



Beykent Üniversitesi  
Yazılım Mühendisliği  
Yazılım Mühendisliği Bitirme Projesi  
2020 – BAHAR  
**Birinci Raporu**

Benimki Nerede? (BN)

Abdurrahman Aydın - 160313020

Ahmet Burak Hapaz - 160313067

Muhammed Burak Sıvık - 160313003

Uğurcan Çakar - 160313024

Utku Erdemir - 160313037

# İçindekiler

<b>1 Benimki Nerede? .....</b>	<b>3</b>
1 a. Projenin Açıklaması.....	3
1 b. Proje Ekibi .....	3
1 c. Projenin Aşamaları .....	4
<b>2 Dönem İçinde Yapılan İşler .....</b>	<b>5</b>
2 a. Seçilmiş Platform Testlerinin Yapılması (Bütün Grup Üyeleri) .....	5
2 b. Veri Tabanının Oluşturulması (Ahmet Burak Hapaz, Utku Erdemir, Abdurrahman Aydın).....	5
2 c. Arka Uç Geliştirme (Abdurrahman Aydın, Utku Erdemir).....	6
2 d. Mobil Ön Uç Biriminin Tasarlanması (Uğurcan Çakar, Utku Erdemir).....	7
2 e. Web Ön Uç Geliştirme (Muhammed Burak Sıvık, Ahmet Burak Hapaz, Abdurrahman Aydın, Utku Erdemir).....	9
<b>3 Sonuç .....</b>	<b>11</b>
3 a. Başarı İle Tamamlanan İşler .....	11
3 b. Başarı İle Tamamlanamayan İşler .....	11
3 c. Kişilerin Görevlerini Yerine Getirme Durumu.....	11

# Yazılım Mühendisliği Bitirme Projesi Grup10 1. Raporu

## 1 Benimki Nerede?

### 1 a. Projenin Açıklaması

Projenin temel amacı kişiler için önemli olan varlıkların kaybolmasını önlemek, kaybolan varlıkların bulunmasına yardımcı olmak ve günümüz koşulları nedeniyle depremde enkaz altında kalan insanların bulunmasına yardım etmektedir. Projenin asıl amaçlarının haricinde bir artısı kullanıcı dostu olmaktır. Çünkü proje içerisindeki ara yüzler, teknolojilerin etkileşime geçmesi ve alış veriş sistemi mümkün olduğunca en basit düzeye indirgenmiştir. Bu işlemler yapılırken Beacon cihazlardan yararlanılmaktadır, mobil uygulamanın geliştirilmesinde tek kod platformu olarak React-Native kullanılmaktadır, web ön uç birimi için ReactJS kullanılmaktadır. Arka uç birimde son dönemlerde yüksek performansından ötürü GoLang ve veri yoğunluğunu kaldırabilmesi için MongoDB kullanılmaktadır.

### 1 b. Proje Ekibi

Tablo – 1: Proje Ekibi

	Numarası	Adı Soyadı	Projedeki Görev Tanımı
1	160313020	Abdurrahman Aydın	<ul style="list-style-type: none"><li>Bütün arka uç biriminin geliştirmektir.</li><li>Web ön uç biriminin geliştirilmesine yardım etmektedir.</li><li>Veri tabanının oluşturulmasına yardım etmektedir.</li></ul>
2	160313067	Ahmet Burak Hapaz	<ul style="list-style-type: none"><li>Veri tabanında ki ilişkileri belirlemek ve yapıyı oluşturmaktır.</li><li>Web ön uç biriminin geliştirilmesine yardım etmektedir.</li></ul>
3	160313003	Muhammed Burak Sıvık	<ul style="list-style-type: none"><li>Web ön uç birimini geliştirmektir.</li><li>Bütün kullanıcı ara yüzlerini ve grafiksel gereksinimleri tasarlamaktır.</li></ul>
4	160313024	Uğurcan Çakar	<ul style="list-style-type: none"><li>Mobil ön uç birimini geliştirmektir.</li><li>Ön uç birimler için durum yönetimini geliştirmektir.</li></ul>
5	160313037	Utku Erdemir	<ul style="list-style-type: none"><li>Proje ekibinin koordinasyonunu ve proje için gerekli kaynakların bulunmasını sağlamaktır.</li><li>Web ön uç biriminin geliştirilmesinde karşılaşılan problemlere çözüm bulmaktır.</li><li>Arka uç birimin geliştirilmesinde karşılaşılan problemlere çözüm bulmaktır.</li><li>Mobil ön uç biriminin genel yapısını oluşturmak ve gerekli platformların kurmaktır.</li><li>Veri tabanının oluşturulmasına yardım etmektedir.</li></ul>

