CHAPTER 1

INTRODUCTION

1.1 Project overview

Dairy management involves keeping good records. Once a government realizes that milk recording should complement breeding programmers, farmers must still be convinced that they have something to gain from keeping records. Here are ways in which milk recording systems in developing countries can be adapted to the needs of farmers and governments.

Dairy management system is a software application to maintain day to day transaction in a milk distributor office. It is a pilot project for new milk distributor to be start soon in the city .the management planned this milk distribution center to operate on the next month. They have a big plan to collect the milks from many different sources and distribute the same for the milk buyers to manage all these they required full-fledged software which will take care all these.

Good dairy management practice also ensures that the milk is produced by healthy animals in a manner that is sustainable and responsible from the animal welfare, social, economic and environmental perspectives. So implementing good dairy farming practice is good risk management for the short and long term future of the dairy farming enterprise. This Guide encourages dairy farmers to adopt 'proactive' preventative practices rather than waiting for problems to occur. This means if the farmers are involved in the dairy management system then the health and welfare of the animals are also preserved.

1.2 Objectives of Proposed System

Milk recording has two main objectives:

1. To help the individual farmer to produce milk more efficiently, i.e. more economically.
2. To provide data for government administrative, research, breeding and extension purposes.

Both objectives are of course equally important, but it should be recognized that if the individual farmer cannot be convinced that he/she has something to gain from the practice, it is very difficult to develop extensive recording systems. One should therefore avoid emphasizing the over-all national benefits too much, because otherwise many farmers will believe that milk recording serves only the interests of various official bodies.

The starting point in all recording schemes should be the farmer, how to get him/her to see that it really pays to keep some simple records. If one succeeds in this, the collection of data, and cooperation in other matters, will be much easier.

In countries like INDIA farmers are the back of bone of the country, these farmers chooses dairy farming other than the field work.so bringing the justice for the farmers and makes the office work easy we are bringing this system into act.

1.3 Database management system (DBMS)

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified – and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks are supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

The DBMS perhaps most useful for providing a centralized view of data that can be accessed by multiple users , for multiple locations , in a controlled manner. End users can software programs are free from having to understand where the data is physically located or on what type of storage media it resides because the DBMS handles all requests.

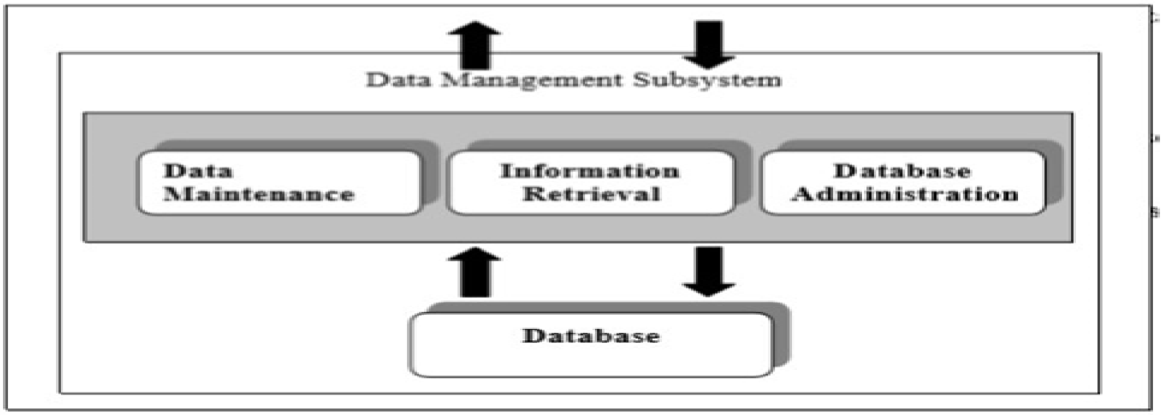


Fig 1.1 Facilities of a DBMS

1.4 Structured Query Language (SQL)

SQL was one of the first commercial Languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, 'A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relation model as described by Codd, it becomes the most widely used database language.

SQL become a standard of the American National Standard Institute (ANSI) in 1986, And of the International Organization for Standardization(ISO) in 1987.Since then, the standard has been revised to include a large set of features. Despite the existing of such standards, most SQL code is not completely portable among different database systems without adjustments.

1.5 Front end development using PHP

PHP is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document.

PHP source code is compiled on-the-fly to an internal format that can be executed by the PHP engine. In order to speed up execution time and not have to compile the PHP source code every time the webpage is accessed, PHP scripts can also be deployed in executable format using a PHP compiler.

PHP is one of the most popular server side scripting languages running today. It is used for creating dynamic webpages that interact with the user offering customized information. PHP offers many advantages; it is fast, stable, secure, easy to use and open source (free).

PHP code is inserted directly into the HTML that makes up a website. When a visitor comes to the website, the code is executed. Because PHP is a server side technology, the user does not need any special browser or plug-ins to see the PHP in action.

Another key advantage of PHP is its connective abilities. PHP uses a modular system of extensions to interface with a variety of libraries such as graphics, XML, encryption, etc. In addition, programmers can extend PHP by writing their own extensions and compiling them into the executable or they can create their own executable and load it using PHP's dynamic loading mechanism.

A huge advantage that PHP offers is its community. Since PHP is an A huge advantage that PHP offers is its community. Since PHP is an open source project, the PHP community is willing to share. If you're looking for a particular script, chances are another user has already created something similar. Check within the PHP community for availability. Likewise, if you have created a function that others might enjoy, be sure to post the code for others.

A PHP scripting block always starts with <?php and ends with ?>. A PHP scripting block can be placed anywhere in the document.

On servers with shorthand support enabled you can start a scripting block with <? and end with ?>.For maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

1.6 Three Schema Architecture

Following are the three levels of database architecture:

- Physical Level
- Conceptual Level
- External Level

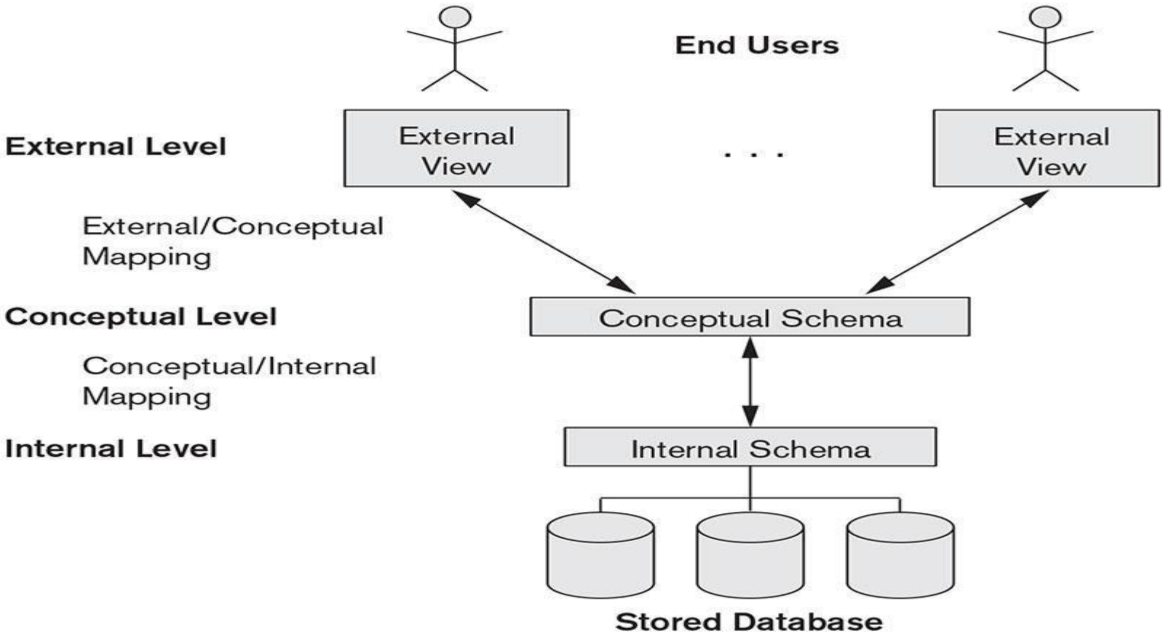


Fig 1.2 The Three Schema Architecture

In the above diagram,

- It shows the architecture of DBMS.
- Mapping is the process of transforming request response between various database levels of architecture.
- Mapping is not good for small database, because it takes more times.
- In External/Conceptual mapping, DBMS transforms a request on an external schema against the conceptual schema.

- In Conceptual/Internal mapping, it is necessary to transform the request form the conceptual to internal levels.

1. **Physical Level:**

- Physical level describes the physical storage structure of data in database.
- It is also known as Internal Level.
- This level is very close to physical storage of data.
- At lowest level, it is stored in the form of bits with the physical addresses on the secondary storage device.
- At highest level, it can be viewed in the form of files.
- The internal schema defines the various stored data types. It uses a physical data model.

2. **Conceptual Level:**

- Conceptual level describes the structure of the whole database for a group of users.
- It is also a known as the data model.
- Conceptual schema is a representation of the entire content of the database.
- This schema contains all the information to build relevant external records.
- It hides the internal details of physical storage.

3. **External Level:**

- External level is related to the data which is viewed by individual end users.
- This level includes a no. of user views or external schemas.
- This level is closest to the user.
- External view describes the segment of the database that is required for a particular user group and hides the rest of the database from that user group.

