



Abschlussprüfung Sommer 2019

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

## **Tierbestandsbuch**

**Erweiterung der Webapplikation zum Führen eines Tierbestandsbuchs für den Argenhof –  
Lebenswürde für Tiere e.V.**

Abgabedatum: Ravensburg, den 30.05.2019

### **Prüfungsbewerber:**

Julia Haggenmiller

Mauerstraße 10

88212 Ravensburg

### **Ausbildungsbetrieb:**

Liebenau Berufsbildungswerk gemeinnützige GmbH

Pfannenstiel 16

88214 Ravensburg



## Inhaltsverzeichnis

---

## Inhaltsverzeichnis

|  |     |
|--|-----|
| Inhaltsverzeichnis .....                 | I   |
| Abbildungsverzeichnis .....              | IV  |
| Tabellenverzeichnis .....                | V   |
| Verzeichnis der Listings.....            | VI  |
| Abkürzungsverzeichnis .....              | VII |
| 1 Einleitung .....                       | 1   |
| 1.1 Projektumfeld .....                  | 1   |
| 1.2 Projektziele .....                   | 1   |
| 1.3 Projektbegründung .....              | 1   |
| 1.4 Projektschnittstellen .....          | 2   |
| 1.5 Projektabgrenzung.....               | 2   |
| 2 Projektplanung.....                    | 2   |
| 2.1 Zeitplanung.....                     | 2   |
| 2.2 Terminplanung.....                   | 3   |
| 2.3 Personalplanung .....                | 3   |
| 2.4 Sachmittelplanung .....              | 3   |
| 2.5 Entwicklungsprozess.....             | 3   |
| 3 Analysephase .....                     | 4   |
| 3.1 Ist-Analyse .....                    | 4   |
| 3.2 Soll-Konzept .....                   | 5   |
| 3.3 Wirtschaftlichkeitsanalyse .....     | 6   |
| 3.3.1 Make-or-Buy-Entscheidung .....     | 6   |
| 3.3.2 Projektkosten.....                 | 6   |
| 3.3.3 Amortisationsdauer .....           | 6   |
| 3.3.4 Nicht-monetäre Vorteile .....      | 7   |
| 3.4 Prozessanalyse.....                  | 7   |
| 3.5 Lastenheft .....                     | 7   |
| 4 Entwurfsphase .....                    | 7   |
| 4.1 Zielplattform .....                  | 7   |
| 4.2 Architekturdesign .....              | 8   |
| 4.3 Entwurf der Benutzeroberfläche ..... | 8   |
| 4.4 Datenmodell .....                    | 9   |

## Inhaltsverzeichnis

---

|     |  |     |
|-----|--|-----|
| 4.5 | Geschäftslogik.....  | 9   |
| 4.6 | Maßnahmen zur Qualitätssicherung.....  | 10  |
| 4.7 | Pflichtenheft/Datenverarbeitungskonzept .....  | 10  |
| 5   | Implementierungsphase .....  | 10  |
| 5.1 | Erstellen der Datenbanktabellen .....  | 10  |
| 5.2 | Implementierung und Test der Pflegedaten .....   | 10  |
| 5.3 | Implementierung und Test der Tierliste .....   | 11  |
| 5.4 | Implementierung der Druckfunktion .....  | 11  |
| 5.5 | Implementierung und Test der automatischen Backups .....                                     | 12  |
| 6   | Einführung .....   | 12  |
| 6.1 | Bereitstellung der Webapplikation auf dem Cloud-Server .....                                 | 12  |
| 6.2 | Schulung des Kunden und Abnahme .....  | 13  |
| 7   | Dokumentation .....  | 13  |
| 8   | Anpassungen .....  | 13  |
| 8.1 | Multiple-Drop-Down anstelle von Textfeldern .....  | 13  |
| 8.2 | Manuelle Tests anstelle von automatisierten Integrations-Tests .....                         | 14  |
| 8.3 | Implementierung der automatischen Backups nach der Veröffentlichung der Webapplikation<br>14 |     |
| 9   | Fazit .....  | 14  |
| 9.1 | Soll-/Ist-Vergleich.....   | 14  |
| 9.2 | Lessons Learned.....   | 15  |
| 9.3 | Ausblick.....  | 15  |
|     | Literaturverzeichnis .....   | 16  |
|     | Anhang.....  | i   |
| A1  | Detaillierte Zeitplanung .....   | i   |
| A2  | Gantt-Diagramm .....   | ii  |
| A3  | Personalkosten .....   | ii  |
| A4  | Sachmittelkosten .....   | ii  |
| A5  | EPK.....   | iii |
| A6  | Lastenheft (Auszug) .....  | iv  |
| A7  | Dialogstruktur .....   | v   |
| A8  | Mockups .....  | vi  |
| A9  | Entity-Relationship-Modell .....   | vii |
| A10 | Tabellenmodell .....   | vii |

## Inhaltsverzeichnis

---

|     |   |      |
|-----|---|------|
| A11 | Klassendiagramm der Controller .....          | viii |
| A12 | Aktivitätsdiagramme.....                      | viii |
| A13 | Pflichtenheft (Auszug).....                   | ix   |
| A14 | Filterung der Tierliste nach dem Status ..... | x    |
| A15 | Test für die Filterung nach dem Status .....  | xi   |
| A16 | Ausdrucken mehrerer Tierseiten .....          | xii  |
| A17 | Test der automatisierten Backups .....        | xii  |
| A18 | Eingabeformular für eine neue Behandlung..... | xiii |
| A19 | Benutzerdokumentation (Auszug) .....          | xiii |
| A20 | Entwicklerdokumentation (Auszug).....         | xiv  |
| A21 | Soll-/Ist-Vergleich.....                      | xiv  |
| A22 | Abnahmeprotokoll.....                         | xv   |

## Abbildungsverzeichnis

---

### Abbildungsverzeichnis

|   |      |
|---|------|
| Abbildung 1: Gantt-Diagramm .....   | ii   |
| Abbildung 2: EPK.....   | iii  |
| Abbildung 3: Dialogstruktur der Pflegeseite .....                                 | v    |
| Abbildung 4: Mockup der Tierliste.....  | vi   |
| Abbildung 5: Mockup des Formulars für neue Behandlungen.....                      | vi   |
| Abbildung 6: Entity-Relationship-Modell .....                                     | vii  |
| Abbildung 7: Tabellenmodell .....   | vii  |
| Abbildung 8: Klassendiagramm der Controller .....                                 | viii |
| Abbildung 9: Aktivitätsdiagramm für die Druckfunktion.....                        | viii |
| Abbildung 10: Aktivitätsdiagramm für die Speicherung einer neuen Behandlung ..... | viii |
| Abbildung 11: Screenshot der Testdurchführung.....                                | xi   |
| Abbildung 12: Screenshot der erstellten Backups .....                             | xii  |
| Abbildung 13: Screenshot des Restores.....  | xii  |
| Abbildung 14: Screenshot des Eingabeformulars .....                               | xiii |
| Abbildung 15: Screenshot der Online-Hilfe.....                                    | xiii |
| Abbildung 16: Abnahmeprotokoll.....   | xv   |

**Tabellenverzeichnis**

---

## Tabellenverzeichnis

|   |     |
|---|-----|
| Tabelle 1: grobe Zeitplanung .....          | 2   |
| Tabelle 2: Kostenaufstellung .....          | 6   |
| Tabelle 3: detaillierte Zeitplanung .....   | i   |
| Tabelle 4: Personalkosten .....             | ii  |
| Tabelle 5: einmalige Sachmittelkosten ..... | ii  |
| Tabelle 6: laufende Sachmittelkosten .....  | ii  |
| Tabelle 7: Soll-/Ist-Vergleich .....        | xiv |
| Tabelle 8: Zeitplanung .....                | xiv |

## Verzeichnis der Listings

---

### Verzeichnis der Listings

|  |     |
|--|-----|
| Listing 1: Beispiel für named scopes .....       | 11  |
| Listing 2: Methode zur Filterung des Status..... | xi  |
| Listing 3: Test zur Filterung des Status.....    | xi  |
| Listing 4: Methode zum Erstellen der PDF.....    | xii |

## Abkürzungsverzeichnis

CRUD.....*Create Read Update Delete*

CSS.....*Cascading Style Sheets*

DBMS.....*Datenbankmanagementsystem*

DRY.....*Don't Repeat Yourself*

ERM.....*Entity-Relationship-Modell*

eRuby.....*embedded Ruby*

HTML.....*Hypertext Markup Language*

MoSCoW.....*Must Should Could Won't*

RoR.....*Ruby on Rails*

Sass.....*Sysntactically Awesome Stylesheets*

SCSS.....*Sassy CSS*



## Einleitung

---

# 1 Einleitung

## 1.1 Projektumfeld

Das Projekt wurde von der Autorin in den Räumlichkeiten des Berufsfeldes „Büro & IT“ des Liebenau Berufsbildungswerk gemeinnützige GmbH durchgeführt, wo die Ausbildung für die Kaufleute für Büromanagement und Fachinformatiker stattfindet. Das Berufsbildungswerk ist eine Tochtergesellschaft der Stiftung Liebenau und bildet junge Menschen mit erhöhtem Förderbedarf in 54 Berufen aus. Daneben bietet es auch berufliche Wiedereingliederung, Weiterbildung und Umschulung an.<sup>1</sup>

In Auftrag gegeben wurde das Projekt vom Argenhof – Lebenswürde für Tiere e.V. Der Argenhof ist ein Gnadenhof in Amtzell, der über 170 Tieren auf einer Fläche von 7,5 ha eine Heimat bietet. Großteils stammen sie aus Beschlagnahmen oder aus anderen Tierheimen und Tierschutzorganisationen. Der Schwerpunkt liegt auf der Resozialisierung von verhaltensauffälligen Hunden, Pferden und anderen Tieren. Der Argenhof finanziert sich als gemeinnütziger Verein nahezu vollständig aus privaten Spenden und wird neben 14 fest angestellten Mitarbeitern von über 50 ehrenamtlichen Helfern unterstützt.<sup>2</sup>

## 1.2 Projektziele

Unser Kunde, der Argenhof – Lebenswürde für Tiere e.V., ist dazu verpflichtet, ein Tierbestandsbuch zu führen, welches alle notwendigen Daten zu jedem einzelnen Tier beinhaltet, das dort lebt. Ziel des Projekts ist es, das Führen des Tierbestandsbuchs zu vereinfachen und dadurch Fehler zu vermeiden und Zeit einzusparen.

Deshalb soll zum vorläufigen Tierbestandsbuch (siehe Abschnitt 1.5 Projektabgrenzung) eine Möglichkeit zur Speicherung der Pflegedaten<sup>3</sup> hinzugefügt werden. Dabei soll es möglich sein, eine bestimmte Eintragung für mehrere Tiere gleichzeitig vornehmen zu können. Für einen besseren Überblick sollen die Behandlungen in einer Übersicht gesammelt werden, mit der Möglichkeit, die Pflegedaten nach mehreren Kriterien zu filtern. Außerdem sollen die Tierdaten<sup>4</sup> auf einer eigenen Seite in Tabellenform dargestellt und ebenfalls nach mehreren Kriterien gefiltert werden können. Da das Tierbestandsbuch dem Veterinäramt regelmäßig bei unangekündigten Kontrollen ausgedruckt vorgelegt werden muss, soll es möglich sein, die einzelnen Seiten mit den Daten zum jeweiligen Tier gesammelt ausdrucken zu können. Zudem soll die Datenbank in regelmäßigen Abständen automatisch gesichert werden. Die Projektziele sind in Abschnitt 3.2 (Soll-Konzept) noch einmal genauer erläutert.

## 1.3 Projektbegründung

Durch das Projekt werden Eingabefehler vermieden, die z.B. dadurch entstehen, dass eine große Menge an Pflegedaten wiederholt eingetragen werden muss. Außerdem werden durch die Reduzierung auf eine einmalige Eingabe auch Arbeitszeit und dadurch Kosten eingespart. Dies ist für den Argenhof von großer Bedeutung, da er von Spenden lebt und dadurch darauf angewiesen ist, das vorhandene Geld und die Zeit möglichst effizient einzusetzen. Zudem ermöglicht die Filterung der Daten eine bessere

---

<sup>1</sup> (Aufgabenfeld Bildung, 2019)

<sup>2</sup> (Der Argenhof, 2019)

<sup>3</sup> Zu den Pflegedaten gehören z.B. Behandlungen vom Tierarzt/Hufschmied, Impfungen, Entwurmungen etc.

<sup>4</sup> Zu den Tierdaten gehören z.B. der Name, das Geschlecht, die Tierart etc.

## Projektplanung

---

Übersichtlichkeit und erleichtert die Suche nach bestimmten Informationen. Durch die Automatisierung der Backups wird außerdem eine höhere Datensicherheit gewährleistet.

### 1.4 Projektschnittstellen

Für die Datenverarbeitung ist der Zugriff auf eine PostgreSQL-Datenbank notwendig. Diese befindet sich auf dem Server, auf dem auch die Webapplikation veröffentlicht wird. Ansonsten ist das Projekt unabhängig von anderen technischen Schnittstellen.

Genehmigt wurde das Projekt von der Betriebsleiterin des Bereichs „Büro und IT“ des Berufsbildungswerks. Die Endbenutzer sind die Verwaltungsmitarbeiter des Argenhofs, die von der Autorin für die Verwendung geschult wurden. Nach Beendigung des Projekts wurde das Tierbestandsbuch zur weiteren Betreuung an zwei Mitauszubildende übergeben, da die Autorin das Berufsbildungswerk verlassen hat.

Der Einstieg in das Projekt erfolgte durch die gemeinsame Erstellung des Lastenhefts mit dem Argenhof und die Abnahme der Webapplikation bildete den Ausstieg.

### 1.5 Projektabgrenzung

Um den Verwaltungsmitarbeitern den Umstieg auf das Online-Tierbestandsbuch zu erleichtern, wurde von der Autorin bereits eine vorläufige Version eingeführt. Diese wurde in Abstimmung mit dem Argenhof implementiert und ist seit Februar 2019 in Benutzung.

