

Zoo Management System

Harshit Maheshwari
Indian Institute of Technology, Kanpur
Computer Science and Engineering
harshitm@iitk.ac.in

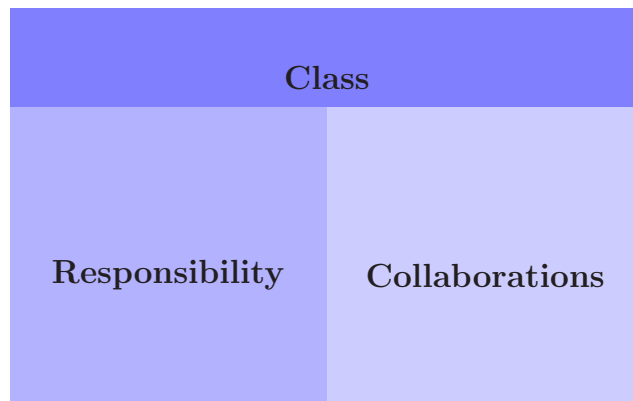


Figure 1.1: CRC model

1 CLASS-RESPONSIBILITY-COLLABORATION MODEL

CRC model is a very useful model used by developers to interact and work closely together to understand the needs of users and design a preliminary software specification which acts as the basis for software development phase. A CRC model looks like shown in 1.1. It consists of

1. **Class** It is simply the class name.
2. **Responsibility** It consists of the functions and responsibilities of the class.
3. **Collaborations** It consists of all the other classes that the class need in order to complete it's responsibilities.

For example, in 2.1 the class **Animals** has the **Responsibility** of eating, entertaining, excretion, sleeping and mating. In order to complete these responsibilities it has to **Collaborate** with Visitors, Admin-Animal and Cages.

2 CRC MODEL FOR ZOO

While designing the CRC model for zoo we have ensured that the use of OO principles is maximized. Also, special focus has been given to code reuse and integrity of the system. The zoo model has been designed into 4 main parts:

2.1 ANIMALS

The CRC model is as shown in 2.1. As evident from the figure, there are two child classes of Animal class: Herbivores and Carnivores. Such a division is made because both the classes have same responsibilities as reflected in Animal class except the function of eating which we override in the child classes.

