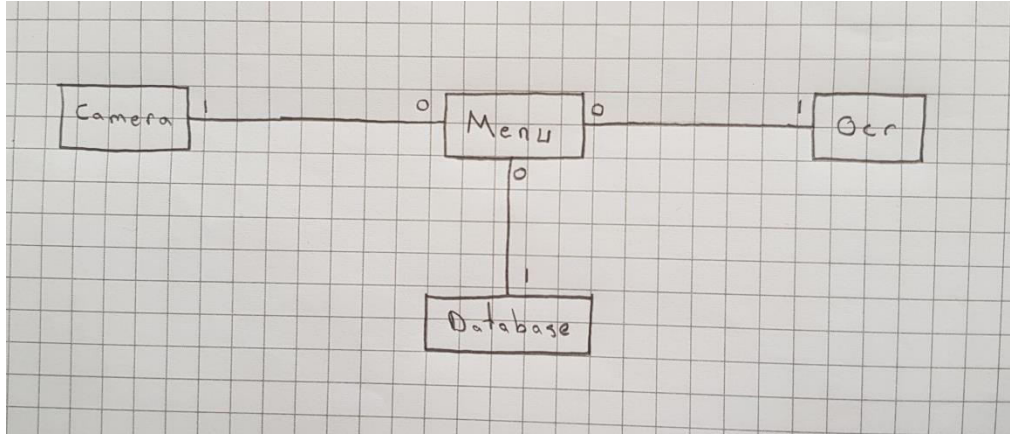


Görüntü İşleme ile Otomasyon Ara Rapor 2

Danışman: Mustafa Dağtekin
Öğrenci: Anıl Ertürk 1358130030

Proje ilerleme/değişiklikler

Aşağıdaki şekilde yeni eklenen Database sınıfı ile projenin güncel halini görüyoruz. Sınıf diyagramı bize kötü bir yöntem olan bidirection association kullanmadığımızı gösteriyor. Menu sınıfımız ana sınıf olarak işlev görüyor. Main metodumuz ve grafik arayüzümüz Menu sınıfında. Dolayısı ile main sınıfın Menu olması, diğer sınıfların belirli işleri yapmak için kullanılan sınıflar olması iyi bir tasarım şekli.



Camera sınıfımızda hiçbir değişiklik yapmadık. Webcam penceresi ile programın kalanını aynı pencereye koymaya çalıştım fakat kamerayı kontrol etmek için kullandığım javacv/opencv sınıfları awt(eski teknoloji grafik arayüz platform) kullanıyor. Dolayısı ile programın geri kalanı swing(daha yeni bir grafik arayüz platform) olduğu için ikisini bir araya getiremedim. Araştırmam sırasında yazılımcıların vardığı kanaatin awt ve swing parçalarını bir arada kullanmamak olduğunu da gördüm.

Ocr sınıfında değişiklikler yaptık.

1) MUHENDISLIK veya BILGISAYAR kelimesi görülünce securityFlag true yapılsın ve sonraki iterasyona geçilsin dedim. Bu iki string de öğrenci numarası olmadığını biliyorum, boşuna bu stringlerin icinde ogrenci numarasi varmi yokmu bakmaya gerek yok.

Eski kod

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

Yeni kod

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

2) Programı test ederken bazen 13 ile başlayan bir kelime değil :13 ile başlayan bir kelime döndürdüğünü fark ettim. Öğrenci kartı üzerinde de :1358130030 şeklinde duruyor öğrenci numarası. Dolayısı ile bu durumdaki kabul edilen durumlar arasına katmak için kodda değişiklikler yaptım.

Eski kod

```
String subString = data.substring(0, 2);  
  
// We already did length test above.  
if ("13".equals(subString) && securityFlag == true)  
{  
    securityFlag2 = true; // Valid student number.  
    // Getting rid of extra characters.  
    studentNumber = Integer.parseInt(data.substring(0, 10));  
  
    securityFlag = false; // Resetting object to be used again  
    return data.substring(0, 10);  
}
```

Yeni kod

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

    if ("13".equals(subString))
    {
        securityFlag2 = true; // Valid student number.

        // Getting rid of extra characters.
        studentNumber = Integer.parseInt(data.substring(0, 10));

        securityFlag = false; // Resetting object to be used again
        return data.substring(0, 10);
    }
    else if (":1".equals(subString))
    {
        securityFlag2 = true; // Valid student number.

