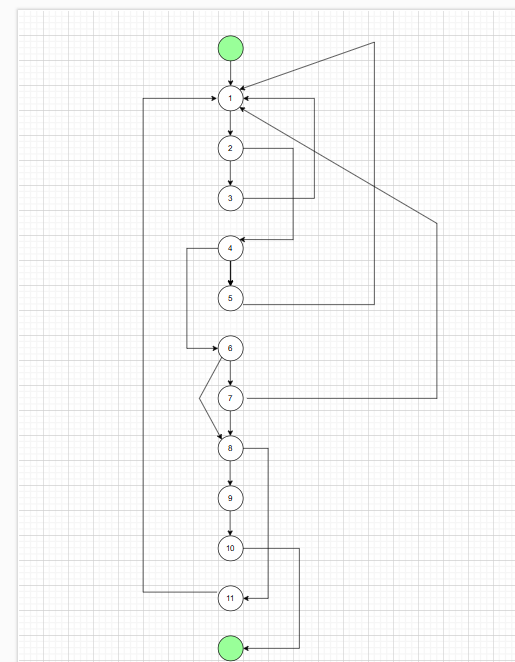
**Mã nguồn 1:**

|  |
| --- |
| public void processRegister()  {  boolean isValid;          do {  *// Nút 1: Lấy input*              String email = edtTK.getText().toString().trim();              String password = edtMK.getText().toString().trim();              String confirmpassword = edtNhapLai.getText().toString().trim();  *// Nút 2-6: Validate*              if (email.isEmpty() || password.isEmpty() || confirmpassword.isEmpty()) {                  Toast.makeText(this, "Vui lòng điền đủ thông tin", Toast.LENGTH\_SHORT).show();                  isValid = false;              } else if (!password.equals(confirmpassword)) {                  Toast.makeText(this, "Mật khẩu không trùng khớp", Toast.LENGTH\_SHORT).show();                  isValid = false;              } else if (isCustomerExists(email)) {                  Toast.makeText(this, "Email đã tồn tại", Toast.LENGTH\_SHORT).show();                  isValid = false;              } else {                  isValid = true;              }              if (isValid) {                  try {  *// Nút 7: Tạo customer*                      Customer customer = new Customer(email, password);  *// Nút 8: Thêm vào DB*                      CustomerDAO customerDAO = new CustomerDAO(RegisterActivity.this);                      customerDAO.themCustomer(customer);  *// Nút 9: Chuyển màn hình*                      Intent intent = new Intent(RegisterActivity.this, LoginActivity.class);                      startActivity(intent);                  } catch (Exception *e*) {  *// Nút 10: Xử lý lỗi*                      isValid = false;                  }              }          } while (!isValid);      }      private boolean validateInput(String *email*, String *password*, String  *confirmpassword*)      {          if (email.isEmpty() || password.isEmpty() || confirmpassword.isEmpty())          {              Toast.makeText(RegisterActivity.this, "Vui lòng điền đầy đủ thông tin", Toast.LENGTH\_SHORT).show();              return false;          }          else          {              if (!password.equals(confirmpassword)) {                  Toast.makeText(RegisterActivity.this, "Mật khẩu không trùng khớp", Toast.LENGTH\_SHORT).show();                  return false;              }              if (isCustomerExists(email))              {                  Toast.makeText(RegisterActivity.this, "Email đã tồn tại", Toast.LENGTH\_SHORT).show();                  return false;              }          }          return true;      }      private boolean isCustomerExists(String *id*)      {          return false;      } |

**i.Lập đồ thị dòng điều khiển cơ bản.**



**ii.Tính số đường thi hành tuyến tính độc lập.**

Để tính số đường thi hành tuyến tính độc lập (cyclomatic complexity) của đoạn mã trên, chúng ta sử dụng công thức:

M=E−N+2P

**Trong đó:**

E: Số cạnh trong đồ thị điều khiển luồng.

N: Số nút trong đồ thị điều khiển luồng.

P: Số thành phần kết nối (thường là P=1 nếu chỉ có một khối mã).

**Phân tích đoạn mã:**

**1. Xác định các khối điều kiện và vòng lặp:**

· do {...} while (!isValid): Nút 1 quay lại khi isValid = false

· if (email.isEmpty()...): Nút 2 kiểm tra form trống

· else if (!password.equals()): Nút 4 kiểm tra password

· else if (isCustomerExists()): Nút 6 kiểm tra email

· try {...} catch: Nút 8 và 11 (try-catch block)

**2. Vẽ đồ thị điều khiển luồng:**

Mỗi khối mã được xem như một nút.

Các nhánh từ điều kiện (nếu/không hoặc các khối try/catch) tạo ra các cạnh.

