# KALINDI COLLEGE

UNIVERSITY OF DELHI

# CROP MANAGEMENT IN NUTRIENT DISTRIBUTION

SUBMITTED BY:

ANSHIKA VARSHNEY 21033570008

GUNJAN BADHAUTIYA 21033570021

PRACHI AGGARWAL 21033570042

**SUBMITTED TO:**

**MS. SWEETY MA'AM**

# ACKNOWLEDGEMENT

We would like to express our sincere thanks of gratitude to our software engineering professor Ms. Sweety as well as our principal to provide such opportunity to do this wonderful and knowledgeable project CROP MANAGEMENT IN NUTRIENT DISTRIBUTION and enlighten us with her knowledge and guidance.

# CONTENTS

- Risk Table

❖ Design Engineering

- Pseudocode

❖ Testing

- Control Flow Diagram
- Cyclomatic Complexity

❖ Future Scope

❖ References

# PROBLEM STATEMENT:

The existing agricultural activities in most of the rural areas are still not compatible and inefficient in this flourishing world resulting losses to farmers economically, emotionally and ethically that in turn are fatal for farmers which leads to suicidal thoughts. The need for a modern farming technique has become critical to address these challenges. This project is concerned about the awareness of new techniques, diversity of biofortified and hybrid crops rich in nutrients, variety of resources such as harvesting machines and other useful appliances with their proper knowledge and utilization of organic fertilizers and natural manures so that they can increase their yield with proper irrigation with modern techniques like drip system, sprinklers and many more. Moreover, they get into traps and frauds because of lack of awareness about current minimum selling price in the market and the resources which they can avail in the market at the low cost. Also, they can reach several Government facilities are provided by government that can help them about which they are not acquainted through this project.

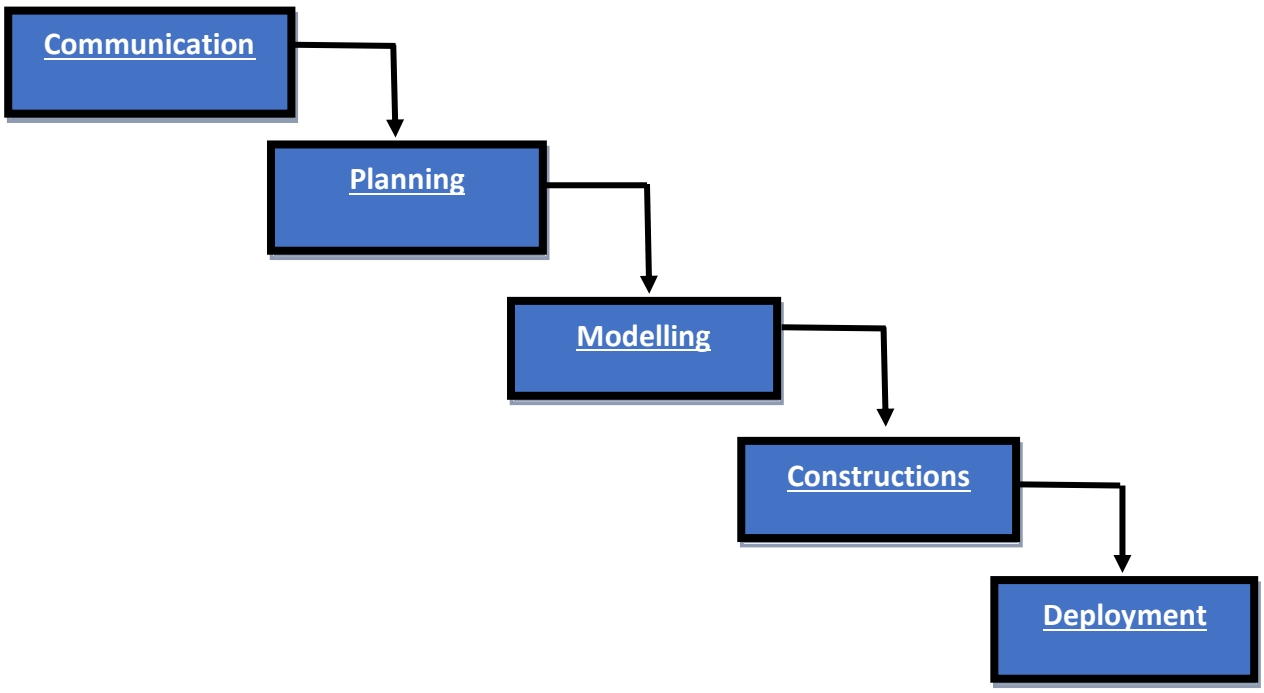
The successful implementation of crop management in nutrient distribution will result in growth in yield of crops making farmer economically stable and optimization of money usage for using resources with efficient government facilities.

# PROCESS MODEL

A software process model is an abstraction of the software development process. The models specify the stages and order of a process. This is a representation of the order of activities of the process and the sequence in which they are performed.

The waterfall model, sometimes called the classic life cycle, suggests a systematic, sequential approach 2 to software development that begins with customer specification of requirements and progresses through planning, modeling, construction, and deployment, culminating in ongoing support of the completed software.

Here for our project we are using Waterfall Model.



## Why we are using Waterfall Model?

- The reason of using this model is that we are clear with all the requirements and the changes tot the requirements are not expected during the development process.

- Also, There is no requirement of feedback at every step and no parallelism is needed.

- Moreover, it is a simple project with least complexity.

- Its is a linear and sequential approach to software development, which means that each phase must be completed before the next one can begin.

# SOFTWARE REQUIREMENT SPECIFICATIONS(SRS)

Software Requirements Specification (SRS) is a document that describes the requirements for a software system or application. It is a comprehensive document that provides a detailed description of what the software system is expected to do and how it is expected to perform. The SRS document is typically used as a reference document throughout the software development lifecycle. It helps to ensure that the software system meets the requirements of the stakeholders and provides a basis for testing, verification, and validation of the system.

## Scope of the Project

This project will reach out the maximum percentage of farmers and concern authorities and will help them grow by providing appropriate data.

The system we are using will be based on android application which can be operated in any kind of phone, tablets and pcs.

## Modules

- User LogIn: Users have to first register themselves for the usage of application. This is the one time process where user have to define username, phone no and password.
- Admin LogIn: The system is unde supervision of admin who manages the different aspects of this application.
- Crop Description: The system will provide user with the information of different-different crops regarding seasons, crop nutrition, high yield and so on.
- Land Description: The system will provide user with information about land where user can check about yield fertility and the resources needed to maintain it.
- Resources: Using this application user can find about the resources availability in different locations and the resources can be fertilizers,

pesticides, insecticides and hardware machines.

- Minimum Selling Prices(MSP): This application elightened the user about the MSP.
- Feedback and Survey: The system has a feedback form, where user can provide feedback into the system along with they can be the part of our ground survey and survey form will be optional for users

## User Functionalities

1. Login
2. Feedback Form Filling
3. Browsing Columns

## Admin Functionalities

1. Login
2. User Dashboard Details
3. Updating pages
4. Reviewing Feedback

## Hardware and Software Requirements

### Hardware:

a. RAM: 750 MB and above.

b. Mobiles should have internet facility.

C. Operating System: Window 10, Mac, and above.

### Software:

a. Front end: XML, JAVA

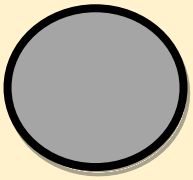
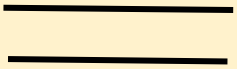
b. Back end: SQL server, C++

# DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of how data moves through a system. It is a modeling technique used in software engineering to describe the flow of data within an information system.

