



Mühendislik Fakültesi
Bilgisayar Mühendisliği
Bölümü

Bitirme Projeleri Kitapçığı 2019-2020 Akademik
Yılı

Haziran 2020

Önsöz

Yaşar Üniversitesi Bilgisayar Mühendisliği Bölümü lisans programı son sınıf öğrencileri, bölüm özgeçmişinin gerçekleştirilmesi doğrultusunda, dört yıllık eğitimleri boyunca edindikleri bilgi ve becerileri gerçek yaşamda karşılaştıkları mühendislik problemleri ve bunların çözümlerine uygulamak amaçlarıyla Bitirme Projeleri yapmaktadırlar.

Bitirme projeleri, her akademik yılın Güz Yarıyılı'nda zorunlu ders olarak alınan COMP 4910 Mezuniyet Tasarım Projesi I, AKTS: 8 (Senior Design Project I, ECTS: 8) ve Bahar Yarıyılı'nda alınan zorunlu ders COMP 4920 Mezuniyet Tasarım Projesi II, AKTS: 8 (Senior Design Project I, ECTS: 8) ile devam eden iki tane dördüncü sınıf dersi çerçevesinde yürütülmektedir. Söz konusu iki derste de her hafta bir saatlik ortak bir açıklamalar ve sorular/cevaplar toplantısı yapılmaktadır. 14 haftalık bir yarıyıldan her öğrenciden beklenen proje katkısının 98 adam-saat olduğu düşünülürse, dört kişilik bir proje takımının bir akademik yılda 784 adam-saat'lik (98 adam-günlük) oldukça kapsamlı bir proje yapabileceği ortaya çıkmaktadır.

Güz Yarıyılı başında 3-4 öğrenciden oluşan proje takımları oluşturulmakta, her proje takımına ayrı bir proje tahsis edilmektedir. Tahsis edilen projelerin Disiplinler arası projeler yapılabilmesi için, proje takımlarının oluşturulmasında Elektrik-Elektronik ve Yazılım mühendisliği gibi farklı mühendislik bölümlerinden bitirme projesi yapan öğrencilerin aynı proje takımında yer alması tercih edilmektedir. Ayrıca, proje konularının sanayi kuruluşlarının gerçek problemleri ile örtüşmesi de arzu edilen bir durumdur, fakat sanayi işbirlikli bitirme projeleri her zaman mümkün olmamaktadır. Sanayi işbirliği ile yürütülen bitirme projelerinde, bölüm öğretim üyesi olan Proje Danışmanı'na ek olarak ilgili sanayi kuruluşundan bir uzman Eş Proje Danışmanı olarak yer almaktadır.

Proje takımları, Güz Yarıyılı'nda, proje probleminin detaylı bir şekilde saptanarak yazılması (RSD: Requirements Specification Document) ve çözüm için ilk taslak tasarımın üretilmesi (DSD: Design Specification Document) aşamalarını tamamlamaktadır. Bahar Yarıyılı'nda ise çözüm için tasarımı son haline getirmekte, gerekirse RSD belgesini revize etmekte, çözümü donanım ve yazılım ortamında gerçekleştirme ve test etme aşamalarını tamamlamakta ve bir Ürün Kılavuzu (PM: Product Manual) üretmektedirler. Proje takımları ayrıca her yarıyıl sonunda kapsamlı bir Sonuç Raporu'nu proje danışmanlarına teslim etmektedirler. Proje saydam sunumu, proje posterleri, proje web sitesi gibi proje çıktıları da Bahar Yarıyılı sonunda proje danışmanlarına teslim edilmekte, ayrıca her proje takımı, en az üç kişilik bir jürinin katıldığı bir herkese açık bir toplantıda, ürettikleri sistemin nasıl çalıştığını gösteren bir tanıtımı (product demonstration) yapmaktadırlar. Mayıs 2020'de yapılan ürün tanıtımları salgın hastalık yasakları sebebiyle Sakai video konferans altyapısı kullanılarak sanal ortamda yapılmıştır.

Başarılı bitirme projelerinin takımlarına, uzun bir süredir her yıl Türkiye Bilişim Derneği, İzmir Şubesi tarafından düzenlenen GBYF: Genç Beyinler Yeni Fikirler bitirme projeleri yarışmasına katılmaları önerilmektedir.

2019-2020 öğretim yılında öğrencilerimizin gerçekleştirdikleri bitirme projelerini özetleyen bu kitapçığın ortaya çıkartılmasına destek veren tüm bölüm öğretim elemanlarımıza teşekkür ederim.

Prof. Dr. Mehmet Ufuk Çağlayan
Bilgisayar Mühendisliği Bölüm Başkanı
Yaşar Üniversitesi

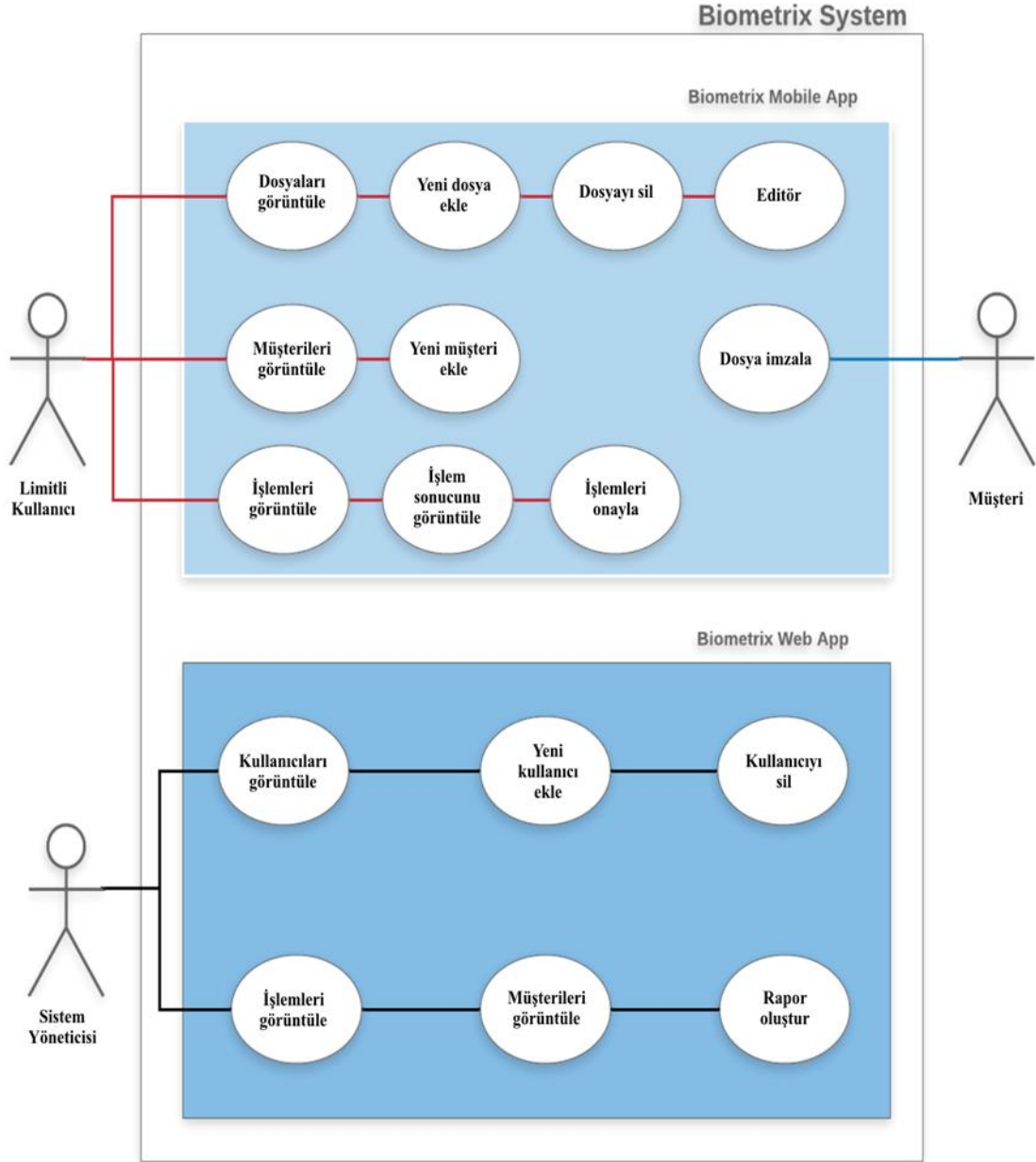
İçindekiler

Biometrix: Artificial Intelligence Assisted Biometric Signature on Blockchain	4
AUTOBOT: Autonomous Robot Car.....	10
OSP: Offer Based Online Service Platform	14
FIWE - Factoring Integers with ECM.....	18
DJAVAC: Automated Load Distribution Engine for Java.....	20
SmartRF	22
SDS: Smart Door System.....	28
ProcessX: Improved Search Functionality with Image Processing for E-Commerce Systems.....	31
EKG-a: A wireless Arduino ECG System.....	36
SLATE: Yapay Zeka ile İşaret Dili-Yazı Çevirisi.....	40
i-Vehicle: Motorless Vehicle Rental System	43
REM-LIFE	46
POF: Performance Optimized Fluids	52
Tears Online : Blockchain Integrated Multiplayer Online Game.....	56
UOA: Universe of Anatomy.....	58
EVOBOX: Akıllı İlaç Kutusu	65

Proje Kodu ve Adı:	Biometrix: Artificial Intelligence Assisted Biometric Signature on Blockchain
Proje Takımı:	Taylan Akbas, Ahmet Şen, Doruk Mestanoğlu
Proje Danışmanları:	Doç Dr. Ahmet Hasan Koltuksuz
Proje Çıktıları	<ol style="list-style-type: none">1. Final Report2. Requirements Specifications Document3. Design Specifications Document4. Product Manual5. Biometrix Mobile App6. Biometrix Data-API7. Biometrix Admin-Web8. Product Web Site9. Poster10. Presentation11. Product Demo Video
Proje Web Adresi:	https://biometrix.pythonanywhere.com/
Proje Özeti 1. Giriş Kimlik doğrulama, veri boyutu ve kullanıcı sayısı arttıkça giderek daha önemli hale gelmektedir. Doğrulama birçok şekilde yapılabilir ancak yöntemler arasında imzanın tarihi oldukça eskiye dayanmaktadır. İmza, hemen her toplumda iki veya daha fazla taraf arasında yapılan anlaşmalarda tarafların kimliğini simgeler. Atılan imza taklit edilemez ve karmaşık kaligrafide olmalıdır. Günümüzde atılan ıslak imzalar, adli grafoloji uzmanlarınca belli kriterler esas alınarak başka imzalarla karşılaştırılabilir. Ancak teknolojinin günlük hayatta kullanılan uygulamalar ve cihazlardaki gelişimi ile kişinin biyometrik imzasını sayısal ortama aktarması mümkün olmuştur. Bu değişim, imza karşılaştırma ve doğrulama yöntemlerini kökten değiştirmiştir. İmzacının stylus (dijital kalem) kullanarak kendi el yazısı ile tablete, cep telefonuna veya bilgisayara attığı imza, normal bir kalem ile attığı ıslak imza gibi kişinin biyometrik verisidir. Bu biyometrik veri genellikle kimlik doğrulama işlemleri için kullanılır. Retina, parmak izi gibi çeşitli doğrulama sistemleri olsa da biyometrik imzanın kullanımı birçok açıdan diğer sistemlerden öndedir. Biyometrik sistemlerde her zaman kullanıcıların hassas verilerinin çalınma endişesi vardır. Bir kez böyle bir zarara uğranırsa geri dönüşü yoktur. Parmak izi veya retina bilgisi değiştirilemez verilerdir. Fakat aynı senaryoda imzacı her zaman kendi imzasını değiştirebilir. İmza verisi son derece dinamik bir yapıya sahiptir. Atılan her imza, yüzlerce ayrı noktadan oluşur. Her bir nokta, koordinat bilgisinden noktanın bağlı hızı bilgisine kadar birçok farklı değer barındırır. Tüm bu değerler kişinin imzasını oluşturur. Bu kompleks veri yapısı, biyometrik imzaları kolay değiştirilir ancak zor taklit edilir kılar. Amacımız, Biometrix sistemi ile kullanıcılara istedikleri bir dosyaya kendi biyometrik imza atmalarını, imzanın doğrulanmasını ve her bir imza işleminin yönetimi ve bakımını kolaylıkla gerçekleştirebilecekleri bir ortamı sağlamaktır. Sistemin en çok önem vereceği konu, saklanacak bilginin biyometrik veri olması sebebiyle güvenlidir. Bu şartlarda güvenliği sağlarken uluslararası standartlara uygun ve son teknoloji yöntemleri kullanmak kritik öneme sahiptir. 2. Gereksinimler Sistemin içinde yönetici, limitli kullanıcı ve müşteri olmak üzere üç ana rol bulunmaktadır. Bu roller, Web ve Mobil olmak üzere iki ayrı uygulamada toplanmıştır. Sistem yönetimi, işlem görüntüleme ve limitli kullanıcıların yönetimi için yöneticilere özel bir Web uygulaması tasarlanmıştır. Mobil uygulama ise kalan iki aktörün ihtiyaçları doğrultusunda tasarlanmıştır. Burada limitli kullanıcıların görevi, imzalanacak dosyaların uygulama editörü vasıtasıyla hazırlanması/düzenlenmesi, yeni müşterileri sisteme kaydetme ve müşterilerin dosyalara imza atmalarını sağlamaktır. Her imzadan sonra yapılan işlemin güvenilirliği ilgili limitli kullanıcıya bir bilgilendirme mesajı döner. Limitli kullanıcı, gelecek rapora göre imzayı onaylayabilir ya da tekrarlanmasını isteyebilir.	

Sistemin çalışması için hem donanımsal hem yazılımsal ihtiyaçlar bulunmaktadır. Biometrix'i kullanacak her sistemin müşterilerinin imzalarını toplayacakları tablet bilgisayara ve dijital kaleme ihtiyacı vardır. Bunun

dışındaki Web uygulamasına tarayıcı ya da herhangi bir mobil cihazın tarayıcısı üstünden erişilebilir. Aşağıdaki tablo Biometrix sisteminin rol ve ihtiyaç dağılımını göstermektedir.



Senaryo	Gereksinim
Yeni müşteri ekle	Yeni bir müşteri ekleyebilmek için, limitli kullanıcının müşterinin isim, soyismi bilgisi, eşsiz bir kimlik numarası (T.C. kimlik numarası vb.) ve 5 adet imzasına ihtiyacı vardır.
Dosya imzala	Bir dosyanın imzalanabilmesi için ilgili müşterinin sisteme kaydı yapılmış olmalıdır. Limitli kullanıcı dosyayı seçtikten sonra müşterinin kendine ait eşsiz kimlik numarasını girer. Müşteri açılan ekranı imzalar ve imzanın sonucu geri döner. Limitli kullanıcı sonuca göre işlemi onaylar veya tekrarlar.
Dosyaları görüntüle	Limitli kullanıcılar daha önce eklenen dosyaları kaydedilen dosya ismiyle bir listeden görüntüleyebilir. Seçilen dosya ve içeriği otomatik olarak editörün içine yerleşir.
Yeni dosya ekle	Limitli kullanıcılar dosyanın içeriğini uygulama editörü yardımıyla hazırladıktan sonra, dosyaya bir isim verme şartıyla sisteme kaydedebilirler.
Dosyayı sil	Limitli kullanıcılar daha önce eklenen dosyaları, önce dosyalar listesinden seçmek şartıyla silebilirler.
Müşterileri görüntüle	Sistemdeki müşterilerin isim, soyismi ve kimlik numaraları limitli kullanıcılar tarafından görüntülenebilir ancak müşterilerin imza bilgilerine erişimleri yoktur.
İşlemleri görüntüle	Sistemde müşterinin imzaladığı her dosya, limitli kullanıcı tarafından onaylanma şartıyla, bir işlem sayılır. İşlemler; imzacı, dosya ve yapılan işlem hakkında detaylı bilgi içerir. Sistemin içindeki tüm işlemler yönetici tarafından görüntülenebilir.
Limitli kullanıcıları görüntüle	Limitli kullanıcılar yönetici paneliden görüntülenebilir.
Yeni limitli kullanıcı ekle	Yeni limitli kullanıcılar; isim, soyismi, mail adresi, şifre ve telefon numarası girilmek şartıyla, sisteme yönetici panelinden eklenebilir.
Limitli kullanıcıyı sil	Limitli kullanıcılar sistem yöneticisi tarafından silinebilir. Bu işlem, limitli kullanıcının uygulamaya erişimini kapatır.
Rapor oluştur	Yöneticiler, sistemdeki işlemlerle ilgili adli vakalarda kullanılmak üzere detaylı raporlar oluşturabilir. Bu raporlar belge içeriği ve imza detaylarının yanında imza tanıma için çizilen grafikleri de içerir.

3. Dizayn ve Implementasyon

Biometrix sistemi 3 ana parçadan oluşur. *Biometrix Client-Mobil*, *Biometrix Admin-Web* ve *Biometrix Data-API*. Her bir parça, projenin gereksinimlerini karşılayacak üzere tasarlanmıştır ve bu her parçanın tasarımı ayrı bakış açıları karşılamak için aşamalara ayrılmıştır.

Sistem servisleri ve kullanıcılar kontekst bakış açısında,
Sistemin soyutlanması ve tekrar kullanılabilir parçaların ayarlanması mantıksal bakış açısında,
Sistemin parçalarını amaçlarına göre ayrılması bağımlılık bakış açısında,
Dizayn modelleri model bakış açısında ve
Kullanıcı arayüz tasarımını arayüz bakış açısında incelenmiştir.

Biometrix Client-Mobil:

Uygulama, müşteri ve limitli kullanıcılara erişim ortamı sağlamak için tasarlanmıştır. İlgili rollerin gerekli aksiyonları alabilmeleri için uygulamada 4 ayrı ekran vardır: Giriş (Login) ekranı, Anasayfa (Dashboard) ekranı, Müşteri (Clients) ekranı ve Dökümanlar (Documents) ekranı.

Giriş: Limitli kullanıcıların uygulamaya giriş yapabilmeleri için önceden sisteme kaydedilmiş olmaları gerekir. Limitli kullanıcı bilgileri ve doğrulama bilgileri Firestore Database’de saklanır. Limitli kullanıcılar sistemdeki kayıtlı mail adreslerini ve şifrelerini girerek uygulamaya erişebilir. Hatalı mail veya şifrede hata mesajı görüntülenir.

Anasayfa: Bu ekran diğer sayfalara geçiş yapabilmek için bir karşılama ekranı olarak tasarlanmıştır. Aynı zamanda sisteme dair bazı bilgileri ve yapılan son işlemlerin görüntülenmesini sağlar.

Müşteriler: Limitli kullanıcılar bu ekranı kullanarak sisteme yeni müşteriler kaydedebilir. Müşteriden imzasını almadan önce isim, soyismi bilgisinin girilmesi ve müşteriye ait eşsiz bir kimlik numarası girilmelidir. İmzacı ekrana dijital kalem yardımıyla 5 (beş) kez imza atmalıdır. İmza atıldıktan sonra sistemden onay ya da red mesajı uygulama tarafında görüntülenir.

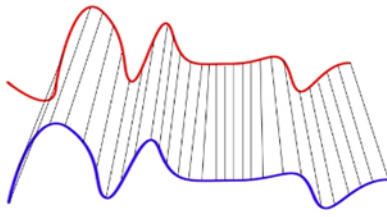
Dosyalar: Bu ekran, uygulamanın birçok aksiyonunu almak için tasarlanmıştır. Ekranın önemli bir kısmını editör doldurur. Dosyalar listesi editörün yanında yer alır ve seçilen dosyaların içeriği editörde görüntülenir. Limitli kullanıcılar dosya içeriğini düzenleyebilir, kaydedebilir ya da silebilir. Dosya seçildikten sonra, müşterinin kimlik numarası girilmelidir. Bu işlemler hazır olunca imzacı imzasını ekranda açılacak imza ekranına atabilir. İmza atıldıktan sonra sistem, imzanın fotoğrafını, doğrulama skorunu, eşik değerini ve fark değerini ekrana döndürür. Limitli kullanıcı gelen sonuca göre işlemi onaylar ya da tekrarlanmasını ister. İmzayla ilgili yapılan tüm işlemler ve sistemin çalışma mantığı bir sonraki bölümde açıklanmıştır.

Biometrix Data-API:

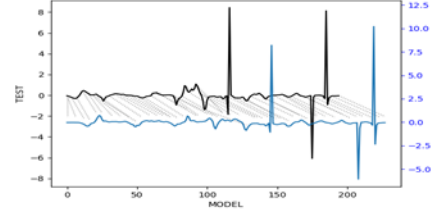
Biometrix sisteminin en önemli parçasını oluşturan bu RESTful API, imza modeli oluşturma, imza tanıma, blockchain işlemlerini yapmaktan sorumludur. Toplamda 8 ayrı isteğe cevap verir ve bu istekler *Client-Mobil* ve *Admin-Web* tarafından yapılır. Bu bölümde API’nin fonksiyonelliği istekler yerine istenen aksiyonlara bölünerek anlatılmıştır.

Model Oluşturma:

Model oluşturma aşlında kendi içinde 4 aşamaya bölünmüştür. Bu aşamalardan ilki, imzanın sayısallaştırılması, mobil uygulama tarafından yapılır. İmzacı imzasını atarken uygulama, imzadaki her bir noktanın bilgilerini kaydeder. Her bir nokta; koordinat bilgisi, basınç, hız, ivme, eğim, *azimuth* açısı ve zaman damgasından oluşur. İmzacının modelini oluşturabilmesi için sistemde 5 farklı imzasının sayısallaştırılmış şekilde kayıtlı olması gerekir.



Dynamic Time Warping Matching



Her imza atıldıktan sonra uygulamadan sunucuya istek yapılır.

İmzalar sunucuya kaydedildikten sonra, ikinci olarak, imzaların her biri için öznelik çıkarımı (*feature extraction*) yapılır. Bu işlem sayesinde toplamda 20 lokal özellik imzadaki her bir nokta için çıkarılmış olur. Özelliklerin her biri aslında imzacının biyometrik imzasını oluşturur. Bu aşamada normalizasyon operasyonları da yapılarak, sayısal değerler birbirine yakın aralıklara getirilir.

Üçüncü aşama, imzaların kendi aralarında karşılaştırılmasıdır. Her bir imza, kendi dışındaki dört imzayla karşılaştırılır. Böylece kayıtlı 5 imzayı ikili eşleştirerek toplamda 10 kez karşılaştırma yapılır. Bu karşılaştırma, imzaların her bir noktası arasındaki mesafeyi hesaplayarak yaptırılır.

Her bir imzanın, yüksek olasılıkla farklı sayıda noktadan oluşması sebebiyle *Euclidian* uzaklık formülünü kullanmak yerine zaman serilerini karşılaştırmada kullanılan ve aslen ses modellerini karşılaştırmak için kullanılan *Dynamic Time Warping* algoritması kullanılmıştır. Algoritma, her bir imza çifti arasındaki benzerliği hesaplar ve bu değer her çift için kaydedilir ve böylece son aşamaya geçilir. Son aşamada, 5 imza için toplamda oluşacak bu 10 değer, aritmetik ortalaması imzacının eşik değerini belirler. Bu değer daha sonra imza doğrulamada kullanılmak üzere kaydedilir.

Modelin Blockchain’de Saklanması:

Sunucu, imzacının 5 imzasını, eşik değerini, kişisel bilgilerini içinde bulunduran özel bir blockchain yapısına ekler. Blok aynı zamanda kendinden önceki bloğun SHA-256 algoritmasını kullanarak hash değerini, bloğun oluşturulduğu zaman damgasını, index numarasını ve her blok için emek ispatı değerini (Proof of Work) kaydeder. Böylece imza sahiplerinin modellerinin değiştirilmesi ya da silinmesi durumunda sistem uyarı verir. Bu zincirde her bir blok müşteriyi temsil eder.

```

"data": {
  "client": "1000",
  "threshold": 2.0984791142026915,
  "signatures": {
    "s1": {},
    "s2": {},
    "s3": {},
    "s4": {},
    "s5": {}
  }
},
"prev_hash": "83d7411b8069a6f1ffa6ae6e6379f967a2d666eb65cc5cf4eeef318b058c5b55",
"timestamp": 1589326857.5779939,
"proof": 95,
"index": "2"

```

İmza Doğrulama:

Uygulamada imza doğrulamaya başlamak için müşterinin imza modelinin oluşturulmuş olması gerekir. Doğrulama süreci, model oluşturmayla neredeyse aynı aşamalara sahiptir. Önce imzacının imzası, uygulama tarafından sayısallaştırılır. Ardından öznelik çıkarımı yapılarak karşılaştırma için hazır hale getirilir. Bu sefer karşılaştırma, test imzası ve modeldeki 5 imza arasında yapılır. Her bir ölçüm sonucu daha sonra aritmetik ortalamaları alınması için saklanır. Bu ortalama değer yapılan testin skorunu verir. İmzanın doğrulanabilmesi için test skorunun müşterinin önceden kaydedilmiş eşik değerinden çıkarılması gerekir. Aradaki fark sıfıra ne kadar yakınsa, doğrulama o kadar başarılıdır. Sistem, farkı + 0.2'ye kadar tüm imzaları doğrulamak üzere ayarlanmıştır. 0.2'den büyük farkları sahte imza olarak tanımlar. Fark ne kadar yüksekse atılan imza o kadar başarısızdır. İmza doğrulama bittikten sonra doğrulama sonucu kullanıcıya geri döndürülür.

İşlem:

İmza doğrulamadan dönen sonuçlar şayet kullanıcı tarafından onaylanırsa, işlem (*transaction*) detayları sunucu tarafından kaydetmeye hazırlanılır. Bu aşamada işlemin detaylı rapor ve dosyaları kaydedilir. Rapor; işlem numarası, müşteri kimlik bilgileri, zaman damgası, müşterinin sayısal imzası, doğrulama sonucu, imza analizi, imza ve imzacının fotoğrafları ve imza atılan konumun koordinat bilgilerinden oluşur. Bu raporlar daha sonra sistem yöneticisi tarafından incelenebilir.

Biometrix Admin-Web:

Projenin son parçası, yönetici rolünde sistemi kontrol edebilme ihtiyacını karşılar. Sistem yöneticileri, web sitesini kullanarak kendi çalışma ortamlarındaki işlemleri takip edebilir yada yapılan bu işlemler hakkında detaylı raporlar oluşturabilir. Bunun dışında, limitli kullanıcıları görüntüleyebilir, silebilir veya yeni kullanıcıları sisteme ekleyebilir.

5. Sonuç

Sistemin imza tanıma algoritmasının performansını ölçmek için testler yapılmıştır. Testler için 5 farklı imzacının imza modelleri oluşturulmuştur. Ardından her bir imzacı için 10 adet gerçek imza, 10 adet profesyonel bir şekilde atılmış sahte imza ve 10 adet diğer kullanıcılardan rastgele alınmış imzalar toplanır. Toplanan imzalar sistemde test edilir. Test sonuçları, biyometrik sistemlerin etkinliğini değerlendirmede kullanılan bir oran olan Equal Error Rate (EER) hesaplaması sonucu bulunmuştur. Biometrix sisteminin EER sonucu: %7'dir.