        // Getting rid of extra characters.
        studentNumber = Integer.parseInt(data.substring(1, 11));

        securityFlag = false; // Resetting object to be used again
        return data.substring(1, 11);
    }
}
```

Database sınıfı yeni sınıfımız. Bir öğrenci numarası veritabanında zaten kayıtlımı sorgusu yapmak, kayıtlıysa sadece o kayıdı editle, değilse yeni kayıt yarat şeklinde işlevleri var.

1) isRegistered metodu. Bu metod bu öğrenci numarasına sahip bir kayıt varmı yokmu sorusunun sonucunu Boolean olarak döndürür.

Bu metod da kaynakları salmayı iki farklı yere yazmak istemediğim için finally ve try with resources ile tek seferde yapmayı denedim fakat finally'de scope dışına çıktığı için kaynak salma yapamadım. Bu bana ilginç geldi çünkü daha yeni java kitabımdan exception handling konusuna çalıştım ve orada bana finally ne olursa olsun(System.exit hariç) çalıştığı için kaynak salma için iyi bir yerdir denmişti. Try with resources i kullanmadım çünkü kullansaydım kodum okunması ve anlaşılması çok zor bir hale gelecekti.

```

public boolean isRegistered(int studentID)
{
    Connection c = null;
    Statement stmt = null;

    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 class1;" );
        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;
}

```

2) insert metodu. Bu metod eğer bu öğrenci numarasıyla açılmış bir kayıt varsa sadece bu hafta da geldiğini veritabanında işaretler, yoksa yeni bir kayıt yarat ve bu hafta geldiğini işaretler şeklinde işlemler yapar.

```

public void insert(int studentID, String date)
{
    Connection c = null;
    Statement stmt = null;
    //DriverManager.registerDriver(new com.mysql.jdbc.Driver());

    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 (isRegistered(studentID))
        {
            stmt.executeUpdate("UPDATE class1 SET " + date + " = 1 "
                + "WHERE studentID = " + studentID);
        }
        else
        {
            stmt.executeUpdate("INSERT INTO `class1` (studentID , " + date+ ") "
                + "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.");
    }
}

```

Veritabanımızda şimdilik tek bir tablo var(class1). Birden çok sınıf desteği için birden çok tablo yaratacağım ilerde ve programın geri kalanında şu anda nasıl hafta seçimi yapılıyor ve bu hafta bilgisi kullanılıyorsa, sınıf seçimi ve kullanımı da öyle olacak. Tabi birde sınıfListesi tablosuna ve bu tablo bilgisini sınıf seçerken görülecek listeye ekleyen metod da yazılacak.

Class1 tablomuzun yapısı 1 adet studentID(int 10) sütünü, 12 adet week1-12(tinyint 1) sütununda oluşmakta.

#	studentID	week1	week2	week3	week4	week5	week6	week7	week8	week9	week10	week11	week12
1	1358130000	✓	■	■	■	■	■	■	■	■	■	■	■
2	1358130031	✓	■	■	■	✓	■	■	■	■	■	■	■
3	1358130032	✓	■	✓	✓	■	■	■	■	■	■	■	■
4	1358130030	✓	■	■	■	■	■	■	■	■	■	■	■
5	135813003	✓	■	■	■	■	■	■	■	■	■	■	■

Menu sınıfında birçok değişiklik ve yenilik yapıldı.

1) Static olan ocr ve db değişkenleri artık static değil. cam değişkeni static olmak zorunda çünkü bu objeyi static main metodunda çağırıyoruz programın başında video yayını başlatması için.

Ayrıca hafta seçimi ve ders seçimi yapılması için kullanılan iki değişken eklendi. İlk başta tarih seçimini her dersin başında admin yapmayacak, system kendisi görecekti tarihin ne olduğunu ve veritabanında gidip o tarihte bu öğrenci geldi şeklinde işaretleyecek diye düşünüyordum. Fakat işin içine girince bunun sıkıntılarını fark ettim. Her ders ve her dönem derslerin tarihleri değişecek. Dolayısı ile şimdilik tarih kullanmadım, hafta1, hafta2, hafta3 , ..., hafta 12 kullandım.

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

String weekNo;
String className;
```

2) photoAndOcrActionPerformed metodunda değişiklik yaptım. Eskiden sadece 13 ile başlayan 10 rakam uzunluğunda sayılar döndürülüyordu ocr.analyze() metodundan. Artık o metod :13 ile başlayan, : hariç 10 rakam uzunluğunda olan sayılarda döndürülüyor.

