

**FARMERS RECORD MANAGEMENT SYSTEM OF MUNICIPAL
AGRICULTURE OFFICE, SAN MIGUEL, BOHOL**

**College of Technology and Allied Sciences
BOHOL ISLAND STATE UNIVERSITY
Zamora, Bilar, Bohol**

**CHYNNA MARIE H. CANO
LITO BOY G. OMUL
JOSHUA S. QUIROL
ANA MAE GRACE A. VICERE**

June 2022

FARMERS RECORD MANAGEMENT SYSTEM OF MUNICIPAL
AGRICULTURE OFFICE, SAN MIGUEL, BOHOL

A Thesis
Presented to the Faculty of the
College of Technology and Allied Sciences
BOHOL ISLAND STATE UNIVERSITY
Bilar Campus, Zamora, Bilar

In Partial Fulfillment
Of the Requirements for the Degree
In Bachelor of Science in Computer Science

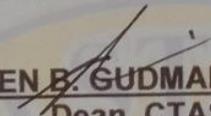
Chynna Marie H. Cano
Lito Boy G. Omul
Joshua S. Quirol
Ana Mae Grace A. Vicere

June 2022

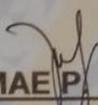
APPROVAL SHEET

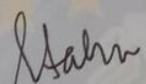
This thesis entitled "Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol" prepared and submitted by *Chynna Marie H. Cano, Lito Boy G. Omul, Joshua S. Quirol and Ana Mae Grace A. Vicere* in partial fulfillment of the requirements for the degree Bachelor of Science in Computer Science has been examined and recommended for acceptance and approval for oral defense.

THE THESIS COMMITTEE


ARLEN B. GUDMALIN, PhD
Dean, CTAS

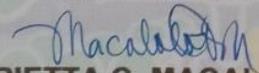

REX VINCENT D. TEJADA
Thesis Adviser

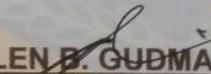

MAE P. BAS, PhD
Editor

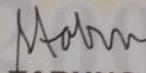

SHEILA G. TABUNO, MSTCS
Chairperson, DCoS

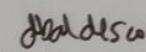
Approved by the Examining Panel during the Oral Examination conducted on May 31, 2022 with rating 1.5 

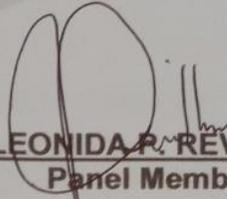
EXAMINING PANEL


MARIETTA C. MACALOT, PhD
Campus Director


ARLEN B. GUDMALIN, PhD
Dean, CTAS

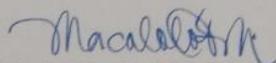

SHEILA G. TABUNO, MSTCS
Panel Expert


DANIE D. BALDESCO
Panel Member


LEONIDA P. REVILLA
Panel Member

Accepted and approved as partial fulfilment of the requirement for the degree Bachelor of Science in Computer Science.

May 31, 2022
Date of Oral Defense


MARIETTA C. MACALOT, PhD
Campus Director

ACKNOWLEDGMENTS

First of all, the researchers of this study would like to thank the Almighty God for His Presence at all times, and for giving the researchers knowledge, wisdom, perseverance, patience and determination to make all things possible.

The researchers wish to convey their sincerest gratitude to all who have given their time and effort to make this research successful.

Mr. Rex Vincent D. Tejada, thesis adviser, for the untiring effort in giving suggestions, advices, guidance and understanding which gave the researchers the idea on the completion of this study;

Mrs. Marcela P. Palma, the client, for accepting the researchers to conduct the study and for giving the idea and information needed;

Dr. Arlen B. Gudmalin, Dean of CTAS, for her supervision and suggestions for the improvement of the study;

Mrs. Sheila G. Tabuno, Chairperson of DCoS, and serve as panel expert during defense, for her supervision, support, and encouragement to the researchers;

Mrs. Leonida P. Revilla and **Mr. Danie Baldesco**, panellist during the defense, for giving and sharing us their ideas and sincere help towards realization of this work despite all the trials encountered;

Dr. Mae P. Bas, for sharing her time, effort and patience in checking the developers' manuscript;

To the parents, **Mr. Borromeo Cano and Mrs. Herminia Cano, Mr. Sofronio Omul and Mrs. Carmelita Omul, Mr. Adriano Quirol Sr. and Mrs. Ernesta Quirol, Mr. Cerino Vicere and Mrs. Flavia Vicere**, for their undying love and support and encouraging the researchers in every decisions made in life and for molding them to be a better person;

To the brothers and sisters of the researchers who gave moral and financial support;

Supportive friends and classmates for the encouragement, bits of advice, ideas, and opinions that they have shared;

And to all who were not mentioned but have contributed for the success of the study, thank you so much from the bottom of our hearts.

The Developers

TABLE OF CONTENTS

	Pages
TITLE PAGE	i
APPROVAL SHEET	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	x
LIST OF PREVIEW	xi
ABSTRACT	xiii
Chapter	
1 THE PROBLEM AND ITS SCOPE	
Rationale	1
Literature Background	2
THE PROBLEM	
Statement of the Problem	6
Scope and Delimitation	7
Significance of the Study	9
RESEARCH METHODOLOGY	
Development Framework	10
Conceptual Diagram of the Proposed Study	10
Block Diagram of the Proposed Study	10
Development Model and Approaches	11

Software Development Tools	13
Environment and Participants	14
Data Collection	14
Operational Definition of Terms	17

2 PRESENTATION OF FINDINGS, ANALYSIS AND INTERPRETATION OF DATA

Existing Operation and Processes	19
Event Specification	20
Need of the Existing Process	22
Proposed System Narrative	23
Use Case Diagram	24
Use Case Narrative	25
Class Diagram	31
Database Structure	32
Program Hierarchy	36
Functional Requirements	38
Non-Functional Requirements	40
Test Cases	40
Minimum Hardware Specification	49
Minimum Software Specification	50
Business Intelligence	50
Screen Layout	55
Economic Performance Evaluation	64

Testing and Evaluation	65
System Usability	65
3 SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS	
Summary of Findings	67
Conclusions	68
Recommendations	68
REFERENCES	69
APPENDICES	70
A. Letter of Intent	71
Letter of Implementation	72
Letter of Questionnaire Distribution	73
B. System Usability Questionnaire	74
Guide Questions for Interview	76
D. Source Code..	78
D. User's Manual	126
RESEARCHERS' BIODATA	134

LIST OF TABLES

Table		Pages
1	Respondents in the System Usability	15
2	Interpretation Guide of the System Usability	15
3	Use Case Narrative- Login to System as Office Head	26
4	Use Case Narrative- Login to System as Staff	26
5	Use Case Narrative- Add Farmers	27
6	Use Case Narrative- Add Crop Information	27
7	Use Case Narrative- Add Livestock Information	28
8	Use Case Narrative- Add Poultry Information	28
9	Use Case Narrative- Add Vegetable Information	29
10	Use Case Narrative- Add Fruit Trees Information	29
11	Use Case Narrative- Add Services Received	30
12	Use Case Narrative- Record Farmers Visit	30
13	Use Case Narrative- Generation of Reports	31
14	Data Structure- Log-In	33
15	Data Structure- User Account	33
16	Data Structure- Membership Form	33
17	Data Structure- Crop Growers	34
18	Data Structure- Livestock Raisers	34
19	Data Structure- Poultry Raisers	34
20	Data Structure- Vegetable Growers	35
21	Data Structure- Fruit Trees	35

22	Data Structure- Services Received	35
23	Data Structure- Farmers Visit	36
24	Initial Investment and Annual Operating Cost	64
25	System Usability Assessment Result	66

LIST OF FIGURES

Figure		Pages
1	Conceptual Diagram of the Study	10
2	Block Diagram of the Proposed System	11
3	Rapid Application Development Diagram (RAD)	12
4	Context Diagram of the Present System	20
5	Event List Diagram- Collection of Information	21
6	Event List Diagram- Data Recording	21
7	Event List Diagram- Generation of Reports	21
8	Top Level of the Present System	22
9	Use Case Diagram	25
10	Class Diagram of the System	32
11	Program Hierarchy	37

LIST OF PREVIEWS

Preview	Pages
1 List of Farmers	51
2 List of Crop Growers	51
3 List of Livestock Raisers	52
4 List of Poultry Raisers	52
5 List of Vegetable Growers	53
6 List of Fruit Trees Growers	53
7 Graphical Report of RSBSA Members	54
8 Graphical Report of Farmers by Commodity	54
9 Log-In Form	55
10 Dashboard	55
11 Farmers List View Form	56
12 Membership Form	56
13 Crop Growers View Form	57
14 Crop Growers Form	57
15 Livestock Raisers View Form	58
16 Livestock Raisers Form	58
17 Poultry Raisers View Form	59
18 Poultry Raisers Form	59
19 Vegetable Growers View Form	60
20 Vegetable Growers Form	60
21 Fruit Trees Growers View Form	61

22	Fruit Trees Growers Form	61
23	Services Received View Form	62
24	Services Received Form.....	62
25	Farmers Visit View Form	63
26	Farmers Visit Form	63

ABSTRACT

The study aimed to develop a record management system for the Municipal Agriculture Office (MAO) of San Miguel, Bohol. Currently, the establishment uses a semi-manual method in recording farmer's information which leads to duplication of records which affect consistency in data record, misplacement, losing of files, and time consuming in the retrieval of farmers' information. The identified problems and issues in the management of recording had led the developers to come up with a solution that could help lessen the problems encountered by the agriculture head and staff. Based on the identified needs, the record management system of the Municipal Agriculture Office was developed with the following features: offline mechanism, recording, data management, administration and report generation. The develop system is capable to handle offline record management with restriction in the input of records exclusive only for office head and staff. The results revealed that the user or respondents "strongly agree" that the system met its system usability, indicating the respondents' expectations were achieved. The developers concluded that the developed system provided a convenient way to improve records organization and management using the computerized system in a customized application program that is based on the actual procedure practiced. It is then highly recommended implementing the Farmers Record Management System to have a well-organized recording of farmer's information.

Chapter 1

THE PROBLEM AND ITS SCOPE

Rationale

One of the key advantages of computerization is the remarkable growth in office productivity. It makes work much easier, on whatever fields they belong, whether on science, research, or agriculture. For a local government office that intends to deliver fast and effective frontline public service to its constituents, the availability of crucial information relative to the functions of that specific office is the prime impetus for innovation on office methods and workflow efficiency.

Thus, an innovation that facilitates the storage, retrieval and management of farmers' information would decisively empower the office to respond to the challenges and needs towards more comprehensive information as stated in the study of Alburo et.al (2014).

Agriculture plays a chiefly role in economy as well as it is considered to be the backbone of economic system. Sufficient quality of information is a prerequisite for progress in all aspects of agriculture. With the rapid increase in the complexity of the technology of farming, there is now a recognized need to improve the record system of farmers. The Philippine Council for Agriculture, Aquatic, Forestry, and Natural Resources Research and Development acknowledges the inadequacy of the database apart from few and scattered research and development efforts on organic farming (Porciuncula, 2014)

Presently, Municipal Agriculture Office (MAO) in San Miguel uses semi-manual system in record management of the organization. According to the head of office and staffs of the Municipal Agriculture Office (MAO), the semi-manual method in recording and monitoring of its farmers' records had resulted in the following problems such as redundancy of records, length and time-consuming, difficulty in accessing and reviewing of documents of farmers in the monitoring of all data, and requires more manpower.

With the problems identified, the developer had identified a computer-based solution that would provide well-organized forms for data processing, recording, and monitoring of all data and make sure that it is accessible at all times. Thus, the Farmers Record Management System was developed.

Literature Background

Article XIV, Section 10 of the Philippine Constitution states that:

“Science and Technology are essential for national development and progress. The state shall give priority to research and development, invention, innovation, and their utilization; and to science and technology education, training, and services. It shall support indigenous appropriate, and self-reliant scientific and technological capabilities, and their application to the country’s productive systems and national life.”

According to this Article, the state shall give priority to adapting Science and Technology and their utilization so that its establishment can easily cope up advancement and changes. The researchers develop a Farmers Record Management System for the San Miguel Municipal Agriculture Office to provide convenience in terms of managing the record of farmers.

Moreover, Republic Act No. 8435, known as “Agriculture and Fisheries Modernization Act of 1997”, states that:

Agriculture and Fisheries Modernization Act of 1997(AFMA) aims to transform the agriculture and fisheries sectors to technology-based, advance and competitive industry; ensure that the small farmers and fisherfolk have equal access to assets, resources and services; guarantee food security; encourage farmers and fisherfolk groups to bond together for more bargaining power; strengthen people’s organizations, cooperatives and non-government organizations by enhancing their participation in decision-making; pursue an aggressive market-driven approach to make the products more competitive in the market; stimulate further processing of agricultural products and make it more marketable; and implement policies that will invite more investors to establish business in the country. (De Venica 1997)

This act aims to modernize Philippine agriculture in order for the country to compete on a global scale. Such facilities are outfitted with the adoption of computer aided record management is aligning to the program. The creation of this system would provide the farmers of Bohol with the storage space they require.

The following related theory are taken from the books and online sources that were used as the foundation for the creation of the application system:

The database management theory is needed to accomplish the goal. Hence, the following theories below are used as a foundation for the creation of this study. The first theory is Edgar F. Codd’s Rule 7 for Relational Database Management System which is the *High-level insert, update, and delete rule*. This rule explains that a system must support set-at-a-time insert, update, and delete operators. This means that data can be retrieved from a relational database in sets constructed of data from multiple rows and/or multiple tables. This rule states that

the insert, update, and delete operations should be supported for any retrievable set rather than just for a single row in a single table.

The second theory is on Edgar F. Codd's 12 Commandments for the Relational Database Management System. Rule number 2 states that all data must be accessible. This rule is known as the guaranteed access rule which is essentially a restatement of the fundamental requirement for primary keys.

There are some related studies that are available and used as references which are implemented in different organization and institution. Here are some examples:

1. Scholarship Database Management System in Institute of Business Administration, University of the Punjab, Lahore, Pakistan (Dabeer, 2017). This system focuses on the management of data of scholars in the said university. The system provides a database for adequate data storing which our system also provides, we only differ on the type of data we manage.
2. Computerized Record Management System of one National High School in the Philippines (Danlog, 2017). The system is designed to provide efficient and accurate recording of records, to maintain and secure the student's records, easy retrieval of student records, and to lessen the workload of the staffs. The system we developed only difference is that they manage student records.

3. Record Management System at the Municipal Agricultural Services, Salcedo, Eastern Samar (Alburo et al., 2014). This study aimed to design, develop and test the Farmers Record Management System to help the Municipality specifically the Office of the Municipal Agricultural Services (OMAS) to come up with a form of an information management system that is effective in collection, storage, management, processing, and generating of pertinent information related to farmers, agriculture, and livestock. It is somewhat the same as our system. They differ in a way that we cover poultry raisers and vegetable growers in our recording process.
4. Farmers Information and Technology Services (FITS) Management System of Carmen, Bohol (Alcala et al., 2018). This study aims to address problems and issues in the management of recording and inventory. The system has the following features: recording, data management, acquisition of inventory, and generation of reports. The system we developed also has the following features as mentioned above, the only difference is that they focused more on services while we focused on farm-related information of farmers and also, we provide dashboard feature in which the summary of total farmers, crop and vegetable growers, livestock and poultry raisers can be found.
5. The Database System of Rice Generic Resources of Bohol Community Seed bank (Galia, 2014). This study aimed to develop and design a Database System for Rice Generic Resources of the Bohol Community Seed bank. The modules of this study include search keys and generating

reports. It will help for easy access of rice information, security of the system's confidentiality & retrieving a huge amount of data. The databases hold all the information about the rice variety. The system we developed also has the following feature which is to generate reports as mentioned above but we have additional feature which is the dashboard feature in which the summary of total farmers, crop and vegetable growers, livestock and poultry raisers can be found.

The existence of these systems and theories mentioned above served as the foundation and conceptualization of the design and development. This helped met the specification and features needed in the implementation of the Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol.

THE PROBLEM

Statement of the Problem

The main purpose of the study was to develop a system that can cater the farmers' record management in Municipal Agriculture Office, San Miguel.

Specifically, the researchers sought to answer the following questions:

1. What is the current process of the farmers' recording management in San Miguel Municipal Agriculture Office?
2. What are the problems encountered by the office head and staff of San Miguel Municipal Agriculture Office in managing records?

3. What is the possible solution to be developed in order to solve the problems encountered?
4. What is the perceived level of acceptability of the developed system by the target users?

The proposed system is to be called Farmers Record Management System and it will integrate the following processes:

1. Implement an offline system that will help the staff in recording farmers' information
2. Design and implement the following modules:
 - a. recording;
 - b. data management;
 - c. administration; and
3. Implement business intelligence technique for decision-support to the office

Scope and Delimitation

This study was conducted at Municipal Agriculture Office (MAO) located in San Miguel, Bohol. It focused on the computerization of recording of the farmers' information of the Municipal Agriculture Office. The system developed was limited only to the MAO Head and Staff who can access to the system. Specifically, the system includes the following features:

1. **Offline Mechanism** - This feature facilitates an offline recording system by providing modular components and methodology, and a convenient interface that match the needs of the office.
2. **Recording** - This module provides recording activities such as recording new farmers, farmers' profile, agricultural commodities, services received and farm-related information of farmers.
3. **Data Management** - Provides the management of data which involves the process of updating, searching, viewing, and retrieval of information of the farmers.
4. **Administration** - This function provides offline administrative tools for the system maintenance, system configuration and facilitates management of the users, privileges, and security. Requires user login to avoid unauthorized personnel manipulate the data.
5. **Reporting** - Generate customized and manageable evaluation reports that will be used as the basis for statistical information needed for assessment and dissemination. It also generates a tabular information of total farmers, crop and vegetable growers, and livestock and poultry raisers.

The study was limited only to the standard operations and procedures in the Municipal Agriculture Office (MAO) including recording of data of farmers, updating data of farmers and generate reports. The users of the system are limited to the office head and staff of the MAO.

Significance of the Study

The study of the Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol aimed to develop a repository of all data related to farmers that will help the staff manage farmers' information well. Furthermore, the study will benefit the following:

Office Head. The head is the one who managed the office, approves the reports and oversee all the processes of the system that made in the office. The overall management of the office is made easier through customized recording system, it will provide a fast and easy way of generating reports and would serve as a repository system providing information that is needed.

Staff. The one who is in-charge and responsible in storing, updating and retrieving farmers' information (crop, livestock, poultry, High Valued Crop Development Program (HVCDP, and fruit trees). The system would be useful in managing farmers' record and giving reports in a more organized and accurate manner.

Future Researchers. The study is beneficial to the researchers because they will be able to apply some knowledge and it may help them improve their communication and thinking skills. They will learn the value of cooperation and hard work which are required for achieving goals.

RESEARCH METHODOLOGY

Development Framework

Figure 1 presents the procedure, technique, and a set of rules that analyze the principles to guide inquiry in a particular field. It also covers the arrangement of the basic functionality of the system that represents the person involved in using and maintaining the system. The system presents the model of the study that follows the principles of input-process-output. These inputs are coming from the staff and office head. The processes include recording, data management, administration, and reports. The output provides decision support to Office Head.

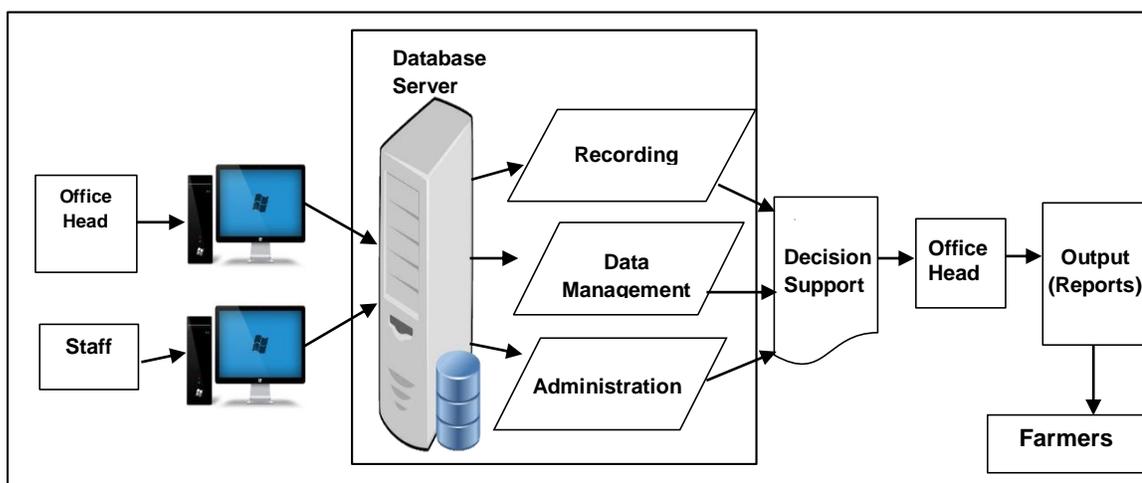


Figure 1. Conceptual Diagram of the System

Block Diagram

Figure 2 shows the block diagram of the Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol. It covers the specification of the basic functionality of the system that represents with the work of the staff and the office head who will manage the records and generate reports.

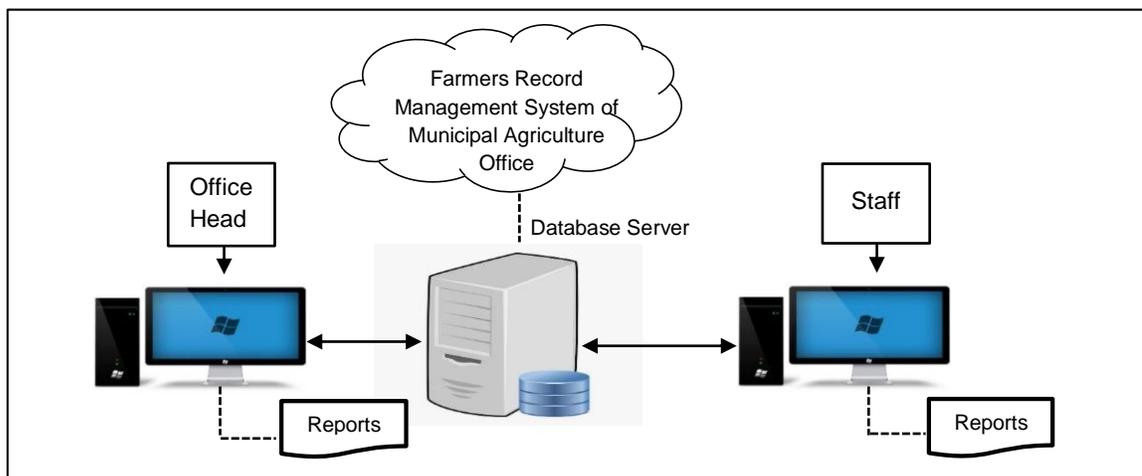


Figure 2. Block Diagram of the System

Development Model and Approaches

The methodology used by the developers was Rapid Application Development (RAD) which has four phases to be followed. It is a software iterative development methodology that involves techniques like software prototyping. Prototyping is suitable for gathering functional requirements if the client does not know what they want.

During the analysis and quick design phase, researchers identify the development's goals as well as current and potential problems that must be addressed. The researchers interviewed the respondents, have a document review as benchmark activity of the forms used and reports generated. They will also observe on how the activity of the office was done specifically in recording, monitoring, and generation of reports to be able to come up with a design of the prototype of the system developed.

After gathering the data needed, prototyping begins by designing tables and some fields need by the system. The client collaborates with the developer as well as present the system to the office head and staff to make sure that the needs are met at every stage of the design process. This was repeatedly done and tested to ensure that the result is correct and follows the procedure.

Next is the testing phase, it follows extensive prototyping and cutting-edge design, the beta system is presented to the office head and staff to ensure that everything is running fine and the user tests the system whether it suits their expectations. For the system usability, the developers provide guide questions.

The final stage is implementation, in which the completed program is launched. The office head and staff are allowed to use the newly developed system. During this stage, all the final changes are made, data conversion, testing and user-training were all done by the developers. The finalization was completed.

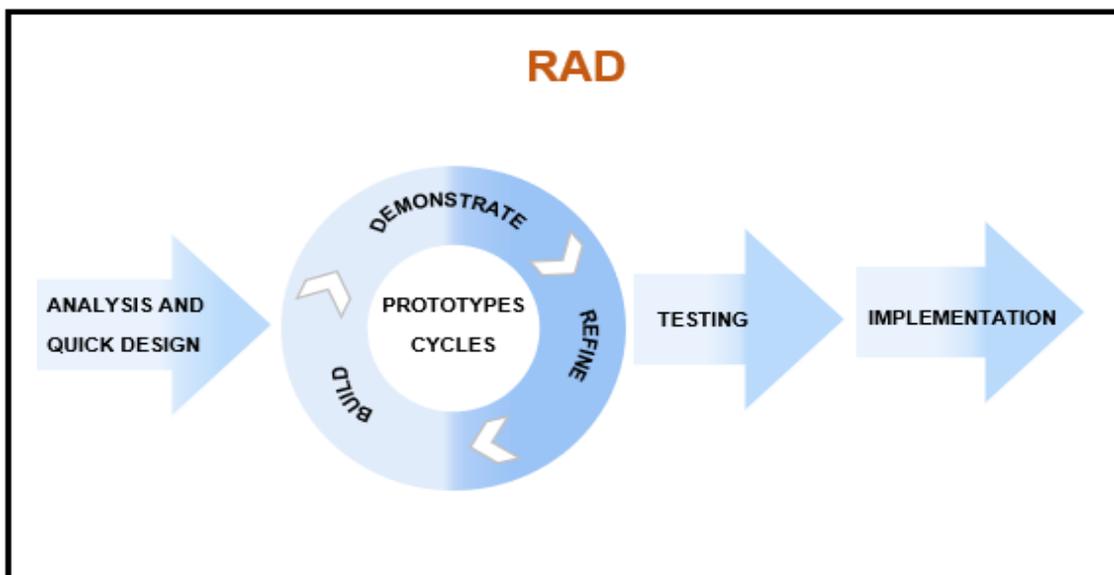


Figure 3. Rapid Application Development Model (RAD) Diagram

The following were the tools used during the analysis and design phase of Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol.

1. **Adobe Photoshop CS6** - is a graphics editing program developed and published by Adobe Systems, a leading paint program from Adobe System. It is used for designing the interface of the proposed system software. This will be used by the researchers in designing and editing the desired theme.
2. **WAMP**- a Windows web development environment tool that allows you to create web applications with Apache, PHP, and MySQL databases that can use to create an application on online viewing and database of the developed system.
3. **Crystal Report** - used in the system to make, generate and view reports of requested information of the farmers.
4. **Microsoft Visual Studio 2012** – an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. This will be used in the database of the Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol.
5. **Software Development Kit (SDK Or Devkit)**- it allows developers to add functionality to their applications and enable them to build the standard components of their apps easier and faster.

Environment and Participants

This study was conducted at the Municipal Agriculture Office, Municipality of San Miguel. It is located at Poblacion, San Miguel, Bohol. It is 50.3 km away from BISU-Bilar Campus. As of March 31, 2022, MAO have 4,901 registered farmers in the municipality. The study's participants were the Office Head who have the right to access the system and can use it to perform tasks such as recording farmers' information, updating information, and generating reports.

The second group of participants were the staff, who have mid-level access to the system and can record farmers' information, update information and request other details to the office head for permission. The farmers are the final set of participants since they were directly involved in using the system by inputting their information into the system. These participants were personally interviewed by the researchers to gather the exact data needed in the study.

Data Collection

The researchers made a letter of approval and asked permission from the Municipal Agriculture Office to conduct a study. Personal interviews, document reviews and actual observations were conducted to gather the needed information. The personal interview was conducted through guide questions where respondents are asked about the operation and process involved in acquiring of information which includes the problems that have been encountered.