Dieses Grundgerüst beinhaltet eine Startseite, auf der ein Passwort eingetragen werden muss, um das System zu entsperren. Außerdem sind Formulare vorhanden, um eine neue Tierübersicht zu erstellen oder die bestehenden Daten zu ändern, sowie zu löschen. Dem Benutzer steht eine Online-Hilfe zur Verfügung, in der die Antworten auf die häufigsten Fragen zu finden sind.

## 2 Projektplanung

### 2.1 Zeitplanung

Der Autorin stand ein Durchführungszeitraum von 70 Stunden zur Verfügung. Eine grobe Zeitplanung findet sich in Tabelle 1: grobe Zeitplanung. Eine detailliertere Übersicht mit den einzelnen Teilaufgaben kann in Anhang A1 (Detaillierte Zeitplanung) eingesehen werden. Die Projektphasen basieren auf dem Wasserfallmodell. Näheres dazu ist in Abschnitt 2.5 (Entwicklungsprozess) zu finden.

| Projektphase    | geplante Zeit in h |
|-----------------|--------------------|
| Analyse         | 3                  |
| Entwurf         | 13                 |
| Implementierung | 36                 |
| Einführung      | 5                  |
| Dokumentation   | 13                 |
| <b>Gesamt</b>   | <b>70</b>          |

Tabelle 1: grobe Zeitplanung

## Projektplanung

---

### 2.2 Terminplanung

Basierend auf der Zeitplanung in Anhang A1 (Detaillierte Zeitplanung) wurde das Gantt-Diagramm<sup>5</sup> in Anhang A2 (Gantt-Diagramm) zur Terminplanung erstellt, wobei von einer durchschnittlichen täglichen Arbeitszeit von 7 Stunden ausgegangen wurde. Da die Zeitangaben nur in Tagen und nicht in Stunden möglich waren, wurden die einzelnen Teilschritte zu größeren Paketen zusammengefasst. Für die Dokumentation, die parallel zum Entwicklungsprozess erfolgt, wurde zu jedem Paket Dokumentationszeit hinzugerechnet. Der detaillierte Ablauf ist in Abschnitt 2.5 (Entwicklungsprozess) genauer beschrieben. Die Einführungsphase wurde auf den 24.05.2019 festgelegt, da an diesem Tag die Übergabe des Tierbestandsbuchs an den Argenhof stattfand. Das Projektende wurde auf den nachfolgenden Werktag gelegt, um die Abnahme etc. noch dokumentieren zu können.

### 2.3 Personalplanung

Die Personalplanung ist in Anhang A3 (Personalkosten) zu sehen. Die Autorin war als Projektleiterin für die Durchführung des Projekts zuständig. Bei der Schulung für die Verwaltungsmitarbeiter des Argenhofs wurde sie von einem der Mitauszubildenden, an den das Projekt übergeben wurde, begleitet. Der Ausbilder stand bei organisatorischen Fragen zur Verfügung. Die Stundensätze sind Pauschalsätze, die sich unter anderem aus dem Bruttostundenlohn, den Sozialaufwendungen des Arbeitgebers und der Ressourcennutzung zusammensetzen.

### 2.4 Sachmittelplanung

Um die anfallenden Kosten möglichst gering zu halten, wurde bei der Auswahl der Software darauf geachtet, dass diese kostenfrei zur Verfügung stehen, z.B. als Open-Source-Software oder durch bereits vorhandene Lizenzen. Deshalb wurde die folgende kostenfreie Software zur Durchführung des Projekts ausgewählt:

- Oracle VM Virtual Box (virtuelle Maschine)
- Ubuntu (Betriebssystem für die virtuelle Maschine)
- Pencil (Software für Mockups)
- Sublime Text (Texteditor)
- SQLite und PostgreSQL (DBMS)
- Ruby on Rails (Web-Framework)

Die einmaligen Kosten sind in Anhang A4 (Sachmittelkosten) zu sehen. Dort wurden die Bustickets für die Fahrt zum Argenhof miteinkalkuliert, wo die Schulung der Mitarbeiter durchgeführt wird. Zusätzlich zum Notebook, das zur Planung und Dokumentierung verwendet wird, wurde für die Durchführung des Projekts ein Desktop-PC angeschafft, auf dem eine virtuelle Maschine mit der Entwicklungsumgebung installiert wurde. Dadurch steht ausreichend Arbeitsspeicher zur Verfügung und durch eine direkte Anbindung ans Internet können Probleme durch die Firewall verhindert werden. Außerdem wurden die laufenden Kosten berechnet, die ebenfalls in Anhang A4 (Sachmittelkosten) zu sehen sind.

### 2.5 Entwicklungsprozess

Aufgrund des geringen Zeitaufwands und, weil alle Anforderungen bereits im Vorfeld im Lastenheft sehr genau definiert wurden, wurde das Projekt nach dem erweiterten Wasserfallmodell durchgeführt. Die

---

<sup>5</sup> Das Diagramm wurde mit dem Tool GanttProject erstellt.

## Analysephase

---

Reihenfolge der einzelnen Teilschritte ist in Anhang A1 (Detaillierte Zeitplanung) aufgelistet. Bei der Festlegung der Reihenfolge wurden die im Lastenheft in Anhang A6 (Lastenheft (Auszug)) festgelegten Priorisierungen miteinbezogen. Es wurden zunächst die Funktionen implementiert, die viele Must-Kriterien enthalten und ganz am Ende die automatischen Backups, da sie nur Could-Kriterien enthalten. Dadurch wurde sichergestellt, dass die essentiellen Funktionen zum Übergabetermin alle implementiert wurden, auch wenn es zu Zeitverzögerungen kommen sollte.

Die Implementierungsphase wurde in iterativen Zyklen durchgeführt, wobei immer ein Feature komplett entwickelt wurde. Parallel zur Implementierung fand auch die Testphase statt. Für jede Funktion wurde entweder schon vor der Implementierung oder direkt im Anschluss ein Test erstellt.

## 3 Analysephase

### 3.1 Ist-Analyse

Auf dem Argenhof werden zwei Mitarbeiter in der Verwaltung beschäftigt, deren Aufgabe es unter anderem ist, das Tierbestandsbuch aktuell zu halten. Dort müssen die Daten aller Tiere enthalten sein, die auf dem Hof leben, wie z.B. die Rasse, das Geschlecht und die Kennzeichnung (Chipnummer oder ähnliches). Der Argenhof besitzt für jedes Tier eine Word-Datei, in der diese Daten lokal auf einem Notebook gespeichert werden. Zusätzlich werden Excel-Tabellen geführt, in denen die Tiere nach Tierart geordnet gesammelt werden. Das Tierbestandsbuch muss dem Veterinäramt regelmäßig bei unangekündigten Kontrollen ausgedruckt vorgelegt werden.

Mit dem Webframework Ruby on Rails wurde bereits eine Webapplikation entwickelt, die das Speichern der Tierdaten erleichtert. Dort werden sie mithilfe von Textfeldern und grafischen Interaktionselementen in Eingabeformulare eingetragen. Alle Tiere werden in einer einfachen Liste gesammelt. Um unbefugte Zugriffe zu verhindern wird das Tierbestandsbuch mit einem Passwort geschützt. Die Bereitstellung erfolgt über Git auf der Cloud Plattform Heroku. Über das Internet kann darauf mittels eines Browsers zugegriffen werden. Heroku stellt außerdem eine PostgreSQL-Datenbank zur Verfügung, in der die Daten gespeichert werden.

Allerdings kann die Webapplikation die ursprünglichen Word-Dokumente noch nicht voll ersetzen, da keine Pflegedaten eingetragen werden können. Die Tiere erhalten unterschiedliche Behandlungen, wie Impfungen, Tierarztbehandlungen oder Behandlungen beim Hufschmied. Die Daten dazu (Art der Behandlung, Datum und Bemerkungen) werden derzeit noch in den Word-Dokumenten für jedes Tier einzeln eingetragen. Diese Vorgehensweise ist jedoch sehr zeitaufwendig und fehleranfällig. Wenn beispielsweise der Hufschmied da war, müssen für 30 bis 40 Pferde und Esel, sowie für etwa 30 Schafe und Ziegen die Behandlungen eingetragen werden. Dabei nimmt schon die Suche nach der richtigen Word-Datei viel Zeit in Anspruch, da auf dem Argenhof durchschnittlich 170 Tiere leben. Dazu kommen die Dateien verstorbener und vermittelter Tiere, die weiterhin aufbewahrt werden müssen. Obwohl die Eintragungen für eine Hufbehandlung bei den meisten Tieren gleich ausfallen, ist die Fehlerquote bei einer derart großen Anzahl dennoch hoch. Die parallele Nutzung der Webapplikation und der Word-Dateien birgt außerdem die Gefahr einer Inkonsistenz in den Daten. Auch gibt es bisher keine Übersicht über die Pflegedaten, um z.B. überprüfen zu können, welche Tiere demnächst geimpft werden müssen. Dazu müssen alle Word-Dateien einzeln geöffnet werden.

## Analysephase

---

Auch andere Eintragungen, die direkt in der Webapplikation vorgenommen werden, sind trotz der vereinfachten Eingabe aufwendig, da das jeweilige Tier in der unsortierten Liste erst gesucht werden muss. Eine Übersicht wie in den Excel-Tabellen ist dort noch nicht vorhanden.

Da das Tierbestandsbuch dem Veterinäramt vorgelegt werden muss, müssen die Daten regelmäßig ausgedruckt werden. Da auch dies für jede Datei einzeln erfolgt, ist der Aufwand sehr hoch und es kann leicht etwas vergessen werden. Werden Dateien aus der Webapplikation heraus ausgedruckt, werden zudem nicht benötigte Komponenten wie das Menü und das Logo mitabgebildet.

Um die Daten zu sichern, werden die Word- und Excel-Dateien in unregelmäßigen Abständen auf einen USB-Stick kopiert. Für die Datenbank der Webapplikation gibt es derzeit keine Sicherung. Bei einem Ausfall sind also unter Umständen nicht alle Daten wiederherstellbar und der USB-Stick bietet wenig Sicherheit. Aus diesen Gründen soll die Webapplikation erweitert werden.

### 3.2 Soll-Konzept

Im Zuge des Projekts soll die Speicherung der Pflegedaten ergänzt werden. Dabei soll es möglich sein, eine bestimmte Eintragung für mehrere Tiere gleichzeitig vornehmen zu können. Dadurch werden die Fehleranfälligkeit und der zeitliche Aufwand reduziert. Für einen besseren Überblick sollen die Behandlungen in einer Übersicht gesammelt werden, mit der Möglichkeit, die Pflegedaten nach mehreren Kriterien zu filtern.

Die Excel-Tabellen sollen durch eine eigene Seite in der Webapplikation ersetzt werden. Dort sollen die Daten in Tabellenform dargestellt und ebenfalls nach mehreren Kriterien gefiltert werden können.

Um das Ausdrucken zu erleichtern, soll eine Komponente eingebaut werden, über die mehrere Tiere ausgewählt und anschließend in einem Zug die jeweiligen Daten ausgedruckt werden können. Dafür wird ein eigenes Druck-Layout angefertigt. Außerdem sollen automatische Backups eingeführt werden, mit denen tägliche und zusätzlich wöchentliche Sicherungen vorgenommen werden.

Die Erweiterung der Webapplikation wird mit Ruby on Rails umgesetzt. Für die Durchführung des Projekts steht ein Desktop-PC zur Verfügung, auf dem eine virtuelle Maschine mit Ruby installiert ist. Das Oberflächenlayout wird mit Bootstrap erstellt und anschließend mit HTML, SCSS und eRuby angepasst und gestaltet.

Während der Entwicklung wird zu Testzwecken eine SQLite-Datenbank genutzt. Nach der Veröffentlichung der Webapplikation auf Heroku werden die Echtdaten dann in einer von der Plattform bereitgestellten PostgreSQL-Datenbank gespeichert. Zur Qualitätssicherung werden automatisierte Tests verwendet.

Für die Schulung stehen die Verwaltungsmitarbeiter des Argenhofs zur Verfügung. Außer der Entwicklerin wird kein weiteres Personal benötigt.

Um das Projekt optimal planen zu können, wurde bereits gemeinsam mit Argenhof ein Lastenheft angefertigt.

## Analysephase

---

### 3.3 Wirtschaftlichkeitsanalyse

#### 3.3.1 Make-or-Buy-Entscheidung

Die angebotenen Hilfen zum Führen eines Tierbestandsbuchs beschränken sich hauptsächlich auf Excel-Tabellen, die zum Download bereitstehen. Diese bieten jedoch keinen Mehrwert gegenüber den bereits vorhandenen Excel-Tabellen des Argenhofs.

Es gibt aktuell lediglich eine Software<sup>6</sup> für Hundeschulen und Pensionen, die unter anderem ein Tierbestandsbuch enthält. Allerdings sind daneben viele weitere Features vorhanden, die vom Argenhof nicht benötigt werden, wie z.B. die Erstellung von Rechnungen für Pensionshunde. Die Software kann zudem nur Hunde erfassen, während auf dem Argenhof etwa 16 verschiedene Tierarten leben. Auch für die große Menge an Tieren ist die Software nicht ausgelegt, was die Arbeit unübersichtlich und zeitaufwendig machen würde. Aufgrund dessen wurde entschieden, die vorhandene Webapplikation zu erweitern, damit die Software genau auf die Bedürfnisse des Argenhofs angepasst werden kann.

#### 3.3.2 Projektkosten

Bei dem Projekt müssen Kosten für das Personal, sowie für die Sachmittel aufgebracht werden. Die Berechnung der einzelnen Posten ist in Kapitel 2 (Projektplanung) genauer aufgeführt. Die Gesamtkosten, die sich daraus ergeben, finden sich in Tabelle 2: Kostenaufstellung.

| Posten                     | Kosten            |
|----------------------------|-------------------|
| Personal                   | 1.500,00 €        |
| einmalige Sachmittelkosten | 504,12 €          |
| laufende Sachmittelkosten  | 26,00 €           |
| <b>Gesamt</b>              | <b>2.030,12 €</b> |

Tabelle 2: Kostenaufstellung

#### 3.3.3 Amortisationsdauer

Da das Berufsbildungswerk eine gemeinnützige Organisation ist und es sich um ein soziales Projekt handelt, wurden die Kosten dafür dem Argenhof in Absprache mit der Betriebsleiterin nicht in Rechnung gestellt. Allerdings ist geplant, das Tierbestandsbuch in Zukunft auch anderen Tierschutzorganisationen zur Verfügung zu stellen. Das Tierbestandsbuch soll dabei zu einem Preis von 400 € pro einmalige Lizenz verkauft werden.

