



**İSTANBUL ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

BİTİRME PROJESİ 2

YOKLAMA YÖNETİMİ OTOMASYONU

Hazırlayanlar : Anıl Ertürk 1358130030

Danışman : Dr.Öğr.Üyesi MUSTAFA DAĞTEKİN

MAYIS - 2019

ÖNSÖZ

Bilgisayarlar günümüzde birçok işin otomasyonunda kullanılıyorlar. Çok hızlı işlem yapabilmeleri, veritabanı ve internet gibi katmanlara bağlı olmaları, insanların hayatını çok kolaylaştıran işler yapmalarını sağlıyor. Eskiden insanların yaptığı monoton işleri daha hızlı ve daha hatasız yapıyorlar. Bu da insanların daha yaratıcı işlerle, daha sevdiği işlerle uğraşmasına yardımcı oluyor. Bilgisayarların birçok kişiyi işsiz bıraktığını savunan [2] gruplarda mevcut. Fakat tarih boyunca teknoloji ilerledikçe insanlar başka işler yapmaya başladı. Ben de insanların kendilerini geliştirip günün şartlarına uyum sağlaması gerektiğini düşünenlerdenim.

İşsizlik tartışması dışında yapay zekanın insanları yok edeceği gibi başka tartışmalarda mevcut. Bazı gruplar yapay zekanın, aynı nükleer silahlar gibi, “great filter” [3] lardan bir tanesi olduğunu düşünüyor.

Modern bilgisayarlar çok amaçlı bilgisayarlar olduğu için istenilen iş için konfigüre edilebiliyorlar. Bu da her işe özel otomasyon programı yazmamıza yardımcı oluyor. Otomasyon birçok sektörde birçok işi yapabilecek bir teknoloji olduğu içinde modern bilgisayarlar ve taşınabilir Java gibi programlama dilleri kullanarak işe özel otomasyon çözümleri sağlayabiliyoruz.

Anıl Ertürk 1358130030

İÇİNDEKİLER

ÖNSÖZ	II
İÇİNDEKİLER	III
ŞEKİL LİSTESİ.....	IVII
TABLO LİSTESİ.....	IVI
KISALTMA LİSTESİ	VIII
ÖZET	IX
SUMMARY	VXI
1. GİRİŞ.....	1
2. GENEL KISIMLAR.....	1
2.1. VERİ TABANI.....	1
2.1.1. class_list	1
2.1.2. calculus1	2
2.1.3.discrete_mathematics	3
2.1.4. introduction_to_programming.....	4
2.1.5.linear_algebra.....	5
2.1.6.physics1	6
2.2. PROGRAMLAR.....	7
2.2.1 Programların Mimarisi	7
2.2.2 Menu Sınıfı.....	8
2.2.3 Camera Sınıfı.....	18
2.2.4 OCR sınıfı	19
2.2.5 Database Sınıfı	25
3. KULLANILAN ARAÇ VE YÖNTEM	31
3.1. SİMÜLASYON	31
3.1.1. Kullanılan Programlama Dilleri ve Platformlar	31
3.1.1. Programın Çalışması	32
4. TARTIŞMA VE SONUÇ	42
KAYNAKLAR	43

ŞEKİL LİSTESİ

Şekil 2.1 : class_list tablosu yapısı.....	1
Şekil 2.2 : class_list tablosu kayıtları.....	1
Şekil 2.3 : calculus1 tablosu yapısı.....	2
Şekil 2.4 : calculus1 tablosu kayıtları.....	2
Şekil 2.5 : discrete_mathematics tablosu yapısı.....	3
Şekil 2.6 : discrete_mathematics tablosu kayıtları.....	3
Şekil 2.7 : introduction_to_programming tablosu yapısı.....	4
Şekil 2.8 : introduction_to_programming tablosu kayıtları.....	4
Şekil 2.9 : linear_algebra tablosu yapısı.....	5
Şekil 2.10 : linear_algebra tablosu kayıtları.....	5
Şekil 2.11 : physics1 tablosu yapısı.....	6
Şekil 2.12 : physics1 tablosu kayıtları.....	6
Şekil 2.13 : Sınıf diyagramı.....	7
Şekil 2.14 : User Sekmesi.....	8
Şekil 2.15 : Admin Sekmesi.....	9
Şekil 2.16 : Output Sekmesi.....	10
Şekil 2.17 : Menu import	10
Şekil 2.18 : Menu, sınıf ve nesne değişkenleri	10
Şekil 2.19 : Menu main	11
Şekil 2.20 : Menu constructor and initComponents	11
Şekil 2.21 : Menu photoAndOcrActionPerformed	12
Şekil 2.22 : Menu submit	13

Şekil 2.23 : Menu manualSubmitActionPerformed	14
Şekil 2.24 : Menu studentNumberActionPerformed	15
Şekil 2.25 : Menu passwordActionPerformed.....	15
Şekil 2.26 : Menu password1ActionPerformed.....	16
Şekil 2.27 : Menu setActionPerformed	16
Şekil 2.28 : Menu weeksActionPerformed	17
Şekil 2.29 : Menu textActionPerformed	17
Şekil 2.30 : Menu excelActionPerformed	17
Şekil 2.31 : Menu main	17
Şekil 2.32 : Camera import	18
Şekil 2.33 : Camera, sınıf ve nesne değişkenleri.....	18
Şekil 2.34 : Camera run	18
Şekil 2.35 : Camera takeAPhoto	19
Şekil 2.36 : OCR import	19
Şekil 2.37 : OCR, sınıf ve nesne değişkenleri	20
Şekil 2.38 : OCR analyze method	21
Şekil 2.39 : Database import	25
Şekil 2.40 : Database, sınıf ve nesne değişkenleri	25
Şekil 2.41 : Database operation	26
Şekil 2.42 : Database isRegistered	27
Şekil 2.43 : Database toText	28
Şekil 2.44 : Database toExcel	29
Şekil 2.45 : Database getClassList	30
Şekil 3.1 : Control Panel penceresi(User sekmesi) ve Web Cam penceresi	32
Şekil 3.2 : Control Panel penceresi(Admin sekmesi) ve Web Cam penceresi	33
Şekil 3.3 : Control Panel penceresi(Output sekmesi) ve Web Cam penceresi	33
Şekil 3.4 : Take photo & OCR	34
Şekil 3.5 : Hafta ve Sınıf seçimi	34

Şekil 3.6 : Set tuşuna bastıktan sonra	35
Şekil 3.7 : Yanlış şifre sonucu	35
Şekil 3.8 : Doğru şifre sonucu	36
Şekil 3.9 : Başarısız OCR sonucu	36
Şekil 3.10 : Başarılı OCR, kayıtsız Numara sonucu	37
Şekil 3.11 : Başarılı OCR, kayıtsız Numara onay sonucu	37
Şekil 3.12 : Başarılı OCR, kayıtlı Numara sonucu	38
Şekil 3.13 : Kayıtlı Numara manuel giriş	38
Şekil 3.14 : Kayıtlı Numara manuel giriş sonucu	38
Şekil 3.15 : Kayıtsız Numara manuel giriş	39
Şekil 3.16 : Kayıtsız Numara manuel giriş sonucu	39
Şekil 3.17 : Export to Text sonucu	40
Şekil 3.18 : Export to Excel sonucu	41

TABLO LİSTESİ

Tablo 3.1 : Projede kullanılan paketler ve onların içlerindeki sınıflar	31
Tablo 3.2 : Projede kullanılan paketler ve onların içlerindeki sınıflar	31
Tablo 3.3 : Projede kullanılan paketler ve onların içlerindeki sınıflar	32

KISALTMA LİSTESİ

GUI	: Graphic User Interface
OCR	: Optical Character Recognition
SE	: Standart Edition
IDE	: Integrated Development Environment

ÖZET

YOKLAMA YÖNETİMİ OTOMASYONU

Otomasyon birçok sektörde birçok işte kullanılan bir teknolojidir. Görüntü işleme de birçok sektörde birçok işte kullanılan bir teknolojidir. Bu projede görüntü işlemenin kullanılacağı bir alan seçilip, otomasyon yapılmıştır. Seçilen kullanım alanı yoklama almadır. Normalde ders boyunca bir imza kağıdı sınıfta dolaşır ve öğrenciler öğrenci sayısı, öğrenci adı soyadı, öğrenci numarası ve imza atarlar. Daha sonra öğretim görevlisi bu kağıttaki bilgileri el ile dijital ortama girer.

Bu proje, bu sürecin her noktasında otomasyon kullanır. İlk nokta, öğrencilerin bilgileri kağıda yazması. Genellikle bu kağıtta dört sütun vardır. Öğrenci sayısı (1, 2, 3 ,4, ...), ad soyad, öğrenci numarası ve imza. Birinci noktanın otomasyonu görüntü işleme ile yapılmıştır. Öğrenciler ders boyu bu bilgileri kağıda yazmak yerine derse girişte öğrenci kartlarını kameraya gösterirler. Program kameraya fotoğraf çekmesini söyler. Program fotoğrafta geçerli bir öğrenci kartı olduğunu teyit eder ve öğrencinin derse geliş bilgisini veritabanına kayıt eder.

İkinci nokta, öğretim görevlisinin bu bilgileri manuel olarak dijital ortama girmesi. İkinci noktanın otomasyonu görüntü işleme ile aldığımız veriler ile yapılır. Veriler otomatik olarak dijital ortama girilir. Veritabanındaki verileri öğretim görevlisi istediği zaman yazılımın grafik arayüzünü kullanarak bir metin dosyasına veya bir Microsoft Excel dosyasına dökülebilir. Metin dosyası seçeneği, ders listesi tablosundaki her ders için bir metin dosyası oluşturur. Her metin dosyasında iki sütun vardır. Birinci sütunda öğrenci numarası, ikinci sütunda ise öğrencinin bu derse geldiği toplam gün sayısı vardır. Microsoft Excel dosyası seçeneği, her ders için bir Microsoft Excel dosyası oluşturur. Her Microsoft Excel dosyasında 13 sütun vardır. Birinci sütunda öğrenci numarası, kalan sütunlarda ise haftalar vardır(hafta1, hafta2, hafta3,...).

SUMMARY

ATTENDANCE MANAGEMENT AUTOMATION

Automation is a technology used in many sectors. Image processing is also a technology used in many sectors. In this project a use case has been selected and automated with image processing. The selected use case is class attendance. Normally, a page is used by students to enter their information(number of students who have signed the page, full name, student id and signature). Then, lecturer manually enters this data to digital platform.

This project automates all stages of this use case. First stage is students entering their information to a page manually. Usually this page has four columns. Number of students (1, 2, 3, 4, ...), full name, student id, signature. The automation of first stage is done with image processing. Students just show their student id card to the camera when they walk in instead of manually entering their information to a page manually. Program tells camera to take a photo. Program makes sure the photo includes a valid student card and saves the attendance data.

Second stage is lecturer entering this data to digital platform manually. Automation of second stage is done with data we got from image processing. Data is entered to digital platform automatically. Lecturer can use the GUI of the program to export the data in database as text or Microsoft Excel files. Export to Text option creates a text file for all classes in class list table. All text files have two columns. First column is student number. Second column is total attendance so far. Export to Excel option creates a Microsoft Excel file for all classes in class list table. All Microsoft Excel files have 13 columns. First column is student number. Rest of the columns are week numbers(week1, week2, week3,...).

1. GİRİŞ

Bilgisayarlar günümüzde her sektörün her işin içine girdi. Her sektörde, her işte bilgisayarlar yapılan işi çeşitli şekillerde kolaylaştırıyor. Bu teknolojik otomasyonun bir parçasıda görüntü işlemedir.

Yapılan işlerin hızlandırılması, bir insana ihtiyaç duyulmaması, bir insanın sahip olduğu zayıflıklardan kurtulmak, işin daha hatasız yapılması, otomasyonun amaçlarındandır.

