# **E-nahtar**

### Github Adresi

<https://github.com/GaziantepPython/E-nahtar>

### Grubu oluşturan öğretmenlerin listesi

Atilla SÜRER, Beyazıt ALAN, Gültekin ÖZÇELİK, Muharrem TAŞÇI, Murat CİNYOL, Şefik YALMAN, Yılmaz YALÇIN, Yusuf Zafer GÜLERARSLAN

# E-nahtar Analiz Raporu

### Kısa Özet

Günümüzde birçok eğitmen, öğrencinin birden fazla duyusuna hitap ettiği için akıllı tahta ile ders işlemenin akılda kalıcılığı arttırdığını ifade etmektedir. Bununla beraber eğitmenler ve eğitim kurumu yöneticileri; akıllı tahtaların yetkisiz kişilerce (öğrenci, kursiyer vb.) eğitim- öğretim faaliyetleri dışında gelişi güzel kullanılmasını engelleme ihtiyacı hissettiklerini belirtmektedirler. Bizlerin bu projedeki amacı; Debian GNU/Linux altyapısını kullanan Pardus işletim sistemi yüklü akıllı tahtaları, geliştireceğimiz yazılımla yetkisiz kişilerin kullanımını açılışta çalışan yazılımla kilitlemek, kullanılacak bir anahtar (USB bellek) yardımıyla da kilidi açmaktır. Ayrıca anahtar takılı değilse cihazın belli bir süre sonunda kapanmasını da sağlayarak enerji tasarrufu yapan çevreci bir cihaza dönüştürmektir.

### Problem Tanımı

Eğitim ortamlarında kullanılan akıllı tahtaya yetkisiz birinin erişimi engellenebilir mi?

Nasıl engellenebilir?

Akıllı tahta kullanılmaması halinde bir süre sonra kendi kendini kapatabilir mi?

### Analiz Süreci

#### İhtiyaç Analizi

Eğitim kurumu yöneticileri ve eğitmenler; eğitim ortamlarında kullanılan akıllı tahtaların

* Derslere verilen aralarda öğrencilerin tahtaları amacı dışında kullanıldığında eğitim ortamını ve öğrencileri olumsuz etkilediğini
* Özensiz kullanımın cihazların ekonomik ömrünü azalttığı, bakım masraflarını arttırdığını
* Arızalanan yedek parça için bekleme sürelerinde tahtaların kullanılamadığı için eğitim öğretim verimliliğinin düştüğünü
* Ders aralarında ve sonunda açık unutulan cihazların enerji tüketmeye devam ettiğini ifade etmektedirler.

Yukarıda açıklanan nedenlerle akıllı tahtaları; istenildiğinde kolayca kilitlemek ve açmak, kullanılmıyorsa da kendi kendini kapatması kullanıcılar tarafından talep edilmektedir.

**E-nahtar** projemizle kullanıcıların bu ihtiyaçları; açık kaynak kodlu yazılımların ortaya çıkış felsefesiyle uyumlu, tamamıyla açık kaynak kodlu ve geliştirmeye açık olacak şekilde yazılıp kullanıcılara sunulacaktır.

#### İçerik Analizi

**E-nahtar** python dilinde yazılacaktır. Böylece Linux işletim sistemi ile çalışan akıllı tahtalarda stabil çalışacaktır.

**E-nahtar** iki kısımdan oluşacaktır.

Birinci kısım, programımız akıllı tahta çalıştırıldığında işletim sistemi ile birlikte otomatik başlayacaktır. Tahta ekranına yetkili kullanıcının belirlemiş olduğu (kurumun logosu, uyarı yazısı, her ikisi vb.) görüntüyü getirecek ve akıllı tahtaya takılı klavye, mouse gibi donanımları kilitleyerek tahtayı yetkisiz kişilerin kullanımına kapatacaktır. Aynı yazılım anahtar (USB bellek) cihaza takıldığında içindeki şifrelenmiş dosyayı okuyup ekrandaki ve çevre donanımlardaki kilitleri kaldıracak ve tahtayı kullanıma açacaktır. Şayet anahtar takılmazsa veya takılan anahtar çıkartılırsa tahta kendini kilit pozisyonuna alacak ve enerji tüketimini sıfıra indirmek, yanı sıra cihazın ekonomik ömrünü uzatmak için belirlenecek sürede tahtanın kendi kendini kapatmasını sağlayacaktır.

İkinci kısım ise; yetkili kişilere ait USB belleklere, tahtaya takıldığında kilidi açan anahtar görevi görecek şifrelenmiş dosyanın eklenmesidir. Bunu yapabilmek içinde kullanıcı dostu bir arayüz yazılacaktır. Bu arayüz yardımıyla USB belleğe tahtanın kilidini açacak anahtar dizisini içeren dosya kolayca eklenebilecektir.

#### Durum Ortam Analizi

Kodlama için, Python programlama dili ve ihtiyaç halinde diğer diller,

Veri saklamak için, Python OS veya SQLite,

USB ile ilgili işlemler yapmak için PYUDEV,

Şifreli veri üretebilmesi için HASHLIB kütüphanesi,

Ara yüz için TKINDER, PYQT, HTML (HTML kullanılırsa WEBVIEW) kullanılacaktır.

İhtiyaç duyulması/duyulmaması halinde gereken eklemeler ve çıkarmalar yapılacaktır.

#### Kullanıcı Analizi

Proje GitHub a “**E-nahtar**” adıyla yüklenecektir. Böylece gerek kullanmak gerekse geliştirmek amacıyla tamamıyla açık kaynak yazılım olarak ihtiyaç duyan her kullanıcıya sunulmuş olacaktır.

Hedef kullanıcı kitlesi Milli Eğitim Bakanlığına bağlı, Pardus işletim sistemiyle çalışan akıllı tahta kullanılan resmi ve özel okul yöneticileri ile öğretmenlerden oluşmaktadır.

# E-nahtar Proje Tasarım Raporu

### Kısa Özet

Projenin başında oluşturulan "Gaziantep Python" isimli grup olarak zoom ve whatsupp üzerinden yapılan toplantılar, yazışmalar ve paylaşımlarla sürekli veri ve bilgi alışverişi yapıldı. Ayrıca oluşturulan bireysel "githup" hesapları bir "githup" grubunda birleştirildi. Yapılan görev dağılımları ve belirlen sürelerin sonucunda projemizin veri, arayüz ve kod tasarımları istenilen seviye gelindi. Tasarımlarda kullanıcı dostu arayüzler yapmaya gayret edildi. Proje logosu olarak da usb bellek şekli ile python logosunun renkleri, kapı anahtarı şeklinde birleştirilerek özgün bir tasarım yapıldı.

## Veri Tasarımı

Projemizde şu aşamada sadece kullanıcı giriş denetimleri için bir veritabanı kullanılacaktır. Projenin ilerleyen zamanlarından kilit ekranda gösterilen bilgiler, yetkili kullanıcı (öğretmen-usb bellek) kaydı gibi bilgilerde veritabanına eklenebilecektir.

Şekil 1:Veri Tasarımı

Şekil 1: Veri Tasarımı

## Ara yüz Tasarımı

Bu ekranda kayıtlı kullanıcılar kullanıcı adı ve şifreleriyle giriş yaparak programın ana menüsüne erişim sağlayabilecektir.

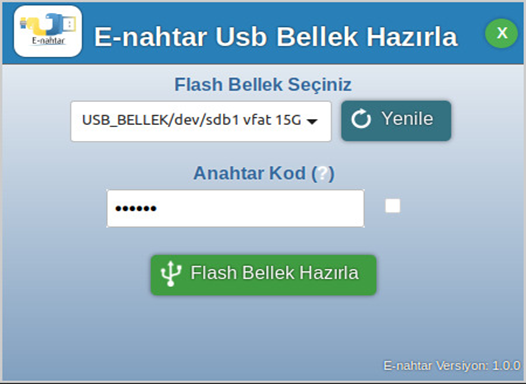
Şekil 2: Programın Giriş Ekranı



Bu ekranda anahtar flash oluşturma, kullanıcı yönetim işlemlerine giriş yapma ve çıkış işlemleri yapılacaktır.

