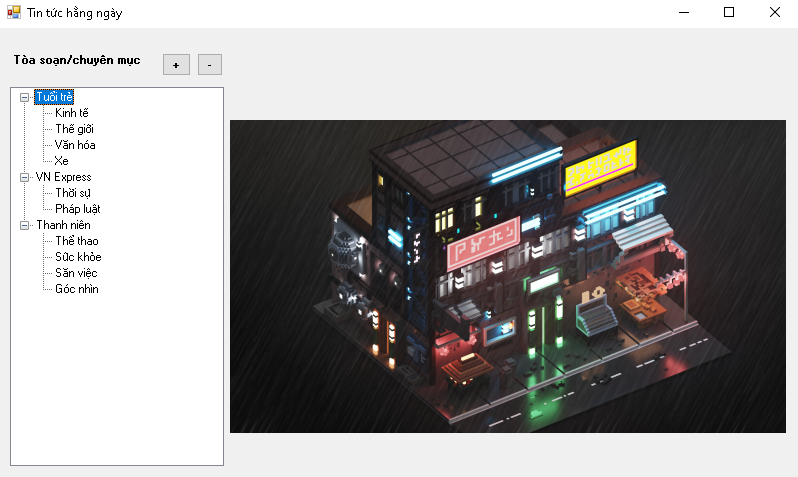
Đinh Trọng Đạt – 1914775



public partial class Form1 : Form

{

private readonly NewFeedManager \_newManager;

public Form1(NewFeedManager newManager)

{

InitializeComponent();

this.\_newManager = newManager;

}

//Sự kiện nhấn chọn node trong Tree View

private void treeView1\_AfterSelect(object sender, TreeViewEventArgs e)

{

pnNews.Controls.Clear();

if(e.Node.Level == 1)

{

var articles = \_newManager.GetNews(e.Node.Parent.Text, e.Node.Text);

foreach (var article in articles)

{

var item = new Component.NewsControl();

item.Size = new Size(519, 101);

item.Dock = DockStyle.Top;

item.SetArticle(article);

pnNews.Controls.Add(item);

}

}

}

//Hiển thị Dữ liệu lên Form

private void Form1\_Load(object sender, EventArgs e)

{

ShowFeedOnTreeView(\_newManager.GetNewFeed());

}

//Sự kiện nút thêm

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

{

var dialog = new NewFeesForm(\_newManager);

if (dialog.ShowDialog() == DialogResult.OK)

{

//\_newManager.SaveChange();

ShowFeedOnTreeView(\_newManager.GetNewFeed());

}

}

//Sự kiện xóa

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

{

if (tvPubLish.SelectedNode == null) return;

if(tvPubLish.SelectedNode.Level == 0)

{

\_newManager.RemovePublish(tvPubLish.SelectedNode.Text);

}

else

{

var publishNode = tvPubLish.SelectedNode.Parent;

\_newManager.RemoveCate(publishNode.Text, tvPubLish.SelectedNode.Text);

}

tvPubLish.SelectedNode.Remove();

}

//Hiển thị thông tin trên treeview

private void ShowFeedOnTreeView(List<Models.Publish> publisher)

{

tvPubLish.Nodes.Clear();

pnNews.Controls.Clear();

foreach (var pub in publisher)

{

var publisherNode = tvPubLish.Nodes.Add(pub.Name);

foreach (var cate in pub.Categories)

{

publisherNode.Nodes.Add(cate.Name);

}

}

tvPubLish.ExpandAll();

}

}

//Class NewParser đọc file xml đưa lên userControl.

public class NewParser

{

public List<Models.Article> ParseXml(string xmlContent)

{

var document = new XmlDocument();

document.LoadXml(xmlContent);

var articles = new List<Models.Article>();

var itemNodes = document.SelectNodes("//item");

foreach (XmlNode node in itemNodes)

{

var news = new Models.Article()

