



ONDOKUZ MAYIS ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

SANAL SINIF ARACI (SaSA)

SEMİNER RAPORU

SABİRE ESRA ÖZCAN

KÜBRA ATASEVER

DİCLE YENİGÜN

BEYZANUR KAYNAK

Danışman

PROF. DR. ERDAL KILIÇ

OCAK, 2021

TEŞEKKÜR

Danışmanımız Sayın Prof. Dr. Erdal KILIÇ'a ve Arş.Gör. Zinnet Duygu AKŞEHİR'e ilgileri ve yardımlarından dolayı teşekkür ederiz.

Ö Z E T

Yazılım yaşamının tüm yönlerine dahil olmaktadır. Eğitim hayatında ve iş hayatında özellikle verimlilik için faydalı olabilmeyi amaçlayan sanal sınıf aracı; aslında kişinin kendi başına da verimli çalışabileceği, yakınlarıyla görüşebileceği bir araç haline gelmektedir. Her ne kadar hedef başlangıçta online eğitim ve scrum toplantıları gibi profesyonel kullanım odaklı olsa da; yazılımı kullanıcının hayal gücünden etkilenerek bir projeye dönüştürmek ve bu projenin daha büyük kitleler tarafından memnuniyetle kullanılmasını sağlamak gelecekte ulaşmak istenilen noktadır.

Bu bağlamda hazırlanan projede hem eğitmenlerin ihtiyacı olan görüntülü/sesli konuşma, veri paylaşımı beyaz tahta gibi imkanlar olacak; hem de öğrencilerin derslerden soyutlanmaması için chat ortamı ve kolayca öğrenip giriş yapabilecekleri bir arayüz olacaktır. İş ortamında ise anlık görev dağılımları yine beyaz tahtada çizilebilecek; hazırlanmış olan dosyalar, dokümanlar sistem üzerinden paylaşılabilecektir.

Özetle bu rapor; günümüzde hala kullanılan ve literatüre geçmiş olan örnekler göz önünde bulundurularak hazırlanmıştır. Bahsedilen imkanları sağlayan bir sistemin hangi teknolojilerle oluşturulduğu, neden bu teknolojilerin tercih edildiği, adım adım hangi teknolojinin nerede ve nasıl kullanıldığını göstermektedir.

Anahtar kelimeler: Çevrimiçi Eğitim, Gerçek Zamanlı İletişim, WebRTC, Eşler Arası İletişim (Peer to Peer Communication), Sinyalizasyon.

İÇİNDEKİLER

I GİRİŞ

1	AMAÇ	2
2	LİTERATÜR ÖZETİ	4
3	TANIM	6
3.1	Sanal Sınıf Nedir?	6
3.2	Sanal Sınıfların Avantajları	7
3.2.1	Fiziksel Ortama Göre Daha Kolay Yönetim	7
3.2.2	Düşük Maliyet	7
3.2.3	Mesafe En Aza İndirilir	8
3.3	Sanal Sınıfların Dezavantajları	8
3.3.1	Daha Fazla Teknolojik Eğitim Gerekir	8
3.3.2	Daha Çok Altyapı Gerekir	8
3.3.3	Verimlilik	9

II MATERYAL

4	WEBRTC	11
4.1	WebRTC Nedir?	11
4.1.1	WebRTC ve Avantajları	12
4.1.2	WebRTC Arayüzünün Amacı	13
4.2	WebRTC API	14

4.2.1	MediaStream (getUserMedia)	14
4.2.2	RTCPeerConnection	15
4.2.3	RTCDataChannel	17
5	SIGNALR	19
5.1	SignalR Nedir?	19
5.2	İletişim Yöntemleri	20
5.2.1	WebSocket	21
5.2.2	Server-sent Events (SSE)	22
5.2.3	Long Polling	23
5.3	Azure SignalR Nedir?	23
5.3.1	Azure SignalR'nin Kullanıldığı Alanlar	24
5.4	Neden WebRTC?	25
III YÖNTEM		
6	YÖNTEM	27
6.1	WebRTC	28
6.1.1	WebRTC'de Ağ İletişimi	28
6.2	Sinyalizasyon Mekanizmaları	29
6.3	NodeJS	30
6.3.1	Socket.IO	31
6.4	PdfJs	32
6.5	MongoDB	32
IV SONUÇ		
7	SONUÇ	37

KAYNAKÇA	38
EKLER	42

ŞEKİLLER LİSTESİ

1	WebRTC Logo [1]	11
2	Dört Neden ile WebRTC	12
3	WebRTC ile Gerçek Zamanlı Veri Paylaşımı [2]	13
4	MediaStream'ın Ses ve Görüntü Akışını Döndürmesi	14
5	P2P (Peer-to-Peer)	15
6	WebRTC İletişim Modülü	17
7	WebRTC-IP	17
8	WebRTC Serverları	18
9	RTCDataChannel	18
10	SignalR ile gerçek zamanlı veri paylaşımı [3]	19
11	Taşıma önceliği [4]	21
12	WebSocket kullanımı	21
13	Konuşma örneği	22
14	SSE kullanımı	22
15	HTTP long polling	23
16	Azure SignalR Hizmetinin ASP.NET Core'da uygulama sunucusuyla kullanılan mimari	24
17	Sanal Sınıf Aracı Oluşturmak	27
18	WebRTC ICE Diyagramı	29

19	NodeJs'in Düşük Gecikme ile Çalışması	31
20	Socket.IO	32
21	MongoDb'nin Logosu[5]	33
22	Genel Mimari	33
23	Use-case Diyagramı	34
24	Faaliyet Diyagramı	35
25	Gantt Diyagramı	43

BÖLÜM: I

GİRİŞ

A M A Ç

Uzaktan eğitim sürecinde yararlı olması planlanan, özellikle içinde bulunduğumuz bu salgın döneminde eğitimde yaşadığımız çeşitli sorunlara çözüm olmak amacıyla bir **Sanal Sınıf Aracı(SaSa)** geliştirilmesi asıl amaçtır.

Sanal sınıf; okul, etüt merkezi, kurs gibi eğitim veren merkezlerdeki eğitim ortamının dijital bir kopyasıdır. Pek çoğu insanın karantina altında kaldığı ve eğitim alanlarının oldukça daraldığı bu dönemde sanal sınıflar bir nevi kurtarıcı olmuştur. Bu sınıflar sayesinde akademik ve sosyal yaşantının, beklenmedik ve olağanüstü durumlarda eğitimi oldukça az etkilemesi asıl hedefler arasındadır.

Uzaktan eğitime bütçe ayıran kurum sayısı artmaktadır. Bununla beraber uzaktan eğitimin dezavantajları, uzaktan eğitim sistemlerine yöneltilen eleştirilerin odak noktası olmaktadır. Oluşturulacak olan sistemde amaç; bu dezavantajların önüne geçebilecek avantajları kullanarak tercih edilmektir. Sanal sınıf araçları, oluşturulan bu çevrim-içi ortamda kullanıcılar arasında gerçek zamanlı görüntü, ses ve veri iletişimini sağlayan araçlardır. Dolayısıyla kullanıcılara hem çok yönlü bir iletişim sağlamak hem de öğrencilerin birlikte çalışabilecekleri bir ortam sunarak aldıkları eğitimi daha eğlenceli bir hale getirmelerini sağlamaktadır. Bütün bunlar göz önüne alındığında, hem öğrencilere hem de öğretmenlere yararlı olması amaçlanan

uygun bir ortam geliřtirmek ve temeli atılan, henüz oluřum ařamasında olan pazarda bir yer edinmek asıl hedeftir. Hedeflenen sistemin planlanan özellikleri ařağıdaki gibidir:

- Görüntülü ve sesli iletişim imkanı
- Dosya paylaşımı imkanı
- Chat ortamı
- Beyaz tahta ortamı
- Güvenli bir ortam yaratabilmek için kullanıcı adı-řifre ile giriş imkanı
- Basit ve kullanıcı dostu bir arayüz tasarımı

LİTERATÜR ÖZETİ

Bu bölümde Sanal Sınıf Aracı ile ilgili literatüre geçmiş çalışmalardan bahsedilmiştir. Yapılan alan araştırmalarına ve mevcut çalışmalara bakılacak olunursa;

Kumar ve arkadaşları [6] yaptıkları çalışmada, kullanıcıların video konferans ile derslere katılabildiği, beyaz tahtaya erişilebilen, sunu ve not paylaşılabilen, notları ve ödevlerin indirilmesine olanak sağlayabilen ve sınav sistemini içeren bir çalışmadan bahsetmişlerdir. Ayrıca projede eş zamanlı görüntü iletişimi için WebRTC Api'lerinden faydalanılmıştır.

Onihunwa ve arkadaşları [7] 2020 senesinde WebRTC teknolojisini kullanarak oluşturdukları sanal sınıf projesini sunmuştur. Uygulamada bütün kullanacıların görüntülü görüşmesi sağlanmakla birlikte, verimlilik ve ölçeklenebilirlik için deneyler yapılmıştır.

Erkan ve Altun [8] 2015 yılında yapmış oldukları çalışmada internet ortamında lisans seviyesinde bir dersin etkileşimli öğretimine olanak sağlayacak bir proje ve bu projenin geliştirme aşamaları anlatılmaktadır. Gerçekleştirilen Java ve Web tabanlı sanal sınıf ve sanal laboratuvar, eğitimde zaman ve ortam kısıtlamalarının aşılabilmesine olanak sağlamaktadır. Burada kullanıcılara kontrol öğeleri sunulmuş ve öznitelikleri değiştirebilme imkanı verilmiştir.

Jansen [9] tarafından hazırlanmış olan tezde WebRTC kullanılarak yapılan video konferans uygulamalarının performans analizini yapmıştır. Bu çalışmada WebRTC'nin arka planda verileri gizlemesi ve güvenlik bazında kontrol etmesinden bahsetmiştir. Bu güvenlik bazında

kontrolün performansa ne derece etkilediği konusunda incelemeler yapmıştır. WebRTC'nin çalışması için bazı tarayıcılarda ek eklentiler gerektiğinden bahsetmiştir. Ayrıca WebRTC'nin mobil cihazlar için gönderilen video akışlarının veri hızlarını yavaşlattığından ve çözünürlüğü düşürdüğünden bahsetmiştir (WebRTC, geliştiricinin platformda yapılan video görüşmesi özellikleri hakkında kolayca bilgi edinmesine izin vermez. Bunun nedeni, WebRTC'nin ağ tıkanıklığını işleme biçiminin, tarayıcının kodunda tamamen gizlenmiş olmasıdır).

