backend-Oberfläche implementiert werden.

Somit lautet die **Forschungsfrage**:

*Wie kann die Anmeldung und Zustellung von Weinen für Weinproben des Regionalverbunds für Weine in der Weinregion Mosel effizient und profitabel durch eine TYPO3-Erweiterung realisiert werden?*

## 2. Stand der Forschung

Der Stand der Forschung beleuchtet verschiedene Erkenntnisse zur Digitalisierung und zur digitalen Transformation.

### 2.1. Modell nach Parviainen et al.

*“The importance of digitalization is becoming understood, but the question now is how to do it in practice in order to best benefit from it.”*  
[Parviainen et al., 2022].

Parviainen et al. stellten sich diese Frage und entwickelten in ihrer Forschungsarbeit einen konzeptionellen Rahmen, um zu verstehen, wie die Digitalisierung in der Praxis umgesetzt werden kann und welche Vorteile sich daraus ergeben. Dieser Rahmen basiert auf dem Plan-Do-Check-Act (PDCA) -Prinzip.

Dieses Rahmenwerk sieht anhand des PDCA-Prinzips vier Schritte vor:

**Im ersten Schritt** wird definiert, wie weit die Digitalisierung für das Unternehmen gehen kann und welche Position das Unternehmen dabei anstrebt. Dieser Schritt kann in vier Teilschritte unterteilt werden: Ausmaße, Treiber, Szenarien und Ziele. Für die Bestimmung der Ausmaße ist die Analyse aktueller Trends und deren Relevanz für die Domäne des Unternehmens wichtig. Ebenfalls ist wichtig, wie weit diese Trends bereits im Fachgebiet verankert sind. Zur Einordnung eignen sich Strength-Weakness-Opportunity-Threat (SWOT) -Analysen. Diese Analysen sind die Grundlage, um ein Unternehmen in der Digitalisierung aufzustellen.

Aus den Ergebnissen der Trendanalysen sollten dann Treiber identifiziert werden. Diese Treiber sollten auf der Grundlage zukünftiger Ergebnisse skalierbar sein: Beispielsweise könnten drastische Maßnahmen erforderlich sein, um drastische Auswirkungen zu verhindern oder dras-

tische Verbesserungen zu erreichen.

Für die relevantesten Treiber sollten Zukunftsszenarien untersucht werden. Dies ist wichtig, um zu wissen, welche Auswirkungen bestimmte Trends in welcher Ausprägung haben werden. Relevant sind hier die Vorteile der Umsetzung des Szenarios, die Kosten der Umsetzung, sowie die Risiken, das Szenario nicht umzusetzen oder doch umzusetzen. Auf dieser Basis kann das beste Szenario ausgewählt werden.

Aus diesem Szenario werden schließlich die Ziele der Digitalisierung abgeleitet. Diese Ziele müssen so formuliert sein, dass sie mit der Ausgangssituation verglichen werden können.

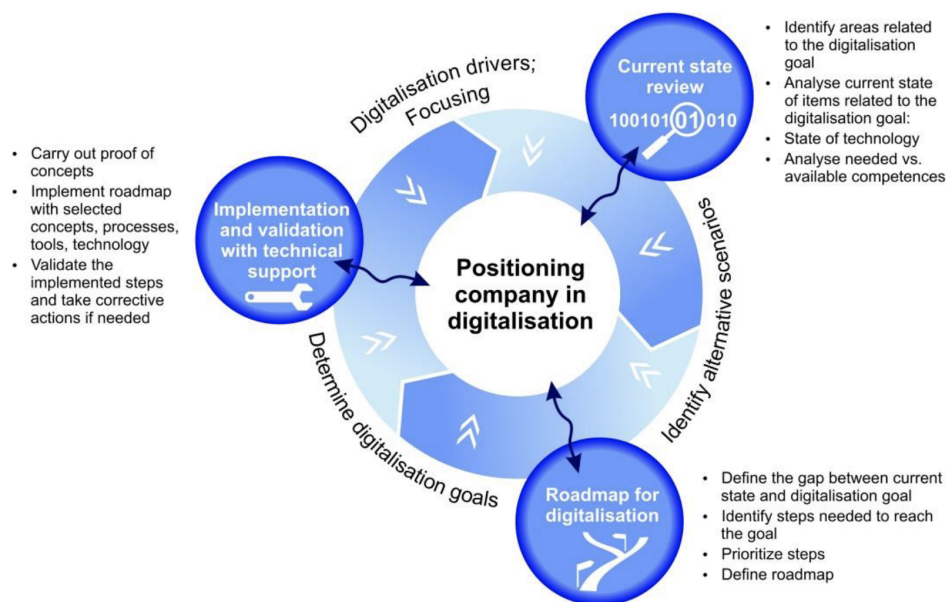


Abbildung 2.1.: Model for tackling digital transformation

Quelle: [Parviainen et al., 2022]

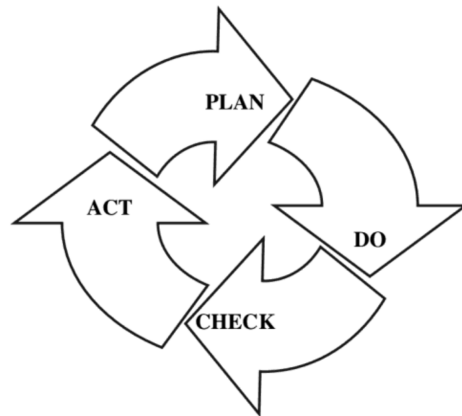


Abbildung 2.2.: Plan-Do-Check-Act (PDCA)

Quelle: [Abraham, 2005]

**Im zweiten Schritt** wird der Ist-Zustand des Unternehmens ermittelt. Dazu wird die aktuelle Positionierung des Unternehmens im Hinblick auf den Zielzustand mit Fokus auf die Digitalisierungsziele betrachtet. Dazu wird der Ist-Zustand im Kontext des Soll-Zustandes anhand definierter Fragen bewertet. Die Auswahl der Fragen unterscheidet sich je nach Art der Ziele. Der gesamte Fragenkatalog kann im Detail der Ausarbeitung von Parviainen et al. entnommen werden.

**Der dritter Schritt** ist die Festlegung der konkreten Schritte, die für den Übergang vom Ist-Zustand zum Soll-Zustand erforderlich sind. Dazu muss zunächst die Lücke zwischen dem Ist- und dem Soll-Zustand identifiziert werden. Relevant ist dabei der aktuelle Stand der Technik und welche Veränderungen notwendig sind, um den Zielzustand zu erreichen. Anschließend sollten die konkreten Schritte identifiziert werden, die erforderlich sind, um diese Lücke zu schließen. Wenn zum Beispiel ein Treiber “interne Effizienz” ist, könnten die Schritte darin bestehen, neue digitale Werkzeuge zu integrieren. Schließlich werden diese Schritte analysiert und priorisiert. Prädestiniert dafür sind Kosten-Nutzen-Analysen, Analysen der Umsetzbarkeit, des Wartungsaufwands und der

Mitarbeiterschulung.

**Der vierte Schritt** befasst sich mit der Umsetzung der in Schritt 3 geplanten Maßnahmen und der Bewertung der erzielten Ergebnisse. Diese Bewertung der Ergebnisse sollte z.B. soziokulturelle Barrieren berücksichtigen, die sich aus den Reaktionen bestimmter Stakeholder ergeben, die möglicherweise negativ auf Veränderungen reagieren oder Schwierigkeiten bei der Einführung neuer Technologien haben. Wenn diese Analyse zeigt, dass die Ziele der Digitalisierung nicht erreicht wurden, sollten Korrekturmaßnahmen in früheren Phasen ergriffen werden [Parviainen et al., 2022].

## 2.2. Modell nach Verhoef et al.

Nach Verhoef et al. lässt sich der hier sogenannte “Prozess der Digitalisierung” in drei Phasen unterteilen. Diese drei Phasen sind *Digitization*, *Digitalization* und *Digital Transformation* [Verhoef et al., 2021]. Die Phase *Digitization* befasst sich mit der Umwandlung analoger Datenstrukturen und Modellen in digitale Datenmodelle, sodass diese digital, in Form von Nullen und Einsen, gespeichert und elektronisch weiterverarbeitet werden können [Dougherty and Dunne, 2012, Loebbecke and Picot, 2015]. *Digitalization* beschreibt den Prozess der Veränderung bestehender Geschäftsprozesse, um mit digitalen Werkzeugen und Datenmodellen zu arbeiten [Li et al., 2016]. Die letzte Phase, die *Digitale Transformation*, beschreibt eine firmenweite Veränderung, die beispielsweise Ergründungen neuer Geschäftsmodelle mit sich bringen könnte [Pagani and Pardo, 2017].

## 2.3. Abwägung in Bezug auf die Problemstellung

In Bezug auf die hier betrachteten Methoden ist es wichtig zu erwähnen, dass der betrachtete Kontext lediglich die Digitalisierungs **eines** Geschäftsprozesses behandelt. Diese Ausarbeitung befasst sich nicht mit

firmenweiten Veränderungen, wie sie von den nahegelegten Modellen abgedeckt ist. Daher sind geringfügige Anpassungen der Methoden unabdinglich. Des Weiteren ist Ressourcenintensivität ein relevanter Gesichtspunkt einer Abwägung, da es ein Projektbudget einzuhalten gilt.

### 2.3.1. Parviainen et al.

Der Autor stellt fest, dass das Modell nach Parviainen et al. hauptsächlich methodisch ausgerichtet ist und für den betrachteten Kontext der Problemstellung unverhältnismäßig ressourcenintensiv ist, da der Umfang der Anforderungen geringfügig ist. Dieses Modell legt den Fokus auf firmenweite Veränderungen anstatt auf einzelne Prozesse und könnte daher in Bezug auf diese Problemstellung als “überdimensioniert” bezeichnet werden.

### 2.3.2. Verhoef et al.

Im Modell nach Verhoef et al. stellt die hier behandelte Problemstellung lediglich Phase eins und Phase zwei der drei Phasen *Digitization*, *Digitalization* und *Digital Transformation* dar. Das ist so, da sich die ersten zwei Phasen mit der Digitalisierung bestimmter Geschäftsprozesse befassen. Phase drei befasst sich mit firmenweiten Veränderungen, die über Geschäftsprozesse hinausgehen [Verhoef et al., 2021]. Firmenweite Veränderungen befinden sich außerhalb des Rahmens dieser Ausarbeitung. Der Autor empfindet das Modell nach Verhoef et al. als agiler und unrestrictiver.

### 2.3.3. Fazit

Aufgrund des kleinen Umfangs der Problemstellung und des kleinen Projektbudgets fällt die Entscheidung auf das kleiner gefasste Modell nach Verhoef et al.. Ziel dieser Entscheidung ist es, vermeidbare Kosten, die keine eminenten Vorteile bringen, zu vermeiden. Nachdem Phase drei des Verhoefer-Modells ausgeklammert wurde, sieht das zu verfolgende Modell aus wie folgt:

**Umsetzungsplanung nach  
Verhoef et al., mit Anpassungen**

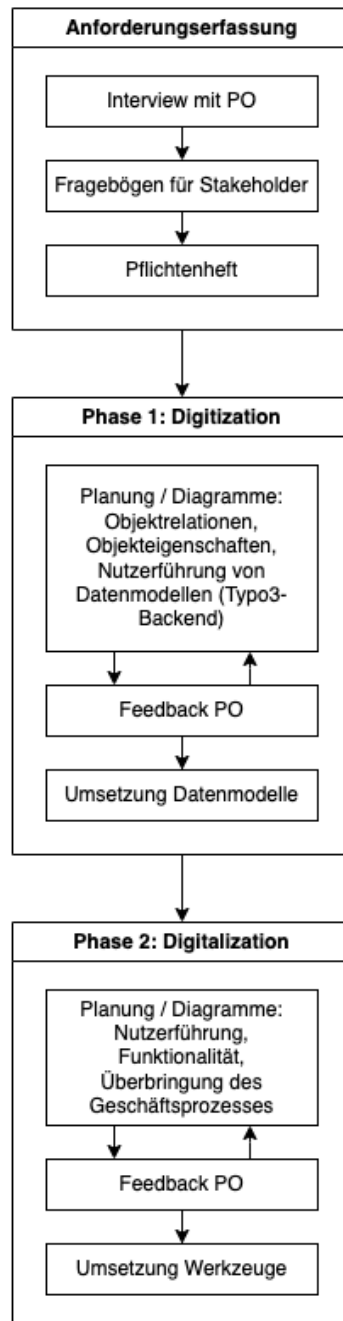


Abbildung 2.3.: Umsetzungsplanung

Quelle: Eigene Darstellung

## 3. Stand der Technik

Der Stand der Technik bezieht sich auf bestehende, praktische Umsetzungen der erforderlichen Technologien. Im Wesentlichen gibt es drei Arten von Technologien, die untersucht werden müssen: Die bestehende Website von WM, Bibliotheken zur Erzeugung von QR-Codes und Bibliotheken zur Erzeugung von PDF-Dateien.

