BÀI TẬP LỚN

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

ĐỀ TÀI: Electrical circuit simulator

**Giáo viên hướng dẫn: Nguyễn Thị Thu Trang**

**Mã lớp: 151965 – Nhóm 15**

**Sinh viên thực hiện:**

| **STT** | **Họ và tên** | **MSSV** | **Email** |
| --- | --- | --- | --- |
| **1** | Thân Cát Ngọc Lan | 20225646 |  |
| **2** | Bùi Thành Long | 20225874 | long.bt225874@sis.hust.edu.vn |
| **3** | Mai Huy Long | 20225735 |  |
| **4** | Phạm Xuân Long | 20225648 | long.px225645@sis.hust.edu.vn |
| **5** | Phan Hoàng Long | 20225738 |  |

**PHÂN CÔNG THÀNH VIÊN TRONG NHÓM**

| **STT** | **Họ và tên** | **MSSV** | **Vai trò** | **Công việc** |
| --- | --- | --- | --- | --- |
| **1** | Thân Cát Ngọc Lan | 20225646 | Thành viên | Dựng biểu đồ lớp, lên ý tưởng logic hệ thống |
| **2** | Bùi Thành Long | 20225874 | Trưởng nhóm | Xây dựng logic hệ thống, viết báo cáo |
| **3** | Mai Huy Long | 20225735 | Thành viên | Xây dựng UI hệ thống |
| **4** | Phạm Xuân Long | 20225648 | Thành viên | Viết báo cáo |
| **5** | Phan Hoàng Long | 20225738 | Thành viên | Lên ý tưởng & xây dựng biểu đồ use case, làm slide thuyết trình |

**ĐẶC TẢ YÊU CẦU BÀI TOÁN**

*Đặc tả yêu cầu*

* 1. *Miêu tả yêu cầu bài toán.*
* Ứng dụng yêu cầu mô phỏng mạch RLC, giúp người dùng thiết kế và kiểm tra các mạch có chứa thành phần điện trở (R), cuộn dây (L), và tụ điện (C). Người dùng có thể xem và kiểm tra đáp ứng tần số, phản hồi mạch, và các thông số quan trọng của mạch.
  1. *Mục tiêu*
* Người dùng có thể thiết kế mạch logic bằng cách kéo thả các thành phần.
* Tính toán giá trị điện thế, cường độ của mạch và của từng thiết bị của mạch.
  1. *Yêu cầu chi tiết*
     1. *Giao diện người dùng*
        1. *Bảng thiết kế mạch*
     + Các thành phần R, L, C nút nguồn, và dây kết nối giữa các thành phần.
       1. *Các thành phần*
     + **Điện trở (R)**: Cho phép người dùng nhập giá trị.
     + **Cuộn dây (L)**: Cho phép người dùng nhập giá trị độ tự cảm (L).
     + **Tụ điện (C)**: Cho phép người dùng nhập giá trị dung kháng (C).
     + **Nút nguồn và tải**: Tạo nguồn cung cấp điện và tải kết nối đầu vào/đầu ra.
     1. *Tính năng chính*
     + **Thiết kế mạch**: Người dùng kéo thả các thành phần và kết nối chúng theo logic mong muốn.
     + **Kiểm tra logic**: Ứng dụng tính toán đầu ra dựa trên đầu vào đã được đặt trong mạch.
     1. *Yêu cầu kĩ thuật*
        1. *Giao diện người dùng*
     + Giao diện desktop.
     + Tối ưu hóa tốc độ xử lý, tương thích với thiết bị.
       1. *Xử lý login và đầu ra*
     + Thuật toán tính toán các thông số RLC như tần số cộng hưởng, Q-factor, đáp ứng tần số.

