



**T.C.
Kocaeli Üniversitesi
Teknoloji Fakültesi
Bilişim Sistemleri Mühendisliği**

**ÖĞRENCİLERİN BAŞVURULARI İÇİN MOBİL KAYIT
TOPLAMA PLATFORMU (SİSTEMİ) GELİŞTİRME**

**Emre Alkan
Mert Bulut
Mehmet Hüseyin Somuncu
Gizem Coşkun**

**2021-2022 GÜZ DÖNEMİ YAZLIM LABORATUVARI-I DERSİ
PROJE - I**

DANIŞMAN-ÖĞRETİM GÖREVLİSİ

**Arş. Gör. Sümeyya İLKİN
Dr. Öğr. Üyesi Önder YAKUT
Öğr. Gör. Alper METİN**

PROBLEM TANIMI

Mobil platformda bir uygulama geliştirilecektir. Mobil uygulamanın veritabanı olacak ve ilgili veritabanı NoSQL ile oluşturulacaktır. Ayrıca veritabanı kısmında veri yapılarında kullanılan B+ ağaç yöntemi kullanılarak hiyerarşik olarak veri kaydı yapılacak ve tutulacaktır.

1.GİRİŞ

Native (Yerel):

Native mobil uygulamalar belirli bir işletim sistemi için yapılan uygulamalardır. Andorid için yapılan uygulamalar sadece android işletim sistemlerinde, ios için yapılan uygulamalar da sadece ios işletim sistemlerinde kullanılabilir. Yani bir uygulama fikrini her iki platformda da kullanılmasını istiyorsak android geliştiriciler android için, ios geliştiriciler ios için kod yazmalıdır.

Cross Platform Geliştirme

Cross platformda uygulama geliştirmek istediğimizde farklı kod tabanlarına ihtiyaç kalmadan birden fazla işletim sistemine uygun bir uygulama yapmış oluruz.

- **Uygulama geliştirmek için kullanılabilecek diller:**

Dart - Flutter
C# - Xamarin
JavaScript - React Native, Ionic, Cordova

- **Cross Platform uygulama geliştirmenin avantajları ve dezavantajları nelerdir?**

Avantajları:

1. Aynı anda birden fazla işletim sisteminde olması daha fazla kullanıcıya erişmeyi sağlar. Bu aynı zamanda uygulamanın gelirinin de artmasını sağlar.
2. Birden fazla geliştirici olmadan tek bir kişi ile yapılabilirdiğinden maliyeti de çok yüksek olmaz.
3. Native uygulama geliştirmeye göre (belirli durumlar dışında, bu durumları karşılaştırma yaptığım kısımda belirteceğim.) daha kısa sürede yapılabilir.

Dezavantajları:

1. Performans açısından native uygulamalara göre düşüklük yaşanır daha yavaştır.
2. Kitaplık ve SDK'ların sınırlı desteğinden dolayı gerekli özelliklerin entegrasyonu çok zor olabilir.
3. Uygulamayı birden fazla platforma uyarlamak zorunda olmak çeşitli cihazlarda tasarım konusunda sorun yaratabilir. Bu da geliştirici için ekstra bir iş yaratır.

Native uygulama geliştirmek:

Android:

Java / Kotlin

IOS:

Object-C / Swift

Avantajları:

1. Uygulamaları tek bir platform için gerçekleştirdiğimiz için performans konusunda çok iyi ve hızlıdır.
2. Tek bir platforma ait olduğu için geliştiriciler ui-ux tasarımını o platformda en iyi çalışacak şekilde yaparlar ve bu da tasarımın harika olmasını dolayısıyla kullanıcının memnuniyetini garanti eder.
3. Uygulamalar cihazın işlevselliği ile uyum sağlar ve cihazın donanımıyla kolayca entegre olur. (gps, kamera, dokunmatik ekran vb.)
4. Uygulamayı google play market ya da app store gibi platformlara yüklerken onların yönergelerine uyumluluk sağlar bu da yükleme işini kolaylaştırır.
5. Uygulamayı kullanmak genellikle belirli işlevler dışında internet bağlantısı gerektirmez.

Dezavantajları:

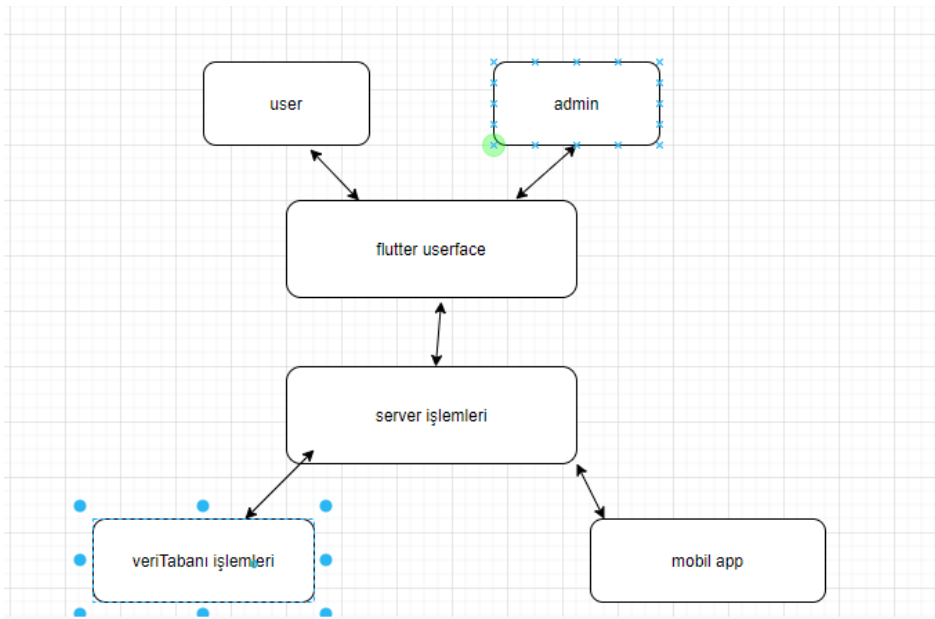
1. Uygulama tek bir platforma özgü yapılırsa bu da aynı uygulamayı farklı platformlarda farklı kod tabanları gerektirir.
2. Her platforma özel uygulama yapmak ise daha fazla geliştiriciye ihtiyaç duyar.
3. Daha fazla geliştirici daha fazla maliyet demektir. Bu yüzden native uygulama yapmak cross platforma göre bazı şartlar dışında daha maliyetlidir.

