|  |  |
| --- | --- |
|  |  |



**Dharmsinh Desai University, Nadiad**

**Faculty of Technology**

Department of Computer Engineering

B. Tech. CE Semester – VI

Subject: Object Oriented Software Engineering

Project Title: Zoo Management System

By:

1)Deval Italiya, Roll no : CE-044, ID : 17CEUOS070

2)Prince Jodhani, Roll no : CE-049, ID : 17CEUON062

**Guided By :** Associate Prof. Jatayu Baxi



**Dharmsinh Desai University, Nadiad**

Faculty of Technology, Department of Computer Engineering

**CERTIFICATE**

This is to certify that Object Oriented Software Engineering entitled “Zoo Management System” is the bonafied report of work carried out by

1)Deval Italiya (17CEUOS070)

2)Prince Jodhani (17CEUON062)

Of Department of Computer Engineering ,Semester VI , academic year 2019-2020, under our supervision and guidance.

Guide HOD

**Prof. Jatayu Baxi** **Dr. C. K. Bhensdadia**

Associate Professor of Department of Head of Department of Computer

Computer Engineering, Dharmsinh Engineering, Dharmsinh Desai

Desai University, Nadiad University, Nadiad

# Contents:

**1. Abstract 4**

**2. Introduction 5**

1. Brief Introduction 5

2. Technology/Platform/Tools used 5

**3. Software Requirement Specifications 6**

**4. Design 11**

I. ER diagram 11

II. Database Schema. 12

III. Class diagram 14

IV. Activity diagram 15

V. Sequence diagram 17

Vl. Use case diagram 20

**5. Implementation Detail 23**

I. Modules created and brief description of each modules. 23

**6. Screen-Shots 25**

**7. Limitations of System 31**

**8. Future Extensions to the System 31**

**9. Conclusion 32**

**Abstract**

The zoo management system is created for the keep the record of all the animals, vets and caretakers. And to manage the data of animals. The zoo management system is managed by the manager. The system is created in C#. And the platform used is visual studio.

**Introduction**

In this system the data of animals, vets and caretakers is managed by the manager. Manager can add, update and delete the data of animal, vets and caretakers. Manager also can generate the cleaning report. The work of the vets is manage other reports. Vets can create food report and health report. Vets can create this report according to the ids of the animal.

**Technology/Platform**

This system is created in C# language and the platform which is used is visual studio.

**Software Requirement Specification**

=>Types of Users

1). Manager

2). Caretaker

3). Veterinarian

**1. Manager**

R 1.1 Manage Animals

Description: Manager should be able to add, update and delete the data of animals.

R 1.1.1 Add Animal Data

Description: Manager can add data of new animal.

Input: Animal Information

Output: Added successfully

R 1.1.2 Update Animal Data

Description: Manager can edit data of existing animal.

Input: Edit Animal Data

Output: Updated successfully

R 1.1.3 Delete Animal Data

Description: Manager can delete the data of existing animal.

Input: Delete Animal Data

Output: Deleted successfully

R 1.2 Manage Caretakers

Description: Manager should be able to add, update and delete the data of caretakers.

R 1.2.1 Add Caretaker Data

Description: Manager can add data of new caretaker.

Input: Caretaker Information

Output: Added successfully

R 1.2.2 Update Caretaker Data

Description: Manager can edit data of existing caretaker.

Input: Edit Caretaker Data

Output: Updated successfully

R 1.2.3 Delete Caretaker Data

Description: Manager can delete the data of existing Caretaker.

Input: Delete Caretaker Data

Output: Deleted successfully

R 1.3 Manage Veterinarians

Description: Manager should be able to add, update and delete the data of veterinarians.

R 1.3.1 Add Veterinarian Data

Description: Manager can add data of new veterinarian.

Input: Veterinarian Information

Output: Added successfully

R 1.3.2 Update Veterinarian Data

Description: Manager can edit data of existing veterinarian.

Input: Edit veterinarian Data

Output: Updated successfully

R 1.3.3 Delete Veterinarian Data

Description: Manager can delete the data of existing veterinarians.

Input: Delete Veterinarian Data

Output: Deleted successfully

R 1.4 Manage Reports

Description: Manager should be able to store data in a way that allows to pull reports about animals combining the information entered by caretakers and veterinarians. The report is about the health checkup, feeding schedule and cleaning.

R 1.4.1 View Health Checkup

Description: Manager should be able to view health checkup report on the base of information given by caretaker and veterinarian.

Input: Click on heath checkup report button

Output: Shows health checkup report

R 1.4.2 View Feeding schedule

Description: Manager should be able to generate feeding schedule report on the base of information given by caretaker and veterinarian.

Input: Click on the schedule of feeding

Output: Shows feeding schedule

R 1.4.2 Manage Cleaning Report

Description: Manager should be able to generate, update and delete cleaning report on the base of information given by caretaker and veterinarian.

R 1.4.2.1 Create Cleaning Report

Description: Manager should be able to create cleaning report

based on the available caretakers.

Input: Add the timing of cleaning

Output: Cleaning report created successfully

R 1.4.2.2 Update Cleaning Report

Description: Manager should be able to update cleaning report

based on the available caretakers.

Input: Update the timing of cleaning

Output: Cleaning report updated successfully

R 1.4.2.3 Delete Cleaning Report

Description: Manager should be able to delete cleaning report

based on the available caretakers.

Input: Delete the timing of cleaning

Output: Cleaning report deleted successfully

R 1.5 Assign Work

Description: Manager should be able to assign the work to caretaker and veterinarian according to their qualification.

R 1.5.1 Assign Work to Caretaker

Description: Manager should be able to assign the work to caretakers according to their qualification.

Input: Input caretaker id and work

Output: Work assign to caretaker successfully

R 1.5.2 Assign Work to Veterinarian

Description: Manager should be able to assign the work to veterinarians according to their qualification.

Input: Input veterinarian id and work

Output: Work assign to veterinarian successfully

**2. Caretaker**

R 2.1 View Animal Data

Description: Caretaker should able to view the data of animals.

Input: Click on view animals data button

Output: Shows the animals data

R 2.2 View Cleaning Report

Description: Caretaker should be able to view cleaning report on the base of information given by manager.

Input: Click on cleaning report button

Output: Shows cleaning report

R 2.3 View Feeding schedule

Description: Caretaker should be able to generate feeding schedule report on the base of information given by veterinarian.

Input: Click on the schedule of feeding

Output: Shows feeding schedule

**3. Veterinarian**

R 3.1 View Animal Data

Description: Veterinarian should able to view the data of animals.

Input: Click on view animals data button

Output: Shows the animals data

R 3.2 Manage Health Checkup Report

Description: Veterinarian should be able to create, update and delete animals health checkup report.

R 3.2.1 Create Health Checkup Report

Description: Veterinarian should be able to create health checkup report.

Input: Add the data of animal health

Output: Health checkup report created successfully

R 3.2.2 Update Health Checkup Report

Description: Veterinarian should be able to update health checkup report.

Input: Update the data of animal health

Output: Health checkup report updated successfully

R 3.2.3 Delete Health Checkup Report

Description: Veterinarian should be able to delete health checkup report.

Input: Delete the data of animal health

Output: Health checkup report deleted successfully

R 3.3 Manage Feeding Schedule

Description: Veterinarian should be able to create, update and delete animals feeding schedule.

R 3.3.1 Create Feeding Schedule

Description: Veterinarian should be able to create feeding schedule.

Input: Add the data of animal feeding schedule

Output: feeding schedule created successfully

R 3.3.2 Update Feeding Schedule

Description: Veterinarian should be able to update feeding schedule.

