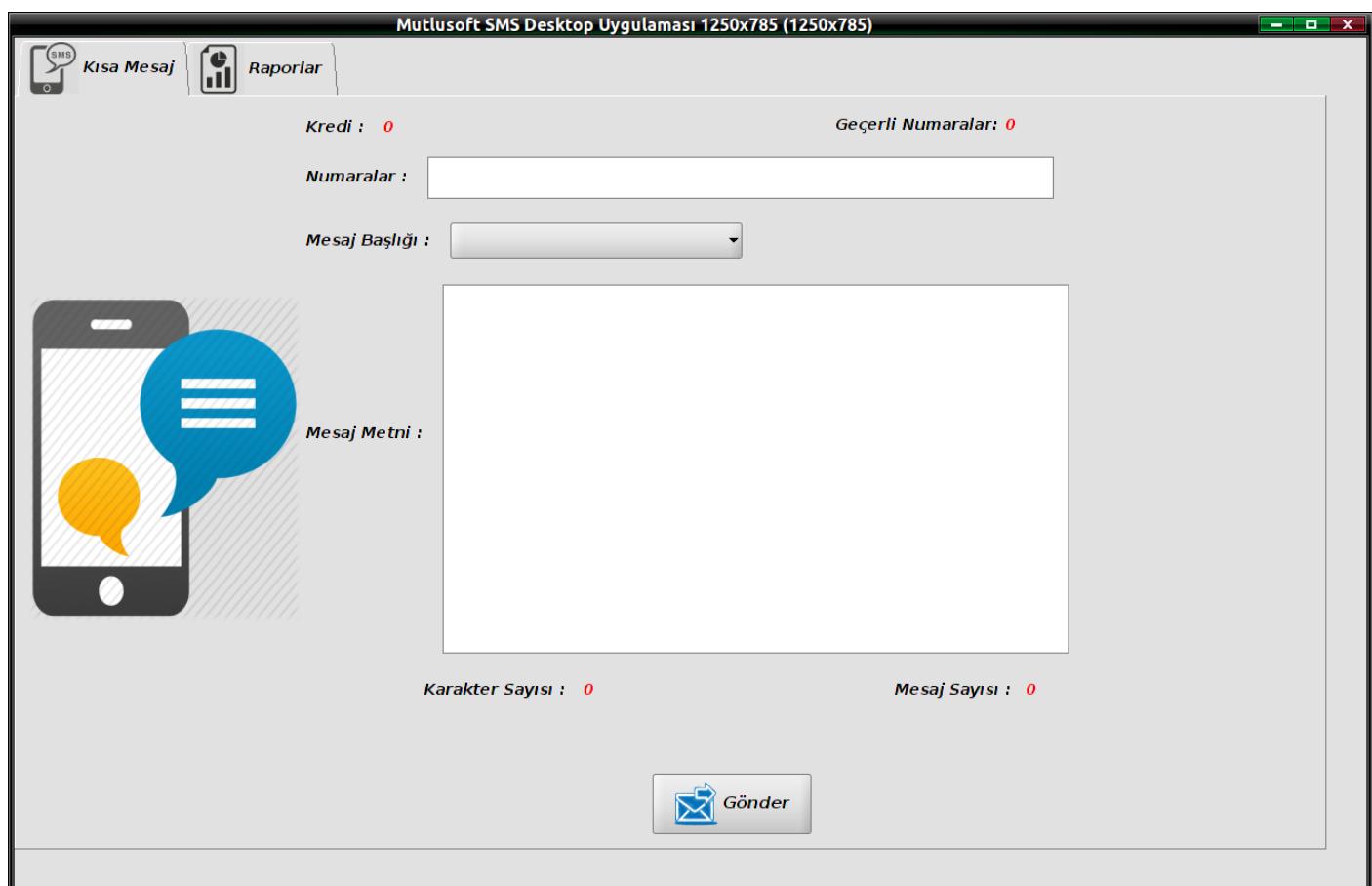


KISIM		YAPRAK NO : 23
YAPILAN İŞ	Mutlusoft SMS Desktop Projesinin Arayüzünün Oluşturulması	TARİH : 23/07/2014

Bugün **Mutlusoft SMS Desktop** projesine başlayacağımız belirtildi. Proje arayüzünün nasıl olması gerekiğiyle ilgili bir sunum yapıldı. **MUTLUCELL API'leri** hakkında bilgi verildi. Program arayüzünü oluştururken **JFormDesigner'in** kullanılacağı söylendi. **JAVA'da GUI** elemanları anlatıldı.

MUTLUCELL API'lerine çalışıldı. Örnek **API** kullanıyla ilgili test sınıfları incelandı, çalıştırıldı. **API'ler XML** içerdigi için **XML** ile ilgili kaba taslak bilgi edinildi. Sonrasında **JFormDesigner'in Eclipse eklentisi** kuruldu. Nasıl kullanıldığı ile ilgili internetten araştırma yapıldı.

Chapter 25 GUI Components: Part 2 ünitesine çalışıldı. Ardından JFormDesigner kullanılarak aşağıdaki arayüz oluşturuldu.



KISIM		YAPRAK NO : 24
YAPILAN İŞ	SMS Projesinin Temel Sınıf ve Metodlarının Oluşturulması	TARİH : 24/07/2014

Toplu SMS projesinde hangi sınıfların olması ve nasıl bir yol izlenmesi gerektiğiyle ilgili mühendisten bilgi alındı. Öncelikle dün oluşturduğumuz **SmsFrame arayüz sınıfını** çağrıran aşağıdaki **Main** sınıfı oluşturuldu.

```
public class Main
{
    public static void main( String[] args )
    {
        SmsFrame frame = new SmsFrame();
        frame.setVisible( true );
    }
}
```

Ardından **Util sınıfı** oluşturuldu. Bu sınıfta **Mutlucell API'leri** kullanılarak **kalan kredi, originator (mesaj başlığı), rapor** için gerekli bilgilerin çekilmesinde kullanılacak metodların oluşturulacağı bilgisi alındı. **mutlucell.com** üzerinden üzerinden 50 kredilik bir hesap açılarak **kullanıcı adı** ve **şifre** belirlendi ve **Util** sınıfından sabit değişkenlere atandı.

```
public static final String USERNAME = "feux075";
public static final String PWD = "qwertyx";
```

Sunucudan **kalan krediyi** çekmek için **HTTPUtil kütüphanesinin httppost metodu** kullanıldı. Krediyi gösteren aşağıdaki metod oluşturuldu.

```
public static double getCredits() throws Exception
{
    String linkCredits = "https://msgw.mutlucell.com/msgw-ws/gtcrdtx";
    String xmlCredits = "<?xml version='1.0' encoding='utf-8'?><smskredi ka="" + USERNAME
+ "" pwd="" + PWD + """/>";
    String result = HTTPUtil.httppost( linkCredits, xmlCredits );
    return Double.valueOf( result.substring( 1 ) );
}
```

Burada geri dönen result değeri **System.out.println(result)** ile ekrana yazdırıldığında \$kredimiktari formatında olduğu için **substring ()** metoduya sayı olan kısım alınıp **Double.valueOf()** metoduya **double** türüne dönüştürüldü.

Ardından **originatorları** çekmek için aşağıdaki metod oluşturuldu. Burada birden fazla **originator** durumu olabileceğiinden **liste** türünde tanımlandı.

```
public static List<String> getOriginators() throws Exception
{
    String xmlOriginator = "<?xml version='1.0' encoding='utf-8'?><smsorig ka="" +
USERNAME + "" pwd="" + PWD
+ """/>";
    String linkOriginator = "https://msgw.mutlucell.com/msgw-ws/gtorgex";
    String result = HTTPUtil.httppost( linkOriginator, xmlOriginator );

    String[] string = result.split( "\t" );
    List<String> originator = new ArrayList<String>();
```

```
for ( String object : string ) {  
    originator.add( object );  
}  
return originator;  
}
```

Projesinin kontrol sınıfı olan **Controller sınıfı** aşağıdaki gibi oluşturuldu.

```
public class Controller extends JPanel  
{  
    public Controller( SmsFrame frame )  
    {  
        this.frame = frame;  
    }  
}
```

Sınıfların birbiriyle iletişimini için **Main sınıfı** içerisinde aşağıdaki kod eklendi.

```
Controller controller = new Controller( frame );
```

KISIM		YAPRAK NO : 25
YAPILAN İŞ	Kredi ve Mesaj Başlığı Verilerinin Arayüze Setlenmesi	TARİH : 25/07/2014

Controller sınıfında **init()** metodu tanımlandı ve **sınıfın yapıcısında** çağrıldı. Burada **init()** metodu **OOP**'in ruhuna uygun olarak çağırılacak metodlar ve olayların barındırılmasında kullanılacak.

```
public void init()
{
    loadCredits();
    loadOriginator();
}
```

loadCredits() ve **loadOriginator()** metodları **kredi miktarı** ve **mesaj başlığını sunucudan çekmek** için kullanılan oluşturulan metodlar. Birbirlerinin gecikmelerinden beklememeleri için ikisinde de **thread** yapısı kullanılmış ve aşağıdaki gibi tanımlanmıştır.

```
private void loadOriginator()
{
    Thread threadOriginator = new Thread() {
        public void run()
        {
            try {

                for ( String string : Util.getOriginators() ) {
                    frame.getCbOriginator().addItem( string );
                }

            } catch ( Exception e ) {

                e.printStackTrace();
            }
        };
        threadOriginator.start();
    }
}

private void loadCredits()
{
    Thread thread = new Thread() {
        public void run()
        {

            try {

                frame.getLbCreditsNo().setText( String.valueOf( Util.getCredits() ) );
            }

            catch ( Exception e ) {

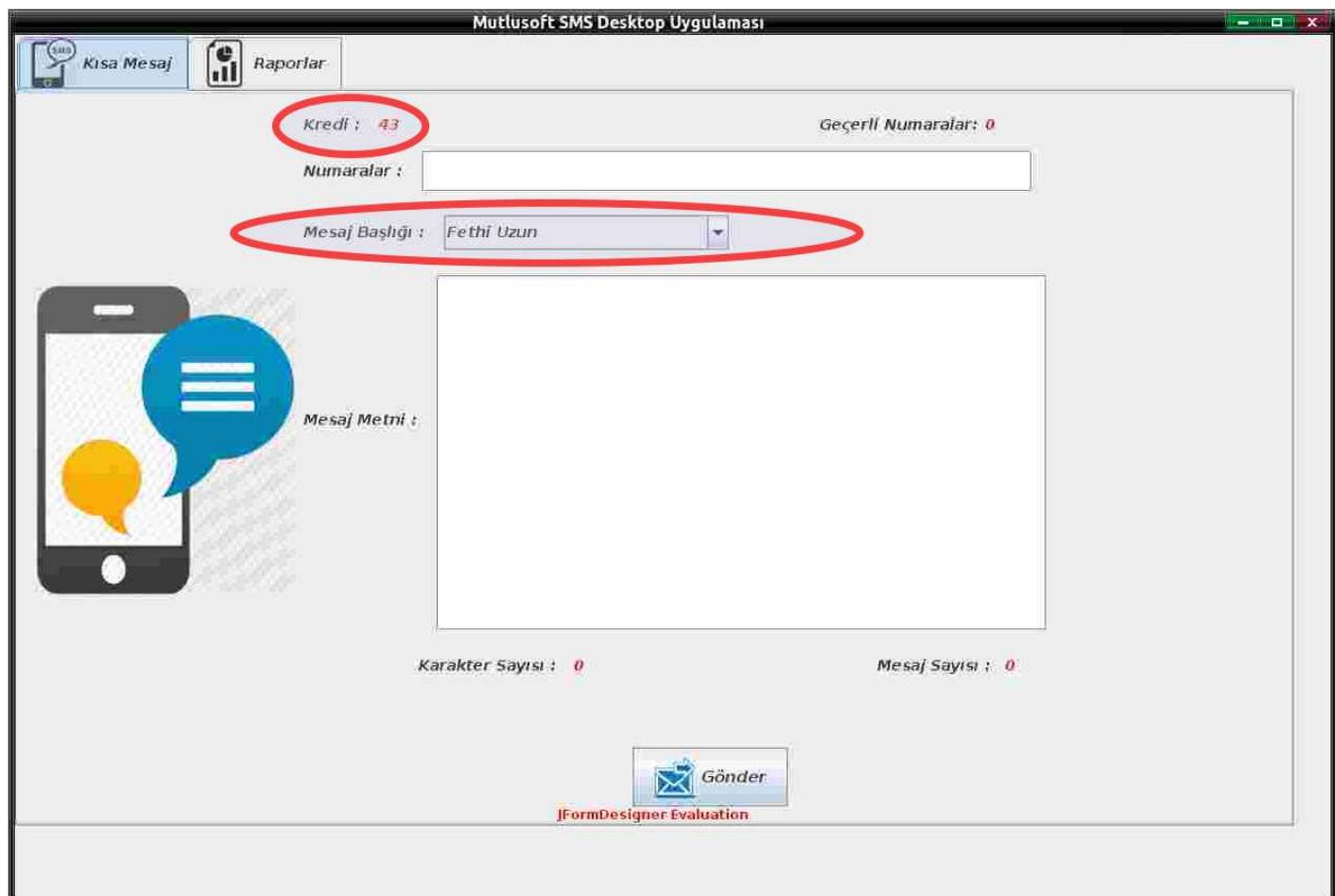
                e.printStackTrace();
            }
        };
        thread.start();
    }
}
```

```
for ( String string : Util.getOriginators() ) {  
    frame.getCbOriginator().addItem( string );
```

koduyla *alınan mesaj başlıklarını* SMS projesinin arayüzündeki “**Mesaj Başlığı**” bölümüne setlenmiş;

```
frame.getLbCreditsNo().setText( String.valueOf( Util.getCredits() ) );
```

koduyla da arayüzdeki “**Kredi**” bölümü setlenmiştir.