In the document review, soft and hard copy documents been reviewed by the researchers to understand the structures of the data and for the possible features to be added. Functionality evaluation were conducted through a system usability testing survey with the target client. It is done to quantify the effectiveness of the developed system. There were 5 respondents in the system usability. It includes 1 Office Head, 3 Staff and 1 IT Expert. Table 1 shows the Summary of Respondents in the System Usability test.

Table 1

Summary of Respondents in the System Usability

N= 5

Respondents	Frequency
Office Head of MAO	1
Staff	3
IT Expert	1

Table 2 shows the interpretation of the results used for system usability. In the system usability, the rating was done based on the System Usability Guidelines developed by MIT Information Services Technology.

Table 2

Interpretation Guide of the System Usability

Weight	Range	Description	Interpretation
7	6.4 – 7.0	Strongly Agree	The respondents strongly believe and are confident that the system is very usable.
6	5.5 – 6.3	Agree	The respondents strongly believe and are confident that the system is very useful

5	4.6 – 5.4	Tend to Agree	The respondents tend to agree that the system is usable.
4	3.7 – 4.5	Neither Agree or Disagree	The respondents are neutral in trusting that the system is usable.
3	2.8 – 3.6	Tend to Agree	The respondents tend not to trust the system is usable.
2	1.9 – 2.7	Disagree	The respondents believe that the system is not usable
1	1.0 – 1.8	Strongly Disagree	The respondents strongly confident that the system is not usable.

To determine the general acceptability of the system, the average weighted mean or the weighted mean score was computed to evaluate and assess the system acceptability level using the following formula:

$$WMS = \frac{1f_1 + 2f_2 + 3f_3 + 4f_4 + 5f_5}{n}$$

Where:

WMS=Weighted Mean Score

f^1 = frequency of respondents given a rate of 1

f^2 = frequency of respondents given a rate of 2

f^3 = frequency of respondents given a rate of 3

f^4 = frequency of respondents given a rate of 4

f^5 = frequency of respondents given a rate of 5

n = total number of respondents

1, 2...5 = constant (rating to the service provided)

The range of the interpretative guide for usability was computed by getting the interval value. The table above shows the interpretative guide that was used to describe the usability of the system.

OPERATIONAL DEFINITION OF TERMS

To facilitate comprehension of the concepts used in this study, the following terms are defined.

Database. It is a software system that is used to manage and organized the collection of data for saving and retrieving purposes. It is also a collection of information of the users that is organized so that it can be easily accessed, managed, and updated.

Farmers. The person who engaged in organic agriculture, raising living organisms for food or raw materials. They are the source of information and that the staff needs to collect.

Municipal Agriculture Office (MAO). It refers to the office, the office head and staff workplace who will use the developed Farmers Record Management System.

Office Head. The person who monitor and oversee the whole system. A person who has the rights to log in the entire content of the developed system. The one who can monitor and generates report.

Staff. The one who is in-charge in storing, retrieving students' information and generation of graphical and tabular reports of the Office Head. The staff is the one who facilitates and entertains the farmers' queries.

Software Development Kit (SDK or Devkit). It is a set of software development tools which allows the creation of applications for a certain software package, software framework, hardware platform, computer system or similar development platform.

Farmers Record Management System of Municipal Agriculture Office. It is a system design created to provide a convenient recording process of farmers' information in Municipal Agriculture Office.

Chapter 2

PRESENTATION OF FINDINGS, ANALYSIS AND INTERPRETATION OF DATA

Existing Operation and Processes

The Municipal Agriculture Office (MAO) is composed of staff and office head which accountable for recording and monitoring the farmers' information inside the office. The office is still using the manual process of recording and tracking the information of a farmer in the Municipality of San Miguel. These are the following present process that takes place.

A. Collection of Information Process

Every household is surveyed by the assigned Municipal Agriculture Staff, who records farmers profile and farm related information. The staff provides membership form for farmers to be filled up. All this data is submitted to the assigned staff in the office for recording and keep in the farmers' record book.

B. Data Recording Process

The Municipal Agriculture Office Head receives the information from the staff assigned in collecting the data. It will be recorded on the record book first and transfer it to Microsoft excel. The staff submit a summary of data to the Office Head. This data will be needed when the Provincial Agriculture Office ask for reports and it will be retrieved when needed.

C. Generation of Reports

The staff of MAO San Miguel will do report about the list of members from the excel file. Category reports, crops and livestock, farmers' profile, farmer's related information based on the list of recorded members of MAO San Miguel and

other relevant information which are related to the organization are extracted manually from the excel file available. Manually scanning one – by – one of the documents in order to come up with the desire output. The in-charge personnel submit all the reports to the Municipal Agriculture Office Head.

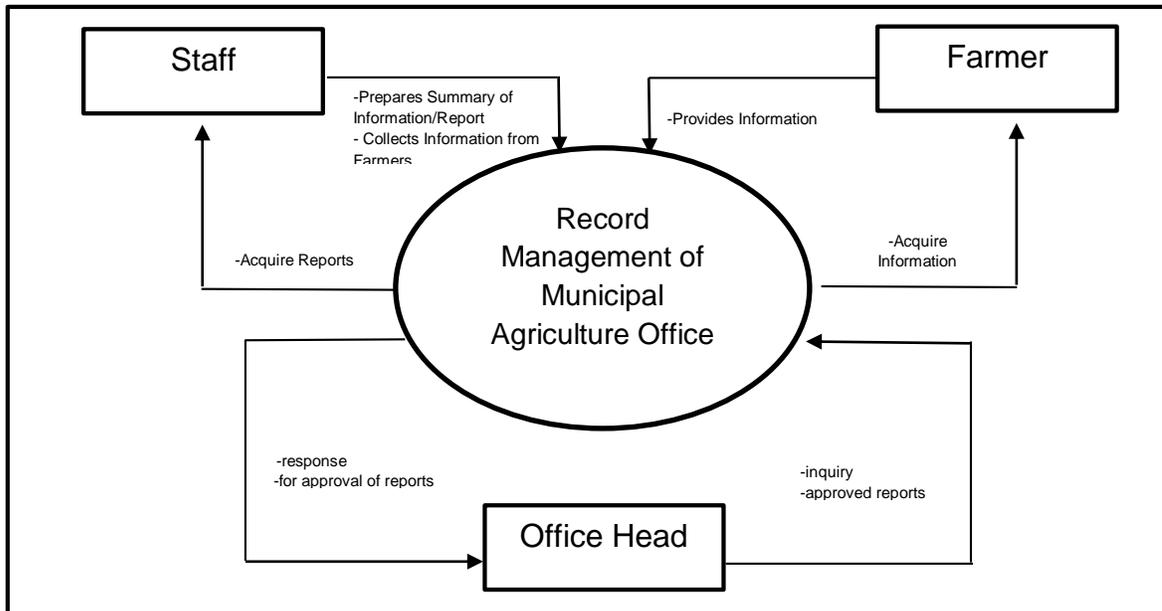


Figure 4. Context Diagram of the Present System

Event Specification

Event List

1. Collection of Information Process
2. Data Recording Process
3. Generation of Reports

Event List Diagrams

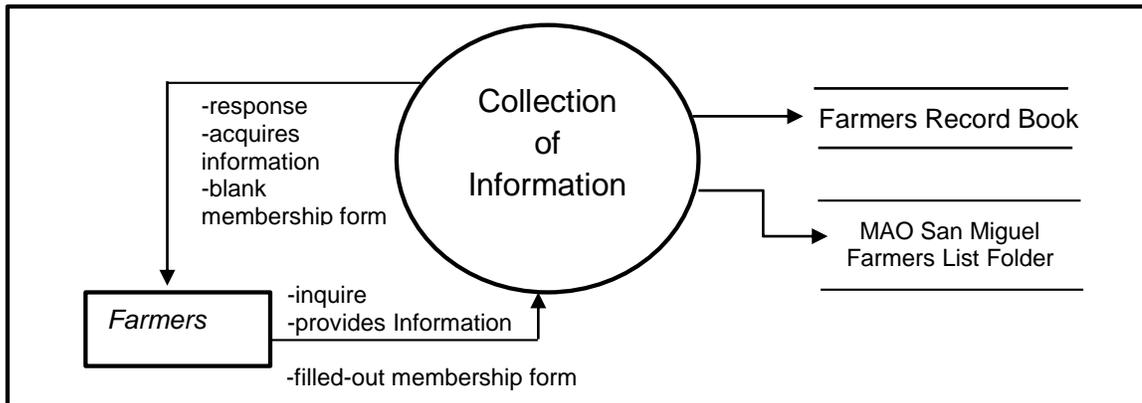


Figure 5. Collection of Information Process (Event 1)

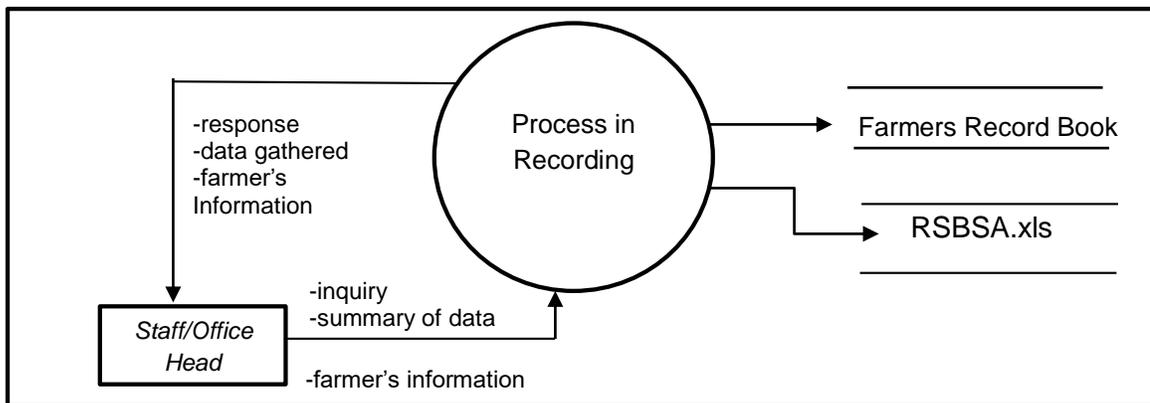


Figure 6. Data Recording Process (Event 2)

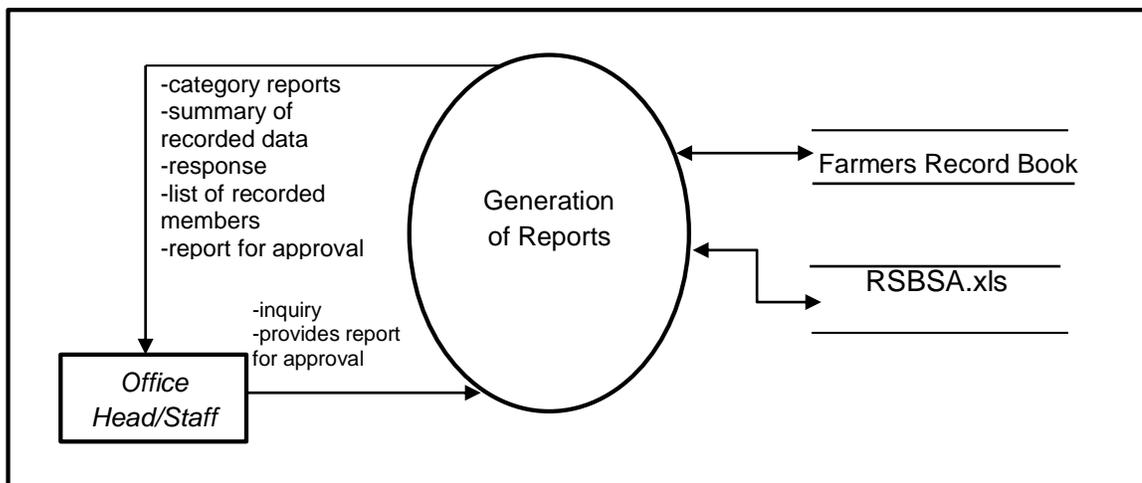


Figure 7. Generation of Reports (Event 3)

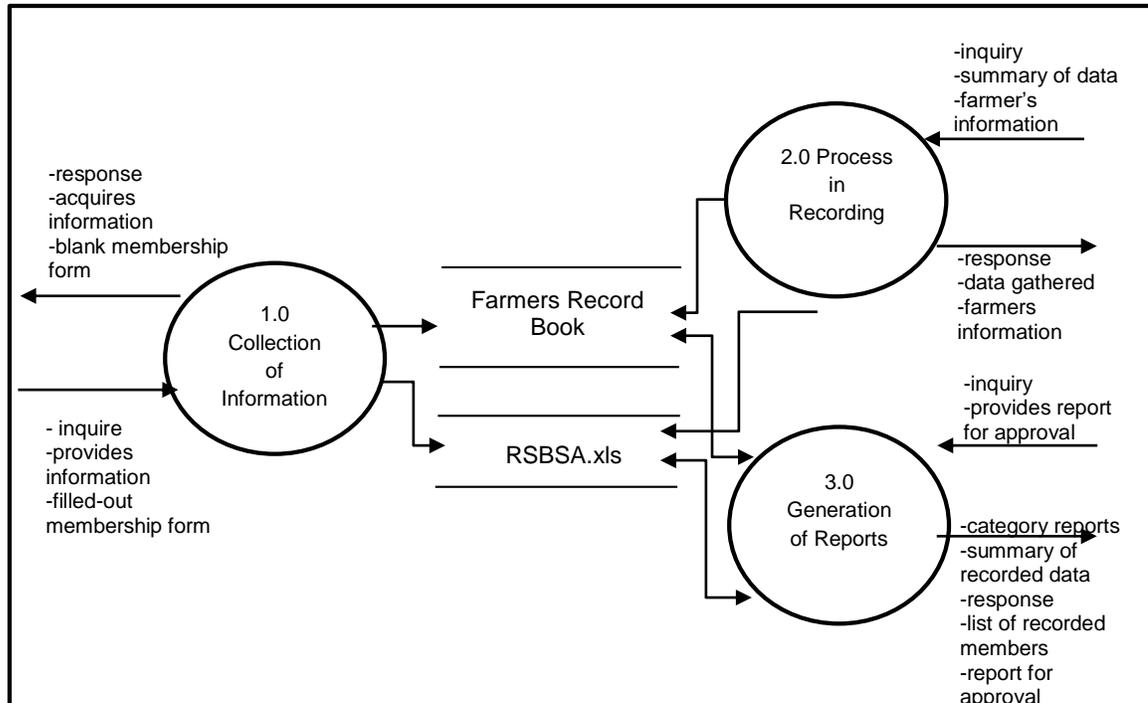


Figure No.8 Top Level of the Present System

Needs of the Existing Operation

The Municipal Agriculture Office (MAO) San Miguel current system relies on a semi-manual method which has need to enhance. The following are the management's needs based on observation and analysis.

1. Computerized and organized system to manage farmers records, and provides faster recording and reduce processing time when retrieving records, as well as to avoid confusion and make it easier to generate reports in a timely and efficient manner.
2. Provide a better data consistency and mechanism in the monitoring of each division as well as collating data for report purposes.

3. Improve the system security and consistency of records to provide integrity of Farmers' Data.

Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol

The researchers gathered all of the data and information about the current system and developed the Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol -a computerized system that processes farmers' information, records them, and generates the required reports.

A. Administration

In the administration, the authorized person should login first to access their responsible module. There are two classifications of accounts: MAO Head and the staff. The office head can fully access the system and serve as the administrator of the system. To access their module, they must enter their user's name and password first into the login form.

B. Acquisition

In the acquisition process, the flow of recording the farmers would start to the Municipal Agriculture Office. The staff that been assigned in gathering the information will survey every household in the municipality. After getting the information, it will be submitted to the assigned staff in the office for recording.

C. Record Management

The MAO staff will input the information of farmers and update this as required. These will be stored and maintain in one table to ensure data integrity.

The office head and staff were the one who will be responsible in keeping, retrieving the records, and generating reports.

D. Report Generation

The system generates tabular and graphical form of report recommendation for proper action and immediate response. This includes Master List Report from different commodities. The system also provides graphical reports generation like Farmers' yearly report. Under the reports are the list of registered members of MAO San Miguel, the information of farmers, and the product/ commodity they've owned. In graphical reports the staff can determine the graph by barangay's and by type of products/commodity.

Use Case Diagram

UML Use case diagrams is usually referred to as behaviour diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). A use case represents a functionality of the system from the viewpoint of the user and describes the goals of their use. Each use case should provide some visible and valuable result to the actors or other stakeholders of the system.

As shown in the diagram, the office head and staff have full access to the system and can record farmers' information, update and manage farmers' record, and generates report.

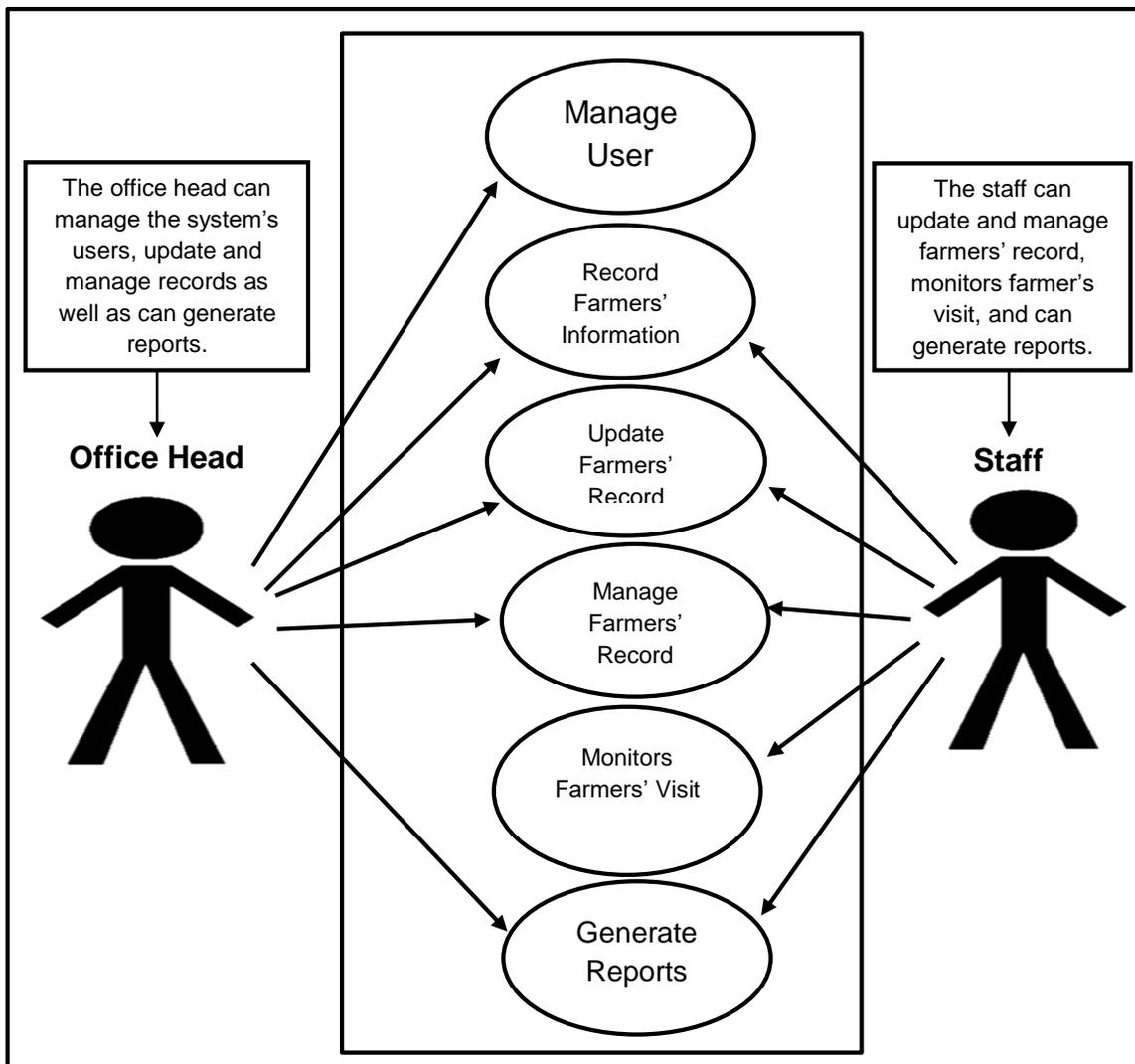


Figure 9. Use Case Diagram of the System

Use Case Narrative

Use case narrative describing a use case that requires both frame context of the use case and represents the dialog between the user (actor or use case) and the use case. In every use case narrative, there are Pre-conditions, Process, and Post-conditions.

Table 3. Use Case Narrative - Login to the system as Office Head

Log-in to the system (UC1)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Context	To access to the system and access the record of farmers.
Primary Actors	Office Head
Stakeholders	Office Head wants to view record of farmers.
Preconditions	None
Triggers	The Office Head input username and password.
Success Guarantee	The Office Head successfully log-in to the system.
Scenarios:	<ol style="list-style-type: none"> 1. The office head access to the system. 2. The log-in form for the office head will show. 3. The office head input username and password. 4. The system checks for the username and password if it matches the database record. 5. The system will go to the main form.

Table 4. Use Case Narrative - Login to the system as Staff

Log-in to the system (UC2)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To access to the system and access the record of farmers.
Primary Actors	Staff
Stakeholders	Staff wants to view record of farmers.
Preconditions	None
Triggers	The Staff input username and password.
Success Guarantee	The staff successfully log-in to the system.
Scenarios:	<ol style="list-style-type: none"> 1. The staff access to the system. 2. The log-in form for staff will show. 3. The staff input username and password. 4. The system checks for the username and password if it matches the database record. 5. The system will go to the main form.

Table 5. Use Case Narrative - Add farmers

To add farmers (UC3)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To add new farmers and its personal information.
Primary Actors	Office Head and Staff
Stakeholders	Office Head, Staff who want to add new farmers.
Preconditions	None
Triggers	The Office Head and Staff will receive a form that been filled up by farmers to be encoded.
Success Guarantee	Added the farmers successfully.
Scenarios:	<ol style="list-style-type: none"> 1. The farmers will provide his/her information. 2. The office head and staff will access the sub menu "file" and select membership form. 3. The staff encoded it to the systems' membership form. 4. The system will go to the crops growers form.

Table 6. Use Case Narrative - Add farmers' crop information

To add farmers' crop information (UC4)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To add new data for farmers crop information.
Primary Actors	Office Head and Staff
Stakeholders	Office Head, Staff who want to add new farmers' crop data.
Preconditions	None
Triggers	The Office Head and Staff will receive the records that been filled up by farmers that to be encoded.
Success Guarantee	Added the farmers' crop information successfully.
Scenarios:	<ol style="list-style-type: none"> 1. The farmers will provide his/her information. 2. The staff encoded it to the systems' crop growers' form. 3. The system will go to the livestock form.

Table 7. Use Case Narrative - Add farmers' livestock information

To add farmers' livestock information (UC5)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To add new data for farmers livestock information.
Primary Actors	Office Head and Staff
Stakeholders	Office Head, Staff who want to add new farmers' livestock data.
Preconditions	None
Triggers	The Office Head and Staff will receive the records that been filled up by farmers that to be encoded.
Success Guarantee	Added the farmers' livestock information successfully.
Scenarios:	<ol style="list-style-type: none"> 1. The farmers will provide his/her information. 2. The staff encoded it to the systems' livestock raisers form. 3. The system will go to the poultry form.

Table 8. Use Case Narrative - Add farmers' poultry information

To add farmers' poultry information (UC6)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To add new data for farmers poultry information.
Primary Actors	Office Head and Staff
Stakeholders	Office Head, Staff who want to add new farmers' poultry data.
Preconditions	None
Triggers	The Office Head and Staff will receive the records that been filled up by farmers that to be encoded.
Success Guarantee	Added the farmers' poultry information successfully.
Scenarios:	<ol style="list-style-type: none"> 1. The farmers will provide his/her information. 2. The staff encoded it to the systems' poultry raisers form. 3. The system will go to the vegetable form.

Table 9. Use Case Narrative - Add farmers' HVCDP (vegetables) information

To add farmers' HVCDP (vegetables) information (UC7)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To add new data for farmers HVCDP (vegetables) information.
Primary Actors	Office Head and Staff
Stakeholders	Office Head, Staff who want to add new farmers' HVCDP (vegetables) data.
Preconditions	None
Triggers	The Office Head and Staff will receive the records that been filled up by farmers that to be encoded.
Success Guarantee	Added the farmers' HVCDP (vegetables) information successfully.
Scenarios:	<ol style="list-style-type: none"> 1. The farmers will provide his/her information. 2. The staff encoded it to the system. 3. The system will go to fruit trees form.

Table 10. Use Case Narrative - Add farmers' fruit trees information

To add farmers' fruit trees information (UC8)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To add new data for farmers fruit trees information.
Primary Actors	Office Head and Staff
Stakeholders	Office Head, Staff who want to add new farmers' fruit trees data.
Preconditions	None
Triggers	The Office Head and Staff will receive the records that been filled up by farmers that to be encoded.
Success Guarantee	Added the farmers' fruit trees information successfully.
Scenarios:	<ol style="list-style-type: none"> 1. The farmers will provide his/her information. 2. The staff encoded it to the system. 3. The system will go to services form.

Table 11. Use Case Narrative - Add services received

To add services received (UC9)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To add services that farmers received.
Primary Actors	Office Head and Staff
Stakeholders	Office Head, Staff who want to add the services received by farmers.
Preconditions	None
Triggers	The Office Head and Staff will receive the records of the list of services received by farmers to be encoded.
Success Guarantee	Added the services successfully.
Scenarios:	<ol style="list-style-type: none"> 1. The farmers will provide his/her information. 2. The staff encoded it to the system. 3. The system will go to farmers visit form.

Table 12. Use Case Narrative - Record farmers visit

To record farmers visit (UC10)	
Scope	Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To record farmers visit.
Primary Actors	Staff
Stakeholders	Office Head, Staff who want to record farmers visit.
Preconditions	None
Triggers	The Office Head and Staff will record farmers visit to be encoded.
Success Guarantee	Record the farmers visit successfully.
Scenarios:	<ol style="list-style-type: none"> 1. The farmers will write his/her information or concern. 2. The staff encoded it to the system. 3. The system will go to the sub menu "reports" and select the report he/she wants to generate.

Table 13. Use Case Narrative - Generation of Reports

Generation of Reports (UC11)	
Scope	Farmers' Record Management System of Municipal Agriculture Office, San Miguel, Bohol
Level	User Goal
Goal in Content	To generate report of farmers' master list, farmers farm related information
Primary Actors	Office Head and Staff
Stakeholders	Office Head, Staff who want to generate reports
Preconditions	None
Triggers	The Office Head and Staff will display the total farmers and farm-related information
Success Guarantee	Display the total farmers and farm-related information successfully
Scenarios:	<ol style="list-style-type: none"> 1. The staff will prepare a report. 2. The office head reviewed and approved the reports made by the staff.

Database Design

Database design is the process of producing data model of the database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be to create a database. System design is an aspect of system progress that serves as building for the system enhances the Farmers Record Management System of Municipal Agriculture Office, San Miguel, Bohol. The purpose of this design is to produce a solution to the problem and develop a new system which would be used in the office.

Class Diagram

A class diagram is an illustration of the relationships and source code dependencies among classes of the system, their relationships (including

inheritance, aggregation and association), and the operations and attributes of the classes (Margaret, 2007).

