

Feedback 1: 61/80 (~50 Stunden)

Userstories (erreicht): 68

Techstories (Abzüge): -7

Qualitätsmanager/in: -4

Usability Engineer: -3

Technischer Architekt/in: -0

Datenmanager/in: -0

US 9 [-12] Stammbaum nicht implementiert

TS 10.2 [-1] TRACE logging fehlt bei ein paar private methoden in DAO

TS 16 [-3] In Entities nur ueber ID (Long) auf Eltern/sport verwiesen statt mit Entities.

TS 18 [-3] Suchparameter koennen nicht reseted werden (ausser durch manuelles loeschen)

Feedback 2: 77/80 (~90 Stunden)

Userstories (erreicht): 80

Techstories (Abzüge): -3

Qualitätsmanager/in: -3

Usability Engineer: -0

Technischer Architekt/in: -0

Datenmanager/in: -0

Userstories Sehr gute Arbeit! Validierung, Exceptionhandling und Usability sind ausgezeichnet

TS 11 [-3] Javadoc: Beschreibung von Interface-Klassen mehrfach nicht vorhanden

Feedback 3: 69/80 (55 Stunden)

Userstories (erreicht): 80

Techstories (Abzüge): -11

Qualitätsmanager/in: -6

Usability Engineer: -0

Technischer Architekt/in: -5

Datenmanager/in: -0

TS 11 [-3] Beschreibung der Interfaces fehlt

TS 16 [-3] verwendung von ids in entity objekt statt auf anderes entity objekt zu verweisen

TS TS24 [-5] alle deine Exceptions sind RuntimeExceptions werden aber explizit via throws angegeben, hier vermischt du zwei Ansätze der Exceptionbehandlung

Feedback 4: 75/80 (140 Stunden)

Userstories (erreicht): 80

Techstories (Abzüge): -5

Qualitätsmanager/in: -0

Usability Engineer: -2

Technische/r Architekt/in: -0

Datenmanager/in: -3

TS 19 [-2]: "Error! Intended update was invalid" statt sprechender Fehlermeldung bei Aenderung eines Pferdes

TS 26 [-3]: Keine Foreign Keys fuer Eltern im Schema (insertData wird ohne datagen nicht ausgefuehrt)

Feedback 5: 71/80 (52 Stunden)

Userstories (erreicht): 80

Techstories (Abzüge): -9

Qualitätsmanager/in: -8

Usability Engineer: -1

Technischer Architekt/in: -0

Datenmanager/in: -0

TS 10 [-3] falsche Loglevel-Konfiguration, nur root:INFO spezifiziert und nicht at:ac:tuwien:sepm:DEBUG

TS 14 [-1] 1x Commitmessage hat keine Zugehörigkeit zur Story

TS 15 [-1] Pferd kann eigenes Elternteil

TS 16 [-3] In den Entities sollten andere Entities über Entity-Objekte verweisen, statt rein über ID (in den Dtos würde dies ausreichend sein, wenn dementsprechend validiert wird)

TS 18 [-1] kein Datepicker bei Geburtsdatum vom Pferd

Feedback 6: 41/80 (67 Stunden)

Userstories (erreicht): 79

Techstories (Abzüge): -38

Qualitätsmanager/in: -13

Usability Engineer: -0

Technischer Architekt/in: -18

Datenmanager/in: -7

US 1 [-1] Name kann aus nur Leerzeichen bestehen

TS 10 [-3] Loglevel wurde nicht richtig konfiguriert (kein DEBUG für sepm folder)

TS 11 [-3] Interfaces selbst werden nicht mit Javadoc documentiert (nur die Methoden darin)

TS 14 [-1] DB wurde einmal commited (aber später wieder gelöscht)

TS 15 [-3] Name wird nicht auf Blank (nur leerzeichen) validiert Es wird nicht überprüft ob Geburtsdatum vor Kindern liegt Änderung von Pferdegeschlecht kann dazu führen das ein Pferd zwei Väter oder Mütter hat (bzw. Vater wird weiblich, Mutter wird männlich)

TS 16 [-3] Auf andere Entities wird (in Entities) nur mit ID verwiesen

TS 22 [-9] Validator wird vom Endpoint aus direkt aufgerufen sollte aber nur vom Service aufgerufen werden Suchparameter bei Pferdesuche sollten in einem DTO zusammengefasst werden

TS 24 [-5] Checked und Unchecked Ansatz für Exception Behandlung wurden vermischt, Die Exceptions wurden behandelt wie Checked Exceptions sind aber von RuntimeException abgeleitet

TS 25 [-4] Parameter für die Suche wurden nicht als QueryParameter sondern als teil der URL übergeben Es wurde an einigen Stellen Code 422 benutzt wo 409 erartet wurde (Fehler bei ändern von Geburtstagen aufgrund von alterseinschränkungen oder Fehler die Elternteile betreffen)

