

A Project/ Dissertation Report
On
FARM MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING



(Established under Galgotias University, Uttar Pradesh Act No. 14 of 2011)

Submitted to:

Mr. Himanshu Sharma

Submitted by:

MD SHAKIR ZULFEQUAR

19021011992 / 19SCSE1010862

MD SOHRAB ALAM

19021011972/19SCSE1010842

SCHOOL OF COMPUTING SCIENCE & ENGINEERING

Galgotias University, Greater Noida

October – 2021

A Project/Dissertation Report

On

FARM MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING



Submitted to:

Mr. Himanshu Sharma

Submitted by:

MD SHAKIR ZULFEQUAR

19021011992 / 19SCSE1010862

MD SOHRAB ALAM

19021011972/19SCSE1010842

SCHOOL OF COMPUTING SCIENCE & ENGINEERING

Galgotias University, Greater Noida

December – 2021



**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA**

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation,
entitled "FARM MANAGEMENT SYSTEM"

in partial fulfillment of the requirements for the award of the SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida,
is an original work carried out during the period of month, Year to Month and Year, under the
supervision of Mr Himanshu, Assistant Professor Department of Computer Science and
Engineering/Computer Application and Information and Science, of School of Computing Science and
Engineering, Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the
award of any other degree of this or any other places.

MD SHAKIR ZULFEQUAR 19SCSE1010862, MD SOHRAB ALAM 19SCSE1010842

This is to certify that the above statement made by the candidates is correct to the best of
my knowledge.

Supervisor Name

Designation

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of MD SHAKIR ZULFEQUAR
19SCSE1010862, MD SOHRAB ALAM 19SCSE1010842 has been held on __21-12-2021__and his/her
work is recommended for the award of

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: December, 2021

Place: Greater Noida

ACKNOWLEDGEMENT

In performing our assignment, we had to take the help and guideline of some respected person, who deserve our greatest gratitude. The completion of this assignment gives us much Pleasure. We would like to show our gratitude Mr. Himanshu Sharma., Project Instructor, Galgotias University who introduced us to the Methodology of work, and whose passion for the “underlying structures” had lasting effect and for giving us a good guideline for assignment throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this project.

Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our assignment. We thank all the people for their help directly and indirectly to complete our assignment.

ABSTRACT

The purpose of Farm Management System is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Farm Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients.

List of Figures

Figure no.	Title	Page no.
1	Data Flow Diagram	19
2	Use Case Diagram	25
3	E R Diagram	30

TABLE OF CONTENTS

S.No	Particulars	Page No
1	Acknowledgement	2
2	Abstract	3
3	Introduction	6
4	Literature Reviews/Comparative study	7-10
5	<u>Feasibility Study</u>	14-18
6	Project Planing	20
7	Required Tools	23
8	Project Profile	25
9	References	43

INTRODUCTION

The "Farm Management System" is designed to overcome the problems that exist in the manual system. This software is supported to eliminate and in some cases reduce the complexity of the existing program. In addition the program is designed for the specific need of the company to perform its functions in a smooth and efficient manner.

The app is minimized as much as possible to avoid errors while entering data. It also provides an error message while entering invalid data. No official information is required for the user to use this program. So all of this proves to be practical. The Farm Management System, as described above, can lead to error-free, secure, reliable and fast system management. It can help the user to focus on what he is doing instead of focusing on keeping records. It will therefore help to streamline the better use of resources.

Every organization, whether large or small, has challenges in overcoming and managing plant data, Farm, Insecticides, Equipments, Pesticides. Every Farm Management System has different Farm requirements, so we create special employee management systems that fit your management needs. This is designed to help you strategically plan, and will help you ensure that your organization has the right level of information and details of your future goals. Also, for those busy managers who are always on the move, our systems come with remote access features, which will allow you to manage your employees at any time, at any time. These programs will ultimately allow you to better manage resources.

Objective of Project on Farm Management System:

The main objective of the Project on Farm Management System is to manage the details of Farm, Crops, Crops, Insecticides, Pesticides. It manages all the information about Farm, Cost Range, Pesticides, Farm. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Farm, Crops, Cost Range, Crops. It tracks all the details about the Crops, Insecticides, Pesticides.

Functionalities provided by Farm Management System are as follows:

- Provides the searching facilities based on various factors. Such as Farm, Crops, Insecticides, Pesticides
- Farm Management System also manage the Cost Range details online for Insecticides details, Pesticides details, Farm.
- It tracks all the information of Crops, Cost Range, Insecticides etc
- Manage the information of Crops
- Shows the information and description of the Farm, Crops
- To increase efficiency of managing the Farm, Crops
- It deals with monitoring the information and transactions of Insecticides.
- Manage the information of Farm
- Editing, adding and updating of Records is improved which results in proper resource management of Farm data.
- Manage the information of Insecticides
- Integration of all records of Pesticides.

Scope of the project Farm Management System

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Farm Management System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

Our project aims at Business process automation, i.e. we have tried to computerize various processes of Farm Management System.

- In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
- In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To assist the staff in capturing the effort spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation.
- The system generates types of information that can be used for various purposes.
- It satisfy the user requirement
- Be easy to understand by the user and operator
- Be easy to operate
- Have a good user interface
- Be expandable
- Delivered on schedule within the budget.

Reports of Farm Management System:

- It generates the report on Farm, Crops, Cost Range
- Provide filter reports on Crops, Insecticides, Pesticides
- You can easily export PDF for the Farm, Cost Range, Insecticides
- Application also provides excel export for Crops, Crops, Pesticides
- You can also export the report into csv format for Farm, Crops, Pesticides

Modules of Farm Management System:

- Farm Management Module: Used for managing the Farm details.
- Pesticides Module : Used for managing the details of Pesticides
- Cost Range Module : Used for managing the details of Cost Range
- Crops Management Module: Used for managing the information and details of the Crops.
- Crops Module : Used for managing the Crops details
- Insecticides Module : Used for managing the Insecticides information
- Login Module: Used for managing the login details
- Users Module : Used for managing the users of the system

Input Data and Validation of Project on Farm Management System

- All the fields such as Farm, Crops, Pesticides are validated and does not take invalid values
- Each form for Farm, Crops, Cost Range can not accept blank value fields
- Avoiding errors in data
- Controlling amount of input
- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.
- Functionality of the entire module/forms.
- Validations for user input.
- Checking of the Coding standards to be maintained during coding.
- Testing the module with all the possible test data.
- Testing of the functionality involving all type of calculations etc.
- Commenting standard in the source files.

The software quality plan we will use the following SQA Strategy:

- In the first step, we will select the test factors and rank them. The selected test factors such as reliability, maintainability, portability or etc, will be placed in the matrix according to their ranks.
- The second step is for identifying the phases of the development process. The phase should be recorded in the matrix.
- The third step is that identifying the business risks of the software deliverables. The risks will be ranked into three ranks such as high, medium and low.

Features of the project Farm Management System:

- Product and Component based
- Creating & Changing Issues at ease
- Query Issue List to any depth
- Reporting & Charting in more comprehensive way
- User Accounts to control the access and maintain security
- Simple Status & Resolutions
- Multi-level Priorities & Severities.
- Targets & Milestones for guiding the programmers
- Attachments & Additional Comments for more information
- Robust database back-end
- Various level of reports available with a lot of filter criteria's
- It contain better storage capacity.
- Accuracy in work.
- Easy & fast retrieval of information.
- Well designed reports.
- Decrease the load of the person involve in existing manual system.
- Access of any information individually.
- Work becomes very speedy.
- Easy to update information

Software Requirement Specification

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

The proposed system has the following requirements:

- System needs store information about new entry of Farm.
- System needs to help the internal staff to keep information of Crops and find them as per various queries.
- System need to maintain quantity record.
- System need to keep the record of Crops.
- System need to update and delete the record.
- System also needs a search area.
- It also needs a security system to prevent data.

Identification of need:

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. there used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records.

The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business .For this reason we have provided features Present system is partially automated (computerized), actually existing system is quite laborious as one has to enter same information at three different places.

Following points should be well considered:

- Documents and reports that must be provided by the new system: there can also be few reports, which can help management in decision-making and cost controlling, but since these reports do not get required attention, such kind of reports and information were also identified and given required attention.
- Details of the information needed for each document and report.
- The required frequency and distribution for each document.
- Probable sources of information for each document and report.
- With the implementation of computerized system, the task of keeping records in an organized manner will be solved. The greatest of all is the retrieval of information, which will be at the click of the mouse. So the proposed system helps in saving the time in different operations and making information flow easy giving valuable reports.

Feasibility Study:

After doing the project Farm Management System, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

A. Economical Feasibility

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

- All hardware and software cost has to be borne by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

B. Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platformst.

C. Operational Feasibility

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

System Design of Farm Management System

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the clients's requirements into a logically working system. Normally, design is performed in the following in the following two steps:

1. Primary Design Phase:

In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimising the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

2. Secondary Design Phase:

In the secondary phase the detailed design of every block is performed.

The general tasks involved in the design process are the following:

1. Design various blocks for overall system processes.
2. Design smaller, compact and workable modules in each block.
3. Design various database structures.
4. Specify details of programs to achieve desired functionality.
5. Design the form of inputs, and outputs of the system.
6. Perform documentation of the design.
7. System reviews.

Preliminary Product Description:

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

Analysts working on the preliminary investigation should accomplish the following objectives:

- Clarify and understand the project request
- Determine the size of the project.
- Assess costs and benefits of alternative approaches.
- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management, with recommendations outlining the acceptance or rejection of the proposal.
- **Benefit to Organization**

The organization will obviously be able to gain benefits such as savings in operating cost, reduction in paperwork, better utilization of human resources and more presentable image increasing goodwill.

- **The Initial Cost**

The initial cost of setting up the system will include the cost of hardware software (OS, add-on software, utilities) & labour (setup & maintenance). The same has to bear by the organization.

Running Cost

Besides, the initial cost the long term cost will include the running cost for the system including the AMC, stationary charges, cost for human resources, cost for update/renewal of various related software.

- **Need for Training**

The users along with the administrator need to be trained at the time of implementation of the system for smooth running of the system. The client will provide the training site.

We talked to the management people who were managing a the financial issues of the center, the staff who were keeping the records in lots of registers and the reporting manager regarding their existing system, their requirements and their expectations from the new

proposed system. Then, we did the system study of the entire system based on their requirements and the additional features they wanted to incorporate in this system.

Reliable, accurate and secure data was also considered to be a complex task without this proposed system. Because there was no such record for keeping track of all the activities, which was done by the Farm Management System on the daily basis.

The new system proposed and then developed by me will ease the task of the organization in consideration. It will be helpful in generating the required reports by the staff, which will help them to track their progress and services.

Thus, it will ease the task of Management to a great extent as all the major activities to be performed, are computerized through this system.