## 1 c. Projenin Aşamaları

Proje içerisinde yaklaşık on dört tane aşama ön görülmüştür ve bu aşamalardan yaklaşık dört tanesi geçtiğimiz dönem, dört ve beş tanesi de bu dönem içerisinde tamamlanmıştır.

**Tablo – 2: İş Paketleri**

No	İş Paketi Adı	Dönem	İçeriği
1	Planlama	12.10.2019 - 08.11.2019	Projenin hatlarının belirlendiği aşamadır.
2	Analiz	08.11.2019 - 27.11.2019	Projede bulunan kullanım durum, fonksiyonel gereksinimler ve kısıtların belirlendiği aşamadır.
3	Tasarım	27.11.2019 - 27.12.2019	Projenin veri tabanının, ara yüzlerinin ve akış senaryolarının belirlendiği aşamadır.
4	Test Planları Oluşturulması	27.11.2019 - 22.01.2020	Projenin bitimindeki test senaryolarının oluşturulduğu aşamadır.
5	Seçilmiş Platform Testlerinin Yapılması	11.02.2020 - 25.02.2020	Projenin gereksinimlerine uygun belirlenmiş platformların kullanım testlerinin gerçekleştirildiği aşamadır.
6	Veri Tabanının Oluşturulması	18.02.2020 - 29.02.2020	Proje için verilerin barındırılacak olduğu veri tabanının temel yapısının oluşturulmasını içeren aşamadır.
7	Arka Uç Birimin Geliştirilmesi	19.02.2020 - 30.03.2020	Kullanıcının ön uçta yaptığı işlemleri veri tabanına belirli kontroller ile kaydedecek olan birimin geliştirildiği aşamadır.
8	Mobil Ön Uç Biriminin Tasarlanması	20.02.2020 - 15.03.2020	Projenin asıl kısmı olan Beacon tarama kısmını oluşturan mobil uygulamanın daha önceden belirlenmiş ara yüzler referans alınarak tasarlandığı kısımdır.
9	Web Ön Uç Biriminin Tasarlanması	25.02.2020 - 17.03.2020	Kullanıcıların ve yöneticilerin kullanacağı web panelinin oluşturulduğu sayfadır.
10	Arka Uç Birimi İle Ön Uç Birimlerinin Bağlanması	30.03.2020 - 04.05.2020	Tasarımlar bittikten sonra daha önceden hazırlanmış olan arka uç birimiyle ön uç birimin bağlanmasını içeren aşamadır.
11	Test Planlarının Gerçekleştirilmesi	04.05.2020 - 12.05.2020	Belirlenmiş olan test planlarının uygulandığı aşamalardır.
12	Karşılaşılan Sorunların Çözülmesi	12.05.2020 - 20.05.2020	Test planları gerçekleştirildikten sonra hata ile karşılaşılan yerlerin çözüme kavuşturulması aşamasıdır.
13	İyileştirme Çalışmaları	20.05.2020 - 27.05.2020	Proje içerisinde kullanıcı deneyimi, performans/kaynak kullanımı açısından iyileştirme çalışmalarını içeren aşamadır.
14	Yayına Alınma	27.05.2020 - 30.05.2020	Projenin internet sitesi üzerinden ve mobil marketlerden yayınlanma aşamasıdır.