A DFD shows how data enters and exits the system, the various processes that transform the data, and the storage locations for the data.

DFD can be used to identify the inputs and outputs of the systems , the process that take place and the relationships between different elements of the systems , and can be used to develop a understanding of system functionalities.

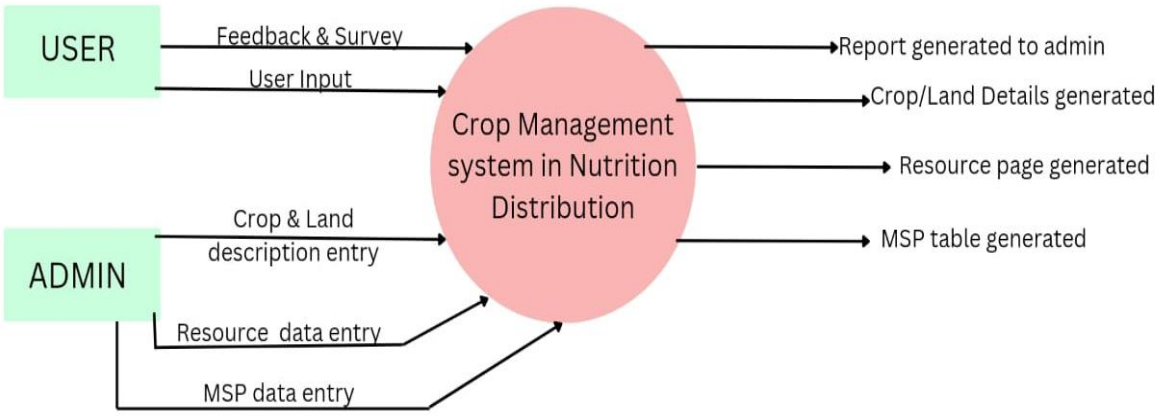
| Symbol | Name | Description |
|---|---|---|
|  | Process | Perform some transformation of input data to yield output data. |
|  | Data Flow | Used to connect processes to each other, to sources or sinks. |
|  | Data Store | Repository of Data. |
|  | External Entity/ Source sink | A source of system inputs or sink of the systems outputs. |

# Context Diagram

Context Diagram show the interactions between the system and other actors (external factors) with which the system is designed to interface.

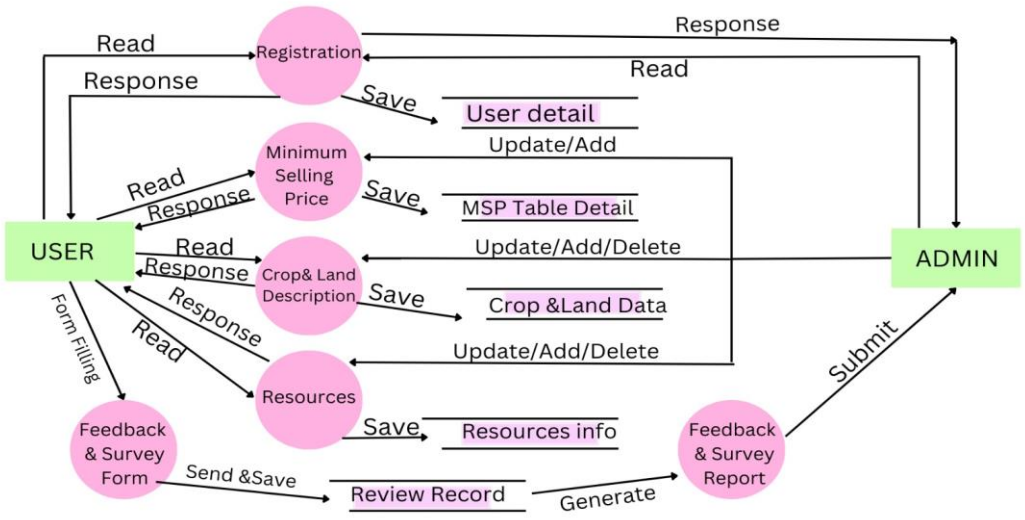
It is the simple model that defines the boundaries and interfaces of the proposed system with external world.

It identifies the entities outside the proposed system that interact with the system.

