**Đánh giá Fresher giai đoạn 2 - Dự án Amis Kế toán**

1. **Thời gian hoàn thành:**

Fresher hoàn thành yêu cầu trong 1 tháng

* Danh mục: 2 tuần
* Nghiệp vụ: 1 tuần
* Hoàn thiện: 1 tuần

Bắt đầu làm bài từ: 26/5/2021

Sau khi hoàn thiện bài thi thí sinh đóng gói Toàn bộ nội dung bài làm vào thư mục và nén lại theo nguyên tắc đặt tên:

Thí sinh nộp bài theo link sau:

1. **Các tiêu chí đánh giá cần đạt được:**

* Kỹ năng lập trình:
  + Kỹ năng sử dụng công cụ Visual Studio, Visual Code, PostgreSQL,
  + Kỹ năng Debug ứng dụng với Visual Studio.
  + Kỹ năng Debug JS sử dụng Chrome Developer Tool.
  + Khả năng phân tích lỗi, tìm lỗi, sửa lỗi.
* Kiến thức lập trình:
  + Lập trình hướng đối tượng với C#
  + Sử dụng thành thạo Javascript, VueJS.
  + Sử dụng EF core.
  + ASP.NET WEB API, thiết kế các API chuẩn RESTFul Service.
  + Thiết kế Database bằng PostgreSQL.
* Tuân thủ MISA Convention:
  + MISA Coding Convention.
  + MISA UI Convention.
  + MISA Database Convention.
* Các tiêu chí khác:
  + Hiểu nghiệp vụ, nhu cầu của bài toán, hiểu tính năng cần làm.

1. **Nội dung yêu cầu:**

**AMIS Kế toán** là sản phẩm phục vụ cho công tác Kế toán dành cho các loại hình doanh nghiệp do MISA cung cấp.

Tiền mặt là một phân hệ nằm trong AMIS Kế toán

Kế toán phải mở sổ kế toán ghi chép hàng ngày liên tục theo trình tự phát sinh các khoản thu, chi, xuất, nhập tiền, ngoại tệ và tính ra số tồn tại quỹ tại mọi thời điểm để tiện cho việc kiểm tra, đối chiếu

**Nhiệm vụ:**

1. Thi công tab Quy trình
2. Các danh mục liên quan
   1. Đối tượng
   2. Tài khoản
3. Thi công tab Thu, chi tiền
   1. Màn hình danh sách Thu, chi tiền
   2. Phiếu chi (Chi khác)

