



**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ**

Bilgisayar Mühendisliği Bölümü

BLOKZİNCİR TABANLI SEÇİM SİSTEMİ

Bekircan AĞAOĞLU

**Danışman
Doç. Dr. F. Erdoğan SEVİLGİN**

**Mayıs, 2018
Gebze, KOCAELİ**



**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ**

Bilgisayar Mühendisliği Bölümü

**BLOKZİNCİR TABANLI
SEÇİM SİSTEMİ**

Bekircan AĞAOĞLU

**Danışman
Doç. Dr. F. Erdoğan SEVİLGİN**

**Mayıs, 2018
Gebze, KOCAELİ**

Bu çalışma06/2018 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümü'nde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	Doç. Dr. F. Erdoğan SEVİLGİN	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Yrd. Doç.Dr. Zafeirakis Zafeirakopoulos	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Doç. Dr. Erchan Aptoula	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

İÇİNDEKİLER

İÇİNDEKİLER.....	vi
ŞEKİL LİSTESİ.....	ix
KISALTMA LİSTESİ.....	x
ÖZET.....	xi
SUMMARY.....	xii
1. GİRİŞ.....	1
2. TEMEL BİLGİLER.....	3
2.1. KRİPTOGRAFİK ÖĞELER.....	3
2.1.1. Açık Anahtarlı Şifreleme.....	3
2.1.2. Sayısal İmzalar.....	3
2.1.3. Kör İmzalar.....	4
2.1.4. Kriptografik Hash Fonksiyonları.....	4
2.1.5. TLS.....	5
2.1. BLOKZİNCİR.....	5
2.1.2. Blokzincir Tabanlı Defter-i Kebir (Distributed Ledger).....	5
3. SİSTEM TASARIMI.....	6
3.1. AKTÖRLER.....	6
3.1.1. Sunucu.....	6
3.1.2. İstemci.....	6
3.1.3. Madenci.....	6
3.1.4. Gözlemci.....	6
3.2. SEÇİM.....	6
3.2.1. Seçimin Nitelikleri.....	7
3.2.2. Seçmen Listesi.....	7
3.2.3. Seçimin Başlatılması.....	8
3.2.3. Seçimin Bitirilmesi.....	8
3.2.3. Oyların Sayımı.....	8
3.3. KİMLİK DOĞRULAMASI.....	8
3.3.1. Jeton.....	8

3.4. İLETİŞİM.....	9
3.4.1. İşlemler.....	10
3.4.2. Komutlar.....	11
3.4.3. İstekler.....	11
3.4.4. Cevaplar.....	11
3.4.5. İşler.....	11
3.4.6. İstemci-Sunucu İletişimi.....	12
3.4.6.1. Oy Kullanma Protokolü.....	12
3.4.6.2. Jeton Edinme Protokolü.....	13
3.4.6.3. Oy Sorgulama Protokolü.....	14
3.4.7. İstemci-Madenci/Gözlemci İletişimi.....	14
3.4.8. Madenci/Gözlemci-Madenci/Gözlemci İletişimi.....	15
3.4.9. Sunucu-Madenci/Gözlemci İletişimi.....	16
4. YAZILIM TASARIMI.....	17
4.1. ARAÇLAR VE KÜTÜPHANELER.....	17
4.2. COMMONS MODÜLÜ.....	17
4.2.1. İşlemler.....	18
4.2.2. Blok Sınıfları.....	19
4.2.3. Komutlar.....	19
4.2.3.1 İstekler.....	21
4.2.4. Cevaplar.....	21
4.2.5. İşler.....	22
4.3. İSTEMCİ.....	23
4.3.1. Kullanıcı Arayüzü.....	23
4.4. SUNUCU.....	28
4.4.1. Veritabanı Tabloları.....	28
4.4.2. Kullanıcı Arayüzü.....	29
4.4.3. Modül Yapısı.....	31
4.4.4. Seçimin Bitirilmesi.....	33

4.5 MADENCİ/GÖZLEMÇİ.....	33
4.5.1. Veritabanı Tabloları.....	33
4.5.2. Kullanıcı Arayüzü.....	34
4.5.3. Modül Yapısı.....	36
4.5.4. İşlem Doğrulama Süresi.....	38
4.5.5. İşlem Doğrulamanın ve Oy Sayımının Paralelleştirilmesi.....	38
5. PROJENİN TESTİ.....	39
5.1. BLOK ZİNCİRİNİN TESTİ.....	33
5.2. SİSTEM TESTİ.....	41
5.3. MADENCİNİN TESTİ.....	41
6. TARTIŞMA VE SONUÇ.....	42
KAYNAKLAR.....	43

ŞEKİL LİSTESİ

ŞEKİL 2.1 Açık Anahtarlı Şifreleme Örneği.....	3
ŞEKİL 2.2 Digital İmza Örneği.....	4
ŞEKİL 2.3 Blok Zinciri Örneği.....	5
ŞEKİL 3.1 Aktörler.....	6
ŞEKİL 3.2 Örnek XML Dosyası.....	7
ŞEKİL 3.3 Örnek Seçmen Listesi.....	7
ŞEKİL 3.4 Sistemin Genel Görüntüsü.....	9
ŞEKİL 3.5 Örnek Oy Verisi.....	10
ŞEKİL 3.6 Oy Kullanma Protokolü.....	12
ŞEKİL 3.7 Jeton Edinme Protokolü.....	13
ŞEKİL 3.8 Örnek Oy Görüntüleme.....	14
ŞEKİL 3.9 Madenci Sorgu Örnekleri.....	14
ŞEKİL 3.10 Madenci/Gözlemci Senkronizasyon Protokolü.....	15
ŞEKİL 4.1 Blok Sınıfları.....	18
ŞEKİL 4.2 İşlem Sınıfları.....	19
ŞEKİL 4.3 Komut Sınıfları.....	20
ŞEKİL 4.4 İstek Sınıfları.....	21
ŞEKİL 4.5 Cevap Sınıfları.....	21
ŞEKİL 4.6 İş Sınıfları.....	22
ŞEKİL 4.7 Kullanıcı Arayüzü Başlangıç Ekranı.....	23
ŞEKİL 4.8 Kullanıcı Arayüzü Sunucu Ekranı.....	24
ŞEKİL 4.9 Jeton Edinme Ekranı.....	25
ŞEKİL 4.10 Oy Kullanılan Ekran.....	26
ŞEKİL 4.11 Madenci/Gözlemci Ekranı.....	27
ŞEKİL 4.12 Sunucu Başlangıç Ekranı.....	29
ŞEKİL 4.13 Sunucu Ekranı.....	30
ŞEKİL 4.14 Sunucu Sınıf Diyagramı.....	32
ŞEKİL 4.15 Madenci Başlangıç Ekranı.....	34
ŞEKİL 4.16 Madenci Ekranı.....	35
ŞEKİL 4.17 Madenci Modül Yapısı.....	37

KISALTMA LİSTESİ

G.T.Ü	: Gebze Teknik Üniversitesi
NIST	: National Institute of Standards and Technology
PKI	: Açık Anahtar Altyapısı (Public Key Infrastructure)
RSA	: Rivest, Shamir ve Adelman
SHA	: Secure Hash Algorithm
IP	: Internet Protocol
XML	: Extensible Markup Language
PKCS	: Public Key Cryptography Standards
TLS	: Transport Layer Security

ÖZET

Günümüzde pek çok şeyin dijitalleşmesine karşın seçimler denetlenebilirlik ve anonimlik gibi gerekliliklerden dolayı hala büyük ölçüde kağıt üzerinde yapılmaktadır.

Kağıt üzerinde yapılan seçimlerde oy kullanmak için belirli bir yere gidip sıra beklemek gerekmekte, oyların sayımı uzun zaman almakta ve oy sayımında hata olup olmadığının kontrolü oldukça zor olmaktadır.

Bu çalışmada yukarıda bahsedilen sorunlara sayısal imza, kör imzalama şeması (blind signature scheme), blokzincir ve dağıtık defter-i kebir teknolojileri kullanılarak çözüm bulunmuş; hızlı, basit, anonim ve güvenilir bir seçim sistemi gerçekleştirilmiştir.

SUMMARY

Despite the digitalization of many things nowadays, the elections are still largely conducted on paper due to requirements such as auditability and anonymity.

In these kind of elections, in order to casting a vote a person have to go a specific place and wait in a queue for a while. In addition to that, counting process takes a long time and it is quite hard to being sure if counting process done without any errors.

In this study we have solved the problems mentioned above by using digital signatures, blind signature scheme, blockchain and distributed ledger technologies and implemented a fast, simple, anonymous and reliable voting system.

1. GİRİŞ

Günümüzde pek çok şeyin dijitalleşmesine karşın seçimler denetlenebilirlik ve anonimlik gibi gerekliliklerden dolayı hala büyük ölçüde kağıt üzerinde yapılmaktadır.

Kağıt üzerinde yapılan seçimlerde oy kullanmak için belirli bir yere gidip sıra beklemek gerekirken, oyların sayımı uzun zaman almakta ve oy sayımında hata olup olmadığının kontrolü oldukça zor olmaktadır. Bundan dolayı seçim sürecinin dijitalleştirilmesine yönelik çeşitli çalışmalar yapılmış ve yapılmaktadır. Günümüzde Fransa, İsviçre, Estonya, Avusturya gibi ülkelerde elektronik ortamdan oy kullanılabilir. Bu sistemlerin resmi olarak kullanılmaya başlanmasıyla birlikte, ülkeden ülkeye kimi farklılıklar gösterse de elektronik oylama sistemlerinin sahip olması gereken bazı temel gereklilikler belirlenmiştir. Örneğin Cenevre Eyalet Konseyi aşağıdaki 11 maddeyi gereklilik olarak belirlemiştir [7]:

- 1) Oy verilmesi engellenememeli ve oylar değiştirilememelidir.
- 2) Oylar, oy pusulalarının okunacağı zamana kadar bilinmemelidir.
- 3) Sadece kayıtlı seçmenler oy verebilmelidir.
- 4) Her seçmen yalnızca bir oy hakkına sahip olmalıdır.
- 5) Oy gizliliği güvence altına alınmalıdır.
- 6) Seçim sistemi DoS saldırılarına karşı dayanıklı olmalıdır.
- 7) Seçmenler kimlik hırsızlığına karşı korunmalıdır.
- 8) Kullanılan oy sayısı ile oy pusulalarının sayısı aynı olmalıdır.
- 9) Herhangi bir seçmenin oy kullandığını kanıtlamak mümkün olmalıdır.
- 10) Oy pusulaları okunurken oy kullanılamamalıdır.
- 11) Sistem denetlenebilir olmalıdır.

Ancak günümüzde kullanılan elektronik seçim sistemlerinin pek çoğunda oylar belirli bir kurum tarafından saklanmakta ve değerlendirilmektedir. Bu da o kurumun sistemlerinin güvenliğine ve kurumun dürüstlüğüne güvenme gerekliliği yaratmaktadır. Bitcoin [4]'in ortaya çıkmasından ve kendini kanıtlamasından sonra böyle bir zorunluluğun olmadığı, blokzincir ve dağıtık güvene (distributed trust) dayanan sistemler insanların güvenini kazanmış ve popüler olmaya başlamıştır. Bu çalışmada da hızlı, basit, anonim ve güvenilir bir seçim sistemi gerçekleştirilmiştir.

2. TEMEL BİLGİLER

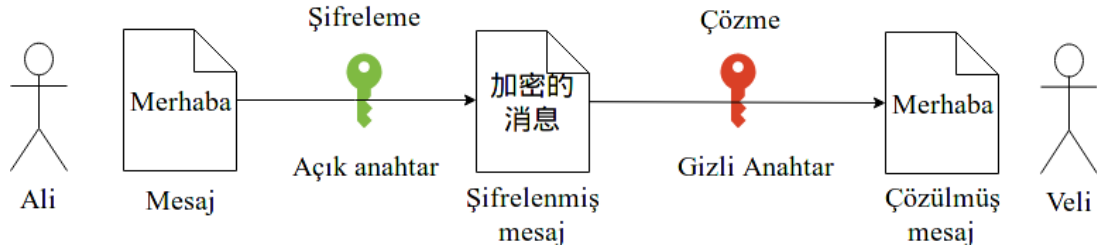
2.1. KRİPTOGRAFİK ÖĞELER

Projede kullanılan protokoller bu başlık altında bahsedilen kriptografik öğeler üzerine inşa edilmiştir.

2.1.1. Açık Anahtarlı Şifreleme

Açık anahtarlı şifreleme [1] veya asimetrik şifreleme, şifreleme ve çözme işlemleri için matematiksel olarak bağlantılı ve birbirinden farklı anahtarların kullanıldığı bir şifreleme sistemidir. Anahtarlardan biriyle şifrelenen veri, diğer anahtar kullanılarak çözülebilir.

Sistemin çalışma şekline dair örnek ŞEKİL 2.1’de gösterilmiştir. Şekilde Ali, Veli’ye iletmek istediği mesajı Veli’nin açık anahtarıyla şifreleyerek şifrelenmiş mesajı elde eder ve Veli’ye gönderir. Veli, gelen mesajı gizli anahtarıyla çözerek mesajı elde eder.

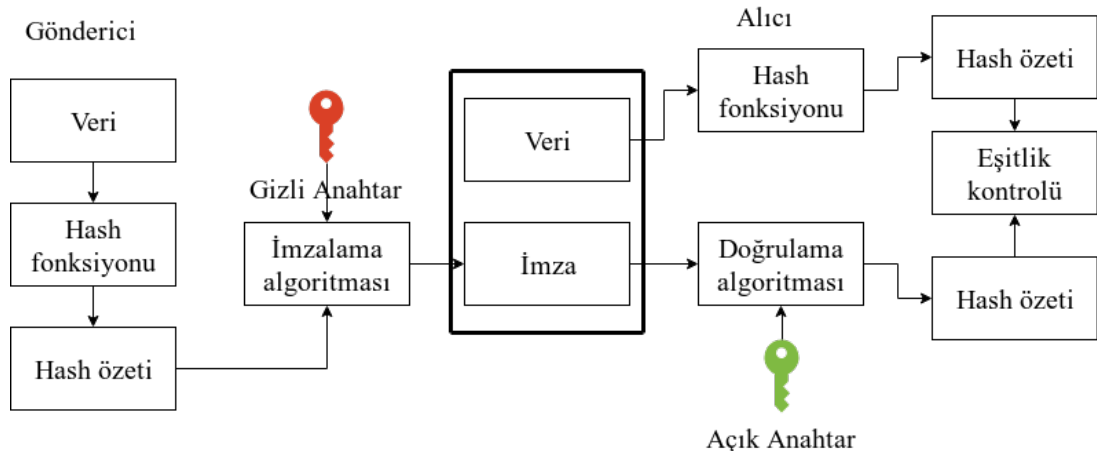


ŞEKİL 2.1 Açık Anahtarlı Şifreleme

2.1.2. Dijital İmzalar

Dijital imzalar [2], açık anahtarlı şifrelemenin bir uygulamasıdır. Bir verinin belirli bir gönderici tarafından gönderildiğinin ve değişmediğinin doğrulanabilmesini sağlar. Bu projede sayısal imzaları elde etmek için RSA algoritması kullanılmıştır.

