



Productive Day

Desktop Application

โดย

นายชัยพล หวาน

รหัสนิสิต 6330085821

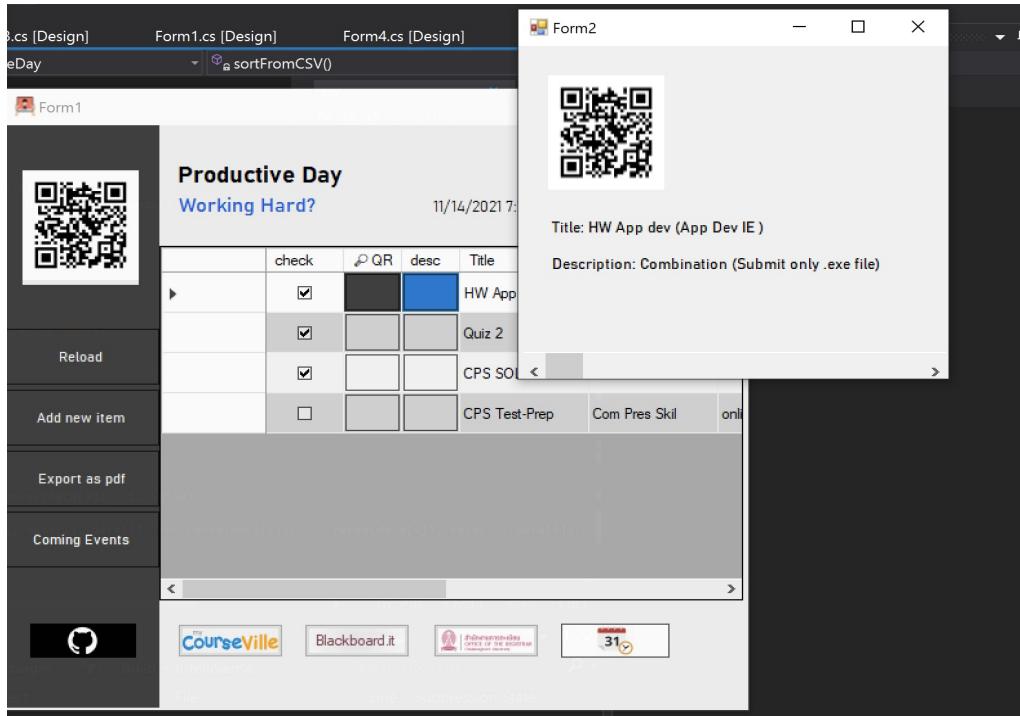
เสนอ

ผศ. ภูมิ เหลืองจามีกร

2104255 App Dev IE (2021/1)

## “Productive Day”

### Desktop Application



#### ที่มาของการพัฒนาแอปพลิเคชัน “Productive Day”

ปัญหานี้ของการ “เรียนออนไลน์” ในทุก ๆ วันนี้ คือภาระงานที่อาจเพิ่มมากขึ้น ซึ่งส่งผลให้การบริหารจัดการเวลา และการจัดระเบียบตารางงานเป็นไปด้วยความยุ่งยาก จึงนำมาสู่แนวคิดที่จะพัฒนา Desktop Application ที่มีชื่อว่า “Productive Day” ที่สามารถช่วยจัดจัดระเบียบตารางงาน ตามลำดับความสำคัญ โดยในแอปพลิเคชันนี้ ได้พัฒนาทอกแบบ UI/UX ที่ง่ายต่อการใช้งาน มาพร้อมกับฟีเจอร์มากมายที่จะอำนวยความสะดวกให้กับผู้ใช้งานได้แก่ การจดบันทึกตารางงาน ตามหัวข้อ รายวิชา รายละเอียดของงาน และวันกำหนดส่งงาน ที่สามารถกดบันทึก (checkbox) งานที่ทำเสร็จเรียบร้อยแล้ว, ปุ่มลิงก์ไปยังหน้าเว็บไซต์ที่ใช้งานบ่อย, ฟังก์ชัน Export as pdf, ฟังก์ชันสำหรับสร้าง QR Code เพื่อช่วยในการเข้าถึงลิงก์งาน ผ่านโทรศัพท์มือถือและแท็บเล็ต ฯลฯ รวมไปถึงฟังก์ชัน Desktop Notification ที่จะแจ้งเตือนงานที่ใกล้ครบกำหนดส่งในอีก 24 ชั่วโมง

