router.post("/login", async (*req*, *res*) *=>* {

(1**)**  *const* { username, password } = *req*.body;

  //simple validation

(2) if (!username || (3)!password) {

(4)   return *res*

      .status(400)

      .json({ success: false, message: "Missing username and/or password" });

  }

  try {

    //check for existing username

(5)    *const* account = await Account.findOne({ username });

(6)   if (!account)

(7)      return *res*

        .status(400)

        .json({ success: false, message: "Incorrect username or password" });

    // Username found

(8)    *const* passwordValid = await argon2.verify(account.password, password);

(9)    if (!passwordValid)

(10)      return *res*

        .status(400)

        .json({ success: false, message: "Incorrect username or password" });

    //All good

(11)    *const* userInfo = await User.findOne({ account: account.\_id });

    //return token

(12)   *const* accessToken = jwt.sign(

      {

        verifyAccount: {

          id: account.\_id,

          isHidden: account.isHidden,

          username: account.username,

          role: account.role,

        },

      },

      process.env.ACCESS\_TOKEN\_SECRET

    );

(13)    *res*.json({

      success: true,

      message: "User logged successfully",

      accessToken: accessToken,

      user: userInfo,

    });

  } catch (error) {

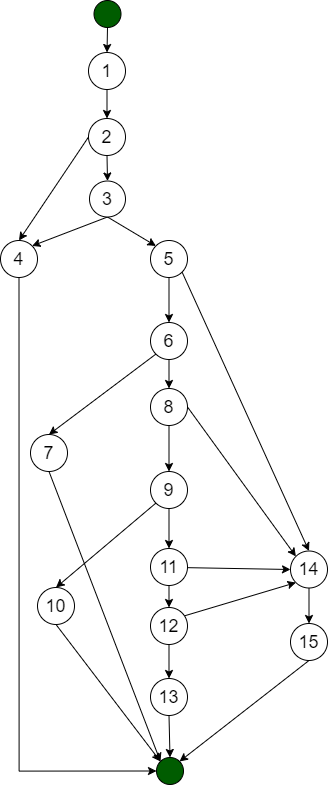
(14)  console.log(error);

(15) *res*.status(500).json({ success: false, message: "Internal server error" });

  }

});

1. Create control flow graph



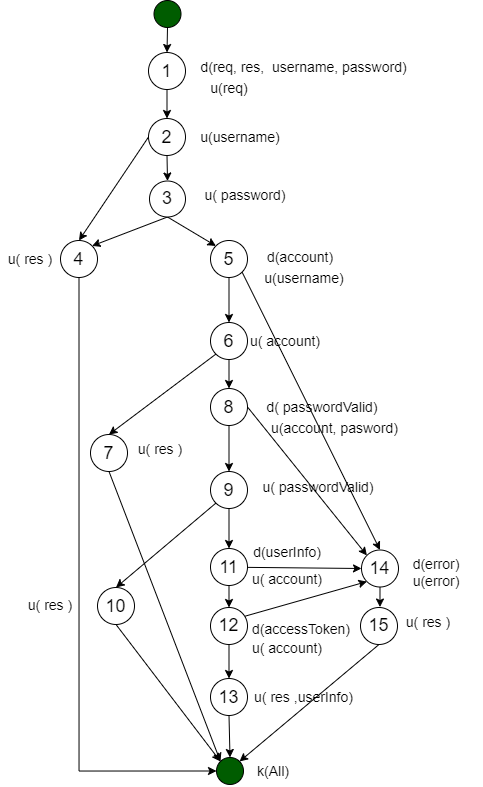
1. Calculate Cyclomatic complexity to identify independent linear execution paths.

V= 24- 17+2 = 9

1. Design test cases for each path

|  |  |  |
| --- | --- | --- |
| Path | Data | Expected result |
| 1-2-3-4 | req.body= {username: “”, password: “1245”} | { success: false, message: "Missing username and/or password" } |
| 1-2-3-5-6-7 | req,body={username: “admin”, password:’12345’} | { success: false, message: "Incorrect username or password" } |
| 1-2-3-5-14 | req.body={username: “”, password: “12345”} |  |
| 1-2-3-5-6-8-9-10 | req.body={username:“minhhoang1” , password: ”123”} | { success: false, message: "Incorrect username or password" } |
| 1-2-3-5-6-8-9-11-12-13 | req.body={username:“minhhoang1”, password: “12345”} | {  "success": true,  "message": "User logged successfully",  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  eyJ2ZXJpZnlBY2NvdW50Ijp7ImlkIjoiNjE4ZGNlYThhNjExZ  jQzNDAyOTYzMjhjIiwidXNlcm5hbWUiOiJtaW5oaG9hb  mcxIiwicm9sZSI6ImFkbWluIn0sImlhdCI6MTYzODMyNj  AzNX0.7r4ILoIHjxlmWn5D2el4yDFzy6X8PMMpl-7PZV  zD\_24",  "user": {  "fullname": "Nguyen Anh Tin",  "address": null,  "dod": null,  "school": null,  "phone": "124567855",  "account": "618dcea8a611f4340296328c",  "avatar": null,  "isHidden": false,  "createdAt": "2021-11-12T09:17:12.000Z",  "updatedAt": "2021-11-12T09:17:12.000Z",  "id": "618dcea8a611f4340296328e"  }  } |
| 1-2-3-5-6-8-14-15 | req.body={username:“minhhoang1”, password: “12345”} xảy ra lỗi trong quá trình giải mã argon2 | { success: false, message: "Internal server error" } |
| 1-2-3-5-6-8-9-11-14-15 | req.body={username:“minhhoang1”, password: “12345”} xảy ra expeption trong quá trình truy suất mongoDB | { success: false, message: "Internal server error" } |
| 1-2-3-5-6-8-9-11-12-14-15 | req.body={username:“minhhoang1”, password: “12345”} xảy ra lỗi trong quá trình tạo accessToken bằng jwt | { success: false, message: "Internal server error" } |

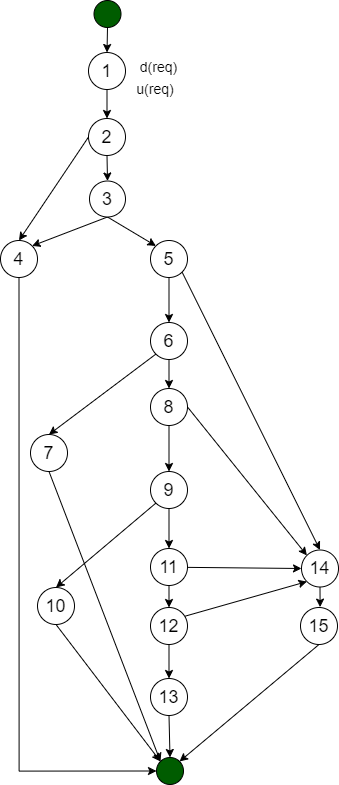
1. Create the general data flow graph



5.Perform lifecycle testing for each variable in the unit

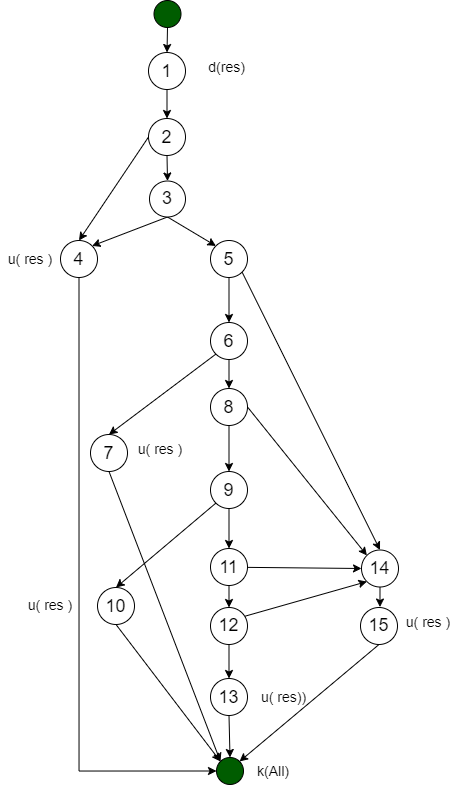
5.1. Kịch bản biến req đều là ~duk.

- Các kịch bản không chứa cặp đôi nào hoạt động bất thường.



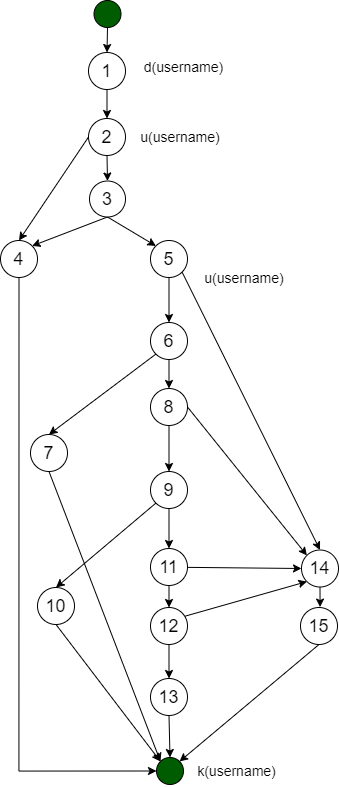
5.3 Kịch bản biến res đều là ~duk

- Các kịch bản không chứa cặp đôi nào hoạt động bất thường.



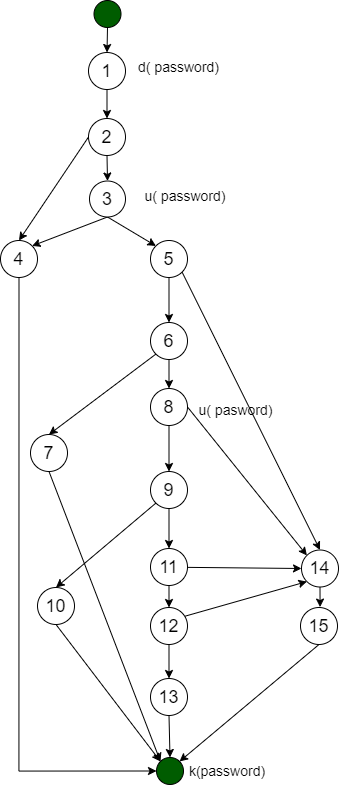
5.4 Kịch bản biến username đều là ~duk và duuk

- Các kịch bản không chứa cặp đôi nào hoạt động bất thường.



5.5 Kịch bản biến password đều là ~dk, ~duk, ~duuk

- Các kịch bản không chứa cặp đôi nào hoạt động bất thường.



5.6 Kịch bản biến account đều là ~d k, ~duk, ~duuk, ~duuuk, ~duuuuk

- Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

A picture containing necklet, scale, accessory

Description automatically generated

5.7 Kịch bản biến passwordValid đều là ~dk, ~duk

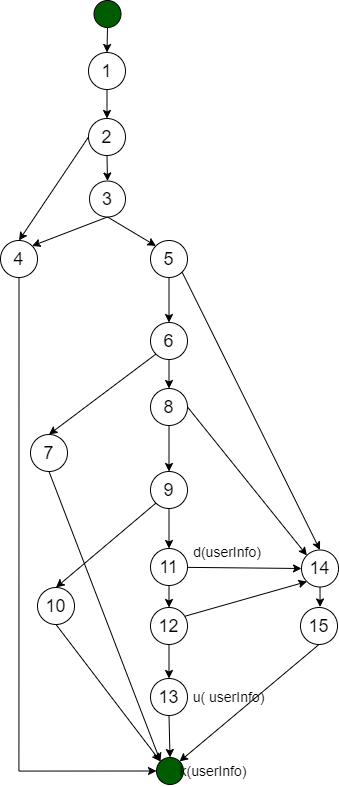
- Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

A picture containing necklet, scale, accessory, scissors

Description automatically generated

5.8 Kịch bản biến userInfo đều là ~dk, ~duk

- Các kịch bản không chứa cặp đôi nào hoạt động bất thường.



5.7 Kịch bản biến error đều là ~duk

- Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