Bu çalışmada bilgisayara bağlı kamera kullanılarak öğrenci kartları okunur. Öğrenci kartının görüntüsündeki text elde edilir. Bu text'ten istenilen veriler alınır ve gerekli dersin gerekli haftasına geldi bilgisi işlenir. Öğretim görevlisi istediği zaman grafik arayüzündeki butonları kullanarak veritabanını bir text dosyası veya Microsoft Excel dosyası olarak çıkartabilir. Amaç, yoklama işini kolaylaştırmaktır. El ile bu bilgilerin bir kağıda girilmesi ve daha sonra öğretim görevlisi tarafından el ile bu kağıttaki bilgilerin tek tek dijital ortama geçirilmesi yerine bütün bu işleri otomatik hale getiriyoruz. Daha detaylı anlatmak gerekirse, Java dili ile yazdığımız yazılımın da kullanıcı için kullanımının kolay olması için grafik arayüz kullandık. Üç sekmeye sahip grafik arayüzümüzün birinci sekmesi yetkili olmayan kullanıcıların kullanacağı sekmedir. Bu sekme üzerinden kullanıcılar kartlarını bilgisayara okutup otomatik olarak bilgilerini veri tabanına işleyebilirler. İkinci sekme, yetkili kişinin manuel öğrenci numarası girmesi için bir bölgeye sahiptir. Bunun dışında bir de hafta ve ders seçiminin yapılacağı bölgeye sahiptir. Üçüncü sekmeden yetkili kişi veritabanını metin veya Microsoft Excel dosyasına dökebilir. Görüntü işleme ve bilgisayarların bize sunduğu diğer hizmetleri kullanarak uzun süre alan işlerin bilgisayar tarafından otomatik olarak yapılması amaçlanmıştır.

OCR için Tesseract kütüphanesi kullanıldı. Tesseract, çeşitli işletim sistemleri için geliştirilen özgür bir optik karakter tanıma motorudur. 1985 ve 1995 yılları arasında Hewlett-Packard tarafından kapalı kaynak bir yazılım olarak geliştirilen Tesseract, sonraki on yıl içerisinde çok az gelişime uğradı. 2005 yılında Hewlett-Packard ve UNLV tarafından açık kaynak olarak piyasaya sürüldü. Tesseract'ın gelişimi şu anda Google tarafından sponsor edilmektedir. Apache Lisansı Sürüm 2.0 altında yayınlanmaktadır. Tesseract, piyasada mevcut en doğru özgür yazılım OCR motorlarından biri olarak kabul edilir.

Genel kısımlarda görüntü işleminin dezavantaj ve avantajları incelenmiştir. Otomasyon yazılımımız için geliştirilen veritabanı tabloları, sınıflar, metotlar ve algoritmalar ayrıntılı bir şekilde anlatılmıştır.

Kullanılan Araç ve Yöntem bölümünde yazılımın ve veritabanının geliştirilmesi için kullanılan platformlar, programlar, yazılım kütüphaneleri, veri tabanı yönetim sistemleri anlatılmıştır. Program çalışırken alınmış ekran görüntüleri ile programın çalışması gösterilmiştir.

Tartışma ve sonuç bölümünde yazılımın iyi ve kötü olduğu noktalar, kötü olduğu noktaların nasıl düzeltildiği incelenmiştir.

2. GENEL KISIMLAR

2.1. VERİ TABANI

2.1.1. class_list

Yoklama alırken bu yazılımın kullanılacağı derslerin listesi bu tabloda tutulur. Tek bir sütuna sahiptir. İlk sütunun adı class_name'dir. Veri tipi varchar(30)'dur.

#	Name	Type	Collation	Attributes	Null	Default
1	class_name	varchar(30)			No	None

Şekil 2.1 : class_list tablosu yapısı

	class_name
<input type="checkbox"/> Edit Copy Delete	calculus1
<input type="checkbox"/> Edit Copy Delete	discrete_mathematics
<input type="checkbox"/> Edit Copy Delete	introduction_to_programming
<input type="checkbox"/> Edit Copy Delete	linear_algebra
<input type="checkbox"/> Edit Copy Delete	physics1

Şekil 2.2 : class_list tablosu kayıtları

Yazılımın test edilebilmesi ve çalıştığının gösterilebilmesi için örnek ders tabloları oluşturuldu. Bu tablolar örnek öğrenci listeleriyle dolduruldu. class_list tablosu dışındaki tabloların tamamı örnek ders tablolarıdır. Hepsinin yapısı aynıdır. Hepsi 13 sütuna sahiptir. İlk sütunun adı öğrenci numarasıdır. Veri tipi int(10)'dur. Kalan 12 sütunların adları sırasıyla week1, week2, week3, week4, week5, week6, week7, week8, week9, week10, week11, week12'dir. 12 sütununda veri tipi tinyint(1)'dir.

2.1.2. calculus1

	#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	studentID	int(10)			No	None	
<input type="checkbox"/>	2	week1	tinyint(1)			No	0	
<input type="checkbox"/>	3	week2	tinyint(1)			No	0	
<input type="checkbox"/>	4	week3	tinyint(1)			No	0	
<input type="checkbox"/>	5	week4	tinyint(1)			No	0	
<input type="checkbox"/>	6	week5	tinyint(1)			No	0	
<input type="checkbox"/>	7	week6	tinyint(1)			No	0	
<input type="checkbox"/>	8	week7	tinyint(1)			No	0	
<input type="checkbox"/>	9	week8	tinyint(1)			No	0	
<input type="checkbox"/>	10	week9	tinyint(1)			No	0	
<input type="checkbox"/>	11	week10	tinyint(1)			No	0	
<input type="checkbox"/>	12	week11	tinyint(1)			No	0	
<input type="checkbox"/>	13	week12	tinyint(1)			No	0	

Şekil 2.3 : calculus1 tablosu yapısı

	studentID	week1	week2	week3	week4	week5	week6	week7	week8	week9	week10	week11	week12
<input type="checkbox"/> Edit Copy Delete	1353130030	0	1	1	0	0	0	0	0	0	0	1	0
<input type="checkbox"/> Edit Copy Delete	1353130000	1	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130001	1	0	0	0	1	0	0	1	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	1358130005	0	0	0	1	0	1	0	1	1	1	0	0
<input type="checkbox"/> Edit Copy Delete	1358130007	1	0	0	0	1	0	1	0	1	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130009	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130010	0	0	1	0	0	1	0	0	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	1358130011	1	1	1	1	1	1	1	1	1	1	1	1
<input type="checkbox"/> Edit Copy Delete	1358130014	0	1	0	1	0	0	0	0	1	1	0	1
<input type="checkbox"/> Edit Copy Delete	1358130015	1	0	1	0	1	1	0	0	1	1	0	1
<input type="checkbox"/> Edit Copy Delete	1358130019	0	0	0	1	0	0	0	1	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	1358130020	1	0	0	0	0	1	0	0	0	1	1	1
<input type="checkbox"/> Edit Copy Delete	1358130021	1	1	1	1	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130022	0	0	1	1	0	1	0	1	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	1358130024	1	1	0	1	0	0	1	0	0	0	1	0
<input type="checkbox"/> Edit Copy Delete	1358130025	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130026	0	0	0	0	0	1	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130028	0	0	1	0	0	0	0	0	1	0	1	0
<input type="checkbox"/> Edit Copy Delete	1358130029	1	0	0	0	0	1	0	1	1	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130030	1	1	1	1	0	1	0	0	0	0	0	1
<input type="checkbox"/> Edit Copy Delete	1368130030	1	0	0	0	0	0	0	0	0	0	0	0

Şekil 2.4 : calculus1 tablosu kayıtları

2.1.3. discrete_mathematics


	#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1	studentID	int(10)			No	None
<input type="checkbox"/>	2	week1	tinyint(1)			No	0
<input type="checkbox"/>	3	week2	tinyint(1)			No	0
<input type="checkbox"/>	4	week3	tinyint(1)			No	0
<input type="checkbox"/>	5	week4	tinyint(1)			No	0
<input type="checkbox"/>	6	week5	tinyint(1)			No	0
<input type="checkbox"/>	7	week6	tinyint(1)			No	0
<input type="checkbox"/>	8	week7	tinyint(1)			No	0
<input type="checkbox"/>	9	week8	tinyint(1)			No	0
<input type="checkbox"/>	10	week9	tinyint(1)			No	0
<input type="checkbox"/>	11	week10	tinyint(1)			No	0
<input type="checkbox"/>	12	week11	tinyint(1)			No	0
<input type="checkbox"/>	13	week12	tinyint(1)			No	0

Şekil 2.5 : discrete_mathematics tablosu yapısı




















































	studentID	week1	week2	week3	week4	week5	week6	week7	week8	week9	week10	week11	week12
<input type="checkbox"/> Edit 1358130001	1358130001	1	1	1	1	1	1	1	1	1	1	1	1
<input type="checkbox"/> Edit 1358130002	1358130002	1	0	0	1	0	0	0	0	0	1	0	0
<input type="checkbox"/> Edit 1358130004	1358130004	0	1	0	0	0	0	0	1	1	0	0	1
<input type="checkbox"/> Edit 1358130005	1358130005	0	0	0	0	1	0	1	0	0	0	0	0
<input type="checkbox"/> Edit 1358130007	1358130007	0	0	0	0	0	1	0	0	0	0	0	1
<input type="checkbox"/> Edit 1358130008	1358130008	0	0	0	0	0	0	0	0	0	0	0	1
<input type="checkbox"/> Edit 1358130009	1358130009	0	0	1	0	0	0	0	0	1	1	0	0
<input type="checkbox"/> Edit 1358130011	1358130011	0	1	0	0	0	1	0	0	0	0	0	1
<input type="checkbox"/> Edit 1358130012	1358130012	1	0	1	1	0	0	0	0	0	1	0	0
<input type="checkbox"/> Edit 1358130014	1358130014	0	0	0	0	0	0	0	0	0	0	1	1
<input type="checkbox"/> Edit 1358130015	1358130015	0	0	0	1	0	1	0	1	0	0	0	0
<input type="checkbox"/> Edit 1358130016	1358130016	0	0	1	0	0	0	0	0	0	0	1	0
<input type="checkbox"/> Edit 1358130017	1358130017	0	1	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit 1358130018	1358130018	0	0	1	0	0	0	1	1	0	1	1	1
<input type="checkbox"/> Edit 1358130019	1358130019	0	1	0	0	0	0	0	1	0	0	0	0
<input type="checkbox"/> Edit 1358130026	1358130026	0	0	0	0	1	0	1	0	0	1	0	1
<input type="checkbox"/> Edit 1358130027	1358130027	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit 1358130028	1358130028	1	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit 1358130030	1358130030	0	1	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Edit 1368130030	1368130030	0	0	1	0	1	0	0	0	0	0	0	0

Şekil 2.6 : discrete_mathematics tablosu kayıtları

2.1.4. introduction_to_programming

	#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1	studentID 	int(10)			No	None
<input type="checkbox"/>	2	week1	tinyint(1)			No	0
<input type="checkbox"/>	3	week2	tinyint(1)			No	0
<input type="checkbox"/>	4	week3	tinyint(1)			No	0
<input type="checkbox"/>	5	week4	tinyint(1)			No	0
<input type="checkbox"/>	6	week5	tinyint(1)			No	0
<input type="checkbox"/>	7	week6	tinyint(1)			No	0
<input type="checkbox"/>	8	week7	tinyint(1)			No	0
<input type="checkbox"/>	9	week8	tinyint(1)			No	0
<input type="checkbox"/>	10	week9	tinyint(1)			No	0
<input type="checkbox"/>	11	week10	tinyint(1)			No	0
<input type="checkbox"/>	12	week11	tinyint(1)			No	0
<input type="checkbox"/>	13	week12	tinyint(1)			No	0

