# ASSIGNMENT NO 1

**Problem Statement** :
To develop the problem under consideration and justify feasibility using concept of knowledge canvas and IDEA matrix.

**Learning Objectives**:

- Understand the concept of IDEA matrix
- Understand the purpose of feasibility analysis

**Learning Outcomes**:

- Check feasibility of proposed system
- Add functionality to the system.

**IDEA Matrix :**

| I | D | E | A |
|---|---|---|---|
| Increase Accuracy | Drive Diagnosis | Educate Users | Accelerate processing |
| Improve Reliability | Deliver Performance | Evaluate Results | Associate Tools |
| Ignore Compatibility Risks | Decrease Complexity Issues | Eliminate Errors | Avoid Updation Errors |

- To increase accuracy and reliability, we are using existing pre-tested tools for utilities.
- Ignore compatibility risk by running the utilities in most basic configurations.
- Decrease compatibility issues by diagnosing the system.
- Deliver performance by optimising data structure
- Results are evaluated to eliminate errors
- To avoid updation error we are approaching the problem through dynamic configurations.

ASSIGNMENT NO 2

**Problem Statement** :
Project problem statement feasibility assessment using NP-Hard , NP-Complete or satisfiability issues using modern algebra and/or relevant mathematical models

**Learning Objectives**:
- Understand the concept of NP-Hard, NP-Complete

**Learning Outcomes**:
- Formulate proper mathematical justification for NP-Completeness of problem
- Show NP-Completeness of proposed system

**Mathematical Model** :
System Description:
Input I= {PTS, DTS, Packet no, Packet Size, Average Frame Rate}

Output ={ Percentage Jitter}

Funnctions={ f1, f2, f3 }

f1 = function to calculate durations
f2= function to calculate average duration and max frequency duration
f3= function to calculate percentage jitter in the video

PTS= { $PTS_i$ | 0<i<N and $PTS_i$ is Presentation Time Stamp}
DTS= { $DTS_i$ | 0<i<N and $DTS_i$ is Decode Time Stamp}
Packet No= { i | 0<i<N }
Packet Size= { Packet $Size_i$ | 0<i<N }
Average Frame Rate= Average of all frame rates

Duration = { $Duration_i$ | 0<i<N and $Duration_i$ is Time Duration between two timestamps }

$$Duration_i = PTS_{(i+1)} - PTS_i$$

For Finding out Time Stamp jitter Present in video,

$$Average\ Duration = \frac{\sum_{i=0}^{N-1} Durationi}{N}$$

$$Maximum\ Frequncy\ Duration = Mode\ of\ (\ Duration\ )$$

$$TimeStamp\ Jitter = Average\ Duration - Maximum\ Frequency\ Duration$$

$$Percentage\ Jitter = \frac{Average\ Duration - Maximum\ Frequency\ Duration}{Average\ Duration} * 100$$

Success case – Shows Percentage jitter.
Failure case – Video contains noise but percentage jitter comes out to be 0.

The number of possible code paths typically increases exponentially with the cyclomatic complexity of a method.Automated test generation tools for path coverage first need to determine which code paths are possible and which ones are not, and then need to be able to generate test inputs that cover all possible paths. Both steps are very time-intensive, and the accuracy of the results cannot really be guaranteed because the required analysis involves problems that are known to be NP-hard or even undecidable (like the infamous halting problem)

ASSIGNMENT NO 3

**Problem Statement** :

Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify objects, morphisms, overloading in functions (if any), and functional relations and any other dependencies (as per requirements).

**Learning Objectives**:

- Reducing the complexity of problem using divide and conquer.
- Making use of object oriented concepts.

**Learning Outcomes**:

- Design a distributed and parallel system using cloud
- Dividing the solution into objects in order to map it into real life entities.

