

Chapter 1

INTRODUCTION

1.1 Introduction to Database Management System

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A database-management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, EnterpriseDB, MongoDB, MariaDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, MemSQL, SQLite and IBM.

A database is not generally portable across different DBMSs, but different DBMSs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Sometimes a DBMS is loosely referred to as a "database".

Applications:

Databases are used to support internal operations of organizations and to underpin online interactions with customers and suppliers.

Databases are used to hold administrative information and more specialized data, such as engineering data or economic models. Examples of database applications include computerized library systems, flight reservation systems, computerized parts inventory systems, and many content management systems that store websites as collections of webpages in a database.

1.2 Introduction to JAVA:

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.^[17] Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture.

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

1.3 Introduction to NetBeans

NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers.

The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++ and HTML5.

NetBeans is cross-platform and runs on Microsoft Windows, macOS, Linux, Solaris and other platforms supporting a compatible JVM.

The editor supports many languages from Java, C/C++, XML and HTML, to PHP, Groovy, Javadoc, JavaScript and JSP. Because the editor is extensible, you can plug in support for many other languages.

The NetBeans Team actively supports the product and seeks feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback. Over 18 million downloads of the NetBeans IDE to date, and over 800,000 participating developers, the NetBeans project is thriving and continues to grow. A new version was released 8.2/october 3, 2016. NetBeans IDE is the official IDE for Java 8. With its editors, code analyzers, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references.

1.4 Introduction to JDBC:

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 Java based data access technology and 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.

1.5 Introduction to SQL

Structure Query Language (SQL) is a programming language to request data from a database, to add, update, or remove data within a database, or to manipulate the metadata of the database. SQL was the first commercial language introduced for E.F Codd's **Relational** model. Today almost all RDBMS (MySQL, Oracle, Infomix, Sybase, MS Access) uses **SQL** as the standard database language. SQL is used to perform all type of data operations in RDBMS. Commonly used statements are grouped into the following categories:

➤ **Data Query Language (DQL)**

- SELECT - Used to retrieve certain records from one or more tables.

➤ **Data Manipulation Language (DML)**

- INSERT - Used to create a record.
- UPDATE - Used to change certain records.
- DELETE - Used to delete certain records.

➤ **Data Definition Language (DDL)**

- CREATE - Used to create a new table, a view of a table, or other object in database.
- ALTER - Used to modify an existing database object, such as a table.
- DROP - Used to delete an entire table, a view of a table or other object in the database

➤ **Data Control Language (DCL)**

- GRANT - Used to give a privilege to someone.
- REVOKE - Used to take back privileges granted to someone.

1.6 Introduction to MySQL:

MySQL is an open-source relational database management system(RDBMS). Its name is a combination of “My”, the name of co-founders Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

phpMyAdmin

phpMyAdmin is a free and open source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services. It is a popular and free open source tool used for administering MySQL with a web browser. Typical operations such as the management of databases, tables, indexes, permissions, and so on are executed with the user interface. Administrators can also use phpMyAdmin to directly execute any SQL statement.

1.7 Introduction to Zoo Management System

Zoo Management is the process by which zoo maintains a record of adoption and healthcare. Any animal should be recorded of its health check and people who adopt the animals. Although, it seems like a mundane task, it involves many aspects including, admission and release of animals from the healthCare Centre and user login Management.

Zoo Management can create a huge burden and unwanted stress for zoo record keeping.. To avoid these issues of large paper use due to keeping record of each animal and backup and restoration of records Zoo Management Systems are used.

A Zoo management system is software designed to organize all the records of animals . These tasks can include keeping track of adoptions, adopters, animals and its injuries from the past or in the present.. Zoo management software often requires very little input from the Admin. The Admin is required to input animal injury information and adoption information—then the software stores the information and displays the information where ever required.

The zoo Management System deals with the database aspects of animals, adoptions, adopters, injury record of animals for long period of time. The outstanding benefit of Zoo Management System is its easy implementation. Other advantages of Zoo Management System are its extensive features and reports.

Features:

Zoo Management System gives you the power to:

- Manage HealthCare and Adoption of animals easily.
- Separate interfaces for users and employees.
- Generate medical record of each animal at the convenience of a mouse click.
- Manage the Adoption Processes according to the Available animals.
- Manage animals which are born inside the Zoo.
- Manage your own information.

Chapter 2

REQUIREMENTS SPECIFICATION

2.1 Hardware Requirements

- Processor: Intel(R) Core(TM) i3 CPU @1.70Ghz
- Installed memory (RAM): 4.00GB
- System type: 64-bit Operating System, x64-based processor
- Graphics: NVIDIA (R) GEFORCE(R)
- Total size of Hard disk: 1TB

2.2 Software Requirements

- Operating System: Microsoft windows 8.1 and above.
- Integrated Development Environment:
 1. Front-end tool:- NetBeans 8.2
 2. Back-end tool:- phpMyAdmin
- Programming Language: JAVA

Chapter 3

SYSTEM DESIGN

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

System designing in terms of software engineering has its own value and importance in the system development process as a whole. To mention it may though seem as simple as anything or simply the design of systems, but in a broader sense it implies a systematic and rigorous approach to design such a system which fulfils all the practical aspects including flexibility, efficiency and security.

Before there is any further discussion of system design, it is important that some points be made clear. As it goes without saying that nothing is created that is not affected by the world in which it's made. So, the systems are not created in a vacuum.

They are created in order to meet the needs of the users. They are not only intended to solve the existing problems, but they also come up with acceptable solutions to the problems that may arise in the future. The whole process of system development, from blueprint to the actual product, involves considering all the relevant factors and taking the required specifications and creating a useful system based on strong technical, analytical and development skills of the professionals.

Let's get back to our discussion about what the system design phase is and the importance of system design in the process of system development. Being another important step in the system development process, system designing phase commences after the system analysis phase is completed. It's appropriate to mention that the output or the specifications taken through the phase of system analysis become an input in the system design phase which in turn leads to workout based on the user defined estimations.