KISIM		YAPILAN İŞ	Mesaj Gönderim Metodunun Oluşturulması	YAPRAK NO : 26
				TARİH : 31/07/2014
<p><i>Util sınıfı</i> içerisinde mesaj göndermek için sendSms() metodu tanımlamamız istendi. Ayrıca mesaj gönderilemezse <i>sunucudan dönen hata kodlarının anlamını döndürecek</i> şekilde metoda eklememiz söylendi. Bunun üzerine Mutlucell API'leri ve hata kodları incelendikten sonra aşağıdaki metod oluşturuldu.</p>				
<pre> public static double sendSms(List<String> numbers, String originator, String message) throws Exception { String resultSend; String[] credits = {}; try { String xmlSend = "<?xml version='1.0' encoding='utf-8'?><smsspack ka="" + USERNAME + "" pwd=""" + PWD + "" org=""" + originator + ""><mesaj><metin>" + xmlEncode(message) + "</metin><nums>" + numbers + "</nums></mesaj></smsspack>"; String linkSend = "https://msgw.mutlucell.com/msgw-ws/snblkex"; resultSend = HTTPUtil.httppost(linkSend, xmlSend); if (resultSend.startsWith("\$")) { credits = resultSend.split("#"); return Double.parseDouble(credits[1]); } if (resultSend.equals("20")) { throw new RuntimeException("Post edilen xml eksik veya hatalı!"); } else if (resultSend.equals("21")) { throw new RuntimeException("Kullanılan originatöre sahip değilsiniz!"); } else if (resultSend.equals("22")) { throw new RuntimeException("Kontörünüz yetersiz!"); } else if (resultSend.equals("23")) { throw new RuntimeException("Kullanıcı adı ya da parolanız hatalı!"); } else if (resultSend.equals("24")) { throw new RuntimeException("şu anda size ait başka bir işlem aktif!"); } else if (resultSend.equals("25")) { throw new RuntimeException(" işlemi 1-2 dk sonra tekrar deneyin!"); } else if (resultSend.equals("30")) { throw new RuntimeException("Hesap Aktivasyonu sağlanmamış!"); } } catch (Exception e) { throw e; } return 0; } </pre>				

Burada metod **telefon numarası**, **mesaj başlığı** ve **mesaj metnini değer olarak alıp** sunucuya gönderdikten sonra **bir hata oluşmamışsa** dönen değer \$id#harcanankredi formatında; ancak \$ ile başlamıyorsa hata var anlamına gelmektedir. Bu nedenle dönen değer \$ ile başlamışsa # işaretine göre ayrıstırılıp sayı dönen kısmını **double** şeklinde geriye döndürülmüştür. \$ ile başlamamışsa ise geriye dönen sayıya göre **hata kodları** aşağıdaki anlamlara gelmektedir

20:Post edilen xml eksik veya hatalı.

21:Kullanılan originatöre sahip değilsiniz

22:Kontörünüz yetersiz

23:Kullanıcı adı ya da parolanız hatalı.

24:Şu anda size ait başka bir işlem aktif.

25:SMSC Stopped (Bu hatayı alırsanız, işlemi 1-2 dk sonra tekrar deneyin)

30:Hesap Aktivasyonu sağlanamamış

Aşağıdaki karakterler **XML'in özel karakterleri** olduğundan mesaj metni içerisindeki bu karakterler karşılıkındaki gibi **encode** edilmeleri gerektiği için bunu gerçekleştirecek **xmlEncode()** metodu oluşturulmuştur.

```
< = &lt;  
> = &gt;  
& = &amp;  
' = &apos;  
" = &quot;
```

```
private static String xmlEncode( String s )  
{  
    s = s.replace( "&", "&amp;" );  
  
    s = s.replace( "<", "&lt;" );  
  
    s = s.replace( ">", "&gt;" );  
  
    s = s.replace( "'", "&apos;" );  
  
    s = s.replace( "\"", "&quot;" );  
  
    return s;  
}
```

KISIM		YAPRAK NO : 27
YAPILAN İŞ	Girilen Numaraların Geçerlilik Kontrolleri	TARİH : 01/08/2014

Uygulamanın arayüzündeki **Numaralar** bölümüne, virgülle ayrılarak girilen **numaraların geçerli olup olmadığını**; rakam hariç **hiçbir karakter girilmesine izin verilmeyecek şekilde** (girildiği zaman numara harici karakterler otomatik silinecek) kontrolünün yapılması istendi. **Geçerli numaraların** arayüzdeki **“Geçerli Numaralar”** bölümüne yansıtılacağı ve Türkiye'de var olan telefon numarası formatlarının dikkate alınması gereği belirtildi.

Kurulan algoritmada geçerli numara kontrolü için **regular expressions(düzenli ifadeler)** kullanılmıştır. Rakam hariç karakterlerin silinmesi için de klavyeden girilen her karakter okunmuş; rakam ve virgül hariç olanların silinmesi sağlanmıştır. Geçerli numara kontrolü için **getValidNo()** metodu oluşturulmuştur. (kod okunabilirliğinin artırılması için) Ayrıca kullanıcının kopyala, kes, yapıştır işlemlerini kullanarak geçersiz karakter girmesini engellemek için de **CTRL+V** kombinasyonu telefon numaraları bölümü için pasif hale getirilmiştir.

```

frame.getTfNumbers().addKeyListener( new KeyListener() {

    @Override
    public void keyTyped( KeyEvent e )
    {

    }

    @Override
    public void keyReleased( KeyEvent e )
    {
        frame.getTfNumbers().setTransferHandler( null );// for block CTRL+V etc.

        String tempInputText = frame.getTfNumbers().getText();// sms metni

        String[] parsedInputText = tempInputText.split( "," );

        getValidNo( parsedInputText, e ); //geçerli no kontrolünü yapan metod
    }

    @Override
    public void keyPressed( KeyEvent arg0 )
    {
        String tempInputText = frame.getTfNumbers().getText();// sms metni

        char[] inputNo = frame.getTfNumbers().getText().toCharArray();

        for ( char c : inputNo ) {

            if ( !Character.isDigit( c ) && c != ',' ) { // virgül ve rakam hariç karakterler siliniyor

                frame.getTfNumbers().setText( tempInputText.substring( 0, tempInputText.length() -
                1 ) );
            }
        }
    }
});
```

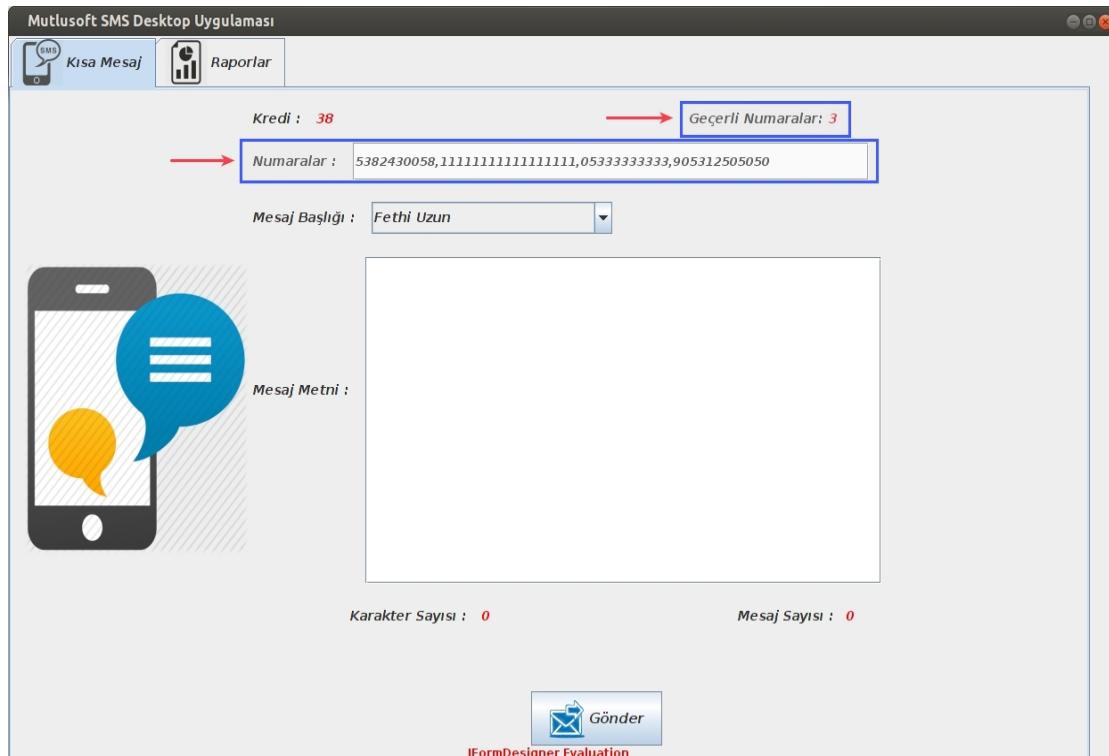
```

private void getValidNo( String[] parsedInputText, KeyEvent e)
{
    if ( e.getKeyCode() == KeyEvent.VK_BACK_SPACE ) {
        numbers.removeAll( numbers );
        for ( String string : parsedInputText ) {
            if ( string.matches( "(90|0)?5(0|3|4|5)(\\d){8}" ) ) {// silinen numaralar listeden de siliniyor
                if ( !this.numbers.contains( string ) ) {// birden fazla eklenen numaralardan sadece birisi listeye ekleniyor
                    this.numbers.add( string );
                }
            }
        }
    }

    for ( String string : parsedInputText ) {// gecerli numaralar listeye ekleniyor
        if ( string.matches( "(90|0)?5(0|3|4|5)(\\d){8}" ) ) {
            9053xxxxxxxx,054xxxxxxxx,55xxxxxxxx vs regular exp
            if ( !this.numbers.contains( string ) ) {

                this.numbers.add( string );
            }
        }
    }
    frame.getLbValidateNumberNo().setText( "" + numbers.size() );// gecerli numara sayisi
}

```



KISIM		YAPRAK NO : 28
YAPILAN İŞ	<i>Mesaj Gönderimi ve Uyarı Mesajlarının Oluşturulması</i>	TARİH : 04/08/2014

Arayüzdeki “**Gönder**” butonunun aktif hale getirilmesi ve tıklandığında mesaj metnini yollaması için gereken kodların yazılması istendi. Bunun için **Gönder** butonuna tıklandığı zaman olması gereken olaylar aşağıdaki gibi oluşturuldu.

```
frame.getBtSend().addActionListener( new ActionListener() {

    @Override
    public void actionPerformed( ActionEvent arg0)
    {

        try {
            toReadySMS();
            loadCredits();
        } catch ( Exception e ) {
            e.printStackTrace();
        }
    }
});
```

Burada çağrılan iki metoddan biri olan **ToReadySMS()**, mesajın gönderim şartlarını sağlayıp sağlamadığını kontrol eden, sağlanmayan şartları kullanıcıya görsel bir uyarı şeklinde bildiren, şartlar sağlandığında mesajı yollayan metoddur. Gönderim şartları ise aşağıdaki gibi yer almaktadır.

- ➔ Geçerli en az bir numara eklenmeli.
- ➔ Boş mesaj gönderilemez, mesaj metni olmalı.
- ➔ Mesaj metni 764 karakterin altında olmalı.
- ➔ Mesaj başlığı seçilmiş olmalı (birden fazla originator varsa).