## วิธีการใช้งานแอปพลิเคชัน

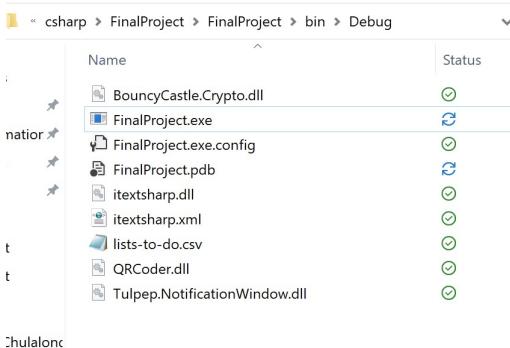
01. รันโปรแกรมโดยเปิดจากไฟล์ .exe โดยให้เตรียมไฟล์ DLL (Dynamic link library) file (BouncyCastle.Crypto.dll, iTextsharp.dll, QRCode.dll, Tulpep.NotificationWindow.dll)  
โดยสามารถ Download ได้จาก <https://drive.google.com/drive/folders/1jUX9M-rNVYGxe-bTkXJB7M1t7lvSAGI?usp=sharing>
02. เมื่อโปรแกรมทำงาน จะมี notification งานที่เกล็คครบกำหนดส่ง (24ชม.) ปรากฏขึ้น  
ในหน้าต่างหลักจะมี Menu Bar ด้านซ้ายมือ และปุ่มลิงก์ไปยังเว็บไซต์ที่ใช้งานบ่อยด้านล่าง
  - ผู้ใช้งานสามารถกดปุ่ม icon เพื่อเปิดหน้าเว็บไซต์ต่างๆ ได้โดยตรง หน้าต่างหลักจะแสดง ข้อความ Greeting, วันที่ เวลา และรายการของตารางงานทั้งหมด
  - โดยผู้ใช้งานสามารถใช้แบบ scroll bar ด้านล่างเพื่อทำการเลื่อนดูข้อมูล
03. ในช่อง check ผู้ใช้งานสามารถกดทำเครื่องหมาย  ในรายการที่ได้ทำเสร็จสิ้นแล้วได้
04. ในช่อง  ผู้ใช้งานสามารถกดปุ่ม เพื่อ generate QR Code ของ url ของรายการนั้นๆ  
โดย QR code ของรายการนั้นๆ จะปรากฏอยู่ในหน้าต่างหลัก  
ถ้าหากไม่พบ QR Code จะมี messageBox แสดงว่า ไม่พบ QR Code
05. ในช่อง desc ผู้ใช้งานสามารถกดปุ่มเพื่ออ่านรายละเอียดของรายการนั้นๆ  
โดยจะแสดงขึ้นมาในหน้าต่างใหม่
06. ปุ่ม Add new item จะทำการเปิดหน้าต่างใหม่ ลักษณะฟอร์ม ให้ผู้ใช้งานกรอกรายการ และกด Submit  
เพื่อยืนยัน เมื่อผู้ใช้งานกดยืนยันเรียบร้อยแล้ว ข้อมูลจะถูกบันทึกไปยัง .csv file  
และหน้าต่างฟอร์มจะหายไปอัตโนมัติ
07. ปุ่ม Calendar เมื่อผู้ใช้งานกด จะทำการเปิดหน้าต่างใหม่ที่แสดงปฏิทิน
08. ผู้ใช้งานสามารถบันทึกตารางงานปัจจุบันในรูปแบบ pdf file ได้ โดยกดปุ่ม Export as pdf  
จะเปิดหน้าต่างใหม่ให้ผู้ใช้งานได้เลือกไฟล์เดอร์ที่ต้องการจะบันทึกไฟล์
09. ผู้ใช้งานสามารถเรียกใช้งาน Desktop Popup Notification ของรายการงานที่เกล็คครบกำหนดส่งได้  
โดยการกดปุ่ม Coming Event
10. ในการนี้ที่ข้อมูลไม่อัพเดท ผู้ใช้งานสามารถกดปุ่ม Reload เพื่อทำการ Refresh โปรแกรมได้

สามารถดูรายละเอียดการทำงานของโปรแกรมได้ในหน้าตัดไป

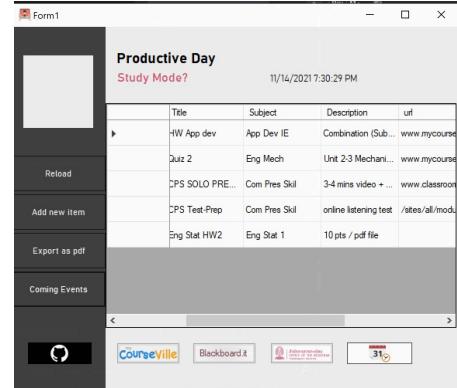
และจาก [jianchayapol/2104255-APP-DEV-IE \(github.com\)](https://github.com/jianchayapol/2104255-APP-DEV-IE)