Sistemin çalışmasına dair örnek ŞEKİL 2.2’de gösterilmiştir. Örnekte gönderici, önceden belirlenmiş bir hash fonksiyonu ile göndereceği verinin hash özetini elde eder, sonrasında da bu hash özetini gizli anahtarı ve imzalama algoritmasını kullanarak imzalar. Daha sonra veriyi imza ile birlikte alıcıya gönderir. Alıcı aynı hash fonksiyonunu kullanarak verinin hash özetini hesaplar ve açık anahtar ve doğrulama algoritmasını kullanarak imzadan imzalanmış olan hash özetini ayıklar. Hesapladığı hash özeti ile imzadan gelen hash özeti aynıysa imzayı doğru kabul eder.



ŞEKİL 2.2 Digital İmza Örneği

2.1.3. Kör İmzalar (Blind Signatures)

Kör imzalar [3], sayısal imzanın bir çeşididir. Bu yöntemde imzalanması istenen mesaj gizlenerek karşı tarafa gönderilir. Dolayısıyla mesajı imzalayan tarafın, mesajın imzalanmasından önce veya sonra mesajın içeriği hakkında bilgisi yoktur. Ancak mesajın ilgili taraf tarafından imzalandığı doğrulanabilir.

2.1.4. Kriptografik Hash Fonksiyonları

Kriptografik hash fonksiyonları, boyutu önemsiz bir bit dizisi alan ve çıktı olarak sabit uzunlukta bir bit dizisi üreten fonksiyonlardır. Üretilen çıktı, girdiye özeldir ve aynı girdi için her seferinde aynı çıktı üretilir. Tek yönlü fonksiyon (one-way function) olarak tasarlanmışlardır. Dolayısıyla çıktıdan girdinin elde edilebilmesinin tek yolu kaba kuvvet aramasıdır (brute-force search).

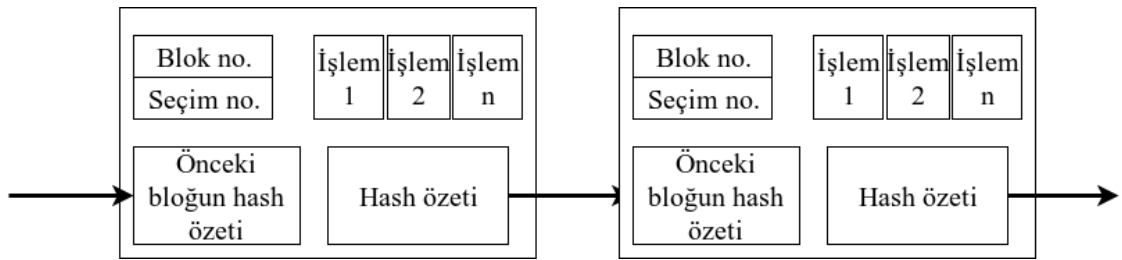
2.1.5. TLS

TLS, bilgisayar ağı üzerinde güvenli iletişim sağlayan kriptografik protokollerdir. Öncelikli amacı, birbiriyle haberleşen iki bilgisayar arasında gizlilik ve veri bütünlüğü sağlamaktır.

2.2. BLOKZİNCİR

Projede blokzincir [4], işlem geçmişinin saklanması amacıyla kullanılmıştır. Blokzincir, blok adı verilen düğümlerden oluşan bir veri yapısıdır. İlk blok dışındaki her blok kendinden önceki bloğun hash özetini içerir. Dolayısıyla herhangi bir bloktaki değişiklik, ilgili bloğun ve sonrasındaki blokların hash özetini etkiler. Bu sayede blok zincirini saklayan taraflar son bloğun hash özetine bakarak aynı işlem geçmişine sahip olduklarını doğrulayabilir.

Bu projedeki blokzincir yapısı ŞEKİL 2.3'te gösterilmiştir. Şekildeki hash özeti; blok numarası, seçim numarası, işlemler ve önceki bloğun hash özeti kullanılarak hesaplanmaktadır.



ŞEKİL 2.3 Blok Zinciri Örneği

2.2.1. Blokzincir Tabanlı Dağıtık Defter-i Kebir (Distributed Ledger)

Projede dağıtık defter-i kebir işlem geçmişinin bağımsız taraflarca doğrulanabilirliğini sağlamak amacıyla kullanılmıştır. Dağıtık defter-i kebirde katılımcıların (peer) her biri, blokzincirin ve henüz blokzincire eklenmemiş işlemlerin kopyasına sahiptir. Blokzincire yeni blok ekleneceği zaman blok tüm katılımcılara iletilir. Her katılımcı bağımsız olarak bloğu doğrular, blokzincire ekler veya reddeder. Böylelikle işlem geçmişi üzerinde dağıtık fikir birliği (distributed consensus) elde edilir.

3. SİSTEM TASARIMI

3.1. AKTÖRLER

Sunucu, istemci, madenci ve gözlemci olmak üzere sistemi oluşturan dört farklı aktör vardır. Bu kısımda aktörler kısaca tanıtılacak ve rapor boyunca ŞEKİL 3.1’de gösterilen sembollerle ifade edilecektir.

3.1.1. Sunucu

Diğer tüm aktörlerin bağlı olduğu, sistemin merkezindeki aktördür. Seçimleri yönetir, kullanıcıları yetkilendirir ve madencilerin iş dağıtımını yapar.

3.1.2. İstemci

Oy kullanabilen, kullandığı oyu ve seçim sonucunu sorgulayabilen aktördür. Oy kullanma işlemini sunucu üzerinden; seçim sonucunu sorgulama işlemini ise madenci veya gözlemci üzerinden yapabilir. Kullandığı oyu sorgulamayı ise her üç aktörden üzerinden de yapabilir.

3.1.3. Madenci

İşlemleri doğrulayan, işlem geçmişini saklayan ve yeni blok üreten aktördür.

3.1.4. Gözlemci

İşlemleri doğrulayan ve işlem geçmişini saklayan aktördür.



ŞEKİL 3.1 Aktörler

3.2. SEÇİM

Bu sistem kullanılarak aynı anda birden fazla seçim yapılabilmektedir ve her seçimin kendine ait nitelikleri, seçmen listesi ve blokzinciri vardır. Bu kısımda seçim sürecinin tasarımı hakkında bilgi verilecektir.

3.2.1. Seçimin Nitelikleri

Her seçim isim, oy verilebilecek en az ve en çok aday sayısı ve aday listesine sahiptir. Kullanıcı bu nitelikleri sunucuya XML dosyası olarak ŞEKİL 3.2’de örnekleri görülen formatta verir. XML etiketlerinin anlamları şunlardır:

- **election:** Dosyanın başlangıcı ile bitişini ifade eder.
- **nvote:** Oy verilebilecek en az ve en çok aday sayısını ifade eder. Örneğin ŞEKİL 3.2’de soldaki dosyada yalnızca bir aday için oy kullanılabilirken sağdaki dosyada en az iki, en çok üç aday için oy kullanılabilir.
- **candidate:** Adayın ismini ifade eder.

Bu niteliklere ek olarak her seçime sunucu tarafından başka seçime atanmamış bir sayı ile oyların şifrelenmesinde ve jeton oluşturulmasında kullanılacak bir anahtar çifti atanır.

<pre><election> <name> seçim 1 </name> <nvote> 1 </nvote> <candidate> aday 1 </candidate> <candidate> aday 2 </candidate> <candidate> aday 3 </candidate> </election></pre>	<pre><election> <name> seçim 2 </name> <nvote> 2 - 3 </nvote> <candidate> aday 1 </candidate> <candidate> aday 2 </candidate> <candidate> aday 3 </candidate> </election></pre>
---	---

ŞEKİL 3.2 Örnek XML Dosyaları

3.2.2. Seçmen Listesi

Seçmenler hem e-posta adresi ile hem de kullanıcı adı ve şifre çifti ile aşağıdaki şekillerde tanımlanabilir:

- “<kullanıcı>@<alan adı>”: Belirtilen e-posta adresi.
- “*@<alan adı>”: Belirtilen alan adına sahip tüm e-posta adresleri.
- “<kullanıcı>”: “<şifrenin SHA256 özeti>”: Belirtilen kullanıcı adı ve şifre

Seçmen listesi sunucuya metin dosyası olarak ŞEKİL 3.3’te örneği görülen formatta verilir.

```
"bekircanagaoglu@gmail.com"
"*@gtu.edu.tr"
"test": "ad57366865126e55649ecb23ae1d48887544976efea46a48eb5d85a6eeb4d306"
```

ŞEKİL 3.3 Örnek Seçmen Listesi

3.2.3. Seçimin Başlatılması

Sunucu seçimin niteliklerini ve seçmen listesini madencilere ve gözlemcilere ileterek seçimi başlatır.

3.2.4. Seçimin Bitirilmesi

Sunucu seçim bittiğinde işlem havuzunda değerlendirilmemiş işlem kalmaması adına tüm madencilere sırasıyla blok oluşturma komutunu verir ve cevap gelene kadar veya blok oluşturma süresi dolana kadar bekler. Sonrasında seçimin gizli anahtarını dağıtarak seçimi bitirir.

3.2.5. Oyların Sayımı

Seçimin bitmesiyle gizli anahtar edinildikten sonra oylar çözülüp sayılabilir.

3.3. KİMLİK DOĞRULAMASI

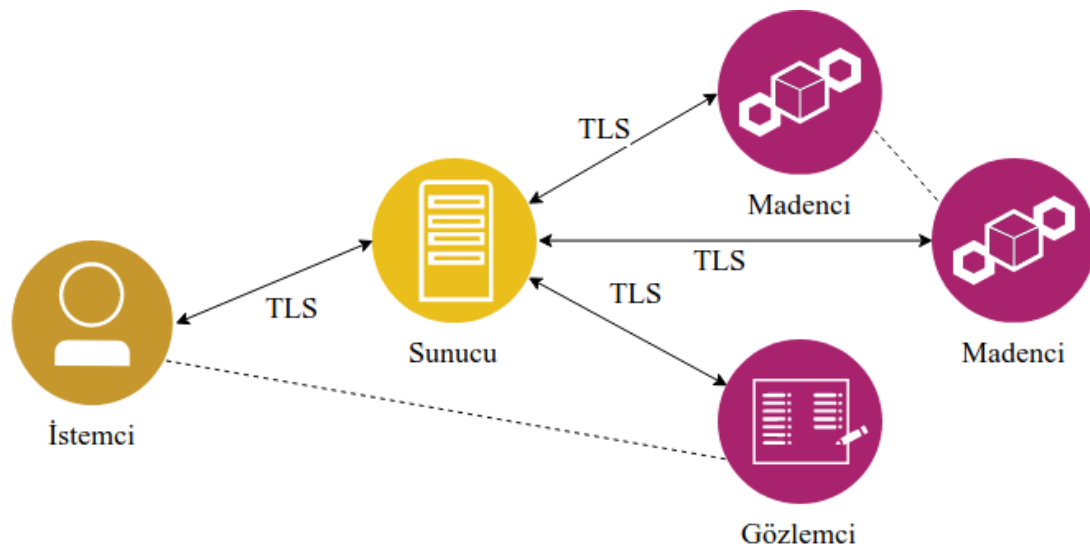
Kimlik doğrulaması e-posta ve e-posta adresine gönderilen tek kullanımlık şifre veya kullanıcı adı ve şifre çifti ile yapılabilir. Yetkilendirme sonrası kullanıcıya oy vermek için kullanabileceği jeton verilir. Her seçmen yalnızca bir adet jeton alabilir.

3.3.1. Jeton

Jeton sahibinin oy verme yetkisi olduğunu ifade eder ancak seçmenin kimliği ile bağdaştırılamaz. Seçmenin açık anahtarının SHA256 özetini ve sunucunun ilgili seçimin anahtarı kullanarak ürettiği imzayı içerir.

3.4. İLETİŞİM

Sistemin genel görüntüsü ŞEKİL 3.4'teki gibidir. Aktörler birbirleriyle çoğunlukla sunucu üzerinden iletişim kurarlar ancak kimi durumlarda doğrudan bağlantı yapabilirler. Sunucu ile olan bağlantılar TLS ile, şekilde kesikli çizgi ile ifade edilen diğer bağlantılar ise soket ile sağlanmıştır. Bu bölümün kalanında iletişimde kullanılan veri tipleri ve protokoller açıklanacaktır.




ŞEKİL 3.4. Sistemin Genel Görüntüsü

3.4.1. İşlemler

Sunucu tarafından imzalanmış hareketlerdir. Ait olduğu seçimin numarasını, zaman bilgisi ve sunucunun imzasını içerir. İşlem tipleri şunlardır:

1. Jeton Oluşturma: Jetonu oluşturan kullanıcının kullanıcı adını veya e-posta adresini içerir.
2. Oy Kullanımı: ŞEKİL 3.5'te içeriği gösterilen oy kullanımı işlemi şunları içerir:
 - Açık anahtar: İstemcinin açık anahtarı.
 - Jeton: Sunucu tarafından verilen jeton.
 - İstemcinin imzası
 - Oy: Oy verisi aşağıdaki sıralamaya sahip bayt dizisinin seçime ait açık anahtarla şifrelenmiş halidir. Seçim bittiğinde sunucunun gizli anahtarı dağıtması sonrası açılabilir.
 - 4 bayt: Oy verilmiş aday sayısı.
 - 4 bayt: İlk adayın numarası.
 -
 - 4 bayt: Son adayın numarası.
 - 4 bayt: Sağlama değeri (Oy verilmiş adayların isimlerinin Java.String.hashCode metodu ile elde edilmiş özet değerlerinin toplamı).
 - 8 bayt: Rastgele sayı.

n	aday 1	...	aday n	sağlama değeri	rastgele sayı	
İmza						

ŞEKİL 3.5 Örnek Oy Verisi

3.4.2. Komutlar

İlgili olduğu seçimin numarasını ve sunucunun imzasını içerir. Sunucu madencilerle bu komutlar aracılığıyla iletişim kurar.

