

# **JAVA SWING BASED - ANIMAL DEPOT MANAGEMENT SYSTEM - SQL CONNECTIVITY USING JDBC**

*A*

*Report*

*Submitted in partial fulfillment of  
the Requirements for the award of  
the Degree of*

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**By**

K. KRISHNA <1602-19-737-077>

Under the guidance of Ms B. Leelavathy



Department of Information  
Technology Vasavi College of  
Engineering (Autonomous)  
(Affiliated to Osmania University)  
Ibrahimbagh, Hyderabad-31

## **BONAFIDE CERTIFICATE**

This is to certify that this project report titled '**Animal Depot Management System**' is a project work of K KRISHNA bearing roll no. 1602-19-737-077 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature

Internal Examiner

Signature

External Examiner

## **ABSTRACT**

The Animal Depot Management System (ADMS) is a graphic user interface that allows the manager and the employees to carry out maintenance and management of database and basically it is the Menagerie / Zoological Park data management system. The three groups of users for this application are the managers, the animal keepers and the animal trainers.

ADMS uses RDBMS for storage and retrieval of data. There will only be one super user (Manager) who has all rights. Super user has the access to add/update the database and employee can only view the content as required. For creating database we have used Oracle's 11g XE which is standard and easy to work with and provides good interface.

This database consists of all around 16 tables and all are included in association with one or more tables.

# INTRODUCTION

## REQUIREMENTS

### List of tables

- Zoo
- Site
- HabitatBiome
- Habitat
- Animal
- EmployeeCommunication
- Employee
- Keeper
- Trainer
- Food
- Shows
- Performs
- CaresFor
- Trains
- Trades
- Eats

### List of attributes with their domain types

- **Zoo:**
  - Address : varchar2(40)
  - Name : varchar2(40)
  - Phone : number(10)
  - City : varchar2(20)
  - Country : varchar2(20)

➤ **Site:**

site\_id : number(20)

Location : varchar2(20)

used\_for : varchar2(20)

➤ **Habitat:**

Enclosureid : number(10)

Biome:varchar2(20)

Sqft : number(10)

Depth : number(10)

➤ **Animal:**

Animalid : number(10)

Name : varchar2(20)

Age:number(10)

Sex : varchar2(20)

Height:number(10)

Weight : number(10)

Species : varchar2(20)

eat\_freq\_week : number(10)

eat\_amount : number(10)

➤ **Employee:**

Fname : varchar2(20)

Lname : varchar2(20)

Empid : number(10)

Pay : number(10)

➤ **Keeper:**

Duty : varchar2(20)

➤ **Trainer:**

Speciality : varchar2(20)

➤ **Food:**

Foodid : number(10)

Name : varchar2(20)

stock\_serving : number(10)

date\_purchased : date

date\_expired : date

➤ **Show:**

start\_time : varchar2(8)

Duration : number(10)

Name : varchar2(20)

Type : varchar2(20)

## **AIM AND PRIORITY OF THE PROJECT**

To create a Java GUI based Animal Depot data management system which handles the data of the zoo parks present in various countries and the data of those zoo parks such as species, employees, etc. This application helps user in easy insertion, deletion, updation and retrieval of the data. The data is updated in the database using JDBC connectivity when ever changes is been made seamlessly.

## **ARCHITECTURE AND TECHNOLOGY**

### **Software used:**

Java Eclipse, Oracle 11g Database, Java SE version 13, SQL\*Plus.

### **Java AWT and SWING:**

Java AWT(Abstract Window Toolkit) and SWING is an API to develop GUI or window-based applications in java.

Java SWING components are platform-independent i.e. components are displayed irrespective to the view of operating system. SWING is lightweight.

The javax.swing package provides classes for SWING API such as JTextField, Label, JTextArea, JButton, JCheckBox, JChoice, JComboBox etc.

### **SQL:**

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use SQL as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

The diagram is an Entity-Relationship (ER) model for a zoo database. It includes the following entities and their attributes:

- Zoo** (Entity): address, city, country, phone, name. It has a 1:1 relationship with **Site** labeled "maintains".
- Employee** (Entity): (name), fname, empid, pay. It has a 1:1 relationship with **Trainer** labeled "is a" and a 1:1 relationship with **Keeper** labeled "is a".
- Habitat** (Entity): enclosureid, temp, humidity, sqft, biome, depth. It has a 1:1 relationship with **Animal** labeled "lives in".
- Animal** (Entity): animalid, name, species, height, weight, sex. It has a 1:1 relationship with **Trainer** labeled "trains" and a 1:1 relationship with **Food** labeled "eats".
- Food** (Entity): foodid, stockServing, purchDate, expiryDate. It has a 1:1 relationship with **Animal** labeled "occurs at".
- Site** (Entity): siteid, location, usedFor. It has a 1:1 relationship with **Zoo** labeled "maintains".
- Trainer** (Entity): trainerid, specialty. It has a 1:1 relationship with **Employee** labeled "is a" and a 1:1 relationship with **Animal** labeled "trains".
- Keeper** (Entity): keeperid, duty. It has a 1:1 relationship with **Employee** labeled "is a" and a 1:1 relationship with **Animal** labeled "cares for".
- Show** (Entity): start time, duration, type, name. It has a 1:1 relationship with **Animal** labeled "performs".

Relationships and Cardinalities:

- maintains** (Zoo to Site): 1:1, mandatory on both sides.
- lives in** (Habitat to Animal): 1:1, mandatory on both sides.
- trains** (Trainer to Animal): 1:1, mandatory on both sides.
- cares for** (Keeper to Animal): 1:1, mandatory on both sides.
- occurs at** (Food to Animal): 1:1, mandatory on both sides.
- eats** (Animal to Food): 1:1, mandatory on both sides.
- performs** (Animal to Show): 1:1, mandatory on both sides.
- is a** (Employee to Trainer): 1:1, mandatory on both sides.
- is a** (Employee to Keeper): 1:1, mandatory on both sides.



## **MAPPING CARDINALITY AND PARTICIATION CONSTRAINTS**

- Zoo is associated with Site in many to one participation as many Sites can have same address.
- Habitat and Site are connect in one to many participation as many habitat may belong same site.
- Habitat and Animal is associated in one to one participation as only one animal can have one habitat or vice-versa.
- Many employees may have same address hence they are connected in one to many participation.
- Atleast one animal need to be handled by keeper hence its a mandatory one to one participation.
- Show and Animal are connect in mandatory one to one participation I.e, the show must have atleast one animal.
- Show is associated with Site in one to many participation as many shows can happen on same site but at different timings.
- Food and Animal are connect and are in many to many participation as different kinds is food is eaten by various animals.
- Trainer and Animal are in association as one trainer can train many animals and many different animals can be trained by same trainee, hence it is many to many participation.

## DDL COMMANDS

```
create table zoo(  
address varchar2(40),  
name varchar2(40) not null,  
phone number(10),  
city varchar2(20),  
country varchar2(20),  
primary key (address)  
);
```

```
create table site(  
siteid number(20),  
location varchar2(20),  
usedfor varchar2(20),  
zooaddress varchar2(40),  
primary key (siteid),  
foreign key (zooaddress) references zoo ON DELETE CASCADE  
);
```

```
create table HabitatBiome(  
biome varchar2(20),  
temp number(10),  
humidity number(10),  
primary key (biome)  
);
```

```
create table habitat(  
enclosureid number(10),  
biome varchar2(20),  
sqft number(10),  
depth number(10),  
siteid number(10) not null,  
primary key (enclosureid),  
foreign key (biome) references habitatbiome,
```

foreign key (siteid) references site ON DELETE CASCADE  
);

create table **animal**(  
animalid number(10),  
name varchar2(20),  
age number(10),  
sex varchar2(20),  
height number(10),  
weight number(10),  
species varchar2(20),eat\_freq\_week number(10),  
eat\_amount number(10),  
enclosure\_id number(10),  
primary key (animalid),  
foreign key (enclosure\_id) references habitat ON DELETE SET NULL  
);

create table **employeecommunication**(  
fname varchar2(20),  
lname varchar2(20),  
walkeetalkeeno number(10),  
primary key (fname, lname)  
);

create table **employee**(  
fname varchar2(20),  
lname varchar2(20),  
empid number(10),  
pay number(10),  
zooaddress varchar2(40) not null,  
primary key (empid),  
foreign key (zooaddress) references zoo ON DELETE SET NULL,  
foreign key (fname, lname) references employeecommunication  
);

```
create table keeper(  
duty varchar2(20),  
empid number(10),  
primary key (empid),  
foreign key (empid) references employee ON DELETE CASCADE  
);
```

```
create table trainer(  
speciality varchar2(20),  
empid number(10),  
primary key (empid),  
foreign key (empid) references employee ON DELETE CASCADE  
);
```

```
create table food(  
foodid number(10),  
name varchar2(20),  
stock_serving number(10),  
date_purchased date,  
date_expired date,  
siteid number(10) not null,  
primary key (foodid),  
foreign key (siteid) references site ON DELETE CASCADE  
);
```

```
create table show(  
start_time varchar2(8),  
duration number(10),  
name varchar2(20),  
type varchar2(20),  
siteid number(10) not null,  
primary key (start_time, name),  
foreign key (siteid) references site ON DELETE CASCADE  
);
```

```
create table performs(
start_time varchar2(8),
show_name varchar2(20),
empid number(10),
animalid number(10),
role varchar2(50),
primary key (start_time, empid, animalid),
foreign key (start_time, show_name) references show,
foreign key (empid) references employee,
Foreign key (animalid) references animal
);
```

```
create table caresfor(
empid number(10),
animalid number(10),
primary key (empid, animalid),
foreign key (empid) references employee,
foreign key (animalid) references animal
);
```

```
create table trains(
empid number(10),
animalid number(10),
skills varchar2(50),
primary key (empid, animalid),
foreign key (empid) references employee,
foreign key (animalid) references animal
);
```

```
create table trades(
zoo_from_address varchar2(40),
zoo_to_address varchar2(40),
animalid number(10),
trade_date date,
```

primary key (zoo\_from\_address, zoo\_to\_address, animalid),  
foreign key (zoo\_from\_address) references zoo,  
foreign key (zoo\_to\_address) references zoo,  
foreign key (animalid) references animal  
);

create table **eats**(  
animalid number(10),  
foodid number(10),  
primary key (animalid, foodid),  
foreign key (animalid) references animal ON DELETE CASCADE,  
foreign key (foodid) references food  
);

## DML COMMANDS

- insert into ZOO values('&ADDRESS', '&NAME', &PHONE, '&CITY', '&COUNTRY');
- insert into SITE values('&SITEID', '&LOCATION', '&USEDFOR', '&ZOOADDRESS');
- insert into HABITATBIOME values('&BIOME', &TEMP, &HUMIDITY);
- insert into HABITAT values('&ENCLOSURE\_ID', '&BIOME', &SQFT, &DEPTH, &SITEID);
- insert into ANIMAL values('&ANIMALID', '&NAME', &AGE, '&SEX', &HEIGHT, &WEIGHT, '&SPECIES', &EAT\_FREQ\_WEEK, &EAT\_AMOUNT, &ENCLOSURE\_ID);
- insert into EMPLOYEECOMMUNICATION values('&FNAME', '&LNAME', &WALKEETALKEENO);
- insert into EMPLOYEE values('&FNAME', '&LNAME', &EMPID, &PAY, '&ZOOADDRESS');
- insert into KEEPER values('&DUTY', &EMPID);
- insert into TRAINER values('&SPECIALITY', &EMPID);
- insert into FOOD values('&FOODID', '&NAME', &STOCK\_SERVING, '&DATE\_PURCHASED', '&DATE\_EXPIRED', &SITEID);
- insert into SHOW values('&START\_TIME', &DURATION, '&NAME', '&TYPE', &SITEID);
- insert into PERFORMS values ('&START\_TIME', '&SHOW\_NAME', &EMPID,

&ANIMALID, '&ROLE');

- insert into CARESFOR values (&EMPID, &ANIMALID);
- insert into TRAINS values (&EMPID, &ANIMALID, '&SKILLS');
- insert into TRADES values ('&ZOO\_FROM\_ADDRESS', '&ZOO\_TO\_ADDRESS', &ANIMALID, '&TRADE\_DATE');
- insert into EATS values (&ANIMALID, &FOODID);

```

SQL> desc zoo;
Name                                     Null?   Type
-----
ADDRESS                                NOT NULL VARCHAR2(40)
NAME                                   NOT NULL VARCHAR2(40)
PHONE                                  NUMBER(10)
CITY                                   VARCHAR2(20)
COUNTRY                               VARCHAR2(20)

SQL> desc site;
Name                                     Null?   Type
-----
SITEID                                NOT NULL NUMBER(20)
LOCATION                               VARCHAR2(20)
USEDFOR                              VARCHAR2(20)
ZOOADDRESS                            VARCHAR2(40)

SQL> desc habitatbiome;
Name                                     Null?   Type
-----
BIOME                                 NOT NULL VARCHAR2(20)
TEMP                                  NUMBER(10)
HUMIDITY                              NUMBER(10)

SQL> desc habitat;
Name                                     Null?   Type
-----
ENCLOSURE_ID                          NOT NULL NUMBER(10)
BIOME                                  VARCHAR2(20)
SQFT                                   NUMBER(10)
DEPTH                                 NUMBER(10)
SITEID                                NOT NULL NUMBER(10)

```

```

SQL> desc animal;
Name                                     Null?   Type
-----
ANIMALID                              NOT NULL NUMBER(10)
NAME                                   VARCHAR2(20)
AGE                                    NUMBER(10)
SEX                                    VARCHAR2(20)
HEIGHT                                 NUMBER(10)
WEIGHT                                 NUMBER(10)
SPECIES                               VARCHAR2(20)
EAT_FREQ_WEEK                         NUMBER(10)
EAT_AMOUNT                            NUMBER(10)
ENCLOSURE_ID                          NUMBER(10)

SQL> desc employeeecommunication;
Name                                     Null?   Type
-----
FNAME                                 NOT NULL VARCHAR2(20)
LNAME                                 NOT NULL VARCHAR2(20)
WALKEETALKEENO                       NUMBER(10)

SQL> desc employee;
Name                                     Null?   Type
-----
FNAME                                 VARCHAR2(20)
LNAME                                 VARCHAR2(20)
EMPID                                NOT NULL NUMBER(10)
PAY                                   NUMBER(10)
ZOO_ADDRESS                           NOT NULL VARCHAR2(40)

SQL> desc keeper;
Name                                     Null?   Type
-----
DUTY                                  VARCHAR2(20)
EMPID                                NOT NULL NUMBER(10)

SQL> desc trainer;
Name                                     Null?   Type
-----
SPECIALITY                           VARCHAR2(20)
EMPID                                NOT NULL NUMBER(10)

```

```

SQL> desc trades;
Name                                     Null?      Type
-----
ZOO_FROM_ADDRESS                       NOT NULL   VARCHAR2(40)
ZOO_TO_ADDRESS                         NOT NULL   VARCHAR2(40)
ANIMALID                              NOT NULL   NUMBER(10)
TRADE_DATE                             DATE

SQL> desc eats;
Name                                     Null?      Type
-----
ANIMALID                              NOT NULL   NUMBER(10)
FOODID                                NOT NULL   NUMBER(10)

SQL> desc food;
Name                                     Null?      Type
-----
FOODID                                NOT NULL   NUMBER(10)
NAME                                   VARCHAR2(20)
STOCK_SERVING                          NUMBER(10)
DATE_PURCHASED                        DATE
DATE_EXPIRED                          DATE
SITEID                                NOT NULL   NUMBER(10)

SQL> desc show;
Name                                     Null?      Type
-----
START_TIME                            NOT NULL   VARCHAR2(8)
DURATION                              NUMBER(10)
NAME                                   VARCHAR2(20)
TYPE                                   VARCHAR2(20)
SITEID                                NOT NULL   NUMBER(10)

SQL> desc performs;
Name                                     Null?      Type
-----
START_TIME                            NOT NULL   VARCHAR2(8)
SHOW_NAME                             VARCHAR2(20)
EMPID                                  NOT NULL   NUMBER(10)
ANIMALID                              NOT NULL   NUMBER(10)
ROLE                                   VARCHAR2(50)

SQL> desc caresfor;
Name                                     Null?      Type
-----
EMPID                                  NOT NULL   NUMBER(10)
ANIMALID                              NOT NULL   NUMBER(10)

SQL> desc trains;
Name                                     Null?      Type
-----
EMPID                                  NOT NULL   NUMBER(10)
ANIMALID                              NOT NULL   NUMBER(10)
SKILLS                                 VARCHAR2(50)

SQL> insert into ZOO values('&ADDRESS', '&NAME', &PHONE, '&CITY', '&COUNTRY');
Enter value for address: 15500 San Pasqual Valley Rd
Enter value for name: San Diego Zoo Safari Park
Enter value for phone: 12345
Enter value for city: Escondido
Enter value for country: USA
old 1: insert into ZOO values('&ADDRESS', '&NAME', &PHONE, '&CITY', '&COUNTRY')
new 1: insert into ZOO values('15500 San Pasqual Valley Rd', 'San Diego Zoo Safari Park', 12345, 'Escondido', 'USA')

1 row created.

SQL> /
Enter value for address: 2000 Meadowvale Rd
Enter value for name: Toronto Zoo
Enter value for phone: 23456
Enter value for city: Toronto
Enter value for country: Canada
old 1: insert into ZOO values('&ADDRESS', '&NAME', &PHONE, '&CITY', '&COUNTRY')
new 1: insert into ZOO values('2000 Meadowvale Rd', 'Toronto Zoo', 23456, 'Toronto', 'Canada')

1 row created.

SQL> /
Enter value for address: 80 Mandai Lake Rd
Enter value for name: Singapore Zoo
Enter value for phone: 45678
Enter value for city: Singapore
Enter value for country: Singapore
old 1: insert into ZOO values('&ADDRESS', '&NAME', &PHONE, '&CITY', '&COUNTRY')
new 1: insert into ZOO values('80 Mandai Lake Rd', 'Singapore Zoo', 45678, 'Singapore', 'Singapore')

1 row created.

```



```
SQL> select * from zoo;
```

```
ADDRESS
```

```
-----
```

NAME	PHONE	CITY
-----		
COUNTRY		
-----		
15500 San Pasqual Valley Rd San Diego Zoo Safari Park USA	12345	Escondido
2000 Meadowvale Rd Toronto Zoo Canada	23456	Toronto

```
ADDRESS
```

```
-----
```

NAME	PHONE	CITY
-----		
COUNTRY		
-----		
80 Mandai Lake Rd Singapore Zoo Singapore	45678	Singapore
London NW1 4RY London Zoo	56789345	London

```
ADDRESS
```

```
-----
```

NAME	PHONE	CITY
-----		
COUNTRY		
-----		
UK		

```
-----
```

NAME	PHONE	CITY
-----		
COUNTRY		
-----		
UK		

```
-----
```

```
-----
```

NAME	PHONE	CITY
-----		
COUNTRY		
-----		
UK		

```
-----
```

NAME	PHONE	CITY
-----		
COUNTRY		
-----		
UK		

```
-----
```

```
-----
```

NAME	PHONE	CITY
-----		
COUNTRY		
-----		
UK		

```
-----
```

```
-----
```

```
SQL> insert into SITE values(&SITEID, '&LOCATION', '&USEDFOR', '&ZOOADDRESS');
```

```
Enter value for siteid: 101
```

```
Enter value for location: Area 1
```

```
Enter value for usedfor: food
```

```
Enter value for zooaddress: 2000 Meadowvale Rd
```

```
old 1: insert into SITE values(&SITEID, '&LOCATION', '&USEDFOR', '&ZOOADDRESS')
```

```
new 1: insert into SITE values(101, 'Area 1', 'food', '2000 Meadowvale Rd')
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for siteid: 102
```

```
Enter value for location: Area 2 North
```

```
Enter value for usedfor: habitat
```

```
Enter value for zooaddress: 2000 Meadowvale Rd
```

```
old 1: insert into SITE values(&SITEID, '&LOCATION', '&USEDFOR', '&ZOOADDRESS')
```

```
new 1: insert into SITE values(102, 'Area 2 North', 'habitat', '2000 Meadowvale Rd')
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for siteid: 103
```

```
Enter value for location: Area 1 West
```

```
Enter value for usedfor: food
```

```
Enter value for zooaddress: 80 Mandai Lake Rd
```

```
old 1: insert into SITE values(&SITEID, '&LOCATION', '&USEDFOR', '&ZOOADDRESS')
```

```
new 1: insert into SITE values(103, 'Area 1 West', 'food', '80 Mandai Lake Rd')
```

```
1 row created.
```

```

SQL> select * from site;

  SITEID LOCATION                USEDFOR
-----
ZOOADDRESS
-----
    101 Area 1                    food
2000 Meadowvale Rd
    102 Area 2 North              habitat
2000 Meadowvale Rd
    103 Area 1 West              food
80 Mandai Lake Rd

  SITEID LOCATION                USEDFOR
-----
ZOOADDRESS
-----
    104 Area 1 East              habitat
80 Mandai Lake Rd

SQL> @insert into HABITATBIOME values('&BIOME', &TEMP, &HUMIDITY);
SP2-0734: unknown command beginning "?insert in..." - rest of line ignored.
SQL> insert into HABITATBIOME values('&BIOME', &TEMP, &HUMIDITY);
Enter value for biome: Tropical Rain Forest
Enter value for temp: 25
Enter value for humidity: 9
old 1: insert into HABITATBIOME values('&BIOME', &TEMP, &HUMIDITY)
new 1: insert into HABITATBIOME values('Tropical Rain Forest', 25, 9)

1 row created.

SQL> /
Enter value for biome: Grassland
Enter value for temp: 15
Enter value for humidity: 4
old 1: insert into HABITATBIOME values('&BIOME', &TEMP, &HUMIDITY)
new 1: insert into HABITATBIOME values('Grassland', 15, 4)

1 row created.

SQL> /
Enter value for biome: Ocean
Enter value for temp: 10
Enter value for humidity: 10
old 1: insert into HABITATBIOME values('&BIOME', &TEMP, &HUMIDITY)
new 1: insert into HABITATBIOME values('Ocean', 10, 10)

1 row created.

SQL> select * from habitatbiome;

BIOME                TEMP  HUMIDITY
-----
Tropical Rain Forest    25      9
Grassland                15      4
Dessert                 30      0
Ocean                   10     10

SQL> insert into HABITAT values(&ENCLOSURE_ID, '&BIOME', &SQFT, &DEPTH, &SITEID);
Enter value for enclosure_id: 1001
Enter value for biome: Tropical Rain Forest
Enter value for sqft: 50
Enter value for depth: 10
Enter value for siteid: 102
old 1: insert into HABITAT values(&ENCLOSURE_ID, '&BIOME', &SQFT, &DEPTH, &SITEID)
new 1: insert into HABITAT values(1001, 'Tropical Rain Forest', 50, 10, 102)

1 row created.

