

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



MẪU THIẾT KẾ

Người hướng dẫn: **GV. Nguyễn Thanh Phước**

Người thực hiện: **Phạm Trung Hoài Sơn – 518H0261**

Bùi Vũ Minh Hiếu – 518H0496

Võ Hùng Anh – 518H0472

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



MẪU THIẾT KẾ

Người hướng dẫn: **GV. Nguyễn Thanh Phước**

Người thực hiện: **Phạm Trung Hoài Sơn – 518H0261**

Bùi Vũ Minh Hiếu – 518H0496

Võ Hùng Anh – 518H0472

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Tụi em cảm ơn thầy đã nhiệt tình chỉ dạy cho em , để tụi em có thể hoàn thành môn học một cách xuất sắc .

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của TS Nguyễn Văn A. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Phạm Trung Hoài Sơn

Bùi Vũ Minh Hiếu

Võ Hùng Anh

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

MỤC LỤC

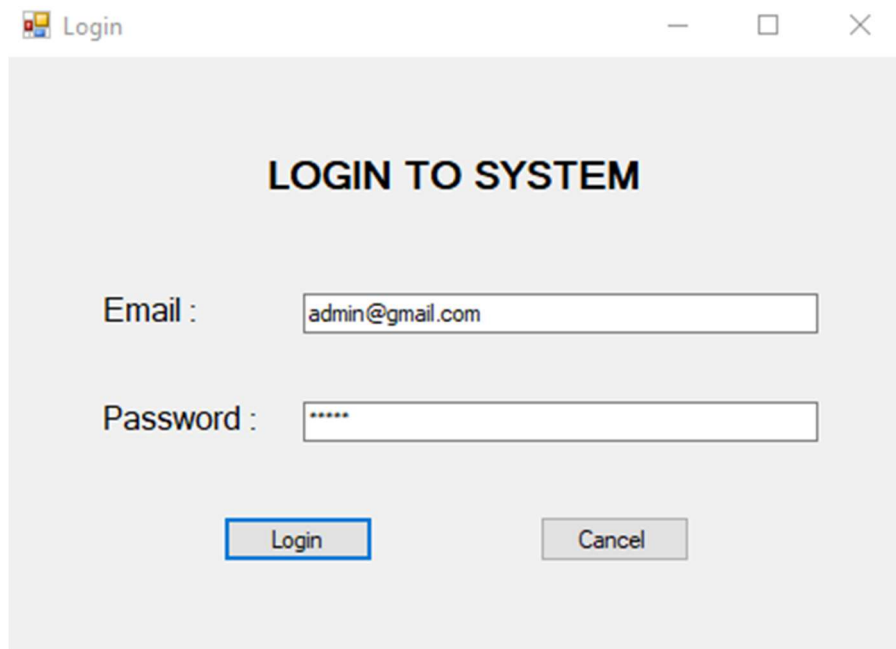
LỜI CẢM ƠN	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	iii
MỤC LỤC.....	1
1) Mô tả	2
2) Giao diện người dùng	2
3) Áp dụng design pattern.....	5
3.1) Singleton Pattern	5
3.2) Strategy Pattern.....	6
3.3) Factory Method Pattern.....	9
3.4) Command Pattern	10
4) Rubric.....	18

1) Mô tả

- Ứng dụng điều khiển các thiết bị trong gia đình là một ứng dụng desktop.
- Mục đích xây dựng ứng dụng : Để có thể giúp người dùng điều khiển các thiết bị trong gia đình một cách thuận tiện hơn, hiệu quả hơn.

