

T.C.
KIRKLARELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

LİSANS BİTİRE PROJESİ

BİTİRME PROJESİNİN ADI: Access kontrol sistemi hazırlanır. RFID tanıma kullanılır, etiket bilgileri veritabanında saklanır. Bilgisayar ortamında gerçekleştirilir.

ÖĞRENCİ ADI SOYADI Alexandr VELICOGLO 1160501602

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME PROJESİ DANIŞMANI: DR. ÖĞR. ÜYESİ SITKI KOCAOĞLU

OCAK, 2021

ONAY SAYFASI

Kırklareli Üniversitesi, Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü,
1160501602 No'lu Alexandr VELICOGLO isimli öğrencisi "RFID access kontrol sistemi "
başlıklı bitirme projesini aşağıda imzası olan danışman önünde başarı ile sunmuştur.

Bitirme Projesi Danışmanı : Dr. Adı SOYADI DR. ÖĞR. ÜYESİ SITKI KOCAOĞLU

Kurum Adı: KIRKLARELİ ÜNİVERSİTESİ

ÖZ

Günümüz **ID** okuma teknolojilerinden olan RFID etiket okuma sistemi ihtiyaç alanı ve sektörlerine göre mesafe farkı ve etiket içi veri dizisinin büyüklüğüne göre tercih farklılığı gösterebilir. Kullanılan etiketlerin zaman diliminde hızlı gerçekleşmesi arz eden durumlarda kullanıcının yalnızca idsi alınabilir böylece okunan verinin veri tabanında varlığı kontrol edilebilir. Böylece ilerleyen algoritmik tasarımlarda işlemler yapılır.

Bu çalışmada ulaşılması kolay komponentlerden arduino nano ve modülü olan RC522 rfid okuma kullanılmıştır. Yazılım esnekliği için Windows işletim sisteminde .Net tabanlı Visual Studio kullanılarak C# dilinde masaüstü uygulaması hazırlanmıştır. Uygulama ile XML veri tabanı ilk açılışta dinamik olarak programca kontrol edilir yoksa kullanıcıya bilgi vererek otomatik oluşturulur. Çalışmada veri tabanının XML olarak kullanılmasının temel sebeplerinden RFID modülü gibi ilk kez kullanılıyor olmasından. Okuma ve yazma hızlarının “tag”lara göre gerçekleştiğinden oldukça yüksek olmasından XML için diyebileceğimiz: indexlemeyi doğru gerçekleştirmeliyiz.

C# uygulaması seri porttan bağlanabilmesi için bilgisayarda ön yüklü seri portları ve arduinonun bağlı olduğu port görülmektedir. Porta bağlanmak için seçilmesi ve Connect butonuna basılması yeterlidir. Uygulamanın belli başlı algoritmaları Try-Catch’e uygun olarak yazılmıştır. Bağlantının gerçekleşmesi durumunda saniyede 5 kez gönderim yapan arduinodan veriler okunur ve ilk açılışta tanımlanan veri tabanı yolunu izleyerek kart idsinin varlığını kontrol eder. Etiketlerde kart idsi aranırken ilk kayıta alınan kart sahibinin adı kayıtlarda saklıdır. Kaydın bulunması durumunda ekrana erişim verildi ve bulunduğunda dair başarı mesajı iletir. Aksi durumda ekranın sağında yer alan Yeni Kayıt grubuna kart idsi otomatik iletilir ve kullanıcıdan kayıt için Ad-Soyad girilmesi beklenir. Veri tabanında kayıt esnalarında boş alanların ileri zamanlarda güvenlik boşluğu oluşturmamaları adına zorunlu alanların doldurulması gerektiği ile ilgili CheckBox atanmıştır, anca verilerin doğruluğundan emin olunduktan sonra seçilen Captcha-türevi kontrol mekanizması kayıt butonunu aktif eder. Çalışabilirliği aktif olan buton XML’de yeni tag ve içerik oluşturur. Kullanılan diğer kontrol mekanizmalarından biri de: uygulama kapatılmaya çalışılırsa ve kullanıcı porttan bağlantısını koparmayı unutursa işletim sistemince sonlandırılması gereken bağlantı kontrol algoritmasıyla kapatılır.