### 3.1. Die bestehende Webseite

Als Mitentwickler des Projektes ist dem Autor bekannt, dass die bestehende Webseite ein TYPO3-Redaktionssystem ist. Das Frontend der Webseite wird mit Webpack und Sass übersetzt. Webpack ist ein Modulbundler [Zimmermann, 2017] und Sass ein CSS-Präprozessor [W3Schools, 2023b]. TYPO3 ist ein Redaktionssystem und PHP-Rahmenwerk, das Daten- und Inhaltspflege in einem geschützten Bereich ermöglicht, Frontend- und Backend-Nutzer und Berechtigungen steuert [TYPO3 Contributors, 2023g] und über die Systemerweiterung “Extbase” eine mächtige Entwickler-schnittstelle bereitstellt, um hochindividualisierte Funktionalitäten zu ermöglichen [TYPO3 Contributors, 2023c].

### 3.2. QR-Code-Bibliotheken

Um mit QR-Codes zu arbeiten, ist es unabdinglich, QR-Codes zu erstellen, da dieselben sonst nicht vorhanden sind. Im Folgenden werden einige Implementationen von QR-Code-Generator-Bibliotheken im Detail betrachtet. Es wird sich auf bereits verwendete Programmiersprachen begrenzt.

### 3.2.1. Javascript-Implementationen

#### Jquery-qrcode

*Jquery-qrcode* ist ein Plugin für JQuery um dynamisch QR-Codes auf Browserseite zu generieren. Jedoch verweist diese Bibliothek selbst auf ihren desolaten Zustand und empfiehlt stattdessen “kjua” zu verwenden [Jung, 2020a]. Somit scheidet jquery-qrcode für nähere Evaluationen aus, da die Software nicht mehr gepflegt wird. *Jquery-qrcode* wurde bis Januar 2023 853 mal von Nutzern mit einem Stern markiert [Jung, 2020a]. Wenn ein Nutzer ein Github-Repository mit einem Stern markiert, zeigt sich der Nutzer interessiert genug, um über Neuigkeiten auf dem Laufenden gehalten werden zu wollen [Github, 2023]. Jquery-qrcode ist MIT-lizenziert [Jung, 2020a].

#### Kjua

*Kjua* ist eine Javascript-Bibliothek, um dynamisch QR-Codes auf der Browserebene zu generieren. Im Gegensatz zu *jquery-qrcode* funktioniert Kjua auch ohne JQuery. Es werden diverse Stilattribute für gestaltete QR-Codes unterstützt [Jung, 2020b]. Kjua kann QR-Codes über HTML-Canvas, Bilder und Vektorgrafiken umsetzen. Das ist bei näherer Betrachtung der Kjua Tech Demo (<https://larsjung.de/kjua/latest/demo>) ersichtlich, jedoch nicht explizit erwähnt. Nach dem aktuellen Stand (Januar 2023) wurde Kjua von 243 Nutzern mit einem Stern markiert. Kjuas letzte Codeänderung liegt circa zwei Jahre zurück. Es gibt eine unbeantwortete, sieben Jahre alte Pull-Request und Issues scheinen ignoriert zu werden. Kjua ist MIT-lizenziert [Jung, 2020b].

#### Soldair/node-qrcode

*Soldair/node-qrcode* ist eine node.js-basierte Implementation eines QR-Code-Generators und bietet somit Funktionalität serverseitig, als CLI, sowohl auch Browserseitig an. Die Readme-Datei zeugt von Länge, ist reich an Beispielen und detailreichen Erklärungen. Der letzte Commit ist zu diesem Zeitpunkt knapp älter als ein halbes Jahr. Somit macht

das Projekt einen moderat gepflegten Eindruck. Die Readme-Datei verweist auf Unit Tests bei Travis, jedoch lief die letzte Pipeline vor circa zwei Jahren, Februar 2021, durch und schlug fehl. Einige Pull-Requests und Issues werden seit Jahren ignoriert [Day, 2020]. Die Bibliothek wurde 74 Millionen mal heruntergeladen und mit 6308 Sternen markiert. Soldair/node-qrcode ist MIT-lizenziert [npm, inc, 2020].

#### 3.2.2. PHP-Implementationen

##### Chillerlan/php-qrcode

Eine arrivierte Lösung für QR-Code-Generierung in PHP ist *chillerlan/php-qrcode*, umgesetzt durch Eugen Rochko et al.. *Chillerlan/php-qrcodes* Etabliertheit zeigt sich durch den zu diesem Zeitpunkt (Jahnuar 2023) auf 5 Millionen stehenden Downloadzähler, mehreren Entwickler und den letzten Commit, der zu diesem Zeitpunkt nicht älter als einen Monat ist. Insgesamt erfolgten bis dato 808 Commits von 6 Entwicklern. Das Projekt verfügt über Unit-Tests, die 90% der Zeilen der Codebase abdecken. Rochko übernahm Teile der Codebase aus dem Java-Projekt “ZXing” und übersetzte diese zu PHP. Es gibt zu diesem Zeitpunkt keine unbeantworteten Issues oder Pull-Requests. *Chillerlan/php-qrcode* basiert auf einer angepassten Version von *kazuhikoarase/qrcode-generator*. Einzig auffällig sind die Commitnachrichten, die zuteils nur aus einem (1) Emoji bestehen. *Chillerlan/php-qrcode* wurde von 1212 Github-Nutzern mit einem Stern markiert. Die Bibliothek ist MIT-lizenziert [Rochko, 2022b].

##### Kreativekorp/barcode

*kreativekorp/barcode* ist eine PHP-Bibliothek zur Generierung von QR-Codes, bereitgestellt von *Kreative Software*, R.G. Bettencourt. Diese Implementation umfasst etablierte Barcode-Formaten und unterstützt eine Vielzahl an Anpassungsmöglichkeiten. Das Projekt wurde bis zum heutigen Tage 189 mal mit einem Stern markiert [Bettencourt, 2018]. Vier von den sechs vorhandenen Commits erfolgten innerhalb von zwei Tagen

in 2016. Ein weiterer Commit erfolgte wenige Tage später. Der aktuellste Commit wurde knapp zwei Jahre später, 2018, veröffentlicht. Das liegt zu diesem Zeitpunkt vier Jahre in der Vergangenheit. Damit ist diese Bibliothek de-facto sechseinhalb Jahre alt und wurde seitdem einmal um Features erweitert. Issues und Pull-Requests werden weitestgehend ignoriert. Die Bibliothek verwendet die MIT-Lizenz [Bettencourt, 2018].

#### **Bacon/BaconQrCode**

*Bacon/BaconQrCode* ist eine PHP-Bibliothek zur Generierung von QR-Codes, bereitgestellt von Ben Scholzen, hinter der Github-Organisation “Bacon”, dessen einziges Mitglied Scholzen darstellt. Verlinkt ist eine Homepage, die zu einer Nginx-“Hello World”-Seite führt. Begleitet wird *BaconQrCode* von etlichen weiteren “Bacon-Projekten” wie Beispielsweise *BaconPdf*, *BaconStringUtils* und *BaconUser* um nur einige zu nennen. Insgesamt machen die stichprobenartig betrachteten Projekte einen desolaten Eindruck mit zuteils aktuellsten Commits von vor zehn Jahren. BaconQrCode stellt das beliebteste und gepflegteste Projekt von Scholzen mit 1508 Sterne-Markierungen und einem aktuellsten Commit von vor zwei Monaten dar. BaconQrCode fällt mit siebzehn Entwicklern auf, die jeweils zumindest einen Commit beigetragen haben. Zu diesem Zeitpunkt fanden 177 Commits statt. Githubs Dependency-Graph verzeichnet nahezu 80.000 Projekte, die BaconQrCode verwenden [Scholzen, 2018] und Packagist meldet 50 Millionen Downloads. Wie *chillerlan/php-qrcode* baut auch Scholzen auf existierende Technik von “ZXing” auf. [Packagist, 2023]. Eine Dokumentation neben der Readme-Datei existiert nicht und diese ist sehr minimalistisch. *BaconQrCode* kann QR-Codes als Rasterbilder und Vektorgrafiken (SVG und EPS) generieren. Spezielle Styles sind nicht erwähnt. Ein Großteil der Issues und Pull-Requests wurden behandelt. BaconQrCode unterliegt einer BSD-2-Clause-Lizenz [Scholzen, 2018].

### 3.2.3. Subjektiver Vergleich im Bezug auf die Problemstellung

Im Folgenden werden subjektive Einschätzungen und Meinungen des Autors über die Eignung der beleuchteten Bibliotheken vorgestellt. Um eine Bibliothek als “am geeignetsten” für einen Anwendungsfall zu kurieren, müssen die konkreten Anforderungen und Constraints für diesen Anwendungsfall beachtet werden. Das ist so, da verschiedene Eigenschaften der Bibliotheken verschiedene Auswirkung in Gewichtung und Richtung je nach Anwendungsfall aufweisen. Hierfür werden die zuvor vorgestellten Bibliotheken zur Erstellung von QR-Codes in den folgenden Attributen gegenübergestellt:

#### **Funktionalität**

Der Umfang, der für diese Problemstellung relevanten Funktionen in Annahme dessen, dass die Bibliothek syntaktisch und pragmatisch korrekt [Heinemann, 2021] ist.

#### **Gepflegtheit**

Das Ausmaß, in dem das Projekt aktiv gepflegt wird und ordnungsgemäß entwickelt zu sein scheint. Hierzu zählen beispielsweise: Bearbeitung von Issues, Bearbeitung von Pull-Requests, Präsenz von Tests, Präsenz einer angemessenen Dokumentation, häufige Commits, mehrere Entwickler, Anzahl der Sterne auf Github (Ausmaß an tieferem, öffentlichem Interesse [Github, 2023]), sowie der Anzahl der Downloads bzw. Installationen, falls verfügbar.

#### **Workflow-Eignung**

Die Eignung einer Bibliothek in existierende Workflows und Constraints übernommen zu werden. Maßgeblich, ob und mit wie viel Aufwand eine Bibliothek in das Projekt übernommen werden kann. Ebenfalls ist relevant, ob die Lizenz einer Bibliothek eine Verwendung gestattet, bzw. welche Bedingungen gelten.

Hierbei werden den verschiedenen Bibliotheken Punkte  $([0, 10])$  in den drei zuvor genannten Kategorien vergeben. Nicht ermittelte Werte wer-

den in ihrer Kategorie durch – repräsentiert, wodurch eine weitere Verwendung ausgeschlossen wird. Die Kumulativpunktzahl  $([0, 30])$  einer Bibliothek beschreibt deren Gesamteignung, nach subjektivem Empfinden des Autors.

#### Kjua

Kjua ist funktional für dieses Projekt gut aufgestellt, da es optisch ansprechende QR-Codes mit Logo unterstützt. Das wird mit acht Punkten in “Funktionalität” bemessen. Kjua macht andererseits einen eher desolaten Eindruck und lässt somit begründete Zweifel an dessen Zukunftssicherheit zu. Eine knappe Dokumentation ist verfügbar. Für die Dokumentation werden zwei Punkte in “Gepflegtheit” angerechnet. Kjua ist als Javascript-Bibliothek nur schwer mit den Anforderungen vereinbar, da der QR-Code in einem PDF eingebunden werden soll. Hierfür ist ein serverseitiger PDF-Generator vorgesehen. Kjuas Lizenz erlaubt Verwendung in kommerziellen, Closed-Source-Projekten [Opensource.org, b]. Die Exklusivität für Nutzung in Webbrowsern schließt eine Einbindung in den Workflow aus.

| Funktionalität | Gepflegtheit | Workflow-Eignung | $\Sigma$ |
|----------------|--------------|------------------|----------|
| 8              | 2            | 0                | 10       |

Tabelle 3.1.: Subjektive Evaluation: Kjua

#### Soldair/node-qrcode

Soldairs Lösung sieht dokumentativ und funktional vielversprechend aus. Hieraus bilden sich acht Punkte in “Funktionalität”. Da diese Bibliothek eine Node.js-Bibliothek ist und Node.js ohne Browser, als Kommandozeilenapplikation, ausgeführt werden kann [OpenJS Foundation, 2023], bietet sich Soldair/node-qrcode als serverseitige Anwendung an. Die Lizenz spricht nicht gegen eine Verwendung. Dennoch wäre es ein vermeidbarer Mehraufwand generierte QR-Codes aus einer Node.JS-Applikation in eine PHP-Umgebung zu übertragen. Das bildet sich mit vier Punkten

in “Workflow-Eignung” ab. Darüberhinaus macht Soldair/node-qrcode einen verbesserungswürdigen Eindruck der Projektpflege, wofür es lediglich drei Punkte gibt.

| Funktionalität | Gepflegtheit | Workflow-Eignung | $\Sigma$ |
|----------------|--------------|------------------|----------|
| 8              | 3            | 4                | 15       |

Tabelle 3.2.: Subjektive Evaluation: Soldair/node-qrcode

### Chillerlan/php-qrcode

Rochkos Lösung macht einen aktiv gepflegten Eindruck und wird von großen Downloadzahlen gestützt. Rochko stellt eine ausführliche Dokumentation und Codebeispiele bereit. Hierfür wird die volle Punktzahl in “Gepflegtheit” vergeben. Der Funktionsumfang übersteigt weit die Anforderungen und gestattet so ausführlich gestaltete QR-Codes [Rochko, 2023], die somit ebenso die hedonische Qualität fördern [König, 2022]. Bewertet wird das mit einer vollen Punktzahl in “Funktionalität”. Chillerlan/php-qrcode lässt sich nahtlos in das existierende Projekt eingliedern, da es als PHP-Bibliothek über Composer eingebunden werden kann und eine eigene API bereitstellt [Rochko, 2022b]. Die von Rochko verwendete Lizenz gestattet eine unkomplizierte Verwendung. Chillerlan/php-qrcode funktioniert sowohl mit PHP 7.x als auch 8.x. Die Bibliothek benötigt zwei weitere Abhängigkeiten. Eine dieser Abhängigkeiten ist ebenfalls von Rochko bereitgestellt [Rochko, 2022c] und weist eine ähnlich gute Projektpflege auf [Rochko, 2022a]. Das wird mit der Maximalwertung in “Workflow-Eignung” berechnet.