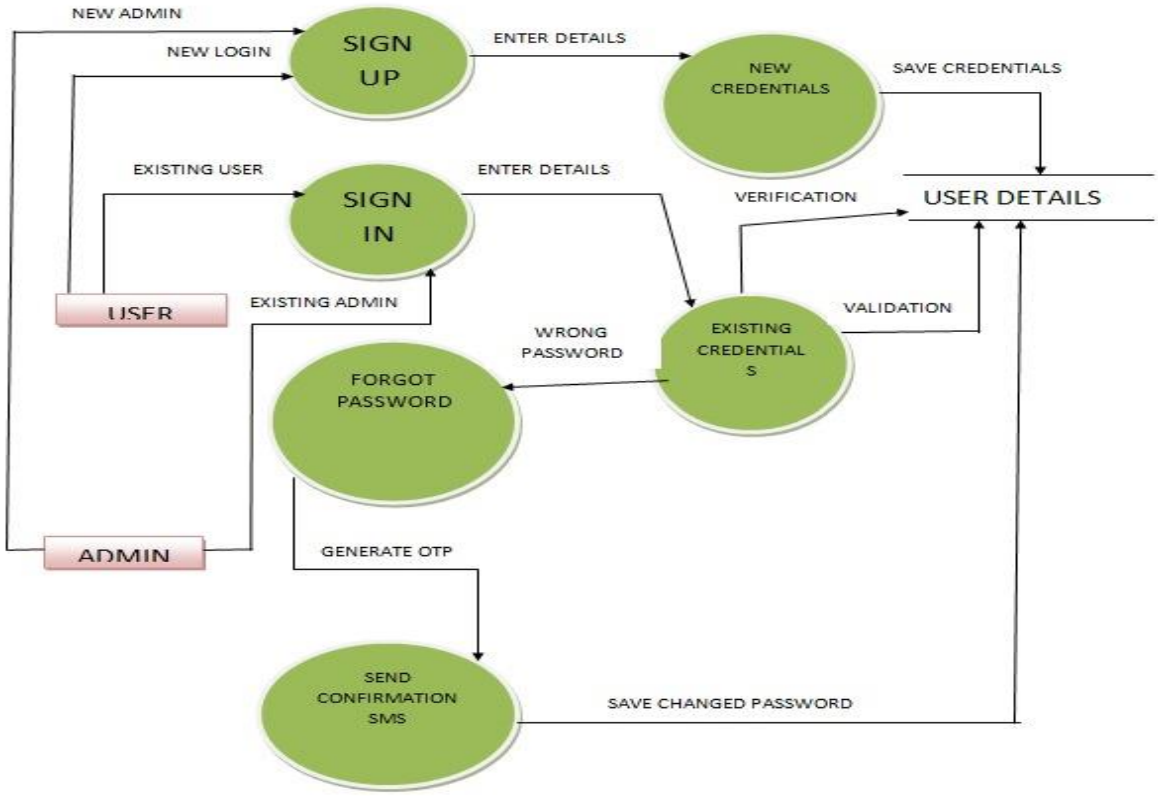


# Leveling

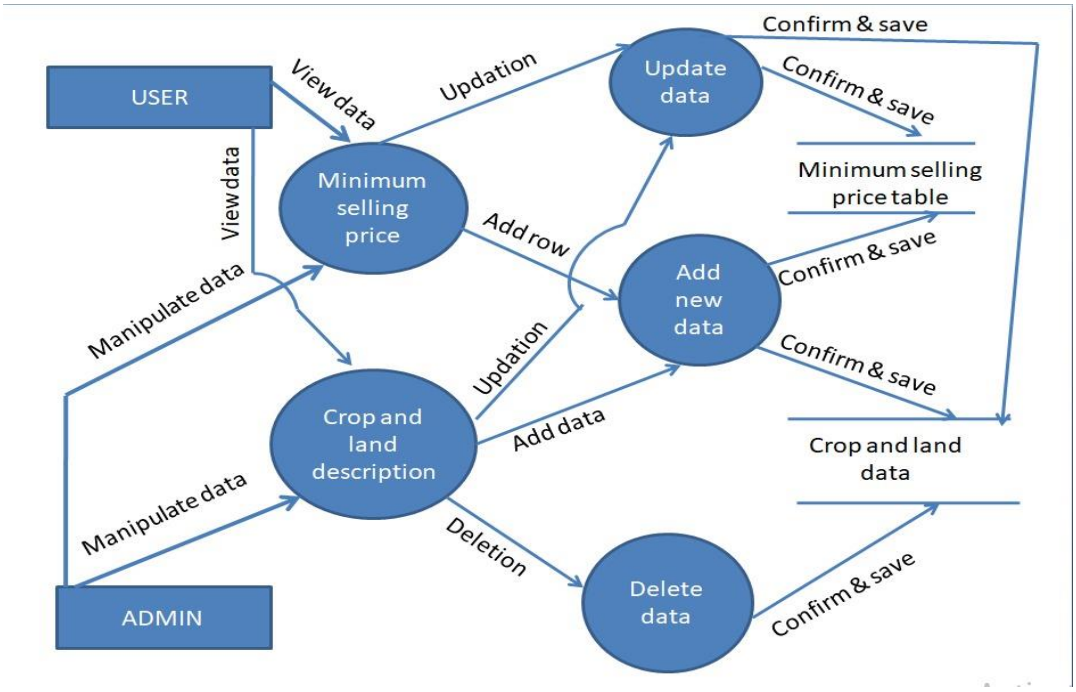
## LEVEL 1:



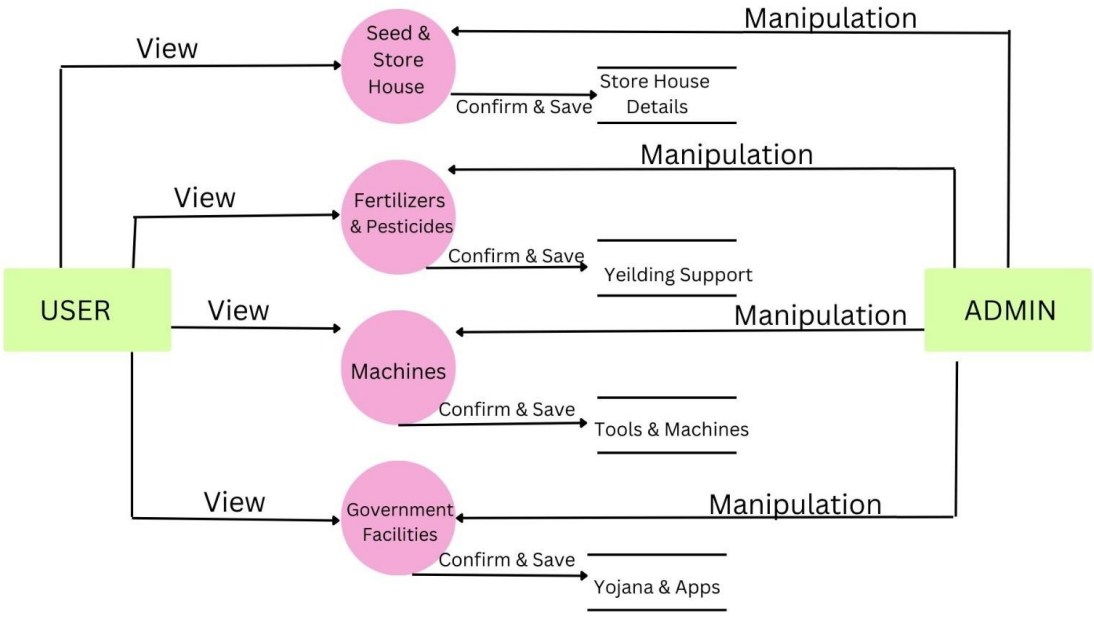

## LEVEL 2.1:



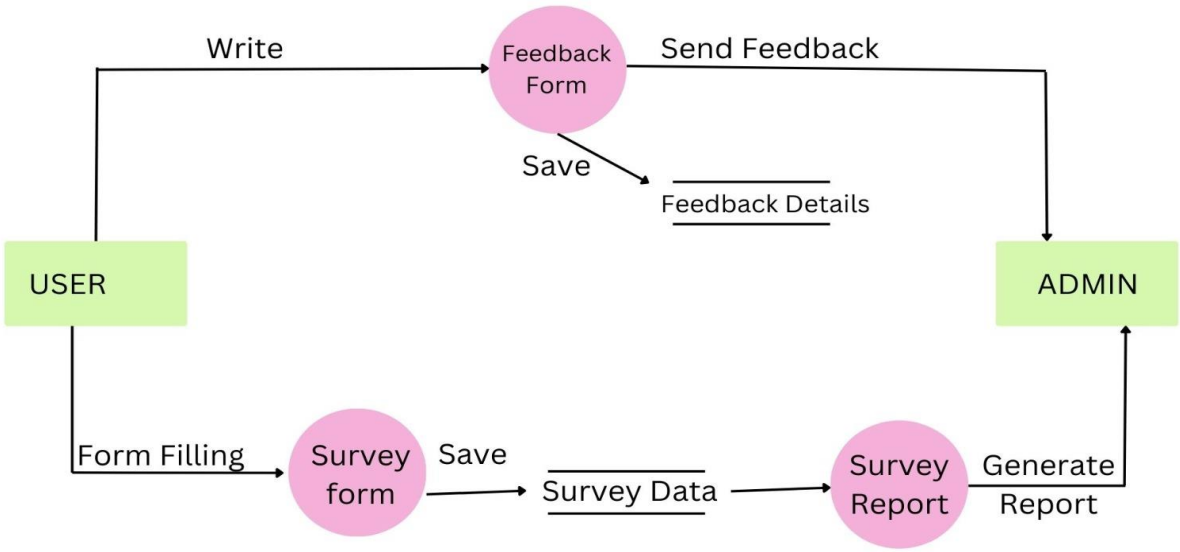
## LEVEL 2.2:




## LEVEL 2.3:



## LEVEL 2.4:


Write
Feedback Form
Send Feedback
Save
Feedback Details
USER
ADMIN
Form Filling
Survey form
Save
Survey Data
Survey Report
Generate Report

# DATA DICTIONARIES:

A data dictionary is a central repository that contains metadata about the data used in a software system. It is a document that provides detailed information about the data elements, their relationships, and their characteristics.

Improves analyst/ user communication.

It is important step in building a database.

1. **Data Elements:** Smallest unit of data that provides functions no further decomposition.
2. **Data Structure:** Group of data element handled as a unit.
3. **Data Flows and Data Stores:** Data flows are data structure in moving and data stores are data structure at rest.