Bu şartlar sağlanmazsa, hangi şart sağlanmadıysa öncelik sırasına göre kullanıcıya bildirilecek; yanlış mesaj yollanmasını engellemek için “... **adet sms gönderilecek onaylıyor musunuz**” uyarısı, mesaj yollandıktan sonra bilgi verme amaçlı olarak da “... **adet sms gönderildi**” mesajı gösterilecek şekilde **ToReadySMS()** metodu yan sayfadaki şekilde kodlanmıştır. **loadCredits()** metodu ise mesaj yollandıktan sonra arayüzdeki **Kredi miktarını** güncellemek için çağrılmıştır. Oluşturulan uyarı ve bilgi mesajlarının bir kısmı aşağıdaki gibidir.



```

private void toReadySMS() throws Exception
{
    if ( frame.getTaSMS().getText().length() < 764 ) {
        if ( numbers.size() > 0 ) {

            if ( frame.getCbOriginator().getSelectedItem() == null )

                JOptionPane.showMessageDialog( frame, "Lütfen başlık seçiniz..!", "Hata",
JOptionPane.ERROR_MESSAGE );
            else {
                if ( frame.getTaSMS().getText().length() > 0 ) {
                    int confirm = JOptionPane.showConfirmDialog( frame, ( smsCount *
numbers.size() ) + " adet sms gönderilecek. Onaylıyor musunuz?", "UYARI",
JOptionPane.YES_NO_OPTION );
                    if ( confirm == JOptionPane.YES_OPTION ) {
                        String inputText = frame.getTaSMS().getText();
                        String org = (String) frame.getCbOriginator().getSelectedItem();
                        double credit = Util.sendSms( numbers, org, inputText );// harcanan kredi

harcanan kredi

                        JOptionPane.showMessageDialog( frame,credit + " adet kredi
harcandı " + smsCount * numbers.size() + " adet sms gönderildi. ", "Bilgi",
JOptionPane.INFORMATION_MESSAGE );
                        numbers.removeAll( numbers );// her gönderimde numara listesi
sıfırlanıyor
                    }
                } else
                    JOptionPane.showMessageDialog( frame, "Mesaj metni boş
bırakılamaz..!", "Hata",
JOptionPane.ERROR_MESSAGE );
            }
        }
    } else
        JOptionPane.showMessageDialog( frame, "Geçerli numara giriniz..!", "Hata",
JOptionPane.ERROR_MESSAGE );
    }
}

```

KISIM		YAPRAK NO : 29
YAPILAN İŞ	Mesaj Metnindeki Karakter Sayılarının Hesaplanması	TARİH : 05/08/2014

Arayüzde yer alan “**Mesaj Metni**” bölümüne girilen metnin karakter sayısının hesaplanması “**Karakter Sayısı**” bölümüne setlenmesi istendi. **Java'da Strings ve Characters** konuları biraz inceledikten sonra aşağıdaki gibi bir metod oluşturulmuştur.

```

private void getCharacterSize()
{
    int occurrences = 0;

    for ( char c : frame.getTaSMS().getText().toCharArray() ) {
        if( c == '^' )
            occurrences++;
        else if ( c == '{' )
            occurrences++;
        else if ( c == '}' )
            occurrences++;
        else if ( c == '[' )
            occurrences++;
        else if ( c == ']' )
            occurrences++;
        else if ( c == '/' )
            occurrences++;
        else if ( c == '~' )
            occurrences++;
        else if ( c == '!' )
            occurrences++;
        else if ( c == '€' ) occurrences++;

    }
    frame.getLbCharacterNo().setText( "" + ( frame.getTaSMS().getText().length() + occurrences ) );
}
}

```

7 bit gsm alfabesi genişleme tablosunda yer alan ^ {} \ [] ~ € karakterleri *escape'lenerek* yazıldıkları için çift karakter sayılırlar. Bu nedenle girilen mesaj metni karakter dizisine atanıp, döngü ile her karakteri ^ {} \ [] ~ € biri olup olmadığını kontrolü yapıldı. Eğer girilen karakter bu karakterlerden biri ise **occurrences** değişkeni bir artırılıp döngü sonunda girilen metnin karakter uzunluğuna eklenmesi sağlandı.

Daha sonra bu metod **init()** içerisinde **Mesaj Metni** bölümünün **keyReleased()** olayında aşağıdaki gibi çağrılmıştır.

```

frame.getTaSMS().addKeyListener( new KeyListener() {

    @Override
    public void keyTyped( KeyEvent e)
    {

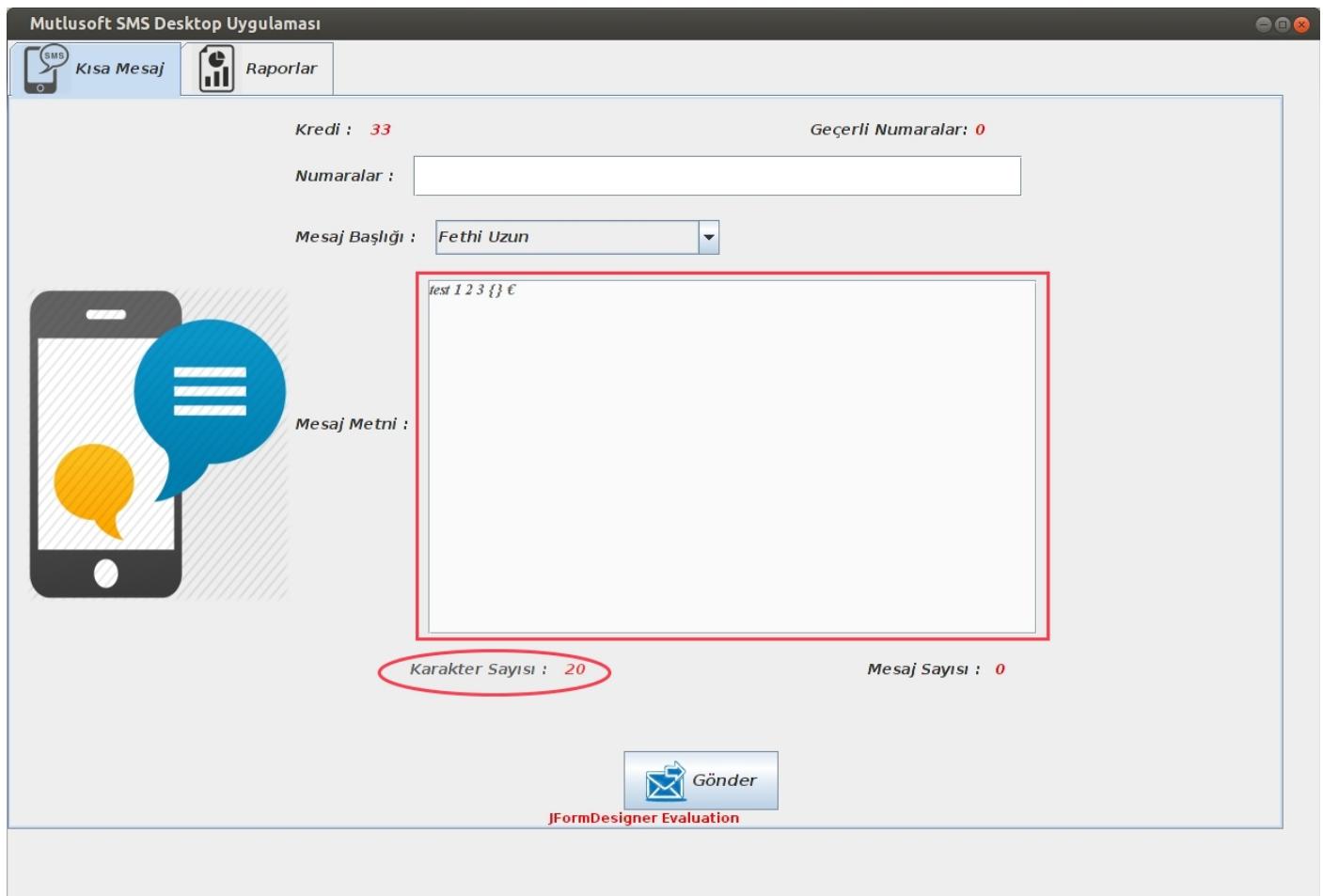
    }

    @Override
    public void keyReleased( KeyEvent e)
    {
        getCharacterSize();
    }
}

```

```
@Override  
public void keyPressed( KeyEvent e)  
{  
}  
}  
};
```

Bu değişikliklerden sonra her karakter girişinde güncellenecek şekilde **“Karakter sayısı:”** bölümü, giren karakterlere bağlı olarak mesaj metninin uzunluğuna setlenmiştir.



KISIM		YAPRAK NO : 30
YAPILAN İŞ	Standart Mesaj Sayısının Hesaplanması	TARİH : 06/08/2014

Arayüzdeki “**Mesaj Sayısı:**” bölümünün setlenmesi istendi. **Mutlucell API'leri** incelendikten sonra aşağıdaki bilgiler gözönüne alındı.

- ➔ 160 karaktere kadar 1 sms olarak gönderilir.
- ➔ 160 karakterden 306 karaktere kadar 2 sms olarak gönderilir.
- ➔ 306 karakterden 459 karaktere kadar 3 sms olarak gönderilir.
- ➔ 459 karakterden 612 karaktere kadar 4 sms olarak gönderilir.
- ➔ 612 karakterden 765 karaktere kadar 5 sms olarak gönderilir.

Sonrasında **Mesaj Sayısı** aşağıdaki gibi setlendi

```

public void keyReleased( KeyEvent e )
{
    countOfChar = frame.getTaSMS().getText().length();
    frame.getLbCharacterNo().setText( "" + countOfChar );

    if ( 0 == countOfChar )
        frame.getLbSMSCountNo().setText( "0" );
    else if ( 0 < countOfChar && countOfChar < 160 )
        frame.getLbSMSCountNo().setText( "1" );

    else if ( 160 < countOfChar && countOfChar < 306 )
        frame.getLbSMSCountNo().setText( "2" );
    else if ( 306 < countOfChar && countOfChar < 459 )
        frame.getLbSMSCountNo().setText( "3" );

    else if ( 459 < countOfChar && countOfChar < 612 ) {
        frame.getLbSMSCountNo().setText( "4" );
        tempTextinput = frame.getTaSMS().getText();

    } else if ( 612 < countOfChar && countOfChar < 765 )
        frame.getLbSMSCountNo().setText( "5" );

    else if ( countOfChar == 745 )
        tempTextinput = frame.getTaSMS().getText();

    else if ( countOfChar > 745 ) {
        frame.getTaSMS().setEnabled( false );
        frame.getTaSMS().setText( tempTextinput );
    }
    frame.getTaSMS().setEnabled( true );
}

```

Ancak sorumluluğunda bulunan mühendis bunun daha kısa ve düzgün bir yoldan yapılmasını ve metodu atanmasını istedi. Ardından **standartSmsCount()** metodu tanımlandı ve “**Mesaj Metni:**” bölümünün **KeyReleased()** olayında çağrıldı.

```
frame.getTaSMS().addKeyListener( new KeyListener() {
...
...
...
public void keyReleased( KeyEvent e)
{
    getCharacterSize();

    standartSmsCount();
}
...
...
}

private void standartSmsCount()
{
    countOfChar = frame.getTaSMS().getText().length();

    if ( 0 == countOfChar ) {

        frame.getLbSMSCountNo().setText( "0" );

        smsCount = 0;

    } else if ( countOfChar < 160 ) {

        frame.getLbSMSCountNo().setText( "1" );

        smsCount = 1;
    } else {

        smsCount = countOfChar / 153;

        if ( smsCount + 1 > 5 ) {

            JOptionPane.showMessageDialog( frame, "Mesaj metni 764 karakteri geçemez..!",

"Error",
            JOptionPane.ERROR_MESSAGE );
        } else
            frame.getLbSMSCountNo().setText( "" + ( smsCount + 1 ) );
    }
}
```

Mutlusoft SMS Desktop Uygulaması

Kredi : **33**

Geçerli Numaralar: **0**

Numaralar :

Mesaj Başlığı : **Fethi Uzun**

 Mesaj Metni :

STAJ 2014
Mutlusoft/Trabzon

Karakter Sayısı : **157**

Mesaj Sayısı : **1**

 Gönder

JFormDesigner Evaluation

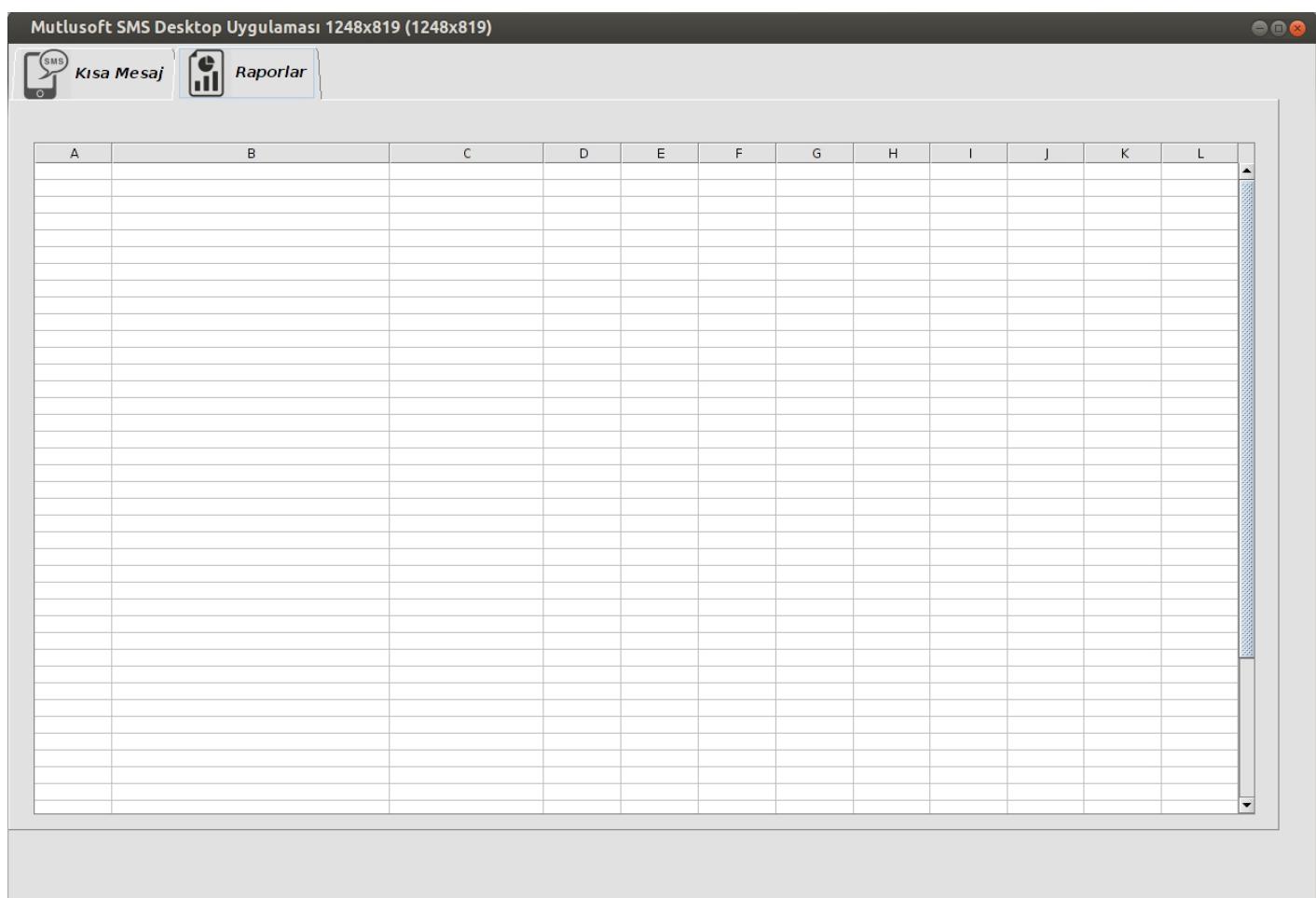
KONTROL SONUCU

KISIM		YAPRAK NO : 31
YAPILAN İŞ	JTable ve getReports() Metodunun Oluşturulması	TARİH : 07/08/2014