**Đếm E và N:**

**N (số nút):**

Toàn bộ khối mã chứa các khối logic chính:

1. Khối bắt đầu.

2. if (email.isEmpty()).

3. if (!password.equals()).

4. if (isCustomerExists()).

5. Khối try/catch bên trong phần kiểm tra thông tin isValid (có 2 nút cho try và catch).

=> Tổng cộng: N=6.

**E (số cạnh):**

1. Start -> 1 (cạnh 1)

2. 1 -> 2 -> 3 (cạnh 2)

3. 3 -> 4 (cạnh 3)

4. 4 -> End (cạnh 4)

5. 3 -> 5 -> 6 -> 7 (cạnh 5)

6. 7 -> 15 -> 16 -> 7 (cạnh 6)

7. 7 -> 8 -> 9 (cạnh 7)

8. 9 -> 13 -> 14 -> End (cạnh 8)

9. 9 -> 10 -> End (cạnh 9)

10. 9 -> 11 -> 12 -> End (cạnh 10)

=> Tổng E = 10 cạnh

P: Chỉ có một khối mã, nên P=1.

**Tính** M:

M = E - N + 2P

M = 10 - 6 + 2(1)

M = 10 - 6 + 2

M = 6

Vậy số đường thi hành tuyến tính độc lập là 6

**iii.Thiết kế test cases cho từng đường.**

***Test case 1:*** Form trống

*Mô tả:* Kiểm tra khi người dùng không nhập đủ thông tin

*Chi Tiết Test Case:*

* Dữ liệu đầu vào:

Dữ liệu đầu vào:

email: ""

password: ""

confirmpassword: ""

Kết quả mong đợi:

+ Toast hiển thị: "Vui lòng điền đầy đủ thông tin"

+ isValid = false, quay lại nhập

***Test case 2:*** Password không khớp

*Mô tả:* Kiểm tra khi password và confirm password không giống nhau

*Chi Tiết Test Case:*

* Dữ liệu đầu vào:

email: "test@gmail.com"

password: "123456"

confirmpassword: "1234567"

* Mong đợi:

+ Toast hiển thị: "Mật khẩu không trùng khớp"

+ isValid = false, quay lại nhập

***Test case 3:*** Email đã tồn tại

*Mô tả:* Kiểm tra khi email đã được đăng ký trước đó

*Chi Tiết Test Case:*

* Dữ liệu đầu vào:

email: "exist@gmail.com" (đã có trong DB)

password: "123456"

confirmpassword: "123456"

* Mong đợi:

+ Toast hiển thị: "Email đã tồn tại"

+ isValid = false, quay lại nhập

***Test case 4:*** Exception khi tạo tài khoản

*Mô tả:* Kiểm tra khi xảy ra lỗi trong quá trình tạo tài khoản

*Chi Tiết Test Case:*

* Dữ liệu đầu vào:

email: "test@gmail.com"

password: "123456"

confirmpassword: "123456"

* Mong đợi:

+ Catch được exception

+ isValid = false, quay lại nhập

***Test case 5:*** Đăng ký thành công

*Mô tả:* Kiểm tra luồng đăng ký hoàn chỉnh không lỗi

*Chi Tiết Test Case:*

* Dữ liệu đầu vào:

email: "newuser@gmail.com"

password: "123456"

confirmpassword: "123456"

* Mong đợi:

+ Tạo được customer mới

+ Thêm vào DB thành công

+ Chuyển sang màn hình Login

***Test case 6:*** Lỗi khi thêm vào database

*Mô tả:* Kiểm tra luồng đăng ký hoàn chỉnh không lỗi

*Chi Tiết Test Case:*

* Dữ liệu đầu vào:

email: "newuser@gmail.com"

password: "123456"

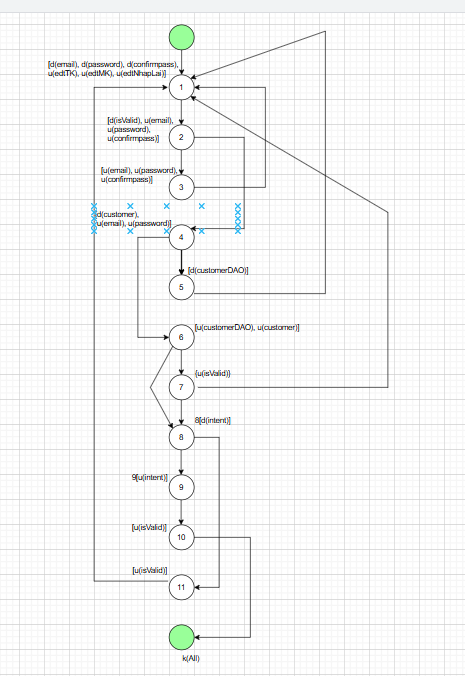
confirmpassword: "123456"

