College of Computing Technology
Faculty of Information Technology
Object Oriented Constructs

College Dublin
Computing • IT • Business

Application Development Project - CCTZoo

Miguelantonio Guerra - 2016324

Group B

Lecturer: Profesor Mark Morrissey

May 13, 2018

Contents

List of Figures

Chapter I.

The Problem

1.1. Problem

We have been tasked to create a Zoo management system. The Zoo has a number of Animals. These Animals are broken down into types:

- Mammal
- Reptile
- Avian
- Aquatic
- Insect

Each Animal has a Zoo keeper that looks after it. Zoo keepers are only allowed to care for animals if they are qualified to do so. A zoo keeper can look after a max of 3 Animal types for a max of 10 animals.

Example Animals:

- Mammal – Tiger (date of birth/date of arrival/gender/offspring/ medication/vaccine/exhibit number)
- Mammal – avian - Bat (date of birth/date of arrival/ flight/gender/offspring/medication/ vaccine/exhibit number).

Our system must be run on test data before the Zoo will accept it. You are required to have a data set of at least 100 animals and 40 zoo keepers .

The system must allow a user to:

- Search for Animals.
- Search for Keepers.
- Add new animals.
- Add new keepers.

- Update animals.
- Update keepers.

1.2. Solution to Problem

Create a user friendly interface that can guide the user through the different functionality that our system will have. Here we set up to create different views on which the user can manipulate the date generated in the model of our system.

We are going to have two sets of data:

- Animals
- Keepers

And these, by the user, are going to be:

- Viewed
- Updated
- Searched
- Incremented (Added)

Chapter II.

Individual Role within the Project

2.1. View of the system

After dividing the task of the system, I was tasked to do the viewing and controller of the system while Asmer and Adelo were going to do the search engine and the model of the system.

For the view the following classes were design:

- For the animal:
  - AddAnimalFrame (see figure 2.1)