Şekil 2.7 : introduction_to_programming tablosu yapısı

	studentID	week1	week2	week3	week4	week5	week6	week7	week8	week9	week10	week11	week12
<input type="checkbox"/>   	1353130030	0	1	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>   	1358130006	1	0	0	1	0	1	0	0	0	0	0	1
<input type="checkbox"/>   	1358130008	0	0	0	0	0	0	0	0	0	1	0	1
<input type="checkbox"/>   	1358130009	0	1	1	1	0	0	1	0	0	0	1	1
<input type="checkbox"/>   	1358130010	0	0	0	0	0	1	0	1	0	0	0	1
<input type="checkbox"/>   	1358130012	1	0	0	1	1	1	1	0	0	1	1	1
<input type="checkbox"/>   	1358130013	0	0	0	0	1	0	0	1	0	0	0	1
<input type="checkbox"/>   	1358130017	1	1	0	0	0	1	1	0	1	0	0	1
<input type="checkbox"/>   	1358130019	0	0	1	0	0	0	0	0	0	1	1	1
<input type="checkbox"/>   	1358130020	1	0	0	0	1	1	1	1	0	1	0	1
<input type="checkbox"/>   	1358130022	0	0	0	0	0	0	0	1	1	1	1	1
<input type="checkbox"/>   	1358130023	1	1	1	1	1	0	1	0	0	0	0	1
<input type="checkbox"/>   	1358130024	0	0	1	0	0	1	0	0	0	1	1	1
<input type="checkbox"/>   	1358130025	1	0	0	1	1	0	0	1	0	0	0	1
<input type="checkbox"/>   	1358130026	0	0	0	0	0	0	0	0	0	0	0	1
<input type="checkbox"/>   	1358130028	1	1	1	1	1	1	1	1	1	1	1	1
<input type="checkbox"/>   	1368130030	1	0	0	0	0	0	0	0	0	0	0	0

Şekil 2.8 : introduction_to_programming tablosu kayıtları

2.1.5. linear_algebra

	#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1	studentID	int(10)			No	None
<input type="checkbox"/>	2	week1	tinyint(1)			No	0
<input type="checkbox"/>	3	week2	tinyint(1)			No	0
<input type="checkbox"/>	4	week3	tinyint(1)			No	0
<input type="checkbox"/>	5	week4	tinyint(1)			No	0
<input type="checkbox"/>	6	week5	tinyint(1)			No	0
<input type="checkbox"/>	7	week6	tinyint(1)			No	0
<input type="checkbox"/>	8	week7	tinyint(1)			No	0
<input type="checkbox"/>	9	week8	tinyint(1)			No	0
<input type="checkbox"/>	10	week9	tinyint(1)			No	0
<input type="checkbox"/>	11	week10	tinyint(1)			No	0
<input type="checkbox"/>	12	week11	tinyint(1)			No	0
<input type="checkbox"/>	13	week12	tinyint(1)			No	0

Şekil 2.9 : linear_algebra tablosu yapısı

	studentID	week1	week2	week3	week4	week5	week6	week7	week8	week9	week10	week11	week12
<input type="checkbox"/> Edit Copy Delete	1353130030	1	1	1	1	1	1	1	1	1	1	1	1
<input type="checkbox"/> Edit Copy Delete	1358130001	1	1	1	0	0	0	1	0	1	0	1	0
<input type="checkbox"/> Edit Copy Delete	1358130003	1	1	1	1	0	1	1	1	0	1	0	1
<input type="checkbox"/> Edit Copy Delete	1358130006	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130008	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130009	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130012	0	0	0	0	1	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130013	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130015	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130018	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130020	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130021	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130023	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130025	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130026	0	0	0	0	0	0	1	0	0	0	0	0
<input type="checkbox"/> Edit Copy Delete	1358130027	0	0	0	0	0	0	1	0	0	0	1	0
<input type="checkbox"/> Edit Copy Delete	1358130028	0	0	0	0	0	0	1	0	0	0	1	0
<input type="checkbox"/> Edit Copy Delete	1358130030	1	1	1	1	1	1	1	1	1	1	1	1

Şekil 2.10 : linear_algebra tablosu kayıtları

2.1.6. physics1

	#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1	studentID	int(10)			No	None
<input type="checkbox"/>	2	week1	tinyint(1)			No	0
<input type="checkbox"/>	3	week2	tinyint(1)			No	0
<input type="checkbox"/>	4	week3	tinyint(1)			No	0
<input type="checkbox"/>	5	week4	tinyint(1)			No	0
<input type="checkbox"/>	6	week5	tinyint(1)			No	0
<input type="checkbox"/>	7	week6	tinyint(1)			No	0
<input type="checkbox"/>	8	week7	tinyint(1)			No	0
<input type="checkbox"/>	9	week8	tinyint(1)			No	0
<input type="checkbox"/>	10	week9	tinyint(1)			No	0
<input type="checkbox"/>	11	week10	tinyint(1)			No	0
<input type="checkbox"/>	12	week11	tinyint(1)			No	0
<input type="checkbox"/>	13	week12	tinyint(1)			No	0

Şekil 2.11 : physics1 tablosu yapısı

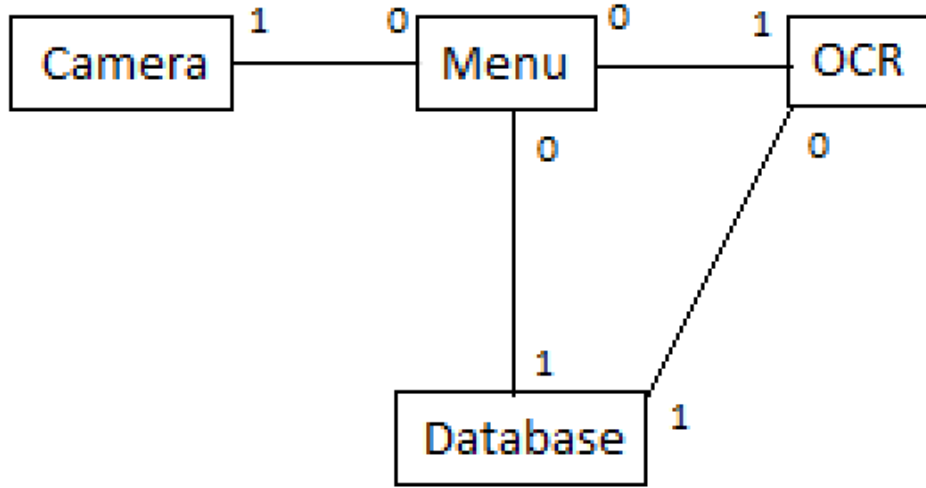
		studentID	week1	week2	week3	week4	week5	week6	week7	week8	week9	week10	week11	week12
<input type="checkbox"/>	Edit Copy Delete	1358130002	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130004	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130005	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130006	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130007	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130008	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130009	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130010	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130011	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130012	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130013	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130014	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130015	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130016	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130019	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130020	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130021	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130022	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130024	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130027	0	0	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/>	Edit Copy Delete	1358130028	0	0	0	0	0	0	0	0	0	0	0	0

Şekil 2.12 : physics1 tablosu kayıtları

2.2. PROGRAMLAR

2.2.1 Programların Mimarisi

Program dört adet sınıftan oluşmaktadır. Sınıfların isimleri Menu, Camera, OCR ve Database'dir. Menu sınıfı main metodunun olduğu ve grafik arayüzün oluşturulduğu sınıftır. Diğer üç sınıf, Menu sınıfına yardımcı sınıflardır. OCR sınıfı da bir Database nesnesine sahip fakat sadece isRegistered metodunu kullanıyor.



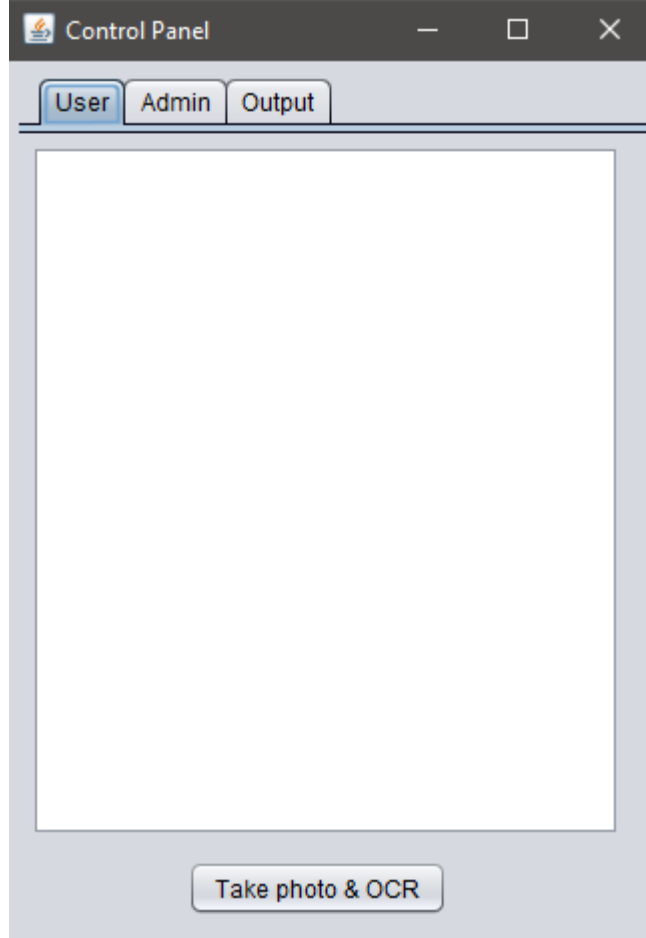
Şekil 2.13 : Sınıf diyagramı

Kötü bir dizayn olan bidirection association (çift yönlü ilişki) [4] yapısından uzak duruldu. Sınıflar ve metotlar mümkün olduğunca birbirinden bağımsız, loosely coupled (az bağımlı) [5] [6] dizayn edildi.

2.2.2 Menu sınıfı

Menu sınıfı GUI'a, main metotuna sahip sınıftır. JFrame içine yerleştirilmiş JtabbedPan'den oluşmaktadır. JTabbedPane üç adet JPanel'e sahiptir, yani üç adet sekme'ımız vardır.

Birinci sekmenin adı "User"dır ve kullanıcıya bilginin verildiği JScrollPane içindeki JTextPane'e sahiptir. Ayrıca "Take Photo & OCR" isimli butona sahiptir.



Şekil 2.14 : User sekmesi

İkinci sekmenin adı “Admin”dir. JSeparator kullanılarak ikiye ayrılmıştır. Üst kısım kartı yanında olmayan kişilerin el ile girişinin yapıldığı kısımdır. Üç adet JLabel’a, bir adet JTextField’a, bir adet JPasswordField’a ve bir adet JButton’a sahiptir. Alt kısım hafta ve ders seçiminin yapıldığı bölümdür. Dört adet JLabel’a, iki adet JComboBox’a, bir adet JPasswordField’a ve bir adet JButton’a sahiptir.

Şekil 2.15: Admin sekmesi

Üçüncü sekmenin adı “Output”tur. JSeperator kullanılarak ikiye ayrılmıştır. Üst kısım, yoklama bilgisini bir metin dosyasına yazmak için kullanılan butona sahiptir. Alt kısım, yoklama bilgisini Microsoft Excel dosyasına yazmak için kullanılan butona sahiptir.



