Department of Computer Engineering

Academic Term: First Term 2023-24

Class: T.E /Computer Sem – V / Software Engineering

Practical No:	**6**
Title:	**Data flow analysis of the Project**
Date of Performance:	**13/09/2023**
Roll No:	**9614**
Team Members:	**Mudabbir(9589),Muhammad(9588), Nathan(9597)**

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

Signature of the Teacher:

Lab Experiment 06

Experiment Name: Data Flow Analysis of the Project in Software Engineering

Objective: The objective of this lab experiment is to introduce students to Data Flow Analysis, a technique used in software engineering to understand the flow of data within a software system. Students will gain practical experience in analyzing the data flow of a sample software project, identifying data dependencies, and modeling data flow diagrams.

Introduction: Data Flow Analysis is a vital activity in software development, helping engineers comprehend how data moves through a system, aiding in identifying potential vulnerabilities, and ensuring data integrity.

Lab Experiment Overview:

1. Introduction to Data Flow Analysis: The lab session begins with an overview of Data Flow Analysis, its importance in software engineering, and its applications in ensuring data security and accuracy.
2. Defining the Sample Project: Students are provided with a sample software project, which includes the data elements, data stores, processes, and data flows.
3. Data Flow Diagrams: Students learn how to construct Data Flow Diagrams (DFDs) to visualize the data flow in the software system. They understand the symbols used in DFDs, such as circles for processes, arrows for data flows, and rectangles for data stores.
4. Identifying Data Dependencies: Students analyze the sample project and identify the data dependencies between various components. They determine how data is generated, processed, and stored in the system.
5. Constructing Data Flow Diagrams: Using the information gathered, students create Data Flow Diagrams that represent the data flow within the software system. They include both high-level context diagrams and detailed level-0 and level-1 diagrams.
6. Data Flow Analysis: Students analyze the constructed DFDs to identify potential bottlenecks, inefficiencies, and security vulnerabilities related to data flow.
7. Conclusion and Reflection: Students discuss the significance of Data Flow Analysis in software development and reflect on their experience in constructing and analyzing Data Flow Diagrams.

Learning Outcomes: By the end of this lab experiment, students are expected to:

· Understand the concept of Data Flow Analysis and its importance in software engineering.

· Gain practical experience in constructing Data Flow Diagrams to represent data flow in a software system.

· Learn to identify data dependencies and relationships within the software components. · Develop analytical skills to analyze Data Flow Diagrams for potential issues and vulnerabilities. · Appreciate the role of Data Flow Analysis in ensuring data integrity,

security, and efficiency.

Pre-Lab Preparations: Before the lab session, students should familiarize themselves with Data Flow Analysis concepts and the symbols used in Data Flow Diagrams. They should review data dependencies and data flow modeling in software systems.

Materials and Resources:

· Project brief and details for the sample software project

· Whiteboard or projector for constructing Data Flow Diagrams

· Drawing tools or software for creating the diagrams

Sample Scenario: Crop Recommendation Feature

1. User Input:
The user selects soil type, climate preferences, and preferred crop types.

2. Data Processing:
Validation and organization of user input.

3. Database Interaction:
Query internal database for soil types, climate conditions, and historical crop data.

4. External Data Sources:
Retrieve real-time weather data from an external weather API.

5. Processing and Analysis:
An algorithm analyzes user preferences, historical crop performance, and current weather data.

6. User Output:
Display a list of recommended crops to the user.

7. Feedback Loop:
Users provide feedback, influencing future recommendations.

8. Notifications and Alerts:
Generate alerts based on significant changes in weather conditions.

Calculations:

User input involves 5 soil types, 3 climate preferences, and 10 crop types.
The internal database contains information on 20 crops.
External weather API provides data on temperature, humidity, and precipitation.
The processing and analysis algorithm complexity is 10 units.
The system generates a recommendation list of 5 crops.
Total Combinations = Soil Types × Climate Preferences × Crop Types

Total Combinations=5×3×10=150

Total Complexity=Internal Database+External Data API+Algorithm
Total Complexity=20+3+10=33

