1) Giriş

Soket programlama ödevi web tabanlı chat uygulama geliştirilmiştir. Steganografi ödevi için masa üstü uygulama geliştirilmiştir. Geliştirilen uygulamalarda farklı teknolojiler kullanılmıştır. Bunlar;

- PHP ve Java web soket
- MongoDB ve MySql
- PHP ve Java - Spring Boot servis.
- Jquery
- Spring Cache ile Hazelcast
- JavaFX

2) Chat Uygulaması

PHP servisi tasarım desenlerine uygun olacak şekilde geliştirilmeye çalışılmıştır. Kullanılan tasarım desenleri Abstract Factory, Command ve Template'dir. PHP servisi hem MySql hem de MongoDB ye bağlantılıdır. MongoDB'de chat uygulamasında kullanıcıların oluşturdukları konular/odalar ile yorumları tutulmaktadır. Bu kısmın MongoDB'de tutulmasının amacı NoSql veri tabanı olmasıdır. NoSql veri tabanları şemasız dinamik tasarımları vardır. Esnek veriler için uygundur ve sabit tablo yapısı bulunmamaktadır. Yorum ve konulara eklenecek yeni alanların (örn oda tipi) olabileceği göz önüne alındığından MongoDB seçilmiştir. MySql veri tabanında ise kullanıcı bilgileri tutulmaktadır. Mysql için oluşabilecek sql injection saldırılarına karşılık php'in kendi güvenli bağlantı sağlayan MySqli sınıfı kullanılmıştır. MySqli sınıfında bulunan Prepare fonksiyonu ile standart yazım örnekleri aşağıda verilmiştir.

1. Örnek standart yazım. Select * from where kullanici = $Değişken
2. Prepare ile yazım. Select * from where kullanici = ?

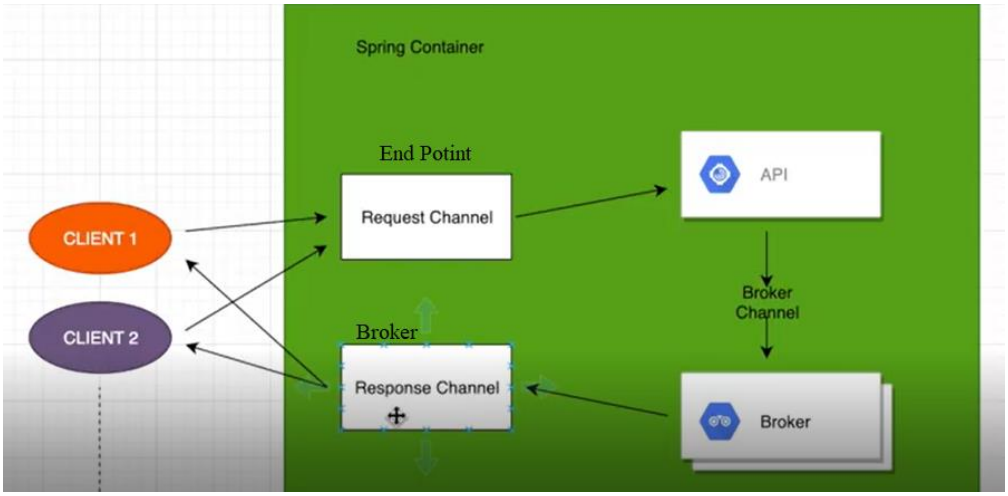
Standart yazımlarda sorguya direk değişken verilmektedir. Bu değişken kullanıcının girmiş olduğu bir değerdir. Kötü niyetli kullanıcılar tarafından girilecek meta karakterler ile veri tabanından bilgilerin çalınmasına neden olabilir. Bunun önlenmesi için girilen girilecek değerlerin ön adımlarda geçirilerek istenmeyen karakterlerden arındırılması gerekmektedir. Prepare fonksiyonu da buna benzer iş yapmaktadır.