1.7 NORMALIZATION:

Here are the most commonly used normal forms:

1). First normal form (1NF):

- As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

2). Second normal form (2NF):

- A table is said to be in 2NF if both the following conditions hold:
- Table is in 1NF (First normal form)

- No non-prime attribute is dependent on the proper subset of any candidate key of table.
- An attribute that is no part of any candidate key is known as non-prime attribute.

3). Third normal form (3NF):

- A table design is said to be in 3NF if both the following conditions hold:
- Table must be in 2NF
- Transitive functional dependency of non-prime attribute on any super key should be removed
- An attribute that is not part of any candidate key is known as non-prime attribute.
- In other words 3NF can be explained like this : A table is in 3NF if it is in 2NF and for each functional dependency X->Y at least one of the following condition hold:
- X is a super key of table
- Y is a prime attribute of table
- An attribute that is a part of one of the candidate keys is known as prime attribute.

Boyce and Codd normal form (BCNF):

It is an advance version of 3NF that’s why it is also referred as 3.5 NF. BCNF is stricter than 3NF. A table compiles with BCNF if it is in 3NF and for every functional dependency X->Y, X should be the super key of the table

4). Fourth Normal Form(4NF): is a normal form used in normalization. Introduced by Ronald Fagin in 1997,4NF is the next level of normalization after Boycee-Codd normal form (BCNF).Whereas the second , third and Boyce-Codd normal forms are concerned with functional dependencies ,4NF is concerned with a more general type of dependency known as multivalued dependency. A tables is in 4NF if and only if ,for every one of its non-trivial multivalued dependies X,Y ,X is a super-key that is ,X is either a candidate key or a super set.

5). Fifth Normal Form(5NF) : A database is said to be decomposes in 5th Normal Form ,if and only if, it is in 4th normal form and if the table further to eliminate redundancy and anomaly and when we joins the decomposed tables by means of candidate keys we should not be losing the original data or any new record set should not arise. In simple words joining two or more decomposed tables should not lose records or new records

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Requirements Related To Project

i) User Requirements

Since end users are the ones who are finally going to use the system, their requirements need to be identified. This involves questioning the end users what their expectations were. The main requirement of the end user is that the system should be easy to use and take less time. In addition to these another important factor was to eliminate the need for database conversion and migration that had to be carried out presently. After conducting interviews with the users a document called the software requirement specification was created. This is the most important document that forms the basis for system development. It should be consistent, complete, unambiguous, traceable and inter-related. This document has the following components:

ii) Functional Requirements:

The functional requirements specify relationship between the inputs and outputs. All the operations to be performed on the input data to obtain output are to be specified. This includes specifying the validity checks on the input and output data, parameters affected by the operations and the other operations, which must be used to transform the inputs into outputs. Functional requirements specify the behaviour of the system for valid input and outputs.

iii) Performance Requirements

This section includes performance of the product that are set by user interaction and studying the existing system of the organization. These are stated in complete measurable terms, so that they can be verified during system evaluation phase.

2.2 Hardware Requirement

Processor : Intel Core Duo 2.0 GHz

RAM : 1 GB or More

Hard disk : 80GB or more

Monitor : 15” CRT, or LCD monitor

Keyboard : Normal or Multimedia

Mouse : Compatible mouse

2.3 Software Requirement

Front End : HTML with CSS and PHP

Back End : My SQL Server

Operation System : Windows 7 with server pack 2 Or Windows 8.1

2.4 User Characteristics

i) Every user

Should be comfortable with basic working of the computer Must carry a login ID and password used for authentication in dairy milk management manager, supervisor and clerk are the employees. These characters only are allowed to authorize to login.

ii) Constraints

The GUI restricted to English Login user mail and password is used for identification of users. There is no facility for a guest login.

CHAPTER 3

SYSTEM DESIGN AND ANALYSIS

3.1 Project Description

System design is essential to develop a model of system before writing any software that is used to control the system or to interact with it during the design process we try to develop system at different levels of abstraction. Design process involves data structures including library function used in the programs. The project is developed using the below objects:-

Planned approach toward working: The working in the organization will be well planned and organized. The data will be stored efficiency with optimal disk space consumption in data stores which will help in retrieval of information as well as its storage under resource constraints.

Accuracy: The level of accuracy in the proposed system will be higher. All operations would conform to integrity constraints and correctness and it will be ensured that whatever information is received at or sent from the center is accurate.

Reliability: The reliability of the proposed system will be high due to the above mentioned reasons. This comes from the fact that only the data which conforms accuracy clause would be allowed to commit back to the disk. Other properties like transaction management and rollback during system or power failure etc. get automatically taken care of by the SQL systems, which is undoubtedly an excellent choice of the DBMS system. Properties of atomicity, consistency, isolation and data security are intrinsically maintained.

No redundancy: In the proposed system it will be ensured that no repetition of information occurs; neither on a physical storage nor on a logical implementation level. This economizes on resource utilization in terms of storage space. Also even in case of concurrent access no anomalies occur and consistency is maintained. In addition to all this, principles of normalization have been endeavored to be followed.

Immediate retrieval of information: The main objective of the proposed system is to provide a quick and efficient platform for retrieval of information. Among the queries allowed for use by sthe user, the query results are made available immediately, without time lapse, irrespective of the complexity of the query.

Ease of operation: The system should be simplistic in design and use. It is such that it can be easily developed within a short period of time and can conform to the financial and resource-related constraints of the organization.

Step 1: Mapping of Regular Entity Types

For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E. Include only the simple component attributes of E as the primary key for R. If the chosen key of E is a composite, then the set of simple attributes that form it will together form the primary key of R.

Step 2: Mapping of Weak Entity Types

For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes(or simple components of composite attributes) of was attributes of R. In addition , include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s). The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

Step 3: Mapping of Binary 1:1 Relationship Type

For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.

Step 4: Mapping of Binary 1: N Relationship Types

For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T that represent the other entity type participating in R. Include any simple attributes of the 1:N relationship type as attributes of S.

Step 5: Mapping of Binary M: N Relationship Types

For each regular binary M:N relationship type R, create a new relation S to represent R. Include as foreign key attribute in S the primary key of the relations that represent the participating entity type; their combination will form the primary key of S. Also Include any simple attributes of the M:N relationship p type as attributes of S.

Step 6: Mapping of Multi valued attributes

For each multi valued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R- of the relation that represents the entity type or relationship type that has A as an attribute. The primary key of R is the combination of A and K. If the multivalued attribute is a composite, we include its simple component.

Step 7: Mapping of N-ary Relationship Types

For each n-ary relationship type R, where n>2, create a new relation S to represent R. Include as foreign key attribute in S the primary key of the relations that represent the participating entity types. So Include any simple attribute of the n-ary relationship type as attribute of S.

3.2 Tables Used

LOGIN
(usermail, username, password)

FARMER
(f_no, f_id, f_name, f_locallity, f_acc, last_paid, f_phone)

EMPLOYEES
(e_id, e_name, e_mail, username, e_pass,e_roll, e_payroll_no)

DELIVERY
(d_id, r_f_no, r_liter, r_dt, r_received_by, r_deliverer)

PAYMENT
(id, p_to, p_date, p_ac, p_method, p_transaction_code, p_transacted_by)

SETTINGS_RATES
(id, from, to, rate)

3.3 Description of Tables

1. **Login page**
   - Login into the account
   - The user should be either Manager, Supervisor or Clerk to access the database.
2. **Farmers**
   - Add farmers into the database and their information.
   - View, Update, and remove farmers information.
3. **Deliveries**
   - Add Deliveries into the database and their information.
   - View, update, and remove delivery information.
4. **Employees**
   - Add employee into the database and their information.
   - This table is only accessed by the Manager neither by Supervisor nor Clerk.
5. **Payments**
   - The record of the Farmer is opened here.
   - The payment for the Farmer and deliverer is processed.
6. **Setting and rating**
   - The cost milk per liter is fixed here.
   - Fixed cost is going to updated and follows the same for payment.