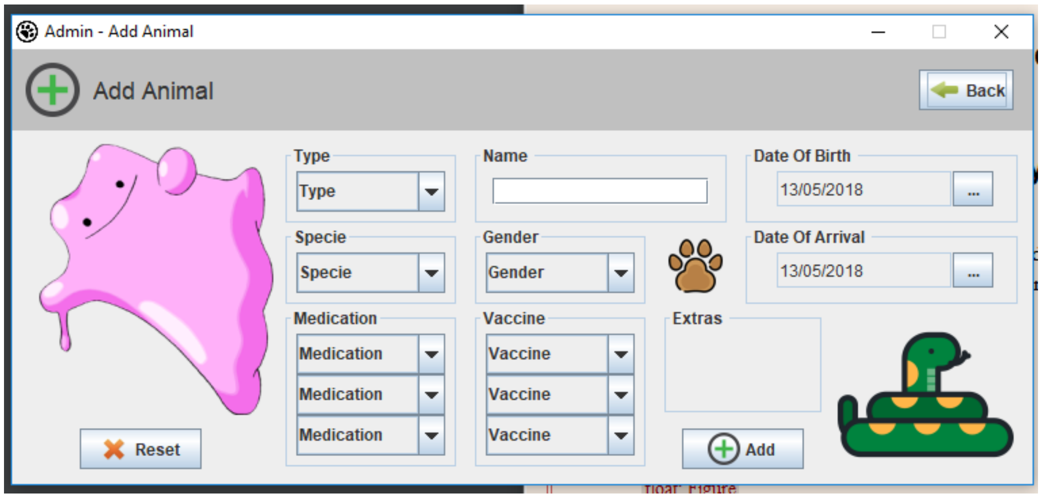
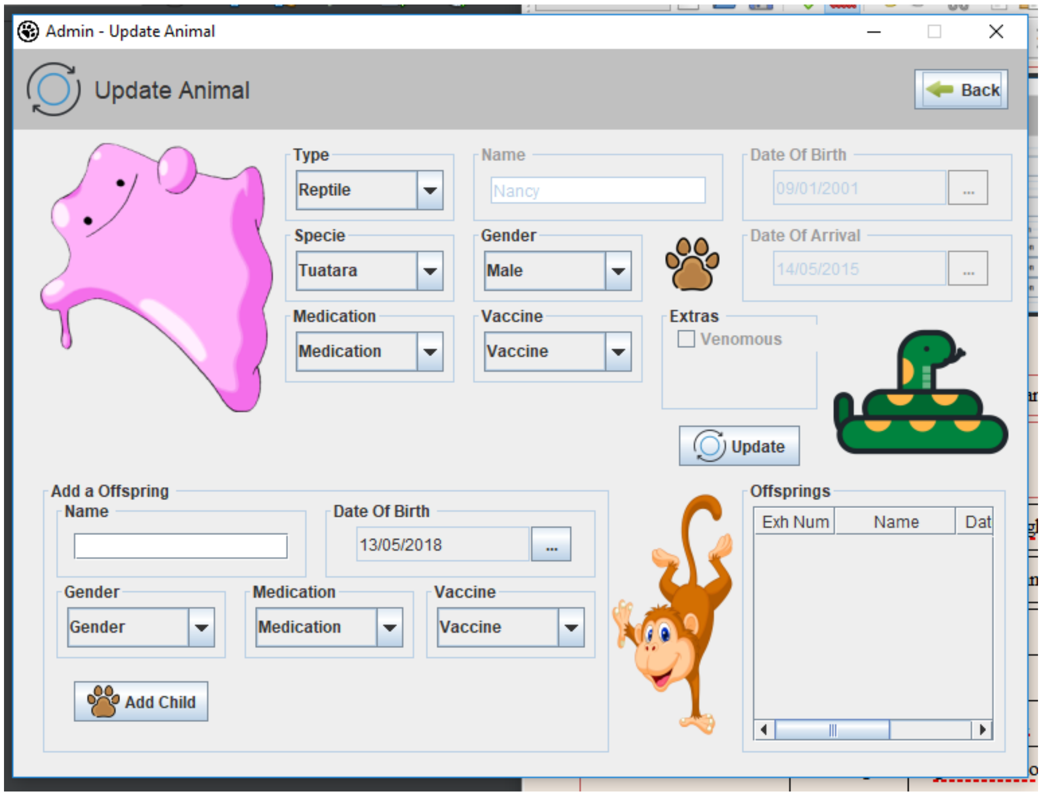


Figure 2.1.: Frame to add an animal to the system.

  - UpdateAnimalFrame (See figure 2.2)


Admin - Update Animal

Update Animal

Back

Type: Reptile

Name: Nancy

Date Of Birth: 09/01/2001

Specie: Tuatara

Gender: Male

Date Of Arrival: 14/05/2015

Medication: Medication

Vaccine: Vaccine

Extras: Venomous

Update

Add a Offspring

Name:

Date Of Birth: 13/05/2018

Gender: Gender

Medication: Medication

Vaccine: Vaccine

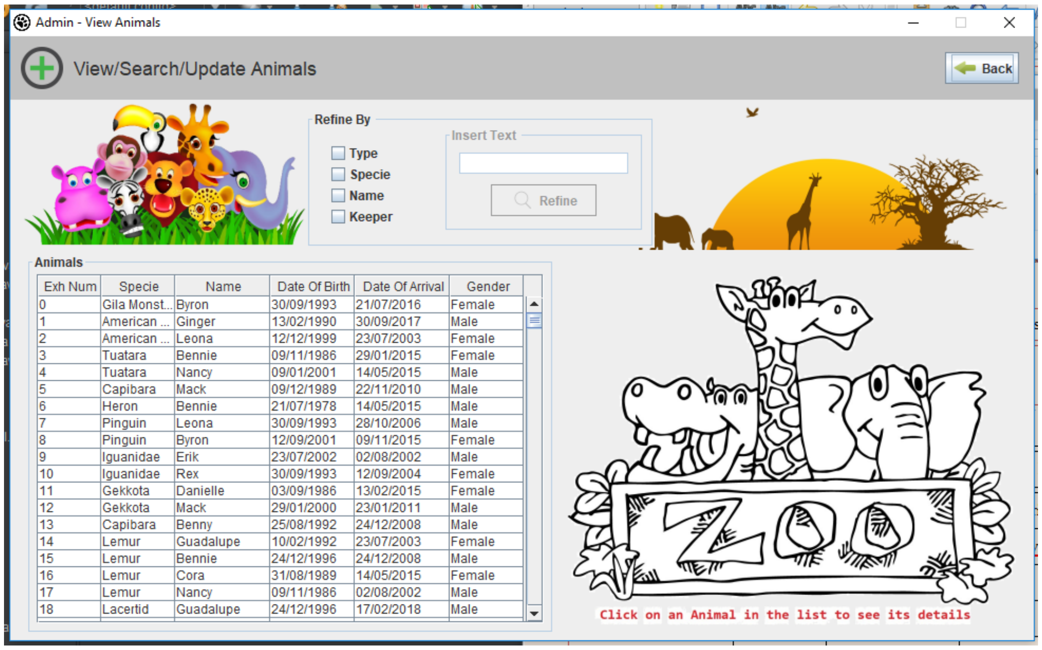
Add Child

Offsprings

Exh Num	Name	Dat

Figure 2.2.: Frame to update an animal to the system.

– ViewAniamlsFrame (See figure 2.3)


Admin - View Animals

View/Search/Update Animals

Back

Refine By: Type, Specie, Name, Keeper

Insert Text

Refine

Animals

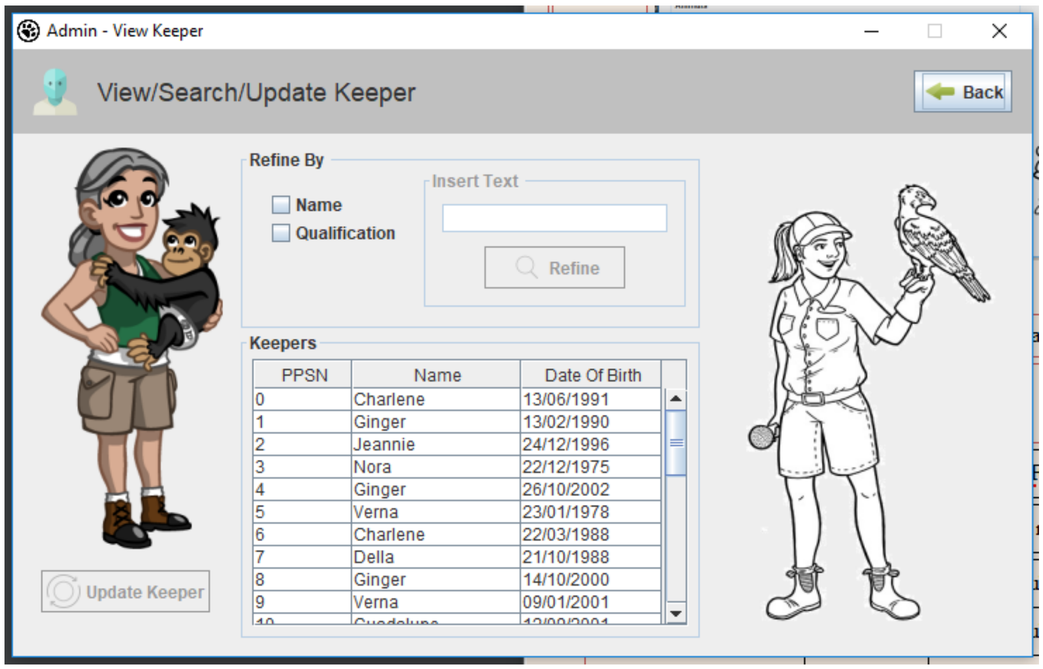
Exh Num	Specie	Name	Date Of Birth	Date Of Arrival	Gender
0	Gila Monst...	Byron	30/09/1993	21/07/2016	Female
1	American ...	Ginger	13/02/1990	30/09/2017	Male
2	American ...	Leona	12/12/1999	23/07/2003	Female
3	Tuatara	Bennie	09/11/1986	29/01/2015	Female
4	Tuatara	Nancy	09/01/2001	14/05/2015	Male
5	Capibara	Mack	09/12/1989	22/11/2010	Male
6	Heron	Bennie	21/07/1978	14/05/2015	Male
7	Pinguin	Leona	30/09/1993	28/10/2006	Male
8	Pinguin	Byron	12/09/2001	09/11/2015	Female
9	Iguanidae	Erik	23/07/2002	02/08/2002	Male
10	Iguanidae	Rex	30/09/1993	12/09/2004	Female
11	Gekkota	Danielle	03/09/1986	13/02/2015	Female
12	Gekkota	Mack	29/01/2000	23/01/2011	Male
13	Capibara	Benny	25/08/1992	24/12/2008	Male
14	Lemur	Guadalupe	10/02/1992	23/07/2003	Female
15	Lemur	Bennie	24/12/1996	24/12/2008	Male
16	Lemur	Cora	31/08/1989	14/05/2015	Female
17	Lemur	Nancy	09/11/1986	02/08/2002	Male
18	Lacertid	Guadalupe	24/12/1996	17/02/2018	Male