PHP sunucusuna ajax ile post isteği bulunurken hangi veri tabanında çalışılacağı BaseCommand adlı değişken ile gönderilmektedir. MongoDB ya da MySql arasında geçiş yapmak için gönderilecek BaseCommand değişkenine MysqlCommand ya da MongoDbCommand verilmelidir. Yorumlar MongoDB ya da kullanıcı bilgileri MySql'de kaydedilse de bütün veri tabanı işlemleri MongoDB'ye ya da MySqle çekilebilmektedir.

PHP soket üzerinden açılan konular ve yorumlar şifrelenmiş biçimde gönderilmektedir. Uygulamada 2 tip oda bulunmaktadır ve bunlardan birisi normal odadır. Normal odalara tüm kullanıcılar girebilirken resim yükleme işlemi bulunmaktadır. Kullanıcıların yükledikleri resimler PHP soket üzerinden gönderilmektedir. Gönderilen resimler uygulamanın sol kısmına eklenmektir. Resimlerin üzerine çift tıklandığında, bilgisayara indirilmektedir.

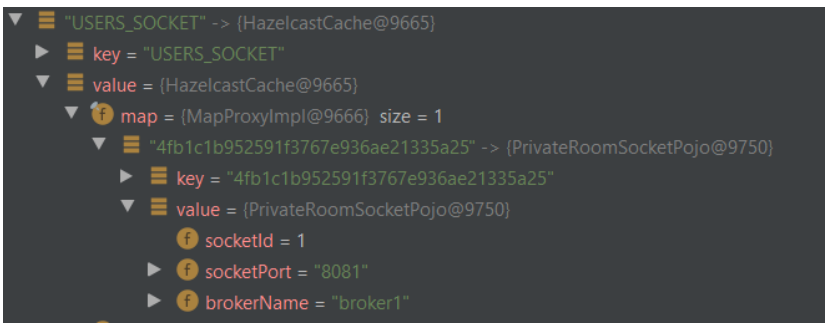
Uygulamada bulunan bir diğer oda çeşidi özel odadır. Bu odaya maksimum 4 kişi girebilmektedir. Uygulamada bir kullanıcı özel oda oluştururken, odaya girebilecek kullanıcıların adlarını girmektedir. Girilen kullanıcı adları Spring Boot ile yazılmış olan servise gönderilmektedir. Servise gelen veriler cacheye kaydedilmektedir. Cache kullanılmasının sebebi, verilere cache bağlanabilecek uygulamalardan erişilebilir olmasıdır. Yapılan uygulamada cacheye erişen sadece 1 servis vardır fakat o servisin bir yedeği olduğu varsayılmaktadır. Sunucuda oluşacak bir hatada, yedek sunucu cache üzerinde tutulan verilere erişebildiğinden uygulamaya çalışmaya devam edecektir.

Normal odalarda resim yüklenebilirken özel odalarda dörtlü görüntülü konuşma (ses aktarımı yok) eklenmiştir. Her 50 mili saniyede kameradan alınan resimler soket üzerinden gönderilmektedir. Resimler base64 formatında gönderilmiştir.
Görüntülü konuşma sisteme bir yük getirmektedir. Bütün görüntülü konuşma işlemlerinin bir soket üzerinden yapılması durumda, sistem yükü kaldıramamaya başlayacaktır. Bu durum kullanıcıyı da etkilemektedir. Kullanıcının sadece bir portu dinlemesi durumunda açılmış olan bütün özel odalardaki veri akışı bilgisayarı yoracaktır. Bunu engellemek için, yoğunluğa göre soket instance kaldırılmıştır. Bunun için her 10 özel odaya ait bir java soket instance bulunmaktadır. Java soket yapısı resim 1 'de verilmiştir.

