

SEPM SS13 / Einzelbeispiel

Michael Ion

1005233

e1005233@student.tuwien.ac.at

Anwendungsfall 4: Beliebte Pferde

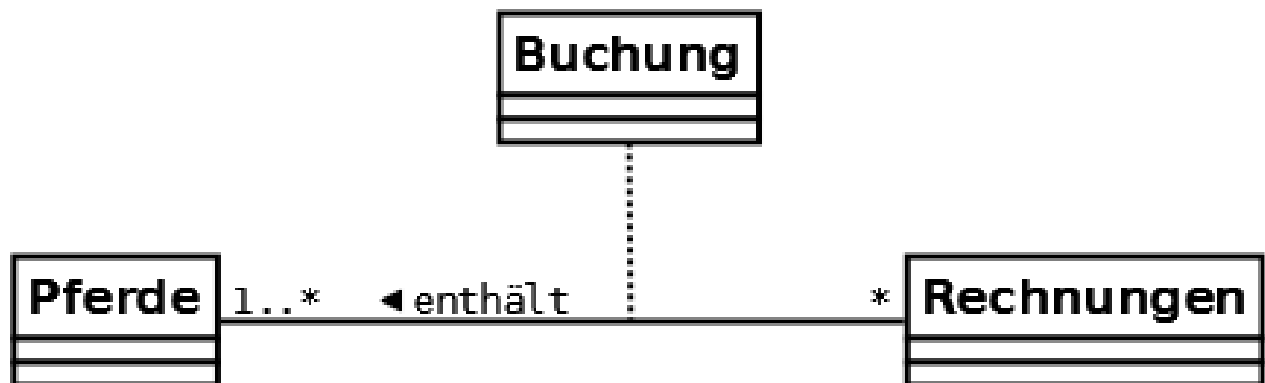
09.03.2013

Stundenliste

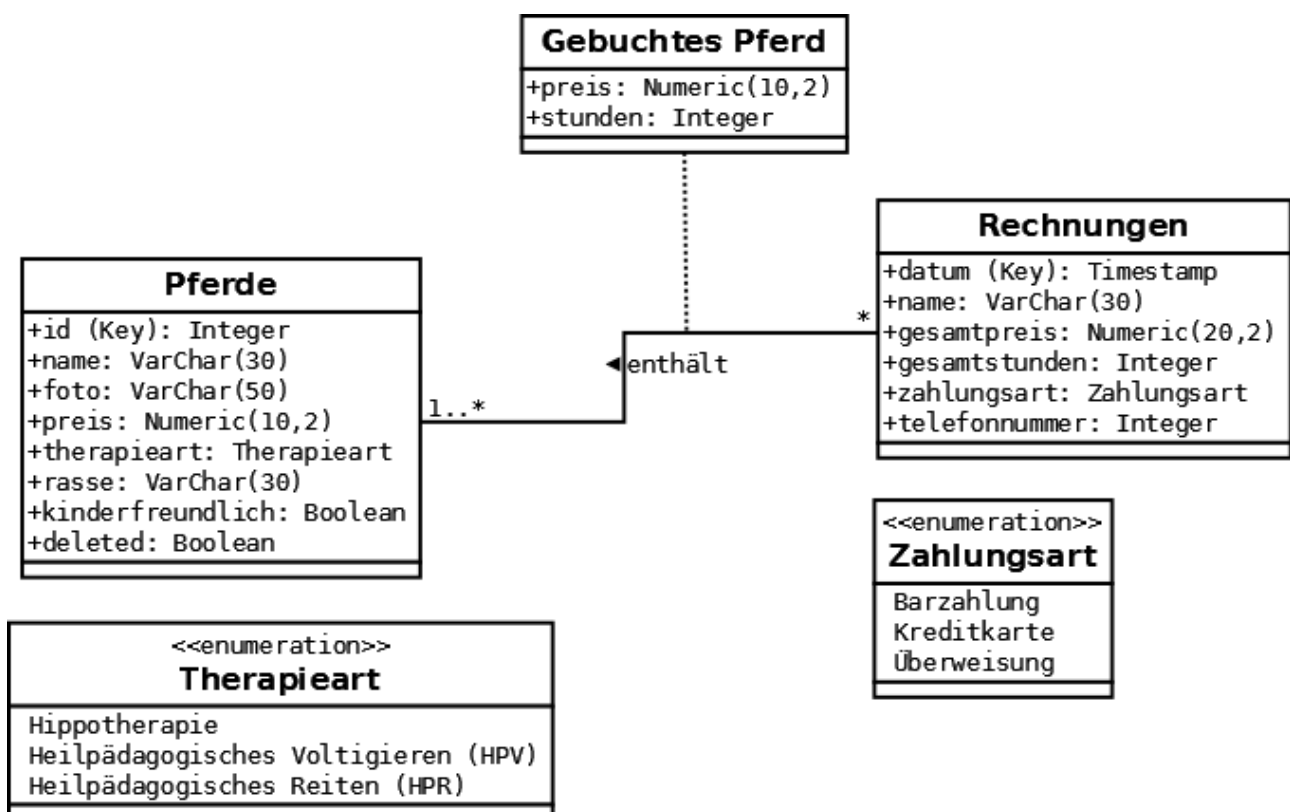
Datum	Start	Dauer (Stunden)	Tätigkeit
08.03.13	13:15:00	1	Lesen von Angabe + Leitfaden, Erstellung von Domänenmodell
09.03.13	15:20:00	2	Domänenmodell, Create-Scripts
10.03.13	14:15:00	2,5	Erstellung der Datenbank, Insert-Scripts
12.03.13	10:25:00	2,5	Einrichten der Entwicklungsumgebung, Erstellen der Datenbankverbindung über Connectionklasse
13.03.13	09:15:00	1,5	Erstellen des Klassendiagramms
14.03.13	15:25:00	2	Erstellen der DAO-Interfaces, Entities
15.03.13	21:30:00	1,5	Erstellung der DAO's, Bearbeiten der Arbeit vom 14.3., 12.3.
17.03.13	14:20:00	3	Erstellung von Testklassen, Bearbeiten der Arbeit vom 15.3. 2013
18.03.13	12:10:00	5	Erweiterung der PferdDao-Testklasse, JDBCpferdDao-Testklasse und Pferd, Implementierung von JDBCpferdDao
19.03.13	13:15:00	1,5	Anwendungsfalldiagramm
