Object Oriented Programming and Software Engineering

Assignment 1 — Individual Project

COURSEWORK: - Control an Elevator

(A C# Project)

Unit Code: - CIS016-2

Submitted to: - University of Bedfordshire

Submitted by: -	University ID	Submission Date
Mayank Agarwal	2210038	November 09, 2023

Contents

Introduction

The elevator system in today's world plays very important role in maintaining life balance in every person of all sectors of the world. It advances the life with the help of technology and reduces time as people can travel over floors within seconds or minutes which makes the work easy and performs small but significant role in our day-to-day life. This is a Elevator Control Project constructed under C# Dot Net Framework under Assignment 1 under unit Object-Oriented Programming and Software Engineering, subject code CIS016-2 which carries 30% of the total weight of the marks. This project checks the concept level of student in C# programming and Object-Oriented concept. This assignment requires to design a working elevator system using C# language and a database to store data. In this project it was required to look for working of design and testing with database running properly.

Problem Statement

Vertical Transportation System, commonly known as "Lifts" in buildings usually faces many challenges like operating, overloading, user dissatisfaction, safety, etc. The problem Statement here states to implement a C# lift elevator control system that could address above challenges and provide an effective solution to it. A small description about the challenges is given below: -

- Many lifts system all around the world usually faces problems of over traffic/crowd management and data record.
- Outdated interface in lifts usually leads to customer dissatisfaction and safety in lifts in one of the major concerns.

Proposed Solution

This solution aims to this design and develop working C# lift controller for 2 floors that is efficient easy to operate. The challenges it solves are summarized below.

- By using algorithms that manages the load balance and traffic and deducts time taken, also a system that records the real time data and location of the lift.
- A simple creativity of control panel to operate the lifts for the users and proper evacuation system during emergency.

Aims and Objectives

This project mainly aims to demonstrate the issues faced in a lift system and develop the skills to provide solutions to day-to-day world issues. The other objective of this assessment is provided below: -

- To have the clear concept of OOPs and C# programming language.
- Designing a smart and attractive user interface of Lift with its controller.
- To have a database to record data and fetch it to showcase it on the running window.

Project Plan

Being this an individual project, I have created this project working plan for myself that showcases the development stages of the project from its planning stages to its completion.

Week No.	Tasks	Priority
Week 3	Identify the requirements (Functional and Non-functional)	Could
Week 3	Create its architecture and prototype	Could
Week 4	Develop the potential and easy to operate user interface	Must
Week 4	Using C# programming provide its working assistance	Must
Week 4	Create a database and insert data by creating tables	Must
Week 5	Using C# fetch the data and showcase in the working window	Must
Week 5	Test the code again and again and make sure there is no error occurred	Must
Week 5	Review with the teachers	Could
Week 6	Submit individual report with source code	Must
Week 6	Project Presentation	Must

Table 1:- Project Plan Table

System Requirements

❖ Hardware Specifications

- A pc having a minimum of 2 Gb RAM and dual core processor embedded in it.
- A monitor, Keyboard, and mouse.
- Proper storage capacity, I prefer to have SSD.

❖ Software Specifications

- Latest version of Visual Studio (mine is visual studio 2022)
- (".net") Framework specification
- Database panel (MySQL, MS Access) to store data provided.
- Visual paradigm or designing tool to draw use cases and activity diagram.

❖ Functional Specifications

- Up/Down request button to call the shaft on the required floor.
- A control panel with buttons to operate the shaft to reach the desired floor.
- A proper working of doors and stoppage of shaft at the proper position that should not make errors to cause accidents.
- Display Actions button to show data provided by buttons to the database in data log table.

System Design

➢ Use Case Design

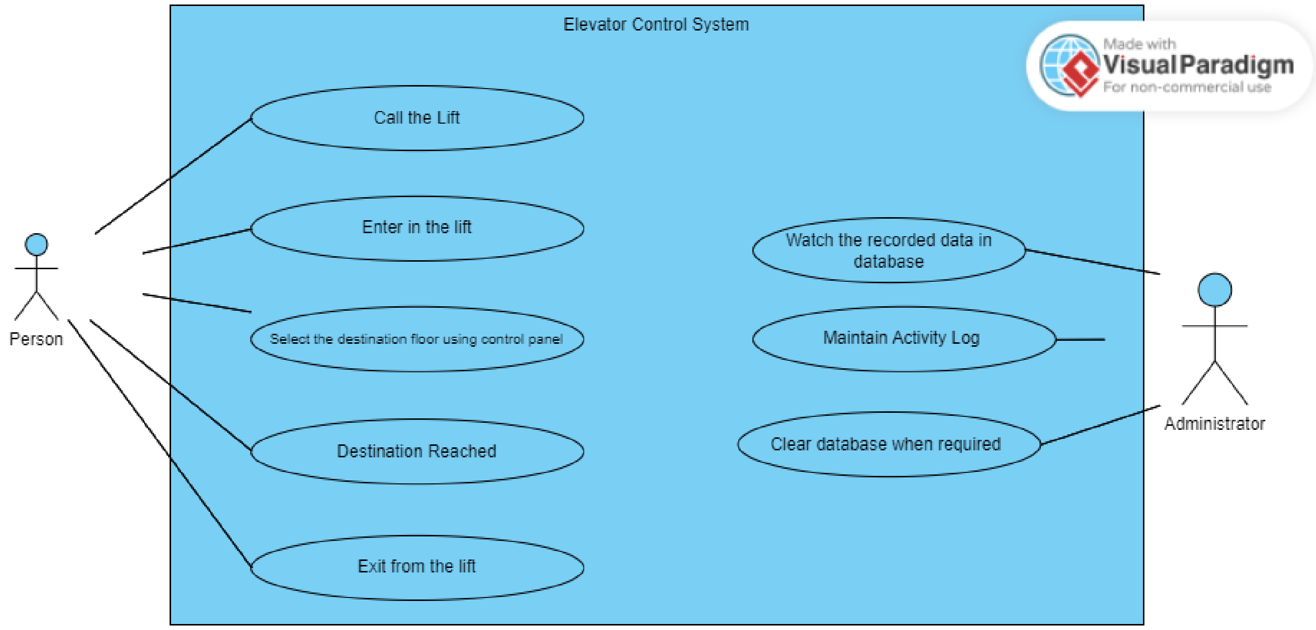

Elevator Control System
Made with Visual Paradigm For non-commercial use
Call the Lift
Enter in the lift
Select the destination floor using control panel
Destination Reached
Exit from the lift
Watch the recorded data in database
Maintain Activity Log
Clear database when required
Person
Administrator

Figure 1:- Use case design

In the diagram above the person need to call the lift, when the shaft reach to the floor where the person is, the doors will open, the person needs to enter the lift then select the desired destination floor number using the control panel, once reaching the destination floor the doors will open and person will exit from the lift at designated floor and the doors will be closed. Similarly, the admin can watch the recorded data in the database and maintain the activity log and whenever required, admin can clear the database table.

