ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Báo cáo bài cuối kì lập trình ứng dụng

Sinh viên: Trần Bảo Thịnh

MSV:1902516

Lớp:DACLC2

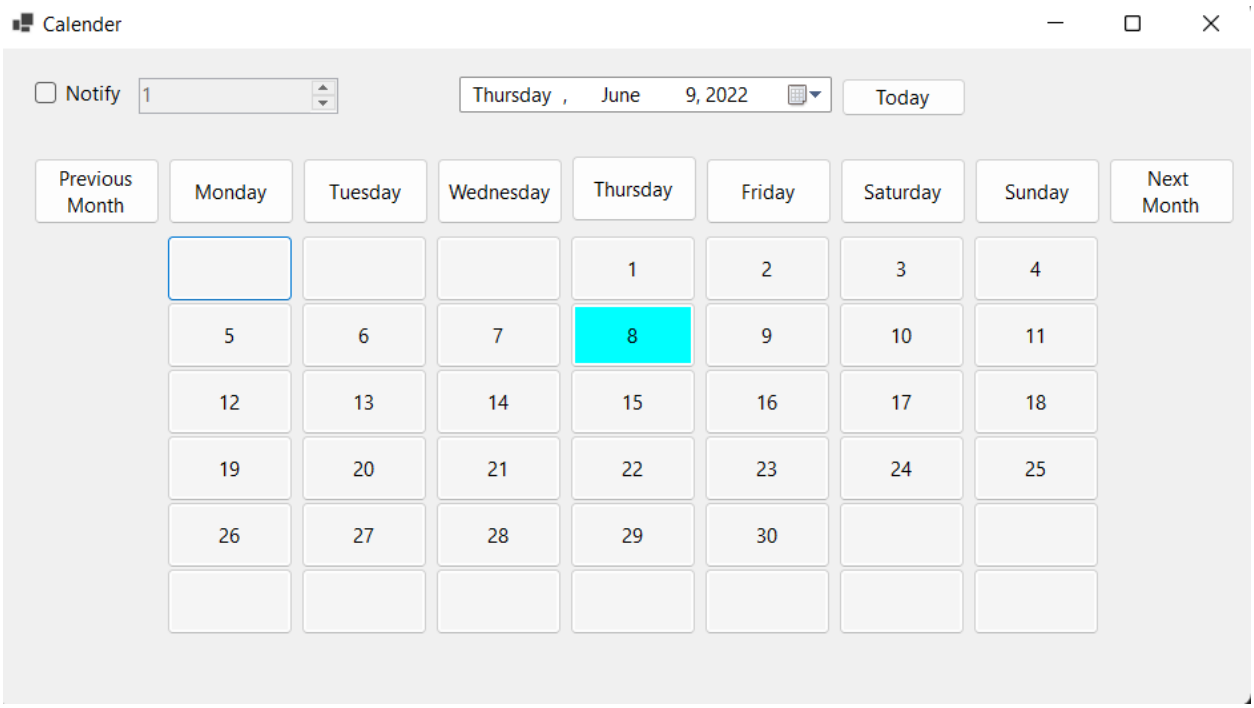
1. Giới thiệu:

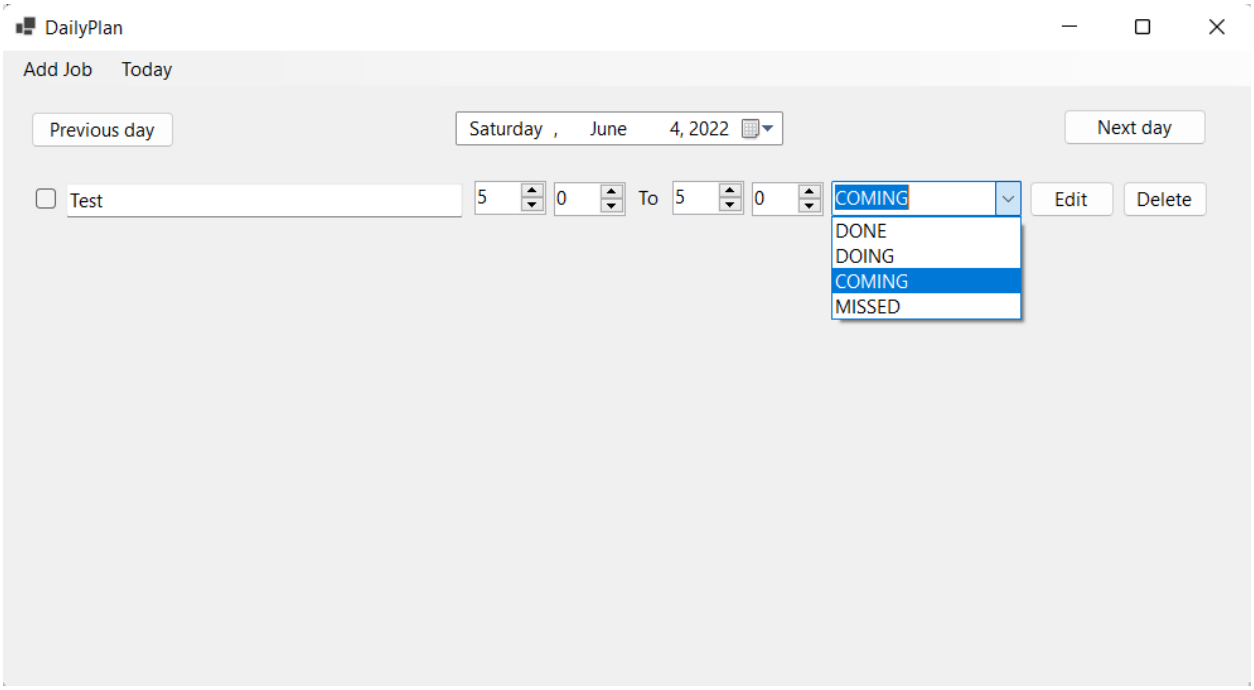
- Hiện tại em đang làm một app về lập lịch làm việc, xem ngày trong tuần/tháng/năm và hiện thông báo.