20.03.13	15:00:00	2	Anwendungsfallbeschreibung, Anwendungsfälle ins Klassendiagramm
21.03.13	09:15:00	2,5	Erstellen des Service Interface, Beginn der Implementierung des Service, allgemeine Refaktorisierung, neue Exceptions
22.03.13	15:00:00	3	Beginn des GUI, Neue Suchfunktion für Pferde, Suchklasse für Pferde erstellt
23.03.13	08:25:00	1,75	Weitere GUI Ausarbeitung
23.03.13	18:45:00	0,75	Weitere GUI Ausarbeitung (PferdPanel)
23.03.13	20:20:00	2	Beginn der RechnungDAOTest-Klasse, JDBCRechnungDAOTest-Klasse, JDBCRechnungDAO
24.03.13	07:05:00	2,5	Weitere GUI Ausarbeitung (PferdPanel)
24.03.13	13:20:00	2	Weitere GUI Ausarbeitung (PferdPanel)
24.03.13	15:45:00	2,25	Weitere GUI Ausarbeitung + Feintuning(PferdPanel)
24.03.13	20:45:00	1,5	Weitere GUI Ausarbeitung, Beginn der Implementierung der ActionListener
25.03.13	08:30:00	1,5	Weitere GUI Ausarbeitung, weitere ActionListener (PferdPanel)
25.03.13	12:30:00	1	Weitere GUI Ausarbeitung
25.03.13	16:30:00	2	Weitere GUI Ausarbeitung(Hinzufügen, BildFilter für FileChooser, Löschen,mehr ActionListener), kleine