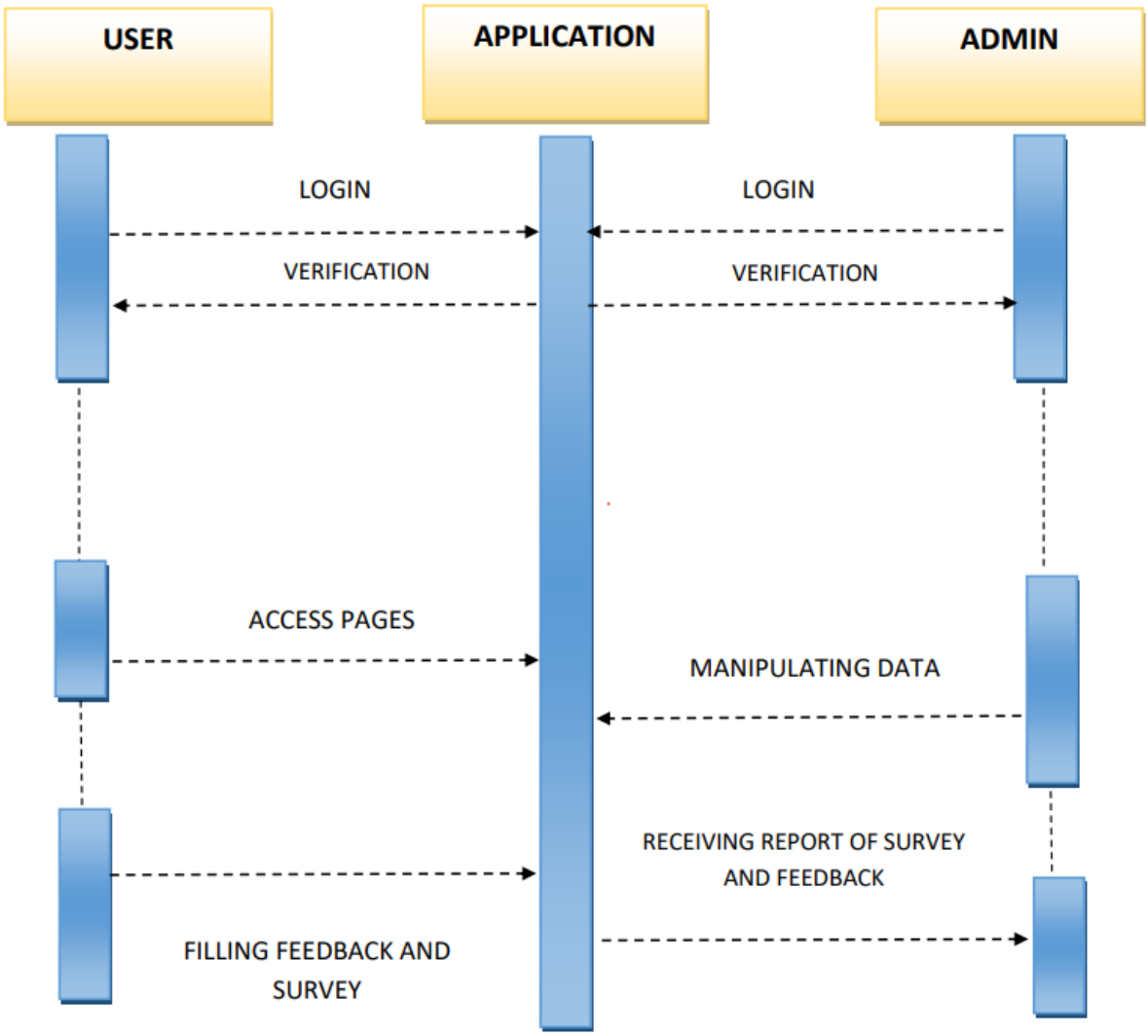
| Data Element | Data Item | Data Type | Description |
|---|---|---|---|
| **Registration** | 1. User Name | Text | Unique User Id |
| | 2. Password | AlphaNumeric | Unique Password for each user |
| **Feedback** | 1. Name | Text | Name of the user |
| | 1. Phone no | Number | Contact number of the user |
| | 2. Age | Number | Current age of user |
| | 3. Is this application helpful? | Text | Yes/No answered by user |

| | | | |
|---|---|---|---|
| | 4. Any Suggestions? | Text | Comment by user |
| | 5. Address | Text | Current residential address of the user |
| | 6. Ratings | Number | 1 to 5 ratings by user |
| **Survey** | 1. Place | Text | Place of crop land |
| | 2. Crop yield last year | Number | Quantity of crop yield last year |
| | 3. MSP of your crop | Number | MSP of the crop last year |
| | 4. Land type | Text | Type of land and soil |
| | 5. Total crop land | Number | Total field area |
| | 6. Crop type | Text | Type of crop |
| | 7. Total labor used | Number | Number of labor used |
| | 8. Amount of fertilizers used | Number | Quantity of fertilizers used |
| | 9. Machine you have | Text | Types of machines used |
| | 10. Water facilities you have | Text | Water facilities using |
| | 11. Ever used another type of crop | Text | Ever used another type of crop |
| | 12. Years in farming field | Number | Experience in farming |
| | 13. Using any government facility | Text | Specifying government facility used |
| | 14. Budget for farming | Number | Budget set for farming |