Dadurch liegt die Gewinnschwelle bei ca.:

$$\frac{2.030,12 \text{ €}}{400,00 \text{ €}} \approx 5 \text{ Lizenzen}$$

Ab dem Verkauf der fünften Lizenz hat sich das Projekt also amortisiert.

---

<sup>6</sup> (Schnatz, 2019)

## Entwurfsphase

---

### 3.3.4 Nicht-monetäre Vorteile

Das Projekt lohnt sich vor allem aufgrund der nicht-monetären Vorteile: Zum einen kann das Projekt den Grundstein für eine weitere Zusammenarbeit mit dem Argenhof legen. Der Argenhof arbeitet nämlich auch im pädagogischen Bereich mit Menschen mit Behinderungen, psychischen Erkrankungen und ähnlichem, wovon die Teilnehmer des Berufsbildungswerks profitieren können.

Zum anderen kann das Berufsbildungswerk durch das soziale Projekt sein Image verbessern und der Bekanntheitsgrad steigt. In Folge dessen können mehr Spender gewonnen werden.

### 3.4 Prozessanalyse

Um die Prozesse bezüglich des Tierbestandsbuchs zu analysieren, wurde für das Eintragen der Pflegedaten eine EPK entworfen, welche in Anhang A5 (EPK) zu sehen ist. Das Online-Tierbestandsbuch soll vor allem den Prozessabschnitt der Verwaltung effizienter gestalten, indem mehrere gleiche Einträge in einem Schritt gemacht werden können.

### 3.5 Lastenheft

Das Lastenheft wurde bereits im Voraus des Projekts von der Autorin gemeinsam mit Verwaltungsmitarbeitern des Argenhofs erstellt. Dadurch wurden die Anforderungen schon früh klar definiert, weswegen die weitere Planung schneller und präziser gemacht werden konnte.

Ein Auszug des Lastenhefts mit den gewünschten Funktionen sind aus Anhang A6 (Lastenheft (Auszug)) zu entnehmen. Dabei wurden die Funktionen mithilfe der MoSCoW-Methode priorisiert und je nach Wichtigkeit in verschiedenen Farben dargestellt. Der Aufbau erfolgte nach Balzert.<sup>7</sup>

## 4 Entwurfsphase

### 4.1 Zielplattform

Bereits das Grundgerüst des neuen Tierbestandsbuchs wurde als Webapplikation erstellt, welches nun auch in dieser Form erweitert werden soll. Der Vorteil davon ist, dass auf Seiten des Clients lediglich ein Webbrowser nötig ist. Die Software muss nicht extra installiert werden, was die Benutzerfreundlichkeit erhöht. Außerdem kann durch die Client-Server-Architektur von mehreren Geräten aus gleichzeitig auf die Software zugegriffen werden. Auch die Wartung wird vereinfacht, da das Wartungspersonal über das Internet einfach auf die Webapplikation zugreifen und z.B. eine neue Version hochladen kann, die anschließend sofort vom User genutzt werden kann.<sup>8</sup>

Auch die Veröffentlichung der Webapplikation auf der Cloud Plattform Heroku soll beibehalten werden. Die Installation und die Wartung eines Webanwendungsservers auf dem Notebook des Argenhofs wäre sehr aufwendig und der PC würde mit großer Wahrscheinlichkeit nicht die nötige Rechenleistung aufbringen können. Extra dafür einen eigenen Server zu kaufen, wäre für den Argenhof zu kostspielig.

---

<sup>7</sup> (Krypczyk & Bochkor, 2018, S. 222)

<sup>8</sup> (Krypczyk & Bochkor, 2018, S. 250-258)

## Entwurfsphase

---

Die Autorin hat sich für Heroku als Plattform entschieden, da dort das Tierbestandsbuch in dem nötigen Umfang kostenlos bereitgestellt werden kann. Dies ist für den Argenhof sehr wichtig, da er sich ausschließlich über Spenden finanziert. Außerdem sind das Setup und die Bedienbarkeit bei Heroku sehr einfach, wodurch bei der Einführungsphase und der Wartung Zeit eingespart werden kann.

Als DBMS wurde PostgreSQL gewählt, da die Autorin bereits in früheren Projekten gute Erfahrungen damit gesammelt hat. Weitere Gründe dafür sind die Plattformunabhängigkeit und die freie Nutzung, weil es sich um eine Open-Source-Software handelt.

### 4.2 Architekturdesign

Auch das Webframework Ruby on Rails (RoR) wurde beibehalten, da es sich sehr gut für die Webapplikation eignet. RoR ist ein Open-Source Web-Framework, das auf der Programmiersprache Ruby basiert.

Für die Implementierung standen außerdem Java und PHP zur Auswahl. Die Autorin hat sich für Ruby entschieden, da es gegenüber Java und PHP einige Vorteile bietet. Bei Ruby ist z.B. viel weniger Code notwendig als für Java. Außerdem ist der Entwicklungsaufwand mit Ruby wesentlich geringer, da es sich zusätzlich noch um eine interpretierte Skriptsprache handelt. Ein Problem von PHP ist die Objektorientierung, da diese erst nachträglich entwickelt wurde und deshalb entsprechende Frameworks benötigt werden. RoR hingegen ist vollständig objektorientiert. Auch, dass die Autorin bereits Erfahrungen mit RoR hat im Gegensatz zu PHP, spielte eine Rolle.<sup>9</sup>

RoR ist nach dem Model-View-Controller Paradigma konzipiert. Dabei werden die Schichten Datenhaltung (Model), Präsentation (View) und Steuerung (Controller) voneinander getrennt. Dadurch können die Interaktionen eines Benutzers sauber von den dadurch veränderten Daten und deren Darstellung getrennt werden.<sup>10</sup> RoR ist geprägt von zwei Prinzipien: „Don’t Repeat Yourself“ (DRY) und „Convention Over Configuration“. Das bedeutet, dass Redundanz im Code vermieden werden und das Standardverhalten für zu konfigurierende Aspekte genutzt werden soll. Dies bewirkt, dass die Entwicklung mit RoR sehr schnell und effizient erfolgen kann.<sup>11</sup>

Die Oberflächen wurden als Webinterface mit HTML und Sass umgesetzt. Sass wurde anstelle von CSS gewählt, da Sass mehr Möglichkeiten bietet und die Arbeit vor allem bei Änderungen vereinfacht. Es lassen sich z.B. Variablen, Nestings und Mixins einsetzen und im Gegensatz zu Less sind auch Schleifen und Fallunterscheidungen möglich. Aufgrund der Nähe zu CSS wurde als Syntax für Sass SCSS eingesetzt.<sup>12</sup>

### 4.3 Entwurf der Benutzeroberfläche

Zuerst wurde die Dialogstruktur entworfen, die auszugsweise in Anhang A7 (Dialogstruktur) zu sehen ist. Diese wurde nach der Erstellung des Pflichtenhefts mit den entsprechenden Identifikationsnummern für jede Funktion ergänzt, um sicherzustellen, dass alle vereinbarten Funktionen enthalten sind.

Für die Planung der Oberflächengestaltung wurden mit der Open-Source-Software Pencil Mockups angefertigt, die auszugsweise in Anhang A8 (Mockups) einsehbar sind. Bei der Gestaltung der Entwürfe wurde unter Miteinbeziehung des Auftraggebers darauf geachtet, das Layout an die schon bestehenden

---

<sup>9</sup> (Ruby on Rails oder php, 2019)

<sup>10</sup> (Lahres & Rayman, 2015)

<sup>11</sup> (Schätzlein, 2019)

<sup>12</sup> (SASS: CSS auf dem nächsten Level?, 2019)



## Entwurfsphase

---

Seiten anzupassen, um Einheitlichkeit zu schaffen. Im Vordergrund stand eine möglichst intuitive Bedienung und eine übersichtliche Gestaltung, da die Verwaltungsmitarbeiter des Argenhofs hauptsächlich andere Aufgabengebiete haben und dadurch teilweise wenig Erfahrung im Umgang mit PCs. Da bereits die vorläufige Version des Tierbestandsbuchs in Verwendung ist, konnten sich die Endbenutzer schon an den Aufbau gewöhnen. Dies erleichtert den Umgang mit den neuen, etwas komplexeren Seiten.

In Absprache mit dem Auftraggeber wurde die Farbgestaltung an das Corporate Design des Argenhofs angepasst, sowie das Logo eingebunden.

### 4.4 Datenmodell

Für das Datenmodell wurde zunächst ein ERM erstellt, das in Anhang A9 (Entity-Relationship-Modell) einzusehen ist. Dort sind die wichtigsten Entitäten, die schon vorhanden waren, leicht transparent dargestellt. Die Entitäten, die im Rahmen des Projekts hinzugefügt wurden, wurden optisch hervorgehoben. Das ERM wurde auf Grundlage des Lastenhefts in Anhang A6 (Lastenheft (Auszug)) erstellt.

Anschließend wurde das ERM normalisiert und in ein Tabellenmodell überführt. Dabei wurde die m:n-Beziehung zwischen den Entitäten `Tier` und `Behandlung` aufgelöst, um Redundanzen und Anomalien zu vermeiden. In Anhang A10 (Tabellenmodell) sind die wichtigsten Relationen mit ihren wesentlichen Attributen zu sehen. Die Benennung der Tabellen und der Fremd- bzw. Primärschlüssel erfolgte nach den Namenskonventionen von RoR.

Da die Daten einer Behandlung (Art, Datum und Beschreibung) in der Regel für mehrere Tiere gleich ist, wurden sie in einer Relation zusammengefasst. Außerdem wurde eine Änderung an der Relation `animals` vorgenommen. Ursprünglich wurden dort alle Kennzeichnungsmöglichkeiten der Tiere (Chip, Tätowierung, Equidenpass) als eigene Attribute dargestellt. Da jedoch jedes Tier nur eine Kennzeichnung besitzt und diese alle vom Datentyp `String` sind, wurden sie in dem neuen Attribut `mark` zusammengefasst.

### 4.5 Geschäftslogik

Für die Planung der Geschäftslogik wurde mit der Software Violet UML Editor zunächst ein grobes Klassendiagramm der Controller erstellt. In Anhang A11 (Klassendiagramm der Controller) sind die geplanten Controller-Klassen zu sehen, die zum Tierbestandsbuch hinzugefügt werden sollen. Diese enthalten Methoden nach dem CRUD-Prinzip und erben alle von `ApplicationController`, was wiederum eine Unterklasse von `ActionController::Base` ist. Action Controller hilft durch Konventionen bei der Erstellung der Controller.<sup>13</sup>

Für die Druckfunktion wurde außerdem das Aktivitätsdiagramm in Anhang A12 (Aktivitätsdiagramme) erstellt, um den Ablauf grafisch darzustellen. Aufgrund der Komplexität war vor der Implementierung noch nicht absehbar, welche konkreten Methoden für die Umsetzung benötigt werden. Deshalb wurde erst im Nachhinein der Controller `DownloadsController` mit den zugehörigen Methoden im Klassendiagramm ergänzt. Das zweite Aktivitätsdiagramm in Anhang A12 (Aktivitätsdiagramme) zeigt den Ablauf bei der Speicherung einer neuen Pflegebehandlung.

---

<sup>13</sup> (Action Controller Overview, 2019)

## Implementierungsphase

---

### 4.6 Maßnahmen zur Qualitätssicherung

Um eine möglichst hohe Qualität der Webapplikation sicherzustellen, wurde jede Funktion, wie schon in Abschnitt 2.5 (Entwicklungsprozess) beschrieben, bereits während der Implementierungsphase getestet. Dies geschah durch Unit- und Integrationstests.

Mithilfe des Tools Guard wurden die Tests automatisch ausgeführt. Dieses Vorgehen stellt eine hohe Qualität der Software sicher, da Fehler frühzeitig erkannt werden und regelmäßig überprüft wird, ob Änderungen an einer Stelle Probleme im restlichen Code bewirken.

Die Grundlage für die automatisierten Tests bildeten die im Pflichtenheft in Anhang A13 (Pflichtenheft (Auszug)) definierten globalen Testfälle. Da die Testung der Webapplikation also parallel zur Implementierung erfolgte, war eine zusätzliche Testphase nicht mehr notwendig.

### 4.7 Pflichtenheft/Datenverarbeitungskonzept

Zum Abschluss der Entwurfsphase wurde ein Pflichtenheft erstellt, das auf dem Lastenheft in Anhang A6 (Lastenheft (Auszug)) basiert. Ausschnitte des Pflichtenhefts sind in Anhang A13 (Pflichtenheft (Auszug)) zu sehen. Wie das Lastenheft, basiert auch die Gliederung des Pflichtenhefts auf Balzert.

## 5 Implementierungsphase

### 5.1 Erstellen der Datenbanktabellen

Die erforderlichen Datenbanktabellen wurden mithilfe von Rails generiert. In den generierten Migrationsdateien wurden anschließend manuell die Fremdschlüssel eingetragen. Außerdem wurden die Relationen in den Models definiert. Über die seed-Datei wurde die Datenbank mit den Werten für die Pflegebehandlungen gefüllt.

### 5.2 Implementierung und Test der Pflegedaten

Um die Oberflächen zu gestalten, wurden zunächst die Controller generiert. In diesem Schritt wurden auch die zugehörigen Views automatisch erstellt. Das Formular wurde in ein Partial<sup>14</sup> ausgelagert, damit es sowohl für das Formular zur Eingabe der neuen Daten, als auch zum Ändern der bestehenden Daten genutzt werden kann. Dadurch wird Redundanz im Code vermieden. Mithilfe von eRuby<sup>15</sup> wurde eine If-Verzweigung eingebunden, damit nur dann ein Button zum Löschen der Behandlung angezeigt wird, wenn es sich um das Änderungsformular handelt.