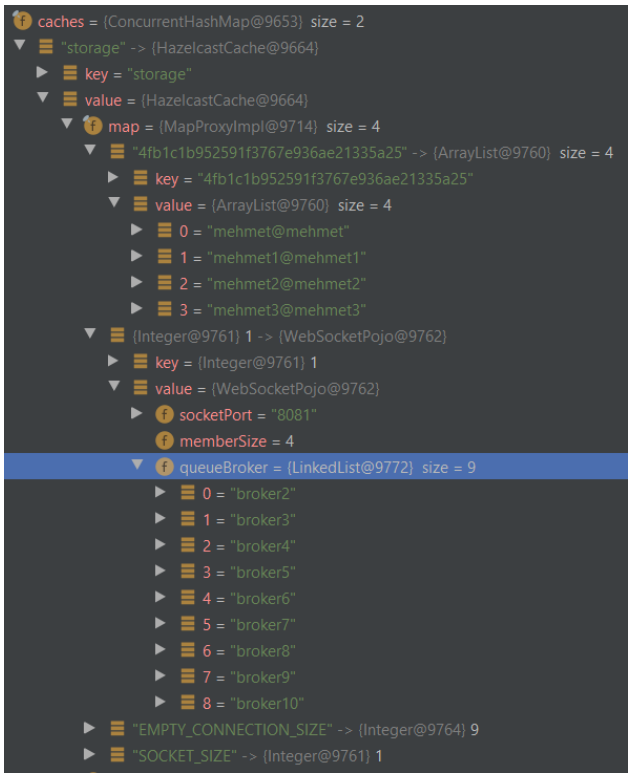


Resim 1

Java sokette sunucular end pointleri dinlemektedir. Yani kullanıcılar gönderecekleri verileri end point adreslerine göndermektedir. Projede kullanılan end point videoTalk dır. Bu durumda sunucunun 8080 portu üzerinde çalıştığı varsayılır ise kullanıcı http://localhost:8080/videoTalk adresine verisini göndermek zorundadır. Gönderilen veri içerisinde resim verisi ve resmin sunucuda gönderileceği broker adı bulunmaktadır. Kullanıcılar ise brokerları dinlemektedir.
Her 10 özel odanın 1 sokete sahip olduğu belirtilmişti. Özel odaların kendilerine ait brokerleri bulunmaktadır. Kullanıcılar sadece kendilerine ait olan sokette kendilerine ait brokerleri dinlemektedir. Bu durumda kullanıcının bir özel odaya giriş yaparken bilmesi gereken bazı bilgiler bulunmakta. Bunlar bağlanacağım soket hangi port üzerinde ve hangi brokeri dinleyeceğim. Tüm bu bilgiler spring boot servisinde özel oda cache üzerinde kaydedilirken aynı zamanda kendisine ait olan soket port ve broker bilgisi de cache üzerine kaydedilmektedir. Soket ve broker bilgileri için daha önceden cache üzerinde kaydedilmiş soket nesneleri gezinerek boş brokera sahip soketlerden birisine üye olunmaktadır. Cache üzerinde tutulan bilgiler resim 2 ve 3' te verilmiştir.



Resim 2



Resim 3

Resim 2'de bir özel odaya verilmiş olan port numarası ve dinleyeceği broker bilgileri görülmektedir. Key bilgisi ise odaların veri tabanlarına kaydedildiğinde verilen id numarasıdır. Resim 3'te ise oluşturulan özel odaya kayıtlı kullanıcılar, 8081 portunda çalışmakta olan soketin boş broker adları ve boş bağlantı sayısı gibi bilgiler görülmektedir.