| Funktionalität | Gepflegtheit | Workflow-Eignung | $\Sigma$ |
|----------------|--------------|------------------|----------|
| 10             | 10           | 10               | 30       |

Tabelle 3.3.: Subjektive Evaluation: Chillerlan/php-qrcode

### Kreativekorp/barcode

Kreativekorp beeindruckt durch Nutzungsbeispiele und Dokumentation in der Readme-Datei, sowie einer Vielzahl unterstützter Barcode-Formate, darunter auch QR-Codes und einiger improvisierter Tests. In Anbetracht dessen, dass die Bibliothek de-facto sechseinhalb Jahre alt ist und seit vier Jahren nicht mehr angepasst wurde, wird eine geringe Wertung von drei Punkten in “Gepflegtheit” vergeben. Null Punkte in “Workflow-Eignung” rechtfertigen sich durch die Abwesenheit jeglicher Unterstützung für Paketmanager, wodurch eine saubere Verwendung in dem Brown-Field-Projekt Änderungen an der Bibliothek selbst erfordern würde. Die Funktionalität wurde aufgrund der desaströsen Gepflegtheit und Eignung nicht näher untersucht, da eine Verwendung selbst mit guter Funktionalität nicht infrage käme.

| Funktionalität | Gepflegtheit | Workflow-Eignung | $\Sigma$ |
|----------------|--------------|------------------|----------|
| -              | 3            | 0                | -        |

Tabelle 3.4.: Subjektive Evaluation: Kreativekorp/barcode

### Bacon/BaconQRCode

BaconQRCode nennt keine speziellen Optionen, um näheren Einfluss auf den generierten QR-Code auszuüben. Es werden Rasterbilder und verschiedene Vektorformate als Renderziel unterstützt. Aufgrund der fehlenden Optionen, die andere Bibliotheken bereitstellen, werden lediglich fünf von zehn Punkten für “Funktionalität” vergeben. Die Projektgepflegtheit ist inkonsistent. Manche Stellen, beispielsweise die Mitwirkenden, Alter des neuesten Commits, Tests und Nutzerzahlen machen einen guten Eindruck, während Punkte wie die Dokumentation, die Organisationswebseite und andere Projekte Sorgen begründen. Weil der Programmcode an sich gut gepflegt aussieht und große Downloadzahlen von häufiger Verwendung sprechen, werden für die “Gepflegtheit” acht Punkte vergeben. Da es sich hierbei um eine PHP-Bibliothek handelt, die über Composer in PHP- 7.x und 8.x Projekte geladen werden kann und eine

das Projekt API bereitstellt, ist die technische Workflow-Eignung gut. Die BSD-2-Clause-Lizenz verkompliziert eine Integration, da somit eine Copyright-Notiz an Nutzer gezeigt werden muss [Opensource.org, a]. Dadurch werden drei Punkte einer vollkommenen Workflow-Eignung abgezogen, wodurch sieben Punkte vergeben werden.

| Funktionalität | Gepflegtheit | Workflow-Eignung | $\Sigma$ |
|----------------|--------------|------------------|----------|
| 5              | 8            | 7                | 20       |

Tabelle 3.5.: Subjektive Evaluation: Bacon/BaconQRcode

### 3.2.4. Fazit

Nach Evaluation der verschiedenen QR-Code-Bibliotheken im Kontext der vorliegenden Problemstellung erweist sich aus Sicht des Autors *chillerlan/php-qrcode* mit 30 Gesamtpunkten als die am besten geeignetste Bibliothek. Somit wird *chillerlan/php-qrcode* als QR-Code Technologie in der Lösung dieser Problemstellung verwendet.

| Bibliothek            | Funkt. | Gepflegtht. | WF.-Eignung | $\Sigma$ |
|-----------------------|--------|-------------|-------------|----------|
| chillerlan/php-qrcode | 10     | 10          | 10          | 30       |
| baconqrcode           | 5      | 8           | 7           | 20       |
| soldair/node-qrcode   | 8      | 3           | 4           | 15       |
| kjua                  | 8      | 2           | 0           | 10       |
| kreativekorp/barcode  | -      | 3           | 0           | -        |

Tabelle 3.6.: Gesamtübersicht: Subjektive Evaluation der QR-Code Bibliotheken

## 3.3. PDF-Generator

Firmenintern ist der PDF-Generator *mpdf/mpdf* [Mpdf Contributors, 2023a] etabliert und wird bereits in zahlreichen Projekten verwendet. Um die projektübergreifende Kontinuität der Technik zu wahren und um somit die Wartbarkeit und Nachhaltigkeit des hier behandelten Softwareproduktes zu fördern, wird sich für den PDF-Generator entschieden, der

bereits firmeninterner Standard ist. Unabhängig dessen ist *mpdf* ein gut gepflegtes Projekt mit einem Alter von mehr als acht Jahren, Sponsoren, 72 Entwicklern, über 31 Millionen Downloads, über 3.900 Sterne-Markierungen, über 800 Commits und regelmäßigen Updates. Dadurch, dass *mpdf* ein Composer-Paket für verschiedene PHP-Versionen ist, ist eine herausragende Workflow-Eignung gegeben [Mpdf Contributors, 2023a].

## 4. Anforderungserfassung

Obwohl bereits ein grober Anriss des Zielsystems bekannt ist, ist es unabdinglich eine Anforderungsanalyse durchzuführen, dies um Details auszuarbeiten [Ebert, 2019]. Hierbei ist es wichtig, kein exzessives Pflichtenheft aufzubauen, denn letztendlich zählt nur, was dem Kunden geliefert wird. Nicht, wie viele gar nicht benötigte Anforderungen umgesetzt wurden. *“Zu viele oder falsche Anforderungen ruinieren Budgets, Termine und die Qualität.”* [Ebert, 2019]. Die Anforderungen eines Produktes sind in drei Kategorien einzuteilen: Funktionale Anforderungen (FAs), nichtfunktionale Anforderungen (NFAs) und Constraints [Heinemann, 2021]. Wie oben erwähnt, sind bereits die Constraints und einige funktionale und nichtfunktionale Anforderungen bekannt. Diese sind:

| Art der Anforderung | Beschreibung   |
|---------------------|--|
| Constraint          | Einbau in brown-field TYPO3  |
| Constraint          | Mitarbeiter-UI in TYPO3-Backend  |
| Constraint          | Nutzer-UI im Frontend  |
| NFA                 | Aus dem Papierformular soll ein Webformular werden   |
| NFA                 | Angaben zum Weingut des Weines sollen aus dem Accountdatensatz anstatt aus dem Webform kommen                                      |
| NFA                 | Registrierte Weiteilnahmen bestehen aus einem Wein und einem Zustand (Ausstehend, Eingegangen, Abgelehnt).                         |
| FA                  | Beim Erstellen einer Weiteilnahme soll ein QR-Code als PDF generiert werden, der den Wein identifiziert                            |
| FA                  | Durch scannen des QR-Codes auf dem mit einer Weiteilnahme erstellten PDFs soll dem Wein der Status “Eingegangen” zugewiesen werden |

Tabelle 4.1.: Initial bekannte Anforderungen

Um nähere Anforderungen zu ermitteln, werden die Befragungstechniken “Interview” und “Fragebogen” verwendet [Heinemann, 2021].

### 4.1. Interview mit Product Owner

Zunächst wird ein Interview mit dem Product Owner (PO) geführt. Ziel dieses Interviews ist es, konkrete Fragen zu Anforderungen zu beantworten und somit konkrete Anforderungen zu formulieren. Aufgrund der individuellen Gesprächsführung wurde sich für ein “Teil-standardisiertes Interview” entschieden. Bei einem Teil-standardisiertem Interview gibt es vordefinierte Fragen, aber auch Freiraum für Improvisation und Persönlichkeit. Für ein gutes Interview ist gute Vorbereitung unabdinglich. Daher wurden bereits sämtliche wichtigen Fragen in einem Fragebogen festgehalten. Dieser Interview-Fragebogen liegt in  $\langle\langle B \text{ Interview-Fragebogen} \rangle\rangle$  anbei. Ebenso ist Vorbereitung auf Seiten des Interviewteilnehmers wichtig, weshalb das Interview einen Tag zuvor angesprochen wurde. Um möglichst objektive und unvorbeeinflusste Antworten zu gewährleisten wird darauf geachtet, keine Suggestivfragen zu stellen [Die SOPHISTen, 2016].

### 4.2. Online-Fragebögen für Stakeholder

Um Fragebögen für Stakeholder formulieren zu können, muss zunächst bekannt sein, wer die Stakeholder sind. “*Ein Stakeholder eines Systems ist eine Person oder Organisation, welche (direkt oder indirekt) Einfluss auf die Anforderungen des betrachteten Systems hat.*” [Pohl and Rupp, 2015]. Daraus ergeben sich die Stakeholdergruppen: “Mitarbeiter WM” und “teilnehmende Weingüter”. Jede dieser Stakeholdergruppen sieht das System aus einer anderen Perspektive [Die SOPHISTen, 2016]. Daher ist es wichtig, Einblicke und Bedürfnisse aller Stakeholdergruppen einzuholen und die individuellen Perspektiven und Bedürfnisse dieser beim Entwurf der Fragebögen zu berücksichtigen. Ebenso ist es wichtig, die wichtigsten Fragen am Anfang zu stellen, da Formulare nicht immer

vollständig ausgefüllt werden [Die SOPHISTen, 2016]. Somit sind auch bei einem nur teilweise ausgefüllten Fragebogen die wichtigsten Fragen beantwortet. Sämtliche Fragen an die Stakeholdergruppe “Mitarbeiter WM” wurden bereits im Interview mit dem PO beantwortet und als Anforderungen festgehalten. Der PO repräsentiert in diesem Falle die Stakeholdergruppe “Mitarbeiter WM” und steht seit geraumer Zeit mit WM im persönlichen, engen Austausch. Daher gibt es keine offenen Fragen, die diese Stakeholdergruppe beantworten könnte. Somit fällt ein Onlinefragebogen für die Stakeholdergruppe “Mitarbeiter WM” weg. Der Fragebogen der Stakeholdergruppe “teilnehmende Weingüter” liegt im Anhang unter  $\langle\langle C \text{ Online-Fragebogen Weingüter} \rangle\rangle$  bei. Um den Aufwand und somit die Hemmschwelle des Ausfüllens eines solchen Online-Fragebogens zu minimieren, gibt es lediglich sechs Ankreuz-Fragen zusätzlich einem optionalen Freitextfeld, um sonstige Wünsche zum Ausdruck zu bringen. Eine Anmeldung über Google ist nicht erforderlich. Der Link zu diesem Online-Fragebogen wurde WM mit der Bitte um Weiterleitung an die Probandenteilnehmer übergeben.

### 4.3. Ergebnisse

Aus dem Interview mit dem PO ergibt sich ein Pflichtenheft. Das Pflichtenheft und das Protokoll zum Interview sind im Anhang unter je  $\langle\langle E \text{ Pflichtenheft} \rangle\rangle$  und  $\langle\langle D \text{ Interview-Protokoll} \rangle\rangle$  zu finden.

Ergebnis dieses Interviews sind zahlreiche Anforderungen und Ideen. Eine der wichtigsten Ideen stellt das projektbezogene, wöchentliche Statusmeeting dar: Jeden Donnerstag soll um 9:30 Uhr der aktuelle Stand des Projektes vorgestellt, diskutiert und Rücksprache gehalten werden. Die wichtigsten Erkenntnisse des Interviews sind:

#### **Endgerät für Scanning und Scanneranwendung**

Gescannt wird von Mobiltelefonen mit einer QR-Code-App wie QRBot. QRBot ermöglicht es Nutzern für jeden aufgerufenen QR-Code eine Vorlagen-URL aufzurufen, um den gescannten Wert als Teil der Url, z.B. als Get-Parameter, zu übergeben [TeaCapps, 2023].

Das ist prädestiniert für API-ähnliche Webseitenaufrufe, um Weine einzuchecken.

#### **Trennung von Weinen nach JAPs im Frontend**

Da es  $n$  JAPs gibt und Weine immer genau einer JAP zugeordnet sein müssen, macht es wenig Sinn alle Weine eines Nutzers auf einmal anzuzeigen. So ist es eine Anforderung, dass die Weinansicht in zwei Ebenen unterteilt ist: Die erste Ebene soll eine Auflistung aller JAP sein und in der Einzelansicht der JAPs sollen alle Weine aufgelistet sein, die dieser JAP angehören. Diese Weine sind ebenso anklickbar und führen zu einer Einzelansicht der Weine.