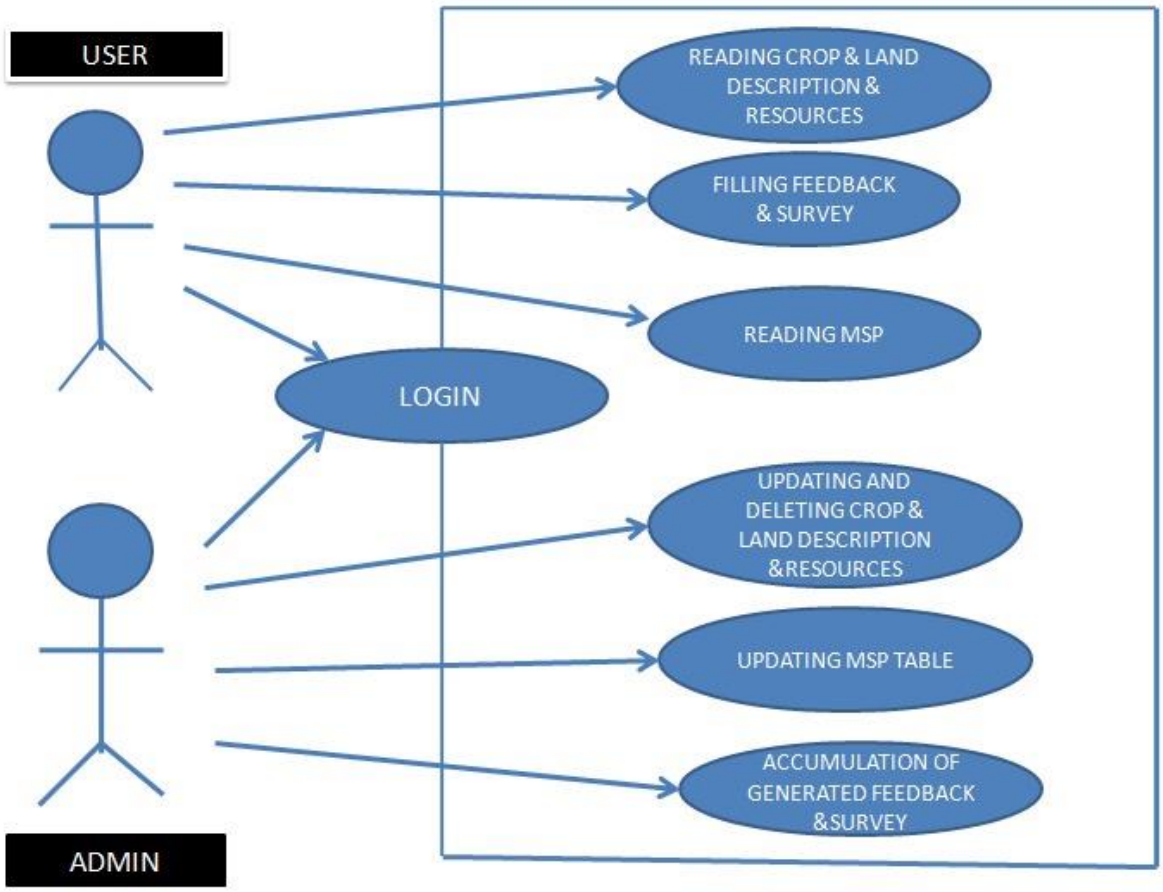
# SEQUENCE DAIGRAM

A sequence diagram is a graphical representation of the interactions among objects in a particular use case or scenario. It depicts the flow of events or messages exchanged between different components of a system, showing the order in which, these events occur.

It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.



# USECASE DIAGRAM



# USE CASES:

## 1.LOGIN

**1.1 Introduction:** This use case describes how a user logs in to the Crop Management System in Nutrition Distribution.

**1.2 Actors:** (i) User

(ii) Admin

**1.3 Pre Condition:** None

**1.4 Post Condition:** If usecase is successful the actor is logged into the system. If not, system state is unchanged.

**1.5 Basic Flow:** This use case starts when actor wishes to login to the Crop Management System in Nutrition Distribution

(i) System request that the actor enter his/her name and password.

(ii) The actor enters his/her name and password.

(iii) System validates name and password and if found correct

allow the actors login to the system.

**1.6 Alternative Flow:**

### 1.6.1 Invalid name and password

If in the basic flow the other enters an invalid username and password, the system displays an error message.The user can choose to either return to the beginning of the basic flow or can use option of forget password to change the

password using OTP.

**1.7 Special Requirements:** None

**1.8 Usecase Requirements:** None

## 2. READING CROP,LAND DESCRIPTION AND RESOURCES PAGE.

**2.1 Introduction:** This usecase includes allowing to read the Crop, land description and resource pages.

**2.2 Actors:** User

**2.3 Pre condition:** User must be logged into the system before this usecase begins.

**2.4 Post-condition:** If the use case is successful then user is able to access the pages.

**2.5 Basic Flow :** After successfully logged into the system the user can start browsing pages.

(i) User can read Crop Description page

(ii) User can read Land Description

(iii) User can access links of different pages provided in the Resources Page

**2.6 Alternative Flow:** None

**2.7 Special Requirements:** None

**2.8 UseCase Requirements:** None

## 3. FILLING FEEDBACK AND SURVEY

**3.1 Introduction:** This use case will allow actor to fill the details in feedback and survey form.

**3.2 Actors:** User

**3.3 Pre -condition:** User must be logged onto the system before the usecase begins.

**3.4 Post-condition:** If this use case successful then the user can fill, edit their feedback and survey forms otherwise the system state is unchanged.

**3.5 Basic Flow:** It starts when the user wants to fill feedback/ survey form (optional).

i) System request that the user enter his/her Name, Address, Phone Number, Age, Is the application helpful? any suggestions to improve it?, and Ratings in the feedback form.

ii) The actors enters his/her details mentioned

iii) Then the optional survey form will display.

iv) System will ask about Place, Crop Yield Last Year, MSP of your crop, Land Type, Total Crop Land, Crop Type, Total Labour used, Amount of Fertilizers used, Machines you have, Water facilities you have, Ever used another type of crops, Years in Farming field, Using any Government facility, Budget for farming.

v) The actor will enter the mentioned details.

**3.6 Alternative Flow:**

### 3.6.1 User do not wish to fill optional form

If user will skip the optional survey form then the user will

proceed to submit button directly and the details of feedback will be stored into the database without survey form.

**3.7 Special Requirements:** None

**3.8 Usecase Requirements:** None

## 4.READING MSP(Minimum Selling Price)

**4.1 Introduction:** This usecase includes allowing to read the MSP(Minimum Selling Price) page.

**4.2 Actors:** User

**4.3 Pre condition:** User must be logged into the system before this usecase begins.

**4.4 Post-condition:** If the use case is successful then user is able to access the page.

**4.5 Basic Flow:** After successfully logged into the system the user can start browsing pages.

(i) User can read MSP table.

**4.6 Alternative Flow:** None

**4.7 Special Requirements:** None

**4.8 UseCase Requirements:** None

## 5.UPDATING AND DELETING CROP,LAND AND RESOURCE PAGES

**5.1 Introduction:** Allow admin to maintain crop,land and resource pages.

**5.2 Actors:** Admin

**5.3 Pre-condition:** The admin must be logged in to the system.

**5.4 Post-condition:** If usecase is successful, admin can add/update/delete pages from the system. Otherwise the system state is unchanged.

**5.5 Basic Flow:** Starts when admin wishes to add/modify/update/delete pages data.

i) If admin wishes to add data then "add data" subflow will be executed

ii) If admin selects update data "update data: subflow will be executed.

iii) If admin selects delete data "delete data" subflow will be executed.

### 5.5.1 Add data

i. The system asks from Admin to enter crop description, land description and resource details.

ii. Then all changes will be saved in crop and land description data and resource info database.

### 5.5.2 Update data

i. The system asks to update/modify the data of the pages from admin.

ii. The admin makes the desired changes to the pages in the system.

iii. After changes the system updates the data of the pages.

### 5.5.3 Delete data

i. The system requests to specify the particular part to be deleted from the pages.

ii. The system prompts the admin to confirm the deletion of the pages.

iii. The admin confirms the deletion.

iv. The system marks the part of page for the deletion.

**5.6 Alternate Flow:**

**5.6.1 Update Cancelled**

If in the update data subflow, the admin decides not to update the pages data, the update is cancelled and the basic flow is restarted at the begin.

**5.6.1 Deletion Cancelled**

If in the delete data subflow, the admin decides not to delete the pages data, the deletion is cancelled and the basic flow is restarted at the begin.

**5.7 Special Requirements:** None

**5.8 UseCase Requirements:** None

## 6.UPDATING MSP(Minimum Selling Price) TABLE:

**6.1 Introduction:** Allow admin to maintain MSP table.

**6.2 Actors:** Admin

**6.3 Pre-condition:** The admin must be logged in to the system.

**6.4 Post-condition:** If usecase is successful, admin can add/update rows or data from the system. Otherwise, the system state is unchanged.

**6.5 Basic Flow:** Starts when admin wishes to add/modify/update MSP table data.

i. If admin wishes to add record in the table then “add data” subflow will be executed.

ii. If admin selects update record “update data: subflow will be executed.

### 6.5.1 Add data

The system asks from Admin to enter MSP records.

Then all changes will be saved MSP table details database.

### 6.5.2 Update data

i. The system asks to update/modify the data of the pages from admin.

ii. The admin makes the desired changes to the table in the system.

iii. After changes the system updates the records of the MSP table.

## 6.6 Alternate Flow:

### 6.6.1 Update Cancelled

If in the update data subflow, the admin decides not to update the records, the update is cancelled and the basic flow is restarted at the begin.