Biometrix, ilerde özel şirketlerden devlet kurumlarına birçok alanda hizmet verebilmek üzere tasarlanmıştır. Sistemin asıl odak noktası kullanıcılara müşterilerinin biyometrik datasını elde edebilme, doğrulama ve bu bilgileri güvenle saklayabilmektir. Kurumlar bu sistemi kullanarak kendi çalışma ortamlarına kolayca entegre edilebilecek, dosya imzalama ya da onaylama işlemlerini kolaylıkla takip edebilecekler. Kurgulanan sistemin ilk prototipi olan Biometrix, potansiyel iş ortaklarına projenin anlatılabilmesi için dizayn edilmiştir



Proje Kodu ve Adı:	AUTOBOT: Autonomous Robot Car
Proje Takımı:	Aleyna Sarışın , Burak Küre, Dođukan Soysal
Proje Danışmanları:	Doç.Dr. Tuncay Ercan
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. Requirements Specifications Document 3. Design Specifications Document 4. Product Manual 5. Product-Hardware Subsystem 6. Product-Software Subsystem 7. Project Web Site 8. Project Powerpoint Presentation 9. Project Poster 10. 2D-3D Simulation Tool
Proje Web Adresi:	https://autobott.wixsite.com/autobotwebsite
Proje Özeti:	
<p>1. Giriş:</p> <p>İlk aşamada endüstriyel operasyonların çoğunun genellikle insanlar tarafından veya robotların manuel kullanımı ile sağlandığını gördük. Bu nedenle projenin konusunu seçerken ve işleyişini tasarlarken teknolojinin gelişimi ile beraber bir şekilde gelişmesinin takımımız için önemli olduğundan emindik. Bu doğrultuda gelişmeye ve yeniliklere her zaman uyum sağlayabilecek bir alt yapı tasarlamak ilk verilen kararlardan biriydi. Bir robot prototipi tasarlanmasının temel nedeni ise, günümüzde dünya çapında sanayileşmenin ve teknolojinin gelişmesiyle beraber insanların yapabildiği bir çok işi robotların daha az enerji ve zaman harcayarak yapabiliyor olmasıdır. İnsanların kontrol etmediği, otonom olarak hareket eden elektronik cihazların özellikle robotların, daha hızlı, verimli ve hatasız olduğu bilinmektedir. Bu proje kapsamında, manuel kullanımda oluşabilecek sorunların çözülmesi ve çözümlerin geliştirilmesi için yeni adımlar atmak temel amaçtır. Bu sorunları robota kazandırılan çeşitli fonksiyonellikler ile birlikte çözülmesi hedeflenmiştir. Geliştirdiğimiz robota eklenen en önemli özellik, çevre ile aktif iletişim halindeyken kapalı alanda bulunduğu ortamın haritasını çıkararak bulunduğu yeri algılayabilmesiydi. Bu hedefe ulaştıktan sonra bulunan ortamda engellerden kaçarken haritasını güncelleyerek istenilen noktaya gidişini aksatmaması robotu günümüzde sanayilerde ve kapalı alanlarda istenilen amaçla kullanıma daha uygun bir hale getirdi. Haritalama, lokalizasyon, yol planlama ve yol takip gibi işlevsellikler robota tamamen kazandırdıktan sonra kullanıcıların daha rahat ve hızlı bir şekilde robotun konumunu ve işleyişini takip edebilmesi için bir simülasyon ortamı oluşturulmuştur.</p> <p>2. Gereksinimler</p> <p>Gereksinimlerin özetini sistem çalışması ve etkin müşteri kullanımı için 3 ana bölüme ayırdık.</p> <p style="padding-left: 40px;">1. Müşteri Gereksinimlerinin Kullanımı</p> <ul style="list-style-type: none"> ● Kullanıcı, sistemin düzgün çalıştığını görmelidir. ● Aktif iletişim sağlandıktan sonra testlerin başarılı olduğunu görmelidir ● Sensörler yardımıyla elde edilen veriler sayesinde kullanıcı, robotun gitmek istediği konum için haritada oluşturulan yolları kolayca gözlemleyebilmelidir. ● Hedefe doğru giderken hatasız bir şekilde yolun takip edildiğini görebilmelidir. ● Kullanıcının, robotun hedefine ulaştığı konusunda bilgilendirildiğinden emin olunması gerekmektedir. <p style="padding-left: 40px;">2. Sistemin Düzgün Çalışması için Gereksinimler</p> <ul style="list-style-type: none"> ● Sistemin donanım entegrasyonu sağlanmalıdır. ● Bilgisayar ve AUTOBOT arasında iletişimin doğru işlediğinden emin olunmalıdır. ● AUTOBOT işlevlerini kontrol etmek için ilk testlerin yapılması gerekmektedir. ● Lidar yardımıyla çevrede engel varlığının taraması gerekmektedir. 	

- Robotun yol bulma algoritmaları yardımıyla en kısa ve uygun yolu seçerek hedef noktaya ulaşması gerekmektedir.
- Hedeflerden kaçarak yol takibi sağlanmalıdır.
- Hedefe ulaştığında geri bildirim vermelidir.

3. Robotun Hedeflenen Fonksiyonellikler için Gereksinimler

- Gerekli donanım bileşenleri kullanılmalıdır.
- AUTOBOT'un hareket edip başarılı bir şekilde çalışabilmesi için ideal/doğru sensörlerin kullanılması gerekmektedir.
- Kullanıcı ve AUTOBOT arasındaki iletişimin başlatılması gerekmektedir.
- Yol planlaması için bir hedef tanımlanmalıdır.
- AUTOBOT'un oluşturduğu harita gözlemlenebilmelidir.
- Belirtilen işlemler yerine getirilmelidir.
- AUTOBOT'un kullanılacağı ortamın tanıtılması gerekmektedir (Harita oluşturma).
- Çevreyi temsil eden gözlemlenebilir bir grafik oluşturulmalıdır.
- Oluşturulan harita yeniden kullanabilmek için kaydedilmelidir.
- AUTOBOT'un konumunu ve takip etmesi gereken yönü bulması gerekmektedir.
- Gidilecek en kısa ve en iyi rotanın belirlenmesi gerekmektedir.
- Kısa mesafedeki engellerin tespit edilmesi gerekmektedir.
- Yol planlamasında engellerden kaçınarak ilerlemektedir.
- AUTOBOT' mevcut konumunu minimum toleransla belirlemelidir.
- Çevrenin 2D/3D haritası oluşturulmalıdır.

3. Tasarım

AUTOBOT projesinin prototip olarak belirtilen tüm işlevselliklerini yerine getirmesi için yazılımsal ve donanımsal katmanlarının doğru bir şekilde tasarlanması ve kullanılması gerekmektedir. Donanım bileşenleri olarak Nvidia'nın Jetson Nano gömülü bilgisayarı, Arduino Mega, L298N motor sürücüsü ve mesafe ölçümü için lidar kullanılmıştır. Bu mesafe ölçümleri haritalama, navigasyon ve engellerden kaçınma için kullanılacaktır. Arduino mega'yı bir mikro işlemci olarak seçtik çünkü birçok I/O portu bulunmaktadır, bu yüzden sensörlerimiz ve motorlarımız için yeterli I/O portumuz bulunmaktadır. Robotun motorlarını L298N motor sürücüsü kullanarak kontrol edilmiştir. Arduino motorlarının hız ve yön bilgilerini L298N motor sürücü kartına göndererek kullanılmıştır. Robot ve uzak bilgisayar arasındaki iletişim Kablosuz bir ağ üzerinden uygulanmıştır. Yazılım platformu ise Arduino IDE, Visual Studio, ROS (Robotic Operating System), Matlab araçları birleştirilerek oluşturuldu. Jetson Nano, donanımdaki verileri işlemek için sistemdeki ana bilgisayar olarak kullanılacaktır bu nedenle temel yapıtaşımızdır. Yazılım mimarimiz çok işlemlidir. Projemizde katmanlı yazılım sistem mimarisine sahip olduğumuz için projemizi hiyerarşik mimariyle tasarladık. Bu katmanlı yapı, lidar ile veri okuma ile başlar. İkinci katmanda bu verileri yerelleştirme,gezinme(yol bulma) ve engellerden kaçınma için kullanıyoruz. Genel olarak, robotumuzun yönü ve hareketleri bu katmanlardan sonra belirlenir. İstenilen işlevsellikleri sağlamak için, yazılım alt sistemlerimizin çoğu, gerekli olduğu bir durumda birbirleriyle iletişim kurabilecek şekilde tasarlanmış bulunmaktadır. Robotun temel olarak gerçekleştirdiği 4 işlevselliği bulunmaktadır. Bunlar sırasıyla şu şekilde tasarlanmıştır,

- Robotun bir hedeften diğer bir hedefe gidebilmesi için öncelikle haritalama işleminin gerçekleştirilmesi
- Lokalizasyon işlemi olarak robotun oluşturulan map üzerinde lidar ve odometri verilerinin kullanılarak anlık olarak nerede olduğunun hesaplanması,
- Haritalama ve lokalizasyon işlemini tamamladıktan sonra istenilen noktaya ulaşmak için en kısa ve verimli yol planının oluşturulması,
- Yol Takip Etme işlemi, robotun oluşturulan yol üzerindeki noktaları olabildiğince yakın şekilde takip edebileceği, aynı zamanda yol üzerinde bir engel ile karşılaşırsa bu engelden kaçınmak için gideceği yönü tekrar hesaplayabileceği şekilde tasarlanmıştır.

4. Uygulama ve Testler

Kullanılan Robot Araç Kiti, tekerlekleri, enkoderli motorları ve çerçeveleri içerir. Lidar, haritalama ve lokalizasyon uygulamalarında kullanılmak üzere robotun bulunduğu ortamdan mesafe bilgilerini toplamak için kullanılır. Arduino Mega, robotun motorlarını kontrol etmek için kullanılmıştır. L298n motor sürücüsü, motorların yönünü ve hızını kontrol etmek için kullanılmıştır. Bir L298N motor sürücüsü iki motorun kontrolünü sağlar, bu yüzden prototip robotun dört motoru için iki L298N motor sürücüsü kullanılmıştır.

Projemiz yazılımsal ve donanımsal bileşenlere sahip olduğu için uygulamaları ve test etme aşamaları iki adımda tamamlanmıştır. Bu adımlar da uygulama kısmı, donanımsal bileşenlerin uygulaması ve yazılımsal bileşenlerin uygulaması olarak ayrılmıştır. Donanımsal uygulamalar ilk olarak, robot araba (tekerlekler, motorlar ve araba kiti) kurulduktan sonra, kontrol edebilmek için robotun üzerine L298N motor sürücüleri ve arduino mega yerleştirilerek ve basit kontroller sağlayarak gerçekleştirilmiştir. Ardından, gömülü bilgisayar robota yerleştirilmiştir ve arduino mega ile seri iletişim bağlantısı kurulmuştur. Daha sonra kapalı ortam hakkında mesafe bilgilerini almak için önemli donanımsal bileşenlerden olan lidar robotun üzerine yerleştirilmiştir. Daha sonra, robot ağ üzerinden uzak bir bilgisayarla kontrol edilmiştir ve test edildiği ortam hakkında mesafe bilgisine erişilmiştir. Projenin yazılım uygulamalarının geliştirme aşamaları aşağıdaki gibi açıklanabilir.

- Robotu uzak bir bilgisayar kontrol etmek,
- Lidar verilerine erişim,
- Uzak bilgisayara anlık lidar verileri gönderme,
- Robotu uzak bir bilgisayarla kontrol etmek, bulunduğu ortamda manuel olarak sürmek ve lidar verilerini toplamak,
- Toplanan lidar verilerini kullanarak haritalama,
- Robotu önceden belirlenmiş bir noktadan başlatarak konum bilgisini hesaplama,
- Harita üzerinde seçilen nokta için yol planlaması,
- Robotun 7. adımda oluşturulan yolu takip etmesi,
- Robotun oluşturulan yolda hareket ederken engellerden kaçınması.

Test aşamalarımız da aynı şekilde yazılımsal ve donanımsal bileşenlerin testini yapma şeklinde gerçekleştirilmiştir. Donanımsal bileşenler proje için önemli olduğu için, aktif ve sorunsuz olarak çalıştığından emin olunması projenin doğru bir şekilde işleyiş gösterebilmesi adına önemlidir. Donanımsal bileşenlerin test edilmesi süreci;

- Tüm bileşenlerin bilgisayara başarıyla bağlandığının test edilmesi,
- Motor sürücülerin ve motorların arduino ile doğru şekilde kontrol edildiğinin test edilmesi,
 - Lidar bağlantısının USB portu üzerinden Jetson Nano'ya bağlandığının test edilme
- Motorların performansının farklı koşullar ve ortamlarda test edilmesi,
- Jetson Nano ve uzak bilgisayar arasındaki bağlantı kablosuz iletişim ile sağlandığından, uzak bilgisayarın kablosuz bileşeninin (Wi-Fi bağlantısı) test edilmesi,
- Şeklinde gerçekleştirilmiştir. Yazılımsal bileşenlerinin testi ise, haberleşme, harita oluşturma, navigasyon ve kullanıcı kontrolü yaparak sağlanmıştır.

Genel olarak prototipin test aşamaları ise şu şekildedir;

- Robot ile olan kablosuz bağlantıda tüm verilerin doğru şekilde ulaştığının test edilmesi,
- Robotun manuel kontrol etmek için gerekli tuş kombinasyonları ile robotun hareketleri arasındaki doğruluğun test edilmesi,
- Anlık olarak lidar verilerine erişilerek doğruluğunun test edilmesi,
- Harita çıkarmak için toplanan lidar verilerinin depolandığının test edilmesi,
- Lidar verileri kullanılarak oluşturulan haritanın, robotun bulunduğu ortam ile tutarlılığının test edilmesi,
- Lokalizasyon işleminin doğru şekilde çalıştığından emin olmak için hesaplanan konum bilgileri ile gerçek konum bilgilerinin gözlemlenerek test edilmesi,
- Yol planlama algoritmaları kullanılarak oluşturulan yolun harita üzerindeki kalıcı engeller ile olan yakınlığının robota engel olup olmadığının test edilmesi (robotun, engeller ile arasında belirli bir mesafe olmalıdır),

- Oluşturulan yolun doğru şekilde takip edildiğinin test edilmesi,
- Oluşturulan yolu takip ederken engellerden kaçınıldığının test edilmesi,
- Simülasyon ortamında robotun konumunun doğru olarak belirtildiğinin test edilmesi,
- Simülasyon ortamında, oluşturulan yolun doğru şekilde gösterildiğinin test edilmesi.

Sonuçlar:

Yapılan bu projede ortaya bir prototip çıkmasını istememizin temel nedeni olarak teknolojik gelişmelerin artmasıyla birlikte robot veya otonom araçların da öneminin artmasını gösterebiliriz. Tasarlanan bu prototip ile bu alana giriş yapmış olup, isteğe göre geliştirilebilecek bir altyapı sağladık. Genel olarak prototipe kazandırmayı hedeflediğimiz Haritalama, Lokalizasyon, Yol Planlama, Yol Takip Etme (Navigasyon) ve Unity Simülasyonu işlevselliğini tamamlamış bulunuyoruz. Harita, robotu kendi ortamında hareket ettirerek ve lidar verilerini toplayarak oluşturulur. Robotun bulunduğu ortamda lidar bileşeni yardımıyla topladığı veriler ile çevresinin haritasını çıkartılır. Ayrıca, robotun gerçek zamanlı konumu, anlık lidar ve odometri verileri kullanılarak elde ediliyor. Yol Planlaması, robotun bulunduğu noktadan, harita üzerinde seçilen herhangi bir noktaya ulaşması için yapıyor. Daha sonra sonra yol takibi(navigasyon), robotun hedef noktaya ulaşmak için yoldaki noktaları etrafındaki engellerden kaçarak takip etmesi, ve otonom bir şekilde hareket etmesi sağlıyor. Bu basamaklar sayesinde robot istenilen işlevselliği göstermiş oluyor. Bunlara ek olarak, kullanıcıların robotun konumunu ve hareketlerini rahatlıkla izleyebilmesi için 2D/3D Simülasyon ortamı sağlanmıştır. Yaptığımız proje otonom olarak kullanılabilen araçlarda görülen işlevselliği göstermektedir. Bu nedenle, bu alanda geliştirilen projenin otonom alanda yapılabilecek gelişmeleri destekleyecek altyapıya sahip olduğunu belirtmek isteriz. Bu prototipin şu sağladığı ve sağlayabileceği yararlar ;

- Otonom bir şekilde bu işlevselliğin gerçekleştirilmesine ve yeni özellikler eklenmesine altyapı sağlamaktadır.(Örneğin Otonom Haritalama)
- Kullanıcıların yararı için tasarlanmış Simülasyon ve Kullanıcı arayüzü ile kullanıcılar programlama bilgisine sahip olmadan, robotun konumunu ve hareketlerini daha kolay takip edebilecek ve yönetebileceklerdir.
- Toplanan lidar verileri ile oluşturulan haritayı kullanan simülasyon programı, robotun yol takibi ve engellerden dinamik olarak kaçınma gibi işlemleri yapay zeka ile öğrenmesi için uygun bir ortam oluşturabilecek durumdadır.
- Rutin işlerde geliştirilmiş verimlilik ve hız sağlayabilir.
- İnsanlar için ortaya çıkabilecek tehlikeleri en aza indirir ve böylelikle tehlikeli bölgelerdeki insan sayısını azaltabilir.
- Üretimi arttırabilir. İnsanlardan farklı olarak, robotlar yorulmaz ve sürekli olarak çalışabilir ve üretimin yavaşlama olmadan devam etmesine izin verir.
- İşçilik maliyetlerini azaltabilir. Bazı basit rutin çalışmalarda, bir insandan çok daha ucuza mal olur.
- İnsanların iş kalitesini arttırmada yarar sağlayabilir. AUTOBOT rutin işleri yapabileceğinden, bu işlerde çalışan insanlar daha kaliteli işlerde zamanlarını harcayabilirler.
- AUTOBOT, insanlar tarafından yapılan manuel kontrol hatalarını ortadan kaldırabilir, çünkü insanlardan daha hatasız bir şekilde çalışabilir.

Son olarak, gelecekte AUTOBOT projesine eklenen işlevselliği geliştirilerek robota tümüyle otonom çalışma yeteneği eklenebilir. Otonom çalışmayı destekleyecek bir işlevsellik olarak Otonom Haritalama örnek gösterilebilir. Bu sayede haritayı otonom bir şekilde oluşturarak tümüyle insansız kullanımı sağlayabilir. Çeşitli alanlarda daha hızlı, verimli ve hatasız çalışma sağlayabilir. 3D lidar kullanılarak ve 3D Haritalama kullanılarak, çevrenin daha detaylı ve renkli bir haritası çıkarılabilir. Çeşitli kullanım alanları için gereksinim duyulduğunda, AUTOBOT farklı boyutlarda tekrar tasarlanabilir ve aktif hale getirilebilir.

Proje Kodu ve Adı:	OSP: Offer Based Online Service Platform
Proje Takımı:	Ege CIFTCI, Ertunga Kuthan KURT, Omer PINAR
Proje Danışmanları:	Doç.Dr. Ahmet Tuncay ERCAN
Proje Çıktıları	1. Final Report 2. Requirements Specifications Document Design 3. Specifications Document Product Manual
Proje Web Adresi:	-

Proje Özeti

1. Giriş

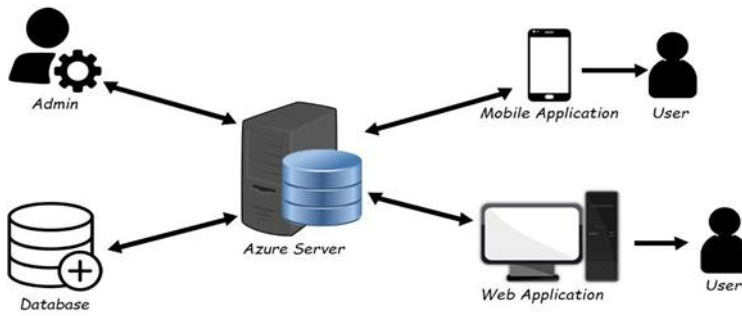
Günümüzde hız, insanların ihtiyaçlarını karşılamada önemli bir gerekliliktir. İnsanlar internet üzerinden hizmetin yaygın olmadığı zamanlarda ihtiyaçlarını giderebilmek için daha fazla çaba harcamıştır. Teknolojinin gelişimiyle insanlar ihtiyaçlarını genelde internet üzerinden online olarak karşılamaya başlamıştır. İnsanların hizmet almak istediği ihtiyaçlarına internet üzerinden karşılık bulabilmesi zaman ve ulaşılabilirlik açılarından insanlara büyük bir fayda sağlamıştır. Bu bilgiler ışığında hareket ederek projemizi oluşturduk.

Bu platform kullanıcıların ihtiyaçlarını hızlı ve güvenli bir şekilde karşılamayı amaçlamaktadır. Bunu sağlamak için projemiz kullanıcıları tek bir platformda bir araya getirir ve istedikleri hizmetlere kolayca erişebilmeleri için bu platformdaki diğer kullanıcılarla etkileşime girme olanağı sağlar. Bu şekilde kullanıcılar ihtiyaçlarına hızlı çözümler bulacaktır. Güvenliği sağlamak için, kullanıcılar sisteme kaydolarken istenen bilgileri tam ve doğru bir şekilde girmeleri halinde platforma erişebileceklerdir. Oluşturulan ilanların gerekliliklerini karşılayabilen kullanıcılar, ilgilendikleri ilan için teklif gönderebilir. İlanın sahibi gelen teklifler arasından kendisine uygun teklifi onaylayabilecektir. Bu seçimi yaparken teklif veren kullanıcıların profilindeki yorumlar ve puanlardan yardım alabilir.

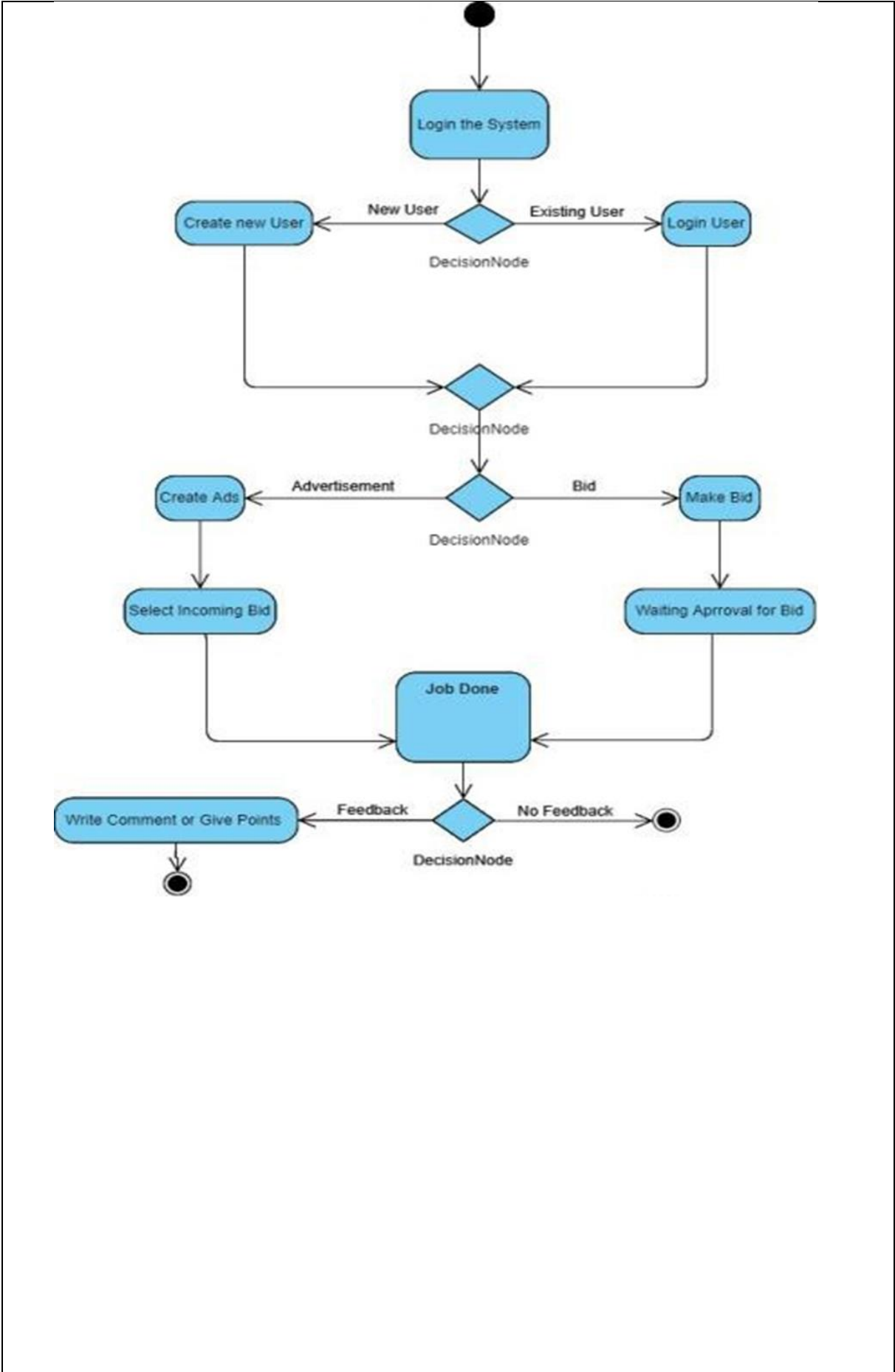
Projemiz, web ve mobil platformlarda geliştirilip, mobil platformda android tabanlı olarak kullanıcılarına hizmet sunacaktır.

2. Tasarım

Offer Based Online Service Platform Yazılım Sistemi Yapısı :

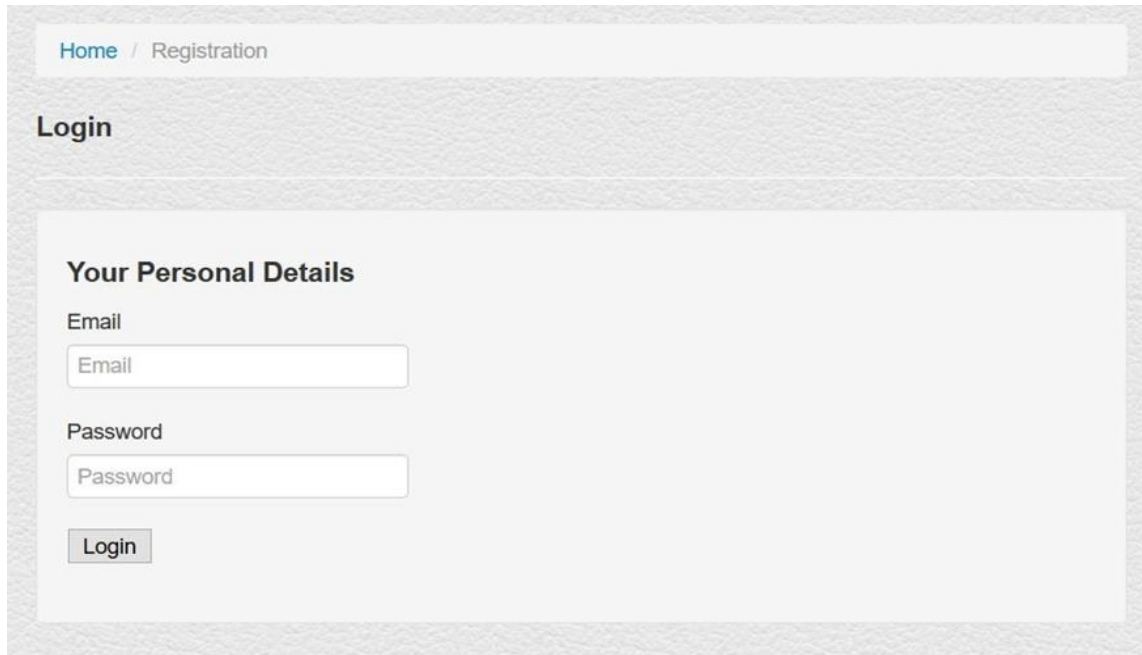
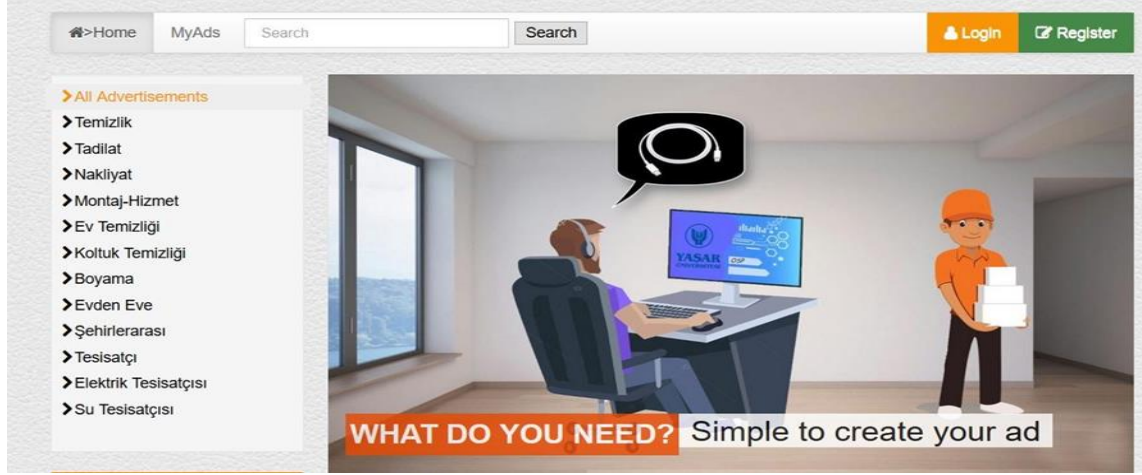


İş sistemlerinin modellenmesinde kullanılan “activity diagram” ile modellenen sistemdeki iş akışı adım adım gösterilmiştir.