- Tên App: Calender

- Phần mềm thực hiện: Visual Studio 2022 + Advanced_Installer.

- Giao diện:





- Icon của phần mềm trên Window: 

- Em hiện đã gửi phần cài đặt và code cho thầy ạ.

- Đồng thời em đã tạo 1 file Readme để hướng dẫn sử dụng ạ.

2. Các bước để thực hiện ý tưởng app (Brainstorm):

- Thiết kế một app lịch đơn giản dễ tiếp cận và thực tiễn với người dùng

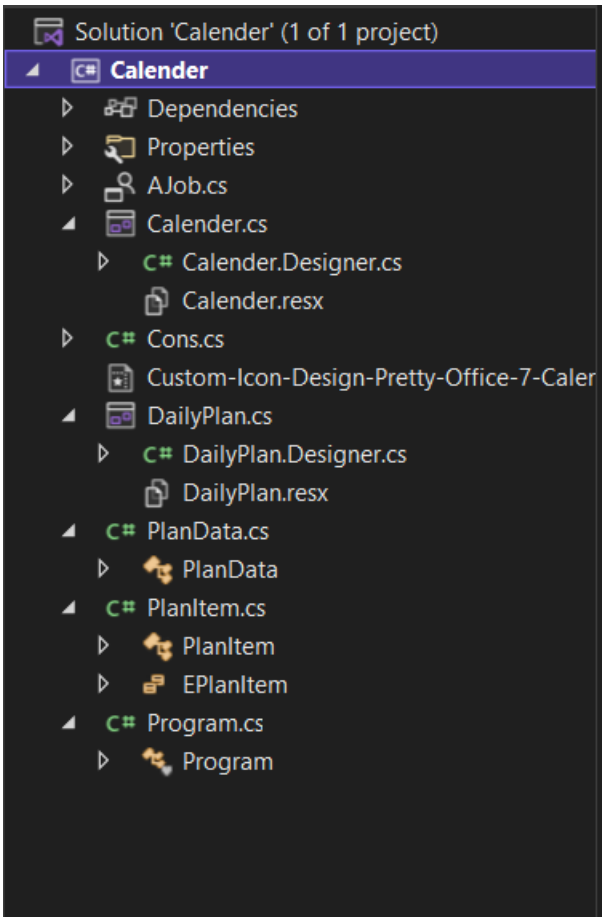
- Thiết kế mô hình lịch có chứa các thứ ngày có thể chuyển đổi giữa ngày tháng năm để thuận tiện cho người sử dụng có thông báo và một nút quay về.

- Thiết kế một mô hình để hiện các công việc.

