

Wendy's Pferdepension

Software Engineering und Projektmanagement SS17

Felix Wagner – 1429339

e1429339@student.tuwien.ac.at

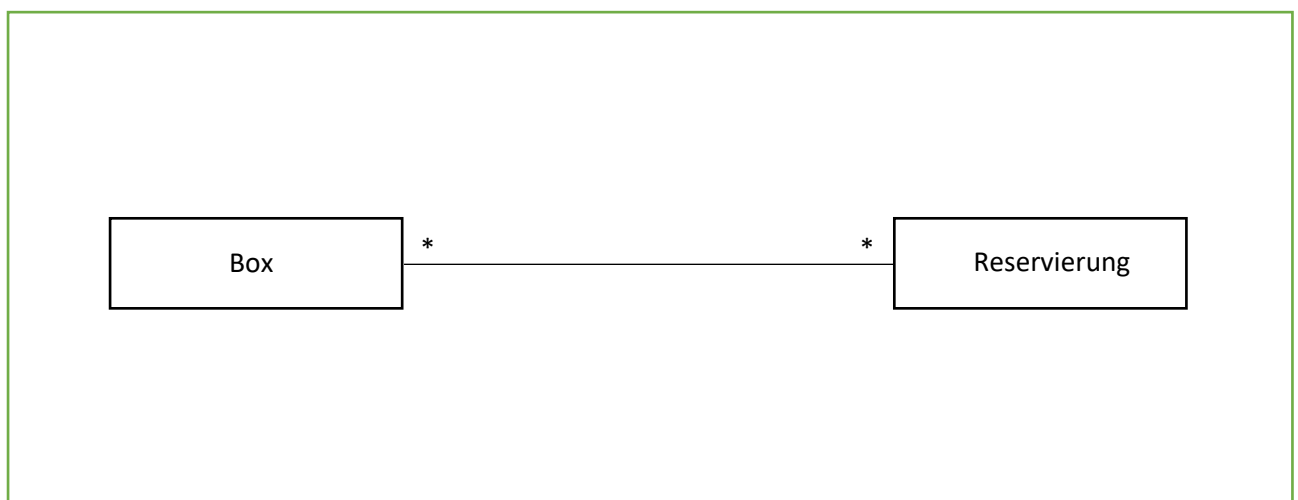
26.03.2017

Stundenliste:

Datum	Beginn	Ende	Erledigt
08.03.2017	19:10	22:30	Angabe, Domänenmodell, Create, Insert
14.03.2017	20:15	23:15	Javaprojekt angelegt, alle nötigen Ressourcen importiert, Verbindung zur Datenbank hergestellt, save/get von Persistenzschicht Box implementiert, Junit Test für save geschrieben
15.03.2017	12:30	13:30	Persistenzschicht Box delete,update implementiert
15.03.2017	16:00	18:00	Persistenzschicht Box search/searchall implementiert; Serviceschicht von Box (Interface+Implementierung)
15.03.2017	20:00	23:00	Scenbuilder eingerichtet, Persistenzschicht Box überarbeitet, tableview erstellt und befüllt
16.03.2017	14:30	17:00	MainController für alle Controller implementiert, Löschen Button implementiert, Bearbeiten Button angefangen zu implementieren
16.03.2017	19:00	22:30	Hinzufügen in Box implementiert (+dazugehörigen Controller und fxml Dateien); Bearbeiten Controller bearbeitet
17.03.2017	10:00	12:00	Gui - Erweiterte Suche; Image-View
17.03.2017	14:00	18:00	Gui Box vervollständigt, mit Fehlerbehandlung, kleinere fehler in der Persistenzschicht noch ausgebessert
17.03.2017	12:00	17:00	Persistenzschicht von Reservierung erstellt und Serviceschicht von Reservierung erstellt
17.03.2017	20:00	00:00	Angefangen Gui Reservierung zu erstellen; Filterbutton und Bearbeitenfenster implementiert
18.03.2017	10:00	12:00	Persistenzschicht von Reservierung SQL Statements überarbeitet und Fehler darin behoben
18.03.2017	15:00	22:00	Persistenzschicht Reservierung checkMetoden erweitert; SQL Statements von absoluten Zeitrahmen zu dynamischen geändert; GUI Reservierung fertig implementiert + Verifizierung von Eingaben
19.03.2017	10:00	12:00	Serviceschicht von Reservierung um computePrice Methode erweitert. Mit Rechnungsbelegung GUI begonnen.
19.03.2017	14:00	15:00	GUI von Rechnungsbelegung fertiggestellt und Verifizierung der Eingabedaten
19.03.2017	17:00	23:00	Durchdenken der Angabe für Statistik, Statistikservice Methoden zum zählen von Tagen implementiert, erstes Barchart implementiert, Gui für Statistik erstellt
20.03.2017	10:30	12:00	Gui von Statistik um Barcharts erweitert; einzelne Box Statistik fertiggestellt
21.03.2017	12:00	15:00	Statistik Serviceschicht weiterimplementiert; Methoden für Suche Beste/Schlechteste Box
21.03.2017	18:00	22:00	Preisanpassungsmethoden in der Serviceschicht fertiggestellt
22.03.2017	12:00	15:00	Preisanpassungsmethoden in GUI integriert
22.03.2017	17:00	22:00	GUI Validierung von Eingaben + Reservierungen bearbeiten ergänzt
23.03.2017	09:00	12:00	Exceptionhandling
23.03.2017	14:00	16:30	Excpetionhandling + Logger ergänzt
23.03.2017	20:00	00:00	Validierung in Service und Persistenzschicht