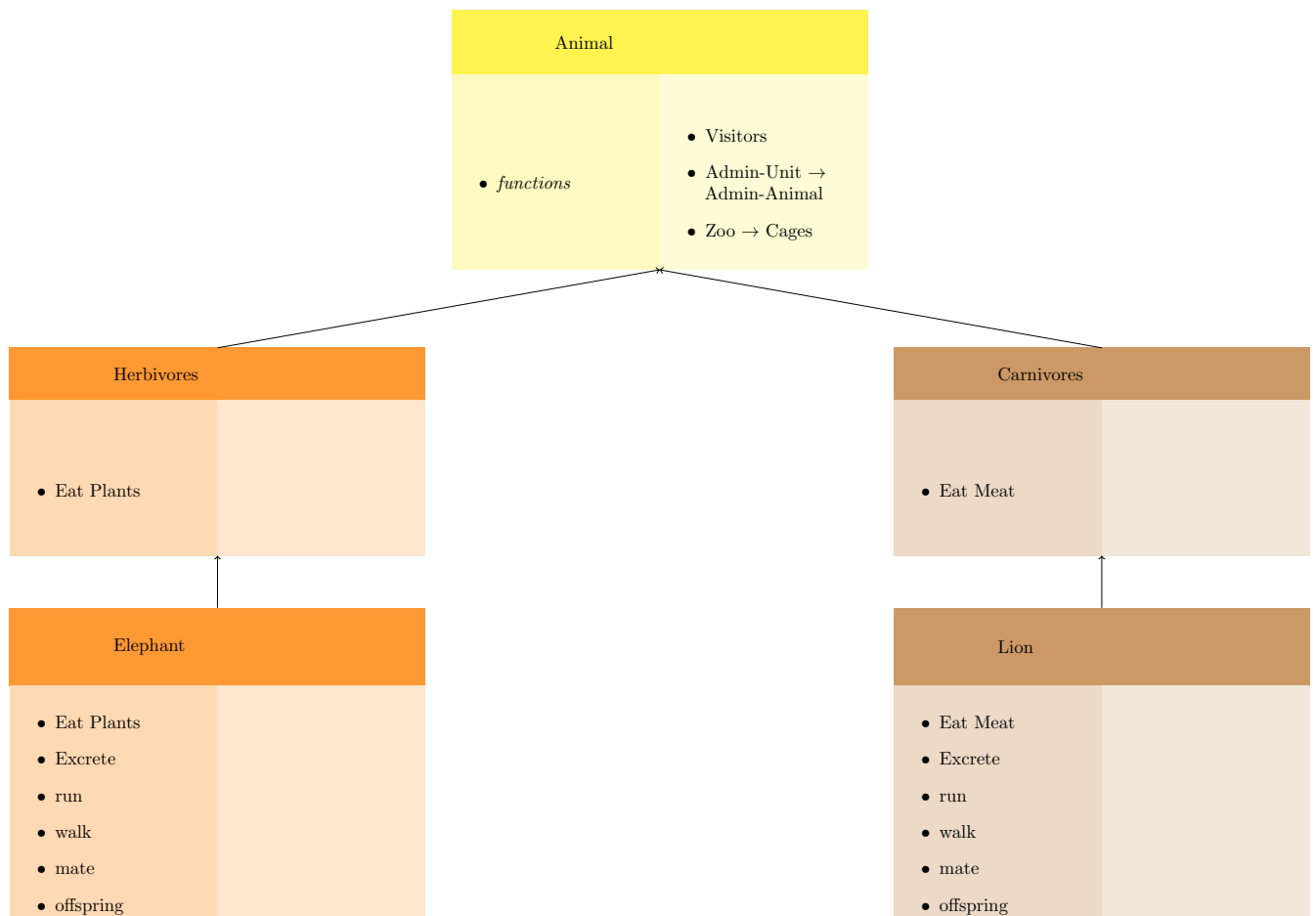


Figure 2.1: CRC Model: Animals

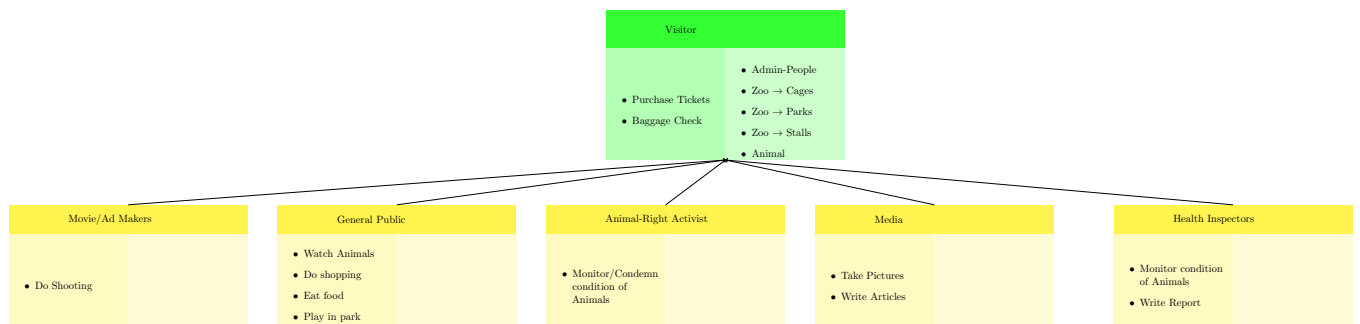


Figure 2.2: CRC Model: Visitors



Figure 2.3: CRC Model: Administrative Unit

2.2 VISITORS

The CRC model is as shown in 2.2. Visitors to zoo can be divided into the following types: Animal Right Activists, General Public, Health Inspectors, Media and Movie/Ad makers. Accordingly, these classes are all child classes of parent class Visitors. They all have some different functions(read responsibilities) which is reflected in 2.2.

2.3 ADMINISTRATIVE UNIT

The CRC model is as shown in 2.3. The administrative unit of any zoo can be classified into the following 3 sub-types: Animal related Administration, Cleaners and Administration for people. These division have been made based on the Collaborations these classes have with other classes. The Admin-animal class is solely collaborating with Animal class, Cleaner class is collaborating with ZooStructure Class and Admin-People class collaborates with Visitors class. These classes again have child classes based on their different responsibilities as shown in 2.3.