**Functional Dependency**:

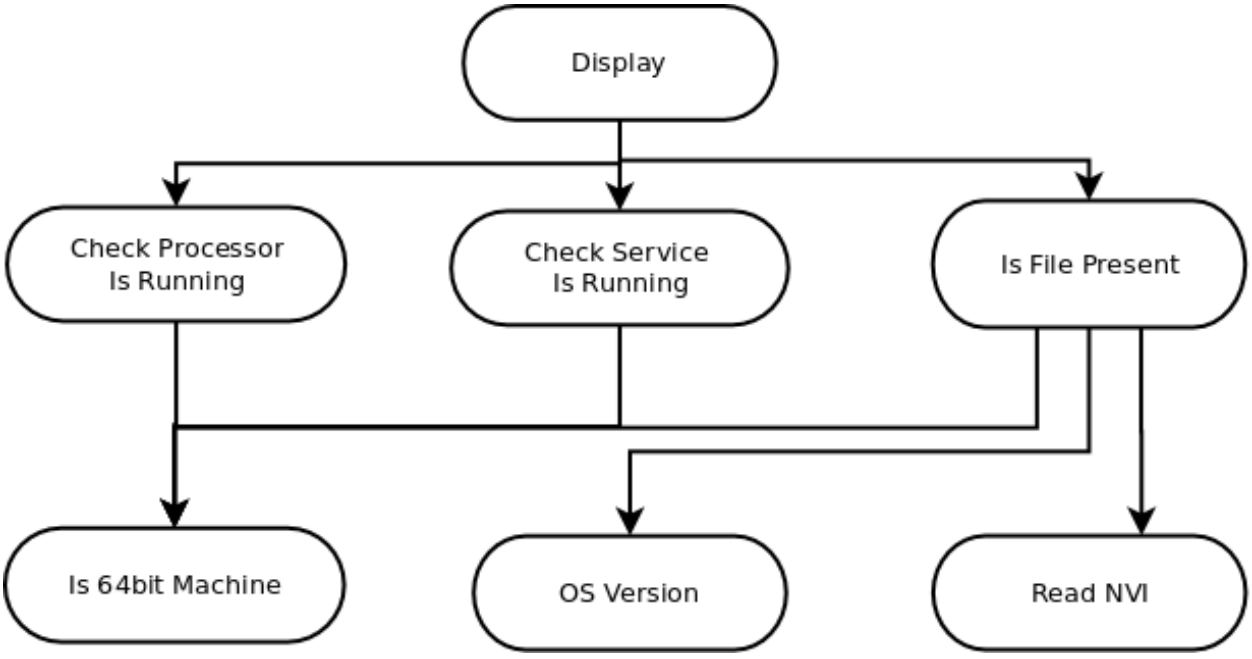


**Figure 1 Functional Dependency**

**Identification of Objects**

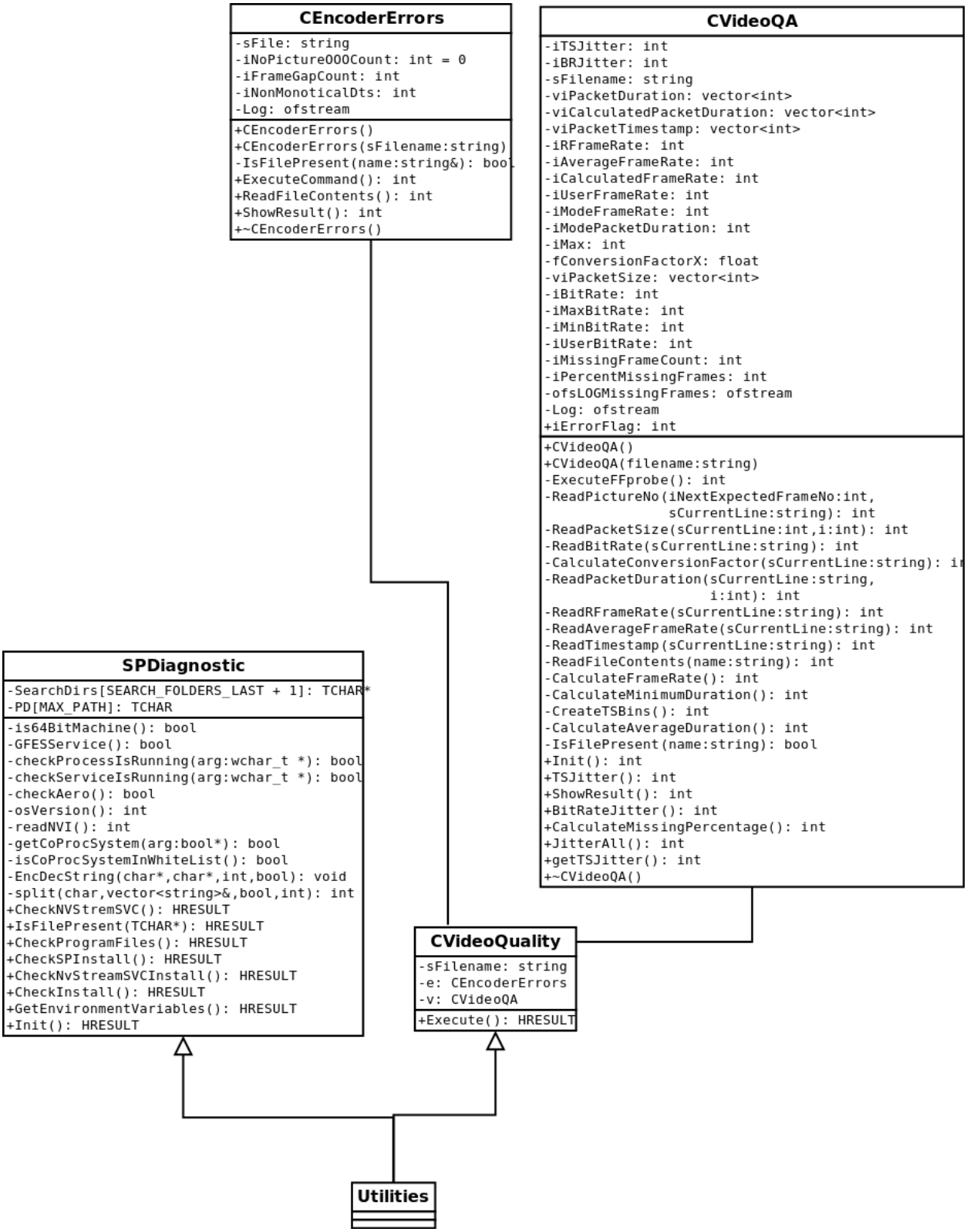


**Figure 2 Class Diagram**

## ASSIGNMENT NO - 04

PROBLEM STATEMENT :

Use of above to draw functional dependency graphs and relevant Software modeling methods, techniques including UML diagrams or other necessities using appropriate tools.

LEARNING OBJECTIVE :

- To modularize the project
- To have clear view of the project.

LEARNING OUTCOME :

- The modules are created.
- Designed UML Diagrams

ER – DIAGRAM :