Şekil 2.16 : Output sekmesi

Menu sınıfımız `javax.swing.JOptionPane` sınıfını import öğrenci listesinde kayıtlı olmayan bir öğrenci eklenmeye çalışırsa pop up bir ekran oluşturup kullanıcıdan onay almak için.

```
import javax.swing.JOptionPane;
```

Şekil 2.17 : Menu import

Menu sınıfı field olarak, daha önce anlatılan üç yardımcı sınıfın nesnelerini oluşturuyor (`cam`, `ocr`, `db`). Ayrıca hafta ve sınıf değerlerini hafızada tutabilmek için iki `String` değişken kullanıyor.

```
static Camera cam = new Camera();
OCR ocr = new OCR();
Database db = new Database();

String weekNo;
String className;
```

Şekil 2.18 : Menu, sınıf ve nesne değişkenleri

Menu sınıfı main metotunda yeni bir GUI thread oluştururken yeni bir Menu objesi yaratıyor. Bu da Menu kurucusunu tetikliyor. Menu kurucusu ise tek göreve sahip, o da initComponents isimli GUI kurulumunu yapan metodu çağırmak.

```
public static void main(String args[])
{
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            new Menu().setVisible(true);
        }
    });
}
```

Şekil 2.19 : Menu main

```
/** Creates new form Menu ...3 lines */
public Menu()
{
    initComponents();
}

/** This method is called from within the constructor to initia:
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents()
{...296 lines } // </editor-fold>
```

Şekil 2.20 : Menu constructor and initComponents

photoAndOcActionPerformed metodu hafta ve sınıf bilgileri seçilmiş ise bir Camera sınıfımızın objesi olan cam'in takeAPhoto metotunu çağırıyor. Bir önceki işlemde kalan studentNumber sıfırlanıyor. Çekilen fotoğraf OCR yapılması için ocr.analyze metotunu çağırılıyor ve geri dönen String değeri String str değişkenine atıyor. Eğer str değişkeninin ilk iki karakteri 13 ise menu objesinin submit metodu çağırılıyor. Değil ise str objesinin tuttuğu string olan "Try again. Make sure your student ID card is about 10 centimeters from the camera." string'i yazdırılıyor.

```

private void photoAndOcrActionPerformed(java.awt.event.ActionEvent evt)
{
    if (weekNo == null || className == null)
    {
        jTextPanel.setText(jTextPanel.getText() + "Set week and class from "
            + "Admin tab first." + "\n\n");
    }
    else
    {
        // Take a photo and save it as id.jpg.
        cam.takeAPhoto();

        // Making sure we reset the studentNumber from last OCR.
        ocr.studentNumber = 0;
        /* OCR the id.jpg. Return student number if photo is valid. Return error
        message(Try again) if it is invalid. */
        String str = ocr.analyze(className);

        // Decide whether image was valid or not. Prompt user to submit or try again.
        if ("13".equals(str.substring(0, 2)))
        {
            // Image was valid.
            submit();
        }
        else
        {
            // Image was not valid.
            jTextPanel.setText(jTextPanel.getText() + str + "\n\n");
        }
    }
}
}

```

Şekil 2.21: Menu photoAndOcrActionPerformed

Submit metodu hafta ve sınıf bilgileri seçilmiş ise ve ocr objesinin analyze metodu sırasında securityFlag2 boolean değişkeni true yapılmış ise Database sınıfının objesi olan db'nin operation metotunu çağırıyor ve sonucu ekrana yazdırıyor. Aksi halde "Gerekli hata mesajı yazdırılıyor."

```
private void submit()
{
    if (weekNo == null || className == null)
    {
        jTextPanel.setText(jTextPanel.getText() + "Set week and class from "
            + "Admin tab first." + "\n\n");
    }
    else
    {
        if (ocr.securityFlag2 == true)
        {
            db.operation(ocr.studentNumber, weekNo, className, ocr.operation);
            jTextPanel.setText(jTextPanel.getText() + "Submitted " + ocr.studentNumber
                + "\n\n");

            // Making sure we reset the studentNumber and securityFlag2 after using them.
            ocr.studentNumber = 0;
            ocr.securityFlag2 = false;
        }
        else
        {
            jTextPanel.setText(jTextPanel.getText() + "You need to take a photo with "
                + "a valid student ID card first.\n\n");
        }
    }
}
```

Şekil 2.22: Menu submit

manualSubmitActionPerformed metodu kartı yanında olmayanların bilgilerinin girişinin yapıldığı metot. studentNumber JTextField'ındaki değeri daha sonra kullanılmak üzere bir adet string değişkenine bir adet de integer değişkenine atıyor. Integer değişkene atamamız aynı zamanda girilen verinin sadece rakam içerip içermediğinin de test ediyor. hafta ve sınıf bilgileri seçilmiş ise ve girilen öğrenci numarası ve şifre gerekli testlerden geçiyorsa öğrenci numarası bu ders'e kayıtlımı değilmi bakıyoruz. Kayıtlı ise db objesinin operation metotunu insert argümanı ile çağırıyoruz. Kayıtlı değil ise db objesinin operation metotunu update argümanı ile çağırıyoruz.

```

private void manualSubmitActionPerformed(java.awt.event.ActionEvent evt)
{
    try
    {
        String studentNoStr = studentNumber.getText();
        // This also checks whether the entered String is a number or not.
        int studentNoInt = Integer.parseInt(studentNoStr);

        if (weekNo == null || className == null)
        {
            jTextPanel.setText(jTextPanel.getText() + "Set week and class from "
                + "Admin tab first." + "\n\n");
        }
        else
        {
            if ("13".equals(studentNoStr.substring(0, 2)) && studentNumber.
                getText().length() == 10 && "dagtekin13".equals(password.getText()))
            {
                if (db.isRegistered(studentNoInt, className))
                {
                    db.operation(studentNoInt, weekNo, className, "update");
                    jTextPanel.setText(jTextPanel.getText() + "Submitted "
                        + studentNoStr + "\n\n");
                }
                else
                {
                    String answer = JOptionPane.showInputDialog("StudentID "
                        + studentNumber.getText() + " is not in student list. Enter "
                        + "\"y\" to add it to student list. Enter \"n\" to take "
                        + "another photo.");
                    if ("y".equals(answer))
                    {
                        db.operation(studentNoInt, weekNo, className, "insert");
                        jTextPanel.setText(jTextPanel.getText() + "Submitted "
                            + studentNoStr + "\n\n");
                    }
                }
            }
            else
            {
                jTextPanel.setText(jTextPanel.getText() + "Student number or password "
                    + "is wrong.\n\n");
            }
        }
    }
}

```

Şekil 2.23: Menu manualSubmitActionPerformed (Part 1 of 2.)

```

        // Reset both fields and focus on studentNumber field.
        studentNumber.setText("");
        password.setText("");
        studentNumber.requestFocus();
    }
    catch (StringIndexOutOfBoundsException | NumberFormatException e)
    {
        jTextPanel.setText(jTextPanel.getText() + "Student number or password "
            + "is wrong.\n\n");

        // Reset both fields and focus on studentNumber field.
        studentNumber.setText("");
        password.setText("");
        studentNumber.requestFocus();
    }
}

```

Şekil 2.23: Menu manualSubmitActionPerformed (Part 2 of 2.)

studentNumberActionPerformed metodu, Admin sekmesinde student id'nin girildiği text field da enter'a basında yapılacakları içerir. Fare ile şifre alanına gidilmesine gerek olmaması için, öğrenci numarası yazıldıktan sonra ENTER'a basıldığında otomatik olarak şifrenin girildiği kutuya gidilmesini sağlar.

```

// Pressing enter in studentNumber textfield takes you to password textfield
private void studentNumberActionPerformed(java.awt.event.ActionEvent evt)
{
    password.requestFocus();
}

```

Şekil 2.24: Menu studentNumberActionPerformed

passwordActionPerformed metodu, Admin sekmesinde student id'nin girildiği kutunun altındaki kutuya şifre girilip ENTER'a basıldığında yapılacakları içerir. Fare ile Submit butonuna basılmasına gerek kalmaması için ENTER'a basıldığında otomatik olarak butonun metodu çağırılır.

```

// Pressing enter in password passwordfield invokes manualSubmit button.
private void passwordActionPerformed(java.awt.event.ActionEvent evt)
{
    manualSubmitActionPerformed(evt);
}

```

Şekil 2.25 Menu passwordActionPerformed

password1ActionPerformed metodu, Admin tabında hafta ve sınıf seçildikten ve şifreyi girdikten sonra ENTER'a basınca yapılacakları içerir. Fare ile Set butonuna basmaya gerek olmadan otomatik olarak Set butonunun metotunu çağırır.

```
private void password1ActionPerformed(java.awt.event.ActionEvent evt)
{
    setActionPerfomed(evt);
}
```

Şekil 2.26 Menu password1ActionPerformed

setActionPerfomed metodu, hafta ve sınıf seçildikten sonra Set butonuna basılınca yapılacakları içerir. Şifrenin doğruluğu test edildikten sonra eğer doğru ise weekNo ve className değişkenlerine atanır JComboBox'lerden alınan veriler.

```
private void setActionPerfomed(java.awt.event.ActionEvent evt)
{
    if ( "dagtekin13".equals(password1.getText()))
    {
        weekNo = weeks.getSelectedItem().toString();
        className = classes.getSelectedItem().toString();

        jTextPanel.setText(jTextPanel.getText() + "Week set to: " + weekNo +
            "\nClass set to: " + className + "\n\n");
    }
    else
    {
        jTextPanel.setText(jTextPanel.getText() + "Wrong password." + "\n\n");
    }

    password1.setText("");
    password1.requestFocus();
}
```

Şekil 2.27 Menu setActionPerfomed

weeksActionPerformed metodu, haftaların bulunduğu JComboBox'tan hafta seçince arka planda yapılacakları içerir. Sınıfları içeren JComboBox'ın uzunluğu sıfır ise db objesinin getClassList metotunu kullanarak sınıf listesini alır ve sınıf JComboBox'ının elemanlarını oluşturur. Bu da sınıf seçebilmemize yarar.

```
private void weeksActionPerformed(java.awt.event.ActionEvent evt)
{
    if (classes.getItemCount() == 0)
    {
        String[] classList = db.getClassList();

        for (String className : classList)
        {
            classes.addItem(className);
        }
    }
}
```

Şekil 2.28 Menu weeksActionPerformed

textActionPerformed metodu, Output sekmesinden veritabanını bir metin dosyasına yazmak için kullanılır. db objesinin toText metotunu çağırır.

```
private void textActionPerformed(java.awt.event.ActionEvent evt)
{
    db.toText();
}
```

Şekil 2.29 Menu textActionPerformed

excelActionPerformed metodu, Output sekmesinden veritabanını bir Microsoft Excel dosyasına yazmak için kullanılır. db objesinin toExcel metotunu çağırır.

```
private void excelActionPerformed(java.awt.event.ActionEvent evt)
{
    db.toExcel();
}
```

Şekil 2.30 Menu excelActionPerformed

Main metodunun ilk kısmının GUI ile ilgili olduğundan bahsetmiştik yukarıda. Şimdi ise ikinci kısmının ne yaptığı anlatılacak. Yeni bir iş parçacığı oluşturulup cam objesinin run metodu çağırılır.

```
// Starting video feed as soon as program starts.
new Thread(new Runnable() {
    public void run() {
        cam.run();
    }
}).start();
}
```

Şekil 2.31 Menu main

2.2.3 Camera sınıfı

Camera sınıfı kamera işlemlerini yaptığımız sınıftır. İki adet metodu vardır.

Kamera kontrolü için gerekli importlar yapılır.

```
import org.bytedeco.javacv.*;
import static org.bytedeco.javacpp.opencv_core.IplImage;
import static org.bytedeco.javacpp.opencv_imgcodecs.cvSaveImage;
```

Şekil 2.32 Camera import

Run ve takeAPhoto metodlarında ortak kullanılan nesneler oluşturulur ve değişkenlere atanır.

```
CanvasFrame canvas = new CanvasFrame("Web Cam");

// Objects used in run and takeAPhoto methods.
FrameGrabber grabber = new OpenCVFrameGrabber(0);
OpenCVFrameConverter.ToIplImage converter = new OpenCVFrameConverter.ToIplImage();
IplImage img;
```

Şekil 2.33 Camera, sınıf ve nesne değişkenleri

run metodu video feed'i sağlayan metottur. Kameradan frame alır ve bu frame'i canvas frame'de gösterir. Program çalıştığı sürece sonsuz döngüde bu metod çalışmaya devam eder.

```
public void run()
{
    try
    {
        /* When application starts, this method is called automatically from
        main method of Menu class. So grabber is started before we start taking
        photos. */
        grabber.start();
        while (true)
        {
            Frame frame = grabber.grab();

            //img = converter.convert(frame);

            canvas.showImage(frame); //converter.convert(img);

            //Thread.sleep(41);
        }
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
```

Şekil 2.34 Camera run

takeAPhoto metodu, analiz etmek istediğimiz zaman fotoğraf çekmemize yarar. Run metodu gibi frame alır ama frame'i image'ye çevirip id.jpg olarak saklar.

```
public void takeAPhoto()
{
    try
    {
        //grabber.start(); // Causes error. So i moved 3 declarations to field

        Frame frame = grabber.grab();

        img = converter.convert(frame);

        cvSaveImage("id.jpg", img);

        //canvas.showImage(converter.convert(img));

        //Thread.sleep(1000);
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
```

2.2.4 OCR sınıfı

Çekilen fotoğrafın analizini yapan sınıftır. Bir adet metota sahiptir.

