

## Veriket – Yazılım Uygulama Testi

### Kısıtlar

- Uygulama C# programlama dili kullanılarak .Net Framework ile veya .Net Core ile gerçekleştirilecek.
- Bir solution içerisinde 2 proje olacak.
  - 1. Proje istenen formattaki logları üreten bir Windows servis projesi.
  - 2. Proje windows servisin ürettiği logları gösteren bir winform uygulaması.
- Windows servis tarafından üretilen loglar CSV formatında **VeriketAppTest.txt** isim dosyada **Tarih, bilgisayar adı, kullanıcı adı** bilgilerini yazacak.
- İki uygulama da tek bir MSI paketinden veya yazılımcı tarafından hazırlanan özel bir programdan tek seferde kurulabilecek.
- Program Windows'un yüklü olduğu sürücü içerisindeki **Program Files** içerisine **VeriketApp** klasörü altına kurulacak.
- Program kurulduktan sonra istenen tüm logları Windows'un yüklü olduğu sürücü içerisindeki **ProgramData** klasörünün altına VeriketApp adındaki bir klasörün altına oluşturulacak.
- Yukarıda bahsedilen klasörlerin yolları kod içerisinde statik olarak kullanılmayacak, gerekli sürücü ve klasör isimleri programatik olarak alınacak.
- Program kurulduktan sonra windows servis kurulumu yapan kullanıcı bilgileri ile otomatik olarak çalışacak. Kurulumu yapan kişi uygulamanın (windows servis) çalışması için ekstra bir işlem yapmamalı.

### Senaryo 1 (Windows Service)

Windows servis, kurulu olduğu bilgisayarın **ProgramData** klasörünün altına **VeriketApp** adındaki bir klasöre **VeriketAppTest.txt** olarak bir dosya oluşturacak. (C:\ProgramData\ dizini dinamik olacak, farklı bilgisayarlarda Windows D,E gibi farklı sürücülere kurulmuş olabilir.)

Windows Service (**Veriket Application Test**) dakikada 1 **VeriketAppTest.txt** dosyasına bir satır veri ekleyecek ve aşağıdaki bilgileri yazacak.

[Tarih] [Bilgisayar Adı] [Windowsta Oturum Açan Kullanıcı Adı]

**Service Name :** Veriket Application Test

**Service Description :** Veriket Applcation Test Service

Örneğin; Aşağıda farklı görevleri yerine getiren Veriket.Resource.Manager gibi.

> Windows Gezini	%0.1	39.8 MB	0 MB/sn	0 Mb/sn	%0
> Hizmet Ana Bilgisayarı: Tanı İlke...	%0	37.5 MB	0 MB/sn	0 Mb/sn	%0
> Discord (32 bit)	%0	34.0 MB	0 MB/sn	0 Mb/sn	%0
> Microsoft Windows Search Dizi...	%0	32.7 MB	0 MB/sn	0 Mb/sn	%0
> Görev Yöneticisi	%0.2	31.7 MB	0.1 MB/sn	0 Mb/sn	%0
> Microsoft Edge	%0.1	23.4 MB	0 MB/sn	0 Mb/sn	%0
> Veriket.Resource.Manager	%0.1	19.9 MB	0 MB/sn	0 Mb/sn	%0
> Hizmet Ana Bilgisayarı: DCOM S...	%0	19.7 MB	0 MB/sn	0 Mb/sn	%0
> Discord (32 bit) (3)	%0	19.2 MB	0 MB/sn	0 Mb/sn	%0
> Başlat	%0	15.9 MB	0 MB/sn	0 Mb/sn	%0
> Zoom Meetings	%0	14.8 MB	0 MB/sn	0 Mb/sn	%0
> Hizmet Ana Bilgisayarı: Uzaktan ...	%0.1	14.8 MB	0 MB/sn	0 Mb/sn	%0
> Microsoft Edge	%0.1	14.2 MB	0 MB/sn	0 Mb/sn	%0
> Hizmet Ana Bilgisayarı: UtcSvc	%0	13.0 MB	0 MB/sn	0 Mb/sn	%0

## Senaryo 2 (WinForm Uygulaması)

Üzerinde bir DataGridView ve bir Buton olan EXE uygulaması olacak. Tıklađımızda boş olarak gelsin. Butona tıklađında windows servisin ürettiđi CSV formatındaki logu okuyup ekrandaki DataGridView içerisine doldursun.