1. Seçimi Başlat: Seçim başlatmak için kullanılan komuttur. Seçimin niteliklerini ve seçmen listesini içerir.
2. Seçimi Bitir: Seçim bitirmek için kullanılan komuttur. Seçime ait gizli anahtarı içerir.
3. Sıradaki Bloğu Oluştur: Sıradaki bloğu oluşturacak madenciye belirten komuttur. Madenciye ait açık anahtarı, blok oluşturmak için sahip olduğu zamanı ve Sunucu-Madenci arasındaki ping zamanını içerir.
4. Bana Bağlan: Sunucu tekrar başlatıldığında kapanmadan önce kendisine bağlı olan madencilere bu komutu gönderir.

3.4.3. İstekler

Sunucu veya istemciler tarafından madenci ve gözlemcilerden bilgi almak için kullanılır. İlgili seçimin numarasını içerir.

1. Oyu Getir: Kullanılan oyu sorgulamaya yarayan istektir. Oy sahibinin açık anahtarını içerir. Seçim henüz bitmemişse oyun içeriği yalnızca sunucu üzerinden öğrenilebilir; madencilerden ve gözlemcilerden ancak oyun blokzincirde olup olmadığı sorgulanabilir.
2. Seçim Sonucunu Getir: Seçimlerin sonucunu sorgulamaya yarayan istektir.

3.4.4. Cevaplar

Madencilerin ve gözlemcilerin, istek ve komutlara verdikleri cevaplardır.

1. Oyu Getir Cevabı: 3.5.2’de bahsedilen oy işlemini içerir.

3.4.5. İşler

Herhangi bir ek bilgi içermeyen verilerdir. Başka bir aktörden bilgi talep etmek için veya yapılmak istenen işlemi belirtmek için kullanılır. Sistemde tanımlı olan işler şunlardır; jeton oluşturma, oy kullanma, oy sorgulama, aday listesini görüntüleme, devam eden seçimleri görüntüleme, devam eden işlemin iptali, belirli bir seçimin niteliklerini görüntüleme, bağlı olan madencileri/gözlemcileri görüntüleme, bitmiş olan seçimleri görüntüleme, tek kullanımlık şifre alma.

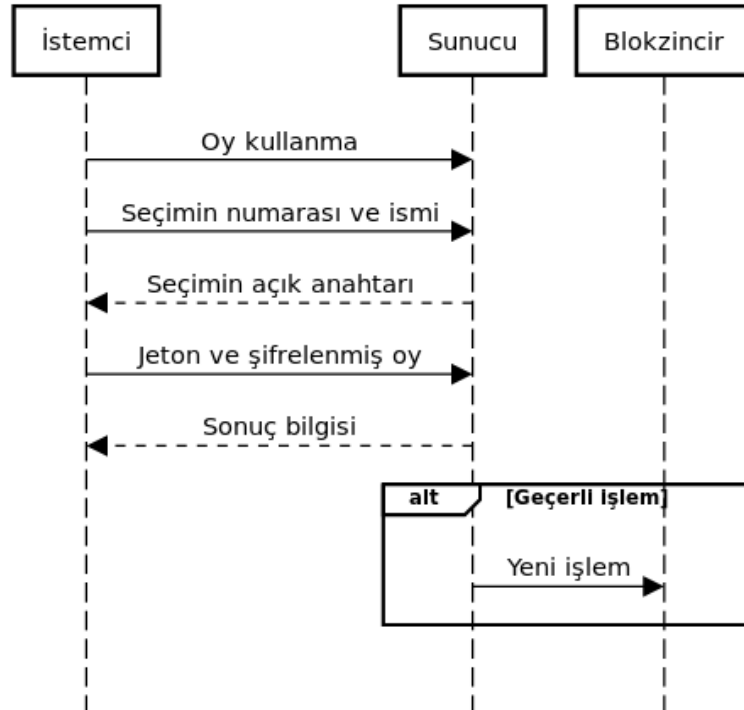
3.4.6. İstemci - Sunucu İletişimi

İstemci sunucu üzerinden:

- Sunucuya bağlı gözlemcileri ve madencileri görüntüleyebilir.
- Devam eden seçimleri görüntüleyebilir ve bu seçimler için:
 - Jeton edinebilir.
 - Oy kullanabilir.
 - Verdiği oyu görüntüleyebilir.

3.4.6.1. Oy Kullanma Protokolü

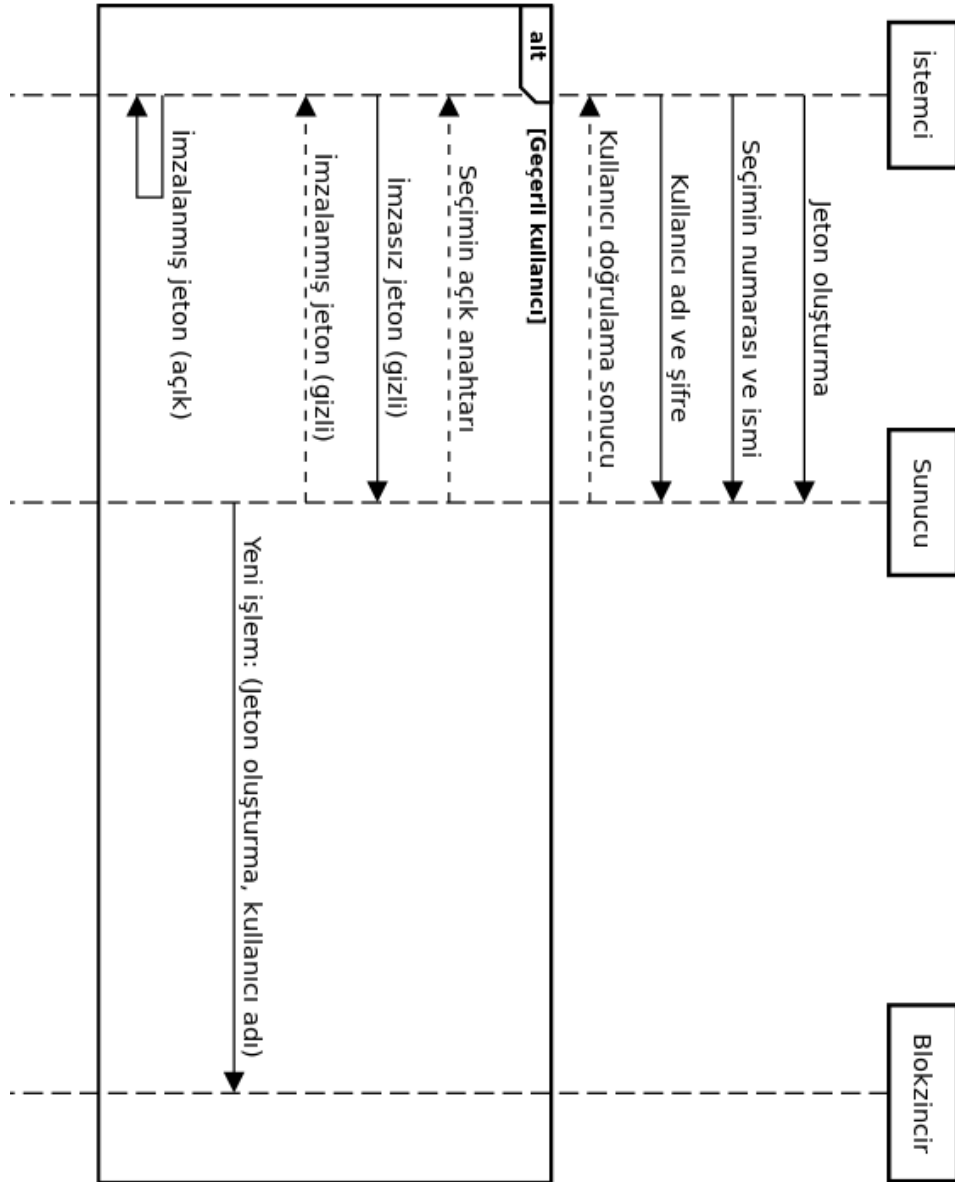
ŞEKİL 3.6’da görüldüğü gibi istemci sunucuya oy kullanma isteğini ve hangi seçim için oy kullanmak istediğini gönderir ve sunucudan ilgili seçimin açık anahtarını alır. Sonrasında oyunu bu anahtarla şifreleyerek jeton ile birlikte sunucuya gönderir, sunucu da işlemin başarılı olup olmadığına dair bilgilendirme mesajı gönderir. Jeton geçerliyse, jetondaki hash değeri ile istemcinin açık anahtarı uyuyorsa ve istemcinin imzası geçerliyse sunucu oyu kabul eder ve blokzincire yayımlar.



ŞEKİL 3.6 Oy Kullanma Protokolü

3.4.6.2. Jeton Edinme Protokolü

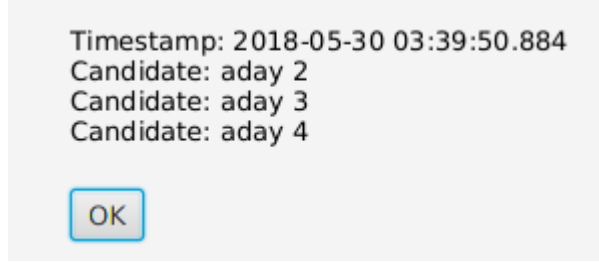
ŞEKİL 3.7’de görüldüğü gibi istemci sunucuya jeton oluşturma isteğini, hangi seçim için jeton oluşturmak istediğini ve kullanıcı adı/şifre bilgilerini gönderir. E-posta ile oy kullanacaksa kullanıcı adı olarak e-posta adresini, şifre olaraksa e-posta adresine gönderilen tek kullanımlık şifreyi kullanır. Sonrasında sunucu kullanıcı bilgilerinin geçerli olup olmadığına dair istemciyi bilgilendirir; kullanıcı bilgileri geçerliyse istemci kör imzalama şemasını kullanarak jeton oluşturur ve sunucuya gönderir. Sunucu jetonu imzalayarak istemciye gönderir ve işlemi blokzincire yayımlar.



ŞEKİL 3.7 Jeton Edinme Protokolü

3.4.6.3. Oy Sorgulama Protokolü

İstemci sunucuya oy sorgulama isteğini ve açık anahtarını iletir. Sunucu da oy blokzincirde ise verilen oyu, değilse oyun blokzincirde olmadığı bilgisini gönderir. ŞEKİL 3.8’de örnek oy sorgulaması sonucu görülebilir.



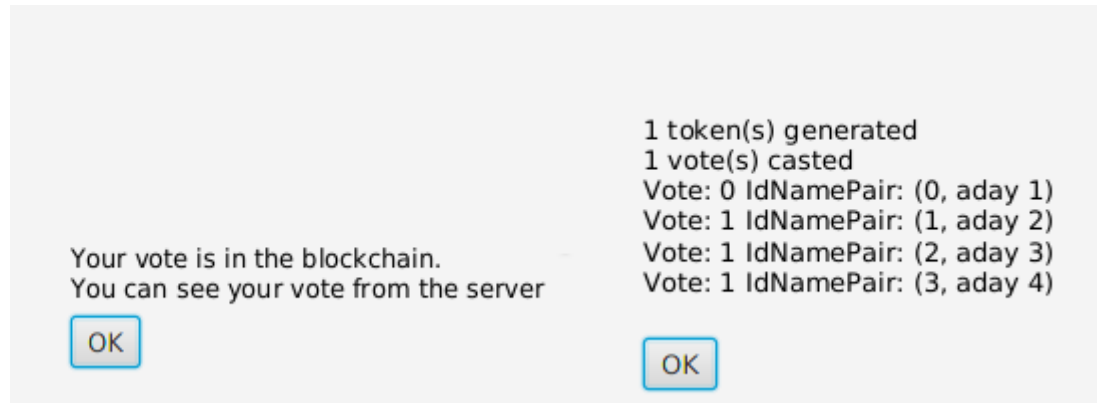
ŞEKİL 3.8 Örnek Oy Görüntüleme

3.4.7. İstemci - Madenci/Gözlemci İletişimi

İstemci madenci/gözlemci üzerinden:

- Devam eden seçimler için oyunun blokzincirde olup olmadığını sorgulayabilir.
- Bitmiş seçimler için:
 - Oyunu görüntüleyebilir.
 - Seçim sonucunu görüntüleyebilir.

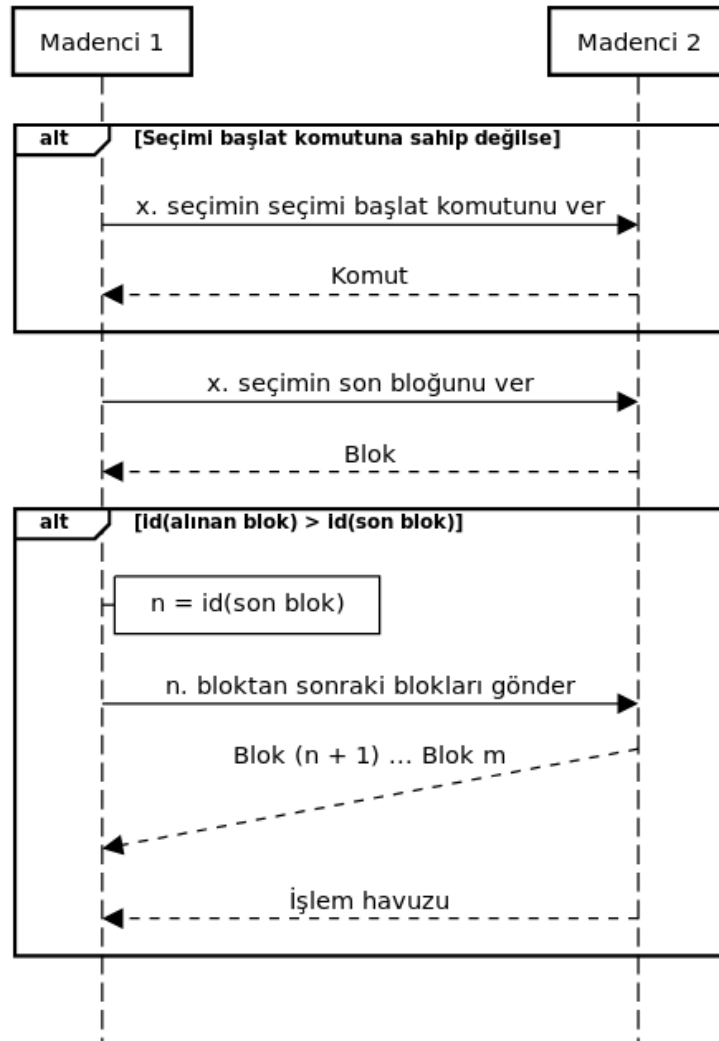
Şekil 3.9’da solda oyun blokzincirde olup olmadığı sorgusunun sağda ise seçim sonucu sorgusunun örneği görülebilir. Blokzincirde olup olmadığı sorgusunda oyun içeriğini göremese de oy istemci tarafından imzalanmış olduğundan oyunun değişmediğinden emin olabilir.