3.4 ER Diagram

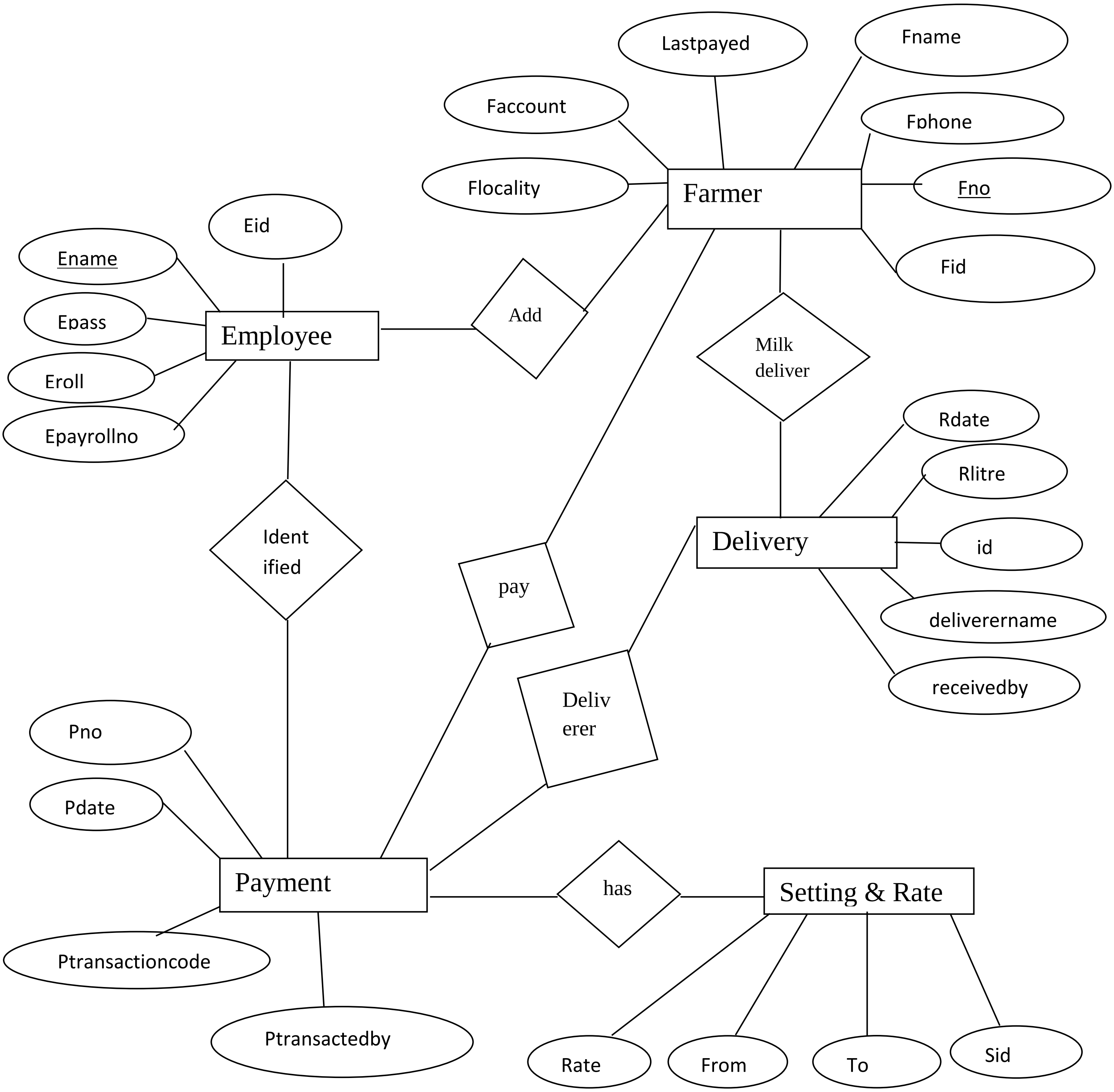
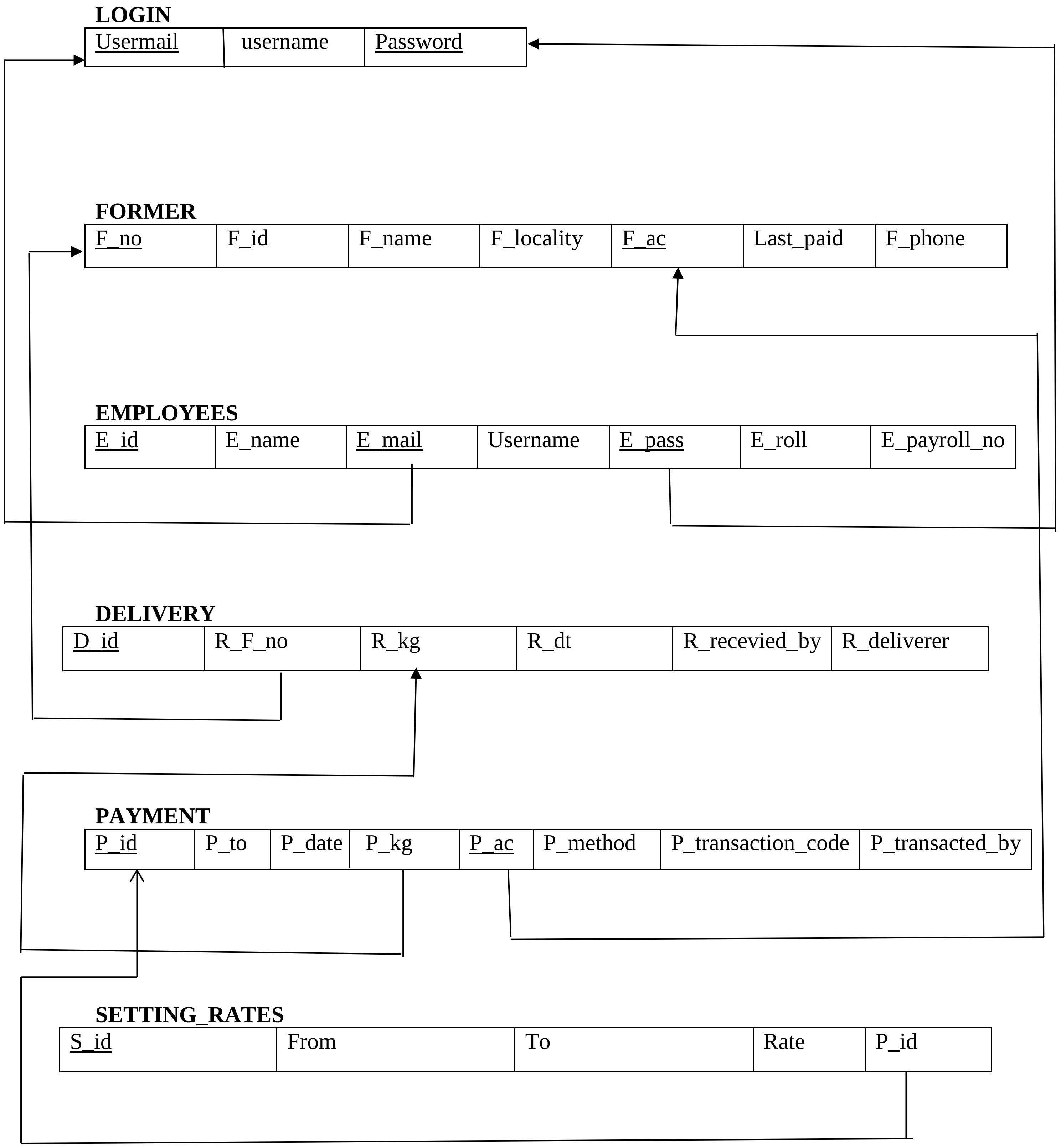


Fig 3.1 ER Diagram of dairy Management System

3.5 Schema Diagram



LOGIN

Usermail	username	Password

FORMER

F_no	F_id	F_name	F_locality	F_ac	Last_paid	F_phone

EMPLOYEES

E_id	E_name	E_mail	Username	E_pass	E_roll	E_payroll_no

DELIVERY

D_id	R_F_no	R_kg	R_dt	R_recevied_by	R_deliverer

PAYMENT

P_id	P_to	P_date	P_kg	P_ac	P_method	P_transaction_code	P_transacted_by

SETTING_RATES

S_id	From	To	Rate	P_id

Fig 3.2 Schema Diagram of Dairy Management System

CHAPTER 4

IMPLEMENTATION

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes.

This phase continues until the system is operating in production in accordance with the defined user requirements.

4.1 CREATE TABLE

- **EMPLOYEES**

```
CREATE TABLE `employees` (`id` int(11) NOT NULL,`e_name` varchar(50) NOT NULL,
`e_mail` varchar(50) DEFAULT NULL,`username` varchar(50) NOT NULL,`e_pass`
varchar(50) NOT NULL,`e_role` varchar(50) DEFAULT NULL,`e_payroll_no` varchar(50)
NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- **FARMERS**

```
CREATE TABLE `farmers` (`f_no` varchar(50) NOT NULL,`f_id` text NOT NULL,
  `f_name` varchar(50) NOT NULL, `f_locallity` varchar(50) DEFAULT NULL,`f_ac`
varchar(50) DEFAULT NULL,`last_paid` date DEFAULT NULL, `f_phone` varchar(20)
DEFAULT NULL);
```

- **SETTINGS_RATES**

```
CREATE TABLE `settings_rates` (`id` int(11) NOT NULL, `from` date NOT NULL,`to`
date NOT NULL,`rate` float NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- **DELIVERY**

```
CREATE TABLE `delivery` ( `id` int(11) NOT NULL, `r_f_no` varchar(50) NOT
NULL,`r_kg` float NOT NULL,`r_dt` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP, `r_received_by` varchar(50) NOT NULL,`r_deliverer`
varchar(50) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- **PAYMENT**

