

CeyPASS Cihaz Paneli

Proje Hakkında

CeyPASS Cihaz Paneli, ZKTeco marka biyometrik cihazların (parmak izi okuyucu, yüz tanıma terminalleri) merkezi olarak yönetilmesini sağlayan bir Windows Forms uygulamasıdır. Uygulama, personel ve kart bazlı erişim kontrolü, cihaz yönetimi, offline veri toplama ve yetkilendirme işlemlerini gerçekleştirir.

Özellikler



Kullanıcı Yönetimi

- Güvenli kullanıcı girişi
- Firma bazlı yetkilendirme
- Kullanıcı oturum yönetimi



Personel ve Kart Yönetimi

- Aktif personel listesi görüntüleme
- Puantajsız kart yönetimi
- Personel ve kart bazlı cihaz yetkilendirme
- Toplu tanımlama ve silme işlemleri



Cihaz Yönetimi

- ZKTeco cihazlarına TCP/IP üzerinden bağlantı
- Cihaz durumu izleme (bağlı/bağlı değil)
- Cihaz bilgilerini görüntüleme (model, seri no, firmware, MAC adresi)
- Kullanıcı ve log kapasitesi takibi
- Saat senkronizasyonu
- Cihaz yeniden başlatma ve kapatma
- Tüm logları ve kullanıcıları silme



Offline Veri Yönetimi

- Cihazlardan offline veri çekme
- Giriş/çıkış kayıtlarını görüntüleme
- Excel formatında veri dışa aktarma
- Offline veri temizleme



Yetkilendirme Sistemi

- Personel bazlı cihaz yetkilendirme
- Kart bazlı cihaz yetkilendirme
- Veritabanı ile senkronize yetki yönetimi
- Toplu yetki ekleme/çıkarma



Otomatik Güncelleme

- Uygulama başlangıcında otomatik güncelleme kontrolü
- Zorunlu güncelleme desteği
- HTTP üzerinden güncelleme paketi indirme

Teknoloji Stack

Framework ve Dil

- .NET Framework 4.7.2
- C# (Windows Forms)

Mimari Katmanlar

Proje, katmanlı mimari (Layered Architecture) prensiplerine göre tasarlanmıştır:

1. **CeyPASSCihazPanel.UI** - Kullanıcı Arayüzü Katmanı
 - Windows Forms uygulaması
 - Kullanıcı etkileşimi ve görsel bileşenler
2. **CeyPASSCihazPanel.Business** - İş Mantığı Katmanı
 - Servis sınıfları
 - İş kuralları ve validasyonlar
 - Cihaz bağlantı yönetimi
3. **CeyPASSCihazPanel.DAL** - Veri Erişim Katmanı

- Repository pattern implementasyonu
- SQL Server veri erişimi
- CRUD operasyonları

4. **CeyPASSCihazPanel.Entities** - Varlık Katmanı

- Model sınıfları
- Veri transfer objeleri (DTO)

Kullanılan Kütüphaneler ve Bağımlılıklar

Kütüphane	Versiyon	Kullanım Amacı
zkemkeeper	1.0	ZKTeco cihaz SDK'sı - Biyometrik cihaz iletişimi
AutoUpdater.NET.Official	1.9.2	Otomatik uygulama güncelleme
Microsoft.Extensions.DependencyInjection	10.0.0	Dependency Injection container
Microsoft.Web.WebView2	1.0.2592.51	Modern web içeriği görüntüleme
System.Data.SqlClient	-	SQL Server veritabanı bağlantısı

Veritabanı

- **SQL Server** (CeyPASS veritabanı)
- Bağlantı bilgileri App.config dosyasında yapılandırılır