The importance of this phase may be understood by reason of the fact that it involves identifying data sources, the nature and type of data that is available. For example, in order to design a zoo management system, there is a need for using inputs, such as, new names,

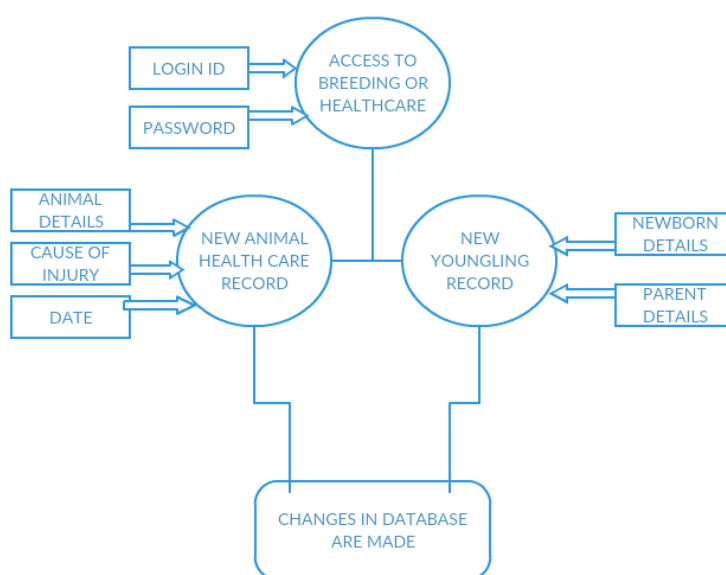
adopter details, additions or deductions etc. This facilitates understanding what kind of data is available and by whom it is supplied to the system so that the system may be designed considering all the relevant factors. In addition, system designing leads to ensure that the system is created in such a way that it fulfils the need of the users and keep them at ease being user-oriented. In terms of the flexibility, one of the main objectives of this phase is that it is intended to design such a system which can be dynamic in nature and responsive to the changes if required. Another important objective is that the phase of system designing is concerned with creating the system which can work efficiently providing the required output and being responsive to the time within a given time limit. The aspect of reliability and physical security of data cannot be ignored. With this respect, the system designing phase ensures security measures of the system effectively and efficiently.

3.1 Data Flow Diagram

A picture is worth a thousand words. A Data Flow Diagram (DFD) is traditional visual representation of the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or combination of both.

It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.



DFD for the Zoo Management System:

Fig:3.1.1

3.2 Logical Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

Logical Database Schema – this schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints

Logical Schema

ANIMAL TABLE

ANIMAL	A_Name	A_Id	S_Id	Gender	B_Date	Admsn_Date	Immigrant
--------	--------	------	------	--------	--------	------------	-----------

BREEDING TABLE

A_Id	Date_Of_Delivery	No_Of_Cubs
------	------------------	------------

CATEGORY TABLE

C_Name	C_Id
--------	------

SPECIES TABLE

S_Name	S_Id	C_Id
--------	------	------

HEALTHCARE TABLE

A_Name	A_Id	Cause_Of_Injury	Admit_Date	Gender	Release_Date
--------	------	-----------------	------------	--------	--------------

ADOPTION TABLE

A_Name	Ad_Id	A_Id	Amount	Adoption_Date
--------	-------	------	--------	---------------

ADOPTION_DETAILS TABLE

Name	Ad_Id	Gender	Phone	No_Of_Adoption
------	-------	--------	-------	----------------

ADOPTLoginID TABLE

Name	LoginID	Password
------	---------	----------

ADMINLoginID TABLE

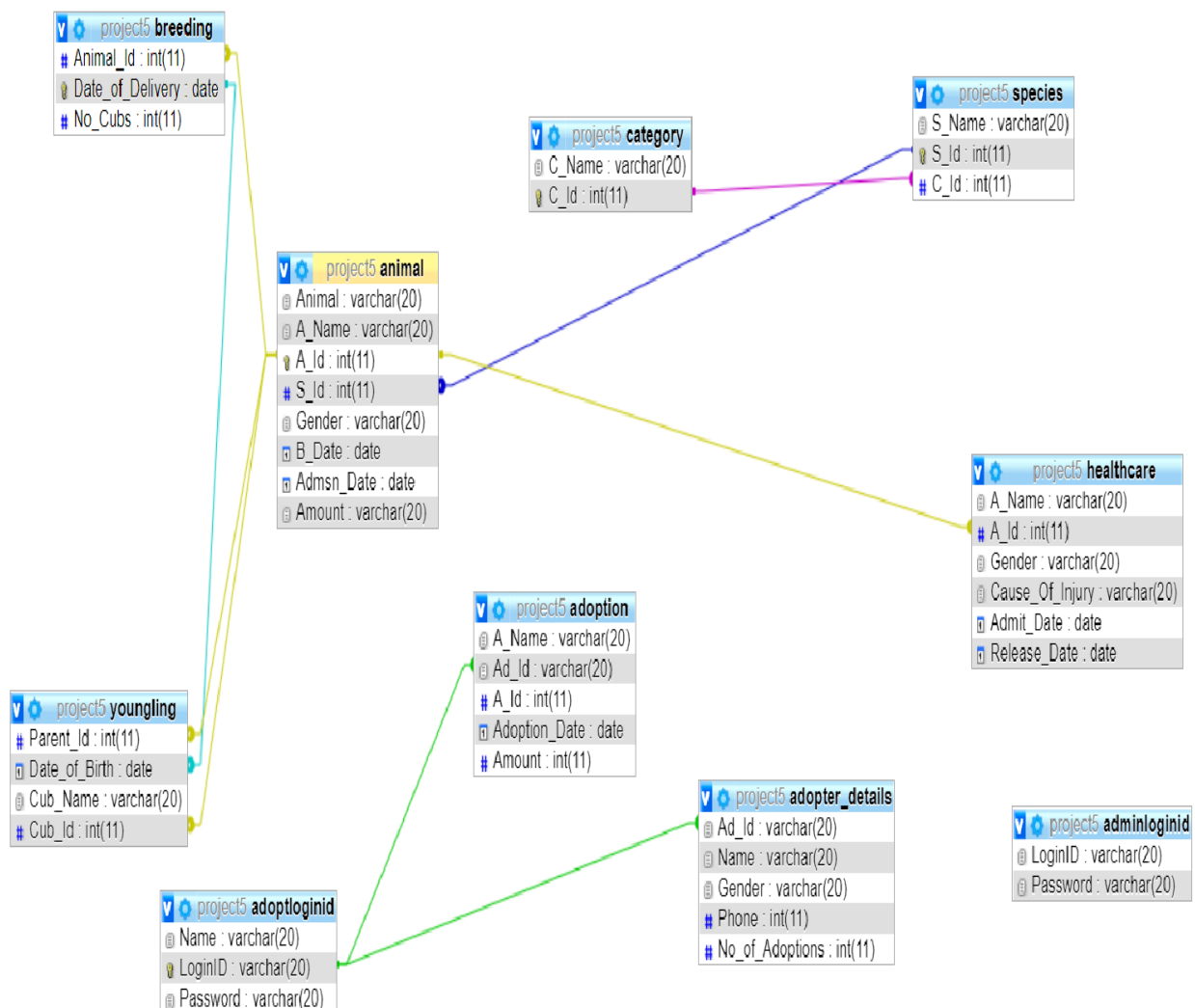
Name	LoginID	Password
------	---------	----------

YOUNGLING TABLE

Parent_Id	Date_Of_Birth	Cub_Name	Cub_Id
-----------	---------------	----------	--------

3.3 E-R Diagram

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.