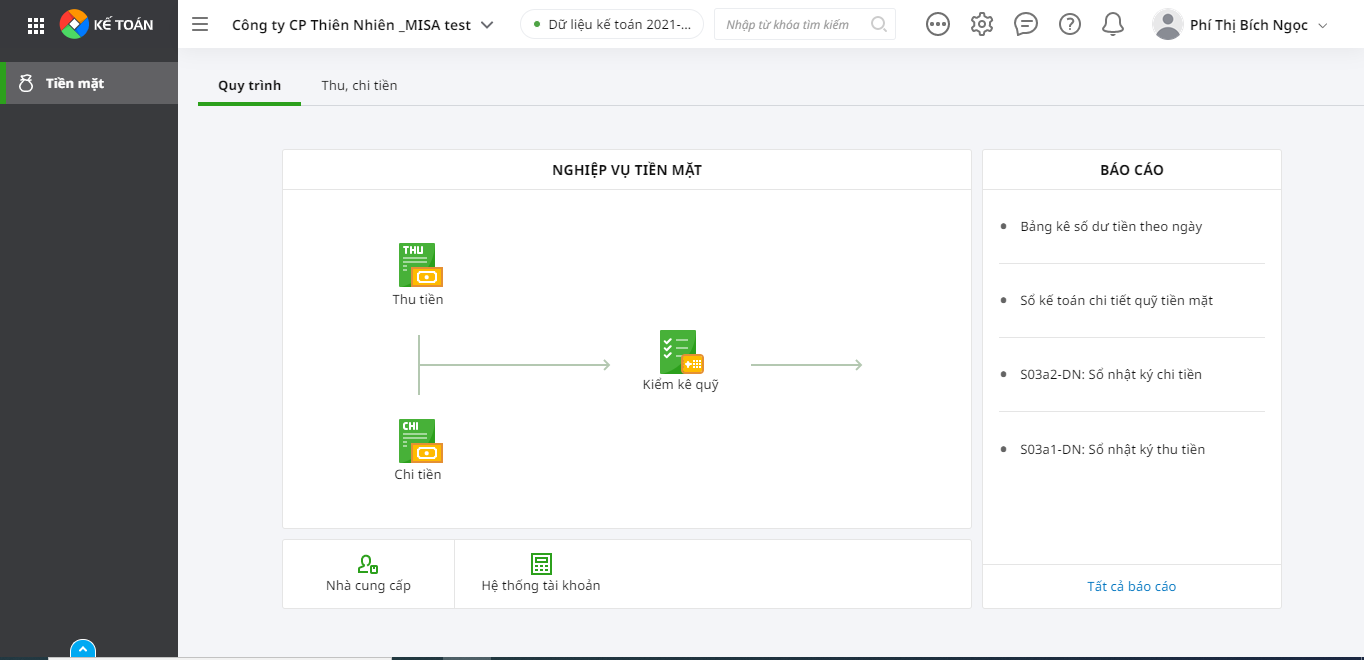
**Tham khảo, lấy icon/resource tại:**

1. Đường dẫn: [https://actapp.misa.vn/app/ca/caprocess](https://actappdc.misa.vn/app/ca/caprocess)
2. Tên đăng nhập/mật khẩu: [EnKiu.Minh@gmail.com/12345678@Abc](about:blank)

**IV. Yêu cầu chi tiết**

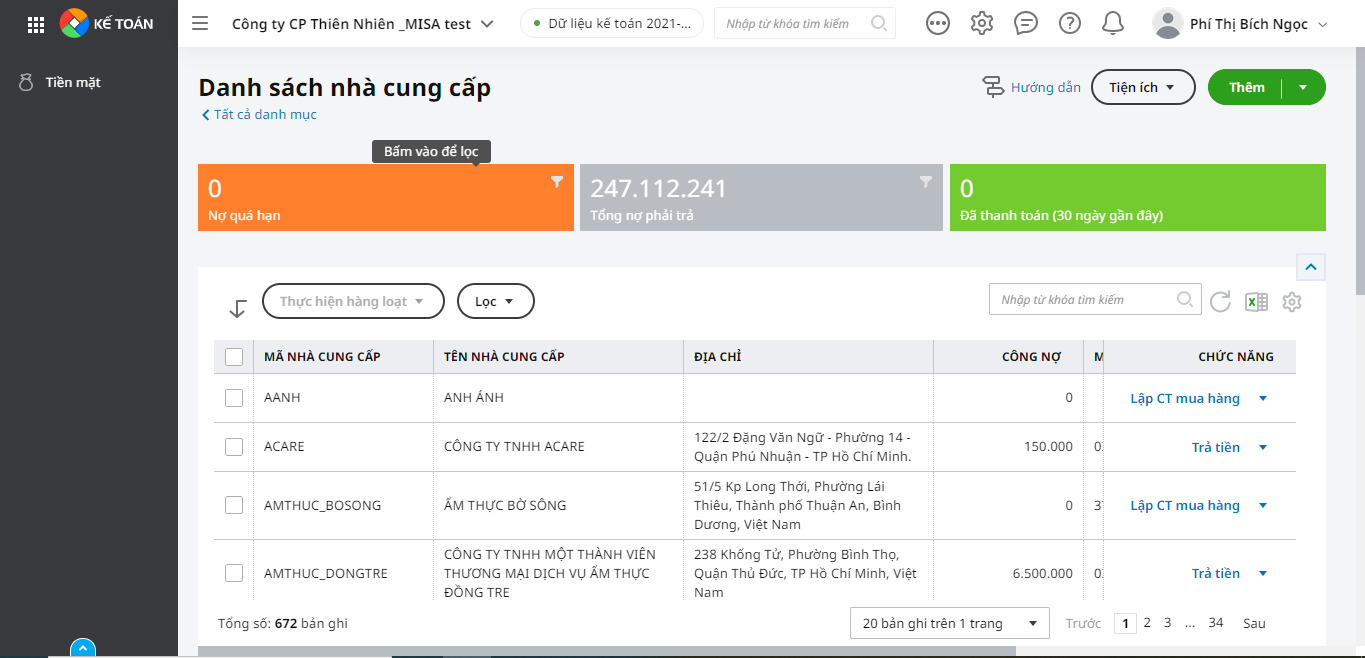
Rule chung: Tự động focus vào ô nhập liệu đầu tiên, có validate các trường require, có sử dụng phím tắt trên các form chi tiết