Ursprünglich sollte die Auswahl der Tiere über das Gem<sup>16</sup> `nested_form`<sup>17</sup> erfolgen, weil dafür JavaScript-Kenntnisse nötig, jedoch nicht vorhanden sind. Dies war technisch aber nicht umsetzbar. Deshalb wurde als Alternative das Gem `chosen-rails`<sup>18</sup> verwendet, wie in Abschnitt 8.1 (Multiple-Drop-Down anstelle von Textfeldern) genauer beschrieben wurde.

---

<sup>14</sup> Sub-Template, das in andere Templates eingebunden werden können

<sup>15</sup> Template-Mechanismus zum Einfügen von Ruby-Code in Textdateien

<sup>16</sup> Plugin für Ruby

<sup>17</sup> (Bates, 2019)

<sup>18</sup> (Ho, 2019)

## Implementierungsphase

---

Nachdem die Methoden zum Speichern, Ändern und Löschen der Pflegedaten implementiert wurden, wurden für die Controller und Models Unit-Tests erstellt. Damit wird sichergestellt, dass eine Fehlermeldung erfolgt, wenn Pflichtangaben fehlen und dass nur auf die Funktionen zugegriffen werden kann, wenn das System entsperrt ist.

Im Anschluss wurde die Oberfläche erstellt, auf der die Pflegedaten in Tabellenform dargestellt und gefiltert werden können. Die Filterfunktion wurde mithilfe von `named_scopes`<sup>19</sup> implementiert, die von RoR bereitgestellt werden. Mit diesen `scopes` können durch Lambdas Datenbankabfragen erstellt und miteinander verkettet werden. Um SQL Injections zu verhindern, wurden Parametrisierungen verwendet. Ein Beispiel dafür ist in Listing 1: Beispiel für `named_scope` zu sehen.

```
scope :name_like, ->(animal) { joins(:animal).where("animals.name LIKE ?", "%#{animal}%" ) }
```

Listing 1: Beispiel für `named_scopes`

### 5.3 Implementierung und Test der Tierliste

Für die Erstellung der Oberfläche der Tierliste konnten Teile der Übersicht für die Pflegedaten übernommen werden. Auch hier wurden für die Filterung `scopes` eingesetzt. Bei der Filterung des Jahres wird unterschieden, ob sie in der Entwicklungs- oder der Produktivumgebung ausgeführt wird. In der Produktivumgebung kommt eine PostgreSQL-Datenbank zum Einsatz, während in der Entwicklungsumgebung eine SQLite-Datenbank verwendet wird. Deshalb muss der `scope` angepasst werden, da die beiden DBMS nicht dieselben Datumsfunktionen kennen.

Eine Ausnahme bildet jedoch die Filterung des Status, um die verstorbenen, vermittelten und/oder auf dem Hof lebenden Tiere anzuzeigen. Dafür wurde aus den folgenden Gründen eine Klassenmethode erstellt: Da es in der Datenbank keine eigenen Spalten für den Status gibt, muss überprüft werden, ob ein Sterbe- oder Abgabedatum eingetragen ist. Ein Tier lebt dann aktuell auf dem Hof, wenn diese Daten fehlen. Deshalb finden für jeden Status bestimmte Datenbankabfragen statt, die mit einem OR verknüpft werden müssen. `Scopes` können die Abfragen jedoch nur mit einem AND verknüpfen, was zu Fehlern bei der Filterung führt, wenn mehrere Status ausgewählt werden. Aus diesem Grund wurde eine Klassenmethode erstellt, bei der die SQL-Abfrage abhängig von den Übergabewerten korrekt zusammengesetzt wird. Der Source Code dieser Methode ist in Anhang A14 (Filterung der Tierliste nach dem Status) zu sehen.

Um zu überprüfen, ob die richtigen Datensätze für die jeweilige Filterung zurückgegeben werden, wurden Unit-Tests des Models erstellt. Ein Ausschnitt der Testklasse, sowie das Testergebnis sind Anhang A15 (Test für die Filterung nach dem Status) zu sehen.

### 5.4 Implementierung der Druckfunktion

Um die Tierliste auszudrucken, wurde ein eigenes Stylesheet erstellt, das beim Ausdrucken zum Einsatz kommt. Dabei werden alle Komponenten außer der Tabelle mit den Tierdaten ausgeblendet. Außerdem wird angezeigt, welche Tierarten und Status bei der Filterung ausgewählt wurden, damit die Ausdrucke auch nach längerer Zeit noch nachvollziehbar sind. Diese Angaben werden nur beim Ausdrucken angezeigt.

---

<sup>19</sup> (ActiveRecord::Scoping::Named::ClassMethods, 2019)

## Einführung

---

Die Implementierung der Funktion für das Ausdrucken mehrerer Tierseiten baute auf dem Aktivitätsdiagramm in Anhang A12 (Aktivitätsdiagramme) auf. Für die Generierung der PDFs aus den HTML-Seiten wurde das Gem `PDFKit`<sup>20</sup> verwendet. `PDFKit` wurde gewählt, weil dort das SCSS-Stylesheet für das Layout bei der Generierung verwendet werden kann. Andere Gems (z.B. `Wicked PDF`<sup>21</sup>) nutzen standardmäßig nur HTML.

Für die Druckfunktion wurde ein eigener Controller erstellt, der die Methoden zur Generierung der PDFs beinhaltet. Dabei werden die IDs der ausgewählten Tiere übergeben und für jedes der Tiere eine PDF mit den Daten erstellt. Diese werden gespeichert und anschließend mithilfe des Gems `CombinePDF`<sup>22</sup> zusammengefügt. Ein Ausschnitt des Codes für diese Methode ist in Anhang A16 (Ausdrucken mehrerer Tierseiten) zu sehen. Um die zusammengefügte PDF im Browser anzuzeigen, wird der Benutzer zu einer statischen Seite weitergeleitet, die einen Link zur PDF enthält. Diese öffnet sich dann in einem neuen Tab und kann über den PDF-Reader ausgelesen werden. Das anschließende Löschen der gespeicherten PDFs ist nicht nötig, da in der Produktion die Dateien von Heroku selbst nur temporär für kurze Zeit gespeichert werden. Da der Testaufbau für den geplanten Integrationstest für die Druckfunktion zu komplex gewesen wäre, wurde anstatt dessen manuell getestet, wie in Abschnitt 8.2 (Manuelle Tests anstelle von automatisierten Integrations-Tests) beschrieben wurde.

### 5.5 Implementierung und Test der automatischen Backups

Bei der Implementierung der automatischen Backups für die Datenbank war eine Anpassung in Bezug auf die Reihenfolge der Durchführung nötig. Nähere Informationen dazu in Abschnitt 8.3 (Implementierung der automatischen Backups nach der Veröffentlichung der Webapplikation). Die Autorin hat sich für die Scheduling-Funktion entschieden, die Heroku bereitstellt. Die Vorteile davon sind die einfache Umsetzung und die Speicherung der Backups, sowie der Webapplikation selbst auf derselben Plattform. Es wurde ein Plan erstellt, nach dem immer um 02:00 Uhr nachts die täglichen und wöchentlichen Backups gemacht werden. Die korrekte Erstellung der Backups wurde am nächsten Tag überprüft. Um dies sicherzustellen, wurde ein Restore mit dem letzten Backup durchgeführt. Die Produktivdaten wurden vorher sicherheitshalber lokal gesichert. Die Ergebnisse sind in Anhang A17 (Test der automatisierten Backups) zu sehen.

## 6 Einführung

### 6.1 Bereitstellung der Webapplikation auf dem Cloud-Server

Damit die Webapplikation über das Internet genutzt werden kann, wurde der neue Code nach der Implementierungsphase bei Heroku hochgeladen. Dies geschah über eine Verbindung mit dem Git-Repository. Anschließend wurde die PostgreSQL-Datenbank, die die Produktivdaten enthält, migriert, um die neuen Tabellen hinzuzufügen.

Beim manuellen Testen des veröffentlichten Tierbestandsbuchs trat ein Fehler bei der Anzeige der PDFs auf, weil die zugehörige SCSS-Datei nicht gefunden wird. Eine Internetrecherche ergab, dass dies an

---

<sup>20</sup> (Pace, 2019)

<sup>21</sup> (Sterrett, 2019)

<sup>22</sup> (GitHub, 2019)

## Dokumentation

---

einem Bug des von RoR bereitgestellten Gems `Sprocket Rails` liegt, das für die Asset Pipeline verwendet wird. Das Problem konnte durch eine Anpassung der Rails-Konfiguration behoben werden.

Eine Migration der alten Daten aus den Word- und Excel-Dateien ist nur manuell möglich. Weil dies mit großem Aufwand verbunden ist, wird die Webapplikation in der Übergangszeit parallel zum alten System genutzt werden. So werden die Daten Schritt für Schritt migriert und den Benutzern fällt die Umstellung leichter, da der Übergang nicht so hart ist und langsam Vertrauen zum neuen Tierbestandsbuch entstehen kann.

### 6.2 Schulung des Kunden und Abnahme

Die Schulung der Verwaltungsmitarbeiter des Argenhofs fand in Verbindung mit der Abnahme statt. Dabei haben die Benutzer alle Funktionen des Tierbestandsbuchs getestet und wurden bei Bedarf von der Autorin unterstützt. Da die Webapplikation intuitiv gestaltet wurde und die Benutzer schon den Umgang mit dem vorläufigen Tierbestandsbuch gewohnt waren, waren keine weiteren Schulungsmaßnahmen notwendig. Die Autorin wurde bei der Schulung von einem Mitauszubildenden begleitet, der das Tierbestandsbuch nach Beendigung des Projekts übernommen hat. So konnten sich die Beteiligten kennenlernen und die zukünftige Zusammenarbeit wurde erleichtert.

Der Test des Tierbestandsbuchs durch den Kunden hat ergeben, dass alle Ziele wie vereinbart umgesetzt wurden, weshalb die Webapplikation ohne Vorbehalt abgenommen wurde. Das Abnahmeprotokoll befindet sich in Anhang A22 (Abnahmeprotokoll).

## 7 Dokumentation

Neben der Projektdokumentation wurden eine Entwickler- und eine Benutzerdokumentation erstellt. Die vorhandene Benutzerdokumentation für das vorläufige Tierbestandsbuch wurde um Beschreibungen der neuen Funktionen erweitert. Um einen schnellen und einfachen Zugriff darauf zu ermöglichen, wurde die Benutzerdokumentation als Online-Hilfe in die Webapplikation integriert. In dem nach Themen geordneten Akkordeon findet der Benutzer schnell eine Antwort auf sein konkretes Problem. Ein Screenshot davon befindet sich in Anhang A19 (Benutzerdokumentation (Auszug)).

In der Entwicklerdokumentation befinden sich Informationen zur Datenbank, zum Deployment usw. Sie soll den Auszubildenden, die das Tierbestandsbuch übernehmen, einen Überblick über den Aufbau und die Funktionsweise geben. Ein Auszug davon befindet sich in Anhang A20 (Entwicklerdokumentation (Auszug)).

## 8 Anpassungen

### 8.1 Multiple-Drop-Down anstelle von Textfeldern

Geplant war, die Tiernamen für die Eintragungen der Pflegedaten mithilfe von Textfeldern und Autovervollständigung zu ermöglichen. Dafür sollte das Gem<sup>23</sup> `nested_form`<sup>24</sup> verwendet werden. Dies

---

<sup>23</sup> Plugin für Ruby

<sup>24</sup> (Bates, 2019)

## Fazit

---

ermöglicht die Verschachtelung von Formularen und das Hinzufügen von neuen Textfeldern über Links. Dabei wird ein Hash<sup>25</sup> erzeugt, in dem die Daten an den Controller übergeben werden.

Allerdings konnten die eingegebenen Namen nicht in der Tabelle `animals_cares` gespeichert werden. Der Grund dafür ist, dass das Gem nicht darauf ausgelegt ist, die Daten in einer Zwischentabelle zu speichern und deshalb zum Beispiel die Benennung des Hashs falsch war (`animals_attributes` anstatt `animal_care_attributes`) und dadurch nicht richtig verarbeitet werden konnte. Um diese Probleme zu beheben hätte man das Gem bearbeiten müssen, was aufwendig gewesen wäre und gute Kenntnisse in JavaScript und JQuery erfordert hätte.

Deshalb wurde von der Autorin in Rücksprache mit dem Auftraggeber beschlossen, anstatt dessen das Gem `chosen-rails`<sup>26</sup> einzusetzen. Dabei werden die Werte in einem Array übergeben und in der Zwischentabelle gespeichert, sobald eine neue Behandlung erstellt wird. Der Benutzer kann beliebig viele Tiere aus einem Drop-Down-Menü auswählen. Durch eine integrierte Suchfunktion wird die Suche erleichtert. Alle ausgewählten Tiere werden im Textfeld angezeigt, über das sie auch ganz einfach wieder entfernt werden können. Die Verwendung des Gems anstelle einer eigenen Implementierung sparte Zeit ein, wodurch der Zeitverlust durch die Fehlersuche teilweise aufgeholt werden konnte. Durch diese Änderung war es nicht mehr nötig, die Seite in zwei Bereiche<sup>27</sup> zu unterteilen, wie es für eine bessere Übersichtlichkeit vorgesehen war. Dadurch konnte das Layout mehr an die bestehenden Seiten angepasst werden. Ein Screenshot davon befindet sich in Anhang A18 (Eingabeformular für eine neue Behandlung).

### 8.2 Manuelle Tests anstelle von automatisierten Integrations-Tests

Um die korrekte Erstellung der PDF-Dateien für die Druckfunktion automatisiert zu testen, wäre ein komplexer Integrationstest nötig gewesen. Die Simulation dieses Testfalls wäre sehr aufwendig gewesen, da es im Testsystem von RoR nicht vorgesehen ist Dateien zu erstellen. Um die richtige Funktionsweise dennoch sicherzustellen, wurde die Druckfunktion manuell über die Benutzeroberfläche getestet. Dabei wurde überprüft, ob die PDF-Dateien wirklich erstellt werden, die richtigen Daten beinhalten, korrekt zusammengefügt und im Browser angezeigt werden.