➢ Activity Design

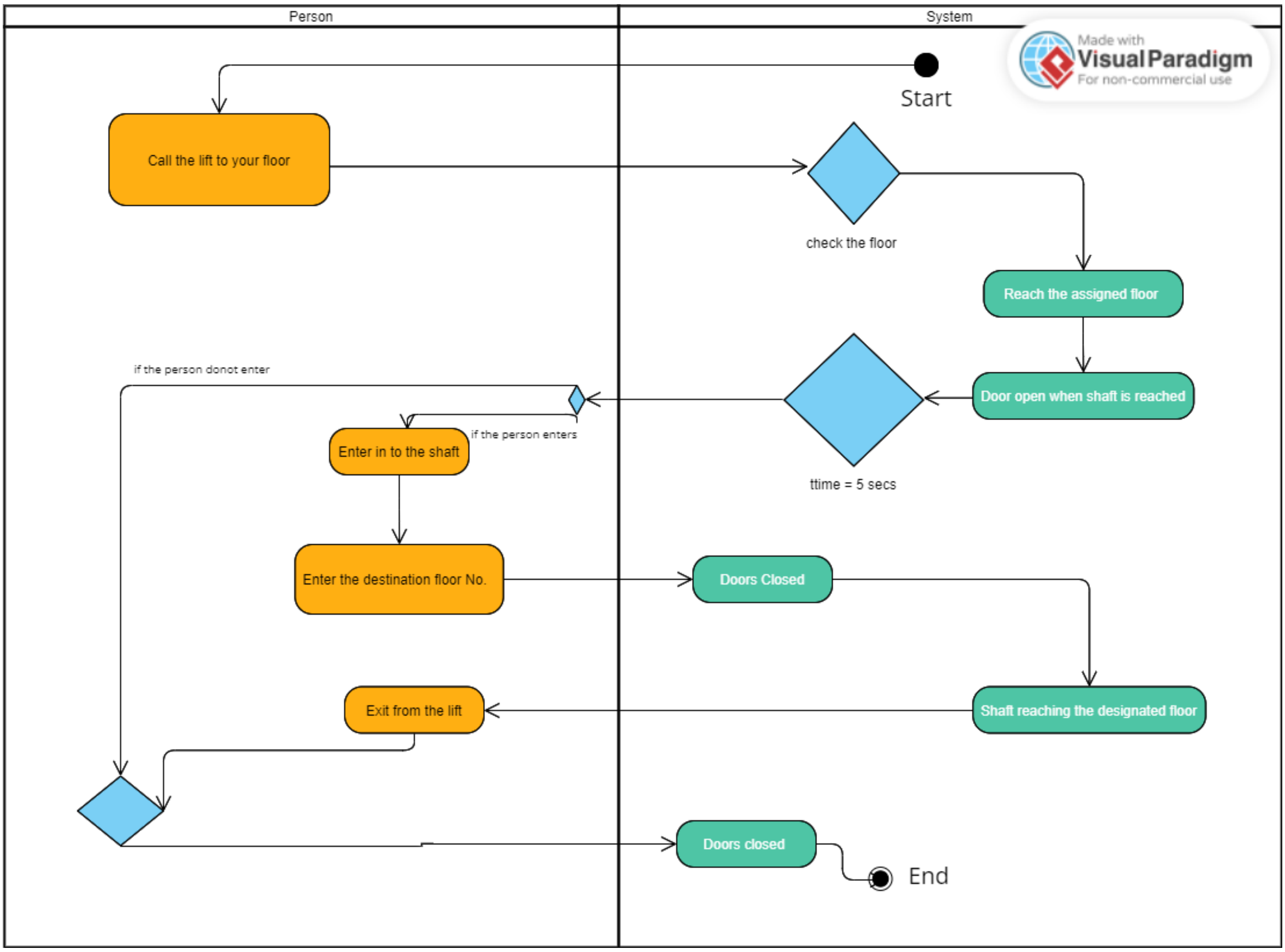


Figure 2:- Activity Design

At first the system starts, then the person will call the lift to its floor, as soon the button is pressed the system checks the floor number and shaft is moved to the floor and doors will be open, if the person enters the shaft and press the destination floor number using control panel the door closes and the shaft moves to the designated floor and the person exits once the door is open at designated floor, if the person doesn't enter at the called floor the door closes and system ends.

About the Language

The language that I have used in accomplishing my assignment is C# commonly pronounced as “C-Sharp”. C# has been used in “(.net Framework)” in latest version of Visual studio 2022. Being Structured by Microsoft, it is specialized for its simplicity and readability.

Standout features of C# is its OOPs capabilities, organized designing pattern which maintains scalability and other aspects of a project.

From the aspects of this project C# helps in designing the user interface using windows forms, where it gets integrated with .NET Framework which helps in functioning and connection with the database. Prominent features of C# is its secured environment, memory management, and its robust capabilities.

About the Database

In this project of elevator control system using C#, I have used Microsoft Access Database to store my data and fetched it from there to display in data log table. MS Access Database is a user-friendly database management system that works well with all sorts of projects and is embedded with all the required features which serves reliable data, stores all kind of information, performs all the requests, and maintains the balance in an organized manner.

MS Access is well designed solution for windows that integrates C# and .NET Framework and artifices for the project. It maintains communication between applications and storage with secured environment.

Prototype

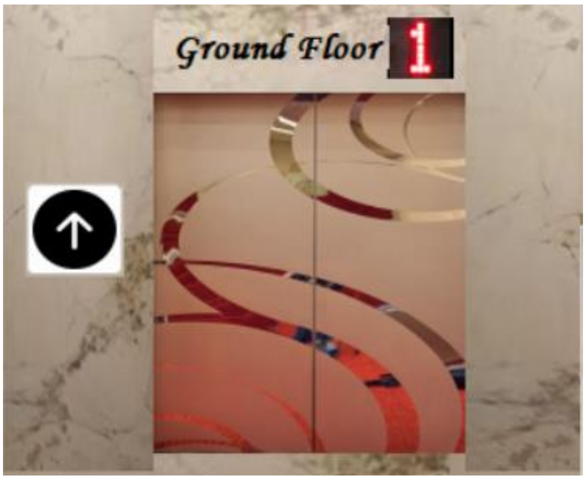


Figure 3:- Lift User interface at ground level

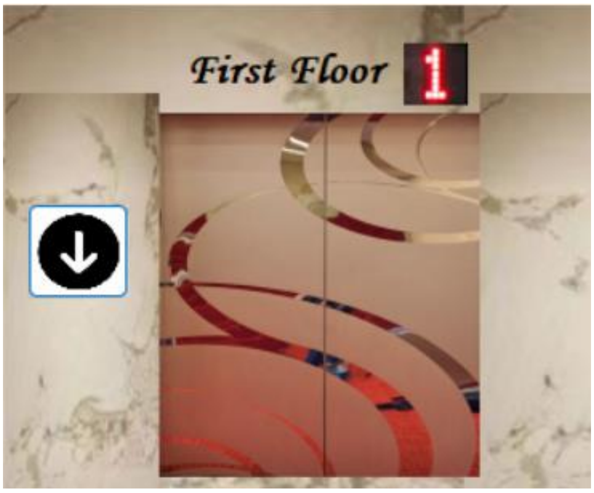
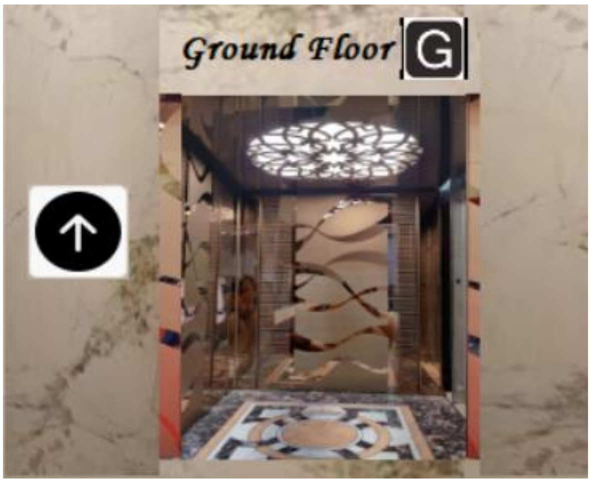


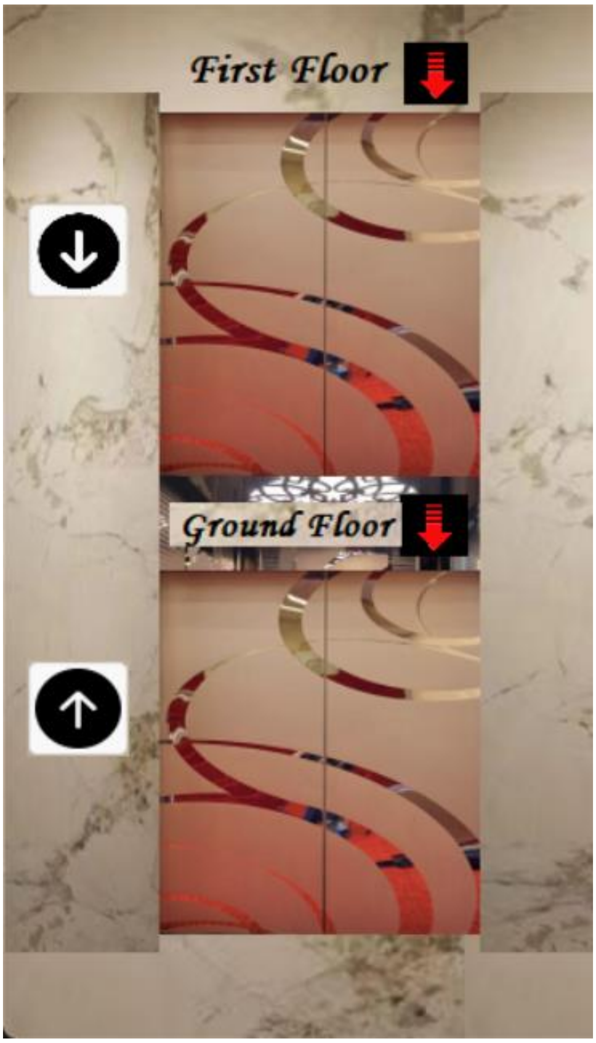
Figure 4:- Lift User interface at first level

This is the floor representation user interface that is designed for lift with a screen showing the current floor on which the inner shaft is and buttons to call the shaft on the desired floor.