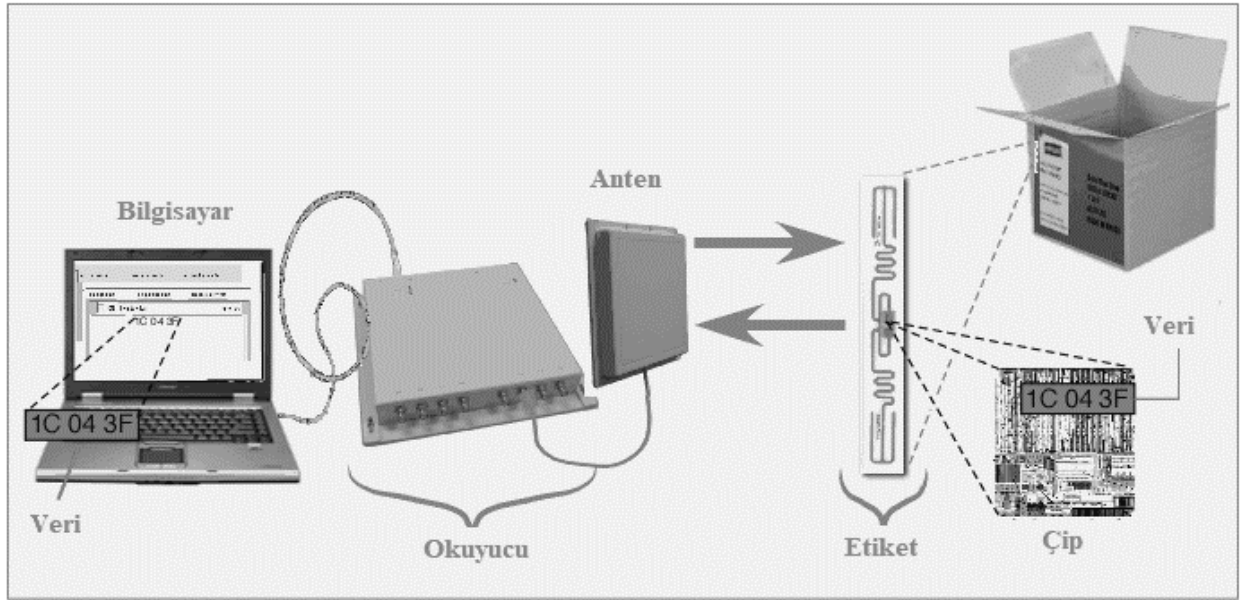
Uygulamada geliştirilmek üzere hazırlanan bu çalışmalar kartların ilk 4 sektöründe bulunan id ye göre işlemleri temsil eder ve kullanımını kolaylaştırmak üzere github’ta açık kaynak olarak yer almaktadır.

RFID Teknolojisi ile İlgili Genel Bilgiler

RFID (Radio Frequency Identification) Teknolojisi, nesnelere iliştirilmiş etiketleri otomatik olarak tanımlamak ve izlemek için elektromanyetik sinyaller kullanır. Etiketler elektronik olarak depolanan bilgileri içerir. Pasif etiketler, yakındaki bir RFID okuyucunun tetiklediği radyo dalgalarından enerji toplar. Aktif etiketler ise dahili bir güç kaynağına (batarya) sahiptir ve RFID okuyucusundan uzun mesafelerde çalışabilir.

Tipik bir RFID Sistemi, etiket, okuyucu, anten ve bilgisayarlar olmak üzere çeşitli donanımlardan oluşur. Her etiketin uygulamanın türüne bağlı olarak kendi salt okunur (read only) veya yeniden yazılabilir (rewrite) bir belleği vardır. RFID okuyucusu, RFID sisteminin menzili içindeki nesneleri (etiketler aracılığıyla) bulmasını sağlayan manyetik alanlar oluşturur. Okuyucu tarafından üretilen yüksek frekanslı elektromanyetik sinyal, sorguya cevap vermek için etiketleri tetikler; sorgu sıklığı saniyede 50 kereye kadar çıkabilir. Sonuç olarak, sistemin ana bileşenleri, yani etiketler ve okuyucu arasında iletişim bu şekilde kurulur, büyük miktarlarda veri üretilebilir (Ajami, 2013).

Temel çalışma prensibi olarak RFID Teknolojisi; bir bilgisayar, bir okuyucu, anten ve en az bir RFID etiketi içeren bir sistemdir. Bilgisayar okuyucuya şifrelenmiş bir veri içeren (ethernet veya seri bağlantı yoluyla) komutlar gönderir. Bu komutlar, düşük seviyeli bir kodlama dili kullanıldığında belirli durumlar için programlanabilir ancak çoğu zaman bu komutlar, üretici tarafından sağlanan önceden belirlenmiş bir komut ailesinden alınır. Etiketleri okuma, etiketleri yazma veya sonlandırma, okuyucu veya anten ayarlarını gerçekleştirme gibi farklı komut türleri vardır.