```

```

SQL> /
Enter value for enclosure_id: 1003
Enter value for biome: Ocean
Enter value for sqft: 100
Enter value for depth: 20
Enter value for siteid: 103
old 1: insert into HABITAT values(&ENCLOSURE_ID, '&BIOME', &SQFT, &DEPTH, &SITEID)
new 1: insert into HABITAT values(1003, 'Ocean', 100, 20, 103)

1 row created.

SQL> /
Enter value for enclosure_id: 1004
Enter value for biome: Dessert
Enter value for sqft: 20
Enter value for depth: 40
Enter value for siteid: 104
old 1: insert into HABITAT values(&ENCLOSURE_ID, '&BIOME', &SQFT, &DEPTH, &SITEID)
new 1: insert into HABITAT values(1004, 'Dessert', 20, 40, 104)

1 row created.

SQL> select * from habitat;

ENCLOSURE_ID BIOME                SQFT    DEPTH    SITEID
-----
1001 Tropical Rain Forest        50       10      102
1002 Grassland                   50       10      101
1003 Ocean                       100      20      103
1004 Dessert                      20       40      104

SQL> insert into ANIMAL values(&ANIMALID, '&NAME', &AGE, '&SEX', &HEIGHT, &WEIGHT, '&SPECIES', &EAT_FREQ_WEEK, &EAT_AMOUNT, &ENCLOSURE_ID);
Enter value for animalid: 101
Enter value for name: Johnny
Enter value for age: 5
Enter value for sex: M
Enter value for height: 50
Enter value for weight: 10
Enter value for species: Duck
Enter value for eat_freq_week: 20
Enter value for eat_amount: 2
Enter value for enclosure_id: 1001
old 1: insert into ANIMAL values(&ANIMALID, '&NAME', &AGE, '&SEX', &HEIGHT, &WEIGHT, '&SPECIES', &EAT_FREQ_WEEK, &EAT_AMOUNT, &ENCLOSURE_ID)
new 1: insert into ANIMAL values(101, 'Johnny', 5, 'M', 50, 10, 'Duck', 20, 2, 1001)

1 row created.

SQL> /
Enter value for animalid: 102
Enter value for name: Chandler
Enter value for age: 8
Enter value for sex: M
Enter value for height: 400
Enter value for weight: 300
Enter value for species: Giraffe
Enter value for eat_freq_week: 10
Enter value for eat_amount: 3
Enter value for enclosure_id: 1002
old 1: insert into ANIMAL values(&ANIMALID, '&NAME', &AGE, '&SEX', &HEIGHT, &WEIGHT, '&SPECIES', &EAT_FREQ_WEEK, &EAT_AMOUNT, &ENCLOSURE_ID)
new 1: insert into ANIMAL values(102, 'Chandler', 8, 'M', 400, 300, 'Giraffe', 10, 3, 1002)

1 row created.

SQL> /
Enter value for animalid: 104
Enter value for name: Ross
Enter value for age: 10
Enter value for sex: M
Enter value for height: 150
Enter value for weight: 400
Enter value for species: Lion
Enter value for eat_freq_week: 15
Enter value for eat_amount: 6
Enter value for enclosure_id: 1004
old 1: insert into ANIMAL values(&ANIMALID, '&NAME', &AGE, '&SEX', &HEIGHT, &WEIGHT, '&SPECIES', &EAT_FREQ_WEEK, &EAT_AMOUNT, &ENCLOSURE_ID)
new 1: insert into ANIMAL values(104, 'Ross', 10, 'M', 150, 400, 'Lion', 15, 6, 1004)

1 row created.

SQL> select * from animal;

ANIMALID NAME                AGE SEX                HEIGHT
-----
WEIGHT SPECIES                EAT_FREQ_WEEK EAT_AMOUNT ENCLOSURE_ID
-----
101 Johnny                    5 M                    50
10 Duck                       20                    2      1001
102 Chandler                  8 M                    400
300 Giraffe                   10                    3      1002
103 Tessa                     7 F                    120
300 Tiger                     15                    5      1003

ANIMALID NAME                AGE SEX                HEIGHT
-----
WEIGHT SPECIES                EAT_FREQ_WEEK EAT_AMOUNT ENCLOSURE_ID
-----
104 Ross                      10 M                    150
400 Lion                      15                    6      1004

```

```

SQL> /
Enter value for fname: willy
Enter value for lname: wonka
Enter value for walkeetalkeeno: 2
old 1: insert into EMPLOYEECOMMUNICATION values('&FNAME', '&LNAME', &WALKEETALKEENO)
new 1: insert into EMPLOYEECOMMUNICATION values('willy', 'wonka', 2)

1 row created.

SQL> /
Enter value for fname: jane
Enter value for lname: goodall
Enter value for walkeetalkeeno: 3
old 1: insert into EMPLOYEECOMMUNICATION values('&FNAME', '&LNAME', &WALKEETALKEENO)
new 1: insert into EMPLOYEECOMMUNICATION values('jane', 'goodall', 3)

1 row created.

SQL> /
Enter value for fname: mary
Enter value for lname: poppins
Enter value for walkeetalkeeno: 4
old 1: insert into EMPLOYEECOMMUNICATION values('&FNAME', '&LNAME', &WALKEETALKEENO)
new 1: insert into EMPLOYEECOMMUNICATION values('mary', 'poppins', 4)

1 row created.

SQL> select * from employeecommunication;

FNAME          LNAME          WALKEETALKEENO
-----
newton         scamander             1
willy          wonka                 2
jane           goodall              3
mary           poppins              4

```

---

```

SQL> /
Enter value for fname: newton
Enter value for lname: scamander
Enter value for empid: 1246545423
Enter value for pay: 75
Enter value for zooaddress: 80 Mandai Lake Rd
old 1: insert into EMPLOYEE values('&FNAME', '&LNAME', &EMPID, &PAY, '&ZOOADDRESS')
new 1: insert into EMPLOYEE values('newton', 'scamander', 1246545423, 75, '80 Mandai Lake Rd')

1 row created.

SQL> /
Enter value for fname: mary
Enter value for lname: poppins
Enter value for empid: 1367865421
Enter value for pay: 0
Enter value for zooaddress: London NW1 4RY
old 1: insert into EMPLOYEE values('&FNAME', '&LNAME', &EMPID, &PAY, '&ZOOADDRESS')
new 1: insert into EMPLOYEE values('mary', 'poppins', 1367865421, 0, 'London NW1 4RY')

1 row created.

SQL> select * from employee;

FNAME          LNAME          EMPID          PAY
-----
ZOOADDRESS
-----
newton         scamander             1234567890          0
2000 Meadowvale Rd
willy          wonka                 2345678901         100
2000 Meadowvale Rd
newton         scamander             1246545423          75
80 Mandai Lake Rd

FNAME          LNAME          EMPID          PAY
-----
ZOOADDRESS
-----
mary           poppins             1367865421          0

```

```

SQL> /
Enter value for duty: clean and feed
Enter value for empid: 2345678901
old 1: insert into KEEPER values('&DUTY', &EMPID)
new 1: insert into KEEPER values('clean and feed', 2345678901)

1 row created.

SQL> /
Enter value for duty: clean
Enter value for empid: 1246545423
old 1: insert into KEEPER values('&DUTY', &EMPID)
new 1: insert into KEEPER values('clean', 1246545423)

1 row created.

SQL> /
Enter value for duty: feed
Enter value for empid: 1367865421
old 1: insert into KEEPER values('&DUTY', &EMPID)
new 1: insert into KEEPER values('feed', 1367865421)

1 row created.

SQL> select * from keeper;

DUTY                                EMPID
-----
clean                                1234567890
clean and feed                       2345678901
clean                                1246545423
feed                                  1367865421

SQL> insert into TRAINER values('&SPECIALITY', &EMPID);
Enter value for speciality: hoop jumping
Enter value for empid: 1234567890
old 1: insert into TRAINER values('&SPECIALITY', &EMPID)
new 1: insert into TRAINER values('hoop jumping', 1234567890)

1 row created.

SQL> /
Enter value for speciality: general
Enter value for empid: 1246545423
old 1: insert into TRAINER values('&SPECIALITY', &EMPID)
new 1: insert into TRAINER values('general', 1246545423)

1 row created.

SQL> /
Enter value for speciality: hoop jumping
Enter value for empid: 1367865421
old 1: insert into TRAINER values('&SPECIALITY', &EMPID)
new 1: insert into TRAINER values('hoop jumping', 1367865421)

1 row created.

SQL> select * from trainer;

SPECIALITY                          EMPID
-----
hoop jumping                         1234567890
balancing                           2345678901
general                             1246545423
hoop jumping                         1367865421

```

```

SQL> /
Enter value for foodid: 13
Enter value for name: fish
Enter value for stock_serving: 32
Enter value for date_purchased: 08-nov-18
Enter value for date_expired: 31-mar-21
Enter value for siteid: 103
old 1: insert into FOOD values(&FOODID, '&NAME', &STOCK_SERVING, '&DATE_PURCHASED', '&DATE_EXPIRED', &SITEID)
new 1: insert into FOOD values(13, 'fish', 32, '08-nov-18', '31-mar-21', 103)

1 row created.

SQL> /
Enter value for foodid: 14
Enter value for name: eggs
Enter value for stock_serving: 99
Enter value for date_purchased: 09-nov-18
Enter value for date_expired: 10-dec-18
Enter value for siteid: 104
old 1: insert into FOOD values(&FOODID, '&NAME', &STOCK_SERVING, '&DATE_PURCHASED', '&DATE_EXPIRED', &SITEID)
new 1: insert into FOOD values(14, 'eggs', 99, '09-nov-18', '10-dec-18', 104)

1 row created.

SQL> select * from food;

   FOODID NAME                STOCK_SERVING DATE_PURC DATE_EXPI SITEID
-----
      11 frozen chicken             80 08-NOV-18 10-AUG-20     101
      12 bamboo                     69 21-OCT-18 10-MAR-19     102
      13 fish                       32 08-NOV-18 31-MAR-21     103
      14 eggs                       99 09-NOV-18 10-DEC-18     104

SQL> /
Enter value for start_time: 17:30:00
Enter value for duration: 25
Enter value for name: Hoop-Jumping Tigers
Enter value for type: Entertainment
Enter value for siteid: 103
old 1: insert into SHOW values('&START_TIME', &DURATION, '&NAME', '&TYPE', &SITEID)
new 1: insert into SHOW values('17:30:00', 25, 'Hoop-Jumping Tigers', 'Entertainment', 103)

1 row created.

SQL> /
Enter value for start_time: 15:00:00
Enter value for duration: 10
Enter value for name: monkey tricks
Enter value for type: Entertainment
Enter value for siteid: 104
old 1: insert into SHOW values('&START_TIME', &DURATION, '&NAME', '&TYPE', &SITEID)
new 1: insert into SHOW values('15:00:00', 10, 'monkey tricks', 'Entertainment', 104)

1 row created.

SQL> /
Enter value for start_time: 12:00:00
Enter value for duration: 12
Enter value for name: Meet A Panda
Enter value for type: Educational
Enter value for siteid: 102
old 1: insert into SHOW values('&START_TIME', &DURATION, '&NAME', '&TYPE', &SITEID)
new 1: insert into SHOW values('12:00:00', 12, 'Meet A Panda', 'Educational', 102)

1 row created.

SQL> select * from show;

START_TI  DURATION NAME                TYPE                SITEID
-----
18:30:00    50 Play With Ducks      Kids                101
17:30:00    25 Hoop-Jumping Tigers  Entertainment       103
15:00:00    10 monkey tricks        Entertainment       104
12:00:00    12 Meet A Panda         Educational          102

SQL> /
Enter value for start_time: 17:30:00
Enter value for show_name: Hoop-Jumping Tigers
Enter value for empid: 2345678901
Enter value for animalid: 103
Enter value for role: guide tigers
old 1: insert into PERFORMS values ('&START_TIME', '&SHOW_NAME', &EMPID, &ANIMALID, '&ROLE')
new 1: insert into PERFORMS values ('17:30:00', 'Hoop-Jumping Tigers', 2345678901, 103, 'guide tigers')

1 row created.

SQL> /
Enter value for start_time: 15:00:00
Enter value for show_name: monkey tricks
Enter value for empid: 1246545423
Enter value for animalid: 102
Enter value for role: prompt monkeys
old 1: insert into PERFORMS values ('&START_TIME', '&SHOW_NAME', &EMPID, &ANIMALID, '&ROLE')
new 1: insert into PERFORMS values ('15:00:00', 'monkey tricks', 1246545423, 102, 'prompt monkeys')

1 row created.

```

```

SQL> select * from performs;

START_TI  SHOW_NAME                                EMPID  ANIMALID
-----
ROLE
-----
18:30:00 Play With Ducks                        1234567890      101
watch over ducks

17:30:00 Hoop-Jumping Tigers                    2345678901      103
guide tigers

15:00:00 monkey tricks                          1246545423      102
prompt monkeys

START_TI  SHOW_NAME                                EMPID  ANIMALID
-----
ROLE
-----
12:00:00 Meet A Panda                          1367865421      104
watch over panda

SQL> insert into CARESFOR values (&EMPID, &ANIMALID);
Enter value for empid: 1234567890
Enter value for animalid: 101
old 1: insert into CARESFOR values (&EMPID, &ANIMALID)
new 1: insert into CARESFOR values (1234567890, 101)

1 row created.

SQL> /
Enter value for empid: 2345678901
Enter value for animalid: 102
old 1: insert into CARESFOR values (&EMPID, &ANIMALID)
new 1: insert into CARESFOR values (2345678901, 102)

1 row created.

SQL> /
Enter value for empid: 1246545423
Enter value for animalid: 103
old 1: insert into CARESFOR values (&EMPID, &ANIMALID)
new 1: insert into CARESFOR values (1246545423, 103)

1 row created.

SQL> /
Enter value for empid: 1367865421
Enter value for animalid: 104
old 1: insert into CARESFOR values (&EMPID, &ANIMALID)
new 1: insert into CARESFOR values (1367865421, 104)

1 row created.

SQL> select * from caresfor;

      EMPID  ANIMALID
-----
1234567890      101
1246545423      103
1367865421      104
2345678901      102

SQL> insert into TRAINS values (&EMPID, &ANIMALID, '&SKILLS');
Enter value for empid: 1234567890
Enter value for animalid: 102
Enter value for skills: understand trainer gestures
old 1: insert into TRAINS values (&EMPID, &ANIMALID, '&SKILLS')
new 1: insert into TRAINS values (1234567890, 102, 'understand trainer gestures')

1 row created.

```



```

SQL> /
Enter value for empid: 2345678901
Enter value for animalid: 103
Enter value for skills: jump through hoops when instructed
old 1: insert into TRAINS values (&EMPID, &ANIMALID, '&SKILLS')
new 1: insert into TRAINS values (2345678901, 103, 'jump through hoops when instructed')

1 row created.

SQL> /
Enter value for empid: 1367865421
Enter value for animalid: 102
Enter value for skills: juggle 3 balls
old 1: insert into TRAINS values (&EMPID, &ANIMALID, '&SKILLS')
new 1: insert into TRAINS values (1367865421, 102, 'juggle 3 balls')

1 row created.

SQL> /
Enter value for empid: 1246545423
Enter value for animalid: 104
Enter value for skills: wave at visitors
old 1: insert into TRAINS values (&EMPID, &ANIMALID, '&SKILLS')
new 1: insert into TRAINS values (1246545423, 104, 'wave at visitors')

1 row created.

SQL> select * from trains;

      EMPID      ANIMALID  SKILLS
-----
1234567890      102 understand trainer gestures
2345678901      103 jump through hoops when instructed
1367865421      102 juggle 3 balls
1246545423      104 wave at visitors

SQL> /
Enter value for zoo_from_address: 80 Mandai Lake Rd
Enter value for zoo_to_address: 2000 Meadowvale Rd
Enter value for animalid: 101
Enter value for trade_date: 21-sep-18
old 1: insert into TRADES values ('&ZOO_FROM_ADDRESS', '&ZOO_TO_ADDRESS', &ANIMALID, '&TRADE_DATE')
new 1: insert into TRADES values ('80 Mandai Lake Rd', '2000 Meadowvale Rd', 101, '21-sep-18')

1 row created.

SQL> /
Enter value for zoo_from_address: London NW1 4RY
Enter value for zoo_to_address: 2000 Meadowvale Rd
Enter value for animalid: 102
Enter value for trade_date: 01-jan-18
old 1: insert into TRADES values ('&ZOO_FROM_ADDRESS', '&ZOO_TO_ADDRESS', &ANIMALID, '&TRADE_DATE')
new 1: insert into TRADES values ('London NW1 4RY', '2000 Meadowvale Rd', 102, '01-jan-18')

1 row created.

SQL> select * from trades;

ZOO_FROM_ADDRESS
-----
ZOO_TO_ADDRESS      ANIMALID  TRADE_DAT
-----
2000 Meadowvale Rd
80 Mandai Lake Rd      103 21-SEP-16

2000 Meadowvale Rd
15500 San Pasqual Valley Rd      104 31-MAR-17

80 Mandai Lake Rd
2000 Meadowvale Rd      101 21-SEP-18

ZOO_FROM_ADDRESS
-----
ZOO_TO_ADDRESS      ANIMALID  TRADE_DAT
-----
London NW1 4RY
2000 Meadowvale Rd      102 01-JAN-18

```



```
SQL> insert into EATS values (&animalid,&foodid);
Enter value for animalid: 101
Enter value for foodid: 14
old 1: insert into EATS values (&animalid,&foodid)
new 1: insert into EATS values (101,14)

1 row created.

SQL> /
Enter value for animalid: 102
Enter value for foodid: 12
old 1: insert into EATS values (&animalid,&foodid)
new 1: insert into EATS values (102,12)

1 row created.

SQL> /
Enter value for animalid: 103
Enter value for foodid: 11
old 1: insert into EATS values (&animalid,&foodid)
new 1: insert into EATS values (103,11)

1 row created.

SQL> /
Enter value for animalid: 104
Enter value for foodid: 14
old 1: insert into EATS values (&animalid,&foodid)
new 1: insert into EATS values (104,14)

1 row created.

SQL> /
Enter value for animalid: 104
Enter value for foodid: 11
old 1: insert into EATS values (&animalid,&foodid)
new 1: insert into EATS values (104,11)

1 row created.

SQL> /
Enter value for animalid: 103
Enter value for foodid: 12
old 1: insert into EATS values (&animalid,&foodid)
new 1: insert into EATS values (103,12)

1 row created.

SQL> /
Enter value for animalid: 101
Enter value for foodid: 11
old 1: insert into EATS values (&animalid,&foodid)
new 1: insert into EATS values (101,11)

1 row created.

SQL> select * from eats;

  ANIMALID      FOODID
-----
      101         11
      101         14
      102         12
      103         11
      103         12
      104         11
      104         14

7 rows selected.
```

# IMPLEMENTATION

## Front end programs and its connectivity

**Java Database Connectivity (JDBC)** is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
public void connectToDB()
{
    try
    {
        connection
        DriverManager.getConnection( "jdbc:oracle:thin:@localhost:1521:xe", "adms", "admin");
        statement= connection.createStatement();

    }
    catch (SQLException connectException)
    {
        System.out.println(connectException.getMessage());
        System.out.println(connectException.getSQLState());
        System.out.println(connectException.getErrorCode());
        System.exit(1);
    }
}
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

## Mandatory JAVA package Imports:

Below mentioned java packages should be placed at the top in every .java file of this application.

```
package AnimalDepot;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.awt.*;
import javax.swing.table.*;
import java.sql.*;
```

### DBConnection.java :

```
public class DBConnection {

    private static DBConnection driver;
    private Connection con;
    private Statement stmt;
    private DBConnection() {
        try {
            DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
        } catch (SQLException e) {
            System.out.println("Message: " + e.getMessage());
        }
        //connect to Oracle
        String username = "AnimalsDB";
        String password = "123";
        try {
            con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", username,
password);
            System.out.println("\nConnected to Oracle DataBase");
            stmt = con.createStatement();
        } catch (SQLException connectException) {
            System.out.println(connectException.getMessage());
            System.out.println(connectException.getSQLState());
            System.out.println(connectException.getErrorCode());
            System.exit(1);
        }
        //insertData();
    }
    public static DBConnection getInstance() {
        if (driver == null)
            driver = new DBConnection();
        return driver;
    }
    //execute any SQL statements alter the database (update, delete, insert)
    public int executeAlter(String sqlstmt) {
        try {
            int a=stmt.executeUpdate(sqlstmt);
            con.commit();
            return a;
        } catch (Exception e) { return 0; }
    }
    //execute query statements
    public ResultSet executeQuery(String sqlstmt) {
        try {
            return stmt.executeQuery(sqlstmt);
        } catch (SQLException e) {
            System.out.println("Message: " + e.getMessage() + "\nUnable to execute:
```

```

" + sqlstmt);
        return null;
    }
}
//disconnect from Oracle database
public void disconnect() {
    try {
        con.close();
    } catch (SQLException e) {
        System.out.println("Message: " + e.getMessage());
        System.out.println("\nFailed to disconnect from Oracle");
    }
}
}
}