1. **Thi công tab quy trình**

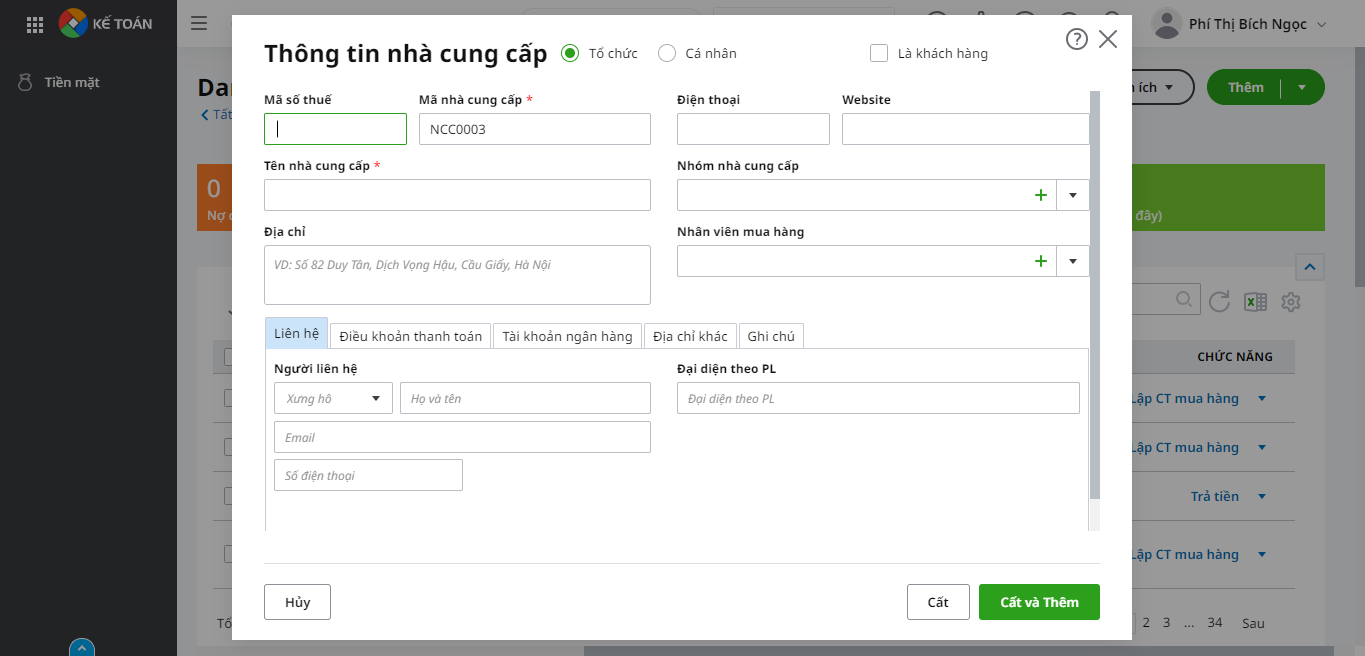
****

* Bấm vào icon chi tiền sẽ mở trang thêm phiếu chi tiền mặt
* Bấm vào icon Nhà cung cấp sẽ mở danh sách nhà cung cấp
* Bấm vào icon Hệ thống tài khoản sẽ mở danh sách Hệ thống tài khoản