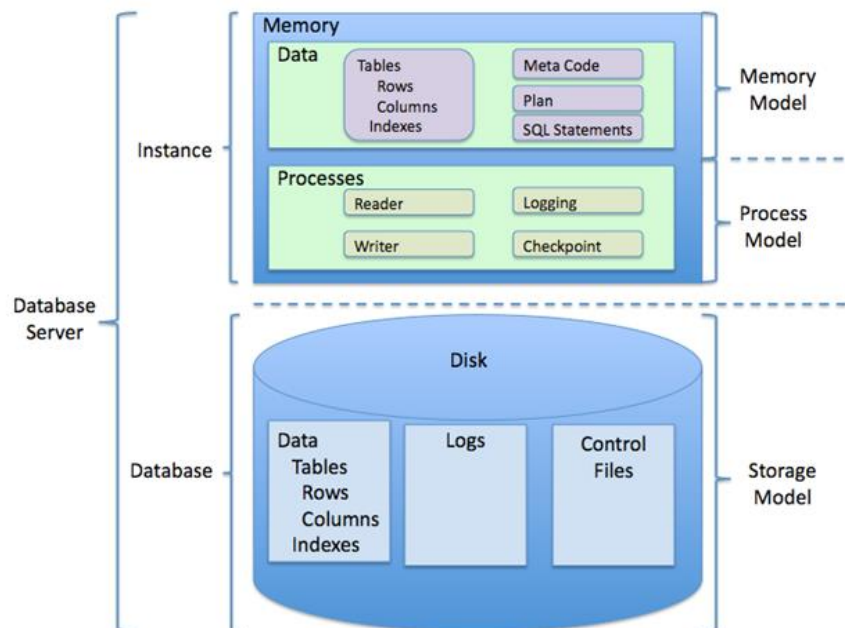
Project Category

Relational Database Management System (RDBMS) : This is an RDBMS based project which is currently using MySQL for all the transaction statements. MySQL is an opensource RDBMS System.

Brief Introduction about RDBSM :

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. Many popular databases currently in use are based on the relational database model.

RDBMSs have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and much more since the 1980s. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use. However, relational databases have been challenged by object databases, which were introduced in an attempt to address the object-relational impedance mismatch in relational database, and XML databases.



Implementation Methodology:

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

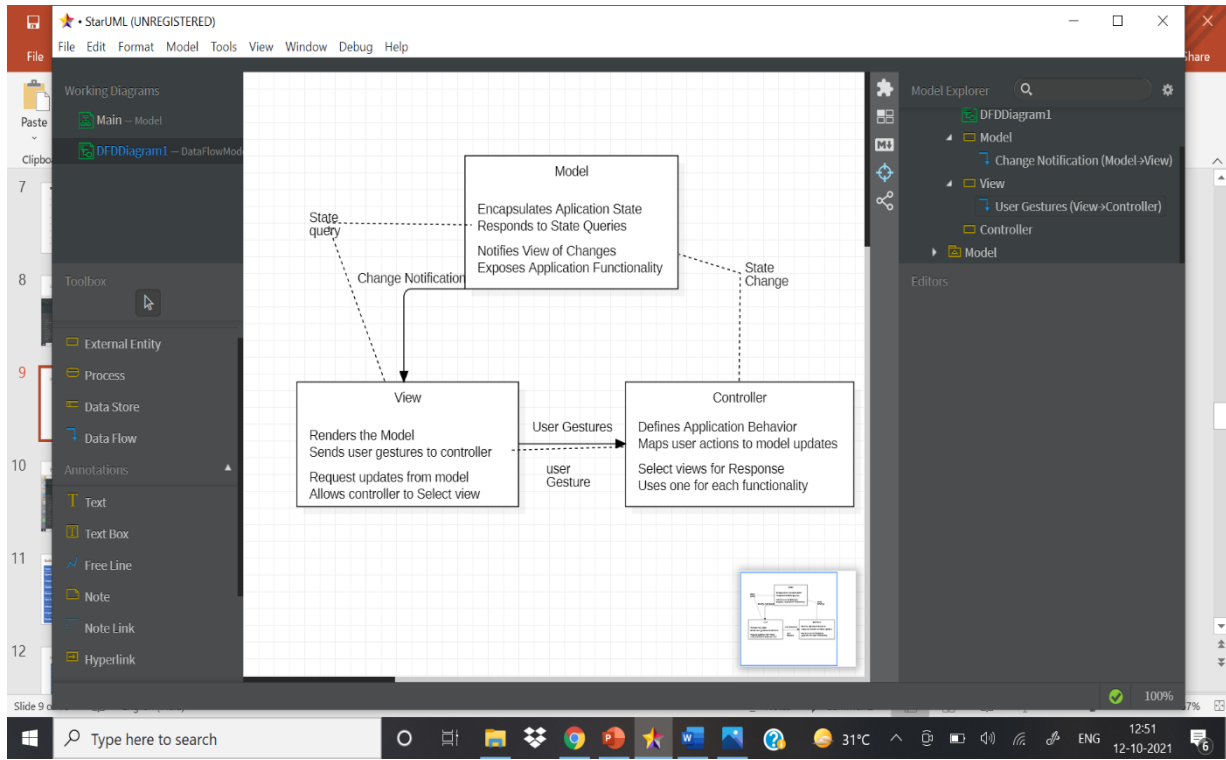
- **Model** - The lowest level of the pattern which is responsible for maintaining data.
- **View** - This is responsible for displaying all or a portion of the data to the user.
- **Controller** - Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then

works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.

MVC (Model View Controller Flow) Diagram

DATA FLOW DIAGRAMS



Project Planning:

Software project plan can be viewed as the following:

- 1) **Within the organization:** How the project is to be implemented? What are various constraints (time, cost, staff)? What is market strategy?
- 2) **With respect to the customer:** Weekly or timely meetings with the customer with presentation on status reports. Customers feedback is also taken and further modification and developments are done. Project milestones and deliverables are also presented to the customer.

For a successful software project, the following steps can be followed:

- Select a project
 - Identifying project's aims and objectives
 - Understanding requirements and specification
 - Methods of analysis, design and implementation
 - Testing techniques
 - Documentation
- Project milestones and deliverables
- Budget allocation
 - Exceeding limits within control
- Project Estimates
 - Cost
 - Time
 - Size of code
 - Duration
- Resource Allocation
 - Hardware
 - Software
 - Previous relevant project information
 - Digital Library
- Risk Management
 - Risk avoidance
 - Risk detection

Cost estimation of the project:

Software cost comprises a small percentage of overall computer-based system cost. There are a number of factors, which are considered, that can affect the ultimate cost of the software such as - human, technical, Hardware and Software availability etc.

The main point that was considered during the cost estimation of **project** was its sizing. In spite of complete software sizing, function point and approximate lines of code were also used to "size" each element of the Software and their costing.

The cost estimation done by me for **Project** also depend upon the baseline metrics collected from past projects and these were used in conjunction with estimation variables to develop cost and effort projections.

We have basically estimated this project mainly on two bases -

- 1) **Effort Estimation** - This refers to the total man-hours required for the development of the project. It even includes the time required for doing documentation and user manual.
- 2) **Hardware Required Estimation** - This includes the cost of the PCs and the hardware cost required for development of this project.

Tools/Platform, Hardware and Software Requirement specifications:

Software Requirements:

Name of component	Specification
Operating System	Windows 98, Windows XP, Windows7, Linux
Language	Java 2 Runtime Environment
Database	MySQL Server
Browser	Any of Mozilla, Opera, Chrome etc
Web Server	Tomcat 7
Software Development Kit	Java JDK 1.7 or Above
Scripting Language Enable	JSP (Java Server Pages)
Database JDBC Driver	MySQL Jconnector

Hardware Requirements:

Name of component	Specification
Processor	Pentium III 630MHz
RAM	128 MB
Hard disk	20 GB
Monitor	15" color monitor
Keyboard	122 keys

Project Profile

There has been continuous effort to develop tools, which can ease the process of software development. But, with the evolving trend of different programming paradigms today's software developers are really challenged to deal with the changing technology. Among other issues, software re-engineering is being regarded as an important process in the software development industry. One of the major tasks here is to understand software systems that are already developed and to transform them to a different software environment. Generally, this requires a lot of manual effort in going through a program that might have been developed by another programmer. This project makes a novel attempt to address the issue of program analysis and generation of diagrams, which can depict the structure of a program in a better way. Today, UML is being considered as an industrial standard for software engineering design process. It essentially provides several diagramming tools that can express different aspects/ characteristics of program such as

Use cases: Elicit requirement from users in meaningful chunks. Construction planning is built around delivering some use cases on each interaction basis for system testing.

Class diagrams: shows static structure of concepts, types and class. Concepts show how users think about the world; type shows interfaces of software components; classes show implementation of software components.

Interaction diagrams: shows how several objects collaborate in single use case.

Package diagram: show group of classes and dependencies among them.

State diagram: show how single object behaves across many use cases.

Activity diagram: shows behavior with control structure. Can show many objects over many uses, many object in single use case, or implementations methods encourage parallel behavior, etc.

The end-product of this project is a comprehensive tool that can parse any vb.net program and extract most of the object oriented features inherent in the program such as polymorphism, inheritance, encapsulation and abstraction.

What is UML?