```

### **MainPage.java :**

```

public class MainPage extends JFrame {

    private JPanel contentPane, panel;
    protected JTabbedPane JTabbedPane1;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    new MainPage().setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public MainPage() {
        setBackground(new Color(0, 128, 128));
        setForeground(new Color(255, 0, 0));
        setFont(new Font("Microsoft JhengHei UI", Font.BOLD, 12));

        setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\\\Users\\cheraka\\eclipse-workspace\\java_eclipse\\images\\icons8-animal-shelter-20.png"));
        setTitle("Animal Depot - main page");
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        addWindowListener((WindowListener) new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                if(JOptionPane.showConfirmDialog(null, "Are you sure? you want to close Application!") == JOptionPane.YES_OPTION){
                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                }
            }
        });
        setBounds(100, 100, 773, 495);
        contentPane = new JPanel();
        contentPane.setBackground(new Color(128, 128, 0));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        panel = new JPanel();
        panel.setBounds(new Rectangle(5, 5, 742, 41));
    }
}

```

```

        contentPane.add(panel);
        panel.setLayout(new GridLayout(0, 1, 0, 0));
        JMenuBar menuBar = new JMenuBar();
        menuBar.setBorder(new MatteBorder(0, 0, 3, 0, (Color) new Color(128,
0, 0)));
        menuBar.setMargin(new Insets(3, 3, 3, 3));
        panel.add(menuBar);
        JMenu mnNewMenu_1 = new JMenu("Employee");
        mnNewMenu_1.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent arg0) {
                if(Login.isLoggedIn)
                {
                    dispose();
                    new Employee().setVisible(true);
                }else {
                    JOptionPane.showMessageDialog(null,"You are not
logged inn, Please Login!","Alert",JOptionPane.WARNING_MESSAGE);
                }
            }
        });
        JMenu mnNewMenu = new JMenu("Login");
        mnNewMenu.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                dispose();
                new Login().setVisible(true);
            }
        });
        mnNewMenu.setForeground(new Color(75, 0, 130));
        menuBar.add(mnNewMenu);
        JMenu mnNewMenu_4 = new JMenu("Log out");
        mnNewMenu_4.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                if(Login.isLoggedIn) {
                    int a=JOptionPane.showConfirmDialog(null, "Are you
sure, You want to logout!");
                    if(a==JOptionPane.YES_OPTION)
                    {
                        Login.isLoggedIn=false;

                        JOptionPane.showMessageDialog(null,"Successfully Logged Out!");
                    }
                }else {
                    JOptionPane.showMessageDialog(null,"You are not
logged inn, Please Login!","Alert",JOptionPane.WARNING_MESSAGE);
                }
            }
        });
        mnNewMenu_4.setForeground(new Color(128, 0, 0));
        menuBar.add(mnNewMenu_4);
        mnNewMenu_1.setForeground(new Color(0, 128, 0));
        menuBar.add(mnNewMenu_1);
        JSeparator separator_4 = new JSeparator();
        mnNewMenu_1.add(separator_4);
        JSeparator separator_5 = new JSeparator();
        mnNewMenu_1.add(separator_5);
        JMenu mnNewMenu_3 = new JMenu("Animals");
        mnNewMenu_3.addMouseListener(new MouseAdapter() {
            @Override

```

```

        public void mouseClicked(MouseEvent arg0) {
            if(Login.isLoggedIn)
            {
                dispose();
                new Animal().setVisible(true);
            }else {
                JOptionPane.showMessageDialog(null,"You are not
logged inn, Please Login!", "Alert",JOptionPane.WARNING_MESSAGE);
            }
        }
    });
    mnNewMenu_3.setForeground(new Color(0, 128, 0));
    menuBar.add(mnNewMenu_3);
    JSeparator separator_2 = new JSeparator();
    mnNewMenu_3.add(separator_2);
    JSeparator separator_3 = new JSeparator();
    mnNewMenu_3.add(separator_3);
    JMenu mnNewMenu_2 = new JMenu("View");
    mnNewMenu_2.setForeground(new Color(0, 128, 0));
    menuBar.add(mnNewMenu_2);
    JMenuItem mntmNewMenuItem_8 = new JMenuItem("Zoo");
    mntmNewMenuItem_8.setBackground(Color.PINK);
    mntmNewMenuItem_8.setForeground(Color.BLACK);
    mntmNewMenuItem_8.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dispose();
            new Zoo().setVisible(true);
        }
    });
    mnNewMenu_2.add(mntmNewMenuItem_8);
    JSeparator separator = new JSeparator();
    mnNewMenu_2.add(separator);
    JMenuItem mntmNewMenuItem_2 = new JMenuItem("Food");
    mntmNewMenuItem_2.setBackground(Color.PINK);
    mntmNewMenuItem_2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dispose();
            new Food().setVisible(true);
        }
    });
    mnNewMenu_2.add(mntmNewMenuItem_2);
    JSeparator separator_6 = new JSeparator();
    mnNewMenu_2.add(separator_6);
    JMenuItem mntmNewMenuItem_9 = new JMenuItem("Show Details");
    mntmNewMenuItem_9.setBackground(Color.PINK);
    mntmNewMenuItem_9.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dispose();
            new Show().setVisible(true);
        }
    });
    mnNewMenu_2.add(mntmNewMenuItem_9);
    JMenuItem mntmNewMenuItem = new JMenuItem("Locations");
    mntmNewMenuItem.setBackground(Color.PINK);
    mntmNewMenuItem.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dispose();
            new Location().setVisible(true);
        }
    });
    JSeparator separator_7 = new JSeparator();
    mnNewMenu_2.add(separator_7);

```

```

        mnNewMenu_2.add(mntmNewMenuItem);
        JPanel panel_1 = new JPanel();
        panel_1.setBackground(Color.PINK);
        panel_1.setBorder(new MatteBorder(2, 3, 3, 3, (Color) Color.CYAN));
        panel_1.setBounds(26, 57, 697, 388);
        contentPane.add(panel_1);
        panel_1.setLayout(null);
        JPanel panel_2 = new JPanel();
        panel_2.setBackground(SystemColor.textHighlight);
        panel_2.setBorder(new MatteBorder(3, 3, 3, 3, (Color) Color.CYAN));
        panel_2.setBounds(135, 96, 476, 212);
        panel_1.add(panel_2);
        panel_2.setLayout(null);
        JLabel lblNewLabel = new JLabel("ANIMAL DEPOT MANAGEMENT SYSTEM");
        lblNewLabel.setBackground(Color.MAGENTA);
        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 20));
        lblNewLabel.setBounds(34, 71, 406, 82);
        panel_2.add(lblNewLabel);
    }
}

```

### **Login.java :**

```

public class Login extends JFrame {

    public JPanel contentPane;
    public JTextField txtUsername, textField;
    public String UserName = "admin", password = "admin";
    public static boolean isLoggedIn = false;
    public Login() {

        setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\cheraka\\eclipse-workspace\\java_eclipse\\images\\icons8-animal-shelter-20.png"));
        setTitle("Animal Depot - Login Page");
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        addWindowListener((WindowListener) new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                if(JOptionPane.showConfirmDialog(null,"Are you sure? you want to close Application!")==JOptionPane.YES_OPTION){
                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                }
            }
        });
        setBounds(100, 100, 773, 495);
        contentPane = new JPanel();
        contentPane.setToolTipText("Login Page\r\n");
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        JPanel panel_2 = new JPanel();
        panel_2.setBorder(new MatteBorder(0, 0, 2, 0, (Color) new Color(75, 0, 130)));
        panel_2.setBounds(5, 51, 742, 394);
        panel_2.setBackground(new Color(0, 191, 255));
        contentPane.add(panel_2);
        panel_2.setLayout(null);
        txtUsername = new JTextField();
        txtUsername.setHorizontalAlignment(SwingConstants.CENTER);
        txtUsername.setBackground(new Color(0, 191, 255));
        txtUsername.setBorder(new MatteBorder(0, 0, 2, 0, (Color) new Color(106, 90, 205)));
    }
}

```

```

txtUsername.setFont(new Font("Times New Roman", Font.BOLD, 20));
txtUsername.setToolTipText("If not registered, go to 'Registration'
page.");
txtUsername.setBounds(327, 78, 267, 23);
panel_2.add(txtUsername);
txtUsername.setColumns(10);
textField = new JTextField();
textField.setFont(new Font("Times New Roman", Font.BOLD, 20));
textField.setHorizontalAlignment(SwingConstants.CENTER);
textField.setBorder(new MatteBorder(0, 0, 2, 0, (Color) new Color(123,
104, 238)));
textField.setBackground(new Color(0, 191, 255));
textField.setBounds(327, 162, 267, 20);
panel_2.add(textField);
textField.setColumns(10);
JButton btnNewButton = new JButton("");
btnNewButton.setBorder(new BevelBorder(BevelBorder.RAISED, new
Color(0, 191, 255), null, null, null));
btnNewButton.setBackground(new Color(253, 245, 230));
btnNewButton.setToolTipText("Login");
btnNewButton.setIcon(new ImageIcon("C:\\Users\\cheraka\\eclipse-
workspace\\java_eclipse\\images\\icons8-login-20.png"));
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton.setForeground(Color.RED);
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        boolean flag1=true,flag2=true;
        if(txtUsername.getText().equals(""))
            flag1=false;
        if(textField.getText().equals(""))
            flag2=false;
        if(flag1==false && flag2==false)
            JOptionPane.showMessageDialog(null,"Username and
Password should not be empty!","Alert",JOptionPane.WARNING_MESSAGE);
        else if(flag1==true && flag2==false)
            JOptionPane.showMessageDialog(null,"Password
should not be empty!","Alert",JOptionPane.WARNING_MESSAGE);
        else if(flag1==false && flag2==true)
            JOptionPane.showMessageDialog(null,"Username
should not be empty!","Alert",JOptionPane.WARNING_MESSAGE);
        else
        {
            flag1=flag2=false;
            if(txtUsername.getText().equals(UserName))
                flag1=true;
            if(textField.getText().equals(password))
                flag2=true;
            if(flag1==false && flag2==false)
                JOptionPane.showMessageDialog(null,"Invalid
Username and Password!","Alert",JOptionPane.WARNING_MESSAGE);
            else if(flag1==true && flag2==false)
                JOptionPane.showMessageDialog(null,"Invalid
Password!","Alert",JOptionPane.WARNING_MESSAGE);
            else if(flag1==false && flag2==true)
                JOptionPane.showMessageDialog(null,"Invalid
Username!","Alert",JOptionPane.WARNING_MESSAGE);
            else {

JOptionPane.showMessageDialog(null,"Successfully Logged in!");

JOptionPane.showMessageDialog(null,"Redirecting to Main
Page","Alert",JOptionPane.WARNING_MESSAGE);

```



```

        isLoggedIn = true;
        dispose();
        new MainPage().setVisible(true);
    }
}

});
btnNewButton.setBounds(505, 261, 89, 33);
panel_2.add(btnNewButton);
JButton btnNewButton_1 = new JButton("");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        dispose();
        new MainPage().setVisible(true);
    }
});
btnNewButton_1.setBorder(new BevelBorder(BevelBorder.RAISED, new
Color(0, 191, 255), null, null, null));
btnNewButton_1.setBackground(new Color(253, 245, 230));
btnNewButton_1.setToolTipText("Go back to Main Page");
btnNewButton_1.setIcon(new ImageIcon("C:\\Users\\cheraka\\eclipse-
workspace\\java_eclipse\\images\\icons8-back-20.png"));
btnNewButton_1.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton_1.setForeground(new Color(0, 0, 0));
btnNewButton_1.setBounds(145, 261, 107, 33);
panel_2.add(btnNewButton_1);
JLabel lblNewLabel = new JLabel("");
lblNewLabel.setToolTipText("User Name");
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setIcon(new ImageIcon("C:\\Users\\cheraka\\eclipse-
workspace\\java_eclipse\\images\\change-user-20.png"));
lblNewLabel.setBounds(145, 78, 46, 23);
panel_2.add(lblNewLabel);
JLabel lblNewLabel_1 = new JLabel("");
lblNewLabel_1.setToolTipText("Password");
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setIcon(new ImageIcon("C:\\Users\\cheraka\\eclipse-
workspace\\java_eclipse\\images\\icons8-forgot-password-20.png"));
lblNewLabel_1.setBounds(145, 162, 46, 23);
panel_2.add(lblNewLabel_1);
JPanel panel = new JPanel();
panel.setBackground(new Color(128, 128, 128));
panel.setBounds(5, 11, 742, 40);
contentPane.add(panel);
panel.setLayout(null);
JLabel lblNewLabel_2 = new JLabel("LOGIN");
lblNewLabel_2.setBounds(243, 0, 212, 40);
panel.add(lblNewLabel_2);
lblNewLabel_2.setForeground(new Color(255, 255, 255));
lblNewLabel_2.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_2.setFont(new Font("Tahoma", Font.BOLD, 25));
}
}

```

### **Employee.java :**

```

public class Employee extends JFrame {

    private JPanel contentPane;
    private JTable table,table_1, table_2,table_3;
    private JTextField
textField,textField_1,textField_2,textField_3,textField_4,textField_5,textField_12;

```

```

private JTextField
textField_6,textField_7,textField_8,textField_9,textField_10,textField_11,textField
_13;
private DefaultTableModel dm, dm1, dm2, dm3;
private DBConnection jdbcConnection;
public Employee() {
    jdbcConnection=DBConnection.getInstance();
    dm = new DefaultTableModel();
    dm1 = new DefaultTableModel();
    dm2 = new DefaultTableModel();
    dm3 = new DefaultTableModel();

    setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\cheraka\\eclip
se-workspace\\java_eclipse\\images\\icons8-animal-shelter-20.png"));
    setTitle("Animal Depot - Employee Manager");
    setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
    addWindowListener((WindowListener) new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            if(JOptionPane.showConfirmDialog(null,"Are you sure? you
want to close Application!")==JOptionPane.YES_OPTION){
                jdbcConnection.disconnect();
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            }
        }
    });
    setBounds(100, 100, 760, 478);
    JMenuBar menuBar = new JMenuBar();
    menuBar.setBackground(new Color(218, 165, 32));
    menuBar.setForeground(new Color(255, 140, 0));
    setJMenuBar(menuBar);

    JMenu mnNewMenu = new JMenu("Main Page");
    mnNewMenu.setForeground(new Color(0, 128, 0));
    mnNewMenu.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            dispose();
            new MainPage().setVisible(true);
        }
    });
    menuBar.add(mnNewMenu);
    JMenu mnNewMenu_1 = new JMenu("Logout");
    mnNewMenu_1.setForeground(new Color(0, 128, 0));
    mnNewMenu_1.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            Login.isLoggedIn=false;
            new MainPage().setVisible(true);
        }
    });
    menuBar.add(mnNewMenu_1);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(0, 128, 128));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
    tabbedPane.setBounds(10, 11, 724, 396);
    contentPane.add(tabbedPane);
    JPanel panel = new JPanel();
    panel.setForeground(new Color(0, 0, 0));
    panel.setBackground(new Color(0, 255, 255));

```

```

        tabbedPane.addTab("Insert", null, panel, null);
        tabbedPane.setForegroundAt(0, new Color(128, 0, 0));
        panel.setLayout(null);
        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(10, 123, 699, 234);
        panel.add(scrollPane);
        table = new JTable();
        table.setFont(new Font("Tahoma", Font.BOLD, 11));
        scrollPane.setViewportView(table);
        JLabel lblNewLabel_7 = new JLabel("First Name");
        lblNewLabel_7.setBounds(10, 11, 89, 14);
        panel.add(lblNewLabel_7);
        JLabel lblNewLabel_8 = new JLabel("Last Name");
        lblNewLabel_8.setBounds(10, 36, 89, 14);
        panel.add(lblNewLabel_8);
        JLabel lblNewLabel_9 = new JLabel("Employee ID");
        lblNewLabel_9.setBounds(10, 61, 89, 14);
        panel.add(lblNewLabel_9);
        JLabel lblNewLabel_10 = new JLabel("Walkee Talkee ID");
        lblNewLabel_10.setBounds(345, 11, 125, 14);
        panel.add(lblNewLabel_10);
        JLabel lblNewLabel_11 = new JLabel("Zoo Address");
        lblNewLabel_11.setBounds(345, 36, 89, 14);
        panel.add(lblNewLabel_11);
        JLabel lblNewLabel_12 = new JLabel("Payment");
        lblNewLabel_12.setBounds(345, 61, 89, 14);
        panel.add(lblNewLabel_12);
        textField_7 = new JTextField();
        textField_7.setBounds(154, 8, 110, 20);
        panel.add(textField_7);
        textField_7.setColumns(10);
        textField_8 = new JTextField();
        textField_8.setBounds(154, 33, 110, 20);
        panel.add(textField_8);
        textField_8.setColumns(10);
        textField_9 = new JTextField();
        textField_9.setBounds(154, 58, 110, 20);
        panel.add(textField_9);
        textField_9.setColumns(10);
        textField_10 = new JTextField();
        textField_10.setBounds(599, 8, 110, 20);
        panel.add(textField_10);
        textField_10.setColumns(10);
        textField_11 = new JTextField();
        textField_11.setBounds(599, 58, 110, 20);
        panel.add(textField_11);
        textField_11.setColumns(10);
        JComboBox<String> comboBox = new JComboBox<>();
        comboBox.setBounds(599, 33, 110, 20);
        panel.add(comboBox);
        JButton btnNewButton_5 = new JButton("Clear All");
        btnNewButton_5.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                textField_7.setText("");textField_8.setText("");textField_9.setText("");
                textField_10.setText("");textField_11.setText("");

            }
        });
        btnNewButton_5.setBounds(10, 89, 89, 23);
        panel.add(btnNewButton_5);
        try {
            ResultSet rSet=jDbConnection.executeQuery("select address from

```

```

zoo");

        while(rSet.next())
        {
            comboBox.addItem((String)rSet.getString("address"));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    JButton btnNewButton_6 = new JButton("Submit");
    btnNewButton_6.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            String firstQuery = "insert into employeecommunication values
('"+textField_7.getText()+"', '"+textField_8.getText()+"', ";
            firstQuery+=textField_10.getText()+")";
            int result1 = jdbcConnection.executeAlter(firstQuery);
            if (result1!=0) {
                String reString="insert into employee values
('"+textField_7.getText()+"', '"+textField_8.getText();
                reString+=", '"+textField_9.getText()+",
"+textField_10.getText()+", '"+(String)comboBox.getSelectedItem()+"')";
                JOptionPane.showMessageDialog(null,jdbcConnection.executeAlter(reString));
            } else {
                JOptionPane.showMessageDialog(null,"Insertion
failed!", "Alert", JOptionPane.WARNING_MESSAGE);
            }
        }
    });
    btnNewButton_6.setBounds(154, 89, 89, 23);
    panel.add(btnNewButton_6);
    JButton btnNewButton = new JButton("View Table");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            loadTable(dm, table,"SELECT
EMPID,WALKEETALKEENO,E.FNAME,E.LNAME,PAY,ZOOADDRESS FROM EMPLOYEE E INNER JOIN
EMPLOYEECOMMUNICATION EC ON E.FNAME=EC.FNAME AND E.LNAME=EC.LNAME");
        }
    });
    btnNewButton.setBounds(609, 89, 100, 23);
    panel.add(btnNewButton);

    JButton btnNewButton_7 = new JButton("Clear Table");
    btnNewButton_7.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dm.setColumnCount(0);
            dm.setRowCount(0);
        }
    });
    btnNewButton_7.setBounds(466, 89, 100, 23);
    panel.add(btnNewButton_7);
    JPanel panel_3 = new JPanel();
    panel_3.setBackground(new Color(0, 255, 127));
    tabbedPane.addTab("Update", null, panel_3, null);
    tabbedPane.setForegroundAt(1, new Color(128, 0, 0));
    panel_3.setLayout(null);
    JScrollPane scrollPane_2 = new JScrollPane();
    scrollPane_2.setBounds(10, 101, 699, 256);
    panel_3.add(scrollPane_2);
    table_2 = new JTable();
    table_2.setFont(new Font("Tahoma", Font.BOLD, 11));
    scrollPane_2.setViewportView(table_2);
    JLabel lblNewLabel = new JLabel("First Name");

```

```

        lblNewLabel.setBounds(10, 11, 89, 14);
        panel_3.add(lblNewLabel);
        JLabel lblNewLabel_1 = new JLabel("Last Name");
        lblNewLabel_1.setBounds(10, 36, 89, 14);
        panel_3.add(lblNewLabel_1);
        JLabel lblNewLabel_2 = new JLabel("Employee ID");
        lblNewLabel_2.setBounds(424, 11, 89, 14);
        panel_3.add(lblNewLabel_2);
        JLabel lblNewLabel_3 = new JLabel("Payment");
        lblNewLabel_3.setBounds(424, 36, 89, 14);
        panel_3.add(lblNewLabel_3);
        textField = new JTextField();
        textField.setBounds(156, 8, 124, 20);
        panel_3.add(textField);
        textField.setColumns(10);
        textField_1 = new JTextField();
        textField_1.setBounds(156, 36, 124, 20);
        panel_3.add(textField_1);
        textField_1.setColumns(10);
        textField_2 = new JTextField();
        textField_2.setBounds(585, 8, 124, 20);
        panel_3.add(textField_2);
        textField_2.setColumns(10);
        textField_3 = new JTextField();
        textField_3.setBounds(585, 33, 124, 20);
        panel_3.add(textField_3);
        textField_3.setColumns(10);
        JButton btnNewButton_1 = new JButton("Modify");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String empid=textField_2.getText();
                ResultSet rs = jdbcConnection.executeQuery("select *
from employee where empid = "+empid);
                try {
                    if (rs.next() == false) {
                        JOptionPane.showMessageDialog(null,"Employee Not
Found!");
                    }
                    else {
                        boolean isNameSame = false;
                        String fname=textField.getText();
                        String lname=textField_1.getText();
                        try {
                            isNameSame =
jdbcConnection.executeQuery("select fname, lname from employee where empid =
"+empid+"and fname = '"+fname+"'and lname = '"+lname+"'").next();
                        } catch (SQLException e1) {
                            System.out.println("select employee
query failed");
                        }
                        String pay=textField_3.getText();
                        if (isNameSame) {
                            int updatePayResult =
jdbcConnection.executeAlter("update employee set pay = "+pay+" where empid =
"+empid);
                            if (updatePayResult != 0) {
                                System.out.println("Success!");
                            } else {
                                System.out.println("error: update
pay failed when name is the same");
                            }
                        }
                    }
                }
            }
        });