TS 28 [-7] Bei der Suche nach Pferden ist es durch das konkatinieren von Parametern direkt in das SQL Statement möglich eine SQL injektion durchzuführen

Feedback 7: 64/80 (64 Stunden)

Userstories (erreicht): 77

Techstories (Abzüge): -13

Qualitätsmanager/in: -7

Usability Engineer: -4

Technischer Architekt/in: -2

Datenmanager/in: -0

US 9 [-3] bei Generationenanzahl=1 kommt das Pferd + die Elterngeneration -> es sollte nur das Pferd angezeigt werden

TS 10 ** [-1] TRACE Log-Level fehlt bei HorseJdbcDao mapRow() und mapRowAll()

TS 11 ** [-3] In keinem Interface wurde dessen Zweck angegeben

TS 14 ** [-3] .idea/ und individual_assignment.iml wurden committed

TS 19 ** [-4] Keine Fehlermeldungen in Suche, das Speichern von Sportarten hat keine vernünftigen Fehlermeldungen

TS 25 ** [-2] der Endpoint zur Suche von Pferden heißt "/horses/search". REST definiert allerdings die Richtlinie, dass keine Verben verwendet werden sollen. Die Aktion soll rein über die HTTP-Methode bestimmt werden

Feedback 8: 49/80 (~40 Stunden)

Userstories (erreicht): 56

Techstories (Abzüge): -7

Qualitätsmanager/in: -5

Usability Engineer: -2

Technischer Architekt/in: -0

Datenmanager/in: -0

US 6 [-12] US funktioniert nicht

US 9 [-12] US nicht implementiert

TS 15 [-1] Änderung des Geschlechts wenn Pferd Kind hat möglich

TS 16 [-4] In den Entities sollten andere Entities über Entity-Objekte verweisen, statt rein über ID (in den Dtos würde dies ausreichend sein, wenn dementsprechend validiert wird),

Backend (Dto): Datumstyp für Geburtsdatum sollte am besten z.B. LocalDate und nicht String sein (so wie in den Entities gemacht)

TS 19 [-2] beim Versuch eine Sportart mit leerem Namen und/oder Description wird keine Fehlermeldung ausgegeben, sondern zurück zur Sportsliste navigiert

Feedback 9: 48/80 (~110 Stunden)

Userstories (erreicht): 70

Techstories (Abzüge): -22

Qualitätsmanager/in: -7

Usability Engineer: -2

Technischer Architekt/in: -10

Datenmanager/in: -3

US 2 [-4] Wird ein Horse bearbeitet und die Sportart ist nicht gesetzt, so scheitert das Bearbeiten. Eltern werden nicht pre-selected und gehen beim Klick auf speichern verloren sofern sind nicht zuvor manuell gesetzt wurden.

US 3 [-2] Wird ein Horse von der Übersicht aus gelöscht, so bleibt das Horse in der Liste. Zwar mit einem Vermerk dass es gelöscht wurde, jedoch können die entsprechenden operationen weiterhin ausgewählt werden.

US 5 [-2] Bei sämtlichen Delete-Operationen verbleibt das Horse in der View und wird erst beim reload entfernt.

US 6 [-2] Suche nach Namen ist case-sensitive. Suche nach Beschreibung ist case-sensitive

TS 10 [-3] Für die Anwendung soll DEBUG als minimales Log-Level gesetzt sein.

TS 14 [-3] Datenbank-Files sollen nicht committed werden.

TS 16 [-1] Entities sollen auf Objekte verweisen nicht auf IDs

TS 19 [-2] SQL-Exception wird angezeigt wenn keine Sportart beim Bearbeiten eines Horses gesetzt wird.

TS 22 [-5] Es werden in unteren Schichten Exceptions geworfen, die HTTP-spezifische Annotationen haben

TS 23 [-5] Es wird bei jedem Key-Event bei der Datumseingabe ein Request für die in Frage kommenden Eltern abgesetzt. Das sind viele unnötige-Requests.

TS 26 [-3] Testdaten entsprechen teilweise nicht den Validierungen (Mutter/Vater zu jung) und Baumansicht nur schwer testbar aufgrund weniger Beziehungen

Feedback 10: 70/80 (~100 Stunden)

Userstories (erreicht): 80

Techstories (Abzüge): -10

Qualitätsmanager/in: -1

Usability Engineer: -1

Technischer Architekt/in: -8

Datenmanager/in: 0

TS 12 [-1] einige Einträge in der Stundenliste keine Zugehörigkeit zu Stories

TS 18 [-1] kein Datepicker für Geburtsdatum beim Horse anlegen

TS 22 [-4] REST-Endpoints sollten DTOs zum Zusammenfassen von Query Parametern verwenden

TS 25 [-4] wenn Name von Pferd/Sportart zu lang wird Status Code 400 statt 422 verwendet

Feedback 11: 61/80(~60 Stunden)

