

KULLANILABİLİRLİK TESTİ ARAÇLARI İLE KULLANICI DAVRANIŞLARININ BELİRLENMESİ VE KULLANICI ARAYÜZLERİNİN PLANLANMASI

Ahmet Gürkan Yüksek^{1*}, Halil Arslan¹, Mustafa Lemi¹ and Gülşah Çiftçi

¹ Department of Computer Engineering, Cumhuriyet University, Turkey

* agyuksekc@cumhuriyet.edu.tr

Özet: Sürekli yeni uygulama ve web sitelerinin geliştirildiği günümüzde web veya mobil uygulamaların hızlı bir şekilde ve en az kod yapısı ile etkileşimli bir şekilde geliştirebilmesi için daha iyi yöntemlere ve araçlara ihtiyaç duyulmaktadır. Bu noktada geliştirilen uygulamaların hem tasarımı hem de kodlandığı programlama dilinin yapısı önem arz etmektedir. Javascript çerçeveleri web temelli uygulamalar için, sayfaların html yapısını sade, basit fakat işlevsel görünüme kavuşturarak uygulamaların daha hızlı sonuçlar vermesini sağlamaktadır. Aynıyeten kullanıcı arayüzüne ait etkileşimli öğeleri geliştirmeye olanak kılmaktadır. Bu çalışmanın asıl amacı, belirli kısıtlar çerçevesinde birtakım görevleri gerçekleştirmek için tasarlanmış olan bir uygulamanın ExtJS SenchaTouch, Angular.js, React.js gibi bazı frameworkler üzerinde platformların birbirleri arasındaki performans kıyaslamalarını ve tasarlanmış olan arayüzlere ilişkin kullanılabilirlik testlerini farklı test araçları kullanarak (Optimal Workshop, Visual Website Optimizer) ölçmek, değerlendirmek ve sonuçlar elde etmek olup geliştiricilere hem tasarım hem de kodlama açısından elde edilen verilere dayalı olarak çıkarımlarda bulunmalarını sağlayacak bilgiler sunmaktır.

Anahtar Kelimeler: Kullanıcı odaklı tasarım, kullanıcı testi, kullanılabilirlik testi, deneysel kullanıcı testi, kullanıcı olayları, kullanıcı izleme, kullanılabilirlik testi araçları, mobil ve web uygulamaları

DETERMINING USER BEHAVIORS WITH USABILITY TESTING TOOLS AND PLANNING USER INTERFACES

Abstract: Nowadays new applications and new web sites are being developed, better methods and tools are needed in order to be able to develop web or mobile applications in a fast and interactive way with minimal code structure. In this point, both the design of the applications developed and the structure of the programming language it is coded are important. For web-based applications, Javascript frameworks provide simple, yet functional, look and feel of the page's html structure, resulting in faster results for applications. In addition, it allows to develop interactive elements of the user interface. The main purpose of this work is to compare the performance of platforms between different platforms on some frameworks such as ExtJS, SenchaTouch, Angular.js, React.js, and the usability tests of the designed interfaces by using different testing tools (Optimal Workshop, Visual Website Optimizer), and to provide developers with information that will enable them to be inferred based on data obtained from both design and coding

Keywords: User-focused design, user testing, usability testing, experimental user testing, user events, user tracking, usability testing tools, mobile and web applications

1-GİRİŞ

Kullanıcı arabirimi (User Interface-UI), geliştirilen her bir programın önemli bir parçasıdır. Program üzerinde kullanıcıların yapacak olduğu görevler kullanıcı ara yüzlerine direkt olarak bağlıdır. İyi planlanmamış bir kullanıcı arayüzü tasarımına sahip olan en mükemmel program dahi işlevselliği

göz önünde bulundurulduğunda kötü bir izlenime ve deneyime neden olacaktır. Uygulamalar üzerinde kullanıcı deneyimlerinin ürünün memnuniyeti açısından önemi bu sebeple çok daha etkili olmaktadır. Geliştirilmiş bir yazılım uygulamasında veya bir web sitesinde kullanıcılar işlemleri, arayüz açısından bakıldığında en kısa sürede ve anlaşılabilir bir yapıda yerine getirilebilecek şekilde tasarlanmalıdır. Bu,