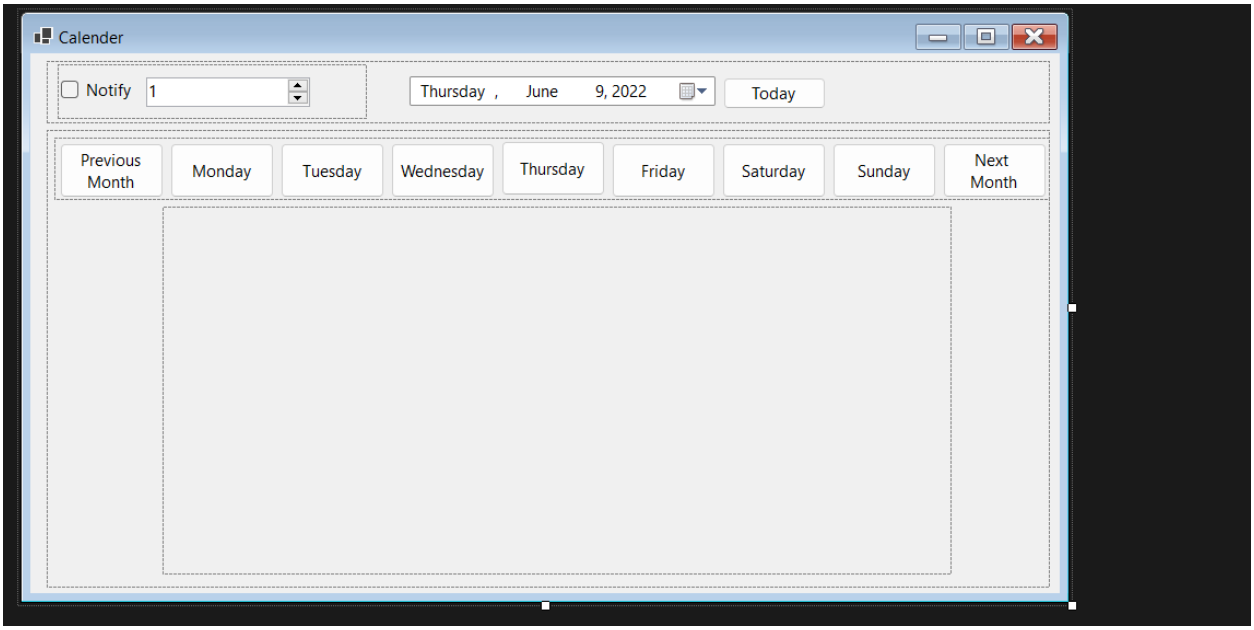
- Thiết kế một thanh công việc có thể chỉnh sửa cũng như xóa

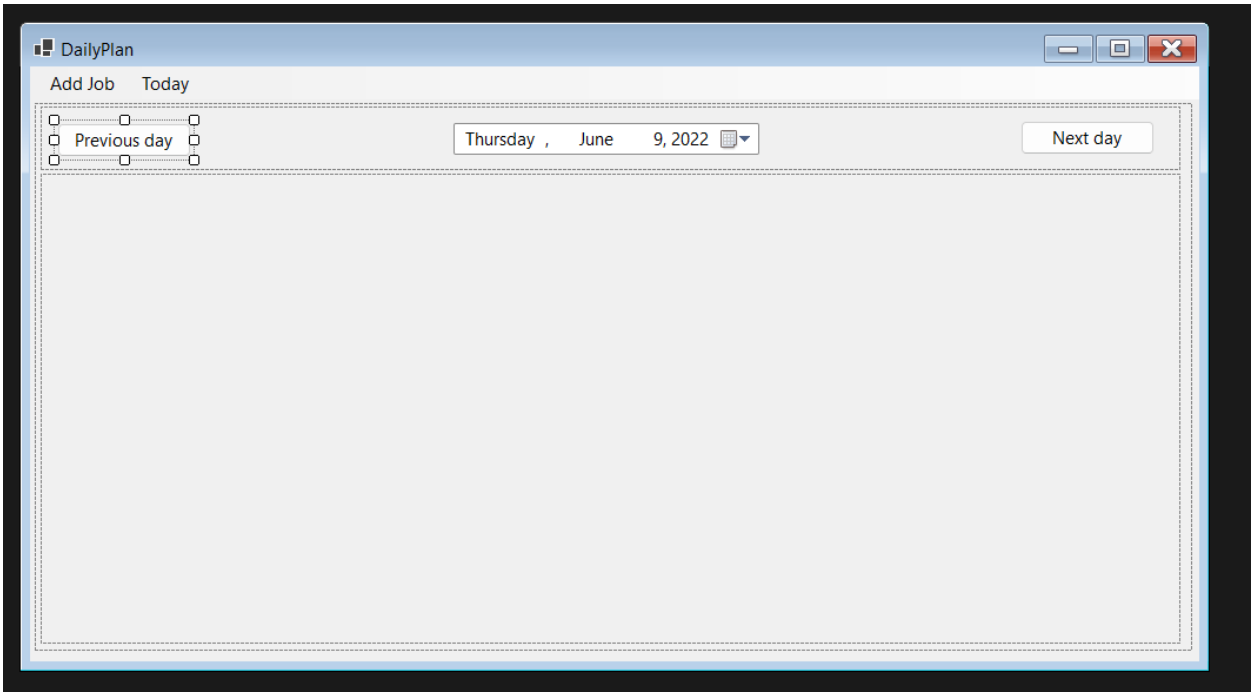
3. Code:

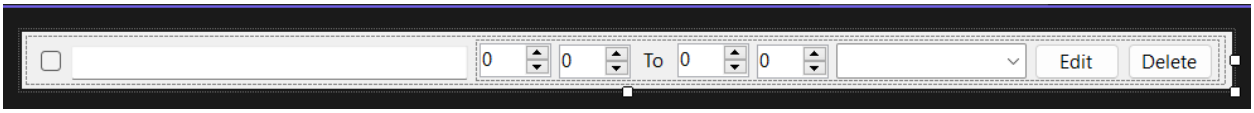
* Khung chương trình


Solution 'Calender' (1 of 1 project)
Calender
Dependencies
Properties
AJob.cs
Calender.cs
Calender.Designer.cs
Calender.resx
Cons.cs
Custom-Icon-Design-Pretty-Office-7-Caler
DailyPlan.cs
DailyPlan.Designer.cs
DailyPlan.resx
PlanData.cs
PlanData
PlanItem.cs
PlanItem
EPlanItem
Program.cs
Program

* Thiết kế:

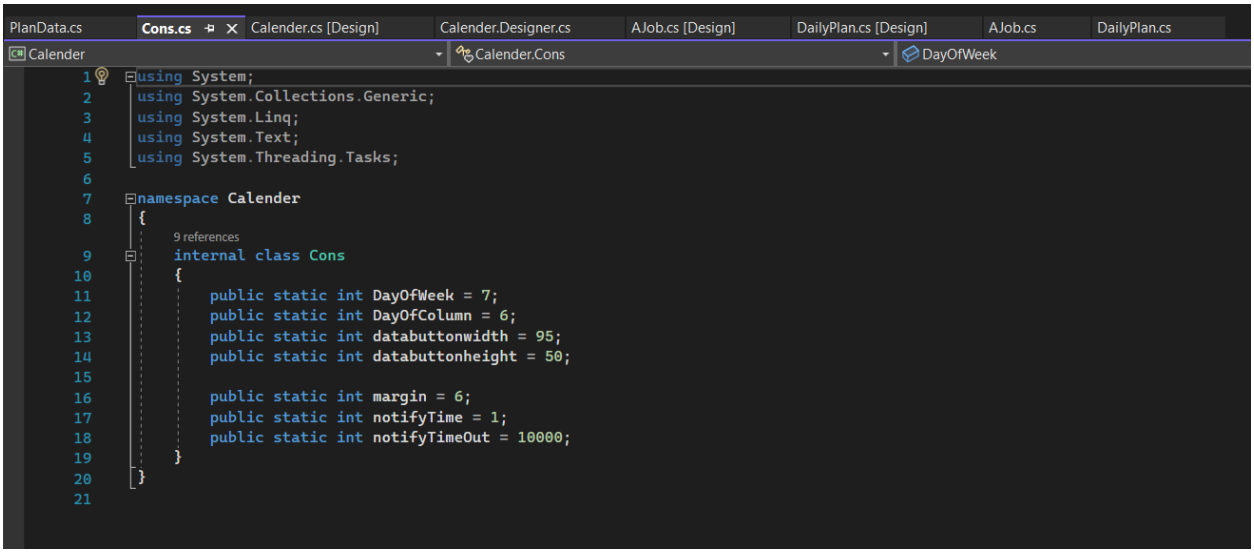






* Phần Con.cs

- Đây là các biến được sử dụng lại nhiều lần nên được lưu chung:



* Phần Calender.cs (phần chính)

- Code chạy chương trình và lưu các thiết lập lịch trước đó vào 1 file data + hẹn giờ notify

```
public partial class Calender : Form
{
    Peoperties
    1 reference
    public Calender()
    {
        InitializeComponent();

        tmNotify.Start();
        appTime = 0;
        LoadMatrix();

        try
        {
            Job = DeserializeFromXML(filePath) as PlanData;
        }
        catch
        {
            SetDefaultJob();
        }
    }
```

- Code tạo button ngày/tháng trong Thiết kế

```
void LoadMatrix()
{

    Matrix = new List<List<Button>>();

    Button oldbtn = new Button() { Width = 0, Height = 0, Location = new Point(0, 0) };
    for (int i = 0; i < Cons.DayOfColumn; i++)
    {
        Matrix.Add(new List<Button>());

        for (int j = 0; j < Cons.DayOfWeek; j++)
        {
            Button btn = new Button() { Width = Cons.databuttonwidth, Height = Cons.databuttonheight}; ;
            btn.Location = new Point(oldbtn.Location.X + oldbtn.Width + Cons.margin, oldbtn.Location.Y);
            btn.Click+= btn_Click;

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

            oldbtn = btn;
        }
        oldbtn = new Button() { Width = 0, Height = 0, Location = new Point(0, oldbtn.Location.Y + Cons.databuttonheight) };
    }

    SetDefaultDate();
}
```