## ไฟล์เอกสารการทำงาน - วิธีการใช้งาน

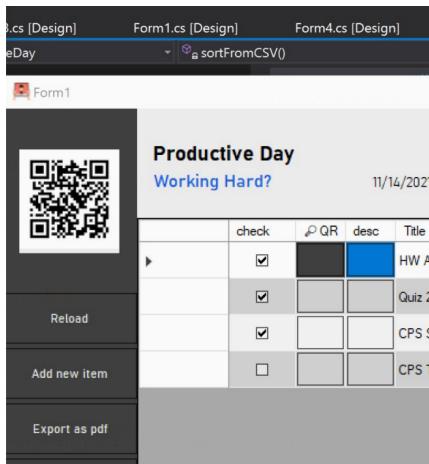
### ● 01



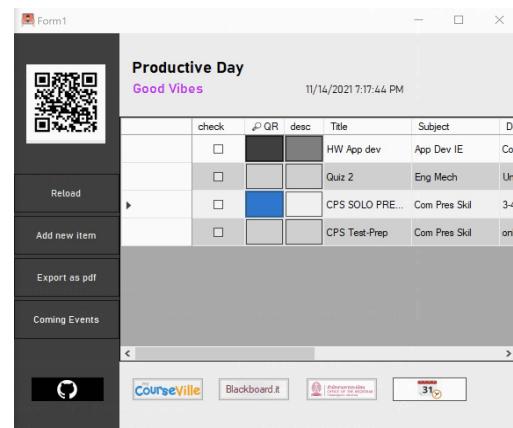
### ● 02 หน้าต่างหลัก



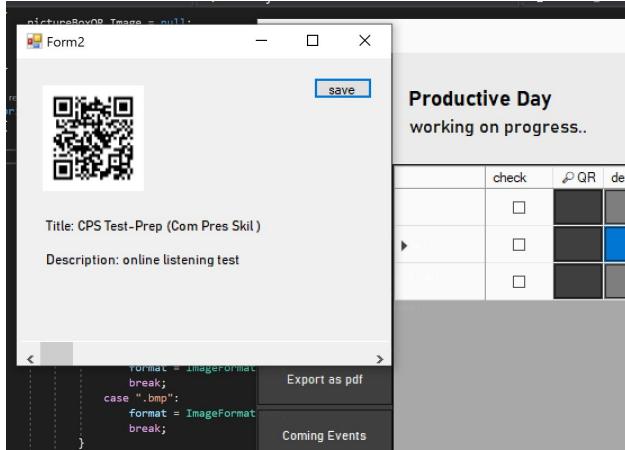
### ● 03 check



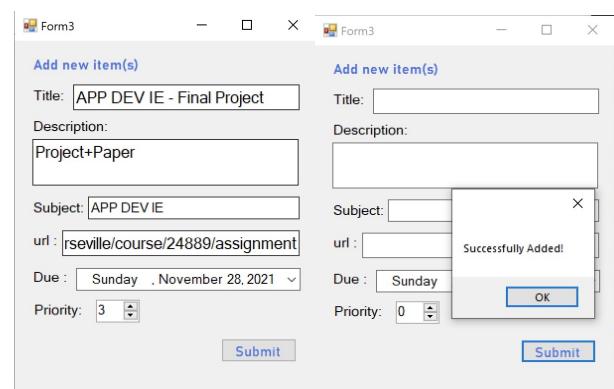
### ● 04 QR



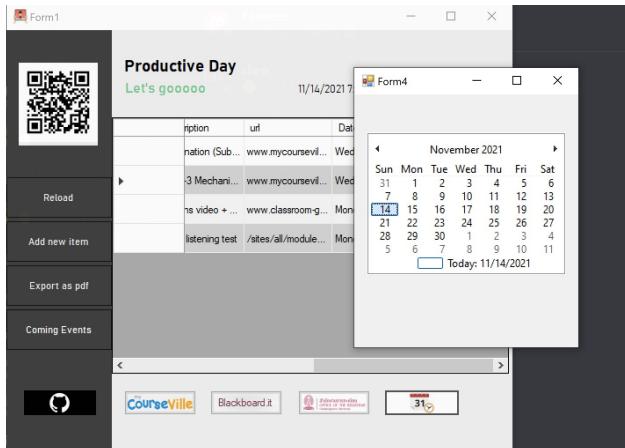
### ● 05 desc



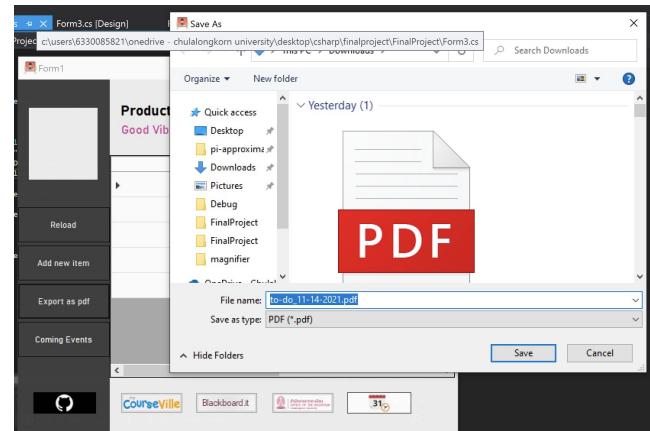
### ● 06 Add new item



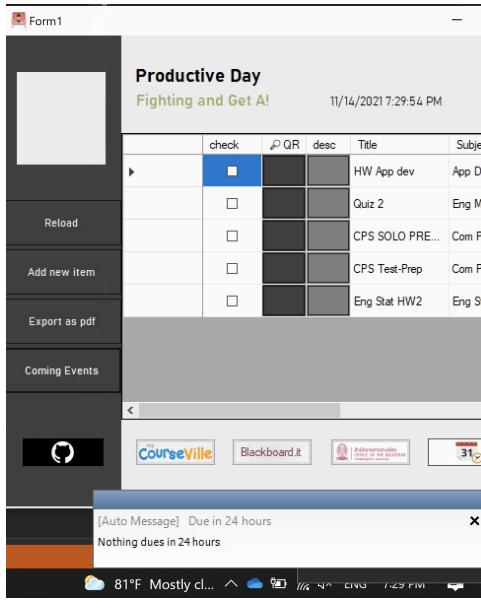
● 07 Calendar



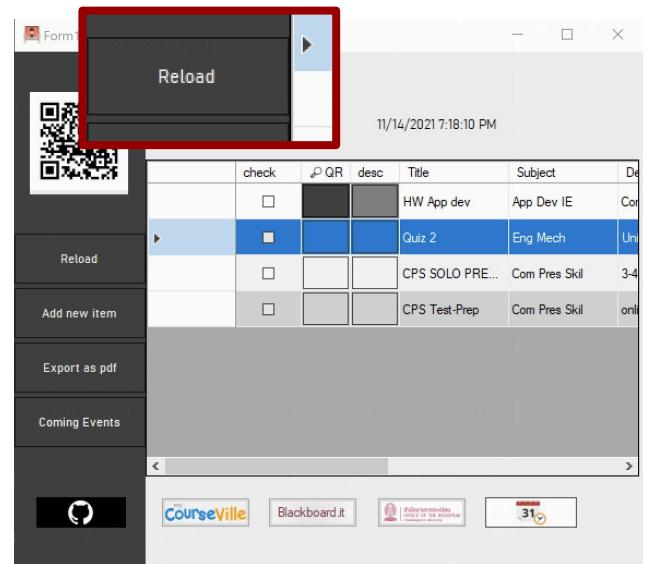
● 08 export as .pdf



● 09 Popup Notification



● 10 Reload



## แนวคิดและการเขียนโปรแกรม

ในการพัฒนาแอปพลิเคชัน ProductiveDay นี้ ได้ใช้ภาษา C# ซึ่งถือว่าเป็นภาษายอดนิยมในการพัฒนา Desktop Application อาศัยหลักการ Object-oriented Programming หรือ OOP อย่างเต็มรูปแบบไม่ว่าจะเป็นการเขียน class และ method, การกำหนด Access Modifier, การเขียน Method Custom Comparator ของ class ตลอดจนการใช้เลือกงาน Data Structure อย่าง SortedSet<T> และ List<T> โดยออกแบบและพัฒนาด้วย UX/UI ที่สวยงาม ทันสมัย และง่ายต่อการใช้งาน รวมถึงการนำความรู้ทั้งจากในและนอกห้องเรียนมาประยุกต์ในการพัฒนาโปรแกรม ไม่ว่าจะเป็น การเขียนและอ่านไฟล์ csv, การใช้ Random, DateTime, UnixTimeStamp, Timer, saveFileDialog, Bitmap, dataGridView และการเลือกใช้งาน library ที่น่าสนใจมากmany เช่น QRCode และ Tulpep.NotificationWindow โดยมีรายละเอียดการทำงานของโปรแกรมดังนี้

### Form1 (หน้าต่างหลัก)

```
class ThingsToDo : IComparable<ThingsToDo>
```

#### Field

```
string title;
string description;
int priorityLevel;
int dueTime;
string subject;
string status;