### 8.3 Implementierung der automatischen Backups nach der Veröffentlichung der Webapplikation

Geplant war, die automatischen Backups am Ende der Implementierungsphase und vor der Einführung umzusetzen. Um die korrekte Erstellung der Backups zu überprüfen, war es jedoch nötig, die Webapplikation erst bereitzustellen und die Datenbank zu migrieren. Deshalb fand die Implementierung der automatischen Backups erst nach der Veröffentlichung des Tierbestandsbuchs auf Heroku statt.

## 9 Fazit

### 9.1 Soll-/Ist-Vergleich

Insgesamt kann das Projekt als erfolgreich angesehen werden, da der Auftraggeber mit dem Tierbestandsbuch zufrieden war und es abgenommen hat. Außerdem konnte die Planung eingehalten werden,

---

<sup>25</sup> Assoziatives Array, das Keys und die zugehörigen Werte enthält

<sup>26</sup> (Ho, 2019)

<sup>27</sup> Eingabe der Behandlungsdaten und Eingabe der Tiernamen

## Fazit

---

wie in Anhang A21 (Soll-/Ist-Vergleich) zu sehen ist. Alle Funktionen, die im Pflichtenheft definiert wurden, wurden umgesetzt. Auch der zeitliche Rahmen von 70 Stunden konnte eingehalten werden, obwohl es teilweise zu Abweichungen kam. Der Zeitaufwand für die Entwurfsphase fiel etwas kürzer aus. Es wurde zum Beispiel Zeit für Änderungen an den Oberflächenentwürfen eingeplant, allerdings hatte der Auftraggeber keine Änderungswünsche. Auch der Aufwand für die Datenmodellierung wurde überschätzt. Dafür fielen die Benutzer- und Entwicklerdokumentation umfangreicher aus als gedacht.

Innerhalb der Implementierungsphase gab es Verschiebungen bei wenigen Teilaufgaben, die jedoch bei anderen Implementierungsschritten wieder ausgeglichen werden konnten: Die Implementierung der Pflegeseite dauerte aufgrund der Anpassungen in Abschnitt 8.1 (Multiple-Drop-Down anstelle von Textfeldern) länger, allerdings war der Zeitaufwand für die Implementierung der automatischen Backups geringer als gedacht. Außerdem konnten Teile der Filterung und der Oberfläche der Pflegeseite für die Tierliste wiederverwendet werden, was ebenfalls den Aufwand reduzierte. Durch die Einhaltung der Anzahl an geplanten Stunden wurden auch die Termin- und Kostenziele erreicht.

Dadurch, dass der Argenhof zufrieden mit dem neuen Tierbestandsbuch ist, ist auch die gewünschte Imageverbesserung des Berufsbildungswerks sehr wahrscheinlich. Das Projekt stellt also nicht nur für den Auftraggeber einen Erfolg dar, sondern auch für das Berufsbildungswerk.

## 9.2 Lessons Learned

Bei der Durchführung des Projekts konnte die Autorin viele wertvolle Erfahrungen sammeln, vor allem in Bezug auf die Terminplanung. Diese konnte nur durch das Leisten von Überstunden eingehalten werden, da mit einer Arbeitszeit von sieben Stunden pro Tag gerechnet wurde, die jedoch aufgrund von anderweitigen Verpflichtungen (Besprechungen etc.) nicht erreicht wurden. Deshalb sollte zukünftig immer etwas Puffer eingeplant werden. Außerdem konnte die Autorin neue Erfahrungen mit der Generierung von PDF-Dateien sammeln und hat neue Sicherheitsaspekte kennengelernt, die beachtet werden müssen (zum Beispiel SQL-Injection).

Die Autorin hat auch gelernt, wie wichtig der Test einer Webapplikation in verschiedenen Browsern ist. Beim Abnahmetest hat sich nämlich herausgestellt, dass der Argenhof den Internet Explorer verwendet und dort die zusammengefügte PDF zum Ausdrucken nicht im Browser angesehen werden kann, sondern immer heruntergeladen werden muss. Für den Kunden stellt dies nach eigener Aussage kein Problem dar, sollte aber bei zukünftigen Entwicklungen beachtet werden.

Insgesamt hatte das Projekt also nicht nur einen großen Nutzen für den Argenhof und das Berufsbildungswerk, sondern auch für die Autorin, die dadurch wichtige Erfahrungen sammeln konnte.

## 9.3 Ausblick

Nach dem Ende des Projekts wurde das Tierbestandsbuch an zwei Mitauszubildende übergeben. Sie sind für die Wartung und die Weiterführung der Webapplikation verantwortlich und unterstützen die Benutzer bei Problemen.

In Zukunft soll das Tierbestandsbuch noch erweitert werden. Zum Beispiel soll es möglich sein, für Tiere ohne Namen (beispielsweise Gänse) eine Gruppe anzulegen, anstelle von einzelnen Seiten. Außerdem ist geplant, das Tierbestandsbuch auch anderen Organisationen wie Tierheimen und weiteren Gnadenhöfen zur Verfügung zu stellen, die ebenfalls Tierbestandsbücher führen müssen.



## Literaturverzeichnis

---

## Literaturverzeichnis

- Action Controller Overview.* (14. Mai 2019). Von Rails Guides: [https://guides.rubyonrails.org/action\\_controller\\_overview.html](https://guides.rubyonrails.org/action_controller_overview.html) abgerufen
- Active Record Basics.* (14. Mai 2019). Von Rails Guides: [https://guides.rubyonrails.org/active\\_record\\_basics.html](https://guides.rubyonrails.org/active_record_basics.html) abgerufen
- ActiveRecord::Scoping::Named::ClassMethods.* (16. Mai 2019). Von Ruby on Rails API: <https://api.rubyonrails.org/classes/ActiveRecord/Scoping/Named/ClassMethods.html> abgerufen
- Aufgabenfeld Bildung.* (13. Mai 2019). Von Stiftung Liebenau-Website: <https://www.stiftung-liebenau.de/bildung/ueber-uns/aufgabenfeld-bildung/#c6513> abgerufen
- Bates, R. (16. Mai 2019). *nested\_form*. Von GitHub: [https://github.com/ryanb/nested\\_form](https://github.com/ryanb/nested_form) abgerufen
- Der Argenhof.* (13. Mai 2019). Von Argenhof-Website: <https://der-argenhof.de/der-argenhof/> abgerufen
- GitHub.* (22. Mai 2019). Von CombinePDF: [https://github.com/boazsegev/combine\\_pdf](https://github.com/boazsegev/combine_pdf) abgerufen
- Ho, T.-C. (16. 5 2019). *chosen-rails*. Von GitHub: <https://github.com/tsechingho/chosen-rails> abgerufen
- Krypczyk, V., & Bochkor, O. (2018). *Handbuch für Softwareentwickler* (1. Ausg.). Bonn: Rheinwerk Verlag GmbH.
- Lahres, B., & Rayman, G. (2015). *Objektorientierte Programmierung - Das umfassende Handbuch*. Rheinwerk Computin.
- Pace, J. (22. Mai 2019). *GitHub*. Von PDFKit: <https://github.com/pdfkit/pdfkit> abgerufen
- Ruby on Rails oder php.* (15. Mai 2019). Von Rundumschlag - Rund ums Business: <http://rundumschlag.de/ruby-on-rails-oder-php/> abgerufen
- SASS: CSS auf dem nächsten Level?* (15. Mai 2019). Von Ionos Digital Guide: <https://www.ionos.de/digitalguide/websites/web-entwicklung/sass/> abgerufen
- Schätzlein, C. (15. Mai 2019). *Ruby on Rails: Eine Einführung*. Von Bader & Jene - Software-Ingenieurbüro: [https://www.bader-jene.de/wp-content/uploads/Einfuehrung\\_Ruby\\_on\\_Rails.pdf](https://www.bader-jene.de/wp-content/uploads/Einfuehrung_Ruby_on_Rails.pdf) abgerufen
- Schnatz, T. (13. Mai 2019). *Tierbestandsbuch*. Von KnoWau: <http://www.knowau.de/kwmenutabpage.php?menuTabPageID=3&current=4> abgerufen
- Sterrett, M. Z. (22. Mai 2019). *Wicked PDF*. Von GitHub: [https://github.com/mileszs/wicked\\_pdf](https://github.com/mileszs/wicked_pdf) abgerufen



## Anhang

## Anhang

## A1 Detaillierte Zeitplanung

|   |           |
|---|-----------|
| <b>Analyse</b>  | <b>3</b>  |
| 1 Wirtschaftlichkeitsanalyse und Make-or-Buy-Entscheidung                                     | 1         |
| 2 Prozessanalyse und Erstellung einer EPK   | 1         |
| 3 Termin-, Sachmittel-, Personal- und Kostenplanung   | 1         |
| <b>Entwurf</b>  | <b>13</b> |
| 1 Planen der Dialogstruktur   | 1         |
| 2 Erstellung des Oberflächenentwurfs  | 3         |
| 3 Entwurf der Datenbank mit einem ERM und einem Tabellenmodell                                | 2,5       |
| 4 Planung der Geschäftslogik und Erstellen eines Klassendiagramms                             | 1,5       |
| 5 Erstellen des Pflichtenhefts  | 5         |
| <b>Implementierung</b>  | <b>36</b> |
| 1 Erstellen der Datenbanktabellen   | 1,5       |
| 2 Implementierung und Test der Pflegedaten  | 14,5      |
| 2.1 Erstellen der Eingabeformulare zum Speichern und Ändern der Pflegedaten                   | 3,5       |
| 2.2 Implementierung und Test der Funktionen zum Speichern, Ändern und Löschen der Pflegedaten | 4         |
| 2.3 Erstellen der Oberfläche zum Anzeigen und Filtern der Pflegedaten                         | 4         |
| 2.4 Implementierung und Test der Funktionen zum Anzeigen und Filtern der Pflegedaten          | 3         |
| 3 Implementierung und Test der Tierliste  | 8         |
| 3.1 Erstellung der Oberfläche zum Anzeigen und Filtern der Tierdaten                          | 4         |
| 3.2 Implementierung und Test der Funktionen zum Anzeigen und Filtern der Tierdaten            | 4         |
| 4 Implementierung und Test der Druckfunktion  | 8,5       |
| 4.1 Erstellen des Drucklayouts  | 2         |
| 4.2 Erstellen der Oberfläche zur Auswahl der zu druckenden Seiten                             | 2,5       |
| 4.3 Implementierung und Test der Funktion zum Ausdrucken mehrerer Seiten                      | 4         |
| 5 Implementierung und Test der automatischen Backups der Datenbank                            | 3,5       |
| <b>Einführung</b>   | <b>5</b>  |
| 1 Bereitstellung der Webapplikation auf dem Cloud-Server                                      | 2         |
| 3 Schulung des Kunden inkl. An- und Abreise   | 3         |
| <b>Dokumentation</b>  | <b>13</b> |
| 1 Erstellen der Projektdokumentation  | 10        |
| 2 Erstellen der Benutzerdokumentation in Form einer Online-Hilfe                              | 2         |
| 3 Erstellen der Entwicklerdokumentation   | 1         |
| <b>Gesamt</b>   | <b>70</b> |

Tabelle 3: detaillierte Zeitplanung

## Anhang

## A2 Gantt-Diagramm

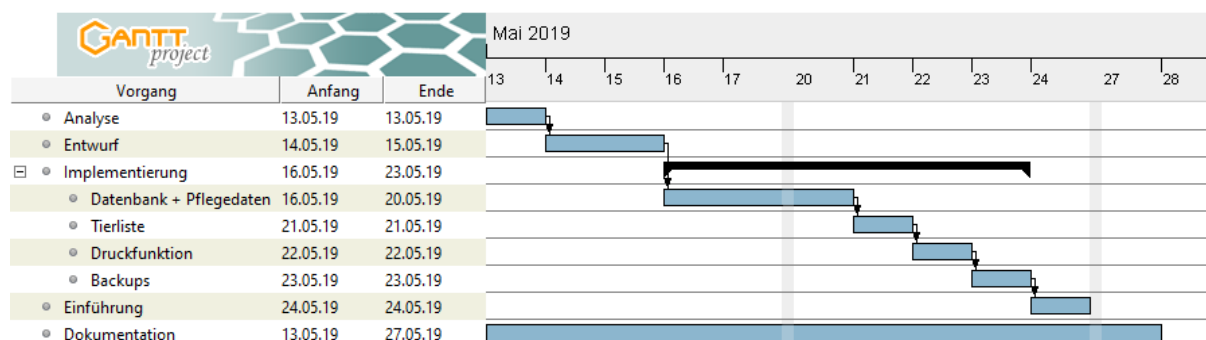


Abbildung 1: Gantt-Diagramm

## A3 Personalkosten

| Mitarbeiter                          | Stundensatz | Dauer in h | Kosten            |
|--------------------------------------|-------------|------------|-------------------|
| Projektleiterin                      | 20,00 €     | 70         | 1.400,00 €        |
| Mitauszubildender                    | 20,00 €     | 3          | 60,00 €           |
| Verwaltungsmitarbeiter des Argenhofs | - €         | 1          | - €               |
| Ausbilder                            | 40,00 €     | 1          | 40,00 €           |
| <b>Gesamt</b>                        |             |            | <b>1.500,00 €</b> |

Tabelle 4: Personalkosten

## A4 Sachmittelkosten

| Bezeichnung        | Einmalige Kosten | Anzahl | Gesamtkosten    |
|--------------------|------------------|--------|-----------------|
| MS Office 2016 Pro | 154,00 €         | 1      | 154,00 €        |
| Fujitsu Desktop-PC | 335,00 €         | 1      | 335,00 €        |
| Busticket          | 7,56 €           | 2      | 15,12 €         |
| <b>Gesamt</b>      |                  |        | <b>504,12 €</b> |

