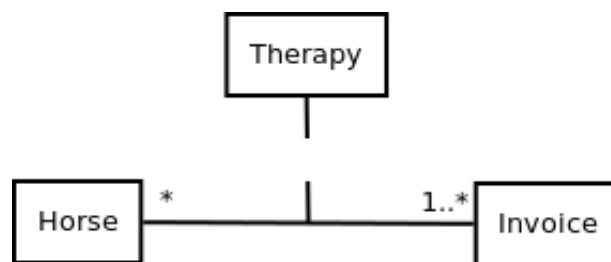
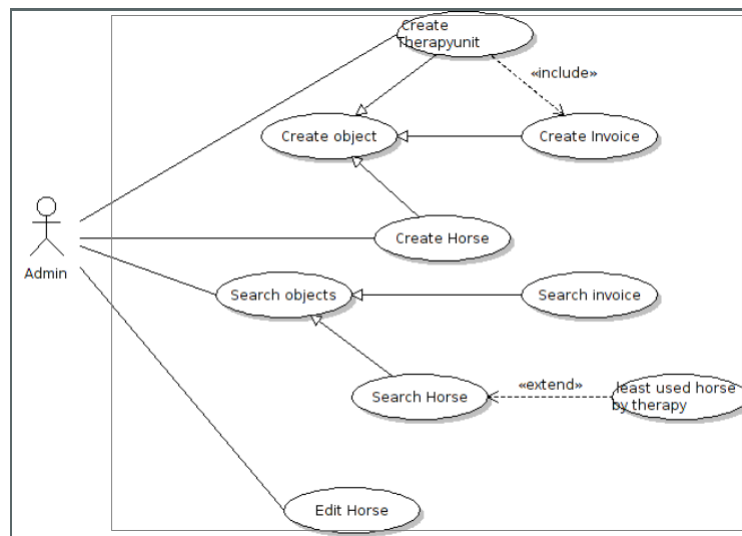

SEPM - Simpel Effiziente Pferdetherapie Manager with Usecase: 3

Maurizio Rinder [mat#:0828852, e0828852@student.tuwien.ac.at]

April 8, 2013



1 Use Cases



1.1 Usecase Description

1. Create Horse

- **Scope:** Creation
- **Actors:** Admin
- **Desc:** The administrator gathers a new horse

-
- **Main Scenario:** The administrator fills in the data needed by the horse entity. This includes the name, a picture, a therapytype and an hourly price. The System/User Interface guides him through the whole data input sequence.
 - **Errors:** Some information were forgotten during the input or the connection to the persistence unit was faulty
 - **Alternatives:** In both cases the administrator gets a feedback saying that it wasn't possible to create the new horse.
 - **Postconditions:** With the next listing of the horses the new horse will be in the list. It is also able find using the search function.

2. Create Therapyunit

- **Scope:** Creation
- **Actors:** Admin
- **Desc:** The administrator is able to create therapyunits which represent a whole therapy

-
- **Main Scenario:** The administrator is able to create a new therapy. For that he has to create therapyunits containing the therapytype, the horse with which the therapy has to be taken, duration and the price. The creation of a therapy involves the creation of new invoice with today's date and several client information.
 - **Errors:** The administrator could forget to give all client credentials to the invoice or the connection or creation of the therapyunits and invoice was faulty
 - **Alternatives:** In both cases the administrator gets a feedback saying that it wasn't possible to create the new Therapy including the invoice. He gets the possibility to correct possible forgotten data inputs and try it again.
 - **Postconditions:** With the creation of the therapy it can be found by listing invoices. From there it will be possible to get full details to the invoice and this will be the listing of the therapyunits.

3. Search Invoice

- **Scope:** Search
- **Actors:** Admin
- **Desc:** The administrator is able to list all invoices and list the details of a particular one

-
- **Precondition:** Invoice is stored in persistence unit.
 - **Main Scenario:** The administrator is able to list all invoices and get a particular one by identifying the invoice-id. For every invoice there is a details view showing all therapyunits.
 - **Errors:** The invoice might not be in the persistence unit and so it can't be found using the invoice id. It is also possible that connecting to the persistence unit is not possible. Another possibility could be that the administrator types in inappropriate data like letters or nothing.
 - **Alternatives:** For the non existence of the invoice there won't be happen anything. The result list remains empty. If there was an error with the persistence unit a feedback will inform the administrator offering to try it again. For last error the administrator will get an error notification telling him what went wrong.

- **Postconditions:** If everything worked fine the administrator will see either a list of invoices or the detailsview of one invoice.