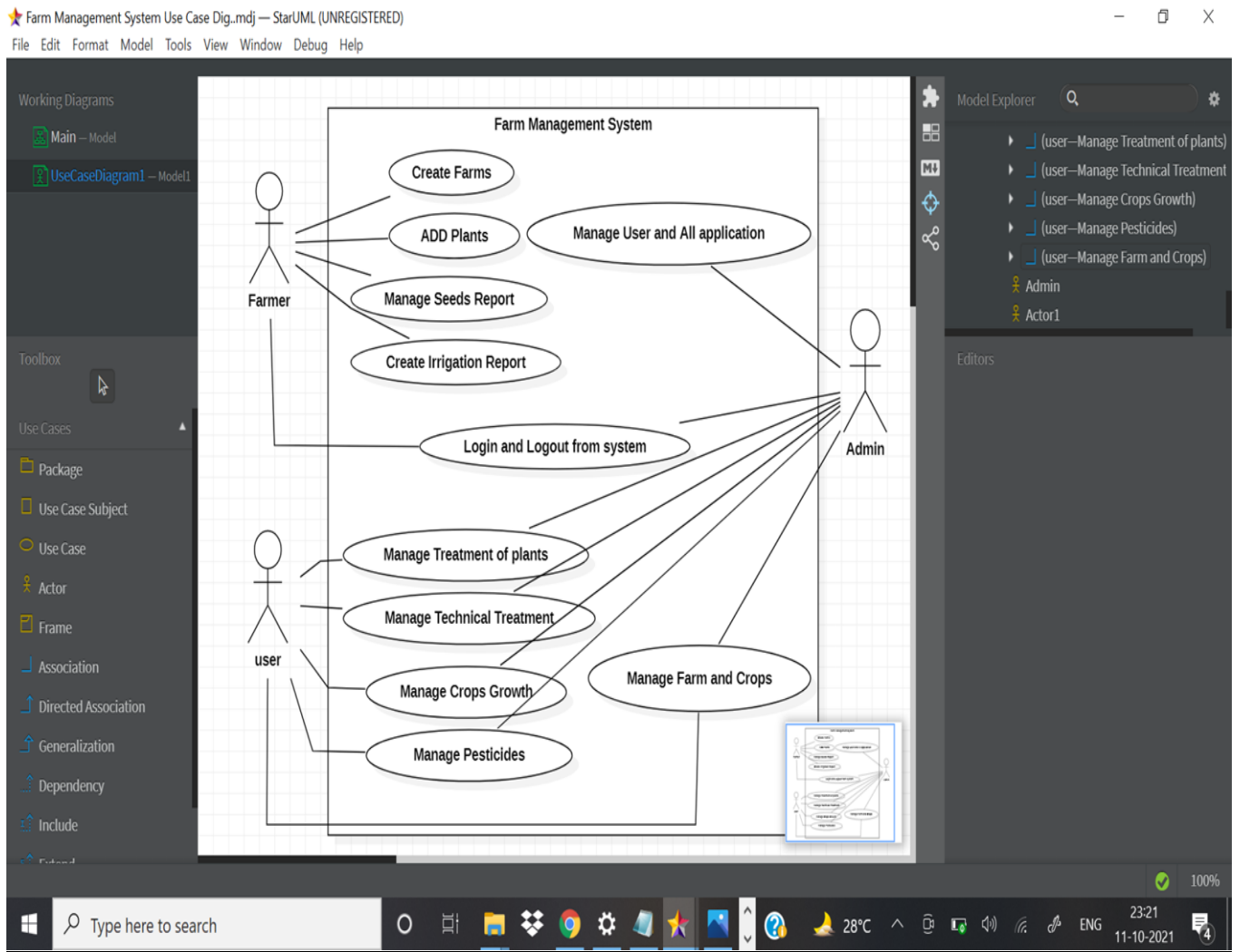
UML stands for Unified Modeling Language is the successor to the wave of Object Oriented Analysis and Design (OOA&D) methods that appeared in the late 80's. It most directly unifies the methods of Booch, Rumbaugh (OMT) and Jacobson. The UML is called a modeling language, not a method. Most methods consist at least in principle, of both a modeling language and a process. The Modeling language is that notation that methods used to express design.

Use Case Model of the Project:

The use case model for any system consists of "use cases". Use cases represent different ways in which the system can be used by the user. A simple way to find all the use case of a system is to ask the questions "What the user can do using the system?" The use cases partition the system behavior into transactions such that each transaction performs some useful action from the users' point of view.

The purpose of the use case to define a piece of coherent behavior without revealing the internal structure of the system. An use case typically represents a sequence of interaction between the user and the system. These interactions consists of one main line sequence is represent the normal interaction between the user and the system. The use case model is an important analysis and design artifact (task). Use cases can be represented by drawing a use case diagram and writing an accompany text elaborating the drawing.

In the use case diagram each use case is represented by an ellipse with the name of use case written inside the ellipse. All the ellipses of the system are enclosed with in a rectangle which represents the system boundary. The name of the system being moduled appears inside the rectangle. The different users of the system are represented by using stick person icon. The stick person icon is normally referred to as an Actor. The line connecting the actor and the use cases is called the communication relationship. When a stick person icon represents an external system it is annotated by the stereo type <<external system>>.



Dataflow Diagram:

Data flow diagram is the starting point of the design phase that functionally decomposes the requirements specification. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flows in the system. A DFD describes what data flow rather than how they are processed, so it does not hardware, software and data structure.

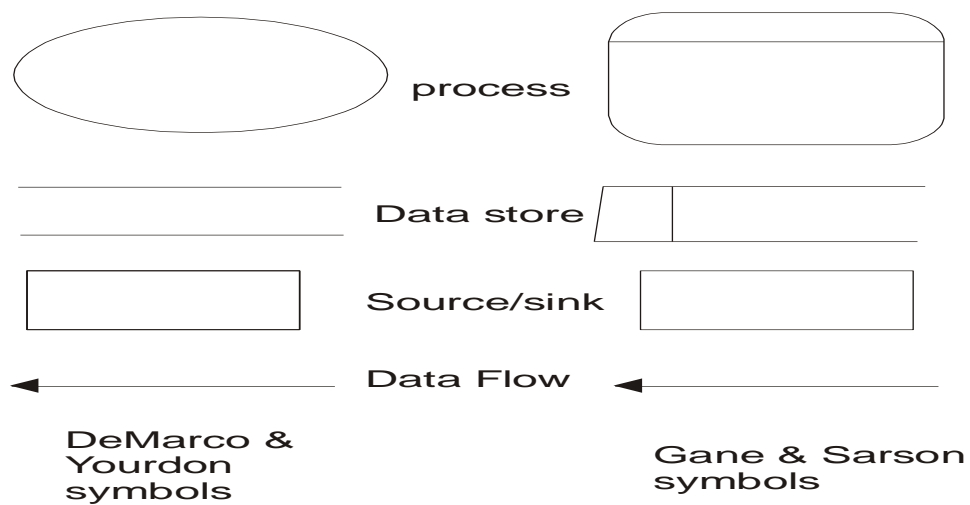
A **data-flow diagram (DFD)** is a graphical representation of the "flow" of data through an [information system](#). DFDs can also be used for the [visualization](#) of [data processing](#) (structured design). A **data flow diagram (DFD)** is a significant modeling technique for analyzing and constructing information processes. DFD literally means an illustration that explains the course or movement of information in a process. DFD illustrates

this flow of information in a process based on the inputs and outputs. A DFD can be referred to as a Process Model.

The data flow diagram is a graphical description of a system's data and how to

Process transform the data is known as Data Flow Diagram (DFD).

Unlike details flow chart, DFDs don't supply detail descriptions of modules that graphically describe a system's data and how the data interact with the system. Data flow diagram number of symbols and the following symbols are of by DeMarco.



There are seven rules for construct a data flow diagram.

- 1) Arrows should not cross each other.
- 2) Squares, circles and files must wears names.
- 3) Decomposed data flows must be balanced.
- 4) No two data flows, squares or circles can be the same names.
- 5) Draw all data flows around the outside of the diagram.
- 6) Choose meaningful names for data flows, processes & data stores.
- 7) Control information such as record units, password and validation requirements are not penitent to a data flow diagram.

Additionally, a **DFD** can be utilized to visualize data processing or a structured design.

This basic DFD can be then disintegrated to a lower level diagram demonstrating smaller steps exhibiting details of the system that is being modeled.

On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process. It is common practice to draw a [context-level data flow diagram](#) first, which shows the interaction between the system and external agents, which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD'), the system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization.

This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. The level 1 DFD is further spreaded and split into more descriptive and detailed description about the project as level 2 DFD. The level 2 DFD can be a number of data flows which will finally show the entire description of the software project.

About ER Diagram:


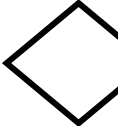
Entity Relationship Diagram

E-R Model is a popular high level conceptual data model. This model and its variations are frequently used for the conceptual design of database application and many database design tools employ its concept.

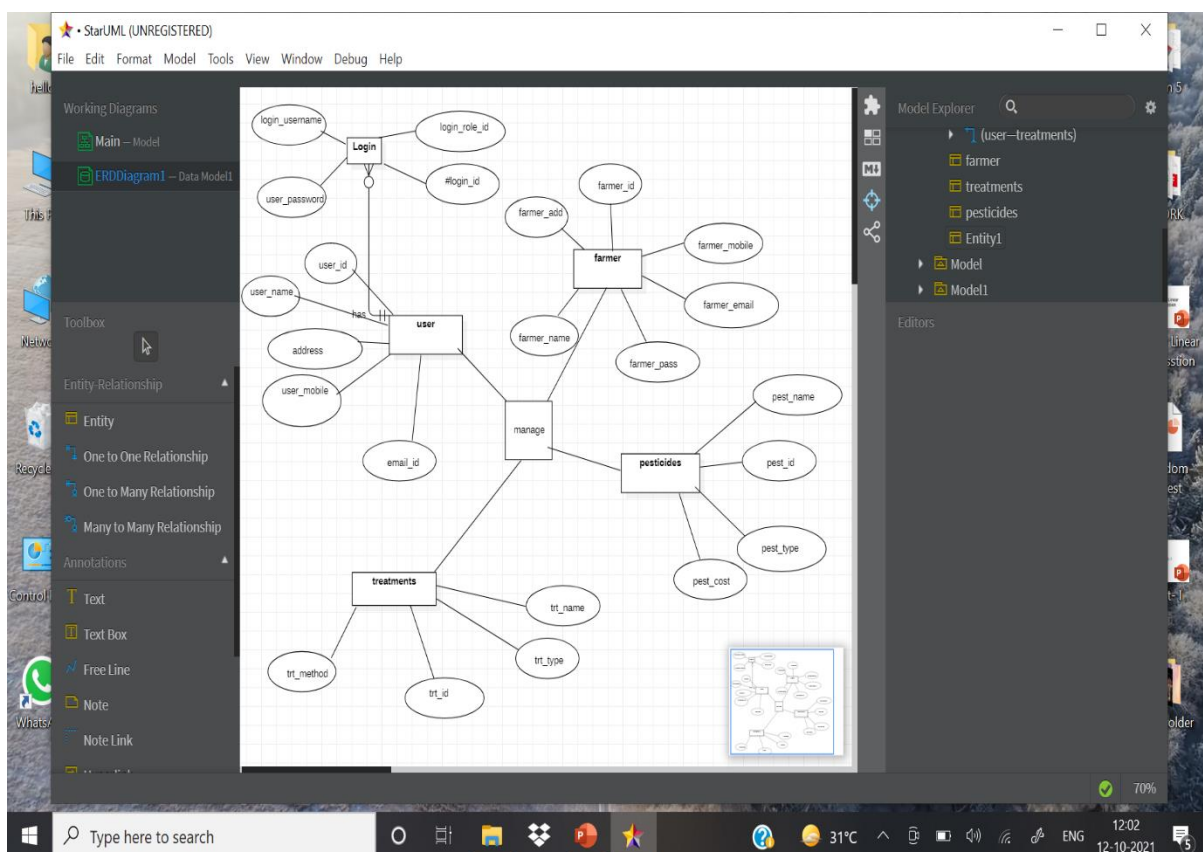
A database that conforms to an E-R diagram can be represented by a collection of tables in the relational system. The mapping of E-R diagram to the entities are:

- Attributes
- Relations
 - Many-to-many
 - Many-to-one
 - One-to-many
 - One-to-one
- Weak entities
- Sub-type and super-type

The entities and their relationships between them are shown using the following conventions.

- An entity is shown in rectangle. 
- A diamond represent the relationship among number of entities. 
- The attributes shown as ovals are connected to the entities or relationship by lines.
- Diamond, oval and relationships are labeled.
- Model is an abstraction process that hides super details while highlighting details relation to application at end.
 - A data model is a mechanism that provides this abstraction for database application.
Data modeling is used for representing entities and their relationship in the database.

