

**ANKARA ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**(BLM4537-A) IOS ile Mobil Uygulama Geliştirme I**

**Öznur KANDAKOĞLU**

**19290250**

**Öğr. Gör. Enver BAĞCI**

**01,2023**

## ÖZET

Bu rapor, Ankara Üniversitesi Bilgisayar Mühendisliği BLM4537 IOS ile Mobil Uygulama Geliştirme I dersi için hazırlanan uygulama hakkında bilgileri kapsamaktadır.

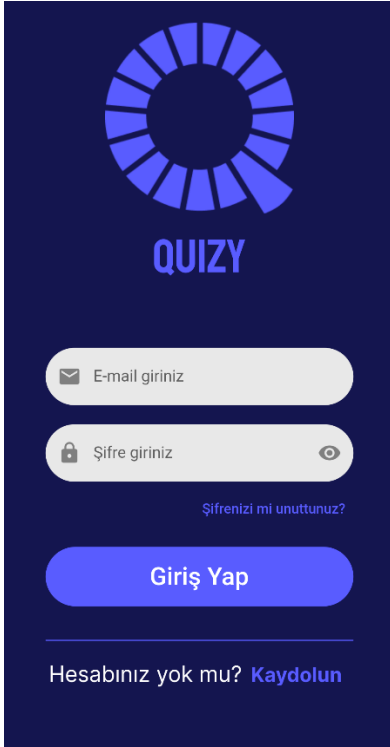
Quiz uygulamaları, kullanıcıların bilgi seviyelerini test etmelerine ve öğrenmelerine olanak tanır. Uygulama içerisinde kullanıcılar sorulara cevap verir ve doğru cevap verdiklerinde puan kazanırlar. Genellikle belirli bir konuda özelleştirilmiş sorular bulunur ve kullanıcılar konuya göre öğrenirler. Örneğin, geliştirmiş olduğum uygulamada kültür, çevre, insan hakları, spor gibi konular içeren soru quizleri oluşturulmuştur.

Uygulamanın amacı; yukarıda yazılmış metin doğrultusunda kullanıcıların bilgi seviyelerini test etmeleri, öğrenirken aynı zamanda eğlenmeleri gibi unsurları içermektedir. Geliştirilen uygulama dört farklı kategoride soru çeşitliği içermektedir. Kullanıcı her kategoride kendini test edebilmektedir.

Uygulamanın kullanımı, arayüzü, tasarımı ve kodlaması raporun devamında detaylı olarak anlatılmıştır.

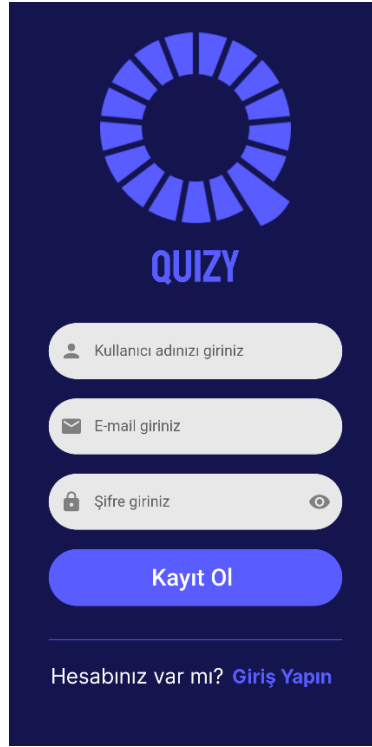
## 1. GİRİŞ EKRANI KULLANIMI VE ARAYÜZÜ

Öncelikli olarak kullanıcının uygulamayı açtığında ilk karşılaşacağı ekran Şekil 1.1’ de gösterilmiştir. Kullanıcı bu ekranla karşılaştığında üç farklı işlemi gerçekleştirebilir. Bunlar kayıt oluşturmak, uygulamaya giriş yapmak ve şifresini unutma durumunda şifresini yenileyebilmektir. Kayıt olmak için kullanıcı “kaydolun” butonuna tıklamalıdır. Bu butona tıkladığı takdirde kullanıcı Şekil 1.2’de gösterilen kayıt ekranı ile karşılaşır. Bu ekranda kullanıcının bazı bilgileri doldurması gerekir. Bunlar kullanıcı adı, mail ve şifreden oluşmaktadır. Kullanıcı bu bilgileri doldurduktan sonra sisteme kaydını tamamlamış olur. Sisteme kaydolan kullanıcı eğer şifresini unutursa Şekil 1.3’te gösterilen ekran sayesinde şifresini sıfırlayabilir.