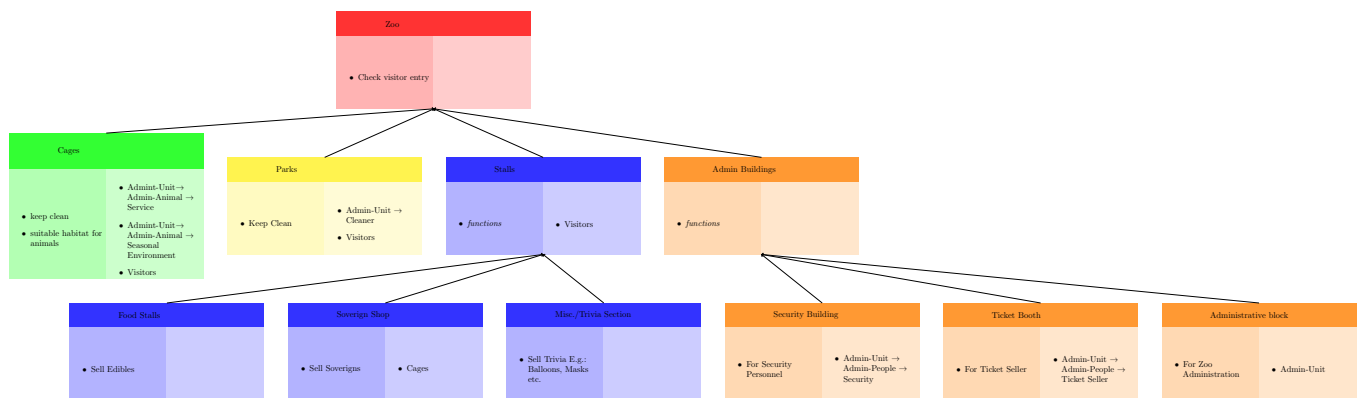


Figure 2.4: CRC Model: Zoo Structure

2.4 ZOO STRUCTURE

This part reflects the physical layout of zoo. The CRC model is as shown in 2.4. A zoo has the following physical infrastructure: cages for animals, parks, stalls and administrative buildings. The stalls can be of different types too: food stalls, sovereign shops and trivia shops. Similarly, the administrative building can be sub-divided into the following child classes: Security building, ticket booths and a main administrative block(read building).

3 CODE STRUCTURE

The entire code is available at: <https://github.com/harshitm26/Tool-Assignments/tree/master/ZooManagementSystem/src>

The name of the class files is self suggestive of the functionality of the class. The following are the code files:

1. Zoo.java
2. **ZooManagementSystem.java** - This is the test file which contains the main method demonstrating the use of the Zoo Management System Code.
3. AdministrativeUnit/AdminAnimal.java
4. AdministrativeUnit/AdminUnit.java
5. AdministrativeUnit/AdminPeople.java
6. AdministrativeUnit/Cleaner.java
7. AdministrativeUnit/Doctor.java
8. AdministrativeUnit/Feeder.java

9. AdministrativeUnit/SeasonalEnvironmentalList.java
10. AdministrativeUnit/Security.java
11. AdministrativeUnit/Service.java
12. AdministrativeUnit/Ticket.java
13. Animals/Animals.java
14. Animals/Carnivores.java
15. Animals/Elephant.java
16. Animals/Herbivores.java
17. Animals/Lion.java
18. Human/Human.java
19. Interface/Eat.java
20. Interface/Fly.java
21. Interface/Reproduce.java
22. Interface/Run.java
23. Interface/Swim.java
24. Visitors/AnimalRightActivist.java
25. Visitors/GeneralPublic.java
26. Visitors/HealthInspectors.java
27. Visitors/Media.java
28. Visitors/MovieMakers.java
29. Visitors/Visitors.java
30. ZooStructure/AdminBuilding.java
31. ZooStructure/Cages.java
32. ZooStructure/FoodStall.java
33. ZooStructure/Parks.java
34. ZooStructure/SecurityBuilding.java
35. ZooStructure/SovereignShop.java

- 36. ZooStructure/Stalls.java
- 37. ZooStructure/TicketBooth.java
- 38. ZooStructure/Trivia.java
- 39. ZooStructure/ZooStructure.java