Userstories (erreicht): 80

Techstories (Abzüge): -19

Qualitätsmanagerin: -8

Usability Engineer: 0

Technische/r Architekt/in: -11

TS 10 [-3] falsche Loglevel-Konfiguration, nur root:INFO spezifiziert und nicht at:ac:tuwien:sepm: DEBUG

TS 15 [-1] Name der Sportart ist nicht in der Länge beschränkt

TS 22 [-4] REST-Endpoints sollten DTOs zum Zusammenfassen von Query Parametern verwenden

TS 16 [-4] In den Entities sollten andere Entities über Entity-Objekte verweisen, statt rein über ID (in den Dtos würde dies ausreichend sein, wenn dementsprechend validiert wird) ** Backend: Datumstyp für Geburtsdatum sollte am besten z.B. LocalDate und nicht Date sein.

TS 22 [-4] REST-Endpoints sollten DTOs zum Zusammenfassen von Query Parametern verwenden

TS 25 [-7] ** URIs sollen keine Verben enthalten. Die Aktion soll rein über die HTTP-Methode bestimmt werden. zB ist /horses/create oder /sports/create verboten ** falscher Status Code der richtigen Klasse verwendet (400 statt 422)

Feedback 12: 59/80 (~110 Stunden)

Userstories (erreicht): 77

Techstories (Abzüge): -18

Qualitätsmanager/in: -10

Usability Engineer: 0

Technischer Architekt/in: -8

Datenmanager/in: 0

US 9. ** [-3] Es gibt einen Button im Tree für die Detailansicht der Pferde, dieser funktioniert jedoch nicht und die Detailseiten können vom Tree aus nicht aufgerufen werden.

TS 10 ** [-3] falsche Loglevel-Konfiguration

TS 11 ** [-3] Kein einziges Interfaces wurde deklariert.

TS 15 ** [-1] Beim erstellen kann das Geburtstag des Kindes vor dem der Eltern liegen.

TS 16 ** [-3] In Entities wurden IDs anstatt Entity-Objekten für die Eltern und Sport verwendet.

TS 25 [-8]

** Die URI /horses/search bricht die Konvention: URIs enthalten keine Verben. Die Aktion soll rein über die HTTP-Methode bestimmt werden.

** Wenn ein Pferd einen Validierungsconflict aufweist wegen einem anderen Pferd wird ein 422er anstatt eines 409ers geworfen. 409 Conflict für Validierungsfehler, wenn sie die Beziehung zu einer anderen Entity betreffen, also einen Konflikt zum Zustand des Systems.

** Suche die kein Ergebnis zurückgibt liefert einen 404 anstatt einer leeren Liste.

Feedback 13: 29/80 (~70 Stunden) (belastend)

Userstories (erreicht): 77

Techstories (Abzüge): -48

Qualitätsmanager/in: -20

Usability Engineer: -8

Technischer Architekt/in: -20

Datenmanager/in: 0

US 5 **[-1] Wenn ein Pferd sich selbst als Mutter oder Vater hat, kann man ihr per link folgen, dann gibt es aber keinen Elternteil mehr, wenn jedoch dann aktualisiert wird erscheint der Link wieder. Inkonsistentes Verhalten.

US 6 ** [-2] Wenn ein Pferd angelegt wird, dann auf details und dort auf loeschen wird man auf die Liste zurueck gerouted und dort ist dann immer noch das Pferd zu sehen, wenn man dieses dann bearbeitet kommt es zu internal Server Errors und seltsamen verhalten der Applikation.

TS 10 [-4] ** Logger falsch konfiguriert ** kein Logging in private Methoden

TS 11 ** [-3] Interfaces selbst nicht dokumentiert * TS 12 ** [-2] Gesamtsumme in Stundenliste fehlt

TS 13 ** [-1] kein Positivtest bei den Endpoints

TS 14 [-5] ** Zugehörigkeit zur Story fehlt bei manchen Commits ** öfters Dateien committed, die nicht committed gehören z.B. db

TS 15 [-9] ** Validierung unvollständig ** Validierung von Abhängigkeiten unvollständig ** zu langer Eingabestring führt zu DB Fehler

TS 16 [-7] ** kein ENUM für Geschlecht ** auf andere Entities mit IDs verwiesen ** java.sql.Date verwendet

TS 17 [-5] ** zu viel Funktionalität bei den Horses ** Horse anlegen unintuitiv ** nicht verwendet für Eingaben

TS 18 ** [-1] kein Drop-Down für Sport bei Horse anlegen

TS 19 ** [-2] keine sinnvolle Fehlermeldung wenn Eingabestring zu lang

TS 22 ** [-4] einzelne RequestParams statt RequestBody bei Suche

TS 24 [-11] ** NullPointerException bei Horse mit alles null ** Endlosschleife, wenn zyklische Abhängigkeiten bei Vater oder Mutter ** explizite Angabe von throws bei unchecked Exception