*Biểu đồ use case*

1. *Biểu đồ use case tổng quan*

* Tác nhân: User, System
* *Các use case với tác nhân User:*

+ Create/Remove elements

+ Run circuit

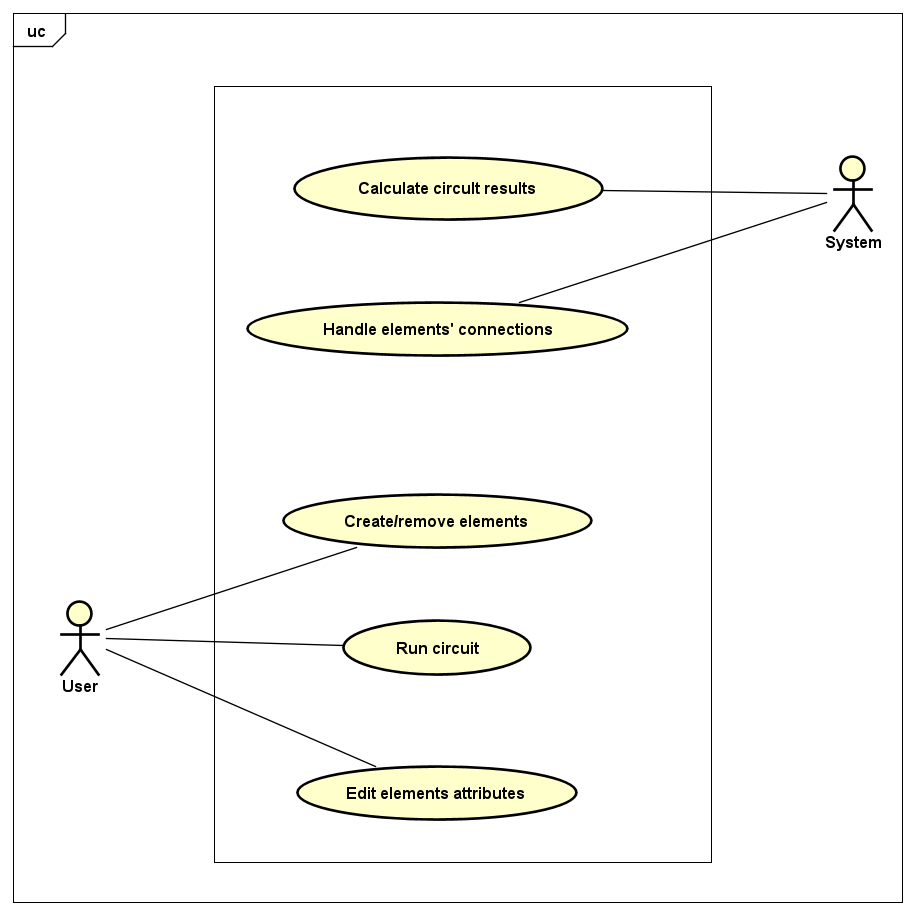
+ Edit elements attributes

* *Các use case với tác nhân System:*

+ Calculate circuit results

+ Handle elements’ connections

* *Hình ảnh cho usecase*

**

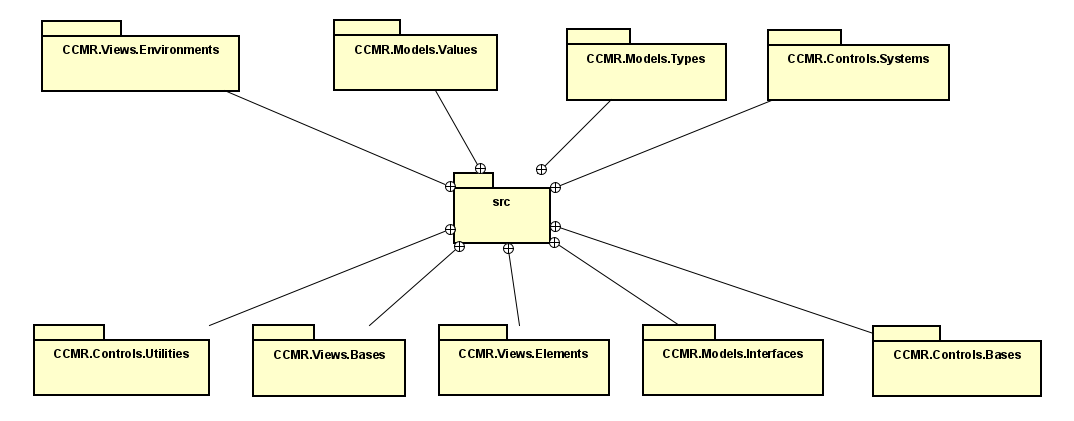
1. *Đặc tả use case*

| **Điều kiện trước** | Người dùng khởi động simulator | |
| --- | --- | --- |
| **Tác nhân** | Người dùng chương trình | |
| **Luồng thực thi chính** | | |
| **No.** | **Thực hiện** | **Hành động** |
| 1 | User | Tạo mới thiết bị |
| 2 | System | Hiện thị thiết bị lên bảng mạch |
| 3 | User | Điền các thông số của thiết bị |
| 4 | User | Kết nối các thiết bị |
| 5 | User | Chạy mạch |