Spring sunucusuna özel oda kaydedilirken boş bağlantı sayısı kontrol edilmektedir. Boş bağlantı sayısı eğer 3'ten küçük ise yeni bir soket instance farklı bir portta çalıştırılmaktadır. Soket instance farklı portlarda çalıştırmak için ayrı bir projede soket kodlanmıştır ve maven kullanılarak jar haline getirilmiştir. Spring sunucusunda ise jar dosyası Runtime sınıfının exec methodu kullanılarak çalıştırılmaktadır. Soket çalıştırılırken main fonksiyonuna port bilgisinin gönderilmesi gerekmektedir. Bundan dolayı WebSocketHelper sınıfı oluşturulmuştur. Bu sınıfta port kuyruğu oluşturulmuş ve içerisine 8081 ile 8091 arasındaki port numaraları pushlanmıştır. Yeni bir soket oluşturulurken bu kuyruktan bir port numarası pop edilerek jar dosyası çalıştırılmıştır. Jar dosyası çalıştırılıp yeni soket çalıştırıldığında cache üzerindeki boş bağlantı sayısı, soket size ve yeni oluşturulan soket nesnesi ile cache güncellenmektedir. Soketlerdeki yoğunluğun düşmesiyle, boşta kalan soketlerin kapanma işlemi zaman yetmediğinden dolayı projeye eklenememiştir. Spring sunucusu kapatıldığında bütün soketler otomatik kapatılmaktadır.

Chat uygulaması PHP üzerinde çalıştığından Java servisine CORS'dan direk istek de bulunamamaktadır. CORS, web tarayıcısı tarafından yönetilen ve ek HTTP başlıkları kullanılarak, bir kökende çalışan web uygulamasının, farklı bir kökende yer alan web uygulamasına erişim izni kontrolünü sağlayan mekanizmadır. Web uygulaması, **internet tarayıcısı** üzerinden farklı bir kökene (protokol, domain ve port) herhangi bir istek gönderirse cross-origin HTTP isteği oluşturmuş olur. Günümüzdeki birçok modern internet tarayıcısı, JavaScript kodu üzerinden başlatılan HTTP isteklerini güvenlik nedenlerinden dolayı kısıtlar. Ajax istekleri, tarayıcı üzerinde gerçekleşirken **same-origin policy**'i (aynı köken politikasını) izler. Bu nedenle, ajax isteği gerçekleştiren bir web sitesi, sadece kendi sitesi üzerindeki kaynaklara erişim sağlama yetkisi vardır. Eğer farklı bir siteye erişmek istiyorsa ilgili sitenin yer aldığı uygulamada CORS başlıklarının uygun şekilde ayarlanması gereklidir.

Java servisinde cors ayarlarında gelen verilerden sadece User_Header başlık bilgisine sahip istekleri kabul edilmiştir.

Chat uygulamasında açılan odalar ve yapılan yorumların şifrelenmesinde daha önceden kendimin kodladığı Tekil Olmayan Matrisler yöntemi ile şifreleme yapılmıştır. Bu şifreleme yöntemi simetrik anahtar kripto sistemlere dahildir. Simetrik kripto sistemlerde, şifreleme anahtarının deşifreleme anahtarından üretilmesi oldukça kolaydır. Çoğu simetrik algoritmalarda, şifreleme ve deşifreleme için kullanılan anahtar aynıdır. Alıcı ve göndericinin güvenli bir şekilde haberleşmesi için, öncelikle bir anahtar üzerinde görüş birliğine varmaları gerekir. Bu algoritmaların güvenliği anahtara bağlı olduğundan anahtar gizli tutulmalıdır.

Tekil Olmayan Matris şifrelemede açık metin karakterine farklı sayılar karşılık getirilir. Boşluk karakteri 0 ile ifade edilir. Bu sayılar M olarak adlandırılan 3 satırlık bir matris şeklinde düzenlenir. Matriste boş kalan pozisyonlar 0'larla doldurulur. Daha sonra M matrisi determinantı +- olan A matrisi çarpılarak B matrisi üretilir. B matrisi şifreli metni oluşturmaktadır. Deşifreleme için A matrisinin tersi hesaplanır ve B şifreli metni ile çarpılır. Sonuçta M orijinal metin elde edilir.