#### **Genaue Nutzerführung für Teilnehmer**

Die exakten Schritte, die ein Teilnehmer durchlaufen muss, um einen Prozess zu durchlaufen. Hierbei betrachten wir den Registrierungsprozess, um ein Teilnehmer zu werden und den Anmeldeprozess eines Weines. Diese Prozesse sind detailliert in *⟨5 Umsetzung⟩* beschrieben.

#### **Nutzerführung für WM-Angestellte**

Gescannte QR-Codes von Weinen sollen den Wein als eingegangen markieren und anschließend dem Mitarbeiter zeigen, welcher Wein eingchecked wurde. Somit dient das Scannen eines Codes ebenso zur Einsicht der Details der gelagerten Flaschen. Sonstige Aktionen sind im TYPO3-Backend mit nativen Werkzeugen erreichbar.

#### **Verschiedene Web-Ansichten**

- Jahreauswahlproben-ListView
- Jahreauswahlproben-SingleView (Hier sind u.a. Weine gelistet)
- Wein-Registrierungsformular
- PDF-Url für Versandbriefe
- Registrierungsseite mit mehreren Schritten

- Mitarbeiter-Seite für gescannte QR-Codes

Der Online-Fragebogen für teilnehmende Weingüter wurde über einen Monat hinweg nicht beantwortet, insofern gibt es keine Ergebnisse zu präsentieren.

## 5. Umsetzung

Infolge der Anforderungsanalyse befasst sich das Kapitel “Umsetzung” mit der Implementation der Anforderungen in dem Brown-Field Projekt [Schwarzer, 2021] in Form einer TYPO3-Extension.

### 5.1. Setup einer TYPO3-Extension

TYPO3-Extensions werden via Composer installiert [TYPO3 Contributors, 2023h]. Um eine TYPO3-Extension zu erstellen, muss also ein Composer-Paket erstellt werden. Um vermeidbare Komplexität zu verhindern, wird das Composer-Paket, welches die hier betrachtete TYPO3-Extension darstellt, lokal in den versionierten Ordner “packages” gelegt. Dieses Verzeichnis wird als Quelle für Composer-Pakete in der Haupt-composer.json-Datei hinterlegt. Somit wird ein Composer-Paket nur für dieses Projekt bereitgestellt, ohne den Aufwand zu haben, der üblicherweise mit dem Bereitstellen eines Paketes einhergeht.

Um das grundlegende Setup einer Extension effizient durchzuführen, wurde eine existierende Extension mit vergleichbarem Funktionsumfang kopiert, umbenannt und eingefügt. Spezifisch ist der “vergleichbare Funktionsumfang”, dass es Datenmodelle und hochpersonalisierte Frontendlogik in Bezug auf die zuvor genannten Datenmodelle gibt.

### 5.2. Digitization

Die Phase der Digitization nach Verhoef et al. befasst sich mit der digitalen Abbildung von Objekten der realen Welt in einer Art und Weise, so dass diese elektronisch weiterverarbeitet werden können [Verhoef et al., 2021, Dougherty and Dunne, 2012, Loebbecke and Picot, 2015]. Das bedeutet, dass in dieser Phase Datenobjekte definiert und implementiert werden. Ein Datenobjekt besteht nach firmeninternen Konventionen aus

zumindest vier Komponenten:

Datenbanktabelle

Die Datenbanktabelle persistiert Informationen.

Domain Model

Das Domain Model (auch Model genannt) ist eine PHP-Klasse, die jeweils die Daten einer Zeile der Datenbanktabelle abbildet.

Repository

Ein Repository ist eine PHP-Klasse, die die Schnittstelle zwischen der Datenbank und der Model-Klasse darstellt.

Table configuration array (TCA)

Der TCA des Modells definiert, wie diese Objekte im TYPO3-Backend dargestellt werden und bearbeitbar sind.

[TYPO3 Contributors, 2023c].

Im Folgenden wurde ein semiformales Diagramm der Objekte und ihren Relationen angefertigt und in Rücksprache mit dem PO finalisiert.

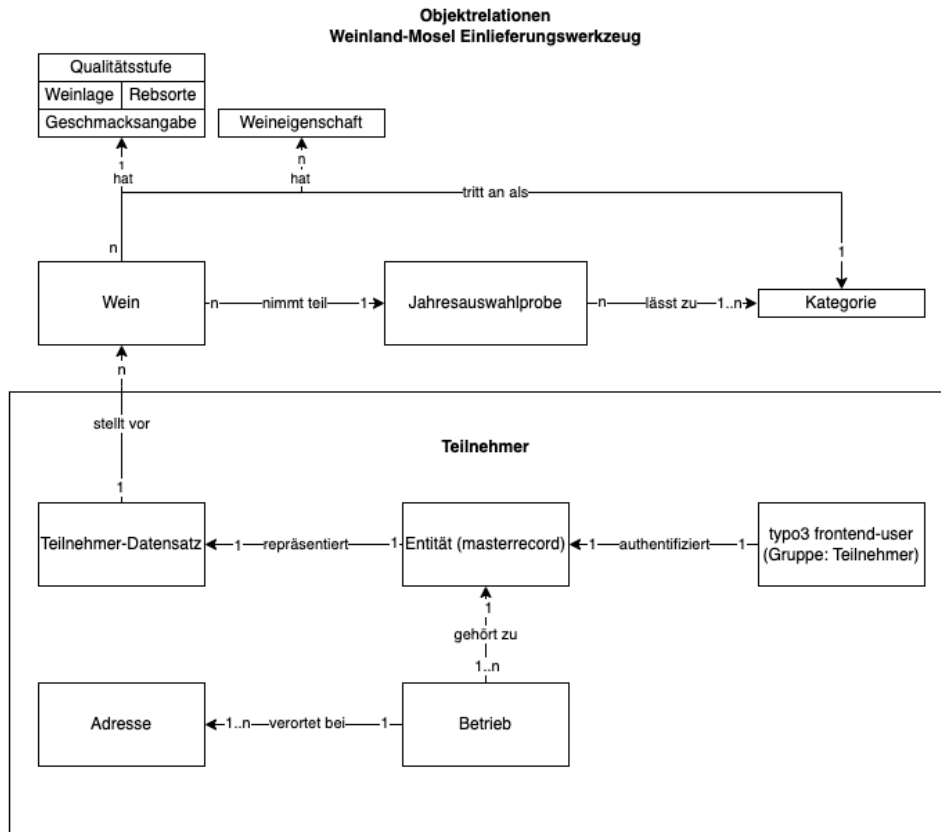


Abbildung 5.1.: Objektrelationen

Quelle: Eigene Darstellung

Nachdem in Erfahrung gebracht wurde, welche konkreten Datenobjekte benötigt werden, wurden Attribute dieser Objekte dem Pflichtenheft entnommen. Diese wurden in einem formalen Klassendiagramm festgehalten und in Rücksprache mit dem PO weiter bis zu festen Datentypen und Auswahlmöglichkeiten konkretisiert. Beispielsweise, dass Wettbewerbskategorien durch TYPO3-Categories repräsentiert werden sollen. Das hat den Vorteil, dass TYPO3-Categories bereits native Bestandteile eines TYPO3-Redaktionssystems sind und alle relevanten Attribute anbieten. Diese sind ein Titel, eine Parentkategorie und eine Beschreibung. Das Parent-Attribut ist benötigt, da  $n$  dieser Attribute einen Attributbaum bilden [TYPO3 Contributors, 2023i]. Somit ist es möglich,

Unterkategorien zu erstellen. Beispiele hierfür sind die Unterkategorien “Trockener Riesling” und “Halbtrockener Riesling” für die Überkategorie “Riesling”. Rebsorten, Geschmack, Weineigenschaften und Qualität sollen eigene Datentypen anstatt einfacher Zeichenfolgen sein. Ziel davon ist, dass sich Nutzer für einen vorgefertigten, nominalen Eintrag in einem Dropdown-Menü entscheiden müssen und diese Auswahlmöglichkeiten immer noch im TYPO3-Backend pflegbar sind. Weinlagen sind im Brown-Field-Projekt bereits vorhanden, also sollen hierfür existierenden Daten eingebunden werden. Pro Wein sollen beliebig viele Weineigenschaften auswählbar sein, Wettbewerbskategorien, Geschmacksrichtung, etc, jeweils nur ein Element. Weitere Notizen zu diesem Gespräch sind im Anhang unter `<?? ??>` zu finden.

Da das Klassendiagramm gegeben lesbare Schrift nicht auf eine Textseite passt, befindet es sich vollseitig im Anhang unter `<F Klassen-diagramm>`. Die weitere Implementation der Datenobjekte ist unkompliziert und besteht hauptsächlich aus repetitivem Schreiben von SQL-Tabellen, Domain-Model-Klassen und TCAs. Um  $m, n$ -Beziehungen wie beispielsweise der Menge der für eine Probe zugelassenen Kategorien “allowedCategories” zwischen  $m$  “Jahresauswahlprobe”-Objekten und  $n$  “Category”-Objekten zu ermöglichen, werden MM-Tabellen (many-to-many) benötigt, diese Beziehungen in Form zweier Foreign Keys speichern. Die Repository-Klassen können “leer” gelassen werden, da zu diesem Zeitpunkt keine erweiterte Auswahllogik für Datenbankankfragen benötigt wird. Wichtig ist hierbei, dass eine Repository-Klasse existiert. Alle unverzichtbaren Schnittstellen werden über die Basisklasse “Repository” geerbt [TYPO3 Contributors, 2023e]. Mit Abschluss der Digitization können alle Datenstrukturen im TYPO3-Backend händisch angelegt, eingesehen, gelöscht und bearbeitet werden.

## 5.3. Digitalization

In der Phase *Digitalization* werden bestehende Geschäftsprozesse so verändert, dass mit digitalen Werkzeugen und Datenmodellen gearbeitet werden kann [Li et al., 2016]. Damit baut diese Phase auf der vorherigen Phase “Digitization” auf, um mit den dort implementierten Datenmodellen zu arbeiten. Im Folgenden werden die Umsetzungen der erforderlichen Geschäftsprozesse beschrieben.

### 5.3.1. Teilnehmerregistrierung

Ein essenzieller Teil des Jahresauswahlprobenwerkzeuges ist die Registrierung von Teilnehmern. Dieses Modul repräsentiert den ersten Berührungspunkt der Winzer mit dem System. Hiervon profitiert WM, weil registrierte Benutzer der Webseite eine Grundvoraussetzung für die Onlineregistrierung von Weinen zu JAPs sind. Dem Pflichtenheft ist zu entnehmen, dass es zwei Kategorien von Teilnehmerregistrierungen gibt:

1. Nutzer ist WM-Mitglied
2. Nutzer ist kein WM-Mitglied

Der primäre Unterschied zwischen Mitgliedern und Nicht-Mitgliedern ist, dass Mitglieder bereits einen Stammdatensatz hinterlegt haben. Dieser Stammdatensatz bildet die Angaben zum Weingut des zu digitalisierendem Anmeldeformulars ab. Nicht-Mitglieder sind dem System noch gänzlich unbekannt und müssen im Zuge der Registrierung ihres Nutzers ihre Stammdaten angeben, während sich Mitglieder lediglich einloggen müssen und eine Schaltfläche “Teilnehmer werden” betätigen. Der mit dem PO ausgearbeitete UX-Flow der Registrierung sieht vor, dass der Nutzer zunächst gefragt wird, ob er Mitglied sei oder nicht. Hierzu gibt es je einen Button. Ist der Nutzer ein Mitglied, wird er auf ein Loginform, mit der Option zur Registrierung weitergeleitet. Nach erfolgreichem Login, wird ein Teilnehmerobjekt erstellt. Wählt der Nutzer “Nein, ich bin kein Mitglied” aus, würde er auf ein Registrierungsformular weitergeleitet, auf um sich einen Nicht-Mitgliederaccount anzu-

legen. Im Zuge dieser Registrierung werden Stammdaten zum Weingut angefragt. Dieser Schritt übersetzt unter anderem den “Einreicher”-Teil des ursprünglichen Anmeldeformulars, anbei in  $\langle\langle A \text{ Anmeldeformular Jahresauswahlprobe} \rangle\rangle$ .

