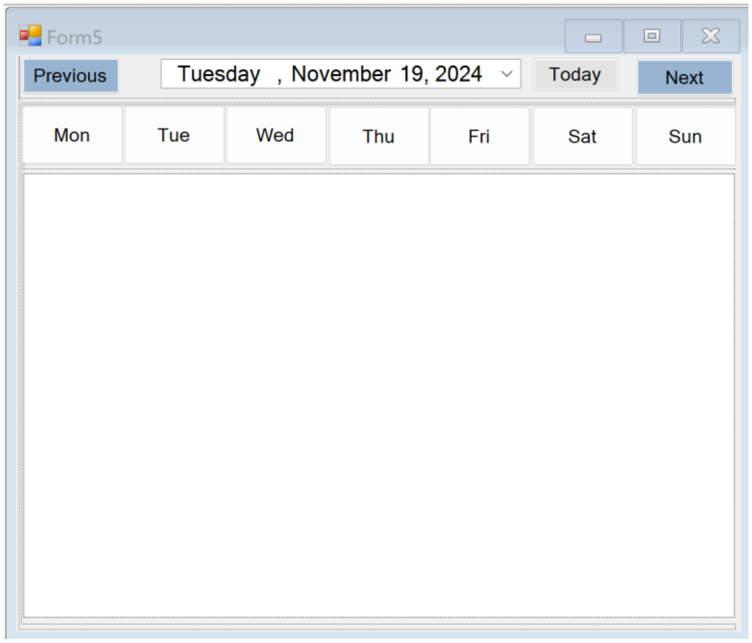
Form 5.

Requirements:

Use DateTimePicker to select the date.

Create a list of Buttons (created by code) to display the days of the month. You can move to the previous month (Previous) or next month (Next) to see the days of that month.

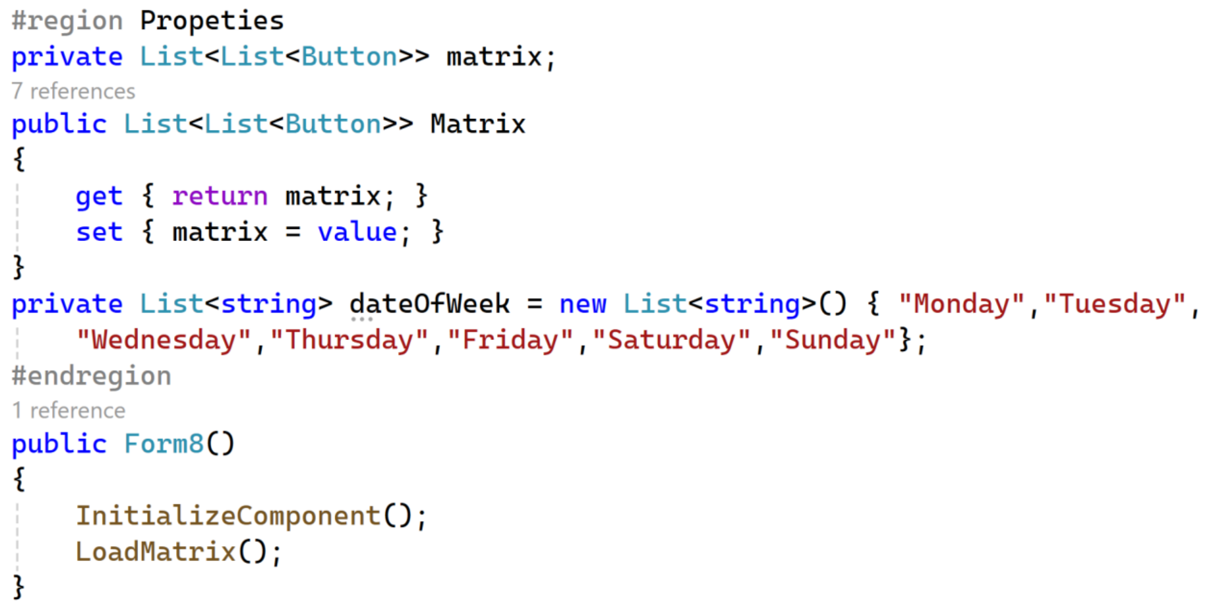
Design:



Code:

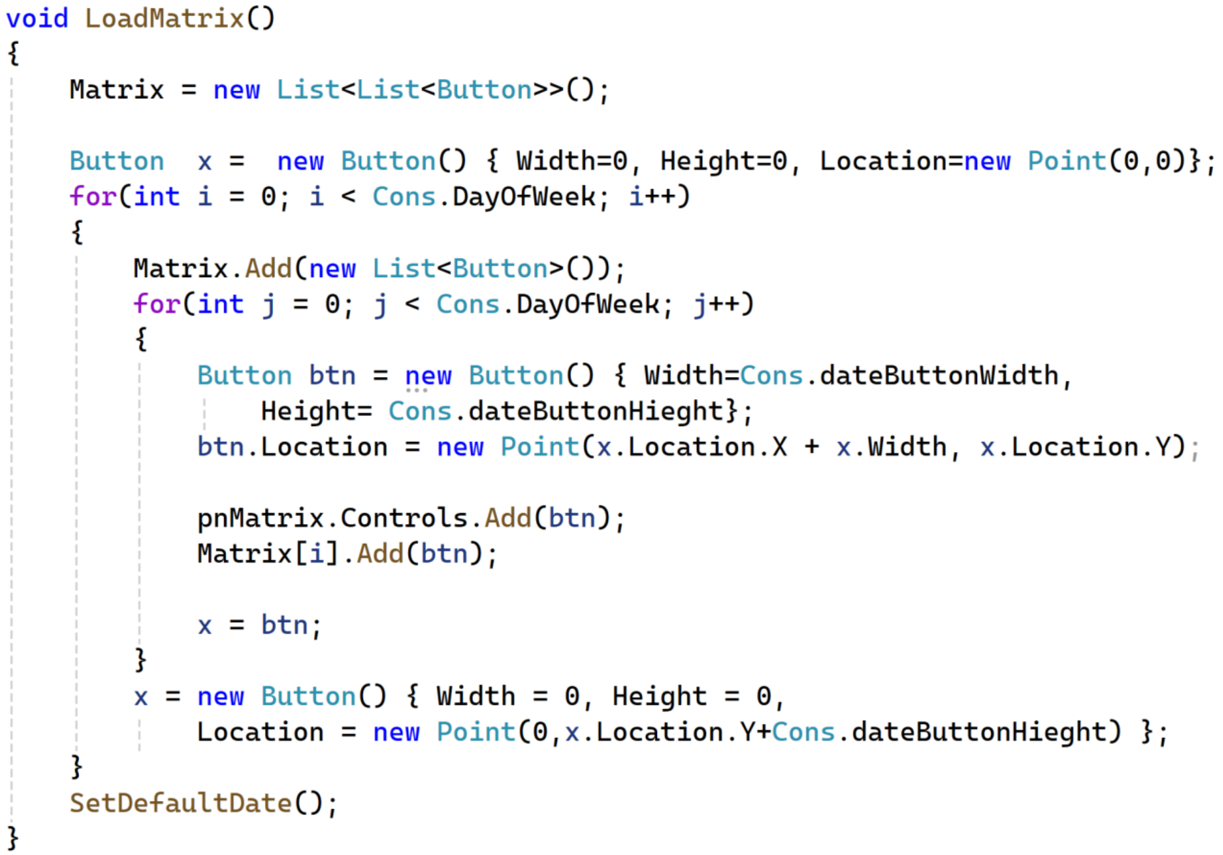
Step 0. Declare consts

Step 1. Create #region



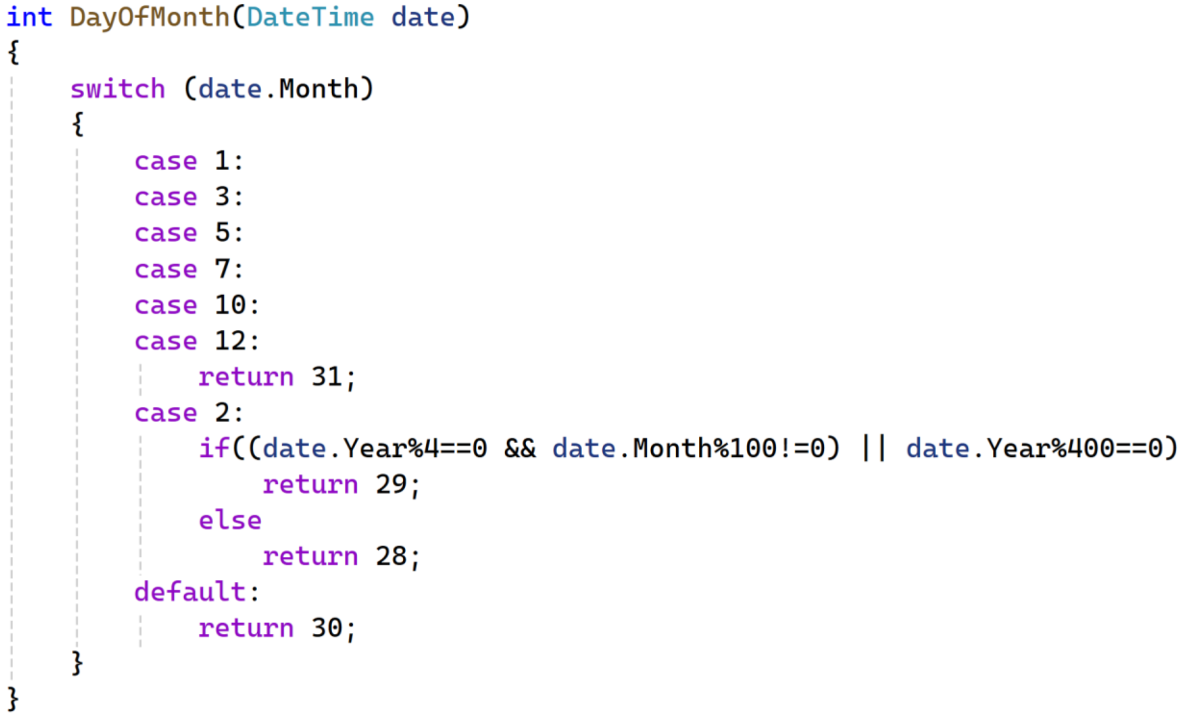
```
#region Propeties
private List<List<Button>> matrix;
7 references
public List<List<Button>> Matrix
{
    get { return matrix; }
    set { matrix = value; }
}
private List<string> dateOfWeek = new List<string>() { "Monday","Tuesday",
    "Wednesday","Thursday","Friday","Saturday","Sunday"};
#endregion
1 reference
public Form8()
{
    InitializeComponent();
    LoadMatrix();
}
```

Step 2.