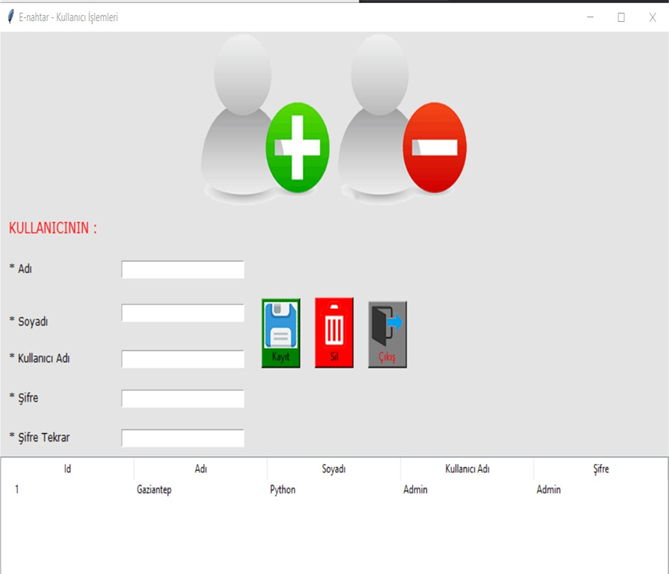
E-nahtar Logosu özgün olarak tasarlanmıştır.

Şekil 3: Programın ana menü ekranı



"E-nahtar Usb Bellek Hazırla" ara yüzü webview ile html, javascript ve css kullanılarak tasarlanmıştır. Bu arayüzle tahtanın kilidini açmak için kullanılacak usb bellekler hazırlanabilecektir.

Şekil 4: E-nahtar Usb Bellek Hazırla Ara yüzü



Kullanıcı işlemleri arayüzü ile kayıtlı kullanıcıları listeleme, kayıt ekleme ve silme işlemleri yapılabilecektir.

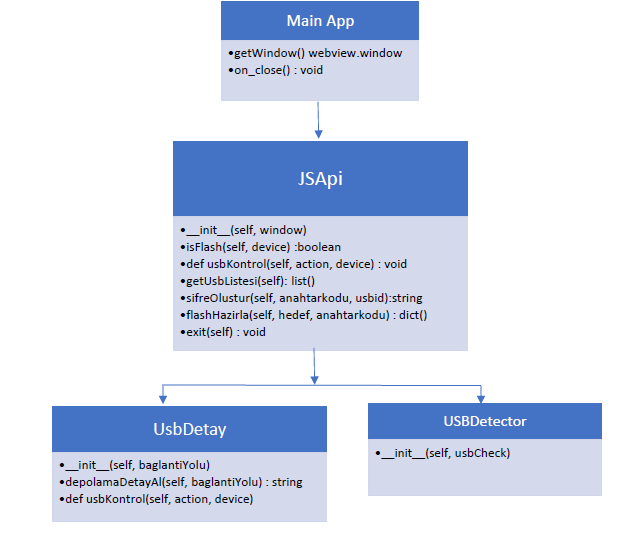
Şekil 5: Kullanıcı İşlemleri Ekranı

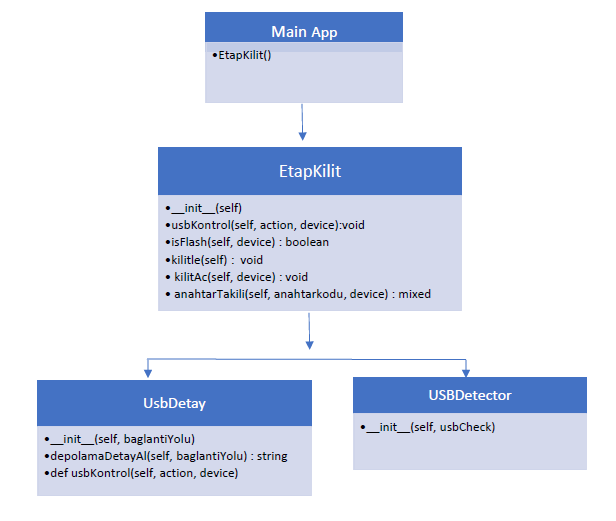
Aşağıda akıllı tahtadan E-nahtar çıkarıldığında karşılaşılacak ekran yer almaktadır. Akıllı tahta kilit ekranının tasarımı Webview kullanılarak yapılmıştır. Böylelikle istenilirse kilit ekranında herhangi bir internet sitesinin de görüntülenmesi mümkün olacaktır.