Holm ve Löf [10] WebRTC uygulamalarının mimarisi ve dizaynı ile ilgili bitirme projesi geliştirmişlerdir. 2019 yılında yazmış oldukları bu tezde WebRTC ve benzeri uygulamaların analizinden bahsedilmiş; akışları benzersiz tutan ölçeklenebilir, yüksek performanslı ve verimli bir WebRTC uygulaması elde etmek için WebRTC uygulamaları için mevcut tasarım modelleri araştırılmış ve özellikle WebRTC üzerine yoğunlaşmıştır. WebRTC uygulamaları için JavaScript teknolojileri kullanılmış, mevcut tasarım modelleri üstünde çalışılmıştır (Bu çalışmalarda JavaScript teknolojileri kullanılarak nasıl uygulanabileceği gösterilmektedir). Tasarım modelleri olarak MCU (Multipoint Control Unit) ve SFU'dan (Selective Forwarding Unit) bahsedilmiştir.

TANIM

Teknolojinin gelişmesiyle ve kullanılan cihazların iyileşmesiyle, eğitimde teknolojinin faydalarından yararlanmak son yıllarda neredeyse gereklilik haline gelmiştir. Günümüzde özellikle içinde bulunduğumuz Covid-19 salgını döneminde uzaktan eğitim ortamlarının ve sanal sınıf projelerinin önemi çok daha iyi anlaşılmıştır.

3.1 *Sanal Sınıf Nedir?*

Sanal sınıf araçları; kullanıcıların çevrim içi olarak katıldığı görüntü, ses ve veri paylaşımı sağlayabildiği bir öğrenme sistemidir. Uzaktan eğitimin gerçekleştirilmesi ve uygulanmasında sanal sınıfın etkisi, eğitimin etkililiği açısından çok önemlidir. Böyle bir ortamda hem öğrenciler öğretmenlerin sunduğu kaynaklardan daha kolay ve etkili bir şekilde yararlanırlar, hem de öğretmenler proje, ödevlendirme ve değerlendirme etkinliklerini daha kolay şekilde gerçekleştirirler.

Günümüzde sanal sınıf aracı olarak kullanılan uygulamalar genellikle Google Classroom, Edmodo ve Schoology olarak verilebilir. Bu sanal sınıflar kendi aralarında özelliklerine ve

türlerine göre değişkenlik gösterir. Bunlar dışında konferans verilebilecek birçok çevrim içi platform ile sanal sınıf ortamı oluşturulabilir.

Sanal sınıflar eğitim haricinde; seminerler, video konferansları, danışmanlık hizmeti gibi uzaktan iletişimle gerçekleştirilebilecek iş etkinliklerinde de kullanılır.

3.2 *Sanal Sınıfların Avantajları*

3.2.1 *Fiziksel Ortama Göre Daha Kolay Yönetim*

Sanal sınıf ortamında fiziksel bir ortam bulunmadığı için, öğretmenler çevresel faktörlerden daha az etkilenir ve öğrencilerle birlikte bilgiye daha çok odaklanabilir. Normal eğitim şartlarında karşılaşılan insan yoğunluğu, kullanılan araç gereçlerdeki eksiklik ya da bozukluk gibi fiziksel problemler, sanal sınıf ortamında bulunmaz.

3.2.2 *Düşük Maliyet*

Alışılmış eğitim ortamlarında birçok faktörün ayarlanması gerekir. Örnek olarak okul, sınıflar, laboratuvarlar için malzemelerin karşılanması, öğrenci sayılarının sınıflara göre ayarlanmasına benzer, daha çok bu fiziksel ortamları yaratmak için oluşan problemlerle karşılaşılır ve bunlar çok maliyet gerektirir. Diğer yandan sanal sınıflar, bilgisayar ve artık mobil cihazlarla da dahil iyi bir internet erişimi sağlandığında daha az maliyetli olabiliyor. Sadece bunlar yeterlidir demek günümüze göre pek gerçekçi değildir ama yine de bilinen sınıf ortamından daha az bir maliyet gerektireceği kesindir.

3.2.3 *Mesafe En Aza İndirilir*

Fiziksel bir ortamda ,başka bir şehre ulaşmak maliyet ve zaman gerektirir. Buna rağmen sanal sınıf ortamında istenilen her yere uzaklık bakılmaksızın kolayca erişilebilir [11].

3.3 *Sanal Sınıfların Dezavantajları*

Sanal sınıfların ve teknolojik ortamların ancak etkili bir şekilde kullanıldığında birçok avantajı vardır ve anlam kazanır. Bunlar gerçekleştirilemediğinde doğal olarak dezavantajları oluşur.

3.3.1 *Daha Fazla Teknolojik Eğitim Gerekir*

Sanal sınıfın sağladığı yararlardan faydalanmak için ,karşıladığı teknolojik araç gereçlere hakim olmak gerekir. Ortamın gerektirdiği teknolojik bilgiye sahip olmamak ya da hangi koşullarda ve zamanlarda kullanılacağını ayarlayamamak faydadan çok zarara sebep olur.

3.3.2 *Daha Çok Altyapı Gerekir*

Günümüzde her ne kadar altyapı sorunları fiber kablolar ile büyük oranda çözülmeye çalışılıyor olsa da hala internete ulaşım konusunda sıkıntılar yaşayan bölgeler vardır. İnternete erişemeyen ya da sınırlı kotayla bağlantı sağlayan insanlar için sanal sınıf zorlayıcı olabilir.

3.3.3 Verimlilik

Eğer sistem düzenli olarak kullanılmazsa, eğitimden alınan verim istenmeyecek bir şekilde düşebilir. Örneğin öğretmen gerekli dokümanları paylaşmış olsa dahi öğrencilerin gerekli dikkati gösterip ilgilenmemesi eğitimden beklenen verimi bir anda düşebilir [11].

Avantajlar ve dezavantajlar değerlendirildiğinde; Sanal sınıflar, bilgisayar ve sağlıklı bir internet erişimi gibi ihtiyaçların karşılanabildiği zamanlarda çok verimlidir. Tabii ki imkanlar doğrultusunda yüz yüze eğitimin yerini tam anlamıyla tutamaz. Böyle düşünüldüğünde sanal sınıf ortamı, fiziksel olarak oluşturulamayan sınıflar için gerekli ve en etkili ortam olmakla beraber geleneksel sınıflar için ise iyi bir teknolojik destektir. Şuan tüm dünyada bulunan salgın durumu için değerlendirildiğinde ise sanal sınıfın, eğitimin aksamaması ve dönemin en verimli şekilde geçirilebilmesi için, gerekli ve zorunlu bir çözüm olduğu görülür.

BÖLÜM: II

MATERYAL

WEBRTC

4.1 *WebRTC Nedir?*



Şekil 1: WebRTC Logo [1]

Web Tabanlı Gerçek Zamanlı İletişim arayüzü (WebRTC-Web Real-Time Communication), W3C (World Wide Web Communication) tarafından tanımlanmıştır. Logosu Şekil 1’de verilen WebRTC, tarayıcılar üzerinden sesli ve görüntülü görüşmeyle birlikte dosya paylaşımını

hedefleyen bir uygulama geliştirme arayüzüdür. WebRTC'nin getirdiği en önemli yenilik; bütün bunları yaparken kullanıcının internet tarayıcısına ekstra bir plug-in (adobe flash, active-x) yüklenmesini zorunlu kılmıyor olmasıdır. WebRTC şu an itibariye Google, Mozilla ve Opera tarafından desteklenmektedir.

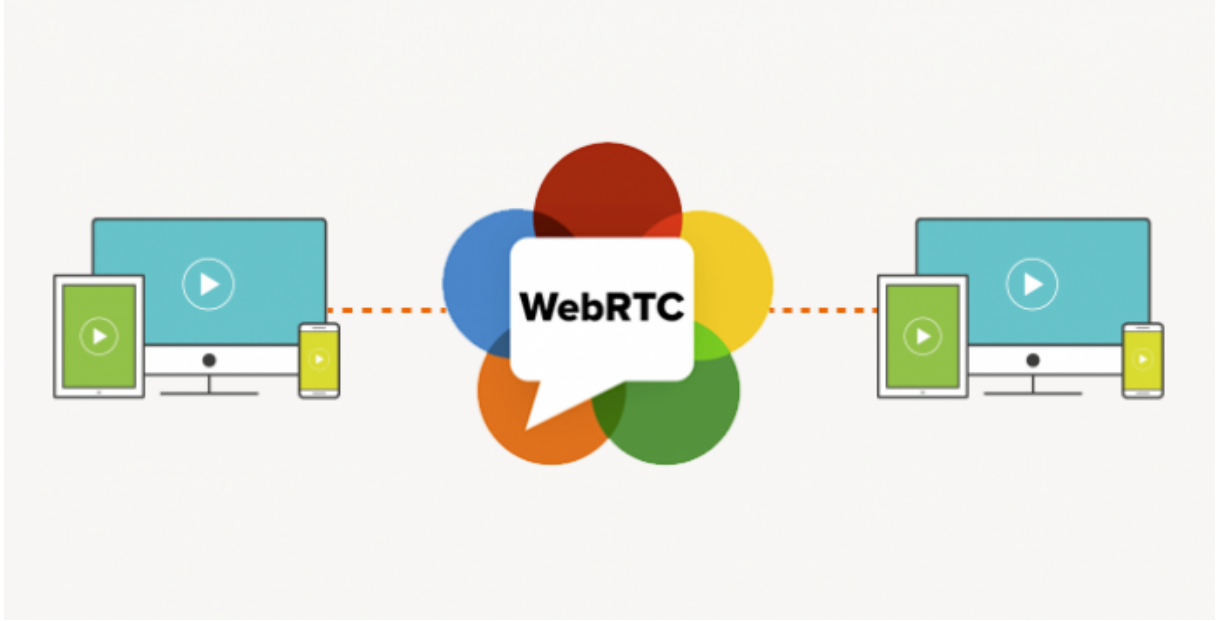
4.1.1 WebRTC ve Avantajları



Şekil 2: Dört Neden ile WebRTC

Şekil 2’de de görüldüğü üzere WebRTC’yi diğer teknolojilerden ayıran özelliklerden başka biri tamamen ücretsiz ve açık kaynak kodlu olmasıdır. Bununla birlikte uçtan uca (peer-to-peer) bağlantı sağladığı için çok düşük maliyetlerle veri transferi gerçekleştirebiliyor olmasıdır (Data aktarımında bir chat uygulaması da düşünebiliriz. Text aktarımı, string aktarımı vs.). Ayrıca karmaşık medya akışı (complex media streaming) sistemleriyle; medyayı kodlaması/çözümlemesi (encoding/decoding) ile uğraşmak gerekmemektedir. Üstelik bütün bu işlemleri hem webde hem de mobilde yapılabilir.