```
CREATE TABLE `payment` ( `id` int(11) NOT NULL,`p_to` varchar(50) NOT NULL,
`p_date` date NOT NULL,`p_ac` bigint(20) NOT NULL,`p_method` varchar(30) NOT
NULL,`p_transaction_code` int(11) NOT NULL, `p_transacted_by` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.2 INSERTING VALUES

```
INSERT INTO ‘login’ (‘userid’,’usermail’,’password’,’userlevel’) VALUES (‘1’,
’nayan@gmail.com’ , ‘nayan123’ , ‘1’);
```

```
INSERT INTO `employees` (`id`, `e_name`, `e_mail`, `username`, `e_pass`, `e_role`,
`e_payroll_no`)VALUES(3,'nayan','nayan@gmail.com','','827ccb0eea8a706c4c34a16891f84e
7b', 'Manager', '3456'),
```

```
INSERT INTO `farmers` (`f_no`, `f_id`, `f_name`, `f_locallity`, `f_ac`, `last_paid`,
`f_phone`, `f_photo`) VALUES ('1', '23456779', 'alexandar jones', 'kk', '9890485987', '2017-
04-30', '0721274242', NULL);
```

```
INSERT INTO `delivery` (`id`, `r_f_no`, `r_kg`, `r_dt`, `r_received_by`, `r_deliverer`)
VALUES (2, '49', 66, '2017-04-07 23:00:00', '', 'kumar');
```

```
INSERT INTO ‘payment’ ( ‘id’ , ‘p_to’, ‘p_date’ , ‘p_ac’, ‘p_method’ , ‘p_transaction_code’
, ‘p_transacted_by’) VALUES (21,’2017-11-31’, ‘’2017-12-31’, ‘ 0721274242’ , ‘cheuue’ ,
‘62468264823’);
```

```
INSERT INTO `settings_rates` (`id`, `from`, `to`, `rate`) VALUES
(4, '2017-01-01', '2017-01-31', 20);
```

4.3 SOURCE CODE

Connecting To Database

```
<?php
define('db_host', 'localhost');
define('db_user', 'root');
define('db_password', '');
define('db_database', 'dairy');
$conn= mysqli_connect(db_host, db_user, db_password, db_database);
mysqli_select_db($conn, db_database);
?>
```

Login Page

```
<?php
if (!defined('PAGE_URL'))define ('PAGE_URL', 'http://localhost/Dairy/');
include("auth.php");
$log = new logmein();
$log->encrypt = false; //set encryption
if($_REQUEST['action'] == "login"){
    $hashed_pass=  md5($_REQUEST['password']);
    if($log->login("logon", $_REQUEST['username'], $hashed_pass) == true){
        //do something on successful login
        header("location:".PAGE_URL);
    }else{
        //do something on FAILED login
      echo "wrong";
      echo "action: ". $_REQUEST['action'] ." , username: ". $_REQUEST['username']. "
password: ".$_REQUEST['password'] ." Hashed: ".$hashed_pass;
      // header("location:".PAGE_URL);
        $log->loginform("login", "loginform", PAGE_URL."auth/login.php");
    }
```

Code for insert:

```
<h1>Add Farmers</h1>

<?php
$validation = array('valid' => true, 'nulls' => ", 'id' => ", 'no' => ");
if (isset($_POST['f_no'])) {
//    foreach ($_POST AS $key => $value) {
//        $_POST[$key] = mysqli_real_escape_string($conn, $value);
//    }
    $validation = validate_farmers($_POST['f_no'], $_POST['f_id'], $_POST['f_name'],
$_POST['f_locallity'], $_POST['f_ac'], $_POST['f_phone'],$conn);
    if ($validation['valid'] == TRUE) {
        $sql = "INSERT INTO `farmers` ( `f_no` ,`f_id` , `f_name` , `f_locallity` ,  `f_ac` ,
`f_phone`  ) VALUES(  '{$_POST['f_no']}' ,'{$_POST['f_id']}' , '{$_POST['f_name']}' ,
'{$_POST['f_locallity']}' ,  '{$_POST['f_ac']}' ,  '{$_POST['f_phone']}'  ) ";
```

```
        mysqli_query($conn,$sql) or die(mysqli_error($conn));
        echo "Farmer Added.<br />";

    } else {
        echo $validation['nulls'];
    }
}
?>
```

Code for Delete:

```
<?php
if ($current_user['role'] != 'Manager') {
    echo "sorry you are not allowed to access this module";
    exit();
}
$e_payroll_no = $_GET['e_payroll_no'];
mysqli_query($conn,"DELETE FROM `employees` WHERE `e_payroll_no` =
'$e_payroll_no' ");
echo (mysqli_affected_rows($conn)) ? "Employee deleted.<br /> " : "Nothing deleted.<br />
";
?>
```

Code for Update:

```
if (isset($_GET['e_payroll_no'])) {
    $e_payroll_no = $_GET['e_payroll_no'];
    if (isset($_POST['submitted'])) {
        foreach ($_POST AS $key => $value) {
            $_POST[$key] = mysqli_real_escape_string($conn, $value);
        }
        $hashed_pass=  md5($_POST['e_pass']);
        $sql = "UPDATE `employees` SET  `e_name` =  '{$_POST['e_name']}' ,  `e_mail` =
'{$_POST['e_mail']}' ,  `e_pass` =  '{$hashed_pass}' ,  `e_role` =  '{$_POST['e_role']}' ,
`e_payroll_no` =  '{$_POST['e_payroll_no']}'   WHERE `e_payroll_no` = '$e_payroll_no' ";
        mysqli_query($conn,$sql) or die(mysqli_error($conn));
        echo (mysqli_affected_rows($conn)) ? "Changes Saved.<br />" : "Nothing changed. <br
/>";
        echo "<a href='index.php'>Back To Employees</a>";
    }
    $result = mysqli_query($conn,"SELECT * FROM `employees` WHERE `e_payroll_no`
='$e_payroll_no'");
    $row = mysqli_fetch_array($result);
    include 'form.php';
}
?
```

CHAPTER 5

TESTING

5.1 INTRODUCTION:

The purpose of testing is to discover errors. Testing is the process of trying to every convincible fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

5.2 LEVELS OF TESTING:

5.2.1 UNIT TESTING:

Unity testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures and Operating procedures .For unit testing first we adopted the code testing strategy, which examined the logic of program

5.2.2 USER ACCEPTANNCE TESTING:

User acceptance testing of the system is the key factor for the success of the system. A system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system at the time of development and making change whenever required. This is done with regard to the input screen design and output screen design.

5.2.3 GUI Testing:

GUI testing is use to ensure the visual clarity of the system, flexibility of the system, user friendliness of the system. The various components which are to be tested are: i. Relative layout ii. Various Link and Buttons.

5.2.4 Validation testing:

At the culmination of black box testing, software is completely assembled is a package. Interfacing errors have uncovered and the correct and final states of tests i.e. validation is defined with a simple definition that validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

5.2.5 Output Testing:

After performing validation testing, the next step is output testing of the proposed system. Since the system cannot be useful if it does not procedure the required output. Asking the user about the user about its required format in which the system is required tests the output displayed or generated by the system under consideration.

TEST CASES

Case Id	Test Case	Test Condition	Excepted Output	Actual Output	Pass / Fail
1	Validation Test Case	Required Field Validation	Mandatory field should not be blank	You have to enter value in mandatory Field	Pass
		Regular Expression	A predefine format should be follow	Check proper format	Pass
		Compare Validation	Check with predefine control	Compare with control	Pass
2	Login on	Username	Username format Must be Input	Username format must be input.	Pass
		Confirmation Password	Password & confirm password match	Password & Confirm password match.	Pass

		Text field	All information must be input	All information must be input	Pass

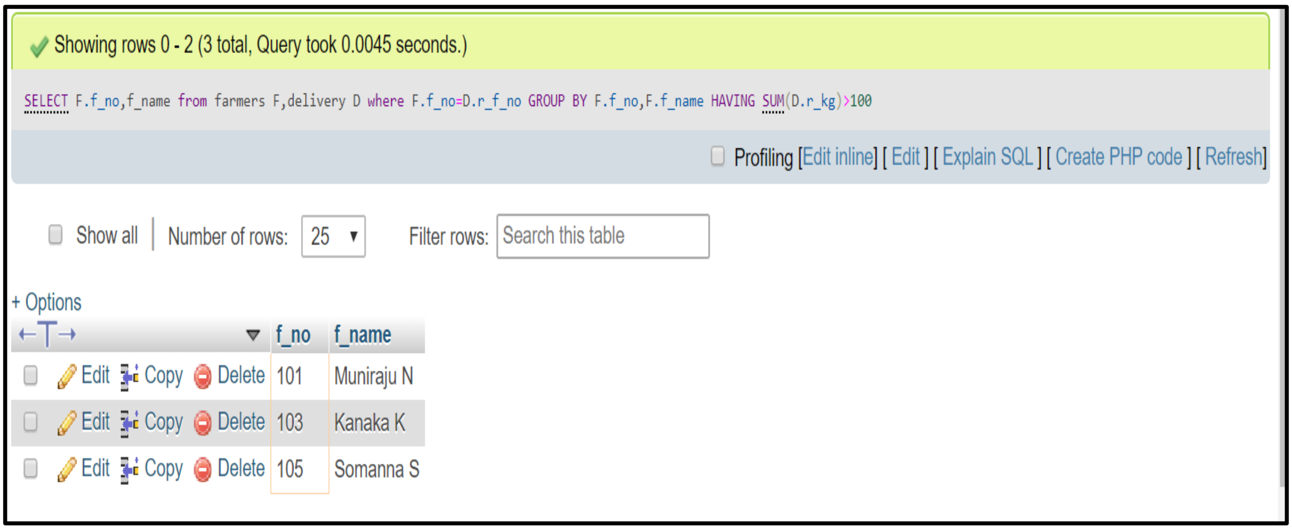
CHAPTER 6

RESULT

A functional Dairy management system was developed using PHP and MYSQL as the side-server and HTML as the client-side. The database used in the Dairy Management System was designed with MYSQL Database Management System (DBMS). All information pertaining to the Result Management System are stored in the system database. The sample output/results are shown in various snapshots.