		Änderungen in JDBCService Löschen erweitert, Bearbeiten
26.03.13	09:00:00	2 Funktionalität hinzugefügt, allgemeine Fertigstellung von PferdPanel
27.03.13	10:15:00	3 Änderungen an Pferdpanel, Table, Löschen Dialog, Neue Tests, Serviceklassen und RechnungDAO angepasst
27.03.13	13:50:00	2 RechnungDAO, RechnungDAOTest, JDBCRechnungDAO erweitert, Logging Statements zu PferdPanel, JDBCPferdDAO hinzugefügt
27.03.13	17:45:00	1 Equals in Buchung, Rechnung, neue Methoden in JDBCRechnungDAO, JDBCPferdDao angepasst, RechnungDAOTest angepasst
27.03.13	20:55:00	1,5 Neue Exceptions, Main, JDBCRechnungDAO und RechnungDAO überarbeitet, neue Methoden in Serviceklassen hinzugefügt
28.03.13	07:30:00	1 ServiceTest erstellt, überflüssige Imports gelöscht, getPopularPferde in JDBCService erweitert
30.03.13	10:00:00	1,5 Implementierung von RechnungPanel, CustomTable in PferdeTable und RechnungTable gespalten, kleine Bugfixes
30.03.13	16:30:00	1,25 RechnungPanel erweitert mit DetailAnzeige
31.03.13	09:00:00	1 AktivTable hinzugefügt, Arbeiten an der Erstellung neuer Rechnungen
31.03.13	13:30:00	1 Weiterarbeiten an Erstellung neuer Rechnung
31.03.13	15:30:00	1,5 Abspeichern der Rechnungen fertiggestellt
31.03.13	18:45:00	1 Code cleanup, Insert into-Skript erweitert
05.04.13	20:00:00	1 implementierte Funktionalitäten zur Findung und Verteuerung der beliebtesten Pferde im GUI
07.04.13	14:00:00	1,25 hinzugefügt: ServiceTests, RechnungDAOTests, JDBCServiceTests
07.04.13	18:10:00	0,75 Tests in JDBCServiceTest und Servicetest hinzugefügt
08.04.13	10:15:00	0,75 Fertigstellen der Dokumentation
Summe:		68,75