ürünün başarısına ve diğer rakipleri arasındaki durumuna önemli açıdan etki etmekte olup aralarında bir takım farklılıklar da yaratabilmektedir. Kullanıcı arabirimlerinin tasarımında, insan ile olan etkileşim en önemli konulardan biridir. Görsel tasarım disiplinleri iletişim tabanlıdır, bu sebeple insan ile makine arasındaki etkileşim göz önünde bulundurulması gereken bir unsurdur (Klaus Hinum). Kullanıcı arayüzünün önemini ifade eden bir diğer gerçek, Myers ve Rosson tarafından gerçekleştirilen bir çalışmada detaylı olarak değinilmiştir. Bu çalışmada, yazılan kodların büyük bir çoğunluğunun (%70' inin) kullanıcı arayüzüne özgü hazırlandığından bahsedilmektedir (Myers, Brad A., Mary Beth Rosson, 1996). Kullanıcı deneyimine olan ihtiyacın bu kadar fazla olmasının altında yatan ana sebep günümüzde mobil cihaz ve bilgisayar türevi cihazların gün geçtikçe insan hayatının önemli bir parçası olması ve birçok hayati işlerin bu cihazlara veya sistemlere bağımlı olarak gerçekleştirilmesindedir. Kullanılabilirlik nitelikleri, bir web sitesinin kullanıcı dostu olduğunu değerlendiren metriklerdir. Ampirik mobil kullanılabilirlik çalışmalarının incelenmesinde, Coursaris ve Kim (Coursaris, Constantinos K., and Dan J. Kim, 2011), mobil kullanılabilirlik testinde en sık kullanılan üç önemli ölçütü şöyle sunmaktadır:

- **Verimlilik:** Ürünün görevlerin hızlı, etkili ve ekonomik bir şekilde gerçekleştirilmesine imkan tanınması veya performansına etki etme derecesi,
- **Etkililik:** Değerlendirmeye alınan kullanıcıların belirli ortamda belirtilen hedefleri veya görevleri doğru ve eksiksiz bir şekilde gerçekleştirebilme durumu,
- **Memnuniyet:** Bir ürünün kullanıcıyı tatmin etme derecesi

Yazarlar, kullanıcı, görev, çevre ve teknolojinin dört değişkeni ile oluşturulan "bağlamsal kullanılabilirlik" çerçevesinde bu kısıtları ortaya atmaktadırlar. Belirlenmiş bir görev senaryosuna giren bir katılımcının, görev için harcanan zamanı ve hata oranı ölçülerek site verimliliği ve etkinliği değerlendirilmektedir. Kullanıcı memnuniyeti, görevleri yerine getirirken katılımcının memnuniyetinin, kafa karışıklığının veya hayal kırıklığının ifade edilmesi ile ölçülebilmektedir. Kullanıcı arayüzlerinin değerlendirilmesinde üç yaklaşım bulunmaktadır. Bunlar; denetleme, kullanıcı ve model tabanlı değerlendirmelerdir. İlk iki yaklaşım, kullanılabilirlik testini uygulayan kişiler tarafından yaygın olarak kullanılan yaklaşımlar olmakla birlikte kapsamlı bir şekilde belgelenmişlerdir (Rosenbaum S, Rohn JA, Humburg J. A, 2000) (Bastien, JM Christian 2010). Modele dayalı yaklaşımlar ise uygulanması maliyetli ve araştırma ekipleri ile sınırlı olarak düşünülebilir. Hazırlanan bu çalışma ikinci bölümde uygulanacak testlere ait ön tanımlama ve belirlemeler, üçüncü kısmında kullanılacak test yöntemleri ve dördüncü bölümde ise sonuçların karşılaştırılması şeklinde organize edilmiştir.

2- KULLANICI TABANLI DEĞERLENDİRME

Kullanıcının doğrudan katıldığı kullanılabilirlik değerlendirme yöntemleridir. Kullanıcılardan bir ürünle ilgili tipik görevleri yerine getirmesi veya ürünü keşfetmeleri istenebilir. Kullanıcı hatalarına veya zorluklara neden olan tasarım kusurlarını belirlemek için davranışlar gözlemlenir ve kaydedilir. Bu gözlemlerde, bir görevi tamamlamak için gereken süre, görev tamamlama oranları ve hataların sayısı ve türü kaydedilir. Tasarım kusurları belirlendikten sonra ürünün ergonomik kalitesini arttırmak için tasarım tavsiyeleri önerilir. Bir kullanıcı testinin uygulanması genel olarak aşağıdaki gibi belirli adımları içerir (Dumas JS, 2003)(Tullis T, Albert B, 2008).