6.1 Quires:

1. SELECT f_no , f_name FROM farmers , delivery WHERE f_no=r_f_no GROUP BY f_no,f_nameHAVING SUM (r_kg)>100 ;



✓ Showing rows 0 - 2 (3 total, Query took 0.0045 seconds.)

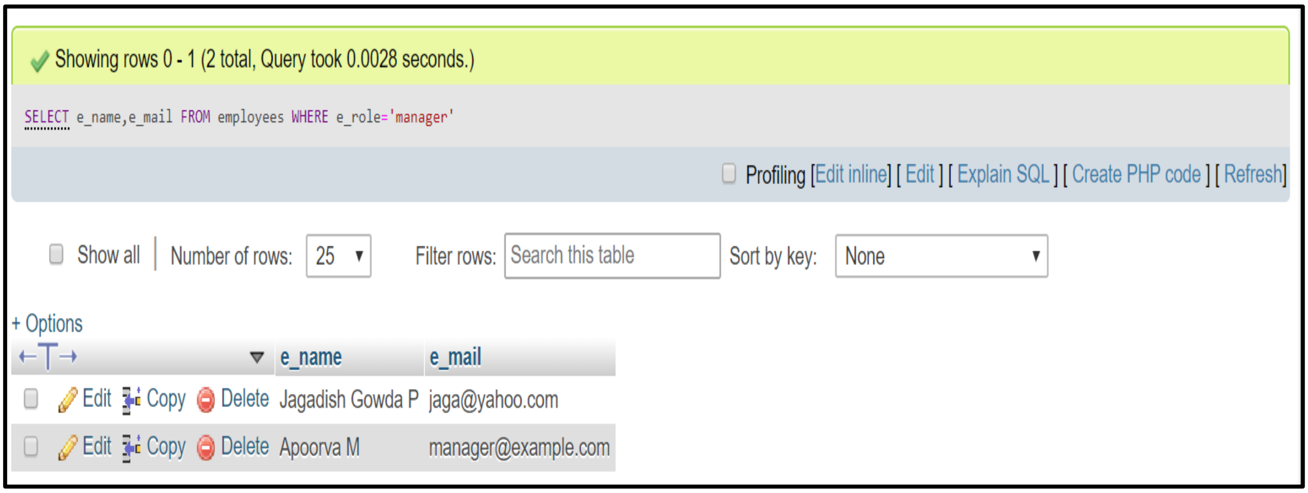
```
SELECT F.f_no,f_name from farmers F,delivery D where F.f_no=D.r_f_no GROUP BY F.f_no,F.f_name HAVING SUM(D.r_kg)>100
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 ▾ Filter rows: Search this table
+ Options
←T→ ▾

				f_no	f_name
□	Edit	Copy	Delete	101	Muniraju N
□	Edit	Copy	Delete	103	Kanaka K
□	Edit	Copy	Delete	105	Somanna S

Figure 6.1 Query 1

2. SELECT e_name,e_mail FROM employees WHERE e_role = 'manager'



✓ Showing rows 0 - 1 (2 total, Query took 0.0028 seconds.)

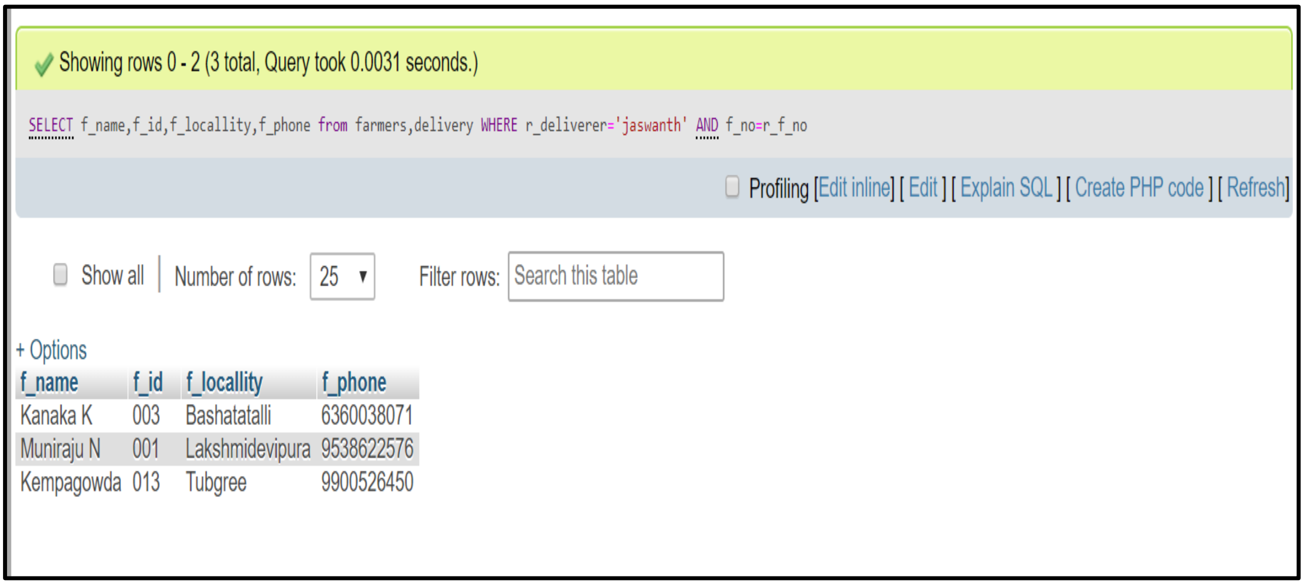
```
SELECT e_name,e_mail FROM employees WHERE e_role='manager'
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾
+ Options
←T→ ▾

				e_name	e_mail
□	Edit	Copy	Delete	Jagadish Gowda P	jaga@yahoo.com
□	Edit	Copy	Delete	Apoorva M	manager@example.com

Figure 6.2 Query 2

3. SELECT f_name,f_id,f_locality,f_phone from farmers,delivery WHERE r_deliverer='jaswanth' AND f_no=r_f_no;



✓ Showing rows 0 - 2 (3 total, Query took 0.0031 seconds.)

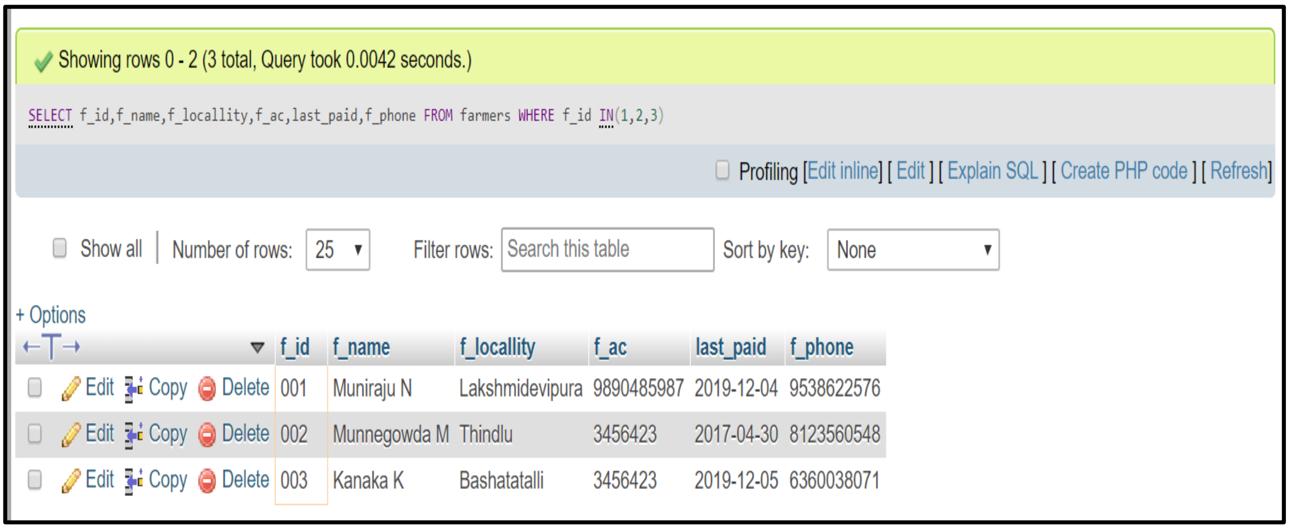
```
SELECT f_name,f_id,f_locallity,f_phone from farmers,delivery WHERE r_deliverer='jaswanth' AND f_no=r_f_no
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
+ Options

f_name	f_id	f_locallity	f_phone
Kanaka K	003	Bashatatalli	6360038071
Muniraju N	001	Lakshmidevipura	9538622576
Kempagowda	013	Tubgree	9900526450

Figure 6.3 Query 3

4. SELECT f_id,f_name,f_locallity,f_ac,last_paid,f_phone FROM farmers WHERE f_id IN(1,2,3);



✓ Showing rows 0 - 2 (3 total, Query took 0.0042 seconds.)