## 2 Dönem İçinde Yapılan İşler

### 2 a. Seçilmiş Platform Testlerinin Yapılması (Bütün Grup Üyeleri)

Bu kısımda genel olarak kullanılması düşünülen platformların kurulumu ve bu platform içinde gerekli olduğu düşünülen kütüphanelerin testleri yapılmıştır. Örneğin başta MongoDB düşünülmemiş lakin bu testlerin ardından MongoDB'ye geçilmiştir. ReactJS'in saf haliyle yola çıkılmış lakin sunucu tarafında ki işlemenin arama motoru optimizasyonu konusunda daha mantıklı olacağını kanısına varılmış ve NextJS kullanılmasında karar kılınmıştır. GoLang üzerinde Mogo ve Bongo arasında seçim bu aşamada yapılmıştır. React-Native için gerekli yan menü ve Beacon Manager testi bu aşamada gerçekleştirilmiştir.

### 2 b. Veri Tabanının Oluşturulması (Ahmet Burak Hapaz, Utku Erdemir, Abdurrahman Aydın)

Bu bölümde veritabanı tasarımı yapılmıştır. Başta ilişkisel bir veritabanı düşünülmüş olsa da çok fazla veri olacağından ötürü platform olarak doküman tabanlı bir veri tabanı olan MongoDB'ye geçilmiştir. Veri tabanı, koleksiyonlar ve içinde tutulacağı veri yapıları oluşturulmuştur. MongoDB'de koleksiyonlar oluştururken koleksiyon adları boşluk içerilerek oluşturulabildiği için komut satırında ve web servislerde kullanılırken önemli bir sorun yaratmıştır. Sorun komut satırında görünüp düzeltilmiştir. MongoDB'nin kendi sitesindeki kaynaklar başta olmak üzere birkaç siteden veri tabanının kullanımı ve işleyişi öğrenilmiştir. Veri tabanında standart olarak normalleştirme (normalizasyon) işlemleri tercih edilmemiştir. Çünkü veri tabanına bağlanıldığında, her bir koleksiyon dosyasını açarken veri tabanına yeni bir bağlantı oluşturulmaktadır. Birleştirme (JOIN) işlemlerinde başka bir koleksiyonla ilişkilendirilmesi performansı düşürebileceğinden ötürü normalleştirilmiş verilerden ziyade koleksiyonlarda benzer verilerin tutulması öngörülmüştür.

```
{
  "user_id": "5e4d83a99510103a245aec83",
  "order_id": "xxxxxxxxxxxxxxxx",
  "order_status": {
    "$numberInt": "0"
  },
  "order_date": {
    "$date": {
      "$numberLong": "946674000000"
    }
  },
  "orders": [{
    "product_id": {
      "$numberInt": "1213213"
    },
    "product_name": "beaconXXX",
    "product_price": {
      "$numberDouble": "0.95"
    }
  }
}
```

```

    }],
    "payment_type": "CARDXXX",
    "total_price": {
        "$numberDouble": "3.1"
    },
    "contact_info": {
        "user_real_name": "xxxxxxxxxxxxxx",
        "user_surname": "xxxxxxxxxxxxxxxxxx",
        "user_address": "xxxx 07 xxxx a CRXXX",
        "user_phone": "+90533345646546"
    }
}

```

### Kod Örneği – 1: Veri Tabanının Oluşturulması / Orders.json

## 2 c. Arka Uç Geliştirme (Abdurrahman Aydın, Utku Erdemir)

Bu bölümde Arka uç kısmında kullanılan Golang'ın kurulumu ve testleri yapılmıştır. Platform olarak Golang platformu seçilmiştir. Bu platform da öncelikle MongoDB için oluşturulan birkaç kütüphane test edilmiş en son olarak Bongo kütüphanesinde karar kılınmıştır. Daha sonra platformun HTTP Request yapısı test edilmiştir, bu test sürecinde girdi ve çıktı yöntemleri ve HTTP methodları test edilmiştir. Golang tarafında karşılaşılan başlıcan en önemli sorun Golang'ın bize sunduğu paket hiyerarşisi olmuştur. Projede kullanılan kaynaklar Golang'ın bize hali hazırda sunduğu dökümanlar ve aynı şekilde diğer kullanılan kütüphanelerin bizlere sunduğu dökümanlar olmuştur. Projede uyulan standartlar genel olarak SOLID prensipleri ve controller ile modeller ayrıştırmak olmuştur. Bu sayede daha okunabilir ve günümüz standartlarına daha uygun kodlar yazılmıştır.

```

package main
import (
    "encoding/json"
    "fmt"
    "net/http"
    "github.com/globalsign/mgo/bson"
)
func login(w http.ResponseWriter, r *http.Request) {
    if r.Method == "POST" {
        var user = findDB(r.FormValue("id"), r.FormValue("pass"))
        if user == nil {
            fmt.Println("error:", user)
            w.Header().Add("Content-Type", "application/json")
            w.Write([]byte(`{"error": true }`))
        } else {
            w.Header().Add("Content-Type", "application/json")
            w.Write([]byte(string(user)))
        }
    }
}