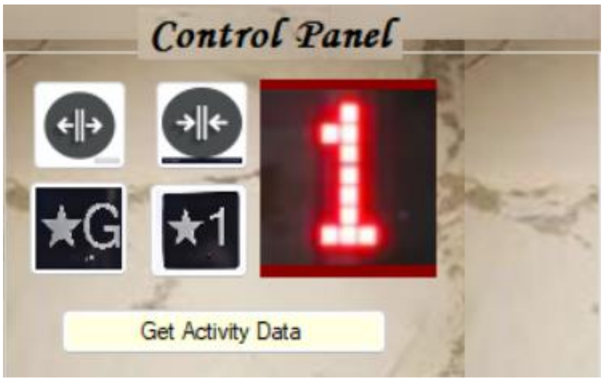
This interface shows the open-door view of lift when it reached the ground floor. The screen above shows the floor number as "G" for ground floor.

Figure 5:- Lift user interface of open door at ground level



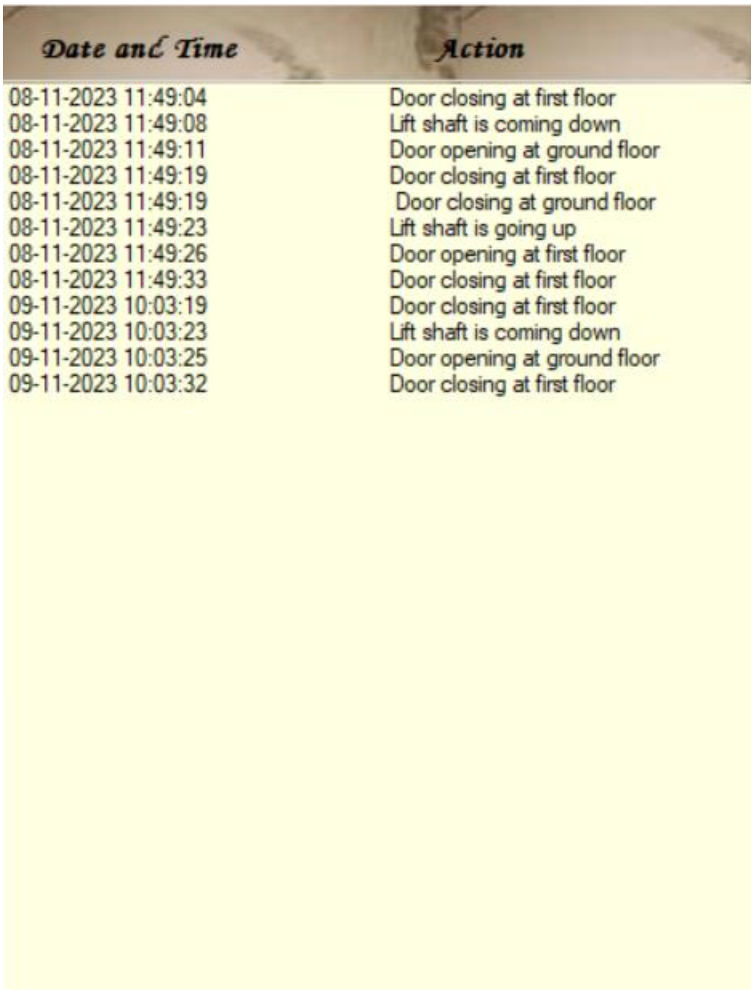
This interface shows the movement of shaft from first floor to ground floor, where the screen showing the movement of shaft downwards and shaft can be seen between two floors going down.

Figure 6:- Lift user interface shows movement of shaft



This user interface showcases the control panel which consists of buttons which helps in movement of lift and help to see the activity data too.

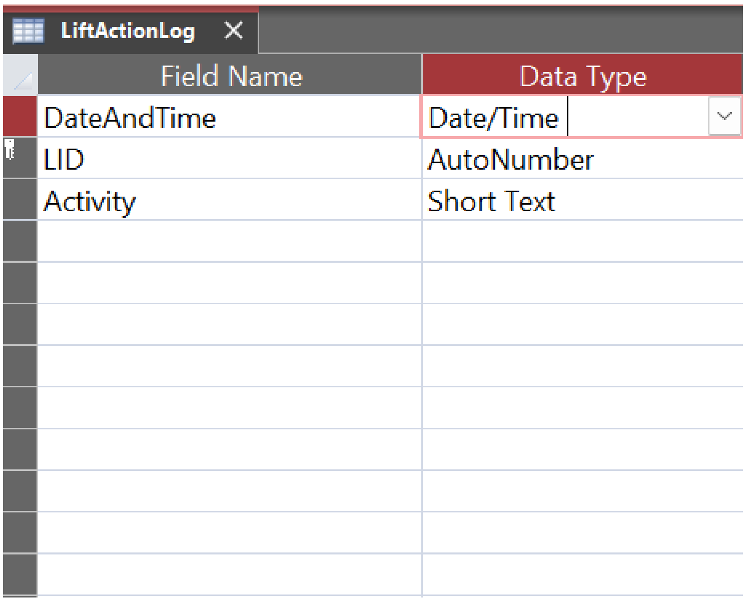
Figure 7:- Lift user interface shows control panel



Date and Time	Action
08-11-2023 11:49:04	Door closing at first floor
08-11-2023 11:49:08	Lift shaft is coming down
08-11-2023 11:49:11	Door opening at ground floor
08-11-2023 11:49:19	Door closing at first floor
08-11-2023 11:49:19	Door closing at ground floor
08-11-2023 11:49:23	Lift shaft is going up
08-11-2023 11:49:26	Door opening at first floor
08-11-2023 11:49:33	Door closing at first floor
09-11-2023 10:03:19	Door closing at first floor
09-11-2023 10:03:23	Lift shaft is coming down
09-11-2023 10:03:25	Door opening at ground floor
09-11-2023 10:03:32	Door closing at first floor

This image shows the data log table present on the working window which displays the action data from the database, as date and time and action performed by the shaft.

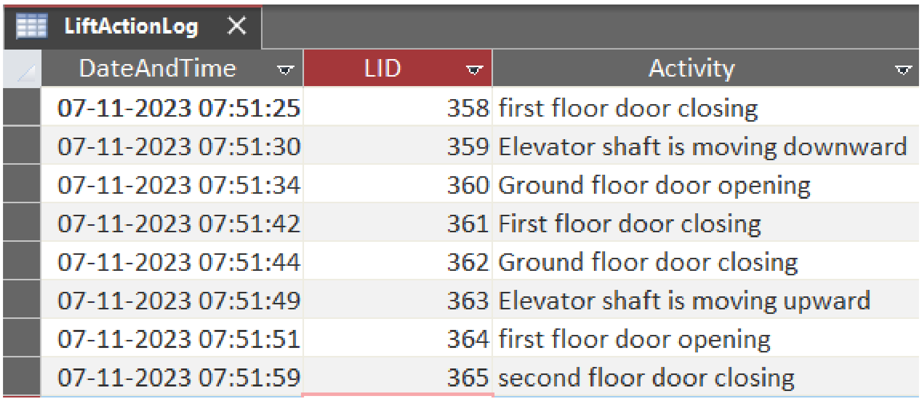
Figure 8:- Lift user interface shows data log table



Field Name	Data Type
DateAndTime	Date/Time
LID	AutoNumber
Activity	Short Text

Figure 9:- Database

The image on the left shows the database has been created namely “Elevator_Datab” which has a table named “LiftActionLog” which has field as DateAndTime (stores date and time), LID (primary key and auto number), and Activity (which stores the action in text form).



