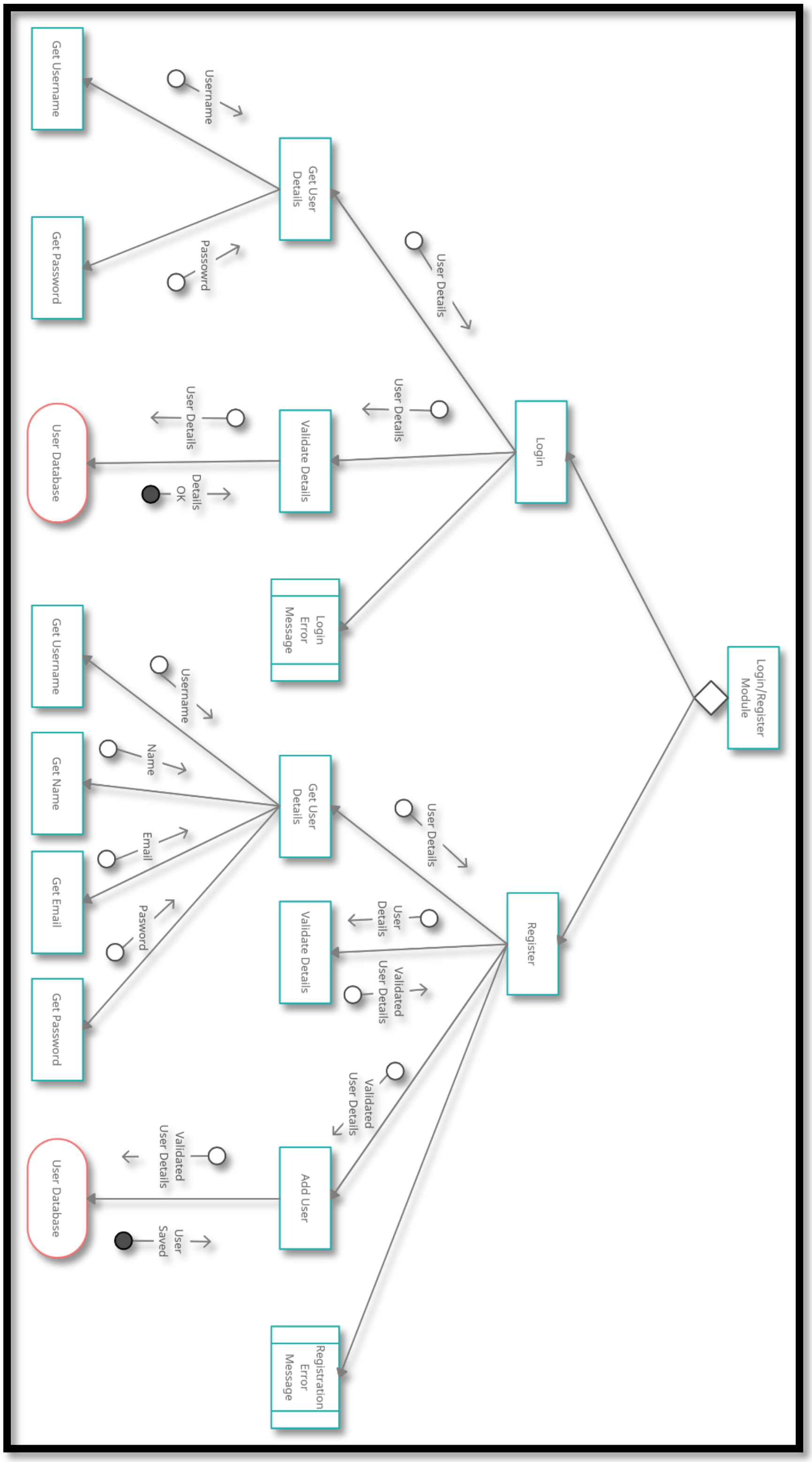