Web Arayüzü :

Web tasarımı kullanıcıların kullanışlı bir arayüz ile karşılaşması istenilerek tasarlanmıştır. Kullanıcıların rahatlıkla anlayabilmesi için basit ve kolay bir tasarım yapılmıştır.



3. Testler

Mobile geliştirme için IOS platformu araştırıldı ve incelendi , çeşitli eksiklikler neticesi ile bu platform iptal edildi. Yapılmış olan Web , mobil uygulamaları ve Web Api servisi çalıştırılıp , uygun bir biçimde çalışıp çalışmadığı test edildi. Test esnasında karşılaşılan hatalar düzeltildi. Azure Sunucularının hatasından dolayı ortak veritabanı erişimi sağlanamamıştır. Bir süre sonra sunucuların hizmete açılması ile Azure server ile ortak veritabanı erişimi sağlanmıştır. Çeşitli testlerin ardından bu sunucunun projemize uygun olduğu gözlemlenmiştir.

4. Sonuç

Offer Based Online Service Platform(OSP) ile insanların ihtiyaçlarını hızlı ve güvenilir bir şekilde halletmesi hedeflenmiştir.

Kullanıcıların hizmet alıp hizmet verdiği çeşitli platformlar vardır fakat bu platformlarda hizmetin bedeli standart bir şekilde fiyatlandırılmıştır. Bu durum üzerinden yola çıkarak kullanıcıların vermek istediği hizmeti kendisinin fiyatlandıracağı, hizmeti alacak olan kullanıcının da gelen teklifler arasından değerlendirme yapabileceği teklif tabanlı bir sistem oluşturmaya karar verildi.

Bunu sağlamak için projemiz kullanıcıları tek bir platformda bir araya getirir ve istedikleri hizmetlere kolayca erişebilmeleri için bu platformdaki diğer kullanıcılarla etkileşime girme olanağı sağlar. Bu şekilde kullanıcılar ihtiyaçlarına hızlı çözümler bulacaktır.

Platformun çalışma mantığını kısaca özetlersek ; sisteme kayıt olan kullanıcı , diğer kullanıcıların açmış olduğu ilanlara teklif vererek talebini iletir ve teklifi onaylanır ise ilan sahibi olan kullanıcıya hizmet verir. Aynı kullanıcı, herhangi bir ihtiyacına istinaden ilan oluşturup , aynı yolu izleyip değerlendirmesini yaparak hizmet alabilir.

Proje Kodu ve Adı:	FIWE - Factoring Integers with ECM
Proje Takımı:	Asena Durukan, Aslı Altıparmak, Elif Özbay, Hasan Ozan Soğukpınar Nuri Furkan Pala
Proje Danışmanları:	Dr. Ögr. Üyesi Hüseyin Hışıl
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Raporu 2. Yazılım Gereksinim Dokümanı 3. Tasarım Dokümanı 4. Kullanım Kılavuzu 5. Yazılım 6. Poster 7. İnternet Sitesi
Proje Web Adresi:	-
Proje Özeti	
<p>1. Giriş</p> <p>Arjen K. Lenstra tarafından 1987 yılında geliştirilen Eliptik Eğri Metodu (EEM), tam sayıları çarpanlarına ayıran bir algoritmadır. Bu algoritma tam sayıları çarpanlarına ayırırken Weierstrass, Edwards ve Montgomery gibi eliptik eğrileri kullanır. Bu projede hızından ve uygunluğundan dolayı Montgomery eğrileri seçilmiştir. Eğriler ve eğrilerin üzerindeki noktalar afin ve projektif koordinatlar ile temsil edilebilir. Projektif koordinatların afin koordinatlardan daha hızlı olması sebebiyle, EEM algoritmasının uygulaması projektif koordinatlar kullanılarak yapılmıştır. EEM algoritmasının ihtiyaç duyduğu modüler ters alma ve Ortak Bölenlerin En Büyüğü (OBEB) işlemi için, hızı ve güncel olması dolayısıyla, safegcd algoritması gerçekleştirilmiş, ECM algoritmasına eklenmesi gelecek çalışması olarak belirlenmiştir. Algoritma, GPU'nun SIMD mimarisi üzerinde optimize ve paralelize edilmiştir. Bu sayede, çok basamaklı birden fazla sayı aynı anda çarpanlarına ayrılır.</p> <p>2. Gereksinimler</p> <p>Eliptik Eğri Metodu'nu gerçekleyebilmek için fonksiyonel ve fonksiyonel olmayan gereksinimler bulunmaktadır. Fonksiyonel gereksinimler üçe ayrılabilir:</p> <ul style="list-style-type: none"> • Matematiksel Gereksinimler: <ul style="list-style-type: none"> ○ Büyük sayı temsili ve işlemleri. ○ Eliptik eğri temsili, nokta temsili ve işlemleri ○ OBEB işlemi ○ Modüler ters alma işlemi • Paralleleştirme Gereksinimleri: <ul style="list-style-type: none"> ○ Çok basamaklı birden fazla sayının aynı anda çarpanlarına ayrılabilmesi. <p style="text-align: center;">Kullanım Gereksinimleri:</p> <ul style="list-style-type: none"> ○ Çok basamaklı bir sayının çarpanlarına kolayca ayrılabilirdiği bir komut satırı. <p style="text-align: center;">Fonksiyonel olmayan bir gereksinim ise OBEB işleminin kısa sürede çalışmasıdır.</p>	

3. Tasarım

Sistem, EEM algoritmasının, GPU'nun SIMD mimarisi üzerinde gerçekleştirilmesinden meydana gelir. Ayrıca yazılım, farklı matematiksel işlemlere ihtiyaç duyar ve bu işlemler farklı kütüphaneler oluşturularak sağlanmıştır. Bu kütüphanelerin isimleri; Çok Basamaklı Sayı Kütüphanesi Montgomery Kütüphanesi, EEM Kütüphanesi'dir. Bu kütüphaneler birçok yapıyı ve fonksiyonu içerir. Sistem beş ana bölümden oluşur:

- Çok Basamaklı Sayı Kütüphanesi'nin kurulması
- Montgomery Eğri Kütüphanesi'nin kurulması
- EEM Kütüphanesi'nin kurulması
- Safegcd'nin gerçekleştirilmesi
- Yazılımın SIMD mimarisinde paralelleştirilmesi

4. Uygulama ve Testler

Projenin gerçekleştirilmesi, EEM algoritmasının C dilinde kodlanması ile başladı. Daha sonra C dilinde yazılmış olan kodlar CUDA diline çevrildi. Bu çevirme sayesinde paralelleştirilme süreci başladı. Bu sürecin sonunda, çok basamaklı birden fazla sayının çarpanlarına ayrılması mümkün oldu.

Proje boyunca yazılan her bir fonksiyon birer birer test edildi. Bütün test prosedürleri C ve CUDA dillerinde kodlandı ve modüllere eklendi. Bütün fonksiyonlar 10000 kere test edildi ve %100 başarı sağlandı.

5. Sonuçlar

Proje, literatür taraması ile başladı. Araştırmaların rehberliğinde, Tasarım Dokümanı oluşturulmaya başlandı. Eş zamanlı olarak, EEM ve safegcd algoritmalarının gerçekleştirilmesine C dili kullanılarak başlandı. Daha sonra, paralelleştirme amacıyla CUDA diline geçildi.

Maliyet Analizi

Bu proje için dokuz ayda toplamda 242 adam günü harcadı. Bir başka deyişle, her proje üyesi 48 adam günü harcadı.

Projenin Faydaları

Projenin faydalarından bir tanesi kriptoloji, kriptanaliz ve ilgili alanlarda çalışacak olan insanlara rehberlik edebilecek bir kaynak sunmak. Yazılımı herhangi biri raporları okuyarak ve kaynak kodları inceleyerek anlayabilir ve kullanabilir. Dahası, grup üyelerinin, bu projenin konusu olan çarpanlarına ayırma problemini ve çözümünü anlamasına yardımcı oldu.

Gelecekteki Çalışmalar

- EEM algoritmasının ikinci fazı uygulanabilir.
- EEM algoritmasının gerçekleştirilmesinde afin koordinatlar kullanılabilir ve bir karşılaştırma yapılabilir.
- Yazılımın paralelleştirme kısmı, yeni GPU'lar veya farklı mimariler üzerinde gerçekleştirilebilir.
- Safegcd algoritmasının ECM algoritmasına eklenmesi.

Proje Kodu ve Adı:	DJAVAC: Automated Load Distribution Engine for Java
Proje Takımı:	Halil Bülent Orhon, Yaren Gündüz Fulya Ertay, Berke Sayan
Proje Danışmanları:	Dr. Öğr. Üyesi Hüseyin Hışıl
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. Requirements Specifications Document 3. Design Specifications Document 4. Product Manual 5. Product-Hardware Subsystem, if any 6. Product-Software Subsystem 7. Any other deliverables
Proje Web Adresi:	https://djavacteam.wixsite.com/djavac
Proje Özeti	
<p>1.Giriş</p> <p>Geliştirdiğimiz DJAVAC projemiz, yazılım alanıyla ilgili belirli bir sorunu çözer. Java programcısının, kod yazarken yaşadığı sıkıntıları çözmeyi amaçlamaktayız. Örneğin, Java programcı aynı metodları birçok kez, editör programda yazarak, zaman kaybetmektedir. Bu nedenle hedefimiz, programcının yazdığı kodları ağ üzerinden alarak, sunucu makinaya göndermektedir. Bu şekilde, sunucu, istemcinin kodlarını çalıştırır ve sonucu tekrar istemciye, arayüz programı aracılığıyla iletir. Java dili kodlayan programcı kendi kodlarını yazar ve bu kodun 'dağıtılmış' anahtar kelime olarak pragması varsa, programcının kodu, sunucu yürütmesine uygun hale getirmek için ayrıştırılır, dağıtılır ve oluşturulur. Sunucu yürütme tamamlandığında, kodun çıktıları veya sonuçları her zamanki gibi istemciye döndürülür. Bu işlemi yaparken istemci kodu başka bir java dosyası olarak kopyalanır ve koddaki diğer değişiklikler burada tamamlanır. Java RMI yardımıyla, RMI arabirimini kullanarak, istemci tarafı kodlarındaki RMI uygulama dosyalarındaki değişiklikleri iletme için daha iyi bir bağlantı sağlar.</p> <p>2.Gereksinimler</p> <p>DJAVAC projesinin çalışabilmesi için bazı işlevsel gereksinimlere ihtiyaç duymaktadır. Örneğin, herhangi bir Java editörü, Java ortamı (JDK, JRE, JVM) ve Windows 10 veya Ubuntu işletim sistemlerine ihtiyaç duymaktadır. Ayrıca, sistem iki kullanıcıya ihtiyaç duymaktadır. Bunlar istemci ve kullanıcı. İstemci ve sunucunun çalışabilmesi için, her biri ilgili kullanıcı arayüz programını çalıştırmalıdır. DJAVAC takım üyeleri, projenin gereken kodlarını, Eclipse Java editör programı kullanarak yazmıştır. Ayrıca, proje çalışabilmek için her iki kullanıcının da belli bir ağa bağlı olması gereklidir. Projemiz, istemciden kodları alıp, sunucu tarafında çalıştırabilmesi için, RMI API ye ihtiyaç duymaktadır. Bunlara ek olarak, DJAVAC projesi işlevsel olmayan gereksinimlere de ihtiyacı vardır. Örneğin, performans gereksiniminde, DJAVAC sistemi ölçeklendirilebilir olmalıdır. İstemci veya sunucu sayısı artarsa, sistem hem istemci hem de sunucular için işlemleri hızlı bir şekilde yürütmelidir. Bu ölçeklenebilirlik dağıtılmış bir şekilde karşılanmalıdır. DJAVAC sistemi, daha erişilebilir olması için istemci ve sunucu arayüzlerine sahip olacaktır. Sistem yalnızca Java editör programını, Java programlama dilini ve RMI mimarisini kullanmalıdır. Diğer bir işlevsel olmayan gereksinim ise, güvenlik gereksinimidir. DJAVAC sisteminde, hem istemci hem de sunucu güvenlik duvarlarını durdurmalıdır. İstemci dağıtılmış kodunu sunucuya gönderdiğinde veya sunucu istemcinin işlevinin sonucunu istemciye döndürdüğünde bu önemlidir. Ayrıca, DJAVAC sisteminin özgünlük mekanizmasına ihtiyacı vardır. İstemci ve sunucu birbirleriyle iletişim kurduğunda, istemcinin veya kullanıcının kimliğini doğrulamak için IP adreslerini ve bağlantı noktası numaralarını girerler. Ayrıca, DJAVAC sistemi TCP / IP sistemini kullanarak iletişim sağlar.</p>	

3. Tasarım

Projemizin tasarım kapsamında iki arayüz dosyamız vardır. Sunucu, “Sunucu arayüz programı” ile etkileşime girer ve İstemci “İstemci arayüz programı” ile etkileşime girer. Sunucu ilk olarak, Central Server, Server ve RMI Server düğmelerine bastıktan sonra, istemcinin kodunu kabul etmeye hazır olacaktır. Ardından, İstemci Dağıt ve Kod Oluşturucu düğmesine ve Yürüt düğmesine basar, istemci kodlarını, sunucuya gönderir ve sunucu kod sonucunu istemciye döndürür. Tüm bu işlemler RMI API kullanılarak tasarlanmıştır. Ayrıca, projemizde yer alan bazı gerekli sınıflar şunlardır:

1. Parser (Dağıtılmış yöntemleri tespit etmek için istemci kodunu küçük parçalara ayırır.)
2. Kod Jeneratör (İstemci ve Sunucu iletişimi için arayüz, uygulama ve ana yöntemleri oluşturur).
3. RMI sunucusu başladıktan sonra istemci kodunu engellemek için kullanılan yöntem, bu yöntem komut penceresinde yürütülmeye başlar ve sunucu sınıfından istemciye döner.
4. Birden çok sunucu ve çoklu istemci iletişim sistemi, çoklu thread yaratılmıştır.

4.Uygulama Ve Testler

Projemizi uygulama kapsamında, 3.Tasarım bölümünde belirttiğimiz metodları kullandık. Ardından projemizi, virtual box üzerinden sanal işletim sistemleri kurarak test ettik. Test işlemlerimizi hem Windows 10 hem de Ubuntu işletim sistemleri üzerinden gerçekleştirdik. Örneğin, bazı JAVA kodları, Eclipse programı kullanılarak, istemci tarafında yazıldı, ardından istemci arayüz programı çalıştırılarak, bu kodlar sunucuya gönderildi, ve ardından RMI Sunucusunda bu kodlar çalıştırılıp, istemci arayüz programına, sonuçları istemcinin makinesine gönderildi. Testlerimizin sonucunda, istemci kodlarının, sunucu üzerinden ne kadar sürede çalıştırılıp geri döndürüldüğü, yazılan kodun uzun ve karmaşık olmasına bağlı olduğunu gözlemledik.

5.Sonuçlar

Sonuç olarak, projemizi daha önce bitirmeyi planladığımız şekilde bitirdik. Öncelikle, COMP4910 dersi kapsamında, danışmanımızla toplantılar yaparak, projemizin ne olması gerektiğini belirledik, daha sonra yazılım ve donanım gereksinimlerimizi belirledik, gerekli yöntemlerin neler olabileceğini araştırdık ve ikinci yarıyılı DJAVAC projesini uygulamaya geçirdik. DJAVAC birden fazla istemciyle çalışmayı sağlar ve bu istemciler birden çok sunucuya bağlanabilir. Bu sistemi uygularken, istemci ve sunucular arasındaki bağlantılarda daha önce «dağıtılmış» pragma olarak adlandırılan bir yöntemi çağırmak için Java RMI kullanılmaktadır. 'Dağıtılmış' pragmaya sahip işlev sunucuya aktarılabilir. Sunucu, istemcinin kodunu derler ve bu kodları JAVA RMI üzerinden istemcinin makinesine döndürmeye hazırdır. Sistemimizi test ettikten sonra, DJAVAC' ın yürütme süresinin, istemcinin kodunun ne kadar uzun ve karmaşık olduğuyula ilgili olduğunu gözlemledik. İstemci kodu TCP / IP ağı üzerinden dağıtıldığında, Sunucu kodunu çalıştırır ve İstemciye geri döndürür. İstemci / Sunucu iletişimi başarıyla sağlandı. İstemciler çok fazla zaman harcamadan daha kaliteli Java kodları yazabilir. Kod üreticileri ve ayrıştırıcı istemciden almak için ek açıklamayı algılamaya izin verir.. Ayrıca, hangi RMI Sunucusunun müsait olduğunu, istemciye o sunucunun IP adresini sorunsuz bir şekilde gönderildiğini gözlemledik. Eğer, Java programcılar çok fazla zaman harcamadan daha kaliteli Java kodları yazmak istiyorsa, DJAVAC kullanmalıdır.

Proje Kodu ve Adı:	SmartRF
Proje Takımı:	Gökhan Yıldız, Ezgi Gönül
Proje Danışmanları:	Prof. Dr. Mehmet Ufuk Çağlayan
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. RSD Final 3. DSD Final 4. PM Final 5. Source Codes (zip) 6. Poster 7. Slide 8. Web page 9. Project Summary 10. Online Presentation
Proje Web Adresi:	https://smartrf.online/ (Ücretsiz süresi dolmuştur) https://gokhyildiz.pythonanywhere.com/

Proje Özeti

1. Giriş

Envanter yönetimi etkin bir tedarik zinciri yönetiminin sağlanması için kritik noktalardan biridir. Envanterleri dikkatli bir şekilde takip etmemek ve stokları düzenli olarak saymamak işletmeler için para ve zaman kaybına neden olmaktadır. Barkod teknolojisi ürün takibini sağlamak için kullanılan yöntemlerden biridir. Fakat bu teknoloji ile depoda bulunan ürün etiketlerinin çalışanlar tarafından manuel şekilde tek tek taranması gerekmektedir. Bu da deponun büyüklüğüyle orantılı olarak zaman kaybına yol açmakta ve ürünlerin gerçek zamanlı izlenebilirliğini engellemektedir. Aynı zamanda barkod teknolojisiyle depo taramasını otomatize etmek mümkün değildir.

SmartRF projesi ile kullanıcılara zaman ve mekan fark etmeksizin uzaktan kontrollü bir envanter yönetim sistemi sunmak amaçlanmıştır. Projede RFID teknolojisi kullanılarak çalışan gücünü minimuma indirip depo taramasının kısa sürede yapılması ve tarama sonucunda etiketlerin mevcut durumunun web ve mobil arayüz aracılığıyla kullanıcıya gösterilmesi sağlanmıştır. Ek olarak depo güvenliğini arttırmak amacıyla tarama işlemleri gece saatlerinde kullanıcı bağımsız gerçekleşecek şekilde otomatize edilmiştir. Bilindiği üzere bir işletmenin faaliyetini yürütebilmesi için ürünlerin müşteri talebinden önce tedarik edilerek stok yapılması oldukça önemlidir. Projede makine öğrenmesi algoritmaları kullanılarak kullanıcıya belirli tarih aralığında satışı yapılan ürünlerin analizini yapma imkanı sunulmuştur. Bu sayede işletme geçmiş veri analizlerini değerlendirip müşteri talebi gerçekleşmeden önceden gerekli ürünleri tedarik edebilecektir. Projenin donanım kısmı Yaşar Üniversitesi Elektrik Elektronik Mühendisliği öğrencileri tarafından tasarlanmıştır. Etiket taramasının tutarlı olması ve RFID okuyucu cihazının uzak menzilli tarama yapabilmesi için RFID okuyucuya anten entegrasyonu yapılmış ve tarama işlemlerinin farklı açılarda yapılması sağlanmıştır. Bu sayede işletmelere düşük maliyetli donanım çözümü sunulmaktadır.

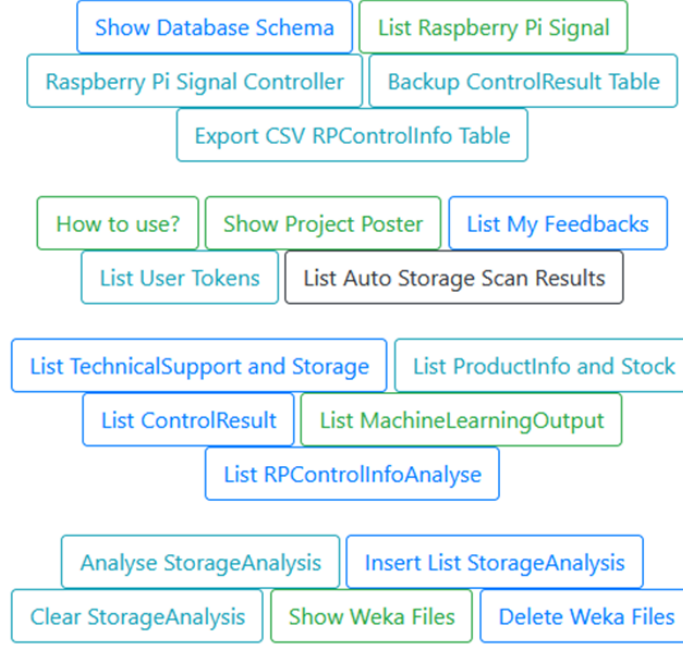
2. Gereksinimler

Proje gereksinimleri temel olarak yazılımsal ve donanımsal olmak üzere ikiye ayrılmaktadır. Yazılımsal gereksinimler kullanıcının envanter yönetim sisteminde ihtiyaç duyacağı fonksiyonlara göre belirlenmiştir. Kullanıcı arayüzünde bulunması gereken fonksiyonlar şu şekildedir:

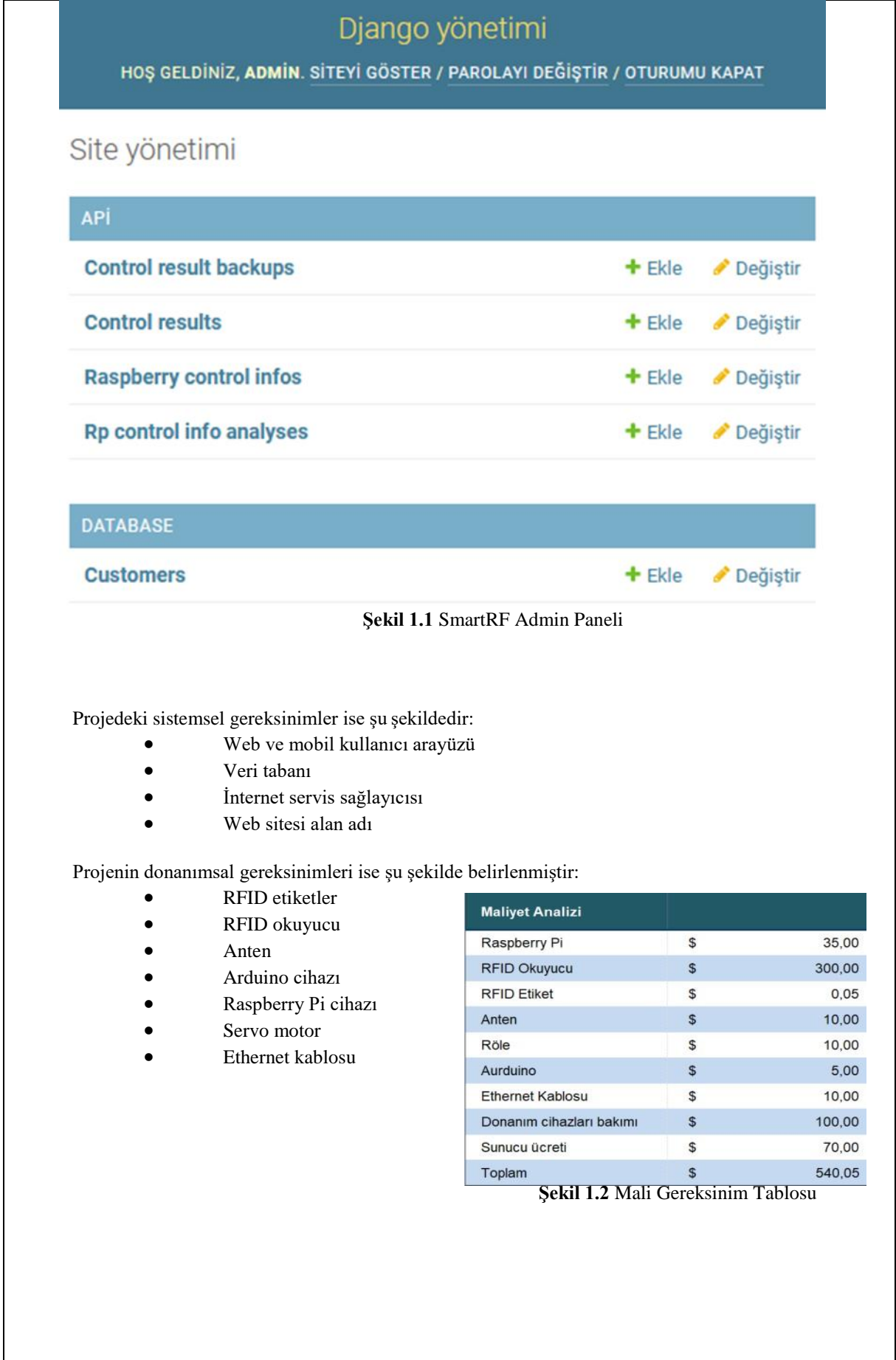
- Statüsü yüksek olan kullanıların depoda bulunan ürün, RFID etiket ve satış gibi depo bilgilerine erişebileceği bir admin paneli bulunmalıdır. Admin paneline giriş yapmadan önce depo güvenliğini sağlamak amacıyla kullanıcının izin seviyesi kontrol edilmelidir.
- Kullanıcının tarama işlemini başlatabilmesi için depo seçmesi ve RFID okuyucuya sinyal göndermesi gerekmektedir.
- Depo taraması sonucunda RFID etiketler için oluşabilecek üç durum söz konusudur. İlk durumda bir etiket veri tabanında kayıtlı olduğu halde tarama sonucunda okunmamış olabilir. Bu ürünün kayıp olduğu anlamına gelmektedir. İkinci durumda ise depo taramasında okunduğu halde veri tabanında kayıtlı olmayan etiket bulunabilir. Son durumda ise bir etiket hem veri tabanında kayıtlı olup hem de tarama sonucunda okunabilir. Kullanıcı yapılan depo tarama işlemlerinden sonra tüm etiketlerin durumunu detaylı şekilde görmelidir.
- Kullanıcının geçmişte yapılan ürün satışlarına makine öğrenmesi algoritmalarıyla analiz yaptırabilmesi için geçmiş ürün satış bilgilerine erişmesi gerekmektedir.
- Analiz sonuçlarının kullanıcıya daha anlaşılır bir şekilde sunulabilmesi için grafiksel bir gösterim gereklidir.
- Sistem depo taraması gece saatlerinde üç kere yapılacak şekilde otomatize edilmeli ve tarama sonucunun bir önceki gün ile karşılaştırılması kullanıcıya sunulmalıdır.