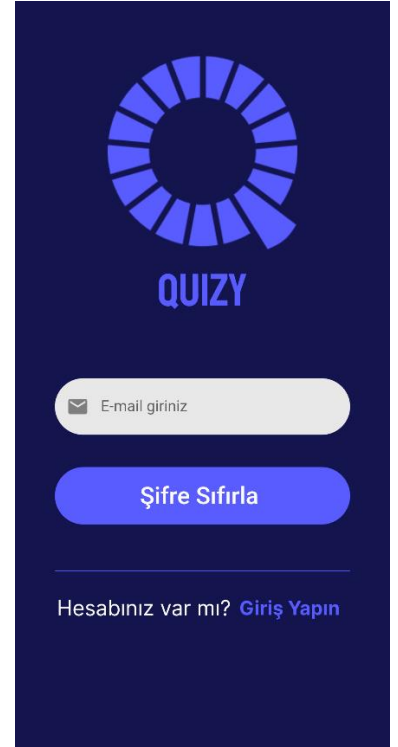
Şekil 1.1, QUIZY uygulamasının giriş ekranıdır. Ekranın üst kısmında QUIZY logoğu ve alt kısmında giriş bilgileri için alanlar yer almaktadır. Giriş alanları şunlardır: E-mail giriniz, Şifre giriniz (gizli) ve Şifrenizi mi unuttunuz? linki. Giriş Yap butonu ve Hesabınız yok mu? Kaydolun linki de ekranın alt kısmında yer almaktadır.

Şekil 1.1 Giriş Ekranı



Şekil 1.2, QUIZY uygulamasının kayıt ekranıdır. Ekranın üst kısmında QUIZY logoğu ve alt kısmında kayıt bilgileri için alanlar yer almaktadır. Kayıt alanları şunlardır: Kullanıcı adınızı giriniz, E-mail giriniz, Şifre giriniz (gizli) ve Kayıt Ol butonu. Hesabınız var mı? Giriş Yapın linki de ekranın alt kısmında yer almaktadır.

Şekil 1.2 Kayıt Ekranı



Şekil 1.3, QUIZY uygulamasının şifre sıfırlama ekranıdır. Ekranın üst kısmında QUIZY logoğu ve alt kısmında şifre sıfırlama alanları yer almaktadır. Şifre sıfırlama alanları şunlardır: E-mail giriniz, Şifre Sıfırla butonu ve Hesabınız var mı? Giriş Yapın linki.

Şekil 1.3 Şifre Sıfırlama Ekranı

```

Flexible DontHaveAnAccount(BuildContext context) {
  return Flexible(
    flex: 10,
    child: Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text(
          "Hesabınız yok mu?",
          style: GoogleFonts.inter(color: Colors.white, fontSize: 22),
        ),
        TextButton(
          onPressed: () async {
            Navigator.of(context).pushReplacement(PageTransition(
              child: SignupPage(),
              type: PageTransitionType.rightToLeftWithFade,
              duration: Duration(milliseconds: 400),
              reverseDuration: Duration(milliseconds: 400)));
          },
          child: Text(
            "Kaydolun",
            style: GoogleFonts.inter(
              color: Color(0xff595CFF),
              fontSize: 20,
              fontWeight: FontWeight.bold),
          ),
        ),
      ],
    ),
  );
}