TS 25 [-8] ** /tree{id} nicht REST konform ** 422 statt 409 an entsprechenden Stellen ** 404 statt leerer Liste bei Suche

Feedback 14: 40/80 (~35 Stunden)

Userstories (erreicht): 73

Techstories (Abzüge): -33

Qualitätsmanager/in: -20 (-28 ohne Cap)

Usability Engineer: -2

Technischer Architekt/in: -11

Datenmanager/in: 0

Anmerkung Pferde & Sport werden über ID referenziert in der Horse Entity

US 1 [-1] Leerer name (nur Leerzeichen) für pferd wird gespeichert

US 6 [-6]

Suche nicht case insensitive

Suche nach vollem Namen (groß/klein) liefert kein Ergebnis

Suche nach teil eines Namens (groß/klein) liefert kein Ergebnis

Suche nach voller Beschreibung (groß/klein) liefert kein Ergebnis

Suche nach teil der Beschreibung (groß/klein) liefert kein Ergebnis

Suche nach Geburtsdatum liefert umgekehrtes Ergebnis (Suche nach 02.04.2021 liefert bspw. Pferd mit Datum 03.04.2021 aber nicht 01.01.2013)

TS 10 [-12]

Loglevel konfiguration nicht wie gefordert (debug für das Paket in dem die Anwendung liegt)
mapRow methode hat in den DAO Klassen kein logging
FamilyTreeServiceImpl hat einen Aufruf von "System.out.println"; printStackTrace in allen Endpoints im catch (bzw. in writeErrorToLog-Methode)

TS 13 [-3] assert statt assertEquals/assertNotEquals/assertNull/etc.

TS 15 [-13]

Ändern des Geburtsdatums eines Elternteils auf "nach Kind" setzt empty value in dropdown (behält referenz aber bei ohne anzeige, es kann noch zum vater navigiert werden)
Selbiges gilt für Änderung des Geschlechts (siehe Ändern des Geburtsdatums)
Geburtsdatum in der Zukunft wird gespeichert
Leerer Name für sport wird gespeichert; Allgemein bei Strings nur Überprüfung auf == null und Längenbeschränkung aber nicht auf leere Strings
Gender wird nur auf null überprüft, nicht auf Richtigkeit
Sportart wird nicht überprüft ob existiert
Elternpferde werden nur auf null und Datum geprüft, nicht auf Existenz
Geschlecht der Eltern wird nicht validiert
"PreparedStatementCallback; Wert zu gross / lang für Feld "DESCRIPTION VARCHAR(255)" da im Validator in der validateNewSport Methode (Zeile 39)
"name.length()>255" anstatt "description.length()>255"

TS 16 [-7]

Tests liegen alle im unit/persistence Ordner, Ordnerstruktur ist falsch
Formatierung bei if Statements tlw. unterschiedlich (mal abstand zwischen Zeichen, mal nicht)
Elternpferde bzw. Sportart werden in der Entity nicht über Objekt sondern ID referenziert.
java.util.Date für Geburtsdatum ist falsch (LocalDate wäre hier besser)

TS 18 Anmerkungen

"Create Sport" unintuitiv auffindbar (button am ende der liste)
Liste von Pferden ist unter "Search Horse" platziert (nicht optimal)

TS 19

[-2] Beim anlegen einer Sportart mit zu langer Beschreibung kommt "Error! Error during sport creation" -- Beim Pferd wird eine sprechende Fehlermeldung ausgegeben.

TS 25 [-11]

/search endpoint für die suche ist nicht RESTful
/tree wäre auch beim Horse besser aufgehoben
POST für die Suche ist nicht REST Konform, POST nur zum Erstellen, Anfragen mit GET
Suche sollte auch unter /horses liegen
BAD_GATEWAY in allen Endpoint Klassen für ServiceException. ServiceException sind kein Grund für einen BAD_GATEWAY, das Gateway funktioniert ja, es ist ein anderer Fehler aufgetreten.

Feedback 15: 57/80 (~37 Stunden)

Userstories (erreicht): 79

Techstories (Abzüge): -22

Qualitätsmanager/in: -14

Usability Engineer: -2

Technischer Architekt/in: -6

Datenmanager/in: 0

US 1 ** [-1] Der name eines Pferdes kann nur aus Leerzeichen bestehen

TS 11 ** [-3] Deklaration der Interfaces fehlt gänzlich.

TS 13 ** [-1] Es wurde öfters assert anstatt von Assert Methoden (aus zum Beispiel org.junit.jupiter.api) bei den Tests verwendet.

TS 15 [-6] ** Länge wird nicht validiert ** Name kann ein leerer String sein und nur aus Leerzeichen bestehen ** Bei einem zu langem String wird nicht validiert und diese Eingabe führt zu einem Datenbankfehler.