- Entities are the basic units used in modeling database entities can have concrete existence or constitute ideas or concepts.
- Entity type or entity set is a group of similar objects concern to an organization for which it maintain data,
- Properties are characteristics of an entity also called as attributes.
- A key is a single attribute or combination of 2 or more attributes of an entity set is used to identify one or more instances of the set.
- In relational model we represent the entity by a relation and use tuples to represent an instance of the entity.
- Relationship is used in data modeling to represent in association between an entity set.
- An association between two attributes indicates that the values of the associated attributes are independent.



Security Testing of the Project

Testing is vital for the success of any software. no system design is ever perfect. Testing is also carried in two phases. first phase is during the software engineering that is during the module creation. second phase is after the completion of software. this is system testing which verifies that the whole set of programs hanged together.

White Box Testing:

In this technique, the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. all logical parts of the software checked once. errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops. When the box testing tests all the independent part within a module a logical decisions on their true and the false side are exercised , all loops and bounds within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

Black Box Testing:

This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. black box testing tests the input, the output and the external data. it checks whether the input data is correct and whether we are getting the desired output.

Alpha Testing:

Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

Beta Testing:

On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of

potential users who agree to use it. The customers then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

Unit Testing:

Each module is considered independently. it focuses on each unit of software as implemented in the source code. it is white box testing.

Integration Testing:

Integration testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules. modules are integrated by using the top down approach.

Validation Testing:

Validation testing was performed to ensure that all the functional and performance requirements are met.

System Testing:

It is executing programs to check logical changes made in it with intention of finding errors. a system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

Implementation and Software Specification Testings

Detailed Design of Implementation

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

Technical Design

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

Test Specifications and Planning

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole.

Programming and Testing

This activity encompasses actual development, writing, and testing of program units or modules.

User Training

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

Acceptance Test

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

Installation Phase

In this phase the new Computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

System Installation

The process of starting the actual use of a system and training user personnel in its operation.

Review Phase

This phase evaluates the successes and failures during a systems development project, and to measure the results of a new Computerized Transystem in terms of benefits and savings projected at the start of the project.

Development Recap

A review of a project immediately after completion to find successes and potential problems in future work.

Post-Implementation Review

A review, conducted after a new system has been in operation for some time, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system.

THE STEPS IN THE SOFTWARE TESTING

The steps involved during Unit testing are as follows:

- a. Preparation of the test cases.
- b. Preparation of the possible test data with all the validation checks.
- c. Complete code review of the module.
- d. Actual testing done manually.
- e. Modifications done for the errors found during testing.

- f. Prepared the test result scripts.

The unit testing done included the testing of the following items:

1. Functionality of the entire module/forms.
2. Validations for user input.
3. Checking of the Coding standards to be maintained during coding.
4. Testing the module with all the possible test data.
5. Testing of the functionality involving all type of calculations etc.
6. Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System Integration, We integrated the modules one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

The steps involved during System testing are as follows:

- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.

The System Testing done included the testing of the following items:

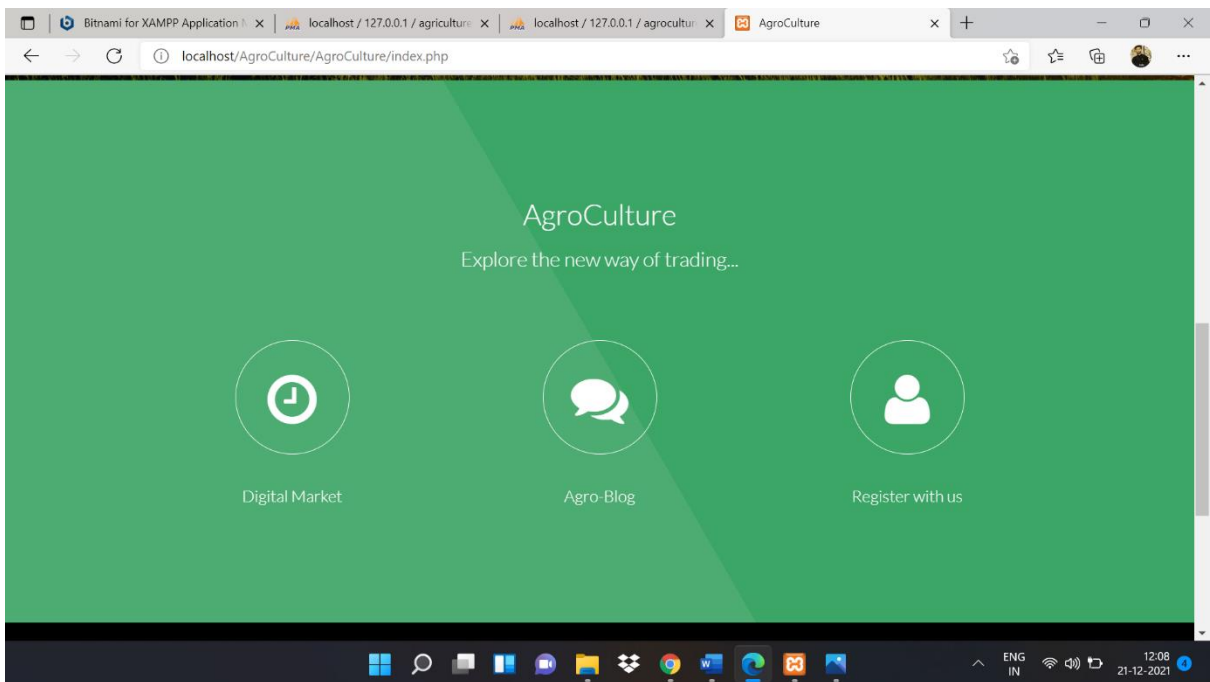
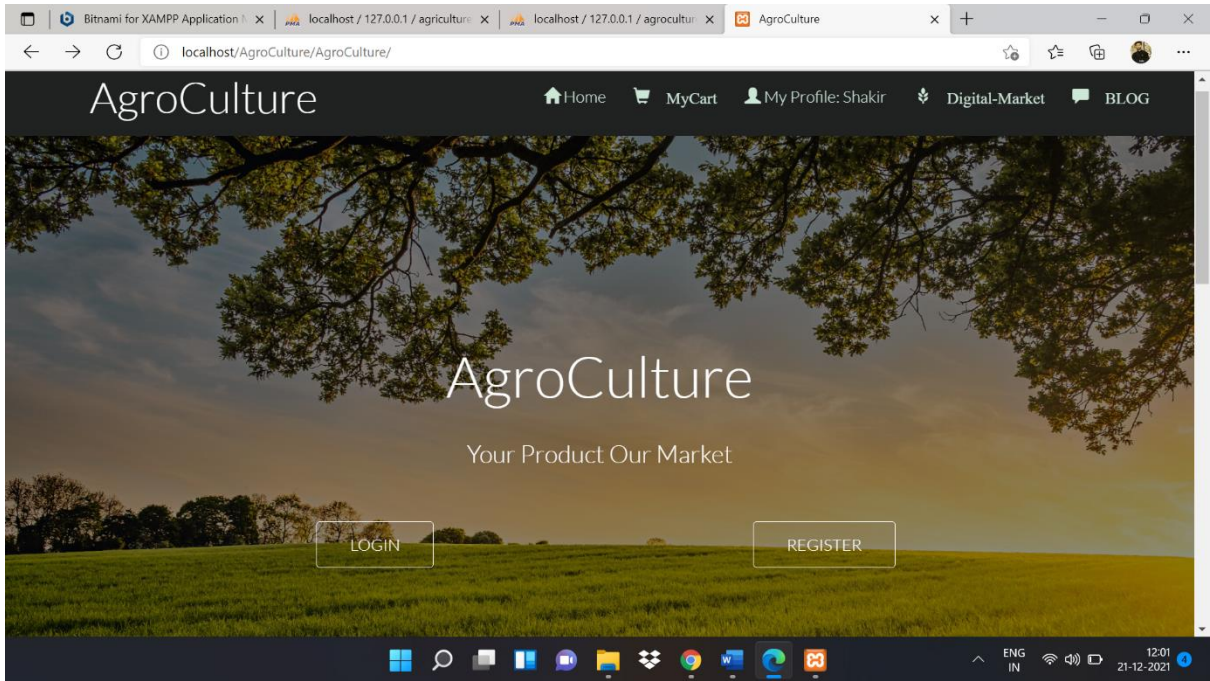
1. Functionality of the entire system as a whole.
2. User Interface of the system.
3. Testing the dependent modules together with all the possible test data scripts.
4. Verification and Validation testing.
5. Testing the reports with all its functionality.