string url;
```

#### Constructor

```
public ThingsToDo(string subject, string title, string description, int priorityLevel, int dueTime, string status, string url)
{
    this.subject = subject;
    this.title = title;
    this.description = description;
    this.priorityLevel = priorityLevel;
    this.dueTime = dueTime;
    this.status = "Waiting..";
    this.url = url;
}
```

Method	Description
<pre>int CompareTo(ThingsToDo other)</pre> <pre>public int CompareTo(ThingsToDo other) {     if (this.getDueTime() != other.getDueTime())     {         if (this.getDueTime() &gt; other.getDueTime()) return 1;         else return -1;     }     if (this.getPriorityLevel() != other.getPriorityLevel())         if (this.getPriorityLevel() &lt; other.getPriorityLevel()) return 1;         else return -1;     return 1; }</pre>	<ul style="list-style-type: none"> <li>- Custom comparator โดย override (Default method) CompareTo กำหนดให้เรียกข้อมูลจากวันกำหนดส่ง และลำดับความสำคัญ (priorityLevel) ของงานตามลำดับ</li> </ul>

getter/setter สำหรับทุกๆ field

public partial class ProductiveDay : Form

Field
<pre>SortedSet&lt;ThingsToDo&gt; s = new SortedSet&lt;ThingsToDo&gt;();  long t = 0; List&lt;string&gt; greetWordList = new List&lt;string&gt;();  greetWordList.Add("Working Hard?"); greetWordList.Add("Cheer Up!"); greetWordList.Add("Let's gooooo"); greetWordList.Add("working on progress..."); greetWordList.Add("Fighting and Get A!"); greetWordList.Add("Good Vibes"); greetWordList.Add("Study Mode?");</pre>