2. TASARIM

A. GENEL AKIŞ ŞEMASI



B. YAZILIM MİMARİSİ



Şekil 1: Sistemin blok şeması

Projemizde kullandığımız veritabanı NOSQL document(belge) bazlı bulut veritabanı olan Firestore veritabanıdır.

NoSql veritabanından bahsetmek gerekirse ; belirli veri modelleri için özel olarak tasarlanmıştır ve modern uygulamalar oluşturmaya yönelik esnek şemalara sahiptir. NoSQL veritabanları uygun ölçekte geliştirme kolaylığı, işlevselliği ve performansı ile geniş çaplı olarak kabul görmüştür.

NoSQL veritabanlarında, verilere erişmek ve onları yönetmek için çeşitli veri modelleri kullanılır. Bu tür veritabanları, özellikle büyük veri hacmi, düşük gecikme süresi ve esnek veri modelleri gerektiren uygulamalar için optimize edilmiştir. Bu gereksinimler, diğer veritabanlarının veri tutarlılığı kısıtlamalarının bir kısmı esnetilerek karşılanır.

Nosql Veritabanı Türleri

- 1-) Anahtar-Değer(Key-Value Pair) NoSQL
- 2- Document(Belge) NoSQL
- 3-GraphQL(Grafik) NoSQL
- 4-Bellek İçi(Amazon DynamoDB)
- 5-Arama NOSQL

İLİŞKİSEL VERİTABANI VE NOSQL VERİTABANI KARŞILAŞTIRMASI

Veri modeli : ilişkisel veritabanlarında ilişkisel model, verileri satır ve sütunlardan oluşan tablolar halinde normalleştirir. Tablolar, satırlar, sütunlar, dizinler, tablolar arasındaki ilişkiler ve diğer veritabanı öğeleri bir şema tarafından kesin bir şekilde tanımlanır. Veritabanı, tablolar arasındaki ilişkilerde başvurusal bütünlük uygular. NoSQL ise performans ve ölçek için optimize edilmiş anahtar-değer, belge ve grafik gibi çeşitli veri modelleri sağlar.

İş Modeli: İşlemsel ve güçlü tutarlılığa sahip çevrimiçi işlem gerçekleştirme (OLTP) uygulamaları için tasarlanan ilişkisel veritabanları, çevrimiçi analitik işlem (OLAP) için uygundur. NoSQL veritabanları, düşük gecikme süreli uygulamaları içeren çeşitli veri erişimi desenleri için tasarlanmıştır. NoSQL arama veritabanları, yarı yapılandırılmış veriler üzerinde analiz için tasarlanmıştır.

Performans: RDBMS için performans genellikle disk alt sistemine bağlıdır. En üst düzey performans için genellikle sorguların, dizinlerin ve tablo yapısının optimize edilmesi gerekir. NOSQL de ise temel donanımın küme boyutu, ağ gecikme süresi ve çağrı yapan uygulama gibi etmenlerin birleşimine bağlıdır.

API'ler : RDBMS için Veri depolama ve alma istekleri, yapılandırılmış sorgu diline (SQL) uygun sorgular kullanılarak iletilir. Bu sorgular ilişkisel veritabanı tarafından ayrıştırılır ve yürütülür. NoSQL ise Nesne tabanlı API'ler, uygulama geliştiricilerinin veri yapılarını kolayca depolamasına ve almasına imkan tanır. Bölüm anahtarları, uygulamaların anahtar-değer çiftlerini, sütun kümelerini veya seri hale getirilmiş uygulama nesneleri ve öznitelikleri içeren yarı yapılandırılmış belgeleri bulmasına imkan tanır.

Projemizde kullandığımız Firestore veritabanı belge tipli bir NoSql veritabanıdır.

Bu tipte bir veritabanında uygulama kodunda genellikle bir nesne veya JSON benzeri bir belgeyle temsil edilir. Belge veritabanları, geliştiricilerin uygulama kodlarında kullandıkları aynı belge modeli biçimini kullanarak bir veritabanında veri depolamasını ve sorgulamasını kolaylaştırır. Belgelerin ve belge veritabanlarının esnek, yarı yapılandırılmış ve hiyerarşik yapısı sayesinde geliştiriciler, uygulamaların gereksinimleri doğrultusunda geliştirme yapabilirler. Bu belge modeli; kataloglar, kullanıcı profilleri ve her belgenin hem benzersiz olduğu hem de zamanla geliştiği içerik yönetim sistemleri ile birlikte oldukça kullanışlıdır.