24.03.2017	00:00	00:30	Ordnerstruktur verbessert und Fehler in der Statistik ausgebessert
24.03.2017	10:00	12:00	Tests für BoxDAO geschrieben
24.03.2017	14:00	18:00	BoxDAO Tests, Serviceschicht Tests, Insert Daten umgeschrieben
25.03.2017	22:00	23:59	Code Dokumentation
26.03.2017	00:01	02:00	Doänenmodell, Anwendungsfallbeschreibung
26.03.2017	10:00	12:30	Anwendungsfallbeschreibung, Anwendungsfalldiagramm
	Summe:	83h20min	

Domänenmodell (konzeptionelle Sichtweise):



Anwendungsfalldiagramm:



Anwendungsfallbeschreibung:

Titel: Box hinzufügen

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein/e Mitarbeiter/in fügt eine Box hinzu.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptscenario: Der/die Mitarbeiter/in fügt eine Box mit den Attributen (Foto, Fenster, Einstreu, Drinnen, Größe, Bereich und Tagessatz) hinzu. Die hinzugefügte Box wird dann im Programm angezeigt.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben. Falls versucht wird, ungültige Werte in der Datenbank zu speichern, wird eine Fehlermeldung ausgegeben.

Alternativscenario: Wird eines der Pflichtfelder (Fenster, Einstreu, Drinnen, Größe, Bereich und Tagessatz) nicht ausgefüllt oder ungültige Werte eingegeben wird eine Fehlermeldung ausgegeben.

Nachbedingung: Der/die Mitarbeiter/in kann die hinzugefügte Box bearbeiten und löschen (andere Use Cases).

Titel: Box löschen

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein Mitarbeiter löscht eine vorhandene Box.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptscenario: Der/die Mitarbeiter/in wählt eine Box aus und kann sie löschen. Vor der Ausführung der Löschung muss der/die Anwender/in einen Dialog bestätigen. Sind noch ausständige Reservierungen für diese Box vorhanden wird diese trotzdem gelöscht und der Mitarbeiter/in benachrichtigt.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben. Existiert die Box in der Datenbank nicht, wird ein Fehler ausgegeben.

Nachbedingung: Die gelöschte Box wird dem/der Mitarbeiter/in nicht mehr im Programm angezeigt.

Titel: Box bearbeiten

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein/e Mitarbeiter/in bearbeitet eine vorhandene Box.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptszenario: Der/die Mitarbeiter/in bearbeitet die Attribute (Fenster, Einstreu, Drinnen, Größe, Bereich und Tagessatz) einer ausgewählten Box.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben. Existiert die Box in der Datenbank nicht, wird ein Fehler ausgegeben. Ebenso wenn versucht wird, ungültige Werte in der Datenbank abzuspeichern.