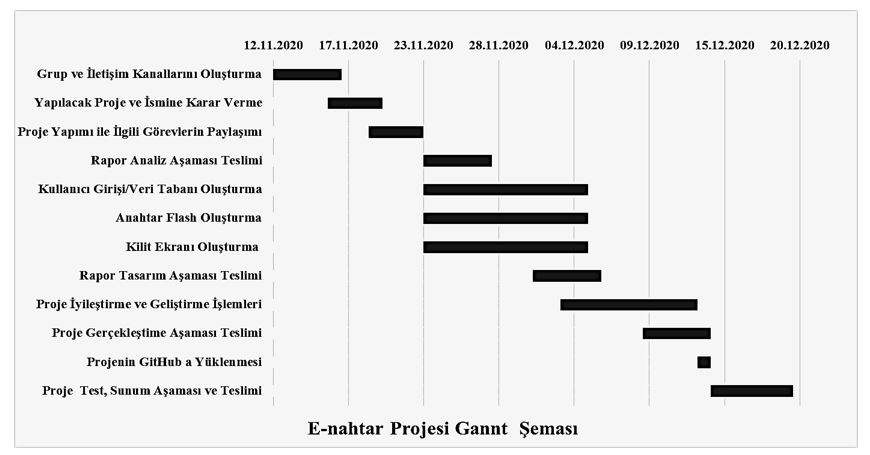
Şekil 6: Akıllı Tahta Kilit Ekranı

## Kod Tasarımı

1. **Flash Bellek Hazırlama**
2. **Kilit Ekranı**



## Zaman Çizelgesi

Yaptığımız proje toplantılarında süreç ile ilgili zaman çizelgesini aşağıdaki gibi oluşturduk.

# E-nahtar Gerçekleştirme Raporu

## Karşılaşılan Sorunlar ve Uygulanan Çözümler

USB bellek listesi getirilirken disk kapasitesini direk olarak bulan bir kod yapısı bulamadık. Bunun için Linux dosya sisteminde subprocess kütüphanesi yardımıyla ilgili sürücüyü listeleyerek, disk kapasitesini bu listeden aldık.

## Proje Bileşenleri ve Görevleri

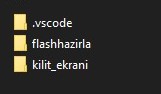
E-nahtar programı ile Milli Eğitim Bakanlığına bağlı okullarda bulunan akıllı tahtalarla uyumlu çalışabilecek şekilde tasarlandı. Akıllı tahtalarda PARDUS işletim sistemi yüklü olduğu için bu işletim sistemine özgün olarak yazıldı.

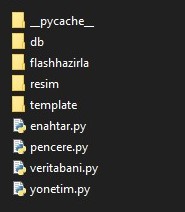
E-nahtar uygulaması ile öğrencilerin akıllı tahtaların eğitim dışı kullanımını engellemek adına hazırlanmıştır. Akıllı tahtalar hazırlanan USB bellek ile açılacak ve USB bellek çıkarıldığında sistem kendini kilitleyip öğrencilerin müdahaleleri engellenmiş olacaktır.

E-nahtar uygulaması şu aşamalardan oluşmaktadır;

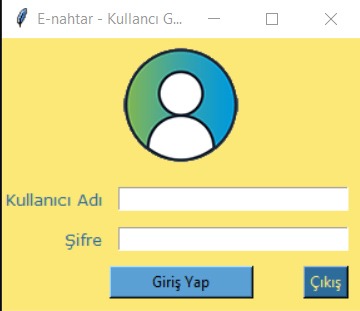
1. Anabilgisayara yüklenecek yönetim paneli ile kullanıcılar tanımlanır,
2. Oluşturulan kullanıcıların flash belleklerine E-anahtar programı yüklenir,
3. Akıllı tahtalara E-nahtar programı yüklenir.

## 1-KULLANICI OLUŞTURMA:

**Flashhazirla** klasörüne tıklanır.



**Enahtar. py** dosyası çalıştırılır.

Program yandaki şekilde görüldüğü gibi açılacaktır.