Figure 2.3.: Frame to view animals of the system.

And the following classes were generated to help build all the views:

- AddOffspringPanel
- ViewAnimalCard
- ViewAnimalsPanel
- ViewOffspringsPanel

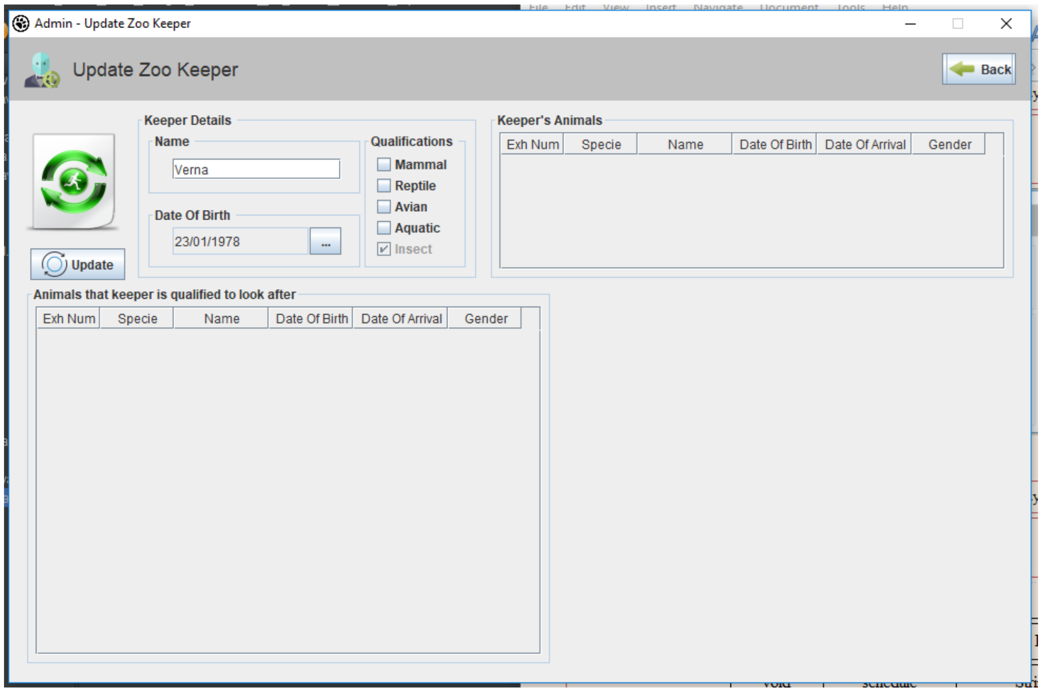
- For the Keepers:
  - ViewKeepersFrame (See figure 2.4)



PPSN	Name	Date Of Birth
0	Charlene	13/06/1991
1	Ginger	13/02/1990
2	Jeannie	24/12/1996
3	Nora	22/12/1975
4	Ginger	26/10/2002
5	Verna	23/01/1978
6	Charlene	22/03/1988
7	Della	21/10/1988
8	Ginger	14/10/2000
9	Verna	09/01/2001
10	[illegible]	[illegible]

Figure 2.4.: Frame to view keepers of the system.