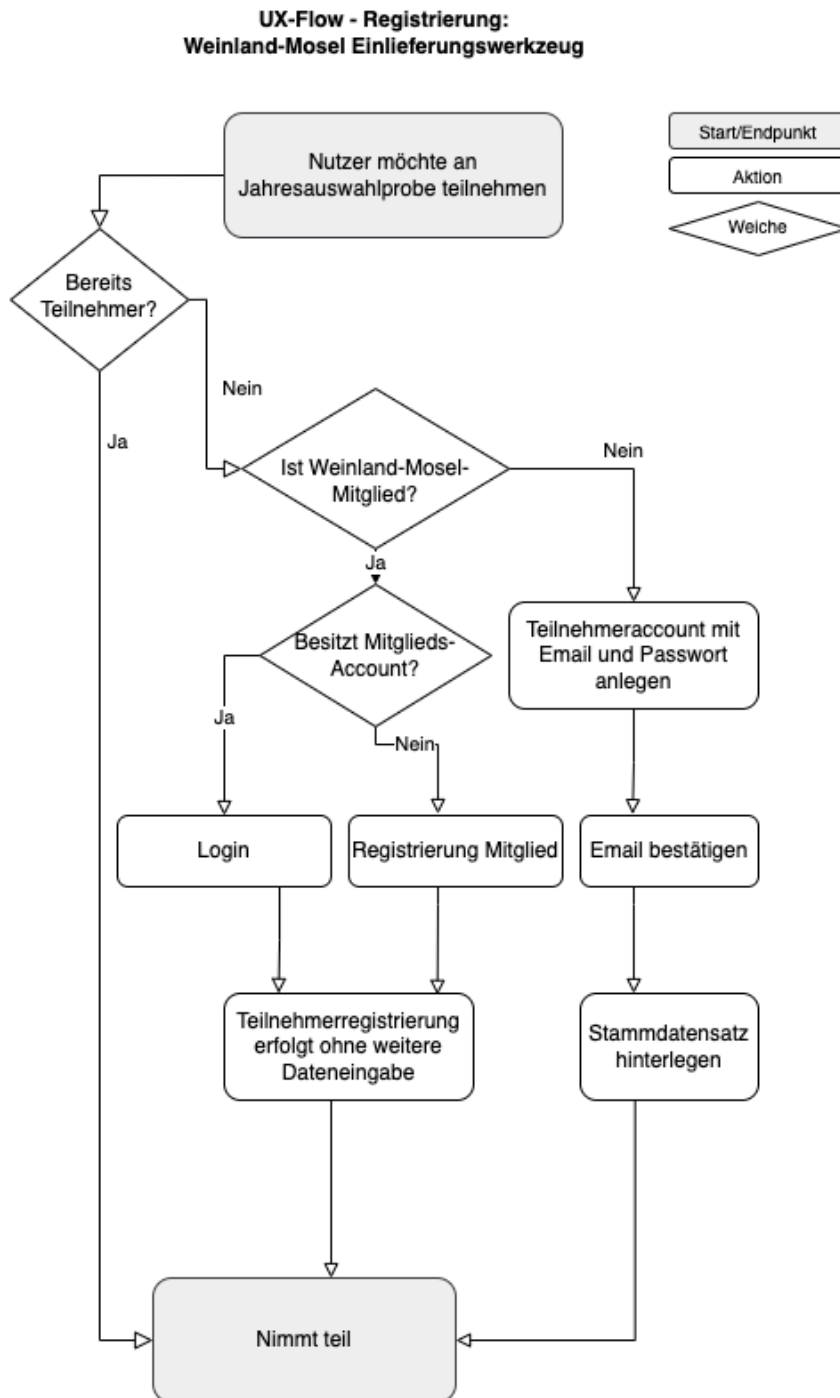


Abbildung 5.2.: UX-Flow: Registrierung

Quelle: Eigene Darstellung

Da das Brown-Field-Projekt bereits Accountlogins und -Registrierungen implementiert und nutzt, werden auf diese Lösungen zurückgegriffen, um einen einheitlichen Workflow beizubehalten. Accountregistrierungen werden über den “femanager” [in2code, 2023] realisiert, während Logins via TYPO3’s nativem Frontend-Nutzer-Login gelöst werden. Das ist explizit von femanager so angedacht: “*Note: Login and a I forgot my password function is part of the core and not part of femanager.*” [in2code, 2023].

Im Folgenden wird der Registrierungsprozess im Detail beschrieben: Grundsätzlich gibt es drei relevante Nutzerzustände vor der Registrierung:

1. Kein Mitglied
2. Mitglied, mit Konto
3. Mitglied, ohne Konto

Diese Prozesse sehen wie folgt aus:

### **Kein Mitglied**

Ist ein Nutzer kein Mitglied, so muss er zunächst einen Account erstellen. Anfangs wählt dieser Nutzer “Ich bin kein Mitglied” auf der Registrierungsseite aus. Daraufhin navigiert der Browser zu einem Registrierungsformular. Hierfür muss eine Email-Adresse und ein Passwort vergeben und den Nutzungsbestimmungen zugestimmt werden. Im Anschluss erhält der Nutzer eine Bestätigungsemail mit einem Aktivierungslink. Wird dieser Link geöffnet, wird der Account freigeschaltet und ein Login-Feld erscheint. Nach erfolgreichem Login wird der Nutzer mit einem Stammdatenformular konfrontiert. Dabei handelt es sich um Angaben zum teilnehmenden Weingut. Wird dieses Formular abgeschickt, ist die Teilnehmerregistrierung abgeschlossen.

### **Mitglied, mit Konto**

Ist ein Nutzer ein Weinland-Mosel-Mitglied und hat bereits ein Mitgliedskonto, muss dieser auf der Registrierungsseite “Ich bin ein Mitglied” auswählen. An dieser Stelle navigiert der Browser zu einem Login-Formular. Hier kann sich das Mitglied anmelden. Tut es dies erfolgreich, erstellt der Controller einen neuen Teilnehmer-Eintrag für den Frontend-Nutzer und fügt den Frontend-Nutzer der Nutzergruppe “Teilnehmer” hinzu. Damit ist die Teilnehmerregistrierung abgeschlossen.

### **Mitglied, ohne Konto**

Ist ein Nutzer ein Mitglied und hat noch kein Mitgliedskonto, muss dieser auf der Registrierungsseite “Ich bin ein Mitglied” auswählen. An dieser Stelle navigiert der Browser zu einem Login-Formular. Auf diesem Login-Formular existiert ein Button “Jetzt registrieren”, sowie ein Hinweistext dazu. Da der Nutzer noch keinen Account hat, muss dieser auf “Jetzt registrieren” klicken. Daraufhin navigiert der Browser zu einem Registrierungsformular, das eine Email-Adresse, ein Passwort und die Zustimmung zur Datenverarbeitung benötigt. Ist dieses Formular abgeschickt, erhält das Mitglied eine Email mit einem Bestätigungslink. Wird dieser Bestätigungslink angeklickt, wird das Mitgliedskonto freigegeben und es öffnet sich ein Login-Formular, beschrieben in  $\langle\langle\textit{Mitglied, mit Konto}\rangle\rangle$ .

### **Umsetzung**

Zunächst wurde ein simples Weichen-Content-Element erstellt. Dieses Content-Element hat die Parameter “question”, “answ-1-link”, “answ-1-text”, “answ-2-link” sowie “answ-2-text”. Der Zweck dieses Content-Elementes ist es, Nutzer basierend auf einer ausformulierten Frage auf eine von zwei Seiten weiterzuleiten. Anschließend wurden Registrierungen über Femanager-Plugin-Content-Elemente realisiert. Anpassungen der versendeten Emails erfolgen durch Überschreiben der Email-Templates von Femanager. Weiterleitungen zu bestimmten Seiten nachdem ein

Nutzer spezielle Events ausgelöst hat können über TypoScript konfiguriert werden [in2code, 2023]. Logins werden über das TYPO3-Native Loginformular gelöst. Im TYPO3-Loginformular kann man Weiterleitungen zu spezialisierten Seiten im Backend-UI festlegen [TYPO3 Contributors, 2023a]. Für alle funktionalen Belange wurde ein TYPO3-Plugin registriert. Dieses Plugin verfügt über einen ActionController, der Nutzeranfragen an PHP-Funktionen (“Actions”) bindet. In diesen Actions wird Fehlerbehandlung durchgeführt, Datenmodelle der Domäne erstellt und in der Datenbank persistiert sowie Daten für die Anzeige im Frontend aufbereitet [TYPO3 Contributors, 2023d]. Neue Datenobjekte werden in Repositories registriert [TYPO3 Contributors, 2023e]. Diese Repositories sind Aggregate des Controllers, werden jedoch nach dem “Inversion of Control”-Prinzip via Dependency Injection instanziiert und der ActionController-Klasse über Methode übergeben [TYPO3 Contributors, 2023b]. Als problematisch erweisen sich bidirektionale Verbindungen zwischen Datenmodellen, wenn die Foreign Keys über das SQL-Schlüsselwort “AUTO\_INCREMENT” in der Datenbank definiert werden. In diesem Fall wollen wir einen MasterRecord, der Betriebsinformationen speichert, bidirektional an ein Teilnehmerobjekt linken. Als ForeignKeys werden hierfür ihre jeweiligen Uids herangezogen, da diese Werte durch “AUTO\_INCREMENT” auf der Datenbankebene gehandhabt werden. Es gilt also, dass ein MasterRecord  $a$  die TeilnehmerUid von einem Teilnehmer  $b$  hält und dass  $b$  die MasterRecordUid von  $a$  hält. Die Problematik hierbei ist, dass diese Uids erst nach dem persistieren in der Datenbank bekannt sind, da diese Werte erst im Zuge der Persistierung erstellt werden. Das ist so, da das “AUTO\_INCREMENT”-Schlüsselwort lediglich zu SQL gehört und SQL nur von der Datenbank ausgeführt wird. Die Lösung hierfür ist es, beide Elemente zu erstellen und zu persistieren, danach ihre Uids gegenseitig bekannt machen um sie danach erneut zu persistieren.

### 5.3.2. Weinregistrierung

Ein Basismerkmal des Jahresauswahlprobenwerkzeuges ist die Möglichkeit, Weine zu Jahresauswahlproben anzumelden. Dieser Schritt übersetzt unter anderem die verbleibenden Formfelder des ursprünglichen Anmeldeformulares, anbei in  $\langle\langle A \text{ Anmeldeforumular Jahresauswahlprobe} \rangle\rangle$  in den digitalen Workflow. Für die Weinanmeldung spielt die Mitgliederschaft eines Teilnehmers keine Rolle. Es wird lediglich ein Frontend-Nutzer der Nutzergruppe "Teilnehmer" benötigt. Dieser Nutzer hat, wenn angemeldet, Zugriff auf eine Auflistung aller zeitlich freigegebenen Jahresauswahlproben. Soweit der Registrierungszeitraum dieser Jahresauswahlprobe den aktuellen Zeitpunkt miteinschließt, wird eine "Jetzt Wein anmelden"-Schaltfläche angeboten. Dadurch, dass Anmeldeformulare elektronisch und automatisiert verarbeitet werden, sinkt der Aufwand, der seitens WM für Anmeldungen gestemmt werden muss. Das ist so, da diese Formular nun nicht mehr von Mitarbeitern bearbeitet werden müssen. Davon profitiert WM, da diese Zeit nun anderweitig genutzt werden kann.

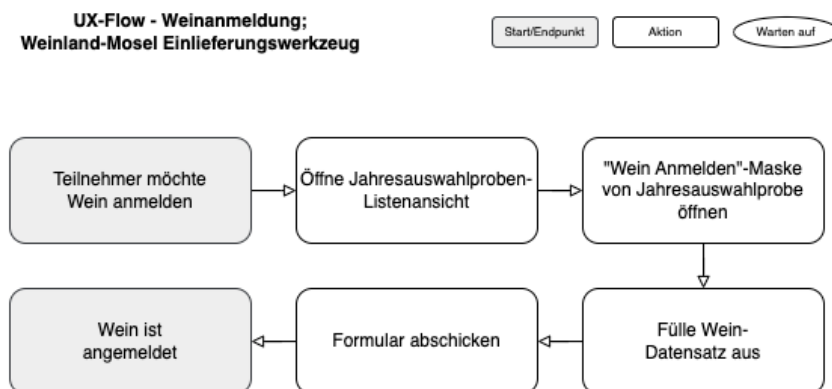


Abbildung 5.3.: UX-Flow: Weinanmeldung

Quelle: Eigene Darstellung

### IT-Sicherheit

Es ist wichtig zu erwähnen, dass solche Überprüfungen, wie das Aktivsein eines Registrierungszeitraumes einer JAP, grundsätzlich im Backend, d.h. serverseitig auf der betroffenen Webseite (in diesem Beispiel der Weinanmeldungsseite) durchgeführt werden. Das Verstecken der zugehörigen Schaltfläche im Frontend dient lediglich der User-Experience und stellt keine Sicherheitsvorkehrung dar. Das ist essenziell, da eine URL, auch wenn für sie keine Schaltfläche existiert, dennoch aufgerufen werden kann. Da Jahresauswahlprobennummern (Unique Identifier (UIDs)) fortlaufend sind, ist es trivial URLs für Weinanmeldungen beliebiger Jahresauswahlproben herzustellen. Insofern ist es von großer Wichtigkeit sicherzustellen, dass der Server solche Anfragen grundsätzlich selbst prüft und gegebenenfalls verneint.

### Das Formular

Aufgrund dessen, dass TYPO3 die Fluid Templating Engine verwendet [TYPO3 Association, 2023], werden Formulare und Formfelder mit den entsprechenden Fluid-Form-ViewHelfern aufgebaut. Diese View-Helper repräsentieren und erstellen gleichnamige HTML-Tags und fügen diesen spezielle Attribute zur Identifizierung in Submit-Aufrufen hinzu [TYPO3 Contributors, 2023f]. Grundsätzlich entstehen hierbei drei Kategorien von Werten, die es im Formular abzubilden gilt:

**Inputfelder** sind triviale Formfelder, die nicht durch andere Datensätze beschränkt werden. Beispiele für Inputfelder sind: Weinbeschreibung, Jahrgang und Alkoholgehalt. Inputfelder wurden mit simplen Input-Tags umgesetzt und erhielten nach Bedarf *required* und *pattern*-Attribute. Diese Attribute beschreiben jeweils, ob ein Formfeld ein Pflichtfeld ist und mit welcher Regular Expression der Formfeldinhalt abzugleichen ist [W3Schools, 2023a]. Die Formfeldwerte können unverändert in der Datenbank persistiert werden.