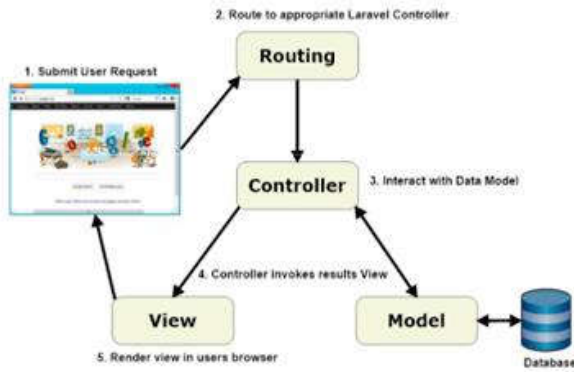
- Test amaçlarının belirlenmesi
- Test katılımcılarının nitelikleri ve seçilmesi
- Görev senaryolarının oluşturulması ve açıklanması
- Kaydedilecek olan veri ölçütlerinin seçimin belirlenmesi
- Test materyalinin ve test ortamının hazırlanması
- Test cihazının seçimi ve test protokolünün tasarımı
- Memnuniyet anketlerinin tasarımı, seçimi ve analizi
- Test sonuçlarının sunumu ve bağlantısı

2-a- Kaç Kullanıcı ile Test Gerçekleştirilmeli ?

Kaç kullanıcının istihdam edileceğine karar vermek hem pratik / ekonomik hem de bilimsel sonuçların elde edilmesini sağlayacaktır. Kullanıcıları bir teste davet ederken asıl amaçlanan en düşük maliyetle (katılımcıların maliyeti, gözlemci maliyeti, laboratuvar imkanlarının maliyeti ve geliştiricilere sağlanan verilerin elde edilmesi için harcanan zaman vs. gibi) bir arayüz tasarımında karşılaşılabilecek kusurları en aza indirmektir (J.M. Christian Bastien). Bu bağlamda deneysel kanıtlara dayanarak test katılımcılarının sayısında arayüz değerlendirmesine herhangi bir katkı sağlamayacak kullanıcıların teste tabi tutulmasına izin verilmemektedir. Doksanlı yıllarda, dört veya beş katılımcı ile arayüze ait kullanılabilirlik sıkıntılarının %80 ile %85' inin ortaya çıkabileceği belirtildi. Buna ek olarak, Spool ve Schröder büyük ölçekli bir kullanılabilirlik değerlendirilmesinin sonuçlarını yayımlayarak. İncelenen web sitelerindeki kullanılabilirlik sorunlarının %85' ini bulmak için 5' ten fazla kullanıcıya ihtiyaç duyacakları sonucuna varmışlardı (Spool J, Schroeder W, 2001). Bu çalışmada yaklaşık 20 katılımcı bir kullanıcı testinde yer almakta olup bu kullanıcılardan, farklı platformlarda geliştirilmiş ve arayüz tasarımlarının da birbirlerinden ayrı olduğu bir uygulama üzerinde istedikleri herhangi bir kişiye iş oluşturabilmesi ve o işe ait gelişmeleri takip edebilmesi istenmektedir. Bu görev 3 farklı platformda gerçekleştirilecektir.