Constructor	
<pre>// MAIN public ProductiveDay() {     InitializeComponent();     timer1.Start();     if (!File.Exists("lists-to-do.csv")) using (StreamWriter w = File.AppendText("lists-to-do.csv"))         notifyWithin24Due();     showOnGrid(); }</pre>	<p>InitializeComponent</p> <p>สั่ง timer ให้เริ่มทำการนับเวลา</p> <p>เรียกใช้ method notifyWithin24Due()</p> <p>เพื่อขึ้น notification popup ที่มุ่งลากยาวของ Desktop ของผู้ใช้งาน</p> <p>และแสดงตารางงานที่ผู้ใช้งานได้จดบันทึกไว้ใน การใช้งานครั้งก่อน</p>

Method	Description
<pre>private void timer1_Tick(object sender, EventArgs e)</pre>	<p>ใช้งาน timer1_Tick โดยทำการนับเวลา จะทำการแสดงผล และ Refresh() เล่าให้มีทุกวินาที และทุก ๆ 10 วินาที จะทำการ random ข้อความ greeting จาก List&lt;string&gt; greetWordList</p>
<pre>private void timer1_Tick(object sender, EventArgs e) {     t += 1;     if (t % 10 == 1)     {         time_now.Text = DateTime.Now.ToString();         Refresh();     }      if(t% 100 == 0)     {         var random = new Random();         int index = random.Next(greetWordList.Count);         greetLabel.Text = greetWordList[index];         greetLabel.ForeColor = randomColor();     } }</pre>	
<pre>private void notifyWithin24Due()</pre>	<p>แจ้งเตือนงานที่ใกล้ครบกำหนดส่ง ในรูปแบบของ Desktop Popup Notification โดยใช้งานบริการของ Tulpep.NotificationWindow</p>
<pre>private void notifyWithin24Due() {     string alert = "";     s = new SortedSet&lt;ThingsToDo&gt;();     int h = 5;     sortFromCSV();     foreach (ThingsToDo thing in s)     {         DateTime anotherDay = UnixTimeStampToDate((thing.getDueTime()));         DateTime today = DateTime.Now;         if ((anotherDay - today).TotalDays &lt;= 1)         {             alert += " " + thing.getTitle() + "[" + thing.getSubject() + "] " + "\r\n";             h++;         }     }      if (alert == "") alert = "Nothing dues in 24 hours\r\n";      Tulpep.NotificationWindow.PopupNotifier popup = new Tulpep.NotificationWindow.PopupNotifier();      popup.ShowGrip = false;     popup.HeaderHeight = 20;     popup.TitlePadding = new Padding(3);     popup.ContentPadding = new Padding(3);     popup.AnimationDuration = 1000;     popup.AnimationInterval = 1;     popup.HeaderColor = Color.FromArgb(125, 164, 200);     popup.ShowCloseButton = true;     popup.TitleText = "[Auto Message] Due in 24 hours ";     popup.ContentText = alert;     popup.Scroll = true;     popup.Size = new Size(400, h*2*8);     popup.Popup(); }</pre>	

```
private static Bitmap generateQRcode(string url)
```

static method ที่ทำการรับ parameter เป็น string ของ url link และ generate QR Code คืนกลับมาในรูปแบบของ bitmap

```
private static Bitmap generateQRcode(string url)
{
    QRCodeGenerator qrGenerator = new QRCodeGenerator();
    QRCodeData qrCodeData = qrGenerator.CreateQrCode(url, QRCodeGenerator.ECCLevel.Q);
    QRCode qrCode = new QRCode(qrCodeData);
    Bitmap qrCodeImage = qrCode.GetGraphic(180);

    Bitmap qr180 = new Bitmap(qrCodeImage, 90, 90);
    return qr180;
}
```

Method button click ต่างๆ สำหรับลิงก์ไปยังหน้าเว็บไซต์ต่างๆ

```
private void githubLink_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("http://www.github.com/jianchayapol");
}

private void mycoursevilleLink_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("http://www.mycourseville.com");
}

private void blackboardLink_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("https://blackboard.it.chula.ac.th/");
}

private void regChulaLink_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("http://www.reg.chula.ac.th");
}
```

```
private void exportAsPdf_Click(object sender,
EventArgs e)
```