Dashboard Panel

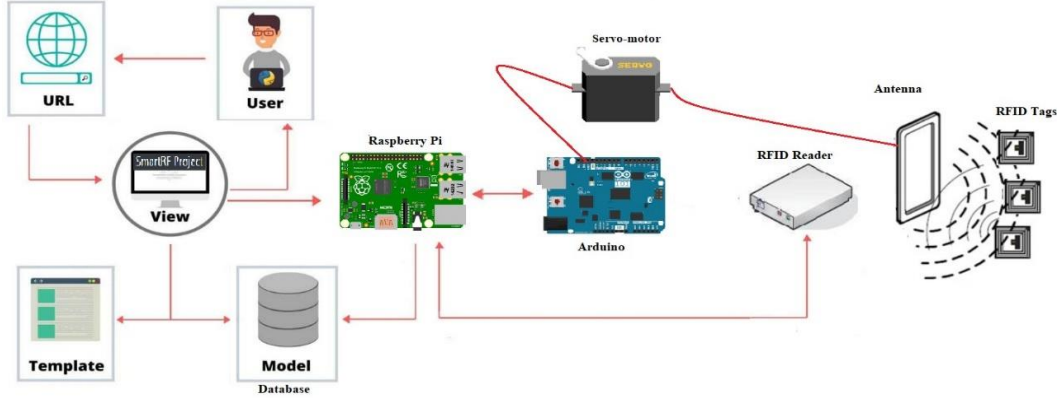


Şekil 1.0 SmartRF Kullanıcı Paneli



3. Tasarım

Sistem gereksinimlerinin tanımlanması sonucunda yazılım ve donanım süreçlerinin tümleşik bir şekilde çalışması amacıyla bir sistem tasarımı oluşturulmuştur.



Şekil 2.0 SmartRF Sistem Tasarımı

İlk olarak kullanıcı arayüzünü oluşturmak amacıyla MVT(Model-View-Templete) mimarisine sahip olmasından dolayı kod tekrarını önleyen ve yüksek güvenlik seviyesine sahip olan Django Web Framework tercih edilmiştir. Her donanım cihazının kendisine ait bir ID numarası vardır. Raspberry pi cihazları belirli zaman aralıklarında kullanıcı tarafından kendisine gönderilen bir tarama işlemi olup olmadığını API vasıtasıyla control etmektedir.

Eğer Raspberry pi cihazı kendisine gönderilen bir tarama sinyali varsa RFID okuyucuyu tetiklemekte ve tarama işlemini başlatmaktadır. RFID okuyuculara bir anten entegre edilmiştir. Bu anten servo motor ile bağlantılıdır. Arduino cihazı tarafından programlanmış olan servo motor farklı açılarda okuma yapmaktadır. Daha sonra veri tutarlılığını sağlamak amacıyla okunan verilerin karşılaştırılması yapıp tutarlı olan veri Raspberry pi cihazı tarafından veri tabanına kaydedilmektedir.

4. Uygulama ve Testler

Uygulama sürecinde birçok teknoloji kullanılmıştır. Kullandığımız teknolojiler şu şekildedir:

- Python, Django Web Framework, Django Rest Framework
- MySQL
- Java, Android Studio
- Weka
- Raspberry Pi, Linux, Free Test and Deployment Servers
- pythonanywhere.com, digitalocean.com, cloudflare.com, godaddy.com
- Nginx, Gunicorn,WSGI



Web projesi Django Web Framework ile Python programlama dili kullanılarak geliştirilmiştir. Tarayıcıdan gelen HTTP isteklerini iletilebilmesi için diğer sunuculara göre daha yüksek performanslı ve CPU kullanımı daha az olan NGINX sunucusu tercih edilmiştir. HTTP isteklerinin Python tarafından yorumlanabilmesi için gerekli protokolleri barındıran WSGI yazılımına ihtiyaç vardır. Bu nedenle ağ geçidi arabirim yazılımı olarak Gunicorn kullanılmıştır. Projeye makine öğrenmesi algoritmalarının entegre edilmesi için kolay kurulumundan dolayı Weka yazılımı tercih edilmiştir. Mobil uygulama Android Studio IDE ile Java dilinde web sitesinin görünümünün arayüzde sunulması amaçlanarak geliştirilmiştir.

SmartRF projesinin uygulama sürecinde kullanıcılar için gerçek hayat senaryoları oluşturulup sistem gereksinimlerinde tanımlanan metodlara, veri tabanı ve sınıf yapılarına testler uygulanmıştır. Test sonuçları girdi, çıktı, beklenen sonuç ve gerçekleşen sonuç olarak değerlendirilmiştir. Eğer gerçekleşen sonuç beklenen sonuçtan farklı ise kullanıcı gereksinimlerine göre metodlar ve veri tabanı yapısı optimize olacak şekilde güncellenmiştir. Proje web sitesinin tüm fonksiyonları bir test sonucu kullanılarak kullanıcı gözünden denetlenmiş ve web sunucusunun herhangi bir hata vermeden çalışması kriteri göz önünde bulundurularak veri bütünlüğü ve güvenliğinin sağlandığından emin olunmuştur. Ayrıca veri tabanına örnek veri seti eklenerek sistemin fonksiyonel yapısının tutarlılığı kontrol edilmiştir. Projenin en önemli noktası olan yazılım ve donanım çözümlerinin aynı anda test edilmesi kısmi bir şekilde gerçekleştirilmiş, Covid-19 sürecinden dolayı depo ortamında yapılması gereken fiziksel testler gerçekleştirilememiştir.

5. Sonuçlar, Öneriler ve Teşekkür

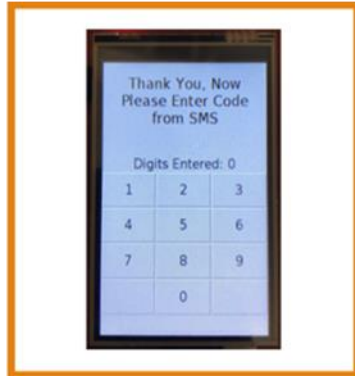
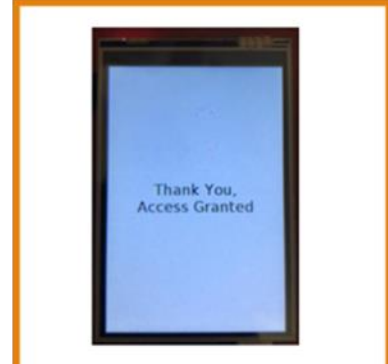
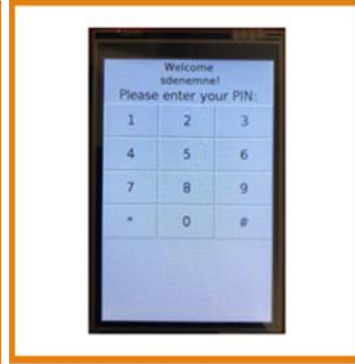
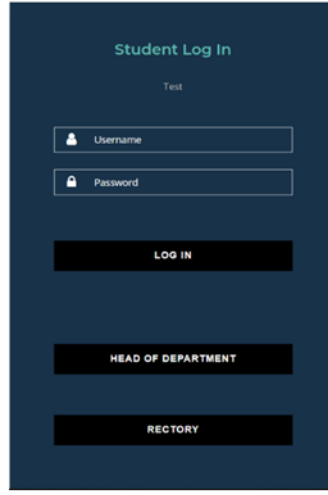
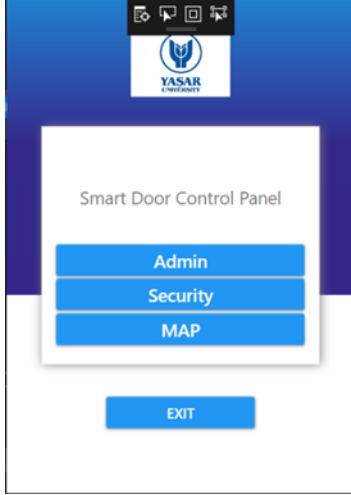
Bir tedarik sisteminin en önemli unsuru envanter takibidir. SmartRF projesi ile kullanıcılara büyük miktarda ürün bilgisinin eş zamanlı olarak kısa süre içinde sunulması sağlanmıştır. Projenin iş gücünü azaltmak, tutarlı ve doğru veri sağlamak, depo güvenliğini arttırmak, veri analizi yapmak, düşük maliyetli donanım çözümleri sunmak gibi bir çok avantajı vardır. Bu da işletmelerin verimliliğini ve rekabet gücünü arttıran anahtar faktörlerden biri olacaktır.

Gelecek çalışmalarda RFID okuyucunun büyük depolarda daha verimli çalışabilmesi ve kısa zamanda daha çok etiket taraması yapılabilmesi için projeye drone teknolojisinin entegre edilmesi ve Android platform için geliştirilen mobil uygulamanın IOS platform için de geliştirilmesi düşünülmektedir. Projede emeği geçen herkese başta İbrahim Zincir, Mustafa Seçmen ve Ceyhan Türkmen olmak üzere sonsuz teşekkürlerimizi sunarız.

Proje Kodu ve Adı:	SDS: Smart Door System
Proje Takımı:	Ahmet Bugra TURPCULU, Ismail Burak TURPCULU, Tolga CANDEMİR
Proje Danışmanları:	Prof. Dr. Mehmet Ufuk Caglayan
Proje Çıktıları	<ol style="list-style-type: none"> 1. Requirements Specifications Document 2. Design Specifications Document 3. Product Manual 4. Final Report
Proje Web Adresi:	https://sdsbitirmev2.azurewebsites.net/
Proje Özeti:	
<p>1. Giriş</p> <p>Çeşitli araştırma alanları teknolojinin büyümesiyle internet üzerine taşınmıştır. Bu nedenle üniversitelerin gelişen teknolojiye ayak uydurması gereklilik haline gelmiştir. Diğer bir yandan teknolojiye ayak uydurmayan üniversiteler öğrenciler tarafından daha az tercih edilmiştir. Bu nedenle, üniversite öğrencilerinin üniversite laboratuvarlarında ve dersliklerinde çalışma saati izni alması için evrak işleriyle uğraşması hem zor hem de zaman alıcı olmuştur. Ayrıca evrak işlerinde gerekli güvenlik önlemleri izlenmesi zorlanmıştır. Gelişen teknoloji sayesinde bu durumlar daha kullanışlı, güvenli ve yararlı hale gelmiştir.</p>	
<p>2. Amaç</p> <p>Smart Door System (SDS), Yaşar Üniversitesi'ndeki öğrencilerin, laboratuvar ve derslik kullanımını artırması hedeflemiştir. Yaşar Üniversitesi mesai saatlerinden sonra dersliklerde ve laboratuvarlarda çalışmak için üniversite tarafından gerekli izin belgesinin doldurulması gerekiyor. İzin belgesinin doldurulma işleminden sonra izin almak isteyen öğrenci, kendi bölümdeki bölüm başkanına imzalatması ve imzalatıldığı izin belgesini de bölüm sekreterliğine vermesi gerekiyor. Bu işlemlerden sonra, bölüm sekreteri de izin kağıdını taratıp, güvenliklere e-posta yoluyla haber vererek, çalışmak istenen dersliğin veya laboratuvarın kilitlenmemesi ve içeride hangi öğrencinin bulunacağını kayıtlı tutmuş oluyor. Bu işlem hem öğrenci tarafından hem de diğer çalışanlar tarafından uğraştırıcı ve zaman alıcıdır. Bizde bu sorundan yola çıkarak, izin belgesini internet ortamına entegre etmeye karar verdik ve Smart Door System(SDS) oluşturduk.</p>	
<p>3. Gereksinimler</p> <p>Okulumuzda mesai saatlerinden sonra sınıflarda ve laboratuvarlarda çalışmak için okul tarafından gerekli izin belgesinin doldurulması gerekiyor. İzin belgesi doldurulma işleminden sonra izin almak isteyen öğrenci, kendi bölümdeki bölüm başkanına imzalatması ve imzalatıldığı izin belgesini de bölüm sekreterliğine vermesi gerekiyor. Bu işlemlerden sonra, bölüm sekreteri de izin kağıdını taratıp, güvenliklere e-posta yoluyla haber vererek, çalışmak istenen sınıfın veya laboratuvarın kilitlenmemesi ve içeride hangi öğrencinin bulunacağını kayıtlı tutmuş oluyor. Bu işlem hem öğrenci tarafından hem de diğer çalışanlar tarafından uğraştırıcı ve zaman alıcıdır. Bizde bu sorundan yola çıkarak, izin belgesini internet ortamına entegre etmeye karar verdik ve SDS oluşturduk.</p>	

4. Tasarım

Sistem tasarımı yapılırken okulumuzun renkleri dikkate alınarak tasarlanmıştır. Masaüstü uygulamasında ve web uygulamasında 3 farklı kullanıcı vardır. Her kullanıcının kendisine özel bir arayüz tasarlanmıştır. Kullanılan arayüzler kullanıcıların sistemi rahat bir şekilde anlaşılması ve kullanılması için oluşturulmuştur.



5. TESTLER

IOS mobil platform incelendi ve denendi. Elde edilen sonuçlara göre, mobil platform sistemimize entegrasi uygun olmadığı tespit edildi ve IOS mobil platformu iptal edildi.

Yapılmış olan web, masaüstü uygulamaları çalıştırılıp denendi Deneme sırasında alınmış olan hatalar düzeltilip tekrar uygulanıp denendi ve bu uygulamalar sorunsuz bir şekilde çalıştığı gözlemlendi.

Amazon AWS sanal sunucusu incelendi ve denendi. Projemizde bulunan sistemler arasındaki bağlantıyı amazon sunucusuyla sorunsuz bir şekilde gerçekleştirdik ama veriler arasındaki iletim süresinin gecikmesinden dolayı Amazon AWS sanal sunucusu iptal edildi. Diğer bir yandan Azure Web sunucu testi incelendi ve denendi.

Azure Web sunucusu projemize sorunsuz şekilde entegre edilip testi yapıldı ve Azure Web sunucusunun sistemimiz için uygun bir sunucu olduğunu gözlemledik.

Donanım tarafında ise Raspberry pi 3B+ cihazı ile 3.5 inch dokunmatik ekran denendi. Gerekli olan dokunmatik ekran kalibrasyonları yapıldı. Ardından cihaza RFID okuyucu USB bağlantısı ile bağlanılıp denendi. Sistem için yazılan kod sorunsuz bir şekilde çalıştığını gözlemledik fakat COVID-19 nedeniyle temin edemediğimiz elektrikli kilit sistemi ile 3 boyutlu yazıcı tarafından yapılacak olan kapı test edilememiştir.

6. SONUÇ

Smart Door System (SDS), Yaşar Üniversitesi'ndeki öğrencilerin, laboratuvar ve sınıf kullanımlarını öğrenci bakımından daha kullanışlı hale getirdik. Yaşar Üniversitesi bakımından ise daha güvenli hale getirdik.

SDS, iki temel uygulamadan oluşmaktadır. Birincisi web uygulaması. Web uygulaması, öğrencilerin rezervasyon yapmak istediği sınıf ve laboratuvarı seçmelerini sağlamaktadır. Seçim işleminden sonra sistem otomatik olarak öğrencini bölüm başkanına mail atarak rezervasyon izni isteyecektir. Bölüm başkanı tarafından onaylanan izin sayesinde okul kartlarıyla, laboratuvar ve sınıfların kapıda bulun kart okuyucusuna okutup, telefonuna gelen SMS ile girişlerini sağlayabilirler.

Masaüstü uygulaması ise okulumuzda bulunan güvenliklerin, Merkezi Akademik Planlama Birimin ve sistemi yönetecek tek yetkilinin kullanılması amaçlanmıştır. Masaüstü uygulamasında, sistemi yönetecek tek yetkili, laboratuvar veya sınıf eklemelerin yapılması, sınıf ve laboratuvar kapasitelerin veya kodların değiştirilmesi işlemlerini yapabilir. Merkezi Akademik Planlama Birimi, sınıflarda veya laboratuvarlar gerçekleşecek derslerin excel dökümanını sisteme entegre ederek öğrencilerin görmesini sağlayacaktır. Güvenlikler, öğrenciler tarafından yapılmış olan rezervasyonları, excel dökümanı olarak alabilecekler ve bu sayede hem güvenlik açısından laboratuvarlarda ve sınıflarda kimlerin olduğunu bilecekler hem de acil durumlarda hangi sınıflarda ve laboratuvarlarda öğrenci olduğunu görebilecekler.

Proje Kodu ve Adı:	ProcessX: Improved Search Functionality with Image Processing for E-Commerce Systems
Proje Takımı:	Okan Çokkeçeci, Orhan Bartu Kutlu, Ahmet Can Poyraz
Proje Danışmanları:	Prof. Dr. Mehmet Ufuk Çağlayan
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Raporu 2. Gereksinimler Teknik Özellikler Belgesi(RSD) 3. Tasarım Özellikleri Belgesi(DSD) 4. Ürün Kılavuzu(PM) 5. iOS ve Android Uygulamaları 6. Proje Poster 7. Proje Sunum Slaytı 8. Proje Özeti 9. Proje Sunum Videosu
Proje Web Adresi:	-
Proje Özeti	
<p>1. Giriş</p> <p>Görüntü İşleme ve Yapay Us, günümüz yazılımları için çok etkili iki alandır ve hala gelişmeye devam etmektedirler. Henüz Yapay Us, hayal edilen uç noktalarına ulaşamamıştır ancak bilgisayar mühendislerinin katkılarıyla amacına doğru ilerlemektedir. Diğer yanda da geniş bir pazara hitap eden E-Ticaret sistemleri bulunmaktadır. Tüm dünyadan satıcılar, satışlarını arttırmak adına E-Ticaret pazarına da dahil olmaktadır. Şu an dünyada yaklaşık olarak 12 ile 24 milyon arasında E-Ticaret sitesi hizmet vermektedir[1]. Deloitte araştırma şirketi tarafından yapılan “2019 Holiday Survey of Consumers” anketine göre[2], çevrimiçi alışveriş zirvedeki yerini almıştır. Telefon ile alışveriş yapmak her geçen yıl olduğu gibi yükselmeye devam etmektedir. Ankete katılan telefon kullanıcılarının %70’i, telefon ile alışveriş yapmayı planlamaktadır. 5 yıl öncesine bakıldığında bu oran %41 olarak bulunmuştur. ProcessX projesi, yapay us dahilinde görüntü işleme ve E-Ticaret sistemlerinin daha etkin bir şekilde, bir arada kullanılacağı çözüm yaratmakta ve müşterilerin ilgilenebileceği ürünleri kendilerine daha hızlı bir yol ile sunarak E-Ticaret sistemlerinin satışlarını arttırmak adına hizmet vermektedir.</p> <p>2. Gereksinimler</p> <ol style="list-style-type: none"> 1. Kullanıcının iOS ve Android uygulamaları aracılığıyla fotoğraf yükleyebilmesi. 2. Web servisin kullanıcı girişi, kullanıcı kaydı, yeni ürün yaratılması, kullanıcı profilinin getirilmesi, tüm ürünlerin getirilmesi, kullanıcının ilgileneceği ürünlerin getirilmesi adına yapılan istekleri başarılı bir şekilde iletmesi. 3. Sunucunun iletilmiş fotoğraf üzerinden yaş ve cinsiyet tahminleri üretmesi. 4. iOS ve Android cihazlarında bulunan uygulamaların sunucudan aldığı verileri doğru bir şekilde kullanıcıya göstermesi. 5. Kullanıcının tüm ürünleri, kendine özel ürünleri ve profil sayfasını görüntülemesi. 	

3. Tasarım

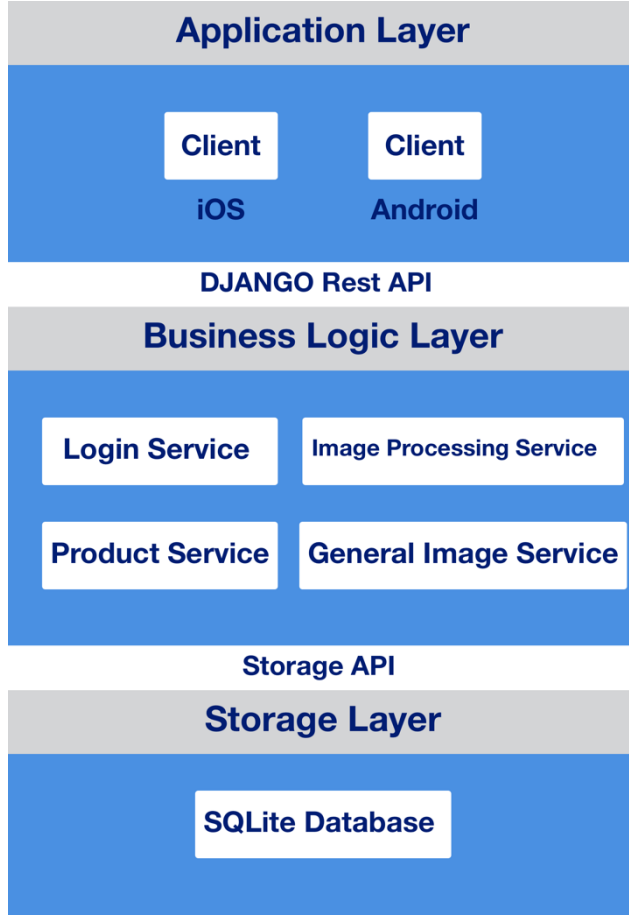


Figure 1. ProcessX Sistem Mimarisi

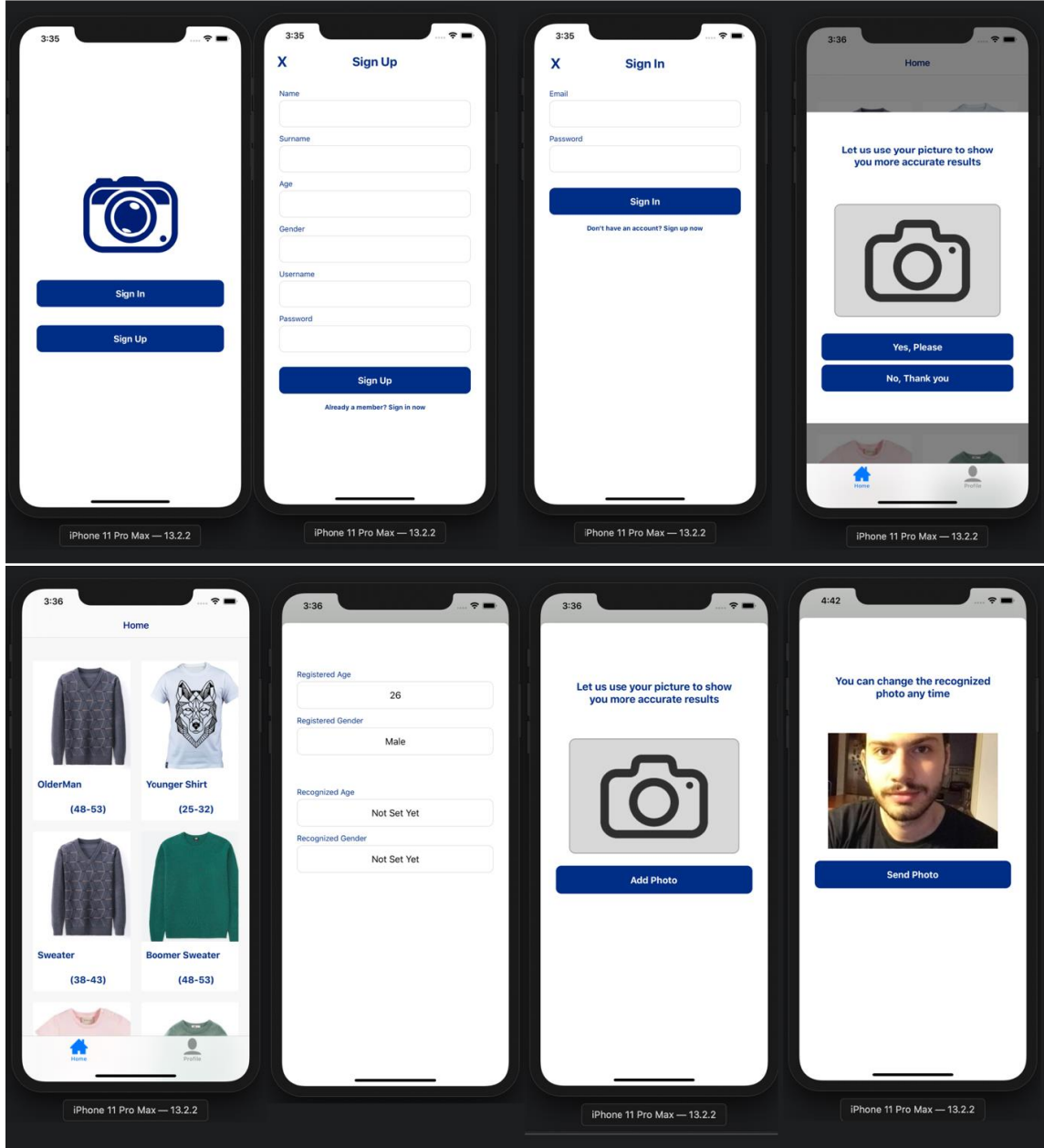
1. KitKat(4.4) veya üst sürümlerini destekleyen tüm Android cihazlar için uygundur.
2. iOS 10 ve üst sürümlerini destekleyen tüm iPhone cihazlar için uygundur.

ProcessX projesi, 25 bin veri kümesiyle eğitilmiş bir model kullanmaktadır[3], sebebi ise projenin amacı sıfırdan görüntü işleme sistemi yapmak değildir. Projenin geliştirilmesine ayrılan süre ve Covid-19 pandemisinin getirdiği organizasyon kısıtlamaları dolayısı ile düşük verili bir model kullanmak projenin amacına çözüm bulamayacaktır. ProcessX, bir sunucuda barınmaktadır. RESTful mimarisinde yazılmış API kullanılarak iOS ve Android platformlarında hizmet vermek üzere geliştirilmiş ProcessX mobil uygulamaları ile iletişime geçmektedir. Mobil uygulamalar bir E-Ticaret modeli örneğidir ProcessX her hangi bir E-Ticaret sistemi içerisine de entegre olabilmektedir.

ProcessX, müşterinin yüklediği fotoğraftan yaşını ve cinsiyetini tahmin ederek kendisinin ilgilenebileceği ürünler sunmaktadır. Öncelikle, web servis aracılığı ile mobil uygulamalarından aldığı verileri sunucusunda işleyip müşteri hakkında tahminler üretmektedir ve elde edilen sonuçlar dahilinde müşteriye ilgilenebileceği ürünler sunmaktadır.