ŞEKİL 3.9 Madenci Sorgu Örnekleri

3.4.8. Madenci/Gözlemci – Madenci/Gözlemci İletişimi

Madenciler birbirleriyle blokzincirleri ve işlem havuzlarını senkronize etmek için iletişim kurarlar. Sunucudan devam eden seçimlerin ve bağlı olan diğer madencilerin listeni alan madenci ŞEKİL 3.10’da görülen protokolü kullanarak senkronize olur. Devam eden seçimin bilgisi yoksa seçim başlatma komutunu ister ve seçimi ilklendirir. Sonrasında seçimin blokzincirindeki son bloğu ister. Gelen bloğun numarası kendisindeki son bloğun numarasından büyükse son bloğundan itibaren olan blokları ister. Blokzincirinin aktarımı tamamlandığında ise işlem havuzunu alır.



ŞEKİL 3.10 Madenci/Gözlemci Senkronizasyon Protokolü

3.4.9. Sunucu - Madenci/Gözlemci İletişimi

Madenci sunucuya bağlandığında sunucu açık anahtarını ve devam eden seçim olup olmadığı bilgisini, devam eden seçim varsa diğer madenci bilgileri ile birlikte, madenciye gönderir. Madenci sunucudan gelen verileri bir kuyrukta tutarak blokszincirleri ve işlem havuzlarını diğer madencilerle senkronize ettikten sonra sunucuya madenci veya gözlemci olduğu bilgisini, madenci ise açık anahtarıyla birlikte, gönderir. Sonrasında ise sunucudan veya istemci ve madencilerden gelecek komut/istekleri bekler.

4. YAZILIM TASARIMI

Yazılım; ortak sınıfları içeren Commons, sunucu ile ilgili sınıfları içeren Server, madenci ve gözlemciyle ilgili sınıfları içeren Miner ve istemciyle ilgili sınıfları içeren Client modülü olmak üzere dört modülden oluşmaktadır.

4.1. ARAÇLAR VE KÜTÜPHANELER

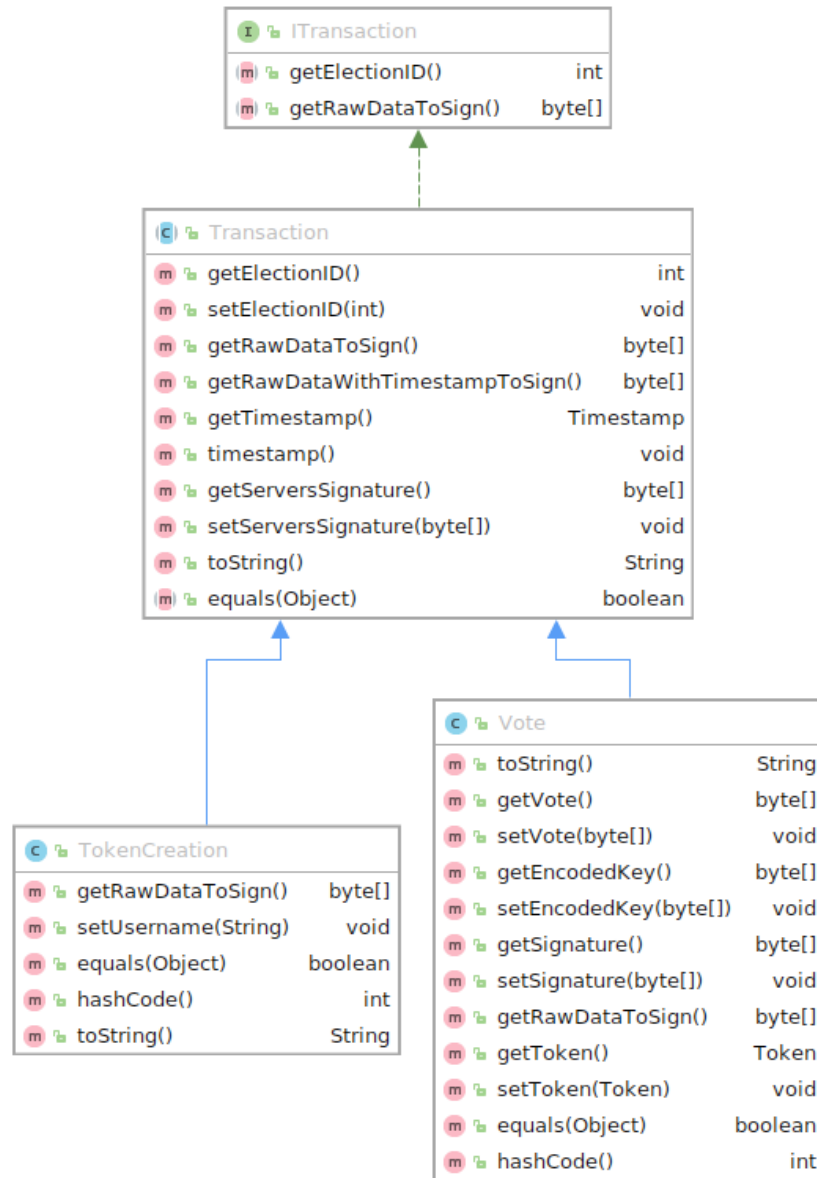
Yazılım Java 1.8 kullanılarak gerçekleştirilmiştir. Proje yönetim aracı olarak Maven, kriptografi kütüphanesi olarak BouncyCastle, e-posta gönderimi için SimpleMailJava uygulama programlama arayüzü (API), veritabanı olarak SQLite ve çeşitli ek araçlar için Apache Commons kütüphanesi kullanılmıştır.

4.2. COMMONS MODÜLÜ

Commons modülü işlem, komut, istek, iş, blok, jeton, kullanıcı ve seçim sınıflarını içerir.

4.2.1. İşlemler

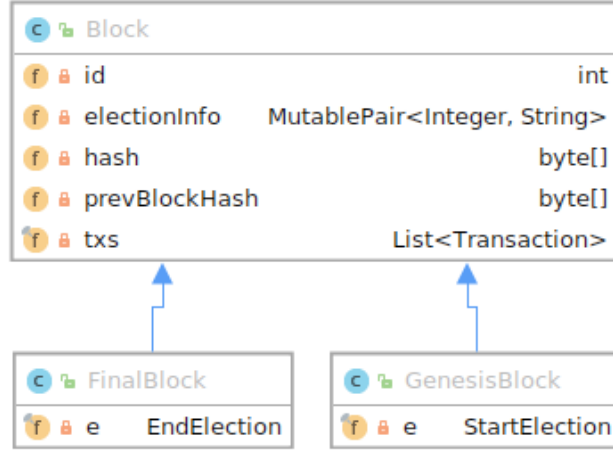
Serileştirilebilir (serializable) bir arayüzden türetilmiş sınıflar olarak gerçekleştirilmişlerdir. Sınıf hiyerarşisi ŞEKİL 4.1’de görülebilir. Seçim numarası, zaman bilgisi ve sunucu imzası üstteki soyut sınıftan gelir, alt sınıflar da kendileriyle ilgili bilgileri ekler.



ŞEKİL 4.1 İşlem Sınıfları

4.2.2. Blok Sınıfları

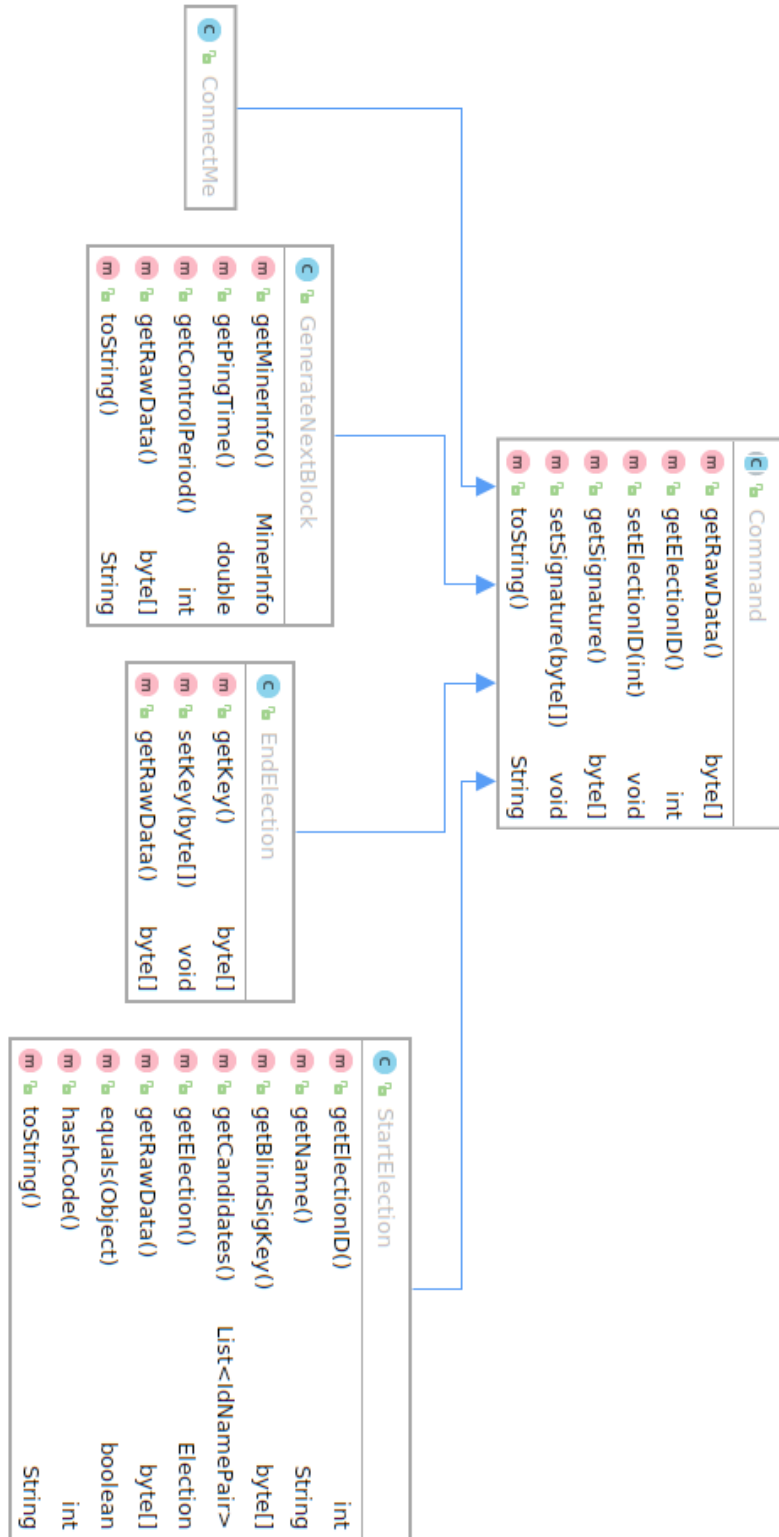
ŞEKİL 4.2’de görülen blok sınıfları serileştirilebilir blok sınıfından türetilmiştir. Blokzincire eklenen ilk blok olan GenesisBlock sınıfı seçim başlatma komutunu, son blok olan FinalBlock sınıfı ise seçimi bitirme komutunu içerir.



ŞEKİL 4.2 Blok Sınıfları

4.2.3. Komutlar

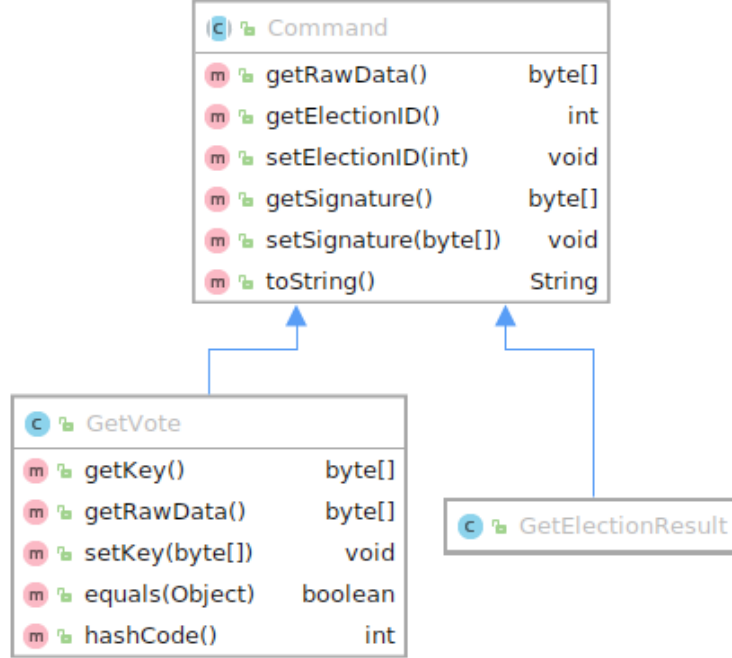
Serileştirilebilir (serializable) bir arayüzden türetilmiş sınıflar olarak gerçekleştirilmişlerdir. Sınıf hiyerarşisi ŞEKİL 4.3’te görülebilir. Seçim numarası ve sunucu imzası üstteki sınıftan gelir, alt sınıflar da kendileriyle ilgili bilgileri ekler.



ŞEKİL 4.3 Komut Sınıfları

4.2.3.1 İstekler

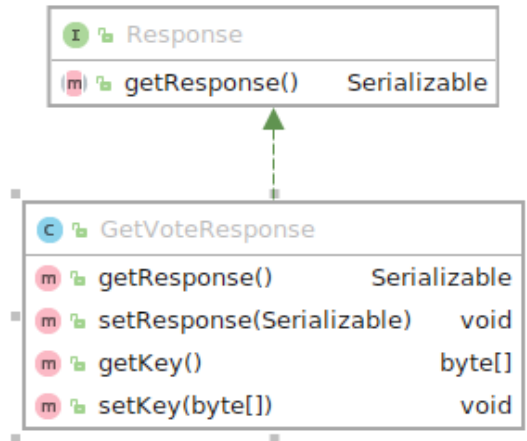
ŞEKİL 4.4'te gösterilen istek sınıfları da komut sınıfından türetilmiş sınıflardır ancak isteklerde sunucu imzası kontrolü yapılmaz.