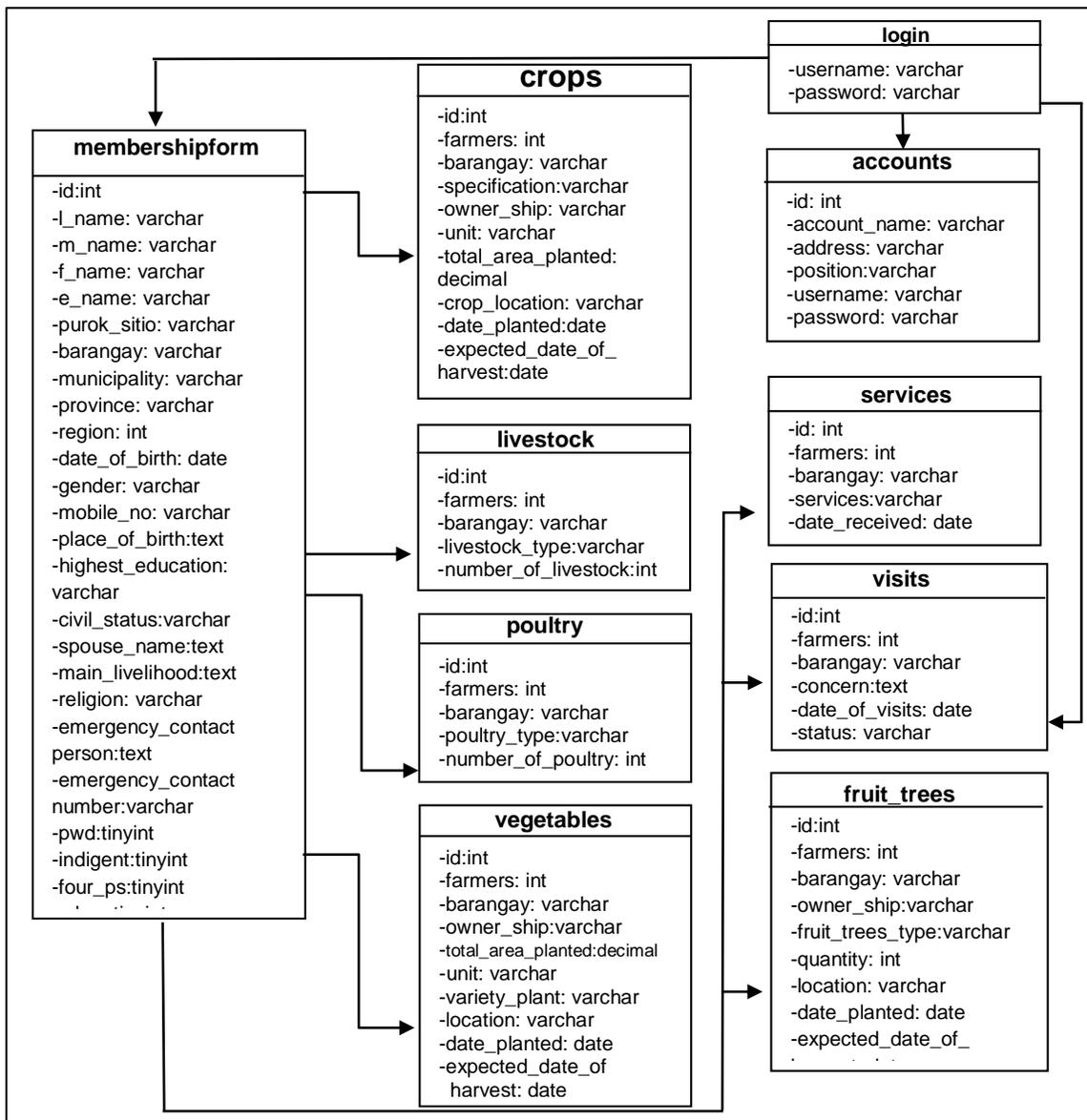


Figure 10. Class Diagram of the System

Data Structure

The tables on the following pages are the database tables used in storing the information that is inputted in the system together with a collection of operations that facilitate searching, sorting, recombination, and similar activities.

Table 14
Data Structure used for Login

Field No.	Field Name	Type	Width	Description
1	username	varchar	20	User's Username
2	password	varchar	20	User's Password

Table 15
Data Structure for Users Account

Field No.	Field Name	Type	Width	Description
1	id	int	11	User's Id
2	account_name	varchar	20	User's Name
3	address	varchar	50	User's Address
4	position	varchar	10	User's Position
5	username	varchar	10	User's Username
6	password	varchar	10	User's Password

Table 16
Data Structure for Membership Form

Field No.	Field Name	Type	Width	Description
1	id	int	5	Id
2	l_name	varchar	25	Last Name
3	m_name	varchar	25	Middle Name
4	f_name	varchar	50	First Name
5	e_name	varchar	5	Extension Name
6	purok_sitio	varchar	20	Purok
7	barangay	varchar	20	Barangay
8	municipality	varchar	20	Municipality
9	province	varchar	20	Province
10	region	int	10	Region
11	date_of_birth	date		Date Of Birth
12	gender	varchar	10	Gender
13	mobile_no	varchar	10	Mobile Number
14	place_of_birth	text		Place Of Birth
15	highest_education	varchar	30	Education
16	civil_status	varchar	20	Civil Status
17	spouse_name	text		Spouse Name
18	main_livelihood	text		Main Livelihood
19	religion	varchar	30	Religion
20	contact_person	text		Person to Contact
21	contact_number	varchar	50	Contact Number

22	pwd	tinyint	1	Pwd Member
23	indigent	tinyint	1	Indigent Member
24	four_ps	tinyint	1	4p's Member
25	rsbsa	tinyint	1	Rsbsa Member
26	image_file	longblob		Farmers Picture

Table 17
Data Structure for Crop Growers

Field No.	Field Name	Type	Width	Description
1	id	int	11	Id
2	farmers	int	11	Farmers Name
3	barangay	varchar	50	Barangay
4	specification	varchar	50	Type of Crop
5	owner_ship	varchar	50	Type Of Owner
6	unit	varchar	50	Unit of Area
7	total_area_planted	decimal	12,2	Area Planted
8	crop_location	varchar	50	Crop Location
9	date_planted	date		Date Planted
10	date_of_harvest	date		Date of Harvest

Table 18
Data Structure of Livestock Raisers

Field No.	Field Name	Type	Width	Description
1	id	int	11	Id
2	farmers	int	11	Farmers Name
3	barangay	varchar	50	Barangay
4	livestock_type	varchar	50	Type Of Livestock
5	number_of_livestock	int	10	No. Of Livestock

Table 19
Data Structure of Poultry Raisers

Field No.	Field Name	Type	Width	Description
1	id	int	11	Id
2	farmers	int	11	Farmers Name
3	barangay	varchar	20	Barangay
4	poultry_type	varchar	20	Type Of Poultry
5	number_of_poultry	int	10	No. Of Poultry

Table 20
Data Structure of Vegetable Growers

Field No.	Field Name	Type	Width	Description
1	id	int	11	Id
2	farmers	int	11	Farmers Name
3	barangay	varchar	50	Barangay
4	owner_ship	varchar	50	Type of Owner
5	total_area_planted	decimal	12,0	Area Planted
6	unit	varchar	50	Unit of Area
7	variety	varchar	50	Vegetable Variety
8	location	varchar	50	Location
9	date_planted	date		Date Planted
10	date_harvest	date		Date Harvest

Table 21
Data Structure of Fruit Trees Growers

Field No.	Field Name	Type	Width	Description
1	id	int	11	Id
2	farmers	int	11	Farmers Name
3	barangay	varchar	50	Barangay
4	owner_ship	varchar	50	Type of Owner
5	fruit_trees_type	varchar	50	Type of Fruit Trees
6	quantity	int	10	No. of Fruit Trees
7	location	varchar	50	Location
8	date_planted	varchar	50	Date Planted
9	date_of_harvest	date		Date ofHarvest

Table 22
Data Structure of Services Received

Field No.	Field Name	Type	Width	Description
1	id	int	11	Id
2	farmers	int	11	Farmers Name
3	barangay	varchar	50	Barangay
4	services	varchar	50	Services Received
6	date_received	date		Date Received

Table 23
Data Structure of Farmers Visit

Field No.	Field Name	Type	Width	Description
1	id	int	11	Id
2	farmers	int	11	Farmers Name
3	barangay	varchar	50	Barangay
4	concern	text		Farmers Concern
5	date_of_visits	date		Date of Visit
6	status	varchar	50	Action to Concerns

Program Hierarchy

A program hierarchy is a chart which shows the breakdown of the system to its lowest manageable levels. Each module is presented by a box, which contains the module name. The hierarchy visualized the relationship between the modules. The figure below shows the program hierarchy which shows all parts of the program, the menu and its submenu of the Farmers' Record Management System of Municipal Agriculture Office.

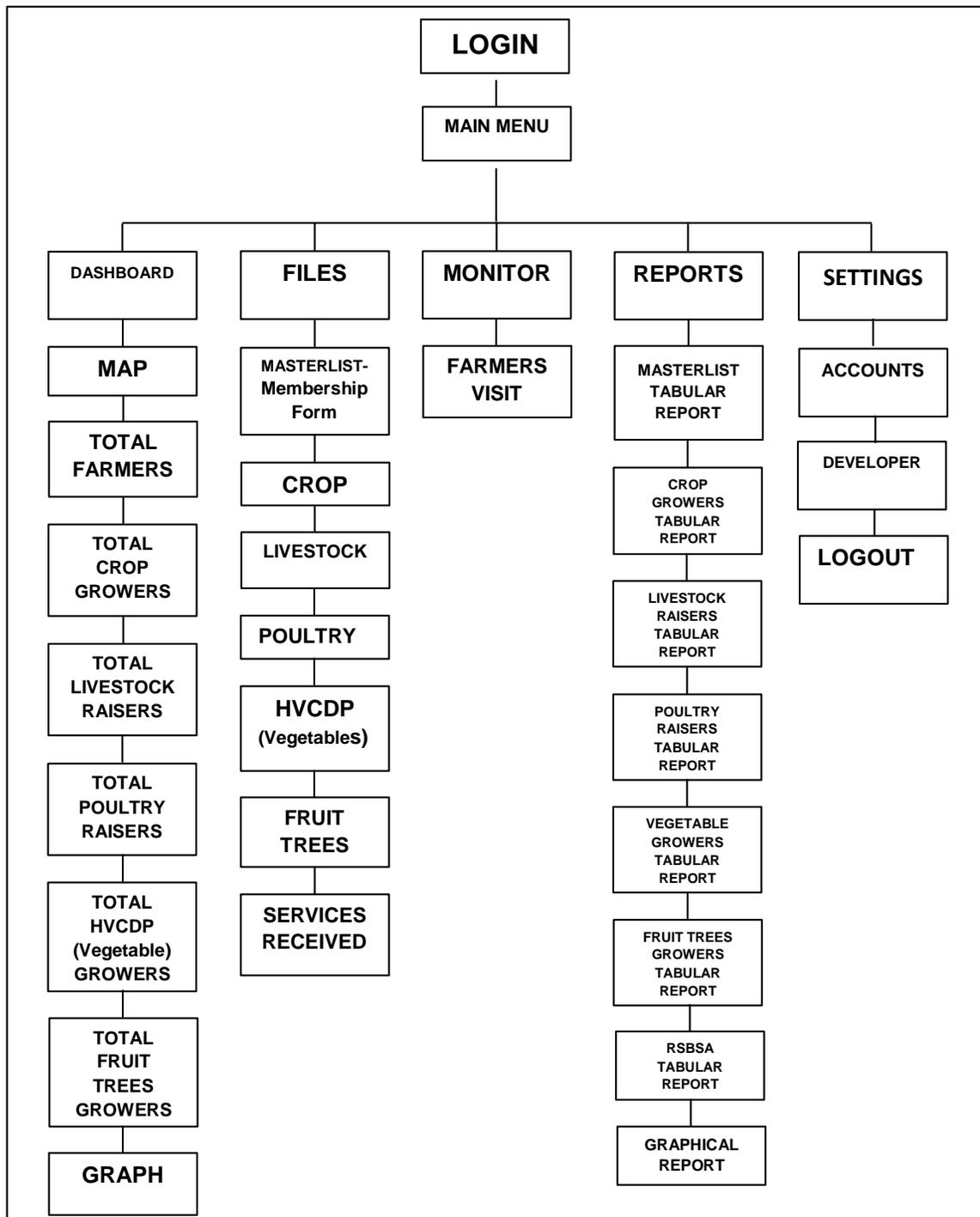


Figure 11. Program Hierarchy

Functional Requirements

A functional requirement defines a functional of a software system or its components. A function is described as a set of inputs, the behaviour and outputs. Functional requirements maybe calculations, technical details, data manipulation, and processing and other specific functionality that define what a system is supposed to accomplish. The functional requirements were formulated with the use of prototype in eliciting the requirement. The functionalities mentioned are based from the existing standard requirements of the information system with the approval and coordination from the respondents as follows:

Process Log In

FREQ 1: Access to the system must be password-protected.

FREQ 2: Access of the system must have a confirmation of the username and password if it is correct.

FREQ 3: The system should have a confirmation of the username and password if it is incorrect.

Process in Record Management

FREQ 4: The system should allow the Office Head to record farmers' personal, crop, livestock, poultry, vegetable, and fruit trees information.

FREQ 5: The system should allow the Staff to record farmers' personal, crop, livestock, poultry, vegetable, and fruit trees information.

FREQ 6: The system should allow the Office Head and Staff to record services received by farmers.

FREQ 7: The system should allow the Office Head and Staff to record farmers visit.

Process in Data Management

FREQ 8: The system should allow the Office Head and Staff to view, search and update farmers' personal, crop, livestock, poultry, vegetable, and fruit trees information.

FREQ 9: The system should allow the Office Head and Staff to view, search, and update services received by farmers.

FREQ10: The system should allow the Office Head and Staff to monitor farmers visit.

Generate Report Process

FREQ 11: The system should allow the Office Head and Staff to generate graphical reports of the total farmers in each barangay, generate tabular reports of farmers' master list, crop, vegetable, and fruit trees growers as well as tabular reports of livestock and poultry raisers which will be used in the decision making of the Farmers Record Management System

Personal System Setting

FREQ 12: The system should allow the Office Head and Staff to change their password and username.

FREQ 13: The system should allow the Staff to request to Office Head to update their password and username.

FREQ 14: The information from the database can be exported.

Non-Functional Requirements

A non-functional requirement shows the operation of a system, rather than functional requirements that define specific behavior of functions. The following are the non-functional requirements of the system:

NFREQ 1. The system must be easy to use and user-friendly so that the functions of the system will be generated properly.

NFREQ 2. The system must be accessible at all times.

Test Cases

A test case is a set of conditions or variables under which a tester will determine whether an application of software system is working correctly or not, a detailed procedure that fully tests a feature. It's also a set of input variables, execution pre-condition, results and execution, developed for a particular objectives or test condition.

These are the test case scenarios conducted during the acceptance testing. The test cases plan is to let the users use the system and follow the instruction in each test case in order to be considered successful. This test case is the guidelines on how to use the system.

Log-In

Test Case 1:

Module: Administrator Login

Severity: 1

Instructions:

1. In the "Login Form", input the username and password.
2. Then click "Login" button.

Expected Results:

1. System displays "Access Granted"
2. Homepage of the system should be displayed
3. User can now access the system features

Cleanup: Click "Cancel" button to close module.

Adding User

Test Case 2:

Module: Adding of users

Severity 1

Instructions:

1. Go to Settings in Main Menu.
2. Fill-up the Data Entry Form
3. Click the "SAVE" button.

Expected Result: The new added users should be successfully saved.

Cleanup: Click “Close” button to close module.

Module: Delete User

Severity: 2

Instructions:

1. Go to Settings in Main Menu.
2. Click the user you want to delete in the data grid view.
3. Click “Delete” button

Expected Result: The users should be successfully deleted.

Cleanup: Click “Close” button to close module.

Module: Update User

Severity: 3

Instructions:

1. Go to Settings in Main Menu.
2. Click the user you want to update in the data grid view.
3. Change the user in the data entry form
4. Click “Update” button

Expected Result: The user should be successfully updated.

Cleanup: Click "Close" button to close module.

Record Management

Test Case 3:

Module: Recording of Farmer

Severity: 1

Instructions:

1. On the main menu, click "File". Then select "MembershipForm"
2. Input the farmers personal information.
3. Click "Save" button
4. Click "Yes" command for confirmation

Expected Result: The farmers information should be successfully save.

Cleanup: Click "Close" button to close module.

Module: Recording of Crop Growers

Severity: 2

Instructions:

1. On the main menu, click "File". Then select "Crop Growers"
2. Input the farmers' crop information.
3. Click "Save" button

4. Click “Yes” command for confirmation

Expected Result: The farmers crop information should be successfully save.

Cleanup: Click “Close” button to close module.

Module: Recording of Livestock

Severity: 3

Instructions:

1. On the main menu, click “File”. Then select “Livestock Raisers”
2. Input the livestock information.
3. Click “Save” button
4. Click “Yes” command for confirmation

Expected Result: The livestock information should be successfully save.

Cleanup: Click “Close” button to close module.

Module: Recording of Poultry

Severity: 4

Instructions:

1. On the main menu, click “File”. Then select “Poultry Raisers”
2. Input the poultry information.
3. Click “Save” button

4. Click “Yes” command for confirmation

Expected Result: The poultry information should be successfully save.

Cleanup: Click “Close” button to close module.

Module: Recording of HVCDP (Vegetables)

Severity: 5

Instructions:

1. On the main menu, click “File”. Then select “Vegetable Growers”
2. Input the vegetable information.
3. Click “Save” button
4. Click “Yes” command for confirmation

Expected Result: The vegetable information should be successfully save.

Cleanup: Click “Close” button to close module.

Module: Recording of Fruit Trees

Severity: 6

Instructions:

1. On the main menu, click “File”. Then select “Fruit Trees Growers”
2. Input the fruit trees information.
3. Click “Save” button

4. Click “Yes” command for confirmation

Expected Result: The fruit trees information should be successfully save.

Cleanup: Click “Close” button to close module.

Module: Recording of Services Received

Severity: 7

Instructions:

1. On the main menu, click “File”. Then select “Services Received”
2. Input the services received by farmers.
3. Click “Save” button
4. Click “Yes” command for confirmation

Expected Result: The services received should be successfully save.

Cleanup: Click “Close” button to close module.

Test Case 4:

Module: Monitoring of Farmers Visit

Severity: 1

Instructions:

1. On the main menu, click “Monitor”. Then select “Farmers Visit”
2. Input the farmers’ information and its concern.

3. Click "Save" button
4. Click "Yes" command for confirmation

Expected Result: The farmers reason of visit should be successfully save.

Cleanup: Click "Close" button to close module.

Test Case 5:

Module: Update Farmers Record

Severity: 1

Instructions:

1. On the main menu, click "File".
2. Input the updated farmers' personal, crop, livestock, poultry, vegetables, fruit trees and services received information respectively.
3. Click "Update" button
4. Click "Yes" command for confirmation

Expected Result: The updated information should be successfully save.

Cleanup: Click "Close" button to close module.

Test Case 6:

Module: Reports

Severity: 1

Instructions:

1. Click the “Report” button on menu.
2. Select the “Tabular Report”
3. Select “Masterlist” button
4. Select “Crop Growers” button
5. Select “Livestock Raisers” button
6. Select “Poultry Raisers” button
7. Select “HVCDP(Vegetable) Growers” button
8. Select “Fruit Trees Growers” button
9. Select “RSBSA Masterlist” button
10. Select “Graphical Report” button

Expected Results:

1. Display the tabular report of farmers’ master list.
2. Display the tabular report of crop growers’ master list.
3. Display the tabular report of livestock raisers master list.
4. Display the tabular report of poultry raisers master list.
5. Display the tabular report of vegetable growers’ master list.
6. Display the tabular report of fruit trees growers’ master list.
7. Display the tabular report of RSBSA members’ master list.
8. Generate a graphical report of total farmers in every barangay.

Technical Requirements

The essential requirements that are needed in the implementation of Record Management System of Municipal Agriculture Office are hardware,

software as well as identification of people involved in the operation and network. These requirements must be present and work together in a newly developed system. It is necessary for proper usage so that the system could be used to its fullest capacity.

Hardware component refers to the physical part of the computer processing unit (CPU). This includes the motherboard, processor, random access memory, and hard disk. However, there were only three (3) components which basically facilitate the processing of data. These components are the Microprocessor, Hard Disk Drive (HDD) and the Random Access Memory (RAM). Thus, these components were just specified in the presentation.

Software often refers to programs of the computers. Otherwise, it is a set of instructions that are needed by the computer for processing, storing and retrieving necessary data. It is a set of machine-readable instructions and produces an output that can be understood by human.

People ware refers to the users of the computer. The system users in the Record Management System of Municipal Agriculture Office are the staff and the office head who are responsible in keeping all the records of farmers.

A network is a system of interconnected electric components or circuits. It connects two or more computers or other devices.

Minimum Hardware Specification

Component	Specification
Microprocessor	Intel Pentium, 1.6 Ghz
Hard Disk Drive	280 Gigabytes
Random Access Memory	4 Gigabytes, DDR3

Minimum Software Specification

The Farmers' Record Management System of Municipal Agriculture Office San Miguel, Bohol requires various software to function properly. This software was enumerated below with its corresponding specifications. The specifications as provided were based on the specifications of the computer units utilized during the development of the system.

Components	Specification
Operating System	Runs only on Windows operating system

Business Intelligence Integration

In the existing theories, business intelligence means the ability of an organization to collect, maintain and organize knowledge. It aims to support better business techniques and decision-making with solutions that take business intelligence (BI) to a whole new level and getting the right information. The system integrated business intelligence specifically in the query and reporting component. The developed system is capable of analyzing data and generating statistical reports thru tabular reports. These generated tabular reports maybe used by the office head and staff MAO as basis for monitoring of farmers and the status of the farmers.

There are (6) tabular reports which includes: farmers' masterlist, crop, vegetable, and fruit trees growers' masterlist, livestock and poultry raisers masterlist. Moreover, two graphical report shows the number of RSBSA member and the number of farmers per commodity.

Preview 1 shows the list of farmers by barangay.

Municipality of San Miguel		Republic of the Philippines Province of Bohol Municipality of San Miguel		Farmers Masterlist Reporting Year: 2022				
Barangay : Bugang								
Farmers Name	Address	Age	Date of Birth	Sex	Mobil Number	Civil Status	Main Livelihood	Registry Status
DEGAMO, TERESA ALBIT N/A	Bugang, San Miguel	0	22/08/2022	Female	09127350778	Separated	TEACHER	NOT YET RBSA
HERNANE, VICERE HERMINIA N/A	Bugang, San Miguel	0	22/08/2022	Female	09124586959	Single	TEACHER	NOT YET RBSA
Barangay : Mahayag								
Farmers Name	Address	Age	Date of Birth	Sex	Mobil Number	Civil Status	Main Livelihood	Registry Status
CANO, OMPYO BORROMEON/A	Mahayag, San Miguel	0	22/08/2022	Male	09303940888	Married	FARMER	NOT YET RBSA
CASING, TUPOS VICTOR N/A	Mahayag, San Miguel	58	22/08/1964	Male	09303564889	Widowed	FARMER	RBSA MEMBER
OMPOY, CANO CHERYL N/A	Mahayag, San Miguel	22	22/08/2000	Female	09303940888	Single	TEACHER	RBSA MEMBER
Prepared by: <u>DEMNA B. TIONGSON</u> Staff			Approved by: <u>MARCELA P. PALMA</u> Office Head			23/08/2022		

Preview 1. List of Farmers

Preview 2 shows the list of crop growers by barangay.

Municipality of San Miguel		Republic of the Philippines Province of Bohol Municipality of San Miguel		Crops Growers Masterlist Reporting Year: 2022				
Barangay : Bayongan								
Farmers	Barangay	Ownership	Crops Specification	Area	Unit	Date Planted	Date Harvested	Registry Status
CANO, OMPYO BORROMEON/A	Bayongan	Owner	Cassava	1.50	Hectare	04/08/2022	12/08/2022	NOT YET RBSA MEMBER
Barangay : Bugang								
Farmers	Barangay	Ownership	Crops Specification	Area	Unit	Date Planted	Date Harvested	Registry Status
CASING, TUPOS VICTOR N/A	Bugang	Heirs	Sweet Potato	1.20	Hectare	07/08/2022	10/08/2022	RBSA MEMBER
HERNANE, VICERE HERMINIA N/A	Bugang	Heirs	Rice	1.20	Hectare	01/08/2022	10/08/2022	NOT YET RBSA MEMBER
Prepared by: <u>DEMNA B. TIONGSON</u> Staff			Approved by: <u>MARCELA P. PALMA</u> Office Head			23/08/2022		

Preview 2. List of Crop Growers

Preview 3 shows the list of livestock raisers by barangay.

 Republic of the Philippines Province of Bohol Municipality of San Miguel Livestock Growers Masterlist Reporting Year: 2022			
Bayongan			
FARMERS	BARANGAY	LIVESTOCK	NUMBER OF LIVESTOCK
OMPOY, CANO CHERYL NA	Bayongan	Pig	5
Bugang			
FARMERS	BARANGAY	LIVESTOCK	NUMBER OF LIVESTOCK
DEGAMO, TERESA ALBITN/A	Bugang	Carabao	8
Mahayag			
FARMERS	BARANGAY	LIVESTOCK	NUMBER OF LIVESTOCK
CANO, OMPOY BORROMEON/A	Mahayag	Cattle	5
Prepared by: <u>DEMNA B. TIONGSON</u> Staff		Approved by: <u>MARCELA P. PALMA</u> Office Head	
23/08/2022			

Preview 3. List of Livestock Raisers

Preview 4 shows the list of poultry raisers by barangay.

 Republic of the Philippines Province of Bohol Municipality of San Miguel Poultry Growers Masterlist Reporting Year: 2022			
Bayongan			
FARMERS	BARANGAY	TYPE	NUMBER OF POULTRY
OMPOY, CANO CHERYL NA	Bayongan	Duck	10
Bugang			
FARMERS	BARANGAY	TYPE	NUMBER OF POULTRY
DEGAMO, TERESA ALBITN/A	Bugang	Chicken	5
Mahayag			
FARMERS	BARANGAY	TYPE	NUMBER OF POULTRY
CANO, OMPOY BORROMEON/A	Mahayag	Chicken	10
CASING, TUPOS VICTOR NA	Mahayag	Duck	5
Prepared by: <u>DEMNA B. TIONGSON</u> Staff		Approved by: <u>MARCELA P. PALMA</u> Office Head	
23/08/2022			

Preview 4. List of Poultry Raisers

Preview 5 shows the list of vegetable growers.

Farmers Name	Location	Variety	Total Area Planted	Unit	Date Planted	Date of Harvest	Registry Status
CANO, OMPOY BORROMEO N/A	Corazon	Carrots	1.00	Hectare	01/08/2022	04/08/2022	NOT YET RSBSA MEMBER
CASING, TUPOS VICTOR N/A	Kagawasan	Tomatoe	2.00	Hectare	04/08/2022	05/08/2022	RSBSA MEMBER
OMPOY, CANO CHERYL N/A	Kagawasan	Lettuce	1.00	Hectare	22/08/2022	22/08/2022	RSBSA MEMBER

Republic of the Philippines
 Province of Bohol
 Municipality of San Miguel
Vegetables Growers Masterlist
 Reporting Year: 2022

Prepared by: DEMNA B. TIONGSON
 Staff

Approved by: MARCELA P. PALMA
 Office Head

23/08/2022

Preview 5. List of Vegetable Growers

Preview 6 shows the list of fruit trees growers.