TS 16 [-4] ** In den Entities sollte auf anderen Entities mit Entity-Objects anstatt von ID Referenzen verwiesen werden. ** für birthday wurde der Datentyp Date verwendet hier hätte LocalDate verwendet werden sollen. Da Date schon als Legacy Class angesehen wird.

TS 19 ** [-2] Wenn bei einem zu langen Namen ein Datenbankfehler auftritt wird dieser als Fehlermeldung im FE angezeigt.

TS 25 [-6] ** Die URI /horses/search bricht die Konvention: URIs enthalten keine Verben. Die Aktion soll rein über die HTTP-Methode bestimmt werden. ** Wenn prinzipiell die Validierung einsetzt ** Wenn ein Pferd einen Validierungsconflict aufweist wegen einem anderen Pferd wird ein 400er anstatt eines 409ers geworfen. 409 Conflict für Validierungsfehler, wenn sie die Beziehung zu einer anderen Entities betreffen, also einen Konflikt zum Zustand des Systems.

Feedback 16: 45/80 (~85 Stunden)

Gesamtpunkte: 45

** Userstories (erreicht): 78

** Techstories (Abzüge): -33

*** Qualitätsmanager/in: -11

*** Usability Engineer: -2

*** Technischer Architekt/in: -18

*** Datenmanager/in: -2

* US 6 ** [-2] Es kann nicht nach der Lieblingssportart gesucht werden. Eine Suche danach liefert immer alle Pferde als Ergebnis.

* TS 11 ** [-3] Kein einziges Interfaces wurde deklariert.

* TS 10 ** [-4] Es wurden zwei Mal printStackTrace im Code verwendet. Auf Standard Streams darf nur mit dem Logger geschrieben werden.

* TS 15 ** [-1] Die Elternteile können nachträglich jünger als das Kind gemacht werden.

* TS 16 ** [-3] In Entities wurden IDs anstatt von Entity-Objekten für die Eltern und Sport verwendet.

* TS 19 ** [-2] Wenn beim Anlegen einer Sportart ein zu langer Name angegeben wird kommt im Frontend keine Fehlermeldung. Der User weiß nicht was er falsch gemacht hat.

* TS 24 [-12] ** Die Exceptions werden gefangen, allerdings werden den Folge-Exceptions nicht die vorherige Exception mitgegeben. Wodurch der Stacktrace also wertvolle Information beim Debuggen verloren gehen.

** Es wurde 10 Mal eine Catch-All Exception "catch (Exception e)" verwendet. Hierbei ist das Abfangen der Exception zu allgemein und eine sinnvolle Behandlung ist hierbei nicht gegeben.

* TS 25 [-6] ** Bei dem Fall, dass ein Elternteil nicht dasselbe Geschlecht haben darf wie das andere Elternteil wird ein 422er anstatt eines 409ers geworfen. 409 Conflict für Validierungsfehler, wenn sie die Beziehung zu einer anderen Entity betreffen, also einen Konflikt zum Zustand des Systems.

** Eine Leere Suche liefert einen 400er zurück. Stattdessen sollte einfach eine leer Liste retourniert werden.

* TS 27 ** [-2] In der Datenbank wurde das Geschlecht als Char umgesetzt und nicht als Enum.

Feedback 17: 68/80 (~55 Stunden)

* Gesamtpunkte: 68

** Userstories (erreicht): 79

** Techstories (Abzüge): -11

*** Qualitätsmanager/in: -7

*** Usability Engineer: 0

*** Technischer Architekt/in: -4

*** Datenmanager/in: 0

* US 1 ** [-1] Es ist möglich das ein Name des Pferdes nur aus Leerzeichen besteht

* TS 11 ** [-3] Kein einziges Interfaces wurde deklariert.

* TS 15 ** [-1] Name kann blank sein, leer bzw aus Leerzeichen bestehen.

* TS 16 ** [-3] In der Horse Entity wurde mit den IDs anstatt mit Entity-Objekten auf die anderen Objekte referenziert.

* TS 25 [-4] ** Die URI /horses/new bricht die Konvention: URIs enthalten keine Verben. Die Aktion soll rein über die HTTP-Methode bestimmt werden.

** Wenn ein Pferd einen Validierungsconflict aufweist wegen einem anderen Pferd wird ein 422er anstatt eines 409ers geworfen. 409 Conflict für Validierungsfehler, wenn sie die Beziehung zu einer anderen Entity betreffen, also einen Konflikt zum Zustand des Systems.