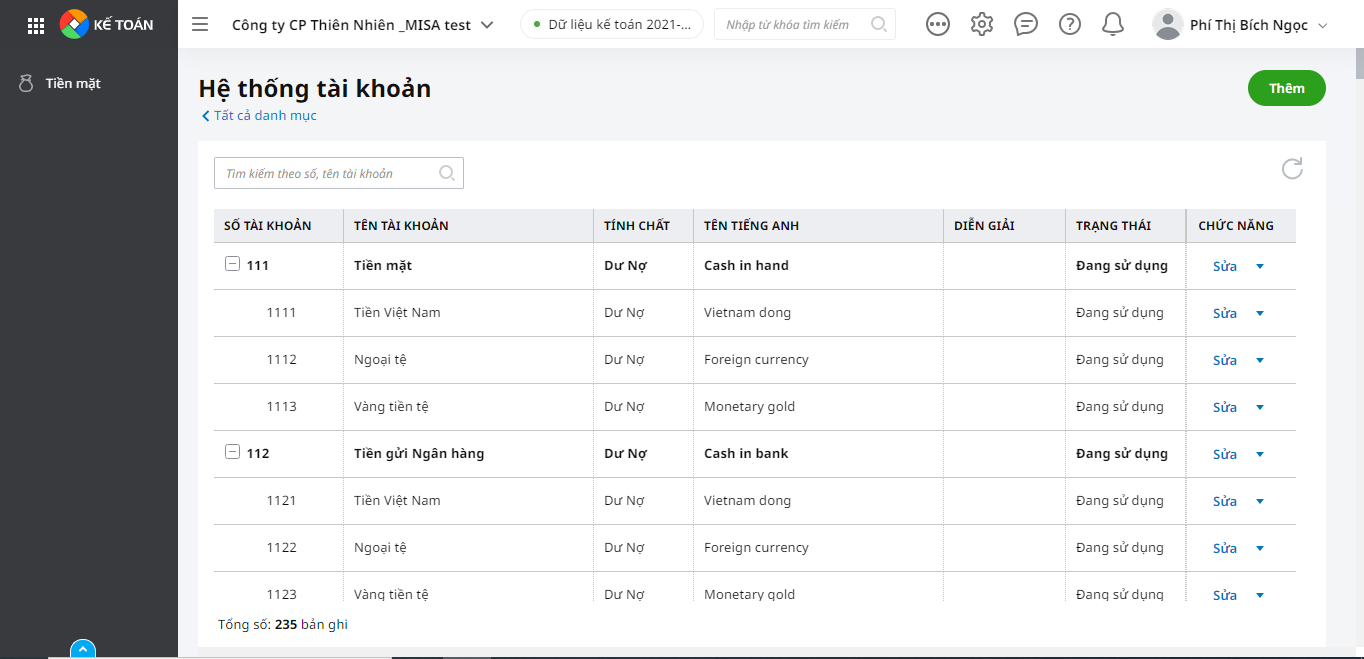
1. Cách danh mục liên quan
   1. Đối tượng
      * Danh sách



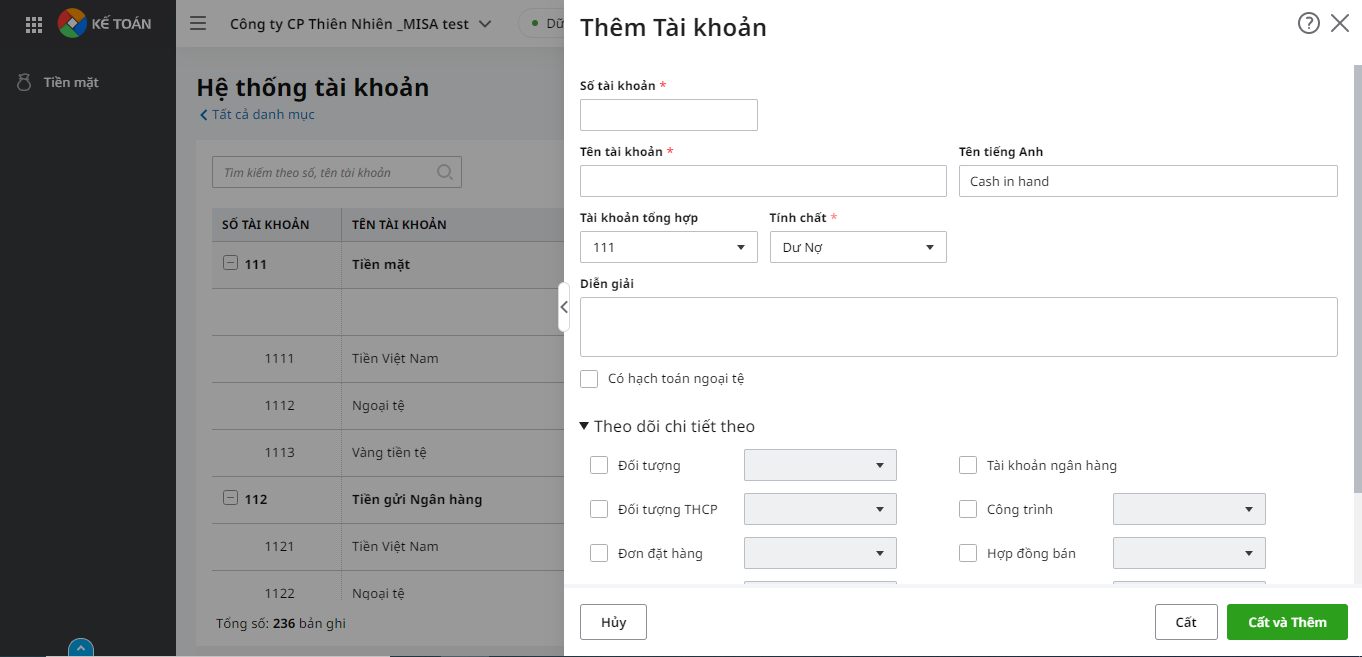
* + Giải pháp phần mềm
    - Danh sách Nhà cung cấp bao gồm các cột: Mã NCC, Tên NCC, Địa chỉ, Mã Số thuế, SĐT, CMTND, Chức năng. Sắp xếp tăng dần theo mã NCC
    - Cho phép tìm kiếm theo tất cả các cột trên
    - Thực hiện các chức năng ở cột chức năng:
      * Xem: View lại toàn bộ thông tin NCC lên form NCC và readOnly toàn bộ thông tin
      * Sửa: View lại toàn bộ thông tin NCC lên form, tự động focus vào ô nhập liệu đầu tiên
      * Xóa: Xóa NCC này khỏi danh sách
    - Danh sách cho phép phân trang. Mặc định là 20 bản ghi trên 1 trang. Cho phép chọn các tùy chọn 10, 20, 50, 100 bản ghi trên 1 trang
    - Chức năng Thêm: Cho phép mở form Thêm NCC
    - Form Chi tiết NCC