Flutter ve Cloud Firestore Bağlantısı :

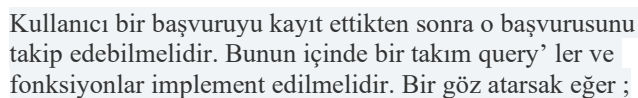
Projemizde 5 farklı başvurunun sisteme kaydedilmesi gereklidir. Bunu gerçekleştirebilmek için bazı adımları sırayla uygulamak gerekir. İlk olarak her bir başvuru için ayrı ayrı Flutter Widget'ları yardımıyla tasarımları yapıldı. Daha sonra her bir başvuru için veri modeli oluşturuldu. Örnek olarak:

```
class YatayGecisBasvuruModel {  
  String? id;  
  String? basvuruTuru;  
  String? ogrenciAd;  
  String? ogrenciTc;  
  String? ogrenciDogumTarih;  
  String? ogrenciEmail;  
  String? ogrenciGsm;  
  String? ogrenciAdres;  
  String? ogrenciUniversite;  
  String? ogrenciFakulte;  
  String? ogrenciBolum;  
  String? ogretimTuru;  
  String? ogrenciSinif;  
  String? ogrenciDisiplin;  
  String? ogrenciNotOrt; ..... }  
}
```

Daha sonra oluşturulan veri modeli , kullanıcının doldurduğu bu bilgilerin veritabanına kaydedilebilmesi için Service Classlarında yazılan fonksiyonlara parametre olarak gönderildi. Örneğin ;

```
class YatayBasvuruService {  
  final FirebaseFirestore _firestore = FirebaseFirestore.instance;  
  
  User? user = FirebaseAuth.instance.currentUser;  
  
  UserModel loggedInUser = UserModel();  
  Future<YatayGecisBasvuruModel> basvuruOlustur(String  
    basvuruTuru, ogrenciAd...) async {  
    var ref =  
      _firestore.collection("yatay_gecis_basvuru");  
    basvuruDurumu = "onay bekliyor";  
    onaylanmaTarihi = "belirsiz";  
    reddedilmeTarihi = "belirsiz";  
    var documentRef = await ref.add({  
  
      'basvuruTuru': basvuruTuru,  
      'ogrenciAd': ogrenciAd,  
      'ogrenciTc': ogrenciTc,  
      'ogrenciDogumTarih': ogrenciDogumTarih,  
      'ogrenciEmail': ogrenciEmail,  
  
    });  
  }  
}
```

Şekil 1.1 Veritabanına kaydedilen başvuru



```
Stream<QuerySnapshot> basvurulariGetir(UserModel
userModel) {
    CollectionReference collectionReference =
        _firestore.collection("yatay_gecis_basvuru");
    Query query =
        collectionReference.where("ogrenciEmail", isEqualTo:
userModel.email);
    //print("email degeri +++:" + loggedInUser.email!);
    return query.snapshots();
    //var ref =
    _firestore.collection("yatay_gecis_basvuru").snapshots();
    //return ref;
}
```

Tasarım ekranındaki bazı kodlara dikkat edersek ;

```
BorderRadius.circular(10)),
),
child: Padding(
padding: const EdgeInsets.all(5.0),
child: Column(
children: [
    SizedBox(
height: 10,
),
    Row(
children: [
        Text(
            "BAŞVURU TÜRÜ : ",
            style: TextStyle(
                fontSize: 14, color: Colors.black),
            textAlign: TextAlign.center,
        ),
        Text(
            "${basvuruListesi["basvuruTuru"]}",
            style: TextStyle(
                fontSize: 14,
                color: Colors.black,
                fontWeight: FontWeight.bold),
            textAlign: TextAlign.center,
        ), .....
    ],

```

12:30

BAŞVURU TÜRÜ : KURUM İÇİ YATAY GECİS
ÖĞRENCİ ADI : Emre Alkan
ÖĞRENCİ EMAIL : 191307077@kocaeli.edu.tr
ÖĞRENCİ TC NO : 22222222222
ÖĞRENCİ YAZIŞMA ADRESİ : İzmit/Kocaeli
ÖĞRENCİ YERLEŞME YILI : 2019
ÖĞRENCİ YERLEŞME PUANI : YKS-315
ÖĞRENCİ FAKÜLTE : Mühendislik Fakültesi
ÖĞRENCİ BÖLÜM : Bilgisayar Mühendisliği
ÖĞRETİM TÜRÜ : 1.Öğretim
ÖĞRENCİ NUMARASI : 191307077
ÖĞRENCİ SINIFI : 1
ÖĞRENCİ NOT ORTALAMASI : 3.15
BAŞVURULAN FAKÜLTE : mühendislik fakültesi
BAŞVURULAN BÖLÜMÜNÜN PUANI : YKS-340