{

Title = node.SelectSingleNode("title").InnerText,

Description = StripHtml(node.SelectSingleNode("description").InnerText),

Link = node.SelectSingleNode("link").InnerText,

PubDate = ParseDate(node.SelectSingleNode("pubDate").InnerText)

};

articles.Add(news);

}

return articles;

}

private string StripHtml(string content)

{

return Regex.Replace(content, "<.\*?>", String.Empty).Trim();

}

private DateTime ParseDate(string dateStr)

{

try

{

return DateTime.Parse(dateStr);

}

catch (Exception)

{

return DateTime.Now;

}

}

//UserControl hiển thị trên panel các thông tin

public NewsControl()

{

InitializeComponent();

}

public void SetArticle(Models.Article news)

{

lbTitle.Text = news.Title;

lbDes.Text = news.Description;

lbDate.Text = news.PubDate.ToString("dd/MM/yyyy HH:mm");

llb.LinkClicked += (sender, args) =>

{

Process.Start(news.Link);

};

}

protected override void OnPaint(PaintEventArgs e)

{

base.OnPaint(e);

e.Graphics.DrawRectangle(Pens.Black, 0, 1, Width - 1, Height - 2);

}

//class newFeedManager

public class NewFeedManager

{

private readonly IO.INewRepo \_newRepo;

private List<Models.Publish> \_publisher;

private readonly RssFeed.RSSReader \_rssReader;

public NewFeedManager(INewRepo newRepo, RssFeed.RSSReader rssReader)

{

\_newRepo = newRepo;

\_rssReader = rssReader;

}

public List<Models.Publish> GetNewFeed()

{

if(\_publisher == null)

{

\_publisher = \_newRepo.GetNews();

}

return \_publisher;

}

public void SaveChange()

{

\_newRepo.Save(\_publisher);

}

public void RemovePublish(string pubName)

{

\_publisher.RemoveAll(x => x.Name == pubName);

SaveChange();

}

public void RemoveCate(string pubName, string cateName)

{

var pub = \_publisher.Find(x => x.Name == pubName);

if (pub == null)

return;

pub.RemoveCate(cateName);

SaveChange();

}

public bool AddCategory(string publishName, string categoryName, string link, bool isExists)

{

var pub = \_publisher.Find(x => x.Name == publishName);

if (pub == null)

{

pub = new Models.Publish()

{

Name = publishName

};

\_publisher.Add(pub);

}

return pub.AddCategory(categoryName, link, isExists);

}

public List<Models.Article> GetNews(string publishName, string categoryName)

{

var publisher = \_publisher.Find(x => x.Name == publishName);

if (publisher == null) return new List<Models.Article>();

var category = publisher.Categories.Find(x => x.Name == categoryName);

if (category == null) return new List<Models.Article>();

if(category.Articles.Count == 0)

{

category.Articles = \_rssReader.GetNews(category.Link);

}

return category.Articles;

}

public class NewRepo : INewRepo

{

private const string filePath = "Data\\data.txt";

public List<Publish> GetNews()

{

var publisher = new List<Publish>();

Publish office = null;

string line;

try

{

using (var stream = new FileStream(filePath, FileMode.Open, FileAccess.Read))

{

using (var reader = new StreamReader(stream))

{

while (!reader.EndOfStream)

{

line = reader.ReadLine();

if (line == null)

{

break;

}

if (line.StartsWith("@"))

{

office = ParsePublishh(line);

publisher.Add(office);

}

else if(line.StartsWith("#")&& office != null)

{

var category = ParseCate(line);

office.Categories.Add(category);

}

}

}

}

}

catch (Exception)

{

}

return publisher;

}

public void Save(List<Models.Publish> pub)

{

using (var stream = new FileStream(filePath, FileMode.Create, FileAccess.Write))

{

using (var sw = new StreamWriter(stream))

{

foreach (var item in pub)

{

sw.WriteLine("@{0}", item.Name);

foreach (var cate in item.Categories)

{

sw.WriteLine("#{0}^{1}", cate.Name, cate.Link);