Eski kod

```
// Decide whether image was valid or not. Prompt user to submit or try again
if ("13".equals(str.substring(0, 2)))
{
    // Image was valid.
    jTextPanel.setText(jTextPanel.getText() + "If your student number is "
        + str + ", press Submit to continue. If not, take another photo.\n\n");
}
else
{
    // Image was not valid.
    jTextPanel.setText(jTextPanel.getText() + str + "\n\n");
}
```

Yeni kod

```
// Decide whether image was valid or not. Prompt user to submit or try again.
if ("13".equals(str.substring(0, 2)) || ":13".equals(str.substring(0, 3)))
{
    // Image was valid.
    jTextPanel.setText(jTextPanel.getText() + "If your student number is "
        + str + ", press Submit to continue. If not, take another photo.\n\n");
}
else
{
    // Image was not valid.
    jTextPanel.setText(jTextPanel.getText() + str + "\n\n");
}
```

3) submitActionPerformed metodundada değişiklikler yapıldı. Eskiden boş bırakılmıştı, database sınıfı ve metodları tamamlanınca içi doldurulacak denmişti. Şimdi tamamlandı.

Eski kod

```
private void submitActionPerformed(java.awt.event.ActionEvent evt)
{
    if (ocr.securityFlag2 = true)
    {
        /* !! TO DO !! Create a class for database stuff, create a static object in Menu class
            field. Then call the method of that object here with student number as argument*/

        // Submit was successful.
        jTextPanel.setText(jTextPanel.getText() + "Submitted " + ocr.studentNumber
            + "\n\n");
    }
}
```

Yeni kod

```
private void submitActionPerformed(java.awt.event.ActionEvent evt)
{
    if (weekNo == null)
    {
        jTextPanel.setText(jTextPanel.getText() + "Set week from admin tab "
            + "first." + "\n\n");
    }
    else
    {
        if (ocr.securityFlag2 == true)
        {
            db.insert(ocr.studentNumber, weekNo);

            // Submit was successful.
            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");
        }
    }
}
```

Eğer hafta seçilmiş ise, ve valid bir öğrenci numarası görüntüden extract edilmiş ise işlemlere devam ediyor.

4) manualSubmitActionPerformed metodundada değişiklikler yapıldı. Eskiden boş bırakılan database kısmı dolduruldu. Hafta seçilmişmi kontrolü eklendi.

Eski kod

```
private void manualSubmitActionPerformed(java.awt.event.ActionEvent evt)
{
    try
    {
        if ("13".equals(studentNumber.getText().substring(0, 2)) && studentNumber.
            getText().length() == 10 && "dagtekin13".equals(password.getText()))
        {
            // Checking if studentNumber contains non digit characters.
            Integer.parseInt(studentNumber.getText());
            /* !! TO DO !! Create a class for database stuff, create a static object in Menu class
            field. Then call the method of that object here with student number as argument*/

            // Submit was successful.
            jTextPanel.setText(jTextPanel.getText() + "Submitted "
                + studentNumber.getText() + "\n\n");
        }
        else
        {
            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();
    }
    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();
    }
}
```

Yeni kod

```
private void manualSubmitActionPerformed(java.awt.event.ActionEvent evt)
{
    try
    {
        if (weekNo == null)
        {
            jTextPanel.setText(jTextPanel.getText() + "Set week from admin tab "
                + "first." + "\n\n");
        }
        else
        {
            if ("13".equals(studentNumber.getText().substring(0, 2)) && studentNumber.
                getText().length() == 10 && "dagtekin13".equals(password.getText()))
            {
                // Checking if studentNumber contains non digit characters.
                db.insert(Integer.parseInt(studentNumber.getText()), weekNo);

                // Submit was successful.
                jTextPanel.setText(jTextPanel.getText() + "Submitted. "
                    + studentNumber.getText() + "\n\n");
            }
            else
            {
                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();
        }
    }
}
```

5) Control paneli grafik arayüzü değiştirildi. Tabbed pane sistemine geçildi. Artık öğrencinin kullandığı tab ve adminin kullandığı tab farklı. Manuel giriş, admin tabına konuldu. Ayrıca hafta ve ders seçimi özellikleride admin tabına konuldu. Aşağıda hafta ve sınıf bilgisini kayıt eden butonun yaptıklarını görüyoruz. Ve bu butonun üzerindeki password field'da enter'a basında ne yapıldığını görüyoruz.