Program ilk kullanım için;

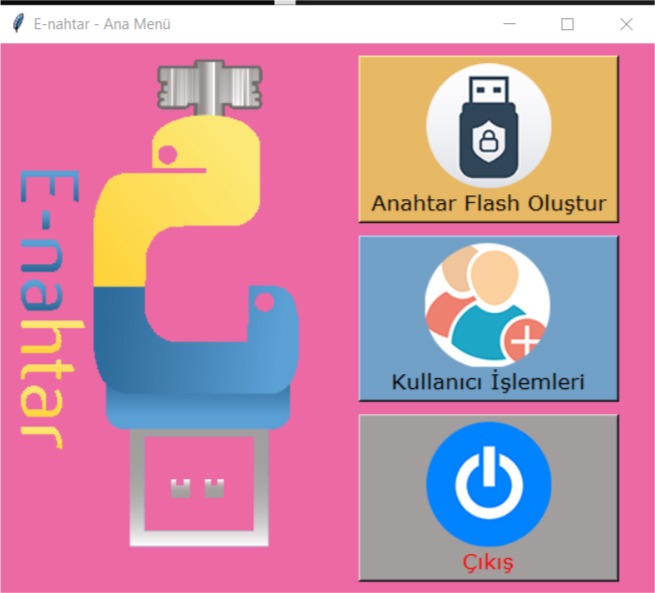
**Kullanıcı Adı : admin**

**Şifre : admin**

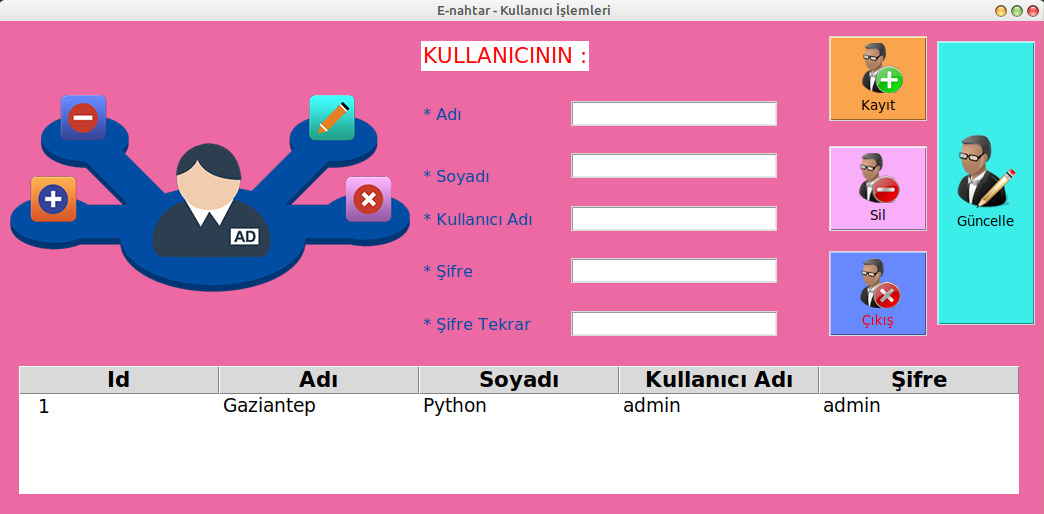
olarak belirlenmiştir.

**Giriş Yap** butonuna tıklanır.

E-nahtar **Ana Menü** penceresi açılır.