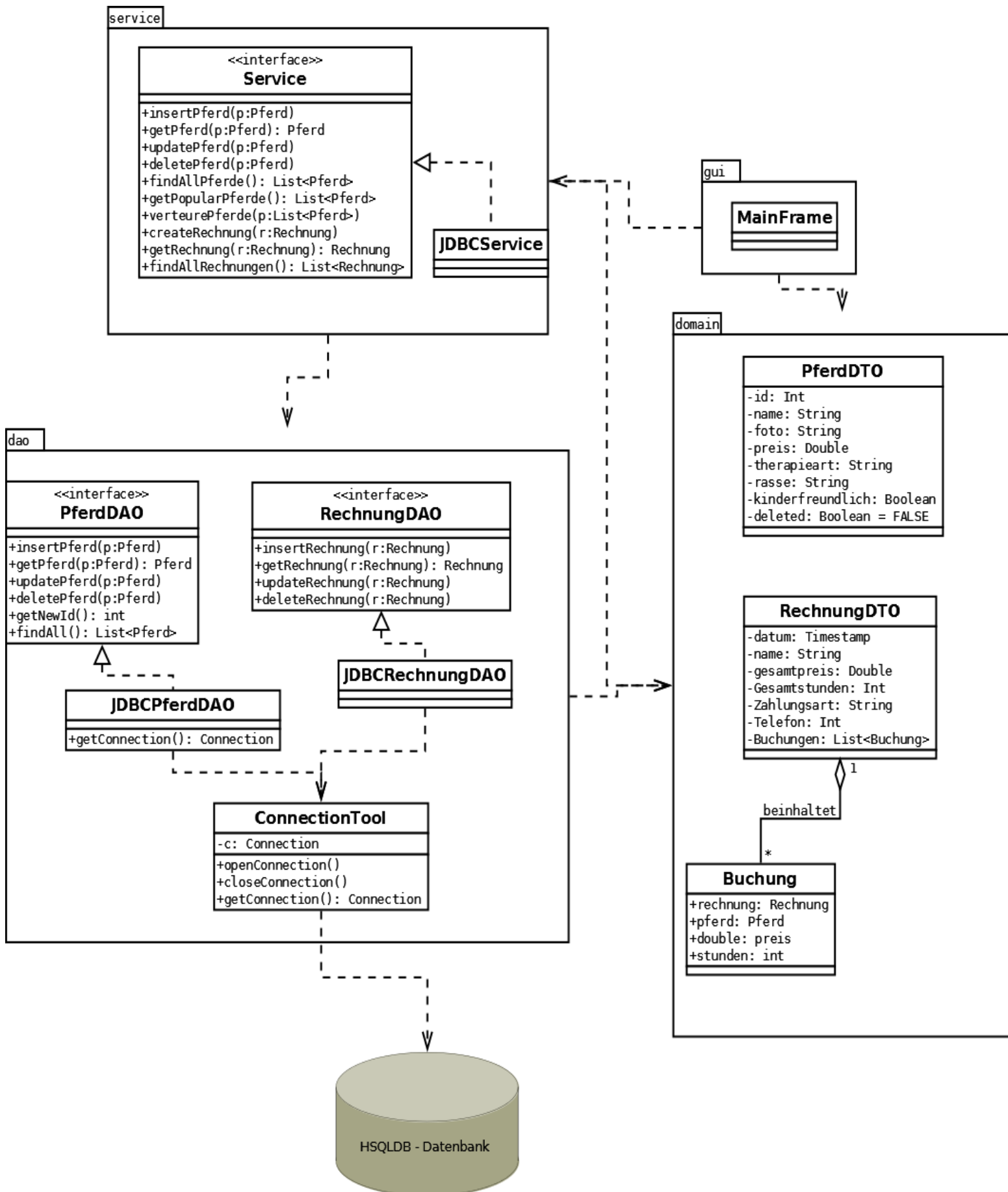
Domänenmodell



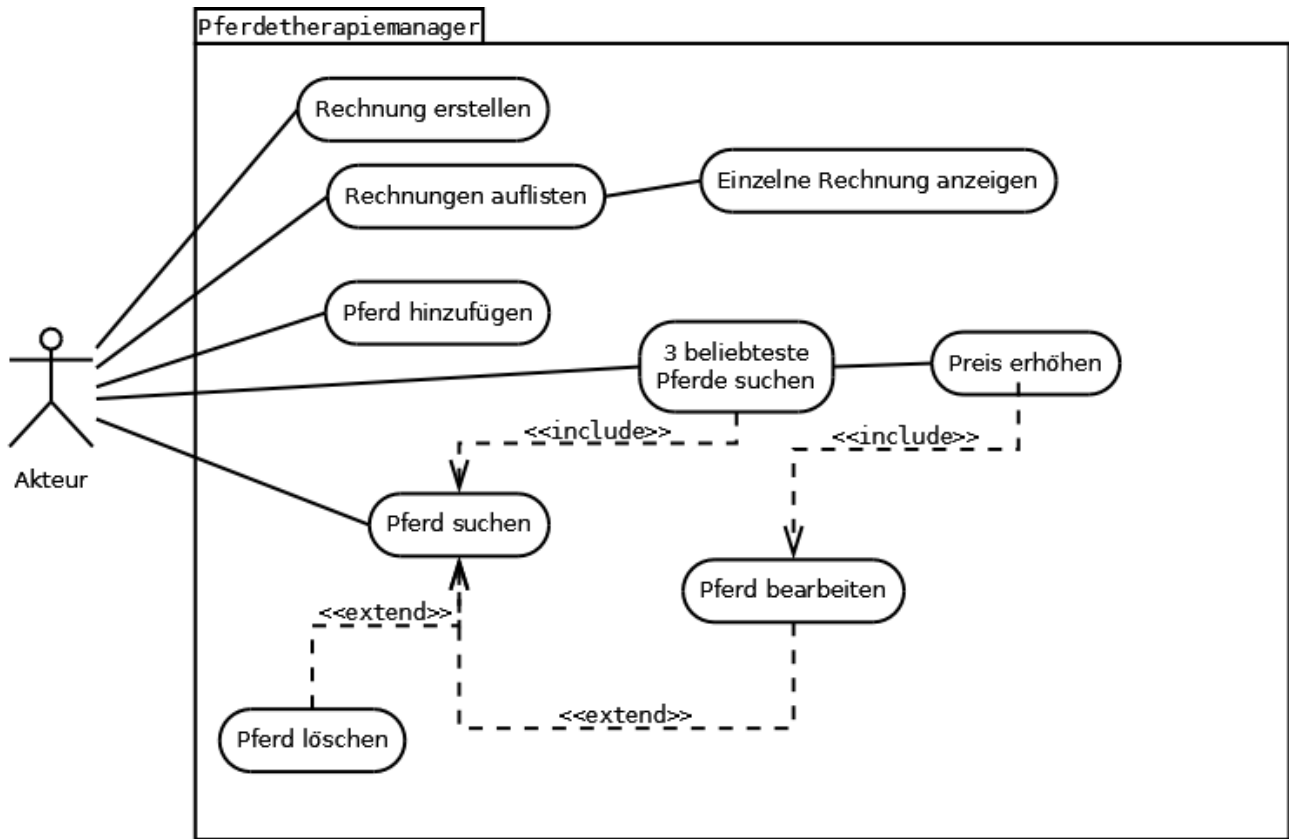
- mit Attributen (zur besseren Planung erstellt):