Chapter 4

IMPLEMENTATION

The special methods used in the project are explained.

4.1 Triggers

A trigger is a named database object that is associated with a table, and that activates when a particular event occurs for the table. Some uses for triggers are to perform checks of values to be inserted into a table or to perform calculations on values involved in an update.

A trigger is defined to activate when a statement inserts, updates, or deletes rows in the associated table. These row operations are trigger events. For example, rows can be inserted by INSERT or LOAD DATA statements, and an insert trigger activates for each inserted row. A trigger can be set to activate either before or after the trigger event. For example, you can have a trigger activate before each row that is inserted into a table or after each row that is updated.

The trigger used in the project is a after insert trigger .In this trigger we are inserting a new row in the Animal table immediately after we insert a new row in Adoption.

4.2 Stored procedure

A procedure (often called a stored procedure) is a subroutine like a subprogram in a regular computing language, stored in database. A procedure has a name, a parameter list, and SQL statement(s). All most all relational database system supports stored procedure, MySQL 5 introduce stored procedure. In the following sections we have discussed MySQL procedure in details and used MySQL 5.6 under Windows 7. MySQL 5.6 supports "routines" and there are two kinds of routines : stored procedures which you call, or functions whose return values you use in other SQL statements the same way that you use pre-installed MySQL functions like pi(). The major difference is that UDFs can be used like any other expression within SQL statements, whereas stored procedures must be invoked using the CALL statement.

4.3 Stored procedure and trigger used in the above application

Stored procedure:

```
DROP PROCEDURE `insertnew`;  
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertnew`(IN `adname`  
VARCHAR(20), IN `aduname` VARCHAR(20), IN `adpwd` VARCHAR(20))  
NOT DETERMINISTIC CONTAINS SQL  
SQL SECURITY DEFINER insert into adoptloginid  
VALUES(adname,aduname,adpwd);
```

trigger:

```
DROP TRIGGER IF EXISTS `AFTER_update`;  
CREATE DEFINER=`root`@`localhost`  
TRIGGER `AFTER_update` AFTER INSERT ON `adoption`  
FOR EACH ROW UPDATE animal  
set Amount_A=(Amount_A-10)  
where A_Id=NEW.A_Id;
```

Chapter 5

TESTING

5.1 Introduction to testing

Verification and validation is a generic name given to checking processes, which ensures that the software confirms to its specifications and meets the demands of users.

- **Validation**

Validation involves checking that the program has implanted meets the requirement.

- **Verification**

Verification involves checking that the program confirms to its specification.

5.2 Stages in the Implementation of Testing

- **Unit Testing**

Each individual unit is tested for correctness. These individual components will be tested to ensure that they operate correctly.

- **Module Testing**

A module is a collection of dependent components such as a function. A module encapsulates related components so can test without other system modules.

- **Sub-system Testing**

This phase involves testing collection of modules, which have been integrated into sub-systems. Sub-systems may be independently designed and implemented.

- **System testing**

The Sub-systems are integrated to make up the entire system. The errors that result from unanticipated interaction between sub-systems and system components are removed.

- **Acceptance testing**

This is the final stage in the testing process before the system is tested for operational use. Any requirement problem or requirement definition problem revealed from acceptance testing are considered and made error free.

- **Test plan**

Careful planning is needed to the most of testing and controlled testing cost.

5.3 Results

Several errors were detected and rectified and the whole project is working as it should with proper output and high efficiency. The several tests performed are as follows:

Test case id	Test case	Input data	Steps to execute the test case	Expected Result	Actual Result	Pass/Fail
1	Admin Login screen.	Wrong username or password.	After entering the data click the login button.	A proper message indicating the error should appear and the user should be redirected to login screen.	A message was displayed saying incorrect username or password.	Pass
2	Adopter Login Screen.	Wrong username or password.	After entering the data click the login button.	A proper message indicating the error should appear and the user should be redirected to login screen.	A message was displayed saying incorrect username or password.	Pass
3	HealthCare Check Screen	Trying to Give an animal Id..	Select the Animal Id And press Submit Button.	Animal, its details and its cause of Injury should be Displayed on the screen .	Animal, its details and its cause of Injury should be Displayed on the screen .	Pass
4	Deletion of entry in HealthCare list screen	Entering of the Animal ID	Upon Entering of the Animal ID	The Animal whose ID has Been Entered must get deleted from the list And the list must be updated	The Animal whose ID has Been Entered must get deleted from the list And the list must be updated	pass

5	Selection through Combo box	Insertion into the first Combo box and clicking the "Selected" button	Upon Insertion into the first Combo box and clicking "Selected" button	The Second Como box should be reflected with values only pertained to the first Combo box	The Second Como box should be reflected with values only pertained to the first Combo box	pass
6	Submit button in Categories Page	Insertion into the second Combo box and clicking "Submit" button	Upon Insertion into the second Combo box and clicking "Submit" button	The data regarding the Animal selected should be reflected in the text boxes provided.	The data regarding the Animal selected should be reflected in the text boxes provided.	pass

Chapter 6

SNAPSHOTS



Figure 6.1: Main Screen.