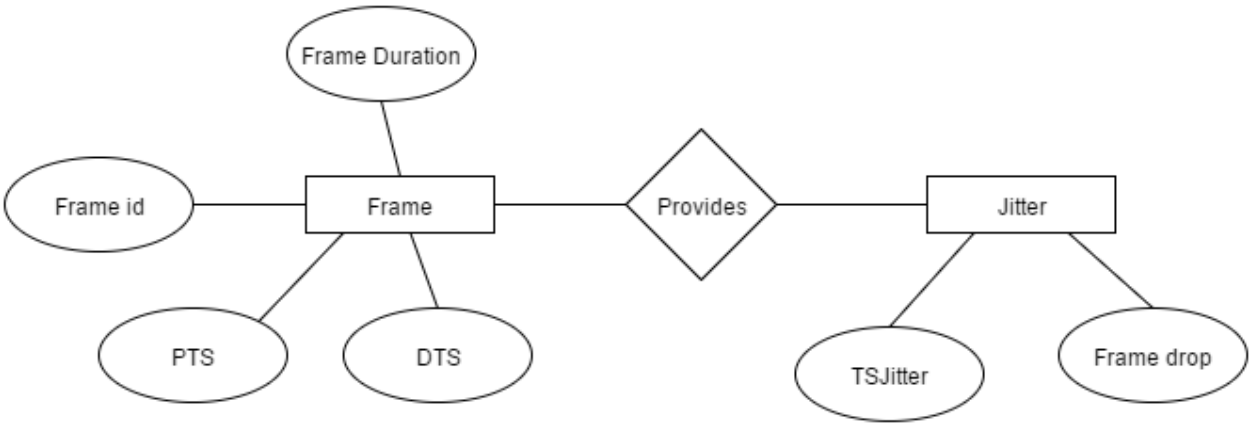


**Figure 1 ER Diagram**

CLASS DIAGRAM:

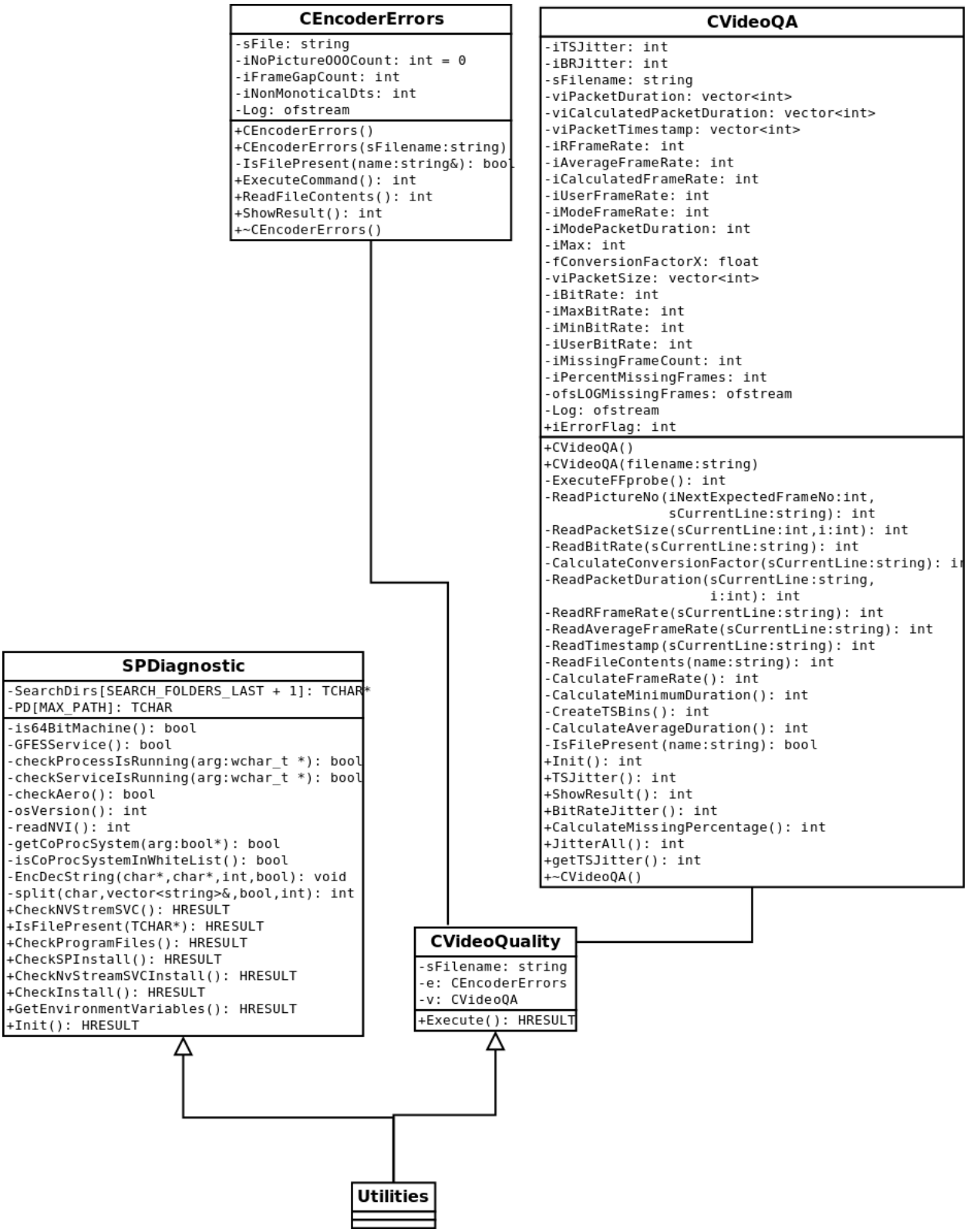


**Figure 2 Class Diagram**

FUNCTIONAL DIAGRAM

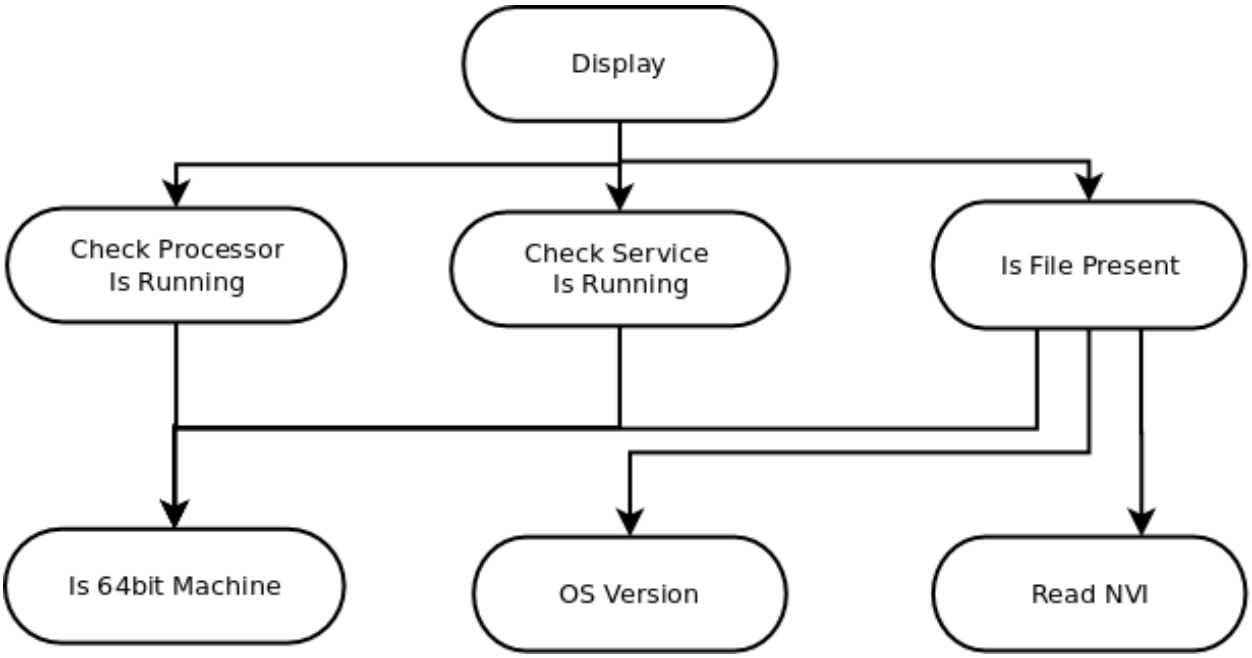