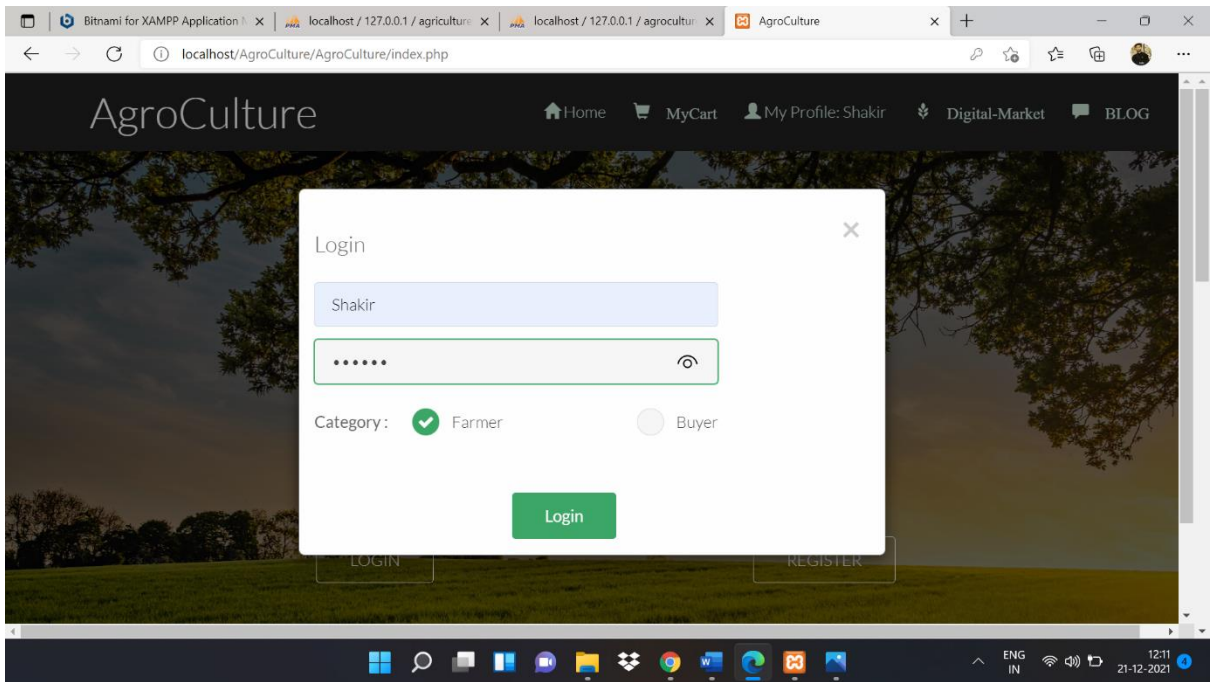
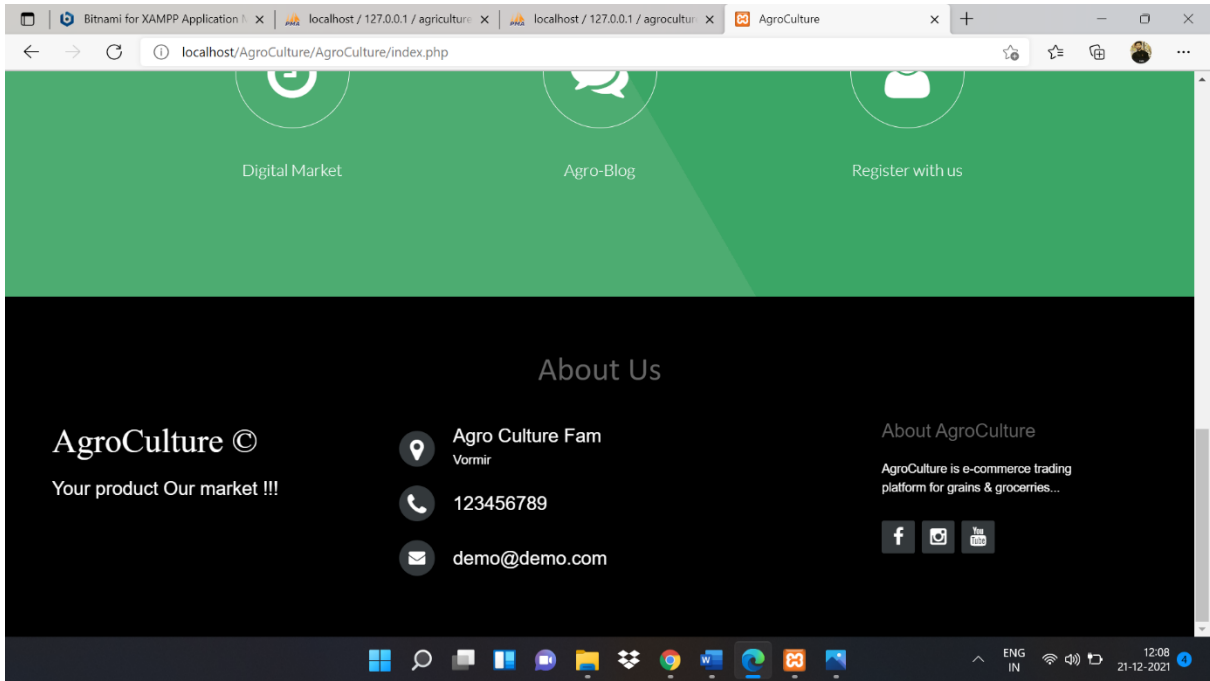
After the completion of system testing, the next following phase was the Acceptance Testing. Clients at their end did this and accepted the system with appreciation. Thus, we reached the final phase of the project delivery.

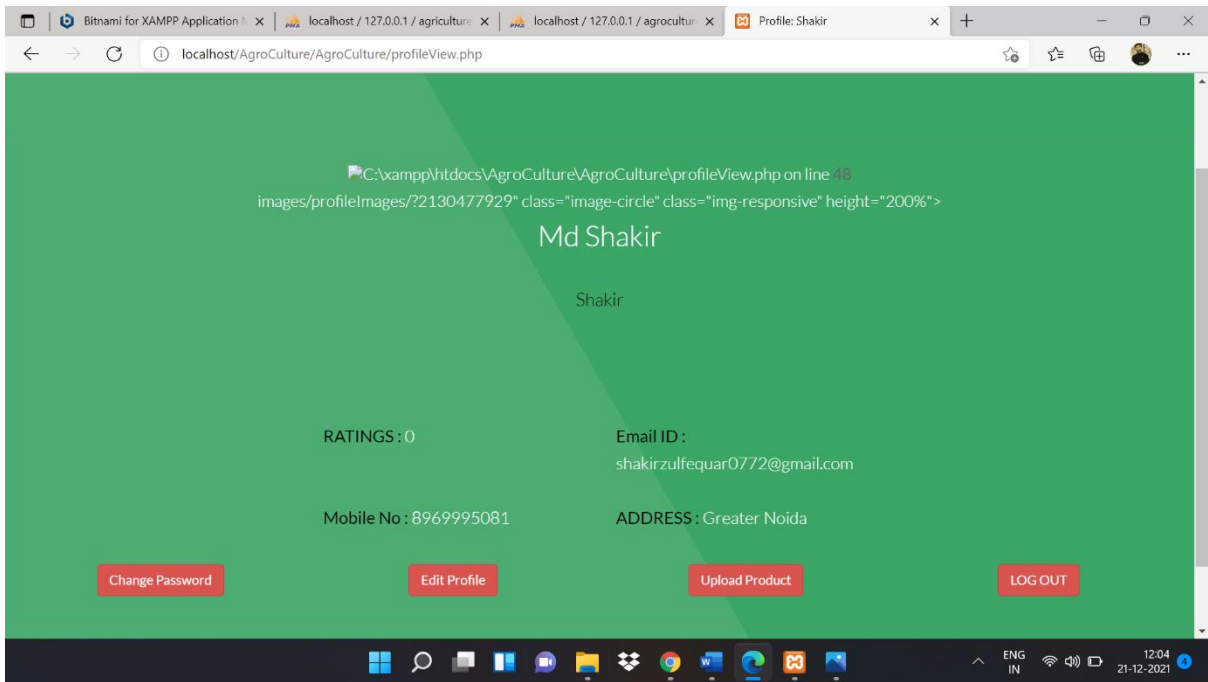
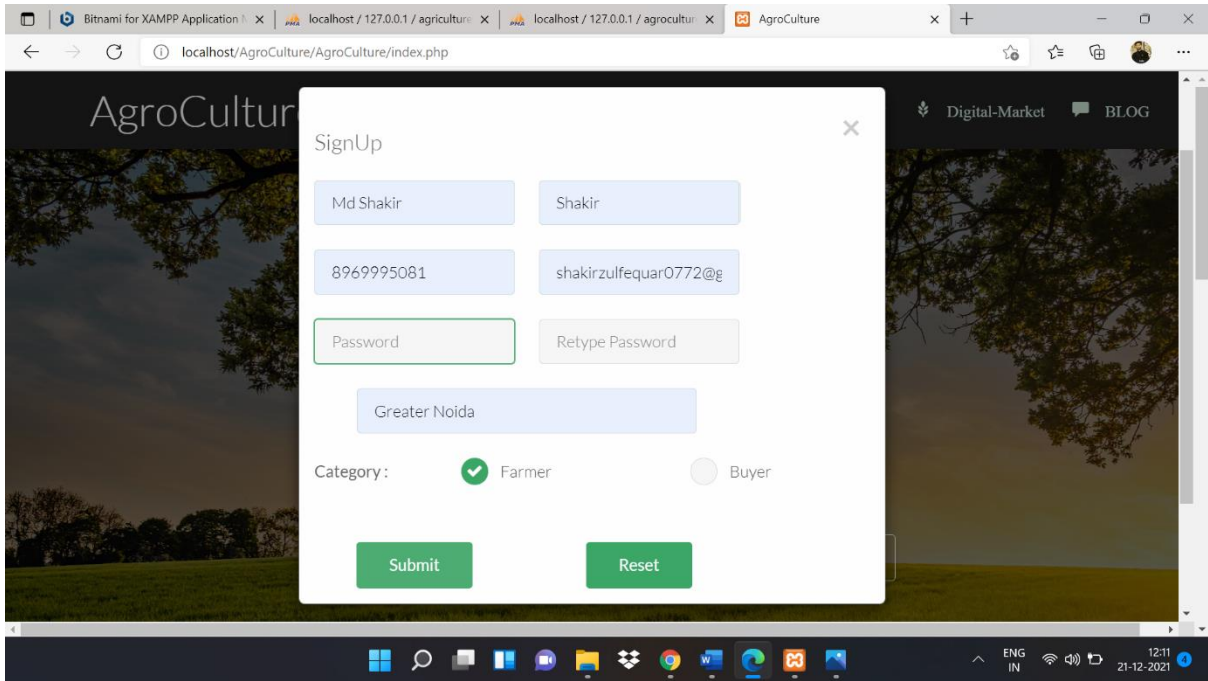
There are other six tests, which fall under special category. They are described below:

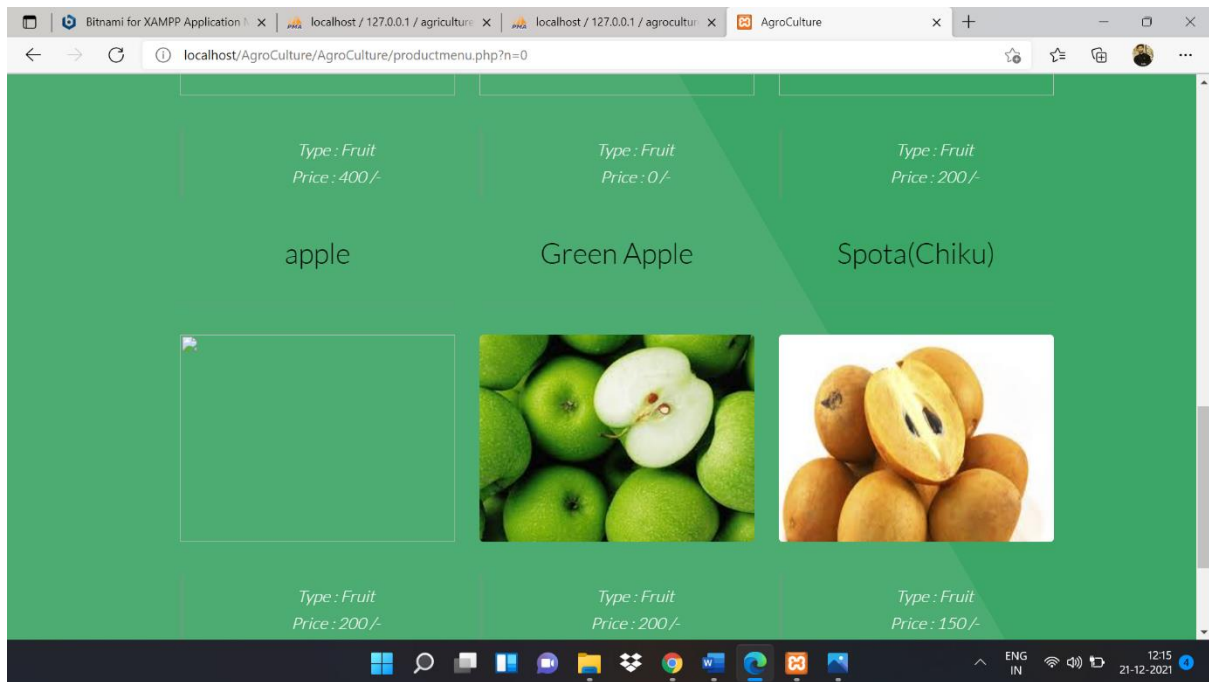
- **Peak Load Test:** It determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand. For example, test the system by activating all terminals at the same time.
- **Storage Testing:** It determines the capacity of the system to store transaction data on a disk or in other files.
- **Performance Time Testing:** it determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.
- **Recovery Testing:** This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss.
- **Procedure Testing:** It determines the clarity of documentation on operation and uses of system by having users do exactly what manuals request. For example, powering down system at the end of week or responding to paper-out light on printer.
- **Human Factors Testing:** It determines how users will use the system when processing data or preparing reports.