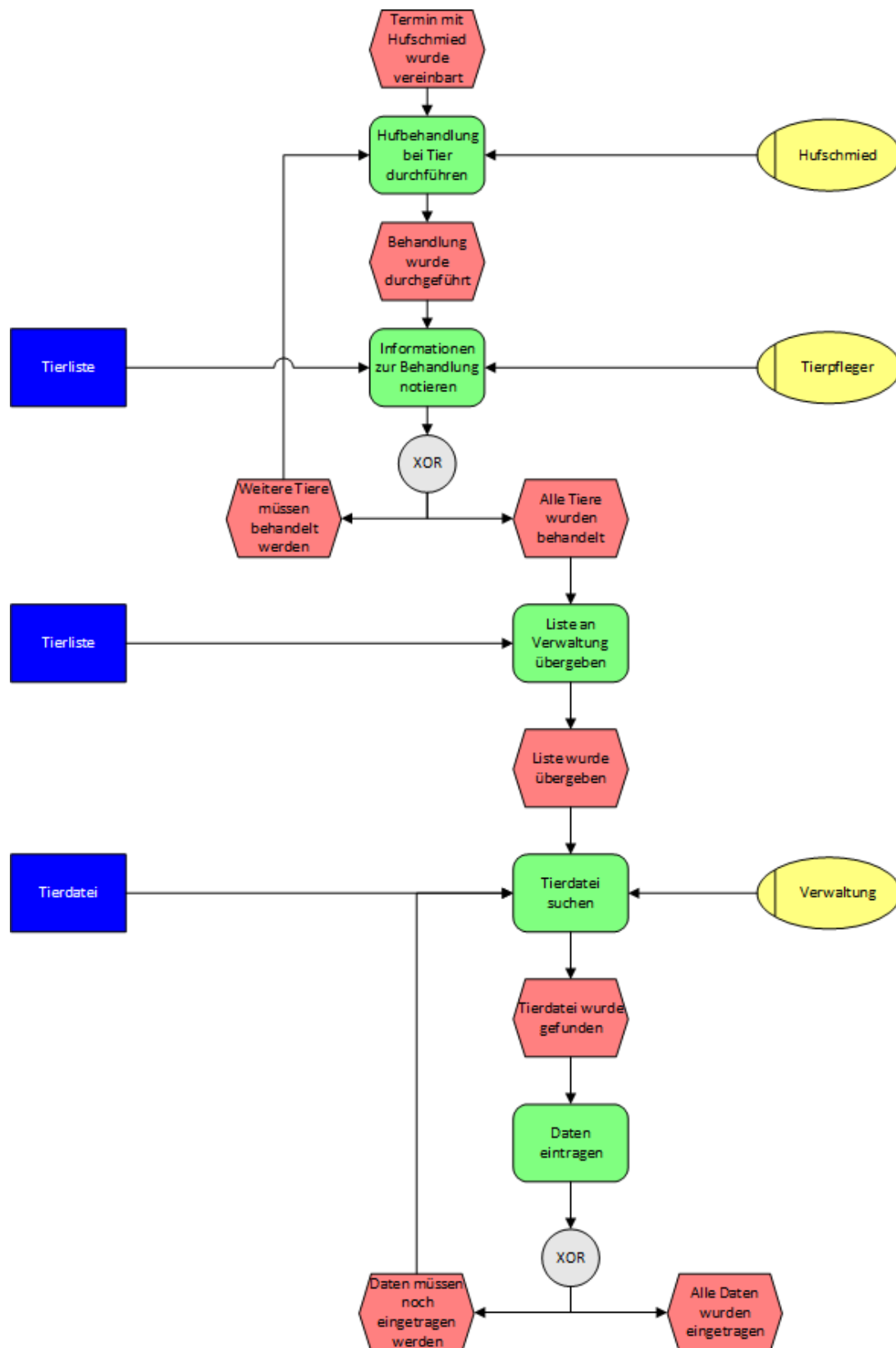
Tabelle 5: einmalige Sachmittelkosten

| Bezeichnung          | laufende Kosten / Monat | Anzahl Monate | Gesamtkosten   |
|----------------------|-------------------------|---------------|----------------|
| Leasingrate Notebook | 26,00 €                 | 1             | 26,00 €        |
| <b>Gesamt</b>        |                         |               | <b>26,00 €</b> |

Tabelle 6: laufende Sachmittelkosten

## Anhang

A5 EPK



### Abbildung 2: EPK

## Anhang

---

### A6 Lastenheft (Auszug)<sup>28</sup>

#### 1. Produktfunktionen

Um auf das System zugreifen zu können, muss es vorher entsperrt werden.

##### 1.1. Pflegedaten

Die Tiere erhalten unterschiedliche Behandlungen. Dazu gehören z.B.:

- o Impfungen,
- o Tierarztbehandlungen,
- o Zeckenschutz,
- o Behandlungen beim Hufschmied und
- o Entwurmungen.

**/LF010/** Der Benutzer kann Pflegedaten für ein Tier abspeichern.

**/LF020/** Der Benutzer kann einen Eintrag für mehrere Tiere gleichzeitig machen.

**/LF030/** Der Benutzer kann bereits gespeicherte Pflegedaten ändern und erweitern.

**/LF040/** Der Benutzer kann bereits gespeicherte Pflegedaten löschen.

**/LF050/** Der Benutzer kann sich die gespeicherten Pflegedaten anzeigen lassen.

**/LF060/** Der Benutzer kann die gespeicherten Pflegedaten nach mehreren Kriterien filtern.

Dazu gehören z.B.

- o der Name eines Tiers,
- o die Tierart,
- o die Art der Pflegebehandlung und
- o das Jahr, in dem die Behandlung stattfand.

##### 1.2. Tierliste

Die Tierliste enthält die wichtigsten Daten der Tiere übersichtlich dargestellt.

**/LF110/** Der Benutzer kann sich eine Übersicht der Tierdaten in einer Tabelle anzeigen lassen.

Zu diesen Daten gehören z.B.:

- o der Name des Tiers,
- o die Rasse,
- o die Farbe,
- o das Geburtsjahr,
- o das Geschlecht,
- o die Kennzeichnung (Chip/Tätowierung/Equidenpass oder ähnliches) und
- o das Sterbedatum.

**/LF120/** Der Benutzer kann die Daten nach einem oder mehreren Kriterien filtern. Dazu gehören z.B.:

- o die Tierart,
- o der Name eines Tiers,
- o ob die Tiere bereits vermittelt wurden und
- o ob die Tiere bereits verstorben sind.

##### 1.3. Drucken

Um das Tierbestandsbuch dem Veterinäramt vorlegen zu können, muss es ausgedruckt werden.

**/LF210/** Der Benutzer kann die Übersichtsseiten der einzelnen Tiere ausdrucken.

**/LF220/** Der Benutzer kann mehrere Übersichtsseiten auf einmal ausdrucken.

---

<sup>28</sup> Must-Anforderungen: **rot**, Should-Anforderungen: **orange**, Could-Anforderungen: **grün**

## Anhang

---

**/LF230/** Der Benutzer kann die Tierlisten ausdrucken.

**/LF240/** Beim Drucken werden nur die Daten ausgegeben und keine weiteren Komponenten.

Dazu gehören z.B.:

- o das Menü,
- o die Fußzeile und
- o die Filterung.

### 1.4. Backups

Die Datenbank wird in regelmäßigen Abständen gesichert.

**/LF310/** Von der Datenbank wird einmal täglich ein Backup erstellt.

**/LF320/** Von der Datenbank wird zusätzlich zum täglichen Backup ein wöchentliches Backup erstellt.

**/LF330/** Die Backups werden automatisch erstellt.

**/LF340/** Die Backups erfolgen außerhalb der Geschäftszeiten.

## 2. Produktdaten

Es sollen (*mindestens*) folgende Daten persistent gespeichert werden.

**/LD10/ Pflegedaten:** Alle Informationen zur Pflege:

- o Art der Behandlung (Text)
- o Datum (Datum)
- o Bemerkungen (Text)
- o Behandeltes Tier (Text)

## A7 Dialogstruktur<sup>29</sup>

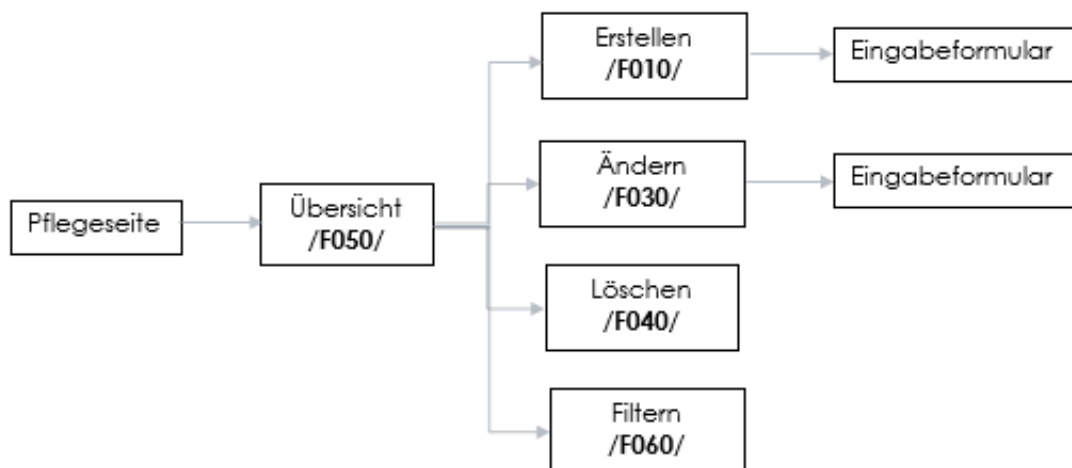


Abbildung 3: Dialogstruktur der Pflegeseite

<sup>29</sup> Die Funktionen referenzieren auf die im Pflichtenheft angegebenen Produktfunktionen in Anhang A13 (Pflichtenheft (Auszug)).

## Anhang

## A8 Mockups

TIERBESTANDSBUCH

Home Hilfe Alle Tiere Pflege Download Sperren

Drucken Neues Tier hinzufügen

Tierart auswählen ▼ Name eingeben Status auswählen ▼ Filtern

[Filter zurücksetzen](#)

| Name                   | Tierart | Rasse                 | Farbe         | Geburtsjahr | Geschlecht | Kennzeichnung | Sterbedatum |
|------------------------|---------|-----------------------|---------------|-------------|------------|---------------|-------------|
| <a href="#">Bello</a>  | Hund    | Deutscher Schäferhund | braun-schwarz | 2009        | männlich   | 01234         |             |
| <a href="#">Spirit</a> | Pferd   | Quarter Horse         | Fuchs         | 1992        | Wallach    | 56789         | 05.03.2015  |
|                        |         |                       |               |             |            |               |             |
|                        |         |                       |               |             |            |               |             |
|                        |         |                       |               |             |            |               |             |
|                        |         |                       |               |             |            |               |             |
|                        |         |                       |               |             |            |               |             |
|                        |         |                       |               |             |            |               |             |
|                        |         |                       |               |             |            |               |             |
|                        |         |                       |               |             |            |               |             |
|                        |         |                       |               |             |            |               |             |
|                        |         |                       |               |             |            |               |             |

58 x 41

Abbildung 4: Mockup der Tierliste

TIERBESTANDSBUCH

Home Hilfe Alle Tiere Pflege Download Sperren

Neue Behandlung

Behandlung:

Behandlung auswählen ▼ Datum auswählen Bemerkung eingeben

Tier(e):

Name eingeben

[weiteres Tier hinzufügen](#)

Speichern

58 x 41

Abbildung 5: Mockup des Formulars für neue Behandlungen

## Anhang

**A9 Entity-Relationship-Modell**

Hier ist ein Ausschnitt des ERM zu sehen. Der hervorgehobene Teil wurde von der Autorin zum bestehenden Modell hinzugefügt.