Feedback 18: 33/80 (~60 Stunden)

Gesamtpunkte: 33

** Userstories (erreicht): 76

** Techstories (Abzüge): -43

*** Qualitätsmanager/in: -20 (-24 ohne Cap)

*** Usability Engineer: -2

*** Technischer Architekt/in: -19

*** Datenmanager/in: -2 * US 6

** [-1] Findet keine Pferde, wenn man Geburtstag von Pferd eingibt

* US 9 ** [-3] Detailansicht kann nicht vom Baum aus aufgerufen werden

* TS 10 [-8] ** Application Loglevel nicht auf mindestens DEBUG gesetzt

** Error Logging fehlt mehrmals in endpoints, Baseurl und Parameter fehlen mehrmals in endpoints, Logger TRACE/DEBUG nicht zu Beginn jeder business logic Methode (bei Validator garnicht)

* TS 15 [-10] ** mangelhafte validierung, wenn validiert wird, dann nur null check bzw. ob eltern generationen anzahl ein nicht negativer long ist ** mangelhafte Validierung zu Problemen mit Abhängigkeiten (keine Prüfung ob Elternpferde existieren, sportart existiert etc.) ** Mangelhafte Abhängigkeitsvalidierung führt zu Datenbank Exceptions

* TS 16 ** [-6] in Entities wird nur mittels IDs auf Eltern und Sportart verwiesen, Geschlecht nicht als Enum definiert

- * TS 19 ** [-2] Nicht aussagekräftige Fehlermeldung beim erstellen
- * TS 24 [-8] ** es werden mehrmals nullpointer exceptions gefangen ** Nullpointer wird mehrfach gethrowt, wodurch NullPointerException Fehler mehrfach reproduzierbar sind
- * TS 25 [-11] ** '/horses/family-tree/{id}' ist nicht REST-konform ** Wenn Name/Description zu lange bei Horse/Sport wird 500 geliefert statt 422 ** Bei Liste von Pferden werden auch die Eltern mitgeliefert, was nicht benötigte Daten in dem Kontext sind
- * TS 27 ** [-2] Geschlecht ist in der Datenbank nicht als enum gespeichert, anmerkung: varchar(30) ist sehr wenig für name/description

Feedback 19: 24/80 (~28h)

- * Gesamtpunkte: 24
- ** Userstories (erreicht): 56
- ** Techstories (Abzüge): -32
- *** Qualitätsmanager/in: -20 (-23 ohne Cap)
- *** Usability Engineer: 0
- *** Technischer Architekt/in: -12
- *** Datenmanager/in: 0
- *US 6 ** [-12] US6 funktioniert nicht
- *US 9 ** [-12] US9 funktioniert nicht
- * TS 10 ** [-3] falsche Loglevel-Konfiguration, sollte at:ac:tuwien:sepm: DEBUG sein nicht TRACE
- *TS 13 [-11]
- ** Endpoint REST Schicht nur 2 Testmethoden
- ** Service Schicht nur 2 Testmethoden
- ** Persistence Schicht nur 2 Testmethoden
- ** Service Schicht kein Negativtest
- ** Persistence Schicht kein Positivtest
- ** Endpoint REST Schicht kein Negativtest
- ** einige Tests nicht sinnvoll
- *TS 14 ** [-1] nur 13 Commits (sollten mindestens 16 sein → regelmäßige Commits)
- *TS 15 [-3]
- ** Geburtsdatum in der Zukunft möglich
- ** Länge von Name des Pferdes und der Sportart wird nicht überprüft
- * TS 16 [-4]
- ** In den Entities sollten andere Entities über Entity-Objekte verweisen, statt rein über ID (in den Dtos würde dies ausreichend sein, wenn dementsprechend validiert wird)
- ** Backend: Datumstyp für Geburtsdatum sollte am besten z.B. LocalDate und nicht Date sein.
- *TS 21 ** [-8] mvn clean package fails: 1 Failed, 1 Error
- *TS 25 ** [-4] falscher Status Code der richtigen Klasse verwendet (400 statt 422)

Feedback 20: 19/80 (~50h)

- ** Userstories (erreicht): 62
- ** Techstories (Abzüge): -43
- *** Qualitätsmanager/in: -17
- *** Usability Engineer: -4
- *** Technischer Architekt: -20 (-26 ohne Cap)
- *** Datenmanager/in: -2
- * US 3