4.1.2 WebRTC Arayüzünün Amacı



Şekil 3: WebRTC ile Gerçek Zamanlı Veri Paylaşımı [2]

WebRTC'nin amacı, internet tarayıcısından basit javascript API'leri ve HTML5 kullanılarak gerçek zamanlı iletişim uygulamalarının geliştirilmesini Şekil 3'teki gibi sağlamaktadır. Web API'yi kullanması gereklidir. Web API sayesinde geliştirilen web uygulamasının yorumlanabilmesinde ise WebRTC destekli tarayıcıda şu öğelerin bulunması gerekir:

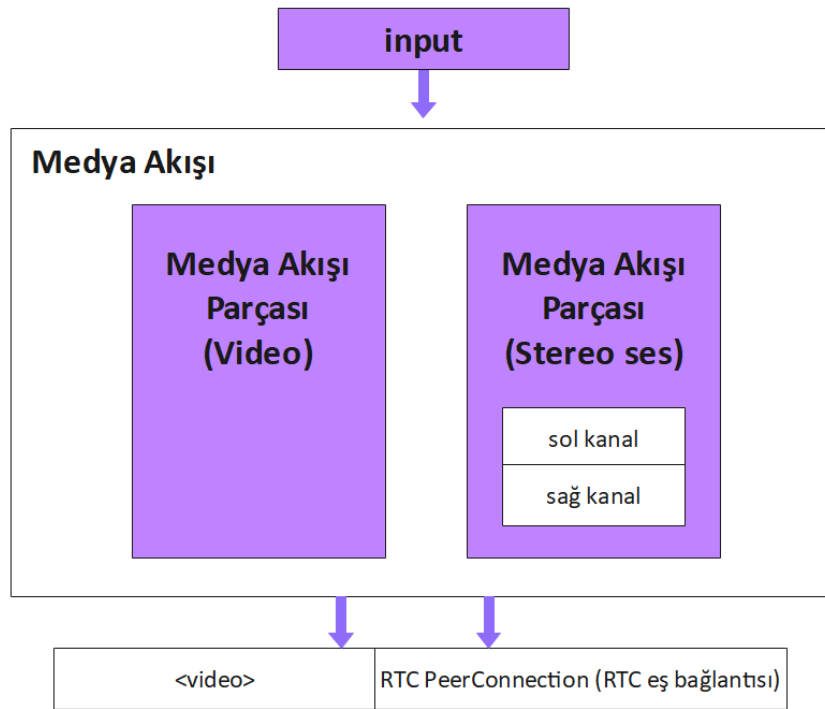
- WebRTC C++ API
- Session Yönetimi, Sinyalleşme
- Ses motoru
- Video motoru(vp9 codec) (H.264 codec'inin yerini almıştır. Vp9 daha hızlıdır, daha az bant genişliği kullanır ve daha performanslıdır. 4K videoları V9 codec sayesinde gerçekleşmektedir.)

- Transport (RTC, RTP protokolünü kullanılmaktadır. Görüşme sağlanabilmesi için tarayıcılarda RTP protokolü olmalıdır.)

4.2 WebRTC API

WebRTC anlık görüntülü/sesli arama ve veri paylaşımı yaptığımız Whatsapp, Snapchat, Messenger gibi uygulamalarda kullanılmaktadır. WebRTC API'sını; MediaStream (Medya Akışları), PeerConnection (Veri akımı ve p2p iletişimin yönetilmesi) ve DataChannel (verinin p2p iletilmesi) olarak üç parçaya bölüp inceleyebiliriz.

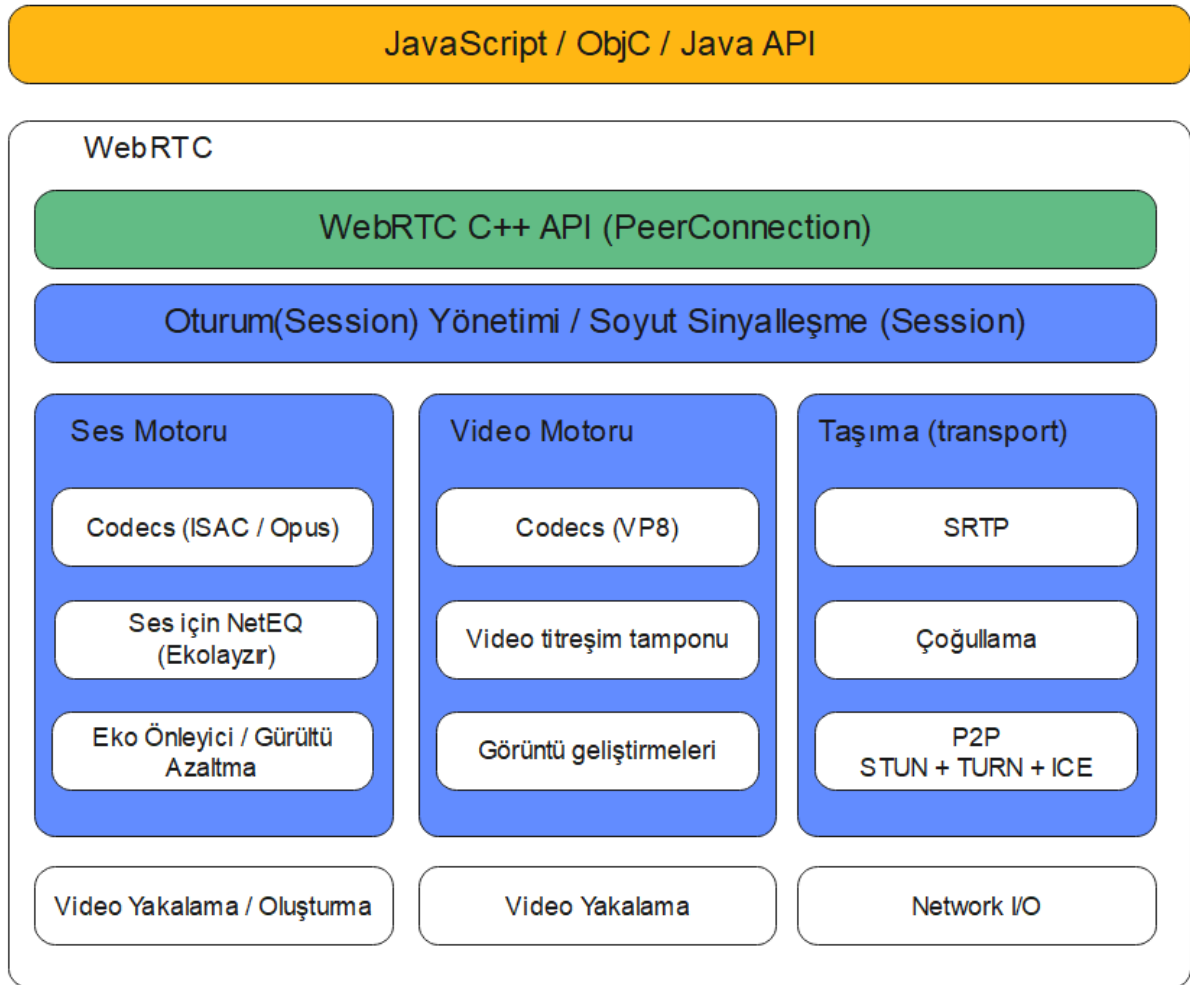
4.2.1 MediaStream (getUserMedia)



Şekil 4: MediaStream'ın Ses ve Görüntü Akışını Döndürmesi

Kullanıcıları yönetmeye ve kullanıcılardan izin istemeye yarayan sinyalleri göndererek Şekil 4'teki ses, görüntü ve veri akışını döndürür. Kısaca kullanıcı aygıtlarını bir tarayıcı ile yönetmeye destek sağlar.

4.2.2 *RTCPeerConnection*



ISAC: Internet SpeechAudio Codec (İnternet Konuşma Ses Codec'i) / Opus: Ses codec'i

SRTP: Secure Real-time Transport Protocol (Güvenli Gerçek Zamanlı Aktarım Protokolü)

P2P: Peer to Peer

Şekil 5: P2P (Peer-to-Peer)

WebRTC, tarayıcılar arasındaki bağlantıyı sunucu olmadan direkt istemcilerle sağlamaktadır. WebRTC kullanılarak istemciler arasındaki iletişimi Şekil 5'te gösterildiği üzere Peer to Peer(uçtan uca bağlantı) kullanarak gerçekleştirmektedir. RTCPeerConnection, uçtan uca iletişimin yönetildiği WebRTC teknolojisinin temel bileşenlerinden biri olmakla beraber ses veya görüntü aramalarında güvenli medya akışını ve bant genişliğini yönetmeye yardımcıdır.

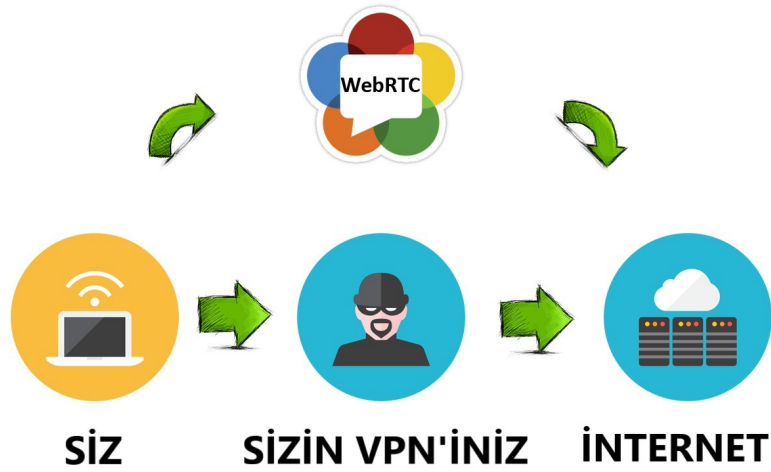
PeerConnection alt katmanlarda gerçekleşen, gerçek zamanlı iletişimi gerçekleştiren bütün işleri geliştiricisinden soyutlar ve eş zamanlı iletişimin ihtiyacı olan önemli işlemleri yönetir. Bu işlemlerden bazıları ise şunlardır:

- Medya içeriğinin karşısında bulunan kullanıcıya gönderilip alınması,
- Medyanın kodlanıp çözümlenmesi, güvenliğin, band genişliğinin ve paket yönetimindeki işlemlerin yapılması,
- Yankının giderilmesi (echo cancellation),
- Gürültünün azaltılıp önlenmesi (noise reduction and suppression).