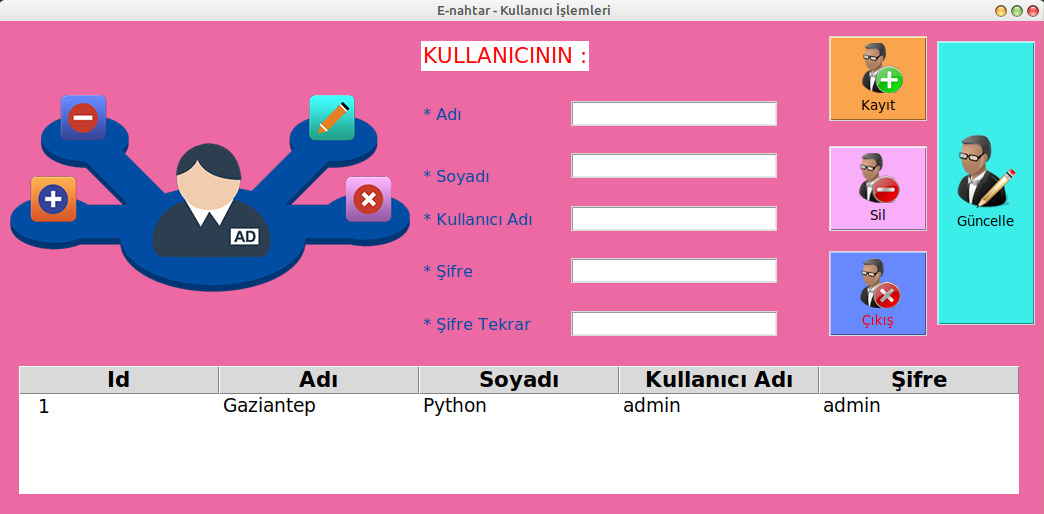


Bu pencerede ***kullanıcı işlemleri*** butonu ile **kullanıcı ekleme, silme, güncelleme** işlemleri yapılır. **Kullanıcı işlemleri** butonuna tıklanır.



**Kullanıcı Ekleme:**

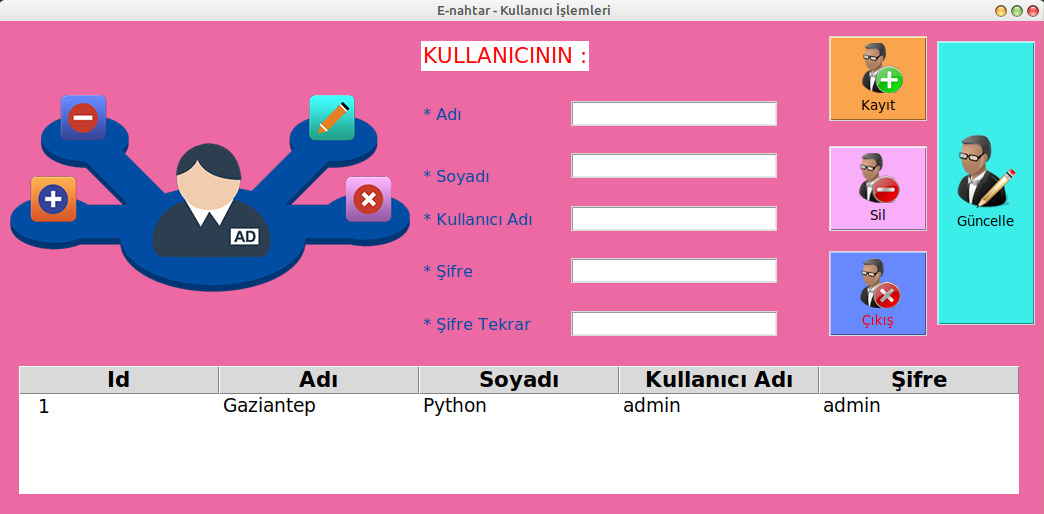
Bu pencerede kullanıcının adı, soyadı, kullanıcı adı ve şifresi belirlenir ve **Kayıt butonuna** tıklanır. Kullanıcı başarıyla eklenmiştir mesajından sonra kullanıcı bilgileri alt bölümde görüntülenir.



**Kullanıcı Silme:**

Var olan bir kullanıcıyı silmek için alt bölümden kullanıcı seçilerek **Sil butonuna** tıklanır.

**Not:** İlk kayıt yönetici bilgisidir. Bu kayıt silinemez.

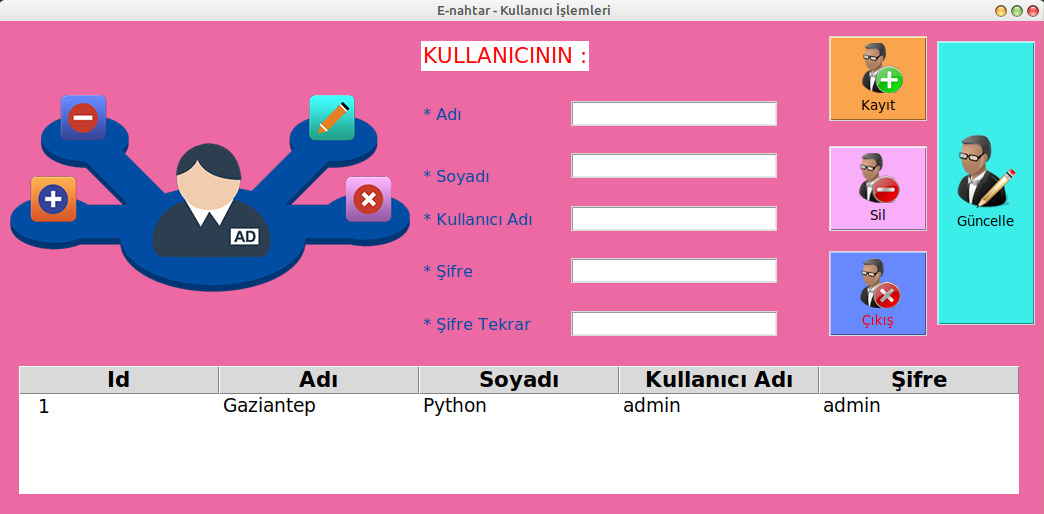


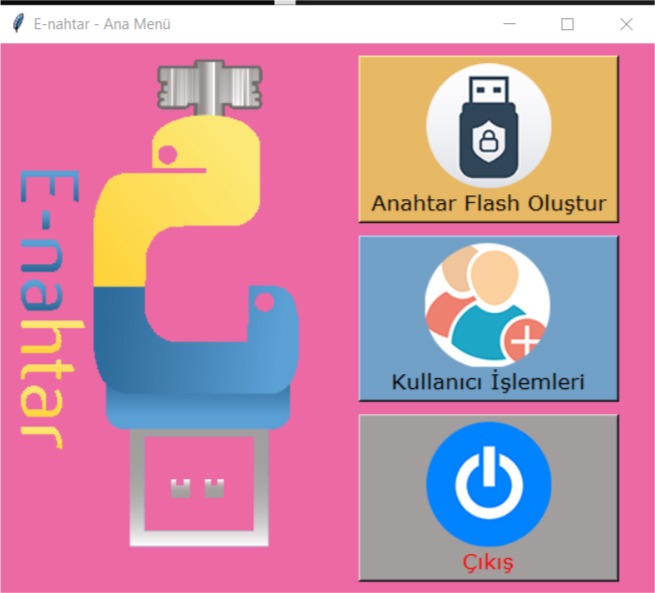
**Kullanıcı Güncelleme:**

Var olan bir kullanıcıya ait bilgilerin güncellenmesi gerektiğinde alt bölümden kullanıcı seçilerek gerekli düzenleme yapılıktan sonra Güncelle **butonuna** tıklanır.

**Çıkış İşlemleri:**

Kullanıcı işlemleri tamamlandıktan sonra **Çıkış butonu** ile E-nahtar Ana Menü ekranına dönülür.





## 2. E-NAHTAR OLUŞTURMA:

***Anahtar Flash Oluştur*** butonu ile kullanıcıların flash belleklerine E-nahtar uygulaması yüklenerek kullanıma hazır hale getirilir.

**Anahtar Flash Oluştur** butonuna tıklanır.



Açılan bu pencerede ilk olarak Flash bellek açılır kutusundan, E-nahtar uygulaması yüklenecek flash bellek seçilir.

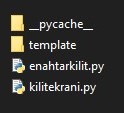
Eğer flash bellek görünmüyor ise **yenile butonuna** tıklanır.

**Anahtar Kod** metin kutusuna bir kod girilir.

**Flash Bellek Hazırla** butonuna tıklanarak E-nahtar uygulaması flash belleğe yüklenir.

## **WhatsApp Image 2020-12-10 at 22.18.45**3. Akıllı tahtalara E-nahtar uygulamasının yüklenmesi:

Klasör içerisinden **kilit\_ekrani** klasörü seçilir.



**enahtarkilit.py** dosyası çalıştırılarak E-nahtar uygulaması akıllı tahtada devreye girer.

E-nahtar uygulaması akıllı tahtalara yüklendikten sonra akıllı tahtalar aşağıdaki açılış ekranı ile açılacaktır.

Kullanıcı bilgileri tanımlanan ve E-nahtar uygulaması yüklenen flash bellekler ile akıllı tahtalar kullanıma açılacaktır.

Programa ait dökümantasyonun taslak halininde ortaya çıktığı bölümdür.

## Github Yükleme Süreci

Yazılım kaynak kodlarının github profiline yükleme işlemini tek bir kullanıcı üzerinden gerçekleştirdik.