**SelectSingle** sind Formfelder, die dem Nutzer eine Auswahl aus  $n$  Elementen aus anderen Datenbanktabellen geben. Der Nutzer muss sich für genau ein Element entscheiden. Beispiele für SelectSingle-Formfelder sind: Weinlage, Qualitätsstufe, Rebsorte und Geschmacksangabe. SelectSingle-Formfelder werden durch Select-HTML-Tags abgebildet. Der TYPO3-Form-ViewHelper für “Select” akzeptiert eine Liste an Auswahlmöglichkeiten und erstellt selbstständig Option-HTML-Tags für diese. Die Formfeldwerte von SelectSingle-Formfeldern sind die UUIDs des jeweils ausgewählten Elementes [TYPO3 Contributors, 2023f].

Aufgrund dessen, dass das Weinlagen-Drop-Down-Menü über 170 Einträge führt, wurde eine Suchmöglichkeit implementiert. Diese ist lediglich ein Textfeld, das bei jeder Eingabe allen Option-Tags der Weinlage, deren Anzeigewert nicht der Suche entspricht, das Stilattribut “display: none;” auferlegt. Somit sind diese nicht mehr sichtbar.

Eine komplexe Ausnahme stellt das SelectSingle-Formfeld “Category” dar, da TYPO3-Kategorien Baumstrukturen sind [TYPO3 Contributors, 2023i]. Um die Eltern-Kind-Beziehung der Baumstruktur ersichtlich zu machen, werden die Option-HTML-Tags einzeln rekursiv gerendert. Zunächst werden sämtliche Kategorien, deren Parent Identifier (PID) 0 ist, dargestellt. Diese Elemente sind direkte Kinder des unsichtbaren Wurzelementes. Für jede dieser Kategorien  $a$  wird nun ein Fluid-Partial aufgerufen, das alle Kategorien  $b$  darstellt, für die gilt:  $b.pid = a.uid$ . Diese Darstellung erfolgt durch einen erneuten rekursiven Aufruf dieses Partials. In jeder Darstellung wird der Kategorienname, geprefixt mit  $n$  Leerzeichen, dargestellt, mit  $n = \text{Rekursionstiefe}$ . Somit entsteht ein Drop-Down-Menü, das alle Kategorien in einer eindimensionalen Liste darstellt. Diese Liste ist nach einer Preorder-Traversierung des Kategoriebaumes sortiert und desto tiefer ein Element im Baum ist, desto weiter ist es eingerückt. Damit sieht das Drop-Down-Menü aus wie eine Baumstruktur.

Diese Herangehensweise erzeugt schlüssigen und lesbaren Programmcode und lässt sich unkompliziert umsetzen. Das senkt Entwicklungskosten und erhöht den Profit des Endkunden, da hierdurch weniger Zeit aufgewandt wird. Rekursiv aufgerufene For-Schleifen, die sich selbst erneut für alle Elemente aufrufen, können jedoch zu einem Performanzproblem führen [Schwarzer, 2022]. Daher wird im Folgenden die Zeitkomplexität dieser Rekursionsfunktion betrachtet. Grundslegend, kann für diese Funktion kein Master-Theorem angewandt werden, da es sich hierbei nicht um einen Divide-and-Conquer-Algorithmus handelt [Schwarzer, 2022]. Das ist so, da das in der Rekursion weitergereichte Problem nicht kleiner wird, sondern gleich groß bleibt. Das verletzt die Bedingung  $b > 1$  des Master-Theorem, dargestellt als  $T(n) = a * T(\frac{n}{b}) + f(n)$ . Betrachten wir den Algorithmus, besteht er aus  $m, m \in \mathbb{N}$  verschachtelten For-Schleifen gleicher Länge. Somit ist die Zeitkomplexität  $O(n^m)$ . Normiert dargestellt beträgt die Zeitkomplexität  $O(n^2)$ . Das lässt sich experimentell bestätigen.

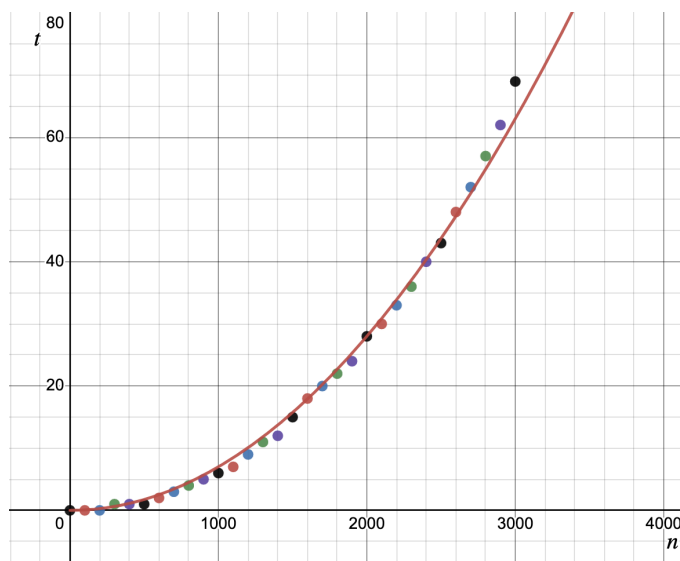


Abbildung 5.4.: Stichprobenartige Laufzeitanalyse des Kategorie-Renderers, gegenüber einer quadratischen Kurve

Quelle: Eigene Darstellung

Auf Optgroup-HTML-Tags wurde bewusst verzichtet. Grund dafür ist, dass Optgroup-Titel an sich nicht als Option auswählbar sind. Das stellt ein Problem dar, da beispielsweise die Kategorie “Riesling”, die die Unterkategorien “Trockener Riesling” und “Halbtrockener Riesling” beinhalten könnte, auch direkt auswählbar sein sollte. Zudem besitzen Kategorie-Elemente kein Attribut das auf die Präsenz von Unterkategorien hindeutet [TYPO3 Contributors, 2023i], womit eine Unterscheidung zwischen Baumblättern und -Zweigen nicht ohne weiteres möglich ist. Diese Entscheidung wäre jedoch benötigt, um zwischen einem Optgroup-Tag und einem Option-Tag abzuwägen.

**SelectMultiple** sind Formfelder, die dem Nutzer eine Auswahl aus  $n$  Elementen aus einer anderen Datenbanktabelle geben. Der Nutzer kann sich für eine beliebige Anzahl dieser, eingeschlossen null, entscheiden. Ein Beispiel für SelectMultiple-Formfelder ist: Weineigenschaften. TYPO3-Fluid implementiert hierfür keinen ViewHelper [TYPO3 Contributors, 2023f], also wurde eine eigene Lösung entworfen: Der Nutzer soll aus einer Menge  $A$  wählen. Für alle Elemente  $a \in A$  wird ein Checkbox-Feld erstellt. Dieses Element trägt den Anzeigewert “<a.title>” und den Wert “<formfeldname>-true”. Ist also eine dieser Checkboxes angehakt, hat sie den zuvor genannten Wert. Falls nicht, trägt sie keinen Wert. Weil alle angehakten Checkboxes dieses Formfeldes den selben Wert tragen, ist es PHP-Seitig trivial eine Liste aller angehakten Checkbox-Ids dieses Formfeldes aus der Liste aller Formfeldparameter zu extrahieren. Hierfür wird die eingebaute PHP-Funktion “array\_keys” verwendet. Diese Funktion gibt alle Keys eines Arrays in Form eines numerisch indizierten Arrays zurück. Der optionale Parameter “filter\_values” bestimmt, dass ausschließlich die Keys der Key-Value-Pairs, die einen bestimmten Wert tragen, extrahiert werden [The PHP Group, 2023]. D.h., der Funktionsaufruf filtert alle Keys und somit alle Formfeld-IDs des Formfeldparameter-Arrays heraus, die den Wert “<formfeldname>-true” haben. Das ist eine Liste aller Formfeld-IDs der Checkboxes des SelectMultiples, die angehakt wurden. Mit der eingebauten PHP-Funktion “array\_map”

wenden wir nun eine Operation auf alle Schlüssel der Liste an, die “strlen(‘formfeldname-’)” Zeichen, von links ausgehend, von der Formfeld-ID entfernt. Somit wird beispielsweise die Formfeld-ID “winekind-18” zu “18” transformiert. Übrig bleiben die UUIDs aller angehakten Elemente  $a$ , in Form einer Zeichenkette. Über die eingebaute PHP-Funktion “intval” ist es trivial diese zu Zahlen zu übersetzen, wodurch die tatsächlichen Objekte aus der Datenbank angefragt werden können.

### 5.3.3. PDF- und QR-Code-Generierung

Das dynamische Erstellen und Ausgeben des Versandbeilageblattes als PDF ist ein essenzieller Bestandteil des Jahresauswahlprobenwerkzeuges, da dieses PDF die Schnittstelle zwischen ankommenden Weinen und dem System darstellt. Daher ist ohne dieses PDF ist der Profit, der durch dieses Werkzeug generiert werden soll, stark eingeschränkt, da einkommende Weine händisch, von Mitarbeitern zugeordnet werden müssten. Wie im *⟨2 Stand der Forschung⟩* erläutert werden hierfür die Bibliotheken “chillerlan/php-qrcode” und “mpdf/mpdf” herangezogen und über Composer installiert.

#### QR-Code-Generierung

Der QR-Code beinhaltet lediglich die Wein-UUID anstatt einer vollständigen URL. Hintergrund dessen ist, dass die URL, die benötigt ist, um einen Wein einzuscannen, bis auf die Wein-UUID immer identisch ist. Somit wird Redundanz verhindert. Es ist Aufgabe der QR-Code-App, die den Code einscannet, aus der Wein-UUID eine vollständige URL herzuleiten. Um Ressourcen zu sparen und somit den Profit zu erhöhen, wird der QR-Code zu einem Base64-kodiertem Bild gerendert. Das ist der Standardrückgabewert des QR-Code-Generators und erfordert somit keine nähere Konfiguration. Ebenfalls lässt sich ein Base64-kodiertes Bild als Quellurl eines IMG-HTML-Tags angeben, womit das Bild eingebettet ist. Hier wird der Profit erhöht, indem Arbeitszeit gespart wird, die sonst in das anderweitige Einbetten eines Bildes in einem PDF mit “mpdf”

fließen müsste. Die PDF-Generierung funktioniert konkret, indem ein neues QRCode-Objekt der QRCode-Klasse erstellt wird. Diese Klasse nimmt ein QROptions-Objekt im Konstruktor, das in diesem Falle einige Stilattribute mit sich trägt. Das QRCode-Objekt bietet nun eine Methode “render” an, die, sofern nicht anders konfiguriert, den QR-Code als Base64-kodiertes Bild zurückgibt [Rochko, 2022b].

### PDF-Generierung

Firmenintern ist es Standard das Aussehen sowie die Inhalte der PDF-Dateien, die “mpdf” erzeugt, mit HTML zu definieren, das an “mpdf” gereicht wird. Um die Gestaltung und die Präsentation von Variablen in der HTML-Zeichenfolge technisch kontinuierlich mit dem restlichen Projekt zu halten und um somit eine gute Wartbarkeit zu gewährleisten, wird diese HTML-Zeichenfolge mit TYPO3-Fluid getemplated. Das heißt, dass eine HTML-Templatedatei bereitgestellt wird, diese mit TYPO3-Fluid befüllt wird und in PHP gerendert wird, um eine HTML-Zeichenkette als Ergebnis zu erhalten. Hierfür wird ein TYPO3-StandaloneView instanziiert, mit einem Pfad zur Template-Datei ausgestattet, Variablen angegeben, die in Fluid verfügbar sein sollen und anschließend über die “render”-Methode zu einem String gerendert [TYPO3 Contributors, 2023k]. Anschließend wird ein “mpdf”-Objekt erstellt und mit einer rudimentären Konfiguration in Form eines Arrays im Konstruktor konfiguriert. Diese Konfiguration definiert in diesem Falle Seitenabstände, Papierformat, Zeichenkodierung und Schriftarten. Abschließend wird dem “mpdf”-Objekt das zuvor generierte HTML übergeben und über die Methode “OutputBinaryData” zu Bytes konvertiert [Mpdf Contributors, 2023b]. Um dieses PDF über die Verbindung an den Nutzer zu übertragen, wird ein TYPO3-Response-Objekt erstellt. Über dieses Response-Objekt werden einige Header gesetzt und direkt übertragen. Dieser Header sind Content-Type und Content-Length. Abschließend werden als Response-Body die Bytes des generierten PDFs abgeschickt. Damit ist die Verbindung beendet und das PDF beim Nutzer angekommen.