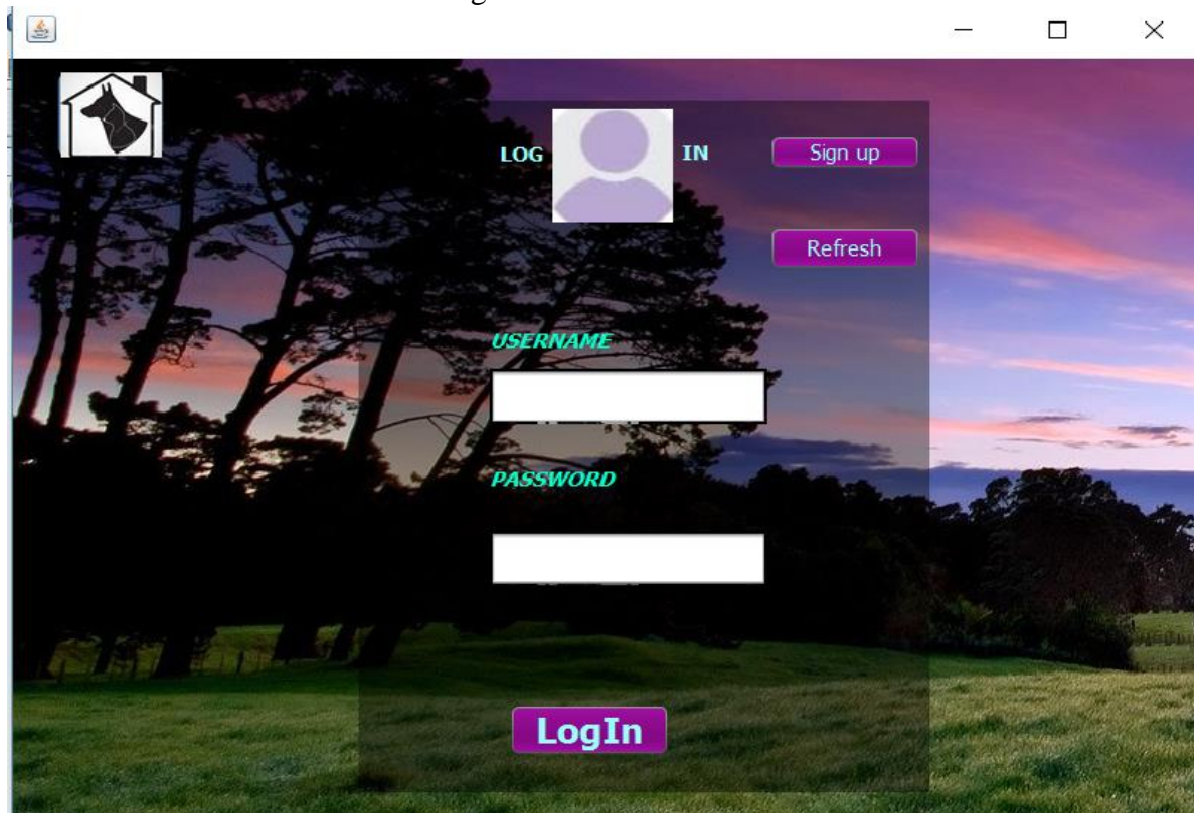
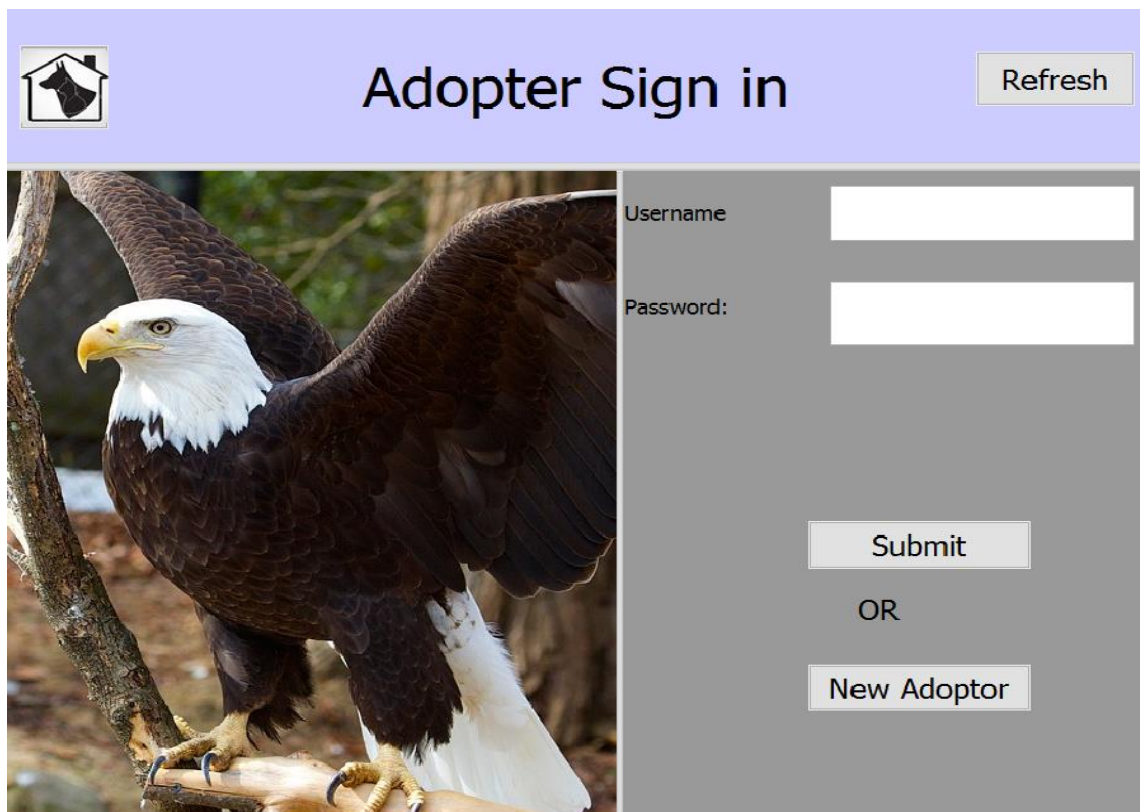
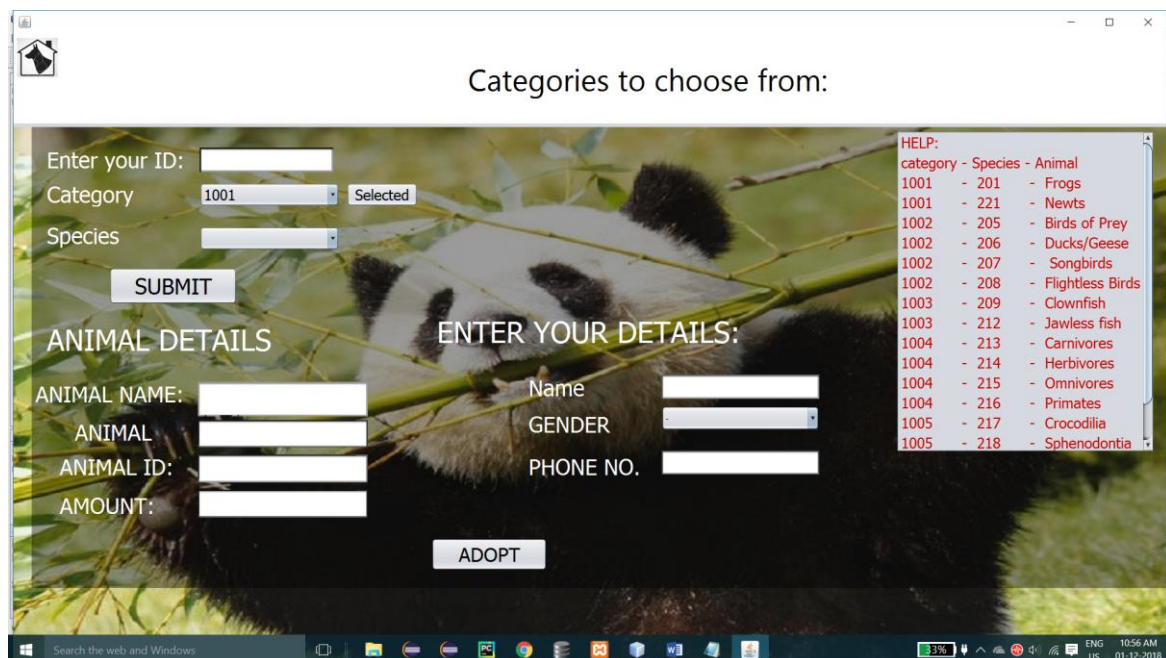


