**Object Oriented Programming and Software Engineering CIS016-2**

**Assignment 1 – Individual Project**

**Control an Elevator**

**Unit Code: - CIS016-2**

**Student Name: - Sushaan Lal Shrestha**

**University ID: - 2027362**

Table of Contents

[**Introduction** 3](#_Toc119063126)

[**Aims** 3](#_Toc119063127)

[**Objectives** 3](#_Toc119063128)

[**Problem** 3](#_Toc119063129)

[**Solution** 3](#_Toc119063130)

[**Requirements** 4](#_Toc119063131)

[**Hardware:** 4](#_Toc119063132)

[**Software:** 4](#_Toc119063133)

[**Implementation** 4](#_Toc119063134)

[**Class diagram** 5](#_Toc119063135)

[**Activity Diagram** 6](#_Toc119063136)

[**Testing** 7](#_Toc119063137)

[**1.** **Load Testing** 7](#_Toc119063138)

[**2.** **Functional Testing** 7](#_Toc119063139)

[**3.** **Button Testing** 10](#_Toc119063140)

[**Conclusion** 11](#_Toc119063141)

[**References** 13](#_Toc119063142)

[**Appendix** 14](#_Toc119063143)

[**Lift.cs Code: -** 14](#_Toc119063144)

[**Door.cs Code: -** 15](#_Toc119063145)

[**Databasecommand.cs Code: -** 16](#_Toc119063146)

[**Globalconnection.cs Code: -** 17](#_Toc119063147)

[**Form1.cs Code: -** 18](#_Toc119063148)

[**Form1.designer.cs Code : -** 26](#_Toc119063149)

[Figure 1: Class Diagram of Elevator 5](#_Toc119063150)

[Figure 2:Activity Diagram Of Elevator 6](#_Toc119063151)

[Figure 3:Elevator Design 7](#_Toc119063152)

[Figure 4:Ground floor door open 8](#_Toc119063153)

[Figure 5:Elevator car going from ground floor to first floor 8](#_Toc119063154)

[Figure 6:Elevator door opens after arriving to first floor 9](#_Toc119063155)

[Figure 7:EMERGENCY Both door open 10](#_Toc119063156)

# **Introduction**

In this project I am going to make an elevator control using C# for two storey building. An elevator is a mechanical device that carries people up and down levels of a building. Elevators are used to transport products and supplies between levels in buildings in addition to moving people. Elevators come in a variety of configurations; some are automated, while others involve physical control. The specifications include a control panel that displays the position of the elevator car. Each floor has an elevator car demand button. When the elevator is approaching the floor, the button will change color. There must be an elevator door that opens with the elevator arrives and closes with the elevator departing. Then, once at the desired floor, the elevator door must open and shuts after everyone exit from the elevator.

# **Aims**

As we know an elevator helps us in many ways. The purpose of this project is to evaluate a difficult issue that we face on a daily basis and use a programming language to create, build, and implement a variety of aspects, guidelines, and processes to the software development process in order to resolve issues.

# **Objectives**

* The ability to go through floors quicker and also to make it easier for the disabled and elderly to move between floors without physically exhausting them.
* Allows us to easily transfer goods
* Makes us feel at ease and relaxed during the ride
* Must have GUI.
* Must have functional button.

# **Problem**

According to this approach, the more delightful and passionate it sounds, the more harmful and life-threatening technology it is. As it advances into the structure, it becomes more dangerous if it stops in the middle due to some technical problem because even if we open the door they cannot get out. Sometimes due to power supply failure people may get trapped inside the elevator. (maddenelevator, 2022)

# **Solution**

Elevators, like other machinery, contain a variety of parts and moving elements that need routine maintenance. Additionally, they should undergo regular inspections to identify problem areas and address little concerns before they grow into larger ones. There is an emergency button for the emergency system so that the user may press it during an incident which will be the initial command together with the fundamental structure and commands. With this, a database is created to record all elevator activity for the purpose of protection and data management. (maddenelevator, 2022)

