

Project Supervisor

Mr. Samyan Qayyum Wahla

Project Members(Group Id Here)

|  |  |
| --- | --- |
| Muhammad Umer Amjad | 2020-CS-61 |
| Muhammad Ahmad | 2020-CS-74 |

Department of Computer Science

University of Engineering and Technology, Lahore

Pakistan

**Project Description:**

A livestock farm named ‘Duko Farms’ is a farm that buys Cattles feeds them for 2 to 3 months and then sells them in a slaughterhouse besides these the farm also has buffalos in it. The milk of buffalos is sold on daily basics to the customers on their doorsteps. They are to manage the cost of food for animals, the salary of the employees, the cost of medicines, the amount of purchase of animals, and the amount at which milk and animal are sold to the market according to their expenditures and weight of animal respectively. First, the owner buys the animals take them to the farm, and stores the cost of purchase and traveling, date of purchase, and their breed. The weight of the animal is overviewed every two weeks and the respective market cost is updated and the estimated profit is also calculated according to the purchase and sale price. After 2 or 3 months of feeding the Cattles, they are to be taken to sell in the slaughterhouse, and on the other hand, milk is sold each day. The purchase price, feeding/medicine cost, Salaries, expenditures on milk, and traveling cost are used to calculate the earned profit. The profits from both slaughtering and milk-producing animals are calculated separately and then are added to total profit. The administrator(owner) also keeps the record of customers who are receiving milk on daily basis their addresses, contact numbers, and CNIC along with the amount of milk they are receiving are recorded each day.

When the app is started a welcome page is displayed with the heading of ‘Duko Farms’ at the top. The app is designed so that only the owner can enter the application. The owner will have to input his credentials to log in to the application. Moreover, he can also change his username and password anytime he wishes to by entering the credentials along with the new ones. In an unfortunate case of forgetting the password owner can click the option of ‘Forget Password’, this action will send an email to the owner's mailing address. The email will be accommodating a code required to make a new password. the owner will enter the code, new password and then confirm it to change the password. after completing the login process user will be able to see different options of viewing Employees, customers, animals, and milk details. another tab of records will also be available there. the records will be having all the records of expenditures of a particular month and show its profit and further details. The employees' tab will show the details like name, CNIC, contact number, address, designation, and salary. The details can further be updated whenever required and an employee can also be deleted when required. The Customers tab will show the details like name, CNIC, contact number, address, and amount of milk they receive and if or not they have paid the bill of the respective month. The details can further be updated whenever required and a customer can also be deleted when required.

**Project Features:**

You are supposed to develop application which is usable by the owner and the administrator. Every data needed to run the farm is stored and displayed by this application. Following features will be implemented in the application.

* **Salaries Management**

Administrator can add/delete the workers and their salaries can be updated

* **Milk Management**

Administrator will keep the record of milk produced each morning and the cost of purification of milk. The estimated expenditures will be calculated. And based on these expenses and the amount at which milk is sold profit will be calculated.

* **Cattle Management**

Administrator can add/delete the animals. Their weight is to be updated every week. Cattle can be viewed by its name. Its pictures, weight (throughout the time in far), purchase price, estimated selling price and estimated profit on it can be viewed by the owner.

* **Food Management**

Administrator will add the type of food with respect to the flock it is used in along with the price off the Food.

* **Employee Management**

The sessional checkup is managed by the administrator and medicine is added if any used.

* **Vaccination record**

This section will keep the record of those animals which had been diagnosed for a particular disease and had been given vaccine and other medical facilities.

* **Travelling Cost**

Travelling Cost is stored with respect to the flock.

* **Profit Calculator**

Profit will be of two types

1)When a flock is marked sold, profit on it is calculated considering cost, selling and purchasing price. The profit is stored to keep the record of how much profit is earned throughout.

2)The amount of milk sold in one month.

**Reporting:**

Owner also requires multiple reports in pdf form that may include.

* **List of Cattles**

List of Cattles include picture, Name ID, Flock name, initial weight, Chang in weight every week, initial price, estimated selling price and profit.

* **List of Employees**

List of Workers include workers CNIC, their salary.

* **List of customers**

List of customers will keep the record of their names, addresses, contact numbers, amount of milk received by each individuals and bill headed towards them.

* **List of Slaughtering Animals**

List of Flocks include sold and stored flocks. If sold then Profit on it. If Stored than Cost till then and Estimated profit..

* **List of Monthly Records**

This List will be having all the records of expenditures of a particular month and show its profit and further details like Amount of milk sold, sale price of animals, food and vaccination cost, salaries of employees

**Technology Stack:**

|  |  |
| --- | --- |
| Language(C#/Java) |  |
| Platform(Web/Desktop) |  |
| Frontend Technology(Simple HTML/Bootstrap/any other library) |  |
| IDEs | List the IDEs here that you will use for development |

**Project Actors:**

In this sections, write the name of actors and brief description, who will be using the system.

**Use Cases:**

All the use cases should be written in the following format. The name of use should be start with a word e.g. Add Student. It should not be as Student/ Student Add. Add Student is separate use case and Edit Student is separate use case:

**Use Case 1(Write the name of the use case):**

|  |  |
| --- | --- |
| Use Case ID | U01 |
| Name |  |
| Actor |  |
| Description | Brief description of the use case. With at least 50 words. |
| Layout in pencil tool |  |
| Validators | Add here the name of validators that will be applied on this page |

**User Interface Details**

In this section, fill the table for summary that which use case will have the required component. Inside each box, write the counts for each component. If component is not used, write zero.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case Id | TextBox | DropDown | Password Box | Table | Date Field | Buttons | AutoComplete | Radio Button | CheckBox | Menu | Text Area | ProgressBar |
| U01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| U02 |  |  |  |  |  |  |  |  |  |  |  |  |

**Classes:**

In this section, we do not require detailed design diagram. But identify the tentative classes with the requirement that you should have at least 10 domain classes and 5 software classes, 3 abstract classes, 2 singelton classes. Fill the following table for details. Note that class name should follow naming conventions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class Name** | **Software/ Domain** | **Is Abstract (Yes/No)** | **Is Singleton (Yes/No)** | **Is the class will has parametrized constructor(Yes/No)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Object Oriented Features:**

**Composition:**

In this section, Identify the at least three solid examples where composition can be perform. Add UML diagram of each example.

**Inheritance:**

At least 2 examples where inheritance will take place with UML diagrams.

**Multiple Inheritance:**

Two example with UML diagrams of multiple inheritance.

**Multi-Level Inheritance:**

Two examples with UML diagrams of multi-level inheritance.

**Polymorphism:**

At least 3 examples with UML diagram for polymorphism.

**Collections:**

In this section, describe how and where you will use the following collections and why you are forced to use these collections.

ArrayList, LinkedList, Queue, Stack, HashSet and TreeSet

**Exceptions:**

In this sections, identify at high level which type of exceptions you can face in your code and what are the solutions. Add more rows in the table as per requirements.

|  |  |  |
| --- | --- | --- |
| Type of Exception | Why this exception will occur | How you will handle the exception |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Data Storage:**

In this sections, describe the five files with their format from where you will read or store data. In case of database table, write down the names of columns for tables. At least 5 files/ tables are required.

**Email Sending:**

In this section, describe the points where you will be required to send the email from the code. Additionally, write down the sample subject and email content.

**Project Plan**

This section should include the implementation plan and work division among the members. All the estimated dates should be before June 15, 2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Id** | **Use Case Name** | **Member Name** | **Estimated Completion Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |