

1. INTRODUCTION

India is one of the agricultural country and its climatic conditions randomly change in many times. Farmers are facing drought condition. Soil plays a major role in agriculture. There are different types of soil for the crop yield. The technique of offering vital vitamins to vegetation by delivering fertilizer aggregate at the side of the irrigation water is called fertigation. Fertilizers is a chemical substance or an organism that has a role in supplying the nutrient substance plants directly or indirectly added to the soil in order for the plant to the grow.

Based on the type of soil, Fertilizer can be injected to soil for the better crop yield production. Fertigation assure equal distribution of fertilizers to all of the vegetation and better absorption of nutrients. Based on recent technology, the climatic condition like temperature, humidity, moisture and Ph in the soil can be analysed and fertilizer like nitrogen, phosphorous and potassium is injected to the crop.

The soil fertility and soil type and fertilizers plays a vital role in agriculture and to check the availability of ground water and rainfall in crop yield production can be improved . There are two types of agricultural land that is dry land and wetland .In dry land ,the cash crops yield more profit for farmers and in wetland wheat and sugarcane yield more profit for farmers. There are different types of agro climatic regions in India and among the types of regions can be divided. The ultimate aim of this project is to display the chemical fertilizers based on crop.

Agriculture is a very essential for everyone's life. now a days agriculture reduces because of farmers revenue. so we have initialize automation in agriculture. Agriculture consist of planting, watering, fertilizing . In agriculture choosing correct fertilizer plays major role because of fertilizers either we could get more yield or vice versa . this project gives farmer correct idea of cost of particular fertilizer to farmers and correct fertilizer to the particular crop. This project helps farmer from getting cheated by retail sellers . Therefore, a forum to especially accommodate fertilizer selection using this application system.

1.1 MOTIVATION

Motive of this application is to save farmer from getting cheated by retail fertilizer sellers Promote farming and educating cultivation group with the cost of the fertilizers and the which kind of fertilizer is used for particular crop .It also create awareness of new fertilizers which are recently released . in this site cultivation group can also know different fertilizer companies and could try different fertilizer so that the cultivation group can get more yield than past . Hence the cultivation group can gain more knowledge on fertilizers and use fertilizers according to their choice and amount or price preference and protect from getting cheated by paying high price for fertilizer or taking wrong fertilizer suggested by seller

1.2 PROBLEM DEFINITION

Major problem for cultivation group is finance and selecting correct fertilizers. cultivation group get cheated by sellers by buying fertilizers at high cost and end up in debts and in future he either sells his land to over come loss or stop cultivation because of loss . Although after implementation of subsidy by government on particular fertilizers cultivation group end's up buying them at same cost or more most because loss of information. Thus an effective and innovative methods need to be develop which will evolve accordingly to the need.

1.3 OBJECTIVE OF PROJECT

To supply each palm with adequate nutrients in balanced proportion to ensure healthy vegetative growth and optimum economic FFB yields. to apply the fertilisers in the prescribed manner over the areas of the estate that are likely to result in the most efficient uptake of nutrients. To integrate the use of mineral fertilisers and palm residues. Primarily, the nutrient balance method is employed first to compute the nutrient requirements of oil palm in a manuring block. This approach assumes that the oil palm agroecosystem has definite components of nutrient removal (demand) from the system and nutrient return to the system The components of nutrient demand are Growth ,Yield ,Nutrient losses through leaching, run-off and erosion Nutrient removed by pest damage and Nutrient non-availability and antagonisms.

2. LITERATURE SURVEY

A literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project.

It is the most important part of your report as it gives you a direction in the area of your research. It helps you set a goal for your analysis - thus giving you your problem statement.

Literature survey is something when you look at a literature (publications) in a surface level, or an Ariel view. It incorporates the study of place people and productions are setting of research. It is phase where the analyst tries to know about what is all the literature related with one range of interest. Also, the relevant literature works are short-listed. Moreover, literature survey guides or helps the researcher to define/find out/identify a problem.

2.1 INTRODUCTION

The main purpose of the literature review work was to survey previous studies on knowledge sharing and intranets. In this, we look into the details about the existing system and try to

reduce the disadvantages of the existing system. We try to improve the performance and the efficiency of the new proposed system and also learn the advantages of proposed system.

2.2 EXISTING SYSTEM

The existing framework it is totally a manual process and during this manual process there is more tendency of doing mistakes in choosing fertilizers . in existing system farmers cannot get required information of fertilizers used for their crop because of retail sellers

2.2.1 DIS-ADVANTAGES

There are few dis-advantages identified in the existing system and are defined below:

- High chances to get cheated
- No exact information
- Time consummation
- No information regarding new fertilizers

2.3PROPOSED SYSTEM

In this application cultivators can have information regarding the fertilizer such as cost of each fertilizer and what kind of fertilizer to use for a particular crop we can over come major loop likes high chances to get cheated , no exact information , time consummation , no information regarding new fertilizers . By this system there will be more advantage to cultivation group

2.3.1ADVANTAGES

There are advantages in the proposed system which could overcome the drawbacks of the existing system and are defined below:

- Less human error
- Strength and strain can be reduced
- High security
- Easy to handle
- Easy to know cost of fertilizers
- Easy record keeping of new fertilizers

3. SYSTEM ANALYSIS

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

3.1 INTRODUCTION

In this phase the requirements are gathered and analysed. User's requirements are gathered in this phase. This phase is the main focus of the users and their interaction with the system.

There are few questions raised:

- Who is going to use the system?
- How will they use the system?
- What data should be input into the system?
- What data should be output by the system?

These general questions are answered during a requirement gathering phase. After requirement gathering these requirements are analysed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.

Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

3.2 SOFTWARE REQUIREMENTS

It deals with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed. The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view. We should try to understand what sort of requirements may arise in the requirement elicitation phase and what kinds of requirements are expected from the software system.

The software requirements that are required for this project are as follows:

- Operating System : Windows 7/8/10
- Php
- Html
- Java script

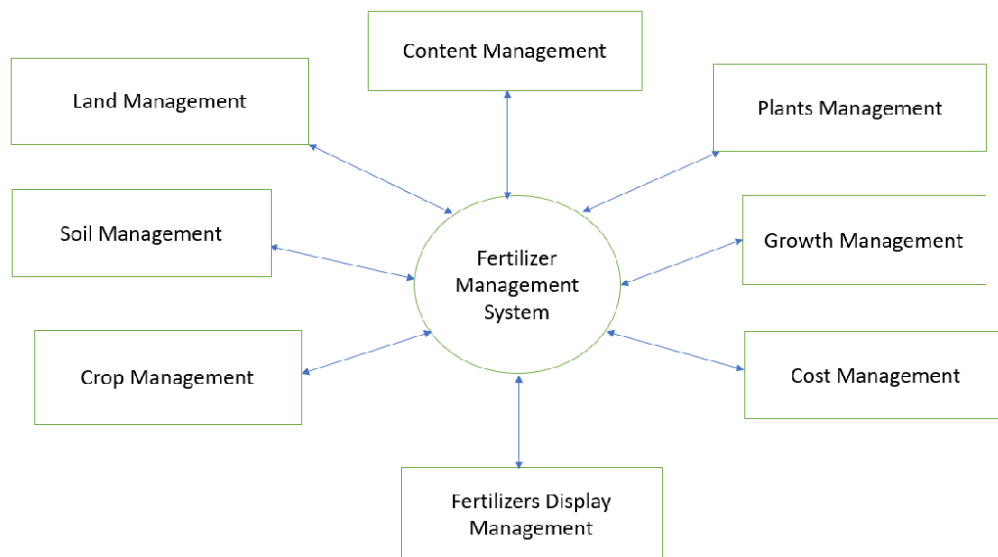
3.3 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, a hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

The hardware requirements that are required for this project are as follows:

- Sd card
- Sim card
- Ram-4gb

3.4CONTENT DIAGRAM OF PROJECT



4.SYSTEM DESIGN

The process of design involves “conceiving and planning out in mind and making a drawing, pattern or a sketch”. The system design transforms a logical representation of what a given system is required to do into the physical reality during development. Important design factors such as reliability, response time, throughput of the system, maintainability, expandability etc., should be taken into account. Design constraints like cost, hardware limitations, standard compliance etc should also be dealt with. The task of system design is to take the description and associate with it a specific set of facilities-men, machines (computing and other), accommodation, etc., to provide complete specifications of a workable system. This new system must provide for all of the essential data processing and it may also do some of those tasks identified during the work of analysis as optional extras. It must work within the imposed constraints and show improvement over the existing system.. At the outset of design a choice must be made between the main approaches. Talks of ‘preliminary design’ concerned with identification analysis and selections of the major design options are available for development and implementation of a system. These options are most readily distinguished in terms of the physical facilities to be used for the processing who or what does the work.

4.1 INTRODUCTION

Software design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints. Software design may refer to either "all the activity involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems" or "the activity following requirements specification and before programming, as in a stylized software engineering process." Software design usually involves problem solving and planning a software solution. This includes both a low-level component design and a high-level, architecture design.

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analysed and specified the software design involves four technical activities – design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

4.2DFD/ER/UML DIAGRAMS

UML stands for Unified Modelling Language which is used in object oriented software engineering. It is a standard language for specifying, visualizing, constructing, and documenting the artefacts of the software systems. UML is different from other common

programming languages like C++, Java, and COBOL etc. It is pictorial language used to make software blueprints.

Although typically used in software engineering it is a rich language that can be used to model an application structures, behaviour and even business processes. There are 8 UML diagram types to help us model this behaviour.

There are two types of UML modelling:

- Structural Modelling
- Behavioural Modelling

Structural Modelling:

Structural model represents the framework for the system and this framework is the place where all other components exist. Hence, the class diagram, component diagram and deployment diagrams are part of structural modelling. They all represent the elements and the mechanism to assemble them.

The structural model never describes the dynamic behaviour of the system. Class diagram is the most widely used structural diagram.

Structural Modelling captures the static features of a system. They consist of the following:

- i. Classes diagrams
- ii. Objects diagrams
- iii. Deployment diagrams
- iv. Package diagrams
- v. Composite structure diagram
- vi. Component diagram

Behavioural Modelling:

Behavioural model describes the interaction in the system. It represents the interaction among the structural diagrams. Behavioural modelling shows the dynamic nature of the system.

They consist of the following:

- Activity diagrams
- Interaction diagrams
- Use case diagrams

All the above show the dynamic sequence of flow in a system.

1. USE CASE DIAGRAM:

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. The "actors" are people or entities operating under defined roles within the system.

As the most known diagram type of the behavioural UML diagrams, use-case diagrams gives a graphic overview of the characters involved in a system, different functions needed by those characters and how these different functions are interacted.

Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors that may influence the system and should be taken into consideration. They provide a good high level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.

A key concept of use case modelling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behaviour in the user's terms by specifying all externally visible system behaviour.

Purpose of Use Case Diagram:

Use case diagrams are typically developed in the early stage of development and are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture
- Drive implementation and generate test cases
- Developed by analysts together with domain experts

Guidelines to draw a Use Case Diagram:

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analysed, the functionalities are captured in use cases. We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

- Functionalities to be represented as use case
- Actors

- Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Usage of Use Case Diagrams:

Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.

These diagrams are used at a very high level of design. This high level design is refined again and again to get a complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, and exceptions. These extra elements are used to make test cases when performing the testing.

Although use case is not a good candidate for forward and reverse engineering, still they are used in a slightly different way to make forward and reverse engineering. The same is true for reverse engineering. Use case diagram is used differently to make it suitable for reverse engineering.

In forward engineering, use case diagrams are used to make test cases and in reverse engineering use cases are used to prepare the requirement details from the existing application.

Use case diagrams can be used for –

- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.

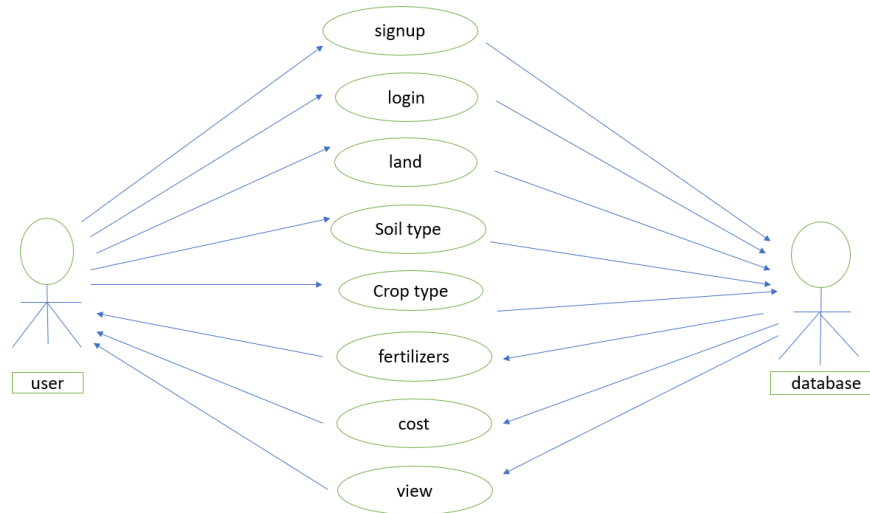


Fig: use case diagram

4.2.2 CLASS DIAGRAM

Class diagrams are the main building blocks of every object oriented methods. It is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Purpose of Class Diagram:

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application; however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as –

- This is the only UML which can appropriately depict various aspects of OOPs concept.
- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.

- Forward and reverse engineering.

Guidelines to draw a Class diagram:

Class diagrams are the most popular UML diagrams used for construction of software applications. It is very important to learn the drawing procedure of class diagram.

Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top level view.

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represent the whole system.

The following points should be remembered while drawing a class diagram –

- The name of the class diagram should be meaningful to describe the aspect of the system.
- Each element and their relationships should be identified in advance.
- Responsibility (attributes and methods) of each class should be clearly identified
- For each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.
- Use notes whenever required to describe some aspect of the diagram. At the end of the drawing it should be understandable to the developer/coder.
- Finally, before making the final version, the diagram should be drawn on plain paper and reworked as many times as possible to make it correct.

Usage of Class Diagram:

Class diagram is a static diagram and it is used to model the static view of a system. The static view describes the vocabulary of the system.

Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.

Generally, UML diagrams are not directly mapped with any object-oriented programming languages but the class diagram is an exception. Class diagram clearly shows the mapping with object-oriented languages such as Java, C++, etc. From practical experience, class diagram is generally used for construction purpose.

Class diagrams are used for –

- Describing the static view of the system.
- Showing the collaboration among the elements of the static view.
- Describing the functionalities performed by the system.
- Construction of software applications using object oriented languages.

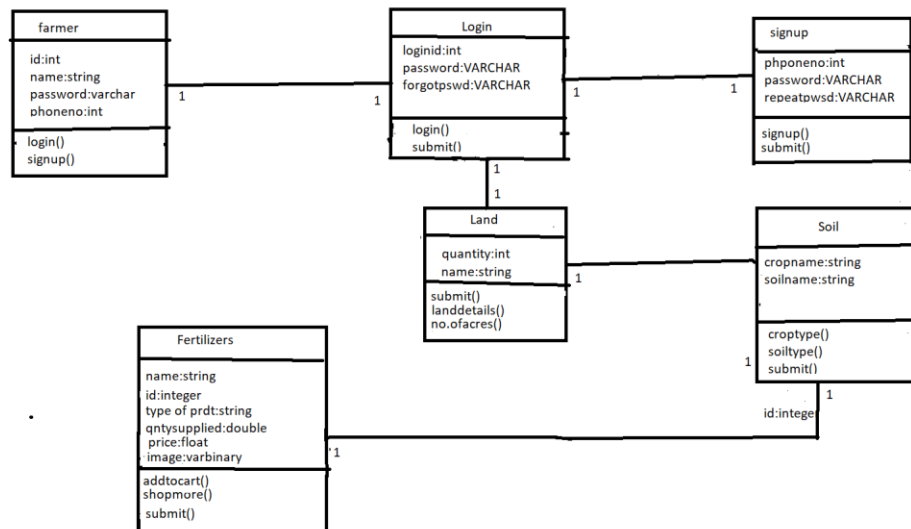


Fig: class diagram

4.2.3 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence diagrams emphasize on time sequence of messages and are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

Purpose of Sequence Diagram:

The purpose of sequence diagrams is to visualize the interactive behaviour of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction.

Sequence diagrams are used to capture the dynamic nature but from a different angle.

The purpose of sequence diagram is –

- To capture the dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects.

Guidelines to draw Sequence diagram:

The purpose of sequence diagrams is to capture the dynamic aspect of a system. So to capture the dynamic aspect, we need to understand what a dynamic aspect is and how it is visualized. Dynamic aspect can be defined as the snapshot of the running system at a particular moment.

The sequence diagram captures the time sequence of the message flow from one object to another.

Following things are to be identified clearly before drawing the sequence diagram --

- Objects taking part in the interaction.
- Message flows among the objects.
- The sequence in which the messages are flowing.

Usage of Sequence Diagram:

To understand the practical application, we need to understand the basic nature of sequence diagram.

The main purpose of sequence diagrams is they are used to capture the dynamic behaviour of a system. However, the specific purpose is more important to clarify and understand.

Sequence diagrams are used to capture the order of messages flowing from one object to another. A single diagram is not sufficient to describe the dynamic aspect of an entire system, so a set of diagrams are used to capture it as a whole.

Sequence diagrams are used when we want to understand the message flow. Message flow means the sequence of control flow from one object to another.

Sequence diagrams can be used –

- To model the flow of control by time sequence.
- For forward engineering.
- For reverse engineering.

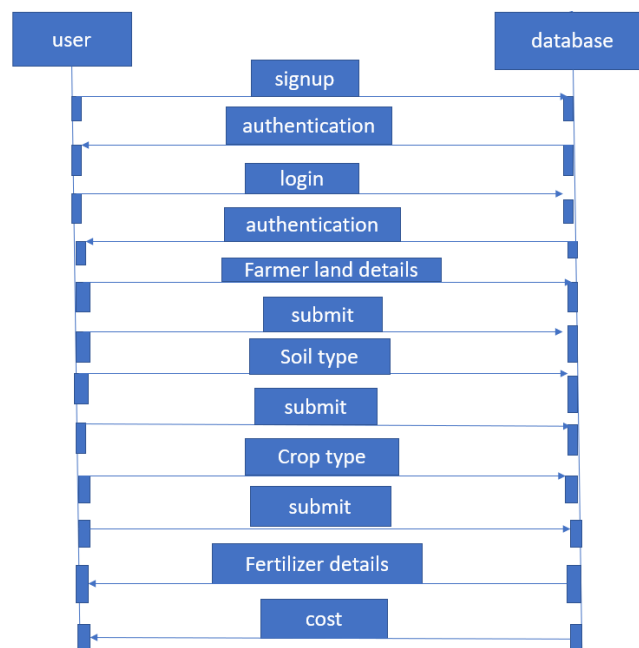


Fig: sequence diagram

4.2.4 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object.

Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements.

In the collaboration diagram, the method call sequence is indicated by some numbering technique. The number indicates how the methods are called one after another. Collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

Purpose of Collaboration Diagram:

The purpose of collaboration diagrams is to visualize the interactive behaviour of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction. Collaboration diagrams are used to capture the dynamic nature but from a different angle.

The purpose of collaboration diagram is –

- To capture the dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects.

Guidelines to draw Collaboration diagram:

The purpose of collaboration diagrams is to capture the dynamic aspect of a system. So to capture the dynamic aspect, we need to understand what a dynamic aspect is and how it is visualized. Dynamic aspect can be defined as the snapshot of the running system at a particular moment.

The collaboration diagram describes the organization of objects in a system taking part in the message flow.

Following things are to be identified clearly before drawing the sequence diagram --

- Objects taking part in the interaction.
- Message flows among the objects.
- Object organization.

Usage of Collaboration Diagram:

To understand the practical application, we need to understand the basic nature of collaboration diagram.

The main purpose of the collaboration diagrams is they are used to capture the dynamic behaviour of a system. However, the specific purpose is more important to clarify and understand.

Collaboration diagrams are used to describe the structural organization of the objects taking part in the interaction. A single diagram is not sufficient to describe the dynamic aspect of an entire system, so a set of diagrams are used to capture it as a whole. Collaboration diagrams are used when we want to understand the structural organization. Structural organization means the visual organization of the elements in a system.

Collaboration diagrams can be used –

- To model the flow of control by structural organizations.
- For forward engineering.
- For reverse engineering.

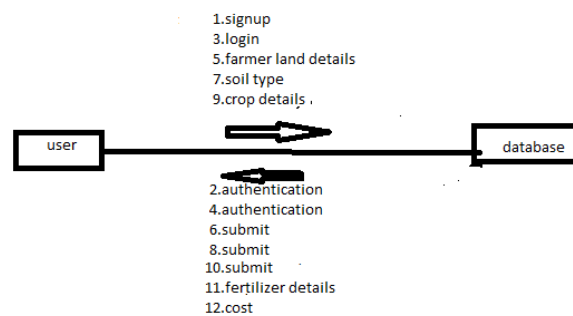


Fig: collaboration diagram

5 IMPLEMENTATION AND RESULTS

5.1INTRODUCTION

Functions are used for placing or storing the code which is to be repeated several times. For example, if we need same code, then we must have to write that code again and again. So in order to remove this we use functions.

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system is giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over an evaluation of change over methods apart from planning. Two major tasks for preparing the implementation are education and training of the users and testing of the system.

5.2METHOD OF IMPLEMENTATION

The more complex the system being implemented, the more involved will be the systems analysis and design efforts required for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then change over to his new fully tested system and the old system is discontinued.

5.2.1TECHNOLOGIES USED

The technologies that are used in the project are:

- i. Php
- ii. Java script
- iii. html

PHP :

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. This tutorial helps you to build your base with PHP.

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

PHP is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning PHP:

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible –

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

Applications of PHP

As mentioned before, PHP is one of the most widely used language over the web. I'm going to list few of them here:

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

In order to develop and run PHP Web pages three vital components need to be installed on your computer system.

- **Web Server** – PHP will work with virtually all Web Server software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server.
- **Database** – PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database.
- **PHP Parser** – In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web Browser. This tutorial will guide you how to install PHP parser on your computer.

PHP Parser Installation

Before you proceed it is important to make sure that you have proper environment setup on your machine to develop your web programs using PHP.

Type the following address into your browser's address box.

`http://127.0.0.1/info.php`

If this displays a page showing your PHP installation related information then it means you have PHP and Webserver installed properly. Otherwise you have to follow given procedure to install PHP on your computer.

This section will guide you to install and configure PHP over the following four platforms –

- [PHP Installation on Linux or Unix with Apache](#)
- [PHP Installation on Mac OS X with Apache](#)
- [PHP Installation on Windows NT/2000/XP with IIS](#)
- [PHP Installation on Windows NT/2000/XP with Apache](#)

Apache Configuration

If you are using Apache as a Web Server then this section will guide you to edit Apache Configuration Files.

PHP.INI File Configuration

The PHP configuration file, php.ini, is the final and most immediate way to affect PHP's functionality.

HTML

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

HTML is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning HTML:

- **Create Web site** - You can create a website or customize an existing web template if you know HTML well.
- **Become a web designer** - If you want to start a carrier as a professional web designer, HTML and CSS designing is a must skill.
- **Understand web** - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.
- **Learn other languages** - Once you understands the basic of HTML then other related technologies like java script, php, or angular are become easier to understand.

Applications of HTML

As mentioned before, HTML is one of the most widely used language over the web. I'm going to list few of them here:

- **Web pages development** - HTML is used to create pages which are rendered over the web. Almost every page of web is having html tags in it to render its details in browser.

- **Internet Navigation** - HTML provides tags which are used to navigate from one page to another and is heavily used in internet navigation.
- **Responsive UI** - HTML pages now-a-days works well on all platform, mobile, tabs, desktop or laptops owing to responsive design strategy.
- **Offline support** HTML pages once loaded can be made available offline on the machine without any need of internet.

Sr.No	Tag & Description
1	<!DOCTYPE...> This tag defines the document type and HTML version.
2	<html> This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
3	<head> This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
4	<title> The <title> tag is used inside the <head> tag to mention the document title.
5	<body> This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.
6	<h1> This tag represents the heading.
7	<p> This tag represents a paragraph.

This **HTML tutorial** is designed for the aspiring Web Designers and Developers with a need to understand the HTML in enough detail along with its simple overview, and practical examples. This tutorial will give you enough ingredients to start with HTML from where you can take yourself at higher level of expertise.

Prerequisites

Before proceeding with this **tutorial** you should have a basic working knowledge with Windows or Linux operating system, additionally you must be familiar with –

- Experience with any text editor like notepad, notepad++, or Edit plus etc.
- How to create directories and files on your computer.
- How to navigate through different directories.
- How to type content in a file and save them on a computer.
- Understanding about images in different formats like JPEG, PNG format.

HTML Tags

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**. Except few tags, most of the tags have their corresponding closing tags. For example, **<html>** has its closing tag **</html>** and **<body>** tag has its closing tag **</body>** tag etc.

Core Attributes

The four core attributes that can be used on the majority of HTML elements (although not all) are –

- Id
- Title
- Class
- Style

The Id Attribute

The **id** attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an id attribute on an element –

- If an element carries an id attribute as a unique identifier, it is possible to identify just that element and its content.
- If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

There are three internationalization attributes, which are available for most (although not all) XHTML elements.

- dir
- lang
- xml: lang

The dir Attribute

When *dir* attribute is used within the **<html>** tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

Value	Meaning
Ltr	Left to right (the default value)
Rtl	Right to left (for languages such as Hebrew or Arabic that are read right to left)

The lang Attribute

The **lang** attribute allows you to indicate the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML. This attribute has been replaced by the **xml: lang** attribute in new XHTML documents.

The values of the *lang* attribute are ISO-639 standard two-character language codes.

The xml: lang Attribute

The *xml: lang* attribute is the XHTML replacement for the *lang* attribute. The value of the *xml: lang* attribute should be an ISO-639 country code as mentioned in previous section.

Generic Attributes

Here's a table of some other attributes that are readily usable with many of the HTML tags.

Attribute	Options	Function
Align	right, left, centre	Horizontally aligns tags
Valign	top, middle, bottom	Vertically aligns tags within an HTML element.
Bgcolor	numeric, hexadecimal, RGB values	Places a background colour behind an element
background	URL	Places a background image behind an element
Id	User Defined	Names an element for use with Cascading Style Sheets.
Class	User Defined	Classifies an element for use with Cascading Style Sheets.
Width	Numeric Value	Specifies the width of tables, images, or table cells.

Height	Numeric Value	Specifies the height of tables, images, or table cells.
Title	User Defined	"Pop-up" title of the elements.

We will see related examples as we will proceed to study other HTML tags. For a complete list of HTML Tags and related attributes please check reference to [HTML Tags List](#)

To learn HTML, you will need to study various tags and understand how they behave, while formatting a textual document. Learning HTML is simple as users have to learn the usage of different tags in order to format the text or images to make a beautiful webpage.

JAVASCRIPT

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

- Java script is the most popular programming language in the world and that makes it a programmer's great choice. Once you learnt Java script, it helps you developing great front-end as well as back-end software using different Java script based frameworks like jQuery, Node.JS etc.
- Java script is everywhere, it comes installed on every modern web browser and so to learn Java script you really do not need any special environment setup. For example Chrome, Mozilla Firefox , Safari and every browser you know as of today, supports Java script.
- Java script helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience.
- JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as Java script Programmer.
- Due to high demand, there is tons of job growth and high pay for those who know JavaScript. You can navigate over to different job sites to see what having JavaScript skills looks like in the job market.
- Great thing about Java script is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.

There are many useful Java script frameworks and libraries available:

- Angular
- React
- jQuery
- Vue.js
- Ext.js

- Ember.js
- Meteor
- Mithril
- Node.js
- Polymer
- Aurelia
- Backbone.js

It is really impossible to give a complete list of all the available Java script frameworks and libraries. The Java script world is just too large and too much new is happening.

Applications of Java script Programming:

As mentioned before, Java script is one of the most widely used programming languages (Front-end as well as Back-end). It has its presence in almost every area of software development. I'm going to list a few of them here:

- **Client side validation** - This is really important to verify any user input before submitting it to the server and Java script plays an important role in validating those inputs at front-end itself.
- **Manipulating HTML Pages** – Java script helps in manipulating HTML page on the fly. This helps in adding and deleting any HTML tag very easily using java script and modify your HTML to change its look and feel based on different devices and requirements.
- **User Notifications** - You can use Java script to raise dynamic pop-ups on the webpages to give different types of notifications to your website visitors.
- **Back-end Data Loading** – Java script provides Ajax library which helps in loading back-end data while you are doing some other processing. This really gives an amazing experience to your website visitors.
- **Presentations** - JavaScript also provides the facility of creating presentations which gives website look and feel. JavaScript provides Reveal JS and Bespoke JS libraries to build a web-based slide presentations.
- **Server Applications** - Node JS is built on Chrome's Java script runtime for building fast and scalable network applications. This is an event based library which helps in developing very sophisticated server applications including Web Servers.

This list goes on, there are various areas where millions of software developers are happily using Java script to develop great websites and others software.

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as Live Script, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first

appearance in Netscape 2.0 in 1995 with the name Live Script. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defined a standard version of the core JavaScript language.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript

The merits of using JavaScript are –

- **Less server interaction** – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** – They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.

- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

JavaScript Development Tools

One of major strengths of JavaScript is that it does not require expensive development tools. You can start with a simple text editor such as Notepad. Since it is an interpreted language inside the context of a web browser, you don't even need to buy a compiler.

To make our life simpler, various vendors have come up with very nice JavaScript editing tools. Some of them are listed here –

- **Microsoft FrontPage** – Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.
- **Macromedia Dreamweaver MX** – Macromedia Dreamweaver MX is a very popular HTML and JavaScript editor in the professional web development crowd. It provides several handy prebuilt JavaScript components, integrates well with databases, and conforms to new standards such as XHTML and XML.
- **Macromedia Home Site 5** – Home Site 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

5.3 EXPLANATIONS OF KEY FUNCTIONS

Source code:

```
<!DOCTYPE html>
<html>
<head>
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title> Login Page </title>
```

```
<style>
```

```
Body {
```

```
    font-family: Calibri, Helvetica, sans-serif;
```

```
    background-color: pink;
```

```
}
```

```
button {
```

```

        background-color: #4CAF50;

        width: 100%;

color: orange;

        padding: 15px;

        margin: 10px 0px;

        border: none;

        cursor: pointer;

    }

form {

    border: 3px solid #f1f1f1;

}

input[type=text], input[type=password] {

    width: 100%;

    margin: 8px 0;

    padding: 12px 20px;

    display: inline-block;

    border: 2px solid green;

    box-sizing: border-box;

}

button:hover {

    opacity: 0.7;

}

.cancelbtn {

    width: auto;

    padding: 10px 18px;

```

```

        margin: 10px 5px;
    }

    .container {
        padding: 25px;
        background-color: lightblue;
    }
</style>
</head>
<body>
<center><h1> Farmer login form</h1></center>
<form>
<div class="container">
<label>Username : </label>
<input type="text" placeholder="Enter Username" name="username" required>
<label>Password : </label>
<input type="password" placeholder="Enter Password" name="password" required>

<input type="checkbox" checked="checked"> Remember me
<button type="button" class="cancelbtn"> Cancel</button>

    Forgot <a href="#"> password? </a>
<br><a href="signup.html"> Sign Up </a>
</div>
</form>

```

```

    <a href="Land1(1).html">

    <button>submit</button>

  </a>

</body>

</html>

!DOCTYPE html>

<html>

<style>

body {font-family: Arial, Helvetica, sans-serif;}

* {box-sizing: border-box;}


/* Full-width input fields */

input[type=text], input[type=password] {

  width: 100%;

  padding: 15px;

  margin: 5px 0 22px 0;

  display: inline-block;

  border: none;

  background: #f1f1f1;

}


/* Add a background color when the inputs get focus */

input[type=text]:focus, input[type=password]:focus {

  background-color: #ddd;

  outline: none;

```

```

}

/* Set a style for all buttons */

button {

    background-color: #4CAF50;

    color: white;

    padding: 14px 20px;

    margin: 8px 0;

    border: none;

    cursor: pointer;

    width: 100%;

    opacity: 0.9;

}

button:hover {

    opacity: 1;

}

/* Extra styles for the cancel button */

.cancelbtn {

    padding: 14px 20px;

    background-color: #f44336;

}

/* Float cancel and signup buttons and add an equal width */

```

```

.cancelbtn, .signupbtn {

    float: left;

    width: 50%;

}

/* Add padding to container elements */

.container {

    padding: 16px;

}

/* The Modal (background) */

.modal {

    display: none; /* Hidden by default */

    position: fixed; /* Stay in place */

    z-index: 1; /* Sit on top */

    left: 0;

    top: 0;

    width: 100%; /* Full width */

    height: 100%; /* Full height */

    overflow: auto; /* Enable scroll if needed */

    background-color: #474e5d;

    padding-top: 50px;

}

/* Modal Content/Box */

```

```
.modal-content {  
  
    background-color: #fefefe;  
  
    margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */  
  
    border: 1px solid #888;  
  
    width: 80%; /* Could be more or less, depending on screen size */  
  
}
```

```
/* Style the horizontal ruler */
```

```
hr {  
  
    border: 1px solid #f1f1f1;  
  
    margin-bottom: 25px;  
  
}
```

```
/* The Close Button (x) */
```

```
.close {  
  
    position: absolute;  
  
    right: 35px;  
  
    top: 15px;  
  
    font-size: 40px;  
  
    font-weight: bold;  
  
    color: #f1f1f1;  
  
}
```

```
.close:hover,  
.close:focus {
```

```
color: #f44336;

cursor: pointer;

}
```

```
/* Clear floats */
```

```
.clearfix::after {

    content: "";

    clear: both;

    display: table;

}
```

```
/* Change styles for cancel button and signup button on extra small screens */
```

```
@media screen and (max-width: 300px) {
```

```
    .cancelbtn, .signupbtn {

        width: 100%;

    }
```

```
}
```

```
</style>
```

```
<body>
```

```
<h1>Sign Up</h1>
```

```
<p>Please fill in this form to create an account.</p>
```

```
<hr>
```

```
<label for="email"><b>User name/Phone number</b></label>
```



```
<input type="text" placeholder="Enter User name/Phone number" name="username"
required>
```

```
<label for="psw"><b>Password</b></label>
```

```
<input type="password" placeholder="Enter Password" name="psw" required>
```

```
<label for="psw-repeat"><b>Repeat Password</b></label>
```

```
<input type="password" placeholder="Repeat Password" name="psw-repeat" required>
```

```
</label>
```

```
<input type="checkbox" checked="checked" name="remember" style="margin-
bottom:15px"> Remember me
```

```
</label>
```

```
<p>By creating an account you agree to our <a href="#" style="color:dodgerblue">Terms &
Privacy</a>.</p>
```

```
<div class="clearfix">
```

```
<button type="button" onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
```

```
</div>
```

```
</div>
```

```
</form>
```

```
<a href="login.html">
```

```
<button>Sign Up</button>
```

```
</a>
```

```
</div>
```

```
<script>
```

```
// Get the modal
```

```
var modal = document.getElementById('id01');
```

```
// When the user clicks anywhere outside of the modal, close it
```

```
window.onclick = function(event) {
```

```
    if (event.target == modal) {
```

```
        modal.style.display = "none";
```

```
    }
```

```
}
```

```
</script>
```

```
</body>
```

```
</html><!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>FARMERS LAND DETAILS</p>
```

```
<form>
```

```
<label for="number of acres">select number of acres:</label>
```

```
<select name="land" id="land">
```

```
<optgroup label="acres">
```

```
<option value="one">one</option>
```

```
<option value="two">two</option>
```

```

<option value="three">three</option>

<option value="four">four</option>

<option value="five">five</option>

<option value="six">six</option>

</optgroup>

</select>

<br><br>

</form>

<a href="soil.html">

<button>submit</button>

</a>

</body>

</html>

html>

<head>

</head>

<body>

<p>TYPES OF SOIL</p>

<form>

<label for="nature of soil">select nature of soils:</label>

<select name="name of the soil" id="name of the soil">

<option value="Alluvial soil">Alluvial soil</option>

<option value="Red soil">Red soil</option>

<option value="Black soil">Block soil</option>

<option value="Arid soil">Arid soil</option>

```

```
<option value="Laterite soil">Laterite soil</option>
```

```
</select>
```

```
<br><br>
```

```
<p>TYPES OF CROP</P>
```

```
<label for="crop type">select crop type:</label>
```

```
<select name="name of the crop" id="name of the crop">
```

```
<option value="Paddy">Paddy</option>
```

```
<option value="Wheat">Wheat</option>
```

```
<option value="Millets">Millets</option>
```

```
<option value="Cotton">Cotton</option>
```

```
<option value="Redgram">Redgram</option>
```

```
</select>
```

```
<br><br>
```

```
</form>
```

```
<a href="11.html">
```

```
<button>submit</button>
```

```
</a>
```

```
</body>
```

```
</html><!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<script>
```

```
var subjectObject = {
```

```

"PADDY": {
  "FERT1": ["1000"],
  "FERT2": ["1000"],
  "FERT3": ["1000"]
},
"COTTON": {
  "FERT4": ["4000"],
  "FERT5": ["5000"]
}
}

window.onload = function() {
  var cropSel = document.getElementById("crop");
  var fertSel = document.getElementById("fert");
  var CostSel = document.getElementById("Cost");
  for (var x in subjectObject) {
    cropSel.options[cropSel.options.length] = new Option(x, x);
  }
  cropSel.onchange = function() {
    //empty Cost- and fert- dropdowns
    CostSel.length = 1;
    fertSel.length = 1;
    //display correct values
    for (var y in subjectObject[this.value]) {
      fertSel.options[fertSel.options.length] = new Option(y, y);
    }
  }
}

```

```

    }

    fertSel.onChange = function() {

        //empty Cost dropdown

        CostSel.length = 1;

        //display correct values

        var z = subjectObject[cropSel.value][this.value];

        for (var i = 0; i<z.length; i++) {

            CostSel.options[CostSel.options.length] = new Option(z[i], z[i]);

        }

    }

}

</script>

</head>

<body>

<h1>Selecting fertilizer based on crop</h1>

<form name="form1" id="form1" action="/action_page.php">

crop: <select name="crop" id="crop">

<option value="" selected="selected">Select crop</option>

</select>

<br>

<br>

fert: <select name="fert" id="fert">

<option value="" selected="selected">Please select crop first</option>

</select>

```


Cost: <select name="Cost" id="Cost">

<option value="" selected="selected">Please select fert first</option>

</select>

</form>

<button>Submit</button>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<style>

.card {

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);

max-width: 300px;

margin: auto;

text-align: center;

font-family: arial;

}

.price {

color: grey;

```

    font-size: 22px;

}

.card button {

    border: none;

    outline: 0;

    padding: 12px;

color: white;

    background-color: #000;

    text-align: center;

    cursor: pointer;

    width: 100%;

    font-size: 18px;

}

.card button:hover {

    opacity: 0.7;

}

</style>

</head>

<body>

<h2 style="text-align:center">Fertilizer 1</h2>

<div class="card">

<imgsrc="C:\Users\bindubhargavi\Desktop\17-578\2.jpg.jpeg" alt="fert"
style="width:100%">

<h1>Combo of 3</h1>

<p class="price">3000rps</p>

<p>These fertilizer is used to paddy</p>

```



```

<h4 style="text-align:center">Fertilizer 2</h4>

<div class="card">

<imgsrc="C:\Users\bindubhargavi\Desktop\17-578\1.jpg.jpeg" alt="fert"
style="width:100%">

<h3>Combo of 3</h3>

<p class="price">1500rps</p>

<p>This frtilizer is used for cotton</p>

<h6 style="text-align:center">Fertilizer 3</h6>

<div class="card">

<imgsrc="C:\Users\bindubhargavi\Desktop\17-578\3.jpg.jpeg"alt="fert"
style="width:100%">

<h5>Combo of 3</h5>

<p class="price">2000rps</p>

<p>This fertilizer is used for wheat</p>

<br>

<br>

<a href="11.html">

<button>Shop More</button>

</a>

<br>

<br>

<p><button>Submit</button></p>

</div>

</body>

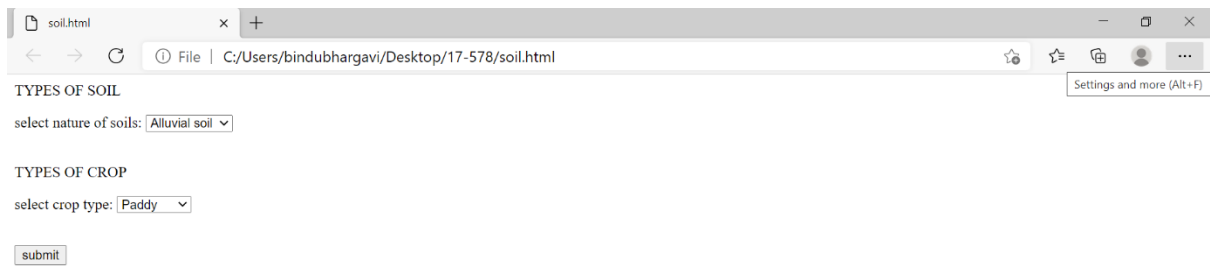
</html>

```

5.4 OUTPUT SCREEN SHOTS:

The first screenshot shows a web browser window with the address bar displaying 'C:/Users/bindubhargavi/Desktop/17-578/login.html'. The page has a pink header with the title 'Farmer login form'. Below the header is a light blue form area containing two input fields: 'Username : Enter Username' and 'Password : Enter Password'. There is a 'Remember me' checkbox, a 'Cancel' button, and a 'Forgot password?' link. A 'Sign Up' link is at the bottom left of the form area. A green 'submit' button is centered below the form area. The Windows taskbar is visible at the bottom.

The second screenshot shows a web browser window with the address bar displaying 'C:/Users/bindubhargavi/Desktop/17-578/signup.html'. The page has a white header with the title 'Sign Up'. Below the header is a light gray form area with the text 'Please fill in this form to create an account.' followed by three input fields: 'User name/Phone number', 'Password', and 'Repeat Password'. There is a 'Remember me' checkbox and a link to 'Terms & Privacy'. A red 'Cancel' button and a green 'Sign Up' button are at the bottom of the form area. A dark gray notification box in the bottom right corner says 'Screenshot saved' and 'The screenshot was added to your OneDrive.' The Windows taskbar is visible at the bottom.





Selecting fertilizer based on crop

crop:

fert:

Cost:



Fertilizer 1

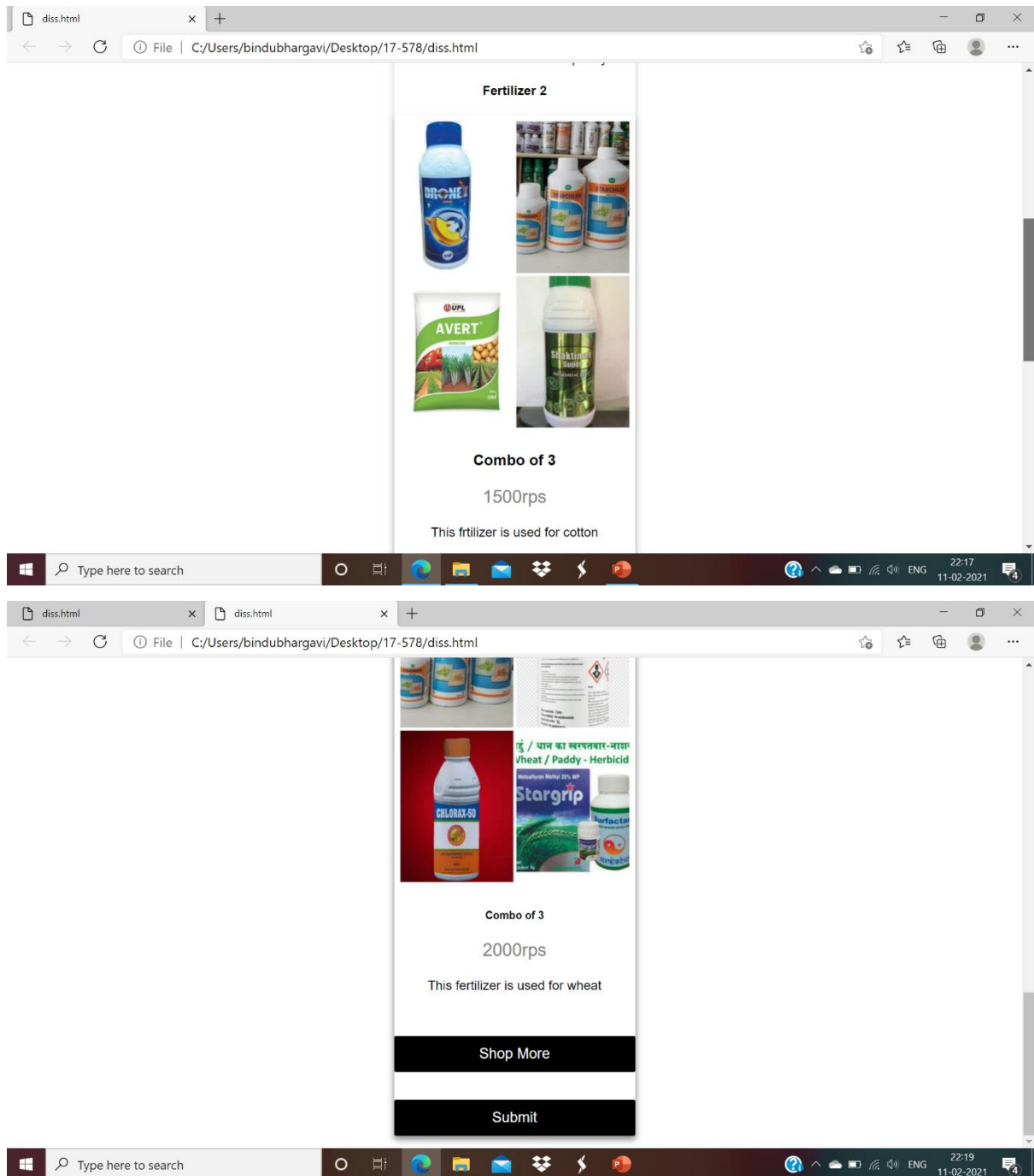


Combo of 3

3000rps

These fertilizer is used to paddy





6. TESTING

6.1 INTRODUCTION

Software Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in

contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing.

In simple terms, Software Testing means Verification of Application Under Test (AUT). Software testing is a critical element of software quality and assurance and represents ultimate review of specifications, design and coding. Testing is an exposure of the system to trial input to see whether it produces correct output.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy and usability. It mainly aims at measuring specification, functionality and performance of a software program or application.

Software Testing can be done in two ways:

1. **Verification:** It refers to the set of tasks that ensure that software correctly implements a specific function.
2. **Validation:** It refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”

Importance of Software Testing:

The importance of software testing is imperative. Software Testing is important because of the following reasons:

1. Software Testing points out the defects and errors that were made during the development phases. It looks for any mistake made by the programmer during the implementation phase of the software.
2. It ensures that the customer finds the organization reliable and their satisfaction in the application is maintained. Sometimes contracts include monetary penalties with respect to the timeline and quality of the product and software testing prevent monetary losses.
3. It also ensures the Quality of the product. Quality product delivered to the customers helps in gaining their confidence. It makes sure that the software application requires lower maintenance cost and results in more accurate, consistent and reliable results.
4. Users are not inclined to use software that has bugs. They may not adopt software if they are not happy with the stability of the application. Testing is important for the product to stay in business.
5. It's important to ensure that the application should not result in any failures because it can be very expensive in the future or in the later stages of the development.

Applications of Software Testing:

- **Cost Effective Development** - Early testing saves both time and cost in many aspects, however reducing the cost without testing may result in improper design of a software application rendering the product useless.
- **Product Improvement** - During the SDLC phases, testing is never a time-consuming process. However diagnosing and fixing the errors identified during proper testing is a time-consuming but productive activity.
- **Test Automation** - Test Automation reduces the testing time, but it is not possible to start test automation at any time during software development. Test automaton should

be started when the software has been manually tested and is stable to some extent. Moreover, test automation can never be used if requirements keep changing.

- **Quality Check** - Software testing helps in determining following set of properties of any software such as
 - Functionality
 - Reliability
 - Usability
 - Efficiency
 - Maintainability
 - Portability

6.1.1 TYPES OF SOFTWARE TESTING:

Software Testing can be broadly classified into two types:

i. Manual Testing:

Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually.

Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software. The difference between expected output and output, given by the software, is defined as a defect. The developer fixed the defects and handed it to the tester for retesting.

Manual testing is mandatory for every newly developed software before automated testing. This testing requires great efforts and time, but it gives the surety of bug-free software. Manual Testing requires knowledge of manual testing techniques but not of any automated testing tool.

Types of Manual Testing:

There are various methods used for manual testing. Each technique is used according to its testing criteria. Types of manual testing are given below:

- White Box Testing
- Black Box Testing

Advantages of Manual Testing:

- It does not require programming knowledge while using the Black box method
- It is used to test dynamically changing GUI design.
- Tester interacts with software as a real user so that they are able to discover usability and user interface issues.
- It ensures that the software is a hundred percent bug-free.
- It is cost-effective.
- Easy to learn for new testers.

Disadvantages of Manual Testing:

- It requires a large number of human resources.
- It is very time-consuming.

- Tester develops test cases based on their skills and experience. There is no evidence that they have covered all functions or not.
- Test cases cannot be used again. Need to develop separate test cases for each new software.
- It does not provide testing on all aspects of testing.
- Since two teams work together, sometimes it is difficult to understand each other's motives, it can mislead the process.

ii. Automation Testing:

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.

Apart from regression testing, automation testing is also used to test the application from load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing.

Advantages of Automation Testing:

- Automation testing takes less time than manual testing.
- A tester can test the response of the software if the execution of the same operation is repeated several times.
- Automation Testing provides re-usability of test cases on testing of different versions of the same software.
- Automation testing is reliable as it eliminates hidden errors by executing test cases again in the same way.
- Automation Testing is comprehensive as test cases cover each and every feature of the application.
- It does not require many human resources, instead of writing test cases and testing them manually, they need an automation testing engineer to run them.
- The cost of automation testing is less than manual testing because it requires a few human resources.

Disadvantages of Automation Testing:

- Automation Testing requires high-level skilled testers.
- It requires high-quality testing tools.
- When it encounters an unsuccessful test case, the analysis of the whole event is complicated.
- Test maintenance is expensive because high fee license testing equipment is necessary.
- Debugging is mandatory if a less effective error has not been solved, it can lead to fatal results.

6.1.2 TESTING ACTIVITIES:

Software level testing can be majorly classified into 4 levels:

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

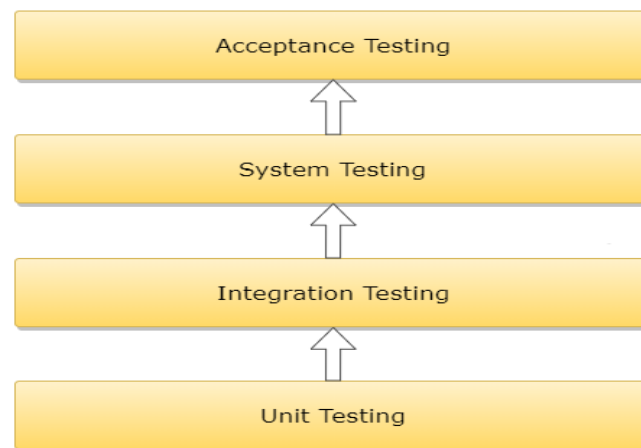


Figure 6.1: Levels of Testing

i. Unit Testing:

Unit Testing is a software testing technique by means of which individual units of software i.e. group of computer program modules, usage procedures and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent module is tested to determine if there are any issue by the developer himself. It is correlated with functional correctness of the independent modules.

Unit Testing is defined as a type of software testing where individual components of a software are tested. Unit Testing of software product is carried out during the development of an application. An individual component may be either an individual function or a procedure. Unit Testing is typically performed by the developer.

Unit testing focuses on the building blocks of the software system, that is, objects and subsystems. There are three motivations behind focusing on components. First, unit testing reduces the complexity of the overall test activities, allowing us to focus on smaller units of the system. Unit testing makes it easier to pinpoint and correct faults given that few computers are involved in this test. Unit testing allows parallelism in the testing activities; that is each component can be tested independently of one another.

The specific candidates for unit testing are chosen from the object model and the system decomposition of the system. In principle, all the objects developed during the development process should be tested.

Objective of Unit testing:

The objective of Unit Testing is:

1. To isolate a section of code.
2. To verify the correctness of code.
3. To test every function and procedure.
4. To fix bug early in development cycle and to save costs.
5. To help the developers to understand the code base and enable them to make changes quickly.
6. To help for code reuse.

Advantages:

- Reduces Cost of Testing as defects are captured in very early phase.
- Unit Tests, when integrated with build gives the quality of the build as well
- Unit Testing allows developers to learn what functionality is provided by a unit and how to use it to gain a basic understanding of the unit API.
- Unit testing allows the programmer to refine code and make sure the module works properly.
- Unit testing enables to test parts of the project without waiting for others to be completed.

ii. Integration Testing:

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

In integration testing, testers test the interfaces between the different modules. These modules combine together to form a bigger component or the system. Hence, it becomes very crucial to validate their behaviour when they work together. Apart from the interfaces, they also test the integrated components. Integration testing is the next level of testing after unit testing. Testers do it after completion of the unit testing phase. Integration testing techniques can be a white box or black box depending on the project requirements.

Objectives of Integration Testing:

Integration testing reduces the risk of finding the defects in integrated components in the System testing phase. Integration defects can be complex to fix and they can be time-consuming as well. Finding them early in the cycle eliminates the risk of making too many changes at the System testing phase. As each of the integrating components has been tested in the integration phase, the System testing can focus on end-to-end journeys and user-specific flows.

- Reducing risk by testing integrating components as they become available.
- Verify whether the functional and non-functional behaviours of the interfaces are designed as per the specification.
- To build confidence in the quality of the interfaces.
- To find defects in the components, system or in the interfaces.
- Prevents defects from escaping to higher test levels of testing i.e System testing.

Guidelines for Integration Testing:

- First, determine the test case strategy through which executable test cases can be prepared according to test data.
- Examine the structure and architecture of the application and identify the crucial modules to test them first.
- Design test cases to verify each interface in detail.
- Choose input data for test case execution. Input data plays a significant role in testing.
- Fix defects and retest.

Advantages of Integration testing:

- It helps to find defects from links and interfaces between the modules.
- It boosts the confidence level of the team in the product as it validates the group of modules together.
- Integration tests run faster than the end to end test scenarios.
- It results in higher code coverage.
- It starts in the early stages when the entire module may not be ready. Hence, it avoids the bugs getting into the system.

iii. System Testing:

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

In other words, System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not. System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product.

The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects within both the integrated units and the whole system. The result of system testing is the observed behaviour of a component or a system when it is tested.

System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behaviour of the system and also the expectations of the customer.

Objectives of System Testing:

The primary objectives of System testing are as below:

- One of the primary objectives of System testing is to reduce risk. Even after individual testing of components, risk of how they will all come together to form a

complete System still exists. System testing eliminates this risk by ensuring that it will function as per customer requirements.

- System testing must verify whether the design of the functional and non-functional behaviours of the system is as per the customer's specifications.
- Validate that the system is complete and will work as expected.
- System testing aims to build confidence in the quality of the system as a whole.
- System testing also aims to find defects and to prevent defects from escaping to higher test levels or production. Additionally, it is the only phase that occurs on the full System just before the User Acceptance testing. So it's critical to find all the possible defects at this stage, and they don't leak to production.
- System Testing results are used by stakeholders to make release decisions. The Entry criteria for User Acceptance testing is the basis completion of System Testing. System testing may also adhere to legal or regulatory requirements or standards.

Guidelines for System Testing:

- The very first step is to create a Test Plan.
- Create System Test Cases and test scripts.
- Prepare the test data required for this testing.
- Execute the system test cases and script.
- Report the bugs. Re-testing the bugs once fixed.
- Regression testing to verify the impact of the change in the code.
- Repetition of the testing cycle until the system is ready to be deployed.
- Sign off from the testing team.

Advantages of System Testing:

- It covers a complete end to end software testing.
- The business requirements and system software architecture are both tested in system testing.
- Appropriate system testing help in relieving after production goes live issues and bugs.
- System testing is led in a situation like a production condition or some of the time it is finished with production parallel test condition where the same data input is feed to the exiting framework and new framework to look at the differences in functionalities removed and added. This causes the client to understand the new framework better and feel great with new functionalities included or existing functionalities revised or removed.

iv. Acceptance Testing:

Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

The standard definition of Acceptance testing is given as, "It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorized entities to determine whether to accept the system or not."

Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use.

The main goal behind acceptance testing is to check whether the developed software product passes the acceptance norms defined on the basis of user and business requirements, so as to declare it acceptable or non-acceptable for its use by the users. Acceptance testing is one of the last types of software testing performed over a software or application. It is conducted by a pool of targeted users to ensure the readiness and quality of the system from user's perspective, which allows the team to meet their needs and expectations.

Objectives of Acceptance Testing:

Following are the three major objectives of Acceptance Testing:

- Confirm that the system meets the agreed-upon criteria.
- Identify and resolve discrepancies, if there are any.
- Determine the readiness of the system for cut-over to live operations. The final acceptance of a system for deployment is conditioned upon the outcome of the acceptance testing. The acceptance test team produces an acceptance test report which outlines the acceptance conditions.

Advantages of Acceptance Testing:

Acceptance testing has the following benefits, complementing those which can be obtained from unit tests:

- Encouraging closer collaboration between developers on the one hand and customers, users or domain experts on the other, as they entail that business requirements should be expressed
- Providing a clear and unambiguous “contract” between customers and developers; a product which passes acceptance tests will be considered adequate (though customers and developers might refine existing tests or suggest new ones as necessary).
- Decreasing the chance and severity both of new defects and regressions (defects impairing functionality previously reviewed and declared acceptable).

6.2 DESIGN OF TEST CASES AND SCENARIOS

6.2.1 TEST CASE DESIGN:

The design of tests for software and other engineering products can be as challenging as the initial design of the product. Test case methods provide the developer with a systematic approach to testing. Moreover, these methods provide a mechanism that can help to ensure the completeness of tests and provide the highest likelihood for uncovering errors in software.

Any Engineered product can be tested in either of the two ways:

1. Knowing the specified function that a product has been designed to perform, tests can be conducted. These tests demonstrate whether each function is full operational and at the same time searches for errors in each function.
2. Knowing the internal workings of a product, tests can be conducted to ensure that internal operations are performed according to specifications and all internal components hence been adequately exercised.

Test case design methods are divided into two types:

1. White-box testing
2. Black-box testing
- 3.

1. White-Box Testing

White –box testing, sometimes called glass-box testing is a test, case designed method that uses the control structure of the procedural design to derive test cases. Using white-box testing methods, the s/w engineer can derive test cases that guarantee that all independent paths within a module have been exercised at least once. Exercise all logical decisions on their true and false sides. Execute all loops at their boundaries and within their operational bounds. Exercise internal data structures to ensure their validity.

Basis path testing is a white-box testing technique. The basis path method enables the test case designer to derive a logical complexity measure of a procedural design and use this measure as a guide for defining a basis set are guaranteed to exercise every statement in the program at least one time during testing.

The following table lists the advantages and disadvantages of white-box testing.

Table 6.2: Advantages and Disadvantages of White box testing

| Advantages | Disadvantages |
|--|---|
| As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively. | Due to the fact that a skilled tester is needed to perform white-box testing, the costs are increased. |
| It helps in optimizing the code. | Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems, as many paths will go untested. |
| Extra lines of code can be removed which can bring in hidden defects. | It is difficult to maintain white-box testing, as it requires specialized tools like code analysers and debugging tools. |
| Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing. | |

2. Black-Box Testing

Black-box testing, also called behavioural testing, focuses on the functional requirements of the s/w. Black-box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements of a program. It is a complementary approach that is likely to uncover a different class of errors that white-box methods could not.

Black-box testing attempted to find errors in the following categories.

- Incorrect or missing functions.

- Interface errors.
- Errors in data structures or external data base access.
- Behaviour or performance errors.
- Initialization and termination errors.

Black-box testing purposely disregards control structure; attention is focused on information domain. By applying black-box techniques, we derive a set of cases that satisfies the criteria test cases that reduce, by a count that is greater than one, the number of additional test cases that must be designed to achieve reasonable testing. Test cases that tell us something about the presence or absence of classes of errors, rather than an error associated only with the specified.

The following table lists the advantages and disadvantages of black-box testing.

Table 6.2: Advantages and Disadvantages of Black box testing

| Advantages | Disadvantages |
|---|---|
| Well suited and efficient for large code segments. | Limited coverage, since only a selected number of test scenarios is actually performed. |
| Code access is not required. | Inefficient testing, due to the fact that the tester only has limited knowledge about an application. |
| Clearly separates user's perspective from the developer's perspective through visibly defined roles. | Blind coverage, since the tester cannot target specific code segments or errorprone areas. |
| Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language, or operating systems. | The test cases are difficult to design. |

6.2.2SCENARIOS

Table 6.3: Testing Scenarios

| S.NO. | Description | Detected Stage | Result | Remarks |
|--------------|--|-----------------------|--|---------------------|
| 1 | Click on the Generate test and train model before uploading the dataset | Unit Testing | Warning indicating to upload dataset first | Result dissatisfied |
| 2 | Click on the Generate test and train model after uploading the dataset | Unit Testing | File Uploaded | Result satisfied |
| 3 | Click on Run Support Vector Classifier before training and testing the model | Unit Testing | Warning indicating to Train and test the model | Result dissatisfied |
| 4 | Click on Run Support Vector | Unit | Display the results | Result |

| | | | | |
|---|--|--------------|--|---------------------|
| | Classifier after training and testing the model | Testing | | satisfied |
| 5 | Click on Run Gaussian Naive Bayes Classifier before training and testing the model | Unit Testing | Warning indicating to Train and test the model | Result dissatisfied |
| 6 | Click on Run Gaussian Naive Bayes Classifier after training and testing the model | Unit Testing | Display the results | Result satisfied |
| 7 | Click on Run Gaussian Naive Bayes Classifier before training and testing the model | Unit Testing | Warning indicating to Train and test the model | Result dissatisfied |
| 8 | Click on Run Random Forest Classifier after training and testing the model | Unit Testing | Display the results | Result satisfied |

7.CONCLUSION

7.1 PROJECT CONCLUSION:

At the end of this project, we have acquired the result of an accurate fertilizer for particular crop. In comparison to existing modules, this proposed module is prevent from getting cheated and more helpful for cultivation group. this application can give accurate information regarding fertilizers.

7.2 FUTURE ENHANCEMENT:

- Our future work will try to represent this into a software application using the new technologies like Artificial Intelligence and Deep Learning.
- We can also sale of fertilizers in future.
- The dataset available on day-to-day processing may become outdated, it is necessary to have updated data for effective.

8.REFERENCE

8.1TEXT BOOKS

- The Joy of PHP Programming: A Beginner's Guide to Programming Interactive Web Applications with PHP and MySQL author – Alan Forbes Latest Edition – Fifth Edition Publisher – Plum Island
- Head First PHP & MySQL Author – Lynn Beighley & Michael Morrison Latest Edition – First Edition Publisher – O'Reilly
- A guide to html author- AshleyMenhennett
- A smarter way to learn java script author-Mark Mayers.

8.2WEBSITES

- www.google.com
- www.tutorialspoint.com
- www.geeksforgeeks.org
- www.w3schools.com
- ieeexplore.ieee.org
- www.wikipedia.org
- www.slideshare.net

8.3PAPER REFERENCES

- A sample research paper on fertilizers by James Smith, Southern Illinois University,2016
- Choosing fertilizers
- Effects of choosing appropriate fertilizers: a review 1, Olajire-Ajayi BL, 2, Dada OV, 3, Wahab OM, and 4, OI Ojo Dept of Forestry Technology, Federal College Of Forestry, Ibadan, Nigeria Dept of Agricultural Engineering, Ladoke Akintola University of Technology, Nigeria.