Input: Update the data of animal feeding schedule

Output: feeding schedule updated successfully

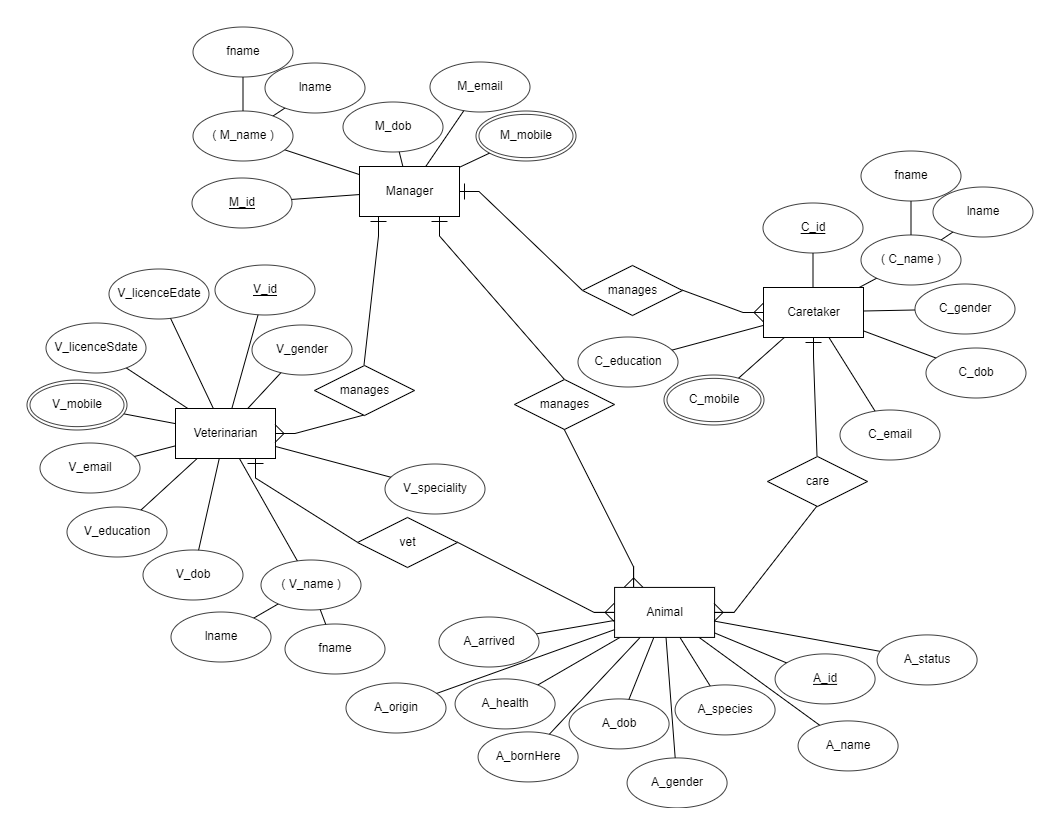
R 3.3.3 Delete Feeding Schedule

Description: Veterinarian should be able to delete feeding schedule.

Input: Delete the data of animal feeding schedule

Output: feeding schedule deleted successfully

**ER Diagram**



**DATABASE SCHEMA**

**Zoo Management System**

**MANAGER**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr\_no.** | **Name** | **Data Type** | **Width** | **Required** | **Unique** | **PK/FK** | **Referenced Table** |
| **1** | **M\_Id** | **INT** | **20** | **YES** | **YES** | **PK** | **-** |
| **2** | **M\_Name** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |
| **3** | **M\_dob** | **DATE** | **-** | **YES** | **-** | **-** | **-** |
| **4** | **M\_email** | **VARCHAR2** | **50** | **YES** | **YES** | **-** | **-** |
| **5** | **M\_mobile** | **NUMBER** | **10** | **YES** | **YES** | **-** | **-** |

**VETERINARIAN**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr\_no.** | **Name** | **Data Type** | **Width** | **Required** | **Unique** | **PK/FK** | **Referenced Table** |
| **1** | **V\_Id** | **INT** | **20** | **YES** | **YES** | **PK** | **-** |
| **2** | **V\_Name** | **VARCHAR2** | **50** | **YES** |  | **-** | **-** |
| **3** | **V\_dob** | **DATE** | **-** | **YES** | **-** | **-** | **-** |
| **4** | **V\_education** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |
| **5** | **V\_email** | **VARCHAR2** | **50** | **YES** | **YES** | **-** | **-** |
| **6** | **V\_mobile** | **NUMBER** | **10** | **YES** | **YES** | **-** | **-** |
| **7** | **V\_licenceSdate** | **DATE** | **-** | **YES** | **-** | **-** | **-** |
| **8** | **V\_licenceEdate** | **DATE** | **-** | **YES** | **-** | **-** | **-** |
| **9** | **V\_Gender** | **VARCHAR2** | **10** | **YES** | **-** | **-** | **-** |
| **10** | **V\_Speciality** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |

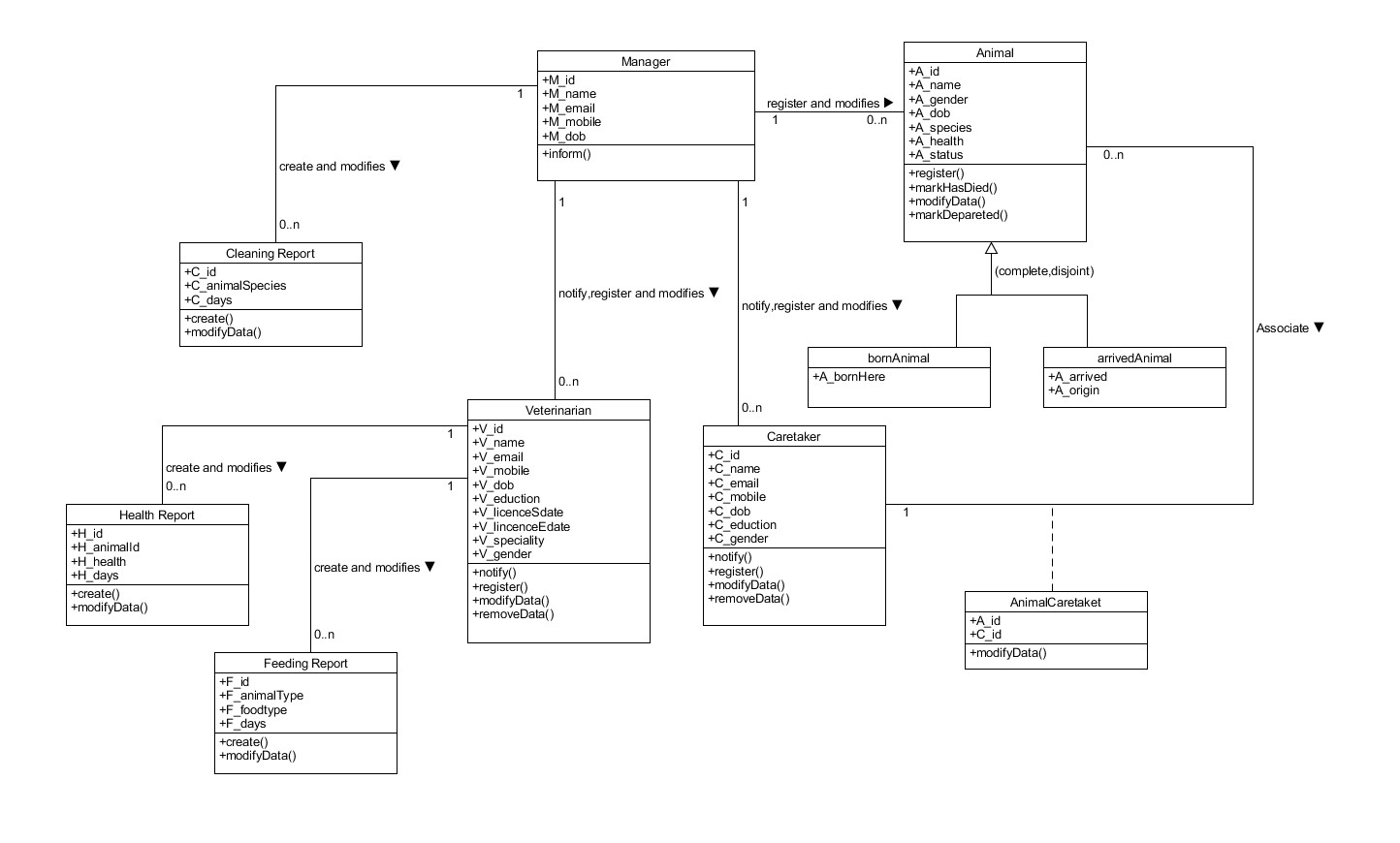
**CARETAKER**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr\_no.** | **Name** | **Data Type** | **Width** | **Required** | **Unique** | **PK/FK** | **Referenced Table** |
| **1** | **C\_Id** | **INT** | **20** | **YES** | **YES** | **PK** | **-** |
| **2** | **C\_Name** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |
| **3** | **C\_Gender** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |
| **4** | **C\_DOB** | **DATE** | **-** | **YES** | **-** | **-** | **-** |
| **5** | **C\_Email** | **VARCHAR2** | **50** | **YES** | **YES** | **-** | **-** |
| **6** | **C\_Mobile** | **NUMBER** | **10** | **YES** | **YES** | **-** | **-** |
| **7** | **C\_Education** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |

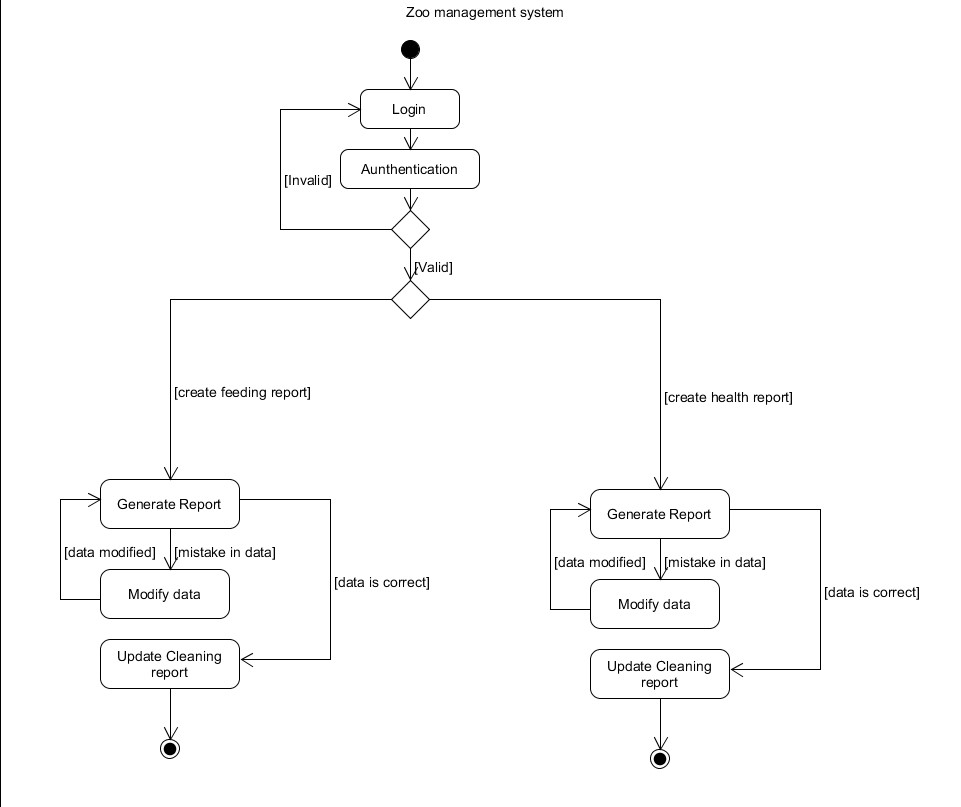
**ANIMAL**

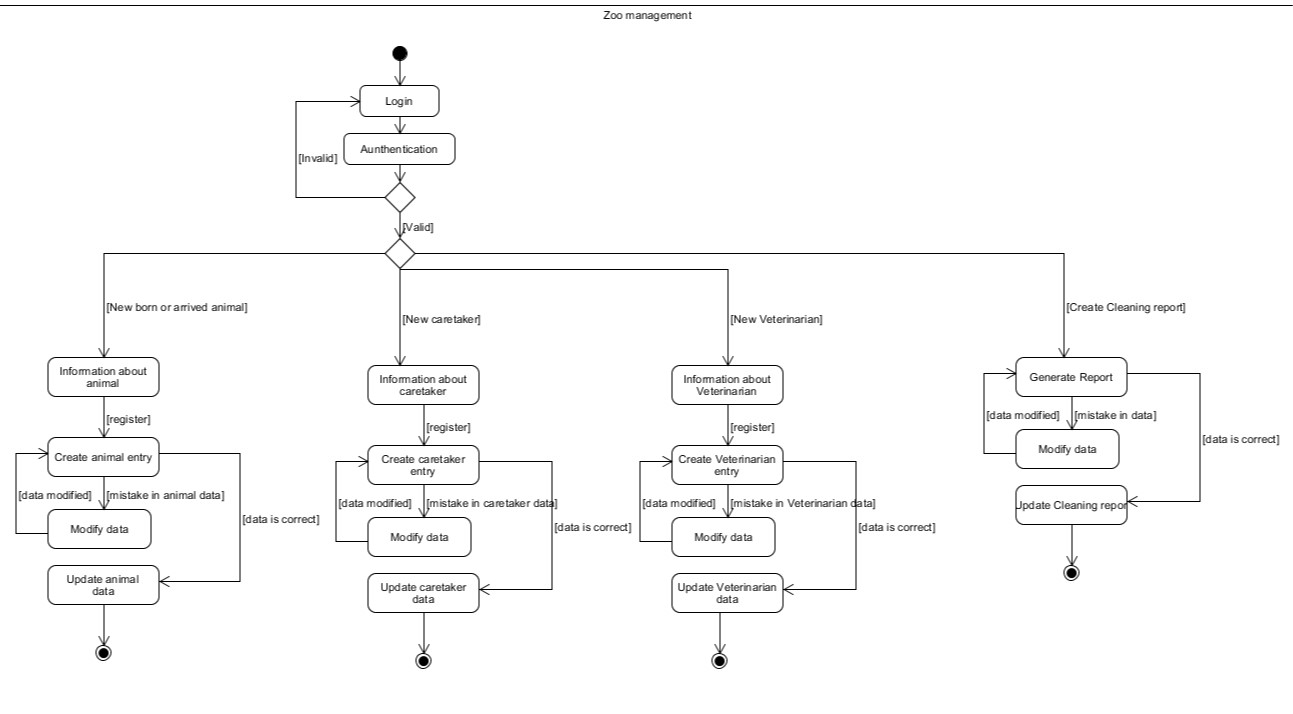
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr\_no.** | **Name** | **Data Type** | **Width** | **Required** | **Unique** | **PK/FK** | **Referenced Table** |
| **1** | **A\_Id** | **INT** | **20** | **YES** | **YES** | **PK** | **-** |
| **2** | **A\_Name** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |
| **3** | **A\_Status** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |
| **4** | **A\_Species** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |
| **5** | **A\_Gender** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |
| **6** | **A\_DOB** | **DATE** | **-** | **YES** | **-** | **-** | **-** |
| **7** | **A\_bornORarrived** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |
| **8** | **A\_Health** | **VARCHAR2** | **50** | **YES** | **-** | **-** | **-** |