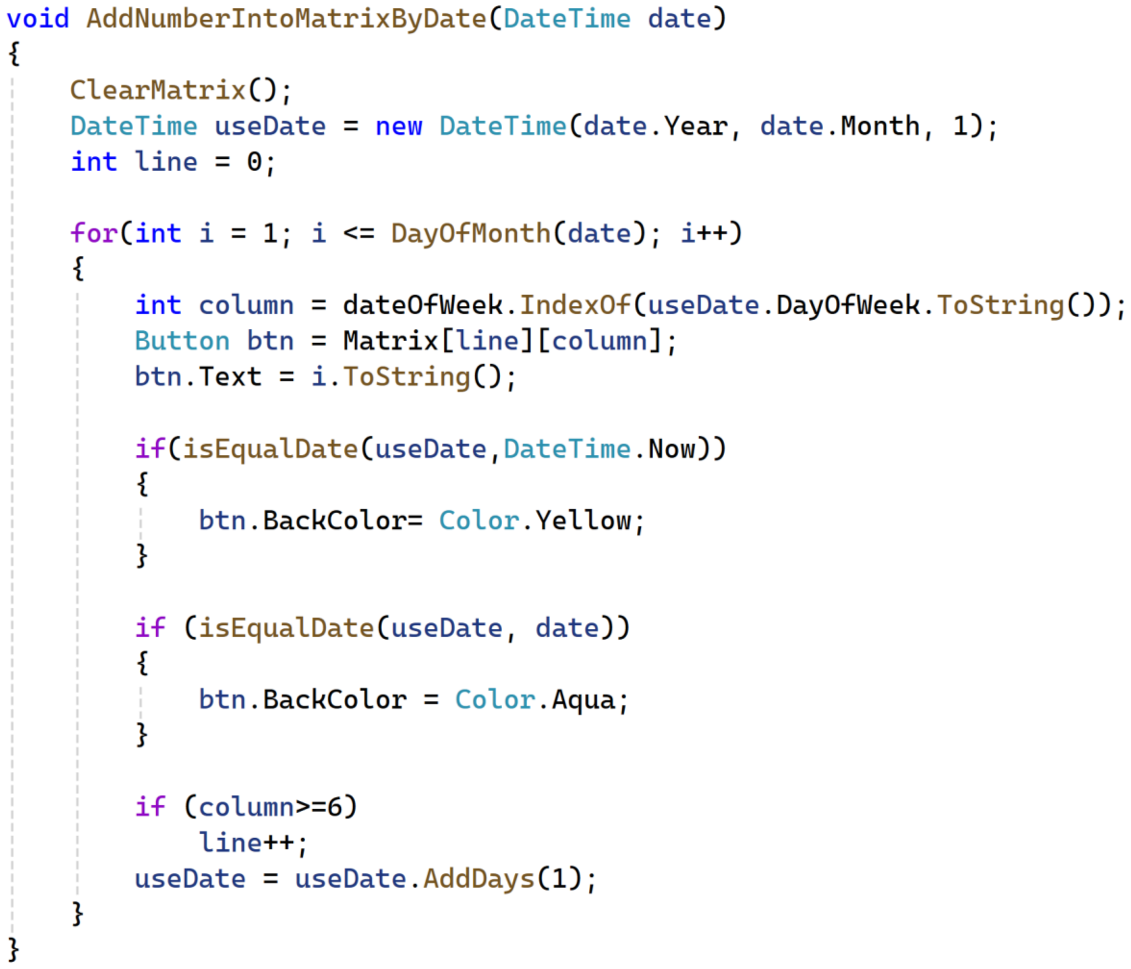
```
void LoadMatrix()
{
    Matrix = new List<List<Button>>();

    Button  x =  new Button() { Width=0, Height=0, Location=new Point(0,0)};
    for(int i = 0; i < Cons.DayOfWeek; i++)
    {
        Matrix.Add(new List<Button>());
        for(int j = 0; j < Cons.DayOfWeek; j++)
        {
            Button btn = new Button() { Width=Cons.dateButtonWidth,
                Height= Cons.dateButtonHieght};
            btn.Location = new Point(x.Location.X + x.Width, x.Location.Y);

            pnMatrix.Controls.Add(btn);
            Matrix[i].Add(btn);

            x = btn;
        }
        x = new Button() { Width = 0, Height = 0,
            Location = new Point(0,x.Location.Y+Cons.dateButtonHieght) };
    }
    SetDefaultDate();
}
```

Step 3.



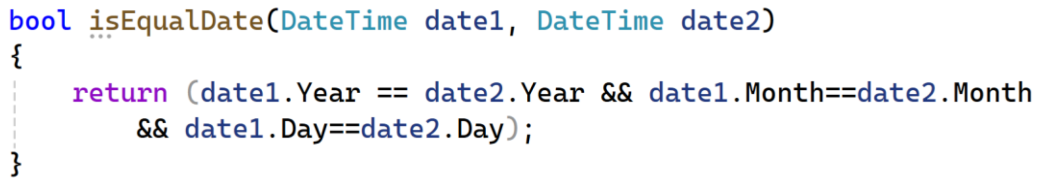
```
int DayOfMonth(DateTime date)
{
    switch (date.Month)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 10:
        case 12:
            return 31;
        case 2:
            if((date.Year%4==0 && date.Month%100!=0) || date.Year%400==0)
                return 29;
            else
                return 28;
        default:
            return 30;
    }
}
```

Step 4.