Klassendiagramm



Anwendungsfalldiagramm



Anwendungsfallbeschreibung

Anwendungsfälle:

- Pferd hinzufügen
- Pferd löschen
- Pferd bearbeiten
- Pferd suchen
- 3 beliebtesten Pferde suchen
 - Preis erhöhen
- Rechnung erstellen
- Rechnung auflisten
 - Einzelne Rechnungen anzeigen

Pferd hinzufügen

1. Identification summary

- Titel: Pferd hinzufügen
- Scope: Service
- Level: User Goal
- Akteure: Angestellter
- Kurzbeschreibung: Der Angestellte fügt ein neues Pferd in die Applikation hinzu.

2. Scenarios

- Vorbedingungen: Man befindet sich im Pferd-Tab und hat auf die Schaltfläche „Pferd hinzufügen“ unter dem Menüpunkt „Aktionen“ gedrückt.
- Hauptszenario: Zum Anlegen eines neuen Pferdes füllt der Anwender die nötigen Felder im Formular aus (). Im System wird dann ein neues Pferd mit den entsprechenden Attributen angelegt.
- Fehlersituationen: Das Formular muss richtig ausgefüllt werden: Es dürfen weder unzulässige Zeichen verwendet (z.B. Buchstaben in ein Preisfeld) oder Pflichtfelder leer gelassen werden. Wenn dies der Fall ist, erscheint eine Fehlermeldung und das Formular wird weiterhin angezeigt
- Alternativszenario: Fehlermeldung bei unzulässiger Eingabe
- Nachbedingungen: Das neue Pferd ist abgespeichert, kann jetzt über neue Rechnungen gebucht werden. Es kann aber auch bearbeitet oder gelöscht werden (andere Use Cases)

3. Non-functional Constraints

Der User soll nicht zu lange darauf warten, dass das Pferd eingefügt wird (max. 5 Sekunden).

Pferd bearbeiten

1. Identification summary

- Titel: Pferd bearbeiten
- Scope: Service
- Level: User Goal
- Akteure: Angestellter
- Kurzbeschreibung: Die Attribute eines in der Applikation bestehenden Pferds werden bearbeitet.

2. Scenarios

- Vorbedingungen: Das Pferd wurde in die Applikation hinzugefügt und erscheint über eine Suche in der Liste der buchbaren Pferde. Das Pferd wurde ausgewählt und auf den „Pferd bearbeiten“-Button gedrückt.
- Hauptszenario: Der Anwender wählt in der Liste der buchbaren Pferde das Pferd, das er zu ändern wünscht aus und passt dessen Attribute im erscheinenden Formular nach Belieben an.
- Fehlersituationen: Es gelten dieselben Fehlerszenarien wie beim Einfügen, die Daten in den Formularfelder müssen zulässig sein.
- Alternativszenario: Fehlermeldung bei unzulässiger Eingabe, dann wird das Formular wieder angezeigt.
- Nachbedingungen: Das Pferd ist nun in der Applikation aktualisiert worden.

3. Non-functional Constraints

Der User soll nicht zu lange darauf warten, dass das neu bearbeitete Pferd wieder abgespeichert wird und in der Tabelle erscheint (max. 5 Sekunden).