Screenshot of the Project









Database and Table Structure of the Project

--

-- Table structure for table `city`

--

```
CREATE TABLE `city` (
  `city_id` int(11) NOT NULL,
  `city_name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

--

-- Table structure for table `country`

--

```
CREATE TABLE `country` (
```

```
`country_id` int(11) NOT NULL,  
`country_name` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Table structure for table `crops`
```

```
--
```

```
CREATE TABLE `crops` (  
  `crops_id` int(11) NOT NULL,  
  `crops_title` varchar(255) NOT NULL,  
  `crops_month` varchar(255) NOT NULL,  
  `crops_description` text NOT NULL,  
  `crops_image` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Table structure for table `insecticides`
```

```
--
```

```
CREATE TABLE `insecticides` (  
  `insecticides_id` int(11) NOT NULL,  
  `insecticides_title` varchar(255) NOT NULL,  
  `insecticides_cost_range` varchar(255) NOT NULL,  
  `insecticides_description` text NOT NULL,  
  `insecticides_image` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

--

-- Table structure for table `pesticides`

--

```
CREATE TABLE `pesticides` (  
  `pesticides_id` int(11) NOT NULL,  
  `pesticides_title` varchar(255) NOT NULL,  
  `pesticides_cost_range` varchar(255) NOT NULL,  
  `pesticides_description` text NOT NULL,  
  `pesticides_image` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

--

-- Table structure for table `role`

--

```
CREATE TABLE `role` (  
  `role_id` int(11) NOT NULL,  
  `role_name` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

--

-- Table structure for table `state`

--

```
CREATE TABLE `state` (  
  `state_id` int(11) NOT NULL,  
  `state_name` varchar(255) NOT NULL,  
  `state_code` varchar(255) NOT NULL,  
  `state_image` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
`state_id` int(11) NOT NULL,  
`state_name` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Table structure for table `user`
```

```
--
```

```
CREATE TABLE `user` (  
  `user_id` int(11) NOT NULL,  
  `user_package_id` varchar(255) NOT NULL,  
  `user_shift_id` varchar(255) NOT NULL,  
  `user_level_id` varchar(255) DEFAULT NULL,  
  `user_username` varchar(255) DEFAULT NULL,  
  `user_password` varchar(255) DEFAULT NULL,  
  `user_name` varchar(255) NOT NULL,  
  `user_add1` varchar(255) NOT NULL,  
  `user_add2` varchar(255) NOT NULL,  
  `user_city` varchar(255) NOT NULL,  
  `user_state` varchar(255) NOT NULL,  
  `user_country` varchar(255) NOT NULL,  
  `user_email` varchar(255) NOT NULL,  
  `user_mobile` varchar(255) NOT NULL,  
  `user_gender` varchar(255) NOT NULL,  
  `user_dob` varchar(255) NOT NULL,  
  `user_image` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Coding related to developing software

Coding of crops.php

```
<?php
    include_once("includes/header.php");
    if($_REQUEST[crops_id])
    {
        $SQL="SELECT * FROM crops WHERE crops_id =
$_REQUEST[crops_id]";
        $rs=mysqli_query($con,$SQL) or die(mysqli_error($con));
        $data=mysqli_fetch_assoc($rs);
    }
    if($_SESSION['crops_details']['crops_level_id'] == 3) {
        $readonly = "readonly";
    }
?>

<div class="crumb">

</div>

<div class="clear"></div>

<div id="content_sec">
    <div class="col1">
        <div class="contact">
            <h4 class="heading colr">Crops Form</h4>
            <?php
                if($_REQUEST['msg']) {
                    ?>
                    <div class="msg"><?=$_REQUEST['msg']?></div>
                <?php
```

```
}
```

```
?>
```

```
<form action="lib/crops.php" enctype="multipart/form-  
data" method="post" name="frm_crops" onsubmit="return validateForm(this)">
```

```
<ul class="forms">
```

```
<li class="txt">Title</li>
```

```
<li class="inputfield"><input  
name="crops_title" type="text" class="bar" required  
value="<?=$data[crops_title]?>"/></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Months</li>
```

```
<li class="inputfield"><input  
name="crops_month" type="text" class="bar" required  
value="<?=$data[crops_month]?>"/></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Description</li>
```

```
<li class="textfield"><textarea  
name="crops_description" cols="" rows="6"  
required><?=$data[crops_description]?></textarea></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Photo</li>
```

```
<li class="inputfield"><input  
name="crops_image" type="file" class="bar"/></li>
```

```
</ul>
```

```
<div class="clear"></div>
```

```
<ul class="forms">
```

```
<li class="txt">&nbsp;</li>
```

```
value="Submit" class="simplebtn"></li>
```

```
value="Reset" class="resetbtn"></li>
```

```
</ul>
```

```
value="save_crops">
```

```
value="<?=$data[crops_image]?>">
```

```
value="<?=$data[crops_id]?>">
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div class="col2">
```

```
<?php if($_REQUEST[crops_id]) { ?>
```

```
<div class="contactfinder">
```

```
Image<h4 class="heading colr"><?=$data['crops_title']?>
```

```
<div></div><br>
```

```
</div>
```

```
<?php } ?>
```

```
</div>
```

```
<div class="col2">
```

```
<?php include_once("includes/sidebar.php"); ?>
```

```
</div>
```

```
</div>
```

```
<?php include_once("includes/footer.php"); ?>
```


Coding of crops-report.php

```
<?php
    include_once("includes/header.php");
    include_once("includes/db_connect.php");
    $SQL="SELECT * FROM crops";
    $rs=mysqli_query($con,$SQL) or die(mysqli_error($con));
    global $SERVER_PATH;
?>
<script>
function delete_crops(crops_id)
{
    if(confirm("Do you want to delete the crops?"))
    {
        this.document.frm_crops.crops_id.value=crops_id;
        this.document.frm_crops.act.value="delete_crops";
        this.document.frm_crops.submit();
    }
}
</script>
<div class="crumb">
</div>
<div class="clear"></div>
<div id="content_sec">
```

```

<div class="col1" style="width:100%">
<div class="contact">
    <h4 class="heading colr">Crops Reports</h4>
    <form name="frm_crops" action="lib/crops.php" method="post">
    <div class="static">
        <table style="width:100%">
            <tbody>
                <tr class="tablehead bold">
                    <td scope="col">Sr. No.</td>
                    <td scope="col">Image</td>
                    <td scope="col">Title</td>
                    <td scope="col">Month</td>
                    <td scope="col">Action</td>
                </tr>
                <?php
                $sr_no=1;
                while($data = mysqli_fetch_assoc($rs))
                {
                ?>
                <tr>
                    <td style="text-align:center; font-
weight:bold;"><?=$sr_no++?></td>
                    <td></td>
                    <td><?=$data[crops_title]?></td>
                    <td><?=$data[crops_month]?></td>

```

```
        <td style="text-align:center"><a
href="crops.php?crops_id=<?php echo $data[crops_id] ?>">Edit</a> | <a
href="Javascript:delete_crops(<?=$data[crops_id]?>)">Delete</a> </td>
```

```
    </tr>
```

```
    <?php } ?>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
<input type="hidden" name="act" />
```

```
<input type="hidden" name="crops_id" />
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<?php include_once("includes/footer.php"); ?>
```

Coding of insecticides.php

```
<?php
    include_once("includes/header.php");
    if($_REQUEST[insecticides_id])
    {
        $SQL="SELECT * FROM insecticides WHERE insecticides_id =
$_REQUEST[insecticides_id]";
        $rs=mysqli_query($con,$SQL) or die(mysqli_error($con));
        $data=mysqli_fetch_assoc($rs);
    }
    if($_SESSION['insecticides_details']['insecticides_level_id'] == 3) {
        $readonly = "readonly";
    }
?>

    <div class="crumb">

</div>

<div class="clear"></div>

    <div id="content_sec">
        <div class="col1">
            <div class="contact">
                <h4 class="heading colr">Insecticides Form</h4>
                <?php
                    if($_REQUEST['msg']) {
                        ?>
                        <div class="msg"><?=$_REQUEST['msg']?></div>
                        <?php
                            }
                    }
                ?>
            </div>
        </div>
    </div>
```

?>

```
<form action="lib/insecticides.php"
enctype="multipart/form-data" method="post" name="frm_insecticides"
onsubmit="return validateForm(this)">
```

```
<ul class="forms">
```

```
<li class="txt">Title</li>
```

```
<li class="inputfield"><input
name="insecticides_title" type="text" class="bar" required
value="<?=$data[insecticides_title]?>" /></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Cost Range</li>
```

```
<li class="inputfield"><input
name="insecticides_cost_range" type="text" class="bar" required
value="<?=$data[insecticides_cost_range]?>" /></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Description</li>
```

```
<li class="textfield"><textarea
name="insecticides_description" cols="" rows="6"
required><?=$data[insecticides_description]?></textarea></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Photo</li>
```

```
<li class="inputfield"><input
name="insecticides_image" type="file" class="bar" /></li>
```

```
</ul>
```

```
<div class="clear"></div>
```

```
<ul class="forms">
```

```
<li class="txt">&nbsp;</li>
```

```
value="Submit" class="simplebtn"></li>
```

```
value="Reset" class="resetbtn"></li>
```

```
</ul>
```

```
value="save_insecticides">
```

```
value="<?=$data[insecticides_image]?>">
```

```
value="<?=$data[insecticides_id]?>">
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div class="col2">
```

```
<?php if($_REQUEST[insecticides_id]) { ?>
```

```
<div class="contactfinder">
```

```
Image<h4 class="heading colr"><?=$data[insecticides_title]?>
```

```
<div></div><br>
```

```
</div>
```

```
<?php } ?>
```

```
</div>
```

```
<div class="col2">
```

```
<?php include_once("includes/sidebar.php"); ?>
```

```
</div>
```

```
</div>
```

```
<?php include_once("includes/footer.php"); ?>
```


Coding of insecticides-report.php