Gönderilen mesajların **Paket ID**, **TARİH**, **DURUM**, **BAŞLIK**, **KREDİ**, **ADET**, **BAŞARILI**, **BAŞARI-SIZ**, **ZAMAN AŞIMI**, **BEKLEYEN**, **DİĞER**, **İADE** durumlarını gösterecek şekilde “**Raporlar**” sekme-sine eklenecek şekilde bir rapor oluşturulması, örnek biçim olarak da aşağıdaki örneğin baz alınılması istendi.

paketid	tarih	durum	baslik	kredi	adet	basarili	basarisiz	zamanasimi	bekleyen	diger	iade
254513	27.08.11 , 14:30	2	MUTLUCELL	10.0	10	8	1	1	0	0	2.0
254514	21.08.11 , 11:00	3	MUTLUCELL	20.0	20	16	3	1	0	0	4.0

Bunun için öncelikle **JFormDesigner'dan** yararlanarak arayüzdeki **Raporlar sekmesinde** aşağıdaki şekilde bir **JTable** oluşturuldu.



Ardından **MutluCell API'lerinden** raporla ilgili kısımlar ayrıntılı çalışıldı. **Util sınıfı** içerisinde aşağıdaki **getReports()** metodu oluşturuldu.

```

public static String getReports() throws Exception
{
    String linkRapor = "https://msgw.mutluCell.com/msgw-ws/gtsummaryex";

    String xmlRapor = "<?xml version='1.0' encoding='UTF-8'?><gonrapor ka="" + USERNAME + ""
pwd="" + PWD + "" tarih="" + "2014-06-25 14:30" + "" + "" bitis="" + "2014-08-22 15:20" + "" />";

    String resultReport = HTTPUtil.httpPost( linkRapor, xmlRapor );

    return resultReport;
}

```

XML içerişine gömülü *başlangıç* ve *bitiş tarihini* sürekli manual girmemek için bitiş anlık sistem zamanı; başlangıcı ise iki ay öncesini alacak şekilde aşağıdaki kod oluşturuldu.

```
Calendar calendar = Calendar.getInstance();

SimpleDateFormat date = new SimpleDateFormat( "yyyy-MM-dd hh:mm" );

String now = date.format( calendar.getTime() );

calendar.add( calendar.MONTH, -2 );

String oneMonthAgo = date.format( calendar.getTime() );
```

Burada *API*de yer alan tarih formatı nedeniyle alınan zaman **"YYYY-MM-DD hh:mm"** tarih formatına dönüştürülmesi gerekiyordu. *JAVA'da* zaman ve tarih yapıları araştırıldıktan sonra yukarıdaki bir dönüştürmenin mümkün olduğu görüldü ve kod buna göre düzenlenendi. Böylece sistem zamanından *geriye dönük iki aylık rapor sorgulaması* yapılabilir hale geldi.

Daha sonra çekilen bilgilerin *JTable'a* nasıl setleneceği ile ilgili araştırmalar yapıldı ancak yapısı itibarıyle kullanımı oldukça karmaşık olan *JTable 'a*, veriler bir türlü setlenemedi. Bu yüzden bu çalışma bir gün sonrasında bırakılarak geriye dönük *bug* araması yapıldı.

KISIM		YAPRAK NO : 32
YAPILAN İŞ	Çekilen Bilgileri Rapor'a Bağlayacak Metodların Oluşturulması	TARİH : 08/08/2014

JTable'la ilgili mühendisten bilgi alındıktan sonra *Util* sınıfı içerisindeki *getReports()* metodu aşağıdaki gibi güncellendi.

```

public static Object[][] getReports() throws Exception
{
    Calendar calendar = Calendar.getInstance();

    SimpleDateFormat date = new SimpleDateFormat( "yyyy-MM-dd hh:mm" );

    String now = date.format( calendar.getTime() );

    calendar.add( calendar.MONTH, -2 );

    String oneMonthAgo = date.format( calendar.getTime() );

    try {
        String linkRapor = "https://smssgw.mutlucell.com/smssgw-ws/gtsummaryex";

        String xmlRapor = "<?xml version='1.0' encoding='UTF-8'?><gonrapor ka="" + USERNAME + "" pwd="" + PWD + "" tarih="" + oneMonthAgo + "" + "" bitis="" + now + "" />";

        String resultReport = HTTPUtil.httpPost( linkRapor, xmlRapor );

        String[] reportArray = resultReport.split( "\t\n" );

        Object[][] reportObject = new Object[reportArray.length / 12][12];

        int counter = 0;

        for ( int i = 0; i < reportArray.length / 12; i++ ) {

            for ( int j = 0; j < 12; j++ ) {
                String str = reportArray[counter];
                if (j == 2)
                    reportObject[i][j] = state( str );
                else
                    reportObject[i][j] = str;

                counter++;
            }
        }

        return reportObject;
    } catch ( Exception e ) {
        e.printStackTrace();
    }
    return null;
}

```

JTable model olarak **Object** türünü kabul ettiği için sunucudan gelen veriler **Object dizisine** atandı. Ancak öncesinde sunucudan sekmelere ayrılmış olan veriler **\t** ve **\n**'e göre ayrılarak sıralı bir şekilde **String dizisine**, ardından da **Object dizisine** atanmışlardır.

İkinci sütunda yer alan **2 durumu beklemeye**, **3 durumu gönderildi** ve **4 durumu iptal edildi** anımlarına gelmekte olup kullanıcı için bir anlam ifade etmediği için **State()** metoduyla rakamlar anımlarıyla değiştirilmişlerdir.

```
public static String state( String state)
{
    String str = null;

    switch ( Integer.parseInt( state ) )
    {
        case 2 :
            str = "BEKLEMEDE";
            break;

        case 3 :
            str = "GÖNDERİLDİ";
            break;

        case 4 :
            str = "İPTAL EDİLDİ";
            break;
        default :
            break;
    }

    return str;
}
```

Ardından **Controller sınıfında** aşağıdaki **loadTable()** metodu **thread** yapısı da kullanılarak oluşturulmuştur. Bu metodla varsayılan tablo modeli ve başlıklar oluşturulmuş, veriler tabloya setlenmiştir.

```

private void loadTable()
{
    Thread thread = new Thread() {
        public void run()
        {

            try {
                String[] tableHeaders = { "Paket ID", "TARİH", "DURUM",
                "BAŞLIK", "KREDİ", "ADET", "BAŞARILI",
                "BAŞARISIZ", "ZAMAN AŞIMI", "BEKLEYEN", "DİĞER",
                "İADE" };
                DefaultTableModel model = new
                DefaultTableModel( Util.getReports(), tableHeaders );
                frame.getTbRapor().setModel( model );
                frame.setTableSize();
                DefaultTableCellRenderer centerRenderer = new
                DefaultTableCellRenderer();
                centerRenderer.setHorizontalAlignment( SwingConstants.CENTER );

                for ( int i = 0; i < 12; i++ ) {

                    frame.getTbRapor().getColumnModel().getColumn( i ).setCellRenderer( centerRenderer );
                }
            }

            catch ( Exception e ) {
                e.printStackTrace();
            }
        };
    };
    thread.start();
}

```

Son olarak *loadTable()* metodu *JTable'in Keyreleased()* olayında çağrılmıştır.

```

frame.getTabPanel().addMouseListener( new MouseListener()
{
    @Override
    public void mouseReleased( MouseEvent arg0)
    {
        loadTable();
    }
}

```

Değişikliklerin ardından *rapor*, bir sonraki sayfadaki şekilde yüklenen hale gelmiştir.

Mutlusoft SMS Desktop Uygulaması

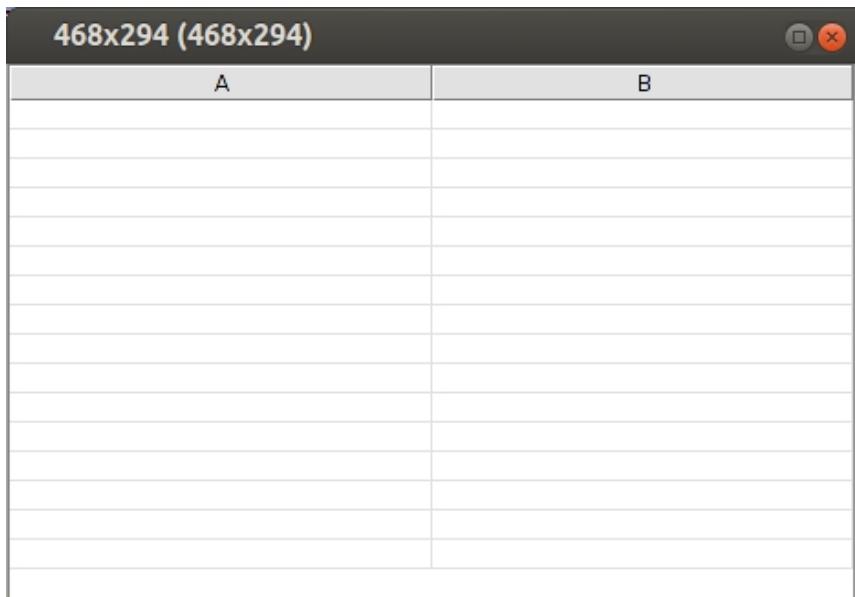


Paket ID	TARİH	DURUM	BAŞLIK	KREDİ	ADET	BAŞARILI	BAŞARISIZ	ZAMAN AŞIMI	BEKLEYEN	DİĞER	İADE
10131254	08.09.14 , 23:25	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
10109213	07.09.14 , 23:38	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9470616	28.07.14 , 11:45	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9470444	28.07.14 , 11:41	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9346022	23.07.14 , 10:56	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313908	21.07.14 , 03:37	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313906	21.07.14 , 03:35	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313904	21.07.14 , 03:31	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313903	21.07.14 , 03:27	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313869	21.07.14 , 02:05	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313844	21.07.14 , 01:13	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313819	21.07.14 , 00:28	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313818	21.07.14 , 00:28	GÖNDERİLDİ	Fethi Uzun	2.0	2	2	0	0	0	0	0.0
9313811	21.07.14 , 00:22	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313771	20.07.14 , 23:57	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9311832	20.07.14 , 14:56	GÖNDERİLDİ	Fethi Uzun	2.0	2	2	0	0	0	0	0.0
9311816	20.07.14 , 14:52	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9311538	20.07.14 , 13:49	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9310089	20.07.14 , 04:03	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9310088	20.07.14 , 04:01	GÖNDERİLDİ	Fethi Uzun	0.0	0	0	0	0	0	0	0.0
9310072	20.07.14 , 03:36	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9309949	20.07.14 , 00:35	GÖNDERİLDİ	Fethi Uzun	2.0	2	2	0	0	0	0	0.0
9309946	20.07.14 , 00:34	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9309923	20.07.14 , 00:12	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9300628	19.07.14 , 03:13	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9300627	19.07.14 , 03:13	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9277158	18.07.14 , 13:37	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9277154	18.07.14 , 13:37	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9262405	18.07.14 , 09:28	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9262373	18.07.14 , 09:22	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9261712	17.07.14 , 23:21	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9261711	17.07.14 , 23:21	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9261683	17.07.14 , 23:02	GÖNDERİLDİ	Fethi Uzun	0.0	0	0	0	0	0	0	0.0
9261682	17.07.14 , 23:02	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9261675	17.07.14 , 22:59	GÖNDERİLDİ	Fethi Uzun	0.0	0	0	0	0	0	0	0.0
9261674	17.07.14 , 22:58	GÖNDERİLDİ	Fethi Uzun	0.0	0	0	0	0	0	0	0.0
9261657	17.07.14 , 22:47	GÖNDERİLDİ	Fethi Uzun	0.0	0	0	0	0	0	0	0.0
9261655	17.07.14 , 22:47	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9248565	17.07.14 01:58	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0

KISIM		YAPRAK NO : 33
YAPILAN İŞ	Ayrıntı Raporunun Arayüz ve Temel Metodlarının Oluşturulması	TARİH : 11/08/2014

Rapor sekmesindeki **Paket ID** sütununa tıklandığında mesajın gönderildiği numara ile mesajın gönderilip gönderilmemiği bilgisini gösteren tablo şeklinde *ayrintı raporu* oluşturmamız istendi.