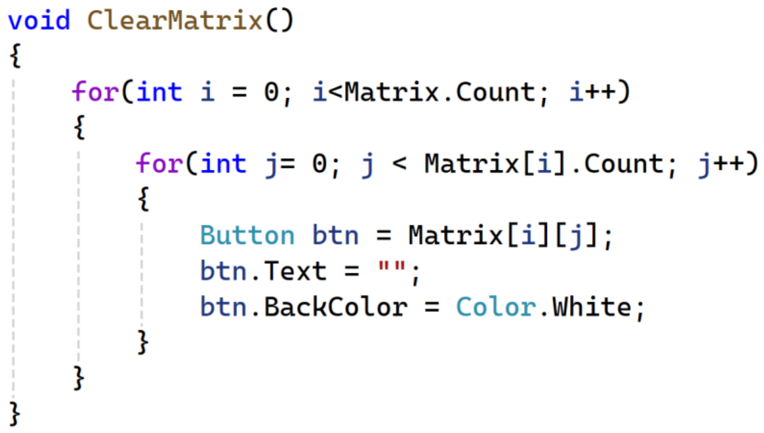
```
void AddNumberIntoMatrixByDate(DateTime date)
{
    ClearMatrix();
    DateTime useDate = new DateTime(date.Year, date.Month, 1);
    int line = 0;

    for(int i = 1; i <= DayOfMonth(date); i++)
    {
        int column = dateOfWeek.IndexOf(useDate.DayOfWeek.ToString());
        Button btn = Matrix[line][column];
        btn.Text = i.ToString();

        if(isEqualDate(useDate,DateTime.Now))
        {
            btn.BackColor= Color.Yellow;
        }

        if (isEqualDate(useDate, date))
        {
            btn.BackColor = Color.Aqua;
        }

        if (column>=6)
            line++;
        useDate = useDate.AddDays(1);
    }
}
```

Step 5.



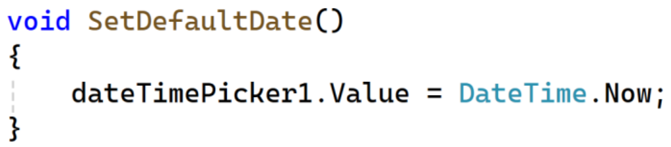
```
bool isEqualDate(DateTime date1, DateTime date2)
{
    return (date1.Year == date2.Year && date1.Month==date2.Month
        && date1.Day==date2.Day);
}
```

Step 6.