**Class Diagram**



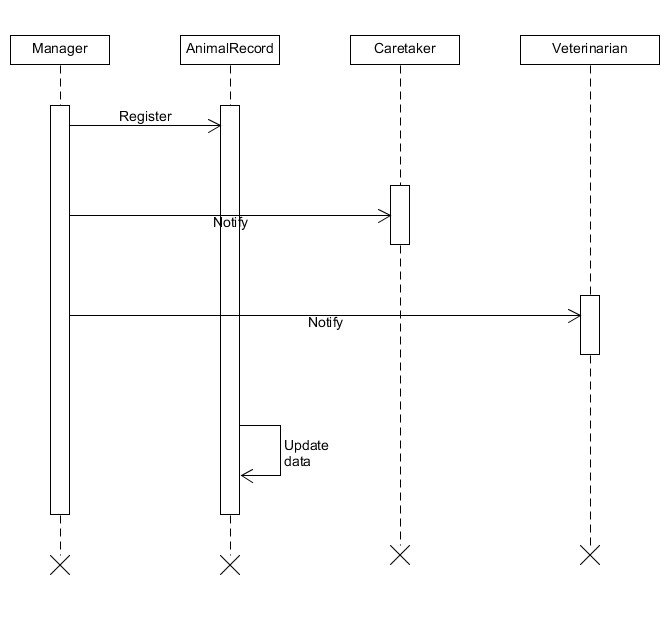
**Activity Diagram**



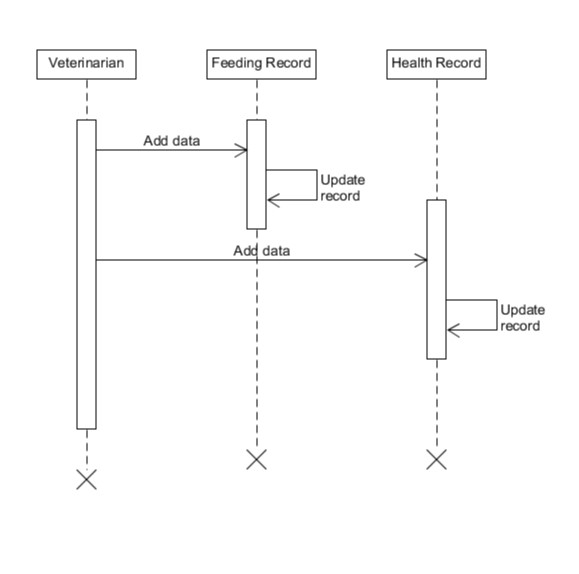


**Sequence Diagram**

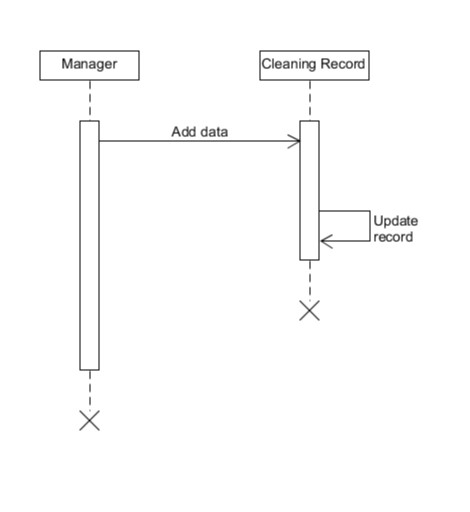
**(1)**



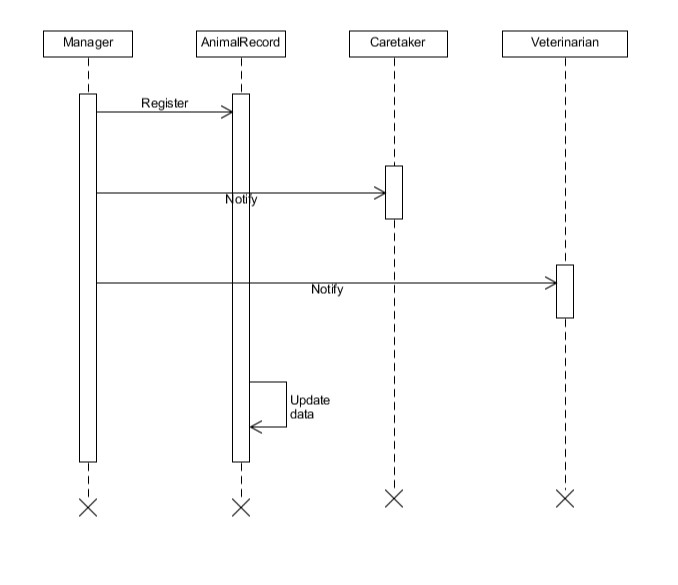
(2)



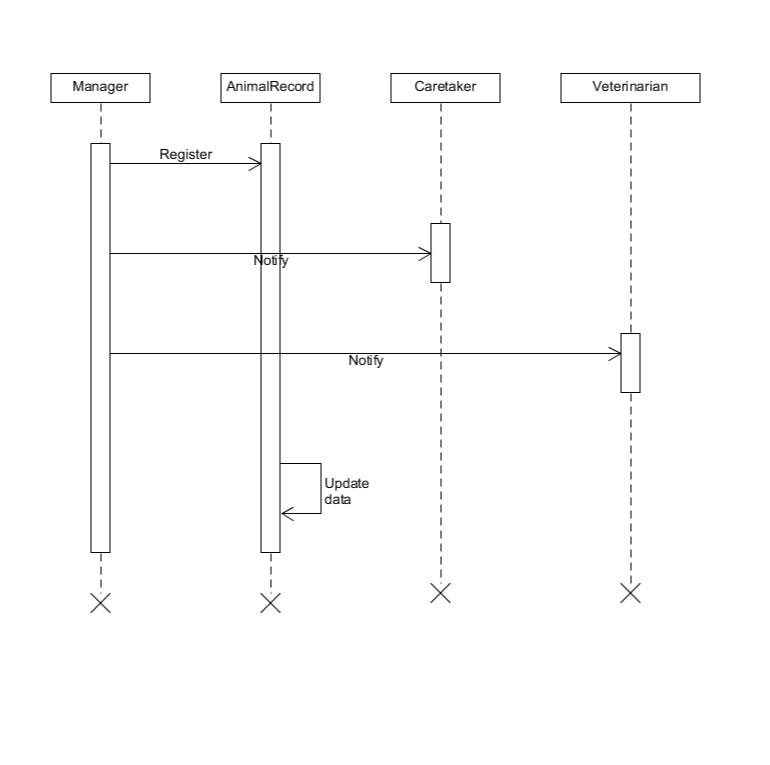
(3)



(4)

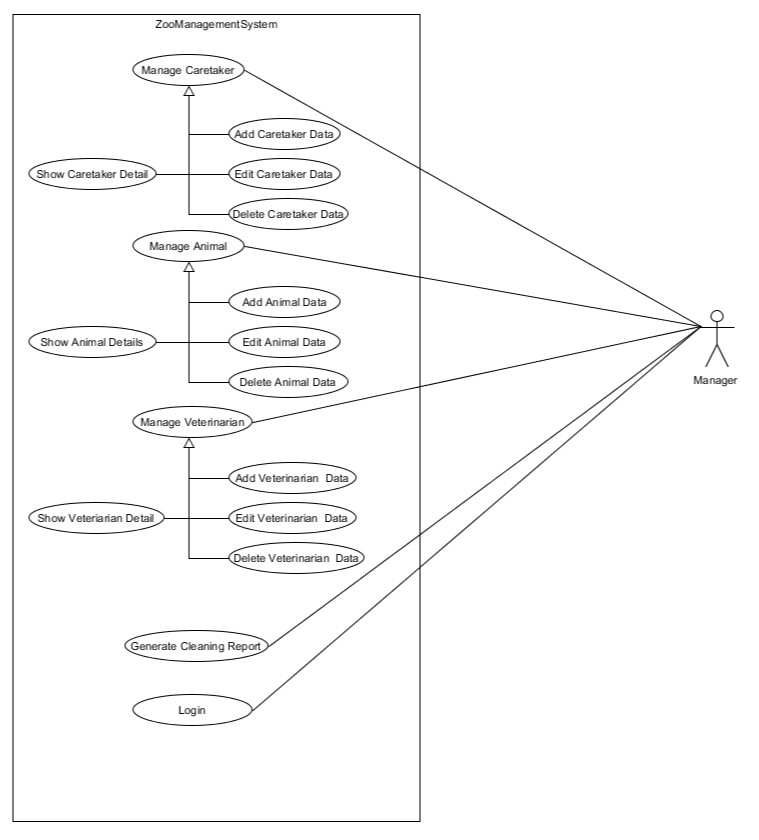


(5)

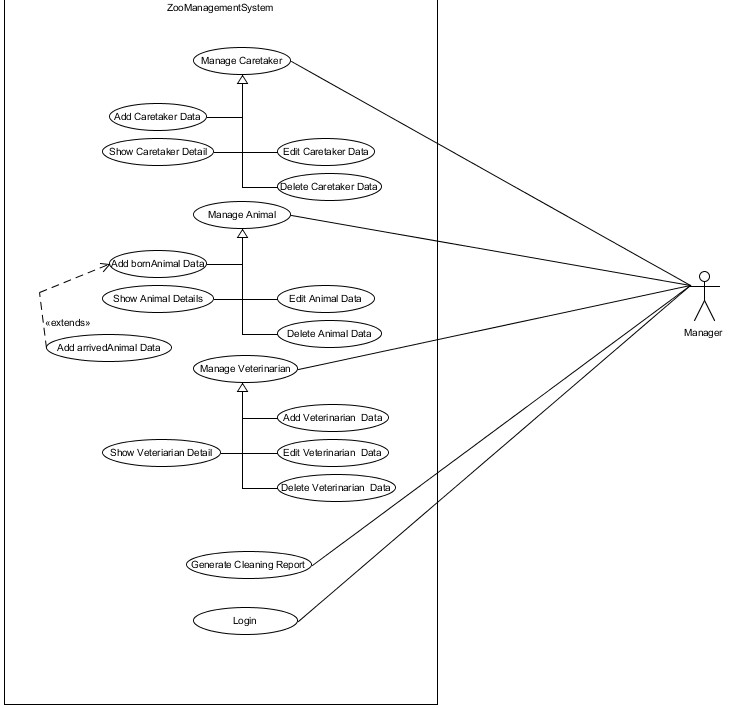


**Usecase Diagram**

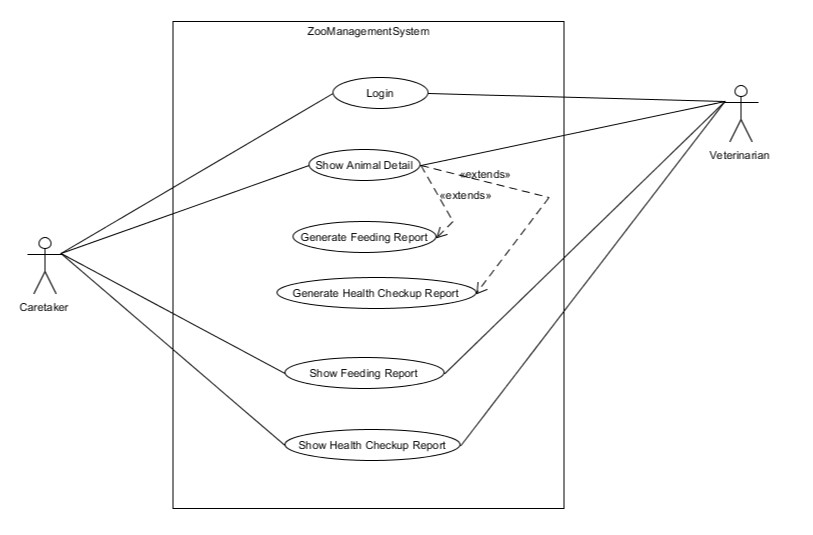
(1)



(2)



(3)



**Implementation Detail**

**Usercase Detail:**

**1. Show Animal**

In this usercase it will show detail of each animal.