| 6 | System | Xử lý logic hiển thị thông tin bảng mạch |

**PHÂN TÍCH THIẾT KẾ BÀI TOÁN**

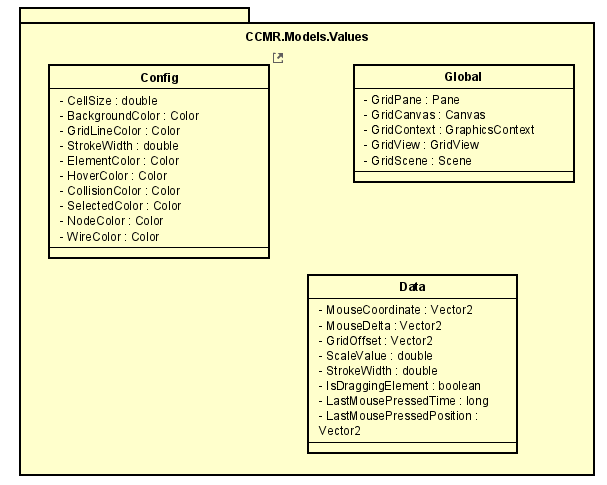
*Class diagram*

1. *Biểu đồ lớp tổng quan:*

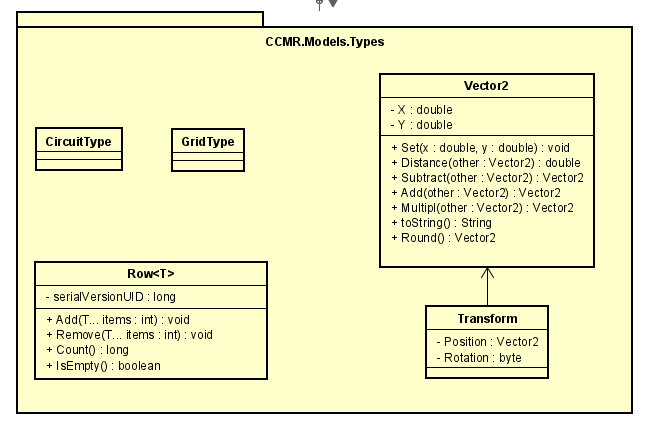
**

* Gói CCMR.Views.Environments: Chứa các đối tượng UI liên quan tới không gian và các thiết bị mạch.
* Gói CCMR.Models.Interfaces: Gồm các mã về interface.
* Gói CCMR.Controls.Utilities:
* Gói CCMR.Models.Types: Chứa các kiểu dữ liệu/đối tượng.
* Gói CCMR.Controls.Elements: Chứa các thành phần mạch.
* Gói CCMR.Controls.Systems: Chứa các đối tượng điều khiển/quản lí.
* Gói CCMR.Models.Values: Chứa các giá trị/tham chiếu toàn cục .trong hệ thống.
* Gói CCMR.Views.Elements: chứa UI của các thiết bị mạch.
* Gói CCMR.Views.Bases: Chứa các object trừu tượng cho các thành phần UI khác.