2) Giao diện người dùng

- Màn hình đăng nhập



Mật khẩu được hiển thị dưới dạng ký tự *

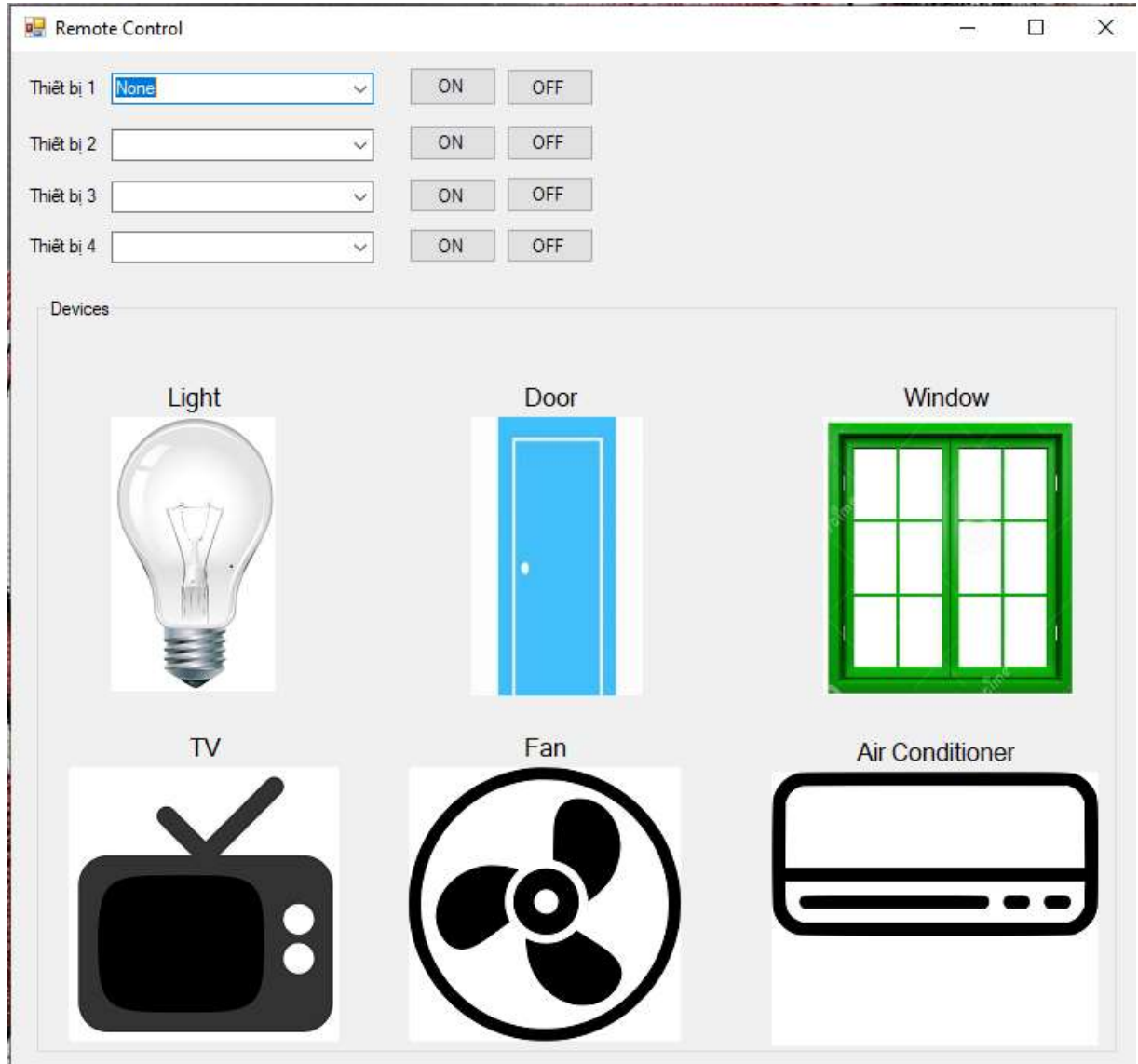


Email or Password is invalid.