2-b- Uygulamada Kullanılacak Olan Frameworkler ve Kullanılabilirlik Testi Araçları

Frameworkler, yazılım geliştiricilerin kullandığı daha önceden hazırlanmış kütüphaneleri barındıran ve bunlara yenilerini ekleyebilme imkânı tanıyan yapıları temsil etmektedir [1]. Gelişmiş frameworklerde formlar arası kontrol, veri tabanı bağlantısı, kullanıcının sisteme girişi ve çıkışı, tema değişikliği gibi kütüphaneler de mevcuttur. Ayrıca frameworkler MVC (Model, View, Controller) gibi bölümlerden oluşarak geliştirilen uygulamaların sistematik açıdan yönetimini kolaylaştırarak daha okunabilir ve düzenlenebilir olmasını sağlamaktadır. Projelerin bir çoğunda aradan belirli bir süre geçtikten sonra düzenleme yapmak zor bir hal almaktadır. Hem kod sayısının artması hem de yazılan kodun nerede olduğunun unutulması revize sürecini uzatmaktadır. Bu yapılar Modelde veri tabanına ait işlemleri, View'da ekrana yansıyacak olan görünüme ait kodları ve Controller'da ise Model ile View arasındaki genel bağlantıyı sağladığı için gerektiğinde kodlara müdahale edilmesini kolaylaştırmaktadır. Aşağıdaki Şekil 1'de örnek MVC akış diyagramı yapısını görebilirsiniz.



Şekil 1- MVC Akış Diyagramı

EXTJS Sencha Touch: Sencha Touch, çapraz platform mobil web uygulamaları geliştirmek için önde gelen MVC tabanlı JavaScript kütüphanesidir. Sencha Touch mobil cihazlar için yüksek performanslı UI bileşenleri sağlamak için donanım hızlandırma tekniklerini kullanmaktadır. Sencha Touch, 50'den fazla dahili UI bileşeni ve tüm önemli mobil platformlar için doğal görünümlü temalarla, iOS, Android, BlackBerry, Windows Phone ve daha pek çok alanda çalışan etkileyici uygulamalar oluşturmak için ihtiyacımız olan birçok şeyi karşılayabilmektedir. [2].

AngularJS: AngularJS, Google'nin desteklediği, javascript MVC kütüphanesidir. MVC (Modal View Controller) verinin, görünümün ve kontrollerin ayrı tutulmasına dayanmaktadır. Dinamik web uygulamaları için yapısal bir framework'tür.

HTML'i şablon dili olarak kullanmanızı sağlar ve uygulamanızın bileşenlerini açıkça ve kısaca ifade etmek için HTML sözdizimini genişletmenize izin vermektedir. AngularJS çalışmak için jQuery gibi kütüphanelere ihtiyaç duymayıp tek sayfa uygulamaları geliştirmeye imkan sağlamaktadır. Diğer kütüphanelerden en önemli farkı çift yönlü olmasıdır. Böylelikle veride olan değişiklik görünümüne, görünümde olan değişiklik de veriye uygulanır. Directive tanımlayarak html etiketlerinin geliştirilmesine olanak sağlamakta ve kod karmaşasını önemli ölçüde önlemektedir [3,4].

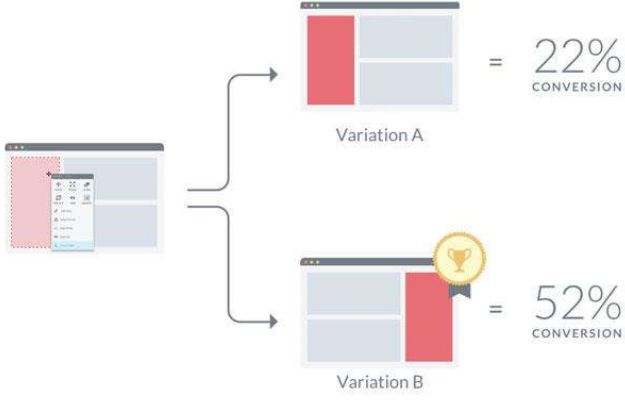
ReactJS: Facebook ve Instagram tarafından oluşturulmuş, arayüz oluşturmak üzere kullanılan bir Javascript kütüphanesidir. Zamana bağlı olarak sürekli değişen verilerin arayüze kolay bir şekilde aktarılıp görüntülenmesini amaçlayarak oluşturulmuştur. Birçok kişi tarafından "MVC" deki "V" yani view olarak kullanılmaktadır. React Javascript ile oluşturulan nesnelere ekrana basılması için kullanılır. JSX adı verilen yeni bir javascript söz dizimini kullanarak kodların çok daha okunaklı bir şekilde (birbiri ile iç içe geçen karmaşık parantez yapıları olmadan) oluşturulmasını sağlamaktadır [5].