**Figure 3 Functional Diagram**

USE CASE ：

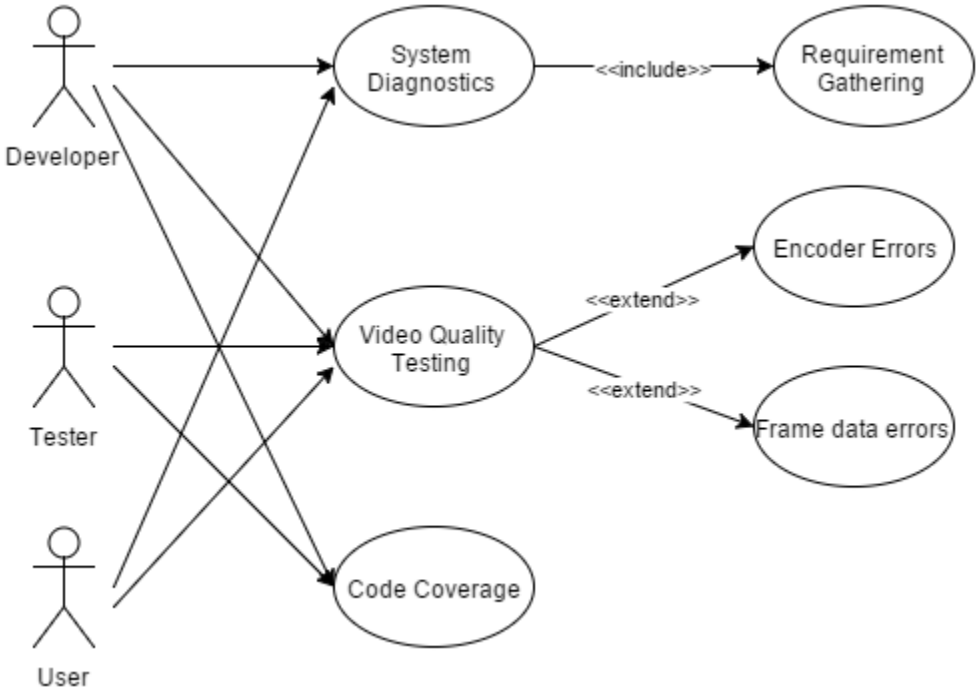


**Figure 4 UseCase**

STATE DIAGRAM :