- Code tạo số ngày/tháng vào button

```
1 reference
void AddNumberIntoMatrixByDate(DateTime date)
{
    ClearMatrix();
    DateTime useDate = new DateTime(date.Year, date.Month, 2);

    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);
    }
}
```

- Kiểm tra năm nhuận

```
1 reference
int DayOfMonth(DateTime date)
{
    switch (date.Month)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            return 31;
        case 2:
            if ((date.Year % 4 == 0 && date.Year % 100 != 0) || date.Year % 400 == 0)
                return 29;
            else
                return 28;
        default:
            return 30; ;
    }
}
```

- Code hiện bảng kế hoạch trong ngày khi click vào button

```
1 reference
void btn_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty((sender as Button).Text))
    {
        return;
    }
    DailyPlan daily = new DailyPlan(new DateTime(dtpkDate.Value.Year, dtpkDate.Value.Month, Convert.ToInt32((sender as Button).Text)), Job);
    daily.ShowDialog();
}
```

- Thiết lập để sử dụng phần giao diện

```
1 reference
private void dtpkDate_ValueChanged(object sender, EventArgs e)
{
    AddNumberIntoMatrixByDate((sender as DateTimePicker).Value);
}

1 reference
private void btnPrevious_Click(object sender, EventArgs e)
{
    dtpkDate.Value = dtpkDate.Value.AddMonths(-1);
}

1 reference
private void btnNext_Click(object sender, EventArgs e)
{
    dtpkDate.Value = dtpkDate.Value.AddMonths(1);
}

1 reference
private void btnToday_Click(object sender, EventArgs e)
{
    SetDefaultDate();
}
```

- Code lưu thông tin lại khi tắt phần mềm

```
private void SerializeToXML(object data, string filePath)
{
    FileStream fs = new FileStream(filePath, FileMode.OpenOrCreate, FileAccess.Write);
    XmlSerializer sr = new XmlSerializer(typeof(PlanData));

    sr.Serialize(fs, data);

    fs.Close();
}

1 reference
private object DeserializeFromXML(string filePath)
{
    FileStream fs = new FileStream(filePath, FileMode.Open, FileAccess.Read);
    try
    {
        XmlSerializer sr = new XmlSerializer(typeof(PlanData));

        object result = sr.Deserialize(fs);
        fs.Close();
        return result;
    }
    catch (Exception e)
    {
        fs.Close();
        throw new NotImplementedException();
    }
}

1 reference
private void Calender_FormClosed(object sender, FormClosedEventArgs e)
{
    SerializeToXML(Job, filePath);
}
```

- Thiết lập bộ tự đếm và Notify

```
1 reference
private void tmNotify_Tick(object sender, EventArgs e)
{
    if (!ckbNotify.Checked)
        return;

    AppTime++;

    if(AppTime < Cons.notifyTime)
        return;

    if (Job == null || Job.Job == null)
        return;

    DateTime currentDate = DateTime.Now;
    List<PlanItem> todayjobs = Job.Job.Where(p => p.Date.Year == currentDate.Year && p.Date.Month == currentDate.Month && p.Date.Day == currentDate.Day).ToList();
    NotifyIcon.ShowBalloonTip(Cons.notifyTimeOut, "Job Schedules", string.Format("you have {0} jobs today", todayjobs.Count), ToolTipIcon.Info);

    AppTime = 0;
}

1 reference
private void numNotify_ValueChanged(object sender, EventArgs e)
{
    Cons.notifyTime = (int)nmNotify.Value;
}

1 reference
private void ckbNotify_CheckedChanged(object sender, EventArgs e)
{
    nmNotify.Enabled = ckbNotify.Checked;
}
```

* Phần Job.cs

- Hiện thông tin

```
1 reference
void ShowInfo()
{
    txbJob.Text = Job.Job;
    nmfromhour.Value = Job.FromTime.X;
    nmfromminutes.Value = Job.FromTime.Y;
    nmtohour.Value = Job.ToTime.X;
    nmtominutes.Value = Job.ToTime.Y;
    cbStatus.SelectedIndex = PlanItem.ListStatus.IndexOf(Job.Status);
    ckbDone.Checked = PlanItem.ListStatus.IndexOf(Job.Status) == (int)EPlanItem.DONE ? true : false;
}
```