```
<?php
    include_once("includes/header.php");
    include_once("includes/db_connect.php");
    $SQL="SELECT * FROM insecticides";
    $rs=mysqli_query($con,$SQL) or die(mysqli_error($con));
    global $SERVER_PATH;
?>
<script>
function delete_insecticides(insecticides_id)
{
    if(confirm("Do you want to delete the insecticides?"))
    {
        this.document.frm_insecticides.insecticides_id.value=insecticides_id;
        this.document.frm_insecticides.act.value="delete_insecticides";
        this.document.frm_insecticides.submit();
    }
}
</script>
<div class="crumb">
</div>
<div class="clear"></div>
<div id="content_sec">
    <div class="col1" style="width:100%">
    <div class="contact">
        <h4 class="heading colr">Insecticides Reports</h4>
```



```

<form name="frm_insecticides" action="lib/insecticides.php"
method="post">
    <div class="static">
        <table style="width:100%">
            <tbody>
                <tr class="tablehead bold">
                    <td scope="col">Sr. No.</td>
                    <td scope="col">Image</td>
                    <td scope="col">Title</td>
                    <td scope="col">Cost Range</td>
                    <td scope="col">Action</td>
                </tr>
                <?php
                $sr_no=1;
                while($data = mysqli_fetch_assoc($rs))
                {
                    ?>
                    <tr>
                        <td style="text-align:center; font-
weight:bold;"><?=$sr_no++?></td>
                        <td></td>
                        <td><?=$data[insecticides_title]?></td>
                        <td><?=$data[insecticides_cost_range]?></td>
                        <td style="text-align:center"><a
href="insecticides.php?insecticides_id=<?php echo $data[insecticides_id]
?>">Edit</a> | <a
href="Javascript:delete_insecticides(<?=$data[insecticides_id]?>)">Delete</a> </td>

```

```
        </tr>
        <?php } ?>
    </tbody>
</table>
</div>
<input type="hidden" name="act" />
<input type="hidden" name="insecticides_id" />
</form>
</div>
</div>
</div>
<?php include_once("includes/footer.php"); ?>
```

Coding of pesticides.php

```
<?php
    include_once("includes/header.php");
    if($_REQUEST[pesticides_id])
    {
        $SQL="SELECT * FROM pesticides WHERE pesticides_id =
$_REQUEST[pesticides_id]";
        $rs=mysqli_query($con,$SQL) or die(mysqli_error($con));
        $data=mysqli_fetch_assoc($rs);
    }
    if($_SESSION['pesticides_details']['pesticides_level_id'] == 3) {
        $readonly = "readonly";
    }
?>
    <div class="crumb">
</div>
<div class="clear"></div>
    <div id="content_sec">
        <div class="col1">
            <div class="contact">
                <h4 class="heading colr">Pesticides Form</h4>
                <?php
                    if($_REQUEST['msg']) {
                        ?>
                        <div class="msg"><?=$_REQUEST['msg']?></div>
                    <?php
                        }
                </div>
            </div>
        </div>
    </div>
```

?>

```
<form action="lib/pesticides.php"
enctype="multipart/form-data" method="post" name="frm_pesticides"
onsubmit="return validateForm(this)">
```

```
<ul class="forms">
```

```
<li class="txt">Title</li>
```

```
<li class="inputfield"><input
name="pesticides_title" type="text" class="bar" required
value="<?=$data[pesticides_title]?>" /></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Cost Range</li>
```

```
<li class="inputfield"><input
name="pesticides_cost_range" type="text" class="bar" required
value="<?=$data[pesticides_cost_range]?>" /></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Description</li>
```

```
<li class="textfield"><textarea
name="pesticides_description" cols="" rows="6"
required><?=$data[pesticides_description]?></textarea></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Photo</li>
```

```
<li class="inputfield"><input
name="pesticides_image" type="file" class="bar" /></li>
```

```
</ul>
```

```
<div class="clear"></div>
```

```
<ul class="forms">
```

```
<li class="txt">&nbsp;</li>
```

```
value="Submit" class="simplebtn"></li>
```

```
value="Reset" class="resetbtn"></li>
```

```
</ul>
```

```
value="save_pesticides">
```

```
value="<?=$data[pesticides_image]?>">
```

```
value="<?=$data[pesticides_id]?>">
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div class="col2">
```

```
<?php if($_REQUEST[pesticides_id]) { ?>
```

```
<div class="contactfinder">
```

```
Image<h4 class="heading colr"><?=$data['pesticides_title']?>
```

```
<div></div><br>
```

```
</div>
```

```
<?php } ?>
```

```
</div>
```

```
<div class="col2">
```

```
<?php include_once("includes/sidebar.php"); ?>
```

```
</div>
```

```
</div>
```

```
<?php include_once("includes/footer.php"); ?>
```


Coding of pesticides-report.php

```
<?php
    include_once("includes/header.php");
    include_once("includes/db_connect.php");
    $SQL="SELECT * FROM pesticides";
    $rs=mysqli_query($con,$SQL) or die(mysqli_error($con));
    global $SERVER_PATH;
?>
<script>
function delete_pesticides(pesticides_id)
{
    if(confirm("Do you want to delete the pesticides?"))
    {
        this.document.frm_pesticides.pesticides_id.value=pesticides_id;
        this.document.frm_pesticides.act.value="delete_pesticides";
        this.document.frm_pesticides.submit();
    }
}
</script>
    <div class="crumb">
</div>
<div class="clear"></div>
    <div id="content_sec">
        <div class="col1" style="width:100%">
            <div class="contact">
                <h4 class="heading colr">Pesticides Reports</h4>
```

```

<form name="frm_pesticides" action="lib/pesticides.php"
method="post">

    <div class="static">

        <table style="width:100%">

            <tbody>

                <tr class="tablehead bold">

                    <td scope="col">Sr. No.</td>

                    <td scope="col">Image</td>

                    <td scope="col">Title</td>

                    <td scope="col">Cost Range</td>

                    <td scope="col">Action</td>

                </tr>

                <?php

                $sr_no=1;

                while($data = mysqli_fetch_assoc($rs))

                {

                ?>

                <tr>

                    <td style="text-align:center; font-
weight:bold;"><?=$sr_no++?></td>

                    <td></td>

                    <td><?=$data[pesticides_title]?></td>

                    <td><?=$data[pesticides_cost_range]?></td>

                    <td style="text-align:center"><a
href="pesticides.php?pesticides_id=<?php echo $data[pesticides_id] ?>">Edit</a> |
<a href="Javascript:delete_pesticides(<?=$data[pesticides_id]?>)">Delete</a> </td>

                </tr>

```



```
<?php } ?>
</tbody>
</table>
</div>
<input type="hidden" name="act" />
<input type="hidden" name="pesticides_id" />
</form>
</div>
</div>
</div>
<?php include_once("includes/footer.php"); ?>
```

Coding of user.php

```
<?php
    include_once("includes/header.php");
    if($_REQUEST[user_id])
    {
        $SQL="SELECT * FROM user WHERE user_id =
$_REQUEST[user_id]";
        $rs=mysqli_query($con,$SQL) or die(mysqli_error($con));
        $data=mysqli_fetch_assoc($rs);
    }
    if($_SESSION['user_details']['user_level_id'] == 3) {
        $readonly = "readonly";
    }
?>
<script>

jQuery(function() {
    jQuery( "#user_dob" ).datepicker({
        changeMonth: true,
        changeYear: true,
        yearRange: "-65:-10",
        dateFormat: 'd MM,yy'
    });
    jQuery('#frm_user').validate({
        rules: {
            user_confirm_password: {
                equalTo: '#user_password'
```

```

        }
    }
});
});
function validateForm(obj) {
    if(validateEmail(obj.user_email.value))
        return true;
    jQuery('#error-msg').show();
    return false;
}
</script>
<div class="crumb">
</div>
<div class="clear"></div>
<div id="content_sec">
    <div class="col1">
        <div class="contact">
            <h4 class="heading colr">User Registration</h4>
            <?php
            if($_REQUEST['msg']) {
            ?>
            <div class="msg"><?=$_REQUEST['msg']?></div>
            <?php
            }
            ?>
            <div class="msg" style="display:none" id="error-
msg">Enter valid EmailID !!!</div>

```

```
<form action="lib/user.php" enctype="multipart/form-data"
method="post" name="frm_user" onsubmit="return validateForm(this)">
```

```
<ul class="forms">
```

```
<li class="txt">Name</li>
```

```
<li class="inputfield"><input
name="user_name" type="text" class="bar" required
value="<?=$data[user_name]?>" <?=$readonly?>/></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">User Name</li>
```

```
<li class="inputfield"><input
name="user_username" type="text" class="bar" required
value="<?=$data[user_username]?>" <?=$readonly?>/></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Password</li>
```

```
<li class="inputfield"><input
name="user_password" id="user_password" type="password" class="bar" required
value="<?=$data[user_password]?>" <?=$readonly?>/></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Confirm Password</li>
```

```
<li class="inputfield"><input
name="user_confirm_password" id="user_confirm_password" type="password"
class="bar" required value="<?=$data[user_password]?>" <?=$readonly?>/></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
<li class="txt">Package</li>
```

```
<li class="inputfield">
```

```
<select name="user_package_id"
class="bar" required/>
```

```
        <?php echo  
get_new_optionlist("package","package_id","package_title",$data[user_package_id])  
; ?>
```

```
    </select>
```

```
  </li>
```

```
</ul>
```

```
<ul class="forms">
```

```
  <li class="txt">Shift</li>
```

```
  <li class="inputfield">
```

```
    <select name="user_shift_id"
```

```
class="bar" required/>
```

```
        <?php echo  
get_new_optionlist("shift","shift_id","shift_title",$data[user_shift_id]); ?>
```

```
    </select>
```

```
  </li>
```

```
</ul>
```

```
<ul class="forms">
```

```
  <li class="txt">Mobile</li>
```

```
  <li class="inputfield"><input
```

```
name="user_mobile" type="text" class="bar" required  
value="<?=$data[user_mobile]?>" /></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
  <li class="txt">Email</li>
```

```
  <li class="inputfield"><input
```

```
name="user_email" id="user_email" type="text" class="bar" required  
value="<?=$data[user_email]?>" onchange="validateEmail(this)" /></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
  <li class="txt">Date of Birth</li>
```

```
        <li class="inputfield"><input
name="user_dob" id="user_dob" type="text" class="bar" required
value="<?=$data[user_dob]?>" /></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
    <li class="txt">Address Line 1</li>
```

```
    <li class="inputfield"><input
name="user_add1" type="text" class="bar" required
value="<?=$data[user_add1]?>" /></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
    <li class="txt">Address Line 2</li>
```

```
    <li class="inputfield"><input
name="user_add2" type="text" class="bar" required
value="<?=$data[user_add2]?>" /></li>
```