# **Requirements**

## **Hardware:**

* RAM 4GB or above
* Intel dual core processor and above

## **Software:**

* C # .net framework
* Microsoft Access
* Database engine 64bit
* Visual Studio

# **Implementation**

The goal was to design an elevator system using C# and the.Net framework. The tasks included creating a GUI with two request buttons for each level, a control panel with two buttons and a display window, and two display areas that indicate the elevator status, i.e., which floor it is now on. A log button causes the history data of the elevator operation to be shown. Task 2 is to write a control program that processes all the events generated by the GUI. Task 3 is to write a log that records all the actions and stores them in a database using MS Access or any other comparable database. Task 4 is to use the delegation and timer to animate all of the events mentioned in Task 2, and Task 5 is to optimize the programs produced in Tasks 1 through 4. The final step is to create a test report that includes the "Marking Matrix with Self-Assessment" table. The implementation code is mentioned in the appendix.

# **Class diagram**

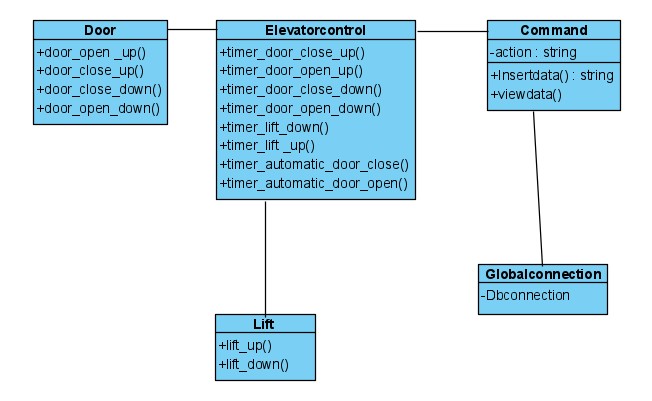


Figure 1: Class Diagram of Elevator

Class diagrams are used in application development to display the system's classes, their properties, actions (or methods), and the relationships between objects.

A class diagram of the elevator is presented above, along with its characteristics and operation mechanism. In this figure, the door has five classes in total, containing all the door's activities. As depicted, the elevator controller controls the elevator's timing. (paradigm, 2022)

# **Activity Diagram**

Diagram

Description automatically generated

Figure 2:Activity Diagram Of Elevator

The figure above indicates that the system's operation is basic. The elevator is used to move people from one storey to another, such as from one to two or two to one. When the button is pressed, the elevator car summons itself, opens the door, and the user enters. The door then closes and reopens when it reaches the target floor. (geeksforgeeks, 2022)

# **Testing**

## **Load Testing**

The purpose of an elevator load test is to ensure that the machines can lift and transport its maximum weight at its top speed without risking the safety of anyone within. A load test for an elevator is necessary since it is a piece of machinery that transports people down and up a building. If any element, component, gadget, cable, or mechanism is defective, it can endanger someone’s life.

## **Functional Testing**

The lift performance and functionality were checked in this testing, and the results were positive. Some of the functional testing’s are shown below: -