Başvuru Durumu : REDDEDİLDİ

PDF Önizleme

Anasayfa'ya Dön

BAŞVURU TÜRÜ : EK MADDE 1 İLE YATAY GECİS
ÖĞRENCİ ADI : Emre Alkan
ÖĞRENCİ EMAIL : 191307077@kocaeli.edu.tr
ÖĞRENCİ TC NO : 22222222222
ÖĞRENCİ YAZIŞMA ADRESİ : İzmit/Kocaeli
ÖĞRENCİ YERLEŞME YILI : 2019
ÖĞRENCİ YERLEŞME PUANI : YKS-318
ÖĞRENCİ FAKÜLTE : Mühendislik Fakültesi
ÖĞRENCİ BÖLÜM : Bilgisayar Mühendisliği
ÖĞRETİM TÜRÜ : 1.Öğretim
ÖĞRENCİ NUMARASI : 191307077
ÖĞRENCİ SINIFI : 4
ÖĞRENCİ NOT ORTALAMASI : 3.15
BAŞVURULAN FAKÜLTE : İLETİŞİM FAKÜLTESİ
BAŞVURULAN BÖLÜMÜNÜN PUANI : AYT-295

Başvuru Durumu : REDDEDİLDİ

PDF Önizleme

Anasayfa'ya Dön

```
// bekleyen başvuru listeleme
Stream<QuerySnapshot> başvurulariGetirAdmin() {
    CollectionReference collectionReference =
        firestore.collection("yatay_gecis_basvuru");
```

```

    Query query =
        collectionReference.where("basvuruDurumu",
isEqualTo: "onay bekliyor");
    //print("email degeri +++:" + loggedInUser.email!);
    return query.snapshots();
    //var ref =
_firestore.collection("yatay_gecis_basvuru").snapshots();
    //return ref;
}

// onaylanan başvuruları listeleme
Stream<QuerySnapshot> onayliBasvurulariGetirAdmin() {
    CollectionReference collectionReference =
        _firebase.collection("yatay_gecis_basvuru");
    Query query =
        collectionReference.where("basvuruDurumu",
isEqualTo: "onaylandı");
    //print("email degeri +++:" + loggedInUser.email!);
    return query.snapshots();
    //var ref =
_firestore.collection("yatay_gecis_basvuru").snapshots();
    //return ref;
}

// reddedilen başvuruları listeleme
Stream<QuerySnapshot>
reddedilenBasvurulariGetirAdmin() {
    CollectionReference collectionReference =
        _firebase.collection("yatay_gecis_basvuru");
    Query query =
        collectionReference.where("basvuruDurumu",
isEqualTo: "reddedildi");
    //print("email degeri +++:" + loggedInUser.email!);
    return query.snapshots();
    //var ref =
_firestore.collection("yatay_gecis_basvuru").snapshots();
    //return ref;
}

// başvuru onaylama fonksiyonu
Future basvuruOnaylaAdmin(String selectedDoc) async {
    FirebaseFirestore.instance
        .collection("yatay_gecis_basvuru")
        .doc(selectedDoc)
        .update({
            'basvuruDurumu': "onaylandı",
            'onaylanmaTarihi': DateTime.now(),
        });
}

// başvuru reddetme fonksiyonu
Future basvuruReddetAdmin(String selectedDoc) async {
    FirebaseFirestore.instance
        .collection("yatay_gecis_basvuru")
        .doc(selectedDoc)
        .update({
            'basvuruDurumu': "reddedildi",
            'reddedilmeTarihi': DateTime.now(),
        });
}

```

1- GENEL YAPI

KOU



Kocaeli Üniversitesi Başvuru Kayıt Sistemi

Sisteme Kayıt Ol

Sisteme Giriş Yap

Admin Paneline Giriş

← Kayıt Formu

Bölüm Seçimi

Öğrenci No

Sınıf

Şifre

Şifre Tekrar

Kayıt Ol

← Kayıt Formu

GSM

Ev adresi


Varsa İş adresi

Üniversite Adı

Fakülte Seçimi

Bölüm Seçimi

← Kayıt Formu



Ad-Soyad

TC NO

Doğum Tarihi

E-mail

GSM

Ev adresi



1992

Şifremi unuttum/sıfırla: [Sıfırla](#)

Sisteme kayıt olmak için: [Kayıt Ol](#)

Admin Girişi için: [Admin Girişe git](#)

3-Öğrenimine ilişkin bilgiler

Öğrenim Türü

☐ 1 Ogretim

☐ 2 Ogretim

Sınıfı

☐ 1

☐ 2


☐ 3

☐ 4

☐ 5 ve Uzeri

YATAY GEÇİŞ EKRANLARI

10:58



1992

KOCAELİ ÜNİVERSİTESİ

YATAY GEÇİŞ BAŞVURU FORMU

1-Başvuru Türü

☐ KURUM İÇİ YATAY GECİS

☐ KURUMLAR ARASI YATAY GECİS

☐ EK MADDE 1 İLE GECİS

☐ YURT DIŞI YATAY GECİS BAŞVURUSU

2-Kişisel Bilgiler

GİZEM COSKUN

3-Öğrenimine ilişkin bilgiler

DGS BAŞVURU EKRANLARI

3-Öğrenimine ilişkin bilgiler
kou
MÜHENDİSLİK FAKÜLTESİ
Bilgisayar Muhendisliği
3-Öğrenci Başvurusuna Ait Bilgiler
GENEL NOT ORTALAMASI
191789288
KAYITLI ÜNİVERSİTEYE YERLEŞME YILI
YERLEŞİLEN PUAN TÜRÜ VE PUANI
Beyan ettiğim bilgilerin veya belgelerin gerçeğe aykırı olması veya daha önce yatay geçiş yapmış olmam halinde hakkımda cezai işlemlerin yürütüleceğini ve kaydım yapılmış olsa dahi silineceğini bildiğimi kabul ediyorum.
Ad-Soyad