- Thiết lập sử dụng giao diện

```
1 reference
private void btnDelete_Click(object sender, EventArgs e)
{
    if (deleted != null)
        deleted(this, new EventArgs());
}

1 reference
private void btnEdit_Click(object sender, EventArgs e)
{
    Job.Job = txbJob.Text;
    Job.FromTime = new Point((int)nmfromhour.Value, (int)nmfromminutes.Value);
    Job.ToTime = new Point((int)nmtohour.Value, (int)nmtominutes.Value);
    Job.Status = PlanItem.ListStatus[cbStatus.SelectedIndex];

    if (edited != null)
        edited(this, new EventArgs());
}

1 reference
private void ckbDone_CheckedChanged(object sender, EventArgs e)
{
    cbStatus.SelectedIndex = ckbDone.Checked ? (int)EPlanItem.DONE : (int)EPlanItem.DOING;
}
```

* Phần DailyPlan

- Khởi chạy khi click vào 1 ngày

```
1 reference
public DailyPlan(DateTime date,PlanData job)
{
    InitializeComponent();

    this.Date = date;
    this.Job = job;

    fPanel.Width = pnlJob.Width;
    fPanel.Height = pnlJob.Height;
    pnlJob.Controls.Add(fPanel);

    dtpkDate.Value = Date;
}
```

- Hiện công việc theo ngày

```
void ShowJobByDate(DateTime date)
{
    fPanel.Controls.Clear();
    if (Job != null && Job.Job != null)
    {
        List<PlanItem> todayJob = GetJobByDay(date);
        for (int i = 0; i < GetJobByDay(date).Count; i++)
        {
            AddJob(todayJob[i]);
        }
    }
}
```