Fotoğraf analizi için gerekli sınıfları import eder.

```
import org.bytedeco.javacpp.*;
import static org.bytedeco.javacpp.lept.*;
import static org.bytedeco.javacpp.tesseract.*;
import javax.swing.JOptionPane;
```

Şekil 2.36 OCR import

Sınıfta kullanılacak objeleri ve değişkenleri field'da tanımlar.

```
int studentNumber;
/* securityflag is used in this class only. It is used to see if the card we
read includes at least one of the words MUHENDISLIK, FAK, BILGISAYAR or MUH.
We do this in order to prevent students from writing their student number on
a paper and scan it to the program. Students wont know about this secret
security step of the program. */
boolean securityFlag = false;
/* We use set and reset security flag in this class but we use it in class
Menu. We use this flag when submit button is clicked in order to see if
current studentNumber variable of ocr object is a valid one. (Starts with 13,
10 characters long, securityFlag is true.
If manualSubmit button is used instead of submit button, these checks are
done in method manualSubmitActionPerformed */
boolean securityFlag2 = false; // Used to
/* This variable is used to determine whether we will create a new record or
update an existing record. */
String operation;
Database db = new Database();
```

Şekil 2.37 OCR, sınıf ve nesne değişkenleri

analyze metodu, menu objesi tarafından çağırılır. menu objesi analyze metotunu çağırırken bir className argümanı verir. analyze metodu ilk olarak OCR işlemini yapmak için gerekli tesseract [7] kütüphanesi kurulumlarını yapar. Daha sonra cam objesinin takeAPhoto metotunda sakladığımız id.jpg dosyasını açar ve içindeki metni çıkarır. OCR işlemi bitince, kullanılan kaynaklar bırakılır.


```

public String analyze(String className)
{
    BytePointer outText;

    TessBaseAPI api = new TessBaseAPI();
    // Initialize tesseract-ocr with English, without specifying tessdata path
    if (api.Init(null, "eng") != 0)
    {
        System.err.println("Could not initialize tesseract.");
        System.exit(1);
    }

    // Open input image with leptonica library
    PIX image = pixRead("D:\\Computer Engineering\\z- Other\\School, Internships"
        + "\\4. Year\\Term 2\\Finishing Project 2\\mavenproject1\\id.jpg");
    api.SetImage(image);
    // Get OCR result
    outText = api.GetUTF8Text();
    //System.out.println("OCR output:\n" + outText.getString());

    // Destroy used object and release memory
    api.End();
    outText.deallocate();
    pixDestroy(image);
}

```

Şekil 2.38 OCR analyze method (Part 1 of 5.)

Görüntüdeki metin çıkartıldıktan sonra, metin kelimelere bölünür ve `studentCardData` String dizisinde tutulur kelimeler. Enhanced for döngüsü kullanarak `studentCardData` dizisinin elemanları üzerinde işlem yapılır. Eğer kelime 10 karakterden daha kısa ise bu kelime üzerinde işlem yapılmaz. `continue` ifadesi kullanılarak sonraki kelimeye geçilir. Kelime 10 karakterden uzunsa ve `MUHENDISLIK` veya `BILGISAYAR` kelimelerini içeriyorsa `securityFlag` boolean değişkeni `true` yapılır ve `continue` ifadesi kullanılarak sonraki kelimeye geçilir. Bunun sebebi güvenlidir. Kullanıcının bilmediği bu sistem sayesinde kullanıcı sistemi aldatamayacaktır. Eğer kelime 10 karakterden uzunsa ve `securityFlag` değişkeni `true` ise bu kelime öğrenci numarası içeriyormu içermiyormu bakmaya başlarız. Kelimenin ilk iki karakterini ve ilk üç karakterini `subString1` ve `subString2` string değişkenlerini atarız. `subString1` değişkeni 13 string'ine eşitse kelimenin ilk 10 karakterini `subString3` değişkenine atarız ve `subString3` öğrenci numarası bu dersin öğrenci listesinde varmı bakarız `db` objesinin `isRegistered` metodu ile. Eğer kayıtlı ise veritabanında bu öğrenci numarasının bu tarihte bu derse geldiğini işaretlemeye hazırlık için bazı işlemler yaparız ve öğrenci numarasını `menu` objesine geri döndürürüz.

```

// Splitting text to String array elements.
String[] studentCardData = outText.getString().split(" ");

// Parsing String array elements
for (String data : studentCardData)
{
    try
    {
        /* I am not checking to see if the string is 10 characters long.
        Because student number is read from image with some extra
        characters attached to it. So it might be longer. */
        if (data.length() < 10)
        {
            continue;
        }

        // Secret security code.
        if (data.contains("MUHENDISLIK") || data.contains("BILGISAYAR"))
        {
            securityFlag = true;
            continue;
        }

        // We already did length test above.
        if (securityFlag == true)
        {
            String subString1 = data.substring(0, 2);
            String subString2 = data.substring(0, 3);

            if ("13".equals(subString1))
            {
                String subString3 = data.substring(0, 10);

                // Valid student id and registered.
                if ( db.isRegistered(Integer.parseInt(subString3), className) )
                {
                    operation = "update";

                    securityFlag2 = true; // Valid student number.

                    // Getting rid of extra characters.
                    studentNumber = Integer.parseInt(subString3);

                    securityFlag = false; // Resetting object to be used again
                    return subString3;
                }
            }
        }
    }
}

```

Şekil 2.38 OCR analyze method (Part 2 of 5.)

Eğer öğrenci numarası bu dersin öğrenci listesinde yoksa `JOptionPane.showInputDialog` kullanılarak kullanıcıya bu öğrenci numarasını bu dersin öğrenci listesine eklemek istermisiniz sorusu sorulur. Eğer evet cevabı gelirse bu öğrenci numarasını öğrenci listesine eklemek için gerekli hazırlıklar yapılır ve öğrenci numarası menu objesine geri döndürülür.

```

}
else // Valid student id but not registered.
{
    String answer = JOptionPane.showInputDialog("StudentID " +
        subString3 + " is not in student list. Enter \"y\" to add"
        + " it to student list. Enter \"n\" to take another photo.");
    if ("y".equals(answer))
    {
        operation = "insert";

        securityFlag2 = true; // Valid student number.

        // Getting rid of extra characters.
        studentNumber = Integer.parseInt(subString3);

        securityFlag = false; // Resetting object to be used again
        return subString3;
    }
    else // If user enters something other than "y".
    {
        break;
    }
}
}

```

Şekil 2.38 OCR analyze method (Part 3 of 5.)

Aynı işlem sadece 13 ile başlayan kelimeler için değil :13 ve 113 ile başlayan kelimeler içinde yapılır.

```

else if (":13".equals(subString2) || "113".equals(subString2))
{
    String subString4 = data.substring(1, 11);

    // Valid student id and registered.
    if ( !db.isRegistered(Integer.parseInt(subString4), className) )
    {
        operation = "update";

        securityFlag2 = true; // Valid student number.

        // Getting rid of extra characters.
        studentNumber = Integer.parseInt(subString4);

        securityFlag = false; // Resetting object to be used again
        return subString4;
    }
    else // Valid student id but not registered.
    {
        String answer = JOptionPane.showInputDialog("StudentID " +
            subString4 + " is not in student list. Enter \"y\" to add it "
            + "to student list. Enter \"n\" to take another photo.");
        if ("y".equals(answer))
        {
            operation = "insert";

            securityFlag2 = true; // Valid student number.

            // Getting rid of extra characters.
            studentNumber = Integer.parseInt(subString4);

            securityFlag = false; // Resetting object to be used again
            return subString4;
        }
        else // If user enters something other than "y".
        {
            break;
        }
    }
}
}
}
}

```

Şekil 2.38 OCR analyze method (Part 4 of 5.)

```

        catch (StringIndexOutOfBoundsException e)
        {
            securityFlag = false; // Resetting object to be used again
            securityFlag2 = false; // Invalid student number.
            return "Try again. Make sure your student ID card is about 10 centimeters "
                + "from the camera.";
        }
    }

    securityFlag = false; // Resetting object to be used again
    securityFlag2 = false; // Invalid student number.
    return "Try again. Make sure your student ID card is about 10 centimeters "
        + "from the camera.";
}
}

```

Şekil 2.38 OCR analyze method (Part 5 of 5.)

2.2.5 Database sınıfı

Veritabanı işlemleri için gerekli sınıflar import edilir.

```

import java.io.File;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

```

Şekil 2.39 Database import

Tüm metotlarda ortak kullanılan Connection ve Statement objeleri field'da tanımlanır.

```

Connection c;
Statement stmt;

```

Şekil 2.40 Database, sınıf ve nesne değişkenleri

Operation metodu, eğer operation parametresi update ise, var olan kayıdı günceller. Operation parametresi insert ise yeni bir kayıt oluşturur.

```
public void operation(int studentID, String weekNo, String className, String
    operation)
{
    Connection c;
    Statement stmt;

    try {
        Class.forName("com.mysql.jdbc.Driver");
        c = DriverManager.getConnection("jdbc:mysql://localhost:3306/test",
            "root", "");
        //System.out.println("Opened database successfully");

        stmt = c.createStatement();
        if ("update".equals(operation))
        {
            stmt.executeUpdate("UPDATE " + className + " SET " + weekNo + " = 1 "
                + "WHERE studentID = " + studentID);
        }
        else if ("insert".equals(operation))
        {
            stmt.executeUpdate("INSERT INTO " + className + " (studentID , " +
                weekNo + ") " + " VALUES ('" + studentID + "', '" + 1 + "')");
        }

        stmt.close();
        c.close();
        //System.out.println("Operation done successfully");
    }
    catch (Exception e) {
        System.err.println(e.getClass().getName() + ": " + e.getMessage());
        //System.out.println("Operation failed.");
    }
}
```

Şekil 2.41 Database operation

isRegistered metodu, parametre olarak alınan studentID bu dersin öğrenci listesinde varmı yokmu sonucunu boolean'a olarak döndürür.

```
public boolean isRegistered(int studentID, String className)
{
    Connection c;
    Statement stmt;

    try {
        Class.forName("com.mysql.jdbc.Driver");
        c = DriverManager.getConnection("jdbc:mysql://localhost:3306/test",
            "root", "");
        //System.out.println("Opened database successfully");

        stmt = c.createStatement();
        ResultSet rs = stmt.executeQuery( "SELECT * FROM " + className + ";" );

        while ( rs.next() ) {
            String currentStudentID = rs.getString("studentID");
            if (studentID == Integer.parseInt(currentStudentID))
            {
                rs.close();
                stmt.close();
                c.close();
                return true;
            }
        }

        rs.close();
        stmt.close();
        c.close();
        return false;
    }
    catch ( Exception e ) {
        System.err.println( e.getClass().getName()+" : "+ e.getMessage() );
        //System.out.println("Operation failed.");
    }
    return false;
}
```

Şekil 2.42 Database isRegistered

toText metodu, classList tablosundaki tüm kayıtlar için bir metin dosyası oluşturur. Her metin dosyası iki sütun içerir. Birinci sütunda öğrenci numaraları vardır. İkinci sütunda bu öğrencinin bu güne kadar geldiği toplam ders sayısı vardır.

```

public void toText()
{
    Connection c;
    Statement stmt;

    try {
        Class.forName("com.mysql.jdbc.Driver");
        c = DriverManager.getConnection("jdbc:mysql://localhost:3306/test",
            "root", "");
        //System.out.println("Opened database successfully");

        // Getting class list.
        String[] classList = getClassList();

        stmt = c.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM " + classList[0] + "");

        for (byte counter = 0; counter < classList.length; counter++)
        {
            //stmt = c.createStatement();
            rs = stmt.executeQuery("SELECT * FROM " + classList[counter] + "");
            File file = new File("D:\\\" + classList[counter] + ".txt");
            PrintWriter writer = new PrintWriter(file);

            while (rs.next())
            {
                String studentID = rs.getString("studentID");
                int participationCounter = Integer.parseInt(rs.getString("week1"))
                    + Integer.parseInt(rs.getString("week2"))
                    + Integer.parseInt(rs.getString("week3"))
                    + Integer.parseInt(rs.getString("week4"))
                    + Integer.parseInt(rs.getString("week5"))
                    + Integer.parseInt(rs.getString("week6"))
                    + Integer.parseInt(rs.getString("week7"))
                    + Integer.parseInt(rs.getString("week8"))
                    + Integer.parseInt(rs.getString("week9"))
                    + Integer.parseInt(rs.getString("week10"))
                    + Integer.parseInt(rs.getString("week11"))
                    + Integer.parseInt(rs.getString("week12"));
                writer.printf("%s: %2d\n", studentID, participationCounter);
            }

            writer.close();
        }

        rs.close();
        stmt.close();
        c.close();
    }
    catch (Exception e) {
        System.err.println( e.getClass().getName()+" : "+ e.getMessage() );
        //System.out.println("Operation failed.");
    }
}