Öncelikle **ReportDetails** adında bir sınıf oluşturuldu. **JFormDesigner** kullanılarak aşağıdaki gibi bir tablo ile *get()* ve *set()* metodları oluşturuldu.



Mutlucell API'leri incelemek **Paket ID**'leri kullanılarak nasıl sorgulama yapılabileceğini öğrenildi. Ardından **Util sınıfı** içerisinde **Paket ID**'si ile sorgulama yapan, alınan verileri **String dizisine** atayan aşağıdaki metodu yazıldı.

```

public static Object[][] getReportDetails(){

    String xml = "<?xml version='1.0' encoding='utf-8'?><smsrapor ka="" + USERNAME + "" pwd="" +
    PWD + "" id="" + getId()
    + ""/>";

    String link = "https://msgw.mutlucell.com/msgw-ws/gtblkrprtex";

    try {
        String result = HTTPUtil.httppost( link, xml );

        String[] reportArrayRD = result.split( "\n" );

        Object[][] reportObjectRD = new Object[reportArrayRD.length / 2][2];

        return reportObjectRD;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

```

Sonrasında **JTable**'da **Object türü** kullanıldığı için **String dizisi iki boyutlu Object dizisine** aşağıdaki gibi dönüştürüldü.

```
int counter = 0;
for ( int i = 0; i < reportArrayRD.length / 2; i++ ) {

    for ( int j = 0; j < 2; j++ ) {

        String str = reportArrayRD[counter];

        if ( j == 1 ){

            reportObjectRD[i][j] = state( str );
        }
        else
        {
            reportObjectRD[i][j] = str;

        }
        counter++;
    }
}
```

KISIM		YAPRAK NO : 34
YAPILAN İŞ	Ayrıntı Raporuna Verilerin Bağlanması	TARİH : 12/08/2014
Dün ayrintı raporu için arayüz ve Util sınıfında temel metodlar oluşturulmuştu. Bugün ise Rapor sekmesinde Paket ID kısmına tiklandığında çıkacak şekilde ayrintı raporu için geriye kalan işlemlerin yapılması istendi.		
Öncelikle Controller sınıfının yapıcısına yeni oluşturduğumuz reportdetails nesnesi eklendi.		
<pre>public Controller(SmsFrame frame, ReportDetails reportdetails) { this.frame = frame; this.reportdetails=reportdetails; init(); }</pre>		
Ardından MainFrame sınıfındaki controller nesnesinin yapıcısına da reportdetails nesnesi verildi.		
<pre>public static void main(String[] args) { SmsFrame frame = new SmsFrame(); ReportDetails reportdetails=new ReportDetails(); try { Controller <u>controller</u> = new Controller(frame, reportdetails); } catch (Exception e) { e.printStackTrace(); } frame.setVisible(true); }</pre>		
Controller sınıfında varsayılan JTable modelinin yer aldığı ve getReportDetails() metodundan aldığı verileri tabloya ekleyen loadRD() metodu thread yapısı kullanılarak oluşturuldu.		
<pre>private void loadRD() { Thread threadRD = new Thread() { public void run() { try { String[] tableHeadersRD = { "Numara", "Durum"}; DefaultTableModel modelRD = new DefaultTableModel(Util.getReportDetails(), tableHeadersRD); reportdetails.getBtDetails().setModel(modelRD); DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer(); centerRenderer.setHorizontalAlignment(SwingConstants.CENTER); for (int i = 0; i < 2; i++) { reportdetails.getBtDetails().getColumnModel().getColumn(i).setCellRenderer(centerRenderer); } } } } }</pre>		

```

        } catch ( Exception e ) {
            e.printStackTrace();
        }
    };
    threadRD.start();
}

```

Tablonun **Paket ID** sütunlarına tıklayınca, bu sütunlarda yer alan **ID değerini** alıp *Util sınıfındaki id değişkenine* tanımlayan ve **ReportDetails** arayüzüne görünür kılan olaylar ve kodlar, **Controller** sınıfındaki **init()** içerisinde aşağıdaki gibi oluşturulmuştur.

```

frame.getTbRapor().addMouseListener(new MouseListener() {

    @Override
    public void mouseReleased(MouseEvent e)
    {

    }

    @Override
    public void mousePressed(MouseEvent e)
    {

    }

    @Override
    public void mouseExited(MouseEvent e)
    {

    }

    @Override
    public void mouseEntered(MouseEvent e)
    {

    }

    @Override
    public void mouseClicked(MouseEvent e) {

        int row = frame.getTbRapor().rowAtPoint( e.getPoint() );

        int col = frame.getTbRapor().columnAtPoint( e.getPoint() );

        if(col == 0){

            String id=(String) frame.getTbRapor().getModel().getValueAt(row, 0);

            Util.setId(Integer.valueOf(id));

            loadRD();

            reportdetails.setVisible(true);
        }
    }
});

```

Mutlusoft SMS Desktop Uygulaması



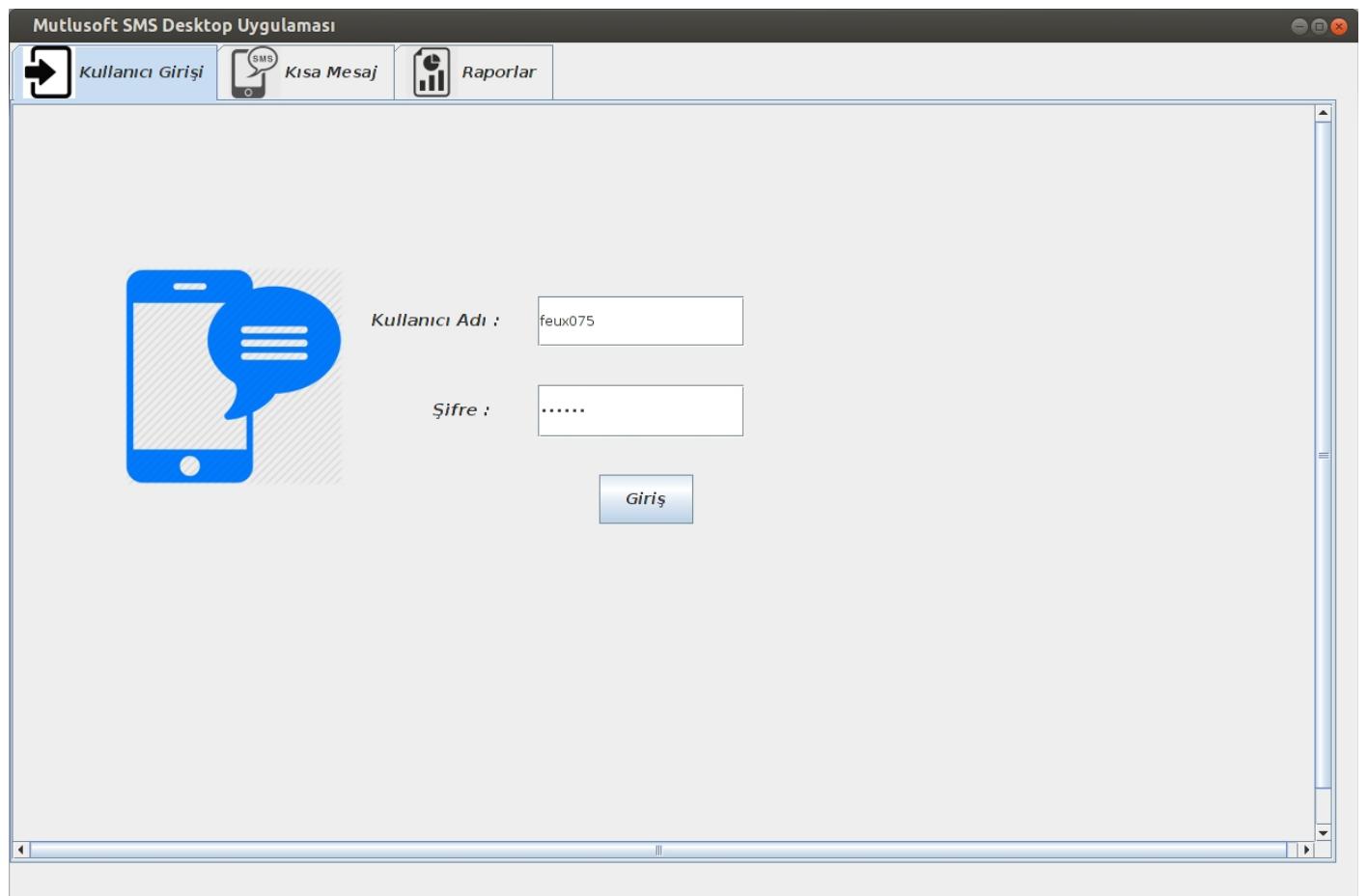
Paket ID	TARİH	DURUM	BAŞLIK	KREDİ	ADET	BAŞARILI	BAŞARISIZ	ZAMAN AŞIMI	BEKLEYEN	DİĞER	İADE
10131254	08.09.14 , 23:25	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
10109213	07.09.14 , 23:38	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9470616	28.07.14 , 11:45	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9470444	28.07.14 , 11:41	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9346022	23.07.14 , 10:56	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313908	21.07.14 , 03:37	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313906	21.07.14 , 03:35	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313904	21.07.14 , 03:31	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313903	21.07.14 , 03:27	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313869	21.07.14 , 02:05	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313844	21.07.14 , 01:13	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313819	21.07.14 , 00:28	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313818	21.07.14 , 00:28	GÖNDERİLDİ	Fethi Uzun	2.0	2	2	0	0	0	0	0.0
9313811	21.07.14 , 00:22	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9313771	20.07.14 , 23:57	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9311832	20.07.14 , 14:56	GÖNDERİLDİ	Fethi Uzun	2.0	2	0	0	0	0	0	0.0
9311816	20.07.14 , 14:5				1	0	0	0	0	0	0.0
9311538	20.07.14 , 13:4				1	0	0	0	0	0	0.0
9310089	20.07.14 , 04:0		Numara			1	0	0	0	0	0.0
9310088	20.07.14 , 04:0		905382430058			0	0	0	0	0	0.0
9310072	20.07.14 , 03:3		905393225813			1	0	0	0	0	0.0
9309949	20.07.14 , 00:3				2	0	0	0	0	0	0.0
9309946	20.07.14 , 00:3				1	0	0	0	0	0	0.0
9309923	20.07.14 , 00:1				1	0	0	0	0	0	0.0
9300628	19.07.14 , 03:1				1	0	0	0	0	0	0.0
9300627	19.07.14 , 03:1				1	0	0	0	0	0	0.0
9277158	18.07.14 , 13:3				1	0	0	0	0	0	0.0
9277154	18.07.14 , 13:3				1	0	0	0	0	0	0.0
9262405	18.07.14 , 09:2				1	0	0	0	0	0	0.0
9262373	18.07.14 , 09:2				1	0	0	0	0	0	0.0
9261712	17.07.14 , 23:2				1	0	0	0	0	0	0.0
9261711	17.07.14 , 23:2				1	0	0	0	0	0	0.0
9261683	17.07.14 , 23:0				0	0	0	0	0	0	0.0
9261682	17.07.14 , 23:0				1	0	0	0	0	0	0.0
9261675	17.07.14 , 22:5				0	0	0	0	0	0	0.0
9261674	17.07.14 , 22:5				0	0	0	0	0	0	0.0
9261657	17.07.14 , 22:47	GÖNDERİLDİ	Fethi Uzun	0.0	0	0	0	0	0	0	0.0
9261655	17.07.14 , 22:47	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0
9248565	17.07.14 01:58	GÖNDERİLDİ	Fethi Uzun	1.0	1	1	0	0	0	0	0.0

KONTROL SONUCU

KISIM		YAPRAK NO : 35
YAPILAN İŞ	Kullanıcı Giriş Sekmesinin Oluşturulması	TARİH : 13/08/2014

Şimdiye dek sabit **kullanıcı adı, parola** ile kullanılan projeye yeni bir **kullanıcı giriş sekmesi** eklenmesi istendi. **mutlucell.com** üzerinde hesabı olan herhangi bir **kullanıcı**, program arayüzünden **kullanıcı adı** ve **şifresini** girerek **toplu mesaj** yollayabilecek.

Öncelikle **JFormDesigner** ile aşağıdaki sekme oluşturuldu.