```

Şekil 1.4 “Hesabınız yok mu? Kaydolun” Kontrol Kodu

```

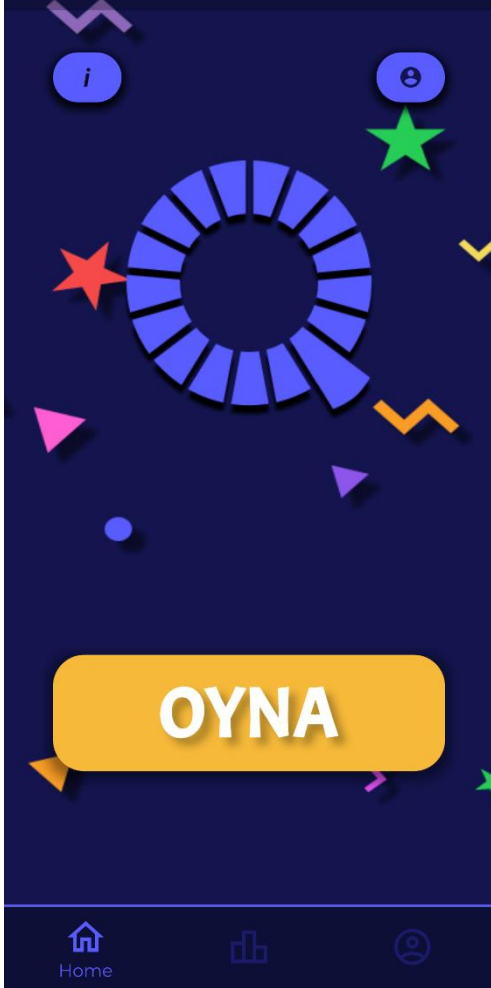
Row DoYouHaveAccountButton(BuildContext context) {
  return Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Text(
        "Hesabınız var mı?",
        style: GoogleFonts.inter(color: Colors.white, fontSize: 22),
      ),
      TextButton(
        onPressed: () {
          Navigator.of(context).pushReplacement(PageTransition(
            child: LoginPage(),
            type: PageTransitionType.rightToLeftWithFade,
            duration: Duration(milliseconds: 400),
            reverseDuration: Duration(milliseconds: 400)));
        },
        child: Text(
          "Giriş Yapın",
          style: GoogleFonts.inter(
            color: Color(0xff595CFF),
            fontSize: 20,
            fontWeight: FontWeight.bold),
        ),
      ),
    ],
  );
}

```

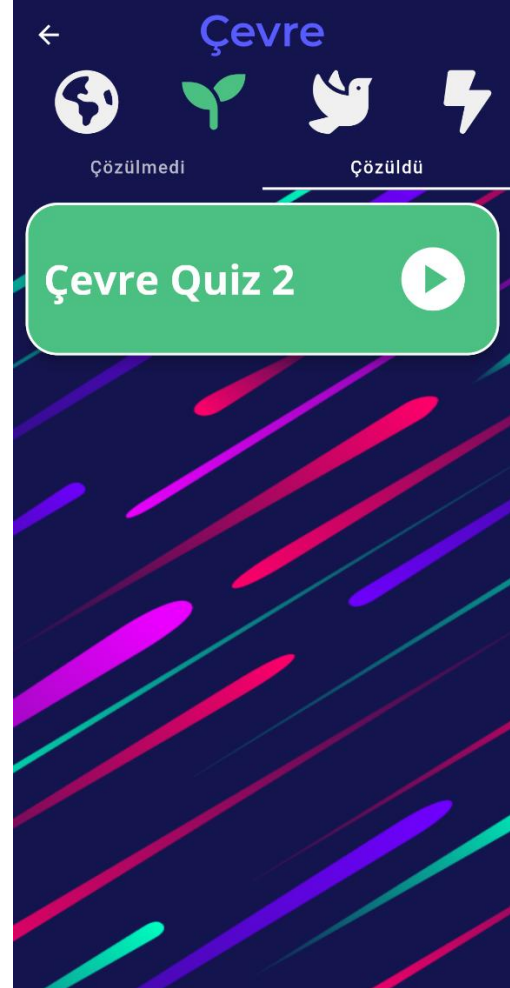
Şekil 1.5 “Hesabınız var mı? Giriş Yapın” Kontrol Kodu

## 2. UYGULAMA KULLANIMI VE ARAYÜZÜ

Kullanıcı uygulamaya giriş yaptıktan sonra Şekil 2.1’de gösterilen ekran ile karşılaşmaktadır. Ekranın en alt kısmında üç farklı element bulunur. Bunlar anasayfayı, liderlik tablosu ekranını ve profili içermektedir. Anasayfa ekranında yer alan oyna butonuna bastıktan sonra Şekil 2.2’de gösterilen ekran ile karşılaşır. Bu ekranın üst kısmında dört farklı kategori yer almaktadır. Bunlar sırasıyla kültür, çevre, insan hakları ve spor sorularını içermektedir. Her kategori kendi içerisinde iki seçenek içermektedir. Bunlar çözüldü ve çözülmedi olarak ikiye ayrılmıştır. Kullanıcı daha önce çözdüğü quizleri çözüldü kısmından ulaşabilir.