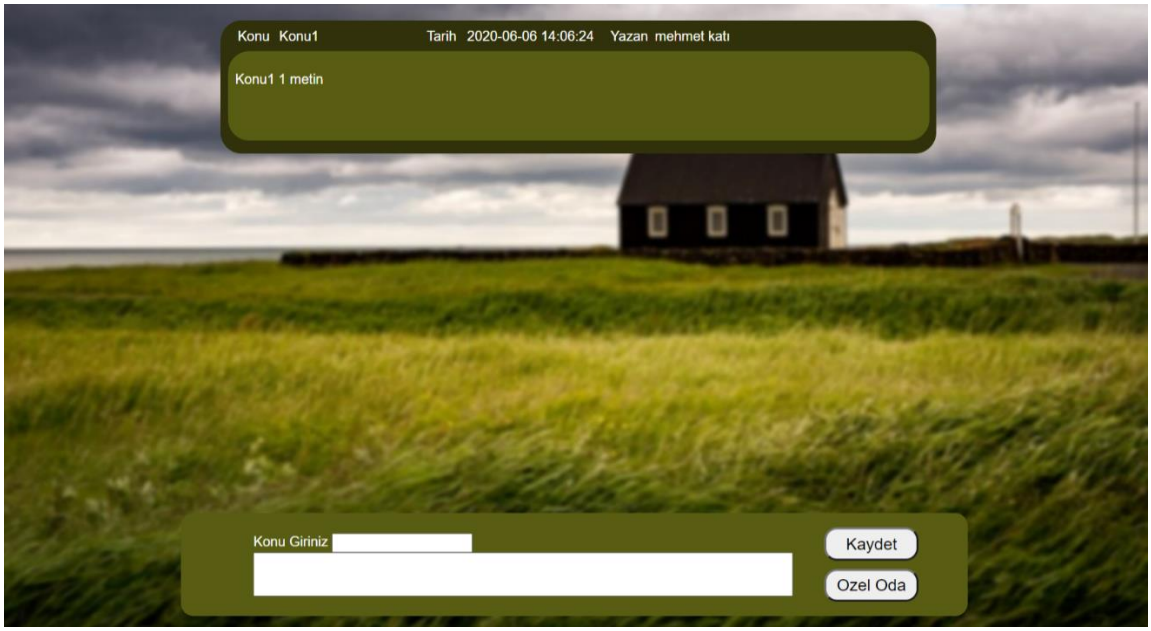
Kodlanmış olan Matris şifrelemede 2 farklı şifreleme tekniği vardır; metin ve sayı şifreleme. Yorum veya konular şifrelenirken hem metin hem de sayı şifreleme yapılmıştır. Şifrelenmiş bir konu resim 4'te görülmektedir.


```
▼ Object
    anahtar: "ş˛˜⌉Ê'_ęyww|GGKv}v{ö□Ä÷ê¿□ÂÅñw"
    konu: "ŸÜ≠ŦR XJ®□£F"
    kullanici: "534.578.431.406.74..89..000080..152.163.177.82.."
    metin: "ĆR̃φ²ȸR̃UYNR XJSG□¡Y□□Y"
    ozelOda: "12631278187.182.353.365."
    tarih: "824.878.830.612.815.614.598.115.126.122.0000117.118.116.238.250.234.243.233.126.122."
  ▶ __proto__: Object
```