**6.7 Special Requirements:** None

**6.8 UseCase Requirements:** None

## 7. ACCUMULATION OF GENERATED FEEDBACK AND SURVEY:

**7.1 Introduction:** Allow admin to collect feedback from users and generated report of survey form.

**7.2 Actors:** Admin

**7.3 Pre-condition:** The admin must be logged in to the system.

**7.4 Post-condition:** If usecase is successful, admin can collect the feedbacks from the users and generated report of conducted survey. Otherwise, the system state is unchanged.

**7.5 Basic Flow:** Starts when admin desires to collect and maintain feedback form survey reports of the users.

i. If the feedback and survey data is available in the database then admin can access and go through that data to maintain application.

ii. Otherwise, the system state will be unchanged.

**7.6 Alternate Flow:** None

**7.7 Special Requirements:** None

**7.8 UseCase Requirements:** None

# FUNCTION POINT:

Function Point Analysis (FPA) is a software metric used to measure the size and complexity of a software system based on the functions it provides. It is a technique used to quantify the functionality of a software application or system.

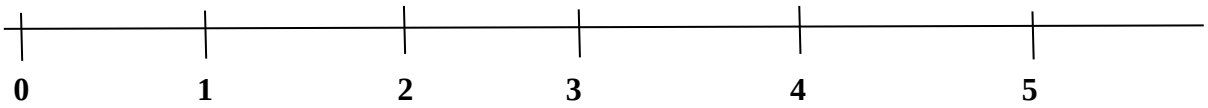
1. Number of external inputs(EI): Each Eis originates from a user is transmitted from another application and provides distinct application oriented data or control information.
2. Number of External Outputs(Eos): Each Eos is derived data within the application that provides information to the user.
3. Number of external inquiries(EQs): Defined as an online input that results in generation of some immediate software res on for of an online output.
4. Number of Internal Logical Files(ILFs): It is a logical grouping of data that resides within the apps boundary and is maintained via external inputs.
5. Number of external interface files(EIFs): It is a logical grouping of data that resides external to the application but provides data that may be of use to the application.

| | | |
|---|---|---|
| **EIs** | **4** | **Filling form by user + User Details +Admin details+ Manipulation by admin** |
| **EOs** | 1 | Report Generate |
| **ILFs** | 3 | Resource Info + MSP Table Details + Crop and Land Data |
| **EIFs** | 0 | NULL |
| **EQs** | 0 | NULL |

| Information Domain values | Count | Simple | Average | Complex | Function Point |
|---|---|---|---|---|---|
| EIs | 4 | 3 | 4 | 6 | 4*4 = 16 |
| EOs | 1 | 4 | 5 | 7 | 1*5 = 5 |
| EQs | 0 | 7 | 10 | 15 | 0*10 = 0 |
| ILFs | 3 | 5 | 7 | 10 | 3*7 = 21 |
| ELFs | 0 | 3 | 4 | 6 | 0*4 = 0 |

Count total=> 16+5+0+21+0 = 42

RATING EACH FACTOR ON A SCALE OF 0 TO 5:




0 -> No Influence, 1-> Incidental, 2-> Moderate, 3-> Average, 4-> Significant, 5-> Essential

| S.No. | Factors | Rating |
|---|---|---|
| 1. | Does the system require reliable backup and recovery? | 5 |
| 2. | Are specialized data communications required to transfer information to or from the application? | 5 |
| 3. | Are there distributed processing functions? | 1 |
| 4. | Is performance critical? | 2 |
| 5. | Will the system run in an existing, heavily utilized operational environment? | 3 |
| 6. | Does the system require online data entry? | 5 |
| 7. | Does the online data entry require the input transaction to be built over multiple screens or operations? | 3 |
| 8. | Are the ILFs updated online? | 4 |
| 9. | Are the inputs, outputs, files, or inquiries complex? | 1 |

| | | |
|---|---|---|
| 10. | Is the internal processing complex? | 3 |
| 11. | Is the code designed to be reusable? | 5 |
| 12. | Are conversion and installation included in the design? | 1 |
| 13. | Is the system designed for multiple installations in different organizations? | 1 |
| 14. | Is the application designed to facilitate change and ease of use by the user? | 2 |

Value of Adjustment Factor(∑Fi) = 41.

**Function Point(FP) = Count Total* [0.65+(0.01*∑F)]**

= 42*[0.65+(0.01*41)]= 42*[0.65+0.41]=42*1.06

= 44.52

**Effort Estimation:**

Effort estimation is predicting time and resources needed to complete a software project.

**Effort= -37+(0.96*FP)**

= -37+(0.96*44.52) = -37+42.74

=5.74

**Total Time = 40% of effort**

= 40/100 * 5.74= 2.296months

= 2.296*30 days= 68.88days ➔ 69days.