* Mong đợi:

+ Catch được exception khi thêm DB

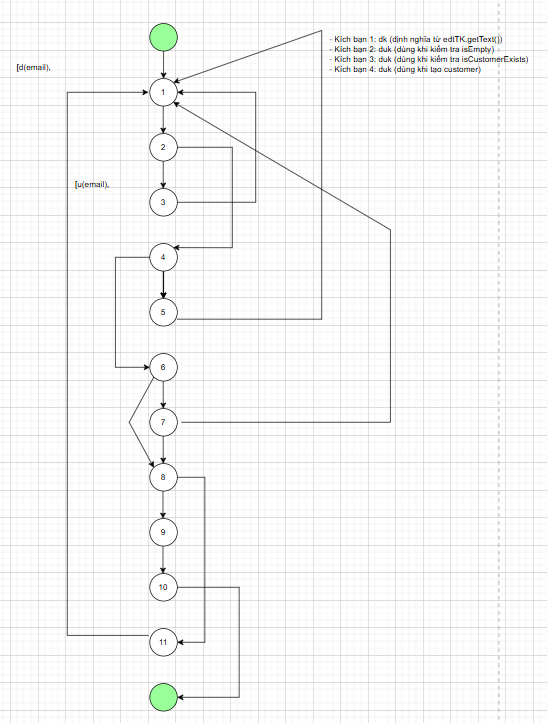
+ isValid = false, quay lại nhập

**iv.Lập đồ thị dòng dữ liệu chung.**

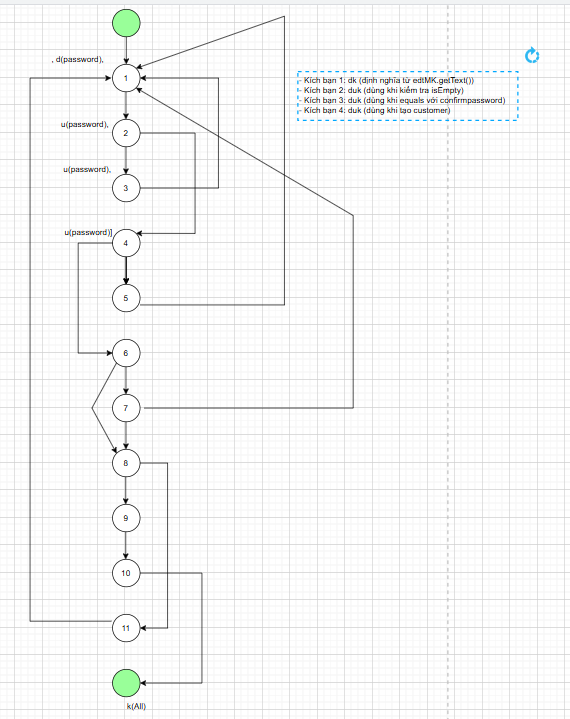


**v.Kiểm thử đời sống của từng biến trong đơn vị mã nguồn đó.**

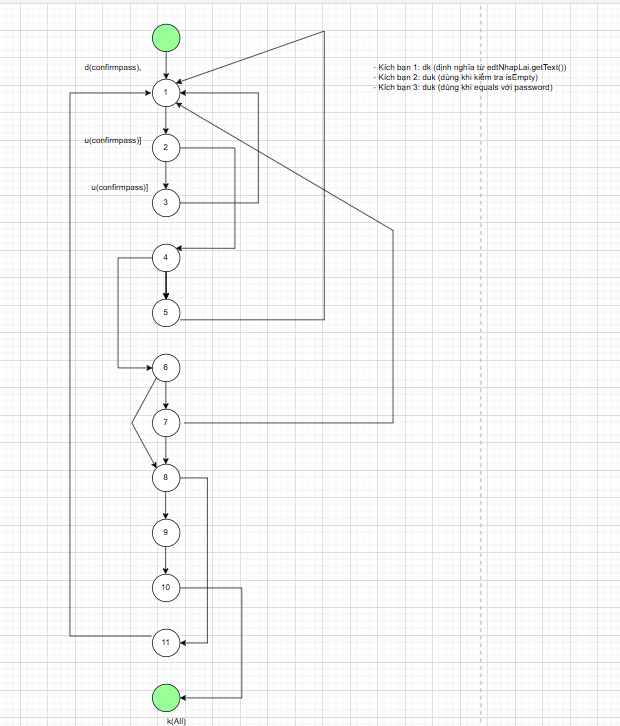
**- Kiểm thử đời sống biến email:**



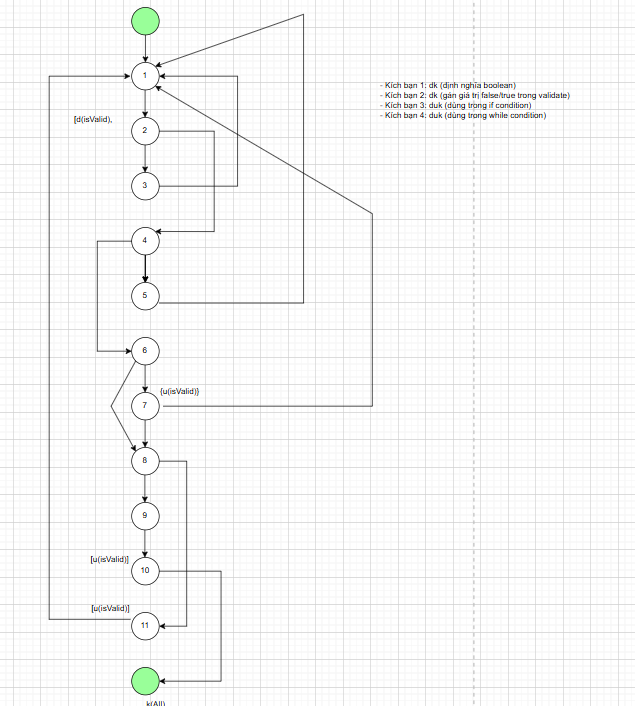