İmplementasyona geldinğinde, öncelikle ProcessX, OpenCV Framework yardımıyla ve önceden eğitilmiş 25 bin veri kümesine sahip modelin desteği ile, bir konsol uygulaması olarak geliştirildi. Sistem gereksinimlerine sahip bir sunucuya taşındı. Ardından E-Ticaret sistemleri ile bağlantı sağlamak adına geliştirilmiş web servisi kullanılarak mobil E-Ticaret platformlarına entegre edilebilecek potansiyele ulaşmıştır. Sonrasında, iOS ve Android E-Ticaret modelleri oluşturuldu ve ProcessX ile iletişimi sağlandı.

3.1 Kullanıcı Arayüzü



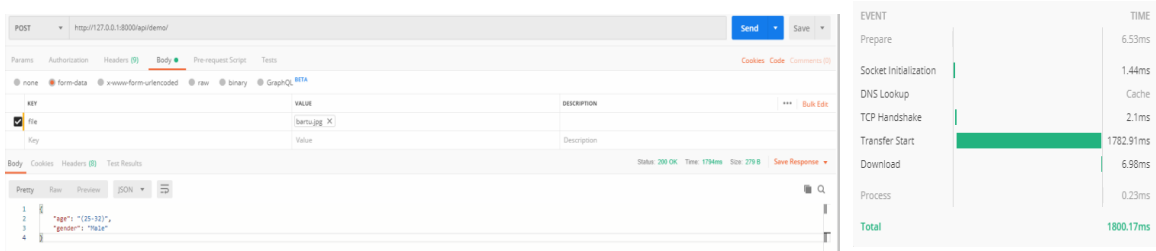
Figur 2. Kullanıcı Arayüzü

4. Uygulama ve Testler

Gender	Total Test	Success	Fail	Success Ratio
Male	25	21	4	84%
Female	27	11	16	40,7%

Tablo 1. Cinsiyet Test Sonuçları

Test aşamasında ProcessX, 25 erkek model ile %84'lük başarılı sonuç elde etmiştir. Fakat 27 kadın modeli ile %40 başarı elde etmiştir. Bunun sebebi, eğitilen yapay us modelindeki test kümesindeki bir takım yetersizliklerden kaynaklandığı gözlemlenmiştir.



Figür 3. Postman yazılımı aracılığıyla yapılan API Testi

Sorgu adresi, başarılı bir şekilde yaş ve cinsiyeti JSON olarak döndürmektedir.

5. Sonuç

5.1 Projenin Faydaları

ProcessX projesi, E-Ticaret sistemleri aracılığıyla alışveriş yapan müşterilere yardımcı olmanın yanında barındırdığı inovatif fonksiyonallite ile ilgilerini çekebilme potansiyeline sahiptir. Aynı zamanda başarılı olması sonucunda E-Ticaret sistemi sahiplerinin kârına da katkı sağlamaktadır.

5.2 Projenin Değerlendirmesi

Sonuç olarak E-Ticaret ve Yapay Usun bir arada büyük bir potansiyeli vardır. Bu potansiyele ulaşılmasında, ProcessX'in büyük bir katkısı olacaktır. Şu an kullanılan model ile de başarılı sonuçlar veren ProcessX daha büyük bir veri kümesi kullanılarak eğitildiğinde, düşük hata payıyla E-Ticaret'i bir sonraki seviyeye taşımaya hazırdır.

5.3 Gelecekte Yapılabilecekler

ProcessX, video üzerinden de müşterinin yaş ve cinsiyetini bulmayı hedeflemektedir. Aynı zamanda, müşterinin iletği fotoğraftan giydiği kıyafetin belirli özelliklerinin de algılanması amaçlanmaktadır. Bu sayede, yaş ve cinsiyetin yanında kıyafet tercihi gibi güçlü bir veri de kullanılarak ProcessX'in ürettiği tahminlerin güçlendirilmesi amaçlanmaktadır. Son olarak da, ProcessX doğası gereği hassas veriler kullanılmaktadır. Fotoğraf paylaşmanın özel hak olmasından dolayı müşterinin uygulamaya duyduğu güveni arttırmak için projenin güvenlik protokollerinin güçlendirmesi amaçlanmaktadır.

6. Referanslar

1. WPForms, LLC, 2019, accessed 23 December 2019 < <https://wpforms.com/ecommerce-statistics/> >
2. Deloitte Development LLC, 2019, accessed 20 May 2020 <https://www2.deloitte.com/content/dam/insights/us/articles/6382_2019-holiday-survey/DEL_Holiday19_ConsumerSurveyFindings.pdf?nc=1>
3. Gil Levi and Tal Hassner. *Age and Gender Classification Using Convolutional Neural Networks*. IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, 2015

Proje Kodu ve Adı:	EKG-a: A wireless Arduino ECG System
Proje Takımı:	Atakan Kayabülbül , İbrahim Fırat Çilek, Enes Eren
Proje Danışmanları:	Prof. Dr. Mehmet Ufuk Çağlayan
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. Requirements Specifications Document 3. Design Specifications Document 4. Product Manual 5. Product-Hardware Subsystem 6. Product-Software Subsystem
Proje Web Adresi:	https://ekg-a-2020.wixsite.com/ekg-a

Proje Özeti

1. Giriş

Projemiz, günümüzde bulunan ağır, kullanımı zor ve sadece hastanelerde bulunan ekg makinalarını bizim ufak, hafif, kullanımı kolay ve daha çok köylerde bulunan sağlık ocaklarında uzaktan erişim olanağına imkan sağlamak amacıyla hitap eden EKG-a'lar kullanarak bir alternatif sunmaktır.

Projemiz EKG-a hem donanımsal hem de yazılımsal bölümler içermektedir. Arduino kullanarak geliştirilmeye daha kolay olanak sağlayacak bir sistemimiz olmuştur. Yazılım tarafında C# ortamında programımızın arayüzünü oluşturduk.

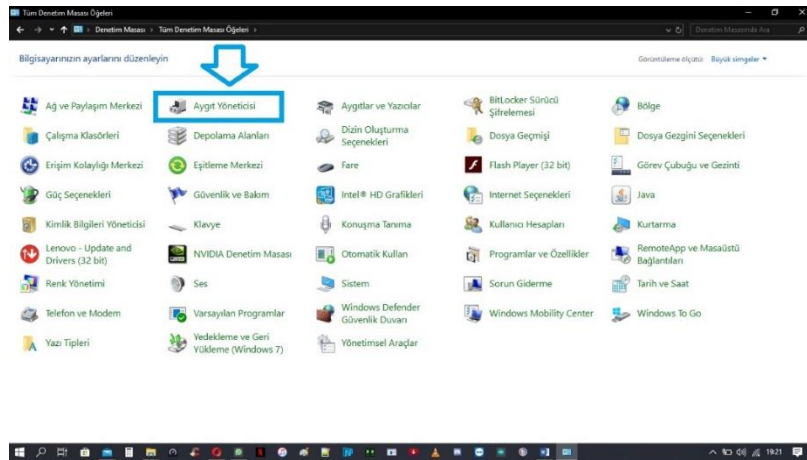
Projemizde 2 arduino nano, 1 kalp atış hızı sensörü, RF Modül ve RF Adaptör ana elementlerdir. EKG-a kullanılırken hastanın sabit ve uzanır bir şekilde yatar pozisyonda olması gerekmektedir. Çok ışıklı ya da gürültülü bir ortam olmaması gerekir. Problar düzgün bağlandığından emin olunduktan sonra test başlatılır. Testin en doğru sonuçlarına 5-10 dk aralığında görmemiz mümkündür.

2. Gereksinimler

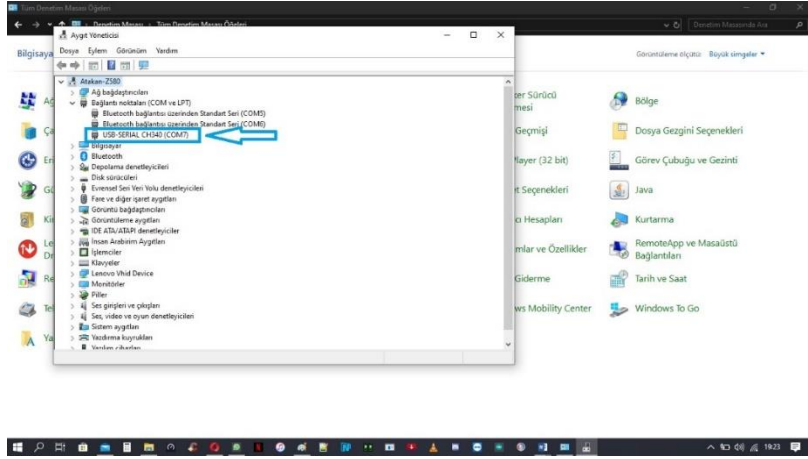
- USB portu olan güncel masaüstü veya dizüstü bilgisayar
- Güç kaynağı veya pil

3. Tasarım

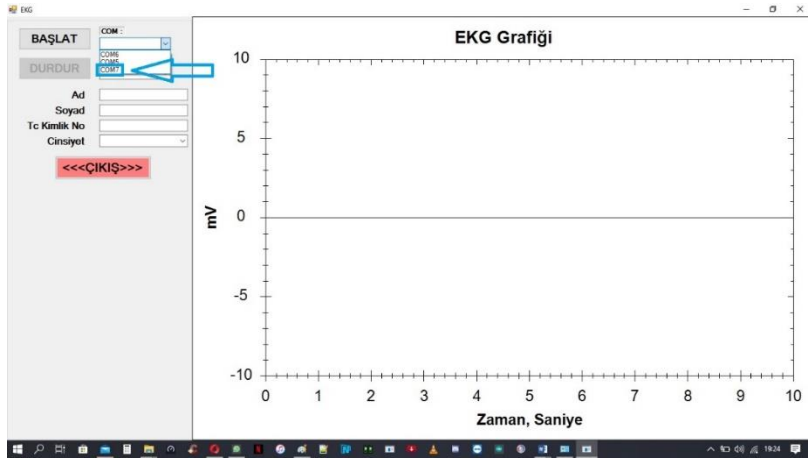
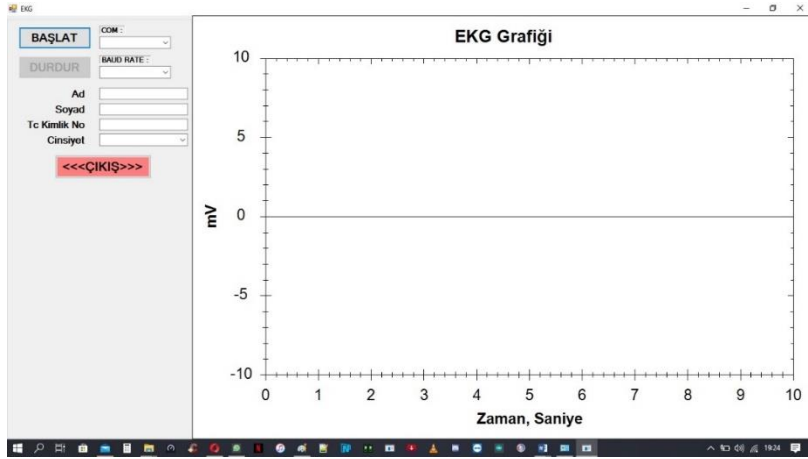
- Masaüstü Arayüzü



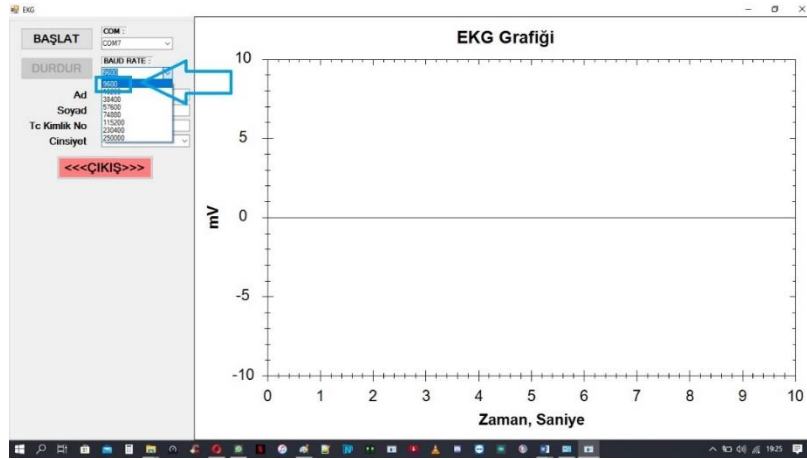
İlk olarak sistemi çalıştırmak için alıcı kutumuzu hangi usb port bölümüne taktığımızı bakıyoruz.



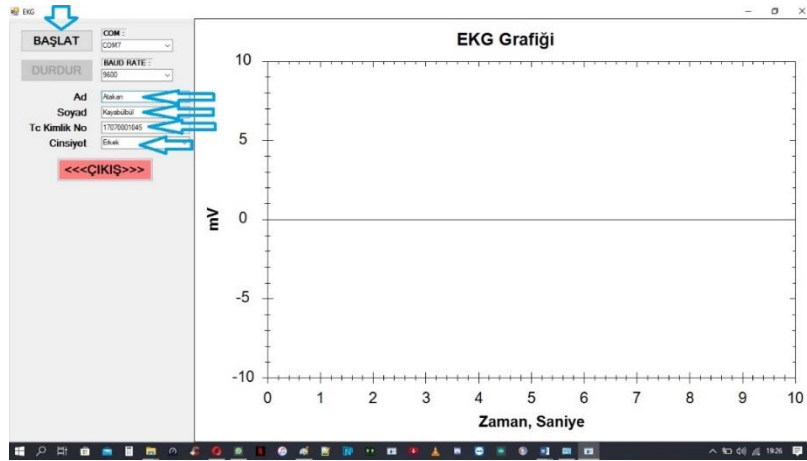
Daha sonra programımızı çalıştırıp bizden istenilen bilgileri giriyoruz.



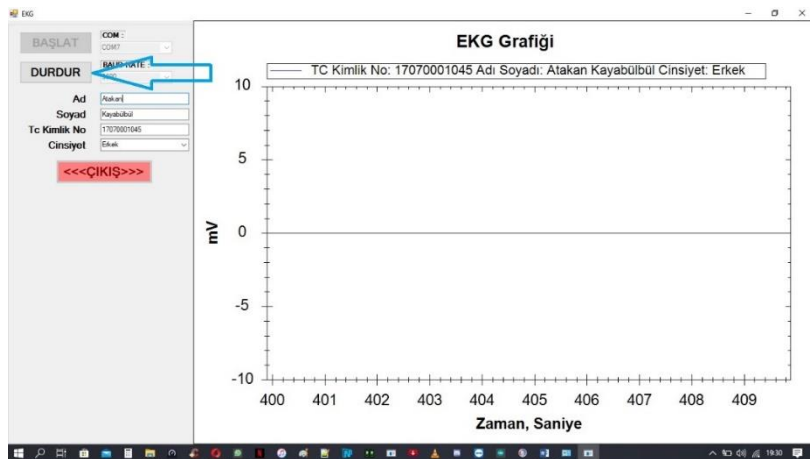
Burada Aygıt Yöneticisinden baktığımız port değerini seçiyoruz.



Baud Rate değerimizi 9600 olarak seçiyoruz.



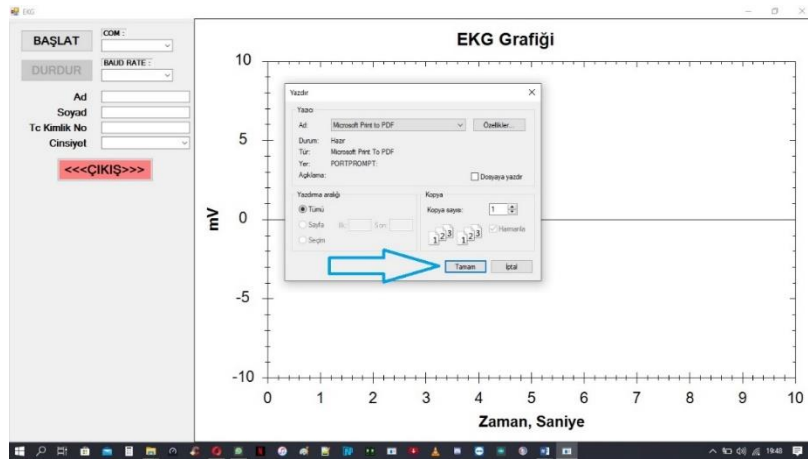
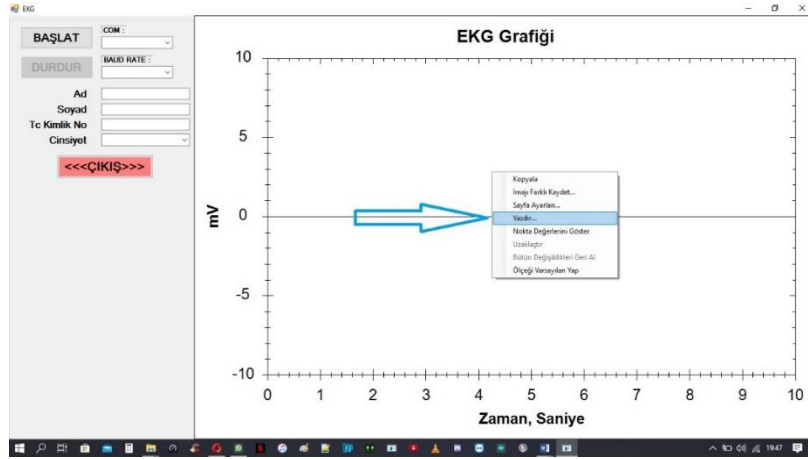
Hastanın Adı, Soyadı, TC Kimlik Numarası bilgilerini girip cinsiyetini seçtikten sonra teste başlayabiliriz.



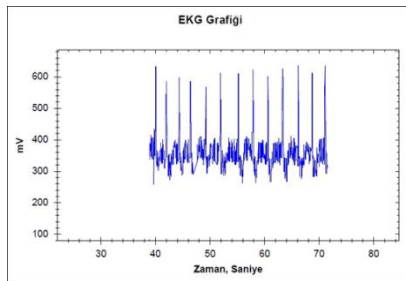
Herhangi bir hata durumunda hasta EKG çekimini iptal etmek için Durdur butonuna basabilir sonrasında ise programdan çıkmak için Çıkış butonuna basabilir.

4. Sonuç

Hasta sonuçları almak için EKG Grafiği ekranına sağ tıklayıp yazdır seçeneğine basmak zorundadır.

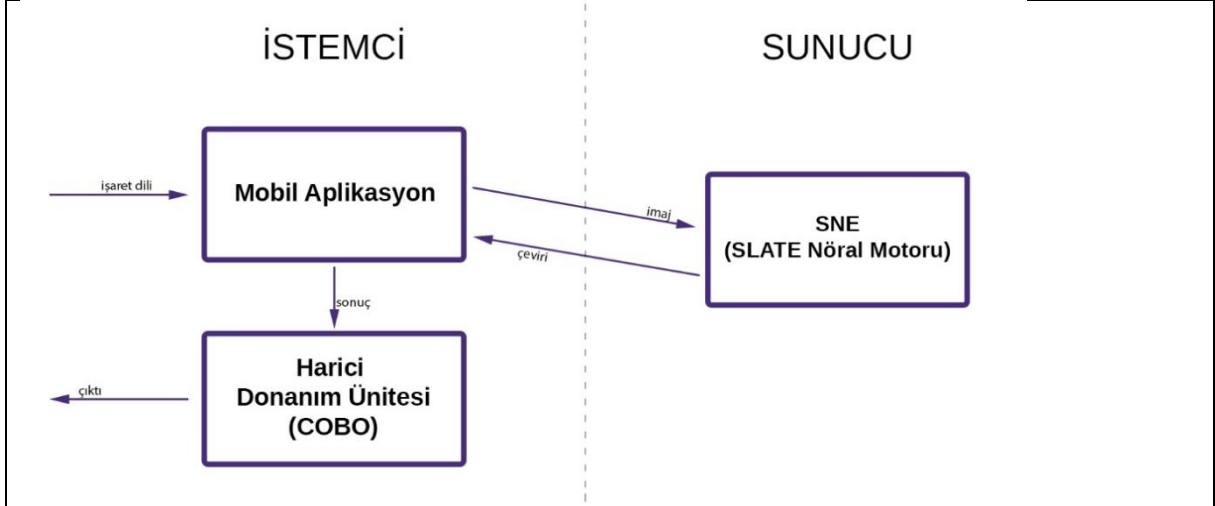


Hasta daha sonrasında Kaydedeceği yeri seçip sonuçlarına PDF ortamında bakabilir, doktoruna iletebilir.



- Sonuç olarak projemiz köylerde bulunan sağlık ocaklarında hastaların, sağlık ocaklarına veya hastanelere gitmeden de bizim geliştirdiğimiz projeyi kullanarak kendilerinin EKG'lerini çekerek test bitiminde doktorların sonuçlara ulaşılması hedeflenmiştir.

Proje Kodu ve Adı:	SLATE: Yapay Zeka ile İşaret Dili-Yazı Çevirisi
Proje Takımı:	Tuna Alaygut , Berkay Bayındır, Alara İşcan
Proje Danışmanları:	Dr. Öğr. Üyesi Mutlu Beyazıt
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. Requirements Specification Document 3. Design Specification Document 4. Product Manual 5. COBO (Harici Donanım Ünitesi) 6. SLATE Nöral Motoru (SLATE Neural Engine) 7. SLATE Web Aplikasyonu 8. SLATE Android Apikasyonu 9. Proje Web Sitesi
Proje Web Adresi:	theslateproject.alaygut.com
Proje Özeti	
<p>Giriş SLATE Projesi, işitme/konuşma engelli kişilerin günlük hayatta karşılaştıkları, işaret dilinin yaygın olmaması ile alakalı, zorluklara çözüm getirme amacıyla kurgulanmış ve gerçekleşmiş bir projedir. Projenin asıl amacı, Türkçe İşaret Dili'ni, yapay zeka (makine öğrenmesi, derin öğrenme), bilgisayarlı görme, mikro denetleyiciler ve web aplikasyon teknolojilerini bir araya getiren bir ürün ile, yazıya dönüştürmektir. Bu ürün, üç farklı bileşenin bir araya gelmesi ile ortaya çıkar. Bu bileşenler işaretleri yazıya çeviren bir nöral motor, kullanıcının direkt olarak etkileştiği bir mobil/web aplikasyonu ve akıllı telefonun arkasına yerleştirilecek, harici ekranlı, bir dış donanım ünitesidir.</p>	
<p>Gereksinimler Projenin gerçekleşmesi için gereken koşullar, ürünü oluşturan herbir bileşen için ayrı ayrı belirlenmiştir. Bu koşullar ve herbir bileşenin, kendi görevine bağlı gereksinimleri yerine getirecek şekilde tasarlanıp gerçekleştirildiğinde, bir bütün olarak karşılamaları gereken gereksinimler şu şekilde özetlenebilir:</p> <ul style="list-style-type: none"> ▪ Akıllı telefonlardan erişilebilen bir uygulama ile devamlı bir, canlı imaj akışı (video) sağlanması ve imajların sunucu tarafında çalışan nöral motora gönderilmesi ▪ Devamlı bir imaj akışından, herbir imajın içindeki ellerin konumlarının, obje tespit algoritmaları ile, belirlenmesi ve ayıklanması ▪ Ayıklanan el imajlarının işaret dilindeki karşılıklarının, imaj sınıflandırma algoritmaları ile belirlenmesi ▪ Yapılan sınıflandırmanın akıllı telefon aplikasyonuna geri gönderilmesi ▪ Akıllı telefonda sınıflandırmanın, Bluetooth teknolojisi ile, donanım ünitesine gönderilmesi ▪ Sonucun donanım ünitesine ait harici ekranda gösterilmesi 	
<p>Tasarım SLATE Projesi, gereksinimleri en verimli şekilde karşılayabilmek ve kullanıcıya akıcı ve bütünleşik bir deneyim sağlayabilmek amacıyla, İstemci-Sunucu mimarisi üzerine tasarlanmıştır. Bu tasarım mimarisi aşağıdaki, Şekil 1 ile özetlenebilir.</p>	



Şekil 1 - SLATE Mimarisi

Bu tasarım mimarisi ürünün genelini kapsamakla birlikte, projenin herbir bileşeni için farklılıklar göstermektedir. Bu farklılıklar doğrultusunda, herbir bileşenin tasarımı şu şekilde özetlenebilir:

Nöral motor, kendi içerisinde üç farklı bileşen barındırmaktadır. Bu bileşenlerin görevleri sırasıyla;

1. Bir imajdaki elleri tespit etmek
2. Tespit edilen ellerin yorumlanması ve yazıya çevrilmesi
3. Çeviri için istemcilerden istek alma ve çevirinin istemcilere gönderilmesi

Web/Mobil aplikasyon ise;

- Akıllı telefonun ön kamerasından bir imaj akışı sağlama ve bu imajları sunucuya gönderme
- Sunucudan cevap alma ve alınan cevabı kullanıcı için görselleştirme
- Alınan cevabı Bluetooth ile harici donanım ünitesine yollama görevlerini gerçekleştirecek şekilde tasarlanmıştır.

Harici donanım ünitesi içerisinde barındırdığı

- Arduino mikrodenetleyici
- HM-10 BLE modülü
- TFT harici ekran

bileşenleri ile Bluetooth üzerinden veri alımı ve üretilen sonucun ekrana yansıtılması görevlerini gerçekleştirir.

Uygulama ve Testler

Ürünün herbir bileşeni kendi operasyonunu istenilen bir şekilde gerçekleştirip gerçekleştirmediği konusunda kapsamlı şekilde, çeşitli senaryolar ile test edilmiştir. Ürünün doğası gereği her bir bileşenin testleri tamamlandıktan sonra ürünün genel operasyonu da test edilmiştir.

Sonuçlar bölümünde anlatılan imkansızlıklar, kusurlar ve varsayımlar dışında herbir bileşenin kendi operasyonunu başarıyla gerçekleştirdiği ve son ürünün de istendiği gibi çalıştığı gözlemlenmiştir.

Sonuçlar

SLATE Projesi sonunda bir kavram ispatı olarak, işaret dili çevirisinin yeterli veri ve yeterli işlem gücü ile yapılabileceği gösterilmiştir.