**2. Search Animal**

In this usercase user can search animal by their name, id or species.

**3. Add animal**

In this usercase manager can add the animal.

**4. Edit Animal**

In this usercase manager can edit or delete the data of animal.

**5. Show Vets**

In this usercase it will show detail of each vets.

**6. Search Vets**

In this usercase user can search vets by their name, id or species.

**7. Add Vets**

In this usercase manager can add the vets.

**8. Edit Vets**

In this usercase manager can edit or delete the data of vets.

**9. Show Caretaker**

In this usercase it will show detail of each caretaker.

**10. Search Caretaker**

In this usercase user can search caretaker by their name, id or species.

**11. Add Caretaker**

In this usercase manager can add the caretaker.

**12. Edit Caretaker**

In this usercase manager can edit or delete the data of caretaker.

**13. Create Cleaning Report**

In this usercase manager can create the cleaning report.

**13. Create Food Report**

In this usercase vet can create the cleaning report.

**13. Create Health Report**

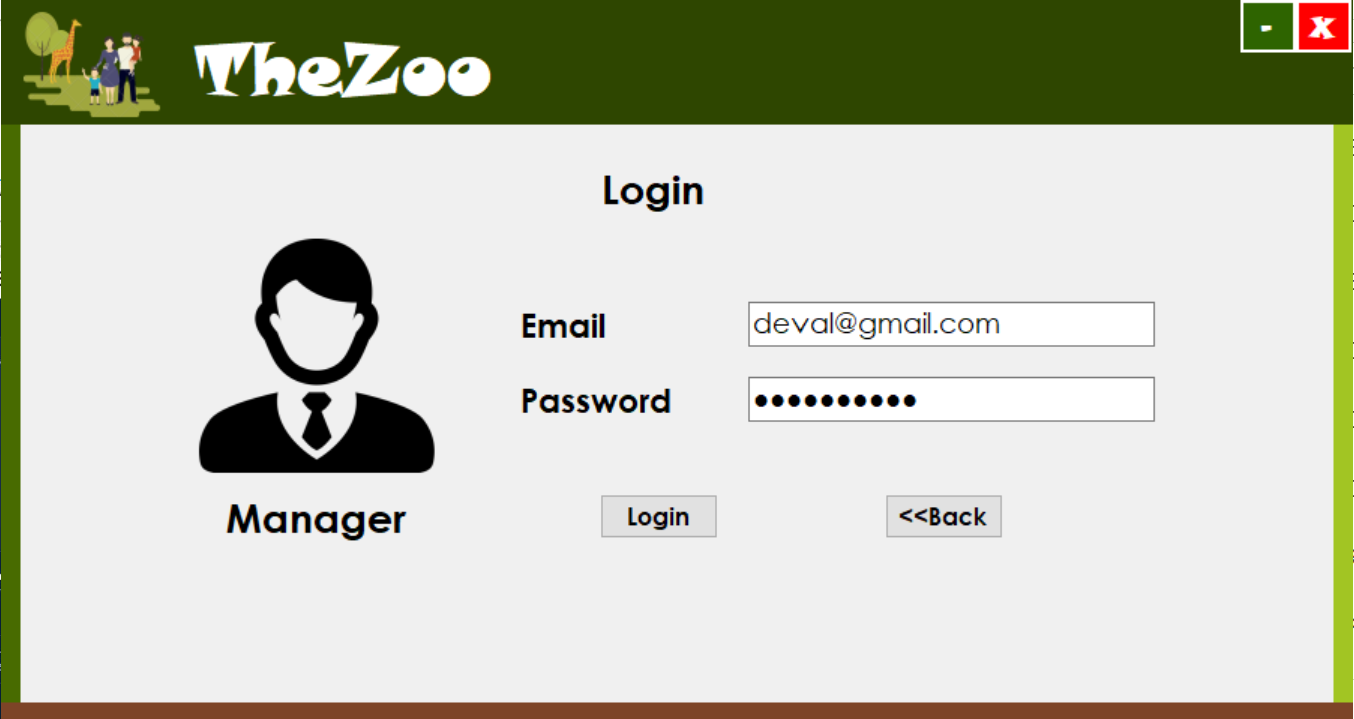
In this usercase vet can create the cleaning report.