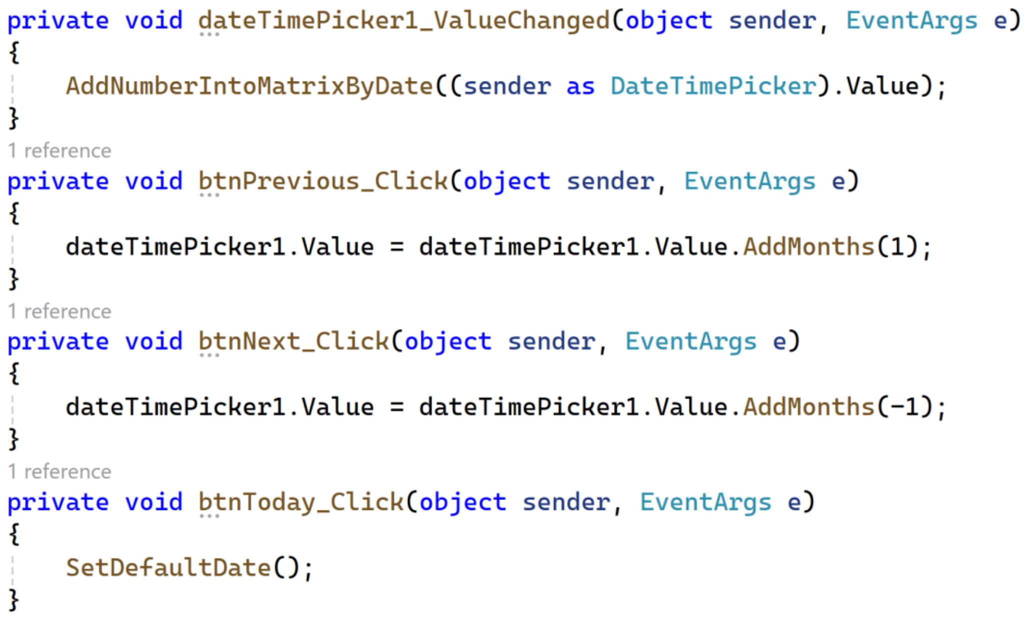
```
void ClearMatrix()
{
    for(int i = 0; i<Matrix.Count; i++)
    {
        for(int j= 0; j < Matrix[i].Count; j++)
        {
            Button btn = Matrix[i][j];
            btn.Text = "";
            btn.BackColor = Color.White;
        }
    }
}
```

Step 7.



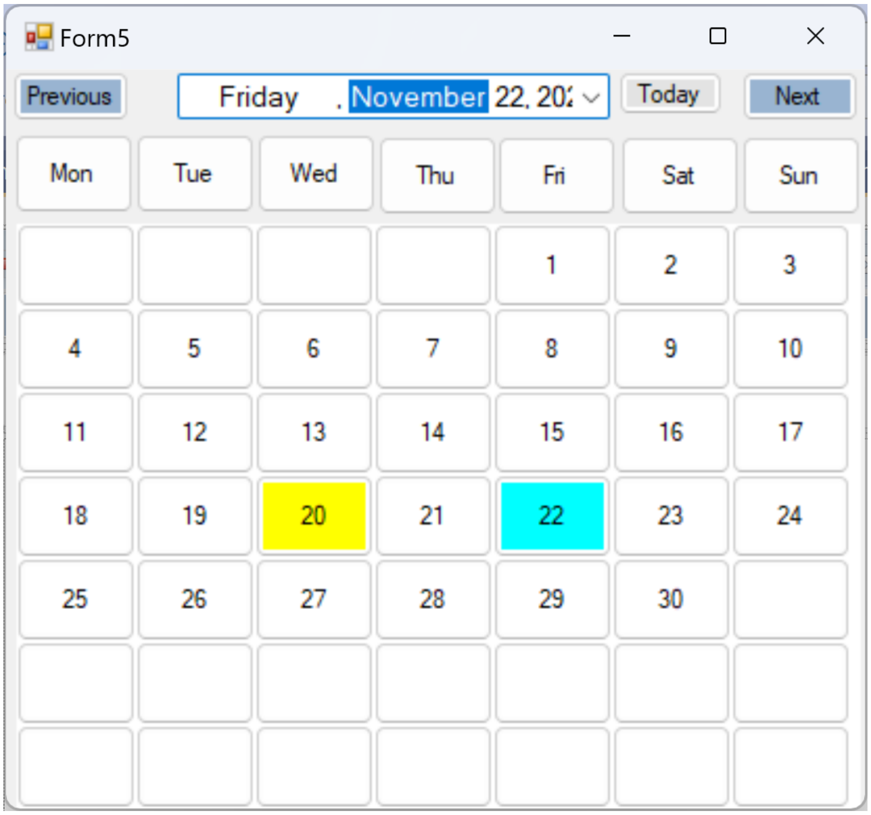
```
void SetDefaultDate()
{
    dateTimePicker1.Value = DateTime.Now;
}
```

Step 8.



```
private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
    AddNumberIntoMatrixByDate((sender as DateTimePicker).Value);
}
1 reference
private void btnPrevious_Click(object sender, EventArgs e)
{
    dateTimePicker1.Value = dateTimePicker1.Value.AddMonths(1);
}
1 reference
private void btnNext_Click(object sender, EventArgs e)
{
    dateTimePicker1.Value = dateTimePicker1.Value.AddMonths(-1);
}
1 reference
private void btnToday_Click(object sender, EventArgs e)
{
    SetDefaultDate();
}
```

Result.