Util sınıfı içerisinde girilen **kullanıcı adı** ve **şifrenin** doğruluğunun kontrolünü yapan aşağıdaki **validateUsername()** metodu oluşturuldu.

```
public static String validateUsername() throws Exception
{
    String linkCredits = "https://smgw.mutlucell.com/smgw-ws/gtcrdtx";
    String xmlCredits = "<?xml version='1.0' encoding='utf-8'?><smskredi ka="" + USERNAME + ""
pwd="" + PWD + "/>";
    String result = HTTPUtil.httppost( linkCredits, xmlCredits );

    return result;
}
```

Controller sınıfı içerisinde **Giriş** butonuna tıklandığında gerçekleşmesi gereken olaylar ve çağrılmaları gereken metodlar için yan sayfadaki gibi bir kodlama yapılmıştır.

```

frame.getBtuser().addActionListener(new ActionListener()
{
    @Override
    public void actionPerformed(ActionEvent arg0) {
        Util.USERNAME=frame.getTfusername().getText();
        Util.PWD=String.valueOf(frame.getPfpassword().getPassword());

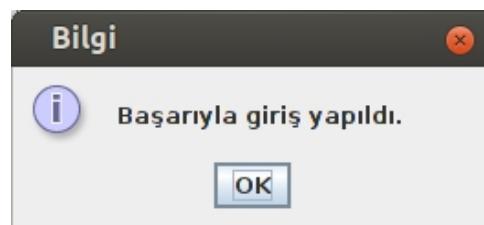
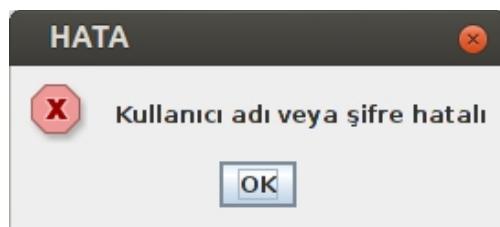
        try {
            if(Util.validateUsername().startsWith("$")){
                JOptionPane.showMessageDialog(frame, "Başarıyla giriş yapıldı.", "Bilgi",
JOptionPane.INFORMATION_MESSAGE);

            }
            else{
                JOptionPane.showMessageDialog(frame, "Kullanıcı adı veya şifre hatalı",
"HATA", JOptionPane.ERROR_MESSAGE);
            } catch (HeadlessException e) {

                e.printStackTrace();
            } catch (Exception e) {

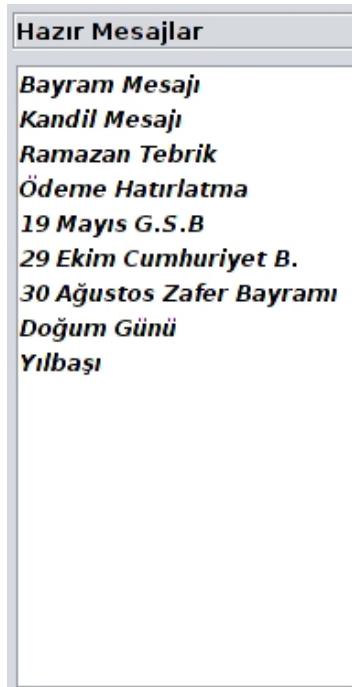
                e.printStackTrace();
            }
        });
    }
});
```

Burada girilen bilgilerin doğruluğuna bağlı olarak kullanıcıya aşağıdaki uyarı veya bilgi mesajları verilmiştir.



KISIM		YAPRAK NO : 36
YAPILAN İŞ	Hazır Mesajlar Bölümünün Arayüze Eklenmesi	TARİH : 14/08/2014

Projeye resmi bayramı kandil, doğum günü vb.ler için **hazır mesajlar** bulunan bir bölüm eklememiz istendi. Gerekli araştırmalar yapıldıktan sonra **Jlist** kullanılarak aşağıdaki bölüm arayüze eklendi.



Ardından **Controller** sınıfında listedeki mesajlara tıklayınca “**Mesaj Metni**” kısmına eklenmesi gereken **hazır mesajlar** hazırlandı ve **ListSelectionListener'a** aşağıdaki gibi eklendi.

```
frame.getLihazirmsj().addListSelectionListener(new ListSelectionListener() {
    @Override
    public void valueChanged(ListSelectionEvent e) {
        if(frame.getLihazirmsj().getSelectedIndex(0)){frame.getTaSMS().setText("Bayramınız
        kutlu olsun.");
    }
    if(frame.getLihazirmsj().getSelectedIndex(1)){frame.getTaSMS().setText("Sayın
        {isim}, Kandilinizi kutlar hayirlara vesile olması dilegiyle sevgi ve saygılarımı sunarım.");
    }
    if(frame.getLihazirmsj().getSelectedIndex(2)){frame.getTaSMS().setText("Sayın
        {isim}, Ramazaninizin bereket, mutluluk, saglik ve afiyet getirmesini dileriz. Nice Ramazanlara");
    }
    if(frame.getLihazirmsj().getSelectedIndex(3)){frame.getTaSMS().setText("Sayın
        {isim}, {Ödeme Günü} tarihinde {Tutar} ödemeniz vardır. Zamanında yaptığınız ödemeleriniz için
        teşekkür ederiz.");
    }
    if(frame.getLihazirmsj().getSelectedIndex(4)){frame.getTaSMS().setText("Modern
        Türkienenin Genc Mimarları, Genclik Bayramınız Kutlu Olsun.");
    }
    if(frame.getLihazirmsj().getSelectedIndex(5)){frame.getTaSMS().setText("Sayın
        {isim}, Cumhuriyet bayramınızı tebrik eder, ulusumuzun ilelebet varlığını temenni ederiz. Nice
        Cumhuriyet bayramlarına");}
    }
}
```

```

        if(frame.getLihazirmsj().isSelectedIndex(6)){frame.getTaSMS().setText("Yüce
ulusumuzun 30 agustos zafer bayramini kutlar, cumhuriyetimizin ilelebet varligini temenni ederiz.");

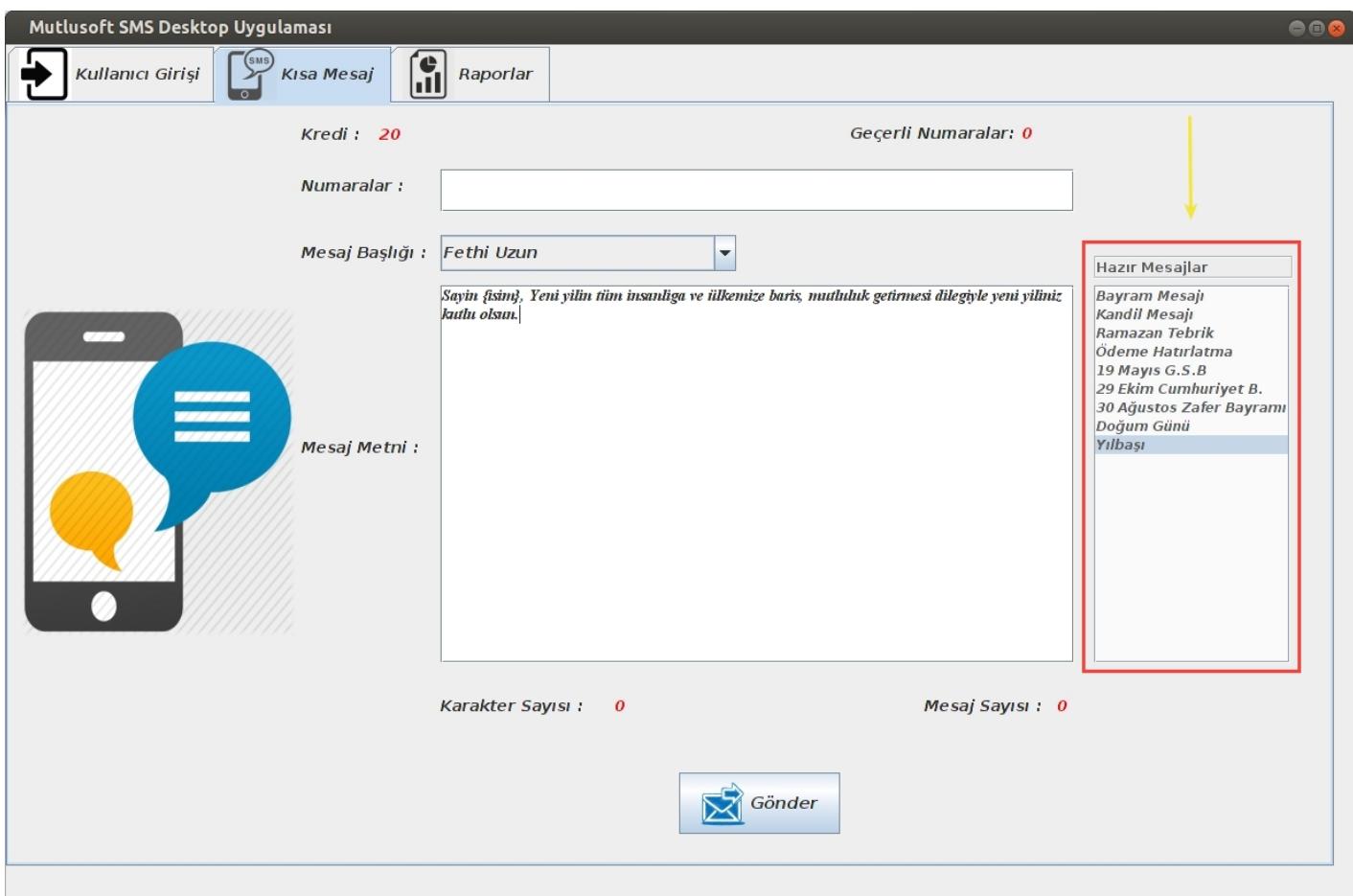
    }
    if(frame.getLihazirmsj().isSelectedIndex(7)){frame.getTaSMS().setText("Sayin
{isim}, Yasaminizda güzel yillar, mutlu yarinlar sizinle olsun. MUTLU YILLAR!");

    }
    if(frame.getLihazirmsj().isSelectedIndex(8)){frame.getTaSMS().setText("Sayin
{isim}, Yeni yilin tüm insanliga ve ülkemize baris, mutluluk getirmesi dilegiyle yeni yiliniz kutlu
olsun.");

    }

}
});
```

Ve proje aşağıdaki gibi son halini aldı.



KISIM		YAPRAK NO : 37
YAPILAN İŞ	Arayüze Rehber Eklenmesi	TARİH : 15/08/2014
Projeye basit bir Rehber eklememiz istendi. JFormDesigner'la , Jlist kullanarak arayüzde basit bir Rehber oluşturuldu ve aşağıdaki gibi kodlandı.		
<pre>frame.getLiRehber().addListSelectionListener(new ListSelectionListener() { String annem="05383169028"; String fethi="05382430058"; String ahmet="05366401661"; String hakan="05393225813"; String burhan="05392442951"; String hfs="05427613966"; String ferhat="05548328881"; String teyzem="05422111507"; String gokan="05436688385"; @Override public void valueChanged(ListSelectionEvent arg0) { String temptext=frame.getTfNumbers().getText(); if(frame.getLiRehber().getSelectedIndex(0)){ frame.getTfNumbers().setText(temptext+","+fethi); numbers.add(fethi); } if(frame.getLiRehber().getSelectedIndex(1)){ frame.getTfNumbers().setText(temptext+","+hakan); numbers.add(hakan); } if(frame.getLiRehber().getSelectedIndex(2)){ frame.getTfNumbers().setText(temptext+","+hfs); numbers.add(hfs); } if(frame.getLiRehber().getSelectedIndex(3)){ frame.getTfNumbers().setText(temptext+","+annem); numbers.add(annem); } if(frame.getLiRehber().getSelectedIndex(4)){ frame.getTfNumbers().setText(temptext+","+ferhat); numbers.add(ferhat); } if(frame.getLiRehber().getSelectedIndex(5)){ frame.getTfNumbers().setText(temptext+","+teyzem); numbers.add(teyzem); } if(frame.getLiRehber().getSelectedIndex(6)){ frame.getTfNumbers().setText(temptext+","+burhan); numbers.add(burhan); } if(frame.getLiRehber().getSelectedIndex(7)){ frame.getTfNumbers().setText(temptext+","+ahmet); numbers.add(ahmet); } if(frame.getLiRehber().getSelectedIndex(8)){ frame.getTfNumbers().setText(temptext+","+gokan); numbers.add(gokan); } }); });</pre>		

Mutlusoft SMS Desktop Uygulaması

Kullanıcı Girişи  Kısa Mesaj  Raporlar 

Kredi : **3.0** Geçerli Numaralar: **3**

Numaralar : **05382430058,05383169028,05422111507**

Mesaj Başlığı : **Fethi Uzun**

Mesaj Metni : **Modern Türkîyenin Genc Mimarları, Genclik Bayramınız Kutlu Olsun.**

REHBER

- Fethi Erdinç UZUN
- Hakan AYDIN
- Haifa Pınar UYGUN
- Annem **Annem**
- Ferhat
- Teyzem
- Burhan GÜNEYDİN
- Ahmet Faruk YAVUZ
- Gökhan ÇETİN