```

```

        ResultSet result1 =
jdbcTemplate.executeQuery("select fname, lname from employee where empid =
"+empid);

        String oldfname = "";
        String oldlname = "";
        try {
            if (result1.next() == true) {
                oldfname = result1.getString(1);
                oldlname = result1.getString(2);
                System.out.println("names saved");
            } else {
                System.out.println("Invalid query:
looking for old name");
            }
        } catch (SQLException e1) {
            System.out.println("select employee
query failed");
        }
        int walkeetalkeeNum = 0;
        ResultSet rs2 =
jdbcTemplate.executeQuery("select walkeetalkeeno from employeecommunication where
fname = '"+oldfname+"' and lname = '"+oldlname+"'");
        try {
            if (rs2.next() == true) {
                walkeetalkeeNum = rs2.getInt(1);
            } else {
                System.out.println("Invalid query:
selecting walkeetalkeeno");
            }
        } catch (SQLException e1) {
            System.out.println("select
employeecommunication query failed");
        }
        int result4 =
jdbcTemplate.executeAlter("update employee set fname = '"+fname+"', lname =
 '"+lname+"' where empid = "+empid);
        if (result4 != 0) {
            int updatePayResult =
jdbcTemplate.executeAlter("update employee set pay = "+pay+" where empid =
"+empid);
            if (updatePayResult!=0)

                JOptionPane.showMessageDialog(null,"Details Successfully Updated!");
            else

                JOptionPane.showMessageDialog(null,"Employee Not Found");
        }else

            JOptionPane.showMessageDialog(null,"Employee Not Found");
        } catch (SQLException e1) {System.out.println("failed
select for id"); }
    }
});
btnNewButton_1.setBounds(191, 67, 89, 23);
panel_3.add(btnNewButton_1);
JButton btnNewButton_2 = new JButton("Clear All");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText("");
        textField_1.setText("");
        textField_2.setText("");
    }
});

```

```

        textField_3.setText("");
    }
});
btnNewButton_2.setBounds(10, 69, 89, 23);
panel_3.add(btnNewButton_2);
JButton btnNewButton_8 = new JButton("Clear Table");
btnNewButton_8.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dm1.setColumnCount(0);
        dm1.setRowCount(0);
    }
});
btnNewButton_8.setBounds(424, 67, 100, 23);
panel_3.add(btnNewButton_8);
JButton btnNewButton_9 = new JButton("View Table");
btnNewButton_9.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        loadTable(dm1, table_2, "SELECT * FROM EMPLOYEE");
    }
});
btnNewButton_9.setBounds(595, 67, 114, 23);
panel_3.add(btnNewButton_9);
JPanel panel_2 = new JPanel();
panel_2.setBackground(new Color(255, 182, 193));
tabbedPane.addTab("Remove", null, panel_2, null);
tabbedPane.setForegroundAt(2, new Color(128, 0, 0));
panel_2.setLayout(null);
JScrollPane scrollPane_1 = new JScrollPane();
scrollPane_1.setBounds(10, 101, 699, 256);
panel_2.add(scrollPane_1);
table_1 = new JTable();
table_1.setFont(new Font("Tahoma", Font.BOLD, 11));
scrollPane_1.setViewportView(table_1);
JLabel lblNewLabel_4 = new JLabel("First Name");
lblNewLabel_4.setBounds(31, 14, 89, 14);
panel_2.add(lblNewLabel_4);
JLabel lblNewLabel_5 = new JLabel("Last Name");
lblNewLabel_5.setBounds(31, 39, 89, 14);
panel_2.add(lblNewLabel_5);
JLabel lblNewLabel_6 = new JLabel("Employee ID");
lblNewLabel_6.setBounds(31, 70, 89, 14);
panel_2.add(lblNewLabel_6);
JButton btnNewButton_3 = new JButton("Clear All");
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField_4.setText("");
        textField_5.setText("");
        textField_6.setText("");
    }
});
btnNewButton_3.setBounds(610, 11, 99, 23);
panel_2.add(btnNewButton_3);
JButton btnNewButton_4 = new JButton("Delete");
btnNewButton_4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        boolean isTrainer=false;
        // need to see if Employee is trainer or keeper
        boolean isKeeper=false;
        String empid=textField_6.getText();
        try {
            isTrainer = jdbcConnection.executeQuery("select * from
trainer where empid = "+empid).next();

```

```

        } catch (SQLException e1) {System.out.println("select trainer
query failed");}
        try {
            isKeeper = jdbcConnection.executeQuery("select * from
keeper where empid = "+empid).next();
        } catch (SQLException e1) { System.out.println("select keeper
query failed");}
        // Delete from Trainer/Keeper + Employee
        if (isKeeper) {
            jdbcConnection.executeAlter("delete from caresfor where
empid = "+empid);
            jdbcConnection.executeAlter("delete from keeper where empid
= "+empid);
        }
        if (isTrainer) {
            jdbcConnection.executeAlter("delete from trains where empid
= "+empid);
            jdbcConnection.executeAlter("delete from performs where
empid = "+empid);
            jdbcConnection.executeAlter("delete from trainer where empd
= "+empid);
        }
        int result1 = jdbcConnection.executeAlter("delete from employee
where empid = "+empid);
        if (result1 != 0) {
            int result2 = jdbcConnection.executeAlter("delete from
employeecommunication where fname = '"+textField_4.getText()+"' and lname =
 '"+textField_5.getText()+"'");
            if (result2 != 0)
                JOptionPane.showMessageDialog(null,"Employee was
Successfully Deleted");
            else
                JOptionPane.showMessageDialog(null,"Employee Not
Found!","Alert",JOptionPane.WARNING_MESSAGE);
        }else
            JOptionPane.showMessageDialog(null,"Employee Not
Found!","Alert",JOptionPane.WARNING_MESSAGE);
    }
});
btnNewButton_4.setBounds(472, 11, 99, 23);
panel_2.add(btnNewButton_4);
textField_4 = new JTextField();
textField_4.setBounds(167, 11, 124, 20);
panel_2.add(textField_4);
textField_4.setColumns(10);
textField_5 = new JTextField();
textField_5.setBounds(167, 36, 124, 20);
panel_2.add(textField_5);
textField_5.setColumns(10);
textField_6 = new JTextField();
textField_6.setBounds(167, 67, 124, 20);
panel_2.add(textField_6);
textField_6.setColumns(10);
JButton btnNewButton_10 = new JButton("View Table");
btnNewButton_10.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        loadTable(dm2, table_1, "SELECT * FROM EMPLOYEE");
    }
});
btnNewButton_10.setBounds(610, 67, 99, 23);
panel_2.add(btnNewButton_10);
JButton btnNewButton_11 = new JButton("Clear Table");

```



```

        btnNewButton_11.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dm2.setColumnCount(0);
                dm2.setRowCount(0);
            }
        });
        btnNewButton_11.setBounds(472, 67, 99, 23);
        panel_2.add(btnNewButton_11);
        JPanel panel_1 = new JPanel();
        panel_1.setBackground(new Color(233, 150, 122));
        tabbedPane.addTab("Search", null, panel_1, null);
        tabbedPane.setForegroundAt(3, new Color(128, 0, 0));
        panel_1.setLayout(null);
        JScrollPane scrollPane_3 = new JScrollPane();
        scrollPane_3.setBounds(10, 102, 699, 255);
        panel_1.add(scrollPane_3);
        table_3 = new JTable();
        table_3.setFont(new Font("Tahoma", Font.BOLD, 11));
        scrollPane_3.setViewportView(table_3);
        JLabel lblNewLabel_13 = new JLabel("First Name");
        lblNewLabel_13.setBounds(21, 22, 90, 14);
        panel_1.add(lblNewLabel_13);
        JLabel lblNewLabel_14 = new JLabel("Last Name");
        lblNewLabel_14.setBounds(21, 60, 90, 14);
        panel_1.add(lblNewLabel_14);
        JTextField_12 = new JTextField();
        textField_12.setBounds(137, 19, 122, 20);
        panel_1.add(textField_12);
        textField_12.setColumns(10);
        textField_13 = new JTextField();
        textField_13.setBounds(137, 57, 122, 20);
        panel_1.add(textField_13);
        textField_13.setColumns(10);
        JButton btnNewButton_12 = new JButton("Search Employee");
        btnNewButton_12.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                loadTable(dm3, table_3, "select * from employee where fname =
'" + textField_12.getText() + "' and lname = '" + textField_13.getText() + "'");
            }
        });
        btnNewButton_12.setBounds(413, 22, 147, 23);
        panel_1.add(btnNewButton_12);
        JButton btnNewButton_13 = new JButton("Search Volunteers");
        btnNewButton_13.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                loadTable(dm3, table_3, "select fname, lname, empid,
zoaddress from employee where pay = 0");
            }
        });
        btnNewButton_13.setBounds(413, 56, 147, 23);
        panel_1.add(btnNewButton_13);
        JButton btnNewButton_15 = new JButton("Clear All");
        btnNewButton_15.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                textField_12.setText("");
                textField_13.setText("");
            }
        });
        btnNewButton_15.setBounds(594, 22, 115, 23);
        panel_1.add(btnNewButton_15);
        JButton btnNewButton_14 = new JButton("Clear Table");
        btnNewButton_14.setBounds(594, 56, 115, 23);

```

```

        panel_1.add(btnNewButton_14);
        btnNewButton_14.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dm3.setColumnCount(0);
                dm3.setRowCount(0);
            }
        });
    }

    void loadTable(DefaultTableModel dm, JTable table, String queryString) {
        dm.setColumnCount(0);
        dm.setRowCount(0);
        ResultSet resultSet = jdbcConnection.executeQuery(queryString);
        ResultSetMetaData rsmd;
        try {
            rsmd = resultSet.getMetaData();
            int cols=rsmd.getColumnCount();
            String c[]=new String[cols];
            for(int i=0;i<cols;i++){
                c[i]=rsmd.getColumnName(i+1);
                dm.addColumn(c[i]);
            }
            Object row[]=new Object[cols];
            while(resultSet.next()){
                for(int i=0;i<cols;i++){
                    row[i]=resultSet.getString(i+1);
                }
                dm.addRow(row);
            }
            if(dm.getRowCount()==0)
                JOptionPane.showMessageDialog(null, "Not Found!");
            else
                table.setModel(dm);
        } catch (SQLException e1) {}
    }
}

```

### **Animal.java :**

```

public class Animal extends JFrame {

    private JPanel contentPane;
    private JTable table,table_1,table_2;
    private JTextField
textField,textField_1,textField_2,textField_3,textField_4,textField_5,textField_6,t
extField_7;
    private JTextField
textField_8,textField_9,textField_10,textField_23,textField_24,textField_25,textFie
ld_26;
    private JTextField
textField_27,textField_28,textField_29,textField_30,textField_22,textField_21,textF
ield_11;
    private DefaultTableModel dm, dm1,dm2;
    private DBConnection jdbcConnection;
    public Animal() {
        jdbcConnection=DBConnection.getInstance();
        dm = new DefaultTableModel();
        dm1 = new DefaultTableModel();
        dm2 = new DefaultTableModel();
        setForeground(new Color(255, 255, 255));

        setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\cheraka\\eclip
se-workspace\\java_eclipse\\images\\icons8-animal-shelter-20.png"));
    }
}

```

```

setTitle("Animal Depot - Animal Manager");
setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
addWindowListener((WindowListener) new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        if(JOptionPane.showConfirmDialog(null,"Are you sure? you
want to close Application!")==JOptionPane.YES_OPTION){
            jdbcConnection.disconnect();
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
    }
});
setBounds(100, 100, 786, 567);
JMenuBar menuBar = new JMenuBar();
menuBar.setBackground(new Color(255, 140, 0));
setJMenuBar(menuBar);
JMenu mnNewMenu = new JMenu("Main Page");
mnNewMenu.setForeground(new Color(0, 128, 0));
mnNewMenu.setBackground(new Color(0, 100, 0));
mnNewMenu.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        dispose();
        new MainPage().setVisible(true);
    }
});
menuBar.add(mnNewMenu);
JMenu mnNewMenu_1 = new JMenu("Logout");
mnNewMenu_1.setBackground(new Color(0, 128, 0));
mnNewMenu_1.setForeground(new Color(0, 128, 0));
mnNewMenu_1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Login.isLoggedIn=false;
        dispose();
        new MainPage().setVisible(true);
    }
});
menuBar.add(mnNewMenu_1);
contentPane = new JPanel();
contentPane.setBackground(new Color(123, 104, 238));
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);
JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
tabbedPane.setBounds(10, 11, 750, 485);
contentPane.add(tabbedPane);
JPanel panel = new JPanel();
panel.setBackground(new Color(0, 255, 255));
tabbedPane.addTab("Insert || Remove", null, panel, null);
tabbedPane.setForegroundAt(0, new Color(128, 0, 0));
tabbedPane.setBackgroundAt(0, new Color(255, 255, 255));
panel.setLayout(null);
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 254, 725, 192);
panel.add(scrollPane);
table = new JTable();
table.setFont(new Font("Tahoma", Font.BOLD, 11));
scrollPane.setViewportView(table);
JLabel lblNewLabel = new JLabel("Animal ID");
lblNewLabel.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
lblNewLabel.setBounds(10, 11, 124, 14);
panel.add(lblNewLabel);

```

```

        JLabel lblNewLabel_1 = new JLabel("Name");
        lblNewLabel_1.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        lblNewLabel_1.setBounds(10, 45, 124, 14);
        panel.add(lblNewLabel_1);
        JLabel lblNewLabel_2 = new JLabel("Age");
        lblNewLabel_2.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        lblNewLabel_2.setBounds(10, 70, 124, 14);
        panel.add(lblNewLabel_2);
        JLabel lblNewLabel_3 = new JLabel("Sex");
        lblNewLabel_3.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        lblNewLabel_3.setBounds(10, 101, 124, 14);
        panel.add(lblNewLabel_3);
        JLabel lblNewLabel_4 = new JLabel("Height");
        lblNewLabel_4.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        lblNewLabel_4.setBounds(10, 126, 124, 14);
        panel.add(lblNewLabel_4);
        JLabel lblNewLabel_5 = new JLabel("Weight");
        lblNewLabel_5.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        lblNewLabel_5.setBounds(372, 11, 124, 14);
        panel.add(lblNewLabel_5);
        JLabel lblNewLabel_6 = new JLabel("Species");
        lblNewLabel_6.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        lblNewLabel_6.setBounds(372, 45, 124, 14);
        panel.add(lblNewLabel_6);
        JLabel lblNewLabel_7 = new JLabel("Eating Frequency");
        lblNewLabel_7.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        lblNewLabel_7.setBounds(372, 76, 124, 14);
        panel.add(lblNewLabel_7);
        JLabel lblNewLabel_8 = new JLabel("Eating Amount");
        lblNewLabel_8.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        lblNewLabel_8.setBounds(372, 101, 124, 14);
        panel.add(lblNewLabel_8);
        JLabel lblNewLabel_9 = new JLabel("Animal ID");
        lblNewLabel_9.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        lblNewLabel_9.setBounds(10, 213, 124, 14);
        panel.add(lblNewLabel_9);
        JButton btnNewButton = new JButton("Delete");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                String id=textField_10.getText();
                try {
                    if (jdbcConnection.executeQuery("select * from animal where
animalid = "+id).next() == false)
                        JOptionPane.showMessageDialog(null, "Animal not found!",
"Alert",JOptionPane.WARNING_MESSAGE);
                    else{
                        jdbcConnection.executeQuery("delete from performs where
animalid = "+id);
                        jdbcConnection.executeQuery("delete from caresfor
where animalid = "+id);
                        jdbcConnection.executeQuery("delete from trains
where animalid = "+id);
                        jdbcConnection.executeQuery("delete from eats

```

```

where animalid = "+id);
                                jdbcConnection.executeQuery("delete from trades
where animalid = "+id);
                                jdbcConnection.executeQuery("delete from animal
where animalid = "+id);
                                JOptionPane.showMessageDialog(null, "Successfully
Deleted!");
                                }
                                } catch (SQLException e) {
                                    System.out.println("error: select query not valid");
                                }
                                }
                            });
                            btnNewButton.setForeground(new Color(128, 0, 0));
                            btnNewButton.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
                            btnNewButton.setBounds(368, 209, 89, 23);
                            panel.add(btnNewButton);
                            JButton btnNewButton_2 = new JButton("Clear All");
                            btnNewButton_2.setForeground(new Color(128, 0, 0));
                            btnNewButton_2.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
                            btnNewButton_2.addActionListener(new ActionListener() {
                                public void actionPerformed(ActionEvent e) {

                                    textField.setText("");textField_1.setText("");textField_2.setText("");textFi
eld_3.setText("");

                                    textField_4.setText("");textField_5.setText("");textField_6.setText("");text
Field_7.setText("");
                                    textField_8.setText("");textField_9.setText("");
                                }
                            });
                            btnNewButton_2.setBounds(10, 162, 89, 23);
                            panel.add(btnNewButton_2);
                            JButton btnNewButton_3 = new JButton("Submit");
                            btnNewButton_3.setForeground(new Color(0, 128, 0));
                            btnNewButton_3.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
                            btnNewButton_3.addActionListener(new ActionListener() {
                                public void actionPerformed(ActionEvent arg0) {
                                    try {
                                        String result="insert into animal values
("+textField.getText()+" , '"+textField_1.getText()+"', '"+textField_2.getText();
                                        result+=",
 '"+textField_3.getText()+"', '"+textField_4.getText()+"', '"+textField_5.getText()+"',
 '"+textField_6.getText();

                                        result+=", '"+textField_7.getText()+"', '"+textField_8.getText()+"', '"+textField_9
.getText()+"");

                                        if(jdbcConnection.executeAlter(result)!=0)
                                        {

                                            JOptionPane.showMessageDialog(null,"Successfully Inserted!");
                                        }else {
                                            JOptionPane.showMessageDialog(null,"Invalid
Details");
                                        }
                                    }catch (Exception e1) {}
                                }
                            });
                            btnNewButton_3.setBounds(179, 162, 89, 23);
                            panel.add(btnNewButton_3);

```

```

textField = new JTextField();
textField.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField.setBounds(179, 8, 145, 20);
panel.add(textField);
textField.setColumns(10);
textField_1 = new JTextField();
textField_1.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_1.setBounds(179, 36, 145, 20);
panel.add(textField_1);
textField_1.setColumns(10);
textField_2 = new JTextField();
textField_2.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_2.setBounds(179, 67, 145, 20);
panel.add(textField_2);
textField_2.setColumns(10);
textField_3 = new JTextField();
textField_3.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_3.setBounds(179, 98, 145, 20);
panel.add(textField_3);
textField_3.setColumns(10);
textField_4 = new JTextField();
textField_4.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_4.setBounds(176, 123, 148, 20);
panel.add(textField_4);
textField_4.setColumns(10);
textField_5 = new JTextField();
textField_5.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_5.setBounds(587, 8, 148, 20);
panel.add(textField_5);
textField_5.setColumns(10);
textField_6 = new JTextField();
textField_6.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_6.setBounds(587, 42, 148, 20);
panel.add(textField_6);
textField_6.setColumns(10);
textField_7 = new JTextField();
textField_7.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_7.setBounds(587, 67, 148, 20);
panel.add(textField_7);
textField_7.setColumns(10);
textField_8 = new JTextField();
textField_8.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_8.setBounds(587, 98, 148, 20);
panel.add(textField_8);
textField_8.setColumns(10);
textField_9 = new JTextField();
textField_9.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_9.setBounds(587, 123, 148, 20);
panel.add(textField_9);
textField_9.setColumns(10);
JLabel lblNewLabel_10 = new JLabel("Habitat ID");
lblNewLabel_10.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
lblNewLabel_10.setBounds(372, 126, 124, 14);
panel.add(lblNewLabel_10);
textField_10 = new JTextField();
textField_10.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
textField_10.setBounds(179, 210, 148, 20);
panel.add(textField_10);
textField_10.setColumns(10);
JSeparator separator = new JSeparator();
separator.setBackground(new Color(85, 107, 47));