Farmers Name	Variety	Type of Ownership	Quantity	Location	Date Planted	Date of Harvest	Registry Status
CANO, OMPOY BORROMEO N/A	Manga	Owner	5	Cambangay Norte	01/08/2022	04/08/2022	NOT YET RSBSA MEMBER
CASING, TUPOS VICTOR N/A	Manga	Heirs	7	San Vicente	05/08/2022	10/08/2022	RSBSA MEMBER
OMPOY, CANO CHERYL N/A	Jackfruit	Owner	5	Bayongan	22/08/2022	22/08/2022	RSBSA MEMBER

Republic of the Philippines
 Province of Bohol
 Municipality of San Miguel
Fruit Trees Growers Masterlist
 Reporting Year: 2022

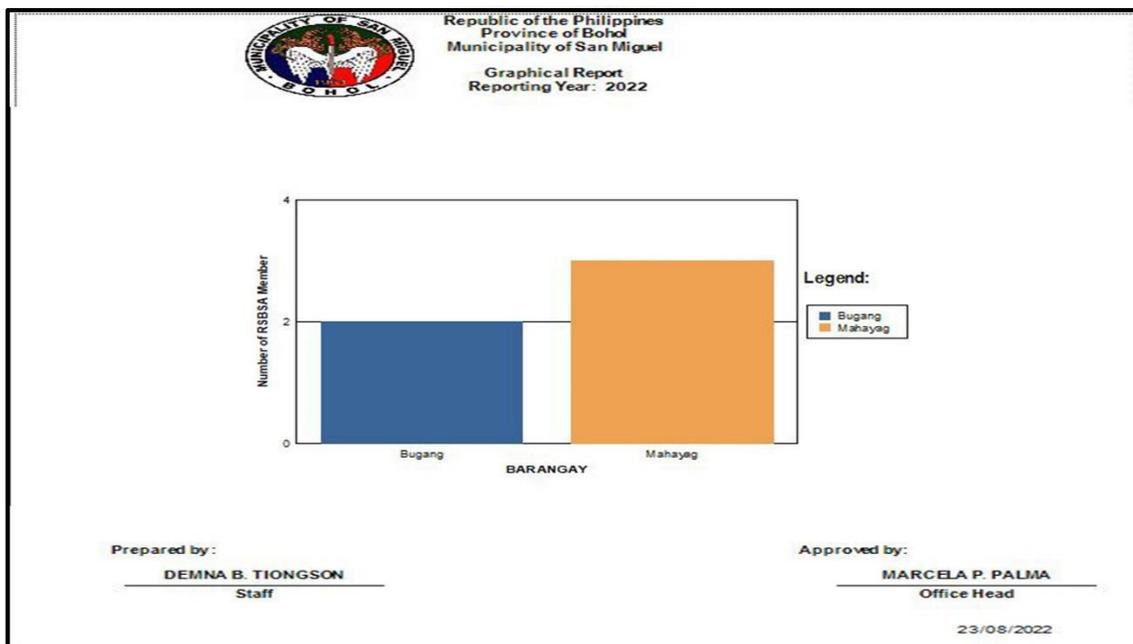
Prepared by: DEMNA B. TIONGSON
 Staff

Approved by: MARCELA P. PALMA
 Office Head

23/08/2022

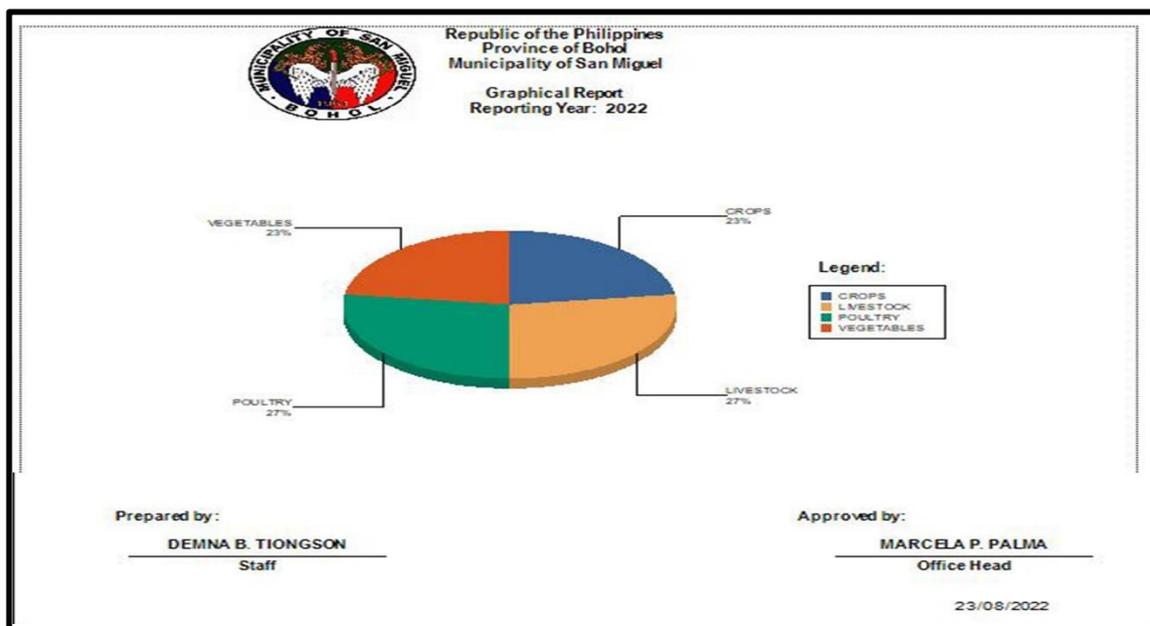
Preview 6. List of Fruit Trees Growers

Preview 7 shows the graphical report of RSBSA member per barangay.



Preview 7. Graphical Report of RSBSA Members

Preview 8 illustrates the graphical report of farmers by commodity.



Preview 8. Graphical Report of Farmers by Commodity

Screen Layout

Screen layout is one of the attributes of the system's user friendliness. It was designed in such a way that the browsers can navigate the system quickly and easily to provide a clear recognition of the tasks the users need to perform.

Preview 9 shows the log-in form of the system.

Welcome to the
Farmers Record
Management System

Develop by
San Miguel Group

Login to you account

LOGIN Cancel

Preview 9. Log In Form

Preview 10 shows the dashboard of the system.

Farmers Record System

MARCELA_PALMA

Dashboard

Files

Monitor

Reports

Settings

Logout

Dashboard

San Miguel Map

Summary of Farmers per Commodity

Description	Count
CROPS	3
LIVESTOCK	4
POULTRY	4
VEGETABLES	3
FRUIT TREES	3

Preview 10. Dashboard

Preview 11 shows the farmers list view form.

The screenshot shows the 'Master File' section of the Farmers Record System. On the left is a sidebar with navigation options: Masterfile (selected), Crops, Livestocks, Poultry, Fruit Trees, HVCDP, Services Received, and Return. The main area features a search bar and a 'Create' button. Below is a table titled 'Farmers List' with the following data:

ID	Full Name	Purok or Sitio	Barangay	Municipality	Province	Region	Date of Birth	Gender	Mobile No.	MAH
2	CANO, OMPOY BORROMELO, N/A	3	Mahayag	San Miguel	Bohol	7	2022-08-22	Male	09303940888	MAH
3	CASING, TUPOS VICTOR, N/A	4	Mahayag	San Miguel	Bohol	7	1964-08-22	Male	09303564889	POBLA
4	DEGAMO, TERESA ALBIT, N/A	5	Bugang	San Miguel	Bohol	7	2022-08-22	Female	09127350778	BUG
5	HERNANE, VICERE HERMINIA, N/A	4	Bugang	San Miguel	Bohol	7	2022-08-22	Female	09124586559	MAH
6	PALMA, OMPOY SERGIO, N/A	4	San Jose	San Miguel	Bohol	7	1987-08-22	Male	09303940888	MAH
7	OMPOY, CANO CHERYL, N/A	4	Mahayag	San Miguel	Bohol	7	2000-08-22	Female	09303940888	BUG
8	PAMAT, CASTRO JOBERT, N/A	4	Bayongan	San Miguel	Bohol	7	1965-08-22	Male	09303940888	POBLA

Preview 11. Farmers List View Form

Preview 12 shows the farmers membership form.

The screenshot shows the 'Membership Form' interface. It is divided into two main sections: 'Personal Info' and 'Address'. The 'Personal Info' section includes fields for First Name, Middle Name, Last Name, Extension, Birthday (22/08/2022), Age (0), Place of Birth, Sex, Highest Formal Education, Religion, Civil Status, Main Livelihood, and Mobile No. There is a 'Select Photo' button next to a placeholder image. The 'Address' section includes dropdown menus for Purok / Sitio, Barangay, Municipality (San Miguel), Province (Bohol), and Region (7). It also has a text field for 'Notify in case of Emergency', a 'Contact Number' field, and checkboxes for 'PWD', 'Member of Indigenous Group', '4PS Member', and 'RSBSA Member'. At the bottom right, there are 'Submit' and 'Close' buttons.

Preview 12. Membership Form

Preview 13 shows the crop growers view form.

Farmers Record System

MARCELA_PALMA

Masterfile

Crops

Livestocks

Poultry

Fruit Trees

HVCDP

Services Received

Return

Search:

Create Edit Delete

Crops List

ID	FARMERS	SPECIFICATION	OWNERSHIP	UNIT	TOTAL AREA PLANTED	CROP LOCATION	DATE PLANTED	EXPECTED HARVEST DATE
2	CANO, OMPOY BORROMEO N/A	Cassava	Owner	Hectare	1.50	Bayongan	04/08/2022	
3	CASING, TUPOS VICTOR N/A	Sweet Potato	Heirs	Hectare	1.20	Bugang	07/08/2022	
8	OMPOY, CANO CHERYL N/A	Peanut	Owner	Square Meter	100.00	Capayas	04/08/2022	
7	HERNANE, VICERE HERMINIA N/A	Rice	Heirs	Hectare	1.20	Bugang	01/08/2022	
9	PALMA, OMPOY SERGIO N/A	Corn	Owner	Square Meter	200.00	Capayas	05/08/2022	

Preview 13. Crop Growers View Form

Preview 14 shows the crop growers fill-up form.

Crops Form

Farmer's Details

Farmer:

Type of Ownership:

Crop's Details

Crops Specification:

Unit:

Date Planted:

Crop Location:

Total Area Planted:

Expected Date Harvest:

Submit Close

Preview 14. Crop Growers Form

Preview 15 shows the livestock raisers view form.

The screenshot displays the 'Livestock' view form. On the left is a sidebar with the user profile 'MARCELA_PALMA' and navigation links: Masterfile, Crops, Livestocks (highlighted), Poultry, Fruit Trees, HVCDP, Services Received, and Return. The main content area is titled 'Livestock' and includes 'Create', 'Edit', and 'Delete' buttons. Below these is a search bar and a table titled 'Livestock List'.

ID	FARMERS	BARANGAY	LIVESTOCK	NUMBER OF LIVESTOCK
1	CANO, OMPOY BORROMEO N/A	Mahayag	Cattle	5
2	CASING, TUPOS VICTOR N/A	Mahayag	Carabao	6
3	DEGAMO, TERESA ALBIT N/A	Bugang	Carabao	8
4	OMPOY, CANO CHERYL N/A	Bayongan	Pig	5
5	PALMA, OMPOY SERGIO N/A	Caluasan	Goat	4
6	PAMAT, CASTRO JOBERT N/A	Garcia	Cattle	4

Preview 15. Livestock Raisers View Form

Preview 16 shows the livestock raisers fill-up form.

The screenshot shows the 'Livestock Form' for data entry. It is divided into two main sections: 'Farmer's Details' and 'Livestock's Details'. The 'Farmer's Details' section has a 'Farmer:' dropdown menu. The 'Livestock's Details' section includes a 'Livestock:' dropdown menu, a 'Number of Livestock:' text input field, and a 'Location:' dropdown menu. At the bottom of the form are 'Submit' and 'Close' buttons.

Preview 16. Livestock Raisers Form

Preview 17 shows the poultry raisers view form.

The screenshot displays the 'Poultry' view form within the 'Farmers Record System'. The sidebar on the left includes a user profile for 'MARCELA_PALMA' and navigation buttons for 'Masterfile', 'Crops', 'Livestocks', 'Poultry' (highlighted), 'Fruit Trees', 'HVCDP', 'Services Received', and 'Return'. The main content area features a search bar, 'Create', 'Edit', and 'Delete' buttons, and a table titled 'Poultry List'.

ID	FARMERS	BARANGAY	TYPE	NUMBER OF POULTRY
1	CANO, OMPOY BORROMELO N/A	Mahayag	Chicken	10
2	CASING, TUPOS VICTOR N/A	Mahayag	Duck	5
3	DEGAMO, TERESA ALBIT N/A	Bugang	Chicken	5
4	OMPOY, CANO CHERYL N/A	Bayongan	Duck	10
5	PALMA, OMPOY SERGIO N/A	Mahayag	Chicken	10
6	PAMAT, CASTRO JOBERT N/A	Capayas	Duck	5

Preview 17. Poultry Raisers View Form

Preview 18 shows the poultry raisers fill-up form

The screenshot shows the 'Poultry Form' for data entry. It is divided into two main sections: 'Farmer's Details' and 'Poultry's Details'. The 'Farmer's Details' section contains a dropdown menu for 'Farmer:'. The 'Poultry's Details' section includes dropdown menus for 'Poultry:', 'Location:', and a text input field for 'Number of Poultry:'. At the bottom, there are 'Submit' and 'Close' buttons.

Preview 18. Poultry Raisers Form

Preview 19 shows the vegetable growers view form.

Farmers Record System

MARCELA_PALMA

HVCDP

Create Edit Delete

Search:

HVCDP List

ID	FARMERS	TOTAL AREA PLANTED	UNIT	VARIETY	LOCATION	DATE PLANTED	EXPECTED DATE OF HARVEST
1	CANO, OMPOY BORROMEO N/A	1	Hectare	Carrots	Corazon	01/08/2022	04/08/2022
2	CASING, TUPOS VICTOR N/A	2	Hectare	Tomatoe	Kagawasan	04/08/2022	05/08/2022
3	OMPOY, CANO CHERYL N/A	1	Hectare	Lefuce	Kagawasan	22/08/2022	22/08/2022
4	PALMA, OMPOY SERGIO N/A	100	Square Meter	Lefuce	Garcia	08/08/2022	10/12/2022
5	PAMAT, CASTRO JOBERT N/A	1	Hectare	Potato	Capayas	22/08/2022	22/08/2022

Preview 19. Vegetable Growers View Form

Preview 20 shows the vegetable growers fill-up form.

HVCDFForm

HVCDF Form

Farmer's Details

Farmer: Type of Ownership:

Vegetable's Details

Variety Plant: Unit: Date Planted:

Location: Total Area Planted: Expected Date Harvest:

Submit Close

Preview 20. Vegetable Growers Form

Preview 21 shows the fruit trees growers view form.

Fruit Trees

Search:

Fruit Trees List

ID	FARMERS	FRUIT TREES TYPE	OWNERSHIP	QUANTITY	LOCATION	DATE PLANTED	EXPECTED DATE OF HARVEST
1	CANO, OMPOY BORROMEO N/A	Owner	Manga	5	Cambangay Norte	01/08/2022	04/08/2022
2	CASING, TUPOS VICTOR N/A	Heirs	Manga	7	San Vicente	05/08/2022	10/08/2022
3	OMPOY, CANO CHERYL N/A	Owner	Jackfruit	5	Bayongan	22/08/2022	22/08/2022
4	PALMA, OMPOY SERGIO N/A	Owner	Lemon	4	Mahayag	04/08/2022	05/08/2022
5	PAMAT, CASTRO JOBERT N/A	Tenant	Mango	20	Cabangahan	05/08/2022	10/11/2022

Preview 21. Fruit Trees Growers View Form

Preview 22 shows the fruit trees growers fill-up form.

Fruit Trees Form

Farmer's Details

Farmer: Type of Ownership:

Fruit Trees Details

Type of Fruit Tree: Location: Date Planted:

Quantity: Expected Date Harvest:

Submit Close

Preview 22. Fruit Trees Growers Form

Preview 23 shows the services received view form.

The screenshot shows the 'Services Received' view in the Farmers Record System. The sidebar on the left contains the following items: MARCELA_PALMA (user profile), Masterfile, Crops, Livestocks, Poultry, Fruit Trees, HVCDP, Services Received (highlighted in green), and Return. The main content area is titled 'Services Received' and features a search bar, 'Create' (green), 'Edit' (blue), and 'Delete' (red) buttons. Below the search bar is a table titled 'Services Received List' with the following data:

ID	FARMERS	BARANGAY	SERVICES	DATE RECEIVED
1	CANO, OMPOY BORROMEO N/A	Mahayag	Corn Seeds	10/08/2022
2	CASING, TUPOS VICTOR N/A	Mahayag	Corn Seeds	10/08/2022
3	OMPOY, CANO CHERYL N/A	Mahayag	Vegetables Seeds	10/08/2022
4	PALMA, OMPOY SERGIO N/A	Cambangay Norte	Vegetables Seeds	11/08/2022
5	PAMAT, CASTRO JOBERT N/A	Cambangay Norte	Corn Seeds	18/07/2022

Preview 23. Services Received View Form

Preview 24 shows the services received fill-up form.

The screenshot shows the 'Services Form' for filling up service records. The form is titled 'Form' and 'Services Form'. It is divided into two main sections: 'Farmer's Details' and 'Services Details'. In the 'Farmer's Details' section, there are two dropdown menus: 'Farmer:' and 'Barangay :'. In the 'Services Details' section, there are two dropdown menus: 'Services Received :' and 'Date Received :'. The 'Date Received :' field is filled with '22/08/2022'. At the bottom of the form, there are two buttons: 'submit' (green) and 'Close' (red).

Preview 24. Services Received Form

Preview 25 shows the farmers visit view form.

The screenshot shows the 'Farmers Record System' window. On the left is a sidebar with a user profile for 'MARCELA_PALMA' and a 'Farmers Visits' menu item. The main area is titled 'Farmers Visits' and contains a search bar, a table of visit records, and buttons for 'Create', 'Edit', and 'Delete'.

ID	FARMERS	BARANGAY	CONCERN	DATE OF VISITS	STATUS
1	CANO, OMPOY BORROMELO N/A	Mahayag	IRRIGATION	01/08/2022	Already been cater
2	CASING, TUPOS VICTOR N/A	Mahayag	SEEDS	10/08/2022	Not yet
3	OMPOY, CANO CHERYL N/A	Mahayag	SEEDLINGS	10/08/2022	Not yet
4	PALMA, OMPOY SERGIO N/A	Camanaga	FERTILIZER	11/08/2022	Already been cater

Preview 25. Farmers Visit View Form

Preview 26 shows the farmers visit fill-up form.

The screenshot shows the 'Farmers Visits' window with the 'Farmers Visit Form' open. The form is divided into two sections: 'Farmer's Details' and 'Farmers Visit Details'. It includes fields for Farmer, Barangay, Concern, Date of Visits, and Action, along with Submit and Close buttons.

Preview 26. Farmers Visit Form

Economic Performance Evaluation

The initial investment is one of the most important parts of any system and design project. The annual operating cost is the total amount required in one year of implementation. This includes office supplies, utilities, system maintenance, and other related expenditures. The purpose of initial investment is to evaluate the economic performance that will result from the operation and development of the new system. The table below presents the required investment and annual operating cost of the system.

Table 24

Initial Investment and Annual Operating Cost

Item	Qty	Unit	Unit Price	Total
A. Initial Investment				
A.1 Hardware				
Desktop Computer	3	Set	Php 23,000.00	Php 69,000.00
Printer	1	Unit	Php 7,500.00	Php 7,500.00
Internet Service Provider	1	Piece	Php 10,000.00	Php 10,000.00
Sub-total				Php 86,500.00
A.2 Software				
Software Licensing			Php 8,000.00	Php 8,000.00
Software Installation			Php 1,000.00	Php 1,000.00
Sub-total				Php 9,000.00
Total Initial Investment Cost				Php 95,500.00
B. Annual Operating Cost				
B.1 Office Supplies				
Bond paper	5	Reams	Php 156.00	Php 780.00
Ball Pen	3	Pieces	Php 8.00	Php 24.00
Stapler	1	Piece	Php 90.00	Php 90.00
Stapler wire	2	Boxes	Php 15.00	Php 30.00
Printer Ink				
Cyan	1	Piece	Php 245.00	Php 245.00
Magenta	1	Piece	Php 245.00	Php 245.00
Yellow	1	Piece	Php 245.00	Php 245.00

Black	1	Piece	Php 245.00	Php 245.00
Folder	10	Pieces	Php 7.00	Php 70.00
Fastener	1	Box	Php 25.00	Php 25.00
B.2 UTILITIES				
Electricity	12	Kw/h	Php 300.00	Php 3,600.00
B.3 GENERAL SERVICES				
System Maintenance	1		Php 1,000.00	Php 1,000.00
Total Annual Operating Cost				Php 6,599.00

Testing and Evaluation

Testing and evaluation were performed to determine the functionality of the system particularly of the provision of expected output, amount of information handled, time/length of information processing, and the proper response of user inputs. This is also the process of evaluating the performance of the system in general. In system testing, a software test professional aims to detect bugs within the interfaces and also within the software as a whole. System usability was used as a parameter in this study.

System Usability

The system usability test had adopted the questionnaire developed by Lewis (1995). This was done to determine the level of system acceptability as perceived by the target users. The system usability test was conducted by the researchers last May 24, 2022 started at 1PM and ended at 3PM, at the Municipal Agriculture Office, Municipality of San Miguel, Bohol. It took 2 hours to do a hands on activity and demonstrate the uses and functions of the system. The respondents were the Office Head, Staff, and IT Technician. The testing was done accordingly, and the entire process have been discussed very well. Based on the result of the rating, most of them rated our system and gave a general rating of 6.42 with an

interpretation of “Strongly Agree”. The outcome generally indicates that the system is usable and the expectations of the respondents as to functions and features have been achieved by the system.

Table 25

System Usability Assessment Result

Criteria for System Usability	Weighted Mean	Rating
1. Overall, I am satisfied with how easy it is to use this system.	6.8	Strongly Agree
2. It was simple to use this system.	7.0	Strongly Agree
3. I can effectively complete my work using this system.	6.2	Agree
4. I am able to complete my work quickly using this system.	6.0	Agree
5. I am able to effectively complete my work using this system.	6.0	Agree
6. I feel comfortable using this system.	6.4	Strongly Agree
7. It was easy to learn to use this system.	6.2	Agree
8. I believe I became productive quickly using this system.	6.6	Strongly Agree
9. The system gives error messages that clearly tell me how to fix problems.	6.2	Agree
10. Whenever I make a mistake using the system, I recover easily and quickly.	6.4	Strongly Agree
11. The information (such as online help, on-screen messages, and other documentation) provided with this system is clear.	6.2	Agree
12. It is easy to find the information I needed.	6.6	Strongly Agree
13. The information provided for the system is easy to understand.	6.8	Strongly Agree
14. The information is effective in helping me complete tasks and scenarios.	6.2	Agree
15. The organization of this system screens is clear.	6.2	Agree
16. The interface of this system is pleasant,	6.6	Strongly Agree
17. I like using the interface of this system.	6.8	Strongly Agree
18. This system has all the functions and capabilities I expect it to have.	6.0	Agree
19. Overall, I am satisfied with this system.	6.8	Strongly Agree
AVERAGE WEIGHTED	6.42	Strongly Agree

CHAPTER 3

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

Summary of Findings

Based on the study of the present system, the researchers were able to identify the following gaps: duplication of records which affect consistency in data record, misplacing and losing of files, time consuming procedure in storage and retrieval of records of farmers, report generation was limited to summary list, and pertinent data report requires sorting the entire list of farmers available by barangay. This was due to the semi-manual method, used in recording the farmers' information. The identified gaps and problems in the office has led the researchers to come up with a solution.

The researchers came up with a computerized system which is the Farmers Record Management System with modules such as administration, record management, and reports. It allows data correction, data update and generate tabular and graphical reports. The developed system was tested and evaluated using the system usability questionnaire by Lewis James R.

Based from the results of the evaluation, the respondents or the users had the general rate of 6.42 with and interpretation of "Strongly Agree" which implies that the system is strongly high on its usability. The result indicates that the systems functions and features provides satisfaction among respondents. Furthermore, it also implies that the system is user-friendly, easy to understand effective, useful and informative.

Conclusions

The recording framework were generated based on the defined needs, with the following modules: data processing, recording, administration, and report generation. The features of the system are applicable to the processes conducted by the office.

Based on the evaluation in Farmers Record Management System of Municipal Agriculture Office, the usability and features of the developed system suits the need and requirement of the client. In addition, putting into consideration the economic performance evaluation which needs an ample amount to be allocated but can be recovered with the expected benefits of the developed system, it has been concluded that the adaptation of the developed system is reasonable and justifiable.

Recommendations

Based from the observation during implementation and on the aforementioned conclusions, the following are recommended:

1. Municipal Agriculture Office in San Miguel must consider the benefits of the developed system in order to enhance record keeping and identify any system bugs and errors.
2. Training and orientation must be conducted by the developers to the users for them to familiarize and be oriented with the functionality of the system.
3. The office should provide regular system maintenance to ensure the system's dependability.

REFERENCES

- Alcala, A., Cortes, M.S., Yleaña, M.V. (2019). Farmers Information and Technology Services (FITS) Management System of Carmen, Bohol. Unpublished Thesis. Bohol Island State University- Bilar Campus, Zamora, Bilar, Bohol.
- Batilong, M., Lomod, M., Visto, J. (2018). Recording and Monitoring System of the Provincial Social Welfare and Development. Unpublished Undergraduate Thesis. Bohol Island State University- Bilar Campus, Zamora, Bilar, Bohol
- Cabradilla, C., Cutab, A.J., Inso, M. (2015). Farm Records Management System of Bohol Island State University. Unpublished Undergraduate Thesis. Bohol Island State University- Bilar Campus, Zamora, Bilar, Bohol.
- Codd, E. F. (1985). Relational database: a practical foundation for productivity. *Communications of the ACM*, 25(2), 109-117. Retrieved: March 27, 2022 from [https:// pdfs.semanticscholar.org/dbc.pdf](https://pdfs.semanticscholar.org/dbc.pdf).
- Dabeer H. (2017). *Scholarship Database Management System*. Retrieved on March 27, 2022 from <https://www.slideshare.net/HafizDabeer/scholarship-database-management-system>.
- Fhatima, 2012. Article XIV, Section 10 of the Philippine Constitution
- GovPH, (2021). Office of the municipal agriculturist. Retrieved May 16, 2022 <https://quezon-nviscaya.gov.ph/offices/economic-development-sector/>
- Lewis J. (September 2009). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. Retrieved on April 2022 from <https://doi.org/10.1080/10447319509526110>
- Massachusetts Institute of Technology (1995). IBM Computer Usability Satisfaction Questionnaires. Retrieved on March 27, 2022 from <http://ist.mit.edu/services/consulting/usability/guidelines>.
- Mickeel, A. (2019). Technological Influence and Society, Alvernia University. Retrieved: March 27, 2022 from [https:// www.bctv.org/technological-influenceonsociety/#:~:text=Technology%20affects%20the%20way%20in,individuals,and%20it%20impacts%20daily%20lives.pdf](https://www.bctv.org/technological-influenceonsociety/#:~:text=Technology%20affects%20the%20way%20in,individuals,and%20it%20impacts%20daily%20lives.pdf).
- Ortega, R.JR., Lagang, J., Busano, J., (2021) Service Management System of Philippine Veteran Affairs Office(FSEO) Tagbilaran City, Bohol. Unpublished Thesis. Bohol Island State University-Bilar Campus, Zamora, Bilar, Bohol.

APPENDICES

Appendix A

Letter of Intent



Republic of the Philippines
Bohol Island State University
Bilar Campus
Zamora, Bilar, Bohol



April 01, 2022

MARCELA P. PALMA
Head of Municipal Agriculture Office
San Miguel, Bohol

Ma'am:

Good day!

We, the 4th Year Students of Bachelor of Science in Computer Science of Bohol Island State University Bilar Campus will conduct a System Development project (Thesis) as requirements for graduation for the degree of Bachelor of Science in Computer Science.

In this connection, we would like to ask permission to conduct a system study (thesis) basing on your recording of farmers record. This study would include interview, observation and review of documents significant for the study and development of the system. We anticipate to have implementation testing (bench marking) of the system developed as part of usability testing.

We assure you that we shall honor secrecy and privacy to all data and information we shall be handling as we go along with our study. Your approval will be a great help to the success of our study.

We anticipate your favorable response regarding this matter.

Thank you very much and more power!

Respectfully yours,

JOSHUA S. QUIROL

CHYMNA MARIE H. CANO

ANA MAE GRACE A. VICERE

LITO BOY G. OMUL
Researchers

Noted:

SHEILA G. TABUNO
Subject Instructor

Recommended by:

SHEILA G. TABUNO
Chairperson, DCoS

REX VINCENT D. TEJADA
Thesis Adviser

Endorsed by:

ARLEN B. GUDMALIN, PhD
Dean, CTAS

Approved by:

MARCELA P. PALMA
Head, Municipal Agriculture Office

Letter of Implementation



Republic of the Philippines
Bohol Island State University
Bilar Campus
Zamora, Bilar, Bohol



May 30, 2022

MARCELA P. PALMA
Head of Municipal Agriculture Office
San Miguel, Bohol

Ma'am:

Good day!

It is our pleasure to inform you that the system "Farmers Record Management System of Municipal Agriculture Office" is now in its final phase. With this, we would like to conduct benchmarking activities as part of the implementation.

It will be conducted on May 30, 2022 at exactly 1:00 PM to 3:00 PM at Municipal Agriculture Office, San Miguel, Bohol. This activity will allow you to assess our developed system and give feedback as well.

By this time, we would like to express our gratitude for allowing us to conduct our thesis study. We are hoping for future collaborations with you, our dear client.

May the Lord continually bless you and your office. Thank you and more power!

Respectfully yours,

[Signature]
JOSHUA S. QUIROL

[Signature]
CHYMNA MARIE H. CANO

[Signature]
ANA MAE GRACE A. VICERE

[Signature]
LITO BOY G. OMUL
Researchers

Noted:

[Signature]
SHEILA G. TABUNO
Subject Instructor

Recommended by:

[Signature]
SHEILA G. TABUNO
Chairperson, DCoS

[Signature]
REX VINCENT D. TEJADA
Thesis Adviser

Endorsed by:

[Signature]
ARLEN B. GUDMALIN, PhD
Dean, CTAS

Approved by:

[Signature]
MARCELA P. PALMA
Head, Municipal Agriculture Office

Letter of Questionnaire Distribution

May 30, 2022

MARCELA P. PALMA

Office Head

Municipal Agriculture Office

Poblacion, San Miguel, Bohol

Dear Ma'am,

Greetings!

It is our pleasure to inform you that the system "Farmers Record Management System of Municipal Agriculture Office" is now in its final phase. With this, we would like to get your views and opinions in the developed system.

We wished to know your feedback as we work for the improvement of the system. It is rightfully needed to hear your side since you will be the one who would use our developed system. Your inputs would be very valuable.

We greatly appreciate if you would make time and answer the questionnaire. The data that would be gathered will be used for rating statistics in our developed system. If there are items you are confused with, feel free to ask and we would gladly assist you.

By this time, we would like to express our gratitude for allowing us to conduct our thesis study. We are hoping for future collaboration with you, our dear client.

May the good Lord continually bless you and your office.

Thank you and more power!

Very truly yours,

Researchers

APPENDIX B

System Usability Questionnaire

Instructions:

- Please rate the usability questioner
- Try to respond to all of the items
- For items that are not applicable, use N/A
- Make sure these fields are filled in

Rating Scale:

- 7 – Strongly Agree
- 6 – Agree
- 5 – Tend to Agree
- 4 – Neither Agree or Disagree
- 3 – Tend to Disagree
- 2 – Disagree
- 1 – Strongly Disagree

Criteria for System Usability	Weighted Mean	Interpretation
1. Overall, I am satisfied with how easy is to use this system.		
2. It was simple to use this system.		
3. I can extremely complete my work using this system.		
4. I am able to efficiently complete my work using this system.		
5. I can effectively complete my work using this system.		
6. I feel comfortable using this system.		
7. It was easy to learn to use this system.		
8. I believe I became productive quickly using this system		
9. The system gives error messages that clearly tell me how to fix problems.		

10. Whenever I make a mistake using the system, recover easily and quickly.		
11. The information (such as online help, on screen messages, and other documentation) provided with the system is clear.		
12. It is easy to find information I needed.		
13. The information provided for the system is easy to understand.		
14. The information provided is effective in helping me complete the tasks and scenarios.		
15. The organization of the information on the system screens is clear.		
16. The interface of this system is pleasant.		
17. I like using the interface of this system.		
18. This system has all the functions and capabilities I expect it to have.		
19. Overall, I am satisfied with this system.		
Total Average		

Based on Lewis J.R (1995) IBM Computer Usability Satisfaction Questioners:
Psychometric Evaluation & Instruction for Use

INTERVIEW GUIDE QUESTIONS

Researchers guide for conducting the interview of the respondents:

Office Head:

1. How is the present manual process of San Miguel done?
2. What are the problems that you encounter with the present process?
3. What are the services that the office offers?
4. Who can avail the services?
5. What are the requirements in order to avail in the services?
6. How the records of farmer's saved and retrieved?
7. Where do you keep the records?

Staff:

1. Who is responsible in gathering all information?
2. How long does it take to retrieve the records?
3. Have you encountered difficulties pertaining in recording of information?
4. What are the effects of these problems to problems?
5. What is the information needed by the different commodities?
6. What are the reports needed by the different commodities?
7. Are you willing to accept new technology such as computerization?
8. What is the current process of the farmers' recording management in San Miguel Municipal Agriculture Office?
9. What are the problems encountered by the office head and staff of San Miguel Municipal Agriculture Office in managing records?

10. What is the possible solution to be develop in order to solve the problems encountered?
11. What is the perceived level of acceptability of the developed system by the target users?

Appendix C

Source Code

Login Form

```

Public Class LoginForm
    Private Sub Button1_Click(sender
As Object, e As EventArgs) Handles
Button1.Click
        Dim account As New Accounts
        Dim access As String
        access =
account.checkAccess(username.Text,
password.Text)
        If Not (access = vbNullString)
Then
            Me.Hide()
MainMenuBar.SidebarLogo1.userLa
bel.Text = access.ToUpper()
            MainPage.ShowDialog()
        Else
            MsgBox("Incorrect
credentials")
        End If
    End Sub

    Private Sub Button2_Click(sender
As Object, e As EventArgs) Handles
Button2.Click
        Me.Dispose()
    End Sub

    Private Sub Button3_Click(sender
As Object, e As EventArgs) Handles
Button3.Click
        If (Button3.Text = "Hide") Then
            password.PasswordChar =
"."
            sender.Text = "View"
        Else
            password.PasswordChar = ""
            sender.Text = "Hide"
        End If
    End Sub
End Class

```

Accounts

```

Public Class AccountsForm
    Public accountId As String
    Private Sub closeBtn_Click(sender
As Object, e As EventArgs) Handles
closeBtn.Click
        Private Function SaveFields() As
Accounts
            Dim domain As New Accounts
            domain.account_name =
account_name.Text
            domain.address = address.Text
            domain.position = position.Text
            domain.username =
username.Text
            domain.password =
password.Text
            Return domain
        End Function
    Private Sub
submitBtn_Click(sender As Object, e
As EventArgs) Handles
submitBtn.Click
        If (
            account_name.Text =
vbNullString OrElse
            address.Text = vbNullString
        OrElse
            position.Text = vbNullString
        OrElse
            username.Text = vbNullString
        OrElse
            password.Text = vbNullString
        ) Then
            MessageBox.Show("Input all
fields!!")
        Else
            Dim domain As New
Accounts
            domain = SaveFields()
            Dim id As New Integer

```

```

        id = domain.SaveMe(domain)

    MessageBox.Show("Successfully
saved.")
    Me.Close()

    End If
    End Sub

    Private Sub
    ServicesForm_Load(sender As
Object, e As EventArgs) Handles
MyBase.Load
        Dim domain As New Accounts
        If Not accountId = vbNullString
Then
            domain.getByld(accountId)
        End If
    End Sub

    Private Sub
    updateBtn_Click_1(sender As
Object, e As EventArgs) Handles
updateBtn.Click
        If (
            account_name.Text =
vbNullString OrElse
            address.Text = vbNullString
        OrElse
            position.Text = vbNullString
        OrElse
            username.Text = vbNullString
        OrElse
            password.Text = vbNullString
        ) Then
            MessageBox.Show("Input all
fields!!")
        Else
            Dim domain As New
Accounts
            domain = SaveFields()
            domain.UpdateMe(domain,
accountId)

    MessageBox.Show("Successfully
Updated.")

```

```

    Me.Close()

    End If
    End Sub

    Private Sub
    account_name_TextChanged(sender
As Object, e As EventArgs) Handles
account_name.TextChanged
        If
        System.Text.RegularExpressions.Re
gex.IsMatch(sender.Text, "[ ^ 0-9]")
Then
            sender.Text = ""
        End If
    End Sub

    Private Sub
    address_TextChanged(sender As
Object, e As EventArgs) Handles
address.TextChanged
        If
        System.Text.RegularExpressions.Re
gex.IsMatch(sender.Text, "[ ^ 0-9]")
Then
            sender.Text = ""
        End If
    End Sub

    Private Sub
    position_TextChanged(sender As
Object, e As EventArgs) Handles
position.TextChanged
        If
        System.Text.RegularExpressions.Re
gex.IsMatch(sender.Text, "[ ^ 0-9]")
Then
            sender.Text = ""
        End If
    End Sub

    Private Sub
    username_TextChanged(sender As
Object, e As EventArgs) Handles
username.TextChanged

```

```

    If
    System.Text.RegularExpressions.Regex.IsMatch(sender.Text, "[ ^ 0-9]")
    Then
        sender.Text = ""
    End If
End Sub

End Class

```

MasterFile Fill-up Form

```
Public Class MasterFileList
```

```

    Private Sub
    createBtn_Click_1(sender As Object,
    e As EventArgs) Handles
    createBtn.Click
        Dim farmers As New Farmers
        FarmerForm.Dispose()
        FarmerForm.submitBtn.Visible
    = True
        FarmerForm.updateBtn.Visible
    = False
        FarmerForm.ShowDialog()

```

```

farmers.getAllFarmers(ListView1)
    End Sub

```

```

    Private Sub
    MasterFileList_Load(sender As
    Object, e As EventArgs) Handles
    MyBase.Load
        Dim farmers As New Farmers

farmers.getAllFarmers(ListView1)

    End Sub

```

```

    Private Sub
    ListView1_Click(sender As Object, e
    As EventArgs) Handles
    ListView1.Click

```

```

    If
    (ListView1.SelectedItems.Count > 0)
    Then
        deleteBtn.Visible = True
        editBtn.Visible = True
    Else
        deleteBtn.Visible = False
        editBtn.Visible = False
    End If
End Sub

```

```

    Private Sub editBtn_Click(sender
    As Object, e As EventArgs) Handles
    editBtn.Click
        Dim farmers As New Farmers

```

```

    If
    (ListView1.SelectedItems.Count > 0)
    Then
        FarmerForm.Dispose()
        FarmerForm.farmerId =
    ListView1.SelectedItems(0).SubItems
    (1).Text

```

```

FarmerForm.submitBtn.Visible =
False

```

```

FarmerForm.updateBtn.Visible =
True

```

```

farmers.getAllFarmersById(ListView
1.SelectedItems(0).SubItems(1).Text
)
        FarmerForm.ShowDialog()

```

```

farmers.getAllFarmers(ListView1)
    Else
    End If

```

```
End Sub
```

```

    Private Sub
    deleteBtn_Click(sender As Object, e
    As EventArgs) Handles
    deleteBtn.Click

```

```

        Dim farmers As New Farmers
        FarmerForm.Dispose()
        Dim result As DialogResult =
        MessageBox.Show("Proceed
        delete?", "Delete",
        MessageBoxButtons.YesNo)
        If result = DialogResult.Yes
        Then
            farmers.Delete(farmers,
            ListView1.SelectedItems(0).SubItem
            s(1).Text)

        farmers.getAllFarmers(ListView1)
        End If
        End Sub
    End Class

```

MasterFile View Form

```
Public Class MasterFileList
```

```

    Private Sub
    createBtn_Click_1(sender As Object,
    e As EventArgs) Handles
    createBtn.Click
        Dim farmers As New Farmers
        FarmerForm.Dispose()
        FarmerForm.submitBtn.Visible
        = True
        FarmerForm.updateBtn.Visible
        = False
        FarmerForm.ShowDialog()

        farmers.getAllFarmers(ListView1)
    End Sub

```

```

    Private Sub
    MasterFileList_Load(sender As
    Object, e As EventArgs) Handles
    MyBase.Load
        Dim farmers As New Farmers

        farmers.getAllFarmers(ListView1)

```

```
End Sub
```

```

    Private Sub
    ListView1_Click(sender As Object, e
    As EventArgs) Handles
    ListView1.Click
        If
        (ListView1.SelectedItems.Count > 0)
        Then
            deleteBtn.Visible = True
            editBtn.Visible = True
        Else
            deleteBtn.Visible = False
            editBtn.Visible = False
        End If
    End Sub

```

```

    Private Sub editBtn_Click(sender
    As Object, e As EventArgs) Handles
    editBtn.Click
        Dim farmers As New Farmers

```

```

        If
        (ListView1.SelectedItems.Count > 0)
        Then
            FarmerForm.Dispose()
            FarmerForm.farmerId =
            ListView1.SelectedItems(0).SubItem
            s(1).Text

            FarmerForm.submitBtn.Visible =
            False

```

```

            FarmerForm.updateBtn.Visible =
            True

```

```

            farmers.getAllFarmersById(ListView
            1.SelectedItems(0).SubItems(1).Text
            )
                FarmerForm.ShowDialog()

```

```

            farmers.getAllFarmers(ListView1)
        Else

```

```
End If
```

```

        crop.getAllCropsById(cropId)
    End If

End Sub

Private Sub
deleteBtn_Click(sender As Object, e
As EventArgs) Handles
deleteBtn.Click
    Dim farmers As New Farmers
    FarmerForm.Dispose()
    Dim result As DialogResult =
MessageBox.Show("Proceed
delete?", "Delete",
MessageBoxButtons.YesNo)
    If result = DialogResult.Yes
Then
        farmers.Delete(farmers,
ListView1.SelectedItems(0).SubItem
s(1).Text)

farmers.getAllFarmers(ListView1)
    End If
End Sub
End Class
Crops Fill-up Form
Public Class CropsForm
    Public cropId As String

    Private Sub
closeBtn_Click(sender As Object, e
As EventArgs) Handles
closeBtn.Click
        Me.Close()
    End Sub

    Private Sub
CropsForm_Load(sender As Object,
e As EventArgs) Handles
MyBase.Load
        Dim farmer As New Farmers

farmer.getAllFarmersNameInComb
oBox(farmers)
        Dim crop As New Crops
        If Not cropId = vbNullString
Then
            crop.getAllCropsById(cropId)
        End If
    End Sub

    Private Function SaveFields() As
Crops
        Dim crop As New Crops
        crop.farmers =
farmers.SelectedValue
        crop.barangay = barangay.Text
        crop.specification =
specification.Text
        crop.owner_ship =
ownership.Text
        crop.crop_location =
crop_location.Text
        crop.unit = unit.Text
        crop.total_area_planted =
total_area_planted.Text
        crop.expected_date_of_harvest
=
expected_date_of_harvest.Value.To
String("yyyy/MM/dd")
        crop.date_planted =
date_planted.Value.ToString("yyyy/M
M/dd")
        Return crop
    End Function

    Private Sub
submitBtn_Click(sender As Object, e
As EventArgs) Handles
submitBtn.Click
        If (
            farmers.Text = vbNullString
        OrElse
            crop_location.Text =
vbNullString OrElse
            barangay.Text = vbNullString
        OrElse
            specification.Text =
vbNullString OrElse
            ownership.Text =
vbNullString OrElse

```

```

        unit.Text = vbNullString
    OrElse

    date_planted.Value.ToString("yyyy/M
M/dd") = vbNullString OrElse

    expected_date_of_harvest.Value.To
String("yyyy/MM/dd") = vbNullString
    OrElse
        total_area_planted.Text =
vbNullString
    ) Then
        MessageBox.Show("Input all
fields!!")
    Else
        Dim crop As Crops
        crop = SaveFields()
        Dim id As New Integer
        id = crop.SaveMe(crop)

    MessageBox.Show("Successfully
saved.")
        Me.Close()

    End If
End Sub

Private Sub
updateBtn_Click(sender As Object, e
As EventArgs) Handles
updateBtn.Click
    If (
        farmers.Text = vbNullString
    OrElse
        crop_location.Text =
vbNullString OrElse
        barangay.Text = vbNullString
    OrElse
        specification.Text =
vbNullString OrElse
        ownership.Text =
vbNullString OrElse
        unit.Text = vbNullString
    OrElse

```

```

    date_planted.Value.ToString("yyyy/M
M/dd") = vbNullString OrElse

    expected_date_of_harvest.Value.To
String("yyyy/MM/dd") = vbNullString
    OrElse
        total_area_planted.Text =
vbNullString
    ) Then
        MessageBox.Show("Input all
fields!!")
    Else
        Dim crop As Crops
        crop = SaveFields()
        Dim id As New Integer
        crop.UpdateMe(crop, cropId)

    MessageBox.Show("Successfully
updated.")
        Me.Close()

    End If
End Sub

Private Sub
unit_SelectedIndexChanged(sender
As Object, e As EventArgs) Handles
unit.SelectedIndexChanged

    End Sub

Private Sub
total_area_planted_TextChanged(se
nder As Object, e As EventArgs)
Handles
total_area_planted.TextChanged
    If Not
System.Text.RegularExpressions.Re
gex.IsMatch(total_area_planted.Text
, "[ ^0-9]") Then
        total_area_planted.Text = ""
    End If
End Sub
End Class
Crops View Form

```

```
Public Class CropsList
```

```
    Private Sub
createBtn_Click_2(sender As Object,
e As EventArgs) Handles
createBtn.Click
        CropsForm.Dispose()
        CropsForm.ShowDialog()
        LoadCropsTable()
    End Sub
```

```
    Private Sub
CropsList_Load(sender As Object, e
As EventArgs) Handles
MyBase.Load
        LoadCropsTable()
    End Sub
```

```
    Private Sub editBtn_Click(sender
As Object, e As EventArgs) Handles
editBtn.Click
        Dim crops As New Crops
        CropsForm.Dispose()
        CropsForm.submitBtn.Visible =
False
        CropsForm.updateBtn.Visible =
True
        If
(ListView1.SelectedItems.Count > 0)
Then
            CropsForm.cropld =
ListView1.SelectedItems(0).SubItem
s(0).Text

crops.getAllCropsById(ListView1.Sel
ectedItems(0).SubItems(0).Text)
            CropsForm.ShowDialog()
            LoadCropsTable()
        Else

            End If
    End Sub
```

```
    Private Sub
deleteBtn_Click(sender As Object, e
As EventArgs) Handles
deleteBtn.Click
        If
(ListView1.SelectedItems.Count > 0)
Then
            Dim crops As New Crops
            Dim result As DialogResult =
MessageBox.Show("Proceed
delete?", "Delete",
MessageBoxButtons.YesNo)
            If result = DialogResult.Yes
Then
                crops.Delete(crops,
ListView1.SelectedItems(0).SubItem
s(0).Text)
                LoadCropsTable()
            End If
        End If

    End Sub
```

```
    Private Sub
TextBox1_TextChanged(sender As
Object, e As EventArgs) Handles
TextBox1.TextChanged
        Dim domain As New Crops
        domain.search(ListView1,
TextBox1.Text)
    End Sub
End Class
```

Livestock Fill-up Form

```
Public Class LivestockForm
```

```
    Public livestockId As String
```

```
    Private Sub
closeBtn_Click(sender As Object, e
As EventArgs) Handles
closeBtn.Click
        Me.Close()
    End Sub
```

```
    Private Function SaveFields() As
Livestock
```

```

    Dim livestock As New Livestock
    livestock.farmers =
farmers.SelectedValue
    livestock.barangay =
barangay.Text
    livestock.livestock_type =
livestock_type.Text
    livestock.number_of_livestock =
number_of_livestock.Text
    Return livestock

```

```
End Function
```

```

Private Sub
submitBtn_Click(sender As Object, e
As EventArgs) Handles
submitBtn.Click
    If (
        farmers.Text = vbNullString
    OrElse
        barangay.Text = vbNullString
    OrElse
        livestock_type.Text =
vbNullString OrElse
        number_of_livestock.Text =
vbNullString
    ) Then
        MessageBox.Show("Input all
fields!!")
    Else
        Dim livestock As New
Livestock
        livestock = SaveFields()
        Dim id As New Integer
        id =
livestock.SaveMe(livestock)

```

```

MessageBox.Show("Successfully
saved.")
    Me.Close()

```

```

    End If
End Sub

```

```

Private Sub
LivestockForm_Load(sender As

```

```

Object, e As EventArgs) Handles
MyBase.Load

```

```
    Dim farmer As New Farmers
```

```

farmer.getALLFarmersNameInComb
oBox(farmers)

```

```
    Dim livestock As New Livestock
```

```
    If Not livestockId = vbNullString
```

```
Then
```

```
    livestock.getByld(livestockId)
```

```
End If
```

```
End Sub
```

```
Private Sub
```

```

updateBtn_Click(sender As Object, e
As EventArgs) Handles
updateBtn.Click

```

```
    If (
```

```
        farmers.Text = vbNullString
```

```
    OrElse
```

```
        barangay.Text = vbNullString
```

```
    OrElse
```

```
        livestock_type.Text =
```

```
vbNullString OrElse
```

```
        number_of_livestock.Text =
```

```
vbNullString
```

```
    ) Then
```

```
        MessageBox.Show("Input all
fields!!")
```

```
    Else
```

```
        Dim livestock As New
```

```
Livestock
```

```
        livestock = SaveFields()

```

```

livestock.UpdateMe(livestock,
livestockId)

```

```

MessageBox.Show("Successfully
Updated.")

```

```
    Me.Close()

```

```
    End If
```

```
End Sub
```

```

Private Sub
number_of_livestock_TextChanged(
sender As Object, e As EventArgs)
Handles
number_of_livestock.TextChanged
If Not
System.Text.RegularExpressions.Regex.IsMatch(number_of_livestock.Text, "[ ^ 0-9]") Then
number_of_livestock.Text =
""
End If
End Sub
End Class

```

Livestock View Form

```
Public Class LivestockList
```

```

Private Sub
createBtn_Click_1(sender As Object,
e As EventArgs) Handles
createBtn.Click
LivestockForm.Dispose()
LivestockForm.ShowDialog()
LoadLivestockTable()
End Sub

```

```

Private Sub
LivestockList_Load(sender As
Object, e As EventArgs) Handles
MyBase.Load
LoadLivestockTable()
End Sub

```

```

Private Sub editBtn_Click(sender
As Object, e As EventArgs) Handles
editBtn.Click
Dim livestock As New Livestock
LivestockForm.Dispose()

```

```
LivestockForm.submitBtn.Visible =
False
```

```
LivestockForm.updateBtn.Visible =
True
```

```

If
(ListView1.SelectedItems.Count > 0)
Then
LivestockForm.livestockId =
ListView1.SelectedItems(0).SubItems(0).Text

```

```

livestock.getByld(ListView1.Selected
Items(0).SubItems(0).Text)
LivestockForm.ShowDialog()
LoadLivestockTable()

```

```
Else
```

```

End If
End Sub

```

```

Private Sub
deleteBtn_Click(sender As Object, e
As EventArgs) Handles
deleteBtn.Click

```

```

If
(ListView1.SelectedItems.Count > 0)
Then

```

```

Dim livestock As New
Livestock
Dim result As DialogResult =
MessageBox.Show("Proceed
delete?", "Delete",
MessageBoxButtons.YesNo)

```

```

If result = DialogResult.Yes
Then

```

```

livestock.Delete(livestock,
ListView1.SelectedItems(0).SubItems(0).Text)

```

```
LoadLivestockTable()
```

```
End If
```

```

End If
End Sub

```

```

Private Sub
TextBox1_TextChanged(sender As
Object, e As EventArgs) Handles
TextBox1.TextChanged

```

```
Dim domain As New Livestock
```

```

        domain.search(ListView1,
        TextBox1.Text)
    End Sub
End Class

```

Poultry Fill-up Form

```

Public Class PoultryForm

    Public poultryId As String

    Private Sub
    closeBtn_Click(sender As Object, e
    As EventArgs) Handles
    closeBtn.Click
        Me.Close()
    End Sub

    Private Function SaveFields() As
    Poultry
        Dim poultry As New Poultry
        poultry.farmers =
        farmers.SelectedVAlue
        poultry.barangay =
        barangay.Text
        poultry.poultry_type =
        poultry_type.Text
        poultry.number_of_poultry =
        number_of_poultry.Text
        Return poultry
    End Function

    Private Sub
    submitBtn_Click(sender As Object, e
    As EventArgs) Handles
    submitBtn.Click
        If (
            farmers.Text = vbNullString
        OrElse
            barangay.Text = vbNullString
        OrElse
            poultry_type.Text =
            vbNullString OrElse
            number_of_poultry.Text =
            vbNullString

```

```

        ) Then
            MessageBox.Show("Input all
            fields!!")
        Else
            Dim poultry As Poultry
            poultry = SaveFields()
            Dim id As New Integer
            id = poultry.SaveMe(poultry)

            MessageBox.Show("Successfully
            saved.")
            Me.Close()

        End If
    End Sub

    Private Sub
    PoultryForm_Load(sender As
    Object, e As EventArgs) Handles
    MyBase.Load
        FarmersComboData(farmers)
        Dim poultry As New Poultry
        If Not poultryId = vbNullString
        Then
            poultry.getByld(poultryId)
        End If
    End Sub

    Private Sub
    updateBtn_Click(sender As Object, e
    As EventArgs) Handles
    updateBtn.Click
        If (
            farmers.Text = vbNullString
        OrElse
            barangay.Text = vbNullString
        OrElse
            poultry_type.Text =
            vbNullString OrElse
            number_of_poultry.Text =
            vbNullString
        ) Then
            MessageBox.Show("Input all
            fields!!")
        Else
            Dim poultry As Poultry

```

```

        poultry = SaveFields()
        poultry.UpdateMe(poultry,
poultryId)

```

```

MessageBox.Show("Successfully
Updated.")
    Me.Close()
    End If
    End Sub

```

```

    Private Sub
number_of_poultry_TextChanged(se
nder As Object, e As EventArgs)
Handles
number_of_poultry.TextChanged
    If Not
System.Text.RegularExpressions.Re
gex.IsMatch(number_of_poultry.Text
, "[ ^0-9]") Then
        number_of_poultry.Text = ""
    End If
    End Sub
End Class

```

Poultry View Form

```
Public Class PoultryList
```

```

    Private Sub
createBtn_Click_1(sender As Object,
e As EventArgs) Handles
createBtn.Click
        PoultryForm.ShowDialog()
        LoadPoultryTable()
    End Sub

```

```

    Private Sub
PoultryList_Load(sender As Object,
e As EventArgs) Handles
MyBase.Load
        LoadPoultryTable()
    End Sub

```

```

    Private Sub editBtn_Click(sender
As Object, e As EventArgs) Handles
editBtn.Click
        Dim poultry As New Poultry

```

```

    If
(ListView1.SelectedItems.Count > 0)
Then
        PoultryForm.Dispose()
        PoultryForm.poultryId =
ListView1.SelectedItems(0).SubItem
s(0).Text

```

```

PoultryForm.submitBtn.Visible =
False

```

```

PoultryForm.updateBtn.Visible =
True

```

```

poultry.GetById(ListView1.SelectedIt
ems(0).SubItems(0).Text)
        PoultryForm.ShowDialog()
        LoadPoultryTable()
    Else
    End If
    End Sub

```

```

    Private Sub
deleteBtn_Click(sender As Object, e
As EventArgs) Handles
deleteBtn.Click
        If
(ListView1.SelectedItems.Count > 0)
Then
            Dim poultry As New Poultry
            Dim result As DialogResult =
MessageBox.Show("Proceed
delete?", "Delete",
MessageBoxButtons.YesNo)
            If result = DialogResult.Yes
Then
                poultry.Delete(poultry,
ListView1.SelectedItems(0).SubItem
s(0).Text)
                LoadPoultryTable()
            End If
        End If
    End Sub

```

```

Private Sub
TextBox1_TextChanged(sender As
Object, e As EventArgs) Handles
TextBox1.TextChanged
    Dim domain As New Poultry
    domain.search(ListView1,
TextBox1.Text)
End Sub
End Class