```

Şekil 2.43 Database toText

toExcel metodu, classList tablosundaki her kayıt için bir Microsoft Excel dosyası oluşturur. Her Microsoft Excel dosyasında 13 sütun vardır. Birinci sütunda öğrenci numaraları, kalan sütunlarda o hafta gelip gelmediği (0, 1) bilgisi vardır.

```
public void toExcel()
{
    Connection c;
    Statement stmt;

    try {
        Class.forName("com.mysql.jdbc.Driver");
        c = DriverManager.getConnection("jdbc:mysql://localhost:3306/test",
            "root", "");
        //System.out.println("Opened database successfully");

        // Getting class list.
        String[] classList = getClassList();

        stmt = c.createStatement();

        for (byte counter = 0; counter < classList.length; counter++)
        {
            String filename = "D:/ " + classList[counter] + ".xls";
            String tablename = classList[counter];
            //stmt = c.createStatement();
            stmt.executeQuery("SELECT * INTO OUTFILE \" " + filename + "\" "
                + "FROM " + tablename);
        }

        stmt.close();
        c.close();
    }
    catch (Exception e) {
        System.err.println( e.getClass().getName()+" : "+ e.getMessage() );
        //System.out.println("Operation failed.");
    }
}
```

Şekil 2.44 Database toExcel

`getClassList` metodu, `class_list` tablosunda kaç kayıt olduğunu sayar. Bu uzunlukta bir String dizisi oluşturur ve sınıf isimlerini tek tek eleman olarak bu diziye atar ve son olarak bu diziye döndürür.

```
public String[] getClassList()
{
    Connection c;
    Statement stmt;

    try {
        Class.forName("com.mysql.jdbc.Driver");
        c = DriverManager.getConnection("jdbc:mysql://localhost:3306/test",
            "root", "");
        //System.out.println("Opened database successfully");

        stmt = c.createStatement();
        ResultSet rs = stmt.executeQuery( "SELECT * FROM class_list;" );

        byte length = 0;
        while (rs.next() && rs.getString("class_name") != null) {
            length++;
        }

        rs = stmt.executeQuery( "SELECT * FROM class_list;" );

        byte counter = 0;
        String [] classList = new String[length];
        while (rs.next()) {
            String className = rs.getString("class_name");
            classList[counter] = className;
            counter++;
        }

        rs.close();
        stmt.close();
        c.close();
        return classList;
    }
    catch ( Exception e ) {
        System.err.println( e.getClass().getName()+" : "+ e.getMessage() );
        //System.out.println("Operation failed.");
    }
    return null;
}
```

Şekil 2.45 Database `getClassList`

3. KULLANILAN ARAÇ VE YÖNTEM

3.1. SİMÜLASYON

3.1.1. Kullanılan Programlama Dilleri ve Platformlar

Programlamada, Sun firmasına ait Java platformunun, Java SE 1.8.2_201 ürünü kullanılmıştır. Java'nın tercih edilmesinin sebebi, çalışma grubunun diğer programlama dillerine göre daha yatkın olması ve java dilinin projede ihtiyaç olunacak her gereksinimi karşılamasıdır. Ayrıca, Java'nın tercih edilmesinin bir başka sebebi de, Java'nın içerdiği geniş paket kitaplığıdır. Projede çalışma grubunun en iyi bildiği IDE olduğu için Netbeans IDE 8.2 kullanılmıştır. Debugging gibi projede kullanılacak özelliklere sahip olduğu için sorun yaşanmamıştır.

Tablo 3.1 : Projede kullanılan paketler ve onların içlerindeki sınıflar

javax.swing	org.bytedeco.javacpp	org.bytedeco.javacpp.lept	org.bytedeco.javacpp.tesseract
JOptionPane	*	*	*

*: Bu paketin tüm sınıfları import edildi.

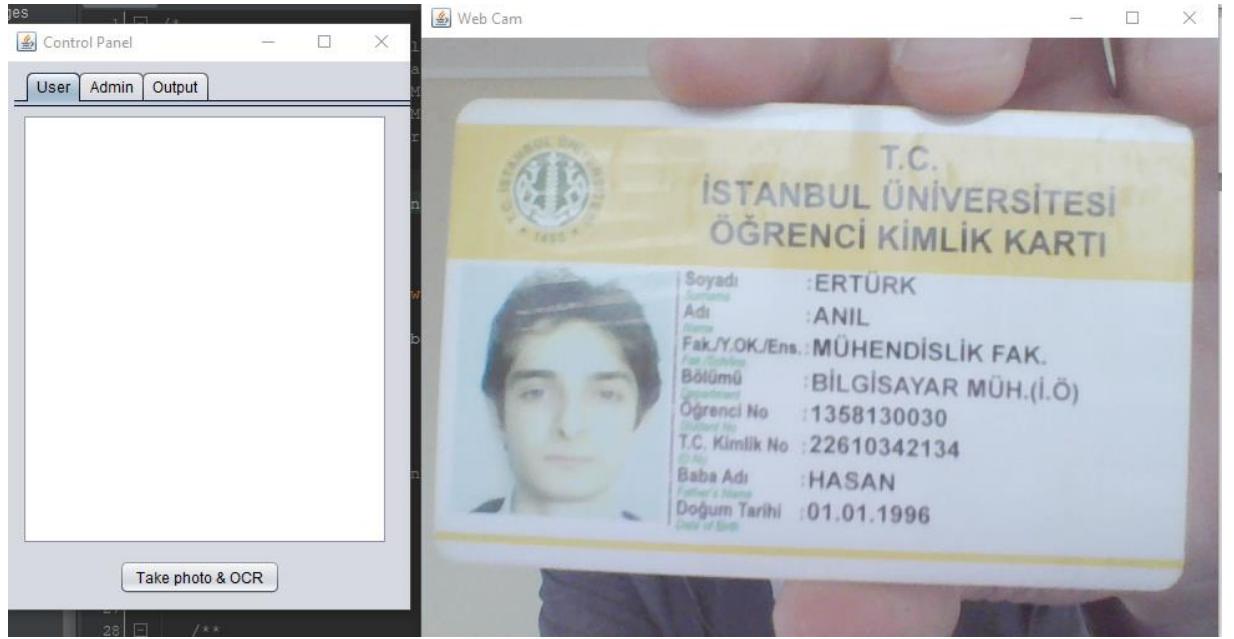
Tablo 3.2 : Projede kullanılan paketler ve onların içlerindeki sınıflar

org.bytedeco.javacv	org.bytedeco.javacpp.opencv_core	org.bytedeco.javacpp.opencv_imgcodecs
*	IplImage	cvSaveImage

Tablo 3.3 : Projede kullanılan paketler ve onların içlerindeki sınıflar

java.io	java.io
File	Connection
PrintWriter	DriverManager
	ResultSet
	Statement

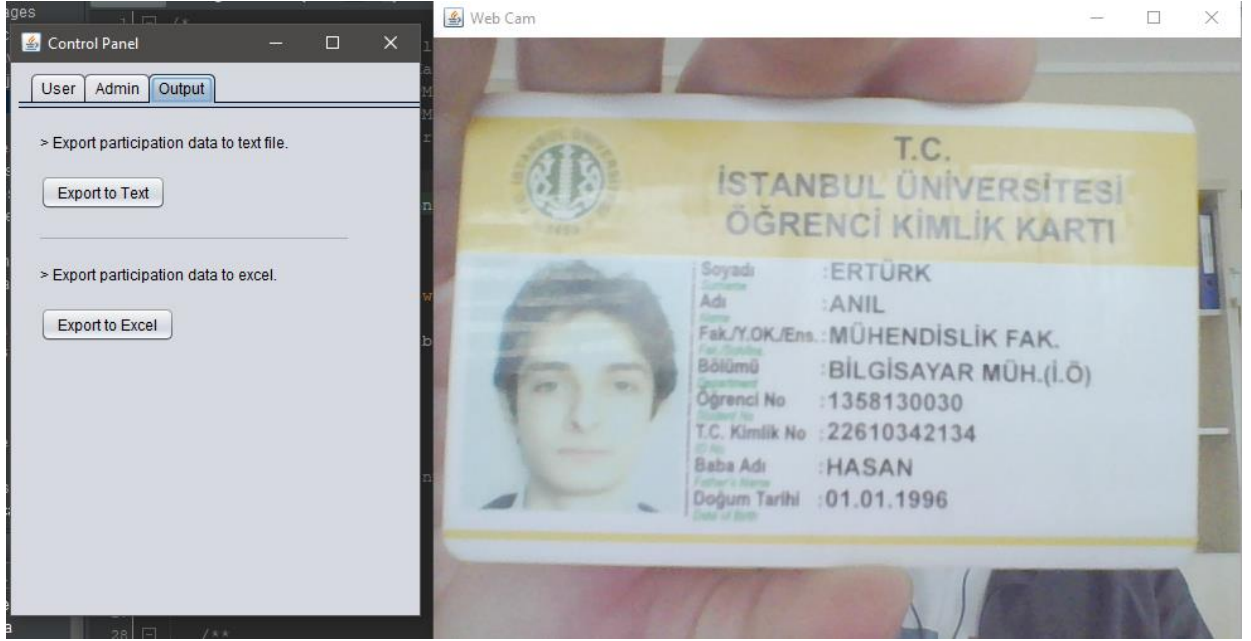
3.1.2. Programın Çalışması



Şekil 3.1 Control Panel penceresi(User sekmesi) ve Web Cam penceresi

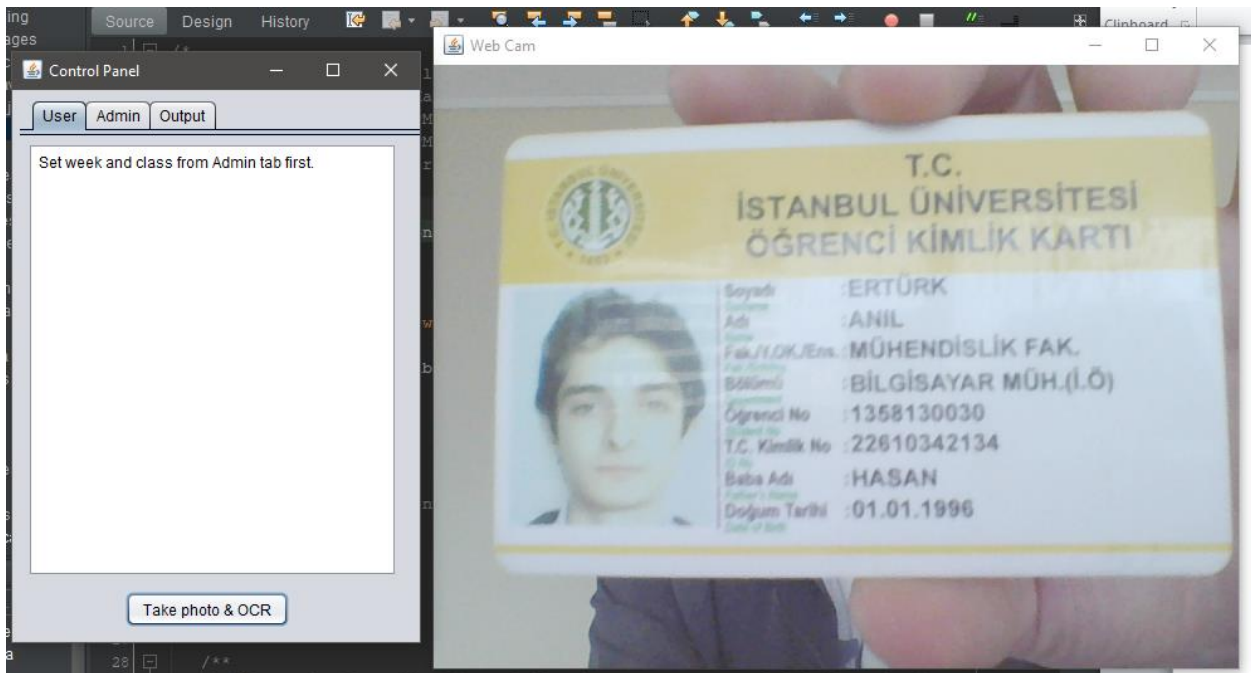


Şekil 3.2 Control Panel penceresi(Admin sekmesi) ve Web Cam penceresi



Şekil 3.3 Control Panel penceresi(Output sekmesi) ve Web Cam penceresi

Hafta ve sınıf seçmeden fotoğraf çekmeye çalıştığımızda program bizi uyarıyor.