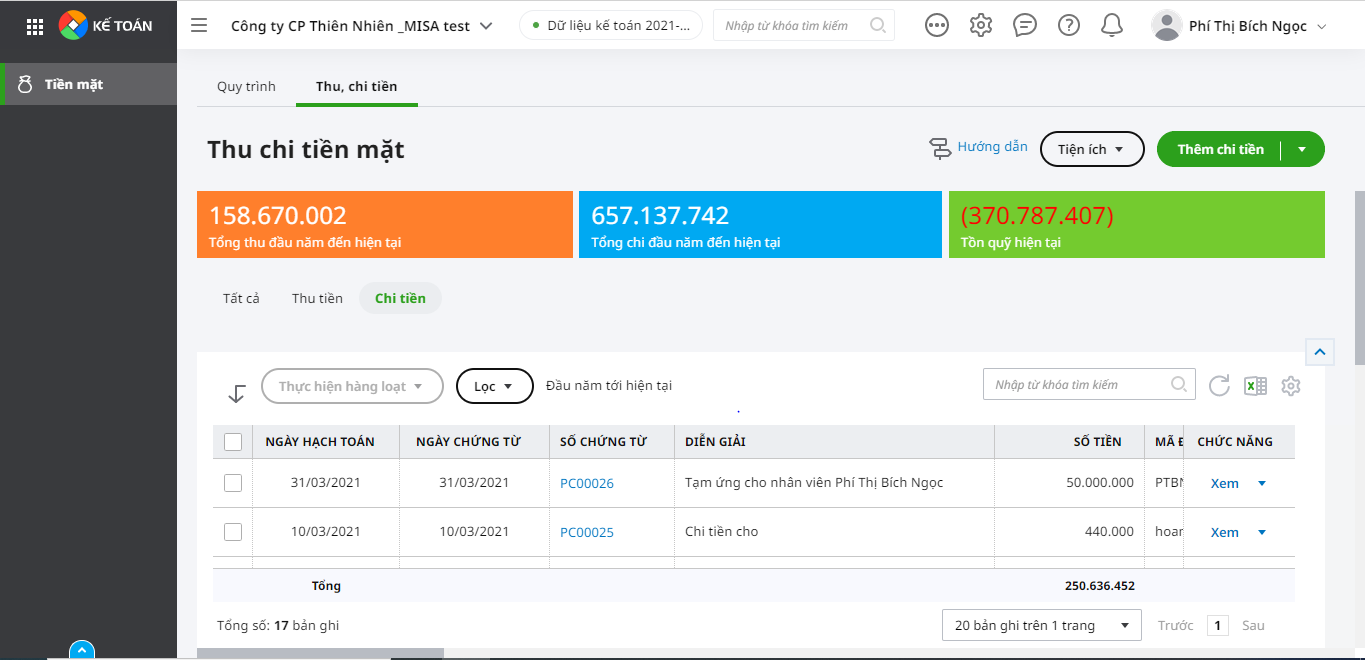
* + Giải pháp phần mềm
    - Thiết kế giao diện ẩn hiện theo các chức năng trên form demo
    - Một số lưu ý:
      * Thêm mới form sẽ tự động focus vào ô nhập liệu đầu tiên
      * Tên nhà cung cấp, Mã nhà cung cấp không được phép bỏ trống và mã tự động tăng theo cấu trúc NCC00001. Lần thêm tiếp theo NCC00002
      * Nhóm Nhà cung cấp cho phép chọn nhiều (Dấu + chỉ cần vẽ không cần làm tính năng)
      * Nhân viên mua hàng: Load từ danh mục nhân viên chỉ được chọn một (Dấu + chỉ cần vẽ không cần làm tính năng)
      * Sử dụng được phím tắt ESC: để đóng form, Ctrl + S: Cất
  1. Hệ thống tài khoản
     + Danh sách