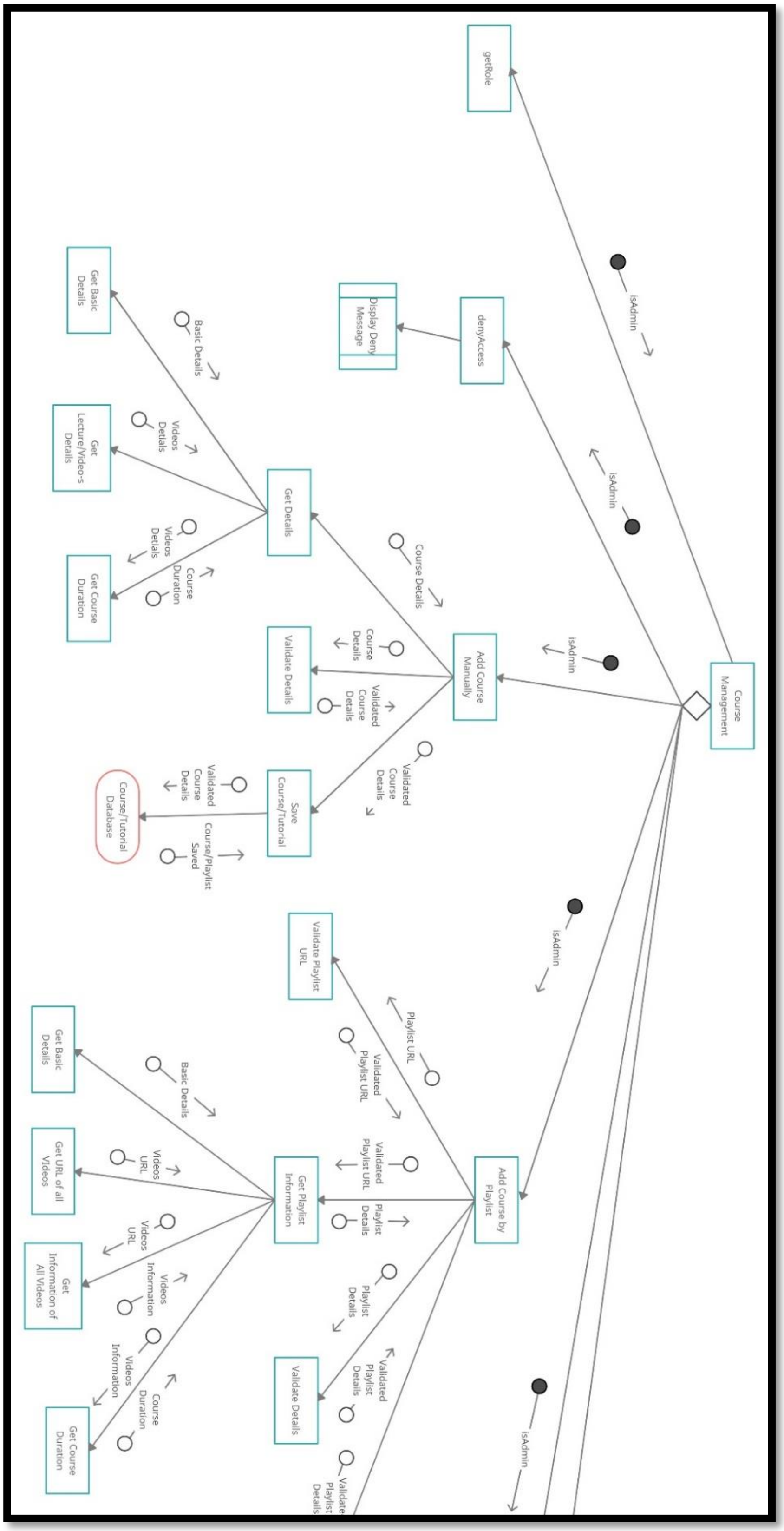