```

```

    } else {
        w.Header().Add("Content-Type", "application/json")
        w.Write([]byte(`{"error": true }`))
    }
}
func findDB(userName string, userPass string) []byte {
    person := &Person{}
    var data []byte
    result := connection.Collection("users").FindOne(bson.M{"user_info.user_name":
userName, "user_info.user_password": userPass}, person)
    if result != nil {
        fmt.Println(result.Error())
    } else {
        person.UserInfos.UserToken = tokenGenerator()
        connection.Collection("users").Save(person)
        user := &Userjon{person.UserInfos.UserToken, false}
        data, _ := json.Marshal(user)
        return data
    }
    return data
}

```

### Kod Örneği – 1: Arka Uç / Login.go

## 2 d. Mobil Ön Uç Biriminin Tasarlanması (Uğurcan Çakar, Utku Erdemir)

Bu bölümde Mobil Ön uç kısmında kullanılan platformun kurulumu ve testleri yapılmıştır. Platform olarak React Native platformu seçilmiştir. Gerek tek platformda yazılan kodların işletim sistemi farketmeksizin çalışması ve gerek öğrenim kolaylığından dolayı platform olarak React Native'i seçilmesinde karar kılınmıştır. Sonrasında ise gerekli kütüphaneler incelenmiştir ve projeye dahil edilmiştir. Projede UI kütüphanesi olarak UIKitten, Iconlar için Eva-Icons, kredi kartı işlemleri için Awesome Card IO, menü için React Native Drawer, resim yükleyebilmek için React Native Image Picker, durum yönetimi için Redux, ekran kaydırma için KeyboardAwareScrollView ve uyumlu tasarım için Responsive Dimensions kütüphaneleri kullanılmıştır. Projede yaşanan önemli sorunlar, kütüphaneler kurulurken yaşanan sıkıntılar ve kurulan kütüphanelerin eski sürümlerde çalışacak durumda olmasıdır. En büyük sıkıntı Beacon Manager kütüphanesinin React Native tarafında çalışacak durumda olmayışıydı. Bu nedenle Android Studio tarafında ara katmanlar yazılması gerekiyordu ve kütüphanenin büyük bir kısmı düzenlenmiştir. Aynı şekilde diğer pek çok kütüphanenin de doğrudan kullanılacak durumda olmayışı platforma bağlı olan kodlarda değişiklik yapmayı gerektirmiştir. Projede kullanılan kaynaklar, geliştiriciler tarafından sağlanan dökümanlar olmuştur. Uyulan standartlar ise Camel Case ve Redux tasarım şablonu ve SOLID prensipleri olmuştur. Kullanım amacı geliştiricilerin kullandığı genel standartları yakalayarak, günümüze uygun ve güzel kodlar yazılmasıdır.

```

import Styles from './Styles';
import React, {Component} from 'react';
import {Image} from 'react-native';
import {Button, Input, Layout} from '@ui-kitten/components';
import { responsiveHeight, responsiveWidth } from "react-native-responsive-dimensions";
import { KeyboardAwareScrollView } from 'react-native-keyboard-aware-scroll-view';