```

Vegetables Fill-up Form

```
Public Class HVCDP
```

```

Private Sub
createBtn_Click_1(sender As Object,
e As EventArgs) Handles
createBtn.Click
    HVCDPForm.Dispose()
    HVCDPForm.ShowDialog()
    LoadVegetablesTable()
End Sub

```

```

Private Sub editBtn_Click(sender
As Object, e As EventArgs) Handles
editBtn.Click
    Dim vege As New Vegetables
    HVCDPForm.Dispose()
    HVCDPForm.submitBtn.Visible
= False
    HVCDPForm.updateBtn.Visible
= True
    If
(ListView1.SelectedItems.Count > 0)
Then
        HVCDPForm.hvcdpfld =
ListView1.SelectedItems(0).SubItem
s(0).Text

vege.getByld(ListView1.Selectedltem
s(0).SubItems(0).Text)
        HVCDPForm.ShowDialog()
        LoadVegetablesTable()
    Else
    End If
End Sub
End Class

```

```

Private Sub
deleteBtn_Click(sender As Object, e
As EventArgs) Handles
deleteBtn.Click
    If
(ListView1.SelectedItems.Count > 0)
Then
        Dim vege As New
Vegetables
        Dim result As DialogResult =
MessageBox.Show("Proceed
delete?", "Delete",
MessageBoxButtons.YesNo)
        If result = DialogResult.Yes
Then
            vege.Delete(vege,
ListView1.SelectedItems(0).SubItem
s(0).Text)
            LoadVegetablesTable()
        End If
    End If
End Sub

```

```

Private Sub
TextBox1_TextChanged(sender As
Object, e As EventArgs) Handles
TextBox1.TextChanged
    Dim domain As New
Vegetables
    domain.search(ListView1,
TextBox1.Text)
End Sub
End Class

```

Vegetables View Form

```
Public Class HVCDPForm
```

```
Public hvcdpfld As String
```

```

Private Sub
closeBtn_Click(sender As Object, e
As EventArgs) Handles
closeBtn.Click
    Me.Close()
End Sub

```

```
Private Function SaveFields() As
Vegetables
```

```
    Dim vege As New Vegetables
```

```
        vege.farmers =
farmers.SelectedValue
        vege.barangay = barangay.Text
        vege.variety_plant =
variety.Text
        vege.owner_ship =
ownership.Text
```

```
        vege.unit = unit.Text
        vege.location =
vegetable_location.Text
        vege.total_area_planted =
total_area_planted.Text
        vege.expected_date_of_harvest
```

```
=
expected_date_of_harvest.Value.To
String("yyyy/MM/dd")
```

```
        vege.date_planted =
date_planted.Value.ToString("yyyy/M
M/dd")
```

```
    Return vege
```

```
End Function
```

```
Private Sub
```

```
submitBtn_Click(sender As Object, e
As EventArgs) Handles
submitBtn.Click
```

```
    If (
```

```
        farmers.Text = vbNullString
```

```
    OrElse
```

```
        barangay.Text = vbNullString
```

```
    OrElse
```

```
        ownership.Text =
vbNullString OrElse
```

```
        unit.Text = vbNullString
```

```
    OrElse
```

```
        vegetable_location.Text =
vbNullString OrElse
```

```
        total_area_planted.Text =
vbNullString OrElse
```

```
        date_planted.Value.ToString("yyyy/M
M/dd") = vbNullString OrElse
```

```
expected_date_of_harvest.Value.To
String("yyyy/MM/dd") = vbNullString
) Then
```

```
    MessageBox.Show("Input all
fields!!")
```

```
Else
```

```
    Dim vege As Vegetables
```

```
    vege = SaveFields()
```

```
    Dim id As New Integer
```

```
    id = vege.SaveMe(vege)
```

```
MessageBox.Show("Successfully
saved.")
```

```
    Me.Close()
```

```
    LoadVegetablesTable()
```

```
End If
```

```
End Sub
```

```
Private Sub
```

```
HVCDPForm_Load(sender As
Object, e As EventArgs) Handles
MyBase.Load
```

```
    FarmersComboData(farmers)
```

```
    Dim vege As New Vegetables
```

```
    If Not hvcdpflid = vbNullString
```

```
Then
```

```
        vege.getByld(hvcdpflid)
```

```
End If
```

```
End Sub
```

```
Private Sub
```

```
updateBtn_Click(sender As Object, e
As EventArgs) Handles
updateBtn.Click
```

```
    If (
```

```
        farmers.Text = vbNullString
```

```
    OrElse
```

```
        barangay.Text = vbNullString
```

```
    OrElse
```

```
        ownership.Text =
vbNullString OrElse
```

```
        unit.Text = vbNullString
```

```
    OrElse
```

```

        vegetable_location.Text =
vbNullString OrElse
        total_area_planted.Text =
vbNullString OrElse

date_planted.Value.ToString("yyyy/M
M/dd") = vbNullString OrElse

expected_date_of_harvest.Value.To
String("yyyy/MM/dd") = vbNullString
) Then
    MessageBox.Show("Input all
fields!!")
Else
    Dim vege As Vegetables
    vege = SaveFields()
    vege.UpdateMe(vege,
hvcdpfld)

MessageBox.Show("Successfully
updated.")
    Me.Close()
    LoadVegetablesTable()

```

```

    End If
End Sub

```

```

Private Sub
total_area_planted_TextChanged(se
nder As Object, e As EventArgs)
Handles
total_area_planted.TextChanged
    If Not
System.Text.RegularExpressions.Re
gex.IsMatch(total_area_planted.Text
, "[ ^0-9]") Then
        total_area_planted.Text = ""
    End If
End Sub
End Class

```

Fruit Trees Fill-up Form

```
Public Class FruitTreesForm
```

```
    Public fruitTreesId As String
```

```

Private Sub
closeBtn_Click(sender As Object, e
As EventArgs) Handles
closeBtn.Click
    Me.Close()
End Sub

```

```

Private Function SaveFields() As
Fruit_Trees
    Dim fruitTrees As New
Fruit_Trees
    fruitTrees.farmers =
farmers.SelectedValue
    fruitTrees.barangay =
barangay.Text
    fruitTrees.fruit_trees_type =
fruit_trees_type.Text
    fruitTrees.quantity =
quantity.Text
    fruitTrees.owner_ship =
ownership.Text
    fruitTrees.location =
trees_location.Text

```

```

fruitTrees.expected_date_of_harvest
=
expected_date_of_harvest.Value.To
String("yyyy/MM/dd")
    fruitTrees.date_planted =
date_planted.Value.ToString("yyyy/M
M/dd")
    Return fruitTrees
End Function

```

```

Private Sub
submitBtn_Click(sender As Object, e
As EventArgs) Handles
submitBtn.Click
    If (
        farmers.Text = vbNullString
    OrElse
        barangay.Text = vbNullString
    OrElse
        trees_location.Text =
vbNullString OrElse

```

```

        quantity.Text = vbNullString
    OrElse
        fruit_trees_type.Text =
vbNullString OrElse
        ownership.Text =
vbNullString OrElse

    date_planted.Value.ToString("yyyy/M
M/dd") = vbNullString OrElse

    expected_date_of_harvest.Value.To
String("yyyy/MM/dd") = vbNullString
    ) Then
        MessageBox.Show("Input all
fields!!")
    Else
        Dim fruitTrees As Fruit_Trees
        fruitTrees = SaveFields()
        Dim id As New Integer
        id =
fruitTrees.SaveMe(fruitTrees)

    MessageBox.Show("Successfully
saved.")
        Me.Close()
        LoadFruitTreesTable()
    End If
End Sub

    Private Sub
updateBtn_Click(sender As Object, e
As EventArgs) Handles
updateBtn.Click
    If (
        farmers.Text = vbNullString
    OrElse
        barangay.Text = vbNullString
    OrElse
        trees_location.Text =
vbNullString OrElse
        fruit_trees_type.Text =
vbNullString OrElse
        quantity.Text = vbNullString
    OrElse
        ownership.Text =
vbNullString OrElse

        date_planted.Value.ToString("yyyy/M
M/dd") = vbNullString OrElse

        expected_date_of_harvest.Value.To
String("yyyy/MM/dd") = vbNullString
    ) Then
        MessageBox.Show("Input all
fields!!")
    Else
        Dim fruitTrees As Fruit_Trees
        fruitTrees = SaveFields()

        fruitTrees.UpdateMe(fruitTrees,
fruitTreesId)

        MessageBox.Show("Successfully
updated.")
        Me.Close()
        LoadFruitTreesTable()

    End If
End Sub

    Private Sub
FruitTreesForm_Load(sender As
Object, e As EventArgs) Handles
MyBase.Load
        FarmersComboData(farmers)
        Dim fruitTrees As New
Fruit_Trees
        If Not fruitTreesId = vbNullString
Then
            fruitTrees.getByld(fruitTreesId)
        End If
    End Sub

    Private Sub
quantity_TextChanged(sender As
Object, e As EventArgs) Handles
quantity.TextChanged
        If Not
System.Text.RegularExpressions.Re
gex.IsMatch(quantity.Text, "[ ^0-9]")
Then

```

```

        quantity.Text = ""
    End If
End Sub
End Class

```

Fruit Trees View Form

```
Public Class FruitTreesList
```

```

    Private Sub
createBtn_Click(sender As Object, e
As EventArgs) Handles
createBtn.Click
        FruitTreesForm.ShowDialog()
        LoadFruitTreesTable()
    End Sub

```

```

    Private Sub editBtn_Click(sender
As Object, e As EventArgs) Handles
editBtn.Click
        Dim fruitTrees As New
Fruit_Trees
        FruitTreesForm.Dispose()

```

```
FruitTreesForm.submitBtn.Visible =
False
```

```

FruitTreesForm.updateBtn.Visible =
True
    If
(ListView1.SelectedItems.Count > 0)
Then

```

```

        FruitTreesForm.fruitTreesId =
ListView1.SelectedItems(0).SubItem
s(0).Text

```

```
fruitTrees.GetById(ListView1.Selecte
dItems(0).SubItems(0).Text)
```

```

FruitTreesForm.ShowDialog()
        LoadFruitTreesTable()
    Else

```

```

    End If
End Sub

```

```

    Private Sub
deleteBtn_Click(sender As Object, e
As EventArgs) Handles
deleteBtn.Click
        If
(ListView1.SelectedItems.Count > 0)
Then
            Dim fruitTrees As New
Fruit_Trees
            Dim result As DialogResult =
MessageBox.Show("Proceed
delete?", "Delete",
MessageBoxButtons.YesNo)
            If result = DialogResult.Yes
Then
                fruitTrees.Delete(fruitTrees,
ListView1.SelectedItems(0).SubItem
s(0).Text)
                LoadFruitTreesTable()
            End If
        End If
    End Sub

```

```

    Private Sub
ListView1_SelectedIndexChanged(s
ender As Object, e As EventArgs)
Handles
ListView1.SelectedIndexChanged
    End Sub

```

```

    Private Sub
TextBox1_TextChanged(sender As
Object, e As EventArgs) Handles
TextBox1.TextChanged
        Dim domain As New
Fruit_Trees
        domain.search(ListView1,
TextBox1.Text)
    End Sub
End Class

```

Services Fill-up Form

```
Public Class ServicesForm
```

```

Public servicesId As String

Private Sub
closeBtn_Click(sender As Object, e
As EventArgs) Handles
closeBtn.Click
    Me.Close()
End Sub

Private Function SaveFields() As
Services
    Dim domain As New Services
    domain.farmers =
farmers.SelectedValue
    domain.barangay =
barangay.Text
    domain.services = services.Text
    domain.date_received =
date_received.Value.ToString("yyyy/
MM/dd")
    Return domain

End Function

Private Sub
submitBtn_Click(sender As Object, e
As EventArgs) Handles
submitBtn.Click
    If (
        farmers.Text = vbNullString
    OrElse
        barangay.Text = vbNullString
    OrElse
        services.Text = vbNullString
    OrElse
        date_received.Value.ToString("yyyy/
MM/dd") = vbNullString
    ) Then
        MessageBox.Show("Input all
fields!!")
    Else
        Dim domain As New
Services
        domain = SaveFields()
        Dim id As New Integer
        id = domain.SaveMe(domain)

        MessageBox.Show("Successfully
saved.")
        Me.Close()

    End If
End Sub

Private Sub
ServicesForm_Load(sender As
Object, e As EventArgs) Handles
MyBase.Load
    Dim farmer As New Farmers

    farmer.getALLFarmersNameInComb
oBox(farmers)
    Dim domain As New Services
    If Not servicesId = vbNullString
Then
        domain.getById(servicesId)
    End If
End Sub

Private Sub
updateBtn_Click_1(sender As
Object, e As EventArgs) Handles
updateBtn.Click
    If (
        farmers.Text = vbNullString
    OrElse
        barangay.Text = vbNullString
    OrElse
        services.Text = vbNullString
    OrElse
        date_received.Value.ToString("yyyy/
MM/dd") = vbNullString
    ) Then
        MessageBox.Show("Input all
fields!!")
    Else
        Dim domain As New
Services

```

```

        domain = SaveFields()
        domain.UpdateMe(domain,
servicesId)

```

```

MessageBox.Show("Successfully
Updated.")
Me.Close()

```

```

    End If
End Sub
End Class

```

Services View Form

```

Public Class ServicesList

```

```

    Private Sub
createBtn_Click_1(sender As Object,
e As EventArgs) Handles
createBtn.Click

```

```

        ServicesForm.Dispose()
        ServicesForm.ShowDialog()
        LoadServicesTable()

```

```

    End Sub

```

```

    Private Sub editBtn_Click(sender
As Object, e As EventArgs) Handles
editBtn.Click

```

```

        Dim domain As New Services
        ServicesForm.Dispose()
        ServicesForm.submitBtn.Visible
= False

```

```

        ServicesForm.updateBtn.Visible
= True
    If

```

```

(ListView1.SelectedItems.Count > 0)
Then

```

```

        ServicesForm.servicesId =
ListView1.SelectedItems(0).SubItem
s(0).Text

```

```

domain.getByld(ListView1.Selectedl
tems(0).SubItems(0).Text)

```

```

        ServicesForm.ShowDialog()
        LoadServicesTable()

```

```

    Else

```

```

    End If
End Sub

```

```

    Private Sub
deleteBtn_Click(sender As Object, e
As EventArgs) Handles
deleteBtn.Click

```

```

    If
(ListView1.SelectedItems.Count > 0)
Then

```

```

        Dim domain As New
Services

```

```

        Dim result As DialogResult =
MessageBox.Show("Proceed
delete?", "Delete",
MessageBoxButtons.YesNo)

```

```

        If result = DialogResult.Yes
Then

```

```

            domain.Delete(domain,
ListView1.SelectedItems(0).SubItem
s(0).Text)

```

```

            LoadServicesTable()

```

```

        End If

```

```

    End If

```

```

End Sub

```

```

    Private Sub
ServicesList_Load(sender As Object,
e As EventArgs) Handles
MyBase.Load

```

```

        LoadServicesTable()

```

```

    End Sub

```

```

    Private Sub
TextBox1_TextChanged(sender As
Object, e As EventArgs) Handles
TextBox1.TextChanged

```

```

        Dim domain As New Services
        domain.search(ListView1,
TextBox1.Text)

```

```

    End Sub

```

```

End Class

```

Farmers Visit Fill-up Form

```

Public Class FarmersVisits

```

```

Public visitsId As String

Private Sub
closeBtn_Click(sender As Object, e
As EventArgs) Handles
closeBtn.Click
    Me.Close()
End Sub

Private Function SaveFields() As
Visits
    Dim domain As New Visits
    domain.farmers =
farmers.SelectedValue
    domain.barangay =
barangay.Text
    domain.concern = concern.Text
    domain.status = status.Text
    domain.date_of_visits =
date_of_visits.Value.ToString("yyyy/
MM/dd")
    Return domain

End Function

Private Sub
submitBtn_Click(sender As Object, e
As EventArgs) Handles
submitBtn.Click
    If (
        farmers.Text = vbNullString
    OrElse
        barangay.Text = vbNullString
    OrElse
        concern.Text = vbNullString
    OrElse
        status.Text = vbNullString
    OrElse
        date_of_visits.Value.ToString("yyyy/
MM/dd") = vbNullString
    ) Then
        MessageBox.Show("Input all
fields!!")
    Else
        Dim domain As New Visits
        domain = SaveFields()
        Dim id As New Integer
        id = domain.SaveMe(domain)

        MessageBox.Show("Successfully
saved.")
        Me.Close()

    End If
End Sub

Private Sub
ServicesForm_Load(sender As
Object, e As EventArgs) Handles
MyBase.Load
    Dim farmer As New Farmers

    farmer.getALLFarmersNameInComb
oBox(farmers)
    Dim domain As New Visits
    If Not visitsId = vbNullString
Then
        domain.getById(visitsId)
    End If
End Sub

Private Sub
updateBtn_Click_1(sender As
Object, e As EventArgs) Handles
updateBtn.Click
    If (
        farmers.Text = vbNullString
    OrElse
        barangay.Text = vbNullString
    OrElse
        concern.Text = vbNullString
    OrElse
        status.Text = vbNullString
    OrElse
        date_of_visits.Value.ToString("yyyy/
MM/dd") = vbNullString
    ) Then
        MessageBox.Show("Input all
fields!!")
    Else

```

```

Else
    Dim domain As New Visits
    domain = SaveFields()
    domain.UpdateMe(domain,
visitsId)

MessageBox.Show("Successfully
Updated.")
    Me.Close()

End If
End Sub

Private Sub
concern_TextChanged(sender As
Object, e As EventArgs) Handles
concern.TextChanged
    If
System.Text.RegularExpressions.Re
gex.IsMatch(sender.Text, "[ ^ 0-9]")
Then
    sender.Text = ""
    End If
End Sub
End Class
Farmers Visit View Form
Public Class FarmersVisitsList

Private Sub editBtn_Click(sender
As Object, e As EventArgs) Handles
editBtn.Click
    Dim domain As New Visits
    FarmersVisits.Dispose()
    FarmersVisits.submitBtn.Visible
= False
    FarmersVisits.updateBtn.Visible
= True
    If
(ListView1.SelectedItems.Count > 0)
Then
        FarmersVisits.visitsId =
ListView1.SelectedItems(0).SubItem
s(0).Text

domain.getByld(ListView1.SelectedIt
ems(0).SubItems(0).Text)

```

```

FarmersVisits.ShowDialog()
LoadVisitsTable()
Else
End If
End Sub

Private Sub
deleteBtn_Click(sender As Object, e
As EventArgs) Handles
deleteBtn.Click
    If
(ListView1.SelectedItems.Count > 0)
Then
        Dim domain As New Visits
        Dim result As DialogResult =
MessageBox.Show("Proceed
delete?", "Delete",
MessageBoxButtons.YesNo)
        If result = DialogResult.Yes
Then
            domain.Delete(domain,
ListView1.SelectedItems(0).SubItem
s(0).Text)
            LoadVisitsTable()
        End If
    End If
End Sub

Private Sub
ServicesList_Load(sender As Object,
e As EventArgs) Handles
MyBase.Load
    LoadVisitsTable()
End Sub

Private Sub
createBtn_Click(sender As Object, e
As EventArgs) Handles
createBtn.Click
    FarmersVisits.Dispose()
    FarmersVisits.ShowDialog()
    LoadVisitsTable()
End Sub

```

```

Private Sub
TextBox1_TextChanged(sender As
Object, e As EventArgs) Handles
TextBox1.TextChanged
    Dim domain As New Visits
    domain.search(ListView1,
TextBox1.Text)
End Sub
End Class

```

Db Config

```

Imports System.Data
Imports MySql.Data
Imports MySql.Data.MySqlClient
Imports System.Reflection

Module DbConfig

    Dim ConnStr As String =
"server=127.0.0.1;" _
    & "uid=root;" _
    & "pwd=;" _
    & "database=farms"

    Public DbConnection As New
MySqlConnection(ConnStr)

    Public Sub OpenConnection()
    Try

Console.WriteLine("Connection
Open")
        DbConnection.Open()
        Catch ex As
MySql.Data.MySqlClient.MySqlExce
ption

MessageBox.Show(ex.Message)
    End Try
    End Sub

    Public Sub CloseConnection()
    Try

Console.WriteLine("Connection
Open")

```

```

        DbConnection.Close()
        Catch ex As
MySql.Data.MySqlClient.MySqlExce
ption

MessageBox.Show(ex.Message)
    End Try
    End Sub

    Public Function Save(ByVal
Domain As Object) As Integer
    Try
        Dim SqlQuery = "INSERT
INTO " +
Domain.GetType.Name.ToString().T
oLower + " VALUES("
        Dim info() As PropertyInfo =
Domain.GetType().GetProperties()
        Dim count As Integer
count = 0

        For Each i As PropertyInfo In
info
            If (count = 0) Then
                SqlQuery = SqlQuery +
"@ " + i.Name
            Else
                SqlQuery = SqlQuery +
"," + "@" + i.Name
            End If
            count = count + 1
            If (info.Length = count)
Then
                SqlQuery = SqlQuery +
"); SELECT LAST_INSERT_ID()"
            End If
        Next
        OpenConnection()
        Dim SQLcmd = New
MySqlCommand(SqlQuery,
DbConnection)

        For Each i As PropertyInfo In
info
            SQLcmd.Parameters.AddWithValue(

```

```

Domain.getFormField(i),
i.GetValue(Domain,
Nothing).ToString())
    Next
    Dim cmd_result As Integer =
CInt(SQLcmd.ExecuteScalar())
    CloseConnection()
    Return cmd_result
Catch ex As
MySql.Data.MySqlClient.MySqlExce
ption

MessageBox.Show(ex.Message)
    Return 0
End Try
End Function

Public Sub Update(ByVal Domain
As Object, ByVal Id As String)
    Try
        Dim SqlQuery = "UPDATE "
+
Domain.GetType.Name.ToString().T
oLower + " SET "
        Dim info() As PropertyInfo =
Domain.GetType().GetProperties()
        Dim count As Integer
        count = 0

        For Each i As PropertyInfo In
info
            If Not (i.Name = "id"
OrElse i.Name = "image_file") Then
                If (count = 0) Then
                    SqlQuery = SqlQuery
+ i.Name + " = @" + i.Name
                Else
                    SqlQuery = SqlQuery
+ "," + i.Name + " = @" + i.Name
                End If
                count = count + 1
            End If
        Next

        SqlQuery = SqlQuery + "
WHERE id = @id"

```

```

OpenConnection()
Dim SQLcmd = New
MySqlCommand(SqlQuery,
DbConnection)

    For Each i As PropertyInfo In
info
        If Not (i.Name = "id"
OrElse i.Name = "image_file") Then

SQLcmd.Parameters.AddWithValue(
Domain.getFormField(i),
i.GetValue(Domain,
Nothing).ToString())
        End If
    Next

SQLcmd.Parameters.AddWithValue(
"@id", Id)
    SQLcmd.ExecuteNonQuery()
    CloseConnection()
    Catch ex As
MySql.Data.MySqlClient.MySqlExce
ption

MessageBox.Show(ex.Message)
    End Try
End Sub

Public Sub Delete(ByVal Domain
As Object, ByVal Id As String)
    Try
        Dim SqlQuery = "DELETE
FROM " +
Domain.GetType.Name.ToString().T
oLower + " WHERE id = @id"
        OpenConnection()
        Dim SQLcmd = New
MySqlCommand(SqlQuery,
DbConnection)

SQLcmd.Parameters.AddWithValue(
"@id", Id)
        SQLcmd.ExecuteNonQuery()
        CloseConnection()

```

```

    Catch ex As
    MySql.Data.MySqlClient.MySqlExce
    ption

```

```

    MessageBox.Show(ex.Message)
    End Try
    End Sub

```

```

    Public Function
    CreateQuery(ByVal SqlQuery As
    String) As DataTable
        OpenConnection()
        Dim table As New DataTable
        Dim adapter As New
    MySqlDataAdapter(SqlQuery,
    DbConnection)
        adapter.Fill(table)
        CloseConnection()
        Return table
    End Function

```

```

    Public Function
    CreateQueryObject(ByVal SqlQuery
    As String) As MySqlDataReader
        OpenConnection()
        Dim reader As
    MySqlDataReader
        Dim cmd As New
    MySqlCommand
        cmd.Connection =
    DbConnection
        cmd.CommandText = SqlQuery
        reader = cmd.ExecuteReader()
        Return reader
    End Function

```

```

    Public Sub SavePicture(ByVal
    PictureBox1 As PictureBox, ByVal id
    As String)
        Dim OpenFileDialog1 As New
    OpenFileDialog

```

```

        Dim cmd As MySqlCommand
        Dim sql As String

```

```

        Dim result As Integer
        Dim caption As String
        Dim arrImage() As Byte
        Dim mstream As New
    System.IO.MemoryStream()

```

```

        PictureBox1.Image =
    Resize(PictureBox1.Image, 80, 0)
        caption =
    System.IO.Path.GetFileName(Open
    OpenFileDialog1.FileName)
        'SPECIFIES THE FILE
    FORMAT OF THE IMAGE

```

```

        PictureBox1.Image.Save(mstream,
    System.Drawing.Imaging.ImageFor
    mat.Jpeg)

```

```

        'RETURNS THE ARRAY OF
    UNSIGNED BYTES FROM WHICH
    THIS STREAM WAS CREATED
        arrImage = mstream.GetBuffer()

```

```

        'GET THE SIZE OF THE
    STREAM IN BYTES
        Dim FileSize As UInt32
        FileSize = mstream.Length
        'CLOSES THE CURRENT
    STREAM AND RELEASE ANY
    RESOURCES ASSOCIATED WITH
    THE CURRENT STREAM
        mstream.Close()
    Try

```

```

        OpenConnection()
        sql = "UPDATE
    `farms`.`farmers` SET image_file =
    @ImageFile WHERE id =" + id
        cmd = New MySqlCommand
    With cmd
        .Connection =
    DbConnection
        .CommandText = sql

```

```

        .Parameters.AddWithValue("@Imag
    eFile", arrImage)

```

```

        result =
        .ExecuteNonQuery()
        End With
        If result > 0 Then
        Else
            MsgBox("Error query",
MsgBoxStyle.Exclamation)
        End If
        Catch ex As Exception
            MsgBox(ex.Message)
        Finally
            CloseConnection()
        End Try

    End Sub

    Public Function Resize(ByVal
imgSource As
System.Drawing.Image, ByVal
intPercent As Integer, ByVal intType
As Integer) As
System.Drawing.Image
        'resize the image by percent
        Dim imgOutput As
System.Drawing.Image
        Dim intX, intY As Integer
        intX = Int(imgSource.Width /
100 * intPercent)
        intY = Int(imgSource.Height /
100 * intPercent)
        Dim bm As Drawing.Bitmap =
New System.Drawing.Bitmap(intX,
intY)
        Dim g As
System.Drawing.Graphics =
Drawing.Graphics.FromImage(bm)

        Select Case intType
            Case 0
                g.InterpolationMode =
Drawing.Drawing2D.InterpolationMo
de.Default
            Case 1

```

```

                g.InterpolationMode =
Drawing.Drawing2D.InterpolationMo
de.High
            Case 2
                g.InterpolationMode =
Drawing.Drawing2D.InterpolationMo
de.HighQualityBilinear
            Case 3
                g.InterpolationMode =
Drawing.Drawing2D.InterpolationMo
de.HighQualityBicubic
            End Select

        g.DrawImage(imgSource, 0, 0,
intX, intY)
        imgOutput = bm
        Return imgOutput
    End Function

    Public Sub
getDashboardCount(ByVal listView
As ListView)
        Dim table As DataTable =
CreateQuery("SELECT * FROM
`dashboard_count`")
        listView.Columns.Clear()
        listView.Items.Clear()

        Dim count As Integer
        count = 0

        For Each columns As
DataColumn In table.Columns

listView.Columns.Add(columns.Colu
mnName.ToString(), 1)
            Next

            Dim lstviewItemImageList As
ImageList = New ImageList

            For Each rows As DataRow In
table.Rows
                Dim list As New
ListViewItem()