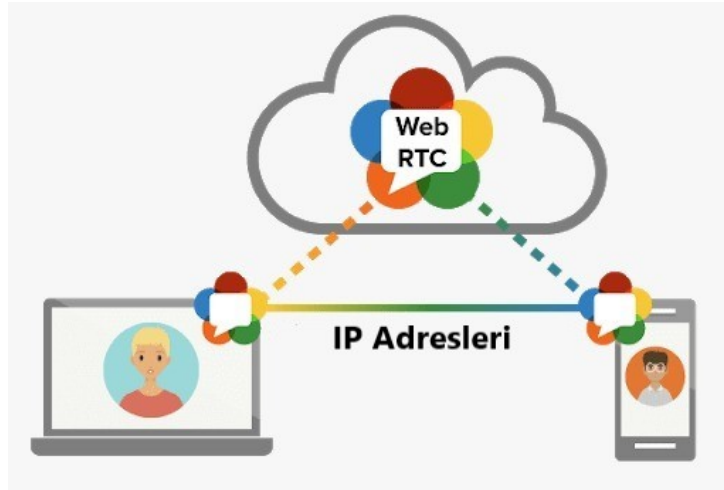
Peer to Peer Connection'da önemli olan, Şekil 6'daki gibi iki tarafın da birbirinin IP'sini bilmesi gerekmesidir.

Normalde bilgisayarlar kendi dış IP'lerini bilmezler, sadece local IP'lerini bilirler. Bu tarz durumlarda ise devreye Stun Server'lar girmektedir. Cihazlar Stun Server'dan IP'lerini öğrenmekte ve Şekil 7'deki gibi birbirlerine paket göndermektedirler. Cihazlar arasında bilgi paylaşımıyla p2p başlamış olmaktadır (Stun Server hizmetini Google başta olmak üzere birçok firma vermektedir.).

Bu bağlantının sağlanamadığı bazı durumlar karşımıza çıkabilmektedir. Cihazlardan biri bir şirket bilgisayarı olabilir veya şirketin wireless'ına bağlı mobil cihaz olabilir. Bu gibi durumlarda o bölgenin firewall'u bu duruma izin vermeyebilir ve bu durumda Turn Server



Şekil 6: WebRTC İletişim Modülü

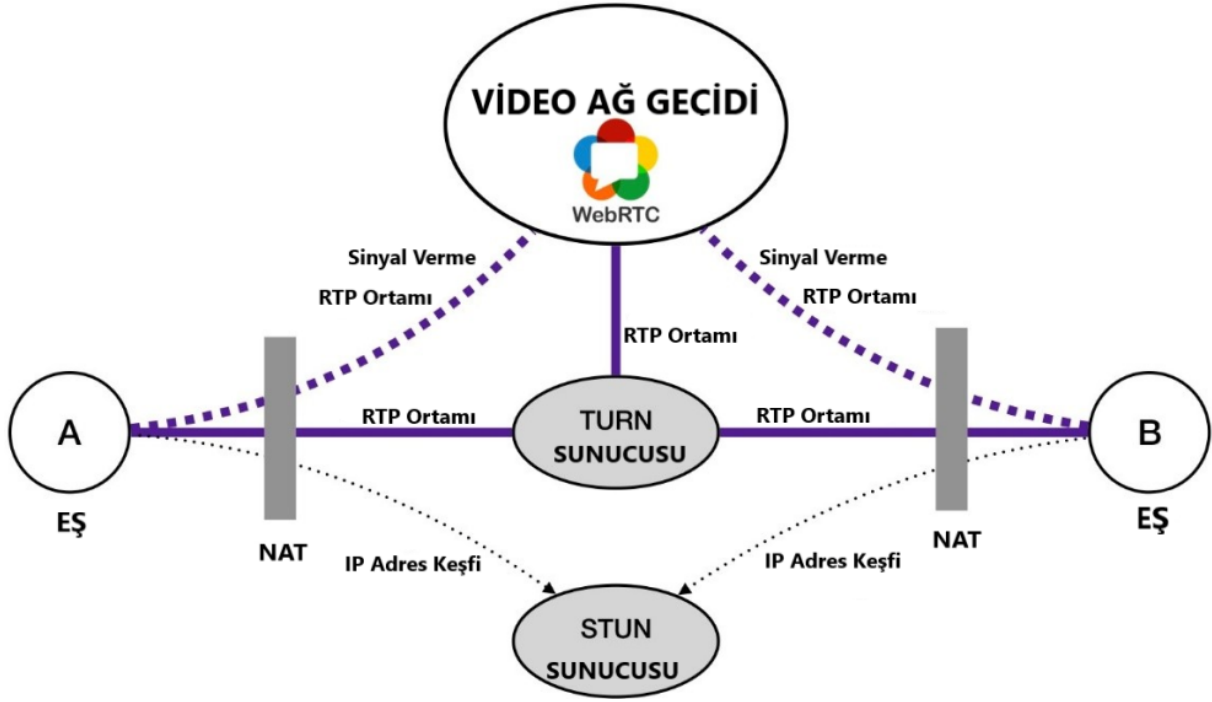


Şekil 7: WebRTC-IP

devreye girmelidir. WebRTC uçtan uca bağlantıyı (p2p connection) sağlayamaz ve bağlantıya Şekil 8'deki gibi Turn Server dahil olmuş olur.

4.2.3 *RTCDataChannel*

Bu API basit ve hızlı bir sohbet geliştirilmesine olanak sağlar. Sunucuda herhangi bir kurulumu gerektirmez. Yeniden yazılabilir ve belirli bir boyuta kadar herhangi bir formattaki



Şekil 8: WebRTC Serverları



Şekil 9: RTCDATAChannel

dosyaları aktarmaya olanak sağlar. Kısaca RTCDATAChannel kullanıcılar arasındaki veri haberleşmesine Şekil 9'daki gibi yardımcı olur. Veri kanalı (data channel), eş bağlantısı (peer connection) sağlanmış bileşenler arasındaki medya akışını (media stream) sağlar [12, 13, 14, 15, 16].

SİGNALR

5.1 *SignalR Nedir?*



Şekil 10: SignalR ile gerçek zamanlı veri paylaşımı [3]

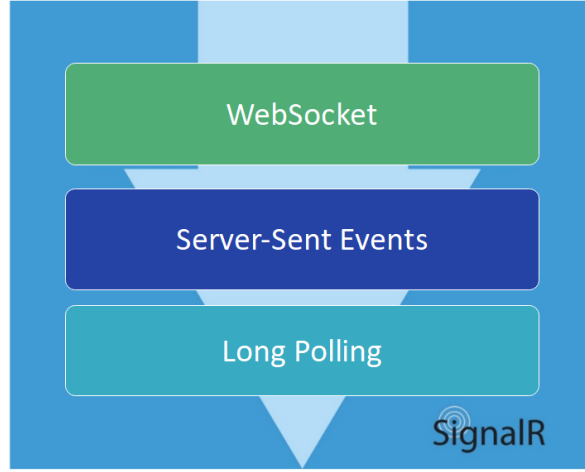
SignalR, açık kaynak kodlu bir .NET kütüphanesidir. Şekil 10'da da görüldüğü gibi gerçek zamanlı sistemlerin geliştirilmesi için yazılmıştır. Normalde kullanılan HTTP bağlantılarında sunucu-istemci arasında iletişim, bulunan her istekte yenileniyorken, SignalR kullanarak kurulan sunucu-istemci arasındaki iletişimde, sürekli olarak bir bağlantı sağlanmış olur. İletişim kurarken HTTP protokolü kullanılırken sayfayı her seferinde yenilemek gerekiyorken, SignalR'de herhangi bir veride değişiklik yapıldığında, sunucu bir Javascript fonksiyonunu çağırır ve sistemde bulunan diğer istemcilere haber gönderir.

SignalR veri transferini gerçekleştirmek için çeşitli bağlantı yöntemleri sağlar. Bunlar Long Polling (Uzun Yoklama), Server Sent Events (Sunucu Tarafından Gönderilen Olaylar), Web Sockets gibi bağlantılardır. Geliştirilen gerçek zamanlı sistemlerde SignalR kütüphanesi, veri transferi için kurulan bu bağlantıları otomatik olarak yapabilmeyi sağlar. Bağlantı kesildiğinde tekrar bağlanmayı kolaylaştırır. Hub ve kalıcı bağlantı (persistent connection) seçeneklerini sağlar.

Günümüzde özellikle sayıları artmış durumda bulunan mesajlaşma uygulamalarında SignalR kullanılabilmektedir. Gelen her yeni mesajı görüntülemek için her seferinde sayfayı yenilemek yerine sadece sayfadaki verileri güncelleyerek iletişimi sağlar. Bunun gibi gerçek zamanlı uygulamaları anlık güncelleyerek iletişimi gerçekleştirmede kolaylık sağlar. Günlük hayatta kullanılan mesajlaşma uygulamaları, gerçek zamanlı veri takip uygulamaları ve iş takip uygulamaları, SignalR'nin sağladığı bu kolaylıklarla geliştirilebilirler. Sanal sınıf oluşturulmasında da SignalR'nin gerçek zamanlı sistemlerin geliştirilmesindeki yararları kullanarak, anlık mesajlaşma kısmı ve anlık veri güncellemeleri gerçekleştirilebilir fakat diğer kısımların geliştirilmesi için tek başına yeterli olmayabilir.

5.2 İletişim Yöntemleri

SignalR'nin en önemli kolaylığı, sunucu ve istemci arasındaki gerçek zamanlı uygulamalarda kullanılacak olan ulaşım (transport) yöntemini, otomatik olarak seçmesidir. SignalR bağlantısı Şekil 11'de görüldüğü üzere, bir HTTP bağlantısı gibi başlayarak, uygunsa daha sonra WebSocket bağlantısının kurulumunu gerçekleştirir.