ŞEKİL 4.4 İstek Sınıfları

4.2.4. Cevaplar

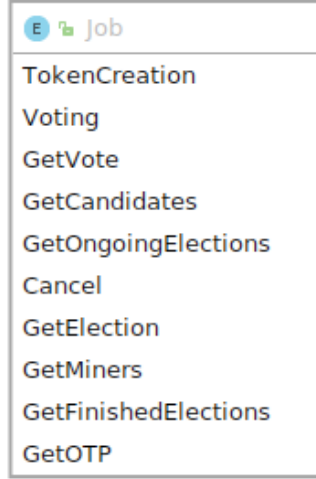
İsteğe olan cevabı içeren, serileştirilebilir cevap arayüzünü gerçekleyen sınıflardır. Sınıf hiyerarşisi ŞEKİL 4.5'te görülebilir.



ŞEKİL 4.5 Cevap Sınıfları

4.2.5. İşler

İşler ŞEKİL 4.6’da görülen serileştirilebilir, sayma (enum) bir sınıf içinde ifade edilmiştir.

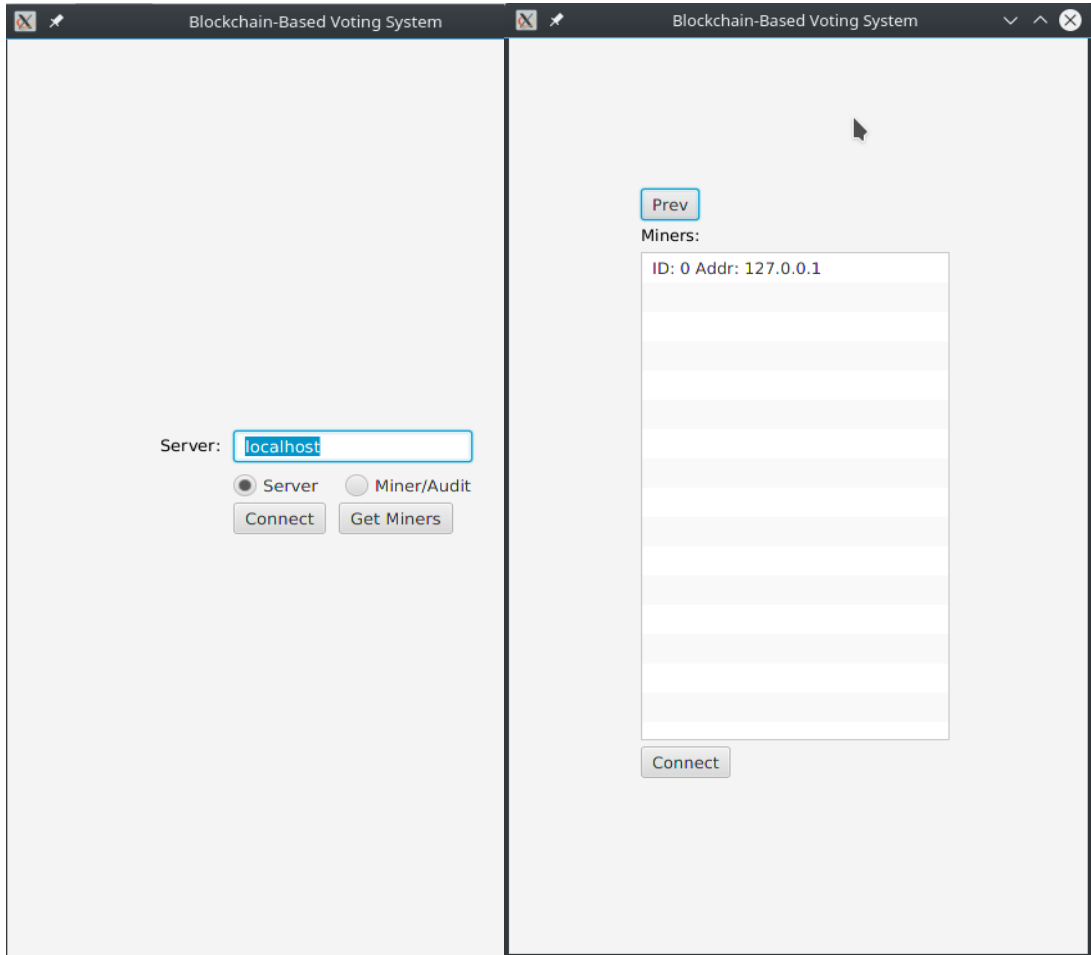


ŞEKİL 4.6 İş Sınıfı

4.3. İSTEMCİ

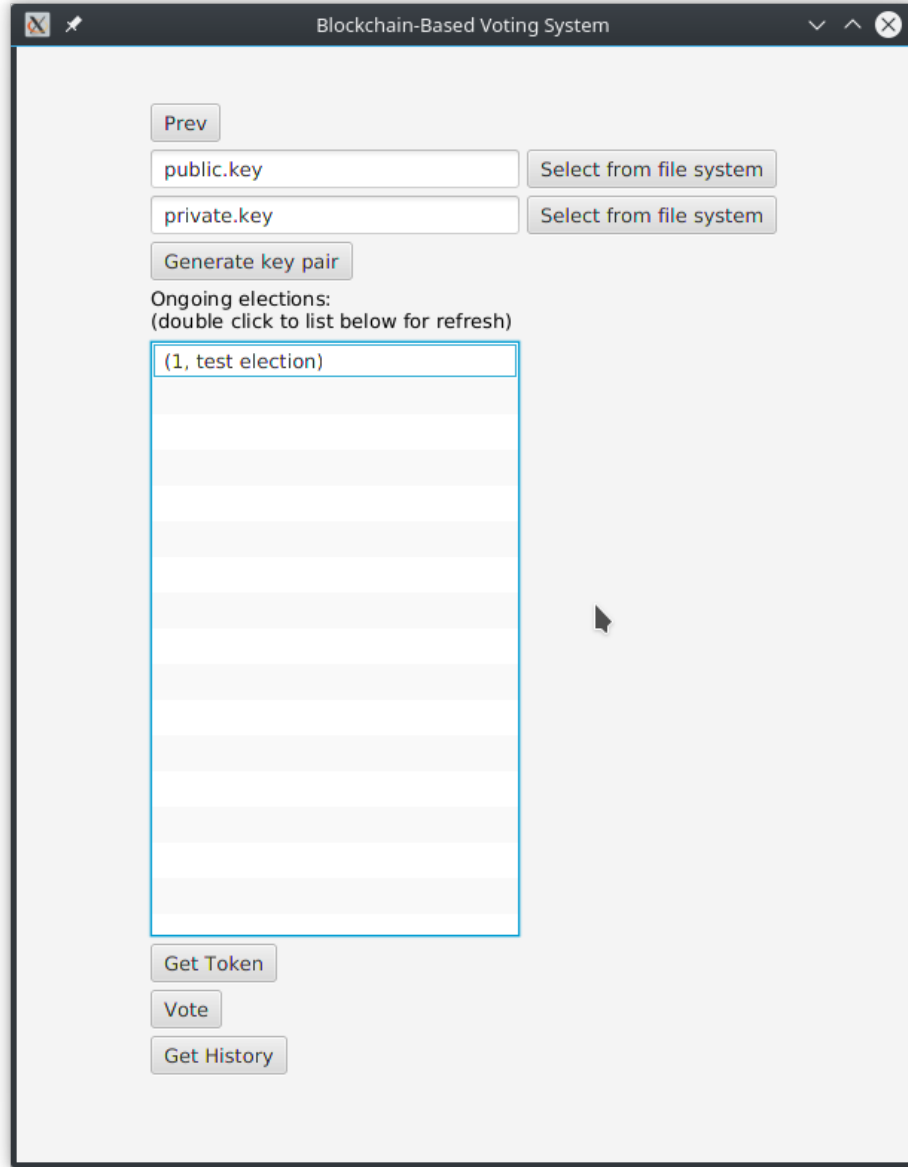
4.3.1. Kullanıcı Arayüzü

İstemci çalıştırıldığında kullanıcı ŞEKİL 4.7’da soldaki arayüz ile karşılaşır. Buradaki Connect butonu metin kutusundaki IP adresini alır, metin kutusunun altındaki radyo butonunda Server seçiliyse sunucunun portundan, Miner/Audit seçiliyse madencinin portundan bağlanır. GetMiners butonu ise sunucudan bağlı olan madenci ve gözlemcilerin listesini alır ve ŞEKİL 4.7’da sağdaki gibi gösterir. Buradaki menüden bağlanılmak istenen madenci seçilerek Connect butonuyla bağlanılabilir veya Prev butonuyla önceki ekrana dönülebilir.



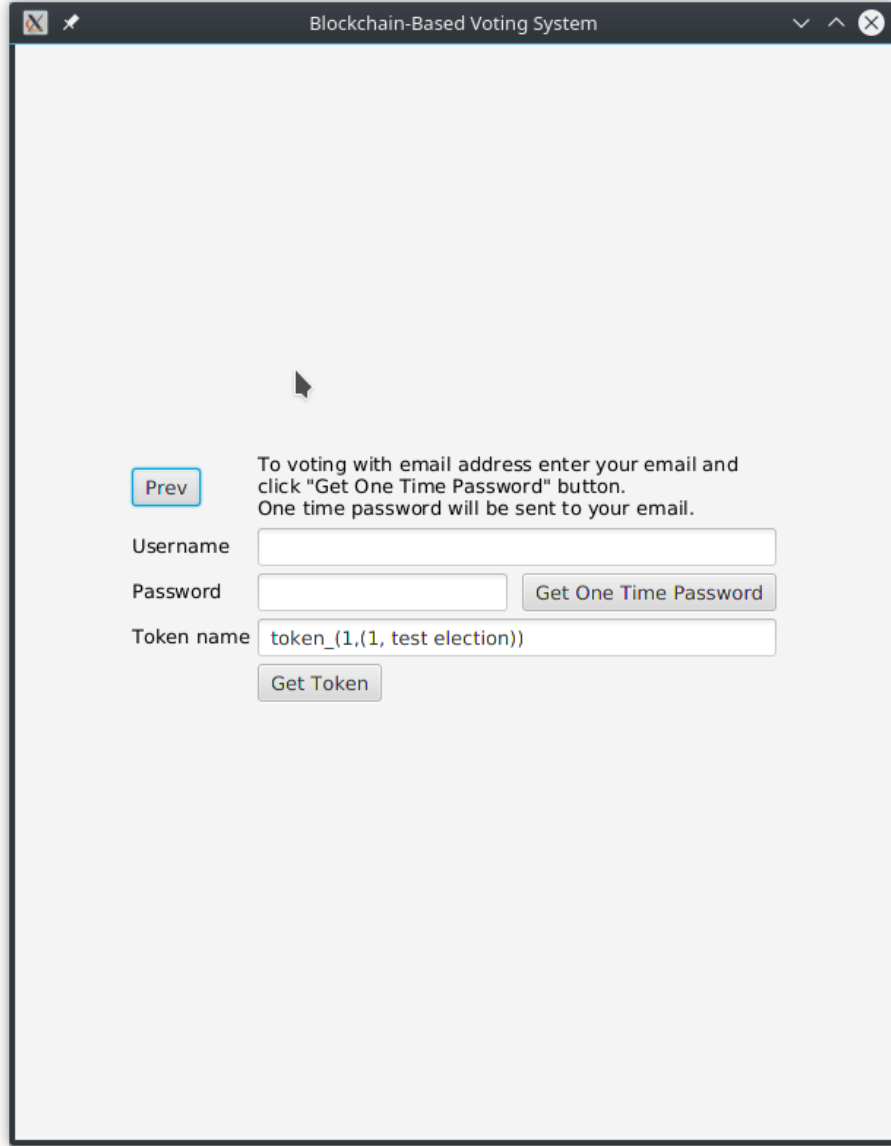
ŞEKİL 4.7 Kullanıcı Arayüzü Başlangıç Ekranı

Sunucuya bağlanıldığında kullanıcı ŞEKİL 4.8’deki ekran ile karşılaşır. Bu ekranda Prev butonu ile önceki ekrana dönebilir, Generate key pair butonu ile anahtar çifti üretebilir veya Select from file system butonları ile mevcut anahtar çiftini seçip kullanabilir. Ekranın ortasında devam eden seçimlerin listesi bulunur, bu liste üzerine çift tıklayarak güncellenebilir. Kullanıcı bu listeden işlem yapmak istediği seçimi seçerek Get Token butonuyla jeton alabilir, Vote butonuyla oy kullanabilir ve Get History butonuyla verdiği oyu sorgulayabilir.



ŞEKİL 4.8 Kullanıcı Arayüzü Sunucu Ekranı

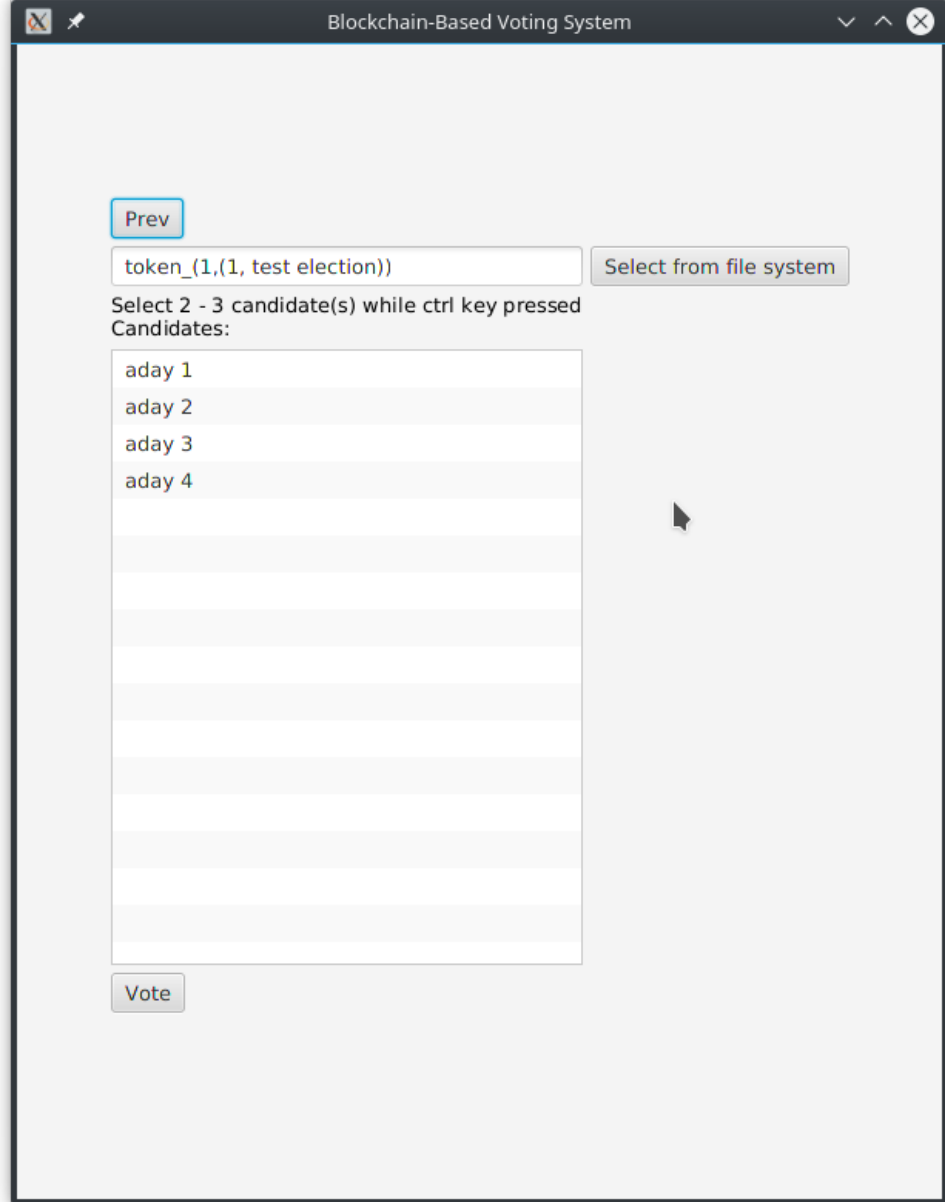
Kullanıcı jeton almak istediğinde ŞEKİL 4.9'deki ekranla karşılaşır. E-posta ile oy kullanacaksa kullanıcı adı için olan metin kutusuna e-posta adresini yazarak Get One Time Password butonuna basar. Sonrasında e-posta adresine gönderilen tek kullanımlık şifreyi parola kısmına yazarak jeton alabilir, jetonun hangi isimde saklanacağını belirtebilir.