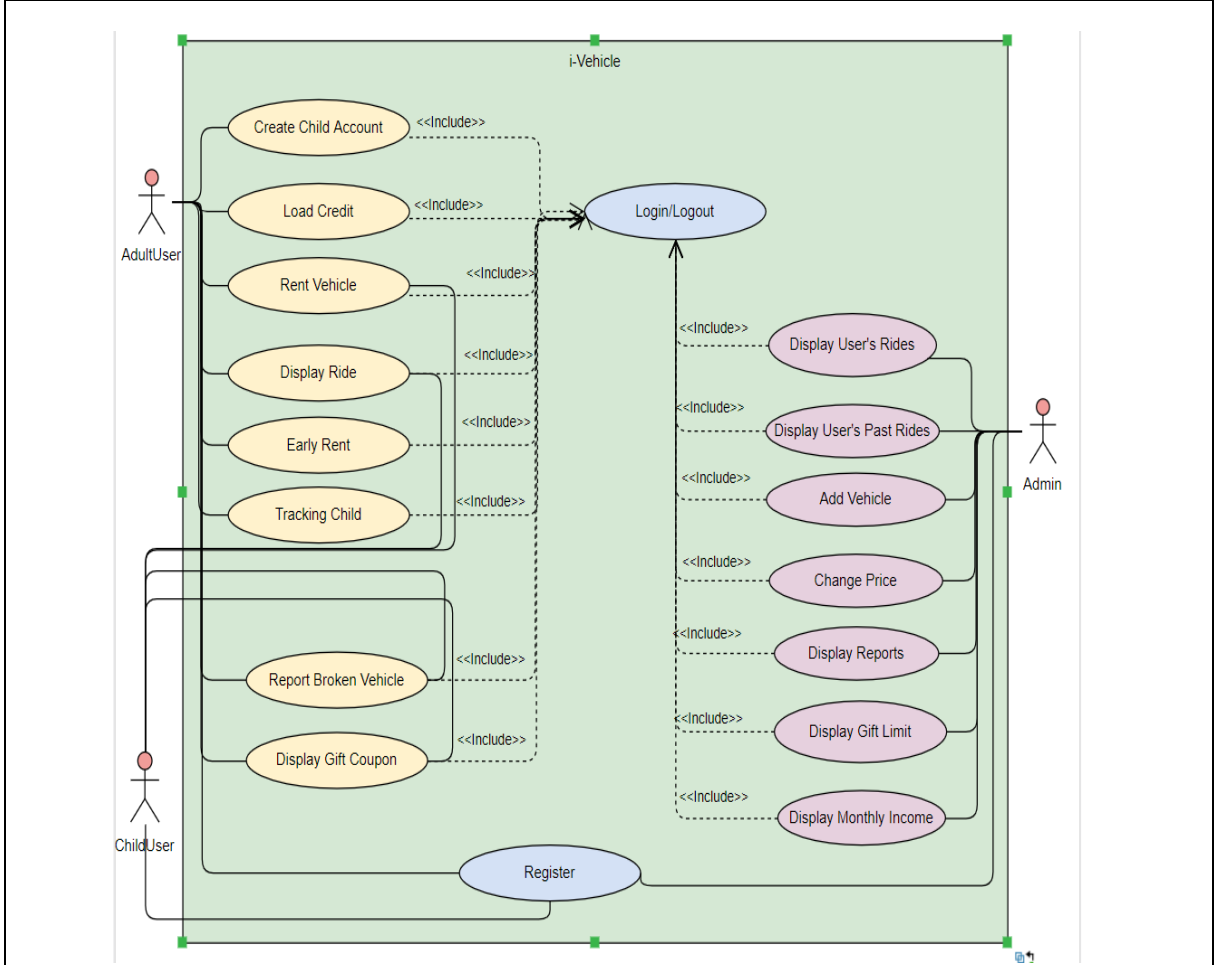
SLATE Projesi aşağıdaki faydaları da beraberinde getirmiştir:

- İşitme/konuşma engelli insanların hayatlarını kolaylaştırma noktasında bir çözüm önerisi ortaya koyulmuştur.
- Sadece proje dahilinde geliştirilen mobil/web uygulamaları değil, herhangi bir partinin de SLATE nöral motordan faydalanarak çeşitli çözümlerin geliştirilmesine olanak sunulmuştur.
- SLATE Projesi ek veriler ile yeni nöral ağların eğitilmesine de olanak sunar. Bu, işaret dilinin bütünüyle (tüm türevleriyle birlikte) çevrilmesine olanak sağlayan bir sistemin kurulmasına önayak olma potansiyeli taşımaktadır.

SLATE Projesinin gerçek potansiyeline ulaşması için öngörülen adımlar şu şekildedir:

- Daha fazla veriye erişim ile daha isabetli nöral ağların eğitilmesi.
- Sunucu tarafında daha güçlü donanım sistemleri ile cevap zamanının kısaltılması.
- Dış donanım ünitesinin prototip aşamasından öteye taşınması.
- Android/iOS işletim sistemleri için native uygulamalar geliştirilmesi.

Proje Kodu ve Adı:	i-Vehicle: Motorless Vehicle Rental System
Proje Takımı:	Baran Göksel Göcen, Gamze Aydemir, İrem Ergüler
Proje Danışmanları:	Dr. Öğr. Üyesi Mutlu BEYAZIT
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. Requirements Specifications Document 3. Design Specifications Document 4. Product Manual 5. Product-Software Subsystem 6. Web Poster
Proje Web Adresi:	
Proje Özeti	
<p>Bisiklet ve skuter kiralama sistemleri büyük şehirlerde çok revaçta olmakla birlikte bazı eksikleri bulunmaktadır. Biz de bu sistemlerin birer kullanıcısı olarak eksik gördüğümüz ve geliştirilebilecek olan kısımları geliştirip yepyeni bir sistem oluşturmayı hedefledik.</p> <p>Projemizde, bisiklet ve skuter sistemleri tek bir sistemde toplanmakta ve eksik özelliklerin kullanıcılara eklenmesinin daha kullanışlı, kolay ve eğlenceli olması hedeflenmektedir. İçinde bulunduğumuz çağın sağladığı teknolojileri kullanarak, prototip oluşturmak için sistemde kullanıcılar için bir Android Uygulaması ve yönetici için bir Web Uygulaması oluşturacak ve hem kullanıcılar hem de yöneticiler için birçok kolaylık sağlayacaktır.</p> <p>Anahtar Kelimeler: MVC Mimarisi, Rest Api, Retrofit, QR kod.</p>	
Proje Amacı	
<p>Bu sistemin yararlarından bazıları; kullanıcıları spor yapmaya ve sağlıklı bir yaşama teşvik etmek için kiralama sistemini kolay ve eğlenceli hale getirmek, çocuk kullanıcı özelliği ile ebeveynlerin çocukları hakkındaki kaygılarını azaltmak ve teknolojik araçlardan uzak, çocuklar için sağlıklı zaman sağlar ve arkadaşlarıyla sosyal ve kaliteli zaman geçirmesini sağlamak. i-Vehicle sadece eğlence için değil aynı zamanda gerçek ulaşım aracı olarak kullanılabilir. Buna göre i-Vehicle, havaya ve çevre temizliğine dikkat ederek küresel ısınmanın önlenmesini desteklemektedir. Gerçek bir ulaşım aracı olarak kullanılmasının bir başka yönü de trafiği azaltmaktır. Özel araçlarla çalışmayı veya okula gitmeyi tercih eden insan sayısı azalacak çünkü i-Vehicle ucuz bir ulaşım yolu olacak. Bu trafiği azaltacak ve trafik kazalarını da nispeten azaltacaktır.</p>	
Yazılım Sistem Ortamı	
<p>Bütün bunlar için 2 ayrı uygulama kullanılmış olup, ortak veritabanı kullanılmıştır. Web Uygulamasında MVC mimarisi kullanılırken, Android Uygulamasında İstemci-Sunucu mimarisi kullanılıp Android Retrofit kütüphanesi ile REST API'ler kullanılmıştır. Web Uygulaması ve web servisler Visual Studio'da C# dilinde yazılmıştır. Admin paneli sayfası tasarımı için HTML, CSS ve JavaScript dilleri kullanılmıştır. Android Uygulaması Android Studio'da Java dilinde yazılmıştır. Kullanıcı ekranları tasarımı için Android Design Support kütüphaneleri kullanılmıştır.</p>	



Şekil 1. Kullanıcılar ve adminin metotlarını gösteren Use Case Diyagramı

Sonuçlar

i-Vehicle Android Uygulaması, kullanıcılar için hazırlanmıştır. Kullanıcılar, e-mail ve şifresi ile uygulamaya giriş sağlayabilirler. Sürüş başlatmak için, QR kod okuma sistemi veya istasyona gitmeden aracı ayırtmak için rezervasyon yöntemi kullanılır. Kullanıcı sürüşü başlattıktan sonra geçen süreyi sürüş sayfasında görüntüleyebilir. Ayrıca isterse, 1 ya da 2 saatlik geri sayaç ayarlayıp bitmesine 3 dakika kala bildirim gelmesini sağlayabilir. Bu özellik, bir kaç dakika için gereksiz para alınmasını önlemek için tasarlanmıştır. Eğer kullanıcının aldığı araçta bir bozukluk varsa bunu raporlayabilir. Raporlanan araç admin tarafından kilitlenecektir. Kullanıcı sürüşünü bitirdikten sonra sürüş bilgilerini görüntüleyebilir. Başlama ve bitirme zamanı, sürüş ücreti ve geçen süre ekranda görüntülenir. Kullanıcı, sürüşlerinden hediye kredi kazanabilmektedir. Her 100 dakikalık sürüş için 10 kredi hediye edilmektedir. Hediye edilen kredi istenilen zamanda toplam krediye eklenebilir. Kullanıcı kredi yükleme işlemini system üzerinden gerçekleştirebilmektedir. En önemli özelliklerden biri olan çocuk hesabı oluşturmadır. Kullanıcı bir çocuk hesabı oluşturur ve artık kendisi bir ebeveynidir. Ebeveyn, çocuğunun sürüşünü harita üzerinden görüntüleyebilir ve çocuğu için kredi yükleyebilir. Bu özellikler çocuk için kapalıdır. Ayrıca kullanıcılar kullanıcı adı ve şifrelerini değiştirebilirler.

i-Vehicle Web Uygulaması, adminler için hazırlanmıştır. Admin, e-mail ve şifresi ile uygulamaya giriş sağlayabilir. Admin, kullanıcıların sürüşlerini ve sürüş bilgilerini görüntüleyebilir. Kullanılan araçlar da admin tarafından görüntülenebilmektedir. Aracın aktifliği, bozuk olup olmadığı ve toplam kaç kere raporlandığı araçlar sayfasındadır. Ayrıca bozuk olarak rapor edilen aracı pasif olarak ayarlar. Admin yeni araç ekleyebilme özelliğine de sahiptir. Yeni aracın tipini seçerek (bisiklet, skuter) dakika başı ücretini de kendi verebilir. Admin, kullanıcıların sürüşlerinden kazanılan kredileri aylık olarak görüntüleyebilir.

Gelecekteki İşler

Kullanıcılar arkadaş ekleyip onlarla mesajlaşıp, sürüş yaptığı rotayı görüntüleyebilir. Kullanıcı kendisine rota oluşturarak o rota üzerinde sürüşünü gerçekleştirebilir. Kullanıcı uygulamayı Türkçe dışında bir dilde kullanabilir. Daha güvenli olması açısından uygulamaya kayıt olurken yaşı kanıtlamak için kimlik kartı onayı gerekebilir

Kaynakça

1. https://www.visual-paradigm.com/solution/usecase/use-case-tool/?gclid=CjwKCAiAluLvBRASEiwAAbX3GX75SbhJVrgtAdkNQAFS77g-n8O5xiMck0m7cEeiRhzFi6FrrVTNtRoCNoQQAuD_BwE
2. <http://is-sanalisti.blogspot.com/2013/07/use-case-analizi-1-use-case-diyagramlar.html>
3. https://www.visual-paradigm.com/support/documents/vpuserguide/94/158/84414_recordanddoc.html
4. <https://www.journaldev.com/9193/web-services-interview-questions-soap-restful>
5. <https://square.github.io/retrofit/>
6. <https://guides.codepath.com/android/consuming-apis-with-retrofit>
7. [https://en.wikibooks.org/wiki/A-level_Computing/AQA/Paper_2/Fundamentals_of_computer_systems/Role_of_an_operating_system_\(OS\)](https://en.wikibooks.org/wiki/A-level_Computing/AQA/Paper_2/Fundamentals_of_computer_systems/Role_of_an_operating_system_(OS))
8. <https://www.journaldev.com/9193/web-services-interview-questions-soap-restful>
9. <https://medium.com/@socraticsol/why-mvc-architecture-e833e28e0c76>
10. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

Proje Kodu ve Adı:	REM-LIFE
Proje Takımı:	Damla Yağmur, İpek Önbiçak, Şölen Gül
Proje Danışmanları:	Dr. Öğr. Üyesi Mutlu Beyazıt
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. Requirements Specifications Document 3. Design Specifications Document 4. Product Manual 5. Product-Software Subsystem
Proje Web Adresi:	https://solgul1996.wixsite.com/sitem

Proje Özeti

1.Giriş

İnsanların şu dönemde çok sık unutkanlık yaşadığı görülmektedir. Bu unutkanlıklar gerek içinde buldukları ortam gerek psikolojik, ekonomik ve sosyal nedenlerden kaynaklanmaktadır. Bu unutkanlıklar kimi zaman önemli kimi zaman da önemsiz şeyler hakkında olmaktadır. Gelişen teknoloji ile birlikte hayatlarımız daha çok sanal ortama doğru yönelmekte. Bu sanal ortama yönelme sırasında telefonlar ilk sırada yer almaktadır. Günümüzde dünyadaki insanların %67'si telefon kullanmaktadır. Bu yüzden büyük bir sorun olan unutkanlığı çözmek için sanal ortamda ilk sırada yer alan telefonlar için bir uygulama yapmak istedik. REM-LIFE... Oluşturduğumuz uygulamanın insanlar için unutkanlığı ortadan kaldırmasını hedefliyoruz. Uygulama sayesinde kişiler istedikleri zamanda ve konumda onlar için önemli olan notları hatırlayabilecek. Kullanıcılar uygulamadan istedikleri notları tarih, açıklama ve konum bilgileriyle birlikte kaydebilecekler.

Bu uygulamayı geliştirmekteki en büyük amacımız insanların unutkanlıkları yüzünden hayatlarında zor durumlarda kalmamaları ve olumsuz etkilenmemeleri. Kullanıcılar uygulamadan istediği notları listesine ekleyebilir ve kaydettiği notlara özel bildirimler alabilir.

Tüm bunlara ek olarak REM-LIFE uygulamasının kullanıcılarına sunduğu bazı özellikler,

- Uygulamaya kayıt olmaları için bir kayıt ekranı
- Uygulamayı rahatça kullanmaya başlamaları için bir giriş ekranı
- Notlarını listeleyip, rahatlıkla görüntüleyebilecekleri not listesi
- İstedikleri notları listelerine ekleyebilecekleri bir not ekleme sayfası
- Aradıkları bazı bilgileri bulabilmeleri için arama sayfası
- Giriş yaparken kullandıkları bilgileri görüntüleyebilecekleri profil sayfası

Bu sayfaların görüntüleri sırasıyla aşağıdaki gibidir.

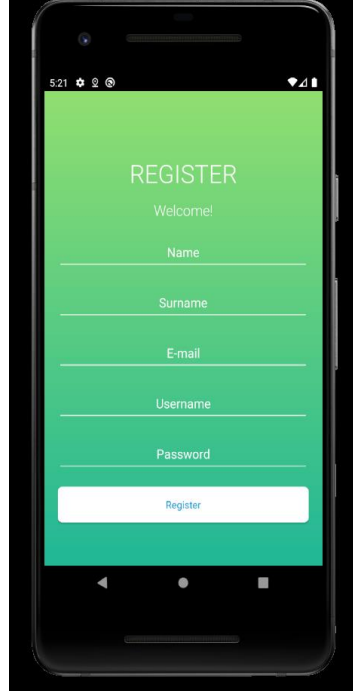


Fig 1. Kayıt Sayfası

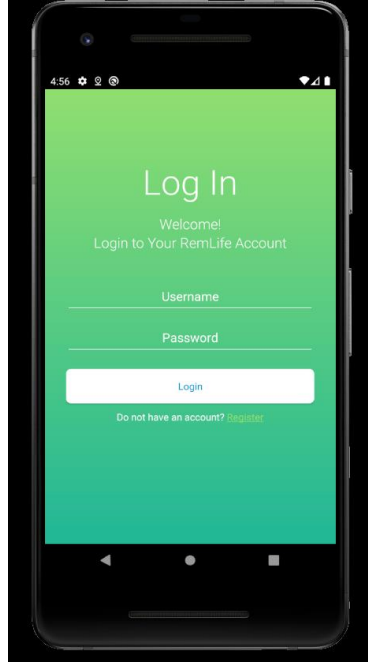


Fig. 2 Giriş Sayfası



Fig 3. Not Listeleme Sayfası

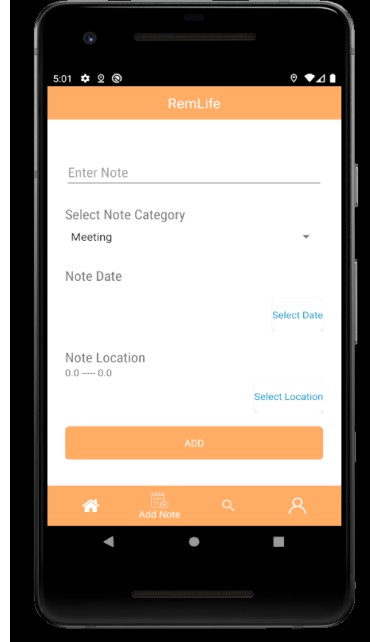


Fig 4. Not Ekleme Ekranı



Fig 5. Arama Ekranı



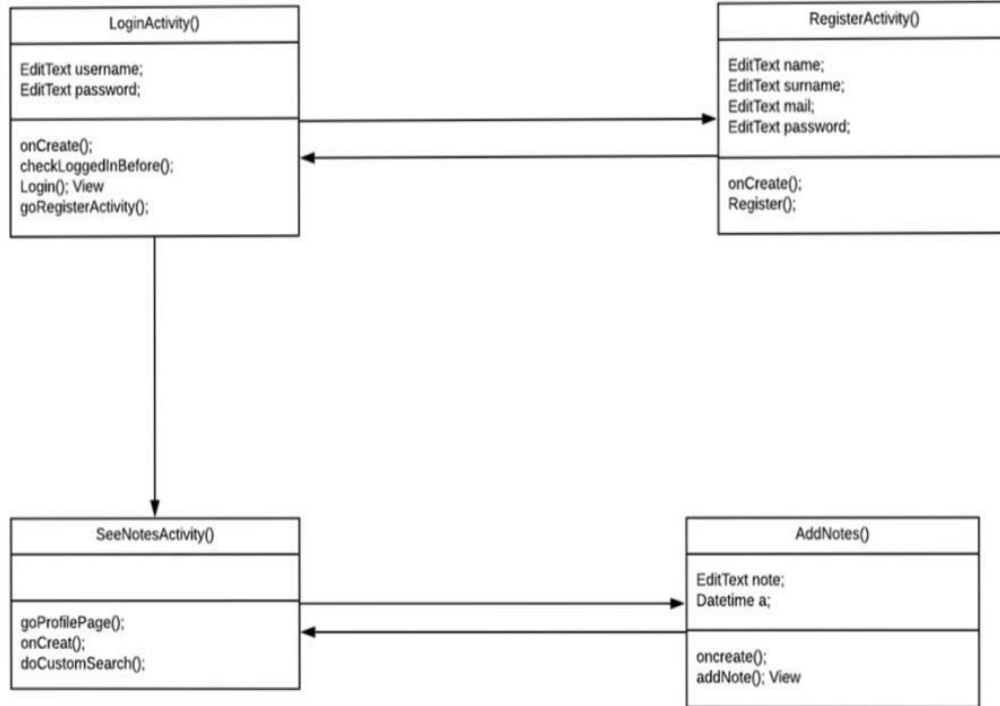
Fig 6. Profil Ekranı

2. Gereksinimler

Öncelikle kullanıcıların, REM-LIFE bir Android uygulaması olduğu için Android işletim sistemine sahip bir telefona ihtiyacı var. Bu özelliği kullanan telefonlarından uygulamayı rahatlıkla kullanabilirler. Uygulamaya not kaydedebilmek için sisteme kayıt olup giriş yapmaları gerekir. Kullanıcı eklemek istediği notlara konum da ekleyebildiği için uygulama konumunuza erişmek ister ve bunu yapmak için internete gereksinim duyar.

3. Tasarım

REM-LIFE arayüzünün kullanıcıları çok yormaması ve zorlamaması adına basit tasarımlar içermektedir. Sistemsel tasarım olarak anlatmak gerekirse kullanıcıların bilgileri veri tabanı yoluyla tutulur. Kullanıcı uygulamaya kayıt olmak, giriş yapmak, notlarını listelemek kısacası kendine özel olan tüm işlemler için bu veri tabanında bulunan verileri kullanır. Kullanıcının uygulamadan yapacağı işlemler için veri tabanına ihtiyaç duyulursa istemci- sunucu mimarisi sayesinde veri tabanından gerekli bilgiler alınır ya da yapılan işleme göre eklenir. Bu istemci sunucu mimarisi veritabanı ve kullanıcı telefonu arasında iletişim sağlamak için kullanılır. Bu iletişimi sağlamak için uygulama web servislerini kullanır. Kullanıcılar için sistemde çeşitli kategorilerde haberler bulunmaktadır. Eğlence, su ve elektrik kesintisi vb. Kullanıcı bu haberler arasında arama yapabilir. Bu haberler sisteme direkt olarak veritabanına etki ederek eklenmez. Uygulamanın giriş sayfasından sistem admini giriş yapar ve bu sayede veritabanına yeni haberler ekler. Bu eklenen haberler de kullanıcı tarafından görüntülenebilir.



Şekil 1. REM-LIFE Sınıf İlişkileri

4.Uygulama ve Testler

Gereksinimlerde de (2. bölüm) belirtildiği üzere REM-LIFE bir Android uygulamasıdır. Android telefonlardan rahatlıkla kullanılabilir. Uygulamanın ara yüzü geliştirilirken Android Studio, veritabanı ve sunucu bağlantıları için Retrofit kullanılmıştır. Uygulamayı kullanan kullanıcıların herhangi bir hata ile karşılaşmaması için uygulama geliştirme sürecinde gerekli testler yapılmıştır. Bu testlere örnek olarak uygulamanın geliştirildiği ortamın hata ayıklama bölümü verilebilir. Uygulamada meydana gelebilecek hataları bulabilmek için kodda bazı bölümlerde log kullanılmıştır. Loglar sayesinde uygulama çalışırken oluşan hatalar anında görüntülenip çözülmüştür. Kullanıcıların yaptığı bazı işlemler sunucuya bağlanmayı gerektirdiği için o kısımda da hatalar oluşmaması için gerekli testler yapılmıştır.

5.Sonuçlar

REM-LIFE uygulamasını geliştirmek için insanların hayatlarındaki önemli bir problemi seçmiştik unutkanlık. Geliştirdiğimiz uygulama ile bunu en az indirdiğimizi düşünüyoruz. Özetle kullanıcılar sisteme rahatlıkla kayıt olup, giriş yapabilirler. Daha sonra unutmak istemedikleri notları zorlanmadan kendi not listelerine kayıt edebilirler. Notlarına tarih, kategori ve konum bilgisi ekleyerek unutkanlık sorunlarını çözebilirler. Not ile birlikte eklenen konum sayesinde bildirim alırlar. Bildirimler şu şekilde çalışır, eğer kullanıcının konumu, herhangi bir notuna eklediği konumun yakınında olursa kişi not hatırlatması olarak bildirim alır. Tüm bunlara ek olarak sistemde çeşitli kategorilerde haberler bulunmaktadır. Kullanıcı bu haberler arasında arama yapabilir.

Kaynakça

[1] Android (2020). Recommendations and latest technologies used for Android developers. Retrieved from <https://developer.android.com/>

Proje Kodu ve Adı:	POF: Performance Optimized Fluids
Proje Takımı:	Baran Budak, Cihanser Çalışkan, İsmail Mekan,
Proje Danışmanları:	Prof. Dr. Mehmet Ufuk Çağlayan
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. Requirements Specifications Document 3. Design Specifications Document 4. Product Manual 5. POF software 6. Poster
Proje Web Adresi:	-

Proje Özeti

1. Giriş

Son yıllarda artan fizik tabanlı ve görselleştirme içeren simülasyonların, bilimsel araştırmalar dışında kullanımı artmış ve bu sistemler dış kaynak olarak oyun ve projeler gibi farklı yerlerde kullanılmaya başlanmıştır. Günümüzde bu sistemlerden biri olan kumaş giysi, gibi dış faktörlere göre şekli değişen objelerin NvidiaPhysics kullanılarak bazı oyunlara eklenip, gün geçtikçe daha fazla sistemde kullanılmaktadır. Bu talepten doğan sonuçla projemiz olarak, parçacık tabanlı sıvı simülasyonlarını bu konuda kaynak ve zaman ayıran büyük şirketler yerine, daha az kaynakla kendi projelerini yapan insanları baz alarak bu ve benzeri simülasyonun daha küçük çaplı projelerde kullanılabilirliğini kanıtlama ve ilgilenmek isteyen insanlar için teşvik örneği olmak istedik. Unity ortamında Nvidianın ürettiği parçacık tabanlı sıvı simülasyonunu entegre edip, bu simülasyonlarda kullanılan yüzey tanımlama algoritmalarından birini uyguladık. Bu bağlamda POF projesi ortaya çıktı.

POF projesi parçacık tabanlı sıvı simülasyonlarında bulunan sıvıların yüzeylerini bulunmasını sağlayıp, bu konuda deneyimli kişilerin algoritmaları uygulayıp, test edip, verilerle birlikte karşılaştırabileceği bir sistemdir. Bu bağlamda bilgisayar grafikleri konusuna veya spesifik olarak parçacık bazlı sıvı simülasyonu konusuna ilgi duyan insanlar için bir platformdur. Kullanım açısından daha kolay olan Unity Platformu tercih edilip, bu tercihle yapılan testlerin sonucu hem görsel hem de veri olarak çıktı alınabilirliği sağlanmıştır. Simülasyon genel olarak, sıvı simülasyonunun entegre edilmesi, simülasyondaki parçacıklara daha hızlı ve stabil ulaşımın sağlanması için hash algoritması uygulanması ve yüzey parçacıklarının bulunması adı altında üç işlemi gerçekleştirip bunların sonucunda sıvıların yüzeylerinin ve test sonuçlarının gösterildiği bir görsel arayüze sahiptir.

2. Gereksinimler

2.1 Proje Gereksinimleri

- Nvidia Flex yapısını Unity ile entegre bir şekilde çalıştırmak.
- Parçacıkların verilerine ulaşmak.
- Parçacıkların bulunduğu hacmi daha küçük küplere bölüp hücre kullanarak Karma (Hash) sistemi oluşturmak.
- Yüzey parçacıklarını tanımlayabilmek.
- Verimliliği arttırmak.
- Kullanılabilirliği arttırmak.
- Test edilebilir bir platform ortamı yaratmak.

2.2 POF sisteminin çalışması gereken sistem gereksinimleri

D3D11 kapasiteye sahip grafik kartı.
NVIDIA: GeForce Game Ready Driver 372.90 veya üstü.
AMD: Radeon Yazılım Versiyon 16.9.1 veya üstü.
Microsoft Visual Studio 2013 veya üstü.
G++ 4.6.3 veya üstü
CUDA 8.0.44 veya üstü
DirectX 11/12 SDK
Windows 7 (64-bit) veya üstü.
Unity 3D 2017.3 versiyon veya üstü.

3. Tasarım

POF sistemini geliştirirken var olan problemlere dayalı olacak şekilde, algoritmalara sahip bir yapıya ayrılmıştır. Bu alt yapıların birbirleriyle iletişimi basitçe:

- Hash algoritması var olan verileri haritalandırır.
- Yüzey tanımlama algoritması bu haritalanan verileri kullanarak, yüzey parçacıklarının belirlenmesi sürecini yönetir.
- Görsel çıktı bölümünde yukarıdaki işlemle yaptığımız yüzey parçacıklarının datasını ve işleme süreçlerini görselleştirip, bir test ortamı hazırlar.

Bu altyapıların kendi içinde çalışma şekillerini anlatırsak:

Karma (Hash) algoritması: Bu algoritma parçacıkların 3 boyutlu uzaydaki yerini küplere bölüp bu küplerin içindeki sıvı parçacıklarını haritalandırmamızı sağlar böylece gerçek hayattaki haritalardaki gibi koordinatlandırma gibi 3 boyutlu düzlemde koordinatlar ve bunlara denk gelen sıvı parçacıklarını bir data yapısı üzerinden daha hızlı bulmamızı sağlar.

Yüzey tanımlama algoritması: Projede bulunan 'yüzey' parçacıklarının tespitinde işimize yarar. Algoritma olarak anlatmak gerekirse, basitçe simülasyondaki her bir parçacık ve onlara ait belirli bir uzaklıktaki komşu parçacıklarını kütle merkezine benzer bir algoritmaya sokup projemizde hedeflediğimiz gibi parçacıkların yüzey olup olmadığını belirleyen algoritmadır.