Friday , November 22, 202

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Form 6.

Movie theater ticket sales program:

The theater has 4 rows of seats, each row has 5 seats, the seats are numbered from 1 to 20 and are divided into 4 rows as shown (picture above):

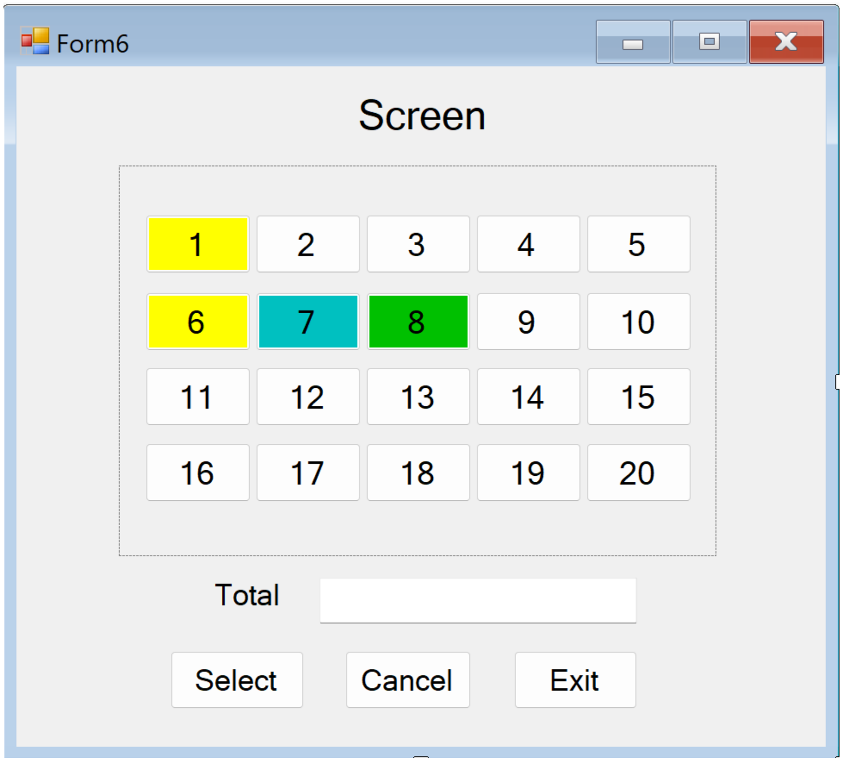
Row 1 (seats 1-5), Ticket price 30,000 VND

Row 2 (seats 6-10), Ticket price 40,000 VND

Row 3 (seats 11-15), Ticket price 50,000 VND

Row 4 (seats 16-20), Ticket price 80,000 VND

The Form presents a seating chart for users to choose the seat they want to buy. This chart also shows the seats that have been sold and the seats that have not been sold by showing different colors (seats that have not been sold are white, seats that have been sold are yellow).



1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

When the user clicks on a location on the diagram:

- If this is an unsold location, the color of this location changes to blue to indicate that this is the currently selected location.
- If this is a currently selected location (blue), the color of this location changes back to white

- If this is a sold location (yellow), a Message box appears to notify the user that the ticket at this location has been sold. After selecting the locations, the user can click on the SELECT or CANCEL button

If clicking on the SELECT button:

- Change the color of the selected locations (blue) on the diagram to yellow (indicating the location where tickets have been sold)
- Print to a Label the total amount to be paid for the number of tickets purchased (depending on the selected locations)

If clicking on the CANCEL button:

- Change the color of the selected locations (blue) on the diagram back to white
- Print to the Label the value 0