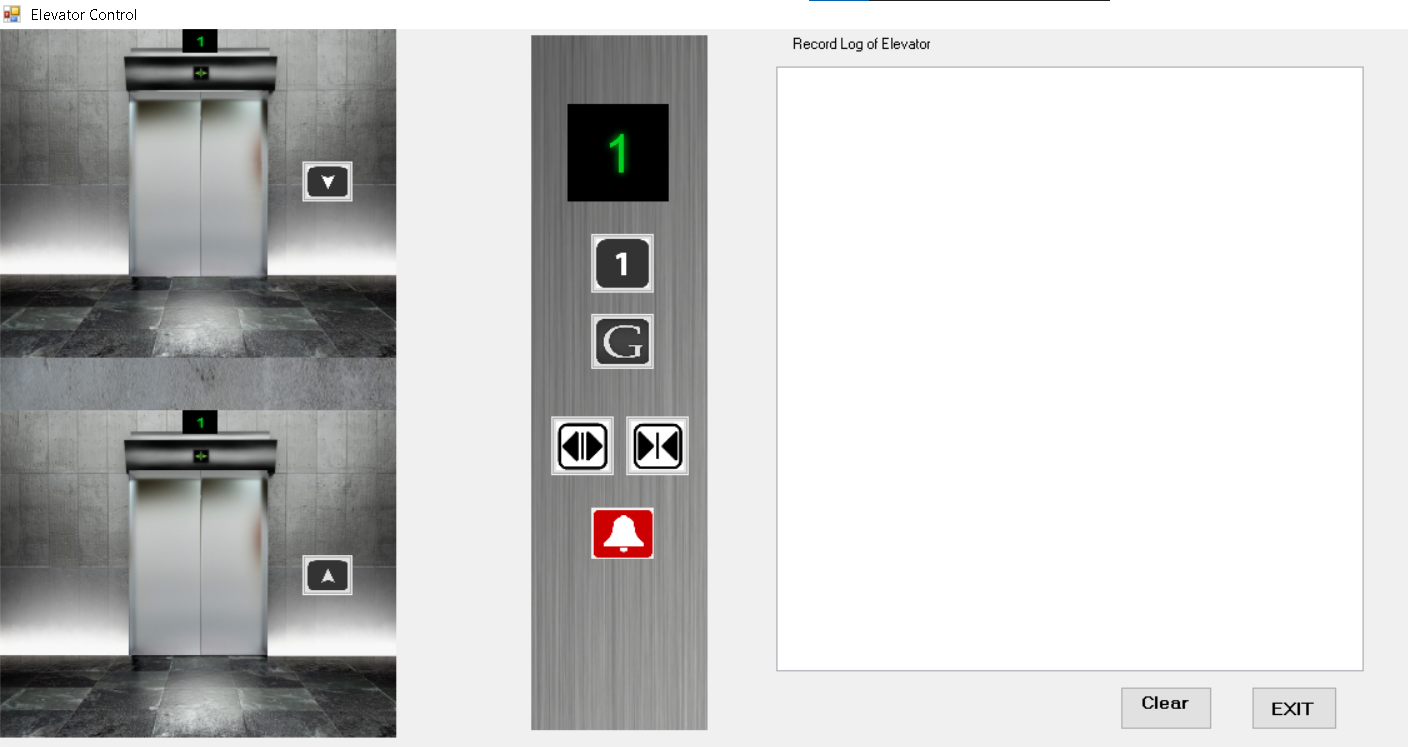


Figure 3:Elevator Design

This shows the elevator with database and the buttons.

Graphical user interface, application

Description automatically generated

Figure 4:Ground floor door open

In this diagram we can see when we click the open button the elevator door opens at the ground floor.

Graphical user interface, application

Description automatically generated

Figure 5:Elevator car going from ground floor to first floor

In this figure when we click the first-floor button the lift goes from ground floor to first floor.

Graphical user interface, application

Description automatically generated

Figure 6:Elevator door opens after arriving to first floor

In this figure we can see that after we have clicked the first-floor button the elevator car has reached the first floor and the door opens automatically for the people to get out at their destination.

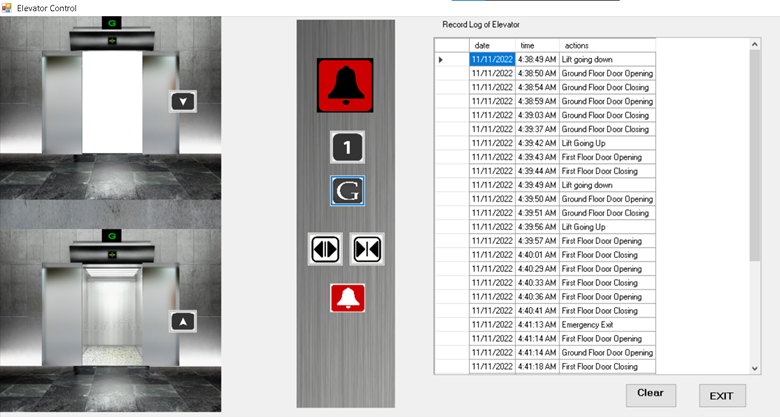


Figure 7:EMERGENCY Both door open

In this above figure we can see that when we press the emergency button both the lift door opens automatically to help the people get out in case of emergency or help the maintenance team solve the problem.