DateAndTime	LID	Activity
07-11-2023 07:51:25	358	first floor door closing
07-11-2023 07:51:30	359	Elevator shaft is moving downward
07-11-2023 07:51:34	360	Ground floor door opening
07-11-2023 07:51:42	361	First floor door closing
07-11-2023 07:51:44	362	Ground floor door closing
07-11-2023 07:51:49	363	Elevator shaft is moving upward
07-11-2023 07:51:51	364	first floor door opening
07-11-2023 07:51:59	365	second floor door closing

Figure 10:- Database with inputs

This figure shows the history of the lift with time and action that is stored in database.

Testing and Evaluation Strategy

➢ Unit Testing

- Purpose: - To check the functions of each constituent of the system

- Key elements: - GUI, Database, algorithms

- Tools: - Visual studio 2022

- Point of reference: - Working of each function properly and exception handling performing correctly.

Integration Testing

- Purpose: - To check the linking between different commands and its working

- Key Elements: - Control Panel, GUI, MS Access Database amalgamation

- Tools: - Visual Studio 2022

- Point of reference: - working of GUI modules as per command from control panel and data attachment to the database.

System Testing

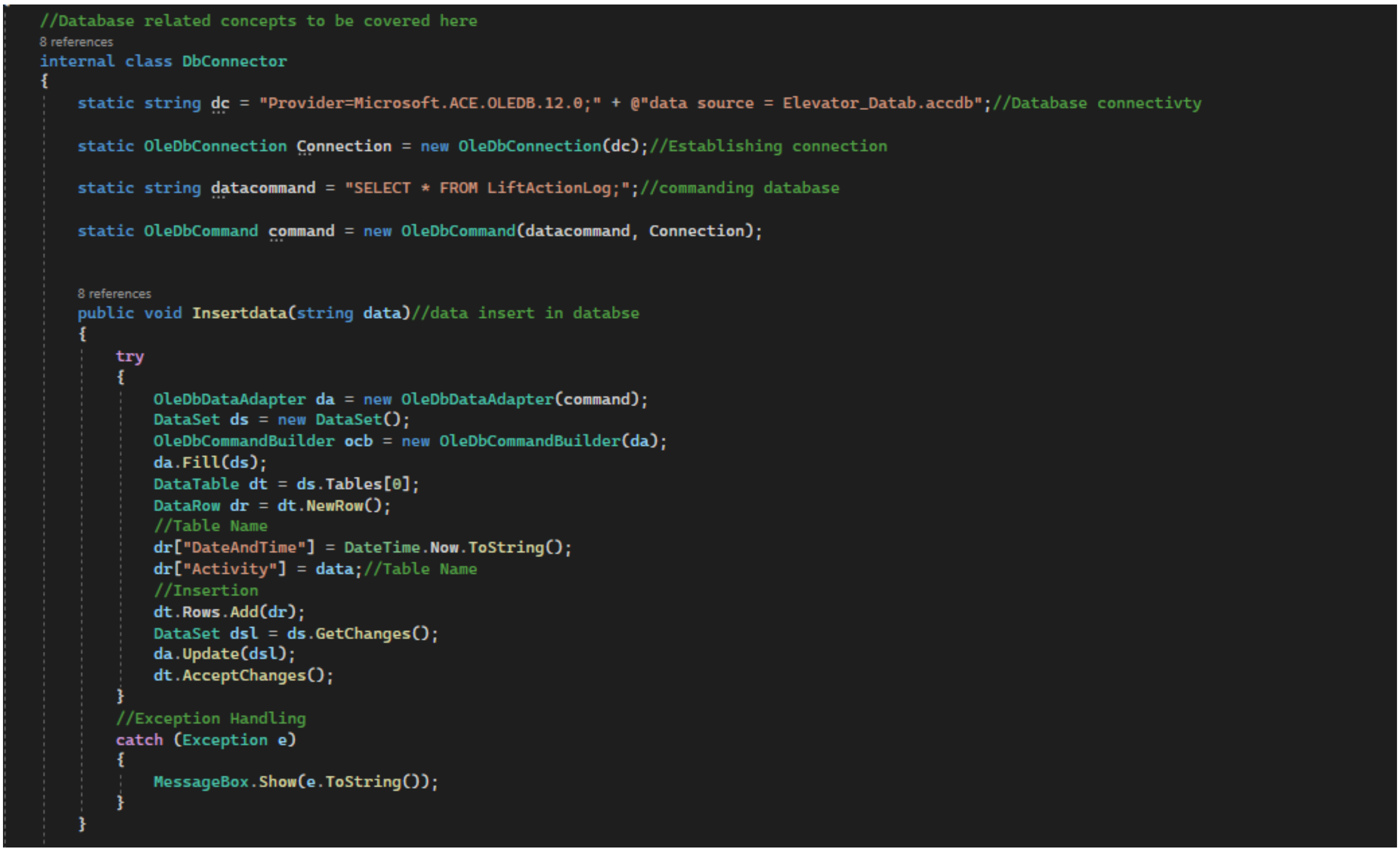
- Purpose: - To check the whole system and accomplishing its requirements

- Key Elements: - working of elevator.

- Tools: - Visual Studio 2022

- Point of reference: - Working of whole elevator system as per the command and it's functioning as per the requirements of the system

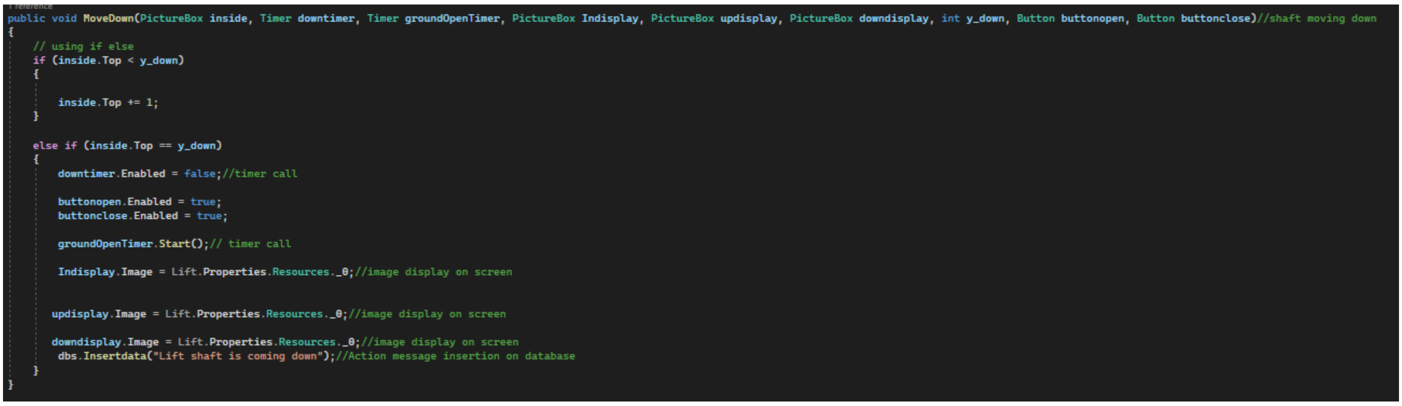
Implementation



```
//Database related concepts to be covered here
8 references
internal class DbConnector
{
    static string dc = "Provider=Microsoft.ACE.OLEDB.12.0;" + @"data source = Elevator_Datab.accdb";//Database connectivty

    static OleDbConnection Connection = new OleDbConnection(dc);//Establishing connection

    static string datacommand = "SELECT * FROM LiftActionLog;";//commanding database

    static OleDbCommand command = new OleDbCommand(datacommand, Connection);


    8 references
    public void Insertdata(string data)//data insert in databse
    {
        try
        {
            OleDbDataAdapter da = new OleDbDataAdapter(command);
            DataSet ds = new DataSet();
            OleDbCommandBuilder ocb = new OleDbCommandBuilder(da);
            da.Fill(ds);
            DataTable dt = ds.Tables[0];
            DataRow dr = dt.NewRow();
            //Table Name
            dr["DateAndTime"] = DateTime.Now.ToString();
            dr["Activity"] = data;//Table Name
            //Insertion
            dt.Rows.Add(dr);
            DataSet dsl = ds.GetChanges();
            da.Update(dsl);
            dt.AcceptChanges();
        }
        //Exception Handling
        catch (Exception e)
        {
            MessageBox.Show(e.ToString());
        }
    }
```