Şekil 11: Taşıma önceliği [4]

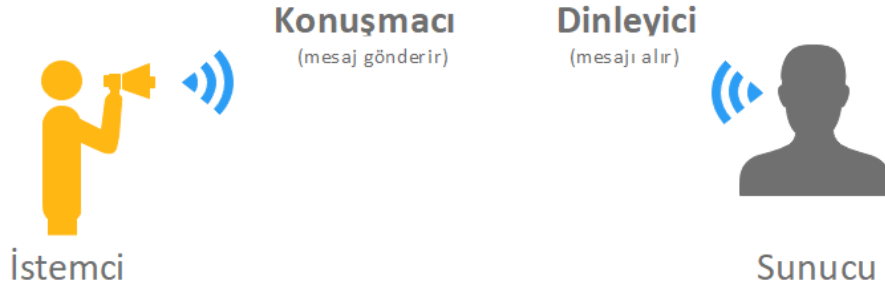
5.2.1 WebSocket



Şekil 12: WebSocket kullanımı

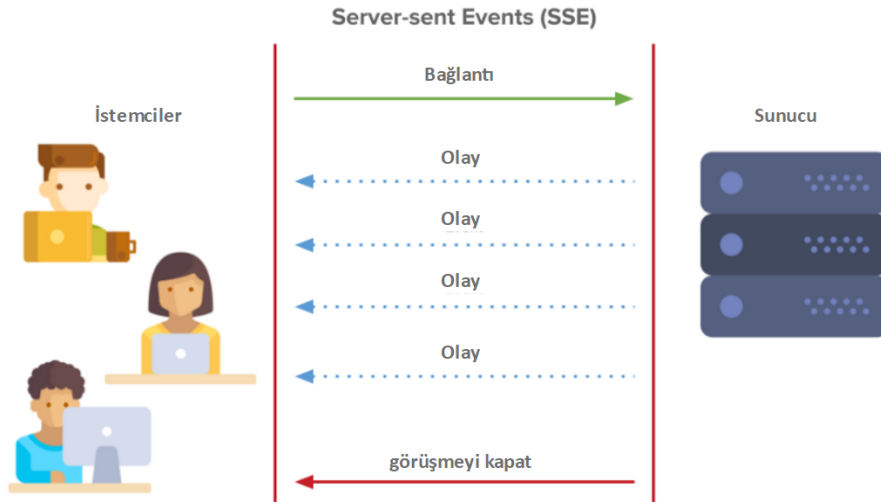
SignalR'nin en iyi iletim yöntemi WebSocket'tir. İstemci tarafındaki tarayıcı sunucuya istek gönderir ve sunucu kalıcı bağlantının oluşumunu gerçekleştirir. Bağlantı oluşturulduğunda istemci ve sunucu arasında TCP üzerinden çift taraflı gerçek zamanlı etkileşimin gerçekleşmesi Şekil 12'deki gibi sağlanır.

WebSocket yöntemi günlük hayattan bir örnek verildiğinde Şekil 13'deki gibi özetlenebilir. İstemci bir konuşmacı görevi yapar ve sunucu da bu mesajları dinleyen konumundadır.



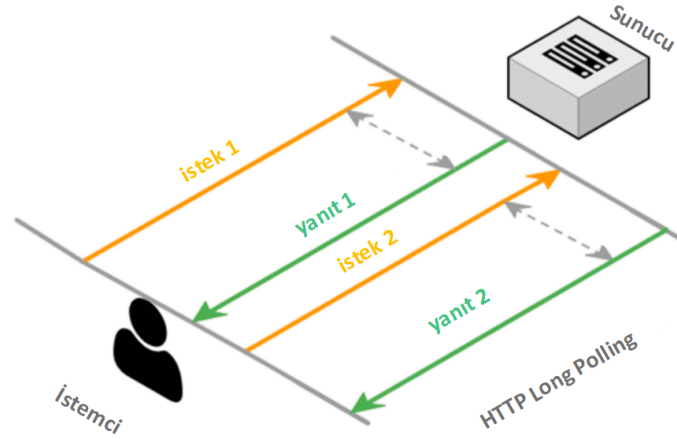
Şekil 13: Konuşma örneği

5.2.2 Server-sent Events (SSE)



Şekil 14: SSE kullanımı

İstemci ve sunucu arasında istek-yanıt ile bağlantı oluşturulur. Event-stream (olay akışı) yeni bir verinin yayınlanması için sunucuyu dinlemeye başlar. Sunucuda oluşan yeni bir veri event-stream üzerinden yayınlanır. İstemci tarafında Javascript üzerinden EventSource objesi ile yeni verilere erişim Şekil 14'deki gibi sağlanabilir.



Şekil 15: HTTP long polling

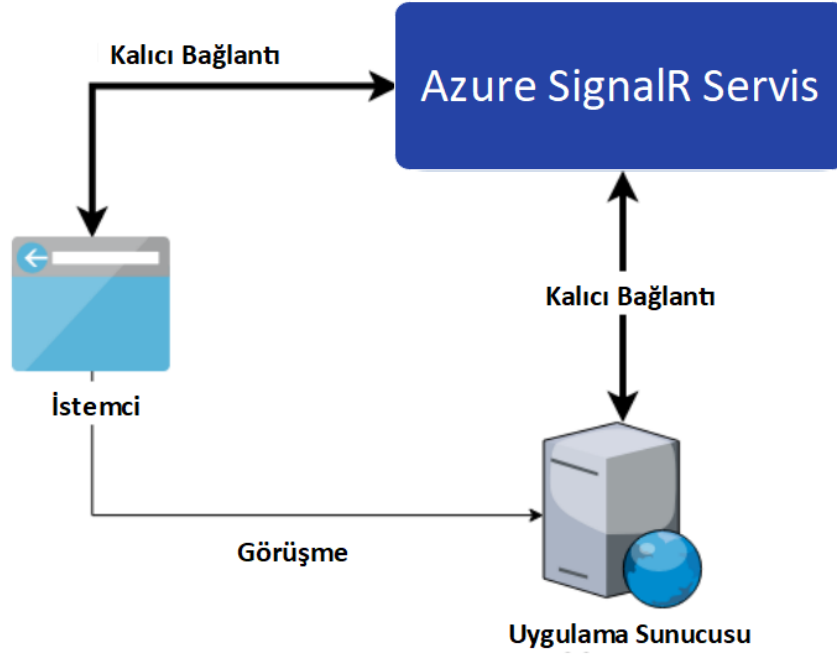
5.2.3 Long Polling

Şekil 15'te görüldüğü gibi, sunucu kendine gelen HTTP isteklerini bekletir ve ancak bir event olduğu durumlarda yanıt gönderir. Yeni bir istemci isteğiyle bağlantı sağlanmaya devam eder [3, 4].

5.3 Azure SignalR Nedir?

SignalR servisi üzerine kurulan ASP.NET Core SignalR uygulamaları ile uyumlu ve bu uygulamaları ölçeklendirmeye yarayan bir hizmettir. SignalR servisi olarak Azure'nin tüm özelliklerini içeren gerçek zamanlı mesajlaşmada kullanılan bir hizmettir. Gerçek zamanlı olarak verileri sunucudan istemciye gönderilmek istenen her senaryo için kullanılmaya imkan tanır.

Bu hizmet, web uygulamalarını gerçek zamanlı ölçeklendirmeye yarayan destek olarak da düşünülmektedir. Doğrudan uygulamada bulunmayan bağlantılar sayesinde maksimum verim aşılmaz [17].



Şekil 16: Azure SignalR Hizmetinin ASP.NET Core’da uygulama sunucusuyla kullanılan mimari

Bu hizmetin getirdiği avantajlardan biri istemciler sadece hizmete bağlanacağı için sunucuya aşırı yüklenme söz konusu olmamasıdır.

Dezavantaj olarak ise bu hizmeti kullanarak kalıcı bağlantılara destek verilemeyeceği ve sunucuda kullanılan .NET sürümü yükseltilmesi gerektiği söylenilebilir.

Şekil 16’ da görüldüğü gibi bu hizmette uygulama sunucusu tarafından istemcilerin istekleri dinlenmekte, bağlantıların yönetildiği görülmektedir.

5.3.1 Azure SignalR’nin Kullanıldığı Alanlar

Azure SignalR’nin kullandığı bazı alanlar ise şunlardır;

- Canlı video yayını yapmak için
- Beyaz tahta uygulamalarında kullanmak için

- Takım toplantısında kullanılmak üzere geliştirilen yazılımlar için
- Canlı sohbet odalarında veya gerçek zamanlı müşteri desteği vermek için
- Gerçek zamanlı konum izlemek için
- Gerçek zamanlı IoT ölçümleri için
- Gerçek zamanlı reklamlarda kullanılmak gibi çok farklı sektörlerde kullanılır.

5.4 Neden WebRTC?

Bahsedilen iki teknolojidten WebRTC'nin hem geliştiriciler hem de kullanıcılar için birtakım avantajları bulunmaktadır. Bu avantajlar, projede WebRTC teknolojisinin kullanılmasına karar verilmesini sağlamıştır.

Avantajları:

- WebRTC eklenti gerektirmez.
- Açık kaynaklı bir projedir.
- Kullanılması için herhangi bir ödeme yapılmasına gerek duyulmaz.
- Web/mobil RTC uygulamaları geliştirmek için geliştiricide telekomünikasyon geçmişi aramaz.
- Uygulamaya multimedya eklemek için medya sunucusuna ihtiyaç duymaz.

BÖLÜM: III

YÖNTEM

YÖNTEM

Bu bölümde projede kullanılacak olan yöntemlerin içeriğinden söz edilmiştir. Bu projede amaca uygun olarak WebRTC teknolojisi kullanılması hedeflenmiştir. Hedeflenen diğer teknolojilerden bazıları ise şunlardır:

- HTML'in CSS ile beraber geliştirilmesi adına JQuery kütüphanesinin kullanılması
- Uygulamanın istemci ve sunucu tarafında JavaScript programlama dili ile geliştirilmesi
- Sunucunun NodeJS kütüphanesiyle oluşturulması ve NodeJS kütüphanesinin yardımıyla, veritabanı için seçilen MongoDB ile bağlantı kurulumunun sağlanması



Şekil 17: Sanal Sınıf Aracı Oluşturmak

Şekil 17 ise NodeJs, WebRTC ve Socket.IO kullanılarak videolu görüşme uygulaması yapılmasını simgelemektedir.