```

```

    For Each columns As
DataColumn In table.Columns

```

```

        If Not
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then

```

```

list.SubItems.Add(rows(table.Column
s.IndexOf(columns.ColumnName.To
String()))

```

```

        Else
            list.Text =
rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
        End If

```

```

    Next

```

```

    listView.Items.Add(list)

```

```

listView.AutoResizeColumns(Column
HeaderAutoResizeStyle.HeaderSiz
e)

```

```

    Next

```

```

End Sub
End Module

```

Files Module

```

Module FilesModule

```

```

    Public Sub
SetFileMenuActive(currentBtn As
Button)

```

```

FilesBar.masterfileBtn.BackColor =
Color.White
    FilesBar.cropsBtn.BackColor =
Color.White
    FilesBar.livestockBtn.BackColor
= Color.White
    FilesBar.fruitBtns.BackColor =
Color.White
    FilesBar.hvcdpBtn.BackColor =
Color.White

```

```

FilesBar.poultryBtn.BackColor =
Color.White

```

```

FilesBar.serviceReceiveBtn.BackCol
or = Color.White

```

```

FilesBar.masterfileBtn.ForeColor =
Color.Black

```

```

    FilesBar.cropsBtn.ForeColor =
Color.Black

```

```

    FilesBar.livestockBtn.ForeColor
= Color.Black

```

```

    FilesBar.fruitBtns.ForeColor =
Color.Black

```

```

    FilesBar.hvcdpBtn.ForeColor =
Color.Black

```

```

    FilesBar.poultryBtn.ForeColor =
Color.Black

```

```

FilesBar.serviceReceiveBtn.ForeCol
or = Color.Black

```

```

'FilesBar.masterfileIcon.BackColor =
Color.FromArgb(126, 88, 68)

```

```

'FilesBar.cropsIcon.BackColor =
Color.FromArgb(126, 88, 68)

```

```

'FilesBar.livestockIcon.BackColor =
Color.FromArgb(126, 88, 68)

```

```

'FilesBar.hvcdpIcon.BackColor
= Color.FromArgb(126, 88, 68)

```

```

'FilesBar.servicesIcon.BackColor =
Color.FromArgb(126, 88, 68)

```

```

        currentBtn.BackColor =
Color.FromArgb(148, 188, 99)

```

```

        currentBtn.ForeColor =
Color.White

```

```

    End Sub

```

```

    Public Sub LoadLivestockTable()
        Dim livestock As New Livestock

```

```
livestock.getAll(LivestockList.ListView1)
```

```
End Sub
```

```
Public Sub LoadCropsTable()  
    Dim crops As New Crops
```

```
crops.getAllCrops(CropsList.ListView1)
```

```
End Sub
```

```
Public Sub LoadPoultryTable()  
    Dim poultry As New Poultry
```

```
poultry.getAll(PoultryList.ListView1)
```

```
End Sub
```

```
Public Sub LoadFruitTreesTable()  
    Dim fruitTrees As New  
    Fruit_Trees
```

```
fruitTrees.getAll(FruitTreesList.ListView1)
```

```
End Sub
```

```
Public Sub  
LoadVegetablesTable()  
    Dim vege As New Vegetables  
    vege.getAll(HVCDP.ListView1)
```

```
End Sub
```

```
Public Sub LoadServicesTable()  
    Dim domain As New Services
```

```
domain.getAll(ServicesList.ListView1)
```

```
End Sub
```

```
Public Sub  
FarmersComboData(ByVal combo  
As ComboBox)  
    Dim farmer As New Farmers
```

```
farmer.getALLFarmersNameInComb  
oBox(combo)
```

```
End Sub
```

```
End Module
```

Main Menu Module

```
Module MainMenuModule
```

```
Public Sub SwitchPanel(ByVal  
panel As Form, ByVal title As String)
```

```
MainPage.MainPanel.Controls.Clear  
(
```

```
    panel.TopLevel = False  
    MainPage.pageTitle.Text = title
```

```
MainPage.MainPanel.Controls.Add(p  
anel)
```

```
    panel.WindowState =  
FormWindowState.Maximized  
    panel.Show()
```

```
End Sub
```

```
Public Sub  
SwitchSidebarPanel(ByVal panel As  
Form)
```

```
MainPage.MenuPanel.Controls.Clea  
r()
```

```
    panel.TopLevel = False
```

```
MainPage.MenuPanel.Controls.Add(  
panel)
```

```
    panel.WindowState =  
FormWindowState.Maximized  
    panel.Show()
```

```
End Sub
```

```
End Module
```

Reports Module

```
Module ReportsModule
```

```

Public Sub
SetReportMenuActive(currentBtn As
Button)

ReportsBar.masterlistReportBtn.Back
Color = Color.White

ReportsBar.cropsGrowersReportBtn.
BackColor = Color.White

ReportsBar.livestockReportsBtn.Back
Color = Color.White

ReportsBar.poultryReportsBtn.Back
Color = Color.White

ReportsBar.vegetablesReportBtn.Ba
ckColor = Color.White

ReportsBar.fruitReportsBtns.BackCo
lor = Color.White

ReportsBar.RSBSAReportBtn.BackC
olor = Color.White

ReportsBar.GraphicalReportBtns.Ba
ckColor = Color.White

ReportsBar.masterlistReportBtn.Fore
Color = Color.Black

ReportsBar.cropsGrowersReportBtn.
ForeColor = Color.Black

ReportsBar.livestockReportsBtn.For
eColor = Color.Black

ReportsBar.poultryReportsBtn.ForeC
olor = Color.Black

ReportsBar.vegetablesReportBtn.For
eColor = Color.Black

ReportsBar.fruitReportsBtns.ForeCol
or = Color.Black

ReportsBar.RSBSAReportBtn.ForeC
olor = Color.Black

ReportsBar.GraphicalReportBtns.For
eColor = Color.Black

    currentBtn.BackColor =
Color.FromArgb(148, 188, 99)
    currentBtn.ForeColor =
Color.White
End Sub

End Module

Accounts Domain

Imports System.Reflection
Imports System.IO
Imports MySql.Data
Imports MySql.Data.MySqlClient

Public Class Accounts : Inherits
JPAClass

    Public Sub New()
        Me.id = vbNullString
        Me.account_name = ""
        Me.address = ""
        Me.position = ""
        Me.username = ""
        Me.password = ""

    End Sub

    Public Overrides Function
SetDomain() As Object
        Dim livestock As New Accounts
        Return livestock
    End Function

    Public Property id As Integer

```

```

Public Property account_name As
String
Public Property address As String
Public Property position As String
Public Property username As
String
Public Property password As
String

```

```

Public Function
getFormField(ByVal column As
PropertyInfo) As String
Return "@" + column.Name
End Function

```

```

Public Sub getAll(ByVal listView
As ListView)
Dim table As DataTable =
MyBase.CreateQuery("SELECT `id`
as `ID`,`account_name` as
`ACCOUNT NAME`,`address` AS
`ADDRESS`,`position` AS
`POSITION`,`username` AS
`USERNAME`,`password` AS
`PASSWORD` FROM `accounts`")
listView.Columns.Clear()
listView.Items.Clear()

```

```

Dim count As Integer
count = 0

```

```

For Each columns As
DataColumn In table.Columns

listView.Columns.Add(columns.Column
Name.ToString(), 1,
HorizontalAlignment.Center)
Next

```

```

Dim lstviewItemImageList As
ImageList = New ImageList

```

```

For Each rows As DataRow In
table.Rows
Dim list As New
ListViewItem()

```

```

For Each columns As
DataColumn In table.Columns

```

```

If Not
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then

```

```

list.SubItems.Add(rows(table.Column
s.IndexOf(columns.ColumnName.To
String())))

```

```

Else
list.Text =
rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
End If

```

```

Next

```

```

listView.Items.Add(list)

```

```

listView.AutoResizeColumns(Column
HeaderAutoResizeStyle.HeaderSiz
e)

```

```

Next

```

```

End Sub

```

```

Public Sub search(ByVal listView
As ListView, ByVal search As String)

```

```

Dim table As DataTable =
MyBase.CreateQuery("SELECT *
FROM `accounts` where
(`account_name` like '%' + search +
`%' or `username` like '%' + search
+ "%')")

```

```

listView.Columns.Clear()
listView.Items.Clear()

```

```

Dim count As Integer
count = 0

```

```

For Each columns As
DataColumn In table.Columns

```

```

listView.Columns.Add(columns.ColumnName.ToString(), 1)
    Next

    Dim lstviewItemImageList As
    ImageList = New ImageList

    For Each rows As DataRow In
    table.Rows
        Dim list As New
        ListViewItem()

        For Each columns As
        DataColumn In table.Columns

            If Not
            (table.Columns.IndexOf(columns.Col
            umnName.ToString()) = 0) Then

                list.SubItems.Add(rows(table.Column
                s.IndexOf(columns.ColumnName.To
                String())))
            Else
                list.Text =
                rows(table.Columns.IndexOf(column
                s.ColumnName.ToString()))
            End If

        Next

        listView.Items.Add(list)

    listView.AutoResizeColumns(Column
    nHeaderAutoResizeStyle.HeaderSiz
    e)
    Next

End Sub

Public Sub getByld(ByVal id As
String)
    Dim query As
    MySqlDataReader =
    MyBase.CreateQueryObject("SELEC

```

```

T * from `accounts` I WHERE I.id = "
+ id)

    If query.HasRows Then
        While (query.Read())

            AccountsForm.account_name.Text =
            query("account_name")

            AccountsForm.address.Text =
            query("address")

            AccountsForm.position.Text =
            query("position")

            AccountsForm.username.Text =
            query("username")

            AccountsForm.password.Text =
            query("password")

        End While
    End If
    DbConfig.CloseConnection()

End Sub

Public Function
checkAccess(ByVal username As
String, ByVal password As String) As
String
    Dim query As
    MySqlDataReader =
    MyBase.CreateQueryObject("SELEC
T * from `accounts` I WHERE
I.username = " + username + " and
password = " + password + """)
    Dim accountname As String
    accountname = ""
    If query.HasRows Then
        While (query.Read())
            accountname =
            query("account_name")
        End While
    DbConfig.CloseConnection()
    Return accountname

```

```

Else
    DbConfig.CloseConnection()
    Return accountname
End If
End Function

End Class

```

Crops Domain

```

Imports System.Reflection
Imports System.IO
Imports MySql.Data
Imports MySql.Data.MySqlClient

Public Class Crops : Inherits JPAClass

    Public Sub New()
        Me.id = vbNullString
        Me.farmers = ""
        Me.barangay = ""
        Me.specification = ""
        Me.owner_ship = ""
        Me.unit = ""
        Me.total_area_planted = 0
        Me.crop_location = ""
        Me.date_planted = New Date()
        Me.expected_date_of_harvest =
= New Date()
    End Sub

    Public Overrides Function
SetDomain() As Object
        Dim crops As New Crops
        Return crops
    End Function

    Public Property id As Integer
    Public Property farmers As String
    Public Property barangay As
String
    Public Property specification As
String

```

```

    Public Property owner_ship As
String
    Public Property unit As String
    Public Property total_area_planted
As Decimal
    Public Property crop_location As
String
    Public Property date_planted As
String
    Public Property
expected_date_of_harvest As String

```

```

    Public Function
getFormField(ByVal column As
PropertyInfo) As String
        Return "@" + column.Name
    End Function

```

```

    Public Sub getAllCrops(ByVal
listView As ListView)
        Dim table As DataTable =
MyBase.CreateQuery("SELECT *
FROM `farmers_crops`")
        listView.Columns.Clear()
        listView.Items.Clear()

        Dim count As Integer
        count = 0

        For Each columns As
DataColumn In table.Columns

listView.Columns.Add(columns.Colu
mnName.ToString(), 1,
HorizontalAlignment.Center)
            Next

            Dim lstviewItemImageList As
ImageList = New ImageList

```

```

        For Each rows As DataRow In
table.Rows
            Dim list As New
ListViewItem()

```

```

        For Each columns As
DataColumn In table.Columns

            If Not
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then

list.SubItems.Add(rows(table.Column
s.IndexOf(columns.ColumnName.To
String()))))
            Else
                list.Text =
rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
            End If

        Next

        listView.Items.Add(list)

listView.AutoResizeColumns(Column
HeaderAutoResizeStyle.HeaderSiz
e)
        Next

    End Sub

    Public Sub search(ByVal listView
As ListView, ByVal search As String)
        Dim table As DataTable =
MyBase.CreateQuery("SELECT *
FROM `farmers_crops` where
(`FARMERS` like '%" + search + "%'
or SPECIFICATION like '%" +
search + "%')")
        listView.Columns.Clear()
        listView.Items.Clear()

        Dim count As Integer
count = 0

        For Each columns As
DataColumn In table.Columns

listView.Columns.Add(columns.Colu
mnName.ToString(), 1)

```

```

        Next

        Dim lstviewItemImageList As
ImageList = New ImageList

        For Each rows As DataRow In
table.Rows
            Dim list As New
ListViewItem()

            For Each columns As
DataColumn In table.Columns

                If Not
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then

list.SubItems.Add(rows(table.Column
s.IndexOf(columns.ColumnName.To
String()))))
                Else
                    list.Text =
rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
                End If

            Next

            listView.Items.Add(list)

listView.AutoResizeColumns(Column
HeaderAutoResizeStyle.HeaderSiz
e)
        Next

    End Sub

    Public Sub getAllCropsById(ByVal
id As String)
        Dim query As
MySqlDataReader =
MyBase.CreateQueryObject("SELEC
T c.`id`, CONCAT(`l_name`,',
`,`f_name`,``,`m_name`,``,`e_name`)
as `farmers`, c.`barangay`,
`specification`, `owner_ship`, `unit`,

```

```

`total_area_planted`, `crop_location`,
`date_planted`,
`expected_date_of_harvest` FROM
`farms`.`crops` `c` join farmers f on
f.id = c.farmers WHERE c.id = " + id)

```

```

    If query.HasRows Then
        While (query.Read())

```

```

CropsForm.farmers.SelectedIndex =
CropsForm.farmers.FindStringExact(
query("farmers"))

```

```

CropsForm.crop_location.Text =
query("barangay")

```

```

CropsForm.specification.Text =
query("specification")

```

```

CropsForm.ownership.Text =
query("owner_ship")
        CropsForm.unit.Text =
query("unit")

```

```

CropsForm.total_area_planted.Text
= query("total_area_planted")
        CropsForm.barangay.Text
= query("crop_location")

```

```

CropsForm.date_planted.Text =
query("date_planted")

```

```

CropsForm.expected_date_of_harve
st.Text =
query("expected_date_of_harvest")

```

```

        End While
    End If
    DbConfig.CloseConnection()

```

```

End Sub

```

```

End Class

```

Farmers Domain

```

Imports System.Reflection

```

```

Imports System.IO
Imports MySql.Data
Imports MySql.Data.MySqlClient
Imports System.Globalization

```

```

Public Class Farmers : Inherits
JPAClass

```

```

    Public Sub New()

```

```

        Me.id = vbNullString
        Me.l_name = ""
        Me.f_name = ""
        Me.m_name = ""
        Me.e_name = ""
        Me.purok_sitio = ""
        Me.barangay = ""
        Me.municipality = ""
        Me.province = ""
        Me.region = 7
        Me.date_of_birth = New Date()
        Me.gender = ""
        Me.mobile_no = ""
        Me.place_of_birth = ""
        Me.highest_education = ""
        Me.civil_status = ""
        Me.spouse_name = ""
        Me.main_livelihood = ""
        Me.religion = ""
        Me.emergency_contact_person
= ""

```

```

        Me.emergency_contact_number = ""
        Me.pwd = 0
        Me.indigent = 0
        Me.four_ps = 0
        Me.rsbsa = 0
        Me.image_file() = vbByte
        Me.date_created =
Date.Now.ToString("yyyy/MM/dd")

```

```

    End Sub

```

```

    Public Overrides Function
SetDomain() As Object

```

```

        Dim farmer As New Farmers
        Return farmer
    End Function

    Public Property id As Integer
    Public Property l_name As String
    Public Property f_name As String
    Public Property m_name As String
    Public Property e_name As String
    Public Property purok_sitio As
String
    Public Property barangay As
String
    Public Property municipality As
String
    Public Property province As String
    Public Property region As Integer
    Public Property date_of_birth As
String
    Public Property gender As String
    Public Property mobile_no As
String
    Public Property place_of_birth As
String
    Public Property highest_education
As String
    Public Property civil_status As
String
    Public Property spouse_name As
String
    Public Property main_livelihood
As String
    Public Property religion As String
    Public Property
emergency_contact_person As
String
    Public Property
emergency_contact_number As
String
    Public Property pwd As Integer
    Public Property indigent As
Integer
    Public Property four_ps As Integer
    Public Property rsbsa As Integer

    Public Property image_file() As
Byte
    Public Property date_created As
String

    Public Function
getFormField(ByVal column As
PropertyInfo) As String
        Return "@" + column.Name
    End Function

    Public Sub
getALLFarmersNameInComboBox(By
yVal combo As ComboBox)
        Dim table As DataTable =
MyBase.CreateQuery("SELECT id ,
CONCAT(`l_name`,`f_name`,`
m_name`,`e_name`) as 'name'
FROM `farmers`")
        Dim comboSource As New
Dictionary(Of String, String)()
        comboSource.Add(0, "")
        For Each rows As DataRow In
table.Rows

            comboSource.Add(rows(table.Column
s.IndexOf("id")),
rows(table.Columns.IndexOf("name")
))

            Next
            combo.DataSource = New
BindingSource(comboSource,
Nothing)
            combo.DisplayMember =
"Value"
            combo.ValueMember = "Key"
        End Sub

    Public Sub getAllFarmers(ByVal
listView As ListView)
        Dim table As DataTable =
MyBase.CreateQuery("SELECT
`image_file` as 'Image',id as 'ID',
CONCAT(`l_name`,`f_name`,`
m_name`,`e_name`) as 'Full
Name', `purok_sitio` as 'Purok or

```

```

Sitio', `barangay` as 'Barangay',
`municipality` as 'Municipality',
`province` as 'Province', `region` as
Region,
DATE_FORMAT(`date_of_birth`,
'%Y-%m-%d') as 'Date of Birth',
`gender` as 'Gender', mobile_no as
'Mobile No.', place_of_birth as 'Place
of Birth', highest_education as
'Highest Education', civil_status as
'Civil Status', spouse_name as
'Spouse Name', main_livelihood as
'Main Livelihood', religion as
'Religion',
emergency_contact_person as
'Emergency Contact Person',
emergency_contact_number as
'Emergency Contact Number', case
when pwd then 'YES' else 'No' end
as 'PWD', case when indigent then
'YES' else 'No' end as 'Indigent', case
when four_ps then 'YES' else 'No'
end as '4PS', case when rsbsa then
'YES' else 'No' end as 'RSBSA'
FROM `farmers`)
    listView.Columns.Clear()
    listView.Items.Clear()

    Dim count As Integer
    count = 0
    listView.Columns.Add("", 0)

    For Each columns As
DataColumn In table.Columns

        If
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then
            Else

listView.Columns.Add(columns.Colum
nName.ToString(), 1,
HorizontalAlignment.Center)
                End If
            Next

                Dim lstviewItemImageList As
ImageList = New ImageList

                For Each rows As DataRow In
table.Rows
                    Dim list As New
ListViewItem()

                    For Each columns As
DataColumn In table.Columns

                        If Not
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then

list.SubItems.Add(rows(table.Colum
ns.IndexOf(columns.ColumnName.To
String())))
                            End If

                        Next

                        Dim imgByte() As Byte
imgByte = rows(0)

                        Dim ms As New
MemoryStream(imgByte)

                        lstviewItemImageList.ImageSize =
New Size(30, 30)
                            listView.SmallImageList =
lstviewItemImageList
                                list.ImageIndex =
lstviewItemImageList.Images.Add(Im
age.FromStream(ms),
Color.Transparent)
                                    listView.Items.Add(list)

                                listView.AutoSizeColumns(Column
nHeaderAutoResizeStyle.HeaderSiz
e)
                                    Next
                                End Sub

```

```

Public Sub
getAllFarmersById(ByVal id As
String)
    Dim query As
MySQLDataReader =
MyBase.CreateQueryOject("SELEC
T `id`, `l_name`, `m_name`,
`f_name`, `e_name`, `purok_sitio`,
`barangay`, `municipality`,
`province`, `region`, `date_of_birth`,
`gender`, `mobile_no`,
`place_of_birth`,
`highest_education`, `civil_status`,
`spouse_name`, `main_livelihood`,
`religion`,
`emergency_contact_person`,
`emergency_contact_number`,
`pwd`, `indigent`, `four_ps`, `rsbsa`,
`image_file` FROM `farmers`
WHERE id = " + id)

    If query.HasRows Then
        While (query.Read())
            Dim imgByte() As Byte
            imgByte =
query("image_file")

            Dim ms As New
MemoryStream(imgByte)

            FarmerForm.l_name.Text
= query("l_name")
            FarmerForm.m_name.Text
= query("m_name")
            FarmerForm.f_name.Text
= query("f_name")
            FarmerForm.e_name.Text
= query("e_name")
            FarmerForm.purok.Text =
query("purok_sitio")

            FarmerForm.barangay.Text =
query("barangay")

            'FarmerForm.birthday.Value =
query("date_of_birth").ToString()

            FarmerForm.gender.Text =
query("gender")

            FarmerForm.mobile_no.Text =
query("mobile_no")

            FarmerForm.place_of_birth.Text =
query("place_of_birth")

            FarmerForm.highest_education.Text
= query("highest_education")

            FarmerForm.civil_status.Text =
query("civil_status")

            FarmerForm.spouse_name.Text =
query("spouse_name")

            FarmerForm.main_livelihood.Text =
query("main_livelihood")
            FarmerForm.religion.Text
= query("religion")

            FarmerForm.emergency_contact_pe
rson.Text =
query("emergency_contact_person")

            FarmerForm.emergency_contact_no
.Text =
query("emergency_contact_number"
)
            FarmerForm.pwd.Checked
= query("pwd")

            FarmerForm.indigent.Checked =
query("indigent")

            FarmerForm.fourPS.Checked =
query("four_ps")

            FarmerForm.rsbsa.Checked =
query("rsbsa")

            FarmerForm.PictureBox1.Image =
Image.FromStream(ms)

```

```

        End While
    End If
    DbConfig.CloseConnection()

```

```
End Sub
```

```
End Class
```

Fruit Trees Domain

```
Imports System.Reflection
Imports System.IO
Imports MySql.Data
Imports MySql.Data.MySqlClient
```

```
Public Class Fruit_Trees : Inherits
JPAClass
```

```

    Public Sub New()
        Me.id = vbNullString
        Me.farmers = ""
        Me.barangay = ""
        Me.fruit_trees_type = ""
        Me.owner_ship = ""
        Me.quantity = 0
        Me.location = ""
        Me.date_planted = New Date()
        Me.expected_date_of_harvest
= New Date()
    End Sub

```

```

    Public Overrides Function
SetDomain() As Object
        Dim fruitTrees As New
Fruit_Trees
        Return fruitTrees
    End Function

```

```

    Public Property id As Integer
    Public Property farmers As String
    Public Property barangay As
String
    Public Property fruit_trees_type
As String
    Public Property owner_ship As
String

```

```

    Public Property quantity As
Decimal
    Public Property location As String
    Public Property date_planted As
String
    Public Property
expected_date_of_harvest As String

```

```

    Public Function
getFormField(ByVal column As
PropertyInfo) As String
        Return "@" + column.Name
    End Function

```

```

    Public Sub getAll(ByVal listView
As ListView)
        Dim table As DataTable =
MyBase.CreateQuery("SELECT *
FROM `farmers_fruit_trees`")
        listView.Columns.Clear()
        listView.Items.Clear()

```

```

        Dim count As Integer
count = 0

```

```

        For Each columns As
DataColumn In table.Columns

```

```

listView.Columns.Add(columns.Colu
mnName.ToString(), 1,
HorizontalAlignment.Center)
        Next

```

```

        Dim lstviewItemImageList As
ImageList = New ImageList

```

```

        For Each rows As DataRow In
table.Rows

```

```

            Dim list As New
ListViewItem()

```

```

            For Each columns As
DataColumn In table.Columns

```

```

        If Not
        (table.Columns.IndexOf(columns.Column
        Name.ToString()) = 0) Then

        list.SubItems.Add(rows(table.Column
        s.IndexOf(columns.ColumnName.To
        String()))

        Else
        list.Text =
        rows(table.Columns.IndexOf(column
        s.ColumnName.ToString()))
        End If

        Next

        listView.Items.Add(list)

        listView.AutoSizeColumns(Column
        HeaderAutoSizeStyle.HeaderSiz
        e)

        Next
    End Sub

```

```

    Public Sub search(ByVal listView
    As ListView, ByVal search As String)
        Dim table As DataTable =
        MyBase.CreateQuery("SELECT *
        FROM `farmers_fruit_trees` where
        ('FARMERS` like '%" + search + "%'
        or `FRUIT TREES TYPE` like '%" +
        search + "%')")
        listView.Columns.Clear()
        listView.Items.Clear()

        Dim count As Integer
        count = 0

        For Each columns As
        DataColumn In table.Columns

        listView.Columns.Add(columns.Colu
        mnName.ToString(), 1)
        Next

        Dim lstviewItemImageList As
        ImageList = New ImageList

```

```

        For Each rows As DataRow In
        table.Rows
            Dim list As New
            ListViewItem()

            For Each columns As
            DataColumn In table.Columns

            If Not
            (table.Columns.IndexOf(columns.Col
            umnName.ToString()) = 0) Then

            list.SubItems.Add(rows(table.Column
            s.IndexOf(columns.ColumnName.To
            String()))
            Else
            list.Text =
            rows(table.Columns.IndexOf(column
            s.ColumnName.ToString()))
            End If
            Next

            listView.Items.Add(list)

            listView.AutoSizeColumns(Column
            HeaderAutoSizeStyle.HeaderSiz
            e)

            Next

        End Sub

        Public Sub getByld(ByVal id As
        String)
            Dim query As
            MySqlDataReader =
            MyBase.CreateQueryObject("SELEC
            T c.`id`, CONCAT('l_name`,',
            ',`f_name`,',',`m_name`,',',`e_name`)
            as `farmers`, c.`barangay`,
            `fruit_trees_type`, `owner_ship`,
            `location`, `date_planted`,
            `expected_date_of_harvest`,
            `quantity` FROM `farms`.`fruit_trees`
            `c` join farmers f on f.id = c.farmers
            WHERE c.id = " + id)

```

```

    If query.HasRows Then
        While (query.Read())

            FruitTreesForm.farmers.SelectedIndex =
            FruitTreesForm.farmers.FindStringExact(query("farmers"))

            FruitTreesForm.barangay.Text =
            query("barangay")

            FruitTreesForm.fruit_trees_type.Text =
            query("fruit_trees_type")

            FruitTreesForm.ownership.Text =
            query("owner_ship")

            FruitTreesForm.quantity.Text =
            query("quantity")

            FruitTreesForm.trees_location.Text =
            query("location")

            FruitTreesForm.date_planted.Text =
            query("date_planted")

            FruitTreesForm.expected_date_of_harvest.Text =
            query("expected_date_of_harvest")

                End While
            End If
            DbConfig.CloseConnection()

        End Sub

```

```
End Class
```

Livestock Domain

```

Imports System.Reflection
Imports System.IO
Imports MySql.Data
Imports MySql.Data.MySqlClient

```

```
Public Class Livestock : Inherits JPAClass
```

```

    Public Sub New()
        Me.id = vbNullString
        Me.farmers = ""
        Me.barangay = ""
        Me.livestock_type = 0
        Me.number_of_livestock = 0
        Me.date_created =
        Date.Now.ToString("yyyy/MM/dd")

    End Sub

    Public Overrides Function SetDomain() As Object
        Dim livestock As New Livestock
        Return livestock
    End Function

    Public Property id As Integer
    Public Property farmers As String
    Public Property barangay As String
    Public Property livestock_type As String
    Public Property number_of_livestock As Decimal
    Public Property date_created As String

    Public Function getFormField(ByVal column As PropertyInfo) As String
        Return "@" + column.Name
    End Function