- **[-2] Pferde mit Kindern können nicht gelöscht werden. Im Backend tritt ein Datenbankfehler auf (Referential integrity constraint violation).
- * US 6
- **[-4] Bei der Suche nach Geburtsdatum sollten alle Pferde aufgeleistet werden, die vorher geboren sind, nicht nachher. Die Suche nach Sportarten funktioniert nicht, gibt immer eine NotFoundException zurück, auch wenn Pferde mit entsprechenden Sportarten existieren.
- * US 9 **[-12] Kein Familienstammbaum implementiert.
- * TS 10 **[-3] DEBUG Logging ist nicht in application.yml konfiguriert
- * TS 11 **[-3] Javadoc: Beschreibung von Interface-Klassen mehrfach nicht vorhanden
- * TS 14 [-4]
- ** Ein Commit gefunden (abgesehen von Initial Commit), bei dem die Zugehörigkeit zur Userstory/Techstory nicht eingetragen ist.
- ** Datenbankdateien (*.db) committed
- * TS 15 [-7]
- ** Geburtsdatum eines Pferdes sollte nicht in der Zukunft liegen können.
- ** Durch Bearbeiten der Elternteile kann ein Kind zwei gleichgeschlechtliche Eltern haben. Elternteile können sowohl beim Erstellen als auch beim Bearbeiten nach Kindern geboren sein.
- ** Zu lange Namen und Beschreibungen führen zu Datenbankfehlern im Backend.
- * TS 19
- **[-4] Wenn beim Erstellen eines Pferdes versucht wird, das einen zu langen Namen zu verwenden, erhält man nur die Fehlermeldung "Error! Could not create horse", aber man weiß nicht, was das Problem ist. Wenn versucht wird, eine Sportart mit Namen rein aus Leerzeichen hinzuzufügen, wird diese zwar korrekterweise nicht erstellt, gibt aber auch hier keine Fehlermeldung. (Benutzer muss wissen, was passiert)
- * TS 23
- **[-5] Die Suche nach mehreren Parametern sollte mittels einer einzigen Datenbankabfrage geschehen. Das geht sehr gut mit einem einzigen SQL Statement. Für jeden Parameter hingegen eine eigene Suche zu performen und die Resultate dann zu kombinieren ist ressourcenverschwendend.
- * TS 24 [-11]
- ** Exceptions werden überwiegend geloggt und weitergeworfen
- ** Ansatz zur Exceptionbehandlung vermischt. Permanentes Wrappen von RuntimeExceptions und diese sind häufig in throws Klauseln deklariert.
- * TS 25
- **[-10] Datenbankexceptions werden nicht immer abgefangen und führen deshalb zu einem InternalServerError (Code 500), dieser sollte nie vorkommen. HTTP-Statuscode ist nicht 201 beim Erstellen von Sportarten; Wenn keine Pferde beim Filtern gefunden werden sollte der Statuscode 200 und nicht 404 sein.
- * TS 27 **[-2] Geschlecht als VARCHAR statt ENUM

Feedback 21: 77/80 (~70h)

- ** Userstories (erreicht): 80
- ** Techstories (Abzüge): -3
- *** Qualitätsmanager/in: 0
- *** Usability Engineer: 0
- *** Technischer Architekt/in: -3
- *** Datenmanager/in: 0

* TS 25 ** [-3] Aufbau von URIs für horse nicht korrekt. Um ein neues Pferd anzulegen sollte nicht ein Endpoint angelegt werden der "/horses/new" heißt dies ist nicht REST-Konform. Gut wäre /horse. (Bei Sports das gleiche)
Notiz: Das kann sich eigentlich nur auf die Frontend URL beziehen, im backend hatte ich keine solche Endpoints.

Feedback 22: 9/80 (~20h)

** Userstories (erreicht): 58
** Techstories (Abzüge): -49
*** Qualitätsmanager/in: -20 (-26 ohne cap)
*** Usability Engineer: -9
*** Technischer Architekt/in: -20 (-26 ohne cap)
*** Datenmanager/in: 0
* US 1 ** [-2] Pferd kann nicht ohne Beschreibung angelegt werden
* US 3 ** [-2] Pferd kann nach Löschen noch als Elternteil gesetzt werden (in Edit view)
* US 5 ** [-2] Es kann von der Detailansicht nicht zu den Eltern navigiert werden
* US 6 [-4] ** Name muss bei Suche vollständig und case sensitive sein (case insensitive like suche gewünscht) ** Beschreibung muss bei Suche vollständig und case sensitive sein
* US 9 ** [-12] Baumansicht nicht vorhanden
* TS 10 [-6] ** Loglevel DEBUG wurde nicht für das Paket von dem Programm definiert ** Einige Methoden in Services und Validator werden gar nicht geloggt ** Anmerkung: Anderes Level als INFO wäre in Validator mehr angebracht ** Anmerkung: ERROR statt WARN wenn Exception geloggt werden (Endpoint)
* TS 11 ** [-3] Interfaces selbst sind nicht mit JacaDoc dokumentiert
* TS 13 ** [-2] Keine Tests für Endpoint Layer
* TS 14 ** [-5] Datenbank wurde oft committed und ist auch am letzten Stand noch im git repo enthalten
* TS 15 [-6] ** Geschlecht und Geburtstag von Eltern können ungeachtet der Eltern-Kind Beziehung geändert werden ** Wenn bereits gelöscht Pferd als Elternteil eingetragen wird, führt das zu einem Datenbankfehler
* TS 16 [-4] ** Enum Klassenname (SEX) nicht CamelCase mit Großbuchstabe am Beginn ** in Entities wird nur über IDs auf andere Entities verwiesen
* TS 18 ** [-1] Bei Suche kann das Geschlecht nicht zurückgesetzt werden
* TS 19 [-8] ** Wenn Sport nicht erstellt werden kann trotzdem Erfolgsmeldung ** Fehlermeldungen nicht aussagekräftig
* TS 22 [-9] ** Methode in Service Layer greift auf HorseDto zu ** Parameter für Suche nicht in DTO gekapselt
* TS 24 [-5] ** Exception wird gefangen und unverändert weitergeworfen ** catch(Exception e) wurde verwendet
* TS 25 [-12] ** Verben werden in URI verwendet (/createhorse; /getall...) ** Klasse HttpParams wird nicht verwendet, Suchparameter werden einfach concatteniert ** Es wird bei Problemen zwischen Eltern und Kindern (Geburtstag) Status Code 422 statt 409 verwendet