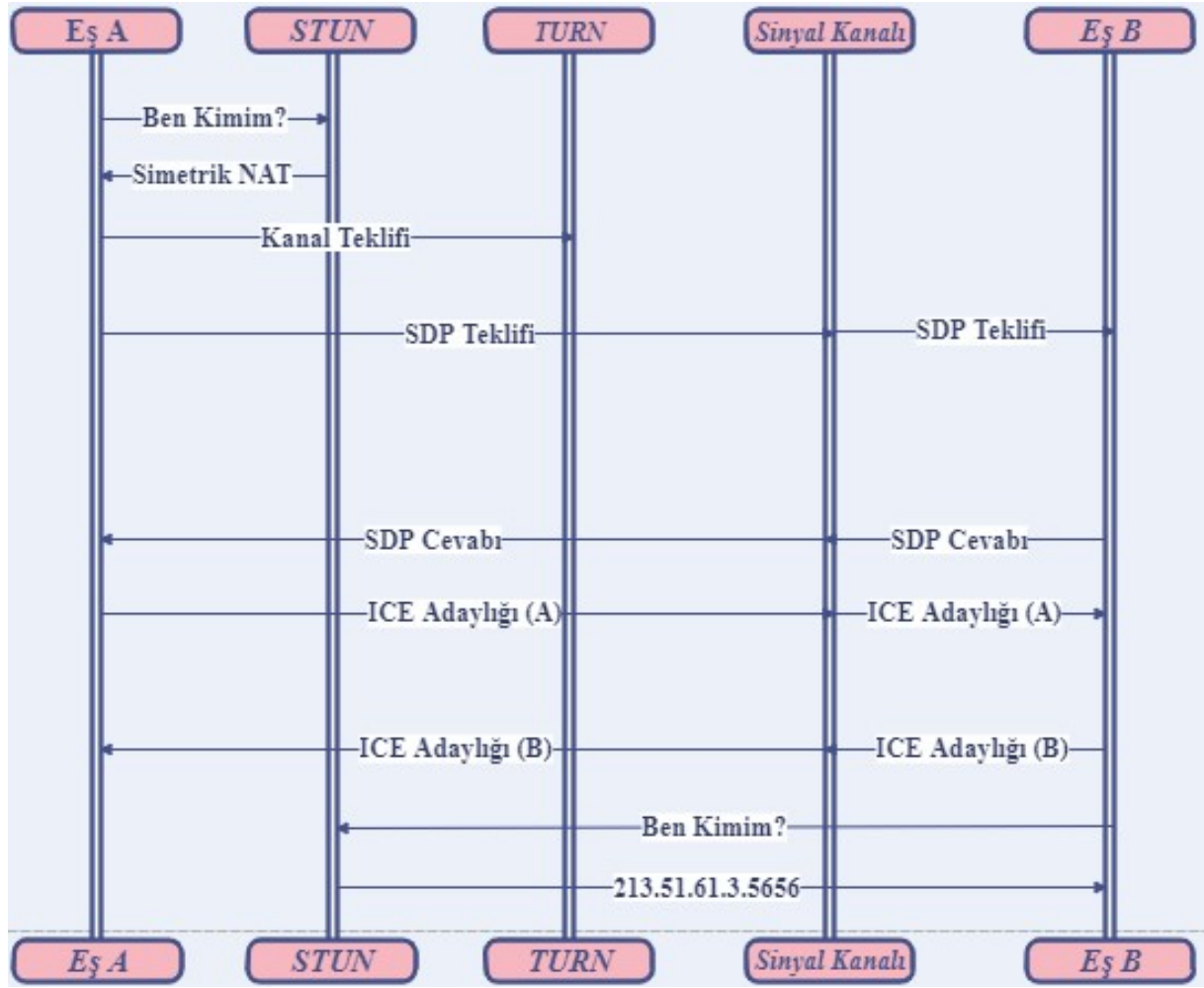
6.1 WebRTC

WebRTC, World Wide Web Consortium (W3C) tarafından oluşturulmuş olan ve gerçek zamanlı video, ses ve veri paylaşım uygulamalarının geliştirilmesine ortam oluşturan bir API'dır. WebRTC teknolojisi, oluşturulan uygulamada kullanıcıların herhangi bir eklenti veya üçüncü taraf yazılımı kullanmadan gerçek zamanlı iletişim kurmalarının mümkün olmasını sağlar. WebRTC tamamen ücretsiz olmakla beraber açık kaynaklıdır [18, 19].

6.1.1 WebRTC'de Ağ İletişimi

WebRTC için sunucu gereksinimi yoktur fakat internette bağlanan cihazların firewall'lar ardında yer alması ve yönlendiricilerin direkt olarak bağlantıya izin vermemesi sebebiyle sunucuya ihtiyaç duyulur. Bu nedenle ICE (Interactive Connectivity Establishment - Etkileşimli Bağlantı Kurulumu) framework'ü kullanılıp bu framework tarafından STUN ve TURN sunucuları kullanılır. Bu teknikle eşler arası ağ iletişimi sağlanır yani iki cihaz arasında bağlantı kurulur. ICE'de ilk önce ağ adresiyle ağ iletişimi gerçekleştirilmeye çalışılır. Ağ adresi sağlanamazsa uygulama, internet ağında yer alan Şekil 18'de görüldüğü gibi STUN sunucusuna "Ben kimim?" mesajı gönderir. Sonra sunucu, cihazın port ve IP verisini döndürür. Eğer cihaz, yönlendiricinin NAT'ı (Network Address Translation – Ağ Adresi Çevrimi) arkasında yer alıyorsa, cihazın ulaşılabilir olmadığı hatasını geri döndürür. NAT arkasındaki cihazlar için STUN sunucusu kullanılarak harici bir ağ adresi alınır. Bazı yönlendiricilerde kısıtlamalar bulunabilir. Bu durumda STUN sunucusundan cihazın public IP adresi alınabilse dahi bağlantı kurulumu gerçekleşmeyebilir. Böylece bu yöntem başarısız olur. Bu durumun çözümü için TURN sunucuları kullanılmaktadır. TURN sunucusu ile bir bağlantı açılarak

trafik yönlendirilebilir. Bu şekilde WebRTC teknolojisiyle eşler arasında çoklu bağlantı gerçekleştirilir [20, 21].



Şekil 18: WebRTC ICE Diyagramı

6.2 Sinyalizasyon Mekanizmaları

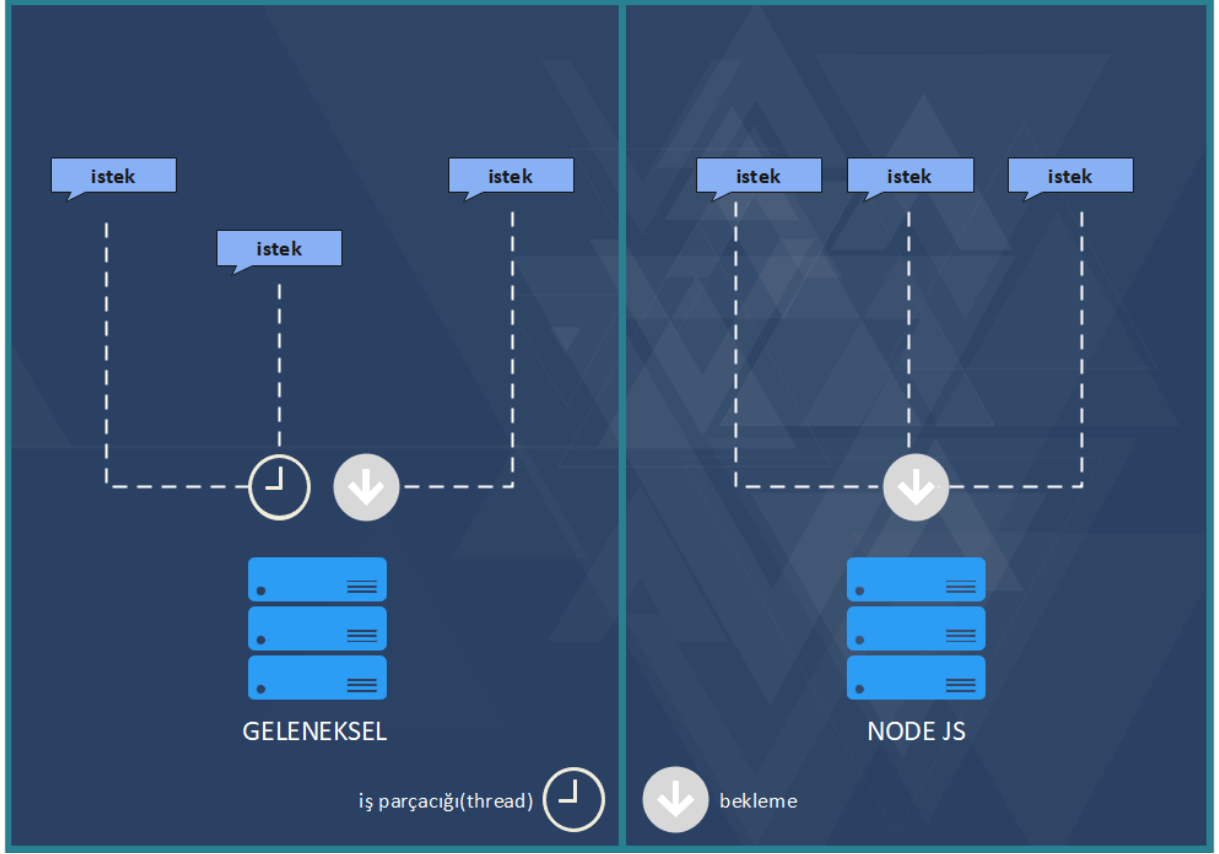
Uzak bir cihaz ile haberleşme yapabilmemiz için harici bir sinyalleşme servisine ihtiyacımız vardır. Burada sinyalizasyon sunucusu olarak NodeJS ve Socket.IO kullanılabilir. Katılımcıların diğer istemciler ile birbirlerini görebilmesi için ağ bağlantısı öncesinde bağlantıyla ilgili bilgiler web sunucusu üzerinden değiş tokuş edilmelidir. Bu iletişimi gerçekleştirecek olan

protokol SDP'dir(Session Description Protocol-Oturum Tanımlama Protokolü). Yani SDP kurulan bağlantıda gönderilip/alınan multimedia içeriğinde hangi özelliklerin bulunduğu tanımlanmasında kullanılır. Bu içerik bilgisinde bulunanlar ise çözünürlük değeri, medya formatı, kullanılan codec'ler ve bilgi güvenliği için gerekli şifreleme bilgisidir. Bu sayede eşler arasında ki veri iletişiminde bir anlaşma sağlanmış olur. Bağlantı bilgilerinin değiş tokuş edilmesinden sonra sunucu, web tarayıcı bağlantılarında şeffaflaşır. Tarayıcı, kimlik doğrulama anahtarlarının denetimini gerçekleştirip güvenli bir ortam oluşturur [22].