# E-nahtar Test Raporu

## Karşılaşılan Sorunlar ve Uygulanan Çözümler

1. Proje planlaması aşamasında;

Yılmaz Yalçın hocamızın WhatsAapp grubu kurması ve gruba arkadaşlarımızı eklemesi ile Gaziantep pyton grubu oluştu. Netacad deki grup sayfası, WhatsApp programı ve Zoom programı üzerinden yapılan çeşitli toplantılar neticesinde hızlıca grup adı, proje türü, zamanlama görev dağılımı gibi konularda çalışmaya başladık.

1. Zaman Planlaması ile ilgili;

Projenin planlanan sürelerde yetiştirilmesi için grup üyeleri oldukça özverili olarak çalıştı. Zaman yönetimi ile ilgili bir sorun ile karşılaşmadık.

1. Kodlama aşamasında;

Python kodlarının çalışması sırasında Linux tabanlı sistemlerde ve Windows tabanlı sistemlerde dosya yolu belirteci olan “/” (bölü)ve “\”(ters bölü) ile ilgili sorun yaşadık. Kaynak kodda ilgili alanlar değiştirilerek sorun giderildi. Daha sonra program geliştirilirken çalışılan işletim sistemi türü ve versiyonu bilgisi alınarak programın ona göre davranması sağlanabilir.

Usb bellek listesi getirilirken disk kapasitesini direk olarak bulan bir kod yapısı bulamadık. Bunun için linux dosya sisteminde subprocess kütüphanesi yardımıyla ilgili sürücüyü listeleyerek, disk kapasitesini bu listeden aldık.

1. Test altyapısı ile ilgili sorunlar;

Projenin testi için programın Fatih projesi kapsamında teslim edilen farklı modellerdeki ve farklı işletim sistemlerine sahip akıllı tahtalarda detaylı olarak test edilmesi gerekmektedir. Bu aşamayı Pandemi dolayısı ile Uzaktan eğitim modeli uyguladığımız için zamanında yerine getiremedik.

Program sanal pardus 19.4 sistemi ve Linux dağıtımlarında test edildi. Mevcut haliyle düzgün olarak çalışmaktadır.

## Test Sürecinde Kullanılan Modüller (Varsa)

Python unittest modülü, assert fonksiyonları program kodlanırken kullanıldı. Program içinde try except yapıları olası hatalarda çökmeyi engellemek için kullanıldı. Örnek olarak flash bellek takılı olup olmadığının kontrolü ;

def anahtarTakili(self, anahtarkodu, device):

try:

usbBellek = getUsbDetay(device.device\_node)

dosyaismi = ".enahtar.dat"

dosyaTamKonumu = usbBellek.mountpoint + "/" + dosyaismi

# ~ print(dosyaTamKonumu)

dosya = open(dosyaTamKonumu, "r")

anahtar = dosya.read()

dosya.close()

durum = hashlib.sha224(

anahtarkodu.encode("utf-8") + device.get('ID\_FS\_UUID').encode("utf-8")).hexdigest() == anahtar

return durum

except Exception as ex:

print(ex)

return False

## Değerlendirme Kriterleri

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KRİTERLER** | Ekip Üyeleri | Eğitmen 1 | Eğitmen 2 | Eğitmen 3 | Ortalama |
| 1. Analiz Raporunun Tamamlanması | Atilla SÜRER  Yılmaz YALÇIN |  |  |  |  |
| 1. Tasarım Raporunun Tamamlanması | Yılmaz YALÇIN  Beyazıt ALAN |  |  |  |  |
| 1. Gerçekleştirim Raporunun Tamamlanması | Atilla SÜRER  Beyazıt ALAN |  |  |  |  |
| 1. Gantt Diagramı | Atilla SÜRER |  |  |  |  |
| 1. Arayüz tasarımı | Şefik YALMAN  Yılmaz YALÇIN |  |  |  |  |
| 1. Veri Tasarımı-Sınıf Tasarımı | Şefik YALMAN |  |  |  |  |
| 1. Kullanıcı Yardım Dökümanı | Murat CİNYOL  Muharrem TAŞÇI |  |  |  |  |
| 1. Programın Çalıştırılması | Şefik YALMAN |  |  |  |  |
| 1. Yazılım Test Çalışması | Gültekin ÖZÇELİK  Muharrem TAŞÇI  Yusuf Zafer GÜLERARSLAN |  |  |  |  |