## 6. Ergebnisse

## 7. Ausblick

# Literaturverzeichnis

- [Abraham, 2005] Abraham, M. (2005). *Sustainability Science and Engineering: Defining Principles*. ISSN. Elsevier Science.
- [Bettencourt, 2018] Bettencourt, R. G. (2018). barcode.php - Generate barcodes from a single PHP file. MIT license. <https://github.com/kreativekorp/barcode>. Zugriff: Januar 2023.
- [Day, 2020] Day, R. (2020). QR code/2d barcode generator. <https://github.com/soldair/node-qrcode>. Zugriff: Januar 2023.
- [Die SOPHISTen, 2016] Die SOPHISTen (2016). Die kleine re-fibel. 3.
- [Dougherty and Dunne, 2012] Dougherty, D. and Dunne, D. D. (2012). Digital science and knowledge boundaries in complex innovation. *Organization Science*, 23(5):1467–1484.
- [Ebert, 2019] Ebert, C. (2019). Vorwort zu systematisches re.
- [Github, 2023] Github (2023). Saving repositories with stars. <https://docs.github.com/en/get-started/exploring-projects-on-github/saving-repositories-with-stars>. Zugriff: Januar 2023.
- [Heinemann, 2021] Heinemann, E. (2021). *Vorlesung: Requirements Engineering*. Hochschule Worms.
- [in2code, 2023] in2code (2023). femanager. <https://docs.typo3.org/p/in2code/femanager/main/en-us/Introduction/Index.html>. Zugriff: März 2023.
- [Jung, 2020a] Jung, L. (2020a). jquery-qrcode readme.md. <https://github.com/lrsjng/jquery-qrcode/blob/master/README.md>. Zugriff: Januar 2023.
- [Jung, 2020b] Jung, L. (2020b). kjua dynamically generated QR codes. <https://larsjung.de/kjua/>. Zugriff: Januar 2023.

- [König, 2022] König, W. (2022). *Vorlesung: User Experience*. Hochschule Worms.
- [Li et al., 2016] Li, F., Nucciarelli, A., Roden, S., and Graham, G. (2016). How smart cities transform operations models: a new research agenda for operations management in the digital economy. *Production Planning & Control*, 27(6):514–528.
- [Loebbecke and Picot, 2015] Loebbecke, C. and Picot, A. (2015). Reflections on societal and business model transformation arising from digitization and big data analytics: A research agenda. *The Journal of Strategic Information Systems*, 24(3):149–157.
- [Mpdf Contributors, 2023a] Mpdf Contributors (2023a). mPDF. <https://mpdf.github.io/>. Zugriff: März 2023.
- [Mpdf Contributors, 2023b] Mpdf Contributors (2023b). Overview mPDF Functions by Category. <https://mpdf.github.io/reference/mpdf-functions/overview.html>. Zugriff: März 2023.
- [npm, inc, 2020] npm, inc (2020). QR code/2d barcode generator. <https://www.npmjs.com/package/qrcode>. Zugriff: Januar 2023.
- [OpenJS Foundation, 2023] OpenJS Foundation (2023). Command-line API. <https://nodejs.org/api/cli.html#command-line-api>. Zugriff: Januar 2023.
- [Opensource.org, a] Opensource.org. The 2-Clause BSD License. <https://opensource.org/licenses/BSD-2-Clause>. Zugriff: Januar 2023.
- [Opensource.org, b] Opensource.org. The MIT License. <https://opensource.org/licenses/MIT>. Zugriff: Januar 2023.
- [Packagist, 2023] Packagist (2023). BaconQrCode is a QR code generator for PHP. <https://packagist.org/packages/bacon/bacon-qr-code>. Zugriff: Januar 2023.

- [Pagani and Pardo, 2017] Pagani, M. and Pardo, C. (2017). The impact of digital technology on relationships in a business network. *Industrial Marketing Management*, 67:185–192.
- [Parviainen et al., 2022] Parviainen, P., Tihinen, M., Kääriäinen, J., and Teppola, S. (2022). Tackling the digitalization challenge: how to benefit from digitalization in practice. *International Journal of Information Systems and Project Management*, 5(1):63–77.
- [Pohl and Rupp, 2015] Pohl, K. and Rupp, C. (2015). *Basiswissen Requirements Engineering*. dpunkt.verlag, Heidelberg, 4., überarbeitete auflage edition. Description based upon print version of record.
- [Rochko, 2022a] Rochko, E. (2022a). A container class for immutable settings objects. Not a DI container. <https://github.com/chillerlan/php-settings-container>. Zugriff: Januar 2023.
- [Rochko, 2022b] Rochko, E. (2022b). A QR code generator and reader with a user friendly API. PHP 7.4+. <https://github.com/chillerlan/php-qrcode>. Zugriff: Januar 2023.
- [Rochko, 2022c] Rochko, E. (2022c). composer.json. <https://github.com/chillerlan/php-qrcode/blob/main/composer.json>. Zugriff: Januar 2023.
- [Rochko, 2023] Rochko, E. (2023). SVG QR Code Generator. <https://smiley.codes/qrcode>. Zugriff: Januar 2023.
- [Scholzen, 2018] Scholzen, B. (2018). barcode.php - Generate barcodes from a single PHP file. MIT license. <https://github.com/bacon/baconqrcode>. Zugriff: Januar 2023.
- [Schwarzer, 2021] Schwarzer, V. (2021). *Vorlesung: Software Architecture*. Hochschule Worms.
- [Schwarzer, 2022] Schwarzer, V. (2022). *Vorlesung: Algorithmen und Datenstrukturen*. Hochschule Worms.

- [Tabrizi et al., 2019] Tabrizi, B., Lam, E., Girard, K., and Irvin, V. (2019). Digital transformation is not about technology. *Harvard business review*, 13(March):1–6.
- [TeaCapps, 2023] TeaCapps (2023). QRBot | QRCode und Barcode Scanner. <https://qrbot.net>. Zugriff: März 2023.
- [The PHP Group, 2023] The PHP Group (2023). PHP: array-keys - Manual. <https://www.php.net/manual/en/function.array-keys.php>. Zugriff: März 2023.
- [TYPO3 Association, 2023] TYPO3 Association (2023). Fluid - A fast and secure templating engine. <https://typo3.org/fluid>. Zugriff: März 2023.
- [TYPO3 Contributors, 2023a] TYPO3 Contributors (2023a). cms-felogin. <https://docs.typo3.org/c/typo3/cms-felogin/main/en-us/Configuration/Index.html>. Zugriff: März 2023.
- [TYPO3 Contributors, 2023b] TYPO3 Contributors (2023b). Dependency Injection. <https://docs.typo3.org/m/typo3/reference-coreapi/11.5/en-us/ApiOverview/DependencyInjection/Index.html>. Zugriff: März 2023.
- [TYPO3 Contributors, 2023c] TYPO3 Contributors (2023c). Extbase reference. <https://docs.typo3.org/m/typo3/reference-coreapi/11.5/en-us/ExtensionArchitecture/Extbase/Reference/Index.html>. Zugriff: Februar 2023.
- [TYPO3 Contributors, 2023d] TYPO3 Contributors (2023d). Extension development / Extbase. <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/ExtensionArchitecture/Extbase/Index.html>. Zugriff: März 2023.
- [TYPO3 Contributors, 2023e] TYPO3 Contributors (2023e). Extension development / Tutorials / Tea in a nutshell / Repository. <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/>

- `ExtensionArchitecture/Tutorials/Tea/Repository.html`. Zugriff: Februar 2023.
- [TYPO3 Contributors, 2023f] TYPO3 Contributors (2023f). form - Fluid ViewHelper Reference 11.5 Documentation. <https://docs.typo3.org/other/typo3/view-helper-reference/11.5/en-us/typo3/fluid/latest/Form/Index.html>. Zugriff: März 2023.
- [TYPO3 Contributors, 2023g] TYPO3 Contributors (2023g). Getting started. <https://docs.typo3.org/m/typo3/tutorial-getting-started/11.5/en-us/Index.html>. Zugriff: Februar 2023.
- [TYPO3 Contributors, 2023h] TYPO3 Contributors (2023h). Managing Extensions. <https://docs.typo3.org/m/typo3/tutorial-getting-started/main/en-us/Extensions/Management.html>. Zugriff: Februar 2023.
- [TYPO3 Contributors, 2023i] TYPO3 Contributors (2023i). System categories. <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/ApiOverview/Categories/Index.html>. Zugriff: März 2023.
- [TYPO3 Contributors, 2023j] TYPO3 Contributors (2023j). TYPO3 Extension Development File Structure. <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/ExtensionArchitecture/FileStructure/Index.html>. Zugriff: Februar 2023.
- [TYPO3 Contributors, 2023k] TYPO3 Contributors (2023k). TYPO3 CMS Fluid View StandaloneView Class Reference. [https://api.typo3.org/10.4/class\\_t\\_y\\_p\\_o3\\_1\\_1\\_c\\_m\\_s\\_1\\_1\\_fluid\\_1\\_1\\_view\\_1\\_1\\_standalone\\_view.html](https://api.typo3.org/10.4/class_t_y_p_o3_1_1_c_m_s_1_1_fluid_1_1_view_1_1_standalone_view.html). Zugriff: März 2023.

- [TYPO3 Contributors, 2023l] TYPO3 Contributors (2023l). What are Content Elements? <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/ApiOverview/ContentElements/Introduction.html>. Zugriff: Februar 2023.
- [Udjaja, 2018] Udjaja, Y. (2018). Ekspanpixel bladsy stranica: Performance efficiency improvement of making front-end website using computer aided software engineering tool. *Procedia Computer Science*, 135:292–301. The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life.
- [Verhoef et al., 2021] Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Qi Dong, J., Fabian, N., and Haenlein, M. (2021). Digital transformation: A multidisciplinary reflection and research agenda. *Journal of Business Research*, 122:889–901.
- [W3Schools, 2023a] W3Schools (2023a). HTML <input> Tag. [https://www.w3schools.com/tags/tag\\_input.asp](https://www.w3schools.com/tags/tag_input.asp). Zugriff: März 2023.
- [W3Schools, 2023b] W3Schools (2023b). Sass Introduction. [https://www.w3schools.com/sass/sass\\_intro.php](https://www.w3schools.com/sass/sass_intro.php). Zugriff: Januar 2023.
- [Zimmermann, 2017] Zimmermann, J. (2017). Webpack - A Detailed Introduction. <https://www.smashingmagazine.com/2017/02/a-detailed-introduction-to-webpack/>. Zugriff: Januar 2023.

# Anhang

# **A. Anmeldeforumular Jahresauswahlprobe**

## Kategorien

| Nr. | Bezeichnung  | Unterkategorien/Kriterien   |
|-----|--|---|
| 1.  | 2021 PiWi-Weißweine trocken  | Rebsortenreine Weine sowie Rebsorten-Cuvées   |
| 2.  | 2021 Elbling Qualitätswein trocken   |   |
| 3.  | 2021 Rivaner/Müller-Thurgau Qualitätswein trocken  |   |
| 4.  | 2021 Weiße Burgundersorten Qualitätswein trocken   | Zugelassen sind:<br>Weißer Burgunder, Grauer Burgunder,<br>Chardonnay und Auxerrois oder Cuvées daraus                    |
| 5.  | 2021 Riesling Qualitätswein trocken - Gutswein   |   |
| 6.  | 2021 Riesling Qualitätswein trocken - Ortswein   | Auch trockene Kabinettweine   |
| 7.  | 2021 Riesling Qualitätswein feinherb - Gutswein  |   |
| 8.  | 2021 Riesling Qualitätswein feinherb - Ortswein  |   |
| 9.  | 2021 Riesling Kabinett feinherb (bis max. 11,5 Vol.-Prozent)   | Bis 11,5 Volumen-Prozent vorhandener Alkoholgehalt  |
| 10. | 2021 Riesling Kabinett lieblich/fruchtsüß (bis max. 70gr/Ltr. Restzucker)  | Bis maximal 70 Gramm Restzuckergehalt je Liter  |
| 11. | 2021 Spätburgunder Rosé Qualitätswein trocken  |   |
| 12. | Sekt/Crémant (Sekt ausschließlich b.A. bzw. Crémant Mosel und aus traditioneller Flaschengärung, mindestens neun Monate Hefelager) | 12.1 Elbling zero dosage bis brut<br>12.2. Riesling zero dosage bis trocken<br>12.3. Burgundersorten zero dosage bis brut |

## Anmeldeformular

Zu jeder Probe nachstehende Angaben ausfüllen und den 2 Probeflaschen (auch bei Schraubverschluss) beifügen!

|                      |  |
|----------------------|--|
| <b>Kategorie Nr.</b> |  |
|----------------------|--|

**Nur vollständig ausgefüllt gültig**

|  |  |                      |  |  |  |  |  |
|--|--|----------------------|--|--|--|--|--|
| Jahrgang:  |  | Rebsorte:            |  |  |  |  |  |
| Weinbezeichnung (z.B. Lagename und/oder andere Bezeichnungen wie Steillage, Alte Reben, Schiefer etc.) |  |                      |  |  |  |  |  |
| Qualitätsstufe:  |  |                      |  |  |  |  |  |
| Geschmacksangabe:  |  |                      |  |  |  |  |  |
| Biowein<br>(Zutreffendes bitte ankreuzen)  |  |                      |  | veganer Wein<br>(Zutreffendes bitte ankreuzen) |  |  |  |
| Spontangärung<br>(Zutreffendes bitte ankreuzen)  |  |                      |  | Barrique<br>(Zutreffendes bitte ankreuzen)     |  |  |  |
| Sonstige Angaben zu Besonderheiten:  |  |                      |  |  |  |  |  |
| AP-Nr. (sofern bereits erteilt):   |  |                      |  |  |  |  |  |
| Säure<br>(g/l):  |  | Restzucker<br>(g/l): |  | Alkoholgehalt<br>(% Vol.):                     |  | Endverbraucherpreis<br>(inkl. MwSt./Sektsteuer): |  |

### ***EINREICHER:***

|  |          |
|--|----------|
| Betriebsname:  |          |
| Ansprechpartner (Vorname/Name):  |          |
| Straße:  | PLZ/Ort: |
| Tel:   | E-Mail:  |
| Internet:  |          |
| Ich erkläre mich hiermit mit den Bedingungen der Ausschreibung einverstanden. Den Datenschutzhinweis habe ich zur Kenntnis genommen. |          |
| Datum und Unterschrift:  |          |

## B. Interview-Fragebogen

Wie stellen Sie sich den Prozess des Einscannens der QR-Codes beim Entgegennehmen der Flaschen vor? Beschreiben Sie den Ablauf.