Fig 6.2: Administrator Sign-In.



The Adopter Sign in screen features a purple header with a house icon and the title "Adopter Sign in". A "Refresh" button is in the top right. The main area is split: the left side shows a bald eagle perched on a branch, and the right side contains a login form. The form has fields for "Username" and "Password:", a "Submit" button, an "OR" separator, and a "New Adoptor" button.

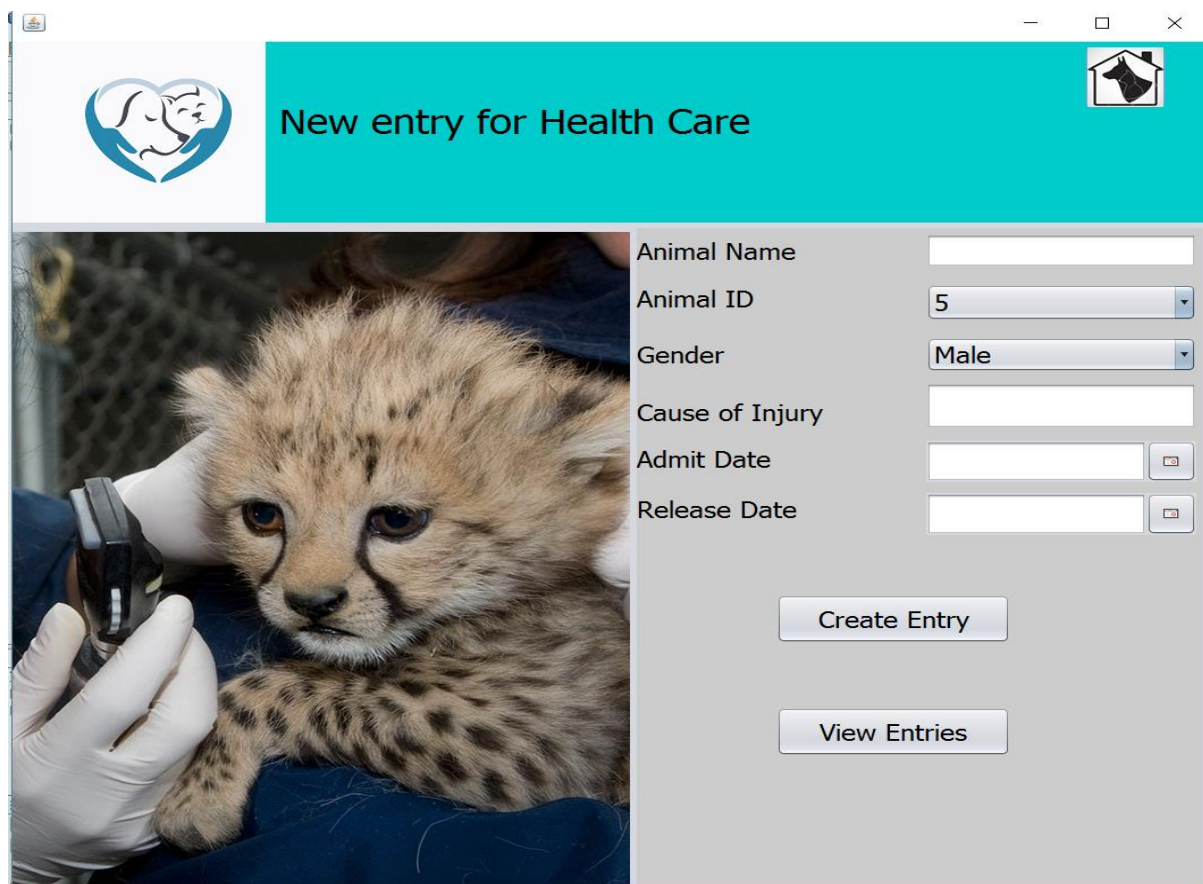
Fig 6.3: Adopter Sign-In.



The Details of animals screen has a white header with a house icon and the text "Categories to choose from:". The main area has a background image of a panda. On the left, there are input fields for "Enter your ID:", "Category" (with a dropdown showing "1001" and a "Selected" button), and "Species". Below these is a "SUBMIT" button. In the center, under "ANIMAL DETAILS", are fields for "ANIMAL NAME:", "ANIMAL", "ANIMAL ID:", and "AMOUNT:". On the right, under "ENTER YOUR DETAILS:", are fields for "Name", "GENDER" (with a dropdown), and "PHONE NO.". At the bottom center is an "ADOPT" button. A "HELP:" sidebar on the right lists categories and species with corresponding IDs.

category	Species	Animal
1001	- 201	- Frogs
1001	- 221	- Newts
1002	- 205	- Birds of Prey
1002	- 206	- Ducks/Geese
1002	- 207	- Songbirds
1002	- 208	- Flightless Birds
1003	- 209	- Clownfish
1003	- 212	- Jawless fish
1004	- 213	- Carnivores
1004	- 214	- Herbivores
1004	- 215	- Omnivores
1004	- 216	- Primates
1005	- 217	- Crocodilia
1005	- 218	- Sphenodontia

Fig 6.4: Details of animals Screen.



New entry for Health Care

Animal Name

Animal ID

Gender

Cause of Injury

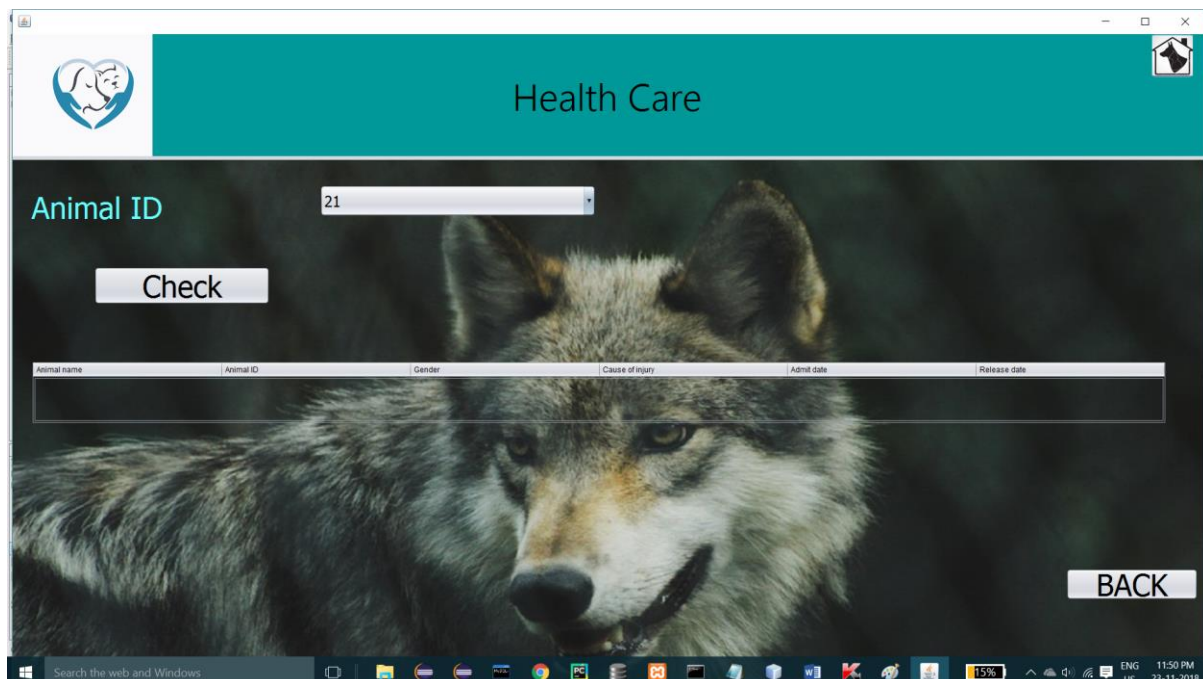
Admit Date

Release Date

Create Entry

View Entries

Fig 6.5: Health Care Screen.



Health Care

Animal ID

Check

Animal name	Animal ID	Gender	Cause of injury	Admit date	Release date
<input type="text"/>					

BACK

Fig 6: HealthCare Check Screen

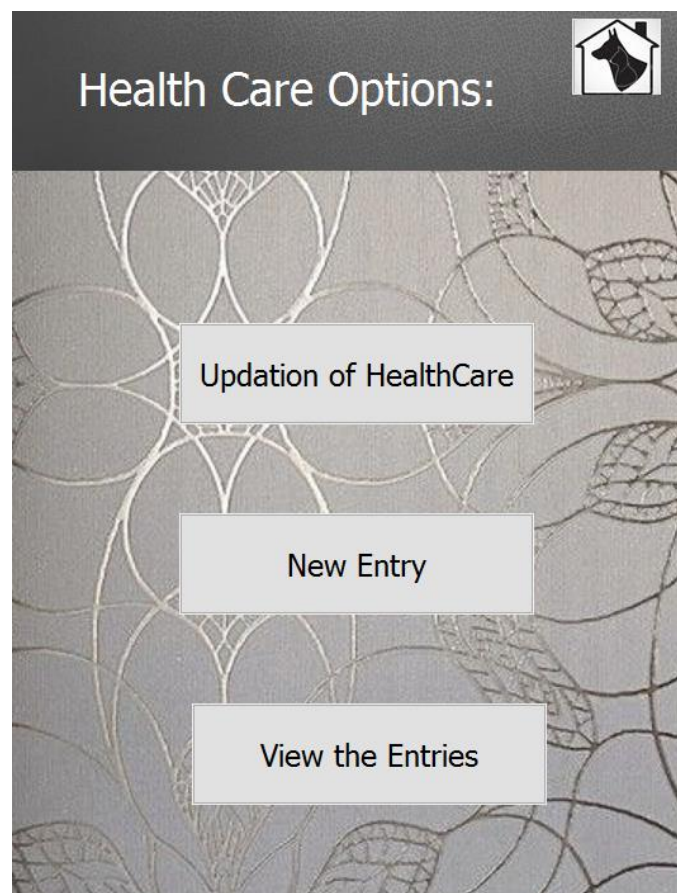


Fig 6.7: HealthCare options Screen.