Görsel Çıktı: Yukarıdaki algoritmaların içine yerleştirilen işlemin yapılma sürelerini tutan sayaçlar ve sonuç olarak aldığımız yüzey parçacıklarının Unity tarafında görsel olarak küre çizerek gösterdik. Böylece, yukarıda bahsettiğimiz sistem ve bağlantılar sonucunda ortaya çıkan yüzey parçacıklarının görselleştirilmesi ve bu algoritmaların işleme sürelerini içeren bir ekran oluşturulup böylece süre bazlı testlerin ve algoritmaların doğru çalışıp çalışılmadığını izleyebildiğimiz bir sistem oluşturuldu.

4. Uygulama ve Testler

Belirli bir sürede olabildiğince fazla resim karesi çıkartabilmek için sistemimizin yeterince hızlı çalıştığından emin olmamız gerekiyordu. Parçacık sayısının oldukça fazla olduğu (Milyonlarca parçacık olabilir fakat bizim sistemimiz yetersiz olduğu için 4000 parçacık kullandık.) sistemimizde tüm verileri teker teker kontrol etmeye çalıştığımızda bu kadar datanın doğruluğunu kanıtlayabilmenin en verimli çözümü olan Unity ortamında görselleştirme çözümüne gittik. Öte yandan çalışan sistemin süre bazlı testlerinin yapılması için algoritmaların başlangıç ve bitişlerini hesaplayan ve bunları ekrana yansıtan sayaçlar ekledik.

Algoritma bazlı aldığımız uygulama ve test sonuçları:

4.1 Hash Sistemi

POF sisteminin uygulama aşamasında karşılaştığımız ilk zorluk, parçacık sayısı arttıkça hesaplamaların süresinin uzaması ve bundan dolayı Unity'nin kendisini kapatmasıydı. Sonuç olarak parçacıkları daha hızlı bulabileceğimiz bir alt yapı inşa etme kararı aldık ve parçacıklara daha hızlı erişmek ve genel performansı arttırmak için hash sistemini yarattık.

Hash sistemi parçacığın 3 boyutlu uzaydaki pozisyonunu hash tablosuna aktararak daha hızlı erişmemize yarar. Parçacıklara sırayla lineer olarak bakmak yerine bunları test sonuçlarımıza göre yaklaşık 4 milisaniye gibi hep aynı sürede bulmamıza yarar. Yukarıda anlattığımız gibi Unity'nin performans problemine çözüm olarak parçacıkları Unity sahne ekranında renklendirmeye çalıştık. Seçtiğimiz parçacığı belirli bir renge boyadık ve komşu parçacıklarını başka renge boyadık. Parçacığın bulunduğu ve bizim hesaplamayı kolaylaştırma açısından yarattığımız hücre adını verdiğimiz küçük küplerin de kenarlarını çizdirdik. Sahnedeki resim karesi ilerledikçe, seçtiğimiz parçacığı, komşularını ve ait olduğu hücreyi takip edebildiğimizden emin olduk.

4.2 Yüzey Parçacık Tanımlayıcı

Öncelikle 'yüzey' dediğimiz kavram projemizde farklı bir anlama sahiptir. Yüzey, parçacığın çevresiyle etkileşen en dışarıdaki parçacıklarına denir.

Yüzey parçacık tanımlayıcının görevi yüzey parçacıkları ayırt edebilmektir. Bu alt yapıyı, parçacık bazlı sıvı simülasyonu konusunda yapılan akademik makaleleri inceleyip POF sistemimize uygulayarak inşa etmeyi başardık. Yüzey parçacıklarını doğru tespit edebildiğimizden emin olmak için ise hesapladığımız yüzey parçacıklarını maviye boyadık ve içeride kalan yüzey olmayan parçacıkları ise beyaz renge boyadık. Farklı sahneler yaratarak ve tüm bu sahneleri resim kareleri şeklinde oynatarak yüzey parçacıklarını her karede doğru şekilde tespit edebildiğimizden emin olduk.

4.3 Performans Testi

Hash sistemi ve yüzey parçacık tanımlayıcı sisteminin doğru çalıştığına emin olduktan sonra en son aşamamız performans testi yapmaktı. Unity ortamında içinde bir rampa, dar bir koridor ve koridorda bir adet küre ve bir adet kapsül objesini parçacıklara çarpması için engel olarak kullandık. Böylece test edilebilecek birden fazla durum(çarpma, akma, engele takılma) tek sahnede test edilebilir oldu.

Test sahnesinde, parçacıklar rampanın üstünden bir yığın halindeyken bırakıldı ve koridor boyunca engellere çarparak ilerledikten sonra parçacıklar koridorun sonuna kadar gelip duruluyor. Bu aşamaları hem grafik olarak hem de değerleri resim karesi başına tablo olarak kaydettik. Bu aşamalar sahnemizin ilk 100 resim karesine karşılık gelmekte. İlk olarak hash sisteminden başlamak gerekirse, hash sistemine aynı sahnede sistemine performans testimizi yaptık ve resim karesi başına ortalama 4-6 milisaniye arasında bir sonuç elde ettik.

Yüzey parçacık tanımlayıcının performansının ise parçacıkların sahnedeki pozisyon karmaşıklığına göre değiştiğini saptadık. Bir hücredeki komşu parçacıkların sayısı arttıkça, hesaplama yükü arttığı için yüzey parçacıklarını tespit etmek daha uzun sürmektedir. Örneğin, parçacıklar rampaya çarptığı an parçacıklar birbirine çarpıp yaklaşmakta ve belirli alanlara yığılmaktadır. Bu yüzden de hesaplama yükü artmakta ve algoritmanın yürütülmesi milisaniye cinsinden daha uzun sürmektedir. Ortalama olarak 400 – 500 milisaniye civarlarında işleme süresine sahiptir.

5. Sonuçlar

Unity gibi kullanılması kolay olan ve daha kullanıcı dostu olan bir platformda simülasyonların çok fazla kullanılmaması bizi bu ortamda POF sistemini inşa etmemizi sağladı. Unity ortamında algoritmaların uygulanabilir, test edilip karşılaştırılabileceği uygun ve kararlı bir sistem yarattık.

Ana amaçlarımızdan biri olan, simülasyonu Unity ortamına entegre edip yüzey parçacıklarını bulmayı hem görsel hem de süre bazlı testlerle hazırlayıp kullanıcılar için inceleyebilecekleri bir ortam oluşturduk.

POF sistemi parçacık bazlı sıvı simülasyonunun performans olarak geliştirilmesinde rol oynar. Sıvı simülasyonu çalışırken her resim karesinde (frame) parçacıkların durumlarını analiz ederek sistemin daha efektif çalışmasını sağlar. POF sisteminin gelecekteki çalışmaları olarak başka metotları da uygulayıp test etmek sistemin daha iyi optimize edilmesini mümkün kılar.

POF sistemi animasyon, film ve oyun gibi çeşitli alanlarda fakat daha çok simülasyon ve bilimsel araştırma ve geliştirme projelerinde kullanılabilir. Projeyi daha da ilerletebilmek için sadece yüzey parçacık tanımlayıcı sisteminin bulunduğu yüzey parçacıklarını çizdirmek performansı daha da artırır.

Proje Kodu ve Adı:	Tears Online : Blockchain Integrated Multiplayer Online Game
Proje Takımı:	Bariş Üçkardeş, Buğra Bozkurt, Oğuz Sert, Utku Süsoy
Proje Danışmanları:	Prof. Dr. Mehmet Ufuk Çağlayan
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. Requirements Specifications Document 3. Design Specificaiton Document 4. Product Manual 5. Product-Software Subsystem 6. Any other deliverables
Proje Web Adresi:	tears-online.com

Proje Özeti

1. Giriş

Günümüzde geliştiricilerin ve oyuncuların önemsedığı en önemli nokta güvenlik ve performanstır. Güvenlik konusunda birçok problem vardır. Bu sorunun nedeni güvenliği sağlamaya çalışırken performansın düşmesidir. Bu projede amaç güvenliği sağlarken performansın olabildiğince optimize edilmesiydi. Bu projede güvenliğin sağlanması amacıyla blockchain entegre edildi. Bunun yanı sıra performans analizleri ve testler yapıldı. Güvenlik ve performans verimliliği sağlandıktan sonra oyun içi detaylandırmalar yapıldı. Bu projenin yapım aşamasında Unity Engine, Visual Studio, Atom, Blender3d, Ethereum gibi programlar kullanıldı. Ayrıca C# , javascript,solidity gibi dillerden yararlanıldı.

2. Gereksinimler

Bizim projemizin tabiri caizse tek bir gereksinimi vardı,oda blockchain entegre edilmiş oyun ortamı.Bu yüzeysel gözüksede bu gereksinininin bir sürü alt başlıkları vardır. Bunlar çoklu oyunculu ortam sağlama,bulut veritabanı desteği ve oyunsallaştırma elementleri.İlk başta gereksinimleri kendi ilgi alanlarımıza ve şuanda edindiğimiz yetkinliklere göre sıraladık.Bu sıralama şöyledir.Çoklu ortam,bulut ve oyunsallaştırma.Burda belirtildiği gibi bu sırada gereksinimlerimizi tamamladık.

3. Tasarım

Oyun içinde oyuncuların aralarında yaptığı her bir trade işlemi, oyuncunun client dan server tarafına bir trade isteği gönderilmesiyle son bulur. Server tarafı, Nethereum yardımıyla bu trade isteğini cevaplar ve daha önceden derlenmiş olan Akıllı Sözleşmenin Bytecode'unu kullanarak kontrat objesine işler. Akıllı Sözleşme objesine sadece trade ile alakalı bilgiler ve oyuncular ile alakalı bilgiler yazılır. Doldurulan kontrat İnfura aracılığıyla Ethereum Test Ağına dağıtılır. Ethereum Test Ağına dağıtılan kontratlara tekrardan erişilerek trade yapan oyuncuların server ve database kısımları kontrol edilerek güncellenir.

4. Testler

İlk test aşamasında, oyunun yerel olarak çalışıp çalışmadığı kontrol edilir. Daha sonra oyunun mekaniği ve oyuncu fonksiyonlarının istenildiği çalışıp çalışmadığı test edilir. Tüm bu aşamalarda oluşabilecek hatalar tespit edilir ve düzeltilir. İkinci test kullanıcı ve sunucu arasındaki bağlantıda yapılır yani oyunun Multiplayer(Çokoyunculu) hale gelip gelmediği kontrol edilir. Ana hedef güvenli bir bağlantı elde etmektir, bu yüzden bunu doğrulamak için clienttan, oluşturulan server a bağlantılar yapılır. Üçüncü aşama ise yazdığımız akıllı kontratı local de test etmek olacaktır. Bu aşamada Ganache-Cli kullanarak yazdığımız kontratı Ganache Local Test Ağına dağıtıp kontratın fonksiyonlarını test ettik. Son test aşaması ise Multiplayer oyunumuzun içinde yapılan her bir trade işlemini Ethereum Test Ağına dağıtmak olacaktır. Bu dağıtım işlemi sonucunda da Ethereum Ağına yapılan her bir dağıtım işleminin oyuncuların fps değerleri üzerinde ki etkisi test edilerek performans analizi yapılmaktadır.

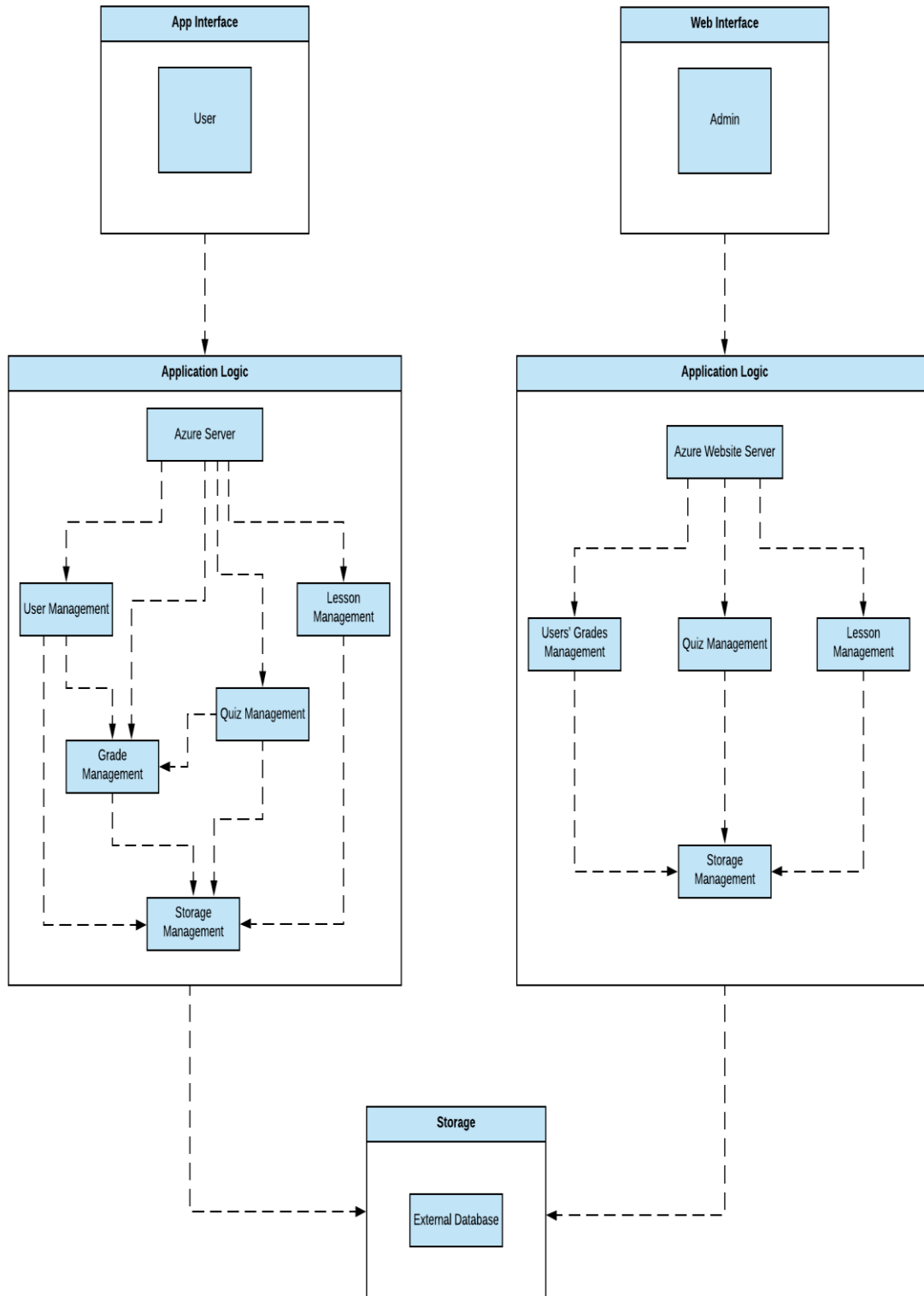
5. Sonuçlar

Hedefimiz blockchain kullanarak güvenli bir oyun çevresi yaratmaktı. Projemizde bu hedefimizi ethereum blockchain teknolojisi kullanarak gerçekleştirdik. Blockchain ile takas işlemlerimizi güvenli bir hale getirip herhangi istenmeyen aksiyonların kolayca fark edilmesine yada önlenmesini sağladık. Projemizden online oyun geliştiricileri faydalanacaktır. Blockchain teknoloji sayesinde oyun geliştiricileri daha güvenli bir oyun deneyimi sunacaklar. Bu durum müşteri güveninin artmasına sebep olacaktır. Projemizi daha ileri taşımak için, takas işlemlerimizin süresini kısaltmamız gerekmektedir. Bu durum R.gerçekleşirse güvenli oyunlar adına yeni yollar ve gelişimlerin kapısı açılacağına inanıyoruz.

Proje Kodu ve Adı:	UOA: Universe of Anatomy
Proje Takımı:	Bariş Dalgıç, Berra Çolak, Büşra Kirpi, Gamze Kızıldağ
Proje Danışmanları:	Prof.Dr. Mehmet Ufuk Çağlayan
Proje Çıktıları	<ol style="list-style-type: none"> 1. Final Report 2. Requirements Specifications Document 3. Design Specifications Document 4. Product Manual 5. Product-Software Subsystem 6. Proje Websitesi 7. Poster
Proje Web Adresi:	-
Proje Özeti	
<p>1) Giriş Universe of Anatomy, tıp öğrencileri için tasarlanmış eğitici bir android mobil uygulamasıdır. Amacımız mobil cihazların ulaşılabilirlik, kişiselleştirilebilirlik ve taşınabilirlik gibi pek çok özelliği sayesinde öğrencilerin sınıf dışında da istediği konuyu öğrenebilmesi, farklı alıştırmalar yaparak kendini geliştirebilmesidir. Öğrenci isterse herbir anatomi sistemiyle ve onun alt başlıkları olan insan anatomisi organlarıyla ilgili bilgileri okuyabilir ve bu konuları 3D objeler, artırılmış gerçeklik yardımıyla daha detaylı inceleyip pekiştirebilir. Bu sayede, öğrenci öğrenmek istediği konuyu sadece metinsel olarak değil; görsel olarak da inceleyebilecektir ve bu objelerin herbir parçasına tıklayarak o parçanın işlevini öğrenecek, aynı zamanda objeyi istediği gibi döndürüp, yaklaştırarak objeyi detaylı olarak görebilecektir. Öğrenci dersin hocasının sisteme eklediği quizlere katılarak hem kendini test edebilecek hem de bu quizlerden not alabilecektir.</p>	
<p>2) Gereksinimler</p> <ul style="list-style-type: none"> • Visual Studio 2017 veya üstü. • Unity 2019.3.13f. • Microsoft Azure Platform. • SQL Server Management Studio (SSMS). • ASP.NET MVC (Web Api) • Vuforia 9 (Unity ile entegre edildi). • Playfab SDK (Unity ile entegre edildi). • İnsan anatomisine ait 3D objeler. • Vuforia'nın çalışabilmesi için "Marker Image". 	

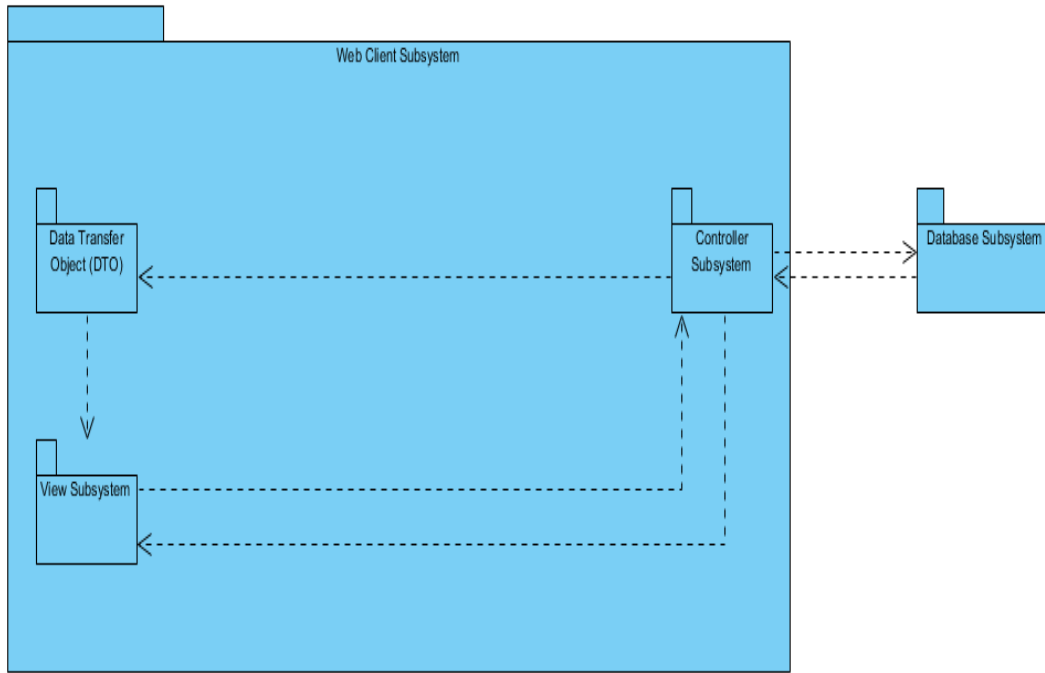
3) Tasarım

- Universe of Anatomy Mobile Application System Design



Öğrenci, uygulama ile etkileşim kurmak için uygulama arayüzünü kullanır. Uygulama mantığı, sunucu ve uygulama arayüzü arasındaki tüm veri aktarımından sorumlu olan ana paketimizdir. Bir öğretim üyesi, bir öğrencinin görebileceği tüm verileri yönetmek için web arayüzünü kullanır. Uygulama mantığı yine ana paketimizdir ve sunucu ile web arayüzü arasındaki veri aktarımını yönetmekten sorumludur. Depolama paketi, tüm verileri depolayan harici bir veritabanı kullanır ve hem uygulama arayüzü hem de web arayüzü bu veritabanından veri alır.

- **Web Client Subsystem Design**



- **Veritabanı Alt Sistemi**

MsSQL Veritabanı Alt Sisteminin görevi verileri düzgün bir şekilde saklamaktır. 2 bölümden oluşur: Kullanıcı Veri Kayıtları (kullanıcı bilgileri), Quizler Veri Kaydı

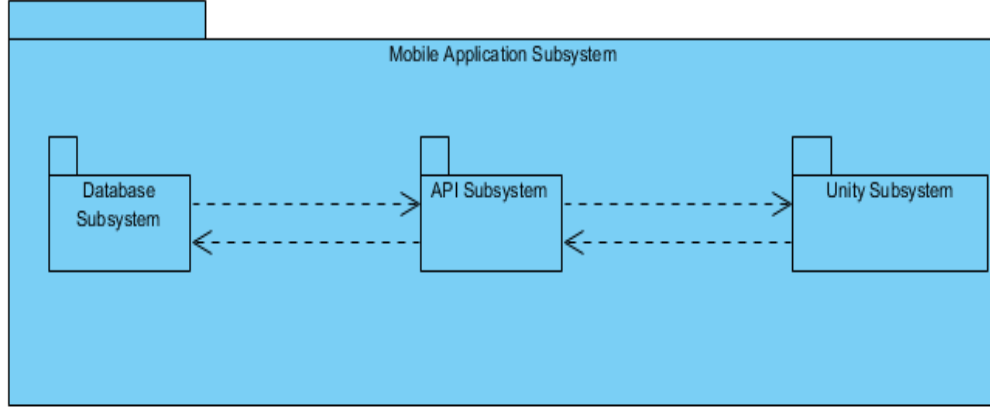
- **Denetleyici Alt Sistemi**

Denetleyici, görünüm alt sistemine gönderilirken modelle birlikte veritabanından alınan verilerin yönetiminde yer alır.

- **Alt Sistemi Görüntüleme**

Web görünümü alt sisteminde, öğrencinin ilerlemelerinin eğitmenlere gösterilmesi için doğru tasarlanmış UI öğeleri olacaktır, eğitmenler öğrencilere kısa sınavlar atayabilir.

- **Mobile Application Subsystem Design**



- **Veritabanı Alt Sistemi**

Mobil Uygulama sistemi tarafından kullanılan Veritabanı Alt Sistemi, öğretim elemanı tarafından yönlendirilen sınav soruları bölümünü içerir.

- **Unity Alt Sistemi**

Bu alt sistem kullanıcıya gösterilen menülerden ve arabirimlerden, bu menülerdeki düğmelerden ve GUI öğelerinden sorumludur. Ayrıca bu alt sistem, anatomi sistemlerini içeren ve öğrencilerin sınavlarını gösteren Organların 3D Modellemesinden sorumludur.

- **API Alt Sistemi**

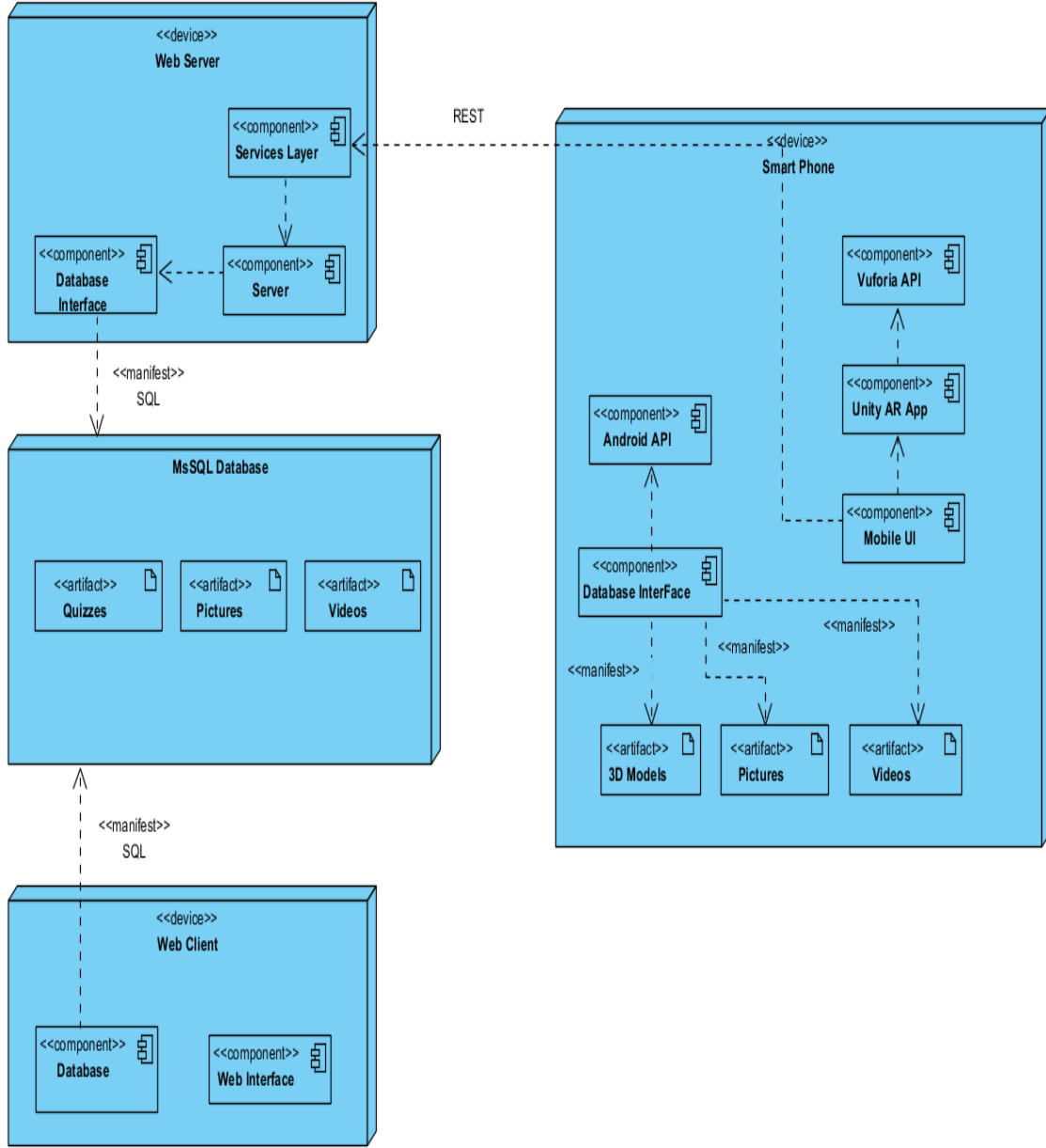
API alt sistemi, veritabanı ve mobil uygulama alt sistemi arasındaki bir bağlantıdır.