Şekil - 1 : RFID Teknolojisi Genel Çalışma Prensibi (Peabody, 2013).

Bilgisayardan gelen komutla okuyucu anten yardımıyla bir radyo sinyali gönderir. Gönderilen bu sinyal aralığında bir RFID etiketi mevcut ise radyo dalgaları tarafından entegre çip içerisine bir akım indüklenir ve içerisindeki veriyi okur. Okunan bu veri tekrar antenler yardımıyla okuyucuya gönderilir. Okuyucu aldığı bu veriyi daha ileri işlemler amacıyla bilgisayara iletir (Peabody, 2013).

Günümüzde perakende, depolama, sağlık, güvenlik, envanter yönetimi, insan kaynakları gibi birçok sektörde çeşitli RFID uygulamaları güvenle kullanılmaktadır. Bu sayede çalışan verimliliği artması ve süreçlerdeki sorunların ortadan kalkması hedeflenir.

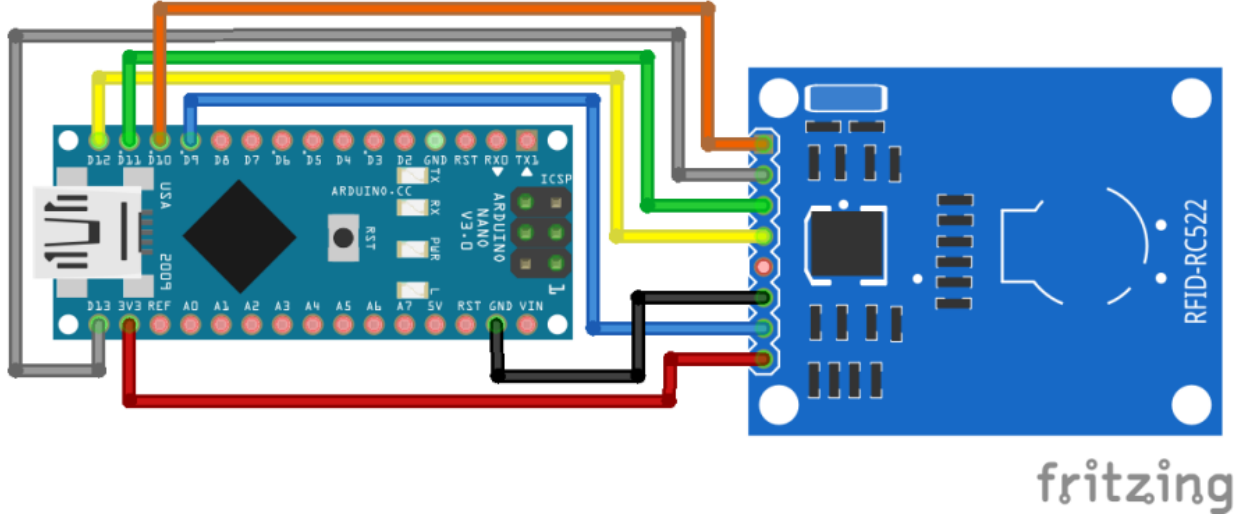
Amerika Birleşik Devletleri'nde Wal-Mart mağazaları teslim edilen tüm ürünlerinin paketlenmesi ve dağıtılması aşamasında pasif etiket tabanlı RFID Teknolojisinden yararlanmaktadır. 2005 yılından itibaren ilk 100 tedarikçisinde uygulamayı başlatmış ve 2006 yılında tüm tedarikçilerine bu uygulamayı standart hale getirmiştir (Yen Chieh Huang, 2011).

Tayland'da bir havalimanının operasyonel verimliliğini artırmak amacıyla yolcu check-in işlemlerinde RFID Teknolojisinden faydalanılmıştır. Yapılan çalışma ile yolcuların havalimanında hareket tarzlarına göre simulasyon modeli geliştirilmiş, bu sayede kaynakların etkin bir şekilde tahsis edilmesi sağlanarak operasyonel çözümler elde edilmiştir (Trakoonsanti, 2016).