}

}

}

}

}

private Publish ParsePublishh(string line)

{

return new Publish()

{

Name = line.Substring(1).Trim()

};

}

private Category ParseCate(string line)

{

var lines = line.Substring(1).Split('^');

return new Category()

{

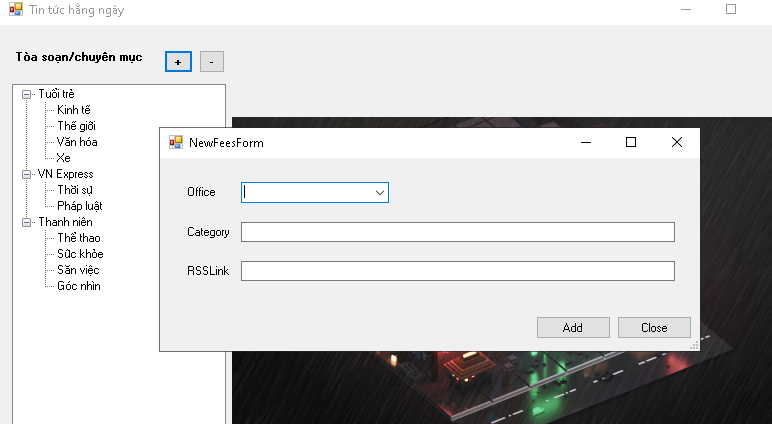
Name = lines[0].Trim(),

Link = lines[1].Trim()

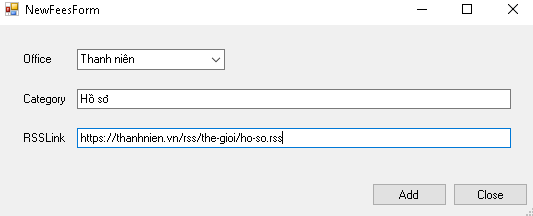
};

}

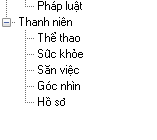
//Chức năng thêm chuyên mục



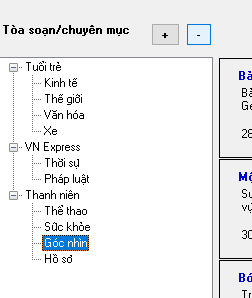
//Màn hình thêm chuyên mục



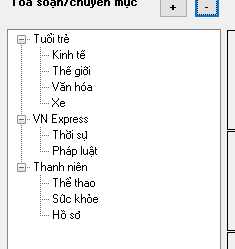
//Kết quả



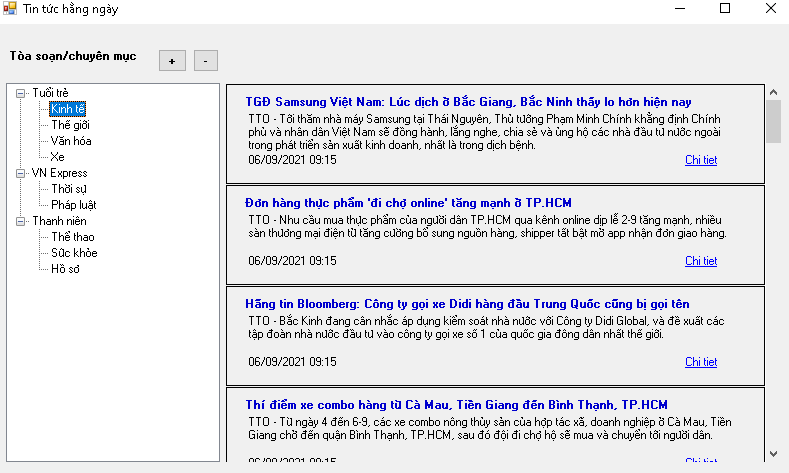
//Xóa chuyên mục



//Kết quả



//Thông tin trong tin tức hằng ngày



//Nhấn vô Link chi tiết



//UserControl