6.3 NodeJS

NodeJs, ağ bağlantılı uygulamalar için geliştirilmiş olan bir çalışma ortamıdır. JavaScript tabanlıdır, açık kaynaklıdır ve eş zamanlı uygulamalar oluşturmak için bize pek çok avantaj sağlar. NodeJs kütüphanesinde barındırılan bütün API'ler eşzamansızdır dolayısıyla hiçbir işlev doğrudan gerçekleşmez. Bu yüzden işlem hiç engellenmemekle beraber yüksek veriyi doğrudan aktarabilir. Böylece Şekil 19'da görüldüğü gibi bize düşük gecikme ve yüksek verimlilik sağlar. Telekonferans ve sanal sınıf gibi gerçek zamanlı iletişim uygulamalarında eş zamanlı işlemler için NodeJs kullanılarak bir sunucu geliştirilebilir [23].

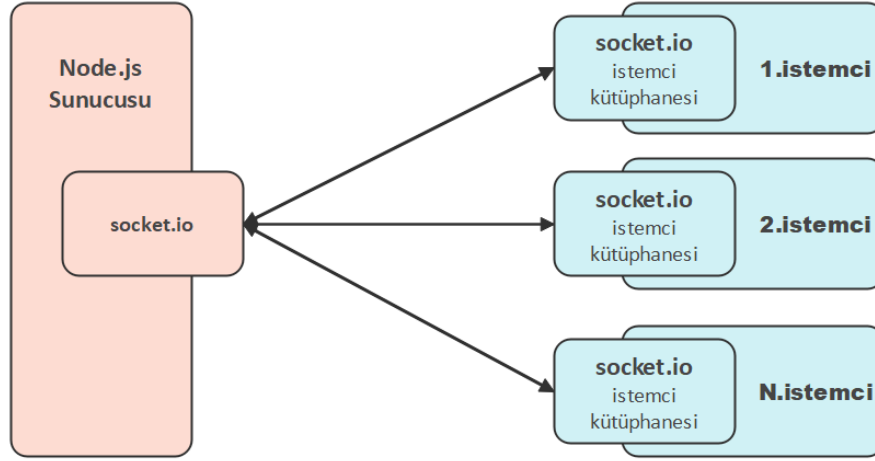
WebRTC ile eş zamanlı iletişim uygulaması oluşturmak için birden fazla programlama dili mevcut olmakla beraber yalnızca JavaScript gibi diller, kullanıcıların beklentisi olan görüntü ve sesli iletişim uygulamasının tamamını sunma becerisine sahiptir. Bu yüzden sanal sınıf uygulamasında NodeJs kullanılması hedeflenmiştir [24].



Şekil 19: NodeJs'in Düşük Gecikme ile Çalışması

6.3.1 Socket.IO

Socket.IO, eş zamanlı iletişim ve karşılıklı veri alışverişine olanak sağlayan uygulamaları geliştirmek için kullanılan bir NodeJs modülüdür. Socket.IO veri alışverişini yapabilmek için WebSocket, AJAX gibi çeşitli yöntemleri kullanır. Socket.IO, bütün platformlarda çalışabilen güvenilir, hız performansı yüksek, eş zamanlı ve çift yönlü iletişimi sağlayan bir yapıdır. Şekil 20 Socket.IO'nun çalışma şekline bir örnektir. Bilinen en güçlü iletişim modüllerinden biridir [25, 19].



Şekil 20: Socket.IO

6.4 PdfJs

PdfJs, HTML5 tabanlı bir JavaScript kütüphanesi olup, PDF (Portable Document Format) şeklindeki dosyaların web sayfalarında gösterilmesi amacıyla oluşturulmuştur. Açık kaynaklıdır. Sanal sınıf uygulamalarında eğiticilerin sunmak istediği dokümanı ekranlarında sunabilmeleri için PdfJS kütüphanesinin kullanılması uygun tercihler arasındadır. Ayrıca eğitici tarafında yapılan değişikliklerin eş zamanlı olarak katılımcılara yansması için scriptler kodlanarak etkileşimin artırılması hedeflenmiştir.

6.5 MongoDB

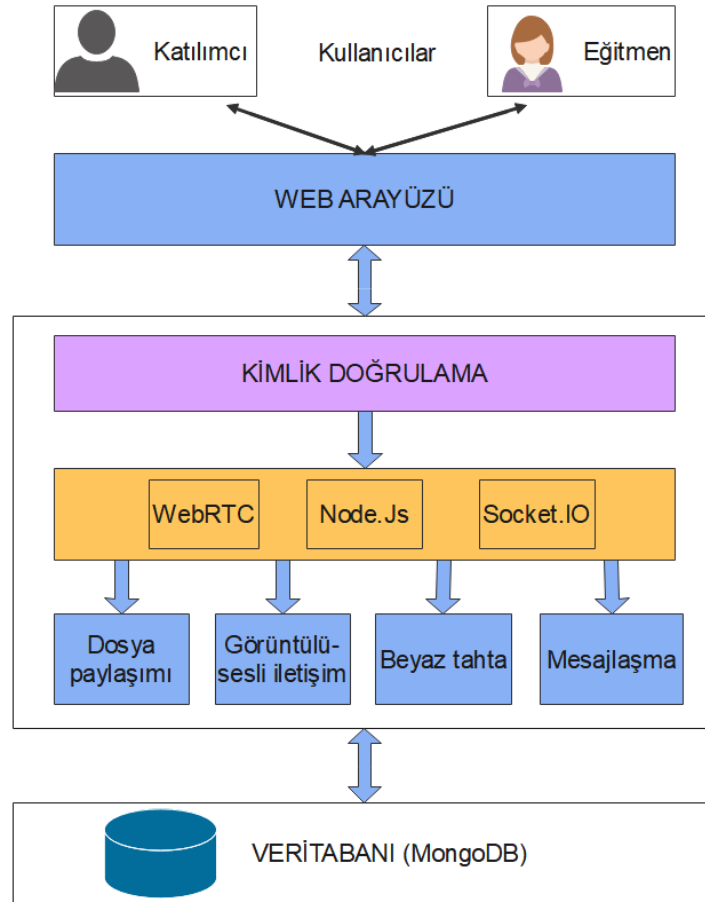
MongoDb, C++ ile geliştirilmiş açık kaynaklı ve ölçeklenebilir bir NoSQL(Not Only SQL) veri tabanı uygulamasıdır. MongoDB, verilerini JSON (JavaScript Object Notation) tipindeki dokümanlarda saklamakta olup NodeJs ile kullanım kolaylığı sağlamaktadır. Eş zamanlı iletişim gerektiren sohbet, telekonferans ve sanal sınıf gibi uygulamalarda veritabanı

işlemlerinin hızlı olması gerekir ve NoSQL veritabanları bu gereksinimler dolayısıyla geliştirilmiştir. NoSQL veritabanları, geleneksel ilişkisel veritabanları gibi kurallara sahip değildir ve daha hızlıdır. Dolayısıyla kullanım rahatlığı ve esnek yapısı sebebiyle sanal sınıf uygulamasında MongoDB veritabanı kullanımı tercih edilebilir. Şekil 21’de MongoDB’nin logosu gösterilmektedir [26, 27].



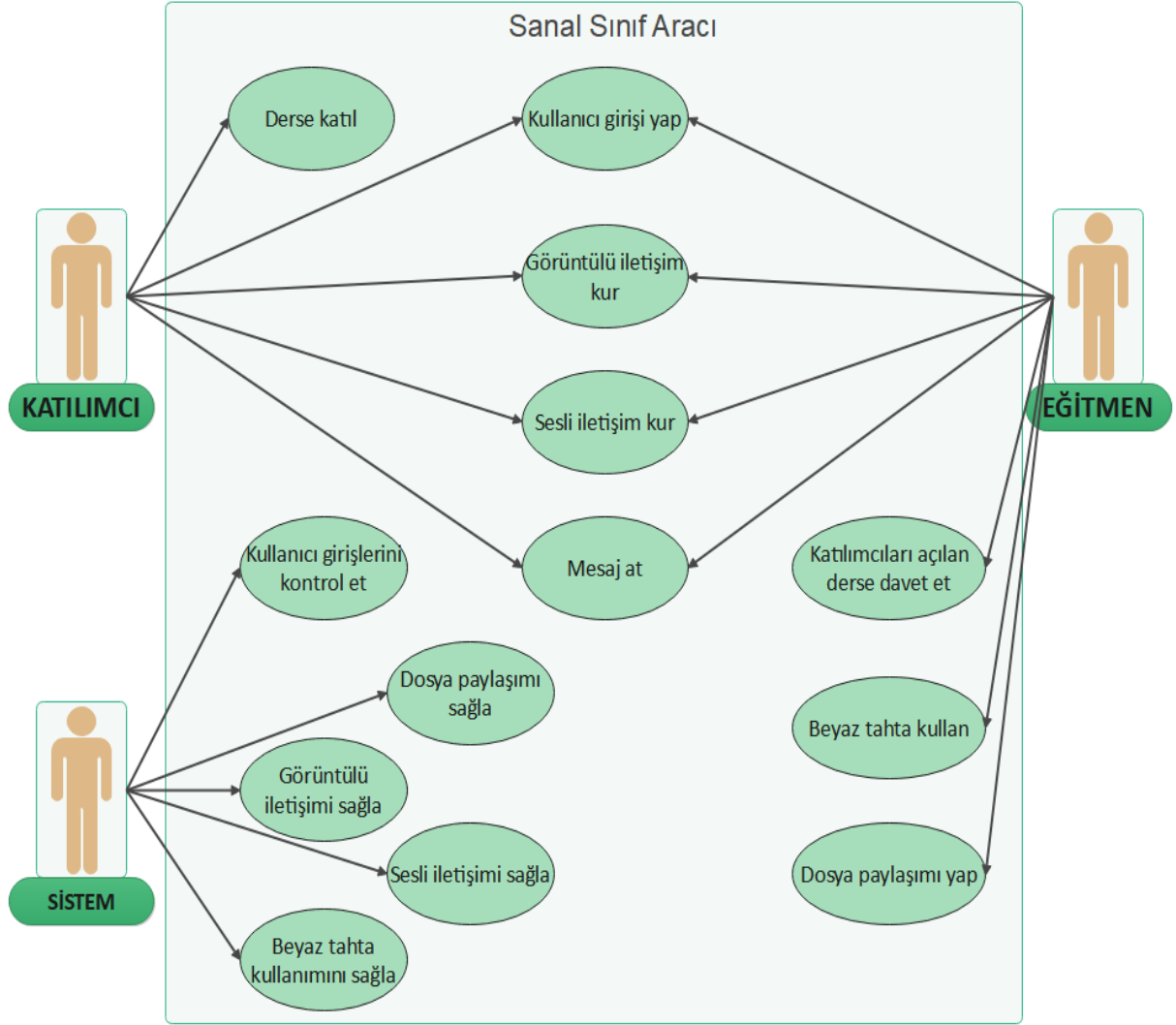
Şekil 21: MongoDB’nin Logosu[5]

Belirlenen bu teknolojilerle gerçekleştirilecek olan sistemin genel mimarisi Şekil 22’de belirtilmiştir.