Projenin Yazılım Sistem Mimarisi

- Proje, iki alt sisteme sahiptir. Biri Web istemcisi alt sistemi, diğeri ise Mobil uygulama alt sistemidir.
- Web istemci alt sistemi sınavları yönetmek için kullanıyor. Bu alt sistemde MVC kullanılmaktadır. Veritabanı sunucusu, istemci / sunucu mimarisini kullanan bir veritabanı uygulamasının arka uç sistemine atıfta bulunmak için kullanılan bir terimdir ve sunucu veri analizi, depolama, veri işleme ve kullanıcıya özgü olmayan diğer görevler gibi görevleri yerine getirir.
- Sınav verileri veritabanından web istemcisi alt sistemine alınır, diğer taraftan bu veriler bir web hizmeti aracılığıyla mobil uygulamaya aktarılır.
- Web istemcisi alt sistemi ve mobil uygulama alt sistemi birbirleriyle bir SQL veritabanı ile ilişkilidir.
- Anatomi evrenini yaratmadan önce, bu sistemin web istemci alt sistemi ve mobil uygulama alt sistemi tarafından oluşturulmasına karar verilmiştir. Bu iki alt sistem arasında veri depolamak ve veri aktarmak için SQL veritabanını kullanılmıştır.

Projenin Yazılım Sistem Ortamı

Universe of Anatomy Mobile Application Software System Environment



Anatomi Mobil Uygulama Evreni, Mobil Uygulama ve Web Sitesi olmak üzere iki alt sistemden oluşmaktadır: Kısa sınavlar, resimler ve videolar web sitesi üzerinden veritabanı sunucusuna yüklenir. Android Mobil Uygulaması, REST API'si aracılığıyla veritabanına yüklenen sınavları, videoları ve resimleri alır. Ardından bunları kullanıcılara gösterir.

- Web sitesinde C # ve Html kullanılır. Mobil uygulamada Unity komut dosyaları için C # kullanılır.
- Bu projede Unity, Vuforia, Azure API Hizmeti, Azure Veritabanı Sunucusu, MVC.NET yazılım araçları kullanılmıştır.
- Bu uygulamayı kullanmak için android 4.2 veya üzeri sürüm ve internet bağlantısı gereklidir.

4) Uygulama ve Testler

Web tarafı & Unity tarafı olmak üzere iki kısımda incelenmiştir

Web tarafında: Data Structure Testi

Veri yapısı, yazılımın neredeyse en önemli parçasıdır, çünkü tüm alt sistemler veriye dayalıdır ve veritabanındaki verilerin tutarlılığı ile ilgili herhangi bir sorun varsa, kullanıcılara gösterilen doğru verilerde bir sapma olacağı anlamına gelir. Bu nedenle veri yapısını ayrıntılı bir şekilde test etmemiz gerekmektedir.

Test sonucu: Kullanıcıdan veriler doğru şekilde alınıp uygun şekilde depolanmaktadır.

Aynı şekilde veriler, veritabanından doğru ve tutarlı bir şekilde gelmektedir.

Unity tarafında: Data Structure Testi

Test edilen sonuçlar aşağıda listelenmiştir:

- Unity ve Visual Studio entegrasyonu başarılıdır.
- 2. Visual Studio'da depolanan verileri Unity ile bağlayabilmekteyiz. Unity ve Web hizmeti bağlantısı başarılıdır.
- 3. Bazen web servisi ile veritabanı bağlantısının başarısız olabileceğini öğrendik.
- 4. Kullanıcı genel bir sistem sınav düğmesine tıklarsa veritabanı bağlantısı kurulduğunda, Unity, insan vücudunun sahip olduğu sınavların sayısına (yani soruları çözmek istediği) dayalı olarak düğmeleri getirebilir ve dinamik olarak başlatabilir ve sorular Unity oynatıcısında düzgün şekilde gösterilir.
- Kullanıcının yanıtı doğruysa, karşılaştırılmaz ve bu karşılaştırmaya dayanarak uygun animasyon başarıyla oynatılır. Puan verileri doğru şekilde hesaplanabilir ve kullanıcıya gösterilebilir
- En yüksek puan değeri başarıyla kalıcı olarak saklanabilir ve her sınav bitiş ekranında gösterilebilir.

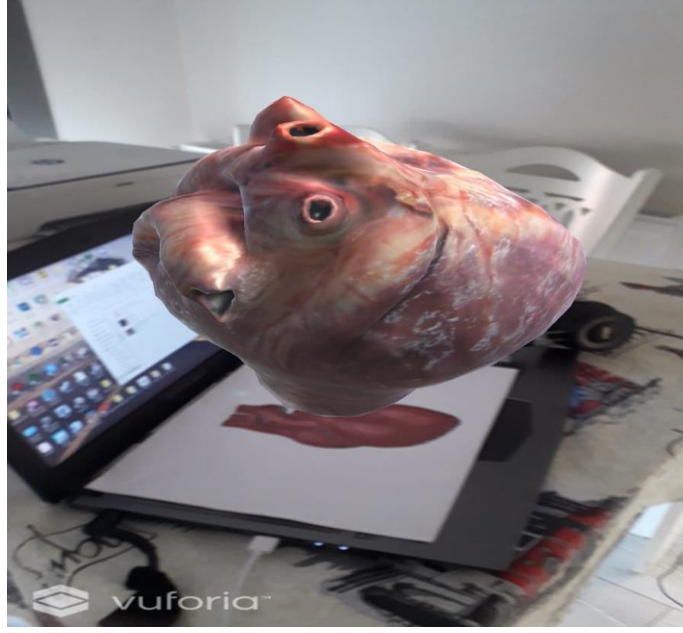
Web tarafında : GUI Testi (WEB UI)

Hem web hem de unity projelerinin test edilmesi gereken GUI kısmı vardır. Düzgün çalışan GUI olmadan, kullanıcılar yazılımımızla etkileşime giremezler. Bu nedenle, her GUI işlevinin doğru çalıştığını görmek için test edilmesi gerekir. Test edilen fonksiyonlar aşağıda listelenmiştir:

- Resimler ve videolar düzgün ve sorunsuz bir şekilde görülür
- Quiz ekleme, silme ve düzenleme butonları doğru çalışıyor.
- Metin kutuları ve girişler düzgün çalışıyor.

Unity tarafında: GUI Testi

- Düğmeler düzgün çalışıyor.
- TextView öğeleri uygun konumlarda ve kullanıcıya doğru şekilde gösteriliyor.
- Açılır listeler düzgün şekilde açılır.
- Kaydırma görünümü doğru kaydırır ve içindeki tüm içeriği gösterir.
- 3D nesnelerin yumurtlama konumları uygun konumlardadır.
- Ekrandaki kullanıcı dokunmaları başarıyla çalışıyor.
- AR 3D nesnelere ortaya çıkar ve düzgün çalışır.



Akıllı telefonda alınan Vuforia ekran görüntüsü.

Unity tarafında: WEB Hizmeti Testi

Web hizmeti, sunucuya ve sunucuya gelen tüm verileri işleyen tüm projenin temel parçasıdır. Test edilen işlevler aşağıda listelenmiştir:

- Denetleyici sınıflarının her yöntemi doğru URL'lerle eşlenir
- Her denetleyici sınıfı verileri düzgün şekilde alır ve ilgili hizmet sınıflarına gönderir.
- Her denetleyici sınıfı, verileri veritabanına iletmek için doğru model nesnelere oluşturur.
- Bazen web hizmeti ve veritabanı bağlantısı başarısız olabilir.

5) Sonuçlar

Yapılan testler sonucunda; ufak tefek sıkıntılar haricinde uygulama amacına uygun ve doğru bir şekilde çalışmaktadır. Uygulama bir web servis aracılığıyla verileri aldığı için güvenlik açısından bir sıkıntı yoktur. Uygulamanın içerik olarak daha fazla geliştirilme aşaması şu anda devam etmekte olup, hedef kitlemiz olan tıp öğrencilerine daha uygun hale getirilecektir. Daha önce de söylediğimiz gibi projenin hedef kitlesi tıp öğrencileridir. Fakat ileride, uygulamanın başka konulara uyarlanabilirlik açısından gayet esnek olmasından dolayı hedef kitlesi genişletilebilir.

Proje Kodu ve Adı:	EVOBOX: Akıllı İlaç Kutusu
Proje Takımı:	Ata Bayraktar, Doruk Tekintaş, Yetkin Yurtsever
Proje Danışmanları:	Prof. Dr. Mehmet Ufuk Çağlayan
Proje Çıktıları	<ol style="list-style-type: none"> 1. EVOBOX Final Raporu 2. EVOBOX Gereksinimler Teknik Özellikler Belgesi(RSD) 3. EVOBOX Tasarım Özellikleri Belgesi(DSD) 4. EVOBOX Ürün Kılavuzu(PM) 5. EVOBOX Android Uygulaması 6. EVOBOX Konteyneri (Fiziksel Kutu)(COVID-19 Pandemisi dolasıyla üretilmedi) 7. EVOBOX Web Sitesi 8. EVOBOX Proje Poster 9. EVOBOX Sunum Slaytları 10. EVOBOX Bitirme Projesi Özeti 11. EVOBOX Video Gösterimi (https://www.youtube.com/watch?v=VDf-Shk0sR8)
Proje Web Adresi:	evobox.tr.ht
Proje Özeti	
<p>1 Giriş</p> <p>1.1 Problem Tanımı</p> <p>huffpost.com[1] adlı internet sitesinin yaptığı araştırmaya göre genç nesillerin yaşlı nesillerden daha unutkan olma olasılıkları daha yüksektir. Bu durum gelecekte daha da unutkan yaşlı nesiller ile karşılaşmamıza sebep olacaktır. İnsanların hayatlarının belirli dönemlerinde çeşitli ilaçlar kullanması gerektiğini düşünülünce ilaçları içmeği unutmak büyük sorunlara yol açabilir. Bu sorunları çözmek için EVOBOX projesi hayata geçmiştir. İnsanların alması gerektiği ilaçları haftalık olarak EVOBOX fiziksel kutusuna doldurduktan sonra, EVOBOX için özel olarak geliştirilen uygulamasına alması gereken ilaçların saatlerini girmesi gerekmektedir. Bu işlem yapıldığında EVOBOX ilaçları belirttiğiniz saatlerde otomatik olarak açarak kutunun altındaki ilaç bardağına düşmesini sağlar. İlaç belirli bir süre ilaç bardağından alınmaz ise ilacın alınmadığını uygulama yolu ile haber verir. Yapılan araştırmalarda Ellie[2] ve Memo Box[3] adlı iki akıllı pil kutusu projesi daha olduğu görüldü. EVOBOX'ın diğer projelerden farkı ise diğer projelerde hastalar kutular açıldıktan sonra ilacı el ile alması gerekir iken EVOBOX'ta ilaçların otomatik olarak ilaç bardağına düşmesidir. Bu yöntem ile insan hatasının yaşanmasını engellemeye çalışılmıştır.</p> <p>1.2 Proje Amacı</p> <p>Projenin amacı, hastaların hayatını kolaylaştırmak, ilaçlarını belirledikleri saatlerde almalarını sağlamak, ilaçlarını eksik veya fazla almalarını önlemek ve ilaçlarını almadıklarında uygulama yolu ile onlara haber vermektir.</p>	

1.3 Proje Faaliyetleri ve Takvimi

- Projenin ana konusuna ve konseptine karar verildi.
- Projenin genel işlevleri ve gerekliliklerine karar verildi.
- EVOBOX Gereksinimler Teknik Özellikler Belgesi(RSD), Revizyon 1.0 tamamlandı.
- EVOBOX Gereksinimler Teknik Özellikler Belgesi(RSD), Revizyon 2.0 tamamlandı.
- Fiziksel kutu, donanım ve yazılım tasarımları yapıldı.
- EVOBOX Tasarım Özellikleri Belgesi(DSD), Revizyon 1.0 tamamlandı.
- COMP4910 için EVOBOX Final Raporu tamamlandı.
- COMP4910 için EVOBOX Web Posterini tamamlandı.
- EVOBOX Tasarım Özellikleri Belgesi(DSD), Revizyon 1.1 tamamlandı.
- EVOBOX Tasarım Özellikleri Belgesi(DSD), Revizyon 2.0 tamamlandı.
- EVOBOX Ürün Kılavuzu(PM), Revizyon 1.0 tamamlandı.
- EVOBOX Ürün Kılavuzu(PM), Revizyon 1.1 tamamlandı.
- EVOBOX Gereksinimler Teknik Özellikler Belgesi(RSD), Revizyon 2.1 tamamlandı.
- EVOBOX Tasarım Özellikleri Belgesi(DSD), Revizyon 2.1 tamamlandı.
- EVOBOX Ürün Kılavuzu(PM), Revizyon 1.2 tamamlandı.
- COMP4920 için EVOBOX Final Raporu tamamlandı.
- EVOBOX Sunum Slaytları tamamlandı.
- EVOBOX Web Posterini tamamlandı.
- EVOBOX Web Sitesi tamamlandı.
- EVOBOX Bitirme Projesi Özeti tamamlandı.
- EVOBOX Video Gösterimi tamamlandı.

2. Gereksinimler

Covid-19 salgını nedeniyle projenin fiziksel kısımları çalışmıyor.

No.	Gereksinimler	Kullanım Senaryosu
1	Kullanıcının Adını, Soyadını, Şifresini, E-mailini ve Telefon numarasını kaydeder.	Kayıt Ekranı
2	Kullanıcı adı ve şifre bilgilerini kontrol eder bilgiler doğru ise kullanıcı sisteme giriş yapar.	Giriş Ekranı
3	Kullanıcı sisteme hastayı kaydeder. Kullanıcı isterse kendisini hasta olarak ekleyebilir.	Hasta Ekleme
4	Var olan bir hastayı sistemden kaldırır.	Hasta Kaldırma
5	Var olan hastanın bilgilerini düzenler.	Hasta Düzenleme
6	Hapın adı, adedi, boyutu ve hangi zamanda alınması gerektiği bilgilerini sisteme ekler.	İlaç Ekleme
7	Var olan bir ilacı sistemden kaldırır.	İlaç Kaldırma
8	Var olan ilacın bilgilerini düzenler.	İlaç Düzenleme
9	Hastayı ilacını içesi için uyarır aynı zamanda girilen zamanda ve adette hapi hap kutusuna bırakır.	İlaç Uyarısı
10	Kullanıcıyı hastanın almadığı ilaçlar konusunda uyarır.	Alınmayan İlaç Uyarısı

Tablo 1 : Gereksinimler ve Kullanım senaryosu ilişkileri

3. Tasarım

a. EVOBOX Genel Sistem Mimarisi

EVOBOX genel sistemi iki katmanlı bir yapı kullanır ve iki ana alt sistemden oluşur.

EVOBOX Yazılım Alt Sistemi

1. EVOBOX Kullanıcı Arayüzü Alt-alt sistemi
2. EVOBOX Uygulama Alt-Alt sistemi
3. EVOBOX Veritabanı Alt-Alt sistemi

ii. EVOBOX Donanım Alt Sistemi

1. EVOBOX Donanım Alt-Alt sistemi
2. EVOBOX Konteyner Alt-Alt sistemi

Katmanlı yapı, farklı sistem ve bileşen türlerine sahip bir donanım/yazılım sistemi oluşturmaya daha uygun bir sistem olduğu için seçildi ve bu bileşenlerin birleşimi bütün sistemi oluşturur. İlk olarak, tüm donanım ve mekanizmaları içeren fiziksel bir konteyner gereklidir ve bu konteynerin içerisinde ilaçların düşme mekanizmasının çalışması için gereken bazı donanımlar vardır. Sonrasında, donanımların yazılım uygulamaları tarafında control edilmesi gerekir.

EVOBOX Donanım Alt Sistemi

EVOBOX Donanım Alt Sistemi EVOBOX Donanım Alt-Alt sistemi ve EVOBOX Konteyner Alt-Alt sistemi olmak üzere iki alt-alt sistemden oluşur.

EVOBOX Konteyner Alt-Alt Sistemi ve Konteyner Tasarımı

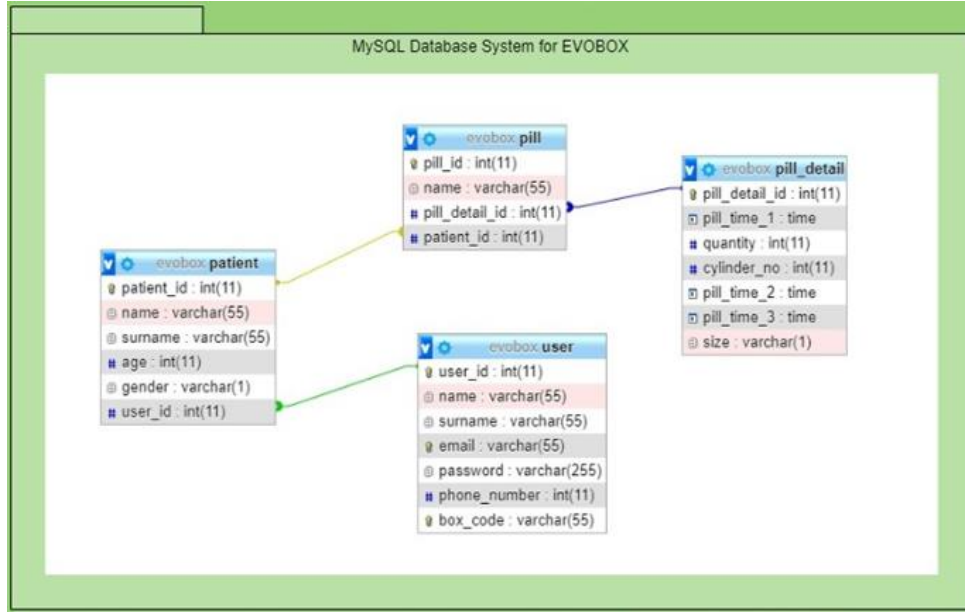
EVOBOX Konteyner hastanın kullanacağı kutudur. Üç farklı boyut hap için üç farklı ilaç silindiri bulunur ve ilaçlar buralara yerleştirilir. Her silindirin altında iki ayrı disk vardır. Disk boyutları hap boyutuna göre değişmektedir. Hapın düşürülmesine yardımcı olmak için kutunun üzerinde 3 plaka bulunur. Her diskin altında bir servo motor vardır. Servo motorlar diskleri döndürür. Servo motorların altında servo motorları tutmak için bir plaka vardır. Servo motorların plakasının altında bir koni vardır. Düşen tüm ilaçlar konide toplanarak ilaç bardağının içine düşer. En aşağı kısımda ilaçların alınıp alınmadığını belirleyen ultrasonik sensor için bir boşluk vardır.

EVOBOX Donanım Alt-Alt Sistemi

EVOBOX Arduino UNO mikrodenetçisini kullanır. Tüm gerekli bileşenler Arduino UNO ile çalışır. Tüm donanım elemanları EVOBOX Konteynerinin içinde kullanıcının ulaşamayacağı bir yerdedir. Arduino UNO ile birlikte bir Wi-Fi modülü, üç adet servo motoru, bir ultrasonik sensör, kablolar, ledler ve bir adet zil bulunur. Wi-Fi modülü internet bağlantısını sağlar, servo motorlar ilaçların düşmesi için kullanılır. Ultrasonik sensör ilaç bardağına yakın bir yere konumlandırılmıştır ve ilaçların alınıp alınmadığını control eder. Kablolar tüm devreyi birbirine bağlar. LED ler ve zil, ilaçların alması için kullanıcıyı uyarmak için kullanılır.

EVOBX Yazılım Alt Sistemi

EVOBX Donanım Alt Sistemi mimarisi aksine, Yazılım Alt Sistemi için farklı bir iç yapı kullanılmıştır. Bu yapı MVC yapısıdır. EVOBX nesne yönelimli bir uygulama ve sınıfları model, görünüm ve kontrolör konseptlerine benzediği için MVC yapısı seçilmiştir. EVOBX Yazılım Alt Sistemi, EVOBX Veritabanı Alt-Alt sistemi, EVOBX Uygulama Alt-Alt sistemi ve EVOBX Kullanıcı Arayüzü Alt-Alt sistemi olmak üzere üç alt-alt sistemden oluşur. EVOBX Veritabanı Alt-Alt Sistemi EVOBX'ın veritabanını temsil eder. MVC yapısındaki model paketidir. EVOBX Uygulama Alt-Alt sistemi EVOBX'ın tüm kontrolörüdür. Aynı şekilde MVC yapısında kontrolör paketidir. EVOBX Kullanıcı Arayüzü Alt-Alt sistemi uygulamanın arayüzünü temsil eder ve kullanıcının uygulamayla etkileşime girdiği kısımdır. MVC yapısındaki görünüm paketidir.



Şekil 1: EVOBX Paket Diyagramı olarak Veritabanı Alt-Alt Sistemi

EVOBX Uygulama Alt-Alt Sistemi

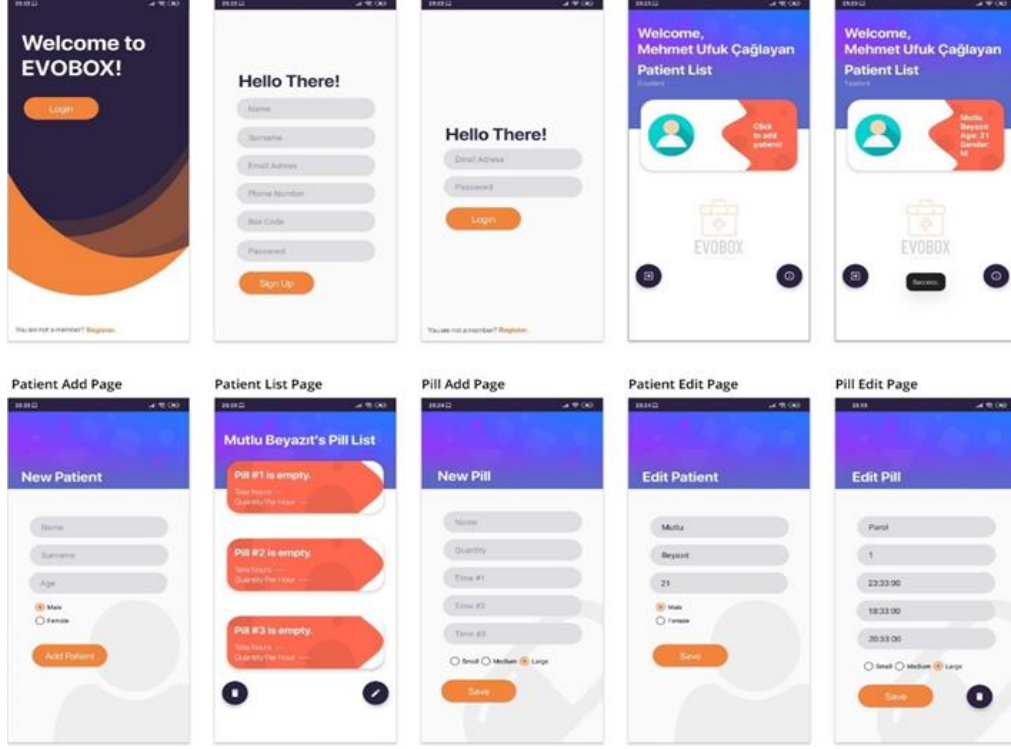
Uygulama Alt-Alt sistemi MVC yapısındaki kontrolör paketidir. EVOBX'ın tüm kontrolör fonksiyonlarını içerir. WebAPI Kontrolör, Android Kontrolör ve Arduino Kontrolör olmak üzere üç paketten oluşur.

Kontrolör Paketlerinin Açıklamaları

- WebAPI Kontrolör paketi “*Server.js*” adlı tek bir dosyadan oluşur. Bu kontrolör metotları get, post ve delete istekleriyle ilgilenir ve veritabanını EVOBX uygulamasına bağlar.
- Arduino Kontrolör paketi “*arduinocode.ino*” adlı tek bir dosyadan oluşur. Bu kontrolör WebAPI isteklerine göre Arduino donanımlarının hareket etmesini sağlar.
- Android Kontrolör paketi 9 adet kontrolör sınıfından oluşur. Her Android kontrolör sınıfı belirli bir kullanıcı arayüzü sayfasının işlevselliğini ele aldığı için aynı sayıda kullanıcı arayüzü vardır. Belirli bir sayfanın düğme yönlendirmelerini işler ve gerekli işlemleri yapmak için gerekli WebAPI metotlarını kullanırlar.

EVOBOX Kullanıcı Arayüzü Alt-Alt Sistemi

Kullanıcı Arayüzü Alt-Alt sistemi MVC yapısındaki görünüm paketidir ve kullanıcının uygulamayla etkileşime girdiği kısımdır.



Şekil 3: EVOBOX Mobil Uygulama Kullanıcı Arayüzü

4. UYGULAMA ve TESTLER

Ürünün Uygulanması Donanım

i. Ana olarak Arduino UNO Mikrodenetleyicisi kullanılmıştır. Arduinonun yanında, ESP 8266 Wi-Fi Modülü, zil, LED'ler, HR-S04 Ultrasonik mesafe sensörü, servo motorlar, devre tahtası, dirençler ve güç kaynağı kullanılmıştır.

Yazılım

- Web-API'yi oluşturan backend programlama için Node.js kullanıldı.
- Android uygulaması için Java kullanıldı. Uygulama WebAPI ile iletişim kurar.
- Arduinoyu programlamak için Arduino IDE kullanıldı.
- Arduinoyu programlamak için C dili kullanıldı.
- Veri tabanı olarak MySQL kullanıldı.
- Arduino IDE ve Android Studio backend programlama için kullanıldı.
- HTML, CSS, Bootstrap ve Javascript frontend programlama için kullanıldı.
- WebAPI geliştirilirken HTTP isteklerinin testi için Postman Uygulaması kullanıldı.

Testler ve Sonuçları

WebAPI Postman uygulaması ile test edildi. Seçilen URL ler uygulamaya doğru istek yöntemlerini verdi ve veri tabanından gerekli verileri döndürdü. Test başarıyla sonuçlandı.

Android uygulaması Android emulâtörü ve gerçek akıllı telefon kullanılarak test edildi. Sayfalar ve tuşlar çalıştı. Test başarıyla sonuçlandı.

İlacın servo motorlar ile düşme mekanizması Pandemi sebebiyle Fiziksel kutu ve silindirler üretilemediği için kartonlar kullanılarak test edildi. Test başarıyla sonuçlandı.

Arduino testi arduino idesindeki seri monitör yardımıyla yapıldı kod arduinoya yüklendi ve işlem seri monitörden izlendi. Bildirim testi için android emulâtörü kullanıldı cihaz bildirimini aldı.

5. SONUÇLAR

a. Projenin Faydaları

Bu proje insanlara ilaçlarını içmesini hatırlatmaya yardımcı olur ve ileride ilaçları içmemek veya fazla içmek sonucu oluşabilecek zararlardan korur. Aynı zamanda EVOBOX uygulaması kullanılarak kendisinin veya refakatçisi olduğu başka bir kimsenin ilaç takibini yapabilir.

Gelecekte Yapılabilecekler

1. COVID-19 sonrası fiziksel kutu üretilebilir.
2. Evinde internet olmayan hastalar için Arduino'ya GSM portu eklenebilir.
3. İlaçları doldurmak için kutuyu açarken QR kodu ile teşhis yapılabilir.
4. Farklı bir dizayn ile taşınabilir bir EVOBOX modeli üretilebilir.

6. REFERANSLAR

- Emling, S. (2013, August 2). Study Reveals Disturbing Trend Among Millennials. Retrieved December 23, 2019, from https://www.huffpost.com/entry/millennial-forgetfulness_n_3695512.
- The Smart Pill Box with Brains & Beauty. (n.d.). Retrieved May 1, 2020, from <https://elliegrid.com/>
- Memo Box Smart Pillbox: Award Winning Pill Organizer & Reminder. (n.d.). Retrieved May 1, 2020, from <https://pillbox.inylogics.com>