Figure 11:- Code Snippets

The above code snippets show database connections which is surrounded by Exception (try-catch block) to handle the errors which helps them from being crashed. To maintain system robustness logical errors were solved and later exception handling was used wherever it found necessary.



```
1 reference
public void MoveDown(PictureBox inside, Timer downtimer, Timer groundOpenTimer, PictureBox Indisplay, PictureBox updisplay, PictureBox downdisplay, int y_down, Button buttonopen, Button buttonclose)//shaft moving down
{
    // using if else
    if (inside.Top < y_down)
    {

        inside.Top += 1;
    }

    else if (inside.Top == y_down)
    {
        downtimer.Enabled = false;//timer call

        buttonopen.Enabled = true;
        buttonclose.Enabled = true;

        groundOpenTimer.Start();// timer call

        Indisplay.Image = Lift.Properties.Resources._0;//image display on screen


       updisplay.Image = Lift.Properties.Resources._0;//image display on screen

       downdisplay.Image = Lift.Properties.Resources._0;//image display on screen
        dbs.Insertdata("Lift shaft is coming down");//Action message insertion on database
    }
}
```

The above code snippet shows commanding of to a timer using if else statement which will help in working of functions in the system.

Testing

Test Case No: 01

Case Description: Request up and down of the shaft

Created and Review by: Mayank Agarwal

Test By: Mayank Agarwal

Date/Time: Monday November 06, 2023, 7:30 AM

Prerequisites: N/A

Test Date: 05 November 2023

Test Result: Pass

Step	Description	Expected Output	Actual Output	Result	Error	Error Status
01	Start the elevator application	Starting of application and MDI window appearing	MDI window Appeared	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR
02	Click on DOWN button to request shaft coming on ground floor	Shaft moving down and door opening at ground floor	Ground floor door opened with shaft ready	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR
03	Click on UP button to request shaft coming on first floor	Shaft moving up and door opening at first floor	First floor door opened with shaft ready	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR

Table 2:- Test case no-1

Test Case No: 02

Case Description: Closing and opening of the doors

Created and Review by: Mayank Agarwal

Test By: Mayank Agarwal

Date/Time: Monday November 06, 2023, 8:20 AM

Prerequisites: N/A

Test Date: 05 November 2023

Test Result: Pass

Step	Description	Expected Output	Actual Output	Result	Error	Error Status
01	Start the elevator application	Starting of application and MDI window appearing	MDI window Appeared	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR
02	Clicking on **open** the door button on control panel	Doors to be open	Door opened	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR
03	Clicking on **close** the door button on control panel	Doors to be close	Door closed	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR

Table 3:- Test case no-2

Test Case No: 03

Case Description: Requesting to reach ground Floor from first floor

Created and Review by: Mayank Agarwal

Test By: Mayank Agarwal

Date/Time: Monday November 06, 2023, 7:50 AM

Prerequisites: N/A

Test Date: 05 November 2023

Test Result: Pass

Step	Description	Expected Output	Actual Output	Result	Error	Error Status
01	Start the elevator application	Starting of application and MDI window appearing	MDI window Appeared	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR
02	Requesting shaft to move down to ground floor by clicking on **G** button on control panel	Movement of lift at ground floor and opening of door as reached	Door opened at ground floor as it reached	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR

Table 4:- Test case no-3

Test Case No: 04

Case Description: Requesting to reach first Floor from ground floor

Created and Review by: Mayank Agarwal

Test By: Mayank Agarwal

Date/Time: Monday November 06, 2023, 8:00 AM

Prerequisites: N/A

Test Date: 05 November 2023

Test Result: Pass

Step	Description	Expected Output	Actual Output	Result	Error	Error Status
01	Start the elevator application	Starting of application and MDI window appearing	MDI window Appeared	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR
02	Requesting shaft to move down to ground floor by clicking on **1** button on control panel	Movement of lift at first floor and opening of door as reached	Door opened at first floor as it reached	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR

Table 5:- Test case no-4

Test Case No: 05

Case Description: Requesting data from database into data log table

Created and Review by: Mayank Agarwal

Test By: Mayank Agarwal

Date/Time: Monday November 06, 2023, 8:40 AM

Prerequisites: N/A

Test Date: 05 November 2023

Test Result: Pass

Step	Description	Expected Output	Actual Output	Result	Error	Error Status
01	Start the elevator application	Starting of application and MDI window appearing	MDI window Appeared	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR
02	Data request from database as clicked on **get activity data** button	Data show on table after command given	Data arrived at data log table	**PASS** FAIL NOT EXECUTED	YES **NO**	NO ERROR

Table 6:- Test case no-5

Evaluation

This project of Elevator control system which has been created in C# language using .NET framework using Visual studio. It has been tested several times and addressed successfully maintaining all the standards of functionality, satisfaction, security, and its performance. It has operated all the functions successfully and demonstrated the results within the time and has stood up on all the expectations with security and data protection. Overall, this system has proved to be reliable and energy efficient with its excellence in operating and made it adaptable in vertical transportation and installation in urban buildings.

Conclusion

At the end of this project, I would like to thank my teacher whose guidance has led to clear my concept of OOPs which helped in completion of this project. This elevator control system has been designed using C# .NET framework. Although this has been quite time taking and hectic in completing than I had thought of but it has been a great experience working with C# and made my concepts clear for real time working projects, I would love to take more such assignments in future and work more on C#.