```
SELECT f_id,f_name,f_locallity,f_ac,last_paid,f_phone FROM farmers WHERE f_id IN(1,2,3)
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options
←T→

	f_id	f_name	f_locallity	f_ac	last_paid	f_phone
□ Edit Copy Delete	001	Muniraju N	Lakshmidevipura	9890485987	2019-12-04	9538622576
□ Edit Copy Delete	002	Munnegowda M	Thindlu	3456423	2017-04-30	8123560548
□ Edit Copy Delete	003	Kanaka K	Bashatatalli	3456423	2019-12-05	6360038071

Figure 6.4 Query 4

CHAPTER 7

CONCLUSION

With the theoretical inclination of our syllabus it becomes very essential to take the utmost advantage of any opportunity of gaining practical experience that comes along. The building block of this Mini Project “DAIRY MANAGEMENT SYSTEM” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

- The planning that goes into implementing a project.
- The importance of proper planning and an organized methodology.
- The key element of team spirit and co-ordination in a successful project.

The project also provided us the opportunity of interacting with our teachers and to gain from Their best experience.

An application has been developed using My Sql and PHP database programming connectivity via Xampp Server so as to meet the requirements of an organization, thereby ensuring quality performance.

The data can be accessed, manipulated and retrieved very easily. To conclude this software has proved to be a user friendly interface

BIBLIOGRAPHY

[1]. Ramez Elmasri, Shamkant B Navathe “Fundamentals of Database Systems”, 7TH, Pearson Edition.

[2]. Robin Nixon “Learning PHP, MySQL & JavaScript”.

[3]. Raghu Ramakrishna “Database Management System”.

[4]. R. M. Menon “Expert Oracle JDBC Programming”.

[5]. https://www.geeksforgeeks.org/

[6]. https://www.w3schools.com/

[7]. https://www.stackoverflow.com/

APPENDIX A

ER Diagram Notations:

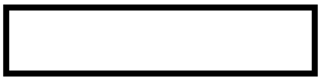
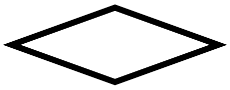
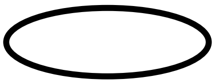
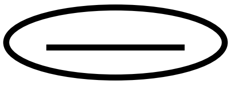
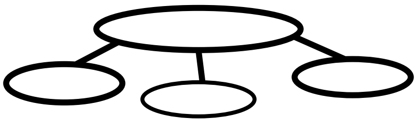
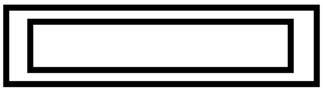
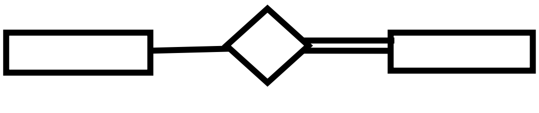
	**ENTITY**
	**RELATIONSHIP**
	**ATTRIBUTE**
	**KEY ATTRIBUTE**
	**COMPOSITE ATTRIBUTE**
	**WEAK ENTITY**
	**TOTAL PARTICIPATION**

Figure A.1: ER Diagram Notations

APPENDIX B

SNAPSHOTS

LOGIN PAGE: This is the login page through which admin going to access the database.

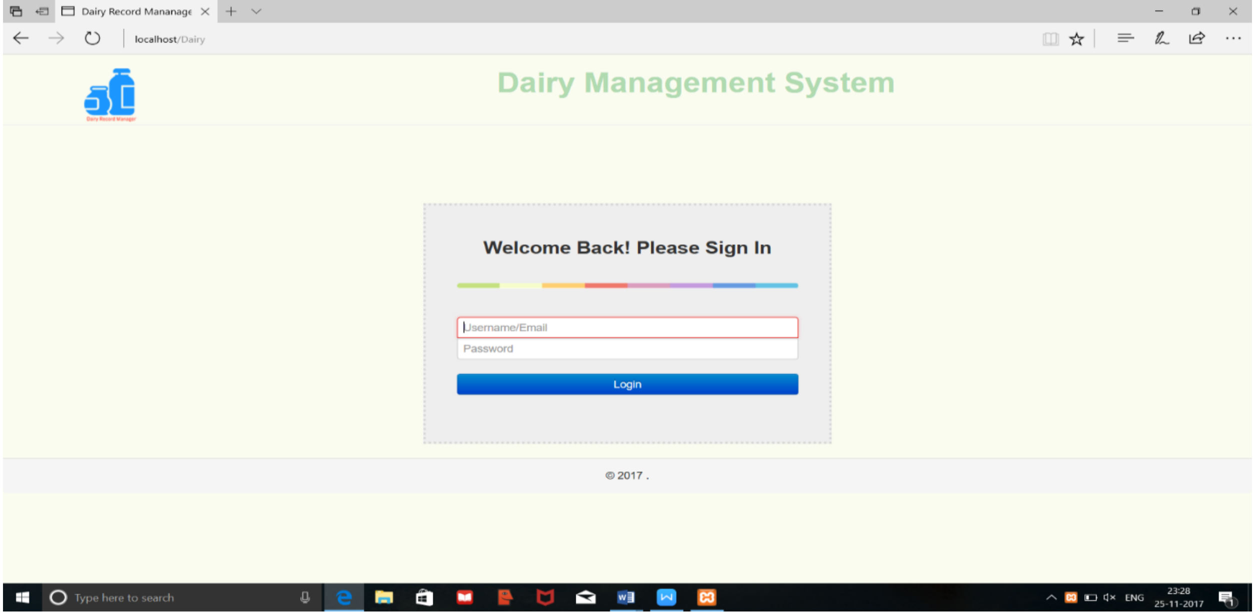


Figure B.1: Login Page

HOME PAGE: this is Home page where we get options like Farmer, Delivery, Payment, Employee and Setting.

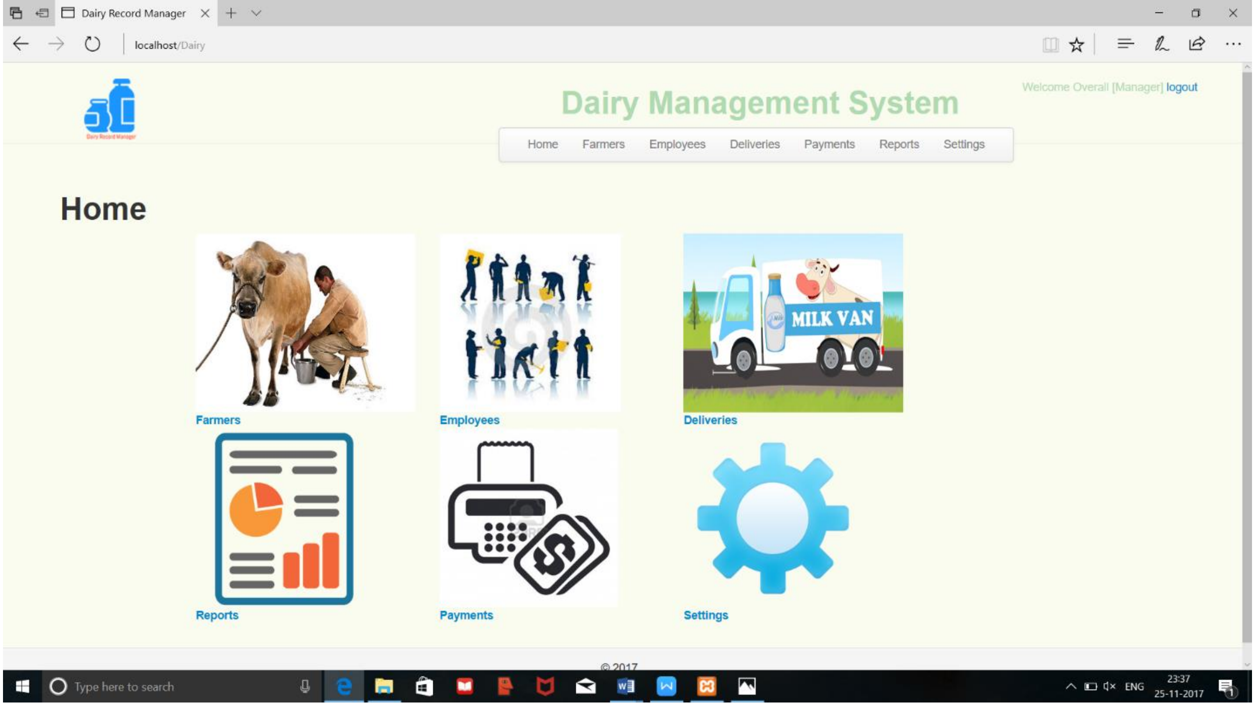
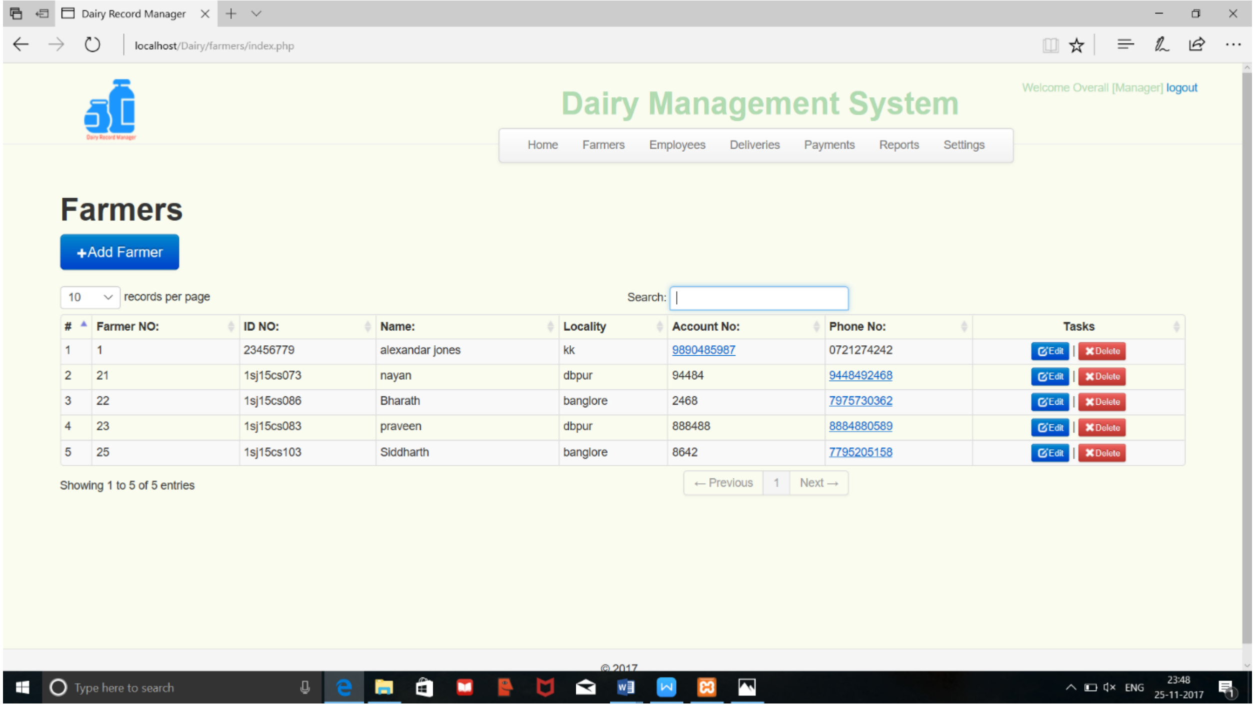