The screenshot shows a web application window titled "Blockchain-Based Voting System". The interface is light gray with a central form area. At the top left of the form is a blue "Prev" button. To its right is instructional text: "To voting with email address enter your email and click 'Get One Time Password' button. One time password will be sent to your email." Below this text are three input fields: "Username" (empty), "Password" (empty), and "Token name" (containing the text "token_(1,(1, test election))"). To the right of the "Password" field is a button labeled "Get One Time Password". Below the "Token name" field is a button labeled "Get Token".

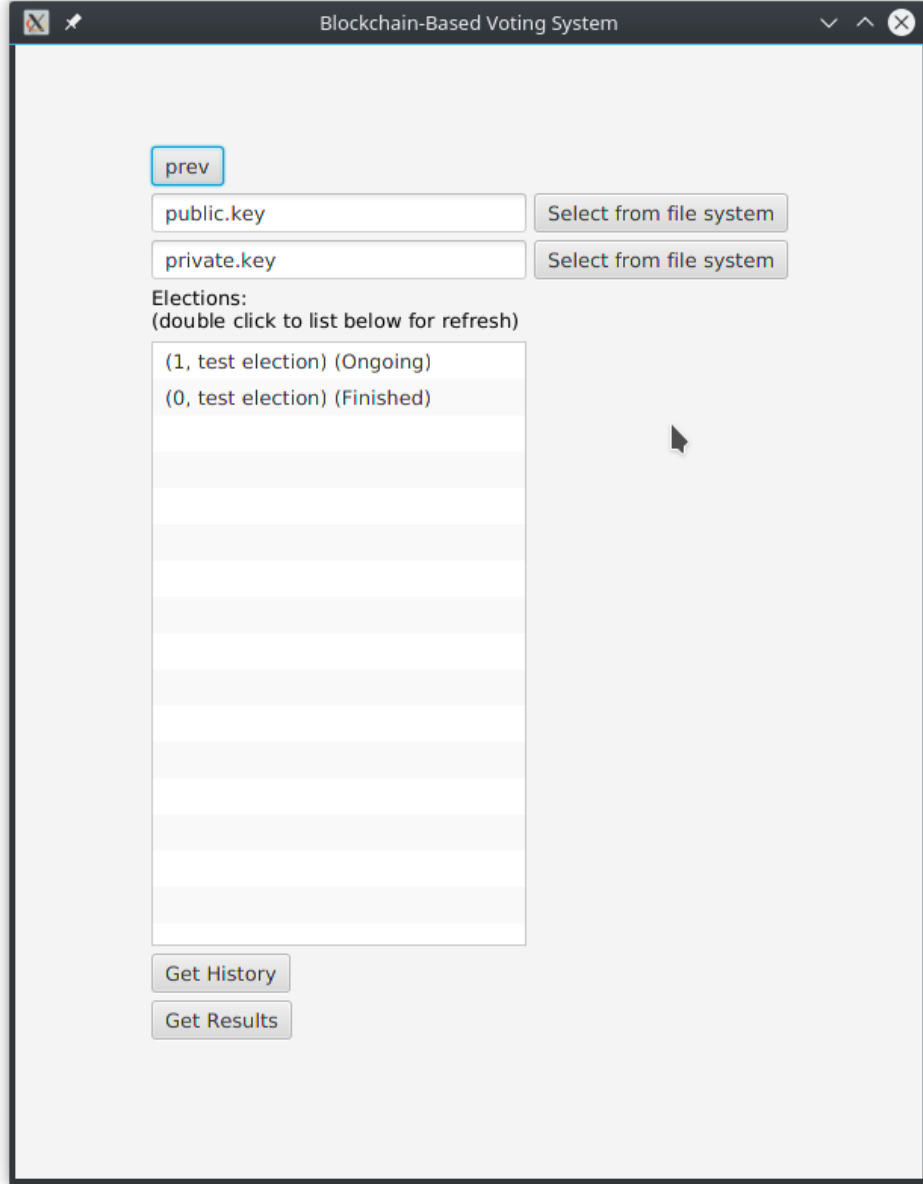
ŞEKİL 4.9 Jeton Edinme Ekranı

Oy verilen ekran ise ŞEKİL 4.10’da gösterilmiştir. Select from file system butonu kullanılarak oy verirken kullanılacak jeton seçilir. Sonrasında ctrl tuşuna basılı tutarak oy verilecek adaylar seçilip Vote butonuna basılarak oy verilir. En az ve en çok kaç aday için oy kullanılabileceği bilgisi aday listesinin üzerinde mevcuttur.



ŞEKİL 4.10 Oy Kullanılan Ekran

Gözetici veya madenciye bağlanıldığında da kullanıcı ŞEKİL 4.11'deki ekranla karşılaşır. Bu ekrandan devam eden ve bitmiş seçimleri görüntüleyebilir ve bu seçimler için 3.4.7. kısımda açıklanmış olan oy sorgulama ve sonuç görüntüleme işlemlerini gerçekleştirebilir.



ŞEKİL 4.11 Madenci/Gözetici Ekranı

4.4. SUNUCU

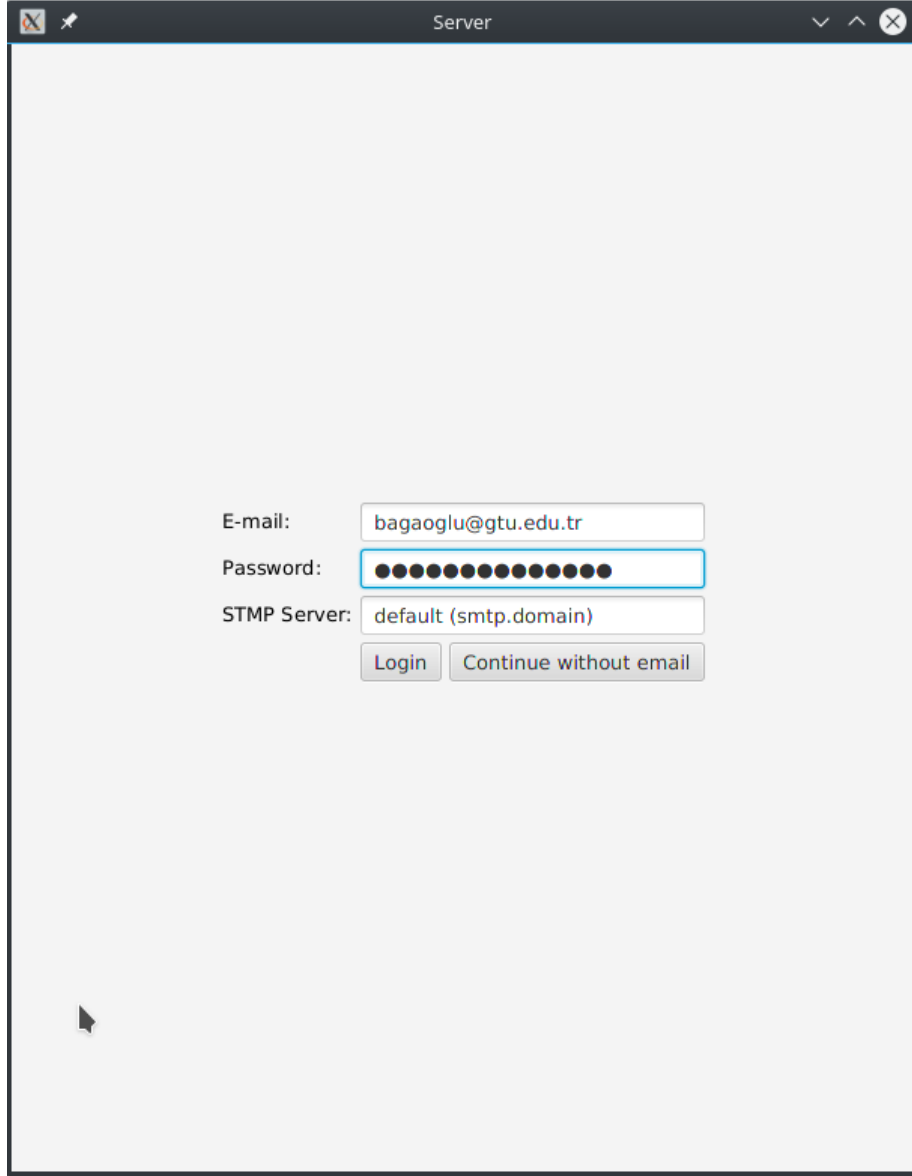
4.4.1. Veritabanı Tabloları

Sunucuda seçim bilgilerini ve bağlı olan madenci/gözlemcileri saklamak için iki farklı veritabanı tablosu kullanılmıştır. Sunucu başladığında bu tabloları okuyarak kaldığı yerden devam eder, değişiklik olduğunda verileri günceller. Tabloların içerikleri şunlardır:

- Seçimler:
 - ID: Tamsayı, Birincil anahtar
 - Seçim: BLOB (seçimin nitelikleri)
 - Kullanıcı bilgileri: BLOB (seçmen listesi, jeton almış olan seçmenler)
 - Seçimin devam edip etmediği bilgisi: Tamsayı
- Madenciler:
 - ID: Tamsayı, Birincil anahtar
 - IP Adresi: BLOB

4.4.2. Kullanıcı Arayüzü

Sunucu başlatıldığında kullanıcı ŞEKİL 4.12'deki ekran ile karşılaşır. E-posta ile oy kullanıcı doğrulama özelliği kullanmak istiyorsa sunucunun e-posta göndermek için kullanacağı bir e-posta bilgisi girer ve Login butonuna basar. Bu özelliği kullanmak istemiyorsa Continue without email butonu ile devam edebilir.

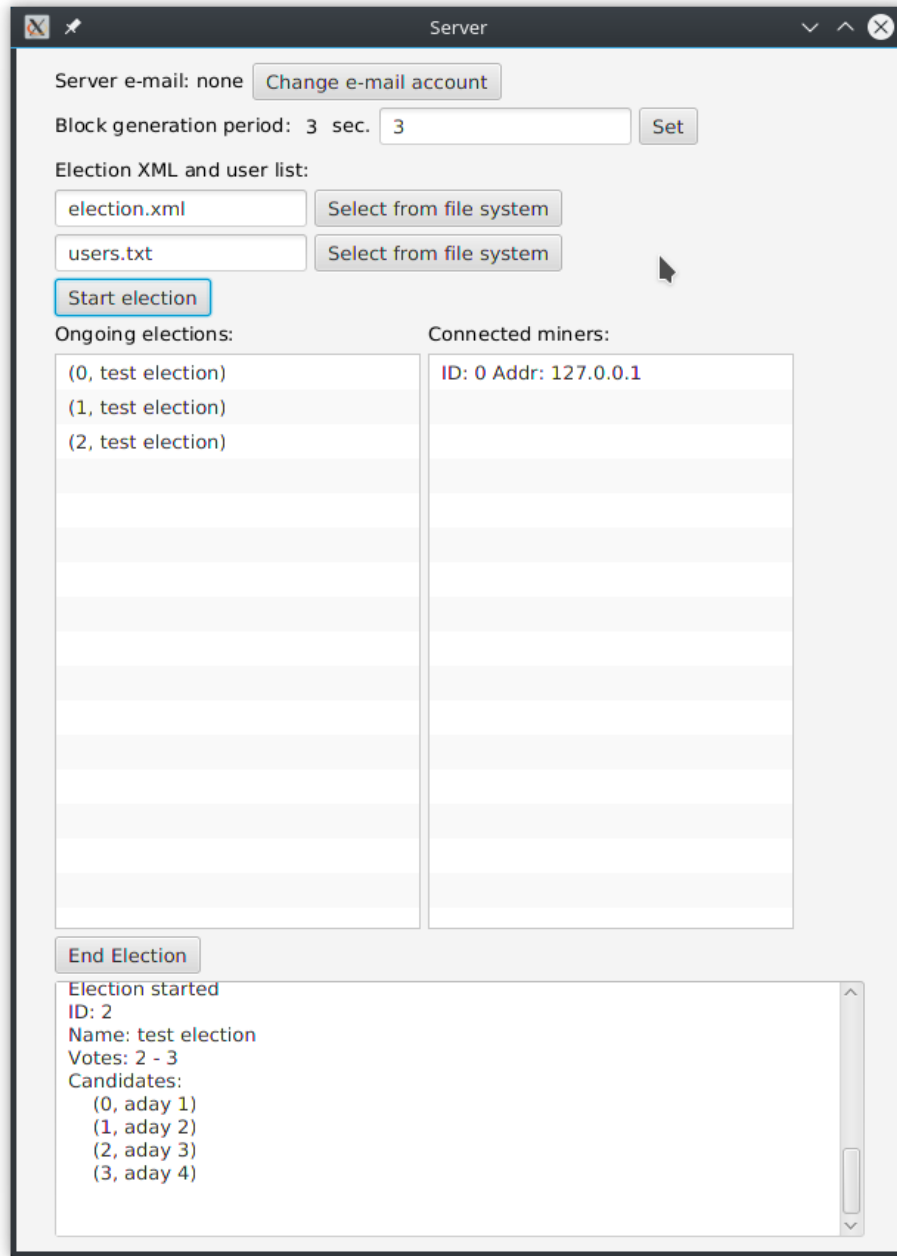


The screenshot shows a window titled "Server" with a light gray background. In the center, there are three input fields and two buttons. The first field is labeled "E-mail:" and contains the text "bagaoglu@gtu.edu.tr". The second field is labeled "Password:" and contains a series of black dots. The third field is labeled "STMP Server:" and contains the text "default (smtp.domain)". Below these fields are two buttons: "Login" and "Continue without email".