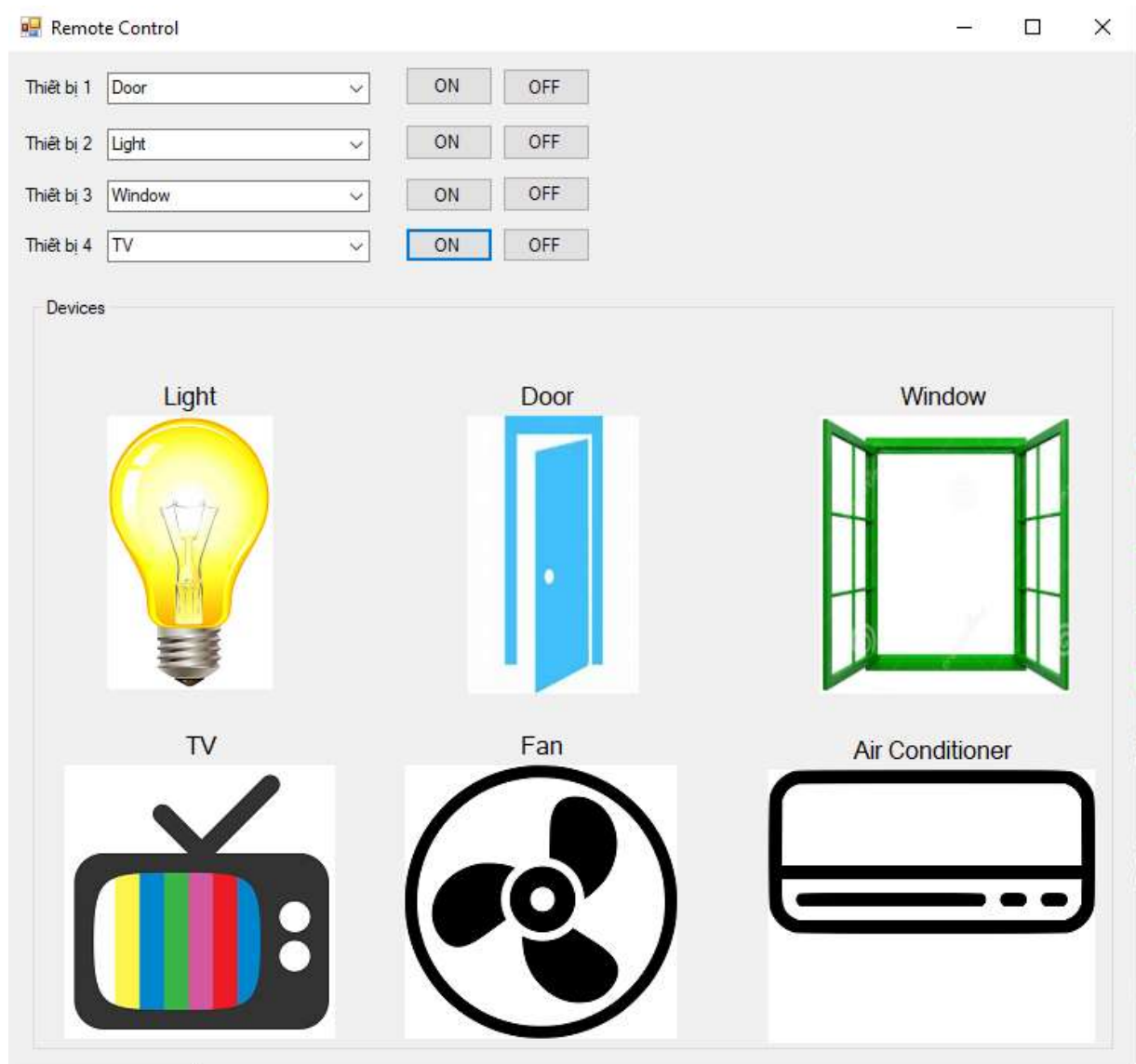
OK

Sẽ hiển thị thông báo lỗi nếu chúng ta nhập sai email hoặc password

- Màn hình điều khiển của ứng dụng:



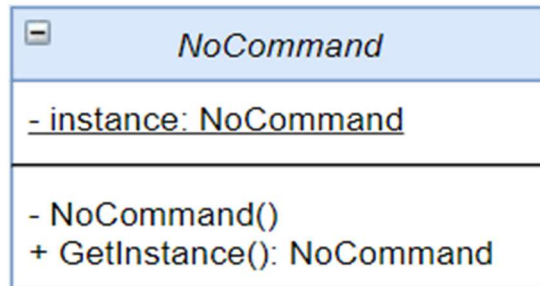
- Các thiết bị khi được bật



3) Áp dụng design pattern

3.1) Singleton Pattern

- Class Diagram



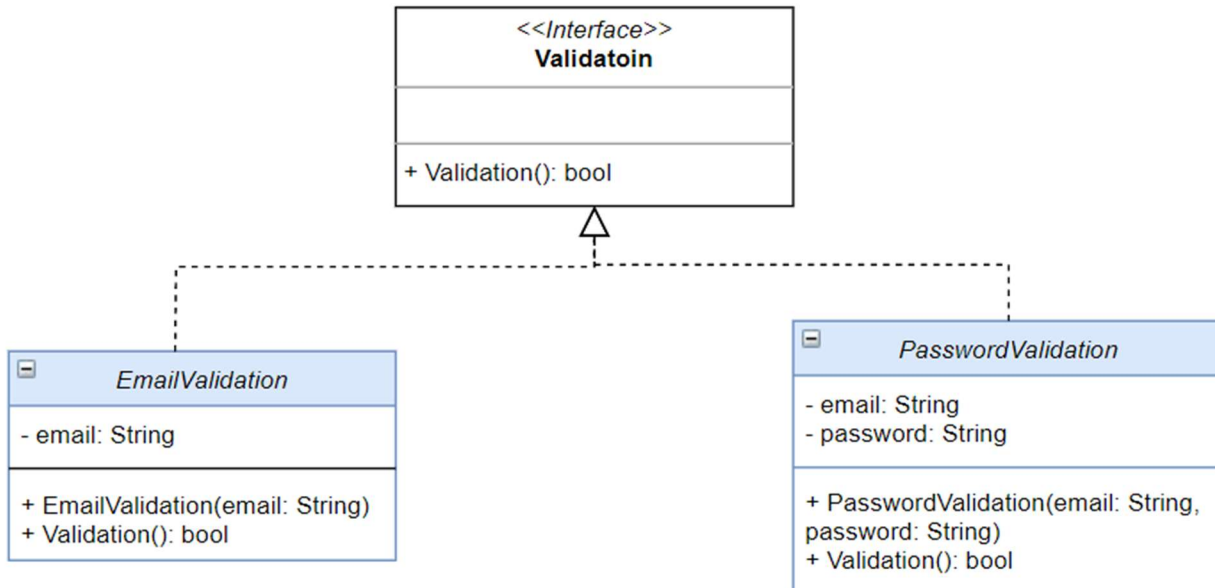
```

public class NoCommand : ICommand
{
    private static NoCommand Instance = new NoCommand();
    private NoCommand() {}
    public static NoCommand GetInstance()
    {
        return Instance;
    }
    public void Execute()
    {
        MessageBox.Show("Please choose device in this slot");
    }
}
  
```

- Khi không chọn một thiết bị nào thì chỉ cần khởi tạo một lần duy nhất (khi không làm gì thì chỉ có hành động duy nhất giống nhau).

3.2) Strategy Pattern

- Class diagram:



- Cho phép chúng ta thay đổi dễ dàng khi thêm những phần tử mới.

```

interface Validation
{
    public Boolean Validation();
}
  
```

```

public class EmailValidation : Validation
{
    private String email;
    private Regex regex = new Regex("^\\w+@\\w+\\.\\w+$");
    public EmailValidation(String email)
    {
        this.email = email;
    }
}
  
```

```

    }
    public bool Validation()
    {
        if (regex.Match(email).Success)
        {
            return true;
        }
        return false;
    }
}