  - UpdateKeepersFrame (See figure 2.5):



Admin - Update Zoo Keeper

Update Zoo Keeper

Keeper Details

Name: Verna

Date Of Birth: 23/01/1978

Qualifications: Mammal [ ], Reptile [ ], Avian [ ], Aquatic [ ], Insect [✓]

Keeper's Animals

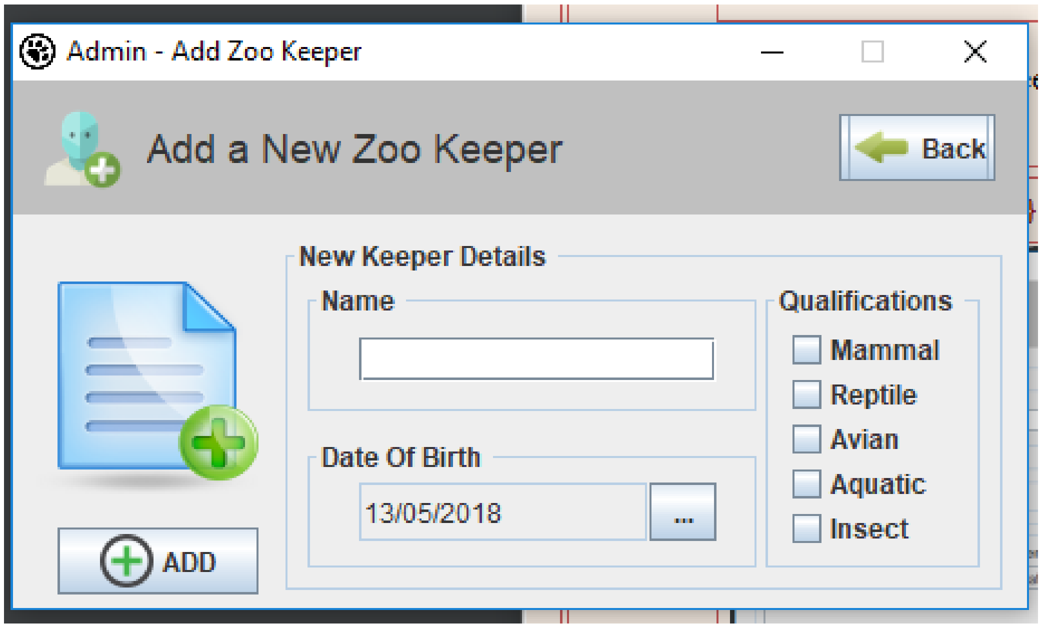
Exh Num	Specie	Name	Date Of Birth	Date Of Arrival	Gender

Animals that keeper is qualified to look after

Exh Num	Specie	Name	Date Of Birth	Date Of Arrival	Gender

Figure 2.5.: Frame to update keepers to the system.

– AddKeepersFrame (See figure 2.6)



Admin - Add Zoo Keeper

Add a New Zoo Keeper

New Keeper Details

Name:

Date Of Birth: 13/05/2018

Qualifications: Mammal [ ], Reptile [ ], Avian [ ], Aquatic [ ], Insect [ ]

Figure 2.6.: Frame to add keepers to the system.

And the following classes to help build this windows:

- ViewKeepersPanel

- ViewKeepersCard

And, I realized that there were going to be elements that were going to be common for all the views, so the following classes were defined:

- CalendarPanel
- ChackBoxPanel
- ComboBoxFieldsPanel
- GenericButton
- Header
- ImageLabel
- MainFrame
- RefineByPanel
- TextFieldPanel

All this classes, but MainFrame, are panels, labels or buttons that contains more elements inside (images, text, tool tip text, etc). MainFrame class is extended by all the views of the system, because all have a common frame parent that extends JFrame.