**Cost Estimation**

Total cost for the project = Total effort * Labour Rate

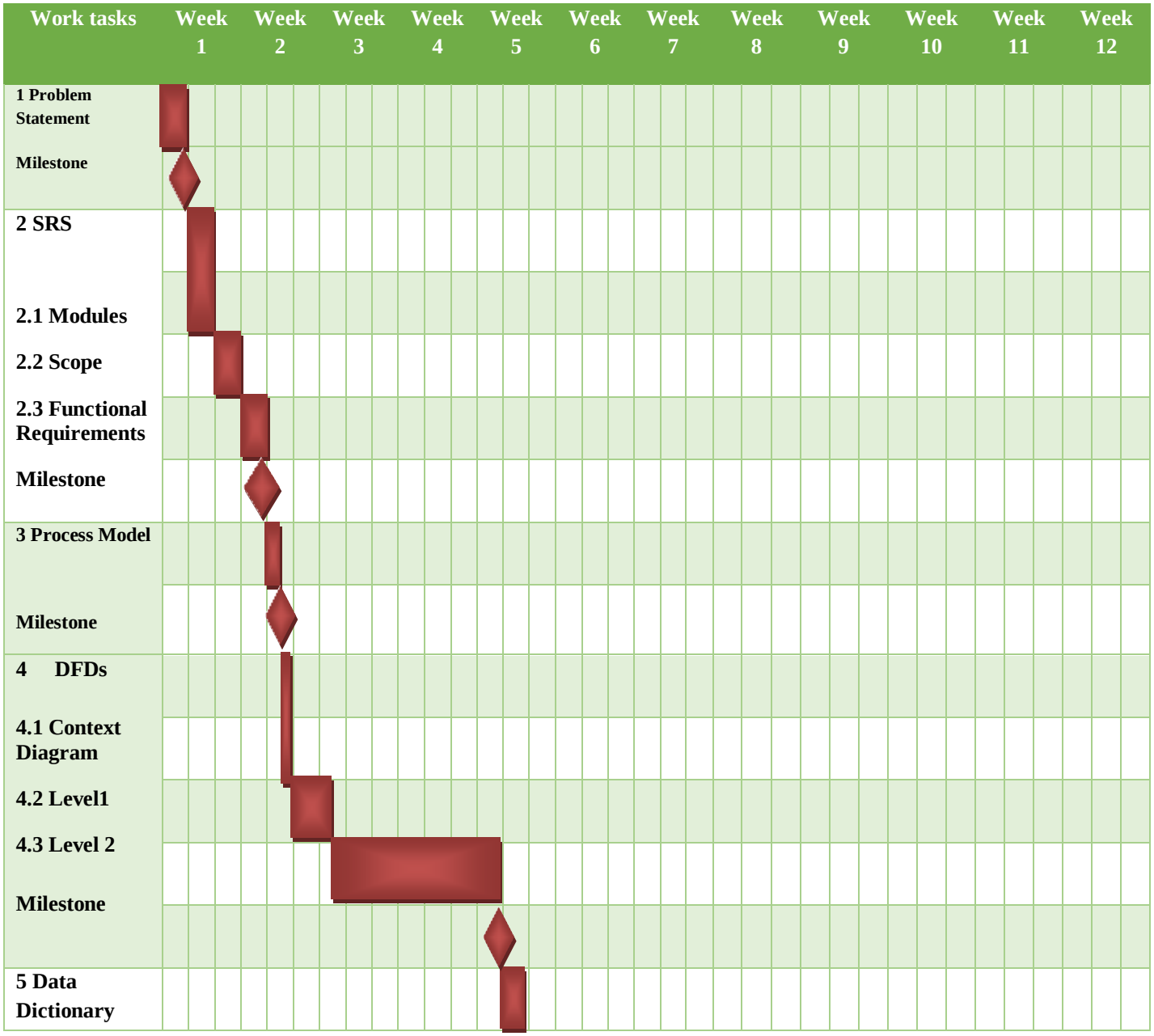
= 5.74*51=292.74

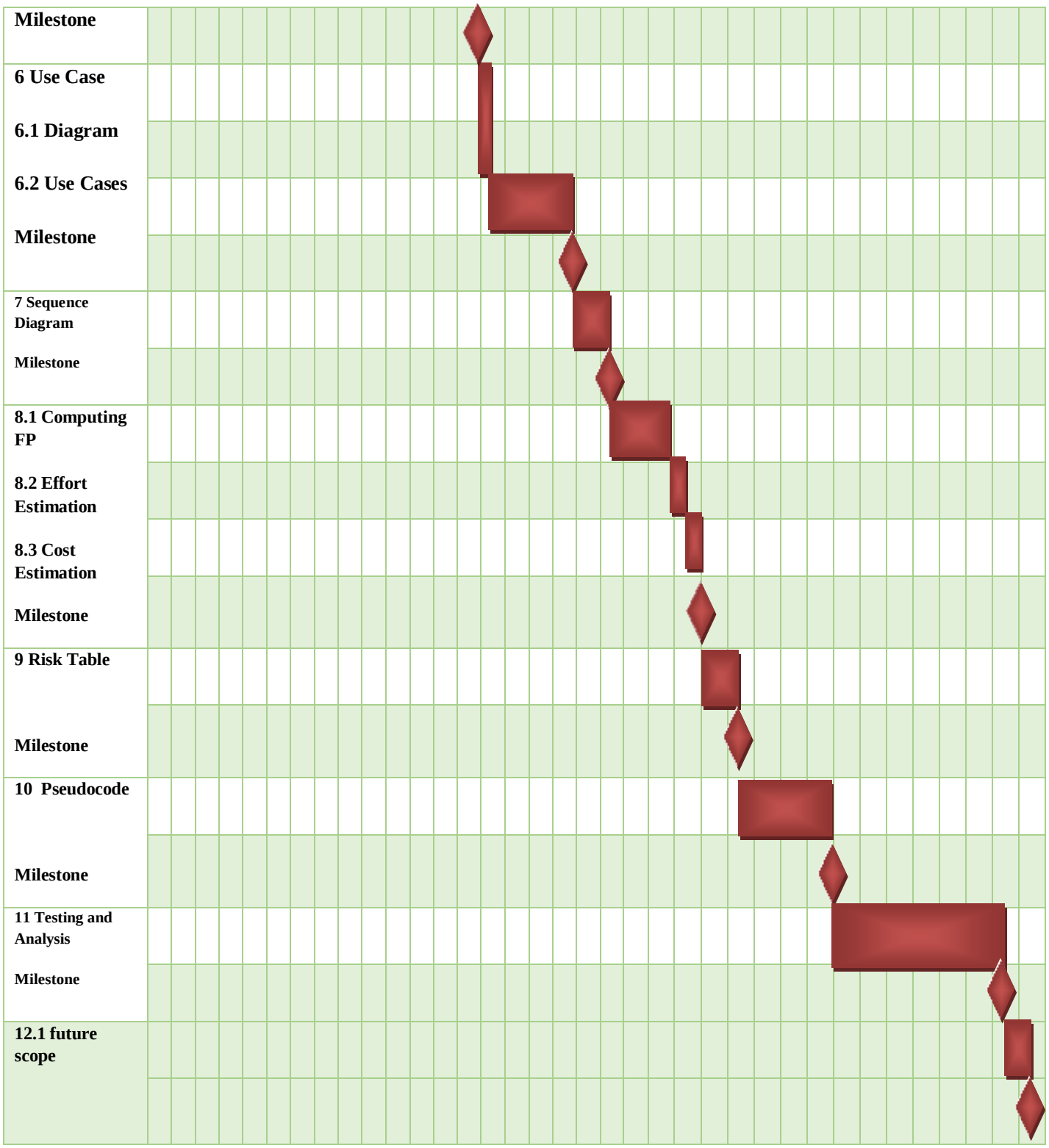
# TIMELINE CHART

A timeline chart is a valuable tool for managing projects as it allows project managers to track progress, identify potential issues, and communicate the project schedule to stakeholders.

It is useful for visualizing project timelines, milestones and deadlines.

Timeline is a chart that depicts how a set of resources are used over time.






- We are working 6 days a week for the project and one day off.

# RISK TABLE

Risk refers to the potential for an event or situation to have a negative impact on project objectives or outcomes. Risks can arise from a wide range of factors, including technical issues, resource constraints, organizational culture, external factors, and human error.

A risk table, also known as a risk matrix or a probability-impact matrix, is a tool used in risk management to assess and prioritize potential risks based on their probability and impact.

Risks:

1. Glitch in Software -due to virus application not working properly/server down.
2. Hardware failure at user side –defect in laptop/phone/computer.
3. Hardware failure at admin side-defect in laptop/phone/computer.
4. Finance problem-Improper budget.
5. Time problem-Improper time management.
6. Staffing limitation-not enough number of employee.
7. Natural calamities-natural disasters affecting the network.
8. Human error-mistake by admin /user.
9. Communication breakdown -Improper communication between the members of organization.

| Risk | Probability | Impact | Risk level |
|---|---|---|---|
| **Glitch in Software** | Low | High | Medium |
| **Hardware failure at user side** | High | High | High |
| **Hardware failure at admin side** | Medium | High | Medium-high |

| | | | |
|---|---|---|---|
| **Finance problem** | Low | Medium | Low-medium |
| **Time problem** | Low | High | Medium |
| **Staffing limitation** | Low | Low | Low |
| **Natural calamities** | Medium | High | Medium-high |
| **Human error** | Medium | High | Medium-high |
| **Communication breakdown** | Low | Medium | Low-medium |

# PSEUDOCODE

Pseudocode is a detailed yet readable description of what a computer program or algorithm must do, expressed in a formally-styled natural language rather than in a programming language. Pseudocode is sometimes used as a detailed step in the process of developing a program.

We are going to create pseudocode for **login module**.

1. START
2. If(login_user == true)
3. Name = Prompt Username.
4. Password = Prompt Password.
5. If(Name==User details && password== User details)
6. Valid user (verified)
7. Else

   Invalid user(non-verified)
8. Else

   newName = Prompt username
9. newPassword = Prompt password
10. PRINT(“Details saved”)
11. Singup successful
12. Print(“Start Login”)
13. END

# TESTING

Testing in software engineering refers to the process of evaluating a software application or system to identify any defects, errors, or bugs that may affect its performance, functionality, or user experience.