Marking Matrix with Assessment

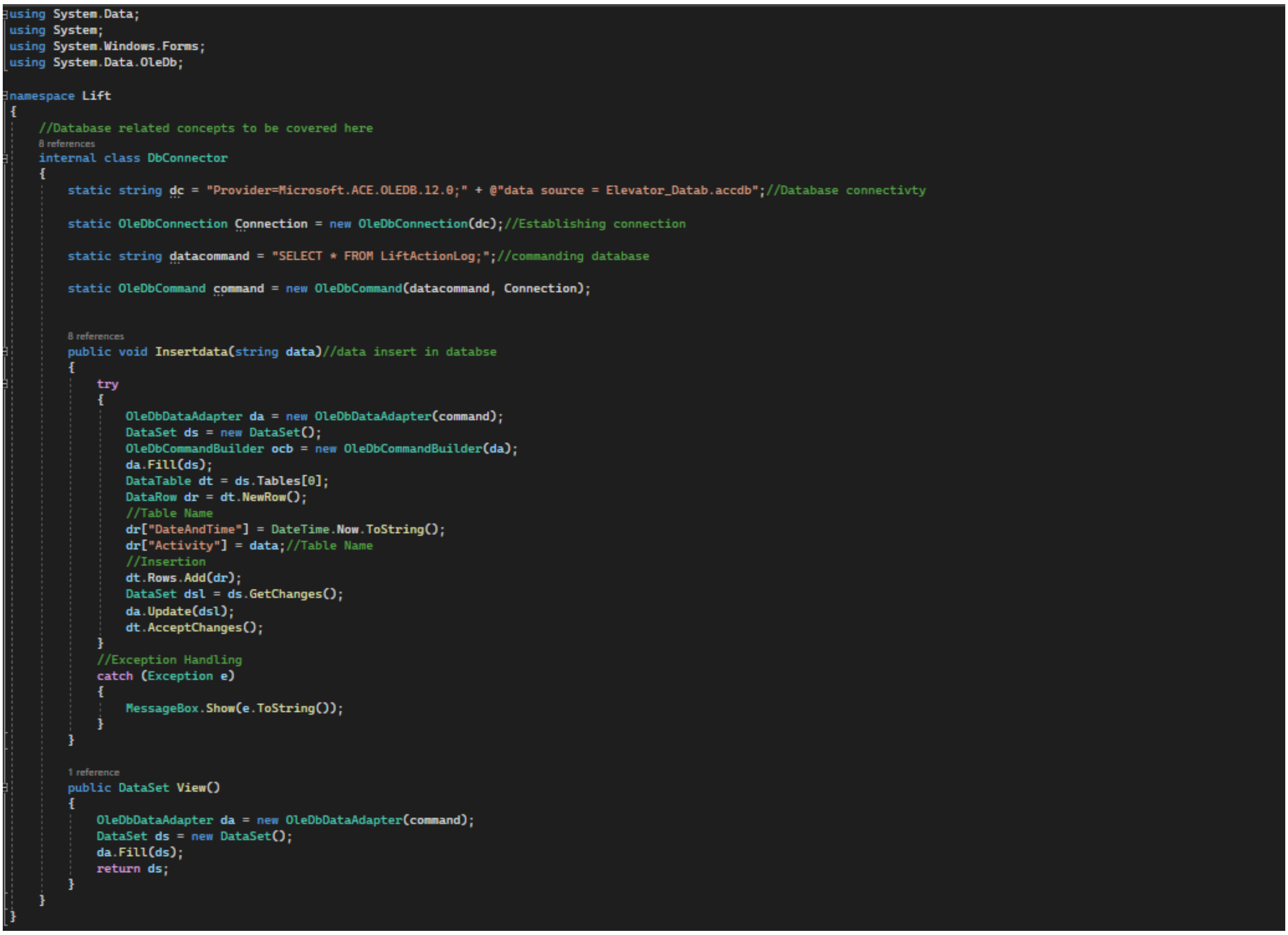
Task Number	Sub-tasks	Possible Marks	Self-assessment (completed Yes/No)	Reference to your testing report	Mark Awarded
Task 1	Complete GUI for Task 1	10	Yes	9	10
	Skeleton of event handlers in place for all buttons	10	Yes	10	10
Task 2	All event handlers are functional	10	Yes	11	10
Task 3	Database (DB) is designed and can be connected	5	Yes	14	5
	Log Information can be retrieved from DB and displayed in the GUI	5	Yes	14	5
	When the log button is pressed, log information is sent to and stored in the DB	5	Yes	14	5
	Use the disconnected model rather than connected model (Data source is updated via DataAdapters Update() method instead of ExecuteNonQuery() method)	5	Yes	14	5
	Using relative path instead of absolute path	5	Yes	14	5
	Avoiding any duplication among the event handlers over the database related functions	5	Yes	14	5
Task 4	Events described in Task 2 animated using delegation and timer	10	Yes	14	10
Task 5	Eliminating logical errors and handling exceptions with try and catch	5	Yes	14	5

	Optimise the efficiency of GUI by implementing multiple tasks concurrently via BackgroundWorker	5	Yes	14	5
	Use state patterns instead of if-else statements to accommodate future changes of the requirement	10	Yes	14	10
Task 6	Testing report	10	Yes	15	10
Total		100			100

Table 7:- Marking Matrix with Assessment

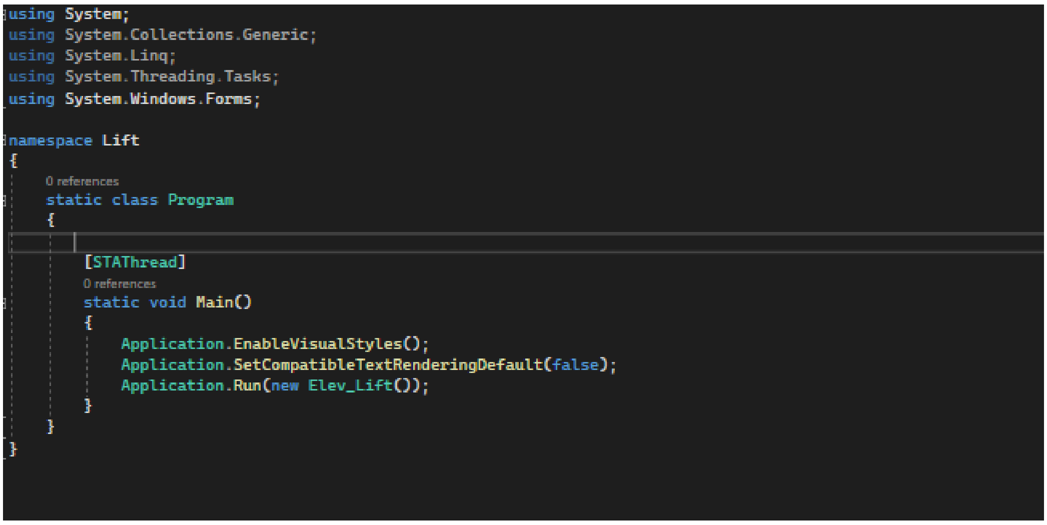
Appendix

- DbConnector.cs



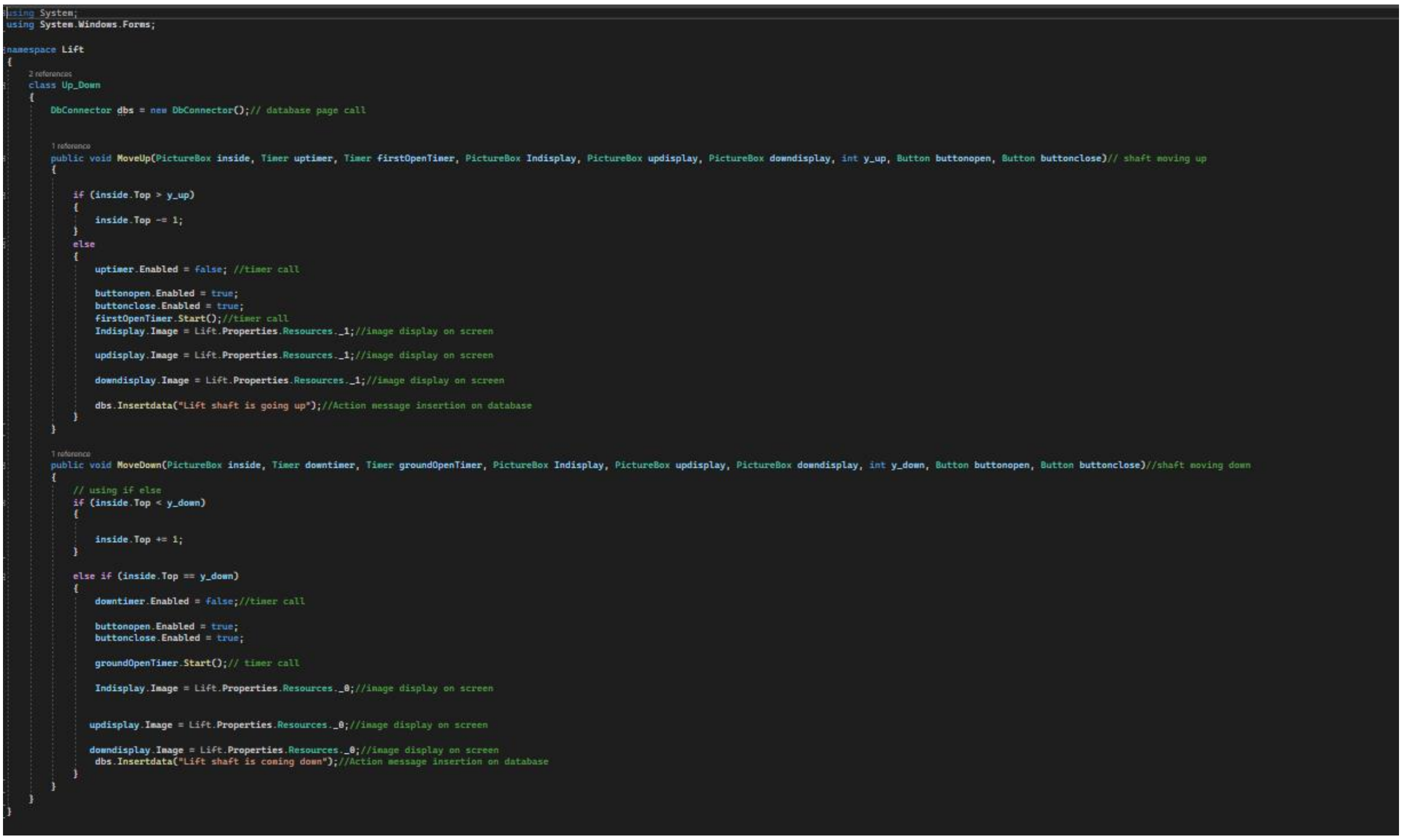
```
using System.Data;
using System;
using System.Windows.Forms;
using System.Data.OleDb;

namespace Lift
{
    //Database related concepts to be covered here
    8 references
    internal class DbConnector
    {
        static string dc = "Provider=Microsoft.ACE.OLEDB.12.0;" + @"data source = Elevator_Datab.accdb";//Database connectivty

        static OleDbConnection Connection = new OleDbConnection(dc);//Establishing connection

        static string datacommand = "SELECT * FROM LiftActionLog;";//commanding database

        static OleDbCommand command = new OleDbCommand(datacommand, Connection);


        8 references
        public void Insertdata(string data)//data insert in databse
        {
            try
            {
                OleDbDataAdapter da = new OleDbDataAdapter(command);
                DataSet ds = new DataSet();
                OleDbCommandBuilder ocb = new OleDbCommandBuilder(da);
                da.Fill(ds);
                DataTable dt = ds.Tables[0];
                DataRow dr = dt.NewRow();
                //Table Name
                dr["DateAndTime"] = DateTime.Now.ToString();
                dr["Activity"] = data;//Table Name
                //Insertion
                dt.Rows.Add(dr);
                DataSet dsl = ds.GetChanges();
                da.Update(dsl);
                dt.AcceptChanges();
            }
            //Exception Handling
            catch (Exception e)
            {
                MessageBox.Show(e.ToString());
            }
        }

        1 reference
        public DataSet View()
        {
            OleDbDataAdapter da = new OleDbDataAdapter(command);
            DataSet ds = new DataSet();
            da.Fill(ds);
            return ds;
        }
    }
}
```