GENEL NOT ORTALAMASI
191789288
KAYITLI ÜNİVERSİTEYE YERLEŞME YILI
YERLEŞİLEN PUAN TÜRÜ VE PUANI
Beyan ettiğim bilgilerin veya belgelerin gerçeğe aykırı olması veya daha önce yatay geçiş yapmış olmam halinde hakkımda cezai işlemlerin yürütüleceğini ve kaydım yapılmış olsa dahi silineceğini bildiğimi kabul ediyorum.
Ad-Soyad GİZEM COSKUN
Tarih : 05-11-2021
imza
BU BÖLÜM ÜNİVERSİTE YETKİLİ BİRİMLERİNCE DOLDURULACAKTIR
<input type="checkbox"/> BAŞVURUSU UYGUNDUR
<input type="checkbox"/> BAŞVURUSU UYGUN DEĞİLDİR
Varsa açıklamalar :
Tarih : .../.../...

11:02	11:02	11:02	11:02
			
KOCAELİ ÜNİVERSİTESİ DGS BAŞVURU FORMU			
1-Kişisel Bilgiler			
GİZEM COSKUN			
87952687456			
21-09-1998			
gizemcsk1@gmail.com			
68957412569			
kocaeli			

YAZ OKULU BAŞVURU EKRANLARI

11:05

KOCAELİ ÜNİVERSİTESİ

1992

Yaz Okulu Başvuru Formu

MÜHENDİSLİK FAKÜLTESİ Bilgisayar Muhendisliği Bölümü
191789288 numaralı gizem coskun isimli öğrencisiyim.

20.../20... Eğitim Öğretim yılı yaz öğretimi kapsamında
aşağıda bilgilerini verdiğim ders/dersleri almak istiyorum.
Kontrol listesinde belirtilen adımları tamamladım.Gereği
için arz ederim.

İmza

E-mail : gizemcsk1@gmail.com

Adres : kocaeli

GSM : 68957412569

Danışman Ad-Soyad

Başvurulan Üniversite

11:06

Başvurulan Üniversite

Yaz Okulu Başlama-Bitiş Tarihleri

Kocaeli Üniversitesi Bilgisayar Muhendisliği Bölümü'nde
sorumlu olunan ders/dersler:

Dersin adı ve kodu	T	U

Yaz öğretimi kapsamında alınacak ders/dersler

Dersin adı ve kodu	T	U

Dilekçe Ekleri

1-Yaz döneminde ders almak istenilen Üniversite ve
Kocaeli Üniversitesinin ilgili bölümlerinin, öğrencinin
Üniversiteye giriş yılındaki taban puanlarını gösteren belge

Öğrenci Danışmanı

Bölüm Başkanı

KOCAELİ ÜNİVERSİTESİ YAZ ÖĞRETİMİ ESASLARI

ESAS 1-(1) Spor Bilimleri Fakültesi dışındaki tüm akademik birimlerde okuyan öğrencilerin, diğer yükseköğretim kurumlarından yaz öğretiminde ders alabilmeleri için; Üniversiteye giriş yılı esas olmak üzere ilgili bölüm/program taban puanın en fazla 40 puan düşük olma koşulu aranır (3) Yaz okulunda diğer yükseköğretim kurumlarından alınacak dersler için öğrencinin kayıtlı olduğu ilgili birimin Bölüm/Anabilim Dalı veya Program Başkanlığının onayı gerekir.(4) Yaz okulunda Bölüm/Anabilim Dalı veya Program Başkanlığı bu konudaki değerlendirmesini; ilgili bölüm müfredatındaki benzer içeriğe veya program yeterliliğine sahip olmakla birlikte, ders(ler)in AKTS/kredi/saat değer(ler)inden herhangi birini dikkate alarak yapar

ESAS 2- (I) Öğrenciler, yaz öğretiminde derslerin AKTS/ kredi/saat değerlerine bakılmaksızın, en çok 3 ders alabilir.

ESAS 3-(I) Güz ve/veya bahar yarıyıllarında kayıt donduran öğrenciler, kayıt dondurdıkları yarıyıllara ilişkin dersleri yaz öğretiminde alamazlar

ESAS 6- (I)Yaz öğretimi kapsamında alınan ders(ler)in notu genel not ortalamasına bakılmaksızın 4,lük sistem üzerinden 2,nin altında ise öğrenci bu ders(ler)den başarısız sayılır.

ESAS 9- (I) Yaz öğretimi aynı öğretim yılına ait üçüncü bir yarıyıl değildir. Yaz öğretiminde alınan ders(ler)in notları, öğrencilerin güz ve bahar yarıyıllarındaki yarıyıl not ortalamalarını ve yarıyıllardaki derslerin koşul durumlarını etkilemez. Ancak, öğrencinin genel not ortalaması (GNO) hesaplanmasına dahil edilir.