Şekil 2.1 Anasayfa Ekranı



Şekil 2.2 Kategori Ekranı

```
String categoryId(String input) {
    switch (input) {
        case "Kültür":
            return "0";
        case "Çevre":
            return "1";
        case "İnsan Hakları":
            return "2";
        default:
            return "0";
    }
}
```

Şekil 2.3 Kategori String

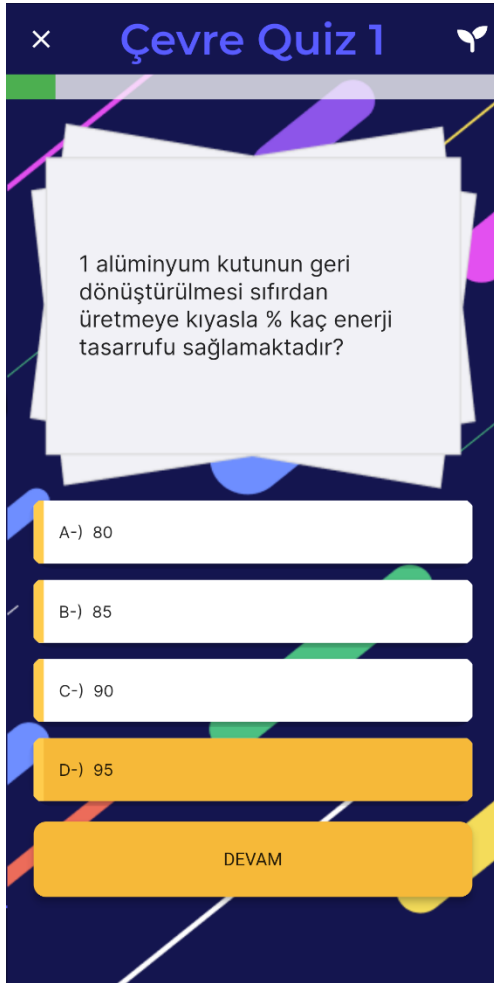
```
void answerControl(String answer) {
    //firebasede bir answer olacak şıktaki texti buraya yollayıp kontrol sağlanması gerekiyor

    if (correctAnswerTextController.text == answer) {
        switch (optionController.text) {
            case "A":
                {
                    setState(() {
                        colorControllerA = Colors.green;
                        score += 10;
                        print("skor=" + score.toString());
                    });
                }
                break;

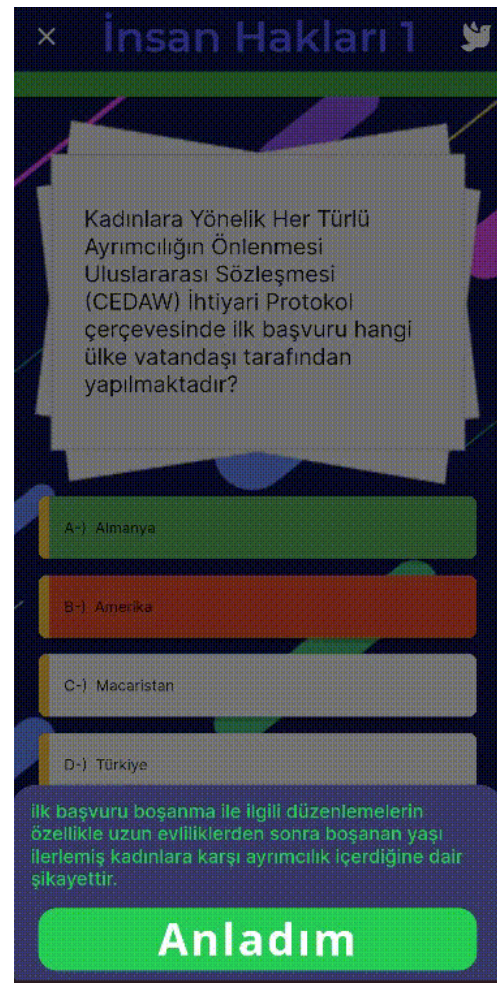
            case "B":
                {
                    setState(() {
                        colorControllerB = Colors.green;
                        score += 10;
                        print("skor=" + score.toString());
                    });
                }
                break;
        }
    }
}
```