Hazır Mesajlar

- Bayram Mesajı
- Kandil Mesajı
- Ramazan Tebrik
- Ödeme Hatırlatma
- 19 Mayıs G.S.B **19 Mayıs G.S.B**
- 29 Ekim Cumhuriyet B.
- 30 Ağustos Zafer Bayramı
- Doğum Günü
- Yılbaşı

Karakter Sayısı : **65** Mesaj Sayısı : **1**

 **Gönder**

Arayüzün son hali

KISIM		YAPRAK NO : 38
YAPILAN İŞ	Rehbere Yeni Kayıt Ekleme	TARİH : 18/08/2014

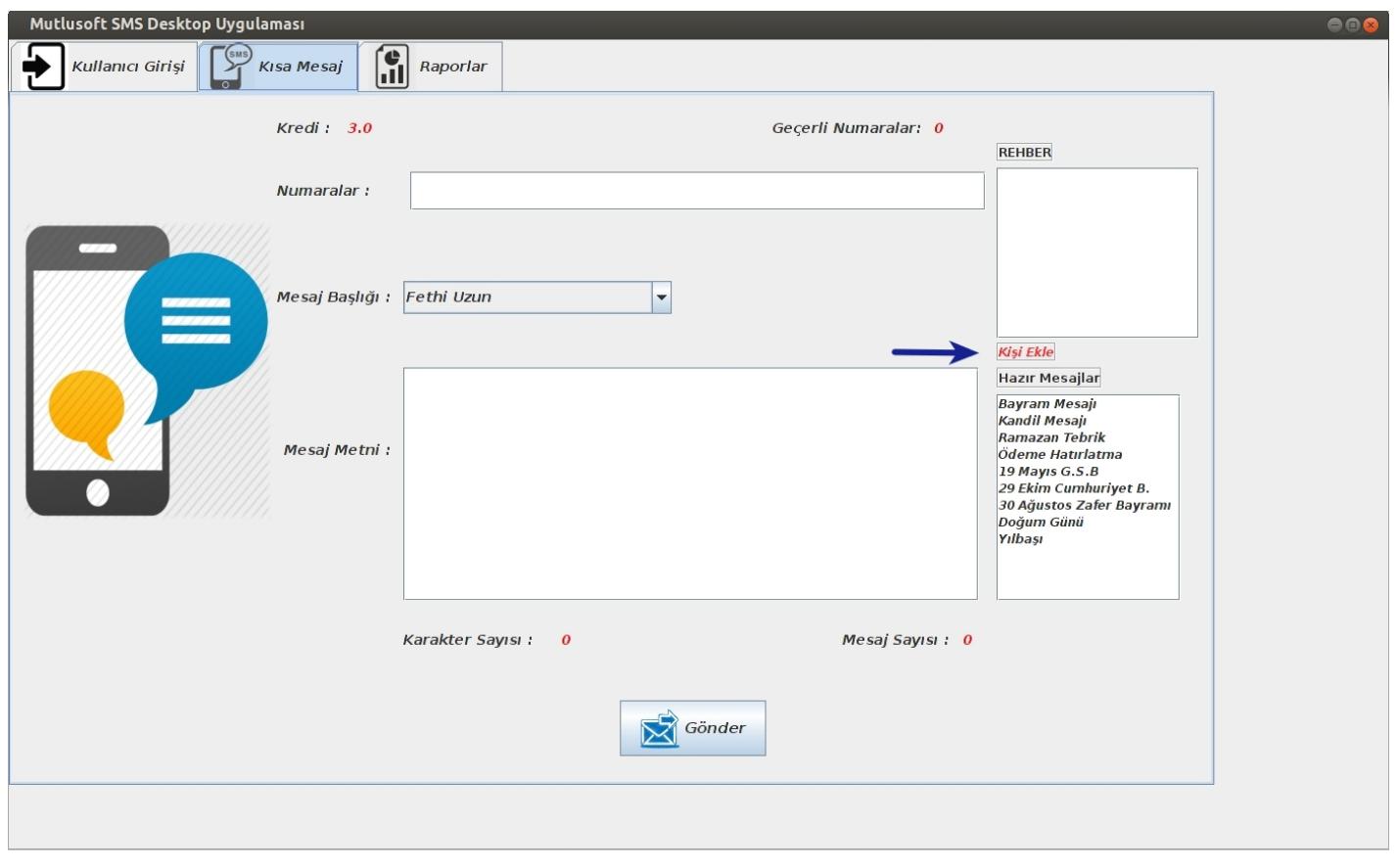
Daha önce oluşturulan **Rehber** bölümüne kullanıcının müdahalesi mümkün değildi. Kullanıcının rehbere istediği kayıtları ekleyebilmesi için arayüze “**Kişi Ekle**” bölümü eklememiz istendi. Bunun üzerine aşağıdaki gibi bir **JDialog** oluşturuldu.



Ardından **varsayılan liste modeli** oluşturan **setListModel()** metodu yazıldı ve **Controller** sınıfının **init()** metodunda çağrıldı.

```
private void setListModel(){
    listModel = new DefaultListModel();
    JList list = new JList(listModel);
    frame.getLiRehber().setModel(listModel);
}
```

Arayüze **Kişi Ekle** butonu eklendi.



Yeni Kayıt penceresine girilen değerler, **Kişi Ekle** butonuna basıldıktan sonra geçici değişkenlere atanması ve **Rehber** bölümüne yeni kayıtın ismini ekleyecek kod aşağıdaki gibi oluşturuldu.

```
frame.getLbAddContact().addMouseListener(new MouseListener() {  
  
    @Override  
    public void mouseReleased(MouseEvent arg0) {  
  
        addContact.setVisible(true);  
  
        addContact.getOkButton().addActionListener(new ActionListener() {  
  
            @Override  
            public void actionPerformed(ActionEvent arg0) {  
  
                String tempAd=addContact.getTfAd().getText();  
                String tempSoyad=addContact.getTfSoyad().getText();  
                String tempNo=addContact.getTfNo().getText();  
  
                noList.add(tempNo);  
                listModel.addElement(tempAd+" "+tempSoyad);  
  
                addContact.setVisible(false);  
                addContact.getTfAd().setText(null);  
                addContact.getTfSoyad().setText(null);  
                addContact.getTfNo().setText(null);  
  
            }  
        });  
    }  
  
    @Override  
    public void mousePressed(MouseEvent arg0) {}  
  
    @Override  
    public void mouseExited(MouseEvent arg0) {}  
  
    @Override  
    public void mouseEntered(MouseEvent arg0) {}  
  
    @Override  
    public void mouseClicked(MouseEvent arg0) {}  
});
```

noList isimli kullanıcının girdiği numaraların eklendiği listedeki numaraların ilgili liste elemanına tıklandığında “**Numaralar**” bölümüne eklenmesi için **Rehber** bölümünün **AddListSelectionListener()** kısmı bir sonraki sayfadaki gibi güncellenmiştir.

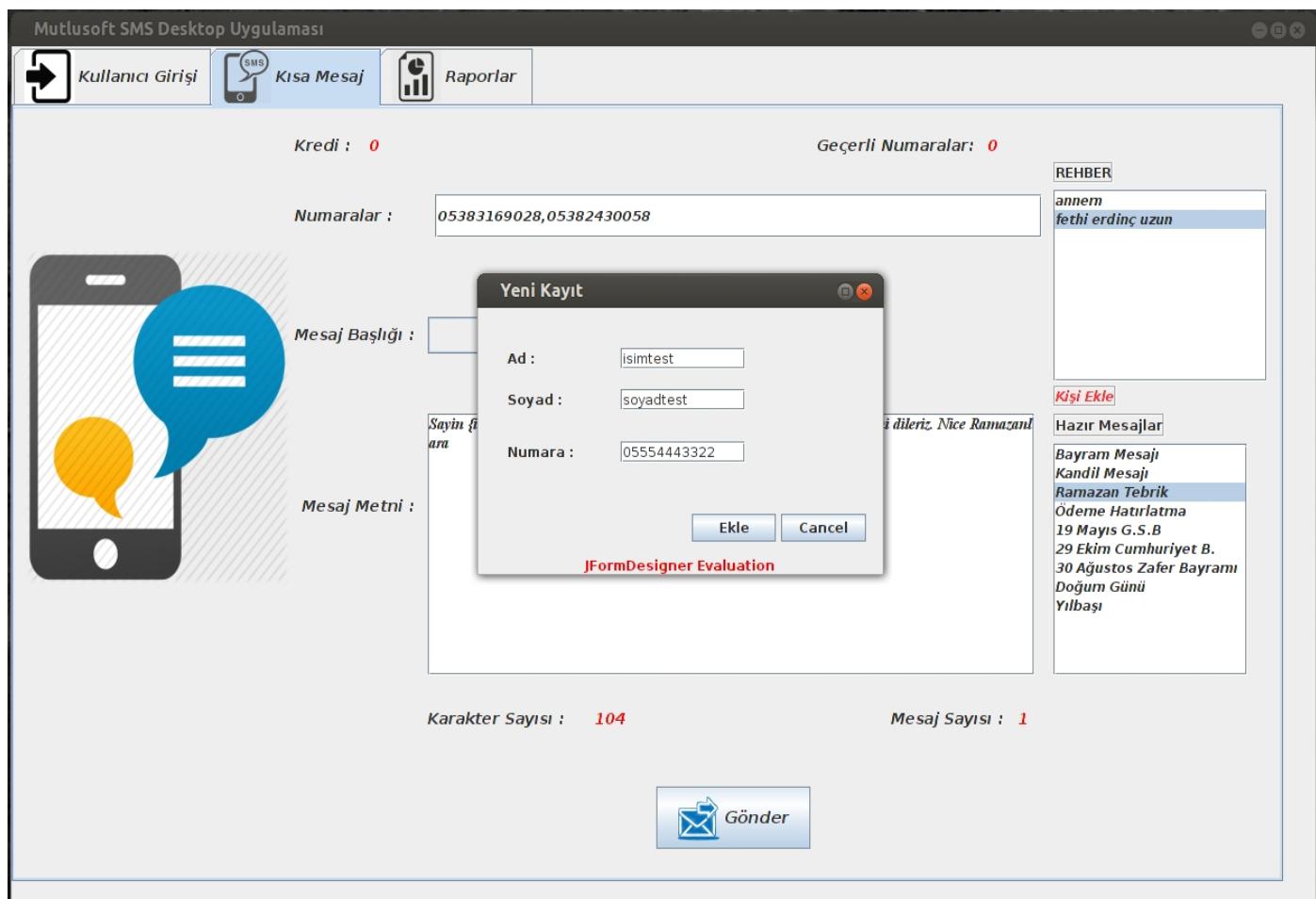
```

frame.getLiRehber().addListSelectionListener(new ListSelectionListener() {

    @Override
    public void valueChanged(ListSelectionEvent arg0) {

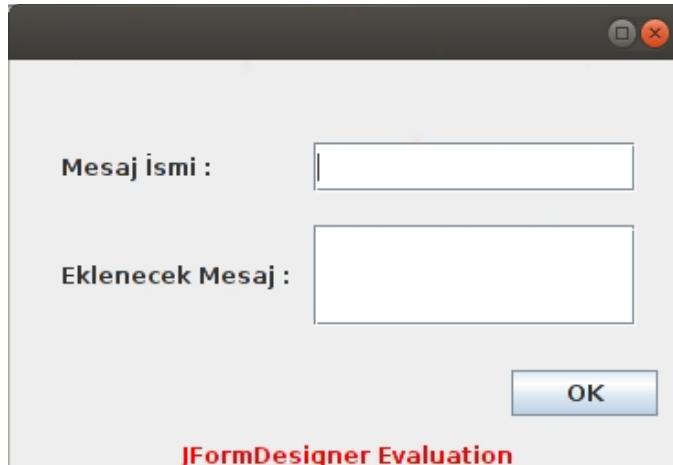
        String temptext=frame.getTfNumbers().getText();
        int index=frame.getLiRehber().getSelectedIndex();
        if(!frame.getTfNumbers().getText().isEmpty()){
            frame.getTfNumbers().setText(temptext+","+noList.get(index));
            numbers.add(noList.get(index));
            temptext=null;
        }
    else{
        frame.getTfNumbers().setText(temptext+noList.get(index));
        numbers.add(noList.get(index));
        temptext=null;
    }
}
});
```

Bu değişiklikler sonrasında program kullanıcının rehbere kendi kayıtlarını eklemesine elverişli hale gelmiş oldu.



KISIM		YAPRAK NO : 39
YAPILAN İŞ	Yeni Hazır Mesaj Ekleme	TARİH : 19/08/2014

Hazır Mesajlar bölümüne kullanıcının da kendi hazır mesajlarını eklemesine olanak sağlamak için araya **Mesaj Ekle** bölümü eklememiz istendi. Bunun üzerine öncelikle Aşağıdaki **JDialog** tasarlandı.