Alternativszenario: Wird eines der Pflichtfelder (Fenster, Einstreu, Drinnen, Größe, Bereich und Tagessatz) nicht ausgefüllt oder ungültige Werte eingegeben wird eine Fehlermeldung ausgegeben.

Nachbedingung: Der/die Mitarbeiter/in sieht die bearbeitete Box im Programm.

Titel: Box suchen

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein/e Mitarbeiter/in sucht nach Boxen mit bestimmten Suchkriterien.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptszenario: Der/die Mitarbeiter/in gibt Suchkriterien (ID, Fenster, Einstreu, Drinnen, Größe, Bereich und Tagessatz) für die gewünschten Boxen an. Das Programm zeigt die gesuchten Boxen an. Der/die Anwender/in kann sich auch alle Boxen anzeigen lassen.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben.

Alternativszenario: Existiert keine Box zu den angegebenen Suchkriterien, wird keine Box ausgegeben.

Nachbedingung: Der/die Mitarbeiter/in kann die gefilterten Boxen bearbeiten und löschen (andere Use-Cases).

Titel: Reservierung anlegen

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein Mitarbeiter legt eine Reservierung für eine/einen Kundin/Kunden an.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptszenario: Der/die Mitarbeiter/in legt eine Reservierung mit den Attributen (Kundenname, Von, Bis, Boxen mit Pferdenamen) für eine/einen Kundin/Kunden an. Der/die Mitarbeiter/in kann für eine Reservierung, mehrere Boxen mit dazugehörigen Pferden anlegen.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben.

Alternativszenario: Gibt der/die Mitarbeiter/in keine Box mit dazugehörigem Pferd an, wird eine Fehlermeldung ausgegeben. Falls die zu reservierenden Boxen schon in dem Zeitraum belegt sind, wird eine Fehlermeldung ausgegeben.

Nachbedingung: Der/die Mitarbeiter/die kann die hinzugefügte Reservierung bearbeiten und löschen (andere Use Cases).

Special Requirements: Der Anwendungsfall „Reservierung anlegen“ wird sehr häufig in Wendy's Pferdepension Verwendung finden.

Titel: Reservierung suchen

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein/e Mitarbeiter/in sucht nach Reservierungen in einem bestimmten Zeitraum.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptszenario: Der/die Mitarbeiter/in gibt einen Zeitraum für die gewünschten Reservierungen an. Das Programm zeigt die gesuchten Reservierungen an. Der/die Anwender/in kann sich auch alle Reservierungen anzeigen lassen.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben.

Alternativszenario: Existiert keine Reservierung in dem angegebenen Zeitraum, wird keine Reservierung ausgegeben.

Nachbedingung: Der/die Mitarbeiter/in kann die gefilterten Reservierungen bearbeiten/löschen (andere Use-Cases).

Titel: Reservierung bearbeiten

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein/e Mitarbeiter/in bearbeitet eine vorhandene Reservierung.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptszenario: Der/die Mitarbeiter/in bearbeitet die Attribute (Kundenname, Von, Bis, Boxen mit Pferden) einer ausgewählten Reservierung. Dabei kann er/sie auch weitere Boxen mit Pferden hinzufügen oder löschen. Eine Reservierung kann nur bis zum Stichtag bearbeitet werden.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben. Existiert die Reservierung in der Datenbank nicht, wird ein Fehler ausgegeben. Falls versucht wird, ungültige Werte in der Datenbank zu speichern, wird eine Meldung ausgegeben.

Alternativszenario: Falls die neu hinzugefügte/n Boxe/n in dem angegebenen Zeitraum schon reserviert sind, wird eine Meldung ausgegeben. Ebenso wenn der Zeitraum geändert wird und sich schon die vorhandenen Boxen mit anderen Reservierungen überschneiden. Ist der Stichtag schon vorbei, wird dem/der Anwender/in eine Meldung ausgegeben.

Nachbedingung: Der/die Mitarbeiter/in sieht die bearbeitete Reservierung im Programm.