Visual Website Optimizer: A/B, Split ve Multivariate gibi çeşitli testleri yapabilmemize imkan sağlamanın yanı sıra tıklanma ve sıcaklık haritalarını göstererek arayüzler üzerinde analiz ve raporlama yaparak elde edilen sonuçlar neticesinde arayüzler üzerinde optimizasyon gerçekleştirebilmemizi sağlayan bir üründür.

Optimal Workshop: Ağaç Testi, First Click Test ve Kart Gruplama gibi bir takım testleri gerçekleştirmemize, gözlem yapmamıza ve geliştirilen uygulama veya sitelere yönelik olarak kullanıcılar için anket yapabilmemize olanak sağlayan bir üründür. Ağaç ve kart gruplama testleri daha çok e-ticaret sitelerinde kullanılmakta olup site hiyerarşisinin kullanıcıların gözünde doğru bir şekilde kurgulanıp kurgulanmadığını tanımlayabilmek için kullanılmaktadır. Siteye gelen kullanıcıların aradıkları herhangi bir ürünü hangi menünün altında aradığına dair bilgileri bize sunmaktadır.

3-TEST YÖNTEMLERİ

A/B Test: A/B testleri ile bir sayfanın farklı versiyonlarının performansları ölçülebilir. Versiyonlar aynı sayfanın farklı tasarımları olacağı gibi, aynı tasarım içinde farklı sayfa öğelerinin yerlerinin değiştirilmesinden de oluşabilir. Bu test genellikle major ve minör şeklinde yapılır. Minör bir A/B testi örneğin "Satın Al" butonunun rengini kırmızı ve turuncu yaparak sonucu gözlemlemek iken daha major bir A/B testi ise genellikle akışları ilgilendirmektedir Şekil 2.



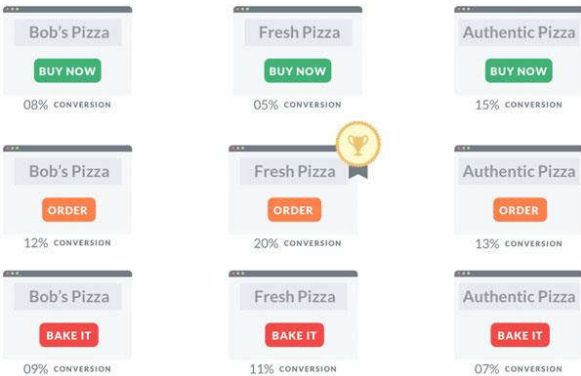
Şekil 2- A/B Testi

Split Test: Çoğunlukla landing page (Yönlendirilen Sayfa) optimizasyonunda kullanılır. Tüm sayfanın A ve B versiyonu kurgulanır. Siteye gelenlerin yarısı A, diğer yarısı ise B sayfasıyla karşılaşır. Bu testten elde edilen sonuçlar neticesinde başlangıç noktasındaki tüm aksiyon değişimlerinin oranı optimize edebilecek durumda olmaktadır Şekil 3.



Şekil 3- Split Test

Multivariate Test: A/B testine çok benzer niteliktedir; fakat tek farkı varyasyonların tek aksiyon içerisinde daha farklı unsurlarla test edilmesi ile ux (Kullanıcı Deneyimi) optimizasyonu sağlanmaktadır Şekil 4.