Hazır Mesajlar bölümüne **Mesaj Ekle** adında bir **JLabel** eklendi. Ardından **varsayılan liste modeli** ile hazır mesajları arayüze ekleyen **setMsjList()** metodu **Controller** sınıfı içerisinde aşağıdaki gibi oluşturuldu ve **init()** metodu içerisinde çağrıldı.

```

private void setMsjList() {
    msjListModel = new DefaultListModel();
    JList list = new JList(msjListModel);
    frame.getLihazirmsj().setModel(msjListModel);

    msjList.add("Bayramınız kutlu olsun.");
    msjList.add("Sayın {isim}, Kandilinizi kutlar hayırlara vesile olması dileğiyle sevgi ve saygılarımı sunarım.");
    msjList.add("Sayın {isim}, Ramazaninizin bereket, mutluluk, saglik ve afiyet getirmesini dileriz. Nice Ramazanlara");
    msjList.add("Sayın {isim}, {Ödeme Günü} tarihinde {Tutar} ödemeniz vardır. Zamanında yaptığınız ödemeleriniz icin tesekkür ederiz.");
    msjList.add("Modern Türkiyenin Genc Mimarları, Genclik Bayramınız Kutlu Olsun.");
    msjList.add("Sayın {isim}, Cumhuriyet bayramınızı tebrik eder, ulusumuzun ilelebet varlığını temenni ederiz. Nice Cumhuriyet bayramlarına");
    msjList.add("Yüce ulusumuzun 30 agustos zafer bayramını kutlar, cumhuriyetimizin ilelebet varlığını temenni ederiz.");
    msjList.add("Sayın {isim}, Yaseminizda güzel yıllar, mutlu yarınlar sizinle olsun. MUTLU YILLAR!");
    msjList.add("Sayın {isim}, Yeni yılın tüm insanlığa ve ülkemize barış, mutluluk getirmesi dileğiyle yeni yılınız kutlu olsun.");

    msjListModel.addElement("Bayram Mesajı");
    msjListModel.addElement("Kandil Mesajı");
    msjListModel.addElement("Ramazan Kutlaması");
    msjListModel.addElement("Ödeme Hatırlatma");
    msjListModel.addElement("G.S.B");
    msjListModel.addElement("Cumhuriyet Bayramı");
    msjListModel.addElement("30 Ağustos Zafer B.");
    msjListModel.addElement("Doğum Günü Mesajı");
    msjListModel.addElement("Yılbaşı");
}

```

Son olarak **Mesaj Ekle** etiketine tıklandığında eklenecek mesaj ile mesaj ismini alan **dialog penceresini** çağırın ve **OK** butonuna tıklandığında bunları setleyen olaylar dizisi aşağıdaki gibi oluşturuldu.

```
frame.getLbAddHazirMesaj().addMouseListener(new MouseListener() {

    @Override
    public void mouseReleased(MouseEvent arg0) {

        addTemplate.setVisible(true);

        addTemplate.getBtOkTemplate().addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent arg0) {

                String tempMsjName=addTemplate.getTfMsjName().getText();

                String tempMsj=addTemplate.getTaMsj().getText();

                msjListModel.addElement(tempMsjName);

                msjList.add(tempMsj);

                addTemplate.setVisible(false);

                addTemplate.getTfMsjName().setText(null);

                addTemplate.getTaMsj().setText(null);

            }
        });
    }

    @Override
    public void mousePressed(MouseEvent arg0) {

    }

    @Override
    public void mouseExited(MouseEvent arg0) {

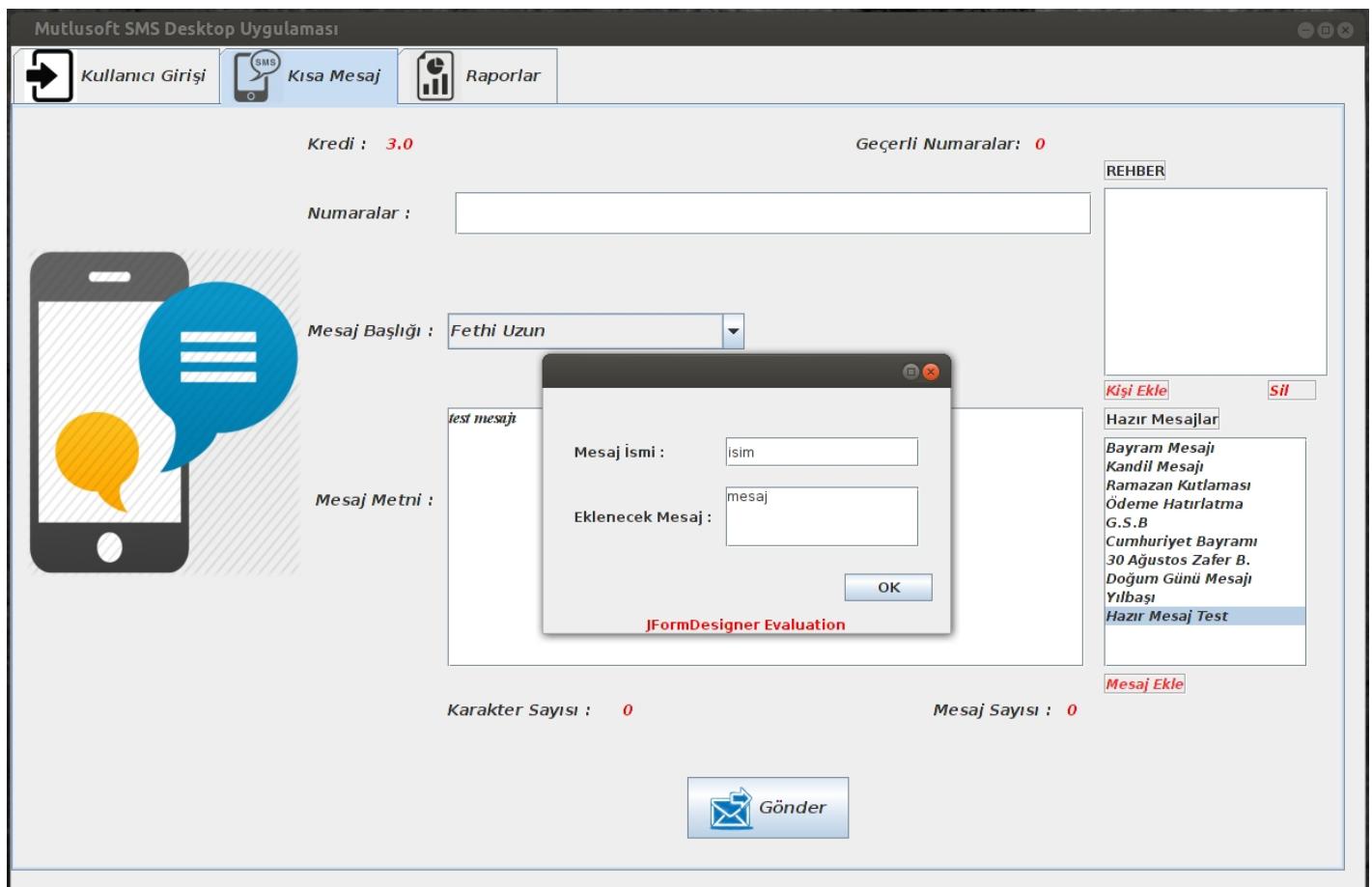
    }

    @Override
    public void mouseEntered(MouseEvent arg0) {

    }

    @Override
    public void mouseClicked(MouseEvent arg0) {

    }
});
```



KONTROL SONUCU

KISIM		YAPRAK NO : 40
YAPILAN İŞ	Rehber ve Hazır Mesajlar için Silme İşlevi	TARİH : 20/08/2014

Rehber ve Hazır Mesajlar bölümlerine *silme işlevi* kazandırmamız istendi. *JList'te silme olayları* araştırıldıktan sonra *rehber için silme işlemi* aşağıdaki kodlarla gerçekleştirildi. **Rehber** altındaki *Sil* etiketine tıklanınca seçili olan elementin *indeksi* alınıp o indeksteki *liste modeli üyesi* ile numaraları tutan *noList* listesinden aynı indeksteki elementi silecek bir yapı oluşturuldu ve **Controller** sınıfının *init()* metodunda çağrıldı.

```
frame.getLbDelete().addMouseListener(new MouseListener() {

    @Override
    public void mouseReleased(MouseEvent e) {

        int indexOf=frame.getLiRehber().getSelectedIndex();

        if(indexOf!=-1)
        {
            noList.remove(indexOf);

            listModel.remove(frame.getLiRehber().getSelectedIndex());

        }
    }

    @Override
    public void mousePressed(MouseEvent e) {}

    @Override
    public void mouseExited(MouseEvent e) {}

    @Override
    public void mouseEntered(MouseEvent e) {}

    @Override
    public void mouseClicked(MouseEvent e) {}
});
```

Ardından aynı yapı **Hazır Mesajlar** için de aşağıdaki gibi uygulandı.

```
frame.getLbTempMesajDelete().addMouseListener(new MouseListener() {

    @Override
    public void mouseReleased(MouseEvent e) {

        int indexOfTemplateMsj=frame.getLihazirmsj().getSelectedIndex();

        if(indexOfTemplateMsj!=-1){

            msjList.remove(indexOfTemplateMsj);

            msjListModel.remove(frame.getLihazirmsj().getSelectedIndex());

        }
    }
});
```

```

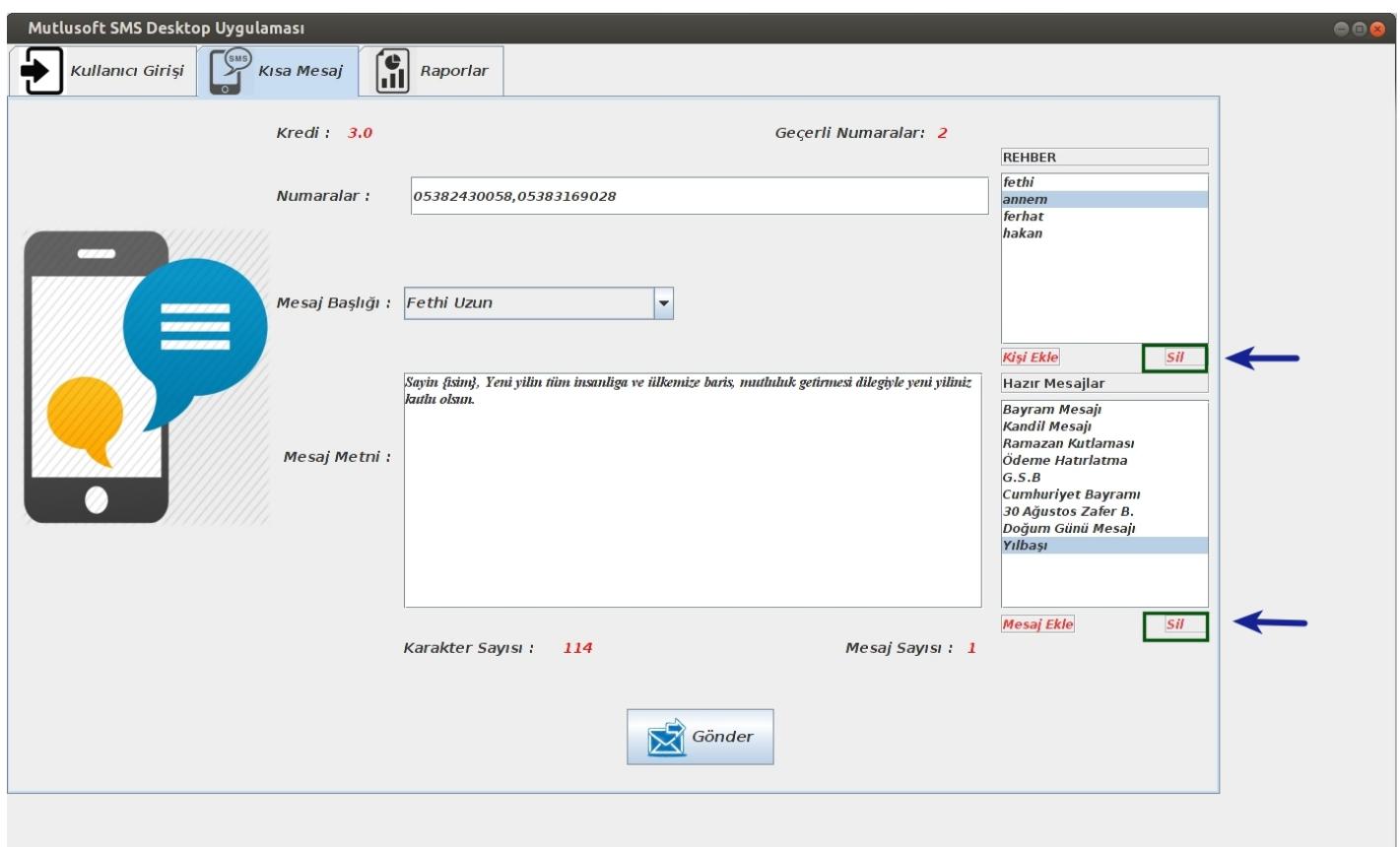
@Override
public void mousePressed(MouseEvent e) {}

@Override
public void mouseExited(MouseEvent e) {}

@Override
public void mouseEntered(MouseEvent e) {}

@Override
public void mouseClicked(MouseEvent e)
);

```



Bu son eklemelerle birlikte ***Mutlusoft SMS Desktop Uygulaması*** tamamlanmış oldu.