## 



Elevator Demo

04/24/2024

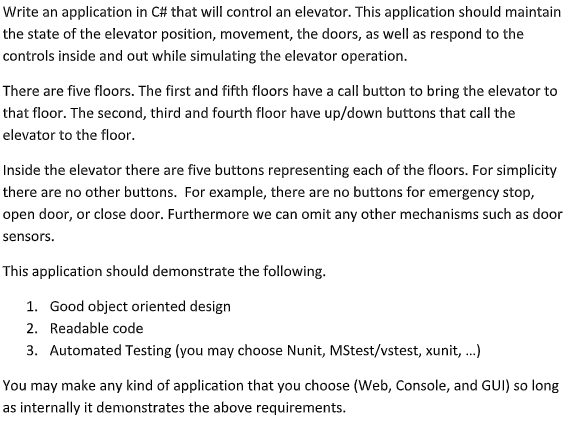
**─**

Alonso Rojas Arango

alonso.rojas.arango@gmail.com

# General Overview

As part of the selection process, a demo was requested to evaluate the theory and practical knowledge of the candidate according to the following requirements:

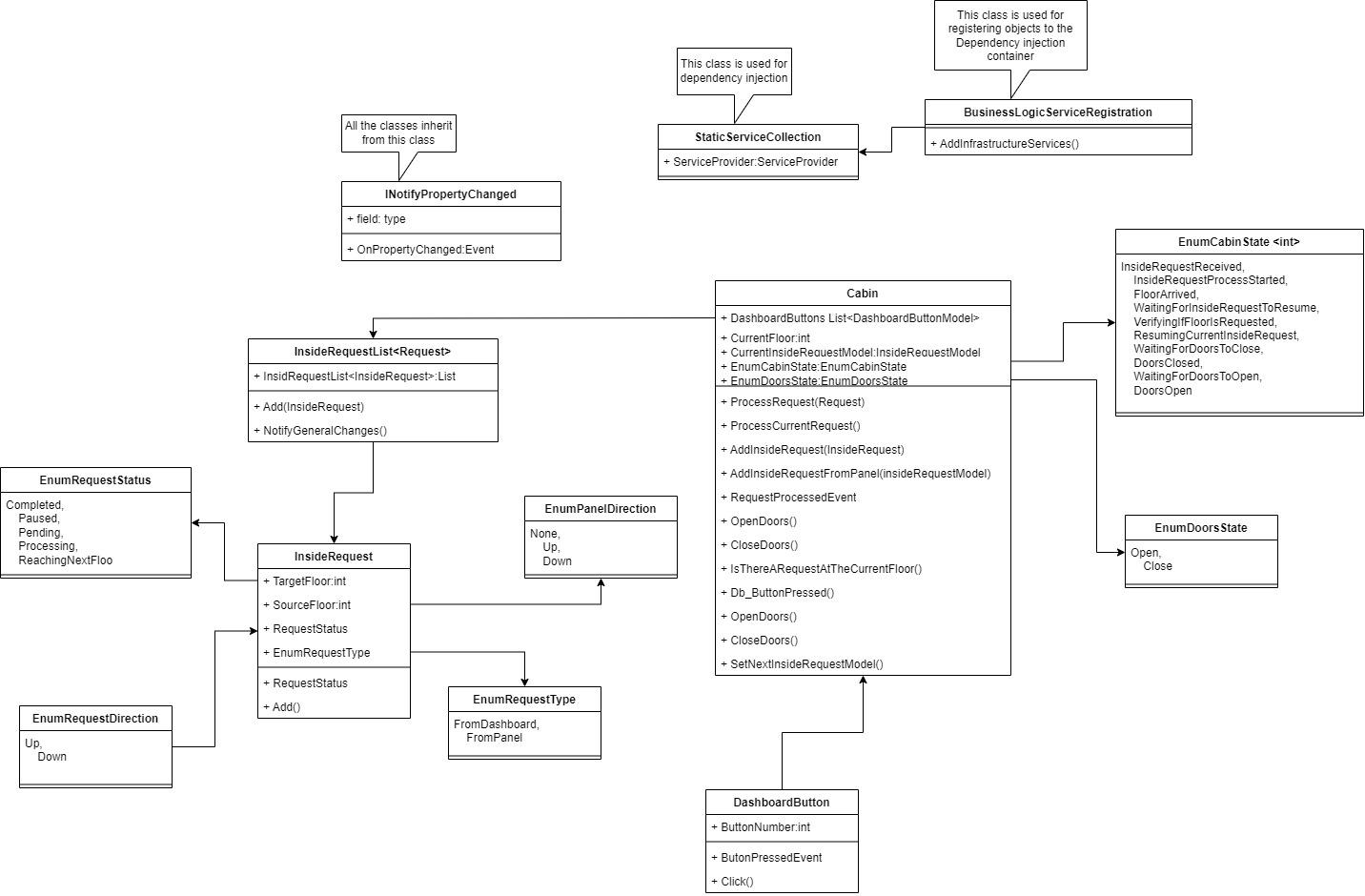


# Objectives

1. Implement the an **object oriented design** approach
2. Create a little project that can demo a **multi layered design**
3. Utilize some Core functionality such as **Dependency Injection**
4. Create a small **front end application to demo the business classes**
5. Demo some uses of front end interacting with the **INotifyProperty** changed object
6. Create the **Unit test project**
7. **Demo hand made dependency injection in the test project**

# Description

Given the aforementioned requirement, a small class diagram was created as a guide

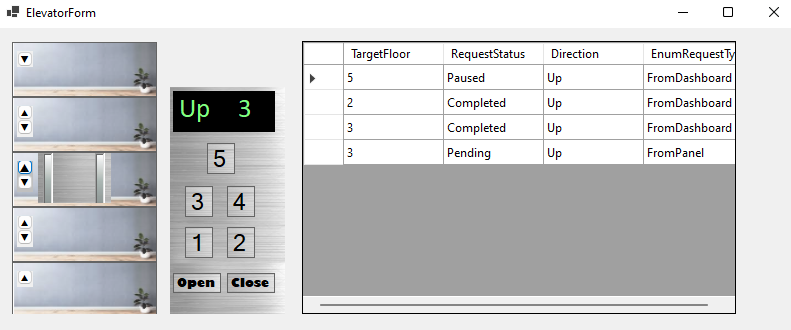


It is important to mention that in the end, the classes were divided into three objects, inheriting from them to demonstrate inheritance. The model classes are inheriting from INotifyPropertyChange and the third class is the Business Layer class that is also inheriting from the model in order not to repeat the same code again or needing to add some references in the BL classes.

The idea of implementing the IProperttyChanged interface is because, at the UI level, the application will be responding and reacting to change notifications, which will allow the UI to refresh the updated model. Most of those notifications are irrelevant at the business layer, but they are essential when updating the UI.

In the UI, a Windows Forms application will be showing the different functionality.

# User Interface Application



The application consists of three important sections:

* The Building Mock
* The Cabin Dashboard
* The Track list

## The Building Mock

On the far left, there is a building simulation with panel buttons on each floor.

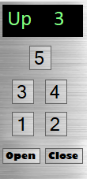
The cabin will be moving with a delayed to demo the most basic Async implementation using await Task.Delay



## The Cabin Dashboard

In the middle, a dashboard has been added to allow you to select the different floors inside the elevator cabin. It also notifies you of the current floor number.

It is important to mention that the Elevator Cabin will need the button Close to be pressed in order to start moving.



## The Track list

And finally, on the right, there is a Data Grid View that keeps track of the events and requests that are being processed by the Elevator Cabin. It provides information such as the Target floor, the request status, Another important thing is the Request Type which tells you whether the request came from the dashboard or the floor panels