Figure B.2: Home Page

FARMER PAGE: In this page user is going to add or edit the Farmer information.



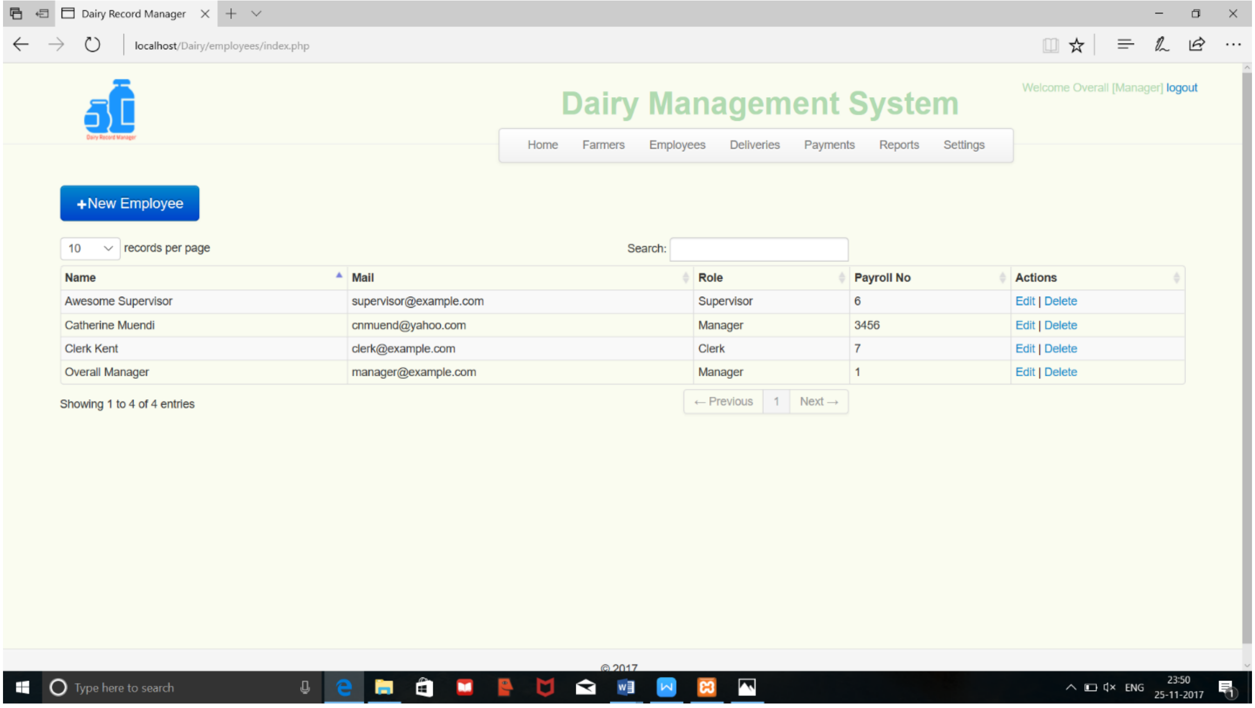
Farmers

#	Farmer NO:	ID NO:	Name:	Locality	Account No:	Phone No:	Tasks
1	1	23456779	alexandar jones	kk	9890485987	0721274242	Edit | Delete
2	21	1sj15cs073	nayan	dbpur	94484	9448492468	Edit | Delete
3	22	1sj15cs086	Bharath	banglore	2468	7975730362	Edit | Delete
4	23	1sj15cs083	praveen	dbpur	888488	8884880589	Edit | Delete
5	25	1sj15cs103	Siddharth	banglore	8642	7795205158	Edit | Delete

Showing 1 to 5 of 5 entries

Figure B.3: Farmer Page

EMPLOYEE PAGE: This page is only for the overall Manager of the system.

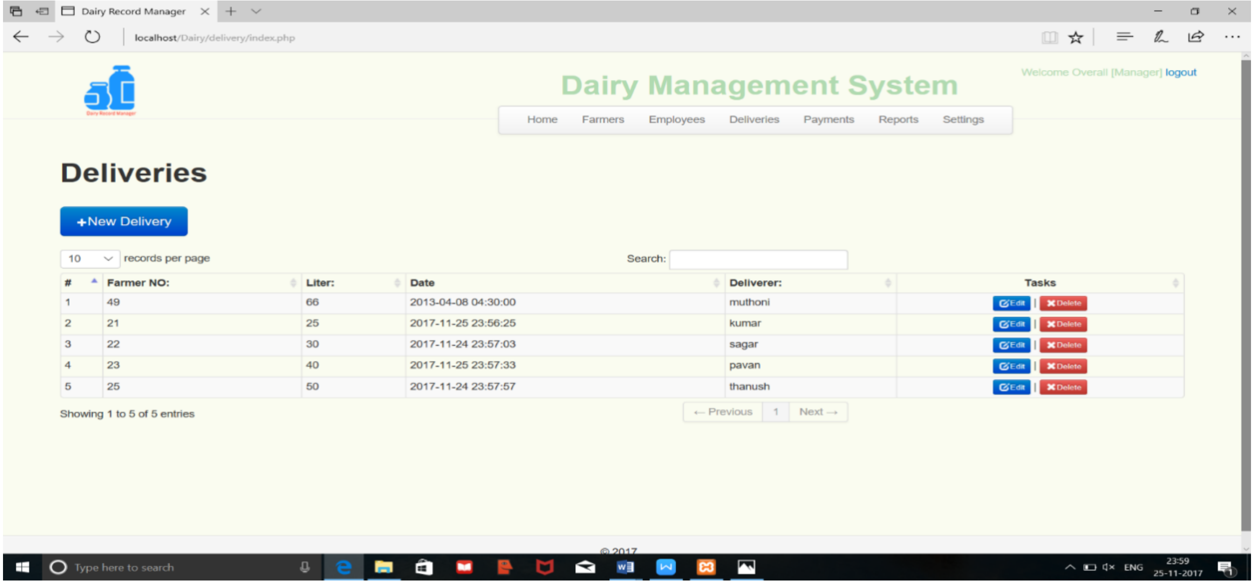


Name	Mail	Role	Payroll No	Actions
Awesome Supervisor	supervisor@example.com	Supervisor	6	Edit | Delete
Catherine Muendi	cnmuend@yahoo.com	Manager	3456	Edit | Delete
Clerk Kent	clerk@example.com	Clerk	7	Edit | Delete
Overall Manager	manager@example.com	Manager	1	Edit | Delete

Showing 1 to 4 of 4 entries

Figure B.4 :Employee Page

DELIVERIES PAGE: In this page Deliverer is going to delivers the milk into the destination location by collecting the milk from Farmer.