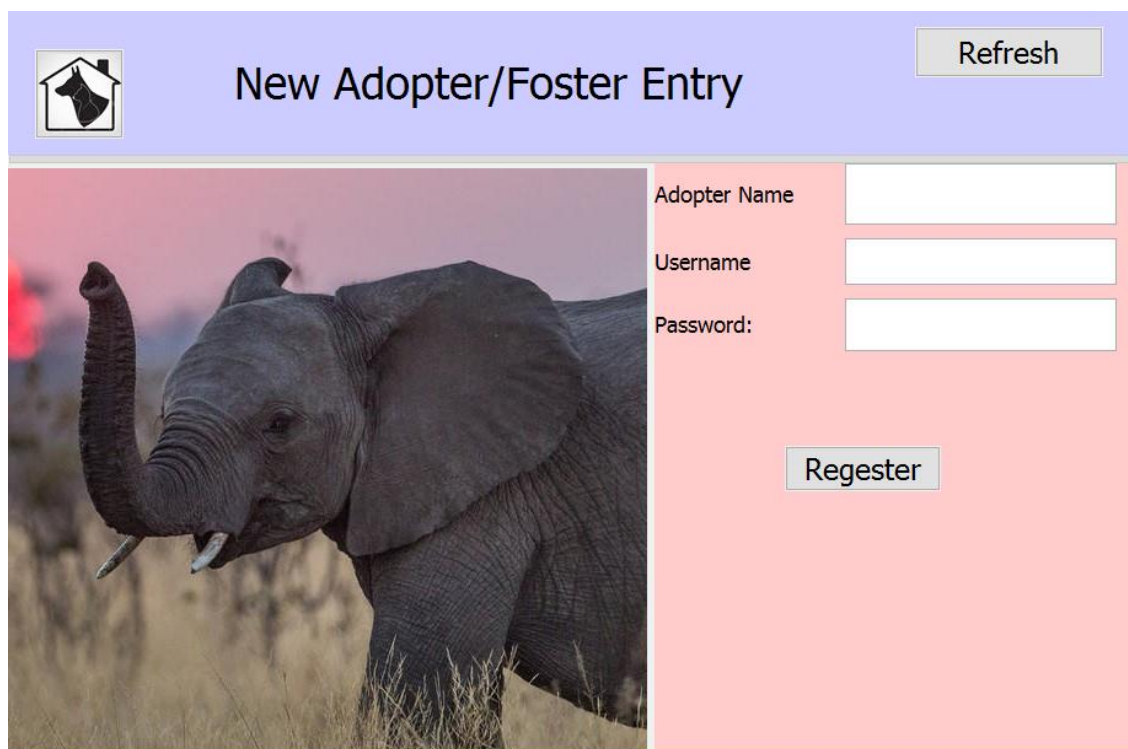
The screenshot displays a registration form titled "New Adopter/Foster Entry" in a light blue header. A "Refresh" button is in the top right corner. On the left, there is a photograph of an elephant. To the right of the photo, there are three input fields labeled "Adopter Name", "Username", and "Password:". Below these fields is a "Register" button. A small house-and-dog icon is located in the top left corner of the form area.

Fig 6.8: Adopter Sign-up Screen



Fig 6.9: Administrator Master Screen.

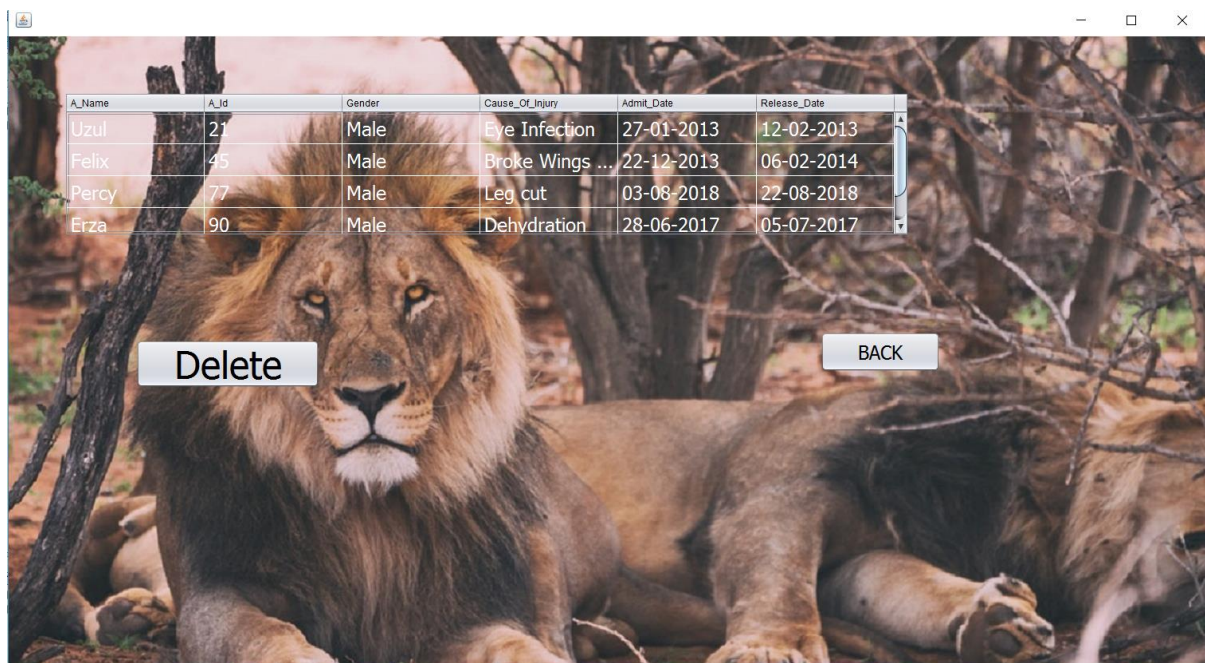
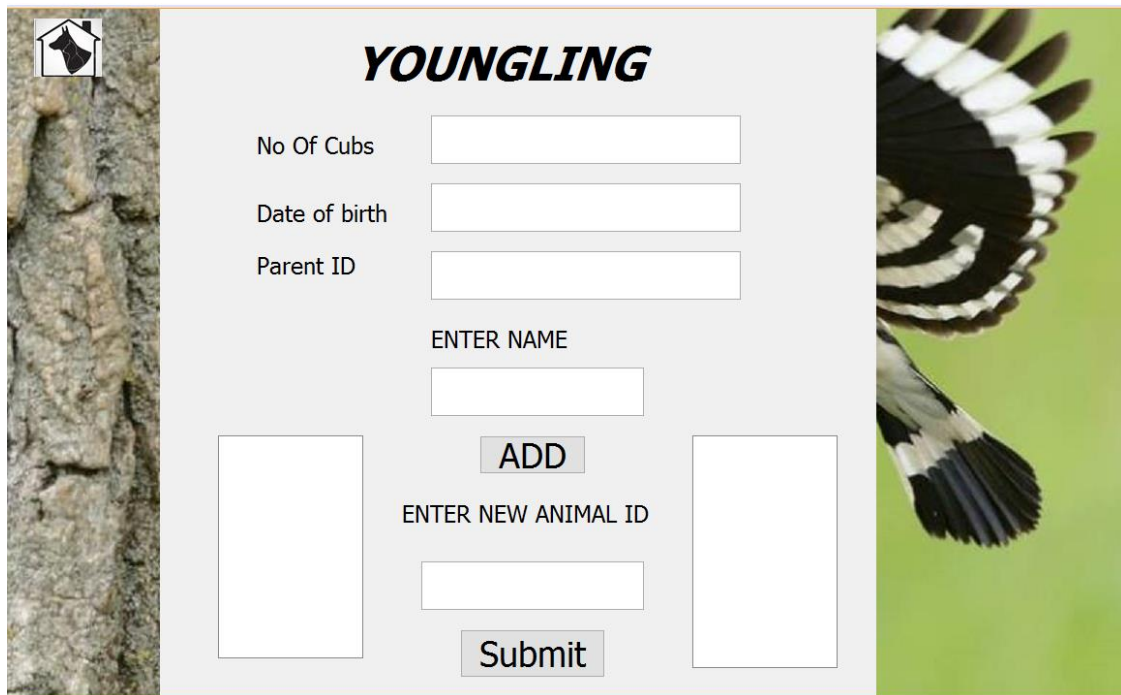