Şekil 22: Genel Mimari

Gerçekleştirilecek sanal sınıf ortamında bulunacak olan kullanıcılar, Şekil 23'teki use-case (kullanım senaryosu) diyagramında belirtilmiştir. Burada sistemin de sanal sınıfın bir parçası olduğu düşünüldüğünde, sistemin görevleri de diyagramda verilmiştir.



Şekil 23: Use-case Diyagramı

Şekil 24'te görülen activity (faaliyet) diyagramında öğretmen ve katılımcıların sistemdeki etkinlikleri verilmiştir. Öğretmen sanal sınıfa giriş yaparak ders oluşturacak ve katılımcıları derse davet edecektir. Katılımcılar daveti aldıktan sonra ders için istek gönderip derse katılabilmektedir. Katılımcılar ve öğretmenler, görüntülü-sesli iletişim kurabilir ve mesajlaşma

BÖLÜM: IV

SONUÇ

SONUÇ

Geliştirilecek olan projede seçilen WebRTC teknolojisi ile amaç, geliştiriciler için zorlayıcı ve maliyetli olmayan; kullanıcılar için ise maksimum fayda sağlayabilen bir sanal sınıf aracı oluşturmak ve birtakım koşullar gereği kullanılması gereken bu aracın yaşam kalitesini arttırması olmuştur.

Raporun ileri kısımlarında asıl problemin ne olduğundan, bu problemin nasıl çözüleceğinden ve çözümü için kullanılabilecek olan teknolojilerden bahsedilmiştir. Bununla birlikte dağıtık bir yazılım olarak maksimum fayda sağlanması için web platformu tercih edilmiş; ücretsiz ve açık kaynaklı olması, düşük maliyetli olması, son kullanıcılar için uygun olması ve eklenti gerektirmemesi özellikleriyle diğer teknolojilere göre daha avantajlı olan WebRTC teknolojisinin kullanılmasına karar verilmiştir. WebRTC teknolojisinin her yönüyle ele alındığı bu raporda, paydaşların kafasında soru işareti bırakmamak adına doğal dil kullanılmış; diyagramlarla da oluşturulacak sistem ve sistem mimarisi basite indirgemıştır.

KAYNAKÇA

- [1] E. Fatih, “Webrtc nedir?” <https://fatihherikci.com/webrtc-nedir/>, Ekim 2019.
- [2] A. Yeren, “Webrtc nedir?” <https://www.mediatick.com.tr/tr/blog/webrtc-nedir>.
- [3] A. Furkan, “Signalr nedir?” <https://medium.com/@furkanalaybeg/signalr-nedir-617f5873711d>, Haziran 2019.
- [4] K. Yunus, “Signalr nedir? nasıl Çalışır?” <https://medium.com/kodcular/signalr-core-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-31b395aec335>, Ekim 2019.
- [5] M. Vuranok, “Mongodb,” <https://medium.com/batech/mongodb-3ccc9449731e>, Temmuz 2019.
- [6] K. Vinodh, K. Punith, V. Sunil, and A. Srinivas, “Sanal sınıf kullanımının uygulanması,” http://www.ksct.iisc.ernet.in/spp/37_series/spp37s/synopsis_exhibition/114_37S1022.pdf, project reference no:37S1022.
- [7] J. Onihunwa, E. Irunokhai, E. Omole, and D. Joshua, “Web tabanlı sanal sınıf sisteminin geliştirilmesi ve uygulanması,” <https://seahipaj.org/>.
- [8] E. Erdem and A. Halis, “Java ve web tabanlı uzaktan eğitim,” https://www.researchgate.net/publication/267790610_Java_ve_WEB_Tabanli_Uzaktan_Egitim_e-Egitim_icin_Sanal_Sinif_ve_Sanal_Laboratuvar_Projesi.

- [9] B. A. Jansen, “Webrtc tabanlı video konferansının performans analizi,” <https://repository.tudelft.nl/islandora/object/uuid%3A505300b0-f421-4470-b63e-cdb9154f7d54>, Ağustos 2016, thesis No: PVM 2016-086.
- [10] H. Simon and L. Alexander, “Webrtc uygulamalarının dizaynı ve mimarisi,” <http://muep.mau.se/handle/2043/29191>, 2019.
- [11] P. Guy, B. Thomas, E. Marcus, and J. Yawna, “The advantages and disadvantages of the virtual classroom and the role of the teacher,” http://www.swdsi.org/swdsi2010/sw2010_preceedings/papers/pa126.pdf, pp. 2–9, Mart 2010.
- [12] A. Büşra Deniz, “Webrtc nedir?” <https://medium.com/@busradeniz/webrtc-nedir-8a483e686162>, Temmuz 2016.
- [13] G. Hakan, “Webrtc avantajları ve teknik zorlukları?” <https://hakanguzel.medium.com/webrtc-avantajlar%C4%B1-ve-teknik-zorluklar%C4%B1-hakk%C4%B1nda-bilmeniz-gereken-her-%C5%9Fey-5925465d619f>, Kasım 2020.
- [14] H. Guzel, “Webrtc’yi anlamak,” <https://hakanguzel.medium.com/webrtcyi-anlamak-d6c0aaedc3d>, Kasım 2020.
- [15] H. Can, “Webrtc tabanlı uygulamaları Ölçeklendirme,” https://www.google.com.tr/url?sa=i&url=https%3A%2F%2Fdzone.com%2Farticles%2Fscaling-webrtc-based-applications%3Fsource%3Dpost_page-----&psig=AOvVaw2OgyS65UJ6CC9nJYo3FTi0&ust=1606463795739000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPCD_pPen-0CFQAAAAAdAAAAABAD.

- [16] A. Ramazan, “Webrtc hakkında,” <http://ramazanaltinata.blogspot.com/2015/11/webrtc.html>, Kasım 2015.
- [17] S. Zhidi, “Azure signalr nedir?” <https://devblogs.microsoft.com/aspnet/azure-signalr-service-now-supports-asp-net/>, Nisan 2019.
- [18] U. Yıldız, “Web tabanlı gerçek zamanlı iletişim teknolojisi ile sanal sınıf uygulaması,” *Research in Education and Teaching*, vol. 5, pp. 295–302, Haziran 2016.
- [19] R. Infotech, “Building a video chat using node.js, webrtc and socket.io,” <https://rydotinfotech.medium.com/building-a-video-chat-using-node-js-webrtc-and-socket-io-52b930346e6c>, Eylül 2020.
- [20] M. D. Network, “Webrtc connectivity,” https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Connectivity, 2020.
- [21] Z. Aydın, “React native’de görüntü ve ses aktarımı için webrtc kullanımı,” <https://zaferayan.medium.com/react-nativede-g%C3%B6r%C3%BCnt%C3%BC-ve-ses-aktar%C4%B1m%C4%B1-i%C3%A7in-webrtc-kullan%C4%B1m%C4%B1-9e4db14de25>, Nisan 2020.
- [22] S. Dutton, “Build the backend services needed for a webrtc app,” <https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/>, Kasım 2020.
- [23] B. Kazoğlu, “Node.js nedir? sağladığı avantajlar ve Örnek uygulama,” <https://www.mshowto.org/node-js-nedir-sagladigi-avantajlar-ve-ornek-uygulama.html>.
- [24] D. Gömleksizozlu, “Nodejs nedir?” <https://medium.com/@duygugmleksizolu/node-js-nedir-ef106bab9c83>, Ekim 2016.

- [25] Y. Sezer, “Nodejs ve socket.io modülü,” <https://www.yusufsezer.com.tr/node-js-socket-io/>.
- [26] Vikipedi, “Mongodb,” <https://tr.wikipedia.org/wiki/MongoDB>.
- [27] Y. Sezer, “Mongodb nedir?” <https://www.yusufsezer.com.tr/mongodb-nedir/>.

E K L E R

Proje yönetiminde bir projenin planlamasının gösterilmesi beklenir. Gantt Diyagramı, tüm faaliyetlerin listesini göstermesi, her aktivitenin başlangıç ve bitiş tarihlerini göstermesi, her aktivitenin zamanlanmış bir süresi olması özellikleriyle avantajlıdır ve projedeki belirli görevlerin koordinasyonunda, planlanmasında ve izlenmesinde yardımcı olması için tercih edilmiştir. Bu projenin planlanması ise Şekil25'te verilmiştir. Grafiğin sol tarafında faaliyetlerin bir listesi vardır. En üst kısımda ise uygun bir zaman çizelgesi yer bulunur. Her çubuk yapılan bir etkinliği temsil etmektedir. Çubuğun bulunduğu yer ve belirttiği tarih, yapılan etkinliğin başlangıç ve bitiş tarihlerini yansıtmaktadır.

Proje Planlayıcısı

Şağı tarafın vurulacak bir dönem seçin. Grafikte gösteren gösterge, bu hâzrenin yonında verilmektedir.



Sekil 25: Gantt Diyagramı