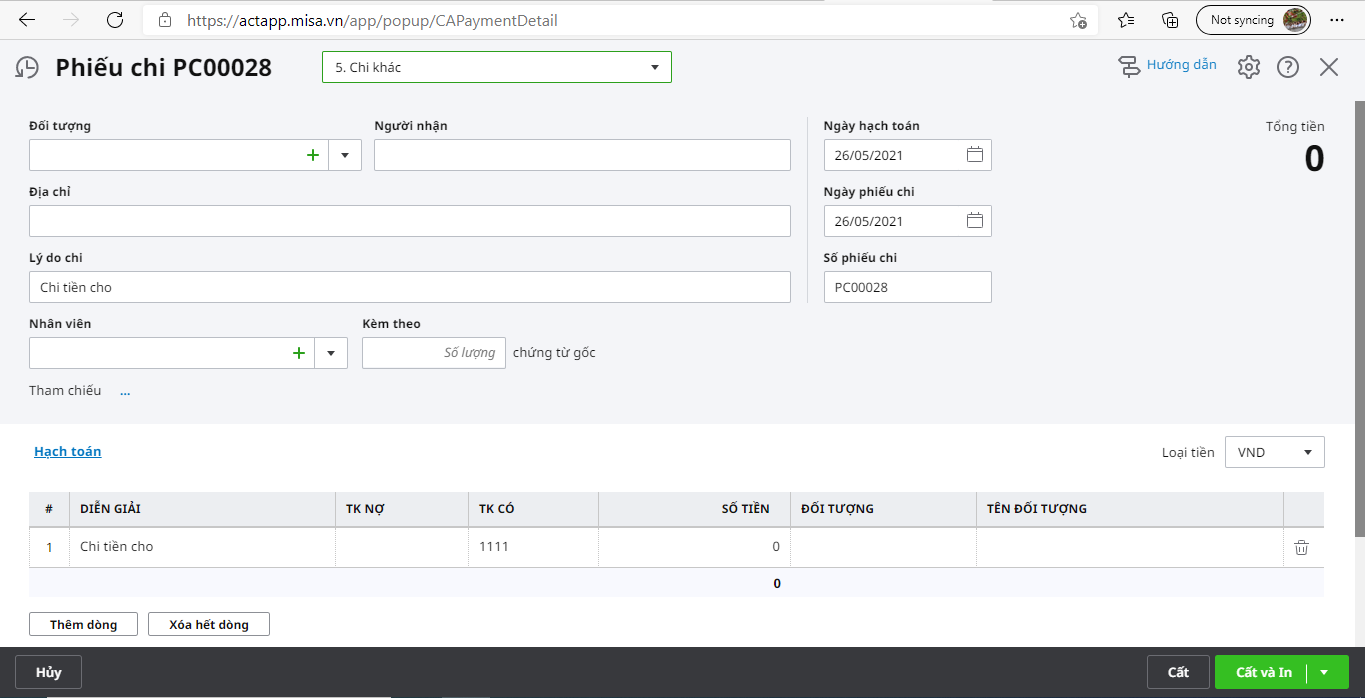
* Giải pháp phần mềm
  + Danh sách hiển thị dạng hình cây phân cấp không phân trang. Sắp xếp theo thứ tự tăng dần TK. Bao gồm các cột như thiết kế. Hiển thị tổng số bản ghi ở chân trang
  + Cột chức năng có các chức năng: Sửa, Xóa
  + Nút Thêm sẽ mở form thêm mới Tài khoản
    - Chi tiết - Form thêm mới Tài khoản



* Giải pháp phần mềm
  + Một số lưu ý
  + Form mở lên sẽ hiển thị ở bên phải và có thể co dãn full màn hình
  + Đảm bảo giao diện ẩn hiện như form demo

1. **Thi công tab Thu chi, tiền**
   1. **Danh sách**
      * **Giải pháp phần mềm**

* Thiết kế: Như trên
* Danh sách bao gồm các cột: Ngày hạch toán, Ngày chứng từ, Số chứng từ, Diễn giải, Số tiền, Đối tượng, Mã Đối tượng, Địa chỉ, Chức năng
* Cho phép tìm kiếm theo thông tin số chứng từ, đối tượng, mã đối tượng
* Thực hiện được các chức năng ở cột chức năng bao gồm Xem, Xóa, Nhân bản
* Danh sách cho phép phân trang. Mặc định là 20 bản ghi trên 1 trang. Cho phép chọn các tùy chọn 10, 20, 50, 100 bản ghi trên 1 trang
* Nút Thêm chi tiền: mở phiếu chi tiền mặt tương ứng
  1. **Chi tiết - Phiếu chi tiền mặt**