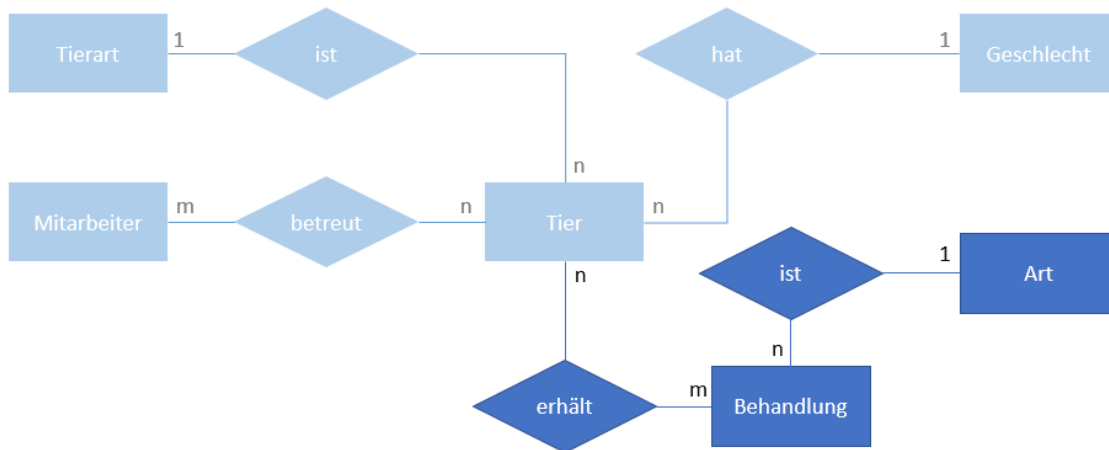


Abbildung 6: Entity-Relationship-Modell

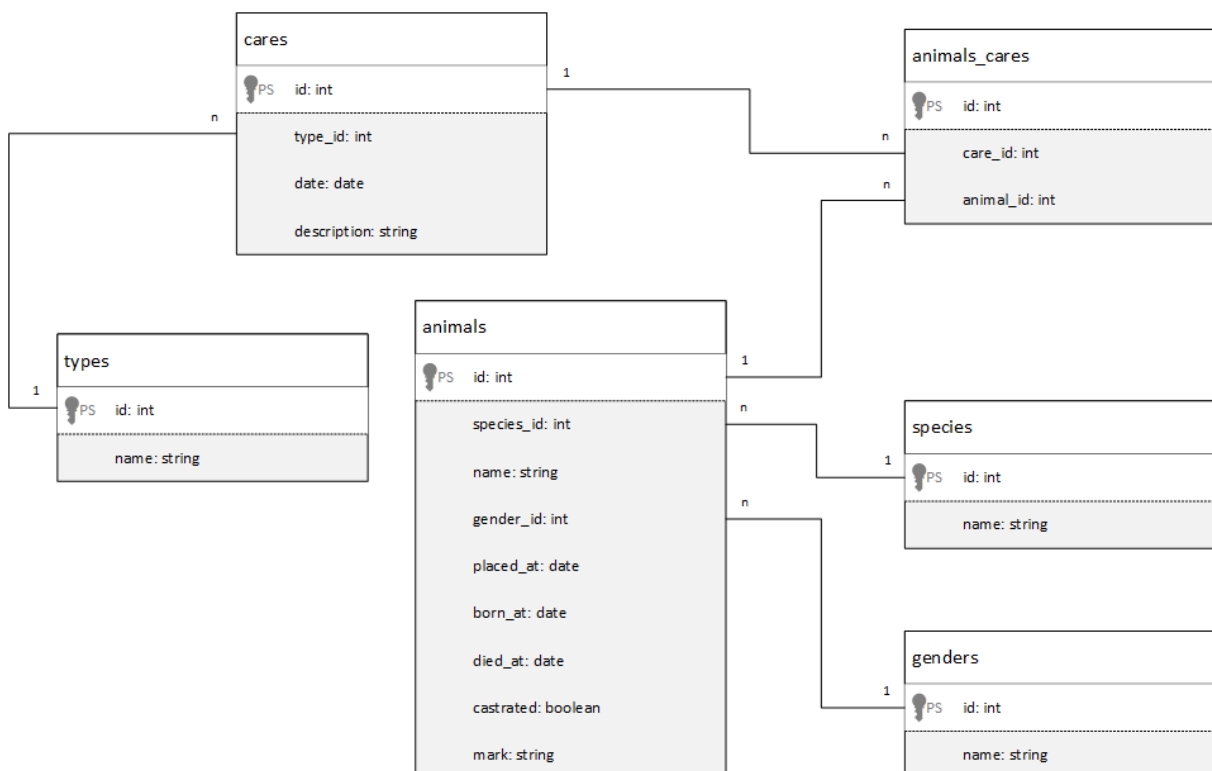
**A10 Tabellenmodell<sup>30</sup>**

Abbildung 7: Tabellenmodell

<sup>30</sup> Von der Autorin wurden die folgenden Tabellen hinzugefügt: **types**, **cares** und **animals\_cares**

## Anhang

## A11 Klassendiagramm der Controller

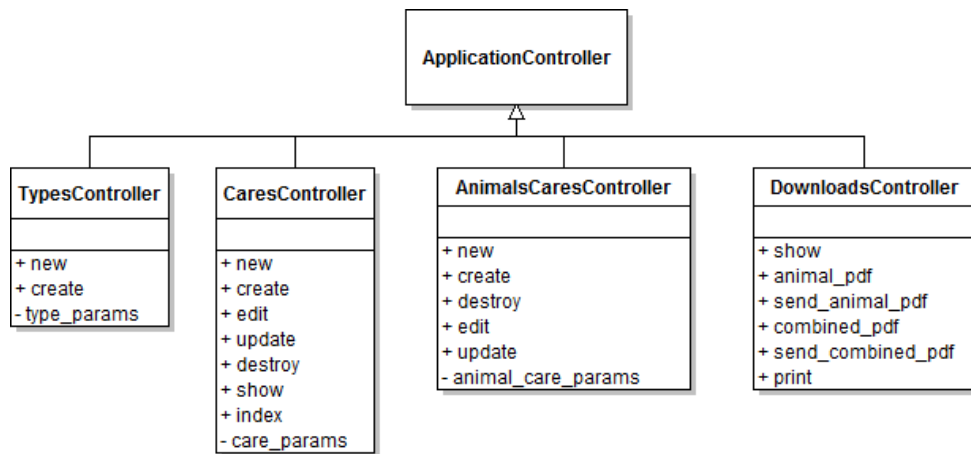


Abbildung 8: Klassendiagramm der Controller

## A12 Aktivitätsdiagramme

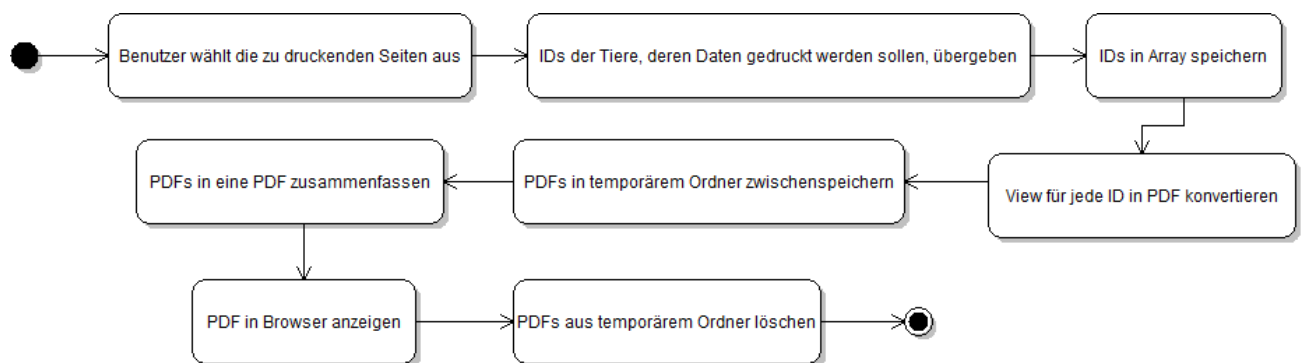


Abbildung 9: Aktivitätsdiagramm für die Druckfunktion

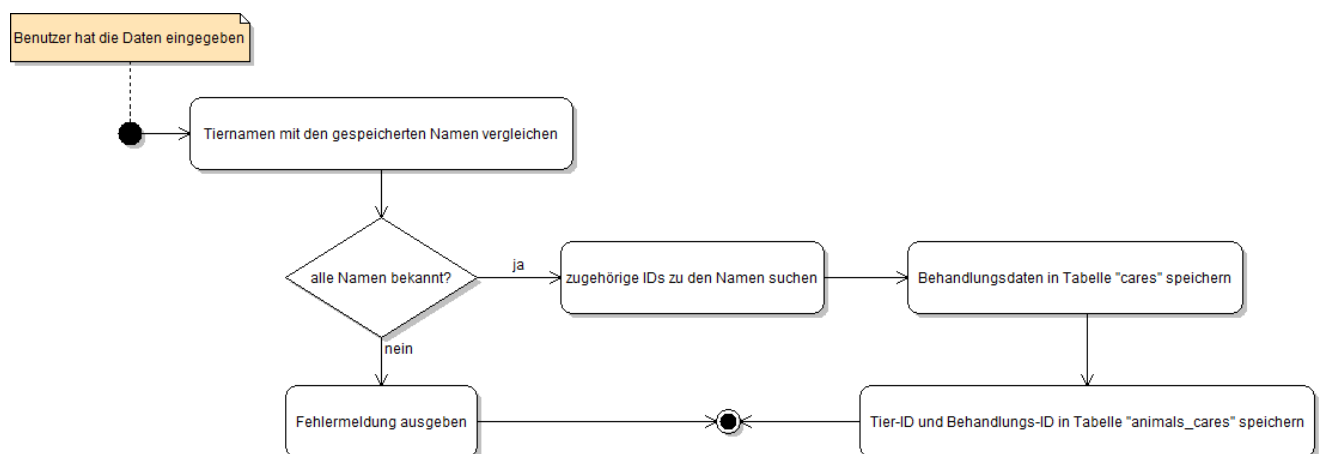


Abbildung 10: Aktivitätsdiagramm für die Speicherung einer neuen Behandlung



## Anhang

---

### A13 Pflichtenheft (Auszug)

#### Funktionen

**/F010/ Speichern der Pflegedaten:** Der Benutzer kann eine neue Pflegebehandlung für ein Tier **abspeichern**. Die Art der Behandlung wird durch ein Drop-Down-Menü ausgewählt. Zur Auswahl stehen die folgenden Arten: Hufschmied, Tierarzt, Impfung, Zeckenschutz und Entwurmung.

Die Angabe der Behandlungsart ist verpflichtend, sowie die Angabe des Datums, an dem die Behandlung stattfand. Das Datum kann über einen Kalender ausgewählt werden. Die Eingabe einer Bemerkung in einem Textfeld ist optional. Es muss mindestens ein Tier angegeben werden, für das der Eintrag gemacht werden soll. Dafür wird der Name eingetragen, wobei der Benutzer durch Autovervollständigung unterstützt wird. Vor dem Speichern muss überprüft werden, ob alle Pflichtangaben gemacht wurden und das angegebene Tier bekannt ist.

**/F020/ Speichern einer Pflegebehandlung für mehrere Tiere gleichzeitig:** Der Benutzer kann **mehrere Tiere für das Abspeichern einer Pflegebehandlung** angeben. Dafür kann er über einen Link weitere Textfelder hinzufügen und auch wieder löschen.

**/F030/ Ändern und Erweitern der Pflegedaten:** Der Benutzer kann die bestehenden Pflegedaten **ändern** und **erweitern**. Dies geschieht über ein Eingabeformular, das wie das Formular für die neuen Behandlungen aufgebaut ist. Die bestehenden Daten sind bereits voreingetragen.

**/F040/ Löschen der Pflegedaten:** Der Benutzer kann die bestehenden Pflegedaten **löschen**. Dabei wird unterschieden zwischen dem Löschen der Behandlung für ein einzelnes Tier und dem Löschen der jeweiligen Behandlung bei allen betroffenen Tieren.

Beim Löschen für ein einzelnes Tier, wird der betroffene Datensatz aus der Tierbehandlungstabelle (siehe /D010/) entfernt.

Beim Löschen einer kompletten Behandlung wird der betroffene Datensatz aus der Behandlungstabelle (siehe /D030/) entfernt und gleichzeitig alle damit verknüpften Datensätze aus der Tierbehandlungstabelle (CASCADE). Wird ein Tier gelöscht, werden alle Datensätze aus der Tierbehandlungstabelle mitgelöscht, die mit dem jeweiligen Tier verknüpft sind. In der Behandlungstabelle bleiben sie jedoch erhalten.

**/F050/ Anzeige der Pflegedaten:** Der Benutzer kann sich die bereits gespeicherten Pflegedaten auf zwei Wegen **anzeigen** lassen. Zum einen werden sie in den Übersichten des jeweiligen Tiers angezeigt. Außerdem gibt es eine Übersicht in Tabellenform, in denen die Pflegedaten für alle Tiere angezeigt werden können.

**/F060/ Filterung der Pflegedaten:** Der Benutzer kann die Pflegedaten in der Tabelle nach einem oder mehreren Kriterien **filtern**. Dazu gehören: Tiername, Tierart, Art der Pflegebehandlung und Jahr.

Standardmäßig werden zunächst alle vorhandenen Daten angezeigt. Über Drop-Down-Menüs kann der Benutzer die gewünschte Tierart und die Art der Pflegebehandlung auswählen. Über ein Textfeld kann nach einem bestimmten Namen gefiltert werden und über einen Kalender kann das Jahr ausgewählt werden. Die Filterung kann über einen Link wieder zurückgesetzt werden.

## Anhang

---

### Globale Testszenarien und Testfälle

*/T010/ Speichern der Pflegedaten:* Max Mustermann erstellt die folgenden Behandlungen:

- **Impfung** am **01.01.2019** gegen **Tollwut** für **Bello**
- **Hufschmied** am **02.02.2019** für **Spirit**

*/T020/ Speichern einer Pflegebehandlung für mehrere Tiere gleichzeitig:* Max Mustermann erstellt die folgende Behandlung:

- **Zeckenschutz** am **03.03.2019** mit **Frontline** für **Bello, Rex** und **Lessie**

*/T030/ Ändern und Erweitern der Pflegedaten:* Max Mustermann ändert die letztgenannte Behandlung folgendermaßen ab:

- Zeckenschutz am 03.03.2019 mit **Zeckenhalsband** für Bello, Rex und Lessie

*/T040/ Löschen der Pflegedaten:* Max Mustermann löscht die folgenden Behandlungen:

- Zeckenschutz am 03.03.2019 mit Zeckenhalsband für **Rex**
- Zeckenschutz am 03.03.2019 mit Zeckenhalsband für **alle**

*/T050/ Anzeige der Pflegedaten:* Max Mustermann lässt sich alle Pflegedaten anzeigen.