Feedback 23: 27/80 (~56 Stunden, Danke an den freundlichen Tutor nebenan)

* Gesamtpunkte: 27
** Userstories (erreicht): 57
** Techstories (Abzüge): -30
*** Qualitätsmanager/in: -17
*** Usability Engineer: -0

*** Technischer Architekt/in: -7

*** Datenmanager/in: -6

* US 4 ** [-7] Eltern nicht implementiert

* US 5 ** [-4] Eltern werden nicht angezeigt

* US 9 ** [-12] Nicht implementiert

* TS 10 ** [-3] Debug Log Level fehlt

* TS 11 ** [-3] Interfaces selbst (Methoden darin schon) sind nie mit JavaDoc kommentiert

* TS 13 ** [-2] Positivtests der Persistence asserten nur assertDoesNotThrow und nicht, welche Daten genau zurück kommen, PositivTest des Endpoint testet auch nicht, ob die erwarteten Daten zurück kommen, sondern nur den Status und content type. Analog große Teile der Service Schicht Tests.

* TS 14 ** [-3] 4x mehr als 2 Stories zusammengefasst

* TS 15 ** [-3] Keine Validierung der Eltern

* TS 16 ** [-3] In Horse und HorseDTO auf Sport mit ID verwiesen * TS 24 ** [-5] Nicht Konsequent: Gemischte Ansätze zur Exceptionbehandlung in Service und Dao

* TS 25 ** [-2] URL nicht REST-Konform: /horses/search

* TS 27 ** [-6] Fremdschlüssel für Eltern fehlen

Feedback 24: 29/80 (~60 Stunden, -11 punkte wegen commits !!):

* Gesamtpunkte: 29

** Userstories (erreicht): 65

** Techstories (Abzüge): -36

*** Qualitätsmanager/in: -20 (27)

*** Usability Engineer: -5

*** Technischer Architekt/in: -9

*** Datenmanager/in: -2

* **US 1** ** [-1] Der Name eines Pferdes kann nur aus Leerzeichen bestehen

* **US 5** ** [-2] Es ist nicht möglich von der Detailseite eines Pferdes auf die Detailansicht eines Elternteils zu navigieren *

* **US 9** * [-12] Die Baumansicht wurde nicht implementiert

* **TS 11** [-6]

** Kein einziges Interfaces wurde deklariert.

** Es wurde überwiegend die kurze Zusammenfassung der Methode vergessen.

* **TS 12** ** [-2] Stundenliste ist nicht im Wurzelverzeichnis.

* **TS 14** [-11]

** Die Commit Messages verweisen nicht immer auf die UserStory / TechStory. ** [Die Commit messages sind öfters nicht aussagekräftig.

** Es wurde sogar wie immer die Datenbank commitet.

** Im Endcommit ist die Datenbank commitet.

* **TS 15** ** Eltern können nachträglich jünger als das Kinde gemacht werden.

* **TS 16** [-8]

** Für Geburtstag wurde String verwendet

** In Entities wurden IDs anstatt von Entity-Objekten für die Eltern und Sport verwendet.

** für birthday wurde der Datentyp Date verwendet hier hätte LocalDate verwendet werden sollen. Da Date schon als Legacy Class angesehen wird.

** Es wurde keine Enumeration für das Geschlecht verwendet.

* **TS 17** ** [-5] UI ist sehr überladen. Alle Funktionalitäten für Pferde befinden sich auf der selben Seite.

***TS 24** ** [-3] Es wurde zweimal eine `DataAccessException` gefangen, ohne der neuen Exception die alte mitzugeben.

* **TS 25** [-6]

** Wenn ein Pferd einen Validierungsconflict aufweist wegen einem anderen Pferd wird ein 422er anstatt eines 409ers geworfen. 409 Conflict für Validierungsfehler, wenn sie die Beziehung zu einer anderen Entities betreffen, also einen Konflikt zum Zustand des Systems.

** Suche die kein Ergebnis zurückgibt liefert einen 404 anstatt einer leeren Liste.

* **TS 27** [-2] ** In der Datenbank wurde das Geschlecht als `Varchar` umgesetzt und nicht als Enum. ** `date_of_birth` ist ein `DATETIME` und sollte ein `DATE` sein, da die Zeit nicht gespeichert wird.