It is an essential part of the software development lifecycle that helps ensure the quality and reliability of the software being developed. It involves running the software under different scenarios and conditions to ensure that it works as intended and meets the requirements of the user.

## CONTROL FLOW DIAGRAM

A control flow diagram is a graphical representation of the control flow or the sequence of execution of instructions or statements in a software program. It is a visual representation of the program's execution flow that helps to illustrate the structure and behavior of the program.

Control flow diagrams are used to model and visualize the control flow of a program, including the decision points, loops, and function calls. The diagrams typically use a combination of symbols and arrows to represent the different elements of the program and the relationships between them.

The symbols used in a control flow diagram may include:

Start and end symbols, which represent the beginning and end of the program

Decision symbols, which represent conditional statements such as if-else or switch-case

Loop symbols, which represent the repetition of a block of code using for or while loops

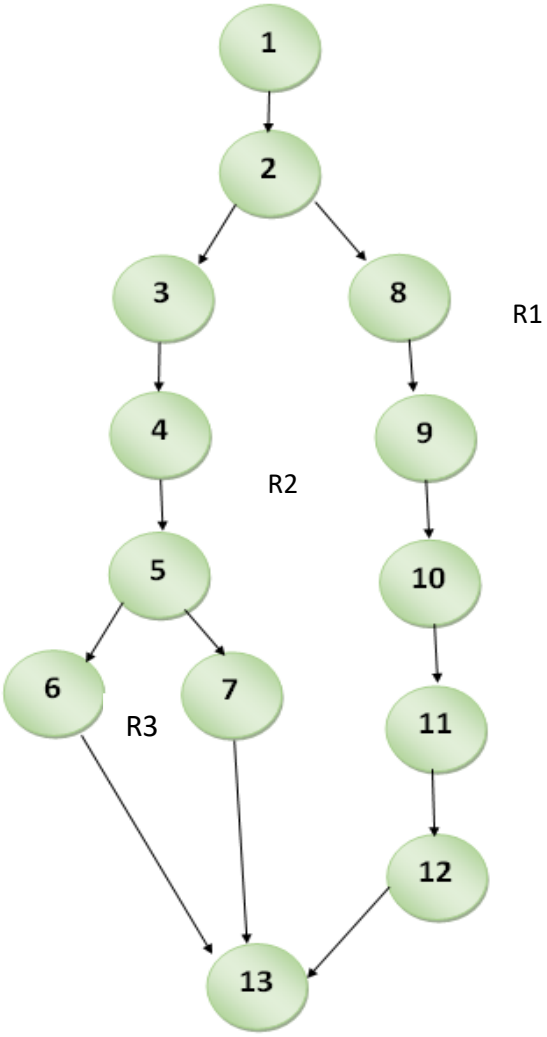
Function call symbols, which represent the invocation of a function or

subroutine

Input/output symbols, which represent the input and output of data to and from the program

The arrows in a control flow diagram represent the flow of execution between the different elements of the program. They indicate the sequence in which the statements are executed and the possible paths through the program.

The control flow graph of pseudocode of login module is :



# CYCLOMATIC COMPLEXITY

Cyclomatic complexity is a software metric that provides a quantitative measure of the logical complexity of a program.

When used in the context of the basis path testing method, the value computed for cyclomatic complexity defines the number of independent paths in the basis set of a program and provides you with an upper bound for the number of tests that must be conducted to ensure that all statements have been executed at least once.

Complexity is computed in one of three ways:

1. The number of regions of the flow graph corresponds to the cyclomatic complexity.

2. Cyclomatic complexity V( G) for a flow graph G is defined as

V( G) = E - N + 2 where E is the number of flow graph edges and N is the number of

flow graph nodes.

3. Cyclomatic complexity V( G) for a flow graph G is also defined as

V( G) = P + 1 where P is the number of predicate nodes contained in the flow graph

Here ,

- **No.of edges are 14**

  **No. of nodes are 13**

  V(G) = Number of edges(E) – Number of nodes(N) + 2

  = 14 – 13 +2

  = 3

- **No. of predicate nodes (P)=2(node 2 and node 5)**

  V(G)=P+1=2+1=3

- Number of regions =V(G)=3

  - **Region 1 :** 1-2-3-4-5-6-13-12-11-10-9-8-1
  - **Region 2 :** 2-3-4-5-7-13-12-11-10-9-8-2
  - **Region 3 :** 5-6-13-7-5

# LINEARLY INDEPENDENT PATHS

- A linearly independent path refers to a path through the code of a program that provides unique and independent information about the behavior of the program.
- More specifically, it is a path in the control flow graph of a program that is not redundant and cannot be expressed as a combination of other paths. Each linearly independent path represents a distinct set of inputs and outputs that can be used to test the program.
- Linearly independent paths are important in software testing because they can help ensure that a program is behaving correctly under a wide range of conditions. By testing each linearly independent path, developers can be more confident that their code is working as intended and that there are no hidden bugs or errors.

Since cyclomatic complexity is 3 ,therefore there will be 3 independent path in our control flow graph.

**Path 1:** 1-2-3-4-5-6-13

**Path 2:** 1-2-3-4-5-7-13

**Path 3:** 1-2-8-9-10-11-12-13

# FUTURE SCOPE

It is sure, and undoubtedly the GDP of all the developing and developed countries is influenced by the agricultural sectors. And, it is the integral component of the annual plan of any state or central government. Advances in machinery can expanded the scale, speed, and productivity of farm equipment, leading to more efficient cultivation of more land. Seed, irrigation, and fertilizers can also vastly improve, helping farmers increase yields. Now, agriculture is in the early days of yet another revolution, at the heart of which lie data and connectivity. Analytics, connected applications, and other emerging technologies could further increase yields, improve the efficiency of water and other inputs, and build sustainability and resilience across crop cultivation and nutrient quality.

Without a solid connectivity infrastructure, however, none of this is possible. If connectivity is implemented successfully in agriculture, the industry could tack on $500 billion in additional value to the global gross domestic product by 2030, according to the research.

It is one of just seven sectors that can contribute $2 trillion to $3 trillion in additional value to global GDP over the next decade.

Demand for food is growing at the same time the supply side faces constraints in land and farming inputs. The world's population is on track to reach 9.7 billion by 2050, requiring a corresponding 70 percent increase in calories available for consumption, even as the cost of the inputs needed to generate those calories is rising.

By 2030, the water supply will fall 40 percent short of meeting global water needs, and rising energy, labor, and nutrient costs are already pressuring profit margins. About one-quarter of arable land is degraded and needs significant restoration before it can again sustain crops at scale. And then there are increasing environmental pressures, such as climate change and the economic impact of catastrophic weather events, and social pressures, including the push for more ethical and sustainable farm practices, such as reduced use of chemicals and water.

This project handles all the problems related to agricultural field that a farmer could face along with increasing the quality of food for the common people like us.It will reach out to all kind of people in our society and contribute in a very positive manner.

Farmer if provided with the right information about which crop to use and how to increase its yield can contribute in total GDP a lot. That's why we provide benefit not only to the farmer for the yield but also the GDP and the quality of crops for everybody.

# REFERENCES


- Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach.8th edition. McGraw-Hill.
- Aggarwal, K. K., & Singh, Y. (2007). Software Engineering.3rd edition. New Age International Publishers.
- https://www.geeksforgeeks.org/
- https://youtube.com/@GateSmashers
- https://chat.openai.com/chat