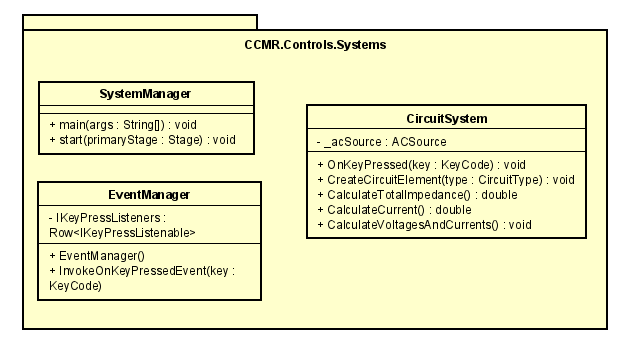
1. *Biểu đồ cho các gói*
   1. *Gói CCMR.Models.Values*

**

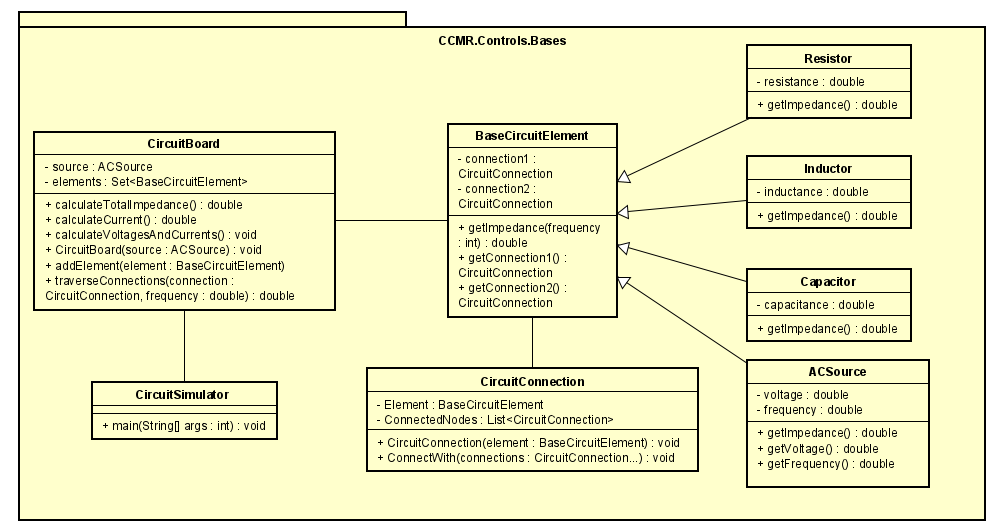
* Class Config: lớp static chứa các giá trị mặc định của hệ thống.
* Class Data: lớp static chứa các giá trị lưu động của hệ thống.
* Class Global: lớp static chứa tham chiếu tới singleton của các đối tượng duy nhất trong hệ thống.
  1. *Gói CCMR.Models.Types*

**

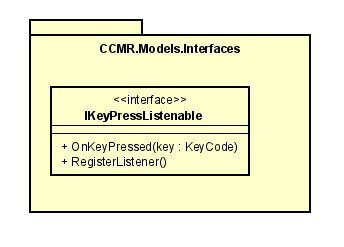
* Enum CircuitType: là các kiểu loại của thiết bị mạch.
* Enum GridType: là các kiểu dạng lưới của GridView (UI).
* Row: lớp generic đại diện cho một danh sách các đối tượng với các chức năng cơ bản như Add(), Remove(), Count().
* Transform: là lớp đại diện cho vị trí và góc xoay của UI của các thiết bị.
* Vector2: là lớp đại diện cho kiểu dữ liệu vector có 2 giá trị double X và Y.
  1. *Gói CCMR.Controls.Systems*

**

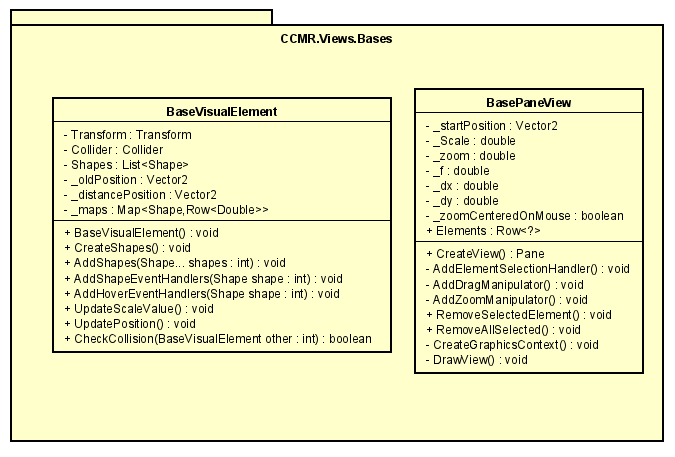
* Class CircuitSystem: là lớp đại diện cho hệ thống mạch, có khả năng thêm các thiét bị cũng như tính toán các giá trị cần thiết.
  1. *Gói CCMR.Controls.Bases*