**- Kiểm thử đời sống biến password:**



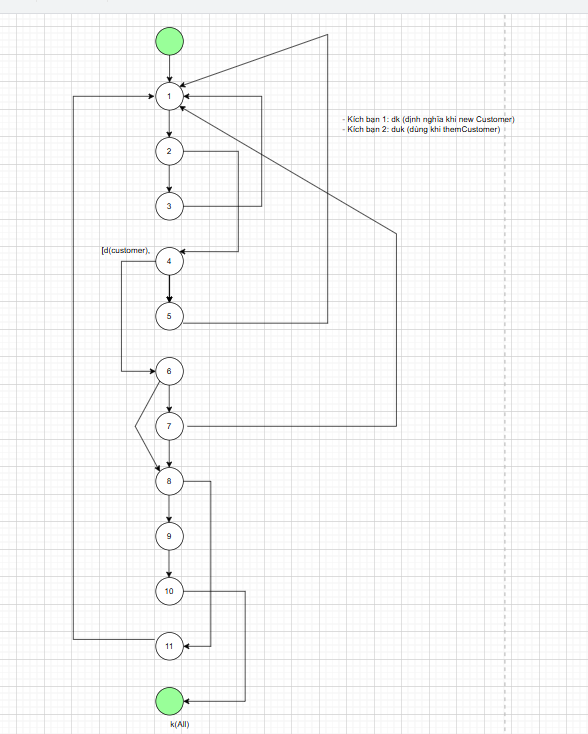
**- Kiểm thử đời sống biến confirmpassword:**



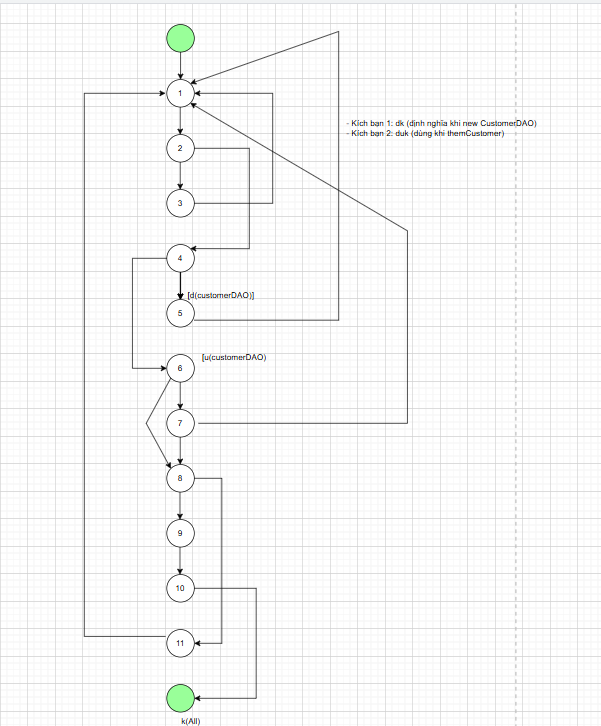
**- Kiểm thử đời sống biến isValid:**



**- Kiểm thử đời sống biến customer:**



**- Kiểm thử đời sống biến customerDAO:**



**- Kiểm thử đời sống biến intent:**