Başvuruyu Kaydet

ÇAP BAŞVURU EKRANLARI

2-Öğrenimine ilişkin bilgiler

Kayıtlı Olduğu Programın Öğretim Türü

☐ Örgün Öğretim

☐ İkinci Öğretim

Kayıtlı Olduğu Programın Öğretim Türü

☐ Örgün Öğretim

☐ İkinci Öğretim

Kayıtlı Olduğu Programın Yarıyılı

☐ 3. Yarıyıl

☐ 5. Yarıyıl

Disiplin Cezası

☐ VAR

☐ YOK

4-ADAYIN BAŞVURDUĞU ÇAP PROGRAMINA İLİŞKİN BİLGİLER

KOCAELİ ÜNİVERSİTESİ

KOCAELİ ÜNİVERSİTESİ


ÇAP BAŞVURU FORMU

1-Kişisel Bilgiler

2-Öğrenimine ilişkin bilgiler

DERS İNTİBAK BAŞVURU EKRANLARI

12:30



KOCAELİ ÜNİVERSİTESİ Mühendislik
Fakültesi Bilgisayar Mühendisliği İNTİBAK
DİLEKÇESİ

Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü
191307077 numaralı Emre Alkan isimli öğrencisiyim.

Daha önce Üniversitesi.....
Fakültesi / Meslek Yüksek
Okulu.....Bölümünde / Programında
aldığım ve aşağıda belirttiğim ders / derslerden muaf
olmak istiyorum.
Gereğinin yapılmasını arz ederim.

İmza

Öğrenci Ad : Emre Alkan

Öğrenci No : 191307077

Bölümü : 191307077

E-mail : 191307077@kocaeli.edu.tr

12:30

Adres : İzmit/Kocaeli

GSM : 5555555555

Muaf olunmak istenen derslerin alındığı üniversite

Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü'nde
muaf olunmak istenen dersler:

Dersin adı ve kodu	T	U
Daha önce aldığım dersin		
Dersin adı ve kodu	T	U

BAŞVURU TÜRÜ : KURUM İÇİ YATAY GECİS

ÖĞRENCİ AD : gizem coskun

ÖĞRENCİ TC : 87952687456

PDF Önizleme

Anasayfaya dön

BAŞVURU TÜRÜ : KURUMLAR ARASI YATAY GECİS

ÖĞRENCİ AD : gizem coskun

ÖĞRENCİ TC : 87952687456

PDF Önizleme

Anasayfaya dön

Yan Menü



KULLANICI ADI
Kullanici@mail.com

YATAY GEÇİŞ BAŞVURU FORMU
DGS BAŞVURU FORMU
YAZ OKULU DİLEKÇESİ
İNTİBAK BAŞVURU DİLEKÇESİ
ÇAP BAŞVURUSU

KAYITLI BAŞVURULARIM

YATAY GEÇİŞ BAŞVURULARIM
İNTİBAK BAŞVURULARIM
ÇAP BAŞVURULARIM
YAZ OKULU BAŞVURULARIM
DGS BAŞVURULARIM

ÇIKIŞ YAP



Ana Sayfa





KULLANICI BİLGİLERİ

Kullanıcı Adı Soyadı :
Kullanıcı E-Mail :
Kullanıcı Telefon No :
Kullanıcı Fakülte :
Kullanıcı Bölüm :
Kullanıcı Üniversitesi :

Çıkış Yap


Kayıt Olma Ekranı - 2


Varsa İş adresi


Üniversite Adı



Fakülte Seçimi


Bölüm Seçimi


Öğrenci No

Sınıf






Şifre

Şifre Tekrar

Kayıt Ol

Kayıt Olma Ekranı


Ad-Soyad

TC NO


Doğum Tarihi

E-mail

GSM

Ev adresi

Varsa İş adresi

Üniversite Adı

Fakülte Seçimi

Bölüm Seçimi

GİRİŞ EKRANI



✉ E-Mail _____

🔒 Parola _____

GİRİŞ YAP

ŞİFRE SIFIRLA
KAYIT OL

Şifre Sıfırlama Ekranı



✉ E-Mail _____

Şifreyi sıfırla

1- KAZANIMLAR

A. Karşılaşılan Sorunlar ve Çözümleri

Başvuru yaparken kaydettiğimiz bilgileri pdf'e çevirirken birkaç gün boyunca sebebini anlayamadığımız hatalar aldık. Sebebini uzun bir süre anlayamadık ya da yanlış anladık. Hatanın çözümünü google'de araştırırken aynı zamanda da farklı arkadaşlarıma sorarak fikir alışverişi yaptık. Bunu araştırırken hatanın neden olabileceği birkaç durum bulduk. Bunlardan bazıları şöyleydi:

Girilen bilgilerde türkçe karakterler bulunmasından ötürü pdf'e çevirilememesi ve farklı bir font kullanımı, farklı font kullanılmak istenirse fontun yolunun belirlenmemesi, girdilerin doğru olmaması gibi birçok neden bulduk. Bizim hatamız ise kullanacağımız fontu, dosyada belirtirken girdilere dikkat etmemek. Kısacası birkaç gün uğraştığımız bu sorunun tek nedeni girdi hatasıydı. Bunun dışında başvuru ekranlarında bilgileri girip kayıt ederken bazı bilgiler null geliyordu. Bilgileri yazdığımız halde başvuruları görüntülediğimiz ekranda hata alıyorduk. Bunun sebebini ise başvuruları getirirken değişkenleri model, başvuruEkranı, başvuruGoruntuleEkranı ekranlarında düzgün bir sıra ile değil farklı sıralarda yazdığımızı farkettik. Hepsini aynı düzende yazdıktan sonra hatasız çalışmaya ve null olan değerlere yazdığımız değerlerin geldiğini gördük. Çözümü yukarıda belirttiğim gibi google araştırmaları, arkadaşlarımla yazdıklarımızı kontrol etmek, youtube videoları izleyerek doğru/yanlışlarımızı görmek ve farklı yöntemleri denemek gibi birçok yol deneyerek hatalarımıza çözümler bulduk.