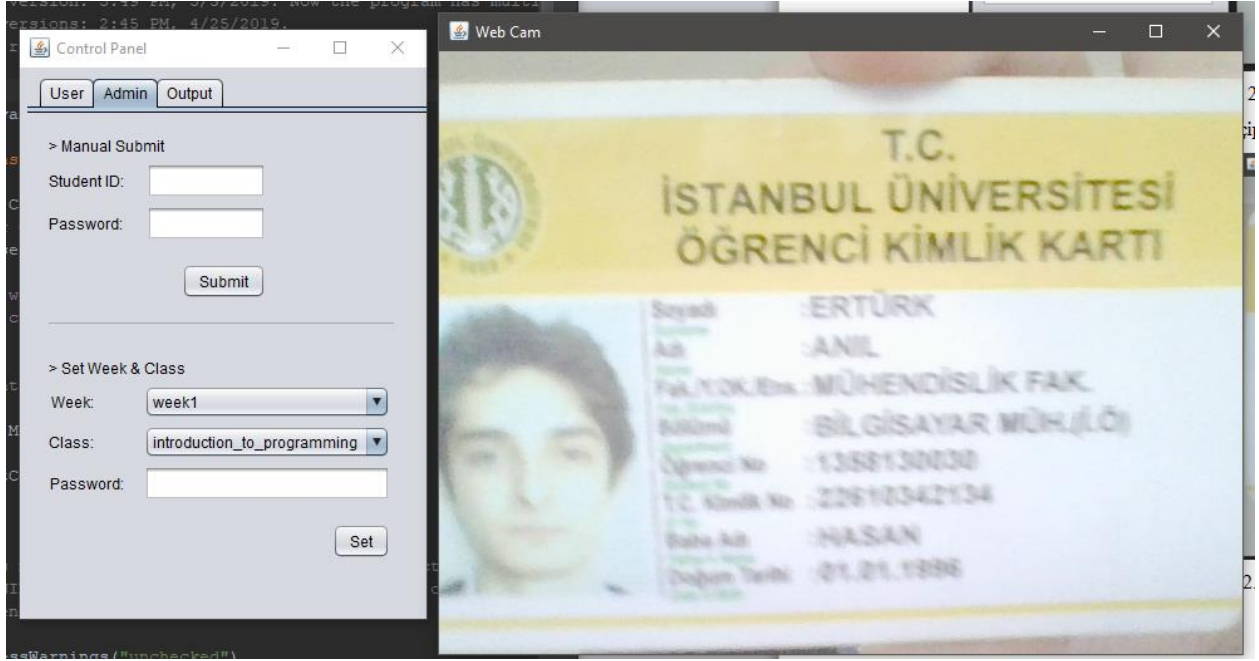
Şekil 3.4 Take photo & OCR

Admin sekmesinden hafta ve sınıf seçip, şifre giriyoruz.



Şekil 3.5 Hafta ve Sınıf seçimi

Şifre girilip Set butonuna basıldığı zaman otomatik olarak şifre kutusuna odaklanılıyor ve kutudaki şifre sıfırlanıyor. Hem güvenlik için hemde yanlış girilen şifrenin hızlıca doğru girilmesi için eklenen bir özellik bu.

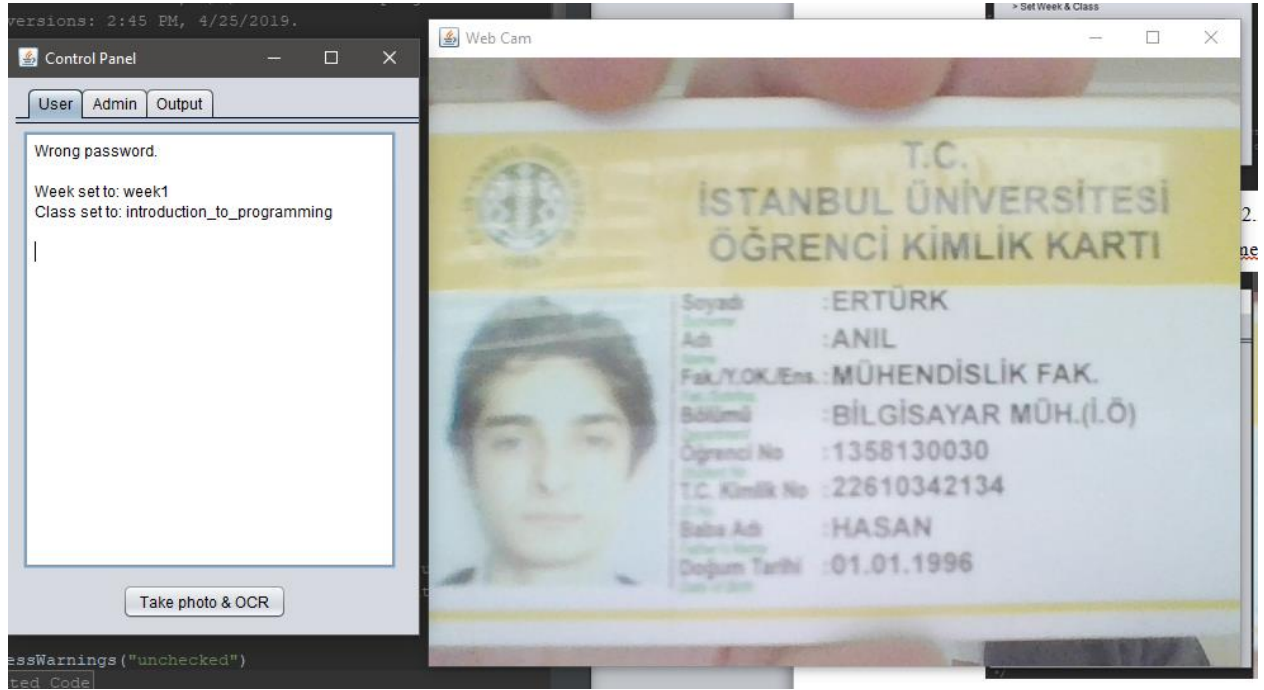


Şekil 3.6 Set tuşuna bastıktan sonra

Yanlış şifre girilirse aşağıdaki mesaj gösteriliyor.