Proje Yapısı

```
CeyPASSCihazPanel/  
├─ CeyPASSCihazPanel.UI/           # Kullanıcı Arayüzü  
│   ├── Forms/  
│   │   ├── loginForm.cs          # Giriş formu  
│   │   └─ anaForm.cs             # Ana uygulama formu  
│   ├── Program.cs                 # Uygulama giriş noktası  
│   └─ App.config                  # Yapılandırma dosyası  
│  
├─ CeyPASSCihazPanel.Business/     # İş Mantığı  
│   ├── Abstractions/  
│   │   ├── IAuthService.cs        # Kimlik doğrulama servisi  
│   │   ├── IDeviceService.cs      # Cihaz yönetim servisi  
│   │   └─ IAdminLookupService.cs  # Yönetim lookup servisi  
│   └─ Services/  
│       ├── AuthService.cs  
│       ├── DeviceService.cs  
│       └─ AdminLookupService.cs  
│  
├─ CeyPASSCihazPanel.DAL/          # Veri Erişim  
│   ├── Abstractions/  
│   │   ├── IUserRepository.cs  
│   │   ├── ICihazRepository.cs  
│   │   ├── IPersonelRepository.cs  
│   │   ├── IKisiCihazYetkiRepository.cs  
│   │   ├── IPuantajsizKartRepository.cs  
│   │   └─ IPuantajsizKartCihazYetkiRepository.cs  
│   └─ Repositories/  
│       ├── SqlUserRepository.cs  
│       ├── SqlCihazRepository.cs  
│       ├── SqlPersonelRepository.cs  
│       ├── SqlKisiCihazYetkiRepository.cs  
│       ├── SqlPuantajsizKartRepository.cs  
│       └─ SqlPuantajsizKartCihazYetkiRepository.cs  
│  
├─ CeyPASSCihazPanel.Entities/     # Varlıklar  
│   └─ Models/  
│       ├── Kullanici.cs           # Kullanıcı modeli  
│       ├── Personel.cs            # Personel modeli  
│       ├── Terminal.cs            # Cihaz modeli  
│       └─ CihazBilgi.cs           # Cihaz bilgi modeli
```

```
|      |— OfflineLog.cs          # Offline log modeli
|      |— PuantajsızKart.cs      # Puantajsız kart modeli
|      |— ...
|
|— CeyPASSCihazPanel.Setup/      # Kurulum projesi
```

Kurulum ve Çalıştırma

Gereksinimler

- Windows 10 veya üzeri
- .NET Framework 4.7.2 Runtime
- SQL Server (2012 veya üzeri)
- Visual Studio 2022 (geliştirme için)
- ZKTeco cihazlar için zkemkeeper COM bileşeni

Veritabanı Yapılandırması

1. App.config dosyasını açın
2. Connection string'i kendi SQL Server bilgilerinize göre güncelleyin:

```
<connectionStrings>
  <add name="CeyPASS"
        connectionString="Server=SUNUCU_ADI\INSTANCE;Database=CeyPASS;User Id=KULLANICI_ADI;Pa:
</connectionStrings>
```

Geliştirme Ortamı Kurulumu

1. **Projeyi klonlayın veya indirin**
2. **Visual Studio'da açın**

CeyPASSCihazPanel.sln dosyasını Visual Studio ile açın

3. **NuGet paketlerini geri yükleyin**

Visual Studio'da: Tools > NuGet Package Manager > Restore NuGet Packages

4. **zkemkeeper COM referansını kaydedin**

- ZKTeco SDK'sını kurun

- COM referansı otomatik olarak gömülü (EmbedInteropTypes=True)

5. Projeyi derleyin

Build > Build Solution (Ctrl+Shift+B)

6. Uygulamayı çalıştırın

Debug > Start Debugging (F5)

Üretim Dağıtımı

1. Release build oluşturun

Configuration: Release

Build > Build Solution

2. Setup projesi ile kurulum paketi oluşturun

- CeyPASSCihazPanel.Setup projesi ile MSI kurulum dosyası oluşturulabilir

3. Otomatik güncelleme yapılandırması

- update.xml dosyasını web sunucusuna yerleştirin
- Program.cs içindeki güncelleme URL'ini güncelleyin