****

**Mục tiêu**: Kế toán tiền mặt muốn lập phiếu chi tiền khác như chi tiền mua dịch vụ điện, nước, chi để ký quỹ, ký cược, chi tạm ứng cho nhân viên

**Phạm vi - Phiếu chi tiền mặt loại Chi Khác**

**Giải pháp phần mềm, chức năng/nghiệp vụ bị ảnh hưởng:**

1. Số phiếu chi:
   1. Không được bỏ trống
   2. Tự động tăng
   3. Khi cất chứng từ: Nếu số chứng từ bị trùng phải đưa cảnh báo cho người dùng lựa chọn.
      1. Nếu người dùng chọn hủy bỏ: không lưu dữ liệu Database.
      2. Nếu người dùng muốn cất chứng từ: Gợi ý tự động tăng số chứng từ, chỉ cất dữ liệu khi số chứng từ không bị trùng.
   4. Mặc định khi thêm số chứng từ được bind sẵn, người dùng có thể nhập số chứng từ khác nhưng phải đảm bảo theo quy tắc trên.
2. Ngày hạch toán
   1. Mặc định bằng ngày hiện tại
3. Ngày phiếu chi
   1. Ngày chứng từ: mặc định bằng ngày hạch toán.
   2. Ngày chứng từ <= ngày hạch toán.
   3. Quy tắc lưu ý:
      1. Khi thêm mặc định ngày hạch toán = ngày chứng từ.
      2. Khi thêm nếu ngày hạch toán=ngày chứng từ
         1. Sửa ngày hạch toán =>ngày chứng từ tự động cập nhập theo.
         2. Sửa ngày chứng từ =>ngày hạch toán không thay đổi.
      3. Nếu ngày hạch toán khác ngày chứng từ=>Sửa ngày hạch toán không ảnh hưởng đến ngày chứng từ và ngược lại.
4. Tổng tiền
   1. Khi thêm mặc định=0.
   2. Tổng tiền= Tổng tiền dưới detail
5. Combo đối tượng
   1. Lấy dữ liệu trên danh mục đối tượng từ bảng account\_object
   2. Tích chọn hiển thị bảng: mã đối tượng, tên đối tượng, địa chỉ, mã số thuế, điện thoại.
   3. Load danh sách là khách hàng, nhà cung cấp
   4. Giá trị hiển thị trên combo là Tên đối tượng
   5. Khi chọn mã đối tượng tự động bind dữ liệu ra các dòng khác có liên quan:
      1. Dòng địa chỉ: bind ra địa chỉ cá nhân hay tổ chức, tương ứng.
      2. Trên grid:
         1. Cột đối tượng trên grid mặc định bind theo đối tượng trên master.
         2. Cột tên đối tượng trên grid mặc định bind theo giá trị "tên đối tượng" trên combo đối tượng.
         3. Người nhận: Người nhận được lưu trong danh mục
         4. Nếu tên đối tượng là cá nhân==>Người nhận mặc định bind theo tên đối tượng trong combo đối tượng.
         5. Nếu tên đối tượng là tổ chức==>Người nhận bind là người đại diện cho tổ chức đó.
         6. Người nhận người dùng có thể nhập bằng tay.
6. Lý do chi:
   1. Chi tiền cho <tên đối tượng>. VD: Chi tiền cho Công ty TNHH Hoa Thanh.
   2. Khi thêm lần đầu mặc định giá trị hiển thị là: chi khác
   3. Dòng diễn giải: Bind dữ liệu diễn giải tương ứng.
7. Trên grid:
   1. Cột diễn giải:
      1. Mặc định bind giá trị theo giá trị diễn giải trên master
      2. Nếu thay đổi giá trị diễn giải master ==> giá trị diễn giải trên grid thay đôi theo
      3. Ngược lại thay đổi giá trị diễn giải trên grid==>giá trị master không đổi.
      4. Trường hợp các giá trị hiện diễn giải master và grid không bằng nhau, thay đổi giá trị trên master==>giá trị master sẽ không bị thay đổi.
   2. TK Có: mặc định 1111
   3. Combo Đối tượng: load tương tự trên Master
   4. Quy tắc thêm dòng, xóa dòng:
      1. Mặc định khi thêm phiếu chi hiện thị một dòng chi tiết ban đầu.
      2. Khi thêm dòng chi tiết tiết tiếp theo, thông tin các trường sẽ được bind mặc định theo dòng chi tiết gần nhất đã thêm
      3. Xóa dòng có thể xóa lần lượt hoặc xóa hết
8. Kèm theo chỉ cho nhập số (>0).
9. Quy tắc về phần chức năng dưới phần footer:
   1. Mặc định tab combo chức năng gồm một số tùy chọn: Cất và thêm, Cất và đóng.
   2. Mặc định ban đầu khi thêm là: Cất và đóng. Các lần thêm tiếp theo chức năng combo hiển thị sẽ là chức năng người dùng lựa chọn trước đó.