4. Search Horse

- **Scope:** Search
- **Actors:** Admin
- **Desc:** The administrator is able to search for horses using different search attributes.

-
- **Main Scenario:** The administrator can search for horses. For that he can use every horse's attribute as single or combination. In addition to the normal search it is also possible to the horse work load by therapytype and in ascending order. This should give the administrator the opportunity to select the least used horse.
 - **Errors:** It may be possible that the given values result in an empty resultset. The administrator might type in non valid data concerning the dataformat.
 - **Alternatives:** If there is a an empty resultset the administrator will only get an empty resultlist. He is able to reset the search results so that can see all horses again. If the administrator types in non valid data the system will tell him that he has to correct that mistakes before allowing a search.
 - **Postconditions:** Resultlist with horses and their information.

5. Edit Horse

- **Scope:** Editing data
- **Actors:** Admin
- **Desc:** The administrator is able to edit and delete horses.

-
- **Main Scenario:** The administrator can select a horse and has the choice wether to delete or edit it.
 - **Errors:** The administrator didn't choose a horse. The access to the data-source was not able. The administrator left out some values or gave some wrong values during updating a horse.
 - **Alternatives:** In all cases there will be an appropriate error message telling the administator what went wrong.
 - **Postconditions:** Depending on what the administrator wants to do the horse will be either deleted from the resultlist or has new values.

Time Track

activity	start time	end time	minutes	hours
Read through task	2013-03-23 14:52:00	2013-03-23 15:09:00	17	0.28
scratching domainmodel	2013-03-23 15:09:00	2013-03-23 15:57:00	48	0.80
Scratching database schema	2013-03-23 16:15:00	2013-03-23 17:05:00	50	0.83
create testdata	2013-03-23 18:06:00	2013-03-23 20:07:00	121	2.02
scratching classdiagram domain	2013-03-23 20:44:00	2013-03-23 21:25:00	41	0.68
scratching classdiagram_DAO	2013-03-24 14:09:10	2013-03-24 15:30:11	81	1.35
domain and dao objects	2013-03-25 13:31:33	2013-03-25 14:48:37	77	1.28
domain and dao objects	2013-03-25 17:36:35	2013-03-25 18:31:37	55	0.92
domain and dao objects	2013-03-25 19:15:14	2013-03-25 20:36:51	81	1.35
domain and dao objects	2013-03-26 15:17:10	2013-03-26 16:23:56	66	1.10
domain and dao objects	2013-03-26 16:55:05	2013-03-26 20:35:04	219	3.65
remanaging planning	2013-03-27 16:43:50	2013-03-27 18:45:29	121	2.02
domain and dao objects refactoring	2013-03-28 19:29:04	2013-03-28 23:24:39	235	3.92
Configure_log4j	2013-03-30 19:20:25	2013-03-30 20:05:10	44	0.73
Logging_HorseDAO	2013-03-30 20:05:25	2013-03-30 21:49:02	103	1.72
Testing	2013-03-31 01:21:17	2013-03-31 03:44:25	143	2.38
domain and dao objects	2013-04-02 14:17:00	2013-04-02 18:47:00	270	4.50
Documentation_finalizing				
usecase_diagram	2013-04-02 22:41:00	2013-04-02 23:52:00	71	1.18
ServiceLayer	2013-04-03 22:26:00	2013-04-04 02:22:59	236	3.93
ServiceLayer finalizing	2013-04-05 18:31:45	2013-04-05 21:20:09	168	2.80
Scratching_gui	2013-04-05 22:27:54	2013-04-05 23:32:00	64	1.07
implementing_ui	2013-04-05 23:58:39	2013-04-06 02:43:15	164	2.73
implementing_ui	2013-04-06 14:04:00	2013-04-06 16:34:00	150	2.50
implementing_ui	2013-04-06 17:46:09	2013-04-06 21:59:02	252	4.20
implementing_ui	2013-04-07 01:42:31	2013-04-07 03:28:34	106	1.77
implementing_ui	2013-04-07 15:12:41	2013-04-07 18:20:00	187	3.12
implementing_ui	2013-04-07 18:46:55	2013-04-07 22:02:02	195	3.25
implementing_ui	2013-04-07 22:42:06	2013-04-08 03:21:21	279	4.65
detailswork_and_fixes	2013-04-08 15:27:22	2013-04-08 19:31:38	244	4.07
detailswork_and_fixes	2013-04-08 19:55:26	2013-04-08 20:48:20	52	0.87
			3940	65.67