Başvurularda yaz okulu başvuruları ve intibak başvurularını tasarlamak ve başvuruları kaydettirmek ve listelemek zor ve zaman alıcıydı. Çünkü bu başvurularda tablo halinde alanlar(fields) var örnek olarak yaz okulunda alınmak istenen 2 veya 3 farklı ders aynı şekilde intibak başvurusu yaparken birden fazla dersten muaf olunmak isteniyorsa liste halindeki bir değeri veritabanına kaydettirmek gerekiyor. Bunu gerçekleştirmek çok zorladı ve birçok hatayla karşılaştık. En sonunda çözüm olarak Firestore'un array of objects (nesne arrayi) için geliştirdiği bir methodu uyguladık.

```
{
    'sorumluOlunanDersler':
    FieldValue.arrayUnion([
        {
            "Ders Adi ve Kodu":
            _dersControllerKou1.text,
            "t": _teoriControllerKou1.text,
            "u": _uygulamaControllerKou1.text,
            "l": _labControllerKou1.text,
        },
        {
            "Ders Adi ve Kodu":
            _dersControllerKou2.text,
            "t": _teoriControllerKou2.text,
            "u": _uygulamaControllerKou2.text,
            "l": _labControllerKou2.text,
        },
        {
            "Ders Adi ve Kodu":
            _dersControllerKou3.text,
            "t": _teoriControllerKou3.text,
            "u": _uygulamaControllerKou3.text,
            "l": _labControllerKou3.text,
        }
    ])
},
{
    'yazOkuluAlinanDersler':
    FieldValue.arrayUnion([
        {
            "Ders Adi ve Kodu":
            _dersControllerYaz1.text,
            "t": _teoriControllerYaz1.text,
            "u": _uygulamaControllerYaz1.text,
            "l": _labControllerYaz1.text,
        },
        {
            "Ders Adi ve Kodu":
            _dersControllerYaz2.text,
            "t": _teoriControllerYaz2.text,
            "u": _uygulamaControllerYaz2.text,
            "l": _labControllerYaz2.text,
        },
        {
            "Ders Adi ve Kodu":
            _dersControllerYaz3.text,
            "t": _teoriControllerYaz3.text,
            "u": _uygulamaControllerYaz3.text,
            "l": _labControllerYaz3.text,
        }
    ]),
}
```

B. Projenin Bize Katkıları

Daha önce mobil uygulama geliştirme konusunda

hiçbirimizin çok fazla deneyimi yoktu. Projeye başlamadan önce yaptığımız;
Hangi platformu ve dili seçmeliyiz, hangisi daha avantajlı, bize verilen sürede hangisini daha pratik ve güzel bir şekilde yapabiliriz gibi araştırmalarımız sayesinde öncelikle mobil uygulama geliştirme hakkında bilgi sahibi olduk. Daha sonra ise seçimlerimize karar verdik ve seçtiğimiz dil hakkında araştırma ve öğrenme sürecine girdik. Yeni bir dil öğrenirken en faydalı şey öğrendiklerimizi uygulamaktı ve hemen uygulama sürecine girdiğimiz için öğrendiklerimizden daha fazlasını da uygularken öğrendik ve bu hem bana hem ekip arkadaşlarıma çok fazla şey kattı. Çünkü hem hata alıp o hataların çözümlerinin neler olabileceğini gördük. Bu da ilerleyen süreçlerde tekrar mobil uygulama geliştirmek istediğimizde bize kolaylık sağlayacak. Bunlara ek olarak Flutter dilini tanımamı ve öğrenmemi sağladı. NoSql nedir hangi veritabanları nosql dir onları öğrendik. Firebase nedir ve nasıl kullanılır bunları öğrendik. Ekibimize de planlı ve programlı çalışmayı benimsememiz gerektiğini gösterdi. Yaptığımız toplantılar ile birlikte beraber çalışıp ekip olmanın kısa zamanda bir proje çıkarmanın nasıl olduğunu öğrendik. Github kullanımına iyice alıştık. 1 ayda daha önce kullanmadığımız bir dili kullanıp öğrenmeyi ve bir ay içinde kendimizi ne kadar geliştirebileceğimizi gördük ve emek verince her şeyin halledildiğini, zamanın yetebileceğini ve önemli olmadığını bir kere daha anlamış olduk.

Proje Sürecini ve Görevleri Belirlerken Nasıl Bir Metot kullandık?

Proje sürecini ve görevleri belirlerken agile metodlarından biri olan Scrum metodunu kullandık. Bu metodu kullanma sebebimize değinecek olursam, belirlediğimiz projemin birçok insana hitap etmesi ve büyük bir proje olması, değişebilecek yani esnek, belki de birçok kez üstünden geçilmesi ve tekrar edilmesi gereken son olarak da ilerleyen süreçlerde isteklerinin ve geri dönüşlerinin bu proje için çok önemli olmasından dolayı Scrum metodu kullanarak görevlendirme ve projeyi yönetme yoluna gittik.