## **Button Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. N | Steps | Expected result | Actual result | Remarks |
| 1 | Press the button (either Up or Down) | Elevator must go to the button pressed floor | Expected result | Pass |
| 2 | Press the open button | Elevator must open the door in the button pressed floor. | Expected result | pass |
| 3 | Press the close button | Elevator must close the door in the button pressed floor. | Expected result | pass |
| 4 | Press the emergency button | Elevator must open the door in both floor | Expected result | Pass |
| 5 | Display | Display should sow the floor in which the elevator car is and show it coming down or (paradigm, 2022) up. | Expected result | pass |

# **Conclusion**

All the procedures and requirements are carefully examined and depicted in a diagram to understand all the information, as specified in the assignment brief. First, a thorough examination of all qualifications is performed, including understanding of C # languages, databases, and designs.

When the information encoded, many issues and crises develop, but as time passes, all the problems are resolved, and all of the criteria are satisfied. Designing the requirements in a correct and human comprehensible manner was the biggest issue that arose, but it was also appropriately addressed. Thorough testing and usability should be performed, and this technology should be utilized in the majority of the buildings.

Marking Matrix with Self-Assessment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Number** | **Sub-tasks** | **Possible Marks** | **Self-assessment (completed Yes/No)** | **Reference to your testing report** | **Mark Awarded** |
| **Task 1** | Complete GUI for Task 1 | 10 | yes | Design GUI | 10 |
| Skeleton of event handlers in place for all buttons | 10 | yes | Implementation | 10 |
| **Task 2** | All event handlers are functional | 10 | yes | implementation | 10 |
| **Task 3** | Database (DB) is designed and can be connected | 5 | yes | Backup and recovery | 5 |
| Log Information can be retrieved from DB and displayed in the GUI | 5 | yes | Backup and recovery | 5 |
| When the log button is pressed, log information is sent to and stored in the DB | 5 | yes | Backup and recovery | 5 |
| Use the disconnected model rather than connected model (Data source is updated via DataAdapters Update() method instead of ExecuteNonQuery() method) | 5 | yes | Implementation | 5 |
| **Task 4** | Events described in Task 2 animated using delegation and timer | 10 | yes | Implementation | 10 |
| **Task 5** | Using relative path instead of absolute path | 5 | yes | Implementation | 5 |
| Avoiding any duplication among the event handlers over the database related functions | 5 | yes | Implementation | 5 |
| Eliminating logical errors and handling exceptions with try and catch | 5 | yes | Implementation | 5 |
| Optimize the efficiency of GUI by implementing multiple tasks concurrently via BackgroundWorker | 5 | yes | Implementation | 5 |
| Use state patterns instead of if-else statements to accommodate future changes of the requirement | 10 | yes | Implementation | 10 |
| **Task 6** | Test report | 10 | yes |  | 10 |
| **Total** |  | 100 |  |  |  |

# **References**

geeksforgeeks, 2022. *geeksforgeeks.* [Online]   
Available at: http://www.geeksforgeeks.com  
[Accessed 9 11 2022].

maddenelevator, 2022. *maddenelevator.* [Online]   
Available at: https://maddenelevator.com/news-and-pr/2021/10/20/the-four-most-common-elevator-problems-discovered-in-routine-inspections/  
[Accessed 2022].

paradigm, v., 2022. *visual paradigm.* [Online]   
Available at: http://www.visual-paradigm.com  
[Accessed 2022].