**Figure 5 Project state diagram**

TASK NETWORK

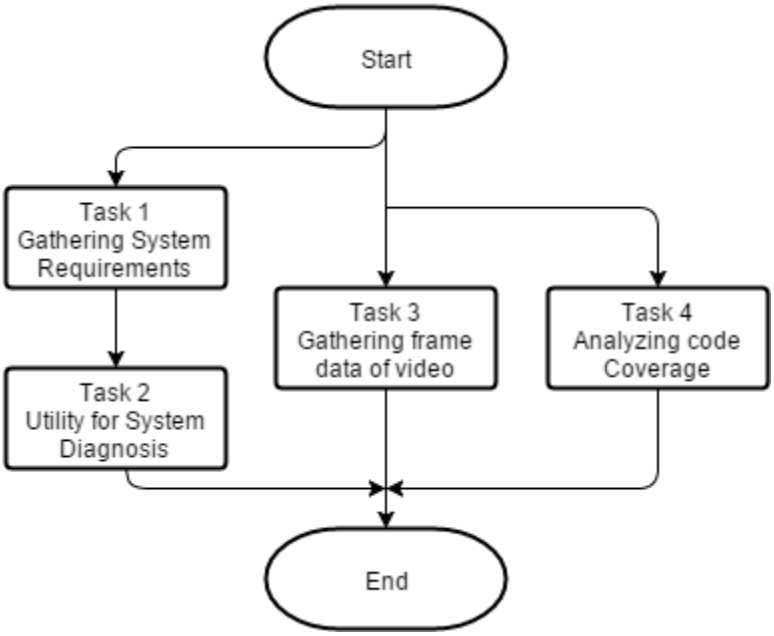


**Figure 6 Task Network**

PLAN

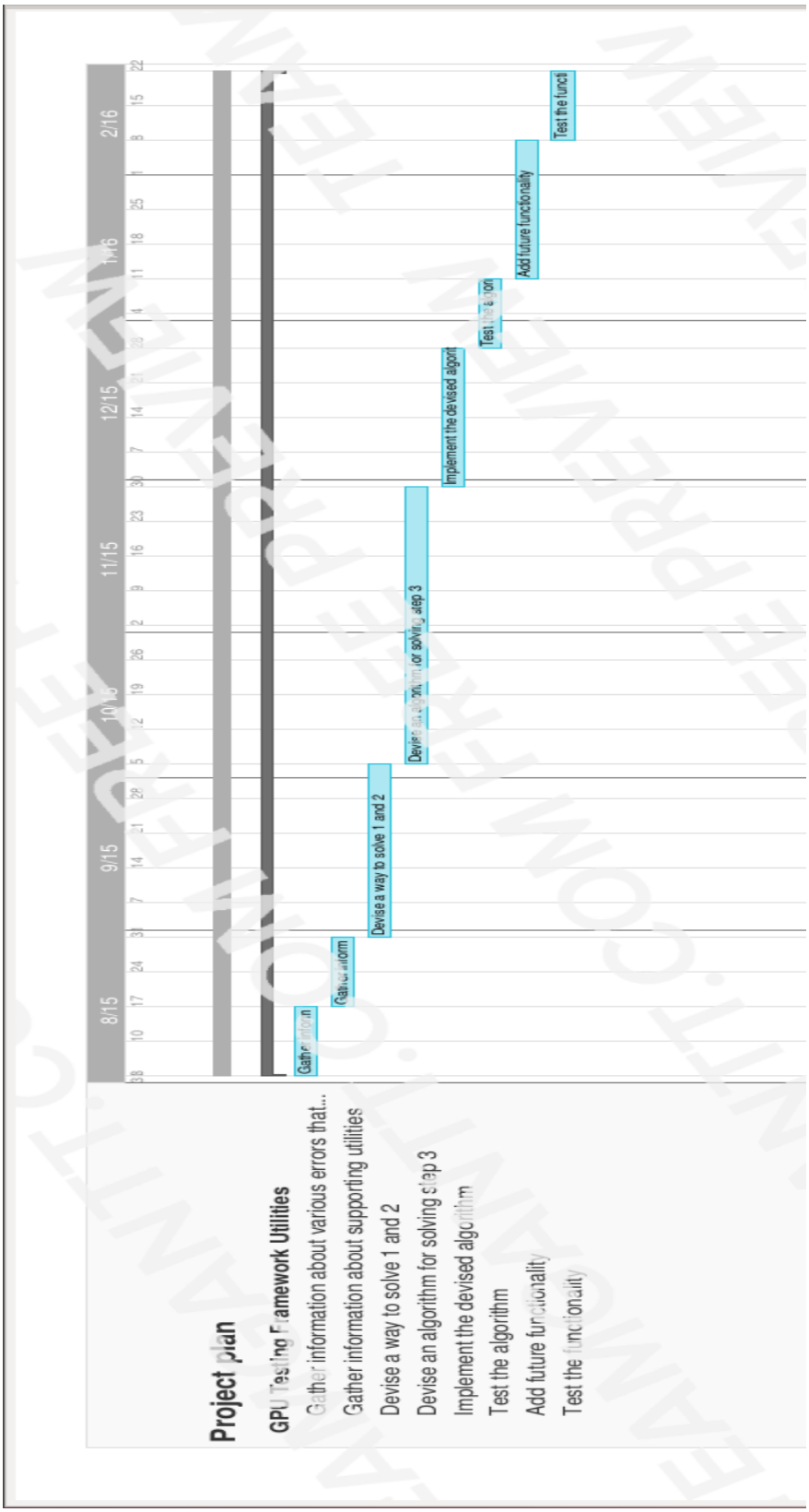


**Figure 7 Plan**

ASSIGNMENT NO 5

**Problem Statement** :
Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

**Learning Objectives**:

- Understand how testing is carried out.

**Learning Outcomes**:

- Test cases for proposed system requirements are generated

Function Principle Testing

| Steps | Description | Input Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|
| 1.1 | Corrupted video is given as input to quality testing tool | Sample.mp4 | Results should show amount of corruption in percentage. | Results showing amount of corruption in percentage. | Pass |
| 1.2 | Normal video is given as input. | Sample2.mp4 | Result should be 0% jitter | Result showing 0%jitter. | Pass |
| 1.3 | Merging and analysis of coverage file. | Coverage file for each test is given as input | 100% coverage. (Ideal result) | 70-80% coverage | Pass |