Özellikle hastanelerde, farklı ve karmaşık yapıları nedeniyle bu teknolojiyi uygulamak çok önemlidir. Son yıllarda, hasta bakım süreçlerinin yol açtığı zararlar giderek artmaktadır. Bilgisayar ve yazılımların gelişmesi, bilgi teknolojilerinin süreçlere dahil edilmesi ile birlikte hasta bakımını kolaylaştırmak, hataları yakalamak ve düzeltmek, kararlara yardımcı olmak ve performans hakkında geri bildirim sağlamak suretiyle bu riskleri azaltmada hayati bir rol oynaması öngörülmektedir (Ajami, 2013).

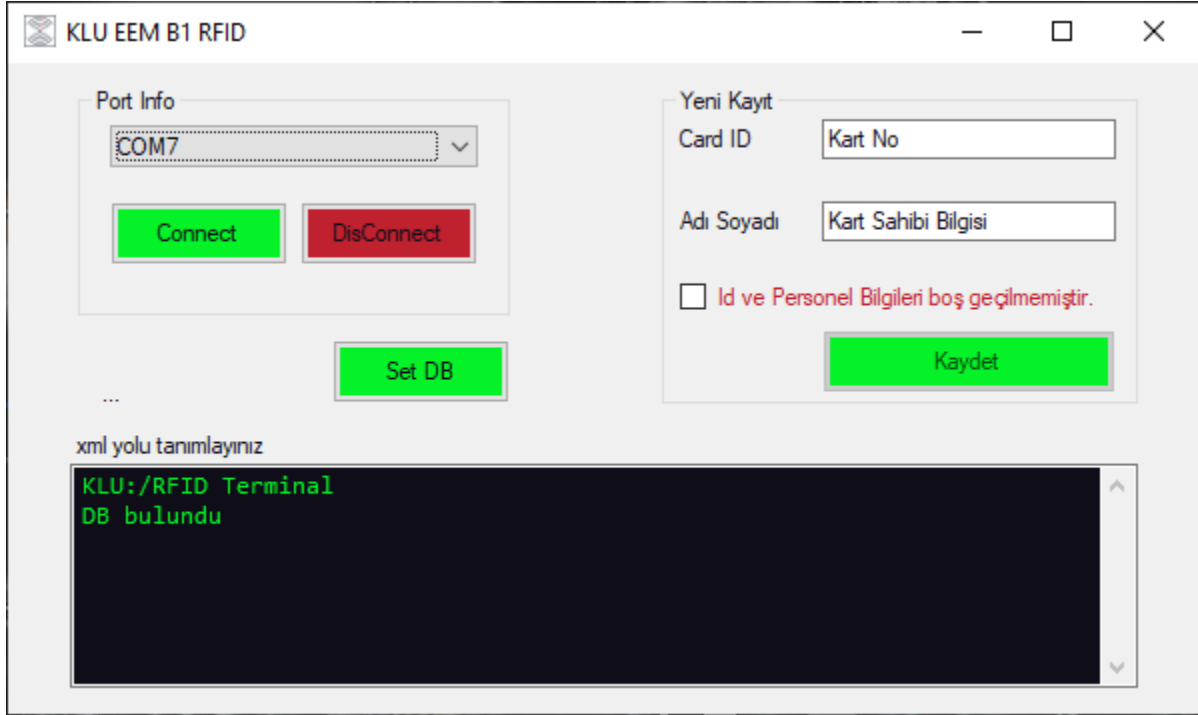
Modül ile İlgili Önemli Notlar

Okuma hızı saniyede 424 kBit, çekilen akım ve voltaj; 13-26mA 3,3V. Haberleşme için kullandığı SPI protokolü Master Input Slave Output pinlerinden data transferini sağlar.



Şekil - 2 : RFID Modülünün Bağlantı Şeması

Zaman kayıplarını en aza indirmek için EEPROM okumada minimum optimize kod kullanılmıştır tüm yük bilgisayar tabanlı uygulamaya iletilmiştir. Sırasıyla Arduino ve C# uygulama kodları aşağıda ekran görüntüleriyle mevcuttur;