```

```

separator.setBounds(10, 196, 466, 2);
panel.add(separator);
JSeparator separator_1 = new JSeparator();
separator_1.setBackground(new Color(85, 107, 47));
separator_1.setBounds(10, 238, 466, 2);
panel.add(separator_1);
JButton btnNewButton_6 = new JButton("View Data");
btnNewButton_6.setForeground(new Color(0, 128, 0));
btnNewButton_6.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
btnNewButton_6.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        loadTable(dm, table,"select * from animal");
    }
});
btnNewButton_6.setBounds(495, 204, 115, 33);
panel.add(btnNewButton_6);
JButton btnNewButton_8 = new JButton("Clear Table");
btnNewButton_8.setForeground(new Color(128, 0, 0));
btnNewButton_8.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
btnNewButton_8.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dm.setColumnCount(0);
        dm.setRowCount(0);
    }
});
btnNewButton_8.setBounds(620, 204, 115, 33);
panel.add(btnNewButton_8);
JPanel panel_1 = new JPanel();
panel_1.setBackground(new Color(0, 250, 154));
panel_1.setLayout(null);
tabbedPane.addTab("Update", null, panel_1, null);
tabbedPane.setForegroundAt(1, new Color(128, 0, 0));
JScrollPane scrollPane_1 = new JScrollPane();
scrollPane_1.setBounds(10, 252, 725, 194);
panel_1.add(scrollPane_1);
table_2 = new JTable();
table_2.setFont(new Font("Tahoma", Font.BOLD, 11));
scrollPane_1.setViewportView(table_2);
JLabel label_11 = new JLabel("Animal ID");
label_11.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
label_11.setBounds(20, 17, 124, 14);
panel_1.add(label_11);
JLabel label_12 = new JLabel("Name");
label_12.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
label_12.setBounds(20, 53, 124, 14);
panel_1.add(label_12);
JLabel label_13 = new JLabel("Age");
label_13.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
label_13.setBounds(20, 91, 124, 14);
panel_1.add(label_13);
JLabel label_14 = new JLabel("Sex");
label_14.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
label_14.setBounds(20, 126, 124, 14);
panel_1.add(label_14);
JLabel label_15 = new JLabel("Height");
label_15.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
label_15.setBounds(20, 158, 124, 14);
panel_1.add(label_15);
JLabel label_16 = new JLabel("Weight");
label_16.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));

```

```

label_16.setBounds(377, 17, 124, 14);
panel_1.add(label_16);
JLabel label_17 = new JLabel("Species");
label_17.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
label_17.setBounds(377, 53, 124, 14);
panel_1.add(label_17);
JLabel label_18 = new JLabel("Eating Frequency");
label_18.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
label_18.setBounds(377, 91, 124, 14);
panel_1.add(label_18);
JLabel label_19 = new JLabel("Eating Amount");
label_19.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
label_19.setBounds(377, 126, 124, 14);
panel_1.add(label_19);
JButton btnClearAll = new JButton("Clear All");
btnClearAll.setForeground(new Color(128, 0, 0));
btnClearAll.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
btnClearAll.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField_21.setText("");
        textField_22.setText("");
        textField_23.setText("");
        textField_24.setText("");
        textField_25.setText("");
        textField_26.setText("");
        textField_27.setText("");
        textField_28.setText("");
        textField_29.setText("");
        textField_30.setText("");
    }
});
btnClearAll.setBounds(20, 207, 89, 23);
panel_1.add(btnClearAll);
JButton btnModify = new JButton("Modify");
btnModify.setForeground(new Color(0, 128, 0));
btnModify.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
btnModify.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            String reString="update animal set name =
            '"+textField_22.getText()+"', age = '"+textField_23.getText();
            reString+=" sex = '"+textField_24.getText()+"',
height = '"+textField_25.getText()+" weight = '"+textField_26.getText();
            reString+=" species =
            '"+textField_27.getText()+" eat_freq_week = '"+textField_28.getText()+"
eat_amount = ";
            reString+=textField_29.getText()+" enclosure_id =
            '"+textField_30.getText()+" where animalid = '"+textField_21.getText();
            if(jdbcTemplate.executeAlter(reString)!=0)
                JOptionPane.showMessageDialog(null,"Successfully
Updated!");
            else {
                JOptionPane.showMessageDialog(null,"Invalid
Details","Alert", JOptionPane.WARNING_MESSAGE);
            }
        }catch (Exception e1) {
        }
    }
});
btnModify.setBounds(154, 207, 89, 23);
panel_1.add(btnModify);
textField_23 = new JTextField();

```



```

        textField_23.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_23.setColumns(10);
        textField_23.setBounds(154, 85, 147, 20);
        panel_1.add(textField_23);
        textField_24 = new JTextField();
        textField_24.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_24.setColumns(10);
        textField_24.setBounds(154, 123, 147, 20);
        panel_1.add(textField_24);
        textField_25 = new JTextField();
        textField_25.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_25.setColumns(10);
        textField_25.setBounds(154, 152, 147, 20);
        panel_1.add(textField_25);
        textField_26 = new JTextField();
        textField_26.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_26.setColumns(10);
        textField_26.setBounds(572, 14, 147, 20);
        panel_1.add(textField_26);
        textField_27 = new JTextField();
        textField_27.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_27.setColumns(10);
        textField_27.setBounds(572, 47, 147, 20);
        panel_1.add(textField_27);
        textField_28 = new JTextField();
        textField_28.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_28.setColumns(10);
        textField_28.setBounds(572, 88, 147, 20);
        panel_1.add(textField_28);
        textField_29 = new JTextField();
        textField_29.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_29.setColumns(10);
        textField_29.setBounds(572, 123, 147, 20);
        panel_1.add(textField_29);
        JLabel label_21 = new JLabel("Habitat ID");
        label_21.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        label_21.setBounds(377, 158, 124, 14);
        panel_1.add(label_21);
        textField_30 = new JTextField();
        textField_30.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_30.setColumns(10);
        textField_30.setBounds(572, 155, 147, 20);
        panel_1.add(textField_30);
        textField_22 = new JTextField();
        textField_22.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_22.setBounds(154, 50, 147, 23);
        panel_1.add(textField_22);
        textField_22.setColumns(10);
        textField_21 = new JTextField();
        textField_21.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        textField_21.setBounds(154, 14, 147, 23);
        panel_1.add(textField_21);
        textField_21.setColumns(10);
        JButton btnNewButton_7 = new JButton("View Data");
        btnNewButton_7.setForeground(new Color(0, 128, 0));
        btnNewButton_7.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        btnNewButton_7.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                loadTable(dm1,table_2,"select * from animal");
            }
        });
    });

```

```

        btnNewButton_7.setBounds(494, 207, 111, 23);
        panel_1.add(btnNewButton_7);
        JButton btnNewButton_9 = new JButton("Clear Table");
        btnNewButton_9.setForeground(new Color(128, 0, 0));
        btnNewButton_9.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        btnNewButton_9.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dm1.setColumnCount(0);
                dm1.setRowCount(0);
            }
        });
        btnNewButton_9.setBounds(615, 207, 104, 23);
        panel_1.add(btnNewButton_9);
        JPanel panel_2 = new JPanel();
        panel_2.setBackground(new Color(221, 160, 221));
        panel_2.setForeground(new Color(238, 130, 238));
        tabbedPane.addTab("View", null, panel_2, null);
        tabbedPane.setForegroundAt(2, new Color(128, 0, 0));
        panel_2.setLayout(null);
        JScrollPane scrollPane_2 = new JScrollPane();
        scrollPane_2.setBounds(62, 111, 626, 298);
        panel_2.add(scrollPane_2);
        table_1 = new JTable();
        table_1.setFont(new Font("Tahoma", Font.BOLD, 11));
        scrollPane_2.setViewportView(table_1);
        JLabel lblNewLabel_11 = new JLabel("Species");
        lblNewLabel_11.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));
        lblNewLabel_11.setBounds(69, 43, 94, 23);
        panel_2.add(lblNewLabel_11);
        textField_11 = new JTextField();
        textField_11.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 12));
        textField_11.setBounds(173, 43, 167, 23);
        panel_2.add(textField_11);
        textField_11.setColumns(10);
        JButton btnNewButton_4 = new JButton("Search by Species");
        btnNewButton_4.setForeground(new Color(0, 128, 0));
        btnNewButton_4.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));
        btnNewButton_4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                loadTable(dm2, table_1, "select animalid, name, weight,
height from animal where species = '"+textField_11.getText()+"'");
            }
        });
        btnNewButton_4.setBounds(512, 27, 167, 23);
        panel_2.add(btnNewButton_4);
        JButton btnNewButton_5 = new JButton("Species Details");
        btnNewButton_5.setForeground(new Color(0, 128, 0));
        btnNewButton_5.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));
        btnNewButton_5.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                loadTable(dm2, table_1, "select animalid, name, age, sex,
species from animal order by species");
            }
        });
        btnNewButton_5.setBounds(512, 77, 167, 23);
        panel_2.add(btnNewButton_5);
    }
    void loadTable(DefaultTableModel dm, JTable table, String queryString) {

```

```

        dm.setColumnCount(0);
        dm.setRowCount(0);
        ResultSet resultSet = jdbcConnection.executeQuery(queryString);
        ResultSetMetaData rsmd;
        try {
            rsmd = resultSet.getMetaData();
            int cols=rsmd.getColumnCount();
            String c[]=new String[cols];
            for(int i=0;i<cols;i++){
                c[i]=rsmd.getColumnName(i+1);
                dm.addColumn(c[i]);
            }
            Object row[]=new Object[cols];
            while(resultSet.next()){
                for(int i=0;i<cols;i++){
                    row[i]=resultSet.getString(i+1);
                }
                dm.addRow(row);
            }
            if(dm.getRowCount()==0)
            {
                JOptionPane.showMessageDialog(null,"Not Found!", "Alert",JOptionPane.WARNING_MESSAGE);
            }else {
                table.setModel(dm);
            }
        } catch (SQLException e1) {}
    }
}

```

**Food.java :**

```

public class Food extends JFrame {

    private JPanel contentPane;
    private JTable table;
    private DefaultTableModel dm;
    private DBConnection jdbcConnection;

    public Food() {
        jdbcConnection = DBConnection.getInstance();

        setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\cheraka\\eclipse-workspace\\java_eclipse\\images\\icons8-animal-shelter-20.png"));
        setTitle("Animal Depot");
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        addWindowListener((WindowListener) new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                if(JOptionPane.showConfirmDialog(null,"Are you sure? you want to close Application!")==JOptionPane.YES_OPTION){
                    jdbcConnection.disconnect();
                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                }
            }
        });
        setBounds(100, 100, 773, 495);
        JMenuBar menuBar = new JMenuBar();
        menuBar.setBackground(new Color(255, 165, 0));
        setJMenuBar(menuBar);
        JMenu mnNewMenu = new JMenu("Main Page");
        mnNewMenu.setForeground(new Color(0, 128, 0));
        mnNewMenu.addMouseListener(new MouseAdapter() {

```

```

        @Override
        public void mouseClicked(MouseEvent e) {
            dispose();
            new MainPage().setVisible(true);
        }
    });
    menuBar.add(mnNewMenu);
    JMenu mnNewMenu_1 = new JMenu("Logout");
    mnNewMenu_1.setForeground(new Color(0, 128, 0));
    mnNewMenu_1.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            Login.isLoggedIn=false;
            dispose();
            new MainPage().setVisible(true);
        }
    });
    menuBar.add(mnNewMenu_1);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(0, 191, 255));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    JPanel panel = new JPanel();
    panel.setBackground(new Color(123, 104, 238));
    panel.setBorder(new MatteBorder(3, 3, 3, 3, (Color) new Color(0, 0,
0)));
    panel.setBounds(10, 11, 737, 116);
    contentPane.add(panel);
    panel.setLayout(null);
    JButton btnNewButton_2 = new JButton("Clear");
    btnNewButton_2.setForeground(new Color(128, 0, 0));
    btnNewButton_2.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));
    btnNewButton_2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dm.setColumnCount(0);
            dm.setRowCount(0);
        }
    });
    btnNewButton_2.setBounds(577, 63, 114, 32);
    panel.add(btnNewButton_2);
    dm = new DefaultTableModel();
    JButton btnNewButton_3 = new JButton("Expiring Food");
    btnNewButton_3.setForeground(new Color(0, 128, 0));
    btnNewButton_3.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));
    btnNewButton_3.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            try {
                dm.setColumnCount(0);
                dm.setRowCount(0);
                StringBuilder str = new StringBuilder();
                str.append("select foodid, name, date_expired from food where
1=1 order by date_expired");
                ResultSet resultSet =
jDbConnection.executeQuery(str.toString());
                ResultSetMetaData rsmd=resultSet.getMetaData();
                //Coding to get columns-
                int cols=rsmd.getColumnCount();
                String c[]=new String[cols];
                for(int i=0;i<cols;i++){

```

```

        c[i]=rsmd.getColumnName(i+1);
        dm.addColumn(c[i]);
    }
    //get data from rows
    Object row[]=new Object[cols];
    while(resultSet.next()){
        for(int i=0;i<cols;i++){
            row[i]=resultSet.getString(i+1);
            dm.addRow(row);
        }
    }
    if(dm.getRowCount()==0)
        JOptionPane.showMessageDialog(null,"Not
Found!","Alert",JOptionPane.WARNING_MESSAGE);
    else
        table.setModel(dm);
    }catch (Exception e1) {}
    }
});
btnNewButton_3.setBounds(328, 63, 193, 32);
panel.add(btnNewButton_3);
JLabel lblNewLabel = new JLabel("Food Details");
lblNewLabel.setForeground(new Color(128, 0, 0));
lblNewLabel.setBackground(new Color(128, 0, 0));
lblNewLabel.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 22));
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setBounds(172, 11, 461, 41);
panel.add(lblNewLabel);
JButton btnNewButton_4 = new JButton("Get Food");
btnNewButton_4.setForeground(new Color(0, 128, 0));
btnNewButton_4.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));
btnNewButton_4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            dm.setColumnCount(0);
            dm.setRowCount(0);
            StringBuilder str = new StringBuilder();
            str.append("select s.siteid, s.location, f.foodid, f.name,
f.stock_serving, f.date_purchased, f.date_expired ");
            str.append("from site s, food f where s.siteid = f.siteid");
            ResultSet resultSet =
jdbcConnection.executeQuery(str.toString());
            ResultSetMetaData rsmd=resultSet.getMetaData();
            //Coding to get columns-
            int cols=rsmd.getColumnCount();
            String c[]=new String[cols];
            for(int i=0;i<cols;i++){
                c[i]=rsmd.getColumnName(i+1);
                dm.addColumn(c[i]);
            }
            //get data from rows
            Object row[]=new Object[cols];
            while(resultSet.next()){
                for(int i=0;i<cols;i++){
                    row[i]=resultSet.getString(i+1);
                    dm.addRow(row);
                }
            }
            if(dm.getRowCount()==0)
                JOptionPane.showMessageDialog(null,"Not
Found!","Alert",JOptionPane.WARNING_MESSAGE);
            else
                table.setModel(dm);
        }
    }
});

```

```

        }catch (Exception e1) {}
    }
});
btnNewButton_4.setBounds(36, 63, 207, 32);
panel.add(btnNewButton_4);
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 138, 737, 286);
contentPane.add(scrollPane);
table = new JTable();
table.setFont(new Font("Tahoma", Font.BOLD, 11));
scrollPane.setViewportView(table);
}
}

```

### **Zoo.java :**

```

public class Zoo extends JFrame {

    private JPanel contentPane;
    private JTable table;
    private JTextField textField;
    private DBConnection jDbConnection;
    private DefaultTableModel dm;
    public Zoo() {
        jDbConnection=DBConnection.getInstance();
        setTitle("Animal Depot");

        setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\\\Users\\\\cheraka\\\\eclipse-workspace\\\\java_eclipse\\\\images\\\\icons8-animal-shelter-20.png"));
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        addWindowListener((WindowListener) new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                if(JOptionPane.showConfirmDialog(null,"Are you sure? you want to close Application!")==JOptionPane.YES_OPTION){
                    jDbConnection.disconnect();
                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                }
            }
        });
        setBounds(100, 100, 773, 495);
        JMenuBar menuBar = new JMenuBar();
        menuBar.setBackground(new Color(255, 140, 0));
        menuBar.setForeground(new Color(255, 140, 0));
        setJMenuBar(menuBar);
        JMenu mnNewMenu = new JMenu("Main Page");
        mnNewMenu.setForeground(new Color(0, 128, 0));
        mnNewMenu.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                dispose();
                new MainPage().setVisible(true);
            }
        });
        menuBar.add(mnNewMenu);
        JMenu mnNewMenu_1 = new JMenu("Logout");
        mnNewMenu_1.setForeground(new Color(0, 128, 0));
        mnNewMenu_1.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                Login.isLoggedIn=false;
                dispose();
                new MainPage().setVisible(true);
            }
        });
    }
}

```

```

    }
    });
    menuBar.add(mnNewMenu_1);
    contentPane = new JPanel();
    contentPane.setBackground(new Color(0, 191, 255));
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    JPanel panel = new JPanel();
    panel.setBackground(new Color(147, 112, 219));
    panel.setBorder(new MatteBorder(3, 3, 3, 3, (Color) new Color(0, 0,
0)));
    panel.setBounds(10, 11, 284, 413);
    contentPane.add(panel);
    panel.setLayout(null);
    JLabel lblNewLabel = new JLabel("Zoo Details");
    lblNewLabel.setForeground(new Color(128, 0, 0));
    lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 24));
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setBounds(10, 35, 235, 49);
    panel.add(lblNewLabel);
    JLabel lblNewLabel_1 = new JLabel("Country");
    lblNewLabel_1.setForeground(new Color(255, 255, 255));
    lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel_1.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
14));
    lblNewLabel_1.setBounds(10, 111, 111, 28);
    panel.add(lblNewLabel_1);
    JTextField textField = new JTextField();
    textField.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 14));
    textField.setBounds(131, 112, 129, 28);
    panel.add(textField);
    textField.setColumns(10);
    JButton btnNewButton = new JButton("Clear");
    btnNewButton.setForeground(new Color(128, 0, 0));
    btnNewButton.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 12));
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dm.setRowCount(0);
            dm.setColumnCount(0);
            textField.setText("");
        }
    });
    btnNewButton.setBounds(10, 336, 111, 39);
    panel.add(btnNewButton);
    dm=new DefaultTableModel();
    JButton btnNewButton_1 = new JButton("View");
    btnNewButton_1.setForeground(new Color(0, 128, 0));
    btnNewButton_1.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));
    btnNewButton_1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            try {
                dm.setColumnCount(0);
                dm.setRowCount(0);
                String queryString="select * from zoo where country =
"+"+textField.getText()+"'";
                jdbcConnection=DBConnection.getInstance();
                ResultSet resultSet=
jdbcConnection.executeQuery(queryString);

                ResultSetMetaData rsmd=resultSet.getMetaData());

```

```

        //Coding to get columns-
        int cols=rsmd.getColumnCount();
        String c[]=new String[cols];
        for(int i=0;i<cols;i++){
            c[i]=rsmd.getColumnName(i+1);
            dm.addColumn(c[i]);
        }
        //get data from rows
        Object row[]=new Object[cols];
        while(resultSet.next()){
            for(int i=0;i<cols;i++){
                row[i]=resultSet.getString(i+1);
            }
            dm.addRow(row);
        }
        if(dm.getRowCount()==0)
            JOptionPane.showMessageDialog(null,"Not
Found!", "Alert", JOptionPane.WARNING_MESSAGE);
        else
            table.setModel(dm);
        }catch (Exception e1) {}
    }
});
btnNewButton_1.setBounds(149, 336, 111, 39);
panel.add(btnNewButton_1);
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(304, 11, 443, 413);
contentPane.add(scrollPane);
table = new JTable();
table.setFont(new Font("Tahoma", Font.BOLD, 11));
scrollPane.setViewportView(table);
}
}

```

**Show.java :**

```

public class Show extends JFrame {

    private JPanel contentPane;
    private DBConnection jdbcConnection;
    private DefaultTableModel dm;
    private JTable table;
    public Show() {
        dm=new DefaultTableModel();

        setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\cheraka\\eclipse-workspace\\java_eclipse\\images\\icons8-animal-shelter-20.png"));
        setTitle("Animal Depot");
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        addWindowListener((WindowListener) new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                if(JOptionPane.showConfirmDialog(null,"Are you sure? you
want to close Application!")==JOptionPane.YES_OPTION){
                    jdbcConnection.disconnect();
                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                }
            }
        });
        setBounds(100, 100, 773, 495);
        JMenuBar menuBar = new JMenuBar();
        menuBar.setBackground(new Color(255, 165, 0));
        setJMenuBar(menuBar);
        JMenu mnNewMenu = new JMenu("MainPage");
    }
}

```



```

mnNewMenu.setForeground(new Color(0, 128, 0));
mnNewMenu.setBackground(new Color(255, 255, 255));
mnNewMenu.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        dispose();
        new MainPage().setVisible(true);
    }
});
menuBar.add(mnNewMenu);
JMenu mnNewMenu_1 = new JMenu("Logout");
mnNewMenu_1.setForeground(new Color(0, 128, 0));
mnNewMenu_1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Login.isLoggedIn=false;
        dispose();
        new MainPage().setVisible(true);
    }
});
menuBar.add(mnNewMenu_1);
contentPane = new JPanel();
contentPane.setBackground(new Color(0, 206, 209));
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);
JButton btnNewButton = new JButton("Clear");
btnNewButton.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
btnNewButton.setForeground(new Color(128, 0, 0));
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dm.setColumnCount(0);
        dm.setRowCount(0);
    }
});
btnNewButton.setBounds(579, 12, 125, 23);
contentPane.add(btnNewButton);
JButton btnNewButton_1 = new JButton("View");
btnNewButton_1.setForeground(new Color(0, 128, 0));
btnNewButton_1.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            jdbcConnection=DBConnection.getInstance();
            jdbcConnection.executeAlter("create view showinfo
as select sh.start_time, sh.duration, sh.name, sh.type, st.location " +
            "from show sh, site st where sh.siteid =
st.siteid");
            ResultSet resultSet= jdbcConnection.executeQuery("select
* from showInfo");
            dm=new DefaultTableModel();
            ResultSetMetaData rsmd=resultSet.getMetaData();
            //Coding to get columns-
            int cols=rsmd.getColumnCount();
            String c[]=new String[cols];
            for(int i=0;i<cols;i++){
                c[i]=rsmd.getColumnName(i+1);
                dm.addColumn(c[i]);
            }
            Object row[]=new Object[cols];
            while(resultSet.next()){