Kullanım

Giriş Yapma

1. Uygulamayı başlatın
2. Kullanıcı adı ve şifrenizi girin
3. Firma seçin (opsiyonel)
4. "Giriş" butonuna tıklayın

Personel Tanımlama

1. Ana formda "Personel/Kart" sekmesini açın
2. Personel listesinden bir personel seçin
3. Yetkili olacağı cihazları işaretleyin
4. "Tanımla" butonuna tıklayın

Toplu Tanımlama

1. "Toplu Tanımla" sekmesini açın
2. Personelleri seçin
3. Cihazları işaretleyin
4. "Toplu Tanımla" butonuna tıklayın

Offline Veri Çekme

1. "Offline Veri" sekmesini açın
2. Cihazları seçin
3. "Veri Çek" butonuna tıklayın
4. Veriler DataGridView'de görüntülenir
5. "Excel'e Aktar" ile dışa aktarabilirsiniz

Cihaz Yönetimi

1. "Cihaz Yönetimi" sekmesini açın
2. Cihaz bilgilerini görüntüleyin
3. İşlemler:
 - Saat Senkronize Et
 - Cihazı Yeniden Başlat
 - Cihazı Kapat
 - Tüm Logları Sil
 - Tüm Kullanıcıları Sil

Mimari Kararlar ve Tasarım Desenleri

Dependency Injection

- Microsoft.Extensions.DependencyInjection kullanılarak IoC container implementasyonu
- Servis ve repository'ler constructor injection ile enjekte edilir
- Loosely coupled, test edilebilir kod yapısı

Repository Pattern

- Veri erişim katmanında repository pattern kullanımı
- Her entity için ayrı repository interface ve implementasyonu
- Veri kaynağı değişikliklerinde esneklik

Service Layer Pattern

- İş mantığı business katmanında servisler içinde kapsüllenir
- UI katmanı sadece servisleri kullanır, doğrudan repository'lere erişmez
- Separation of concerns prensibi

Asenkron Programlama

- Cihaz işlemleri için async/await pattern kullanımı
- UI thread'in bloke olmaması için Task-based operations
- Responsive kullanıcı deneyimi

Veritabanı Şeması

Ana Tablolar

- **Kisiler** - Personel bilgileri
- **PuantajsizKartlar** - Puantajsiz kart bilgileri
- **Cihazlar** - Terminal/cihaz bilgileri
- **KisiCihazYetki** - Personel-cihaz yetkilendirme ilişkisi
- **PuantajsizKartCihazYetki** - Kart-cihaz yetkilendirme ilişkisi
- **Kullanıcılar** - Sistem kullanıcıları

Güvenlik

- Kullanıcı kimlik doğrulaması
- Firma bazlı veri izolasyonu
- SQL injection koruması (parametrelili sorgular)
- Bağlantı string'leri yapılandırma dosyasında

Performans Optimizasyonları

- Connection pooling (SQL Server)
- Asenkron cihaz işlemleri
- Lazy loading için ComboBox dropdown genişliği hesaplama
- DataGridView için sanal mod desteği (büyük veri setleri için)

Hata Yönetimi

- Try-catch blokları ile exception handling
- Kullanıcıya anlamlı hata mesajları
- Log yazma mekanizması (UI üzerinde)
- Cihaz bağlantı hatalarında otomatik yeniden bağlanma

Bilinen Sınırlamalar

- Sadece ZKTeco marka cihazlar desteklenir
- Windows işletim sistemi gereklidir
- SQL Server veritabanı bağımlılığı
- Aynı anda tek kullanıcı oturumu

Gelecek Geliştirmeler

- ☐ Web tabanlı yönetim paneli
- ☐ Mobil uygulama desteği
- ☐ Çoklu dil desteği
- ☐ Detaylı raporlama modülü
- ☐ Gerçek zamanlı cihaz durumu bildirimleri
- ☐ API entegrasyonu

Lisans

Bu proje özel mülkiyettir ve telif hakkı koruması altındadır.

İletişim ve Destek

Sorularınız veya sorunlarınız için lütfen sistem yöneticinizle iletişime geçin.

Versiyon: 1.0

Son Güncelleme: 2025

Geliştirici: Tahir Koca