- 1- Öncelikle projenin gereksinimlerini tam olarak belirledik.
- 2- Gereksinimleri belirledikten sonra bir öncelik sıralaması yaptık ve onları listeledik.
- 3- Öncelikleri belirledikten sonra projeyi küçük kısımlara ayırdık. Böyle yapmak projede aşama aşama ilerlenmesine ve projede bir değişiklik gerektiğinde bunu proje üzerinde uygulanmasına kolaylık sağlar.
- 4- Daha sonra bu projeyi ekibimiz ile parçalara ayırdık ve her kısım için de ayrı gereksinimleri belirledik.
- 5- Öncelik sırasına göre ilk kısımdan başladık. Seçtiğimiz kısım için projenin ne zamana kadar yetiştirilmesi gerekiyorsa ve büyüklüğüne göre o kısma ortalama bir süre verdik. Daha sonra projeye başladık ve sprint boyunca günlük toplantılar yaparak (bu toplantılara normalde müşteri için yapılyorsa müşteri de katılır çünkü ilerleyeşi onun da görmesi ve eğer değişiklik istiyorsa belirtmesi önemlidir.) projemizin ne durumda olduğunu, değişiklik yapmak isteyip istemediğimizi yani genel olarak 10-15 dakikalık toplantılar ile genel ilerleyiş hakkında değerlendirmeler yapılır. Böyle toplantıların yapılması projenin doğru ve düzenli ilerlemesinde, eğer bir değişiklik yapılacaksa çok ilerlenmeden hemen değişikliğin uygulanmasına ve projenin aksamamasını en önemlisi de zaman kaybının yaşanmamasını sağlar.
- 6- Sprint bittikten sonra test aşamasına geçtik ve o kısım test edilmeye başlandı. Bir sorun çıktığında ise o sorun çözdük ve yeniden test ettik. Test sonucunda sorunsuz olduğu

belirlendikten sonra o kısım için sprint bitti ve diğer kısım için sprint başladı. Böyle kısımlara ayırmamız bir sorun çıktığında tüm projeyi değil de sadece o kısmı etkiler ve kısımlar da küçük olduğu için sorunu düzeltmek projenin tamamını düzeltmeye göre daha kolay olur.

7- Diğer sprintler için de aynı aşamalar uygulandı ve en son bittiğinde yeniden bir test aşamasına geçtik.

8- Test aşaması da sorunsuz bir şekilde bittikten sonra projemiz tamamlanmış ve hazır hale geldi.

C. Tartışma

Mobil uygulama geliştirilirken kaç kişi ile çalışır ve nasıl görevlendirme yapılması daha doğru olur?
Hangi geliştiricilere ihtiyaç var?

Mobile Developer:

Kodlama ustalığını alan ve uygulama geliştiren yazılım geliştiricisidir. Akıllı telefon, tabletler vs. için yeni teknolojiler geliştirir.

Backend Developer:

Uygulamanın veritabanı ve işletim sisteminin düzgün ve verimli entegrasyonundan sorumludur. Tüm ön uç dinamiklerinin arka uç yazılımına anında yanıt vermesini sağlamaktan sorumludur.

UI – UX Designer:

UI arayüz tasarımını ifade ederken, UX deneyim üzerine kurulmuş bir tasarım sürecini anlatan geliştiricilerdir. UI Uygulamanın tasarımının işleyişinden, hızından, estetiğinden yani tüm grafiklerden sorumludur. Yazılan koda bir dış görünümüm sağlar. UX ise tasarım yapmaktan çok kullanıcıların site üzerindeki etkileşimin nasıl olacağıyla ilgilenir.

Quality Assurance Specialist:

Uygulamanın test aşaması ile ilgilenir ve projenin başından sonuna kadar ayrılmaz bir parçasıdır. Projenin son aşamasına kadar hazır olmasını sağlamak için hataları ve eksiklikleri tespit eder. Daha az geliştirici ile hatta tek bir kişi ile bile yapılabilir ama zaman, kalite vb. konularda düşüklük görülür. Bu yüzden yukarıdaki geliştiricilerin olması gerekir. Proje erken bitirilmek istenirse sayılar arttırılabilir.

KAYNAKÇA

- [0] <https://flutter.dev/docs>
- [1] <https://firebase.flutter.dev/docs/overview/>
- [2] <https://flutter.dev/docs/development/data-and-backend/firebase>
- [3] https://www.youtube.com/watch?v=PHf7NH2_gQs&t=458s
- [4] <https://www.syncfusion.com/>
- [5] <https://firebase.google.com/docs>
- [6] <https://flutter.dev/docs/development/ui/widgets/basics>
- [7] <https://flutter.dev/docs/development/ui/widgets>
- [8] <https://www.btkakademi.gov.tr/portal/course/google-flutter-ile-mobil-uygulama-kursu-10150#!/about>
- [9] <https://stackoverflow.com/questions/tagged/flutter>
- [10] https://www.youtube.com/watch?v=kY7QPqXc83A&list=PLjOFHn8uDrvR-nZtbKtV6NX_-4GaBkGNg