```

```

        for(int i=0;i<cols;i++)
            row[i]=resultSet.getString(i+1);
        dm.addRow(row);
    }
    if(dm.getRowCount()==0)
        JOptionPane.showMessageDialog(null,"Not
Found!","Alert",JOptionPane.WARNING_MESSAGE);
    else
        table.setModel(dm);
    }catch (Exception e1) {}
    }
});
btnNewButton_1.setBounds(61, 12, 125, 23);
contentPane.add(btnNewButton_1);
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBorder(new MatteBorder(3, 3, 3, 3, (Color) new
Color(128, 0, 0)));
scrollPane.setBounds(61, 59, 643, 331);
contentPane.add(scrollPane);
table = new JTable();
table.setFont(new Font("Tahoma", Font.BOLD, 11));
scrollPane.setViewportView(table);
}
}

```

### **Location.java :**

```

public class Location extends JFrame {

    private JPanel contentPane;
    private JTable table;
    private JTextField
textField,textField_1,textField_2,textField_3,textField_4;
    private DBConnection jdbc;
    protected DefaultTableModel dm;
    public Location() {
        jdbc=DBConnection.getInstance();
        dm=new DefaultTableModel();

        setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\cheraka\\eclip
se-workspace\\java_eclipse\\images\\icons8-animal-shelter-20.png"));
        setTitle("Animal Depot");
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        addWindowListener((WindowListener) new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                if(JOptionPane.showConfirmDialog(null,"Are you sure? you
want to close Application!")==JOptionPane.YES_OPTION){
                    jdbc.disconnect();
                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                }
            }
        });
        setBounds(100, 100, 996, 495);
        JMenuBar menuBar = new JMenuBar();
        menuBar.setBackground(new Color(255, 140, 0));
        setJMenuBar(menuBar);
        JMenu mnNewMenu = new JMenu("Main Page");
        mnNewMenu.setForeground(new Color(0, 128, 0));
        mnNewMenu.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent arg0) {
                dispose();
            }
        });
    }
}

```

```

        new MainPage().setVisible(true);
    }
});
menuBar.add(mnNewMenu);
JMenu mnNewMenu_1 = new JMenu("Logout");
mnNewMenu_1.setForeground(new Color(0, 128, 0));
mnNewMenu_1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        Login.isLoggedIn=false;
        dispose();
        new MainPage().setVisible(true);
    }
});
menuBar.add(mnNewMenu_1);
contentPane = new JPanel();
contentPane.setBackground(SystemColor.textHighlight);
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);
JPanel panel = new JPanel();
panel.setBackground(new Color(123, 104, 238));
panel.setBorder(new MatteBorder(3, 3, 3, 3, (Color) new Color(0, 0,
0)));

panel.setBounds(10, 11, 350, 413);
contentPane.add(panel);
panel.setLayout(null);
JLabel lblNewLabel = new JLabel("Enclosure ID");
lblNewLabel.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 12));
lblNewLabel.setForeground(new Color(255, 250, 250));
lblNewLabel.setBounds(40, 83, 94, 14);
panel.add(lblNewLabel);
JLabel lblNewLabel_1 = new JLabel("Square Feet");
lblNewLabel_1.setForeground(new Color(255, 250, 250));
lblNewLabel_1.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));

lblNewLabel_1.setBounds(40, 203, 94, 14);
panel.add(lblNewLabel_1);
JLabel lblNewLabel_2 = new JLabel("Temperature");
lblNewLabel_2.setForeground(new Color(255, 250, 250));
lblNewLabel_2.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));

lblNewLabel_2.setBounds(40, 152, 94, 14);
panel.add(lblNewLabel_2);
JLabel lblNewLabel_3 = new JLabel("Depth");
lblNewLabel_3.setForeground(new Color(255, 250, 250));
lblNewLabel_3.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));

lblNewLabel_3.setBounds(40, 267, 94, 14);
panel.add(lblNewLabel_3);
JLabel lblNewLabel_4 = new JLabel("Biome");
lblNewLabel_4.setForeground(new Color(255, 250, 250));
lblNewLabel_4.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
12));

lblNewLabel_4.setBounds(40, 326, 94, 14);
panel.add(lblNewLabel_4);
textField = new JTextField();
textField.setFont(new Font("Microsoft YaHei UI Light", Font.BOLD,
11));

textField.setBounds(172, 80, 128, 20);
panel.add(textField);
textField.setColumns(10);

```

```

        textField_1 = new JTextField();
        textField_1.setFont(new Font("Microsoft YaHei UI Light", Font.BOLD,
11));
        textField_1.setBounds(172, 200, 128, 20);
        panel.add(textField_1);
        textField_1.setColumns(10);
        textField_2 = new JTextField();
        textField_2.setFont(new Font("Microsoft YaHei UI Light", Font.BOLD,
11));
        textField_2.setBounds(172, 149, 128, 20);
        panel.add(textField_2);
        textField_2.setColumns(10);
        textField_3 = new JTextField();
        textField_3.setFont(new Font("Microsoft YaHei UI Light", Font.BOLD,
11));
        textField_3.setBounds(172, 264, 128, 20);
        panel.add(textField_3);
        textField_3.setColumns(10);
        textField_4 = new JTextField();
        textField_4.setFont(new Font("Microsoft YaHei UI Light", Font.BOLD,
11));
        textField_4.setBounds(172, 323, 128, 20);
        panel.add(textField_4);
        textField_4.setColumns(10);
        JLabel lblNewLabel_5 = new JLabel("Location Details");
        lblNewLabel_5.setForeground(new Color(128, 0, 0));
        lblNewLabel_5.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
22));
        lblNewLabel_5.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel_5.setBounds(40, 11, 260, 46);
        panel.add(lblNewLabel_5);
        JButton btnNewButton = new JButton("Clear All");
        btnNewButton.setFont(new Font("Microsoft Sans Serif", Font.BOLD, 11));
        btnNewButton.setForeground(new Color(128, 0, 0));
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                textField.setText("");
                textField_1.setText("");
                textField_2.setText("");
                textField_3.setText("");
                textField_4.setText("");
            }
        });
        btnNewButton.setBounds(40, 379, 89, 23);
        panel.add(btnNewButton);
        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(380, 98, 590, 326);
        contentPane.add(scrollPane);
        table = new JTable();
        table.setForeground(Color.BLACK);
        table.setBackground(new Color(255, 255, 255));
        table.setFont(new Font("Tahoma", Font.BOLD, 11));
        scrollPane.setViewportView(table);
        JButton btnNewButton_2 = new JButton("View By ID");
        btnNewButton_2.setForeground(new Color(0, 128, 0));
        btnNewButton_2.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
        btnNewButton_2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    dm.setColumnCount(0);
                    dm.setRowCount(0);

```

```

        String queryString="select animalid, name, species
from animal where enclosure_id = "+textField.getText();
        ResultSet resultSet=jdbc.executeQuery(queryString);
        ResultSetMetaData rsmd=resultSet.getMetaData();
        int cols=rsmd.getColumnCount();
        String c[]=new String[cols];
        for(int i=0;i<cols;i++){
            c[i]=rsmd.getColumnName(i+1);
            dm.addColumn(c);
        }
        Object row[]=new Object[cols];
        while(resultSet.next()){
            for(int i=0;i<cols;i++){
                row[i]=resultSet.getString(i+1);
                dm.addRow(row);
            }
        }
        if(dm.getRowCount()==0)
            JOptionPane.showMessageDialog(null,"Not
Found!","Alert",JOptionPane.WARNING_MESSAGE);
        else
            table.setModel(dm);
    }catch (Exception e1) {
    }
    }
});
btnNewButton_2.setBounds(380, 11, 136, 23);
contentPane.add(btnNewButton_2);

JButton btnNewButton_4 = new JButton("View By Temp");
btnNewButton_4.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
btnNewButton_4.setForeground(new Color(0, 128, 0));
btnNewButton_4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            dm.setColumnCount(0);
            dm.setRowCount(0);
            String queryString="select
enclosure_id,h.biome,s.siteid,location,usedfor,humidity,zooaddress from habitat h,
site s, habitatbiome hb " +
            "where h.siteid = s.siteid and h.biome =
hb.biome and temp = "+textField_2.getText();
            ResultSet resultSet=jdbc.executeQuery(queryString);
            ResultSetMetaData rsmd=resultSet.getMetaData();
            int cols=rsmd.getColumnCount();
            String c[]=new String[cols];
            for(int i=0;i<cols;i++){
                c[i]=rsmd.getColumnName(i+1);
                dm.addColumn(c[i]);
            }
            Object row[]=new Object[cols];
            while(resultSet.next()){
                for(int i=0;i<cols;i++){
                    row[i]=resultSet.getString(i+1);
                    dm.addRow(row);
                }
            }
            if(dm.getRowCount()==0)
                JOptionPane.showMessageDialog(null,"Not
Found!","Alert",JOptionPane.WARNING_MESSAGE);
            else
                table.setModel(dm);

```

```

        }catch (Exception e1) {}
    }
});
btnNewButton_4.setBounds(380, 64, 136, 23);
contentPane.add(btnNewButton_4);

JButton btnNewButton_3 = new JButton("Suitable Habitat");
btnNewButton_3.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
btnNewButton_3.setForeground(new Color(0, 128, 0));
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            dm.setColumnCount(0);
            dm.setRowCount(0);
            String queryString="select * from habitat where
sqft = "+textField_1.getText()+" and depth = "+textField_3.getText()+" and biome =
'"+textField_4.getText()+"'";
            ResultSet resultSet=jdbc.executeQuery(queryString);
            ResultSetMetaData rsmd=resultSet.getMetaData();
            int cols=rsmd.getColumnCount();
            String c[]=new String[cols];
            for(int i=0;i<cols;i++){
                c[i]=rsmd.getColumnName(i+1);
                dm.addColumn(c[i]);
            }
            Object row[]=new Object[cols];
            while(resultSet.next()){
                for(int i=0;i<cols;i++)
                    row[i]=resultSet.getString(i+1);
                dm.addRow(row);
            }
            if(dm.getRowCount()==0)
                JOptionPane.showMessageDialog(null,"Not
Found!", "Alert",JOptionPane.WARNING_MESSAGE);
            else
                table.setModel(dm);
        }catch (Exception e1) {}
    }
});
btnNewButton_3.setBounds(619, 11, 128, 23);
contentPane.add(btnNewButton_3);
JButton btnNewButton_5 = new JButton("Not in use");
btnNewButton_5.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));
btnNewButton_5.setForeground(new Color(0, 128, 0));
btnNewButton_5.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            dm.setColumnCount(0);
            dm.setRowCount(0);
            ResultSet resultSet=jdbc.executeQuery("select siteid,
location, zooaddress from site where usedfor is null");
            ResultSetMetaData rsmd=resultSet.getMetaData();
            int cols=rsmd.getColumnCount();
            String c[]=new String[cols];
            for(int i=0;i<cols;i++){
                c[i]=rsmd.getColumnName(i+1);
                dm.addColumn(c[i]);
            }
            Object row[]=new Object[cols];
            while(resultSet.next()){

```

```

        for(int i=0;i<cols;i++){
            row[i]=resultSet.getString(i+1);
        }
        dm.addRow(row);
    }
    if(dm.getRowCount()==0)
        JOptionPane.showMessageDialog(null,"Not
Found!", "Alert",JOptionPane.WARNING_MESSAGE);
    else
        table.setModel(dm);
    }catch (Exception e1) {}
    }
});
btnNewButton_5.setBounds(619, 64, 128, 23);
contentPane.add(btnNewButton_5);
JButton btnNewButton_1 = new JButton("Clear Table");
btnNewButton_1.setForeground(new Color(0, 128, 0));
btnNewButton_1.setFont(new Font("Microsoft Sans Serif", Font.BOLD,
11));

    btnNewButton_1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            dm.setColumnCount(0);
            dm.setRowCount(0);

        }
    });
    btnNewButton_1.setBounds(842, 11, 128, 23);
    contentPane.add(btnNewButton_1);
}
}

```

## Github links and folder structure:

### Link:

<https://github.com/KulkarniKrishna/Animal-Depot-Management-System>

### Folder structure:

The screenshot shows the GitHub repository page for KulkarniKrishna/Animal-Depot-Management-System. The repository has 1 pull request, 0 stars, and 0 forks. The file list for the 'Assignment\_2' directory is as follows:

File Name	Commit Message	Time Ago
bin	Delete text.txt	44 minutes ago
icons	Add files via upload	42 minutes ago
src	Update DBConnection.java	43 minutes ago
1602-19-737-077_ADMS_Project_Report.pdf	Add files via upload	5 minutes ago

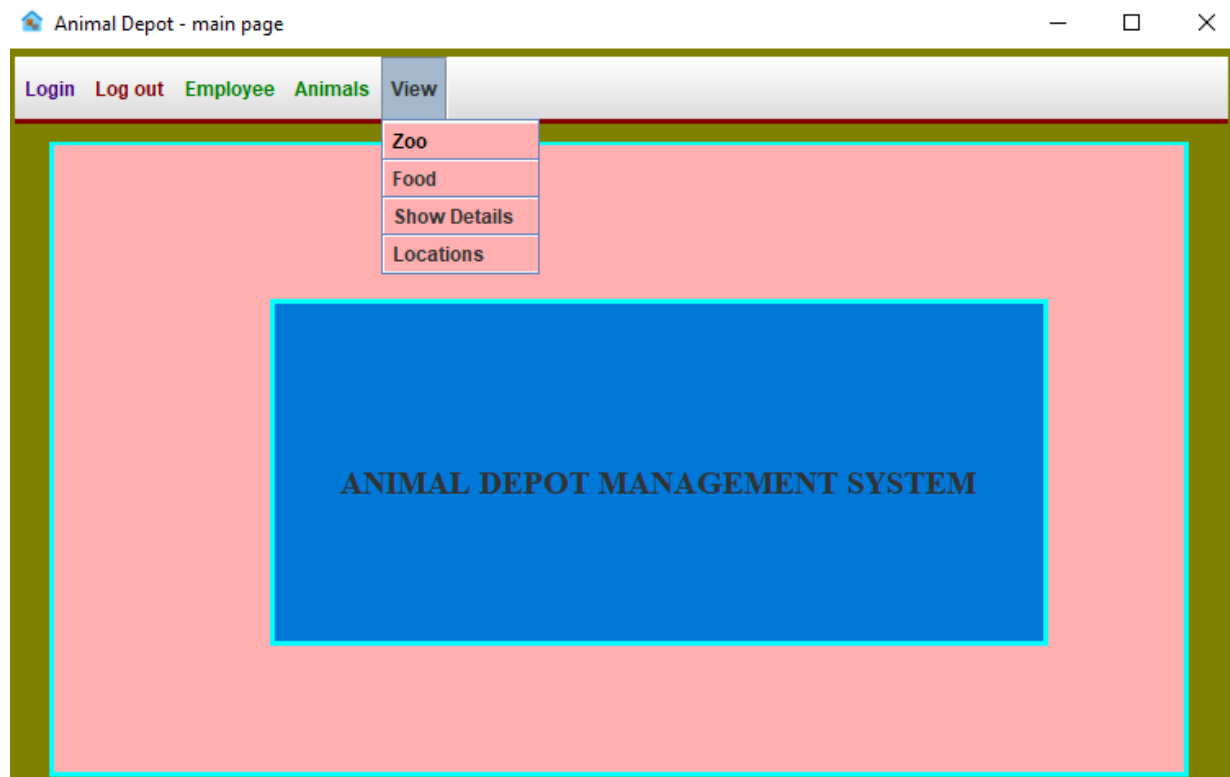
main	Animal-Depot-Management-System / Assignment_2 / src /	Go to file	Add file	...
KulkarniKrishna	Update DBConnection.java	e5d444f	1 hour ago	History
..				
Animal.java	Add files via upload		1 hour ago	
DBConnection.java	Update DBConnection.java		1 hour ago	
Employee.java	Add files via upload		1 hour ago	
Food.java	Add files via upload		1 hour ago	
Location.java	Add files via upload		1 hour ago	
Login.java	Add files via upload		1 hour ago	
MainPage.java	Create MainPage.java		1 hour ago	
Register.java	Add files via upload		1 hour ago	
Show.java	Add files via upload		1 hour ago	
Zoo.java	Add files via upload		1 hour ago	

## TESTING

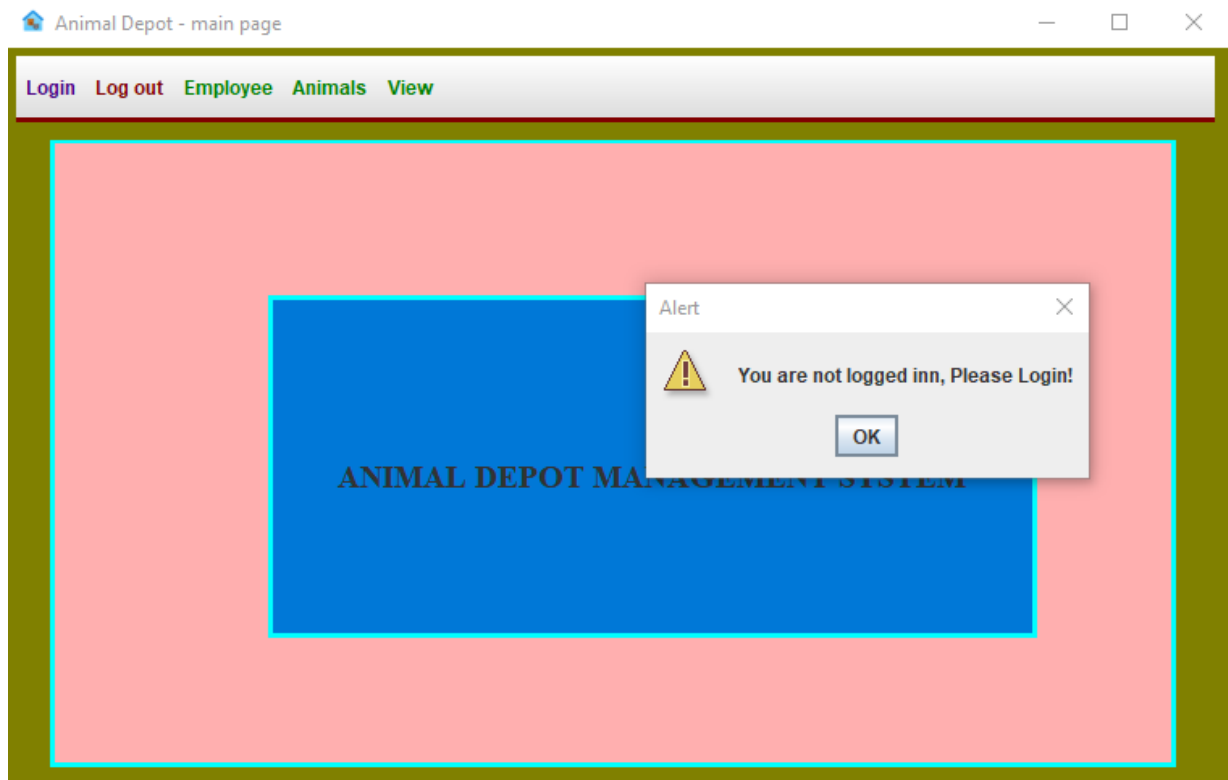
In MainPage module -

User can only access all of the options present in “view” even without logging in.

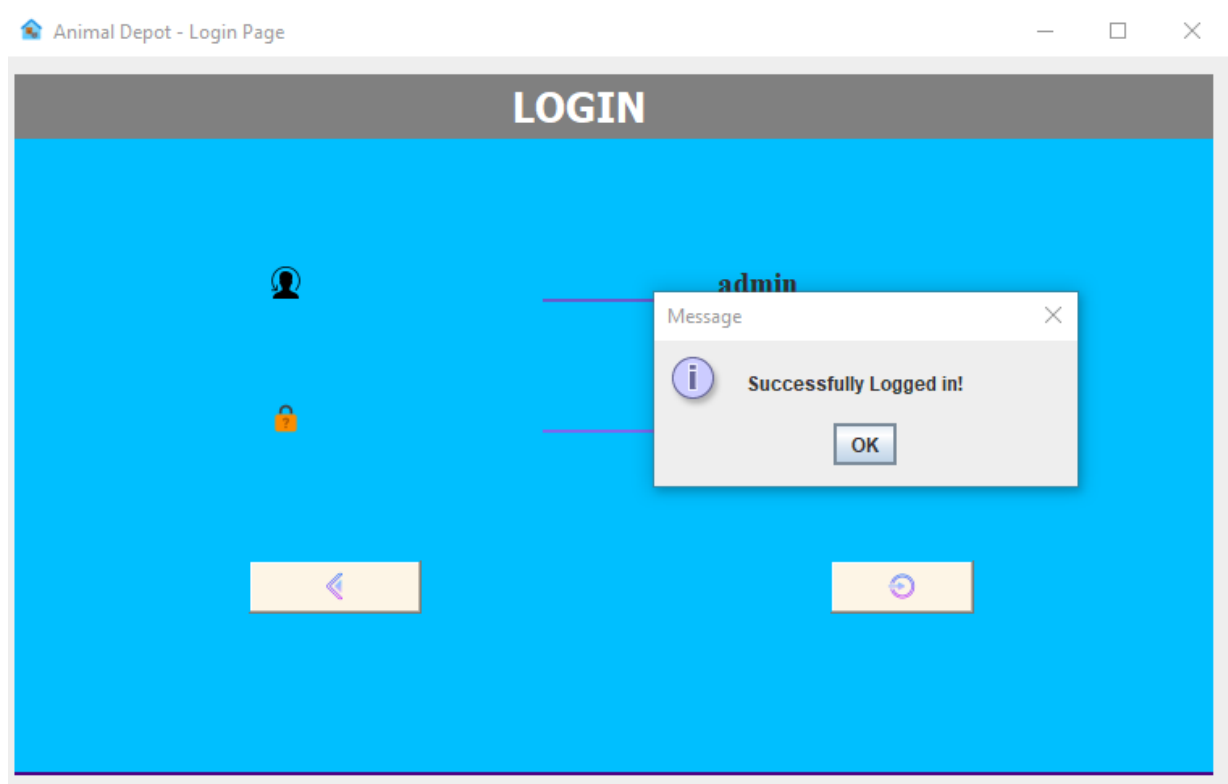
In order to access Employee and Animals Module user need to be logged in.