```

```

export default class DeviceDetail extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name: 'Uğur',
      surname: 'Çakar',
      email: 'ugurcakar@hotmail.com',
      phone: '5555555555',
      password: 'baguvix',
      passwordVisible: false,
    };
  }
  onPasswordIconPress = () => {
    this.setState({
      passwordVisible: !this.state.passwordVisible,
    });
  };
  setEmail = () => {};
  render() {
    return (
      <KeyboardAwareScrollView style={Styles.container}>
        <Image style={{height:responsiveHeight(30),width:responsiveWidth(100),
justifyContent:"center", alignItems: "center",resizeMode: 'contain'}}
          source={{uri: 'url', }}
        />
        <Layout style={Styles.formContainer} level="1">
          <Input
            style={Styles.input}
            value={this.state.name}
            label='Cihaz adı'
            labelStyle={Styles.customizeLabelStyle}
            textStyle={Styles.customizeTextStyle}
            disabled={true}
          />
          <Input
            style={Styles.input}
            value={this.state.name}
            label='Türü'
            labelStyle={Styles.customizeLabelStyle}
            textStyle={Styles.customizeTextStyle}
            disabled={true}
          />
          <Input
            style={Styles.input}

```



```

        value={this.state.name}
        label='Güven aralığı'
        labelStyle={Styles.customizeLabelStyle}
        textStyle={Styles.customizeTextStyle}
        disabled={true}
      />
      <Input
        style={Styles.input}
        value={this.state.name}
        label='Batarya seviyesi'
        labelStyle={Styles.customizeLabelStyle}
        textStyle={Styles.customizeTextStyle}
        disabled={true}
      />
      <Input
        style={Styles.input}
        value={this.state.name}
        label='Cihazın güncel uzaklığı'
        labelStyle={Styles.customizeLabelStyle}
        textStyle={Styles.customizeTextStyle}
        disabled={true}
      />
    </Layout>
    <Button style={Styles.save} size="giant">
      Edit
    </Button>
  </KeyboardAwareScrollView>
);
}
}

```

**Kod Örneği – 1: Mobil Ön Uç Biriminin Tasarlanması / DeviceDetail.js**

## **2 e. Web Ön Uç Geliştirme (Muhammed Burak Sıvık, Ahmet Burak Hapaz, Abdurrahman Aydın, Utku Erdemir)**

Bu bölümde Web Ön uç kısmında kullanılan platformun kurulumu ve testleri yapılmıştır. Platform olarak React.js platformu seçilmiştir. Bu platform içindeki çeşitli Framework kütüphaneleri denenmiş olup, en sonunda Next.js Framework’u seçilmiştir. Sonrasında ise farklı amaçlarla kullanmak için çeşitli kütüphaneler incelenmiş olup, proje de API’yi bağlamak için Axios kütüphanesi, durum yönetimi amacı ile de Redux kütüphanesi seçilmiştir. Projede yaşanan başlıca sorunlar, kütüphaneler kurulurken ve test edilirken yaşanan hatalar, kullanımdan kaldırılan kütüphanelerden dolayı yaşanan sorunlar, tasarım ile ilgili alınan hatalar olarak gösterilebilir. Projede kullanılan kaynaklar, genellikle kullanılan kütüphanelerin geliştiricileri tarafından hazırlanan dökümanlar olmuştur. Bunların yanı sıra çeşitli geliştiricilerin, açık kaynaklı olarak paylaştığı dökümanlar, çözümler ve kodlar yegane kaynaklar arasında olmuştur. Uyulan standartları göz önüne alınacak olursa, kullanılan Redux kütüphaneleri ve çoğu kütüphanelerin yapıları, uyduğumuz en önemli standartlardan olmuştur. Kullanılan Camel Case