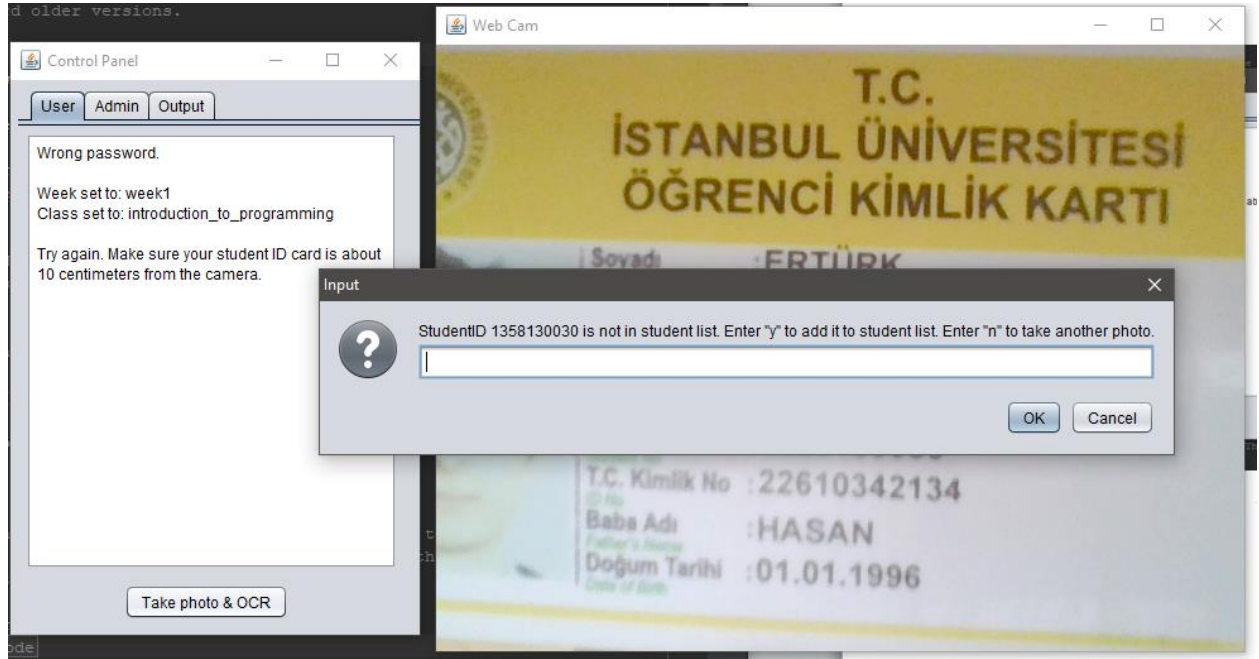
Şekil 3.7 Yanlış şifre sonucu



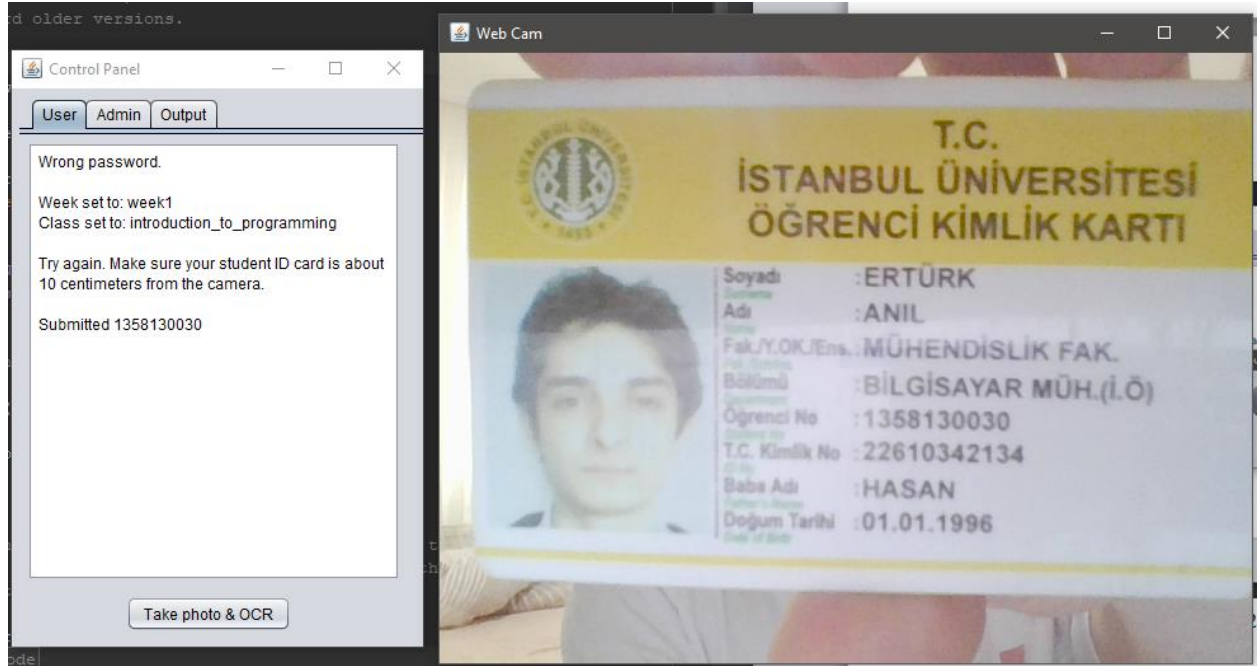
Şekil 3.8 Doğru şifre sonucu



Şekil 3.9 Başarısız OCR sonucu



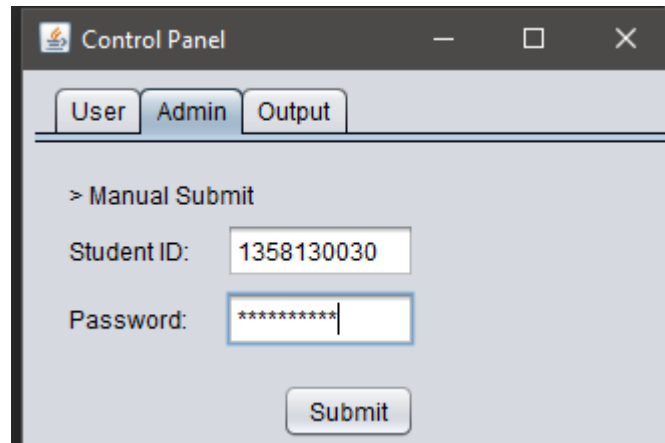
Şekil 3.10 Başarılı OCR, kayıtsız Numara sonucu



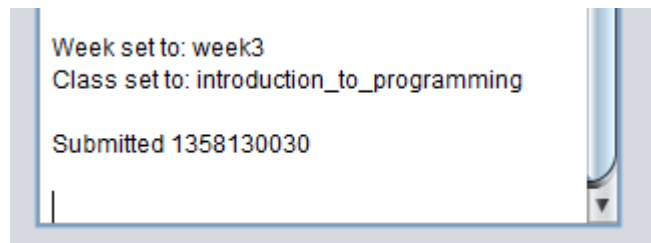
Şekil 3.11 Başarılı OCR, kayıtsız Numara onay sonucu



Şekil 3.12 Başarılı OCR, kayıtlı Numara sonucu



Şekil 3.13 Kayıtlı Numara el ile giriş



Şekil 3.14 Kayıtlı Numara el ile giriş sonucu

Control Panel

User Admin Output

> Manual Submit

Student ID: 1358130099

Password: *****

Submit

Şekil 3.15 Kayıtsız Numara manuel giriş

Control Panel

User Admin Output

> Manual Submit

Student ID: 1358130099

Password: *****

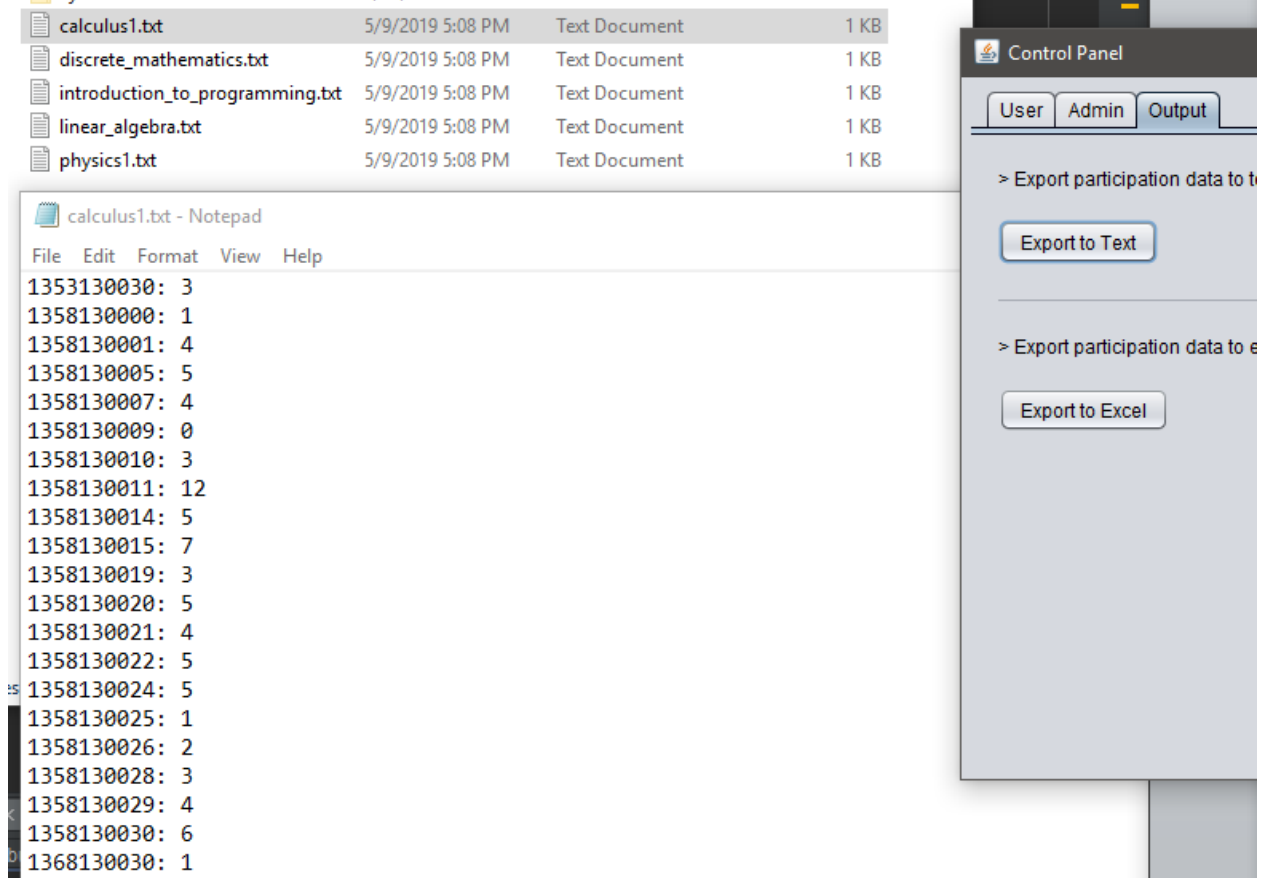
Submitted 1358130099

Input

? StudentID 1358130099 is not in student list. Enter "y" to add it to student list. Enter "n" to take another photo.

OK Cancel

Şekil 3.16 Kayıtsız Numara manuel giriş sonucu



Şekil 3.17 Export to Text sonucu

The screenshot displays a file explorer window with a list of files, a Microsoft Excel spreadsheet, and a Control Panel dialog box.

File Explorer List:

File Name	Date/Time	Type	Size
calculus1.txt	5/9/2019 5:08 PM	Text Document	1 KB
calculus1.xls	5/9/2019 5:09 PM	Microsoft Excel 97-...	1 KB
discrete_mathematics.txt	5/9/2019 5:08 PM	Text Document	1 KB
discrete_mathematics.xls	5/9/2019 5:09 PM	Microsoft Excel 97-...	1 KB
introduction_to_programming.txt	5/9/2019 5:08 PM	Text Document	1 KB
introduction_to_programming.xls	5/9/2019 5:09 PM	Microsoft Excel 97-...	1 KB
linear_algebra.txt	5/9/2019 5:08 PM	Text Document	1 KB
linear_algebra.xls	5/9/2019 5:09 PM	Microsoft Excel 97-...	1 KB

Microsoft Excel Spreadsheet:

A	B	C	D	E	F	G	H	I	J
1353130030	0	1	1	0	0	0	0	0	0
1358130000	1	0	0	0	0	0	0	0	0
1358130001	1	0	0	0	1	0	0	1	0
1358130005	0	0	0	1	0	1	0	1	1
1358130007	1	0	0	0	1	0	1	0	1
1358130009	0	0	0	0	0	0	0	0	0
1358130010	0	0	1	0	0	1	0	0	0
1358130011	1	1	1	1	1	1	1	1	1
1358130014	0	1	0	1	0	0	0	0	1
1358130015	1	0	1	0	1	1	0	0	1
1358130019	0	0	0	1	0	0	0	1	0
1358130020	1	0	0	0	0	1	0	0	0
1358130021	1	1	1	1	0	0	0	0	0
1358130022	0	0	1	1	0	1	0	1	0
1358130024	1	1	0	1	0	0	1	0	0
1358130025	0	0	0	0	0	0	1	0	0
1358130026	0	0	0	0	0	1	1	0	0
1358130028	0	0	1	0	0	0	0	0	1
1358130029	1	0	0	0	0	1	0	1	1
1358130030	1	1	1	1	0	1	0	0	0
1368130030	1	0	0	0	0	0	0	0	0

Control Panel Dialog Box:

Control Panel

User Admin Output

> Export participation data to text file.

Export to Text

> Export participation data to excel.

Export to Excel

Şekil 3.18 Export to Excel sonucu

4. TARTIŞMA VE SONUÇ

Yoklamanın görüntü işleme ile otomasyonu sayesinde hem öğrencilere hemde öğretmene büyük miktarda zaman kazandırılacak. Özellikle öğretmenin her hafta her ders için her öğrencin yoklama bilgisini otomatik olarak digital ortama geçirebilmesi ve istediği formatta geçirebilmesi projenin amacıydı. Dolayısı ile projemizin amacına ulaştığımızı söyleyebiliriz.

Programın ilk modellerinde kullanıcı iki buton kullanıyordu. İleri modellerde isRegistered metodu ve sınıfların öğrenci listeleri(tabloları) oluşturulduktan sonra ikinci butonun(submit butonu) gereksiz olduğu anlaşıldı ve programın bu noktada otomatikleştirildi.

Proje sırasında karşılaşılan sorunlardan bir tanesi, öğrenci kartında öğrenci numarasına bitişik bir iki nokta olmasıydı. Bu iki nokta bazen iki nokta olarak, bazen 1 olarak metine çeviriliyordu. Bazen hiç metine çevirilmiyordu. Dolayısı ile sadece 13 ile başlayan kelimelere değil 13, 113 ve :13 ile başlayan kelimelere bakarak bu OCR sorununun üstesinden gelebildik.

Programa kasıtlı olarak gizli güvenlik sistemleri eklendi kullanıcıların sistemi kandırmasının önlenmesi için. Sadece öğrenci numarası varmı yokmu bakmak yerine, çekilen fotoğrafta BILGISAYAR veya MUHENDISLIK kelimeleri varmı yokmu ona da bakıldı. Eğer bu iki kelimedenden birisi görülmez ise, öğrenci numarası olsa bile kabul edilmiyor çekilen fotoğraf.

KAYNAKLAR

1. DEITEL, Paul, DEITEL, Harvey, 2015, Java How to Program, Pearson, USA, One Lake Street, Upper Saddle River New Jersey 07458, 0-13-380780-0
2. Technological unemployment, [online] https://en.wikipedia.org/wiki/Technological_unemployment
3. Great Filter, [online] https://en.wikipedia.org/wiki/Great_Filter
4. Bidirectional Association, [online] <https://stackoverflow.com/questions/5360795/what-is-the-difference-between-unidirectional-and-bidirectional-jpa-and-hibernate>
5. Loose Coupling, [online] https://stackoverflow.com/questions/226977/what-is-loose-coupling-please-provide-examples?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa
6. Loose Coupling, [online] <https://softwareengineering.stackexchange.com/questions/244476/what-is-decoupling-and-what-development-areas-can-it-apply-to>
7. Tesseract Kütüphanesi, [online] <https://github.com/bytedeco/javacpp-presets/tree/master/tesseract>
8. JavaCV kütüphanesi, [online] <https://github.com/bytedeco/javacv>