ŞEKİL 4.12 Başlangıç Ekranı

Sonrasında ŞEKİL 4.13'deki ekran gelir. Buradaki Change e-mail account butonu ile önceki kısımda tanımladığı e-posta adresini değiştirebilir, bir alt satırındaki metin kutusu ve Set butonu ile de blok oluşturma periyodunu saniye cinsinden ayarlayabilir.

Seçim başlatmak için 3.2.1. bölümde gösterilen formatta XML dosyası ve 3.2.2. bölümde gösterilen formatta seçmen listesi gereklidir. Select from file system butonları kullanılarak bu dosyalar seçilir ve Start election butonuyla seçim başlatılabilir. Devam eden seçimler ve bağlı olan madenci/gözlemciler bu butonun altında listelenir. Sonlandırılmak istenen seçim listeden seçilerek End Election butonuyla sonlandırılabilir. Alt alttaki metin kutusunda ise bilgilendirme mesajları gözükür.

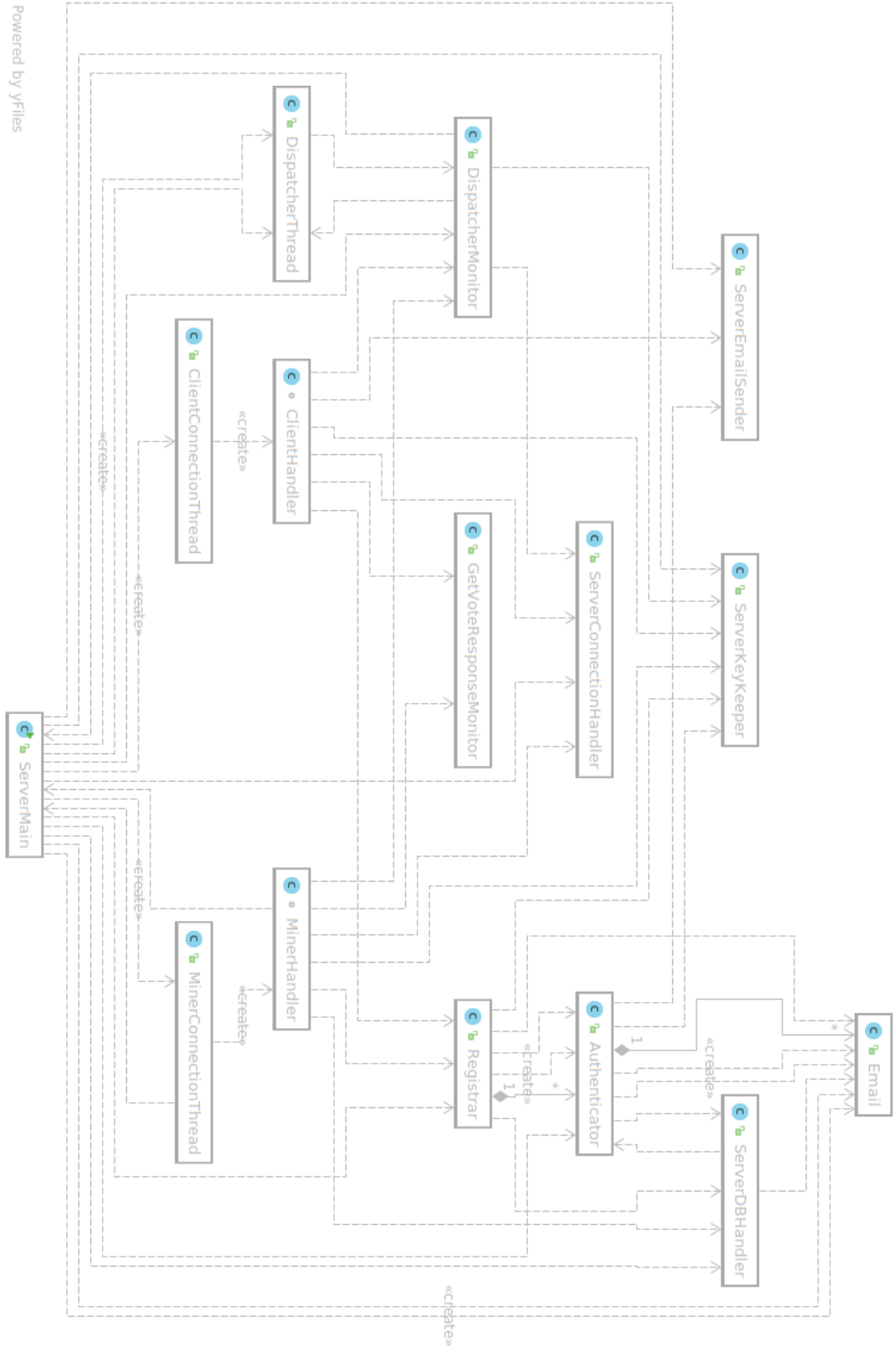


ŞEKİL 4.13 Sunucu Ekranı

4.4.3. Modül Yapısı

Sunucunun içerdiği sınıflar ŞEKİL 4.14’te görülebilir. Sınıfların işlevleri şunlardır:

- ServerEmailSender: E-posta gönderimini yapan sınıftır.
- ServerKeyKeeper: Sunucuya ve seçimlere ait anahtar çiftlerini saklayan sınıftır.
- ServerDBHandler: Veritabanı işlemlerini yapan sınıftır.
- Registrar/Authenticator: Kullanıcı yetkilendirmesini yapan sınıflardır. Authenticator sınıfı tek bir seçim için kullanıcıları yetkilendiren sınıftır. Registrar sınıfı ise yeni seçim başladığında ilgili seçim için bir Authenticator nesnesi oluşturur, böylelikle her seçimde farklı seçmen listesi kullanılabilmesi sağlanır.
- DispatcherThread: Madencilere blok oluşturma işini veren sınıftır.
- MinerHandler: Madencilerle iletişimi sağlayan sınıftır.
- ClientHandler: İstemcilerle iletişimi sağlayan sınıftır.
- ServerConnectionHandler: Verileri madencilere yayınlayan sınıftır.



ŞEKİL 4.14 Sunucu Sınıf Diyagramı

4.4.4. Seçimin Bitirilmesi

Seçim bitirmek istendiği zaman sunucu tüm madencilere sırasıyla blok oluşturma komutu gönderir ve blok oluşturma periyodu kadar cevap bekler. Madenciler işlem havuzunda işlem varsa yeni blok oluşturur ve daha fazla işlem olup olmadığı bilgisini ekler. Böylelikle seçim bittiğinde işlem havuzunda işlenmemiş veri kalması engellenmiş olur.

4.5. MADENCİ/GÖZLEMCİ

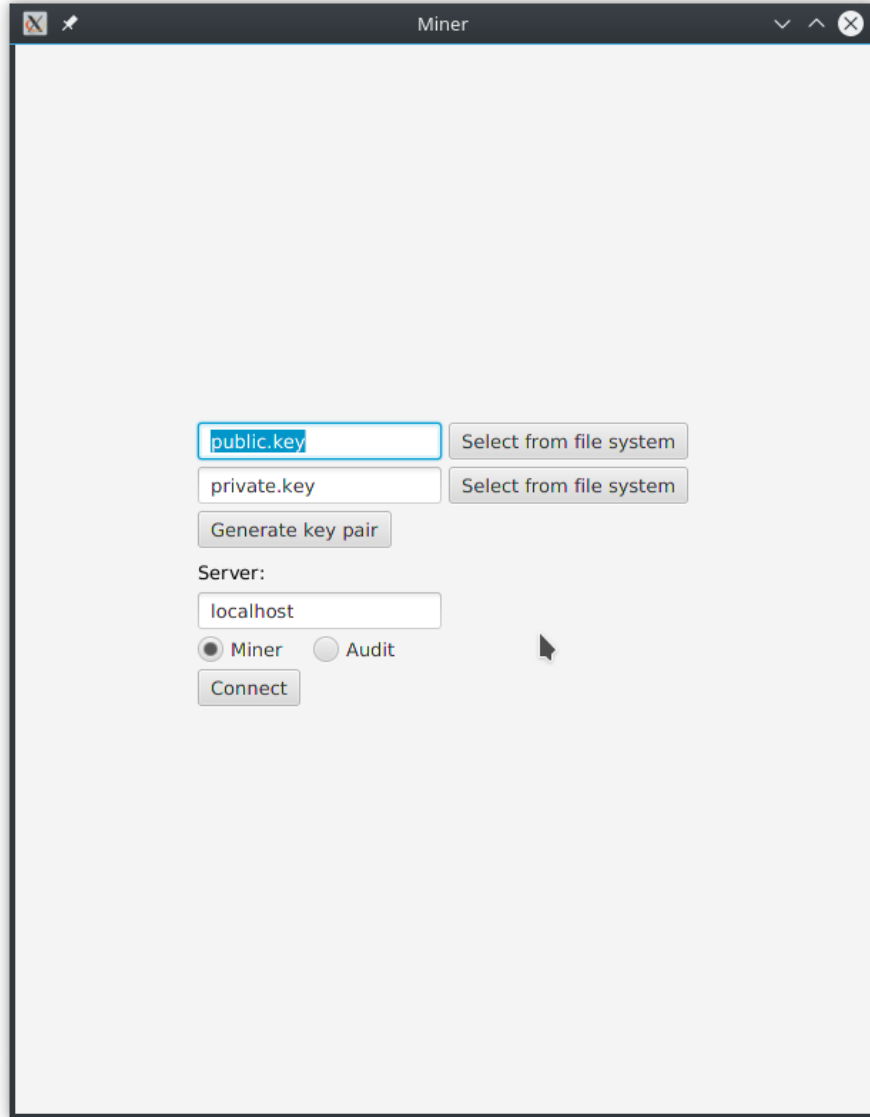
4.5.1. Veritabanı Tabloları

Madenci modülünde seçimler, blokzincirleri ve işlem havuzlarını saklamak için üç farklı veritabanı tablosu bulunur. Madenci başlatıldığında bu tabloları okuyarak kaldığı yerden devam eder, değişik olduğunda verileri günceller. Tabloların içerikleri şunlardır:

- Seçimler:
 - EID: Tamsayı, Birincil anahtar (Seçimin numarası)
 - Seçimi başlatma komutu: BLOB
 - Seçimin devam edip etmediği bilgisi: Tamsayı
- Blokzincirler
 - EID: Tamsayı (Bloğun hangi seçime ait olduğu bilgisi)
 - Blok: BLOB
- İşlem Havuzları:
 - EID: Tamsayı, Birincil anahtar
 - İşlem havuzu: BLOB

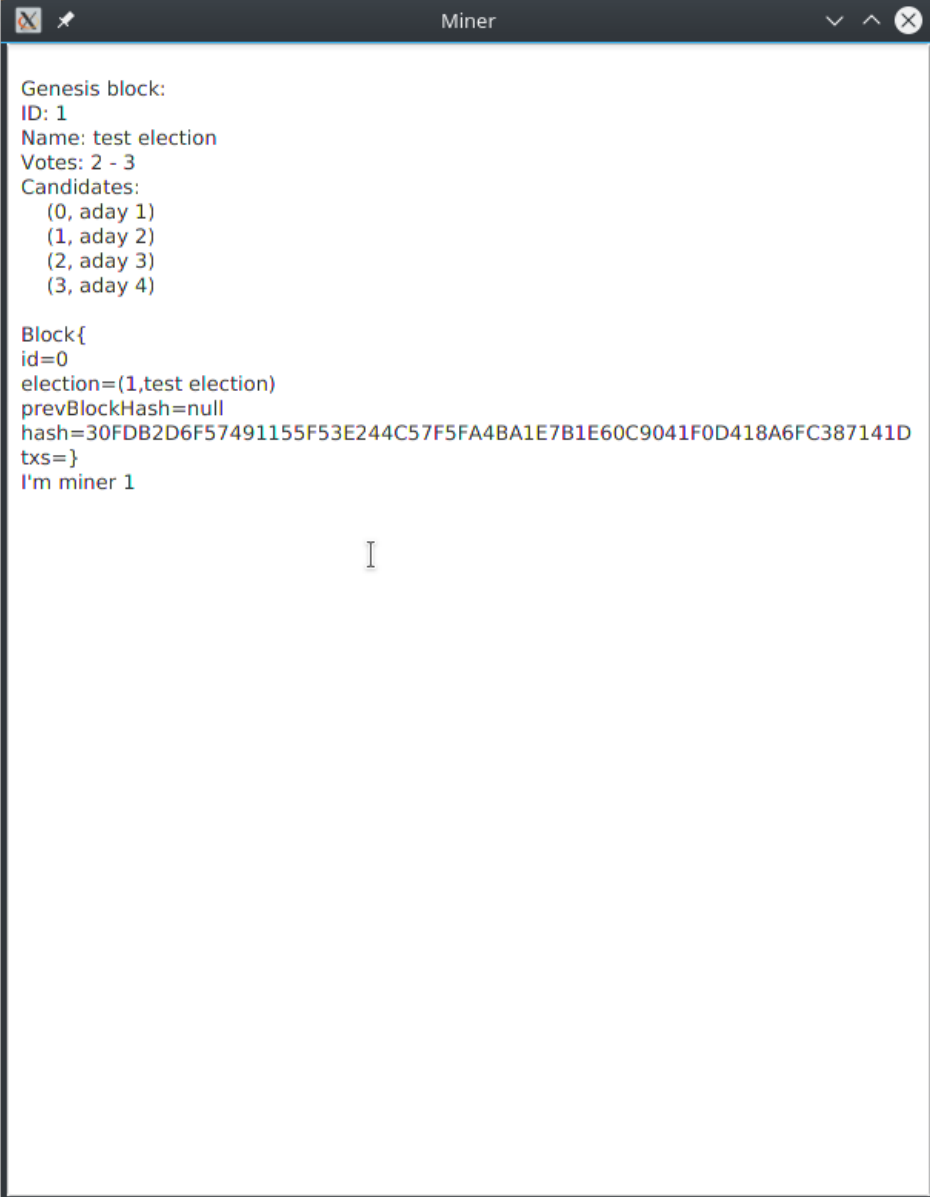
4.5.2. Kullanıcı Arayüzü

Program başlatıldığında ŞEKİL 4.15'teki ekran gelir. Buradan anahtar çifti oluşturulabilir ve var olan anahtar çifti seçilebilir. Ayrıca sunucuya madenci mi gözlemci mi olarak bağlanılacağı belirtilir. Gözlemci seçildiğinde anahtar çiftine gerek olmadığından o kısım kaybolur.