```

```

public class PasswordValidation : Validation
{
    private String password;
    private String email;

    public PasswordValidation(String email, String password)
    {
        this.email = email;
        this.password = password;
    }
    public bool Validation()
    {
        string connectionString =
        "datasource=127.0.0.1;port=3306;username=root;password=;database=dptdt
u;";
    }
}

```

```

    string query = "select * from accounts where email = '" + email + "' and
password = '" + password + "'";

    MySqlConnection databaseConnection = new
    MySqlConnection(connectionString);

    MySqlCommand commandDatabase = new MySqlCommand(query,
    databaseConnection);

    commandDatabase.CommandTimeout = 60;

    MySqlDataReader reader;

    try
    {
        databaseConnection.Open();
        Debug.WriteLine("Connect success !");
        reader = commandDatabase.ExecuteReader();
        if (reader.HasRows)
        {
            return true;
        }
        else
        {
            Debug.WriteLine("No rows found.");
            return false;
        }
        databaseConnection.Close();
    }
    catch (Exception ex)
    {

```

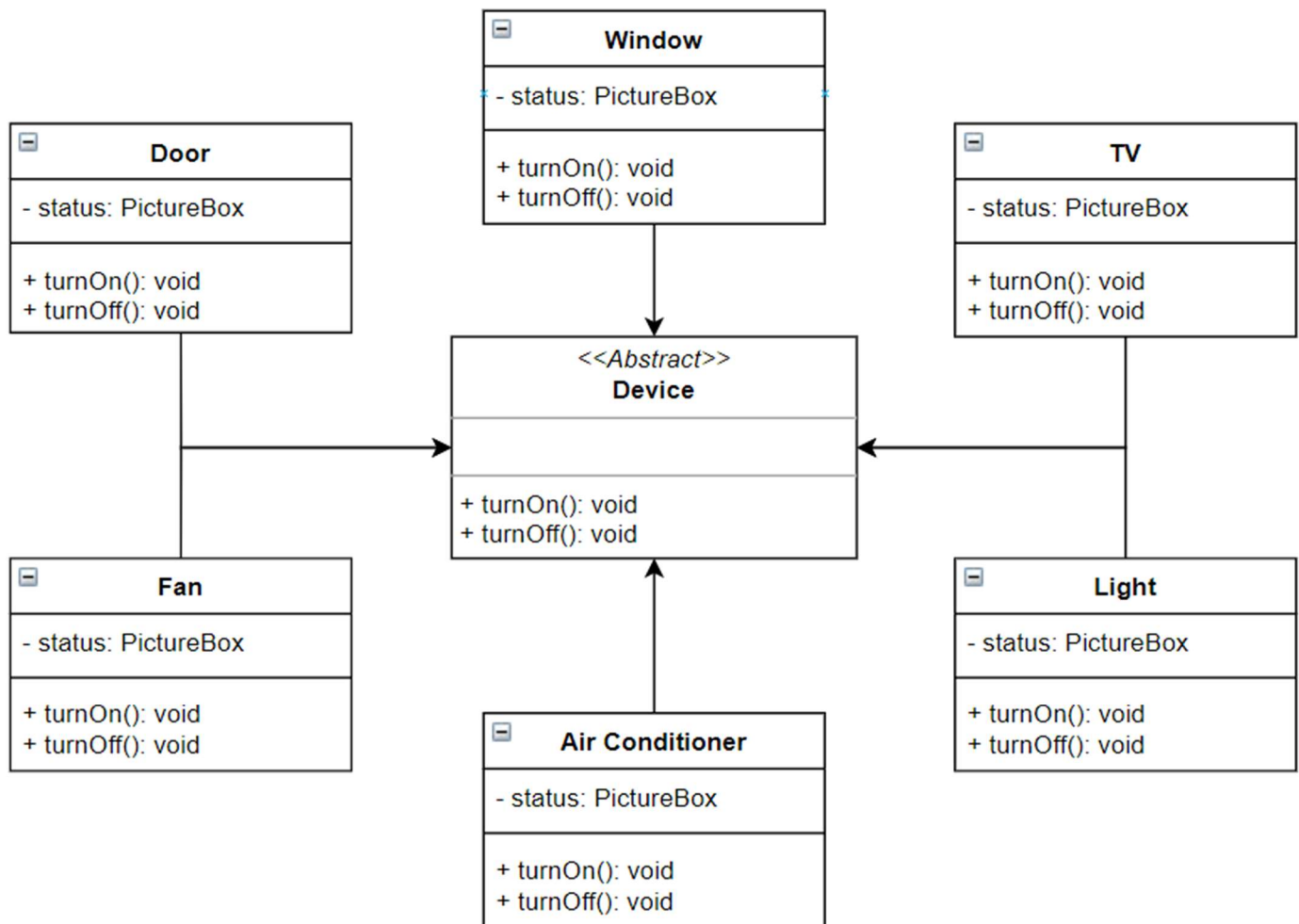
```

        Debug.WriteLine(ex.Message);
        return false;
    }
}