Deliveries

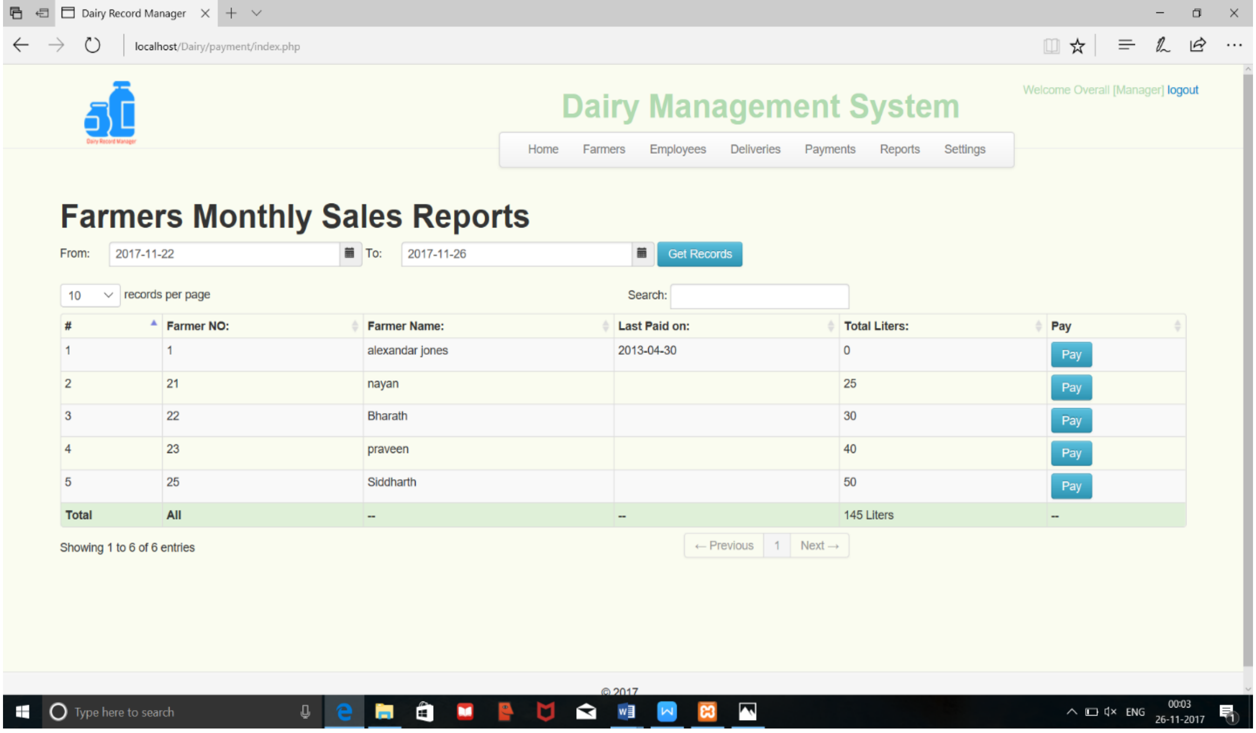
10 records per page Search:

#	Farmer NO:	Liter:	Date	Deliverer:	Tasks
1	49	66	2013-04-08 04:30:00	muthoni	Edit | Delete
2	21	25	2017-11-25 23:56:25	kumar	Edit | Delete
3	22	30	2017-11-24 23:57:03	sagar	Edit | Delete
4	23	40	2017-11-25 23:57:33	pavan	Edit | Delete
5	25	50	2017-11-24 23:57:57	thanush	Edit | Delete

Showing 1 to 5 of 5 entries ← Previous 1 Next →

Figure B.5: Deliveries Page

PAYMENT PAGE: In this page both Farmer and Deliverer is going get the payment through the bank.



Farmers Monthly Sales Reports

From: 2017-11-22 To: 2017-11-26

10 records per page Search:

#	Farmer NO:	Farmer Name:	Last Paid on:	Total Liters:	Pay
1	1	alexandar jones	2013-04-30	0	Pay
2	21	nayan		25	Pay
3	22	Bharath		30	Pay
4	23	praveen		40	Pay
5	25	Siddharth		50	Pay
Total	All	--	--	145 Liters	--

Showing 1 to 6 of 6 entries ← Previous 1 Next →

Figure B.6: payment Page

REPORT PAGE: In this page milk by each Farmer and Total Farmer delivery is reported.

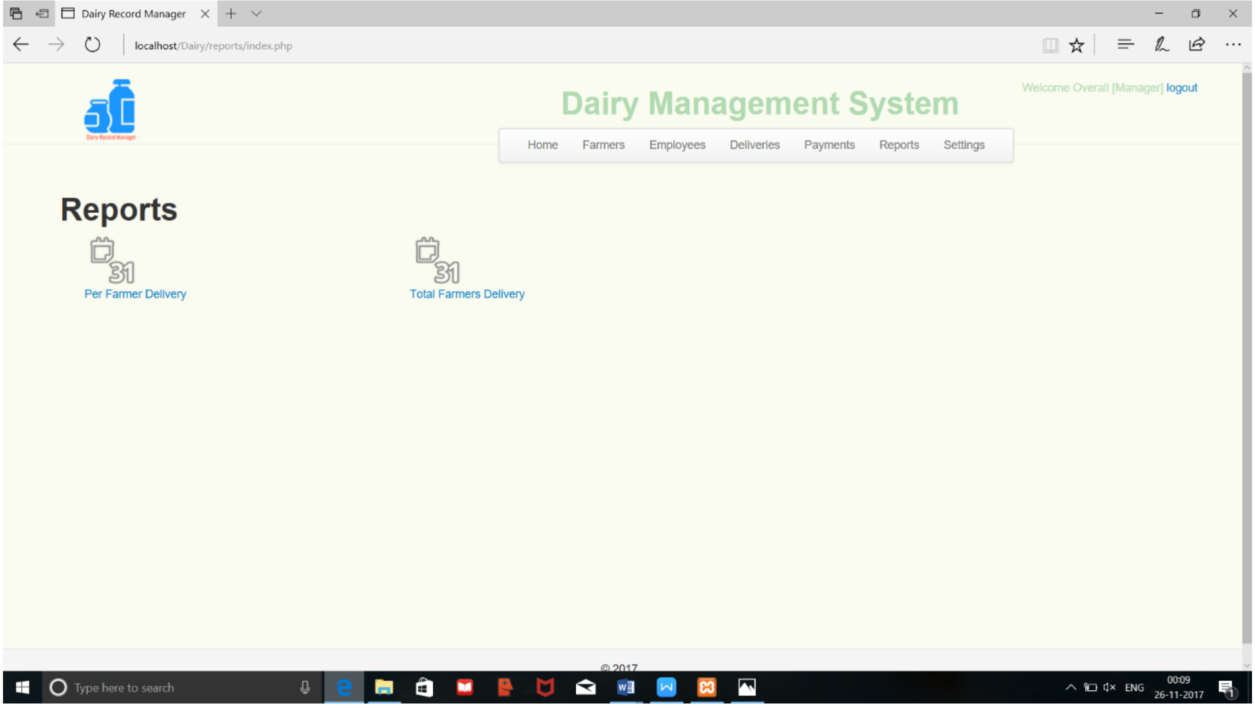
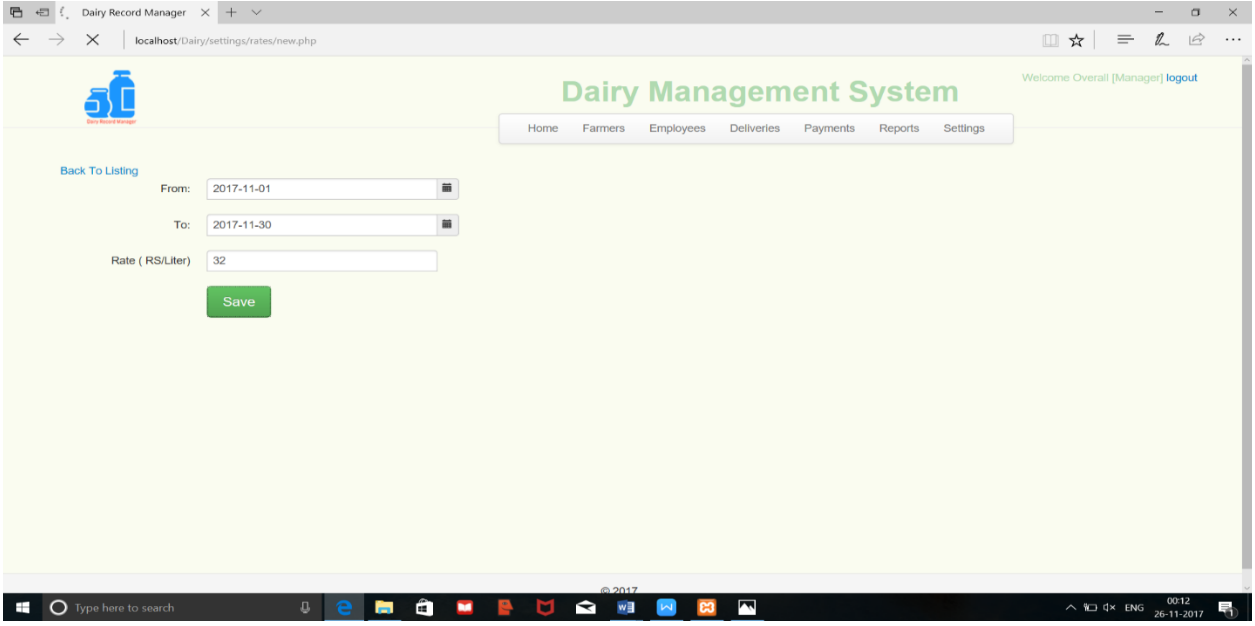


Figure B.7: report Page

SETTING RATE PAGE: By considering this page user is going to fix the cost of the milk per litre.



From: 2017-11-01

To: 2017-11-30

Rate (RS/Liter) 32

Figure B.8: setting rate Page