ŞEKİL 4.15 Madenci İlk Ekran

Sunucuya bağlanıldığında ise gelen verilerin ve blokların gösterildiği ŞEKİL 4.16’teki ekran gelir.



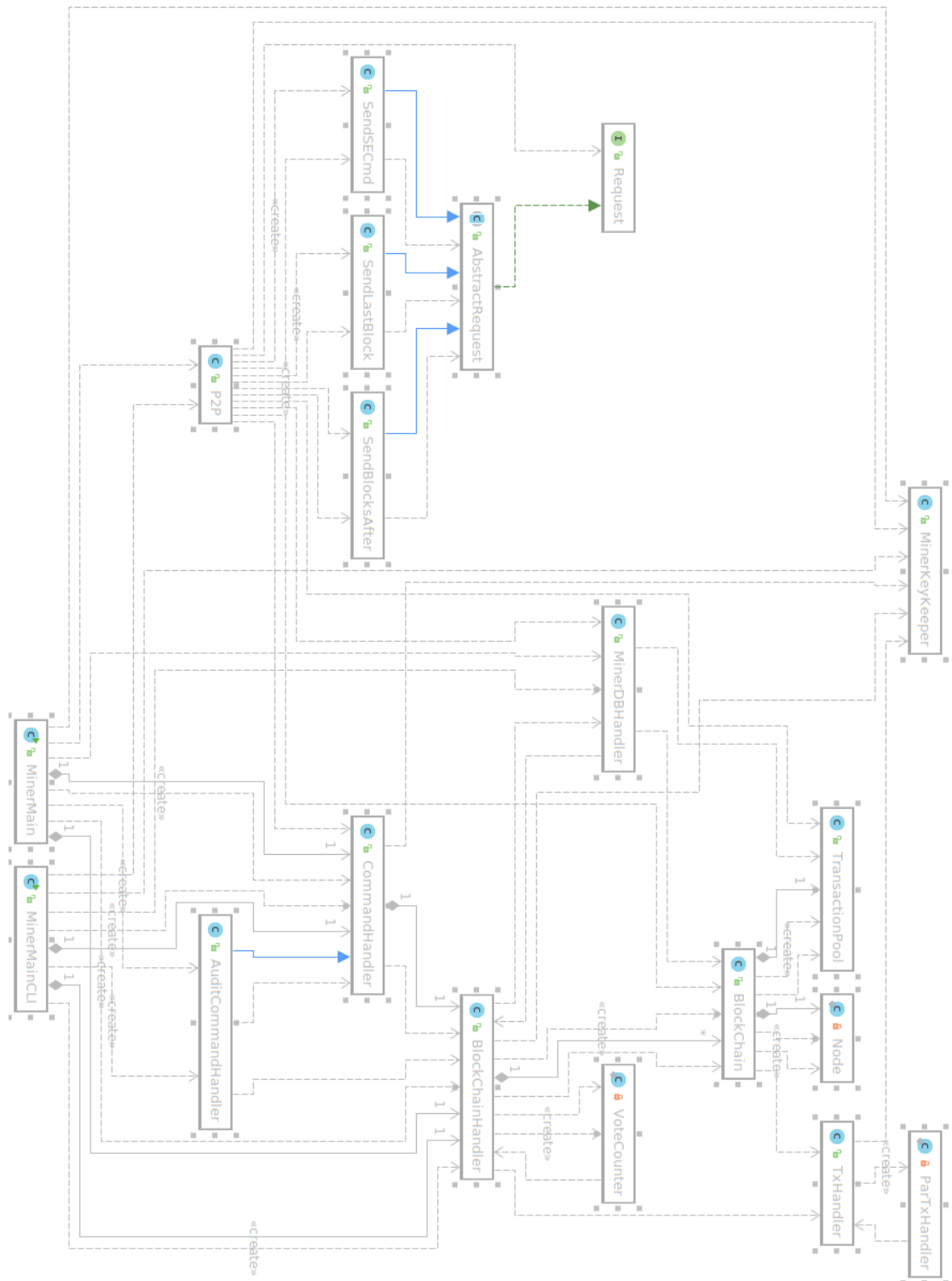
```
Genesis block:  
ID: 1  
Name: test election  
Votes: 2 - 3  
Candidates:  
  (0, aday 1)  
  (1, aday 2)  
  (2, aday 3)  
  (3, aday 4)  
  
Block{  
  id=0  
  election=(1,test election)  
  prevBlockHash=null  
  hash=30FDB2D6F57491155F53E244C57F5FA4BA1E7B1E60C9041F0D418A6FC387141D  
  txs=}  
I'm miner 1
```

ŞEKİL 4.16 Madenci Ekranı

4.5.3. Modül Yapısı

Madencinin içerdği sınıflar ŞEKİL 4.17’da gösterilmiştir. Sınıfların işlevleri şunlardır:

- MinerKeyKeeper: Kriptografik anahtarları saklayan sınıftır.
- MinerDBHandler: Veritabanı işlemlerini yapan sınıftır.
- P2P: Sunucu dışındaki aktörlerle olan iletişimi sağlayan sınıftır.
- BlockchainHandler/Blockchain: İşlem geçmişini saklayan sınıflardır. Blockchain handler her seçim için yeni bir blokzincir oluşturur ve blokzincirleri yönetir.
- CommandHandler/AuditCommandHandler: Sunucudan gelen komutları yerine getiren sınıflardır.
- Request Sınıfları: P2P haberleşmede sadece madenciler tarafından kullanılan isteklerdir.
- TxHandler: İşlemleri doğrulayan sınıftır.



4.5.4. İşlem Doğrulama Süreci

Sunucudan gelen 3.4.2.3te bahsedilen blok oluşturma komutu ping zamanını ve blok oluşturma periyodunu içerir. Madenci bu komutu aldığı anda $periyot - 4 * ping$ süresine zamanlayıcı kurar ve zamanlayıcı bitene kadar işlemleri doğrular. Eğer tüm işlemler doğrulanmadan zaman biterse doğrulayabildiği işlemler ile yeni bloğu oluşturur.

4.5.5. İşlem Doğrulamanın ve Oy Sayımının Paralleştirilmesi

Blok oluştururken yapılan işlem havuzundaki işlemlerin doğrulanması ve bitmiş seçimler için yapılabilen oy sayımı Java'nın ForkJoin çerçevesi (ing. framework) kullanılarak paralelleştirilmiştir. Yapılacak iş en fazla 175 iş içeren bloklara bölünerek ForkJoin çerçevesinin thread havuzuna verilerek 4 gerçek 4 sanal çekirdek içeren Intel i7-4700HQ işlemcisinde aşağıdaki sonuçlar elde edilmiştir:

- Havuzdaki n işlemin doğrulanmasında:
 - 500 işlem için 4,71 hızlanma
 - 1000 işlem için 4,75 hızlanma
 - 2500 işlem için 3,97 hızlanma
- Blokzincirdeki n oyun sayılmasında:
 - 500 oy için 2,08 hızlanma
 - 1000 oy için 4,21 hızlanma
 - 2500 oy için 3,39 hızlanma

elde edilmiştir.

Yığın boyutu olarak 50, 100, 125, 150, 175, 200, 250, 500, 750 boyutlarında yığınlar denenmiş, en iyi sonuçlar en fazla 175 iş içeren yığınlarda elde edilmiştir.

5. PROJENİN TESTİ

5.1. BLOK ZİNCİRİNİN TESTİ

Bu bölümde önceki çalışmada yapılanlar anlatılmıştır.

Blok zincirininin gerçekleşmesinde Bitcoin ve Kripto Para Teknolojileri [5] isimli kursun ödevlerinden yararlanıldı. Bu ödevlerde blok, işlem gibi temel sınıflar verilip blok zinciri ve işlem işleyici sınıfların gerçekleşmesi istendi ve ödev tesliminde çeşitli testler uygulandı. Gerçeklenen blok zinciri 27 testin 23ünden geçti. Geçemediği 4 test, projede iş-kanıtı (Proof-of-Work) dışında, dürüst madencilerin blokzincirlerinde oluşacak dallanmanın önüne geçen bir blok üretme algoritması kullanılmasından dolayı gerçekleşme gereği duyulmayan, blok zincirinin dallanmasıyla ilgili olan testlerdi.

Geçilen testler şunlardır:

- Herhangi bir işlem içermeyen bloğun işlenmesi.
- Tek bir geçerli işlem içeren bloğun işlenmesi.
- Birden fazla geçerli işlem içeren bloğun işlenmesi
- Çift harcama içeren bloğun işlenmesi
- Yeni kök (genesis) bloğun işlenmesi.
- Önceki bloğun hash özeti hatalı olan bloğun işlenmesi.
- Çeşitli geçersiz işlemler içeren bloğun işlenmesi.
- Önceki blokta harcanmış bir girdinin tekrar kullanılmaya çalışılması.
- Kendi dalında olmayan bir girdinin kullanılmaya çalışılması.
- Önceki bloklarda olan bir girdinin kullanılmaya çalışılması.
- Doğrusal blok zinciri oluşturulması.
- Doğrusal blok zinciri oluşturulduktan sonra kök bloğun üzerine kabul edilmemesi gereken blok eklenmesi.
- Havuzda işlem yokken blok oluşturulmaya çalışılması.
- Havuzda tek bir işlem varken blok oluşturulmaya çalışılması.
- Havuzda tek bir işlem varken iki kere blok oluşturulmaya çalışılması.

- Havuzda en uzun blok zincirinde bulunan bir işlem varken blok oluşturulmaya çalışılması.
- Havuzdaki işlem en uzun blok zincirinde kullanılmış bir girdiyi kullanıyorken blok oluşturulmaya çalışılması.
- Havuzdaki işlem en uzun blok zincirinde yokken ve en uzun blok zincirinde kullanılmış bir girdiyi kullanmıyorken blok oluşturulmaya çalışılması.
- Havuzdaki işlemlerin tamamı geçersizken blok oluşturulmaya çalışılması.
- En az iki kere havuza işlem eklenmesi ve blok oluşturulması.
- Havuza işlem eklenmesi, blok oluşturulması, oluşturulan bloğun üzerine o bloktaki girdiyi kullanan işlem içeren bir blok eklenmesi.
- Havuza işlem eklenmesi, blok oluşturulması, kök bloğun üzerine önceki bloktaki girdiyi kullanan bir blok eklenmesi.
- Kök bloğun üzerine birden fazla blok eklenmesi, blok oluşturulması.

Başarısız olunan testler ise şunlardır:

- Kök bloğun üzerinde birden fazla blok eklenmesi
- Doğrusal blok zinciri oluşturulduktan sonra kök bloğun üzerine kabul edilmesi gereken blok eklenmesi.
- Aynı uzunlukta iki dal oluşturulduktan sonra üretilen bloğun doğru dala eklenmek üzere oluşturulması.

5.2. SİSTEM TESTİ

Sistem biri ARM ikisi Intel mimarisine sahip üç farklı bilgisayarla yerel ağda test edildi. Bilgisayarların biri sunucu, üçü madenci olarak kullanıldı. Blok oluşturma süresi üç saniye olarak ayarlandıktan sonra aynı anda iki seçim başlatıldı, test için yazılan istemci programı eş zamanlı olarak iki seçimde de dörder thread ile 2500 adet oy kullandı ve kullandığı oyları ekrana yazdı. Oy kullanma işlemi devam ederken her madenci bir kere kapatılıp açıldı. Oy kullanma işleminin sonunda seçim sonuçları doğrulandı, sonrasında sunucu kapatılıp açılarak yeni bir seçim başlatıldı. Böylelikle madencilerin ve sunucunun kaldığı yerden başlatılabilmesi, işlem kaybı olmaması, tüm işlemlerin doğrulanmasının yetiştirilememesi durumunda blok oluşturulması gibi durumlar test edildi.

5.3. MADENCİNİN TESTİ

Madencinin sunucu ile konuşan sınıfına tam yol kapsama (ing. full path coverage) testi uygulanmıştır.

6. TARTIŞMA VE SONUÇ

Proje kapsamında anonim, güvenilir ve şeffaf bir seçim sistemi geliştirilmiştir. Ancak kör imzalar, anonimlik için yeterli değildir. Kullanıcılar ve işlemler, işlem zamanı veya kullanılan bilgisayarın özellikleri (IP adresi, vb.) kullanılarak eşleştirilebilir. Dolayısıyla anonimlik elde etmek için jetonun üretilmesiyle oy kullanma arasında zaman farkı olmalı ve en azından açık anahtarı kaydetme ve oy kullanma işlemleri Tor [8] benzeri anonim iletişim kanalları üzerinden yapılmalıdır. Örneğin USB üzerinden çalıştırabilir bir işletim sistemi olan Tails [9], internete Tor aracılığıyla bağlanır ve bu amaç için kullanılabilir.

Sistem, giriş bölümünde bahsedilen, Cenevre Eyalet Konseyi'nin gerekliliklerin çoğunu yerine getirmektedir. Yerine getirmediği gereklilikler şunlardır:

1. Seçim sistemi DoS saldırılarına karşı dayanıklı olmalıdır.
2. Seçmenler kimlik hırsızlığına karşı korunmalıdır.
3. Herhangi bir seçmenin oy kullandığını kanıtlamak mümkün olmalıdır.

KAYNAKLAR

- [1] Diffie, Whitfield, and Martin Hellman. "New directions in cryptography." *IEEE transactions on Information Theory* 22.6 (1976): 644-654.
- [2] Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM* 21.2 (1978): 120-126.
- [3] Chaum, David. "Blind signatures for untraceable payments." *Advances in cryptology*. Springer US, 1983.
- [4] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008): 28.
- [5] Narayanan, A., Bitcoin and Cryptocurrency Technologies [online], Coursera, <https://www.coursera.org/learn/cryptocurrency> [Ziyaret Tarihi: 20 Aralık 2017]
- [6] Buterin, V., Proof of Stake FAQ [online], <https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ> [Ziyaret Tarihi: 20 Aralık 2017]
- [7] kernel, Internet Voting: A Requiem for the Dream [online], <http://phrack.org/issues/69/11.html#article> [Ziyaret Tarihi: 20 Aralık 2017]
- [8] , Tor: Overview [online], <https://www.torproject.org/about/overview.html.en> [Ziyaret Tarihi: 31 Aralık 2017]
- [9] , Tails - About, <https://tails.boum.org/about/index.en.html> [Ziyaret Tarihi: 31 Aralık 2017]