สำหรับการ export ตารางงานที่แสดงอยู่เป็น file pdf โดยใน method นี้จะทำการ setting (saveFileDialog) default filetype เป็น pdf และ default filename เป็นวันเดือนปีปัจจุบัน มีการใช้ try/catch ในการจัดการ error ต่างๆ

```

private void exportAsPdf_Click(object sender, EventArgs e)
{
    if (dataGridView1.Rows.Count > 0)
    {
        SaveFileDialog sfd = new SaveFileDialog();
        sfd.Filter = "PDF (*.pdf)|*.pdf";
        string d = DateTime.Now.ToString("dd/MM/yyyy");
        sfd.FileName = "to-do_" + d.Replace("/", "-") + ".pdf";
        bool fileError = false;
        if (sfd.ShowDialog() == DialogResult.OK)
        {
            if (File.Exists(sfd.FileName))
            {
                try
                {
                    File.Delete(sfd.FileName);
                }
                catch (IOException ex)
                {
                    fileError = true;
                    MessageBox.Show("It wasn't possible to write the data to the disk." + ex.Message);
                }
            }
        }
    }
}

```

```

        }

        if (!fileError)
        {
            try
            {
                PdfPTable pdfTable = new PdfPTable(dataGridView1.Columns.Count);
                pdfTable.DefaultCell.Padding = 3;
                pdfTable.WidthPercentage = 100;
                pdfTable.HorizontalAlignment = Element.ALIGN_LEFT;

                foreach (DataGridViewColumn column in dataGridView1.Columns)
                {
                    PdfPCell cell = new PdfPCell(new Phrase(column.HeaderText));
                    pdfTable.AddCell(cell);
                }

                foreach (DataGridViewRow row in dataGridView1.Rows)
                {
                    foreach (DataGridViewCell cell in row.Cells)
                    {
                        if (cell.Value != null && cell.Value is string) pdfTable.AddCell(cell.Value.ToString());
                        else pdfTable.AddCell(" ");
                    }
                }

                using (FileStream stream = new FileStream(sfd.FileName, FileMode.Create))
                {
                    Document pdfDoc = new Document(PageSize.A4, 10f, 20f, 20f, 10f);
                    PdfWriter.GetInstance(pdfDoc, stream);
                    pdfDoc.Open();
                    pdfDoc.Add(pdfTable);
                    pdfDoc.Close();
                    stream.Close();
                }

                MessageBox.Show("Data Exported Successfully !", "Info");
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error :" + ex.Message);
            }
        }
    }
    else
    {
        MessageBox.Show("No Record To Export !", "Info");
    }
}

```

```
public static long ToUnixTimestamp(DateTime target)
```

รับ parameter เป็น DateTime

แปลงและคืนค่าในรูปแบบจำนวนเต็ม long ของ UnixTimeStamp

```
public static long ToUnixTimestamp(DateTime target)
{
    var date = new DateTime(1970, 1, 1, 0, 0, 0, target.Kind);
    var unixTimestamp = System.Convert.ToInt64((target - date).TotalSeconds);

    return unixTimestamp;
}
```

Private Color randomColor() ทำการสุ่มและคืนค่าสี ในรูปแบบ Color

```
private Color randomColor()
{
    Random rnd = new Random();
    Color c = Color.FromArgb(rnd.Next(0, 255), rnd.Next(0, 255), rnd.Next(0, 255));
    return c;
}
```

private void sortFromCSV()

เปิดและอ่านไฟล์ csv ที่บันทึกตารางงาน

แล้วสร้าง new ThingsToDo object

เก็บตารางงานที่ยังไม่ถึงกำหนดส่ง ไว้ใน Data Structure แบบ SortedSet<ThingsToDo>

เพื่อเรียง priority ของข้อมูลจากสำคัญมาก-สำคัญน้อย  
ตามรูปแบบที่ได้กำหนดไว้ใน method CompereTo

```
private void sortFromCSV()
{
    string[] lines = System.IO.File.ReadAllLines("lists-to-do.csv");
    DataTable dt = new DataTable();
    if (lines.Length > 1)
    {
        for (int i = 0; i < lines.Length; i++)
        {
            string[] data = lines[i].Split(',');
            if (data.Length != 7) continue;
            DataRow dr = dt.NewRow();
            if (ToUnixTimestamp(DateTime.Now) - long.Parse(data[4]) < 12 * 60 * 60) // 12 hours to be clear
            {
                ThingsToDo tmp = new ThingsToDo(data[0], data[1], data[2], int.Parse(data[3]), int.Parse(data[4]), data[5], data[6]);
                s.Add(tmp);
            }
        }
    }
    return ;
}
```

```

lists-to-do.csv - Notepad
File Edit Format View Help
-----
Eng Mech ,Quiz 2,Unit 2-3 Mechanics ,10,1637089932,waiting..,www.mycourseville.com
App Dev IE,HW App dev,Combination (Submit only .exe file),20,1637089932,waiting..,www.mycourseville.com
Com Pres Skill,CPS SOLO PRESENTATION,3-4 mins video + Engineering topic,11,1638126732,waiting..,www.classroom-google
Com Pres Skill,CPS Test-Prep,online listening test,4,1638126732,waiting..,/sites/all/modules/courseville/files/ckfind
Eng Stat 1,Eng Stat HW2,10 pts / pdf file,2,1638213132,waiting..

