Harbiye projesinde bir chat ekranın yapılması istenildi. Bunun üzerine chat ekranı için çalışmalara başlanıldı.

Chat ekranının tasarımı aşağıda anlatılan şekilde yapıldı.

* Chat ekranı haritanın sol üst köşesinde açılacak şekilde görüntülenmesi sağlandı.
* Chat ekranının sol bölümünde networkde bulunulan kullanıcılar listview kullanılarak listelendi.
* Sağ bölümünde ise seçilen kullanıcıya gönderilen ve alınan mesajların aktığı bir alan ayarlandı.

Kullanıcıların bulunduğu listview itemları Tab olacak şekilde ayarlandı. Her tab içerisine kullanıcının urn bilgisi ve name bilgisine yer verildi.



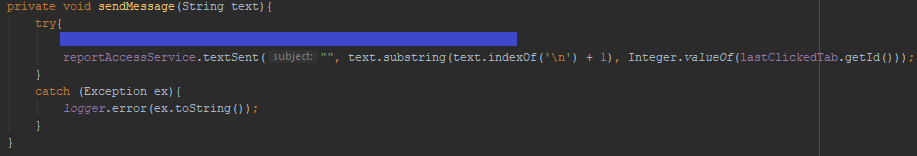
Yukarıdaki kodda bulunan Platform.runlater kullanılmıştır. Bunun nedeni aşağıda açıklanmıştır.

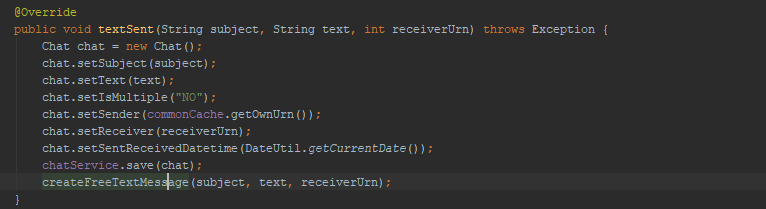
Bilindiği gibi çok uzun zaman alan işlemleriJavaFX main thread’de çalıştırmak uygun bir durum değildir. Sebebi ise bildiğiniz gibi çok uzun süren bir işlem arayüzü kilitleyecek ve kullanıcının program bozuk deyip kapatmasına yani kullanamamasına sebep olacaktır. Tabi bu hiç de istenilen bir durum değildir. Bu durumu çözmek için uzun süren işlemler arka planda çalıştırılmalıdır ki programımız kilitlenmiş havasına girip kapatılmaya sebep olmasın. Arka planda çalıştırılan bir thread oluşturdukdan sonra “UI Thread” (Javafx Application Thread) deki bir label’e işlemler ile ilgili bilgi yazdırmak istenebilir. Yani bir background thread’den UI thread’e veri aktarmak istenebilir. Bu durumda Platform.runlater yapısı kullanılır.

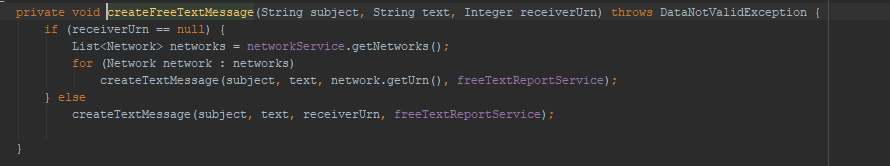
Platform.runLater(new Runnable(){ public void run(){ ana threaddeki elamana burdan bilgi aktarın. }});

**ChatPane’ de FreeText Message Gönderilmesi**

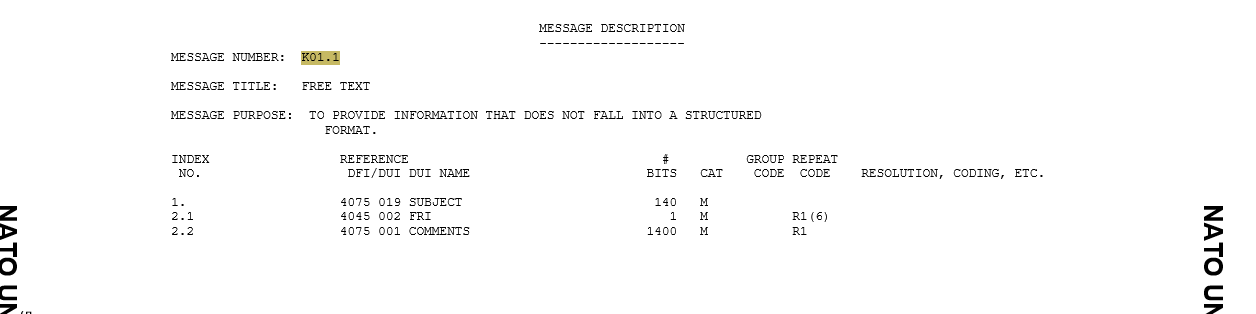
ChatPane’de yazılan bir mesajın karşı tarafdaki alıcıya göndeilmesi için Nato standartlarında geçen Free Text Message kullanıldı.