Mit was soll dieser Code gescannt werden? Soll der Scanner in der Applikation eingebaut sein, oder soll das System auch mit Drittanbieter-Apps funktionieren?

Von welchem Endgerät wird gescannt?

Welche Fallbacks soll es geben, sollte ein Code nicht scanbar sein? Z.B.: Der Code-Inhalt in Text unter dem Code, der auch von Hand eintippbar sei.

Gegen welche Missbrauchsszenarien sollte der QR-Code geschützt sein? Sollte ggf. ein Passwort nach dem Einscannen verlangt werden? -> Diskurs über versch. Authentifizierungsmethoden und Abwägung über Aufwand der dadurch entsteht.

Was soll der Kunde beim Versand selbst machen?

Wenn es ein PDF zum Ausdruck zum Aufbringen des Empfängers gibt, sollte man auch ein PDF zum Ausdruck des Absenders generieren, falls die relevanten Daten vorliegen?

In welcher Form sollten Mitarbeiter die ausstehenden und eingegangenen Weine sehen? Reicht eine einfache Liste, oder sind Export- und Filtermöglichkeiten erwünscht? Wenn ja: Welche Filter (auch Sortierungen)? Welche Exportformate (Excel kann auch csv öffnen)?

Wenn nun ein Wein als “eingegangen” vermerkt ist, sollte ein Mitarbeiter das rückgängig machen können? Sollte ein Mitarbeiter Weine löschen können? Wenn eines der beiden ja: Einzeln, oder als Bulk-Action? (Bulk-Actions sind teuer/aufwändig umzusetzen)

Welche Informationen soll der Kunde über seine Sendunge(n) sehen?

Soll auch ein Kunde in der Lage sein, seine eigene Weinsendung(en) aus dem System zu löschen oder zu verändern? (Eventuell vertippt man sich) Wenn ja, bis wann? Bis zur Eingangsbestätigung? Ist hierbei eine Bulk-Action wichtig?

Die ELN ist eine inkrementell erhöhte Ganzzahl. Ist der tatsächliche Wert dieser wichtig, oder ist es lediglich wichtig, dass sie eindeutig ist? Der einfachste/günstigste Weg wäre es, sie in der Datenbank als "auto\_increment" zu deklarieren. Dann hätte man niemals, auch über x Auswahlproben hinweggehend, die selbe ELN zwei mal. Das verkürzt und vereinfacht die Entwicklung, den entstehenden Code, und die Nutzererfahrung auf Seiten von Weinland Mosel.

## C. Online-Fragebogen Weingüter

# Jahresauswahlprobe - Registrierungstool

1. Sind Sie mit der aktuellen Arbeitsweise der Anmeldung ihrer Weine zur Jahresauswahlprobe zufrieden?

*Markieren Sie nur ein Oval.*

- ☐ Ja  
☐ Nein

2. Wünschen Sie sich eine Digitalisierung dieser Arbeitsweise, die ihre Formulare verkürzt, an den Rechner verschiebt und es ihnen ermöglicht zu verfolgen, ob ihre Sendungen eingegangen sind?

*Markieren Sie nur ein Oval.*

- ☐ Ja  
☐ Nein

3. Wie viele Weine stellen Sie pro Jahresauswahlprobe ungefähr auf?

---

4. Mit welchem Gerät greifen Sie primär auf das Internet zu?

*Wählen Sie alle zutreffenden Antworten aus.*

- ☐ iPhone
- ☐ Android Phone
- ☐ iPad
- ☐ Android Tablet
- ☐ Laptop/Macbook
- ☐ Desktop-Rechner

5. Haben Sie Zugriff auf einen Drucker?

*Markieren Sie nur ein Oval.*

- ☐ Ja
- ☐ Nein

6. Haben Sie sonstige Wünsche und Anregungen bezüglich der Anmeldung zu Jahresauswahlproben?

---

---

---

---

---

---

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt.

Google

## D. Interview-Protokoll

|             |  |
|-------------|--|
| Interviewer | Leon Etienne (Entwickler)  |
| Befragter   | Jochen Stange (Product Owner)  |
| Zeit        | 27.01.2023, 10:00  |
| Ort         | Zoom   |
| Notizen     | Dieses Protokoll ist nicht wortlautkorrekt, sondern zeigt lediglich zeitgleich verfasste Notizen |

**Wie stellen Sie sich den Prozess des Einscannens der QR-Codes beim Entgegennehmen der Flaschen vor? Beschreiben Sie den Ablauf.** - Paket kommt an - Wird aufgemacht - Ziehen qr code raus - Haben entweder Handscanner oder handy und scannen den code ein - system gibt an, welche ELN aufgeklebt werden muss - wein wird eingelagert

Die ELN muss im Verein mit der JAP unique sein. Die ELN ist dann im Format ELN-Jahresauswahlprobennummer z.b. 19-1 gesetzt. Das steht für ELN 19, JAP 1. Auf Datenkbankebene unique. Es wird bei 1 anfangen zu zählen.

**Mit was soll dieser Code gescannt werden? Soll der Scanner in der Applikation eingebaut sein, oder soll das System auch mit Drittanbieter-Apps funktionieren?** Smartphone ist am sinnvollsten, da feedback von Webseite Backup-Funktion die ELN händisch einzugeben. Das ginge dann auch mit einem Handscanner.

**Welche Fallbacks soll es geben, sollte ein Code nicht scanbar sein? Z.B.: Der Code- Inhalt in Text unter dem Code, der auch von Hand eintippbar sei.** ELN händisch im typo3 backend suchen und bearbeiten.

Die ELN muss also auch unter dem qr code stehen, aber das ist für uns out-of-scope. Das ist nicht der QR-Code, den wir generieren, sondern der, der beim Einlagern auf die Flasche geklebt wird. Diese lässt Weinland Mosel im Voraus drucken.

**Gegen welche Missbrauchsszenarien sollte der QR-Code geschützt sein? Sollte ggf. ein Passwort nach dem Einscannen verlangt werden? ->Diskurs über versch. Authentifizierungsmethoden und Abwägung über Aufwand der dadurch entsteht.** Verwenden wir einen API-Key. Übermittlung via POST. Eingetragen in qr-code app. Es gibt einen api key. Der soll im backend änderbar sein. Gehashed mit typo3 argoni2i. Gespeichert in der Datenbank.

**Gibt es im Lager Wlan?** Out of scope

**Wie wollen Sie Weinanmeldungen zwischen Jahresauswahlproben im Frontend trennen? Wichtig: Aktuelle Proben nach Anmeldeschluss sollten immer noch sichtbar sein und alte Proben ggf. gar nicht mehr angezeigt werden.** Es gibt eine Liste mit JAPs. Da werden verschiedene aufgelistet. Backenduser können die Anmeldung (Weine an JAPs anmelden) an- und abschalten. Vergangene Proben werden nicht angezeigt. Die Probe gilt als vergangen, wenn die Proben probiert wurden. Die Anmeldeöglichkeit und "Probe ist vorbei"-Feststellung im Anmeldetool ist ein Clone vom „active until“ im „access-panel. Mit Dateselector.

**Was soll der Kunde beim Versand selbst machen?** Wenn weinland-moselmitglied: Ggf account erstellen mit Mitgliedsnummer <schon implementiert> <muss händisch von WM freigeschalten werden> In account einloggen

Wenn Nichtmitglied: Ggf account erstellen ohne Mitgliedsnummer <muss händisch freigeschalten werden>

Eine JAP auswählen Einen Wein anmelden, Webform zu Weindaten ausfüllen. PDF downloaden, ausdrucken im Paket dem Wein beilegen. Das PDF muss Daten zum Wein beinhalten, und den qr-code mit der ELN. (zur Zuordnung) Nummernschwund, falls Weine gelöscht werden, ist kein Problem. 196-1 darf auf 194-1 folgen. Adressierung wird selbst gemacht. Frankierung auch.

**In welcher Form sollten Mitarbeiter die ausstehenden und eingegangenen Weine sehen? Reicht eine einfache Liste, oder sind Export- und**

**Filtermöglichkeiten erwünscht? Wenn ja: Welche Filter (auch Sortierungen)? Welche Exportformate?** - CSV soll ausgabbar sein - Filter nicht notwendig, aber probenspezifisch

**Wenn nun ein Wein als “eingegangen” vermerkt ist, sollte ein Mitarbeiter das rückgängig machen können? Sollte ein Mitarbeiter Weine löschen können? Wenn eines der beiden ja: Einzeln, oder als Bulk-Action? (Bulk-Actions sind teuer/aufwändig umzusetzen)** - Was TYPO3 by-default kann, nichts eigenes bauen

**Welche Informationen soll der Kunde über seine Sendunge(n) sehen?**  
- ob sie eingegangen ist, mit emailbenachrichtigung

**Soll auch ein Kunde in der Lage sein, seine eigene Weinsendung(en) aus dem System zu löschen oder zu verändern? (Eventuell vertippt man sich) Wenn ja, bis wann? Bis zur Eingangsbestätigung?** Nach Einreichung (nicht Eingang bei WM) nicht mehr veränderbar und auch nicht löschar.

**Die ELN ist eine inkrementell erhöhte Ganzzahl. Ist der tatsächliche Wert dieser wichtig, oder ist es lediglich wichtig, dass sie eindeutig ist? Der einfachste/günstigste Weg wäre es, sie in der Datenbank als “auto\_increment” zu deklarieren. Dann hätte man niemals, auch über x Auswahlproben hinweggehend, die selbe ELN zwei mal. Das verkürzt und vereinfacht die Entwicklung, den entstehenden Code und die Nutzererfahrung auf Seiten von Weinland Mosel.** - siehe oben

**Weiteres** Es soll einen Single-View für Weine geben, der anhand einer ELN aufrufbar ist. Hier werden bekannte Details gezeigt.

**Zum Formular, das Winzer ausfüllen:** Das Formular besteht zwei Schritten: Schritt 1: Daten ausfüllen Schritt 2: Zusammenfassung anzeigen entweder zurück, weiter bearbeiten, oder submit “wein verbindlich anmelden”

## E. Pflichtenheft

| Art der Anforderung | Beschreibung  |
|---------------------|---|
| Constraint          | Einbau in brown-field TYPO3   |
| Constraint          | Kunden-UI im Frontend   |
| FA                  | Mitglieder bei WM können sich einen Nutzeraccount erstellen   |
| FA                  | Nicht-Mitglieder bei WM können sich einen Nutzeraccount erstellen   |
| FA                  | Angemeldete Nutzer sehen eine Übersicht aller aktiven JAPs  |
| FA                  | Angemeldete Nutzer können mehrere Weine bei aktiven JAPs anmelden   |
| FA                  | Eine Weinanmeldung fragt Weindaten mit einem Webform ab   |
| FA                  | Vor Abschluss des Formulars wird eine Übersicht der Daten gezeigt   |
| FA                  | Das Formular lässt sich mit einem “Jetzt verbindlich anmelden”-Button abschicken  |
| FA                  | Angemeldete Nutzer sehen angemeldete Weine für JAPs inklusive Versandstatus   |
| FA                  | Für angemeldete Weine lassen sich PDF-Dateien herunterladen, die die ELN als Barcode und eine Beschreibung des Weines enthalten |
| NFA                 | Das Formular mit Weindaten und dem QR-Code lässt sich auch direkt nach Anmeldung eines Weines herunterladen                     |
| FA                  | Mitarbeiter von WM können eingetragene Weine als TYPO3-Objekte manipulieren   |
| FA                  | Mitarbeiter von WM können JAPs im Backend zeitgesteuert für Anmeldungen freigeben   |
| FA                  | Mitarbeiter von WM können JAPs im Backend zeitgesteuert für die Einsicht freigeben  |
| FA                  | Mitarbeiter von WM können einen gesicherten API-Link anhand einer ELN aufrufen um einen Wein als “eingegangen” zu markieren     |
| FA                  | Wird ein Wein via dem QR-Code als “eingegangen” markiert, bekommt der Winzer eine Email   |
| NFA                 | Der API-Link ist mittels eines API-Keys gesichert   |

| Art der Anforderung | Beschreibung   |
|---------------------|--|
| NFA                 | Der API-Key wird in POST-Daten übergeben   |
| NFA                 | Die ELN wird url-kodiert übergeben   |
| FA                  | Mitarbeiter von WM können den API-Key im Backend verändern                       |
| NFA                 | Nach Aufruf eines API-Links wird gezeigt, welcher Wein gescannt wurde            |
| FA                  | Es gibt einen Single-View für Weine, der über eine ELN in der URL erreichbar ist |
| NFA                 | Der Wein-Single-View zeigt bekannte Weindaten                                    |
| FA                  | Mitarbeiter von WM können Weineintragungen je nach JAP als CSV exportieren       |

## F. Klassendiagramm