Person-Months=Total Complexity×Conversion Factor

Person-Months=33×20=660

WHITEBOARD DIAGRAMS

1. COMPONENT DIAGRAM:

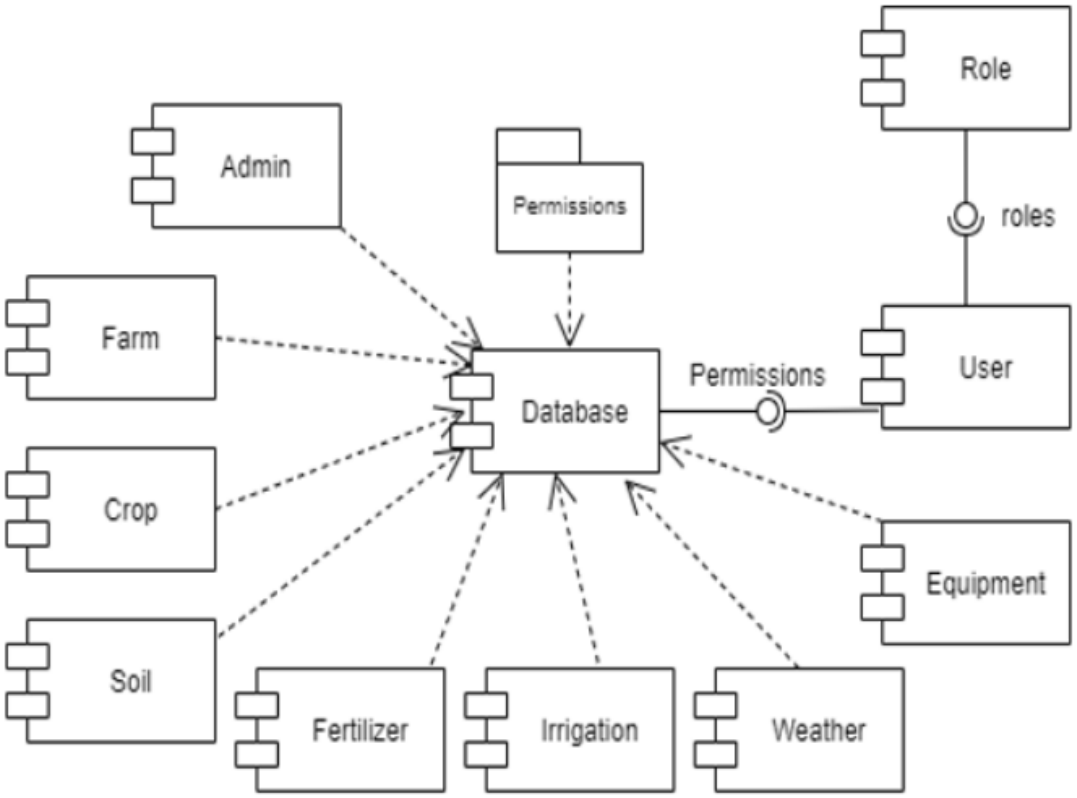

Figure 4: Component Diagram

2. ZERO LEVEL DFD

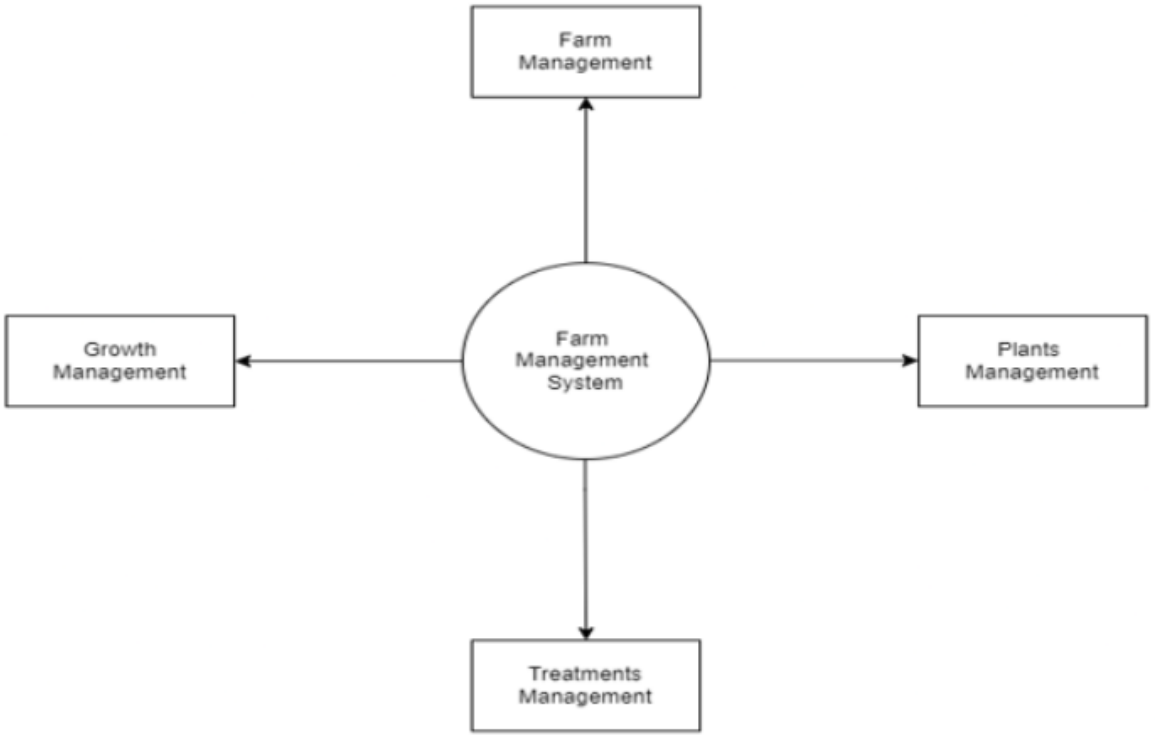

Figure 1: Zero Level DFD

3. ONE LEVEL DFD:

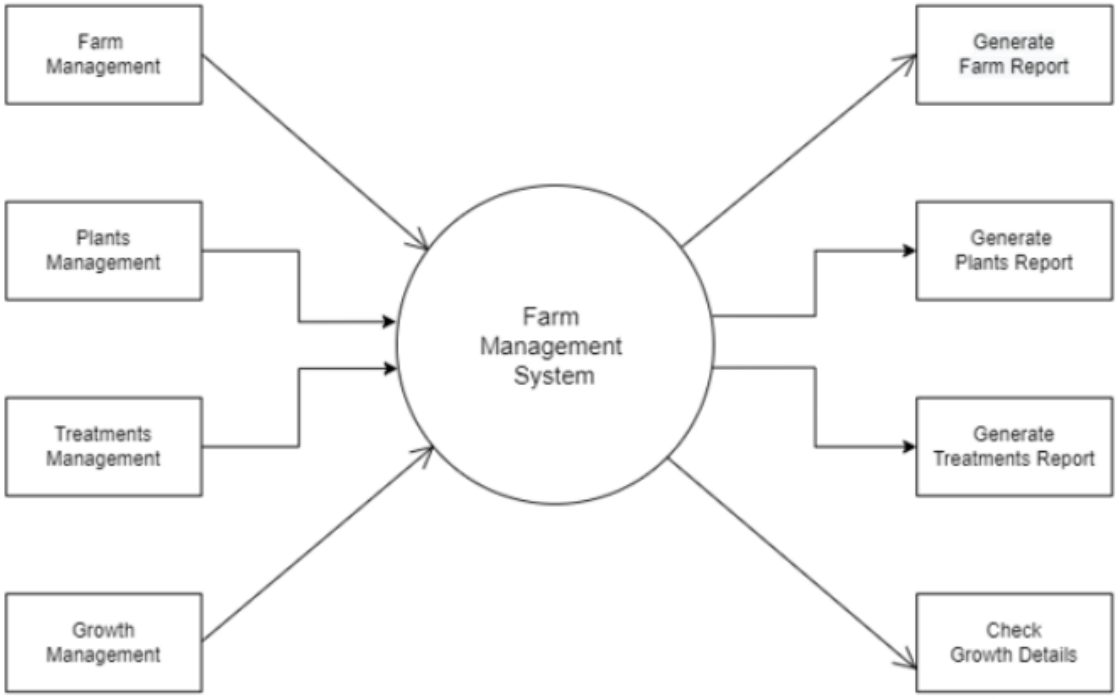

Figure 2: One Level DFD

4. TWO LEVEL DFD:

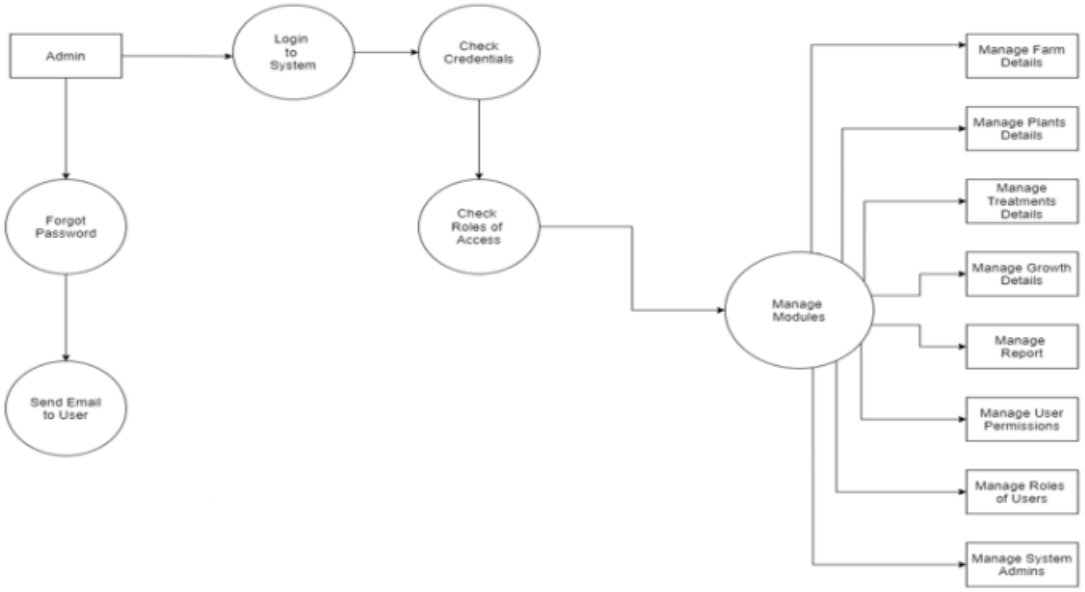

Figure 3: Two Level DFD

Conclusion: The lab experiment on Data Flow Analysis of a software project equips students with essential skills in understanding data flow within a system. By constructing and analyzing Data Flow Diagrams, students gain insights into how data is processed, stored, and exchanged, enabling them to identify potential issues and security concerns. The practical experience in Data Flow Analysis enhances their ability to design efficient and

secure software systems that ensure data integrity and meet user requirements. The lab experiment encourages students to apply Data Flow Analysis in real
world software development projects, promoting better data management and system design practices.