```

`private void showOnGrid()`

ทำงานในลักษณะวนลูปเพื่อแสดงผลข้อมูลที่ถูกเรียงตามลำดับ priority เรียบร้อยแล้ว ออกทางตาราง gridView dataGridView1

```

private void showOnGrid()
{
    s = new SortedSet<ThingsToDo>();

    sortFromCSV();

    DataTable dt = new DataTable();

    dt.Columns.Add("Title");
    dt.Columns.Add("Subject");
    dt.Columns.Add("Description");
    dt.Columns.Add("url");
    dt.Columns.Add("Dates");
    dt.Columns.Add("Status");

    foreach (ThingsToDo item in s)
    {
        DataRow dr = dt.NewRow();
        dr["Subject"] = item.getSubject();
        dr["title"] = item.getTitle();
        dr["Description"] = item.getDescription();
        dr["Dates"] = UnixTimeStampToDate(item.getDueTime()).ToString("yyyy-MM-dd HH:mm:ss");

        dr["Status"] = item.getStatus();
        dr["url"] = item.getUrl();

        dt.Rows.Add(dr);
    }

    if (dt.Rows.Count > 0)
    {
        dataGridView1.DataSource = dt;
    }
    for(int i=0 ; i< dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].DefaultCellStyle.BackColor = Color.WhiteSmoke;
        if (i % 2 != 0)
        {
            dataGridView1.Rows[i].DefaultCellStyle.BackColor = Color.LightGray;
        }

        else if (dataGridView1.Rows[i].Cells[8].Value.ToString().Equals("Complete.."))
        {
            dataGridView1.Rows[i].DefaultCellStyle.BackColor = Color.Green;
        }
    }
}

```

## Method เพิ่มเติม

```
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    var senderGrid = (DataGridView)sender;
    if (e.ColumnIndex == 1)
    {
        if (senderGrid.Columns[e.ColumnIndex] is DataGridViewButtonColumn &&
        e.RowIndex >= 0)
        {
            string urlLink = dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString();
            if (urlLink == "") MessageBox.Show("URL NOT FOUND");
            else
            {
                Bitmap qr = generateQRcode(urlLink);
                pictureBox4.Image = qr;
            }
        }
    }
    else if (e.ColumnIndex == 2)
    {
        if (senderGrid.Columns[e.ColumnIndex] is DataGridViewButtonColumn &&
        e.RowIndex >= 0)
        {
            string urlLink = dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString();
            if (urlLink != "")
            {
                Bitmap qr = generateQRcode(urlLink);
                form2.showQRform2(qr);
            }

            string title = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString() + " (" + dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString() +
            string des = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();
            form2.showDesc(title, des);
            form2.Show();
        }
    }
}
```

```
private void calendar_show_Click(object sender, EventArgs e)
{
    form4.Show();
}
```

```
private void reload_Click(object sender, EventArgs e)
{
    showOnGrid();
    Refresh();
}
```

```
private void comingEvents_Click(object sender, EventArgs e)
{
    notifyWithin24Due();
}
```

```
private void addItem_Click(object sender, EventArgs e)
{
    form3.Show();
    showOnGrid();

}
```

### Form2 (หน้าต่างแสดงรายละเอียดงาน)

public partial class Form2 : Form

Constructor	
<pre>public Form2() {     InitializeComponent(); }</pre>	InitializeComponent

Method	
<pre>public void showDesc(string title, string desc) {     label1.Text = (" Title: " + title);     label2.Text = (" Description: " + desc); }</pre>	รับ parameter เป็น string ของ title และ desc แล่ทำการแสดงข้อความรายละเอียดของงานผ่าน label1 และ label2
<pre>public void showQRform2(Bitmap picture) {     Bitmap x = new Bitmap(picture, pictureBoxQR.Size);     pictureBoxQR.Image = x; }</pre>	รับ parameter เป็น Bitmap qr_code ที่ได้ generate ขึ้นจาก Form1 แล่แสดงภาพที่รับมาใน pictureBoxQR
<pre>private void Form2_FormClosing(object sender, FormClosingEventArgs e) {     pictureBoxQR.Image = null;     e.Cancel = true;     this.Hide(); }</pre>	FormClosing Method set ค่าให้ช่องฟอร์มไว้ เมื่อกดปุ่มปิด

```

private void button1_Click(object sender, EventArgs e)
{
    SaveFileDialog sfd = new SaveFileDialog();
    sfd.Filter = "Images|*.png;*.bmp;*.jpg";
    string d = DateTime.Now.ToString("dd-MM-yyyy");
    sfd.FileName = "desc_" + d.Replace("/", "-") + ".png";

    using (Bitmap bmp = new Bitmap(this.Width, this.Height))
    {
        this.DrawToBitmap(bmp, new Rectangle(Point.Empty, bmp.Size));
        ImageFormat format = ImageFormat.Png;
        if (sfd.ShowDialog() == System.Windows.Forms.DialogResult.OK)
        {
            string ext = System.IO.Path.GetExtension(sfd.FileName);
            switch (ext)
            {
                case ".jpg":
                    format = ImageFormat.Jpeg;
                    break;
                case ".bmp":
                    format = ImageFormat.Bmp;
                    break;
            }

            bmp.Save(sfd.FileName, format);
        }
    }
}