Also, it was created a ManagersDashboard view to allow the user to move between all the view and functionality of it, see figure 2.7.

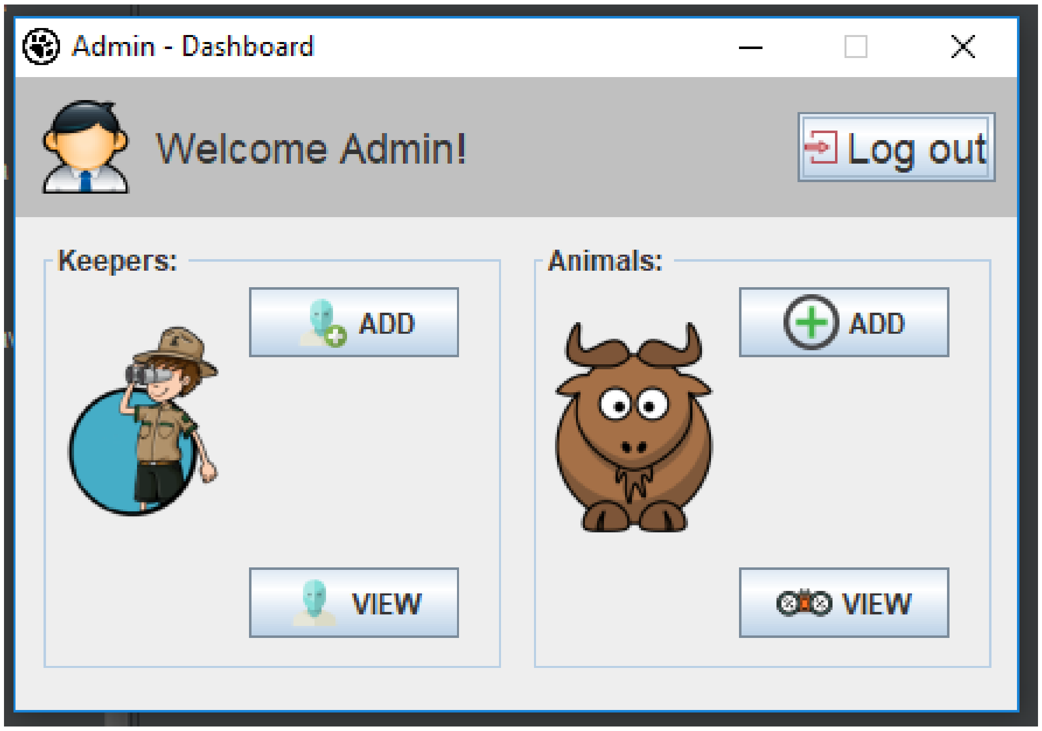


Figure 2.7.: Frame to add keepers to the system.

2.2. Controller of the system

Now, when controlling the view, I created one class that controls each one of them, as listed below:

- AdminDashboardController controls AdminDashBoard
- AddAnimalController controls AddAnimalFrame
- UpdateAnimalController controls UpdateAnimalFrame
- ViewAnimalsController controls ViewAnimalsFrame
- AddKeeperController controls AddKeeperFrame
- UpdateKeeperController controls UpdateKeeperFrame
- ViewKeepersController controls ViewKeepersFrame

Chapter III.

Conclusion

Even though every one of us had its own tasks to develop this program it was not going to go anywhere if we did not set rules at the beginning for the designing of the model, the view and the controller. This way of programming gives us a common goal to seek without every one following its own path and getting a common mess at the end of it.

The design of general classes helped us create frames and components in a fast and easy manner to be able to control every one of the from their respective classes.

At starters I did not create any general class but ext time it will be better to examine the problem as a whole from the beginning to see and create all this common classes of each view.