Titel: Reservierung löschen

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein/e Mitarbeiter/in löscht eine vorhandene Reservierung.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptszenario: Der/die Mitarbeiter/in wählt eine Reservierung aus und kann sie löschen. Vor der Ausführung der Löschung muss der/die Anwender/in einen Dialog bestätigen. Die Reservierung kann nur bis zum Stichtag gelöscht werden.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben. Existiert die Reservierung in der Datenbank nicht, wird ein Fehler ausgegeben.

Alternativszenario: Ist der Stichtag der Reservierung schon vorbei, wird dem/der Anwender/in eine Fehlermeldung ausgegeben.

Nachbedingung: Die gelöschte Box wird dem/der Mitarbeiter/in nicht mehr im Programm angezeigt.

Titel: Rechnung ausstellen

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein/e Mitarbeiter/in stellt eine Rechnung aus.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptscenario: Der/die Mitarbeiter/in wählt eine Reservierung aus und kann zu dieser eine Rechnung ausstellen lassen. Dem/der Anwender/in wird in einem Dialogfenster die Rechnungsnummer, der Kundenname, die reservierten Boxen mit Pferden, der dazugehörige Tagessatz und die Gesamtsumme ausgegeben. Die Rechnung kann erst nach Beginn des Stichtages ausgestellt werden.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben. Existiert die Reservierung in der Datenbank nicht, wird ein Fehler ausgegeben.

Alternativscenario: Hat der Stichtag der Reservierung noch nicht begonnen, wird dem/der Mitarbeiter/in eine Fehlermeldung ausgegeben.

Nachbedingung: Die bezahlte Reservierung wird dem/der Mitarbeiter/in nicht mehr im Programm angezeigt.

Special Requirements: Der Anwendungsfall „Reservierung anlegen“ wird sehr häufig in Wendy's Pferdepension Verwendung finden.

Titel: Statistik anlegen

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein/e Mitarbeiter/in legt eine Statistik an.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptscenario: Der/die Mitarbeiter/in kann eine Statistik zu den Reservierungen und Boxen anlegen. Der/die Anwender/in kann sich nur über einzelne Boxen, als auch über alle Boxen, die Statistik anzeigen lassen. Aus dieser geht hervor, wie oft eine Box beziehungsweise alle Boxen an welchem Wochentag gebucht wurde/n und wie oft sie in einem von dem/der Mitarbeiter/in angegebenen Zeitraum insgesamt gebucht wurde/n. Dem/der Benutzer/in wird die Statistik grafisch über Balkendiagramme angezeigt.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben. Wird versucht, mit ungültigen Daten in der Datenbank zu suchen, wird eine Fehlermeldung ausgegeben.

Alternativscenario: Gibt es keine Reservierungen zu einer Box wird im Balkendiagramm nichts angezeigt.

Titel: Preisanpassung

Scope: Verwaltung

Level: User Goal

Aktoren: Mitarbeiter/in

Kurzbeschreibung: Ein/e Mitarbeiter/in erstellt eine Preisanpassung.

Vorbedingung: Mitarbeiter/in startet das Programm.

Hauptszenario: Der/die Mitarbeiter/in kann ausgehend von der Statistik eine Preisanpassung erstellen. Dabei kann der/die Anwender/in entscheiden ob er/sie sich die Preisanpassung, zu der Box mit der niedersten oder höchsten Kapazitätsauslastung, an einem Tag oder in einem angegebenen Zeitraum, anzeigen lassen will. Er/Sie hat weiters die Möglichkeit die Preisanpassung prozentuell oder in absoluten Werten durchzuführen. Dem/die Mitarbeiter/in wird der alte Preis vor der Preisanpassung und der neue Preis nach der Anpassung angezeigt.

Fehlerszenario: Kann keine Verbindung zur Datenbank hergestellt werden, wird eine Fehlermeldung ausgegeben. Wird versucht, mit ungültigen Daten in der Datenbank zu suchen, wird eine Fehlermeldung ausgegeben.

Alternativszenario: Gibt es gar keine Reservierungen wird 0.0 bei den Summen angezeigt.