```

```

    Public Sub getAll(ByVal listView As ListView)
        Dim table As DataTable =
        MyBase.CreateQuery("SELECT *
        FROM `farmers_livestocks`")
        listView.Columns.Clear()
        listView.Items.Clear()

        Dim count As Integer

```

```

        count = 0

        For Each columns As
            DataColumn In table.Columns

            listView.Columns.Add(columns.ColumnName.ToString(), 1,
                HorizontalAlignment.Center)
            Next

            Dim lstviewItemImageList As
                ImageList = New ImageList

            For Each rows As DataRow In
                table.Rows
                Dim list As New
                    ListViewItem()

                For Each columns As
                    DataColumn In table.Columns

                    If Not
                        (table.Columns.IndexOf(columns.ColumnName.ToString()) = 0) Then

                        list.SubItems.Add(rows(table.Columns.IndexOf(columns.ColumnName.ToString())))
                    Else
                        list.Text =
                            rows(table.Columns.IndexOf(columns.ColumnName.ToString()))
                    End If

                Next

                listView.Items.Add(list)

            listView.AutoSizeColumns(ColumnHeaderAutoResizeStyle.HeaderSize)
            Next

        End Sub

        Public Sub search(ByVal listView
            As ListView, ByVal search As String)
            Dim table As DataTable =
                MyBase.CreateQuery("SELECT *
                FROM `farmers_livestocks` where
                ('FARMERS` like '%" + search + "%'
                or LIVESTOCK like '%" + search +
                "%')")
            listView.Columns.Clear()
            listView.Items.Clear()

            Dim count As Integer
            count = 0

            For Each columns As
                DataColumn In table.Columns

                listView.Columns.Add(columns.ColumnName.ToString(), 1)
                Next

                Dim lstviewItemImageList As
                    ImageList = New ImageList

                For Each rows As DataRow In
                    table.Rows
                    Dim list As New
                        ListViewItem()

                    For Each columns As
                        DataColumn In table.Columns

                        If Not
                            (table.Columns.IndexOf(columns.ColumnName.ToString()) = 0) Then

                            list.SubItems.Add(rows(table.Columns.IndexOf(columns.ColumnName.ToString())))
                        Else
                            list.Text =
                                rows(table.Columns.IndexOf(columns.ColumnName.ToString()))
                        End If

                    Next

                    listView.Items.Add(list)

                End Sub

                listView.AutoSizeColumns(ColumnHeaderAutoResizeStyle.HeaderSize)
                Next
    
```

```

        listView.Items.Add(list)

listView.AutoSizeColumns(ColumnHeader.AutoSizeStyle.HeaderSize)
    Next
End Sub

Public Sub getById(ByVal id As String)
    Dim query As
    MySqlDataReader =
    MyBase.CreateQueryObject("SELECT I.`id`, CONCAT(I.`name`,',',
    `f_name`,``,`m_name`,``,`e_name`)
    as `farmers`, I.`barangay`,
    `livestock_type`,
    `number_of_livestock` FROM
    `livestock` I inner join farmers f on
    f.id = I.farmers WHERE I.id = " + id)

    If query.HasRows Then
        While (query.Read())

LivestockForm.farmers.SelectedIndex =
LivestockForm.farmers.FindStringExact(query("farmers"))

LivestockForm.barangay.Text =
query("barangay")

LivestockForm.livestock_type.Text =
query("livestock_type")

LivestockForm.number_of_livestock.
Text = query("number_of_livestock")

        End While
    End If
    DbConfig.CloseConnection()
End Sub

```

End Class

Poultry Domain

```

Imports System.Reflection
Imports System.IO
Imports MySql.Data
Imports MySql.Data.MySqlClient

Public Class Poultry : Inherits
JPAClass

    Public Sub New()
        Me.id = vbNullString
        Me.farmers = ""
        Me.barangay = ""
        Me.poultry_type = ""
        Me.number_of_poultry = 0
        Me.date_created =
Date.Now.ToString("yyyy/MM/dd")

    End Sub

    Public Overrides Function
SetDomain() As Object
        Dim poultry As New Poultry
        Return poultry
    End Function

    Public Property id As Integer
    Public Property farmers As String
    Public Property barangay As
String
    Public Property poultry_type As
String
    Public Property
number_of_poultry As Decimal
    Public Property date_created As
String

    Public Function
getFormField(ByVal column As
PropertyInfo) As String
        Return "@" + column.Name
    End Function

```

```

Public Sub getAll(ByVal listView
As ListView)
    Dim table As DataTable =
    MyBase.CreateQuery("SELECT *
FROM `farmers_poultry`")
    listView.Columns.Clear()
    listView.Items.Clear()

    Dim count As Integer
    count = 0

    For Each columns As
    DataColumn In table.Columns

listView.Columns.Add(columns.Colum
columnName.ToString(), 1,
HorizontalAlignment.Center)
        Next

    Dim lstviewItemImageList As
    ImageList = New ImageList

    For Each rows As DataRow In
    table.Rows
        Dim list As New
        ListViewItem()

        For Each columns As
        DataColumn In table.Columns

            If Not
            (table.Columns.IndexOf(columns.Col
            umnName.ToString()) = 0) Then

list.SubItems.Add(rows(table.Colum
ns.IndexOf(columns.ColumnName.To
String()))))
                Else
                    list.Text =
                    rows(table.Columns.IndexOf(column
                    s.ColumnName.ToString()))
                End If

            Next

listView.Items.Add(list)

```

```

listView.AutoSizeColumns(Colum
nHeaderAutoSizeStyle.HeaderSiz
e)
    Next

End Sub

Public Sub search(ByVal listView
As ListView, ByVal search As String)
    Dim table As DataTable =
    MyBase.CreateQuery("SELECT *
FROM `farmers_poultry` where
(`FARMERS` like '%" + search + '%"
or TYPE like '%" + search + "%'")")
    listView.Columns.Clear()
    listView.Items.Clear()

    Dim count As Integer
    count = 0

    For Each columns As
    DataColumn In table.Columns

listView.Columns.Add(columns.Colu
mnName.ToString(), 1)
        Next

    Dim lstviewItemImageList As
    ImageList = New ImageList

    For Each rows As DataRow In
    table.Rows
        Dim list As New
        ListViewItem()

        For Each columns As
        DataColumn In table.Columns

            If Not
            (table.Columns.IndexOf(columns.Col
            umnName.ToString()) = 0) Then

list.SubItems.Add(rows(table.Colum
ns.IndexOf(columns.ColumnName.To
String()))))
                Else
                    list.Text =
                    rows(table.Columns.IndexOf(column
                    s.ColumnName.ToString()))
                End If

            Next

listView.Items.Add(list)

```

```

        Else
            list.Text =
rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
            End If

        Next

        listView.Items.Add(list)

listView.AutoSizeColumns(Column
nHeaderAutoSizeStyle.HeaderSiz
e)
        Next

    End Sub

    Public Sub getByld(ByVal id As
String)
        Dim query As
MySqlDataReader =
MyBase.CreateQueryObject("SELEC
T `l`.`id`,concat(`f`.`l_name`,`
`,`f`.`f_name`,``,`f`.`m_name`,`
`,`f`.`e_name`) AS
`farmers`,`l`.`barangay`,
`poultry_type`,`number_of_poultry`
FROM `poultry` l join
`farms`.`farmers` `f` on((`f`.`id` =
`l`.`farmers`) WHERE l.id = " + id)

        If query.HasRows Then
            While (query.Read())

PoultryForm.farmers.SelectedIndex
=
CropsForm.farmers.FindStringExact(
query("farmers"))

PoultryForm.barangay.Text =
query("barangay")

PoultryForm.poultry_type.Text =
query("poultry_type")

```

```

PoultryForm.number_of_poultry.Text
= query("number_of_poultry")

        End While
    End If
    DbConfig.CloseConnection()

    End Sub

```

```
End Class
```

Services Domain

```

Imports System.Reflection
Imports System.IO
Imports MySql.Data
Imports MySql.Data.MySqlClient

Public Class Services : Inherits
JPAClass

    Public Sub New()
        Me.id = vbNullString
        Me.farmers = ""
        Me.barangay = ""
        Me.services = ""
        Me.date_received = New Date()
    End Sub

    Public Overrides Function
SetDomain() As Object
        Dim domain As New Services
        Return domain
    End Function

    Public Property id As Integer
    Public Property farmers As String
    Public Property barangay As
String
    Public Property services As String
    Public Property date_received As
String

```

```

Public Function
getFormField(ByVal column As
PropertyInfo) As String
    Return "@" + column.Name
End Function

Public Sub getAll(ByVal listView
As ListView)
    Dim table As DataTable =
MyBase.CreateQuery("SELECT *
FROM `farmers_services`")
    listView.Columns.Clear()
    listView.Items.Clear()

    Dim count As Integer
    count = 0

    For Each columns As
DataColumn In table.Columns

listView.Columns.Add(columns.Colum
nName.ToString(), 1,
HorizontalAlignment.Center)
        Next

        Dim lstviewItemImageList As
ImageList = New ImageList

        For Each rows As DataRow In
table.Rows
            Dim list As New
ListViewItem()

            For Each columns As
DataColumn In table.Columns

                If Not
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then

list.SubItems.Add(rows(table.Colum
nName.ToString()))
                    Else

                        list.Text =
rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
                        End If

                Next

            listView.Items.Add(list)

listView.AutoSizeColumns(Column
nHeaderAutoResizeStyle.HeaderSiz
e)

            Next

        End Sub

        Public Sub search(ByVal listView
As ListView, ByVal search As String)
            Dim table As DataTable =
MyBase.CreateQuery("SELECT *
FROM `farmers_services` where
('FARMERS` like '%" + search + '%"
or `SERVICES` like '%" + search +
"%')")
            listView.Columns.Clear()
            listView.Items.Clear()

            Dim count As Integer
            count = 0

            For Each columns As
DataColumn In table.Columns

listView.Columns.Add(columns.Colum
nName.ToString(), 1)
                Next

                Dim lstviewItemImageList As
ImageList = New ImageList

                For Each rows As DataRow In
table.Rows
                    Dim list As New
ListViewItem()

```

```

        For Each columns As
DataColumn In table.Columns

            If Not
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then

list.SubItems.Add(rows(table.Column
s.IndexOf(columns.ColumnName.To
String()))
            Else
                list.Text =
rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
            End If
        Next
        listView.Items.Add(list)

listView.AutoResizeColumns(Column
HeaderAutoResizeStyle.HeaderSiz
e)
    Next
End Sub

Public Sub getById(ByVal id As
String)
    Dim query As
MySqlDataReader =
MyBase.CreateQueryObject("SELEC
T I.`id`, CONCAT(`I_name`,`
`,`f_name`,``,`m_name`,``,`e_name`)
as `farmers`, I.`barangay`, `services`,
`date_received` FROM `services` I
inner join farmers f on f.id = I.farmers
WHERE I.id = " + id)

    If query.HasRows Then
        While (query.Read())

ServicesForm.farmers.SelectedIndex
=
ServicesForm.farmers.FindStringExa
ct(query("farmers"))

ServicesForm.barangay.Text =
query("barangay")

```

```

ServicesForm.services.Text =
query("services")

ServicesForm.date_received.Text =
query("date_received")

        End While
    End If
    DbConfig.CloseConnection()
End Sub

End Class

Vegetables Domain
Imports System.Reflection
Imports System.IO
Imports MySql.Data
Imports MySql.Data.MySqlClient

Public Class Vegetables : Inherits
JPAClass

    Public Sub New()
        Me.id = vbNullString
        Me.farmers = ""
        Me.barangay = ""
        Me.total_area_planted = 0
        Me.unit = ""
        Me.variety_plant = ""
        Me.owner_ship = ""
        Me.location = ""
        Me.date_planted = New Date()
        Me.expected_date_of_harvest
= New Date()
    End Sub

    Public Overrides Function
SetDomain() As Object
        Dim fruitTrees As New
Fruit_Trees
        Return fruitTrees
    End Function

    Public Property id As Integer
    Public Property farmers As String

```

```

    Public Property barangay As
String
    Public Property owner_ ship As
String
    Public Property total_ area_ planted
As Decimal
    Public Property unit As String
    Public Property variety_ plant As
String
    Public Property location As String
    Public Property date_ planted As
String
    Public Property
expected_ date_ of_ harvest As String

    Public Function
getFormField(ByVal column As
PropertyInfo) As String
    Return "@" + column.Name
    End Function

    Public Sub getAll(ByVal listView
As ListView)
    Dim table As DataTable =
MyBase.CreateQuery("SELECT *
FROM `farmers_vegetables`")
    listView.Columns.Clear()
    listView.Items.Clear()

    Dim count As Integer
count = 0

    For Each columns As
DataColumn In table.Columns

listView.Columns.Add(columns.Colum
nName.ToString(), 1,
HorizontalAlignment.Center)
        Next

    Dim lstviewItemImageList As
ImageList = New ImageList
    For Each rows As DataRow In
table.Rows
        Dim list As New
ListViewItem()

```

```

        For Each columns As
DataColumn In table.Columns

            If Not
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then

list.SubItems.Add(rows(table.Colum
nName.ToString()))
            Else
                list.Text =
rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
            End If

        Next

        listView.Items.Add(list)

listView.AutoResizeColumns(Column
HeaderAutoResizeStyle.HeaderSiz
e)

    Next

    End Sub

    Public Sub search(ByVal listView
As ListView, ByVal search As String)
    Dim table As DataTable =
MyBase.CreateQuery("SELECT *
FROM `farmers_vegetables` where
(`FARMERS` like '%" + search + '%"
or `VARIETY` like '%" + search +
"%')")
        listView.Columns.Clear()
        listView.Items.Clear()

        Dim count As Integer
count = 0

        For Each columns As
DataColumn In table.Columns

```

```

listView.Columns.Add(columns.ColumnName.ToString(), 1)
    Next

    Dim lstviewItemImageList As
    ImageList = New ImageList

    For Each rows As DataRow In
    table.Rows
        Dim list As New
        ListViewItem()

        For Each columns As
        DataColumn In table.Columns

            If Not
            (table.Columns.IndexOf(columns.Col
            umnName.ToString()) = 0) Then

                listView.SubItems.Add(rows(table.Column
                s.IndexOf(columns.ColumnName.To
                String()))

                Else
                    list.Text =
                    rows(table.Columns.IndexOf(column
                    s.ColumnName.ToString()))
                End If

            Next

            listView.Items.Add(list)

        listView.AutoSizeColumns(Column
        HeaderAutoSizeStyle.HeaderSiz
        e)
        Next

    End Sub

    Public Sub getByld(ByVal id As
    String)
        Dim query As
        MySqlDataReader =
        MyBase.CreateQueryObject("SELEC
        T c.`id`, CONCAT('f_name`,',
        ',`f_name`,``,`m_name`,``,`e_name`)
        as `farmers`,
        c.`barangay`,`owner_ship`,`total_are
        a_planted`,`unit`,`variety_plant`,
        `location`,`date_planted`,
        `expected_date_of_harvest` FROM
        `farms`.`vegetables` `c` join farmers f
        on f.id = c.farmers WHERE c.id = " +
        id)

        If query.HasRows Then
            While (query.Read())

                HVCDPFForm.farmers.SelectedIndex
                =
                HVCDPFForm.farmers.FindStringExa
                ct(query("farmers"))

                HVCDPFForm.barangay.Text =
                query("barangay")
                HVCDPFForm.variety.Text
                = query("variety_plant")

                HVCDPFForm.ownership.Text =
                query("owner_ship")
                HVCDPFForm.unit.Text =
                query("unit")

                HVCDPFForm.vegetable_location.Text
                = query("location")

                HVCDPFForm.total_area_planted.Text
                = query("total_area_planted")

                HVCDPFForm.date_planted.Text =
                query("date_planted")

                HVCDPFForm.expected_date_of_har
                vest.Text =
                query("expected_date_of_harvest")

            End While
        End If
        DbConfig.CloseConnection()
    End Sub
End Class

```

Visits Domain

```
Imports System.Reflection
Imports System.IO
Imports MySql.Data
Imports MySql.Data.MySqlClient
```

```
Public Class Visits : Inherits
JPAClass
```

```
Public Sub New()
    Me.id = vbNullString
    Me.farmers = ""
    Me.barangay = ""
    Me.concern = ""
    Me.status = ""
    Me.date_of_visits = New Date()
End Sub
```

```
Public Overrides Function
SetDomain() As Object
    Dim domain As New Visits
    Return domain
End Function
```

```
Public Property id As Integer
Public Property farmers As String
Public Property barangay As
String
Public Property concern As String
Public Property date_of_visits As
String
Public Property status As String
```

```
Public Function
getFormField(ByVal column As
PropertyInfo) As String
    Return "@" + column.Name
End Function
```

```
Public Sub getAll(ByVal listView
As ListView)
    Dim table As DataTable =
MyBase.CreateQuery("SELECT *
FROM `farmers_visits`")
```

```
listView.Columns.Clear()
listView.Items.Clear()
```

```
Dim count As Integer
count = 0
```

```
For Each columns As
DataColumn In table.Columns
```

```
listView.Columns.Add(columns.Colu
mnName.ToString(), 1,
HorizontalAlignment.Center)
```

```
Next
Dim lstviewItemImageList As
ImageList = New ImageList
```

```
For Each rows As DataRow In
table.Rows
```

```
    Dim list As New
ListViewItem()
```

```
        For Each columns As
DataColumn In table.Columns
```

```
            If Not
(table.Columns.IndexOf(columns.Col
umnName.ToString()) = 0) Then
```

```
list.SubItems.Add(rows(table.Column
s.IndexOf(columns.ColumnName.To
String())))
```

```
            Else
```

```
                list.Text =
```

```
rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
```

```
            End If
```

```
        Next
```

```
listView.Items.Add(list)
```

```
listView.AutoResizeColumns(Colum
nHeaderAutoResizeStyle.HeaderSiz
e)
```

```
Next
```

```
End Sub
```

```

Public Sub search(ByVal listView
As ListView, ByVal search As String)
    Dim table As DataTable =
    MyBase.CreateQuery("SELECT *
FROM `farmers_visits` where
(` FARMERS` like '%" + search + "%'
or `CONCERN` like '%" + search +
"%')")
    listView.Columns.Clear()
    listView.Items.Clear()

    Dim count As Integer
    count = 0

    For Each columns As
    DataColumn In table.Columns

listView.Columns.Add(columns.Colum
columnName.ToString(), 1)
        Next

    Dim lstviewItemImageList As
    ImageList = New ImageList

    For Each rows As DataRow In
    table.Rows
        Dim list As New
        ListViewItem()

        For Each columns As
        DataColumn In table.Columns

            If Not
            (table.Columns.IndexOf(columns.Col
            umnName.ToString()) = 0) Then

list.SubItems.Add(rows(table.Colum
s.IndexOf(columns.ColumnName.To
String()))))
            Else
                list.Text =
                rows(table.Columns.IndexOf(column
s.ColumnName.ToString()))
            End If
        End If
    End For
    End For

```

```

Next
        listView.Items.Add(list)
        listView.AutoSizeColumns(Column
nHeaderAutoSizeStyle.HeaderSiz
e)
    Next
    End Sub

    Public Sub getById(ByVal id As
    String)
        Dim query As
        MySqlDataReader =
        MyBase.CreateQueryObject("SELEC
T l.`id`, CONCAT(`l_name`,`
`,`f_name`,``,`m_name`,``,`e_name`)
as `farmers`, l.`barangay`,`concern`,
`date_of_visits`,`status` FROM
`visits` l inner join farmers f on f.id =
l.farmers WHERE l.id = " + id)

        If query.HasRows Then
            While (query.Read())

FarmersVisits.farmers.SelectedIndex
FarmersVisits.farmers.FindStringExa
ct(query("farmers"))

FarmersVisits.barangay.Text =
query("barangay")

FarmersVisits.concern.Text =
query("concern")

FarmersVisits.date_of_visits.Text =
query("date_of_visits")
                FarmersVisits.status.Text
                = query("status")

            End While
        End If
        DbConfig.CloseConnection()

    End Sub

End Class

```

APPENDIX D

USER'S MANUAL

A. Accessing the System

Steps:

1. To Login type "User Name"
2. Type User's Password
3. Click "LOGIN or Press Enter key" Button to continue

B. Adding login user

Steps:

1. Select "Add New User"
2. Fill up "Username"
3. Type "password"
4. Click "Add User"
5. Confirmation "Do you want to add user?"
6. Click "OK" to confirm
7. Click "Exit" to close

C. Membership Form

Steps:

1. In the Main Menu, Select "Files"
2. Select "Membership Form" to add member
3. Input require information
4. Click "Save"
5. Confirmation "Do you want to save information?"
6. Click "OK" to confirm
7. Click "Refresh" to clear inputs
8. Click "Exit" to close

D. Crop Growers Form

Steps:

1. In the Main Menu, Select "Files"
2. Select "Crop Growers Form" to add crop information
3. Input require information
4. Click "Save"
5. Confirmation "Do you want to save information?"
6. Click "OK" to confirm

7. Click "Refresh" to clear inputs

8. Click "Exit" to close

E. Livestock Raisers Form

Steps:

1. In the Main Menu, Select "Files"

2. Select "Livestock Raisers Form" to add livestock information

3. Input require information

4. Click "Save"

5. Confirmation "Do you want to save information?"

6. Click "OK" to confirm

7. Click "Refresh" to clear inputs

8. Click "Exit" to close

F. Poultry Raisers Form

Steps:

1. In the Main Menu, Select "Files"

2. Select "Poultry Raisers Form" to add poultry information

3. Input require information

4. Click "Save"
5. Confirmation "Do you want to save information?"
6. Click "OK" to confirm
7. Click "Refresh" to clear inputs
8. Click "Exit" to close

G. Vegetable Growers Form

Steps:

1. In the Main Menu, Select "Files"
2. Select "Vegetable Growers Form" to add vegetable information
3. Input require information
4. Click "Save"
5. Confirmation "Do you want to save information?"
6. Click "OK" to confirm
7. Click "Refresh" to clear inputs
8. Click "Exit" to close

H. Vegetable Growers Form

Steps:

1. In the Main Menu, Select "Files"
2. Select "Fruit Trees Growers Form" to add fruit trees information
3. Input require information
4. Click "Save"
5. Confirmation "Do you want to save information?"
6. Click "OK" to confirm
7. Click "Refresh" to clear inputs
8. Click "Exit" to close

I. Update Farmers Record

Steps:

1. Search to the "Search Box"
2. Search the Name you wanted to edit
3. "Double tap" and edit it
4. Click "Update" to change information
5. Confirmation "Do you want to update?" to confirm
6. Click "OK"

J. Delete Farmers Record

Steps:

1. Search to the "Search Box"
2. Search the Name you wanted to delete
3. "Double tap" and delete
4. Click "Delete" to remove information
5. Confirmation "Do you want to delete?" to confirm
6. Click "OK"

K. Services Received Form

Steps:

1. In the Main Menu, Select "Files"
2. Select "Services Received Form" to record services received by farmers
3. Input require information
4. Click "Save"
5. Confirmation "Do you want to save information?"
6. Click "OK" to confirm
7. Click "Refresh" to clear inputs
8. Click "Exit" to close

L. Farmers Visit Form

Steps:

1. In the Main Menu, Select "Monitor"
2. Select "Farmers Visit Form" to record farmers concern
3. Input require information
4. Click "Save"
5. Confirmation "Do you want to save information?"
6. Click "OK" to confirm
7. Click "Refresh" to clear inputs
8. Monitor concern if it's been cater or not
9. Click "Exit" to close

M. Dashboard

Steps:

1. On the Main Menu select "Dashboard"
2. View the map of San Miguel
3. View the total number of farmers, crop growers, livestock raisers, poultry raisers, vegetable growers, and fruit trees growers.
3. Click "Close" when done

N. Setting

Steps:

1. On the Main Menu Select “Settings”
 - a. Select “Accounts”
 - i. Account Form will show
 - b. Select “Developer”
 - ii. Developers Profile will show

O. Reports

Steps:

1. On the Main Menu select “Reports”
2. On the left side, there is the print icon
3. After viewing the records we can now print right away
4. After report printed click “Close” when done

P. Log Out

Steps:

1. On the Main Menu Select “Logout”
2. Click “LogOut
- 3.”It will logout and automatically back to “login user”

RESEARCHER'S BIODATA

Name : Chynna Marie Hernane Cano
Birth Date : April 17, 2000
Place of Birth : Mahayag, San Miguel, Bohol
Age : 22
Home Address : Mahayag, San Miguel, Bohol
Email Address : chymariee17@gmail.com
Religion : Roman Catholic
Citizenship : Filipino



Father's Name : Borromeo Ompoy Cano
Mother's Name : Herminia Hernane Cano

EDUCATIONAL BACKGROUND

Elementary : Mahayag Elementary School
 Mahayag, San Miguel, Bohol
 2011-2012

Secondary : Mahayag National High School
 Mahayag, San Miguel, Bohol
 2017-2018

Tertiary : Bachelor of Science in Computer Science
 Bohol Island State University- Bilar Campus
 Zamora, Bilar, Bohol
 2021-2022

Work Experience : On the - Job Training
 Bohol Island State University-Bilar Campus
 Zamora, Bilar, Bohol

RESEARCHER'S BIODATA

Name : Lito Boy Omul
Birth Date : December 12, 1993
Place of Birth : Montehermo, Carmen Bohol
Age : 28
Home Address : San Jose, Talibon, Bohol
Email Address : reallitoboyomul@gmail.com
Religion : Apostolic Pentecostal (WPF)
Citizenship : Filipino



Father's Name : Sofronio Omul
Mother's Name : Carmelita Omul

EDUCATIONAL BACKGROUND

Elementary : San Isidro Elementary School
 2006-2007

Secondary : Francisco L. Adlaon High School

Tertiary : Bachelor of Science in Computer Science
 Bohol Island State University- Bilar Campus
 Zamora, Bilar, Bohol
 2021-2022

Work Experience : On the - Job Training
 Bohol Island State University-Bilar Campus
 Zamora, Bilar, Bohol

RESEARCHER'S BIODATA

Name : Joshua Sastrellas Quirol
Birth Date : March 27, 2000
Place of Birth : Mahayag, San Miguel, Bohol
Age : 22
Home Address : Mahayag, San Miguel, Bohol
Email Address : josh.ualoruiq2700@gmail.com
Religion : Roman Catholic
Citizenship : Filipino
Father's Name : Adriano Ampo Quirol Sr.
Mother's Name : Ernesta Sastrillas Quirol



EDUCATIONAL BACKGROUND

Elementary : Mahayag Elementary School
 Mahayag, San Miguel, Bohol
 2011-2012

Secondary : Mahayag National High School
 Mahayag, San Miguel, Bohol
 2017-2018

Tertiary : Bachelor of Science in Computer Science
 Bohol Island State University- Bilar Campus
 Zamora, Bilar, Bohol
 2021-2022

Work Experience : On the - Job Training
 Bohol Island State University-Bilar Campus
 Zamora, Bilar, Bohol

RESEARCHER'S BIODATA

Name : Ana Mae Grace Amila Vicere
Birth Date : March 12, 1999
Place of Birth : Bagumbayan, Pilar, Bohol
Age : 23
Home Address : Poblacion, Pilar, Bohol
Email Address : annamaegrace.vecere@gmail.com
Religion : Apostolic Pentecostal (AMPI)
Citizenship : Filipino
Father's Name : Cerino Feliselda Vicere
Mother's Name : Flavia Amila Vicere



EDUCATIONAL BACKGROUND

Elementary : Pilar Central Elementary School
 Poblacion, Pilar, Bohol
 2011-2012

Secondary : Pilar Technical Vocational High School
 Poblacion, Pilar, Bohol
 2017-2018

Tertiary : Bachelor of Science in Computer Science
 Bohol Island State University- Bilar Campus
 Zamora, Bilar, Bohol
 2021-2022

Work Experience : On the - Job Training
 Bohol Island State University-Bilar Campus
 Zamora, Bilar, Bohol