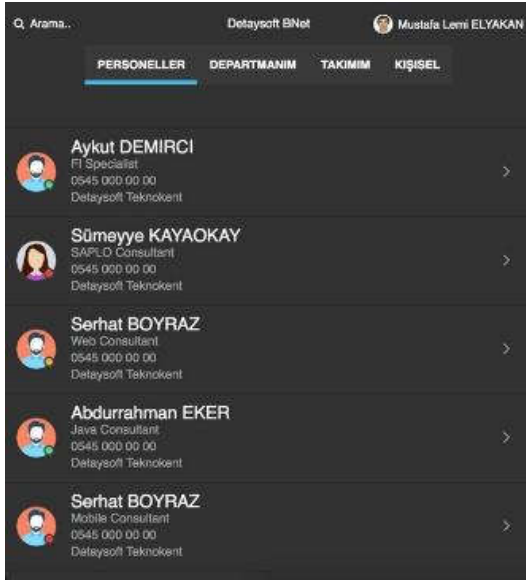
Şekil 4- Multivariate Test

4-SONUÇ VE TARTIŞMALAR

Kurumsal bir yazılım firmasının personeller üzerinde iş ve plan takibini yapabildiği bir uygulamada kullanılabilirlik testleri ile kullanıcı deneyimleri dikkate alınarak arayüz tasarımlarında farklı prototipler ile iyileştirmeler sağlanmaya çalışılmıştır. Kullanılan farklı frameworklerin performansları ise bir takım görevlerin yerine getirilmesinde zaman farklarının oluşmasına neden olmuştur. Aşağıdaki şekillerde Şekil 6-11 uyarlanan son prototipe ait ekran görüntüleri ve test sonuçları yer almaktadır. Şekil 5'te kullanıcılara yönlendirilen iş oluşturma görevine ait istatistiksel sonuçlar yer almaktadır. İlgili görev sekiz kullanıcı üzerinde deneyerek direct success, indirect success, direct failure, indirect failure, direct skip ve indirect skip gibi bir takım görev tamamlama tiplerine ait bilgiler elde edilmeye çalışılmıştır. Direct success, kişinin görevi hiç bir aksaklık olmadan doğrudan başarılı bir şekilde tamamlamasını, indirect success görevi oluştururken farklı ekran görünüşleri arasında gezinme yaptıktan sonra doğru yolu bulmasını, direct failure görevi direct hatalı olarak tamamlamasını, indirect failure başlangıçta doğru yolu bulup da sonradan görevi yanlış olarak tamamlamasını, direct skip görevi hiç tamamlamadan doğrudan test aşamasından çıkmasını, indirect skip ise görevin bir kısmını gerçekleştirip de tam olarak istenilen çalışmayı yerine getirmeden testi bırakmasını ifade etmektedir.