//tạo từng accounting

//for (int i = 0; i < accoutings.Count; i++)

//{

// Accouting accouting = accoutings[i];

// accouting.Idaccounting = Guid.NewGuid();

// accouting.Idpayment = payment.Idpayment;

// \_context.Accoutings.Add(accouting);

// try

// {

// await \_context.SaveChangesAsync();

// }

// catch (DbUpdateException)

// {

// if (AccoutingExists(accouting.Idaccounting))

// {

// return Conflict();

// }

// else

// {

// throw;

// }

// }

// //CreatedAtAction("GetAccouting", new { id = accouting.Idaccounting }, accouting);

//}

{

"paymentobjectcode": "111",

"paymentobjectname": "222",

"paymentaddress": "hanoi",

"reasonpay": "chi tiền",

"paymentemployeecode": "nv34",

"numberoflicense": 0,

"accountingdate": "2021-06-16T08:53:16.013Z",

"paymentdate": "2021-06-16T08:53:16.013Z",

"typeofmoney": "VND",

"totalmoney": 0,

"paymentnumber": "PC231",

"idpayment": "3fa85f64-5717-4562-b3fc-2c963f66afa6",

"accoutings": [

{

"description": "cho cho ai",

"accountdebtnumber": "111",

"accountreceivenumber": "123",

"money": 0,

"objectcode": "cv23",

"objectname": "anh ánh",

"idaccounting": "3fa85f64-5717-4562-b3fc-2c963f66afa4",

"idpayment": "43f3b670-82fa-4650-9917-f2659be4853e"

},

{

"description": "cho cho ai",

"accountdebtnumber": "111",

"accountreceivenumber": "123",

"money": 15,

"objectcode": "cj5678",

"objectname": "cgv",

"idaccounting": "3fa85f64-5717-4562-b3fc-2c963f66afa5",

"idpayment": "43f3b670-82fa-4650-9917-f2659be4853e"

}

]

}

//lấy thông tin phiếu chi

// GET: api/Payments/paging?pageNumber=1&pageSize=2

[HttpGet("paging")]

public async Task<ActionResult<IEnumerable<Payment>>> GetPaymentsPaging(int pageNumber, int pageSize)

{

return await \_context.Payments.OrderBy(x => x.Paymentnumber).Skip((pageNumber - 1) \* pageSize).Take(pageSize).ToListAsync();

}

// số bản ghi phiếu chi

// GET: api/Payments/length

[HttpGet("length")]

public IActionResult GetLengthPayments()

{

var rowEffects = \_context.Payments.Count();

return Ok(rowEffects);

}

// tìm kiếm phiếu chi theo số chứng từ, đối tượng, mã đối tượng

// GET: api/Payments/search?keyword=nv&pageNumber=1&pageSize=10

[HttpGet("search")]

public async Task<ActionResult<IEnumerable<Payment>>> SearchPayments(string keyword, int pageNumber, int pageSize)

{

keyword = keyword.ToLower();

var payments = await \_context.Payments.Where(

p => p.Paymentnumber.ToLower().Contains(keyword) ||

p.Paymentobjectcode.ToLower().Contains(keyword) ||

p.Paymentobjectname.ToLower().Contains(keyword)).OrderBy(x => x.Paymentnumber).Skip((pageNumber - 1) \* pageSize).Take(pageSize).ToListAsync();

return payments;

}

// số bản ghi phiếu chi trong tìm kiếm

// GET: api/Payments/lengthSearch?keyword=nv

[HttpGet("lengthSearch")]

public IActionResult GetLengthSearchPayments(string keyword)

{

keyword = keyword.ToLower();

var rowEffects = \_context.Payments.Where(

p => p.Paymentnumber.ToLower().Contains(keyword) ||

p.Paymentobjectcode.ToLower().Contains(keyword) ||

p.Paymentobjectname.ToLower().Contains(keyword)).Count();

return Ok(rowEffects);

}