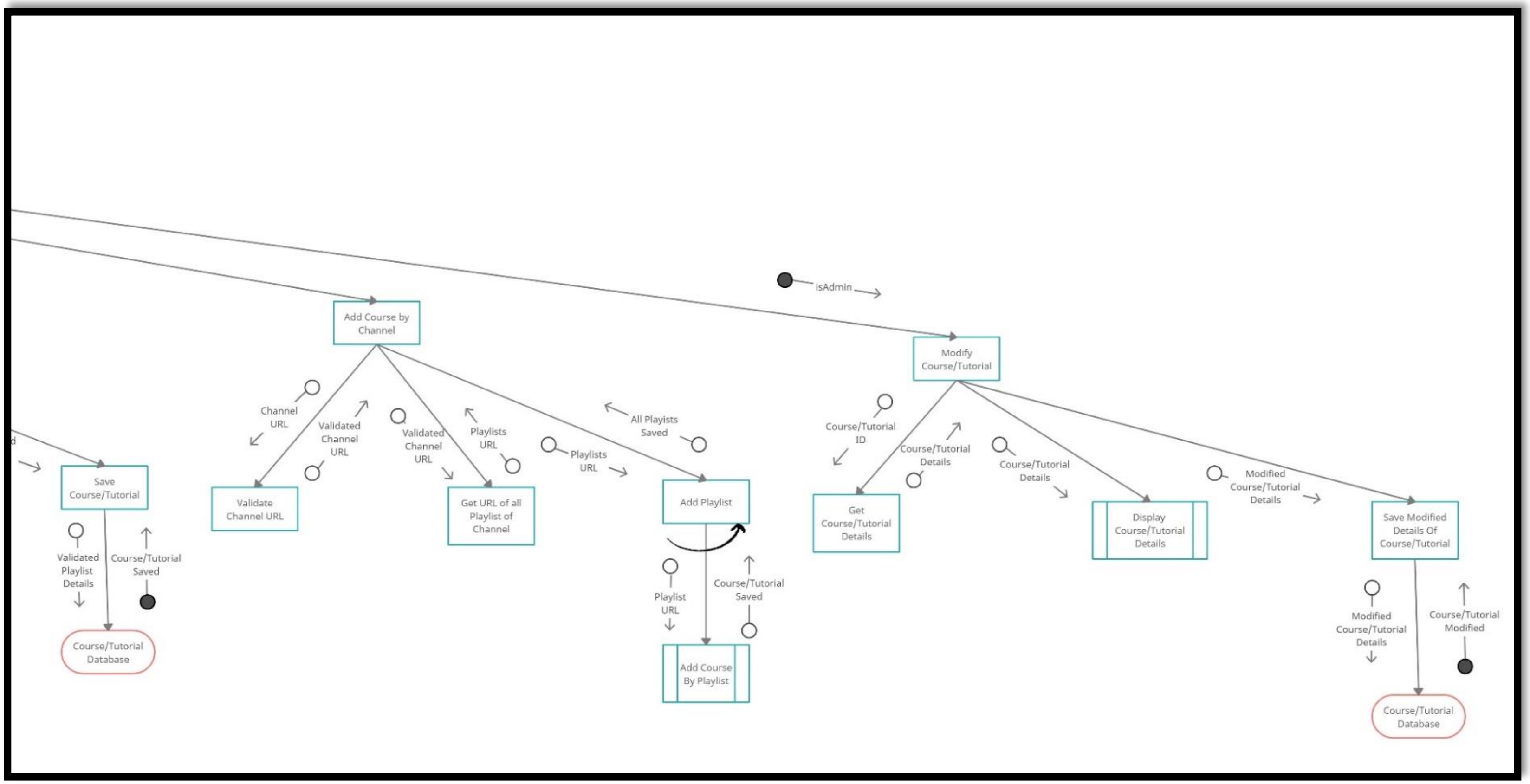
Structure Chart