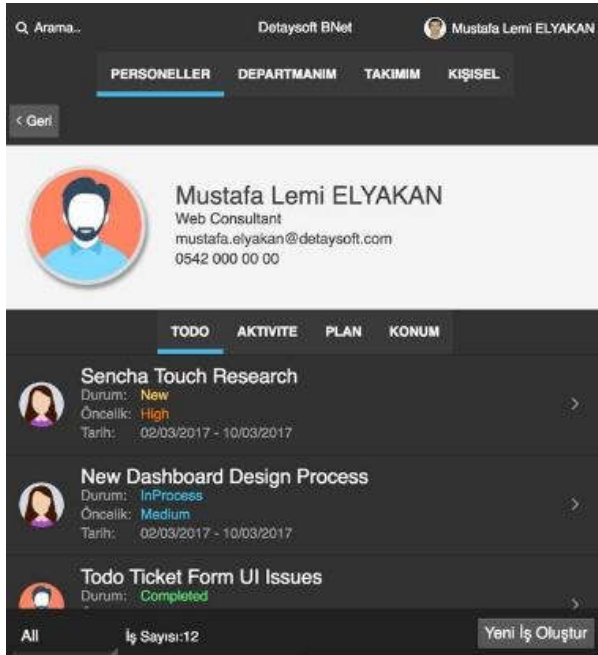


Şekil 5-Göreve Ait Test Sonuçları



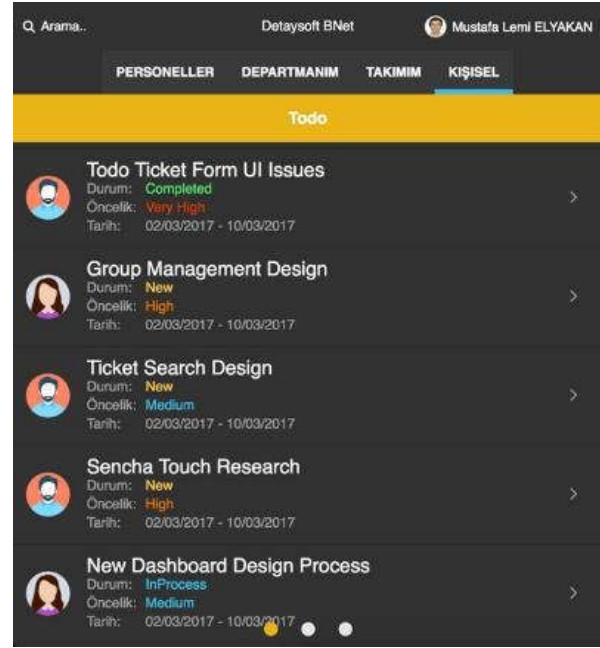
Şekil 6- Personel Listesi

Şekil 6 kullanıcının sisteme ilk giriş yaptığında karşılaştığı görüntüdür. Bu ekran üzerinde kullanıcı çalışma arkadaşlarına ait kişisel bilgileri ve aynı zamanda liste üzerindeki herhangi bir kişiye tıklayarak personelin işlerini, planlarını, günlük aktivitelerini ve konumlarını da görüntüleyebilmektedir.



Şekil 7- Personel Bilgileri

Şekil 7' deki görünüm personelin iş durumlarına ve önceliğine, planına, aktivitesine ve konumuna ait bilgileri içermektedir. Bu ekran üzerinde kullanıcılar "Yeni İş Oluştur" butonuna tıklayarak ilgili kişiye doğrudan iş atayabilmektedir.



Şekil 8- Personelin Kişisel Görevleri

Şekil 8' deki görünüm ise personelin kişisel işlerini, planlarını ve aktivitelerini görüntüleyebildiği bölümdür. Bu ekran üzerinde kullanıcı herhangi bir göreve tıklayarak anlık olarak üzerinde düzenleme yapabilmekte ve işi atayan personele de bildirim olarak düzenleme yapıldığına dair bilgilendirme mesajı gönderilmektedir.



Şekil 9- ReactJS'de Viewlar Arası Yönlendirmeler

```
import React, { Component } from 'react';
import './Button.css';

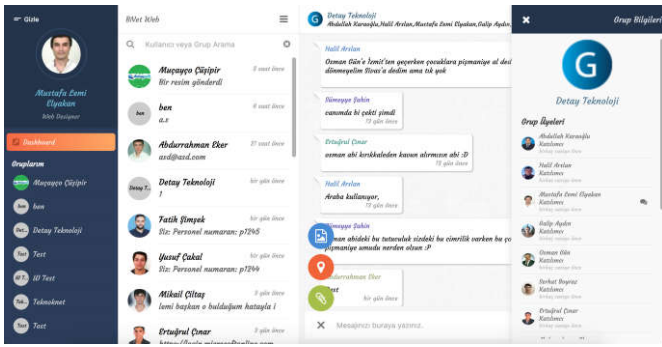
const classNames = require('classnames');

export class Button extends Component {
  render() {
    return (
      <button
        ref={this.props.buttonRef}
        className={classNames('rce-button', this.props.type, this.props.className)}
        style={{
          backgroundColor: this.props.backgroundColor,
          color: this.props.color,
          borderColor: this.props.backgroundColor
        }}
        disabled={this.props.disabled}
        onClick={this.props.onClick}>
        {this.props.icon ?
          <span className="rce-button-icon-container">
            {(this.props.icon.float === 'right' || !this.props.icon.float) ?
              <span style={{ float: this.props.icon.float, fontSize: this.props.icon.size }} className="rce-button-
              <span style={{ float: this.props.icon.float === 'left' ?
                <span>{this.props.text}</span>
              </span>
            </button>
          );
        };

Button.defaultProps = {
  text: '',
  disabled: false,
  type: null,
  icon: null,
  backgroundColor: '#9979aa',
  color: 'white',
  className: null,
  buttonRef: null,
};

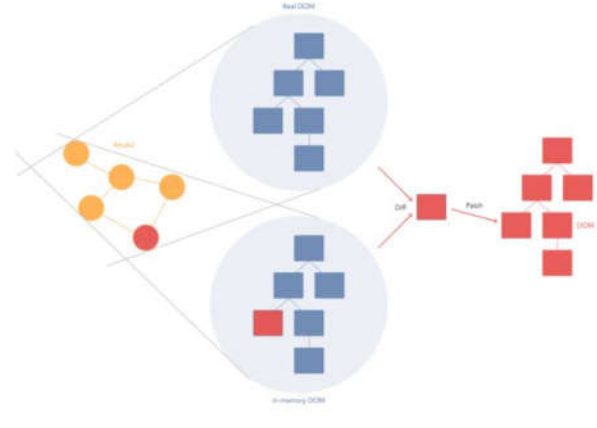
export default Button;
```