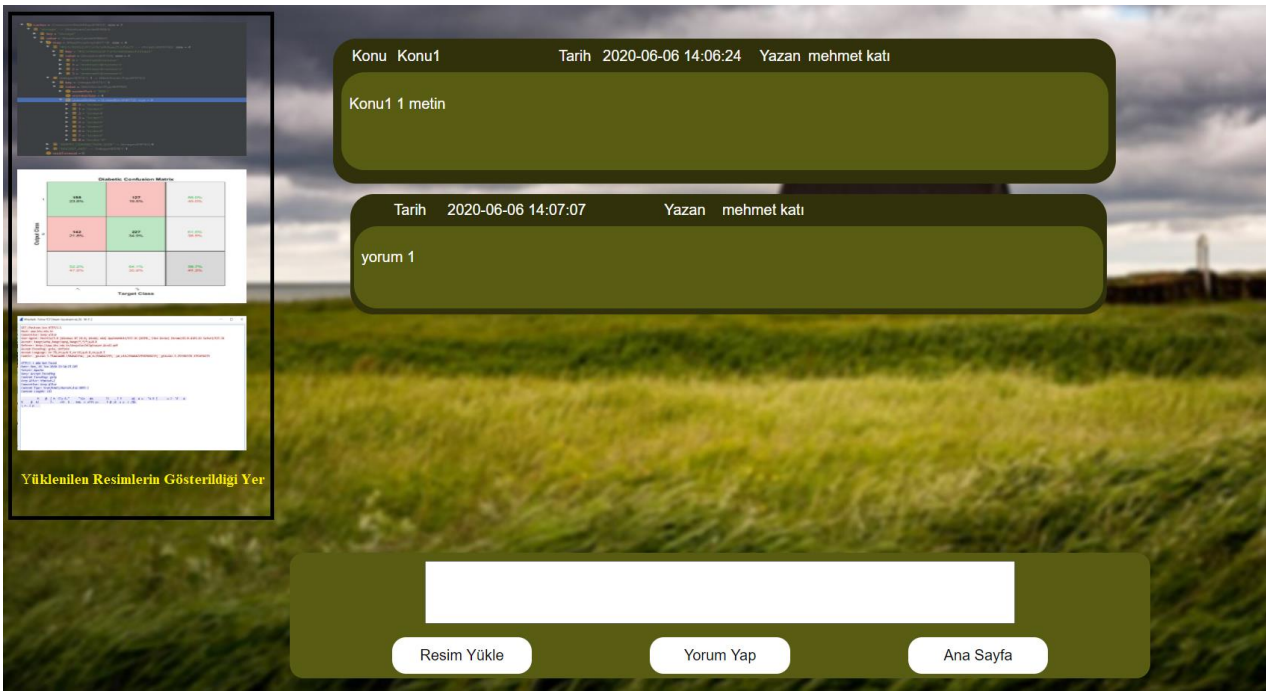

Resim 4

Verilen resimde anahtar, konu ve metin alanları metin şifreleme, kullanici, ozelOda ve tarih alanları ise sayı şifreleme ile şifrelenmiştir.

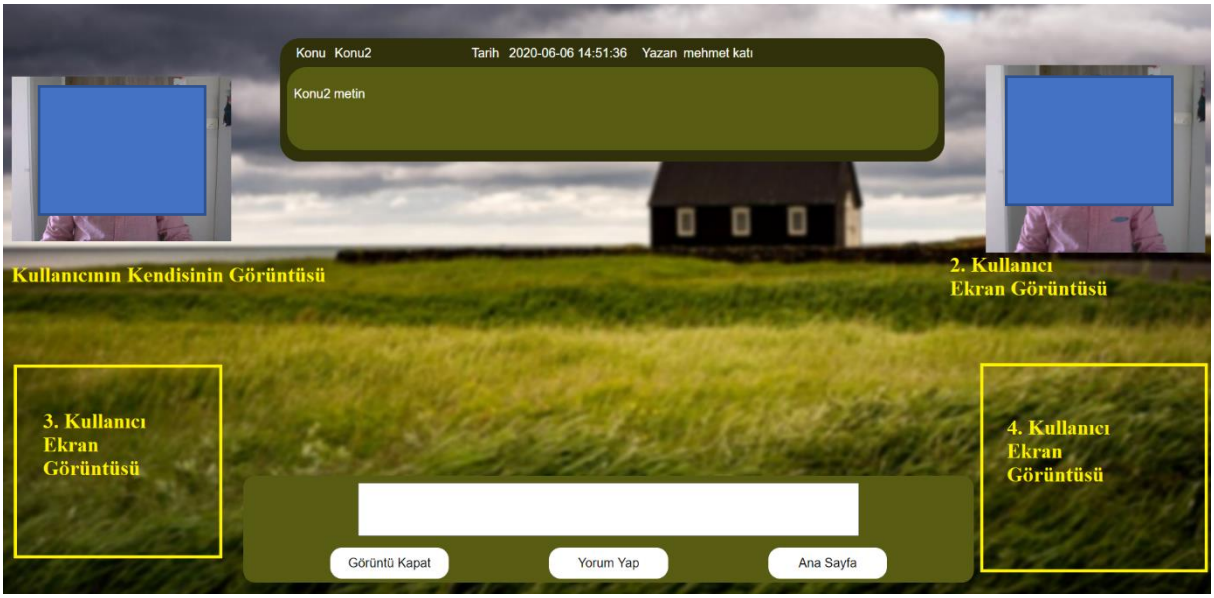
Chat uygulamasından alınan görüntüler resim 5 6 ve 7'de verilmiştir. Resim 5 konuların/odaların yayınlandığı sayfanın ekran görüntüsü, resim 6 normal odaya ait ekran görüntü ve resim 7 ise özel odaya ait ekran görüntüsüdür.