```

3.3) Factory Method Pattern

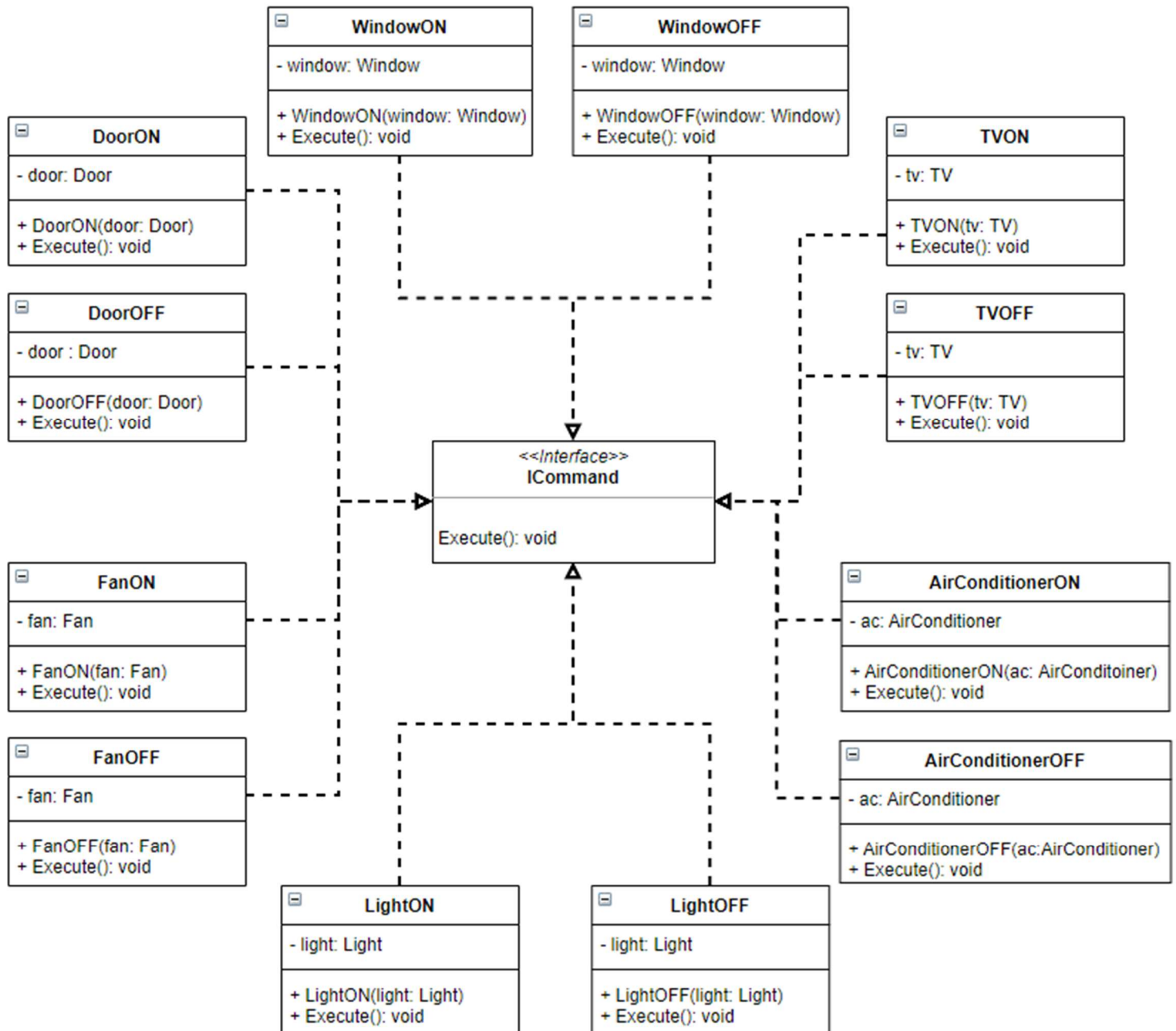
- Class diagram:



- Giúp cho việc khởi tạo những thiết bị mới một cách dễ dàng .

3.4) Command Pattern

- Class diagram:



- Gửi yêu cầu đến các thiết bị để thực hiện các hành động theo yêu cầu đó vào các thời điểm khác nhau

```
public class AirConditionerON : ICommand
{
    private AirConditioner ac;

    public AirConditionerON(AirConditioner ac)
    {
        this.ac = ac;
    }

    public void Execute()
    {
        this.ac.On();
    }
}

public class AirConditionerOFF : ICommand
{
    private AirConditioner ac;

    public AirConditionerOFF(AirConditioner ac)
    {
        this.ac = ac;
    }

    public void Execute()
    {
```



```
        this.ac.Off();  
    }  
}
```

```
public class DoorON : ICommand
```

```
{  
    private Door door;  
  
    public DoorON(Door door)  
    {  
        this.door = door;  
    }  
  
    public void Execute()  
    {  
        this.door.On();  
    }  
}
```

```
public class DoorOFF : ICommand
```

```
{  
    private Door door;  
  
    public DoorOFF(Door door)  
    {  
        this.door = door;  
    }  
}
```

```
public void Execute()  
{  
    this.door.Off();  
}  
}
```

```
public class FanON : ICommand  
{  
    private Fan fan;  
  
    public FanON(Fan fan)  
    {  
        this.fan = fan;  
    }  
  
    public void Execute()  
    {  
        this.fan.On();  
    }  
}  
  
public class FanOFF : ICommand  
{  
    private Fan fan;  
  
    public FanOFF(Fan fan)
```

```
{  
    this.fan = fan;  
}  
  
public void Execute()  
{  
    this.fan.Off();  
}  
}
```

```
public class LightON : ICommand  
{  
    private Light light;  
  
    public LightON(Light light)  
    {  
        this.light = light;  
    }  
  
    public void Execute()  
    {  
        this.light.On();  
    }  
}  
  
public class LightOFF : ICommand  
{
```

```
private Light light;

public LightOFF(Light light)
{
    this.light = light;
}

public void Execute()
{
    this.light.Off();
}
}
```

```
public class TVON : ICommand
{
    private TV tv;

    public TVON(TV tv)
    {
        this.tv = tv;
    }

    public void Execute()
    {
        this.tv.On();
    }
}
```

```
public class TVOFF : ICommand
{
    private TV tv;

    public TVOFF(TV tv)
    {
        this.tv = tv;
    }

    public void Execute()
    {
        this.tv.Off();
    }
}
```

```
public class WindowON : ICommand
{
    private Window window;

    public WindowON(Window window)
    {
        this.window = window;
    }
    public void Execute()
    {
        this.window.On();
    }
}
public class WindowOFF : ICommand
{
    private Window window;
    public WindowOFF(Window window)
    {
        this.window = window;
    }
    public void Execute()
    {
        this.window.Off();
    }
}
```

4) Rubric

Phạm Trung Hoài Sơn : 33%

Bùi Vũ Minh Hiếu : 33%

Võ Hùng Anh : 33%