Şekil 10- ReactJS'de Component Oluşturma



Şekil 11- ReactJS ile Hazırlanan Uygulama Arayüzü

Şekil 9,10 ve 11' de ReactJS Javascript framework'ü ile uygulamanın view yapısına ve arayüz görünümüne ait görsellere yer verilmiştir. Uygulamada kullanılan componentler (bileşenler, örn: Button,Dropdown,...) javascript ve html kullanılarak belirli özellikler verilerek hazırlanmıştır. Şekil 10' da "Button" bileşeninin oluşturulmasına ait kod yapısı bulunmaktadır. ReactJS' de arayüzde kullanılacak olan componentler (bileşenler) gerçekleştirecek oldukları işlevlerine göre özelleştirilip kullanılabilir. ReactJS' in performansına etki eden en önemli sebeplerden birisi Virtual DOM (Sanal Döküman Nesne Modeli) olarak adlandırılan sanal bir mekanizma kullanmasıdır. Virtual DOM, gerçek DOM'a ait render edilen yapının bir kopyasını oluşturmaktadır. Bu sayede herhangi bir element üzerinde değişiklik yapıldığı zaman ReactJS gerçek DOM'u tekrar bütünüyle derlemez.Ancak, Sanal DOM ve gerçek DOM arasındaki farklılıkları tespit ederek sadece değişen elemanları yeniden render etmektedir. Şekil 12' de bu işlemlerin gerçekleşme adımları anlatılmaya çalışılmıştır.



Şekil 12- ReactJS Gerçek DOM ve Sanal DOM Yapısı

Uygulamalar geliştirilirken sistemin performansına hem kullanılacak olan frameworklerin çeşidi hem de yazılan kodların verimliliği etki etmektedir. Bir uygulama farklı yazılım geliştiricileri tarafından birçok farklı şekilde kodlanabilir. Yazılım dünyasında gün geçtikçe yeni teknolojiler ortaya çıkmakta ve daha önce kullanılan eski sistemlerin, gerektiği durumlarda güncellenmesine sebep olmaktadır.

5-KAYNAKLAR

- Bastien, JM Christian. "Usability testing: a review of some methodological and technical aspects of the method." *International journal of medical informatics* 79.4 (2010): e18-e23.
- Coursaris, Constantinos K., and Dan J. Kim. "A meta-analytical review of empirical mobile usability studies." *Journal of usability studies* 6.3 (2011): 117-171.
- Dumas JS. User-based evaluations. In: Jacko JA, Sears A, editors. *The Human-Computer Interaction Handbook*. Mahwah, New Jersey: Lawrence Erlbaum Associates; 2003, p. 1093-1117.
- J.M. Christian Bastien, "Usability Testing Some Current Practices and Research Questions" UFR Sciences Humaines & Arts 57006 Metz, France
- Klaus Hinum, *Human centred Design for Graphical User Interfaces*. Dieselgasse 5a/13 A-1100 Vienna Austria
- Myers, Brad A., and Mary Beth Rosson. "Survey on user interface programming." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1992.
- Rosenbaum S, Rohn JA, Humburg J. A toolkit for strategic usability: results from workshops, panels, and surveys. In: Little R, Nigay L, editors. *Proceedings of*

the ACM CHI 2000 Conference on Human Factors in Computing Systems. New York: ACM Press; 2000, p. 337-344.

Spool J, Schroeder W. Testing Web sites: five users is nowhere near enough. In: Proceedings of the Conference extended abstracts on Human Factors in Computing Systems, CHI'2001. New York: ACM Press; 2001.

Tullis T, Albert B. Measuring the user experience. Amsterdam: Morgan Kaufman; 2008.

[1]<https://www.mediaclick.com.tr/blog/framework-nedir>

[2]<https://www.sencha.com/products/touch/#overview>

[3] <https://docs.angularjs.org/guide/introduction>

[4]<http://www.seonet.com.tr/blog/web-tasarim-terimleri/angularjs-nedir->

[5] <http://blog.cronom.com/reactjs-nedir>