*/T060/ Filterung der Pflegedaten:* Max Mustermann filtert die Pflegedaten nach den folgenden Kriterien:

- **Impfungen** im Jahr **2019**
- **Zeckenschutz** für **Hunde**

### A14 Filterung der Tierliste nach dem Status

```
def self.status_like(status)
  query = ""
  i = 0
  if status.include?("verstorben")
    query += "died_at IS NOT NULL"
    i += 1
  end
  if status.include?("vermittelt")
    if i > 0
      query += " OR "
    end
    query += "placed_at IS NOT NULL"
    i += 1
  end
  if status.include?("lebend")
    if i > 0
      query += " OR "
    end
  end
end
```

## Anhang

---

```

    end
    query += "placed_at IS NULL AND died_at IS NULL"
  end
  where(query)
end

```

Listing 2: Methode zur Filterung des Status

### A15 Test für die Filterung nach dem Status

```

class AnimalTest < ActiveSupport::TestCase
  test "class method :status_like" do
    assert Animal.respond_to?(:status_like)

    status = ["verstorben"]
    expected = [@verstorben]
    assert_equal expected, Animal.status_like(status).to_a

    status = ["vermittelt", "lebend"]
    expected = [@vermittelt, @lebend]
    assert_equal expected, Animal.status_like(status).to_a

    status = ["verstorben", "vermittelt", "lebend"]
    expected = [@verstorben, @vermittelt, @lebend]
    assert_equal expected, Animal.status_like(status).to_a
  end
end

```

Listing 3: Test zur Filterung des Status

```

foo@learnenough:~/environment/tierbestandsbuch$ ruby -I test test/models/animal_test.rb -n test_class_
method_status_like
Started with run options -n test_class_method_status_like --seed 186

 1/1: [=====] 100% Time: 00:00:00, Time: 00:00:00
Finished in 0.40371s
1 tests, 8 assertions, 0 failures, 0 errors, 0 skips

```

Abbildung 11: Screenshot der Testdurchführung

## Anhang

## A16 Ausdrucken mehrerer Tierseiten

```
class DownloadsController < ApplicationController
  def combined_pdf
    animals = Animal.animal_id(params[:animal_ids])
    pdf = CombinePDF.new
    animals.each do |a|
      page = AnimalPdf.new(a).to_pdf
      pdf << CombinePDF.load("#{a.name}.pdf")
      pdf.save "combined.pdf"
    end
    redirect_to print_pdf_path
  end
end
```

Listing 4: Methode zum Erstellen der PDF

## A17 Test der automatisierten Backups

## MANUAL BACKUPS &amp; DATA EXPORTS

[Learn more](#) about data exports and backups.

| Name | Size    | Database | Created     | Finished    | Create Manual Backup |
|------|---------|----------|-------------|-------------|----------------------|
| a002 | 29.1 KB | DATABASE | 7 hours ago | 7 hours ago | Download Delete      |
| a001 | 29.1 KB | DATABASE | a day ago   | a day ago   | Download Delete      |

Abbildung 12: Screenshot der erstellten Backups

```
foo@learnenough:~/environment/tierbestandsbuch$ heroku pg:backups:restore a002 DATABASE_URL --app tierbestandsbuch
WARNING: Destructive Action
This command will affect the app tierbestandsbuch
To proceed, type tierbestandsbuch or re-run this command with
--confirm tierbestandsbuch

> tierbestandsbuch
Starting restore of a002 to postgresql-flat-10976... done

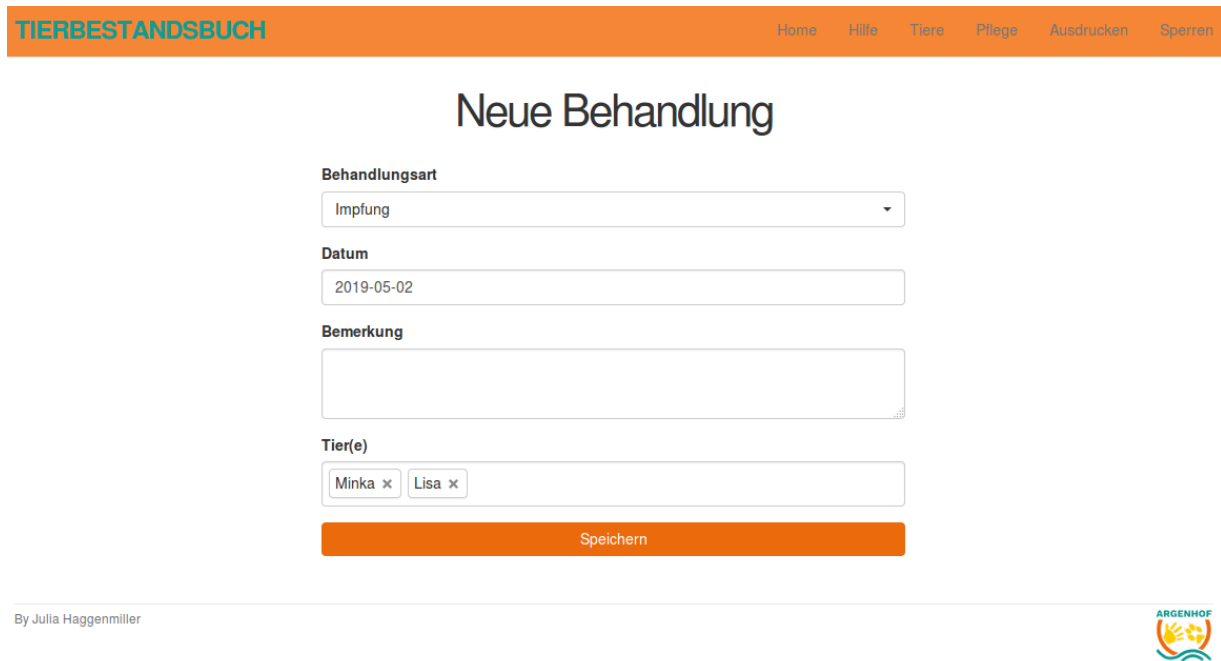
Use Ctrl-C at any time to stop monitoring progress; the backup will continue restoring.
Use heroku pg:backups to check progress.
Stop a running restore with heroku pg:backups:cancel.

Restoring... done
foo@learnenough:~/environment/tierbestandsbuch$
```

Abbildung 13: Screenshot des Restores

## Anhang

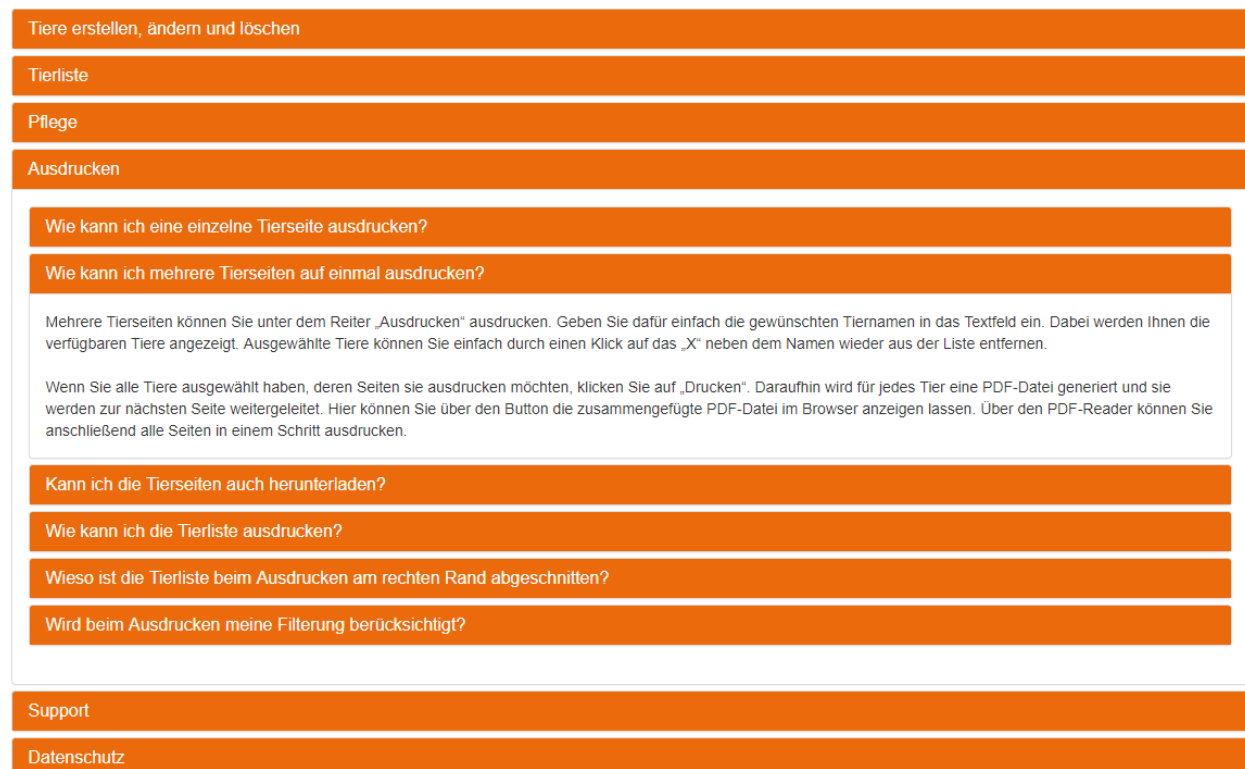
## A18 Eingabeformular für eine neue Behandlung



The screenshot shows the 'Neue Behandlung' (New Treatment) form. At the top, there is a navigation bar with the title 'TIERBESTANDSBUCH' and links for 'Home', 'Hilfe', 'Tiere', 'Pflege', 'Ausdrucken', and 'Sperrern'. The main heading is 'Neue Behandlung'. Below this, there are several input fields: 'Behandlungsart' (Treatment type) with a dropdown menu showing 'Impfung'; 'Datum' (Date) with a text box showing '2019-05-02'; 'Bemerkung' (Remarks) with a large text area; and 'Tier(e)' (Animal(s)) with a list of animals 'Minka' and 'Lisa', each with a close button (x). At the bottom of the form is an orange button labeled 'Speichern' (Save). In the footer, it says 'By Julia Haggenmiller' on the left and the 'ARGENHOF' logo on the right.

Abbildung 14: Screenshot des Eingabeformulars

## A19 Benutzerdokumentation (Auszug)



The screenshot shows a sidebar menu on the left with orange buttons for 'Tiere erstellen, ändern und löschen', 'Tierliste', 'Pflege', 'Ausdrucken', 'Support', and 'Datenschutz'. The main content area is titled 'Ausdrucken' and contains several sections with orange headers and white text boxes. The sections are: 'Wie kann ich eine einzelne Tierseite ausdrucken?', 'Wie kann ich mehrere Tierseiten auf einmal ausdrucken?' (which includes a paragraph explaining how to print multiple pages by selecting animals and clicking 'Drucken'), 'Kann ich die Tierseiten auch herunterladen?', 'Wie kann ich die Tierliste ausdrucken?', 'Wieso ist die Tierliste beim Ausdrucken am rechten Rand abgeschnitten?', and 'Wird beim Ausdrucken meine Filterung berücksichtigt?'.

Abbildung 15: Screenshot der Online-Hilfe

## Anhang

---

### A20 Entwicklerdokumentation (Auszug)

#### Entwicklungsumgebung:

Für die Weiterentwicklung steht ein Desktop-PC (Kennzeichnung "PC-FI19") zur Verfügung. Darauf ist eine virtuelle Maschine (LearnEnough VM v.1.4) mit der Entwicklungsumgebung installiert. Der Source Code befindet sich im Ordner "environment". Um den Code auszuführen, muss über die Konsole in den Ordner "tierbestandsbuch" gewechselt und der installierte Webserver gestartet werden (Befehl: "rails server"). Die Applikation ist dann im Browser unter der Adresse "localhost:3000" erreichbar.

#### Test-Datenbank:

Für die Entwicklung und zu Testzwecken wird eine SQLite-Datenbank benutzt. Diese ist auf der virtuellen Maschine bereits installiert. Der Zugriff erfolgt über SQLite Manager. SQLite-Manager kann als Add-On bei Firefox installiert werden (nur bei den älteren Versionen!). Damit können die Datenbanken im Ordner "db" geöffnet werden. Achtung: Alle Dateien, nicht nur ".sqlite"-Dateien anzeigen lassen!

#### Produktiv-Datenbank:

In der Produktivumgebung wird eine PostgreSQL-Datenbank verwendet, die von Heroku bereitgestellt wird. Der Zugriff darauf erfolgt über pgAdmin. pgAdmin ist bereits auf dem Projekt-Rechner installiert. Achtung: Bei Verwendung des Proxys ist kein Verbindungsaufbau möglich, da AWS blockiert wird! Die Informationen zur Verbindungsherstellung befinden sich auf der Website von Heroku.

#### Deployment:

Um die Webapplikation zu veröffentlichen, muss sie zuerst bei Bitbucket hochgeladen werden. Anschließend wird der Code über das Git-Repository bei Heroku hochgeladen. Bei Bedarf muss die Produktivdatenbank migriert werden.

### A21 Soll-/Ist-Vergleich

| Kriterium  | Soll       | Ist        | Abweichung |
|------------|------------|------------|------------|
| Funktionen | 16 Stück   | 16 Stück   | 0 Stück    |
| Zeit       | 70 Stunden | 70 Stunden | 0 Stunden  |
| Termin     | 27.05.2019 | 27.05.2019 | 0 Tage     |
| Kosten     | 2.030,12 € | 2.030,12 € | 0 €        |

Tabelle 7: Soll-/Ist-Vergleich

| Projektphase    | geplante Zeit in h | tatsächliche Zeit in h | Differenz |
|-----------------|--------------------|------------------------|-----------|
| Analyse         | 3                  | 3,5                    | + 0,5     |
| Entwurf         | 13                 | 10,5                   | - 2,5     |
| Implementierung | 36                 | 36                     | 0         |
| Einführung      | 5                  | 5                      | 0         |
| Dokumentation   | 13                 | 15                     | + 2       |
| <b>Gesamt</b>   | <b>70</b>          | <b>70</b>              |           |

Tabelle 8: Zeitplanung

## Anhang

## A22 Abnahmeprotokoll



# Abnahmeprotokoll

**Abnahmegegenstand:**

Tierbestandsbuch Version 2.0 vom 24.05.2019

**Abnahmebeteiligte:**Auftraggeber:

Argenhof - Lebenswürde für Tiere e.V.  
Argenhof  
88279 Amtzell

Projektleiter:

Julia Haggenmiller  
Liebenau Berufsbildungswerk gemeinnützige GmbH  
Außenstelle Lok-Center  
Pfannenstiel 16  
88212 Ravensburg

**Abnahmeergebnis:**

- ☒ Das Abnahmeobjekt wurde geprüft ohne Mängelanzeige. Die Abnahme erfolgt ohne Vorbehalt.
- ☐ Der Abnahmegegenstand wurde geprüft und unter Vorbehalt abgenommen. Die Mängel sind innerhalb der vorgegebenen Frist zu beheben und die Lösung ist mittels Nachprüfung nochmals abzunehmen.
- ☐ Der Abnahmegegenstand wurde geprüft. Die Abnahme wird verweigert.

24.05.2019

Ort, Datum

Gnadenhof - Lebenswürde für Tiere e.V.  
Argenhof  
88279 Amtzell  
Info: 07522 707967  
www.der-gnadenhof.de

Unterschrift Auftraggeber

24.5.19

Ort, Datum

J. Haggenmiller

Unterschrift Projektleiter

Abbildung 16: Abnahmeprotokoll