- Program.cs



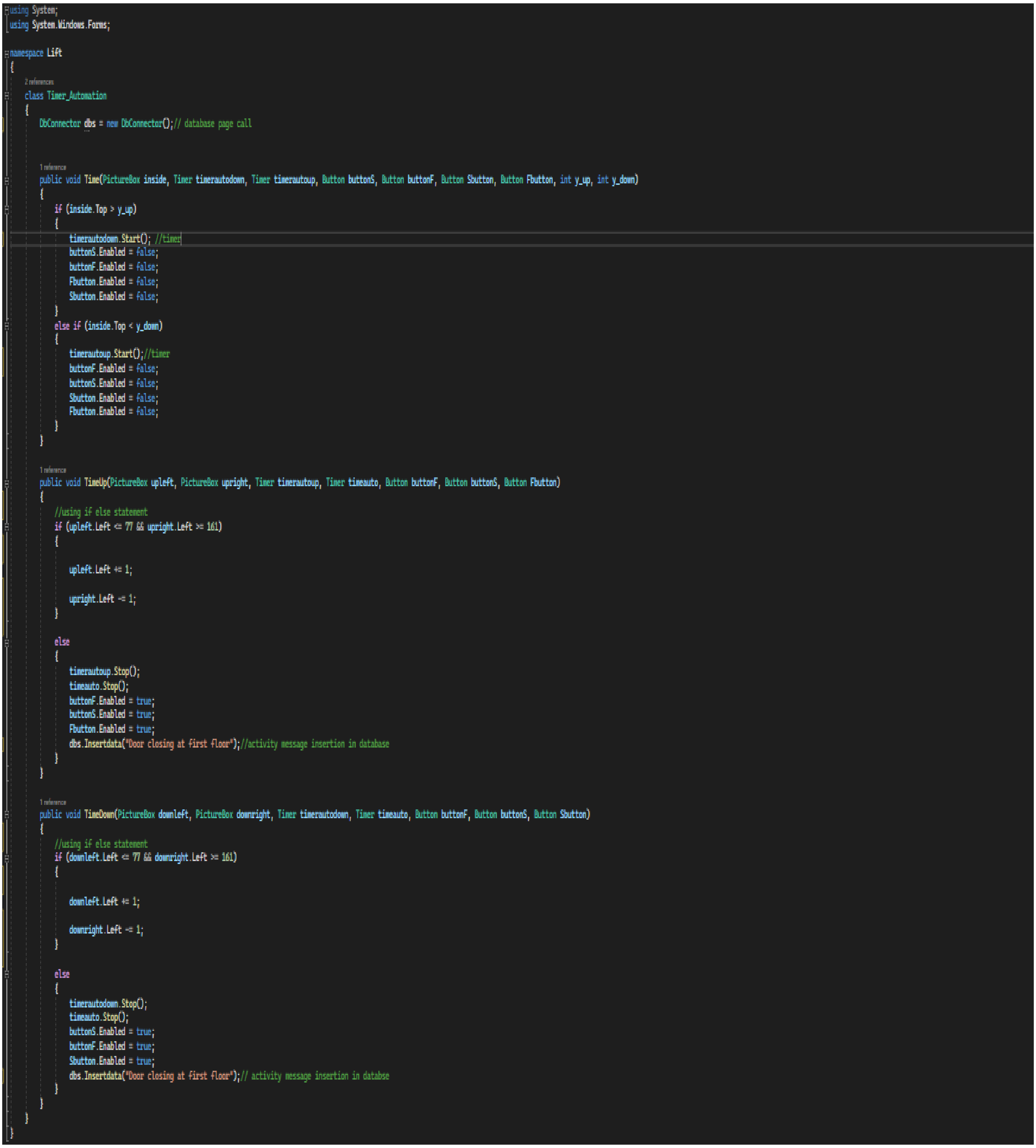
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Lift
{
    0 references
    static class Program
    {

        [STAThread]
        0 references
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Elev_Lift());
        }
    }
}
```

- Up_Down.cs



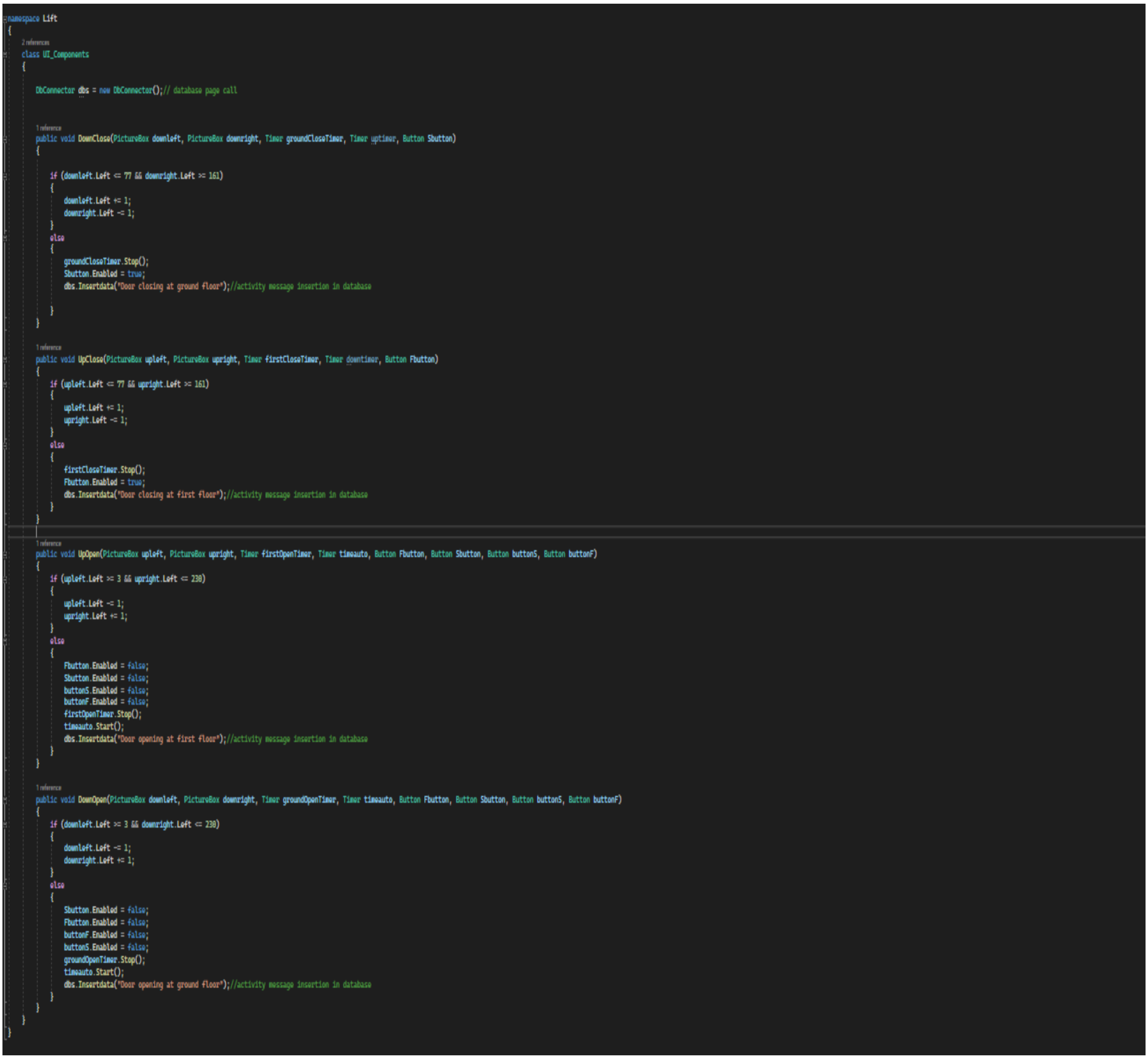
```
using System;
using System.Windows.Forms;