# **Appendix**

## **Lift.cs Code: -**

Graphical user interface, text, application, email

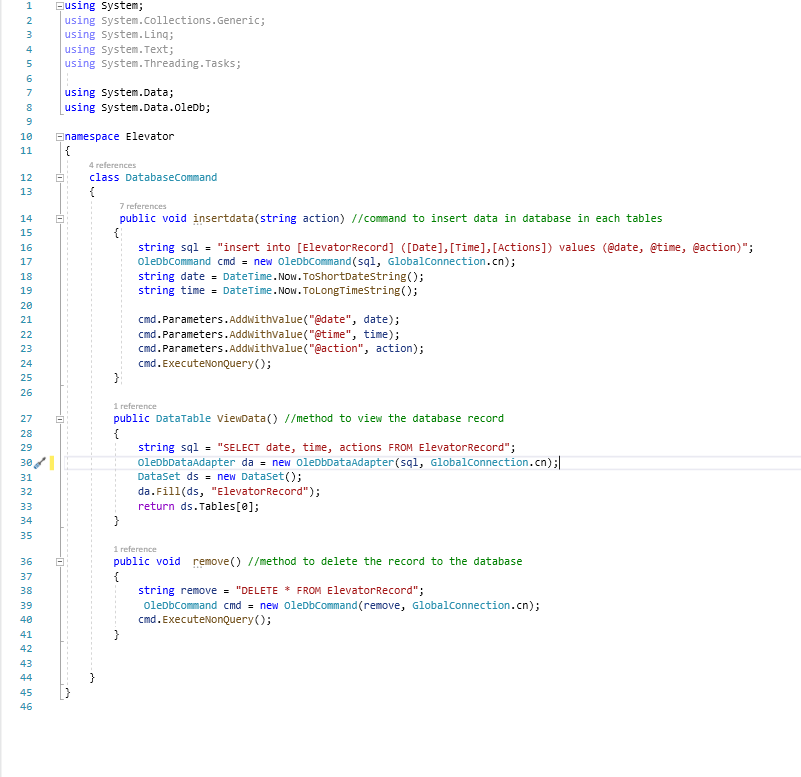
Description automatically generated

## **Door.cs Code: -**

Text

Description automatically generated

## **Databasecommand.cs Code: -**



## **Globalconnection.cs Code: -**

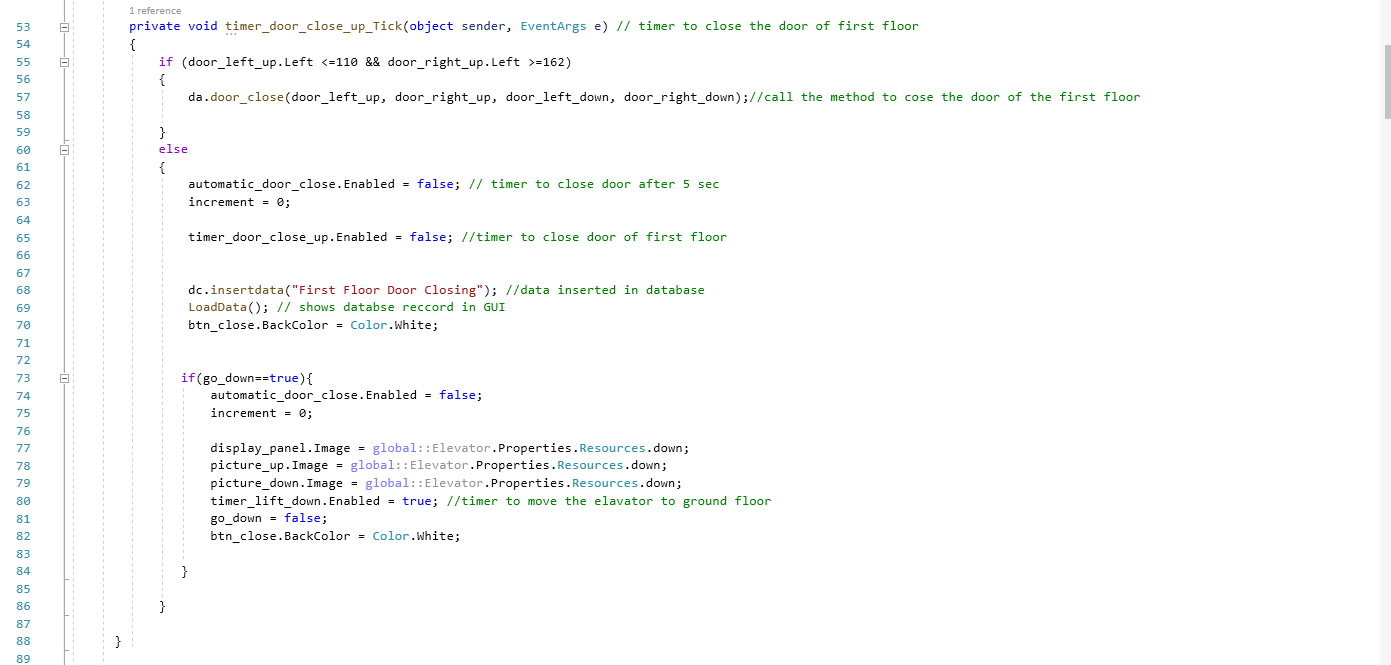
Graphical user interface, text, application, email