Pferd löschen

1. Identification summary

- Titel: Pferd löschen
- Scope: Service
- Level: User Goal
- Akteure: Angestellter
- Kurzbeschreibung: Ein Pferd, das in der Liste der buchbaren Pferde erscheint, wird ausgewählt und gelöscht.

2. Scenarios

- Vorbedingungen: Um es löschen zu können, muss ein Pferd zuerst in die Applikation hinzugefügt werden, um ausgewählt zu werden. Das Pferd wurde mittels Suche gefunden.
- Hauptszenario: Das Pferd wird in der Liste ausgewählt und mittels Tastendruck auf einen „Löschen“-Button (und anschließendem Dialog, ob es wirklich gelöscht werden soll) aus der Liste entfernt.

- Fehlersituationen: ---
- Alternativszenario: Der User kann den Vorgang abbrechen in dem er im „Wollen sie das Pferd wirklich löschen“-Dialog auf „Nein“ klickt.
- Nachbedingungen: Das Pferd wurde aus der Liste entfernt, kann also nicht mehr gebucht und zu Rechnungen hinzugefügt werden. Allerdings erscheint es noch auf alten Rechnungen, die es beinhalten.

3. Non-functional Constraints

Der User soll nicht zu lange darauf warten, dass das ausgewählte Pferd gelöscht wird und nicht mehr in der Tabelle erscheint (max. 5 Sekunden).

Pferd suchen

1. Identification summary

- Titel: Pferd suchen
- Scope: Service
- Level: User Goal
- Akteure: Mitarbeiter
- Kurzbeschreibung: Es wird eine Funktion zur Suche von Pferden gewährleistet. Zu Beginn wird eine Liste aller Pferde dargestellt, die dann nach Auswahl verschiedener Attribute und Parameter eingeschränkt wird.

2. Scenarios

- Vorbedingungen: Es sind genügend Pferde in der Applikation eingefügt, sonst ist die Suche überflüssig.
- Hauptszenario: Der User sieht zunächst eine Liste aller verfügbaren Pferde. Dann kann er über Parameter im selben Fenster die Liste einschränken, z.B. kann er nur Pferde einer bestimmten Therapieart anzeigen lassen. Nachdem er seine gewünschten Einstellungen vorgenommen hat, kann er die Ansicht aktualisieren und sieht nun eine Ergebnisliste an Pferden, die seinen Wünschen entsprechen.
- Fehlersituationen: ---
- Alternativszenario: Es kann eine leere Liste erscheinen, weil kein gespeichertes Pferd den Suchanforderungen entspricht.
- Nachbedingungen: Über die Suchfunktion kann der User leichter an das Pferd kommen, das seinen Wünschen am ehesten entspricht und es nun schneller bearbeiten, buchen oder löschen (andere Use Cases).

3. Non-functional Constraints

Der User sollte nicht zu lange auf die Suchergebnisse warten (max. 5 Sekunden).

3 beliebtesten Pferde suchen

1. Identification summary

- Titel: 3 beliebtesten Pferde
- Scope: Service
- Level: User Goal
- Aktoren: Mitarbeiter
- Kurzbeschreibung: Diese Funktion zeigt die drei meistgebuchten Pferde an.

2. Scenarios

- Vorbedingungen: Es müssen zumindest drei Pferde in der Applikation verfügbar sein.
- Hauptszenario: Es wird im Menü „Aktionen“ auf die Option gedrückt, die drei beliebtesten Pferde anzuzeigen. Diese werden dann angezeigt.
- Fehlersituationen: Wenn weniger als 3 Pferde verfügbar sind, wird eine entsprechende Fehlermeldung angezeigt.
- Alternativszenario: ---
- Nachbedingungen: Es werden die 3 beliebtesten Pferde angezeigt und nun kann deren Preis erhöht werden (anderer Use Case).

3. Non-functional Constraints

Die Dauer der Ausgabe der 3 beliebtesten Pferde sollte nicht zu lange dauern (max. 5 Sekunden)

Preis erhöhen

1. Identification summary

- Titel: Preis erhöhen
- Scope: Service
- Level: User Goal
- Aktoren: Mitarbeiter
- Kurzbeschreibung: Der stündliche Preis der drei beliebtesten Pferde wird um 5% angehoben.