namespace Lift
{
    2 references
    class Up_Down
    {
        DbConnector dbs = new DbConnector();// database page call


        1 reference
        public void MoveUp(PictureBox inside, Timer uptimer, Timer firstOpenTimer, PictureBox Indisplay, PictureBox updisplay, PictureBox downdisplay, int y_up, Button buttonopen, Button buttonclose)// shaft moving up
        {

            if (inside.Top > y_up)
            {
                inside.Top -= 1;
            }
            else
            {
                uptimer.Enabled = false; //timer call

                buttonopen.Enabled = true;
                buttonclose.Enabled = true;
                firstOpenTimer.Start();//timer call
                Indisplay.Image = Lift.Properties.Resources._1;//image display on screen

                updisplay.Image = Lift.Properties.Resources._1;//image display on screen

                downdisplay.Image = Lift.Properties.Resources._1;//image display on screen

                dbs.Insertdata("Lift shaft is going up");//Action message insertion on database
            }
        }

        1 reference
        public void MoveDown(PictureBox inside, Timer downtimer, Timer groundOpenTimer, PictureBox Indisplay, PictureBox updisplay, PictureBox downdisplay, int y_down, Button buttonopen, Button buttonclose)//shaft moving down
        {
            // using if else
            if (inside.Top < y_down)
            {

                inside.Top += 1;
            }

            else if (inside.Top == y_down)
            {
                downtimer.Enabled = false;//timer call

                buttonopen.Enabled = true;
                buttonclose.Enabled = true;

                groundOpenTimer.Start();// timer call

                Indisplay.Image = Lift.Properties.Resources._0;//image display on screen


               updisplay.Image = Lift.Properties.Resources._0;//image display on screen

               downdisplay.Image = Lift.Properties.Resources._0;//image display on screen
                dbs.Insertdata("Lift shaft is coming down");//Action message insertion on database
            }
        }
    }
}
```

- Timer_Automation.cs



```
using System;
using System.Windows.Forms;

namespace Lift
{
    2 references
    class Timer_Automation
    {
        DbConnector dbs = new DbConnector();// database page call


        1 reference
        public void Time(PictureBox inside, Timer timerautodown, Timer timerautoup, Button buttonS, Button buttonF, Button Sbutton, Button Fbutton, int y_up, int y_down)
        {
            if (inside.Top > y_up)
            {
                timerautodown.Start(); //timer
                buttonS.Enabled = false;
                buttonF.Enabled = false;
                Fbutton.Enabled = false;
                Sbutton.Enabled = false;
            }
            else if (inside.Top < y_down)
            {
                timerautoup.Start();//timer
                buttonF.Enabled = false;
                buttonS.Enabled = false;
                Sbutton.Enabled = false;
                Fbutton.Enabled = false;
            }
        }

        1 reference
        public void TimeUp(PictureBox upleft, PictureBox upright, Timer timerautoup, Timer timeauto, Button buttonF, Button buttonS, Button Fbutton)
        {
            //using if else statement
            if (upleft.Left <= 77 && upright.Left >= 161)
            {

                upleft.Left += 1;

                upright.Left -= 1;
            }

            else
            {
                timerautoup.Stop();
                timeauto.Stop();
                buttonF.Enabled = true;
                buttonS.Enabled = true;
                Fbutton.Enabled = true;
                dbs.Insertdata("Door closing at first floor");//activity message insertion in database
            }
        }

        1 reference
        public void TimeDown(PictureBox downleft, PictureBox downright, Timer timerautodown, Timer timeauto, Button buttonF, Button buttonS, Button Sbutton)
        {
            //using if else statement
            if (downleft.Left <= 77 && downright.Left >= 161)
            {

                downleft.Left += 1;

                downright.Left -= 1;
            }

            else
            {
                timerautodown.Stop();
                timeauto.Stop();
                buttonS.Enabled = true;
                buttonF.Enabled = true;
                Sbutton.Enabled = true;
                dbs.Insertdata("Door closing at first floor");// activity message insertion in databse
            }
        }
    }
}
```

- UI_Components.cs



```
namespace Lift
{
    2 references
    class UI_Components
    {

        DbConnector dbs = new DbConnector();// database page call


        1 reference
        public void DownClose(PictureBox downleft, PictureBox downright, Timer groundCloseTimer, Timer uptimer, Button Sbutton)
        {

            if (downleft.Left <= 77 && downright.Left >= 161)
            {
                downleft.Left += 1;
                downright.Left -= 1;
            }
            else
            {
                groundCloseTimer.Stop();
                Sbutton.Enabled = true;
                dbs.Insertdata("Door closing at ground floor");//activity message insertion in database

            }
        }

        1 reference
        public void UpClose(PictureBox upleft, PictureBox upright, Timer firstCloseTimer, Timer downtimer, Button Fbutton)
        {
            if (upleft.Left <= 77 && upright.Left >= 161)
            {
                upleft.Left += 1;
                upright.Left -= 1;
            }
            else
            {
                firstCloseTimer.Stop();
                Fbutton.Enabled = true;
                dbs.Insertdata("Door closing at first floor");//activity message insertion in database
            }
        }

        1 reference
        public void UpOpen(PictureBox upleft, PictureBox upright, Timer firstOpenTimer, Timer timeauto, Button Fbutton, Button Sbutton, Button button5, Button buttonF)
        {
            if (upleft.Left >= 3 && upright.Left <= 238)
            {
                upleft.Left -= 1;
                upright.Left += 1;
            }
            else
            {
                Fbutton.Enabled = false;
                Sbutton.Enabled = false;
                button5.Enabled = false;
                buttonF.Enabled = false;
                firstOpenTimer.Stop();
                timeauto.Start();
                dbs.Insertdata("Door opening at first floor");//activity message insertion in database
            }
        }

        1 reference
        public void DownOpen(PictureBox downleft, PictureBox downright, Timer groundOpenTimer, Timer timeauto, Button Fbutton, Button Sbutton, Button button5, Button buttonF)
        {
            if (downleft.Left >= 3 && downright.Left <= 238)
            {
                downleft.Left -= 1;
                downright.Left += 1;
            }
            else
            {
                Sbutton.Enabled = false;
                Fbutton.Enabled = false;
                buttonF.Enabled = false;
                button5.Enabled = false;
                groundOpenTimer.Stop();
                timeauto.Start();
                dbs.Insertdata("Door opening at ground floor");//activity message insertion in database
            }
        }
    }
}
```