Description automatically generated

## **Form1.cs Code: -**

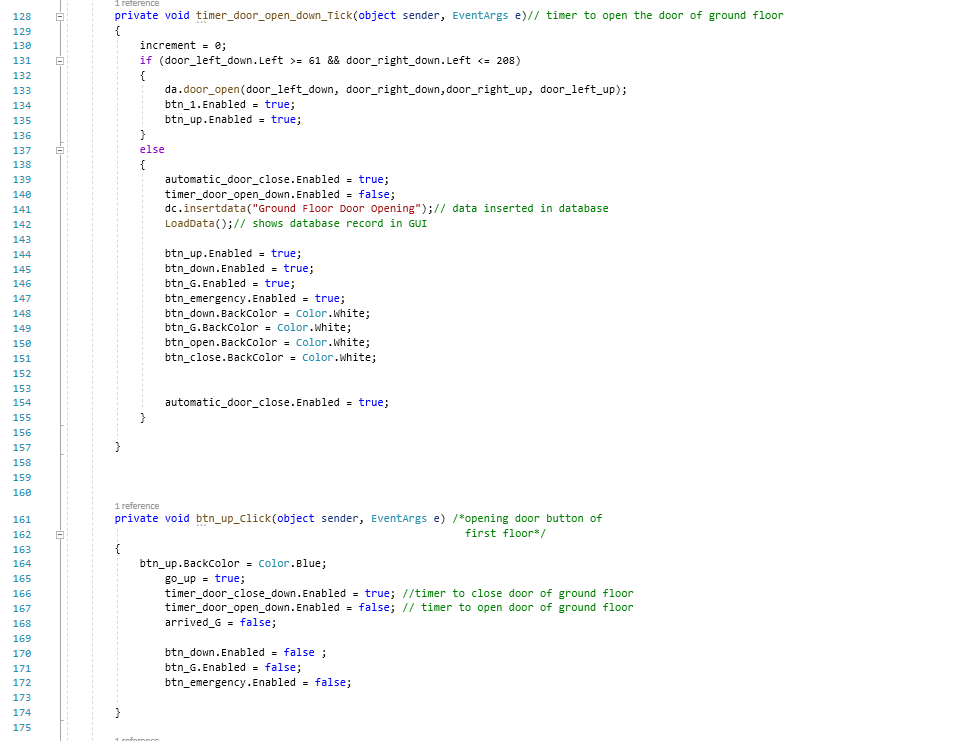
Graphical user interface, text, application