2. Scenarios

- Vorbedingungen: Es wurde der Use Case „3 beliebteste Pferde suchen“ ausgeführt und nun werden jene aufgelistet.
- Hauptszenario: Über einen Klick auf einen Button kann in der Anzeige der drei beliebtesten Pferde deren Preis angehoben werden. Die stündliche Gebühr erhöht sich dann um 5%.
- Fehlersituationen: ---
- Alternativszenario: ---
- Nachbedingungen: Der Preis wurde um 5% erhöht.

3. Non-functional Constraints

Der User sollte nicht zu lange darauf warten, dass der Preis des neuen Pferds in der angezeigten Tabelle aktualisiert wird (max. 5 Sekunden).

Rechnung erstellen

1. Identification summary

- Titel: Rechnung erstellen
- Scope: Service
- Level: User Goal
- Akteure: Mitarbeiter
- Kurzbeschreibung: Der Mitarbeiter erstellt eine neue Rechnung indem er die gewünschten Pferde auswählt und die jeweils gebuchten Stunden einträgt.

2. Scenarios

- Vorbedingungen: Es müssen Pferde in der Applikation verfügbar sein, um Rechnungen erstellen zu können.
- Hauptszenario: Der User wählt ein Pferd aus und kann dann auf „Zur Rechnung hinzufügen“ klicken. Ein Fenster erscheint, wo er die gewünschten Stunden spezifiziert. Danach erscheint das Pferd mit den gebuchten Stunden und dem Gesamtpreis in der Rechnung.
- Fehlersituationen: Die Eingabe der gewünschten Stunden muss ein gültiger Wert sein, sonst erscheint eine Fehlermeldung und es muss erneut eingegeben werden.
- Alternativszenario: ---
- Nachbedingungen: Die Rechnung wird permanent in die Applikation eingespeichert, sie kann nicht mehr gelöscht bzw. bearbeitet werden.

3. Non-functional Constraints

Der User sollte nicht zu lange darauf warten, dass die neue Rechnung abgespeichert und in der Rechnungstabelle angezeigt wird (max. 5 Sekunden).

Rechnungen auflisten

1. Identification summary

- Titel: Rechnungen auflisten
- Scope: Service
- Level: User Goal
- Akteure: Mitarbeiter
- Kurzbeschreibung: Es werden alle in der Applikation gespeicherten Rechnungen aufgelistet.

2. Scenarios

- Vorbedingungen: Man hat auf den Tab „Rechnungen“ gedrückt und sieht nun ein Panel mit allen Rechnungen.
- Hauptszenario: Es werden alle in der Applikation gespeicherten Rechnungen aufgelistet.
- Fehlersituationen: Es wird nichts aufgelistet, wenn keine Rechnungen vorhanden sind.
- Alternativszenario: ---
- Nachbedingungen: Jetzt kann eine bestimmte Rechnung ausgewählt werden um mehr Informationen darüber zu erhalten.

3. Non-functional Constraints

Der User sollte nicht zu lange darauf warten, dass alle Rechnungen gefunden und aufgelistet werden (max. 5 Sekunden).

Einzelne Rechnung anzeigen

1. Identification summary

- Titel: Einzelne Rechnung anzeigen
- Scope: Service
- Level: User Goal
- Akteure: Mitarbeiter
- Kurzbeschreibung: Aus der Auflistung aller Rechnungen werden einzelne zur Detaildarstellung ausgewählt.

2. Scenarios

- Vorbedingungen: Es wurde der Use Case „Rechnungen auflisten“ durchgeführt und nun liegt eine Liste aller in der Applikation gespeicherten Rechnungen vor.
- Hauptszenario: Eine Rechnung aus der Liste wird ausgewählt und es wird auf „Details anzeigen“ geklickt. Es erscheint ein neues Fenster mit weiteren Informationen zur Rechnung (welche Pferde gebucht wurden, zu wieviele Stunden, zu welchem Preis)
- Fehlersituationen: ---
- Alternativszenario: ---
- Nachbedingungen: ---

3. Non-functional Constraints

Der User sollte nicht zu lange darauf warten, dass alle Informationen zu einer Rechnung gefunden werden.