Nato standartlarındaki Free Text Message Raporunun Vmf formatının bir bölümü aşağıdaki resimde gösterilmiştir.

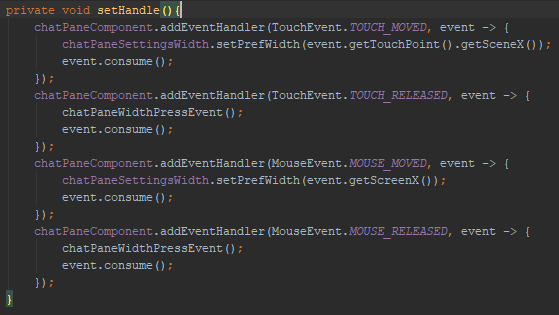


**ChatPane Ekranın Genişliğinin Sürükle Bırak ile Değiştirilmesi**

Chat ekranının genişliğinin gerektiğinde sürükle bırak ile değiştirebilmesi için çalışmalar yapıldı.

Öncelike chatPane’nin en dışına bir Hbox eklendi. Bunun eklenmesinin sebebi arkadaki alanın genişlik değerinin alınabilmesidir. Bu hbox’ın visible özelliği ilk açılışda false olarak ayarlanmıştır. Daha sonra chatPane üzerine 3 saniye basılı tutuldukdan sonra hbox görünür hale gelmektedir.

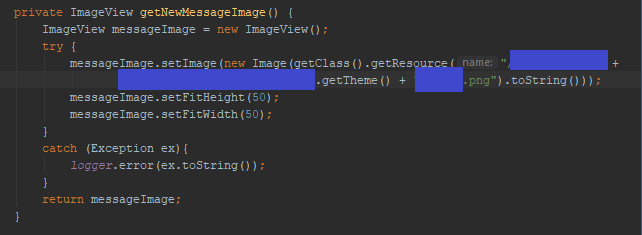
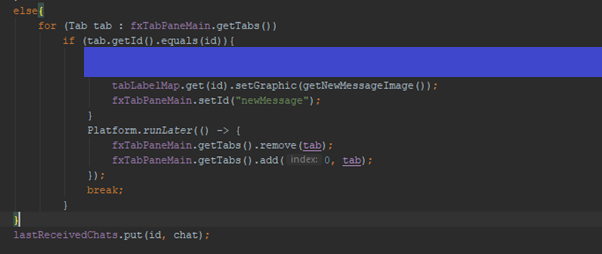
En dışa eklenen hbox’a 4 adet addEventHandler yazımıştır. Bunlardan iki tanesi touch, diğer iki taneside mouse için yazılmıştır.



Yukarıdaki fonksiyon içeriğindende anlaşıldığı gibi, hbox’ın sürüklenip bırakıldığı alanın lokasyon değerleri bir değişkene aktarılmıştır. Daha sonrada diğer gerekli alanlarada atamalar yapılıp genişlik değerinin dinamik bir şekilde ayarlanılması sağlanmıştır.

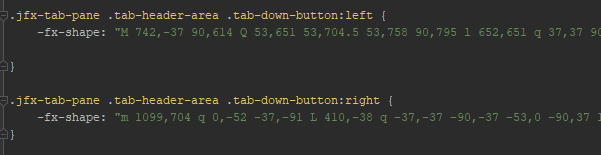
**ChatPane Ekranın’da Yeni Mesaj Geldiğinde Gelen Mesajın En Üstte Taşınması ve Resminin Değişmesi**

Aşağıdaki kod bloğunda anlaşıldığı üzere, yeni bir mesaj geldiği anda, listview’ın sıfırıncı indexine eklenir. Ayrıca getNewMessage fonksiyonu çağrılarak yeni mesajın resminin diğerlerinden farklı olması sağlanmıştır.

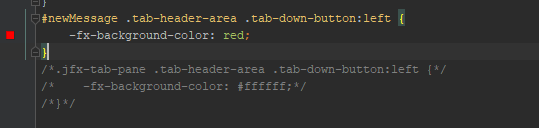


**ChatPane Ekranın’da Yeni Mesaj Geldiğinde Tab Pane Okların Css ile Değiştirilmesi**

Yeni mesaj geldiğinde tabPane’nin yukarı ok ikonunun renginin kırmızı olarak değiştirilmesi için çalışmalar yapılmıştır. Bunun için internet üzerinden herhangi bir kaynak bulunamamıştır. Bunun üzerine JFXTabPane’nin base css’i incelenmiştir. İncelemeler sonucunda aşağıdaki kod bloklarında verilen css’ler bulunmuştur.



Bulunan çözüm kendi css’imize aşağıdaki şekilde eklenmiştir.



Görüldüğü üzere css yazılırken id verilerek yazılmıştır. Bunun sebebi ise yeni mesaj geldiğinde tabPane’nin id değeri newMessage olarak ayarlanmaktır. Bu sayede yukarıda belirtilen css kullanılmaktadır. Diğer durumlarda bir id değeri verilmektedir. Böylece diğer durumlarda varsayılan css’ler kullanılmaktadır.