**

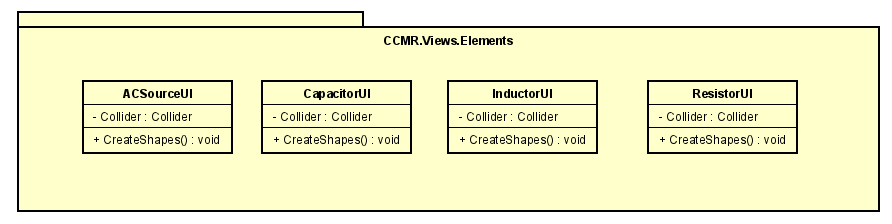
* Class BaseCircuitElement: là lớp trừu tượng định nghĩa cho các thiết bị mạch trong hệ thống.
* Class CircuitConnection: là đối tượng đại diện cho các đầu kết nối giữ thiết bị này với thiết bị khác.
  1. *Gói CCMR.Models.Interfaces*

**

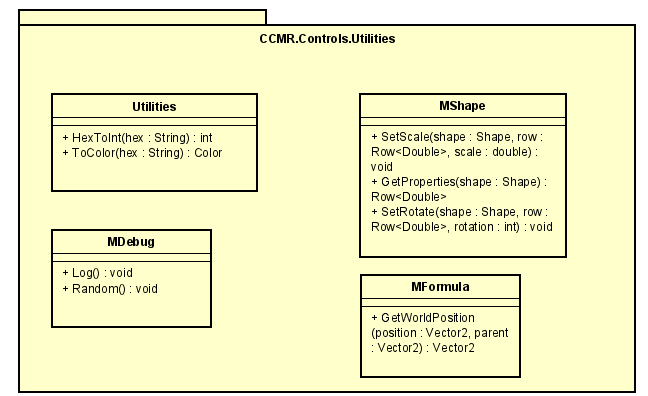
* Interface IKeyPressListenable: cung cấp khả năng lắng nghe sự kiện khi có phím keyboard được nhấn.
* Interface ISelectable: cung cấp khả năng chọn/bỏ chọn đối với các đối tượng UI.
  1. *Gói CCMR.Views.Bases*

**

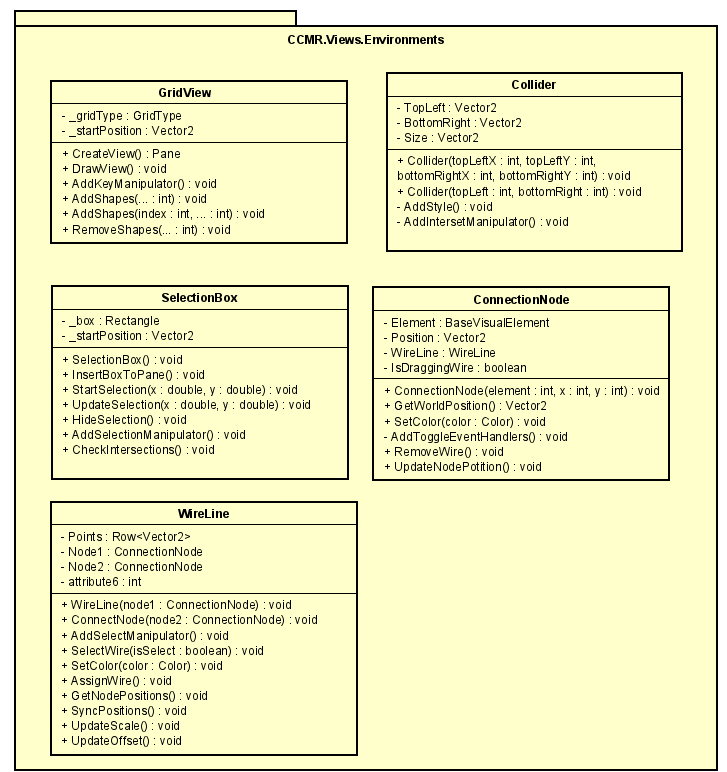
* Class BasePaneView: là lớp trừu tượng dùng để tạo dựng lên các UI Pane phức tạp hơn (GridView) với các phương thức để tạo UI và tương tác với môi trường.
* Class BaseVisualElement: là lớp trừu tượng dùng để tại dựng lên các UI Element cho các thiết bị mạch với các phương thức tạo dựng UI, cập nhật vị trí và tương tác với môi trường.
  1. *Gói CCMR.Views.Elements*