Şekil - 3 : Masaüstü Uygulamasının Ekran Görüntüsü

Arduino içeriği;

```
# include <SPI.h>
# include <MFRC522.h>
# include <EEPROM.h>

#define RST_PIN 9
#define SS_PIN 10
#define ledPin 13

MFRC522 mfrc522(SS_PIN, RST_PIN);

String lastRfid = "";
int j = 0;

MFRC522::MIFARE_Key key;
void setup()
{
    Serial.begin(9600);
    SPI.begin();
    mfrc522.PCD_Init();
    pinMode(ledPin, OUTPUT);
}

void loop()
{
    if (!mfrc522.PICC_IsNewCardPresent())
    {
        return;
    }
    if (!mfrc522.PICC_ReadCardSerial())
    {
        return;
    }
    String rfid = "";
    for (byte i = 0; i < mfrc522.uid.size; i++) //mfrc522.uid.size
    {
        rfid += mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ";
        rfid += String(mfrc522.uid.uidByte[i], HEX);
    }
    rfid.trim();
    rfid.toUpperCase();
    Serial.print(rfid);
    delay(200);
}
```

Böylece saniyede 5 kez okuma gerçekleşir ve yalnızca mifare kartların tanımlı uzunluğu kadar fongü döner (varsayılan 4 sektör HEX)

C# İçeriği;

```
public Form1()
{
    InitializeComponent();
}
string[] portlar = SerialPort.GetPortNames();
int aa = 0;
string ardGelenStr = "";
string dbName = "rfidDB.xml";
string path = "";
private void Form1_Load(object sender, EventArgs e)
{
    buttonYeniKayit.Enabled = false;
    foreach (string portAdi in portlar)
    {
        comboBoxSerials.Items.Add(portAdi);
        comboBoxSerials.SelectedIndex = 0;
    }
    if
(!File.Exists(@Environment.GetFolderPath(Environment.SpecialFolder.Desktop) +
"/rfidDB.xml"))
    {
        textBoxConsole.AppendText("\r\nDesktop->rfidDB not exist! please press
SetDB");
    }
    else
    {
        textBoxConsole.AppendText("\r\nDB bulundu");
    }
}

private void button1_Click(object sender, EventArgs e)
{
    timer1gibi.Start();
    if (serialPortArd.IsOpen == false)
    {
        serialPortArd.PortName = comboBoxSerials.Text;
        serialPortArd.BaudRate = 9600;
        //serialPortArd.Open();
        //labelMessage.Text = "conn++";
        try
        {
            serialPortArd.Open();
            textBoxConsole.AppendText("\r\nconn");
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
}
```



```

timer1gibi.Stop();
if (serialPortArd.IsOpen == true)
{
    serialPortArd.Close();
    textBoxConsole.AppendText("\r\nendisconn");
}
else
{
    textBoxConsole.AppendText("\r\nendisconn error");
}
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    if (serialPortArd.IsOpen == true)
    {
        serialPortArd.Close();
    }
}

private void timer1gibi_Tick(object sender, EventArgs e)
{
    try
    {
        aa = 0;
        //labelMessage.Text = "";
        ardGelenStr = serialPortArd.ReadExisting();
        labelMessage.Text = ardGelenStr.ToString();
        if (ardGelenStr.ToString() != "")
        {
            textBoxConsole.AppendText("\r\n" + ardGelenStr.ToString());
            bool varmi = idAra(ardGelenStr.ToString());
            if (varmi == true)
            {
                textBoxConsole.AppendText("\r\nErişime izin verildi!!");
            }
        }
    }
    catch (Exception)
    {
    }
}

private bool idAra(string v)
{
    XDocument xdoc = XDocument.Load(@path + "/rfidDB.xml");
    foreach (XElement contact in xdoc.Element("rfid").Elements("card"))
    {
        XAttribute numAttr = contact.Attribute("id");
        textBoxConsole.AppendText("\r\nAranan: " + v);
        if (numAttr.Value.ToString() == v)
        {
            textBoxConsole.AppendText("\r\nKayıt mevcuttur!");
            return true;
        }
    }
}