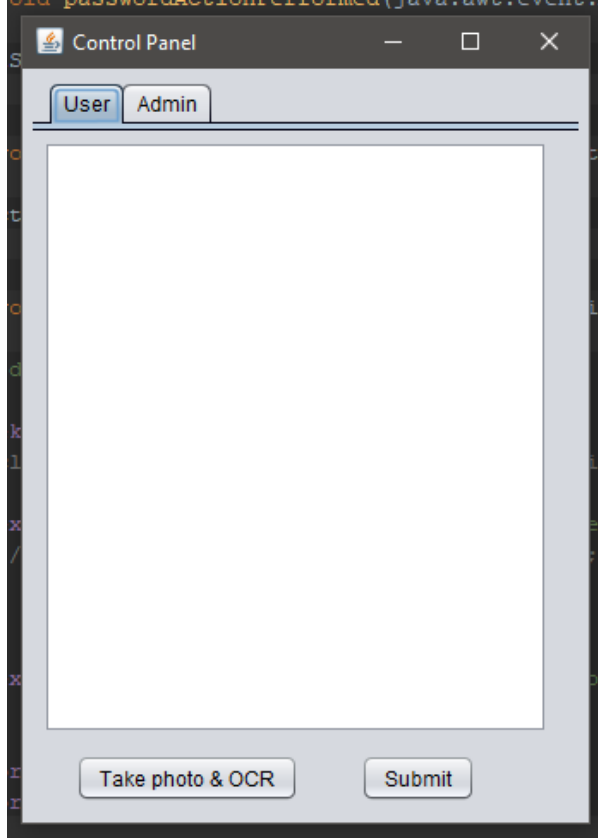
```
private void password1ActionPerformed(java.awt.event.ActionEvent evt)
{
    saveActionPerformed(evt);
}

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

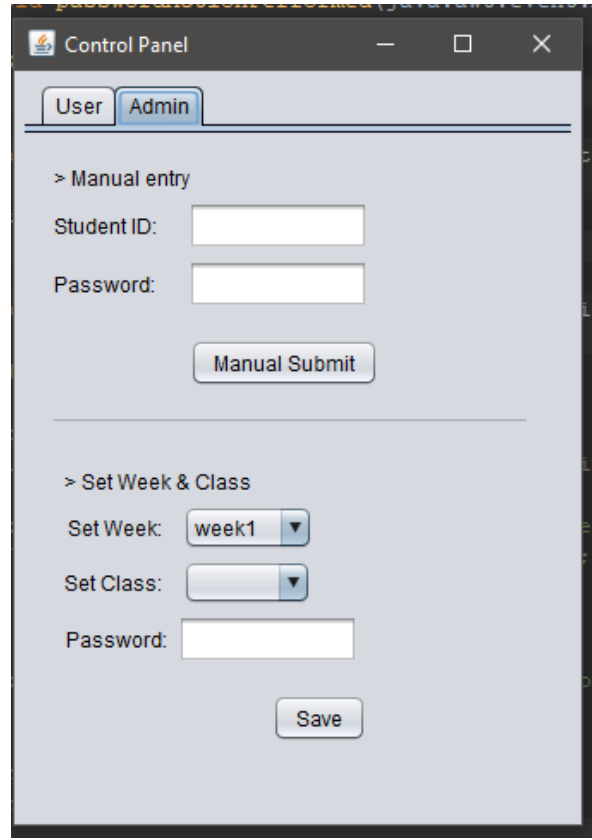
        jTextPanel.setText(jTextPanel.getText() + "Week set to: " + weekNo + "\n\n"
            //"\nClass set to: " + className + "\n\n");
    }
    else
    {
        jTextPanel.setText(jTextPanel.getText() + "Wrong password." + "\n\n");
    }

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

Sonuç



The image shows a web application window titled "Control Panel". It has two tabs: "User" and "Admin". The "User" tab is selected. The main content area is a large, empty white rectangle. At the bottom of the window, there are two buttons: "Take photo & OCR" and "Submit".



The image shows the same "Control Panel" window, but with the "Admin" tab selected. The main content area is divided into two sections. The first section is titled "> Manual entry" and contains two text input fields labeled "Student ID:" and "Password:". Below these fields is a button labeled "Manual Submit". The second section is titled "> Set Week & Class" and contains two dropdown menus labeled "Set Week:" (with "week1" selected) and "Set Class:". Below these dropdowns is a text input field labeled "Password:". At the bottom of this section is a button labeled "Save".