**Screen-shots**

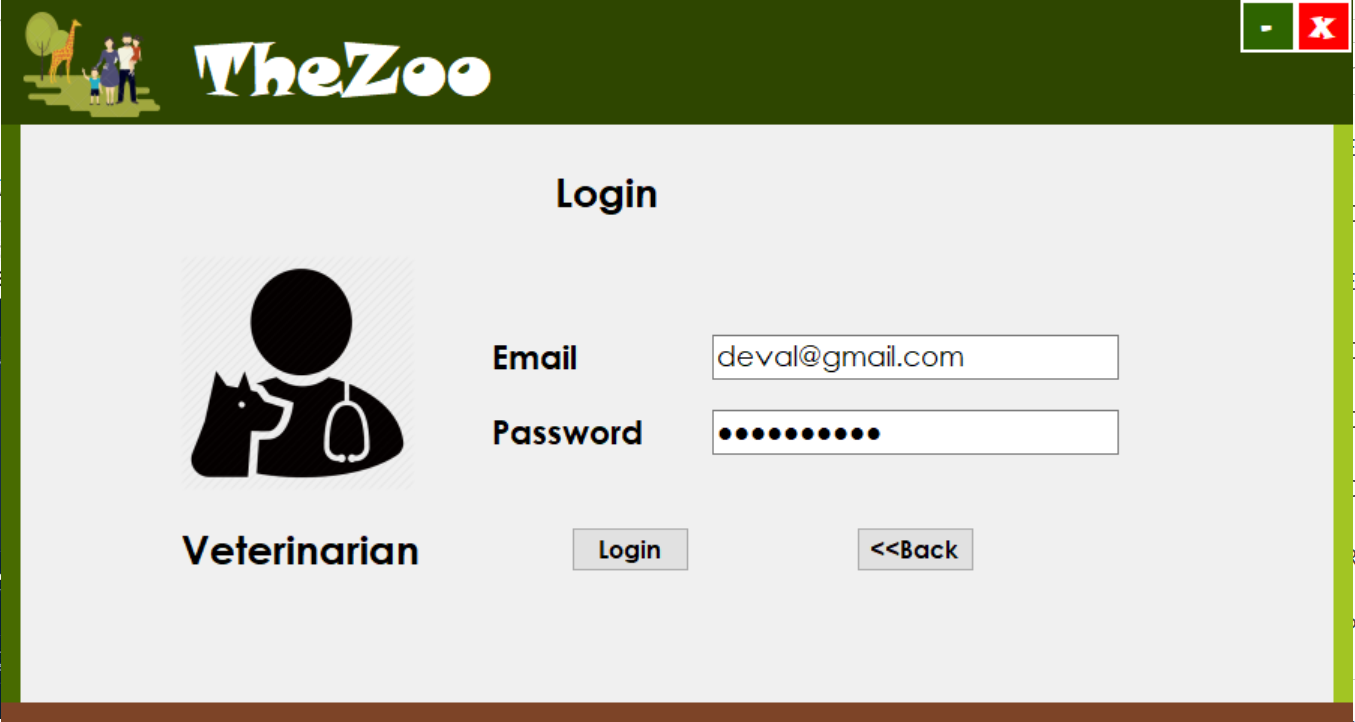
**Home**

****

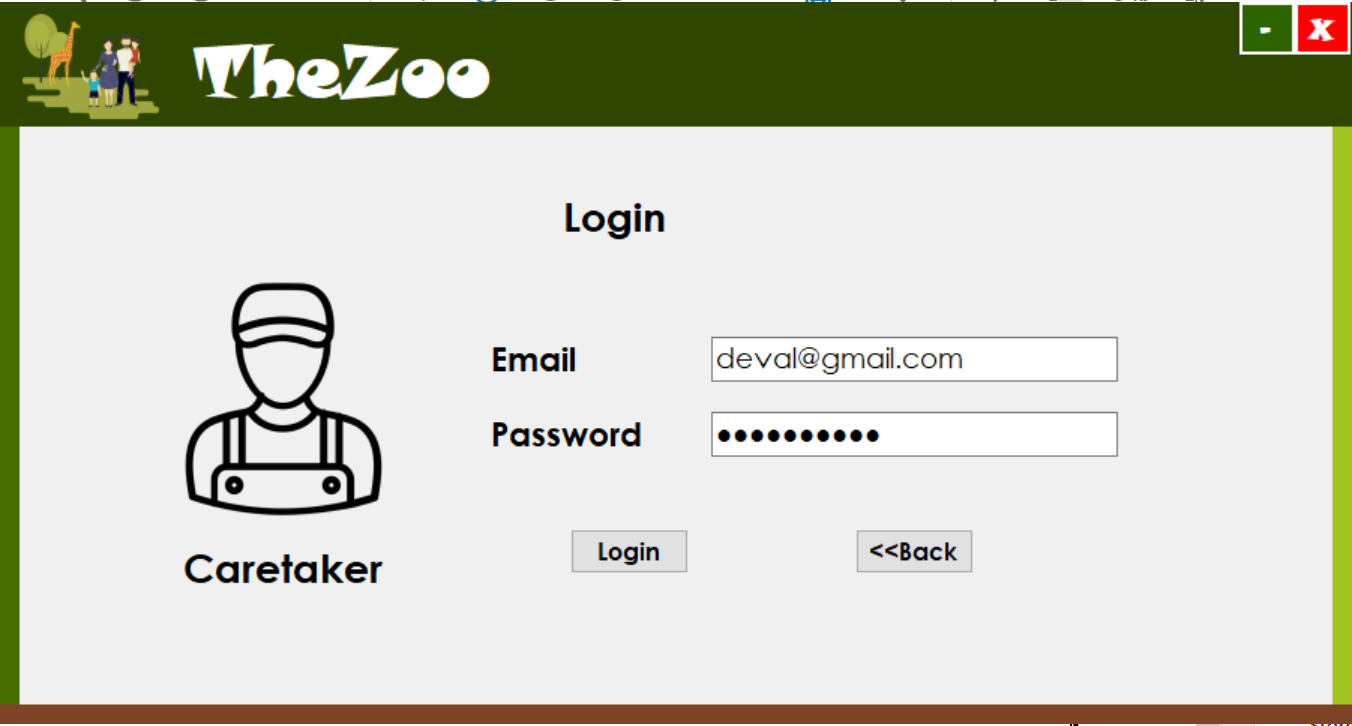
**Manager Login**

****

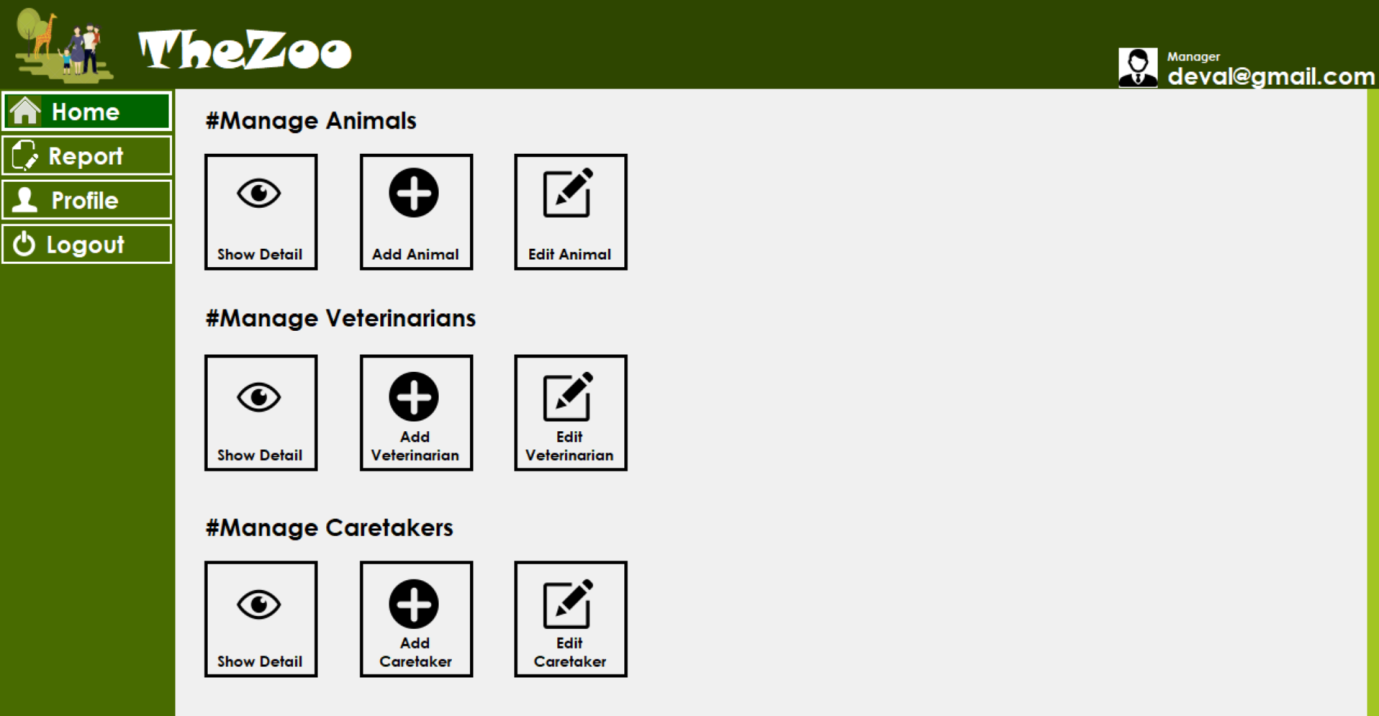
**Veterinarian Login**

****

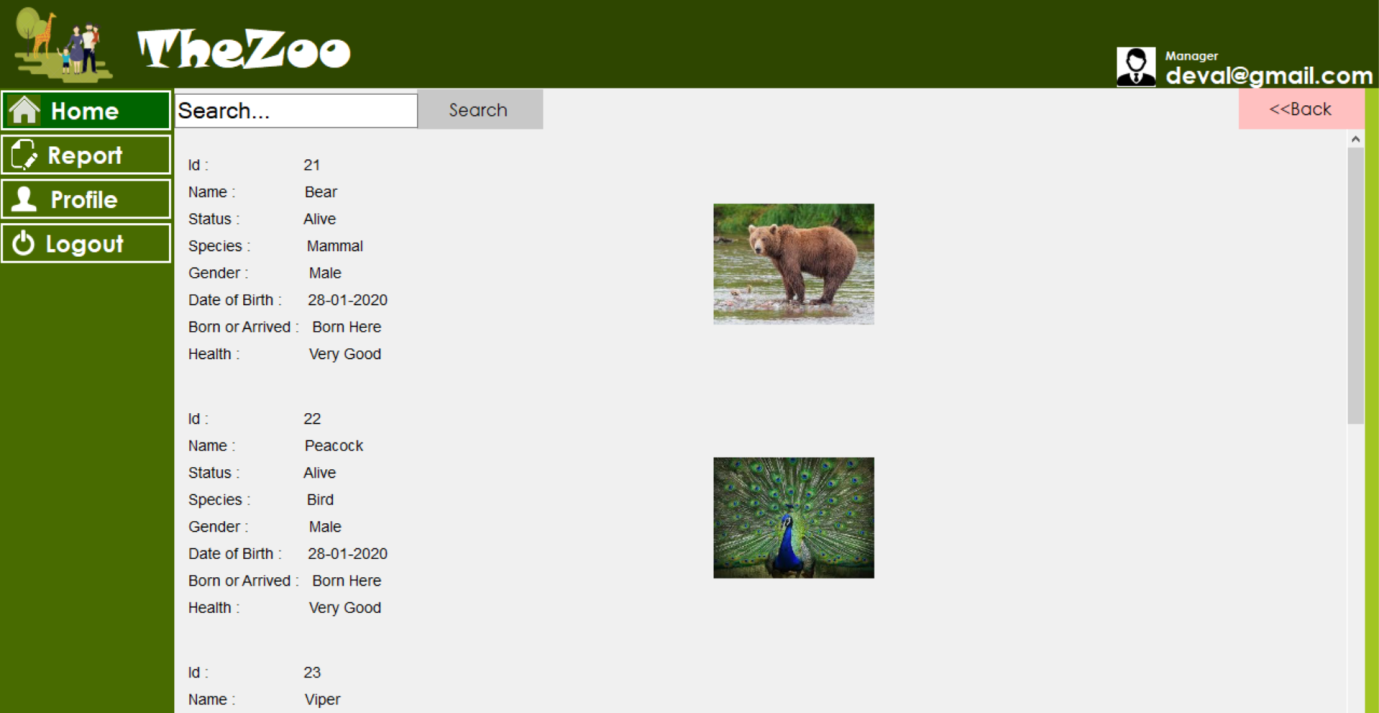
**Caretaker Login**

****

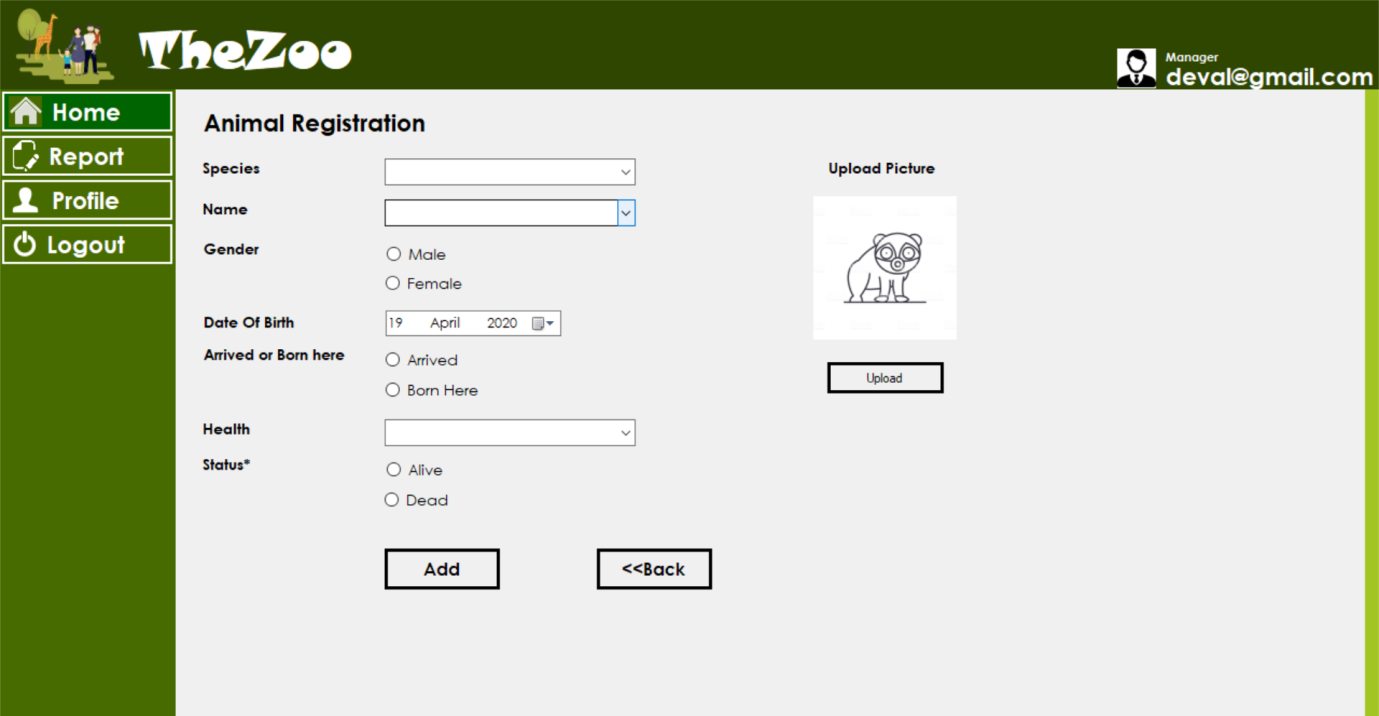
**Manager Home**

****

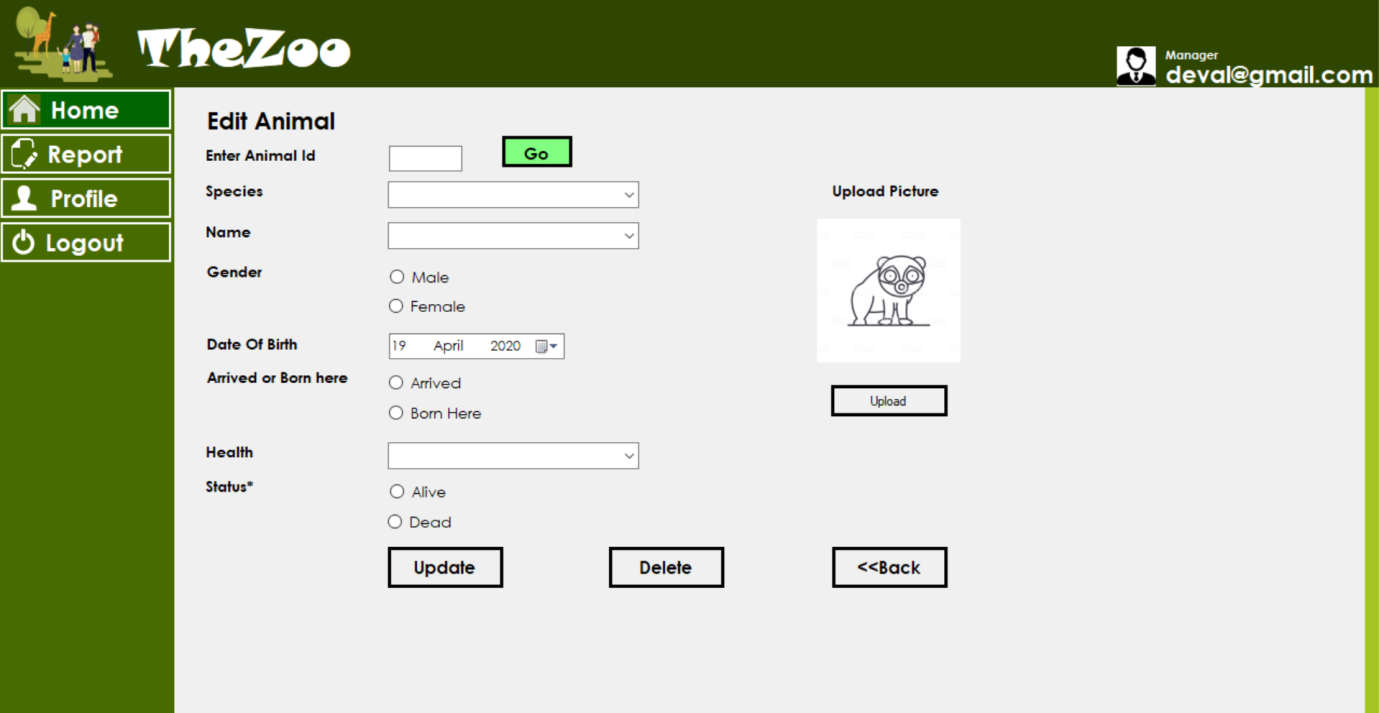
**Show Animal**

****

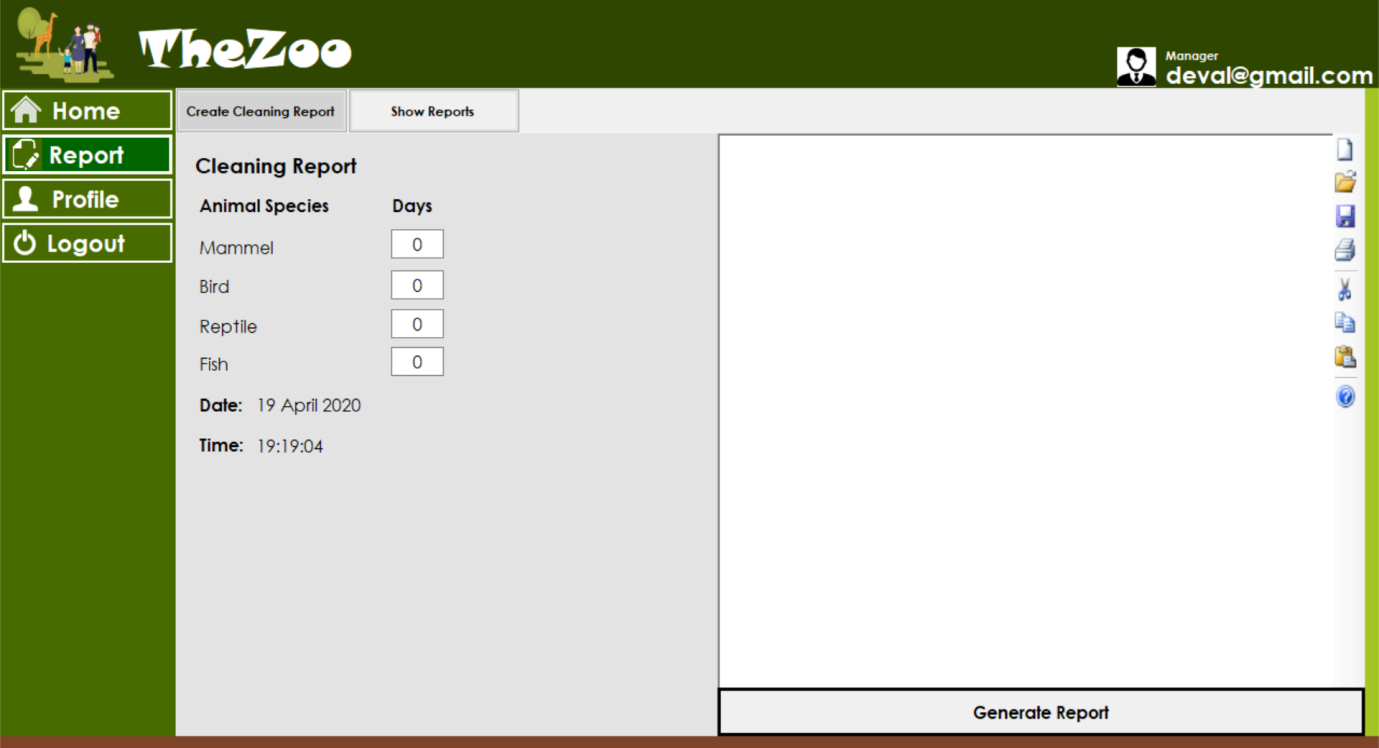