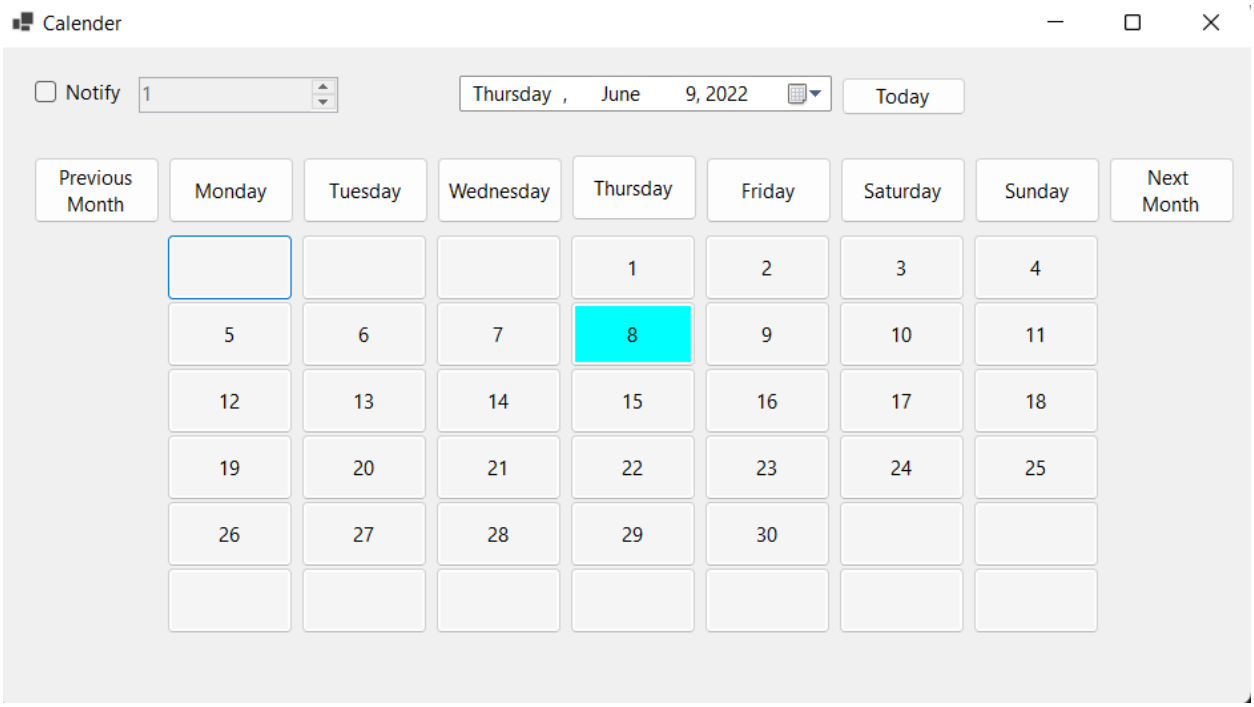
- Thiết lập giao diện sử dụng

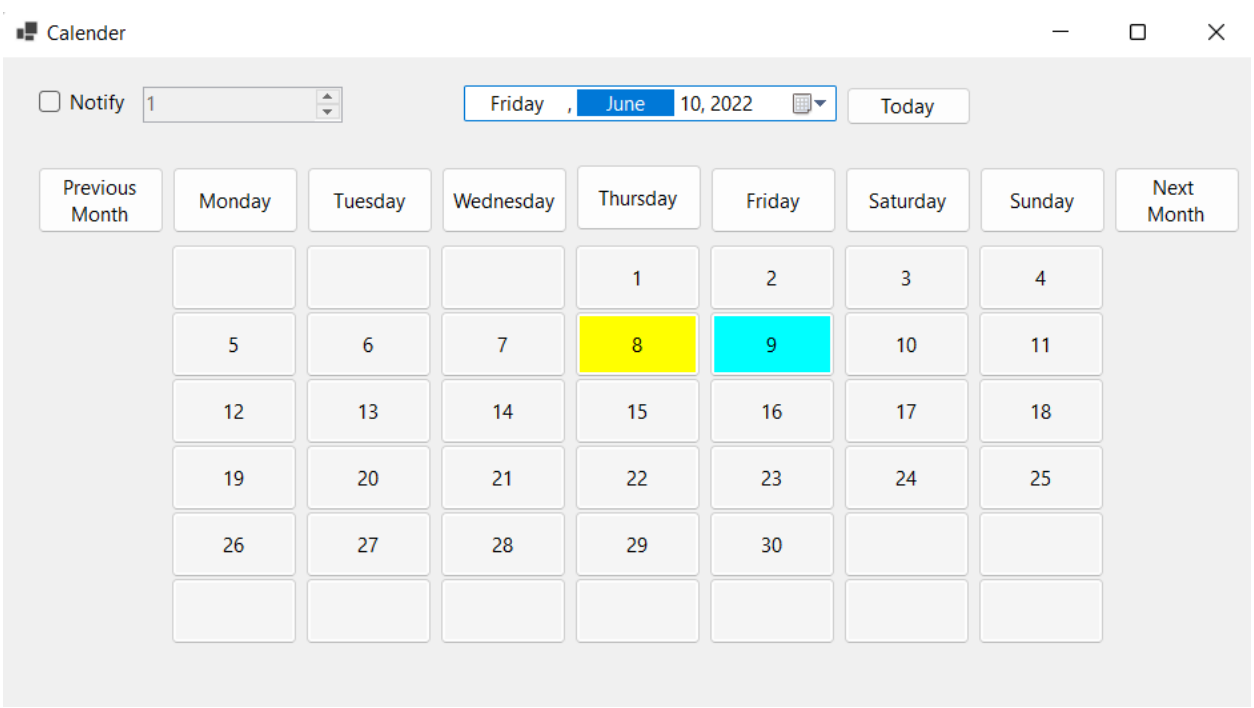
```
1 reference
void aJob_Deleted(object sender, EventArgs e)
{
    AJob uc = sender as AJob;
    PlanItem job = uc.Job;

    fPanel.Controls.Remove(uc);
    Job.Job.Remove(job);
}

1 reference
void aJob_Edited(object sender, EventArgs e)
{

}

2 references
List<PlanItem> GetJobByDay(DateTime date)
{
    return Job.Job.Where(p => p.Date.Year == date.Year && p.Date.Month == date.Month && p.Date.Day == date.Day).ToList();
}

1 reference
private void dtpkDate_ValueChanged(object sender, EventArgs e)
{
    ShowJobByDate((sender as DateTimePicker).Value);
}

1 reference
private void btnPreviousDay_Click(object sender, EventArgs e)
{
    dtpkDate.Value = dtpkDate.Value.AddDays(1);
}

1 reference
private void btnNextDay_Click(object sender, EventArgs e)
{
    dtpkDate.Value = dtpkDate.Value.AddDays(-1);
}

1 reference
private void mnsiAddJob_Click(object sender, EventArgs e)
{
    PlanItem item = new PlanItem() { Date = dtpkDate.Value };
    Job.Job.Add(item);
    AddJob(item);
}

1 reference
private void mnsiToday_Click(object sender, EventArgs e)
{
    dtpkDate.Value = DateTime.Now;
}
```

* Các phần còn lại mà em không đưa ra là các class miền được đặt để sử dụng lại nhiều lần cũng như được quy hoạch để xử lý thông tin code dễ dàng hơn

4. Hướng dẫn sử dụng:

- Khởi động: phần mềm Calender:





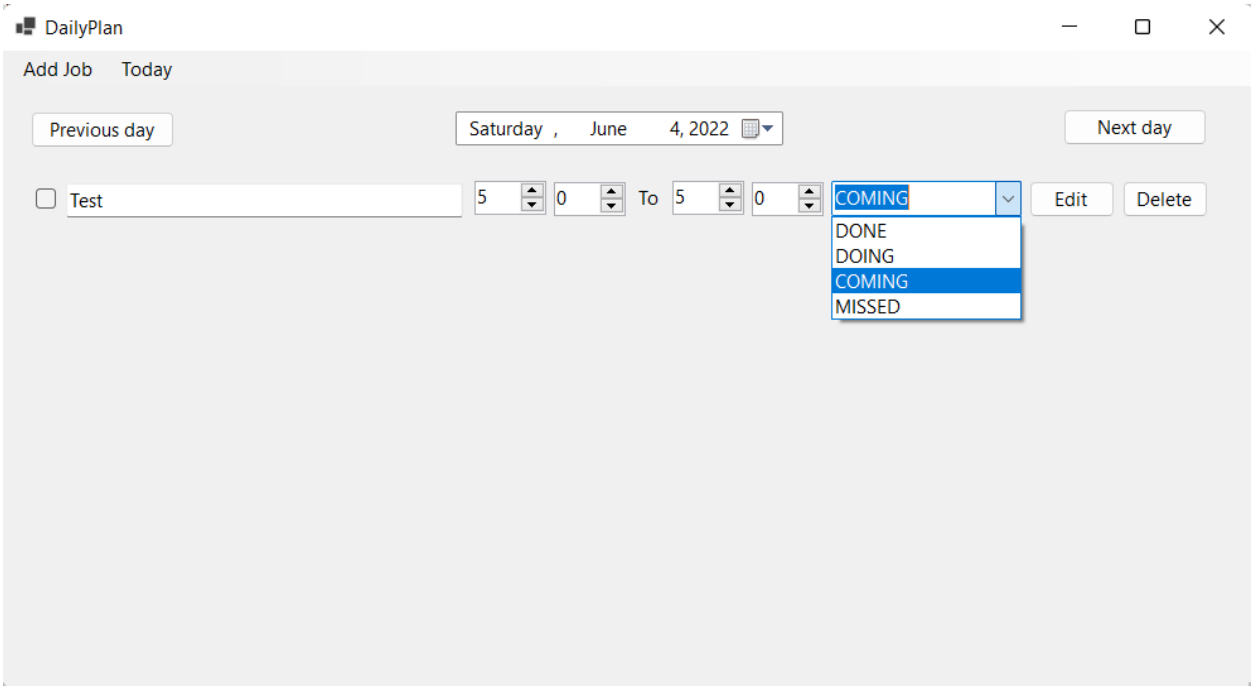
- Phần mềm hiển thị thứ ngày tháng ở chính giữa
- Ở đây ngày màu vàng là ngày hôm nay và ngày màu xanh là ngày được chọn.
- Ta có thể chọn tháng trước hoặc tháng sau hay chọn theo tháng năm.
- Có ô chọn quay về hôm nay
- Ô thông báo:

+ Ô thông báo chọn tíc để thông báo rồi chọn thời gian (ở đây tính theo phút)

+ Sau thời gian tương ứng ứng dụng sẽ hiển thị ở phần notify window để báo xem mình còn những việc gì trong hôm nay.

- Chọn 1 ngày để lập lịch:

+ Giao diện lập lịch:



+ Ở đây cũng có chỗ chọn ngày tháng năm và 2 nút ngày trước và ngày sau

+ Ở thanh các kế hoạch trong ngày thì có ô tick để đánh dấu đã hoàn thành, ô ghi chú, ô ghi thời gian công việc (giờ/phút), ô để kiểm tra công việc, ô sửa và ô xóa.

+ Có 2 thanh phía trên cùng bên trái để them việc và quay lại ngày hôm nay

+ Khi tick hay ghi chú hay sửa thời gian cũng như kế hoạch phải bấm lại vào ô edit để lưu

* Vì phần mềm không có tự động xóa nên mình nên tự xóa để tránh tiêu hao bộ nhớ cũng như hiệu suất phần mềm.

* Hiện tại phần mềm có thể nâng cấp được tiếp một số thao tác khó hơn cũng như dễ tiếp cận cho người dùng mà chưa có: Đồng hồ, lịch âm, đánh dấu những ngày đặc biệt, tự xử lý tác vụ,…

5.Tham khảo

http://www.kettic.com/winforms_ui/csharp_guide/calendar_get_started.shtml

Youtube KTeam

Youtube Jeov Dev

6.Các Bug đã gặp phải:

* Hầu hết các vấn đề em gặp phải là do sai cấu trúc code cũng như đánh máy sai, thiếu.