Fig 6.10: HealthCare List Screen



YOUNGLING

No Of Cubs

Date of birth

Parent ID

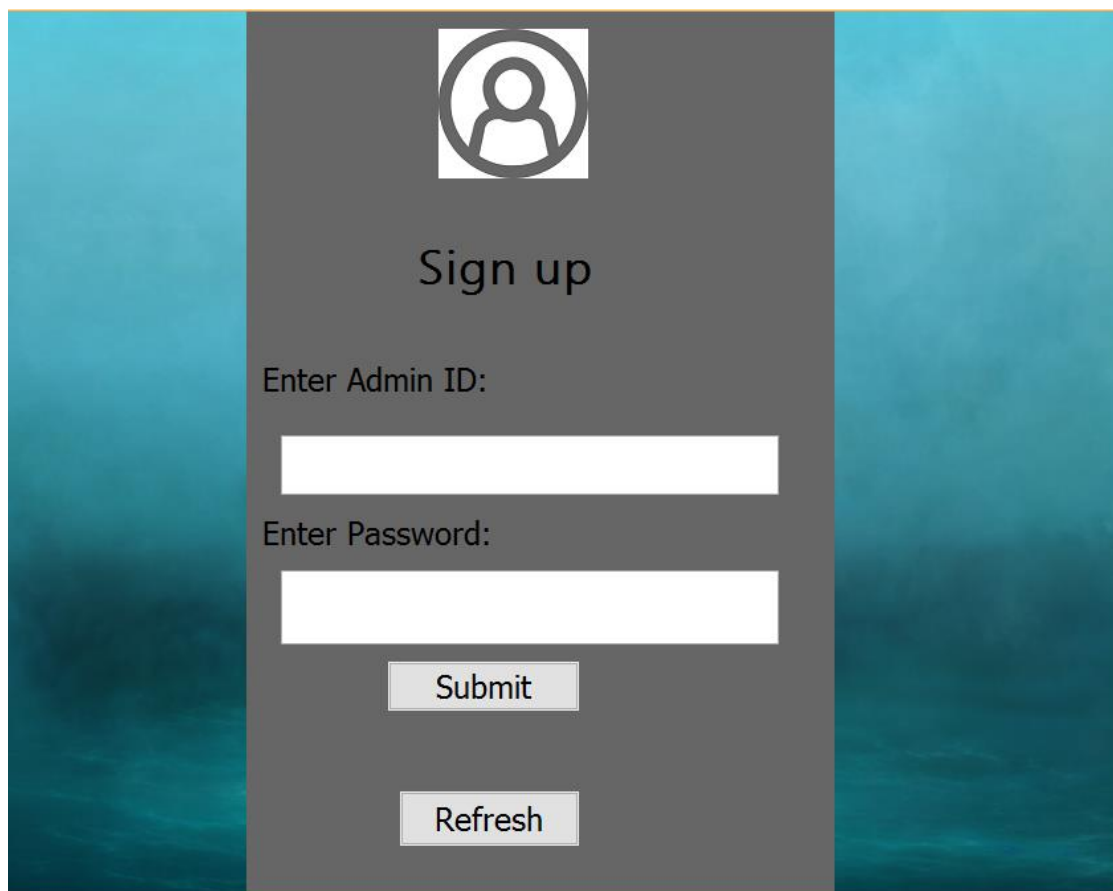
ENTER NAME


ADD

ENTER NEW ANIMAL ID

Submit

Fig 6.11: Adding details of new born Screen.





Sign up

Enter Admin ID:

Enter Password:

Submit

Refresh

Fig 12: Administrator sign-up Screen

Chapter 7

CONCLUSION

This project presents the modelling, analysis and implementation of a Zoo Management System (clone). Especially, this paper also proposed the module of how a basic Zoo Management System works which can assist in designing the extended models. The advantage of the proposed approach is the clear presentation of the Zoo Management system behaviour and readiness for implementation.

FUTURE ENHANCEMENT

This mini project was developed to fulfil user requirement; however there is scope to improve performance of the Zoo Management System in the area of user interface, database performance and query processing time etc.

The future enhancements that are possible in the project are as follows:

- Some enhancement in the code can be made for improved performance.
- Improvement in Graphics and Database handling can be made.
- In future reports can be generated.
- Web Interface for net Zoo Management System Management.

BIBLIOGRAPHY

Books:

- Fundamental of DATABASE SYSTEMS – Ramez Elmasri, Shamkanth B Navathe, 7th Edition, Pearson Education, 2017.
- The Complete Reference JAVA – Herbert Schildt, 7th Edition, Mc Graw Hill Education.
- Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.

Websites & links:

- <http://www.wikipedia.org>
- <http://www.google.com>
- <http://www.stackoverflow.com>