```

เมื่อผู้ใช้ทำการกดปุ่ม button1 (save)  
จะเรียกใช้งาน saveFileDialog เพื่อทำการ  
save หน้าต่าง form  
ที่แสดงอยู่ในรูปแบบไฟล์รูปภาพ

### Form3 (ฟอร์มกรอก new Item)

public partial class Form3 : Form

#### Constructor

```

public Form3()
{
    InitializeComponent();
}

```

InitializeComponent

#### Method

private void submit\_Click(object sender, EventArgs e)

รับ input จาก Form3

```

private void submit_Click(object sender, EventArgs e)
{
    string title_in = title.Text;
    string desc_in = desc.Text;
    String due_date = dateTimePicker1.Value.ToString("yyyy-MM-dd");
    long due_unix = ToUnixTimestamp(dateTimePicker1.Value);
    string subject_in = subject.Text;
    string url_in = url.Text;
    int priority_in = (int) numericUpDown1.Value;

    string s;
    List<string> csvLines = new List<string>();

    StreamReader sr = new StreamReader("lists-to-do.csv");

    while (((s = sr.ReadLine()) != null)) //null = all files are read.
    {
        csvLines.Add(s);
    }
    sr.Close();

    StreamWriter sw = new StreamWriter("lists-to-do.csv");
    foreach (string line in csvLines)
    {
        sw.WriteLine(line + "\r\n");
    }

    title_in = title_in.Replace(", ", " ");
    desc_in = desc_in.Replace(", ", " ");
    url_in = url_in.Replace(", ", " ");
    if (url_in == "-" || url_in == " ") url_in = "";
    subject_in = subject_in.Replace(", ", " ");

    sw.WriteLine("{0},{1},{2},{3},{4},{5},subject_in, title_in, desc_in, priority_in,due_unix,"waiting..",url_i
    sw.Close();

    title.Text= "";
    desc.Text = "";
    subject.Text="";
    url.Text = "";
    numericUpDown1.Value = 0;
    MessageBox.Show("Successfully Added!");
    this.Close();
    return;
}

```

โดยเขียนโปรแกรมป้องกัน case sensitive ต่างๆของการใช้งานไฟล์ CSV ที่อาจจะเกิดขึ้น เช่น กรณี input ที่ผู้ใช้กรอกข้อมูลที่มี “,” ปรากฏอยู่ ทำการเปิดไฟล์ และอ่านข้อมูลที่บันทึกไว้ในไฟล์ CSV แล้วเก็บไว้ใน List<string> csvLines ชั่วคราว ทำการเปิดไฟล์เพื่อเขียน โดยเขียนทุกๆบรรทัดเก่าที่ได้เก็บไว้ ใน csvLines จากนั้น เพิ่มข้อมูลบรรทัดใหม่ อีกหนึ่งบรรทัด เป็นข้อมูลที่เพิ่งรับมาจาก TextBox ต่างๆที่ผู้ใช้กรอก

```

public static long ToUnixTimestamp(DateTime target)
{
    var date = new DateTime(1970, 1, 1, 0, 0, 0, target.Kind);
    var unixTimestamp = System.Convert.ToInt64((target - date).TotalSeconds);

    return unixTimestamp;
}

```

รับ parameter เป็น DateTime แปลงและคืนค่าในรูปแบบจำนวนเต็ม long ของ UnixTimeStamp

```

private void Form3_FormClosing(object sender, FormClosingEventArgs e)
{
    e.Cancel = true;
    this.Hide();
}

```

FormClosing Method  
set ค่าให้ซ่อนฟอร์มไว้ เมื่อกดปุ่มปิด

### Form4 (ຈົບທຶນ)

public partial class Form4 : Form

#### Constructor

```
public Form4()
{
    InitializeComponent();
}
```

InitializeComponent

#### Method

```
private void Form4_FormClosing(object sender, FormClosingEventArgs e)
{
    e.Cancel = true;
    this.Hide();
}
```

FormClosing Method

set ค่าให้ช่องฟอร์มໄວ້ເນື່ອກດັບປຸ່ມປິດ