Description automatically generated with medium confidence

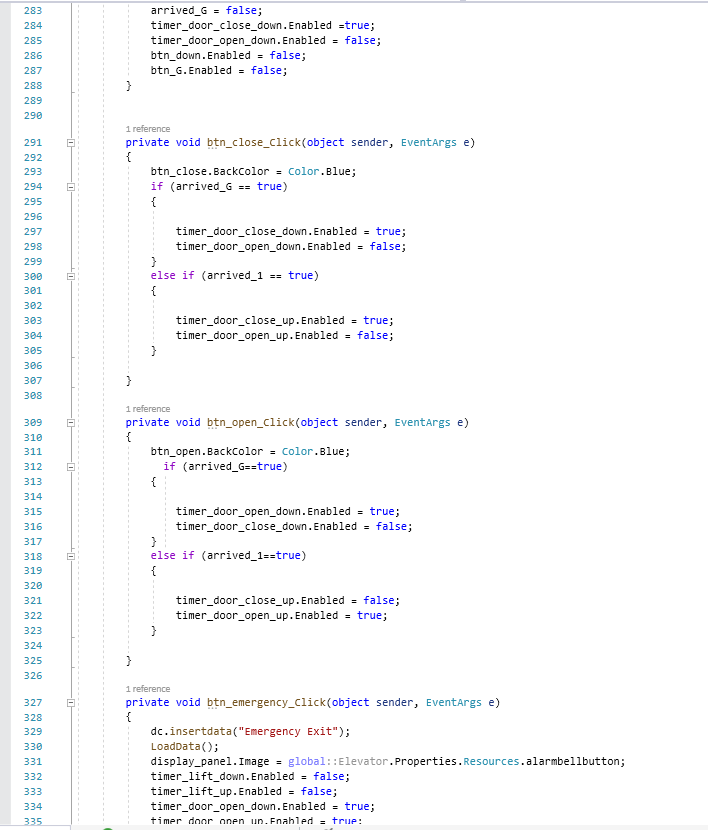


Text

Description automatically generated





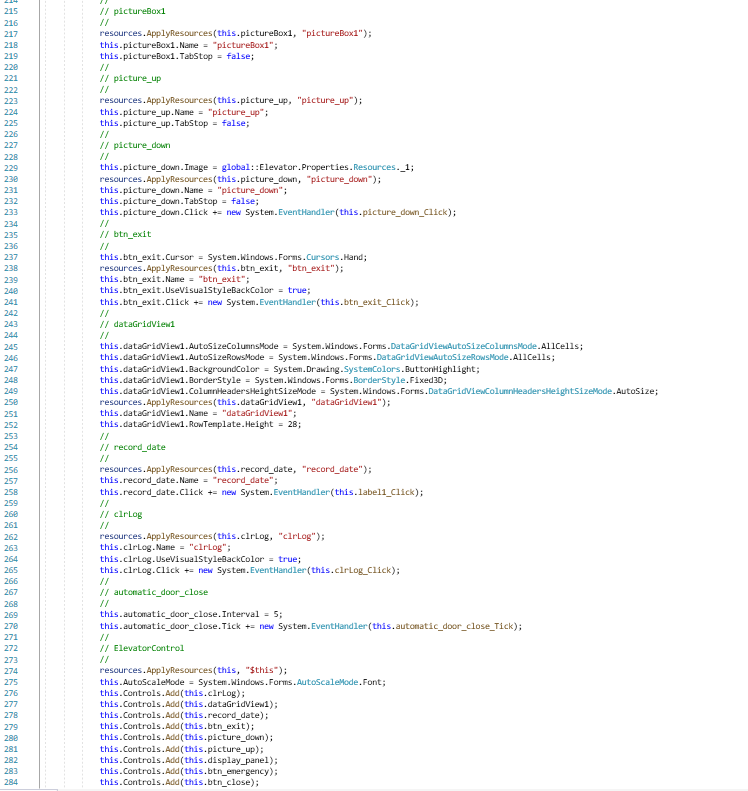
   Graphical user interface, text, application, email

Description automatically generated

## **Form1.designer.cs Code : -**





  Table

Description automatically generated