**Add Animal**

****

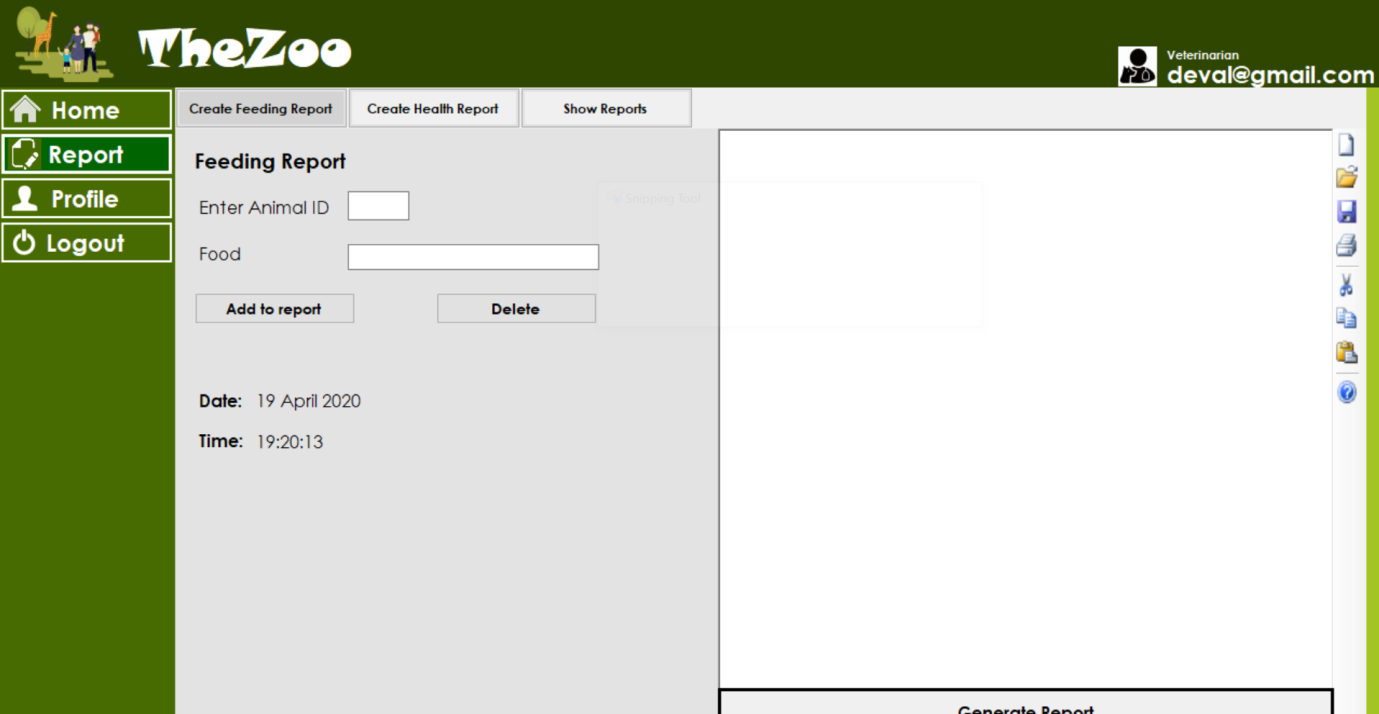
**Edit Animal**

****

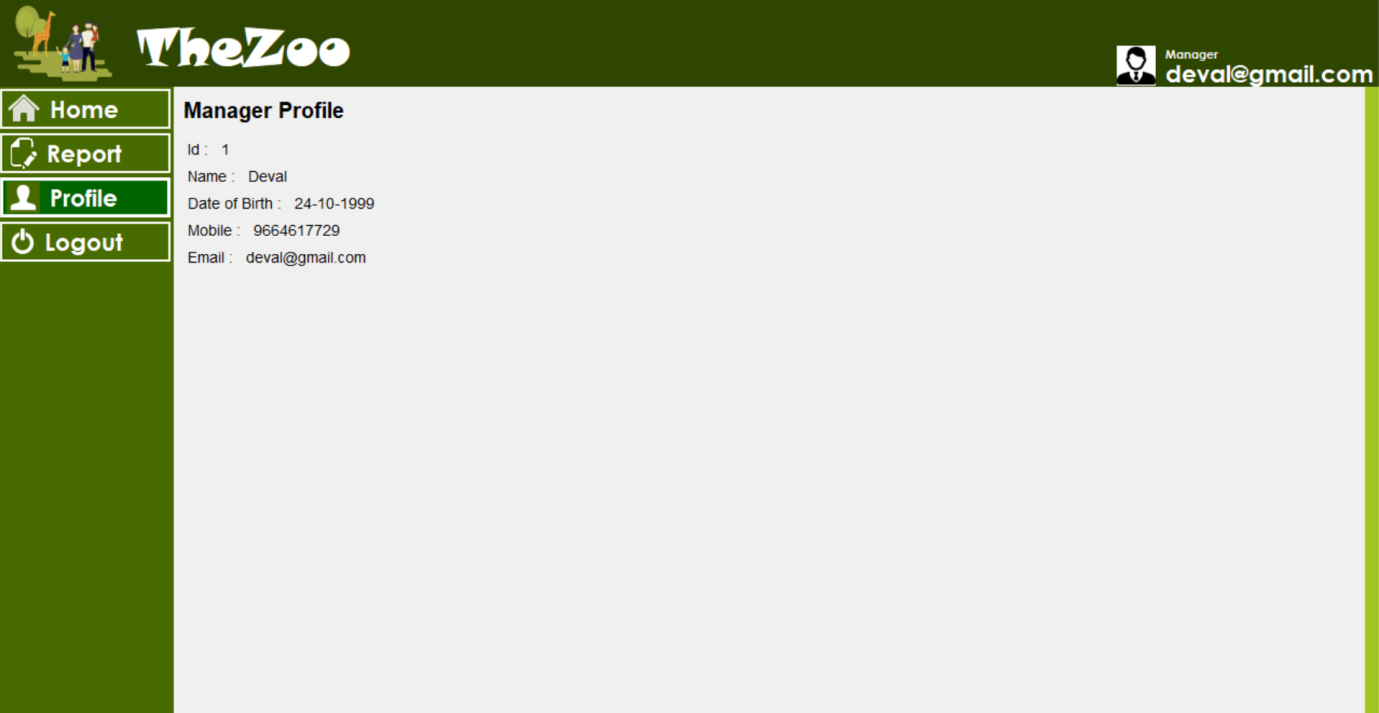
**Create Cleaning Report**

****

**Create Food Report**

****

**Show Profile**

****

**Limitations of System**

In this zoo system there is no customer side.

This system is lack of security.

There is ticket booking system.

**Future Extensions**

In this we can add customer side.

Make design more effective and user friendly.

We can add online ticket booking.

**Conclusion**

The functionality implemented in the system was done after

understanding all the system modules according to the requirements.

Functionalities that are successfully implemented in the system are:

-Add, Update and Delete Animal

-Create Reports

-Add, Update and Delete Vet

-Add, Update and Delete Caretaker

After the implementation and coding of system, comprehensive testing was performed on the system to determine the loopholes and possible flaws in the system.