```

```

        break;
    }
}
textBoxConsole.AppendText("\r\nKayit mevcut değil!!");
textBoxCardId.Text = v;
return false;
}

private void button3_Click(object sender, EventArgs e)
{
    DialogResult d = new DialogResult();
    d = MessageBox.Show("Yalnızca ilk çalışmada DB atanmalı! Onaylıyor musunuz?", "DB Oluşturma", MessageBoxButtons.YesNo);
    if (d == DialogResult.Yes)
    {
        setDBXML();
    }

    /*
    OpenFileDialog file = new OpenFileDialog();
    file.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Desktop);
    //file.InitialDirectory = "C:\\";
    file.Filter = "Folders|\\n"; //Folders|\\n XML File |*.xml
    file.RestoreDirectory = true;
    file.Title = "Veri tabanı dosyasını seçiniz";
    file.ShowDialog();

    */
}

private void setDBXML()
{
    try
    {
        FolderBrowserDialog diag = new FolderBrowserDialog();
        if (diag.ShowDialog() == System.Windows.Forms.DialogResult.OK)
        {
            path= diag.SelectedPath; //selected folder path
            textBoxConsole.AppendText("\r\nSeçilen directory: " + path);
            if (File.Exists(@path + "/rfidDB.xml"))
            {
                //MessageBox.Show("file exist");
                textBoxConsole.AppendText("\r\nDB mevcut");
            }
            else
            {
                textBoxConsole.AppendText("\r\nDB bulunamadı...");
                MessageBox.Show(path);
                using (FileStream fs = File.Create(@path + "/rfidDB.xml"))

                textBoxConsole.AppendText("\r\ndosya otomatik oluşturuldu" + "
rfidDB.xml");

                dbIntroYaz();
            }
        }
    }
    catch { }
}

```

```

        }
        //MessageBox.Show(folder);
        labelDBdir.Text = path + "\\\" + dbName;
    }
}
catch (Exception)
{
    MessageBox.Show("Seçilen klasöre dosya oluşturulamadı!");
}
}

private void dbIntroYaz()
{
    XDocument xdoc = new XDocument();
    XElement el1 = new XElement("rfid");
    XAttribute attr = new XAttribute("id", "DA E3 76 82"); //DA E3 76 82
    XElement el2 = new XElement("card", "system");
    el2.Add(attr);
    el1.Add(el2);
    xdoc.Add(el1);
    xdoc.Save(@path + "/rfidDB.xml");
    textBoxConsole.AppendText("\r\ndosya içeriği otomatik oluşturuldu");
}

private void button1_Click_1(object sender, EventArgs e)
{
    dbIdBul();
}

private void dbIdBul()
{
    XDocument xdoc = XDocument.Load(path+"/rfidDB.xml");
    foreach (XElement contact in xdoc.Element("rfid").Elements("card"))
    {
        XAttribute numAttr = contact.Attribute("id");

        textBoxConsole.AppendText("id: " + numAttr.Value.ToString());
        textBoxConsole.AppendText("name: " + contact.Value);
    }
}

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.Checked)
    {
        buttonYeniKayit.Enabled = true;
    }
    else
    {
        buttonYeniKayit.Enabled = false;
    }
}

private void buttonYeniKayit_Click(object sender, EventArgs e)
{
    newKayit(textBoxCardId.Text, textBoxAdSad.Text);
}

```

```

        textBoxAdSad.Text = "";
        textBoxCardId.Text = "";
    }

    private void newKayit(string id, string ad)
    {
        XDocument xdoc = XDocument.Load(path + "/rfidDB.xml");
        XElement root = xdoc.Element("rfid");
        XAttribute attr = new XAttribute("id", textBoxCardId.Text);
        XElement el2 = new XElement("card", textBoxAdSad.Text);
        el2.Add(attr);
        root.Add(el2);
        xdoc.Save(path + "/rfidDB.xml");
    }

    private void button2_Click_1(object sender, EventArgs e)
    {
        /*
        XDocument xdoc = new XDocument();
        XElement el1 = new XElement("rfid");
        XAttribute attr = new XAttribute("id", "DA E3 76 82"); //DA E3 76 82
        XElement el2 = new XElement("card", "system");
        el2.Add(attr);
        el1.Add(el2);
        xdoc.Add(el1);
        xdoc.Save(@"idDB.xml");
        */
        XDocument xdoc = XDocument.Load("idDB.xml");
        XElement root = xdoc.Element("rfid");
        XAttribute attr = new XAttribute("id", textBoxCardId.Text);
        XElement el2 = new XElement("card", textBoxAdSad.Text);
        el2.Add(attr);
        root.Add(el2);
        xdoc.Save("idDB.xml");
    }
}

```

Uygulamada işlemler ayrı metotlar halinde yazılmasına özen gösterilmiştir böylece bellek yönetimi, try-catch kullanılarak programsal kesmelerin önüne geçilmesi gibi optimizasyonlar yapılmıştır.

Kaynakalar:

Trakoonsanti, 2016

Yen Chieh Huang, 2011

Peabody 2013

Ajami, 2013

Yazılımlar için en başta StackOverFlow olmak üzere ve diğer topluluklardan yararlanılmıştır