Resim 5



Resim 6



Resim 7

3) Steganografi

Geliştirilecek masa üstü uygulamasında UI tasarımlar için geliştirilmeye başlanılan JavaFX kullanılmıştır. Metnin resme gömülmesi için kullanıcıdan alınan tüm metin bitlere çevrilmiştir. Bit çevirme işlemi yapılırken kullanıcının girmiş olduğu Türkçe karakterler İngilizce karakterlere çevrilmiştir. Bunun sebebi veri gömme ve okuma işleminin 8 bite göre ayarlanmasıdır. Türkçe karakterler ise 9 bittir. Metnin bitlere dönüşüm işlemi bittiğinde 0 ve 1 lerden oluşan bir string – char dizisi oluşmaktadır. Kullanıcı tarafından yüklenilen resmin her pikselinin RGB değerlerine erişilerek en anlamsız biti yerine char dizisinden gelen 1 bit yerleştirilir. Bu işlem char dizisinin sonuna kadar devam eder. Char dizisinin sonuna gelindiğinde metnin bittiğini belirtmek için sonlandırma karakteri olarak resme "00000000" bitlere gömülerek işlem tamamlanmış olur. Veri gömülmüş yeni resim, kullanıcının resmi yüklemiş olduğu dizine kaydedilmektedir.

4) Uygulamanın Çalıştırılması

Chat uygulamasının çalıştırılması için xampp yüklü ise \htdocs klasörünün içerisine vendor ve phpUygulama klasörleri atılır. PHP soketin çalıştırılması için htdocs klasöründen komut penceresi açılır. Komut penceresine "(Path)\xampp\php\php.exe phpUygulama\socket.php" komutu çalıştırılır.

MongoDB bağlantı bilgileri NoSqlDatabase/Factory_Eleman.php dosyasında bulunurken MySql bağlantı bilgileri SqlDatabase/Factory_Eleman.php dosyasında bulunmaktadır.

Görüntülü konuşmada hazırlanan java servisinde jar haline getirilirken sorun oluşmaktadır. Sorun jar haline getirildiğinde, projenin içerisinde bulunan websocket.jar dosyasının çalıştırılamamasıdır. Bunun nedeninin maven tarafından oluşturulan yoldan kaynaklandığı düşünülmektedir. Bu problem zaman kısıtından dolayı çözülemediğinden projenin tamamı yüklenmiştir. Proje intellij idea'da açılarak çalıştırılması gerekmektedir.

Steganografi de proje jar haline getirilmiştir. Klasörde bulunan run.bat dosyası ile çalıştırılmaktadır.