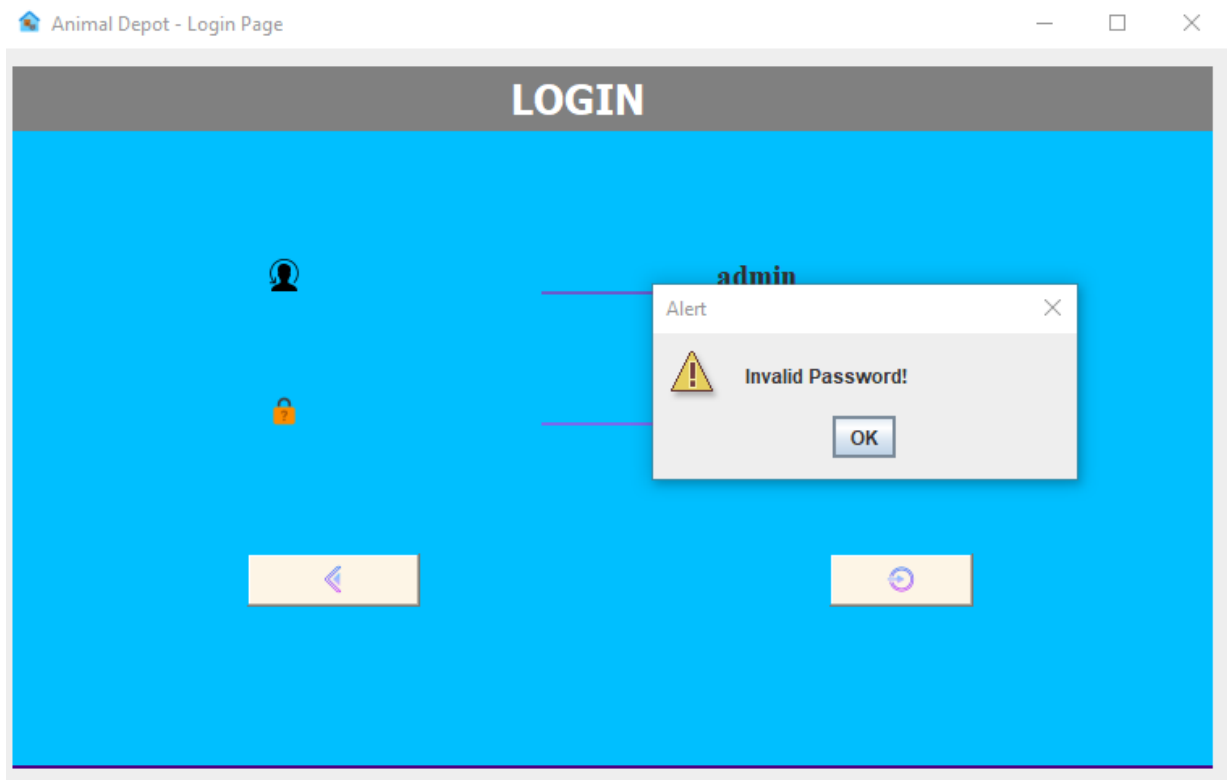
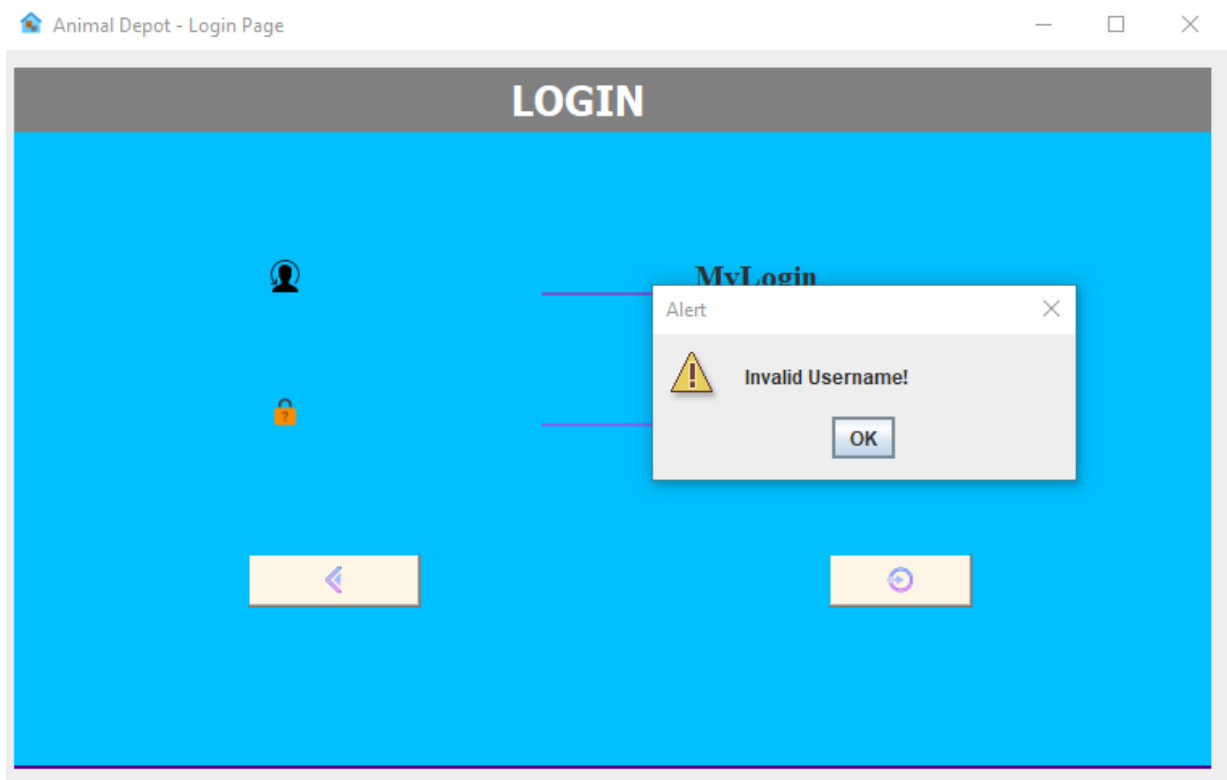


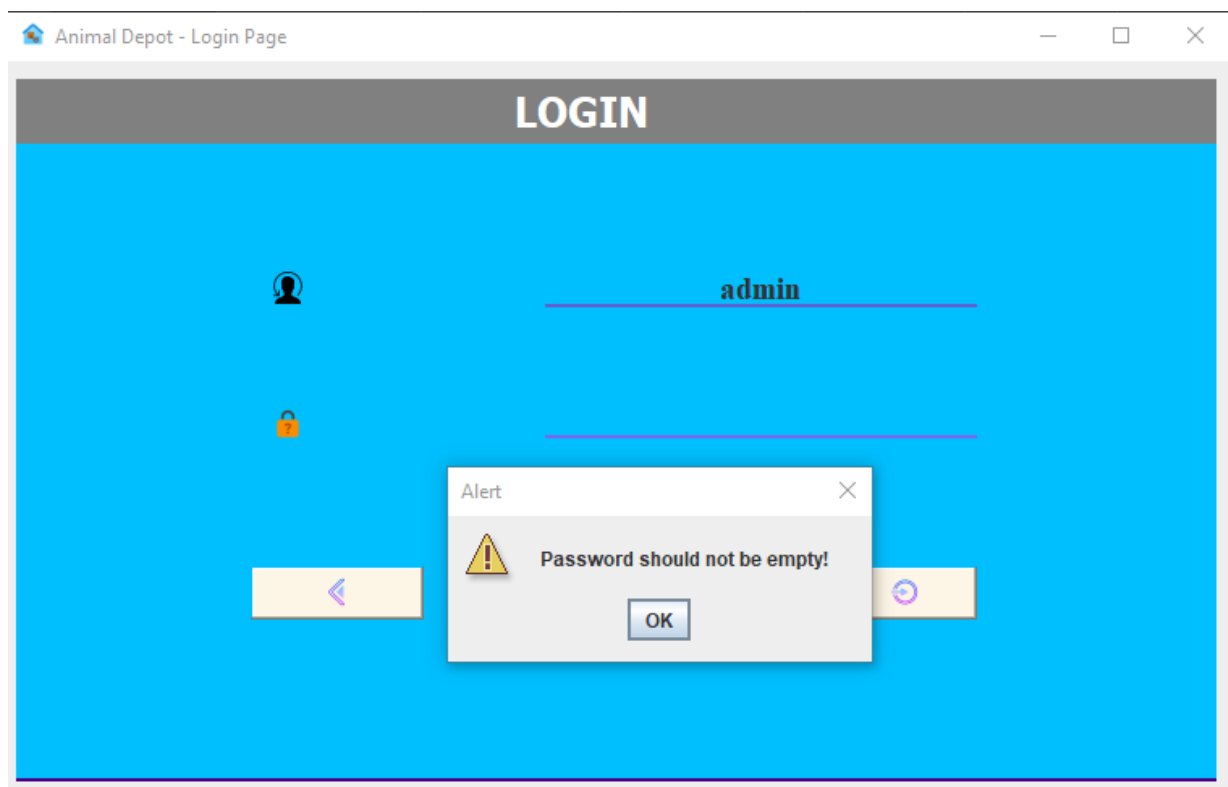
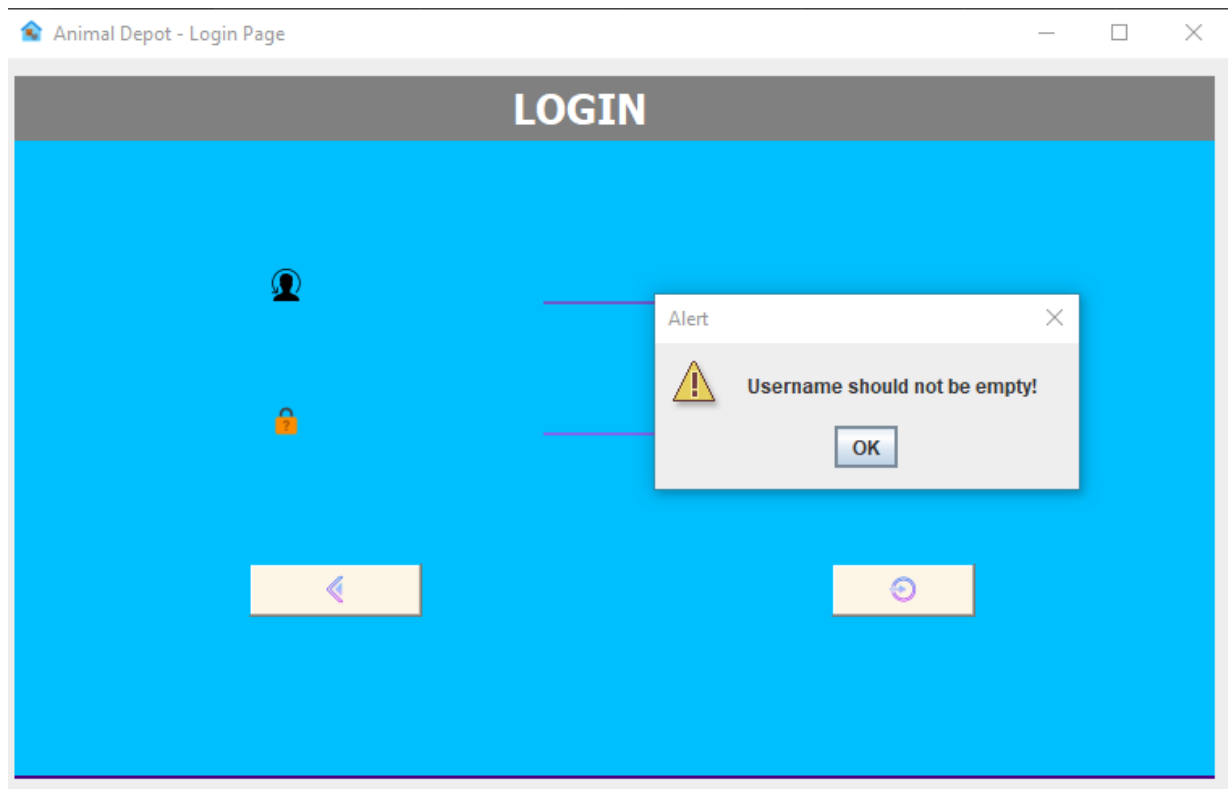


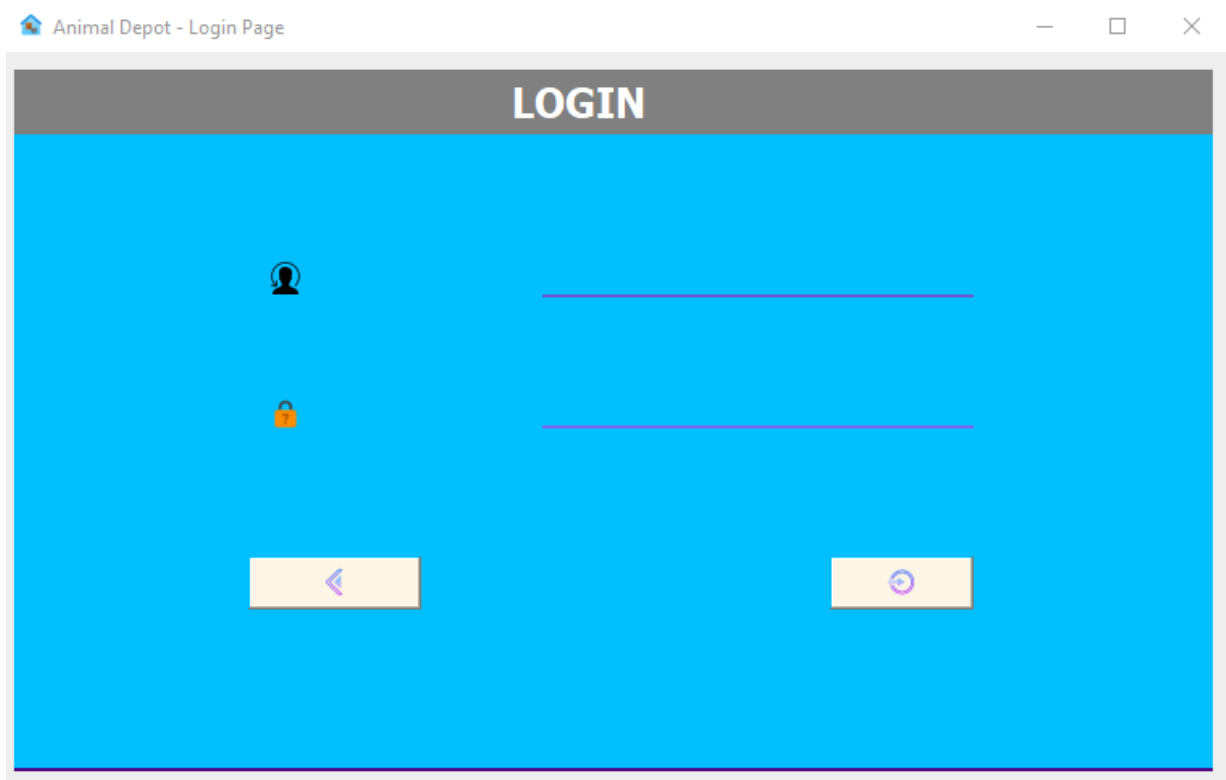
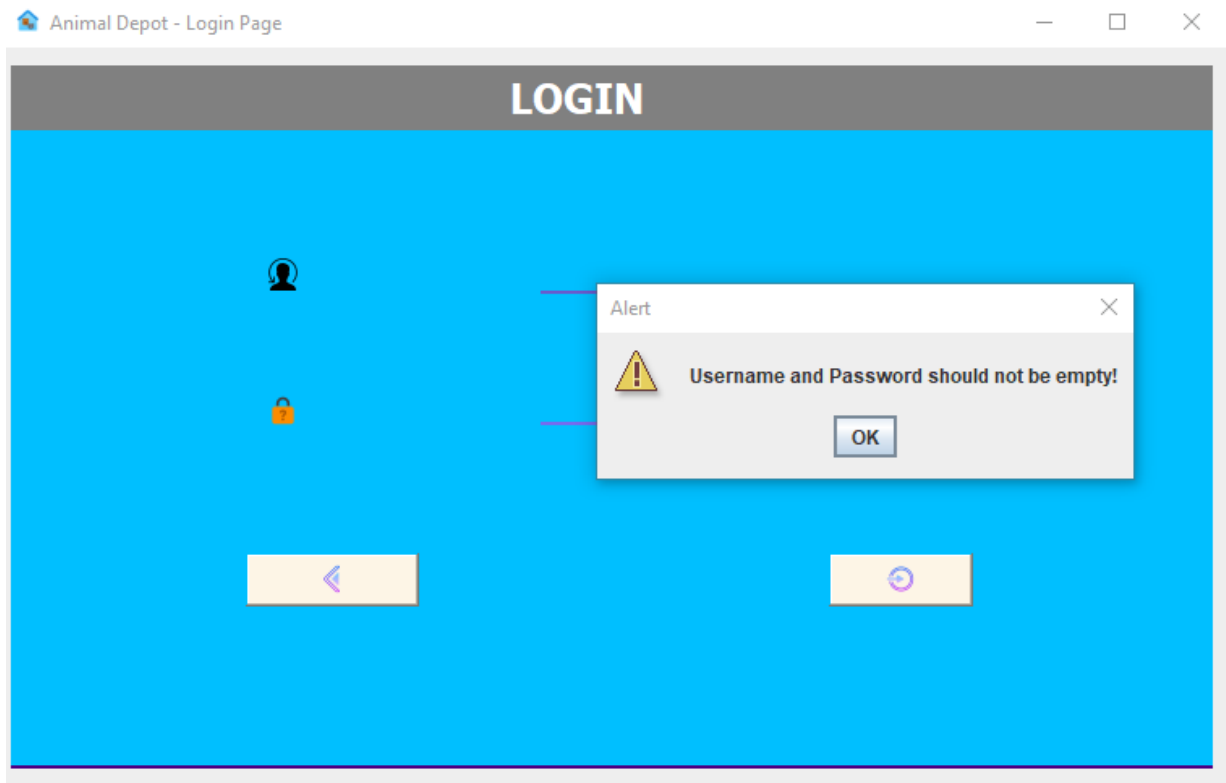


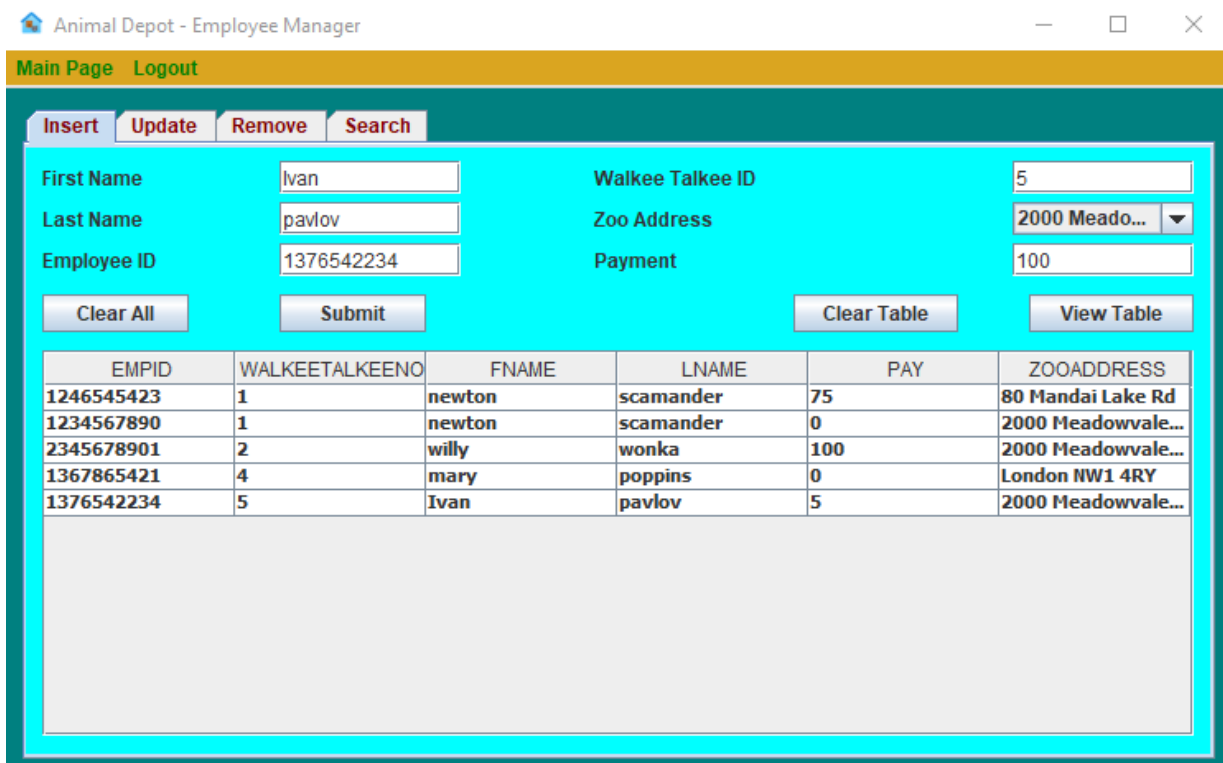
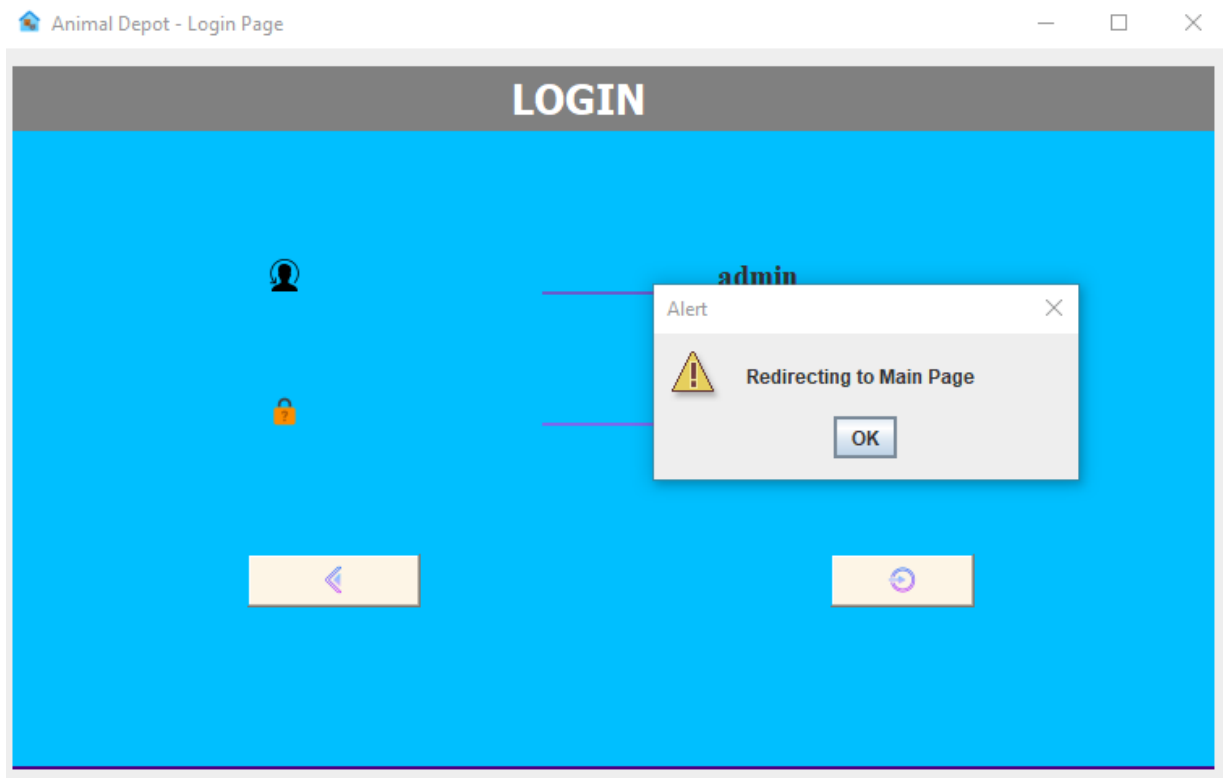
In Login module, data considered - Username = admin, Password = admin.  
(case sensitive)











Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name: Ivan Walkee Talkee ID: 5

Last Name: pavlov Zoo Address: 2000 Meado... ▼

Employee ID: 100

Clear All Clear Table View Table

Message

Successfully inserted

OK

EMPID	WALK	ME	PAY	ZOOADDRESS
1246545423	1		75	80 Mandai Lake Rd
1234567890	1		0	2000 Meadowvale...
2345678901	2	willy wonka	100	2000 Meadowvale...
1367865421	4	mary poppins	0	London NW1 4RY

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name: Ivan Walkee Talkee ID: 5

Last Name: pavlov Zoo Address: 2000 Meado... ▼

Employee ID: 1376542234 Payment: 100

Clear All Submit Clear Table View Table

EMPID	WALKEETALKEENO	FNAME	LNAME	PAY	ZOOADDRESS
1246545423	1	newton	scamander	75	80 Mandai Lake Rd
1234567890	1	newton	scamander	0	2000 Meadowvale...
2345678901	2	willy	wonka	100	2000 Meadowvale...
1367865421	4	mary	poppins	0	London NW1 4RY

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name	robert	Walkee Talkee ID	3
Last Name	ross	Zoo Address	2000 Meado... ▼
Employee ID	1234567890	Payment	150

Clear All Submit Clear Table View Table

Message

ORA-00001: unique constraint (ANIMAL.SDB.SYS\_C007571) violated

OK

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name	robert	Employee ID	1029384756
Last Name	ross	Payment	140

Clear All Modify Clear Table View Table

Message

Employee Not Found!

OK

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name  Employee ID

Last Name  Payment

Clear All Modify Clear Table View Table

FNAME	LNAME	EMPID	PAY	ZOOADDRESS
newton	scamander	1234567890	0	2000 Meadowvale Rd
willy	wonka	2345678901	100	2000 Meadowvale Rd
newton	scamander	1246545423	75	80 Mandai Lake Rd
mary	poppins	1367865421	0	London NW1 4RY
Ivan	pavlov	1376542234	200	2000 Meadowvale Rd

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search


First Name  Employee ID

Last Name  Payment

Clear All Modify Clear Table View Table

FNAME	LNAME	EMPID	PAY	ZOOADDRESS
newton	scamander	1234567890	0	2000 Meadowvale Rd
willy	wonka	2345678901	100	2000 Meadowvale Rd
newton	scamander	1246545423	75	80 Mandai Lake Rd
mary	poppins	1367865421	0	London NW1 4RY
Ivan	pavlov	1376542234	150	2000 Meadowvale Rd

Message

 Details Successfully Updated!

OK



Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name

Last Name

Employee ID

Delete Clear All

Clear Table View Table

FNAME	LNAME	EMPID	PAY	ZOOADDRESS
newton	scamander	1234567890	0	2000 Meadowvale Rd
willy	wonka	2345678901	100	2000 Meadowvale Rd
mary	poppins	1367865421	0	London NW1 4RY
Ivan	pavlov	1376542234	200	2000 Meadowvale Rd

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name

Last Name

Employee ID

Delete Clear All

Clear Table View Table

FNAME	LNAME	EMPID	PAY	ZOOADDRESS
newton	scamander	1234567890	0	2000 Meadowvale Rd
willy	wonka	2345678901	100	2000 Meadowvale Rd
mary	poppins	1367865421	0	London NW1 4RY
Ivan	pavlov	1376542234	200	2000 Meadowvale Rd
robert	scamander	1246545423	0	80 Mandai Lake Rd

Message

Employee successfully removed!!

OK

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name: robert

Last Name: brown

Employee ID: 1234321578

Delete Clear All

Clear Table View Table

Alert

Employee Not Found!

OK

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name: newton

Last Name: bellan

Search Employee Clear All

Search Volunteers Clear Table

Message

Not Found!

OK

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name

Last Name

Search Employee Clear All

Search Volunteers Clear Table

FNAME	LNAME	EMPID	ZOOADDRESS
newton	scamander	1234567890	2000 Meadowvale Rd
mary	poppins	1367865421	London NW1 4RY

Animal Depot - Employee Manager

Main Page Logout

Insert Update Remove Search

First Name

Last Name

Search Employee Clear All

Search Volunteers Clear Table

FNAME	LNAME	EMPID	PAY	ZOOADDRESS
newton	scamander	1234567890	0	2000 Meadowvale Rd

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Animal ID: 106 Weight: 250

Name: Emilia Species: Horse

Age: 9 Eating Frequency: 32

Sex: F Eating Amount: 7

Height: 28 Habitat ID: 1001

Clear All Modify View Data Clear Table

ANIMALID	NAME	AGE	SEX	HEIGHT	WEIGHT	SPECIES	EAT_FRE...	EAT_AMO...	ENCLOSU...
101	Johnny	5	M	50	10	Duck	20	2	1001
102	Chandler	8	M	400	300	Giraffe	10	3	1002
103	Tessa	7	F	120	300	Tiger	15	5	1003
104	Ross	10	M	150	400	Lion	15	6	1004
105	Louis	5	M	50	10	elephant	90	9	1003
106	Emilia	9	F	28	250	Horse	32	7	1001

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Animal ID: 106 Weight: 250

Name: Emilia Species: Horse

Age: 9 Eating Frequency: 32

Sex: F Eating Amount: 7

Height: 28 Habitat ID: 1001

Clear All Modify View Data Clear Table

Message

Successfully Updated!

OK

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Animal ID: 109 Weight: 250

Name: Emilia Species: Horse

Age: 9 Eating Frequency: 32

Sex: F Eating Amount: 7

Height: 28

Clear All Modify Clear Table

Alert  
Invalid Details  
OK

ANIMALID	NAME	AGE	SEX	HEIGHT	WEIGHT	SPECIES	EAT_FRE...	EAT_AMO...	ENCLOSU...
101	Johnny	5	M	50	10	Duck	20	2	1001
102	Chandler	8	M	400	300	Giraffe	10	3	1002
103	Tessa	7	F	120	300	Tiger	15	5	1003
104	Ross	10	M	150	400	Lion	15	6	1004
105	Louis	5	M	50	10	elephant	90	9	1003
106	Emilia	9	F	28	250	Horse	32	7	1001

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Animal ID: 106 Weight: 250

Name: Emilia Species: Horse

Age: 12 Eating Frequency: 22

Sex: F Eating Amount: 7

Height: 32 Habitat ID: 1002

Clear All Submit

Animal ID: Delete View Data Clear Table

ANIMALID	NAME	AGE	SEX	HEIGHT	WEIGHT	SPECIES	EAT_FRE...	EAT_AMO...	ENCLOSU...
101	Johnny	5	M	50	10	Duck	20	2	1001
102	Chandler	8	M	400	300	Giraffe	10	3	1002
103	Tessa	7	F	120	300	Tiger	15	5	1003
104	Ross	10	M	150	400	Lion	15	6	1004
105	Louis	5	M	50	10	elephant	90	9	1003
106	Emilia	12	F	32	250	Horse	22	7	1002

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Animal ID: 106 Weight: 250

Name: Emilia Species: Horse

Age: 12 Eating Frequency: 22

Sex: F Eating Amount: 7

Height: 32 Habitat ID: 1002

Clear All Submit

Message

Successfully Inserted!

OK

Animal ID: View Data Clear Table

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Animal ID: 101 Weight: 250

Name: Emilia Species: Horse

Age: 12 Eating Frequency: 22

Sex: F Eating Amount: 7

Height: 32 Habitat ID: 1002

Clear All Submit

Message

Invalid Details

OK

Animal ID: Delete View Data Clear Table

ANIMALID	NAME	AGE	SEX	HEIGHT	WEIGHT	SPECIES	EAT_FRE...	EAT_AMO...	ENCLOSU...
101	Johnny	5	M	50	10	Duck	20	2	1001
102	Chandler	8	M	400	300	Giraffe	10	3	1002
103	Tessa	7	F	120	300	Tiger	15	5	1003
104	Ross	10	M	150	400	Lion	15	6	1004
105	Louis	5	M	50	10	elephant	90	9	1003

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Animal ID	<input type="text"/>	Weight	<input type="text"/>
Name	<input type="text"/>	Species	<input type="text"/>
Age	<input type="text"/>	Eating Frequency	<input type="text"/>
Sex	<input type="text"/>	Eating Amount	<input type="text"/>
Height	<input type="text"/>	Update ID	<input type="text"/>

Clear All Submit

Animal ID

Alert

Animal not found!

OK

Clear Table

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Animal ID	<input type="text"/>	Weight	<input type="text"/>
Name	<input type="text"/>	Species	<input type="text"/>
Age	<input type="text"/>	Eating Frequency	<input type="text"/>
Sex	<input type="text"/>	Eating Amount	<input type="text"/>
Height	<input type="text"/>	Update ID	<input type="text"/>

Clear All Submit

Animal ID

Message

Successfully Deleted!

OK

Clear Table

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Animal ID  Weight

Name  Species

Age  Eating Frequency

Sex  Eating Amount

Height  Habitat ID

Clear All Submit

---

Animal ID  Delete View Data Clear Table

ANIMALID	NAME	AGE	SEX	HEIGHT	WEIGHT	SPECIES	EAT_FRE...	EAT_AMO...	ENCLOSU...
101	Johnny	5	M	50	10	Duck	20	2	1001
102	Chandler	8	M	400	300	Giraffe	10	3	1002
103	Tessa	7	F	120	300	Tiger	15	5	1003
104	Ross	10	M	150	400	Lion	15	6	1004
106	Emilia	9	F	28	250	Horse	32	7	1001

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Species  Search by Species

Species Details

ANIMALID	NAME	WEIGHT	HEIGHT
101	Johnny	10	50



Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Species

Search by Species

Species Details

ANIMALID	NAME	AGE	SEX	SPECIES
101	Johnny	5	M	Duck
102	Chandler	8	M	Giraffe
106	Emilia	9	F	Horse
104	Ross	10	M	Lion
103	Tessa	7	F	Tiger

Animal Depot - Animal Manager

Main Page Logout

Insert || Remove Update View

Species

Search by Species

Species Details

Alert

⚠ Not Found!

OK

Animal Depot

Main Page

Logout

## Zoo Details

Country

Clear

View

ADDRESS	NAME	PHONE	CITY	COUNTRY
15500 San Pa...	San Diego Zo...	12345	Escondido	USA

Animal Depot

Main Page

Logout

## Zoo Details


Country

Clear

View

ADDRESS	NAME	PHONE	CITY	COUNTRY
---------	------	-------	------	---------

Alert

 Not Found!

OK

Animal Depot
Main Page Logout

## Food Details

Get Food
Expiring Food
Clear

SITEID	LOCATION	FOODID	NAME	STOCK_SERVING	DATE_PURCHA...	DATE_EXPIRED
101	Area 1	11	frozen chicken	80	2018-11-08 00:0...	2020-08-10 00:...
102	Area 2 North	12	bamboo	69	2018-10-21 00:0...	2019-03-10 00:...
103	Area 1 West	13	fish	32	2018-11-08 00:0...	2021-03-31 00:...
104	Area 1 East	14	eggs	99	2018-11-09 00:0...	2018-12-10 00:...

Animal Depot
Main Page Logout

## Food Details

Get Food
Expiring Food
Clear

FOODID	NAME	DATE_EXPIRED
14	eggs	2018-12-10 00:00:00
12	bamboo	2019-03-10 00:00:00
11	frozen chicken	2020-08-10 00:00:00
13	fish	2021-03-31 00:00:00

Animal Depot

Main Page Logout

**Location Details**

Enclosure ID

Temperature

Square Feet

Depth

Biome

**Clear All**

**View By ID** **Suitable Habitat** **Clear Table**

**View By Temp** **Not in use**

Alert

Not Found!

OK

Animal Depot

Main Page Logout

**Location Details**

Enclosure ID

Temperature

Square Feet

Depth

Biome

**Clear All**

**View By ID** **Suitable Habitat** **Clear Table**

**View By Temp** **Not in use**

Alert

Not Found!

OK

Animal Depot

Main Page Logout

**Location Details**

Enclosure ID

Temperature

Square Feet

Depth

Biome

**Clear All**

**View By ID** **Suitable Habitat** **Clear Table**

**View By Temp** **Not in use**

ENCLOSURE_ID	BIOME	SQFT	DEPTH	SITEID
1003	Ocean	100	20	103

Animal Depot

Main Page

Logout

Location Details

Enclosure ID

Temperature

Square Feet

50

Depth

10

Biome

Tropical Rain Forest

Clear All

View By ID

Suitable Habitat

Clear Table

View By Temp

Not in use

ENCLOSURE_ID	BIOME	SQFT	DEPTH	SITEID
1001	Tropical Rain Forest	50	10	102

Animal Depot

Main Page

Logout

Location Details

Enclosure ID

1003

Temperature

Square Feet

Depth

Biome

Clear All

View By ID

Suitable Habitat

Clear Table

View By Temp

Not in use

ANIMALID	NAME	SPECIES
103	Tessa	Tiger

Animal Depot

Main Page

Logout

Location Details

Enclosure ID

1001

Temperature

Square Feet

Depth

Biome

Clear All

View By ID

Suitable Habitat

Clear Table

View By Temp

Not in use

ANIMALID	NAME	SPECIES
101	Johnny	Duck

Animal Depot

Main Page Logout

### Location Details

Enclosure ID

Temperature

Square Feet

Depth

Biome

ENCLOSUR...	BIOME	SITEID	LOCATION	USEDFOR	HUMIDITY	ZOOADDRE...
1004	Dessert	104	Area 1 East	habitat	0	80 Mandai L...

Animal Depot

Main Page Logout

### Location Details

Enclosure ID

Temperature

Square Feet

Depth

Biome

ENCLOSUR...	BIOME	SITEID	LOCATION	USEDFOR	HUMIDITY	ZOOADDRE...
1001	Tropical Rai...	102	Area 2 North	habitat	9	2000 Meado...

Animal Depot

Main Page Logout

### Location Details

Enclosure ID

Temperature

Square Feet

Depth

Biome

SITEID		LOCATION	ZOOADDRESS
105		Area 5	2000 Meadowvale Rd

Animal Depot

Main Page Logout

### Location Details

Enclosure ID

Temperature

Square Feet

Depth

Biome

Clear All

View By ID Suitable Habitat Clear Table

View By Temp Not in use

Alert

⚠ Not Found!

OK

Animal Depot - main page

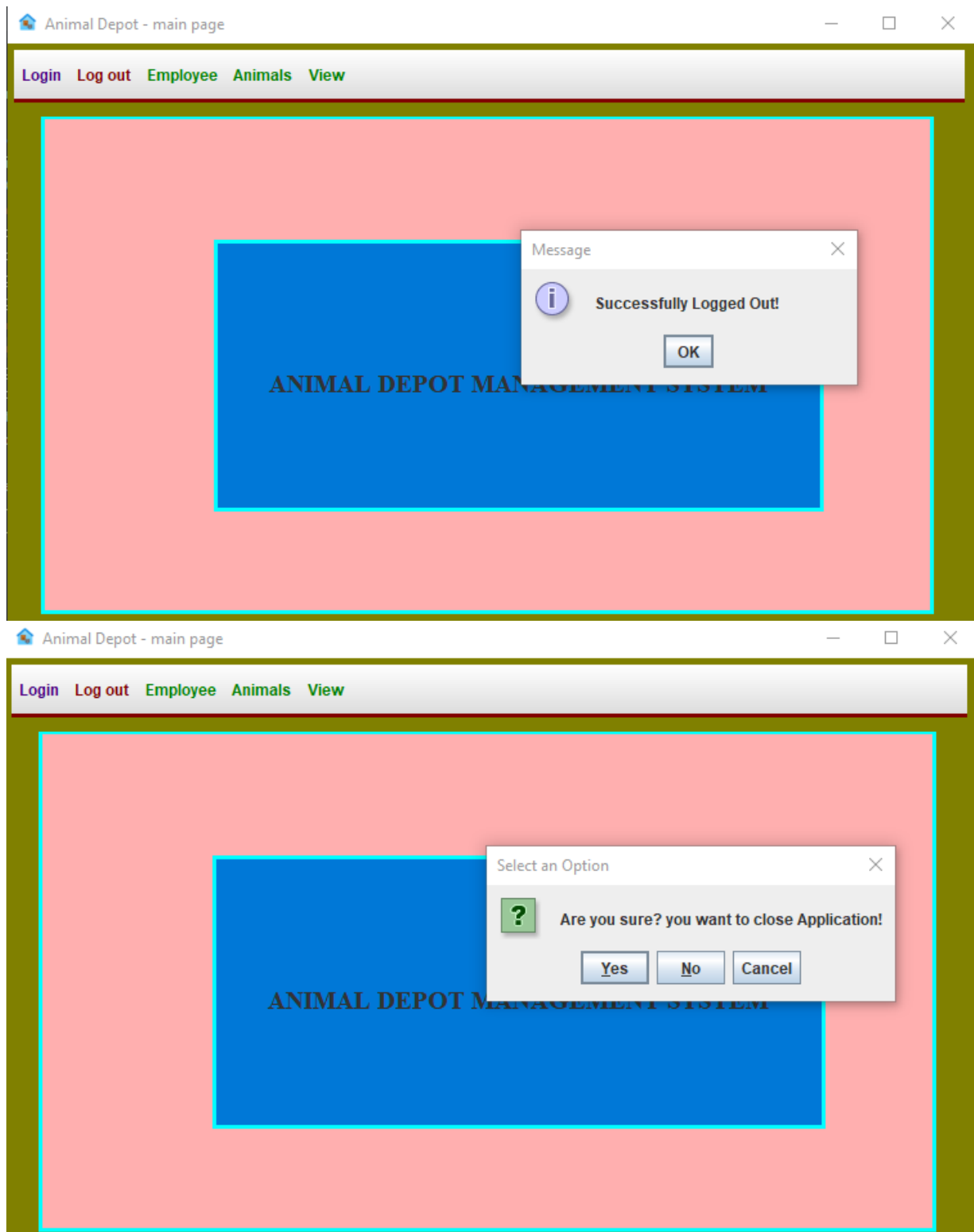
Login Logout Employee Animals View

ANIMAL DEPOT MANAGEMENT SYSTEM

Select an Option

⚠ Are you sure, You want to logout!

Yes No Cancel



## DISCUSSION AND FUTURE WORK

- ✓ Adding more security by using the feature Triggers T-SQL for accessing AADB.



- ✓ Improving the Interface of the Application by making more user friendly.
- ✓ Adding more functionalities such as searching, viewing based on the tags selected by user.
- ✓ Convenient to switch from one page to other i.e., not always going to main page and redirecting to desired page.

## REFERENCES

<https://www.w3schools.com/sql/>

<https://www.oracletutorial.com/oracle-basics/>

<https://www.javatpoint.com/java-swing>

<https://www.youtube.com/watch?v=oeswfZz4IW0&list=PLS1QulWo1RIbYMA5Ijb72QHaHvCrPKfS2&index=3>

# THANK YOU