yazım standardı da, yazılan kodların biz geliştiriciler dahil, gören herkesin kolayca anlayabilmesi ve kod karışıklığının önlenmesi açısından uyulması gereken önemli standartlardan birisi olmuştur.

```
import { Button, Input } from 'antd';
import ProfileCard from './shared/ProfileCard';
import { Edit } from 'react-feather';
const ProductOne = () => (
  <div>
    <ProfileCard name={'John Doe'} avatar="/static/images/face1.jpg"
images={['/static/images/unsplash/4.jpg', '/static/images/unsplash/6.jpg']} location={'London, Uk'}
imageHeight={230}/>
    <div style={{ padding:'20px' }}>
      <Input style={{ marginBottom:'20px',width:'365px',marginRight:'10px'}} placeholder="İsim" />
      <Button shape="circle" type="primary" size="large" className="phone">
        <Edit size={16} strokeWidth={1} />
      </Button>
      <Input style={{ marginBottom:'20px' , width:'365px', marginRight:'10px'}}
placeholder="Soyisim " />
      <Button shape="circle" type="primary" size="large" className="phone">
        <Edit size={16} strokeWidth={1} />
      </Button>
      <Input style={{ marginBottom:'20px' , width:'365px', marginRight:'10px'}}
placeholder="Telefon " />
      <Button shape="circle" type="primary" size="large" className="phone">
        <Edit size={16} strokeWidth={1} />
      </Button>
      <Input style={{ marginBottom:'20px' , width:'365px', marginRight:'10px'}}
placeholder="Adres " />
      <Button shape="circle" type="primary" size="large" className="phone">
        <Edit size={16} strokeWidth={1} />
      </Button>
    </div>
    <div className="text-center mt-3 mb-3">
      <Button type="primary" float='center'>DEĞİŞİKLİKLERİ KAYDET</Button>
    </div>
  </div>
);
export default ProductOne;
```

## Kod Örneği – 2: Web Ön Uç Geliştirme / ProductOne.js

### 3 Sonuç

#### 3 a. Başarı İle Tamamlanan İşler

Bugüne kadar altıncı numaralı iş paketine kadar olan kısımların hepsi başarıyla tamamlanmıştır. Yedinci, sekizinci ve dokuzuncu paketin büyük bir kısmı bitmiştir. Onuncu pakete geçilmeden önce platform testleri sırasında gerekli altyapı kurulmuştur.

#### 3 b. Başarı İle Tamamlanamayan İşler

Planlanan bütün işler başarıyla gerçekleşmiştir. Sadece veri tabanı oluşturulması sırasında başlangıçta ilişkisel veri tabanı düşünülmüş fakat ardından veri hacminin büyük olabileceği ve ilişkileri oluşturmayla zaman kaybetmemek için doküman tabanlı bir veri tabanına geçilmiştir.

#### 3 c. Kişilerin Görevlerini Yerine Getirme Durumu

**Tablo – 3: Proje Ekibinin Görevlerini Yerine Getirme Durumu**

	Numarası	Adı Soyadı	Projedeki Görevini Yerine Getirme Durumu
1	160313020	Abdurrahman Aydın	<ul style="list-style-type: none"><li>Veri tabanı tasarımına yardım etmiştir. (%100)</li><li>Web servislerin istenilen kısmını geliştirmiştir. (%100)</li><li>Web uygulaması kısmında kendine verilen görevleri yerine getirmiştir. (%100)</li></ul>
2	160313067	Ahmet Burak Hapaz	<ul style="list-style-type: none"><li>Veri tabanının tasarımında büyük rol oynamıştır. (%100)</li><li>Web uygulaması kısmında kendine verilen görevi yerine getirmiştir. (%100)</li></ul>
3	160313003	Muhammed Burak Sıvık	<ul style="list-style-type: none"><li>Kendinden istenen ara yüzleri ve grafikleri tasarlamıştır. (%100)</li><li>Web uygulamasının geliştirilmesinde beklenen yere kadar gelmiştir. (%100)</li></ul>
4	160313024	Uğurcan Çakar	<ul style="list-style-type: none"><li>Mobil uygulamanın geliştirilmesinde kendine verilen görevi yerine getirmiştir. (%100)</li><li>Durum yönetimini hazırlamıştır. (%100)</li></ul>
5	160313037	Utku Erdemir	<ul style="list-style-type: none"><li>Web ön uç biriminin geliştirilmesinde karşılaşılan problemlere çözüm bulmuştur. (%100)</li><li>Arka uç birimin geliştirilmesinde karşılaşılan problemlere çözüm bulmuştur. (%100)</li><li>Mobil ön uç biriminin genel yapısını oluşturmuştur ve gerekli platformlarını kurmuştur. (%100)</li><li>Veri tabanının oluşturulmasına yardım etmiştir. (%100)</li></ul>

Not: Görevleri yerine getirme oranı günümüze kadar olan kendine verilmiş görevleri yerine getirme oranı olarak varsayılmıştır.