```
</ul>
```

```
<ul class="forms">
```

```
    <li class="txt">City</li>
```

```
    <li class="inputfield">
```

```
        <select name="user_city" class="bar"
required/>
```

```
        <?php echo
get_new_optionlist("city", "city_id", "city_name", $data[user_city]); ?>
```

```
    </select>
```

```
</li>
```

```
</ul>
```

```
<ul class="forms">
```

```
    <li class="txt">State</li>
```

```
    <li class="inputfield">
```

```

class="bar" required/>
                                <select name="user_state"
                                <?php echo
get_new_optionlist("state","state_id","state_name",$data[user_state]); ?>
                                </select>
                                </li>
                                </ul>
                                <ul class="forms">
                                <li class="txt">Country</li>
                                <li class="inputfield">
                                <select name="user_country"
                                <?php echo
get_new_optionlist("country","country_id","country_name",$data[user_country]); ?>
                                </select>
                                </li>
                                </ul>
                                <ul class="forms">
                                <li class="txt">Photo</li>
                                <li class="inputfield"><input
name="user_image" type="file" class="bar"/></li>
                                </ul>
                                <div class="clear"></div>
                                <ul class="forms">
                                <li class="txt">&nbsp;</li>
                                <li class="textfield"><input type="submit"
value="Submit" class="simplebtn"></li>
                                <li class="textfield"><input type="reset"
value="Reset" class="resetbtn"></li>
                                </ul>

```

```
value="save_user">
        <input type="hidden" name="act"
value="<?=$data[user_image]?>">
        <input type="hidden" name="avail_image"
value="<?=$data[user_image]?>">
        <input type="hidden" name="user_id"
value="<?=$data[user_id]?>">
        </form>
    </div>
</div>
<div class="col2">
    <?php if($_REQUEST[user_id]) { ?>
        <div class="contactfinder">
            <h4 class="heading colr">Profile of
<?=$data['user_name']?></h4>
            <div></div><br>
            </div>
            <?php } ?>
        </div>
    <div class="col2">
        <?php include_once("includes/sidebar.php"); ?>
    </div>
</div>
<?php
    if($_SESSION['user_details']['user_level_id'] != 1)
    {
?>
    <script>
```



```
jQuery( "#user_level_id" ).val(3);
```

```
jQuery( "#user_level" ).hide();
```

```
</script>
```

```
<?php
```

```
    }
```

```
?>
```

```
<?php include_once("includes/footer.php"); ?>
```

Coding of login.php

```
<?php
    include_once("includes/header.php");
    if($_REQUEST[car_id])
    {
        $SQL="SELECT * FROM car WHERE car_id = $_REQUEST[car_id]";
        $rs=mysqli_query($con,$SQL) or die(mysqli_error($con));
        $data=mysqli_fetch_assoc($rs);
    }
?>

    <div class="crumb">
</div>
<div class="clear"></div>
    <div id="content_sec">
        <div class="col1">
            <div class="contact">
                <h4 class="heading colr">Login To Your
Account</h4>
                <div class='msg'><?=$_REQUEST['msg']?></div>
                <form action="lib/login.php" method="post"
name="frm_car">
                    <ul class="forms">
                        <li class="txt">Username</li>
                        <li class="inputfield"><input
name="user_user" type="text" class="bar" required /></li>
                    </ul>
                    <ul class="forms">
                        <li class="txt">Password</li>
```

```
        <li class="inputfield"><input
name="user_password" type="password" class="bar" required /></li>
```

```
</ul>
```

```
<div class="clear"></div>
```

```
<ul class="forms">
```

```
    <li class="txt">&nbsp;</li>
```

```
    <li class="textfield"><input
type="submit" value="Submit" class="simplebtn"></li>
```

```
    <li class="textfield"><input
type="reset" value="Reset" class="resetbtn"></li>
```

```
</ul>
```

```
    <input type="hidden" name="act"
value="check_login">
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div class="col2">
```

```
    <?php include_once("includes/sidebar.php"); ?>
```

```
</div>
```

```
</div>
```

```
<?php include_once("includes/footer.php"); ?>
```

Coding of login-home.php

```
<?php
    include_once("includes/header.php");
    if($_REQUEST[car_id])
    {
        $SQL="SELECT * FROM car WHERE car_id = $_REQUEST[car_id]";
        $rs=mysqli_query($con,$SQL) or die(mysqli_error($con));
        $data=mysqli_fetch_assoc($rs);
    }
?>

    <div class="crumb">
</div>

<div class="clear"></div>

    <div id="content_sec">
        <div class="col1">
            <div class="contact">
                <h4 class="heading colr">Farm Management
Dashboard</h4>
                <ul class="login-home">
                    <li><a href="crops.php">Add
Crops</a></li>
                    <li><a href="insecticides.php">Add
Insecticides</a></li>
                    <li><a href="pesticides.php">Add
Pesticides</a></li>
                    <li><a href="crops-report.php">Crops
Report</a></li>
                    <li><a href="insecticides-
report.php">Insecticides Report</a></li>
```

```
<li><a href="pesticides-  
report.php">Pesticides Report</a></li>
```

```
<li><a href="change-  
password.php">Change Password</a></li>
```

```
<li><a  
href="/lib/login.php?act=logout">Logout</a></li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
<div class="col2">
```

```
<?php include_once("includes/sidebar.php"); ?>
```

```
</div>
```

```
</div>
```

```
<?php include_once("includes/footer.php"); ?>
```

Coding of db_connection.php

```
<?php
global $con;
$con = mysqli_connect("127.0.0.1", "root", "", "farm_management_system");

if (!$con) {
    echo "Error: Unable to connect to MySQL." . PHP_EOL;
    echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;
    echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;
    exit;
}
?>
```

Coding of functions.php

```
<?php
session_start();
$SERVER_PATH = "http://127.0.0.1/farm_management_system/";
##Function for generating the dynamic options #####
function get_new_optionlist($table,$id_col,$value_col,$selected=0, $cond = 1)
{
    global $con;
    $SQL="SELECT * FROM $table WHERE $cond ORDER BY $value_col";
    $rs=mysqli_query($con, $SQL);
    $option_list="<option value="">Please Select</option>";
    while($data=mysqli_fetch_assoc($rs))
    {
        if($selected==$data[$id_col])
        {
            $option_list.="<option value='$data[$id_col]'
selected>$data[$value_col]</option>";
        }
        else
        {
            $option_list.="<option
value='$data[$id_col]'>$data[$value_col]</option>";
        }
    }
    return $option_list;
}
##Function for generating the dynamic options #####
function get_checkbox($name,$table,$id_col,$value_col,$selected=0)
```

```

{
    global $con;
    $selected_array=explode(",",$selected);
    $SQL="SELECT * FROM $table ORDER BY $value_col";
    $rs=mysqli_query($con, $SQL);
    $option_list="";
    while($data=mysqli_fetch_assoc($rs))
    {
        if(in_array($data[$id_col],$selected_array))
        {
            $option_list.="<input type='checkbox' value='$data[$id_col]'
name=".". $name."[]" id='$name' checked>$data[$value_col]<br>";
        }
        else
        {
            $option_list.="<input type='checkbox' value='$data[$id_col]'
name=".". $name."[]" id='$name'>$data[$value_col]<br>";
        }
    }
    return $option_list;
}
?>

```

System Analysis:

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Farm Management System to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The

system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Existing System of Farm Management System:

In the existing system the exams are done only manually but in proposed system we have to computerize the exams using this application.

- Lack of security of data.
- More man power.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials

Proposed System of Farm Management System:

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

- Security of data.
- Ensure data accuracy's.
- Proper control of the higher officials.
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required.

Data Dictionary:

This is normally represented as the data about data. It is also termed as metadata some times which gives the data about the data stored in the database. It defines each data term encountered during the analysis and design of a new system. Data elements can describe files or the processes.

Following are some major symbols used in the data dictionary

- = equivalent to
- + and
- [] either/ or
- () Optional entry

Following are some rules, which defines the construction of data dictionary entries:

1. Words should be defined to understand for what they need and not the variable need by which they may be described in the program .
2. Each word must be unique. We cannot have two definition of the same client.
3. Aliases or synonyms are allowed when two or more enters shows the same meaning. For example a vendor number may also be called as customer number.
4. A self-defining word should not be decomposed. It means that the reduction of any information in to subpart should be done only if it is really required that is it is not easy to understand directly.

Data dictionary includes information such as the number of records in file, the frequency a process will run, security factor like pass word which user must enter to get excess to the information.

Conclusion of the Project Farm Management System:

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points...

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objectives of the project.
- The description of Purpose, Scope, and applicability.
- We define the problem on which we are working in the project.
- We describe the requirement Specifications of the system and the actions that can be done on these things.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system.
- Finally the system is implemented and tested according to test cases.

Future Scope of the Project:

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add printer in future.
- We can give more advance software for Farm Management System including more facilities
- We will host the platform on online servers to make it accessible worldwide
- Integrate multiple load balancers to distribute the loads of the system
- Create the master and slave database structure to reduce the overload of the database queries
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of Farm and Crops. Also, as it can be seen that now-a-days the players are versatile, i.e. so there is a scope for introducing a method to maintain the Farm Management System. Enhancements can be done to maintain all the Farm, Crops, Crops, Insecticides, Pesticides.

We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them. In the last we would like to thank all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is developed there by underlining success of process.

Limitation of Project on Farm Management System

Although I have put my best efforts to make the software flexible, easy to operate but limitations cannot be ruled out even by me. Though the software presents a broad range of options to its users some intricate options could not be covered into it; partly because of logistic and partly due to lack of sophistication. Paucity of time was also major constraint, thus it was not possible to make the software foolproof and dynamic. Lack of time also compelled me to ignore some part such as storing old result of the candidate etc.

Considerable efforts have made the software easy to operate even for the people not related to the field of computers but it is acknowledged that a layman may find it a bit problematic at the first instance. The user is provided help at each step for his convenience in working with the software.

List of limitations which is available in the Farm Management System:

- Excel export has not been developed for Farm, Crops due to some criticality.
- The transactions are executed in off-line mode, hence on-line data for Crops, Insecticides capture and modification is not possible.
- Off-line reports of Farm, Pesticides, Crops cannot be generated due to batch mode execution.

References and Bibliography:

- Google for problem solving
- <http://www.javaworld.com/javaworld/jw-01-1998/jw-01-Credentialreview.html>
- Database Programming with JDBC and Java by O'Reilly
- Head First Java 2nd Edition
- <http://www.jdbc-tutorial.com/>
- Java and Software Design Concepts by Apress
- <https://www.tutorialspoint.com/java/>
- <http://www.javatpoint.com/java-tutorial>
- <https://docs.oracle.com/javase/tutorial/>
- <http://www.wampserver.com/en/>
- <http://www.JSP.net/>
- <http://www.tutorialspoint.com/mysql/>
- <http://d.apache.org/docs/2.0/misc/tutorials.html>