Login Module:



Course Management Module:





isAdmin
Save Course/Tutorial
Validated Playlist Details
Course/Tutorial Saved
Course/Tutorial Database
Add Course by Channel
Channel URL
Validated Channel URL
Validate Channel URL
Validated Channel URL
Playlists URL
Get URL of all Playlist of Channel
Playlists URL
All Playists Saved
Add Playlist
Playlist URL
Course/Tutorial Saved
Add Course By Playlist
Modify Course/Tutorial
Course/Tutorial ID
Course/Tutorial Details
Get Course/Tutorial Details
Course/Tutorial Details
Display Course/Tutorial Details
Modified Course/Tutorial Details
Save Modified Details Of Course/Tutorial
Modified Course/Tutorial Details
Course/Tutorial Modified
Course/Tutorial Database

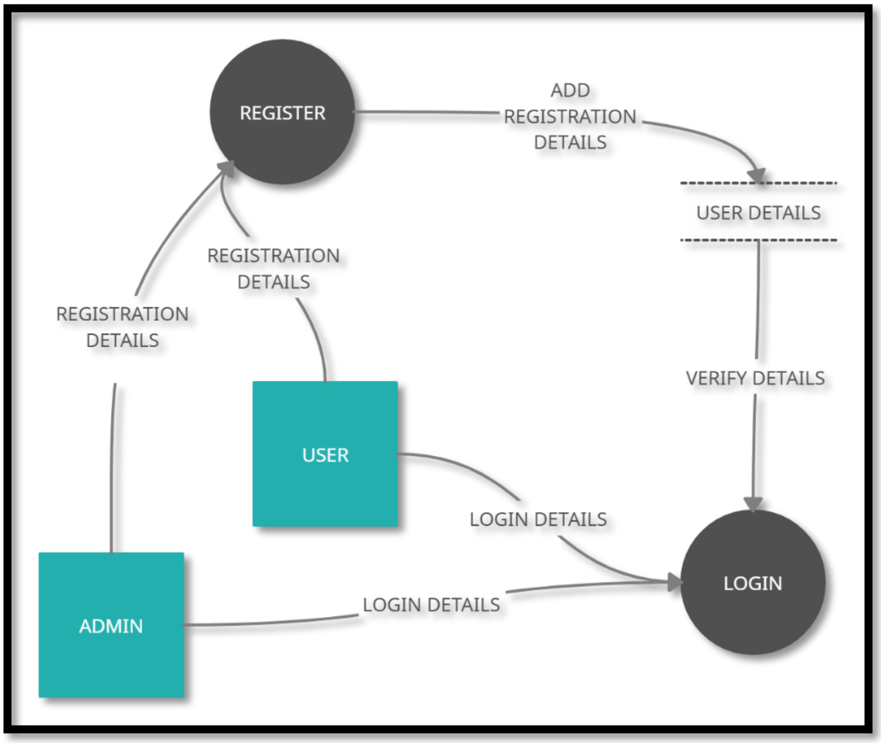
Testing

We will be performing Software Testing on the **Login Sub-Module of Login/Register Module** and **Add Course by Playlist Sub-Module Of Course Management Module**.

- **Test Cases for Login Sub-Module**:
  - **Test Case 1**
    - Title: Login User for Valid Inputs
    - Steps:
      - Go to The Login Screen of the System
      - Enter the Username in the Username Input Field
      - Enter the Password in the Password Input Field
      - Press the Login Button on the Screen
    - Input:
      - Username: Alphanumeric of length 5 – 25
      - Password: Alphanumeric with a Special Character of minimum 8 and maximum 24
    - Precondition: The User/Admin Should be registered on the System
    - Assumption: The User/Admin should be connected to the Internet
    - Expected Result: The User/Admin is given Access to the System
    - Actual Result: As Expected
    - Status: Passed
  - **Test Case 2**
    - Title: Login User for Valid Inputs
    - Steps:
      - Go to The Login Screen of the System
      - Enter the Username in the Username Input Field
      - Enter the Password in the Password Input Field
      - Press the Login Button on the Screen
    - Input:
      - Username: not Alphanumeric of length 5 – 25
      - Password: not Alphanumeric with a Special Character of minimum 8 and maximum 24
    - Precondition: The User/Admin Should be registered on the System
    - Assumption: The User/Admin should be connected to the Internet
    - Expected Result: The User/Admin is not given Access to the System and an Error Message is displayed.
    - Actual Result: As Expected
    - Status: Passed

Black Box Testing

We will perform Boundary Value Analysis, Robustness and Equivalence Partition testing on the **Login Sub-Module of the Login/Register Sub-Module**



1. **Boundary Value Analysis**

For the Login Sub-Module we have the following Inputs and their Condition:

a. 5 ≤ Length of Username ≤ 25

b. 8 ≤ Length of Password ≤ 24

There will be $4 * n + 1 = 4 * 2 + 1 = 9$ Test Case for the Conditions will be:

Test Case	Length of Username	Length of Password	Expected Output
1.	5	8	Valid
2.	6	8	Valid
3.	10	8	Valid
4.	24	8	Valid
5.	25	8	Valid
6.	10	8	Valid
7.	10	9	Valid
8.	10	23	Valid
9.	10	24	Valid

2. **Robustness Testing**:

For the Login Sub-Module we have the following Inputs and their Condition:

a. 5 ≤ Length of Username ≤ 25

b. 8 ≤ Length of Password ≤ 24

There will be $6 * n + 1 = 6 * 2 + 1 = 13$ Test Case for the Conditions will be:

Test Case	Length of Username	Length of Password	Expected Output
1.	5	8	Valid
2.	6	8	Valid
3.	10	8	Valid
4.	24	8	Valid
5.	25	8	Valid
6.	10	8	Valid
7.	10	9	Valid
8.	10	23	Valid
9.	10	24	Valid
10.	4	8	Invalid
11.	26	8	Invalid
12.	10	7	Invalid
13.	10	25	Invalid

3. **Equivalence Partition**:

For the Login Sub-Module we have the following Inputs and their Condition:

a. 5 ≤ Length of Username ≤ 25

b. 8 ≤ Length of Password ≤ 24

The Equivalence Partitioning for the Conditions will be:

Length of Username

Invalid	Valid	Invalid
≤ 4	5 - 25	≥ 26

Length of Password

Invalid	Valid	Invalid
≤ 7	8 - 24	≥ 25

White Box Testing

For White Box Testing we will perform the Path Testing by which works as follows:

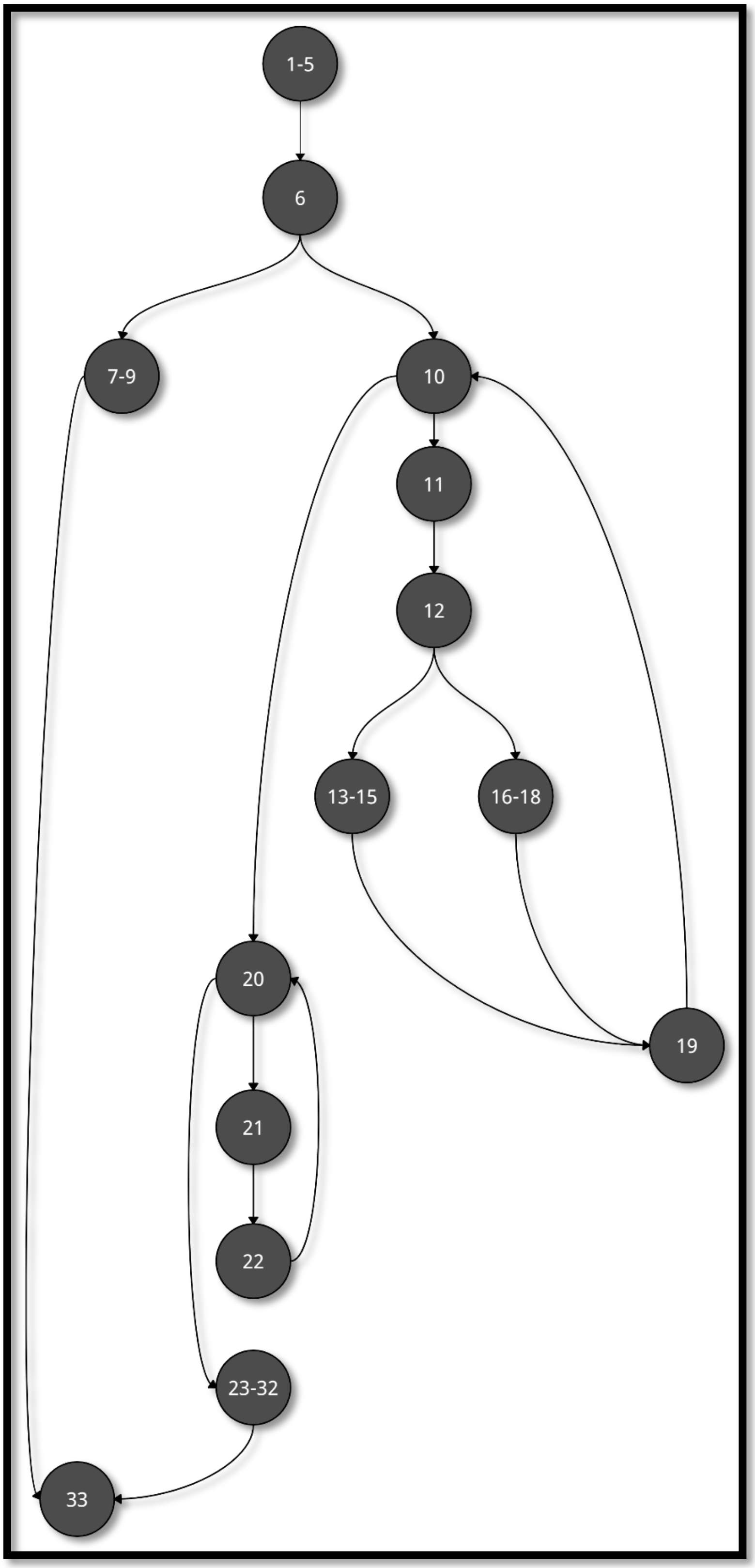
a. Construct the Flow Graph from Source Code
b. Calculate the Cyclomatic Complexity
c. Find the Independent Paths

Pseudocode

```
01: function addPlaylist(playlistID) {
02:     Create videosURL array, videos array, playlist Object
03:     set durationPlayist = 0
04:     videosURL = getVideosURL(playlistID)
05:     playlistData = getPlaylistData(playlistID)
06:     if(videosURL is NULL or playlistData is NULL)
07:     {
08:         exit
09:     }
10:     foreach (url in videosURL) {
11:         videoObject = getVideoObject(url)
12:         if (videoObject is not NULL)
13:         {
14:             Add videoObject to videos Array/List
15:         }
16:          else {
17:             Report Error and Continue
18:         }
19:     }
20:     foreach (video in videos) {
21:         durationPlayist = durationPlayist + video.durationVideo
```

```
    }
    playlist = {
        name:playlistData.title,
        playlist_id:playlistData.id,
        thumbnail:playlistData.bestThumbnail.url,
        description:playlistData.description,
        category:playlistData.author.name
        duration:durationPlayist,
        playlistVideos: videos
    }
    return savePlaylistToDatabase(playlist)
}
```

Flow Graph



Calculating Cyclomatic Complexity

By using the McCabe's Cyclomatic Matrix for a Graph G with n nodes and E Edges, the Cyclomatic Complexity is given by:

$$\boldsymbol{V(G) = E - N + 2 * P}$$

Thus, for the Generated Flow Graph of the Add Course/Tutorial by Playlist Sub-Module:

$$\boldsymbol{E = 17}$$
$$\boldsymbol{N = 14}$$
$$\boldsymbol{P = 1}$$

Cyclomatic Complexity of the constructed Graph will be:

$$\boldsymbol{V(G) = 17 - 14 + (2 * 1)}$$

$$\therefore \boldsymbol{V(G) = 5}$$

Independent Paths

The number of Independent Paths in a Flow Graph is equal to the **Cyclomatic Complexity V(G)**

Thus, for the Generated Flow Graph we will have 5 Independent Paths:

a. (1-5) -> 6 -> (7-9)-> 33
b. (1-5) -> 6 -> 10 -> 11 -> 12 -> (13-15) -> 19 -> 10-> 20 -> (21-22) -> 20 -> (24-32) -> 33
c. (1-5) -> 6 -> 10 -> 11 -> 12 -> (16-18) -> 19 -> 10-> 20 -> (21-22) -> 20 -> (24-32) -> 33
d. (1-5) -> 6 -> 10 -> 20 -> (21-22) -> 20 -> (24-32) -> 33
e. (1-5) -> 6 -> 10 -> 11 -> 12 -> (13-15) -> 19 -> 10-> 20 -> (24-32) -> 33