**

* 4 class ACSource, Capacitor, Inductor, Resistor: Kế thừa từ lớp BaseCircuitElement, đại diện cho các thiết bị mạch trong hệ thống, chứa các giá trị riêng và tính toán, trả về các giá trị tương ứng.
  1. *Gói CCMR.Controls.Utilities*

**

* Class MDebug: Chứa các hàm tiện ích dùng để in log ra console của IDE.
* Class MFormula: Chứa các hàm tiện ích phục vụ cho việc tính toán trong hệ thống.
* Class MShape: chứa các hàm tiện ích tác động tới các đối tượng Shape trong UI.
  1. *Gói CCMR.Views.Environments*

**

* Class Collider: kế thừa từ Rectangle, đại diện cho viền ngoài của UI, có khả năng phát hiện va chạm với các Collider khác khi được kéo thả chống chéo lên nhau.
* Class ConnectionNode: kế thừa từ Circle, đại diện cho các đầu kết nối của UI, có khả năng kết nối với các ConnectionNode khác khi được chọn.
* Class GridView: Kế thừa từ BasePaneView, là UI của background lưới trong hệ thống, có khả năng kéo thả, phóng to thu nhỏ theo input từ chuột.
* Class SelectionBox: là UI cho hộp chọn, sử dụng khi kéo thả trên màn hình để chọn nhiều thiết bị cùng lúc.
* Class WireLine: Kế thừa từ Polyline, là UI của các đường dây dẫn nối giữa các thiết bị mạch.

*Các design pattern được sử dụng*

***Singleton***  - Mô tả: Sử dụng tính chất của 'static' để đảm bảo trong hệ thống chỉ có duy nhất một instance của đối tượng.  
 - Ưu điểm:  
   + Đảm bảo chỉ có duy nhất một instance của đối tượng trong toàn bộ hệ thống.  
   + Làm giảm thiểu lượng tham chiếu chéo trên các đối tượng, đảm bảo tính 'clean code' của hệ thống.  
 - Nhược điểm:   
   + Khó kiểm tra và bảo trì do singleton làm giảm khả năng unit test của hệ thống.  
   + Vi phạm nguyên tắc 'Single Responsibility' của SOLID.  
 - Ứng dụng: Pattern này được sử dụng ở CCMR.Models.Values.Global để lưu trữ tham chiếu tới các đối tượng độc nhất trong hệ thống.

***Observer***- Mô tả: Sử dụng một đối tượng (Publisher) làm trung gian để thông báo một sự kiện tới các đối tượng theo dõi (Listener).  
 - Điểm mạnh:   
   + Giảm thiểu sự phụ thuộc giữa các đối tượng mà vẫn đảm bảo được các đối tượng đó có thể kích hoạt hành động khi có một sự kiện xảy ra.  
   + Dễ dàng quản lý các đối tượng nào được cho phép nhận sự kiện hay không do ốc thể dễ dàng thêm hoặc bớt chúng vào danh sách listener.  
   + Khả năng mở rộng hệ thống tốt do không tạo sự gò bó và độc lập với các phần khác của hệ thống.  
 - Điểm yếu  
   + Khó kiểm soát thứ tự thông báo sự kiện tới các đối tượng lắng nghe.  
   + Không thể hủy/đảo ngược thông báo.  
   + Tiêu tốn bộ nhớ nếu không được quản lý tốt.  
 - Ứng dụng: Pattern này được sử dụng ở CCMR.Models.Interfaces.IKeyPressListenable (interface cho listener) và CCMR.Controls.Systems.EventManager (đối tượng thông báo).