Şekil 2.4 Quiz Ekranı Cevap Kontrolü Kodu

Kullanıcı çözmek istediği kategoriye seçtikten sonra Şekil 2.3'teki ekran ile karşılaşır. Ekranın üstünde yer alan bar sayesinde kullanıcı testin bitmesine kalan soru sayısının oranını anlayabilir. Kullanıcı soruyu cevapladıktan sonra seçtiği cevap sarı renge döner ve cevabından eminse devam butonuna basar. Şekil 2.4'te ise kullanıcının devam dedikten sonra karşısına çıkan ekran gösterilmiştir. Bu ekranda soruya yanlış cevap verilmesi durumunda karşılaştığımız ekrandır. Bizim seçtiğimiz cevap yanlış ise o seçenek kırmızı renge döner, eğer doğru ise yeşil renk olur. Yanlış cevap vermesi durumunda karşısında beliren ekranda ufak bir bilgilendirme metni yer alır. Database'e her soru daha öncesinde girilmiş ve yanlış cevap vermesi durumunda çıkacak olan bilgilendirme metni kaydedilmiştir. Kullanıcıya bilgilendirme yapıldıktan sonra "anladım" butonuna basarak quize kaldığı yerden devam etmesi sağlanır.



Şekil 2.3 Quiz Ekranı



Şekil 2.4 Yanlış Cevaplanmış Soru Örneği

```

void showAlertDialog(BuildContext context) {

  Widget cancelButton = TextButton(
    child: Text("Testten çık", style: TextStyle(color: Colors.red, fontWeight: FontWeight.bold)),
    onPressed: () {
      Navigator.pop(context);
      Navigator.of(context, rootNavigator: true).pop('dialog');
    },
  );
  Widget continueButton = TextButton(
    child: Text("Devam et", style: TextStyle(color: Color(0xff26CE55), fontWeight: FontWeight.bold)),
    onPressed: () {
      Navigator.of(context, rootNavigator: true).pop('dialog');
    },
  );

  // set up the AlertDialog
  AlertDialog alert = AlertDialog(
    backgroundColor: Color(0xff14154F),
    title: Text("Uyarı!", style: TextStyle(color: Colors.red, fontWeight: FontWeight.bold)),
    content: Text("Testten çıkarsanız bir daha bu testten puan kazanamayacaksınız!!", style: TextStyle(c
    actions: [
      cancelButton,
      continueButton,
    ],
  );
}

```

*Şekil 2.5 Buton ve Alarm Kontrol Kodu*

Şekil 2.3'te gösterilen ekranın üst bölümünde yer alan bar tamamen yeşil olduğunda yani testi tamamladığında kullanıcı Şekil 2.5'te yer alan tebrikler ekran ile karşılaşır. Bu ekranda kullanıcı toplam kaç soruda doğru yaptığını ve buna oranla kaç puan aldığını görebilmektedir. Ekran üzerinde yer alan “geri dön” butonu ile de kullanıcı tekrar kategori sayfasına yönlendirilir.





Şekil 2.5 Tebrikler Ekranı

```
class AfterGameScreen extends StatefulWidget {  
  final int score;  
  
  const AfterGameScreen({Key? key, required this.score}) : super(key: key);  
  
  @override  
  State<AfterGameScreen> createState() => _AfterGameScreenState();  
}  
  
class _AfterGameScreenState extends State<AfterGameScreen> {  
  late double correctNumber;  
  bool isPlaying = false;  
  late ConfettiController _controllerTopCenter;  
  
  @override  
  void dispose() {  
    _controllerTopCenter.dispose();  
    super.dispose();  
  }  
  
  @override  
  void initState() {  
    correctNumber = widget.score / 10;  
    _controllerTopCenter =  
      ConfettiController(duration: const Duration(seconds: 5));  
    super.initState();  
  }  
}
```

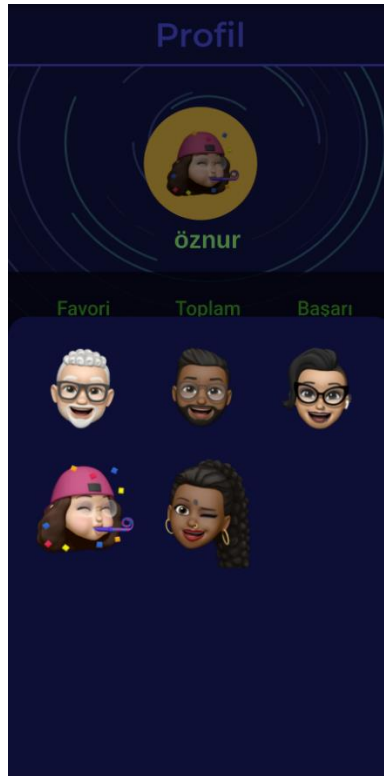
Şekil 2.6 AfterGameScreen Class'ı

### 3. UYGULAMA PROFİLİ KULLANIMI VE ARAYÜZÜ

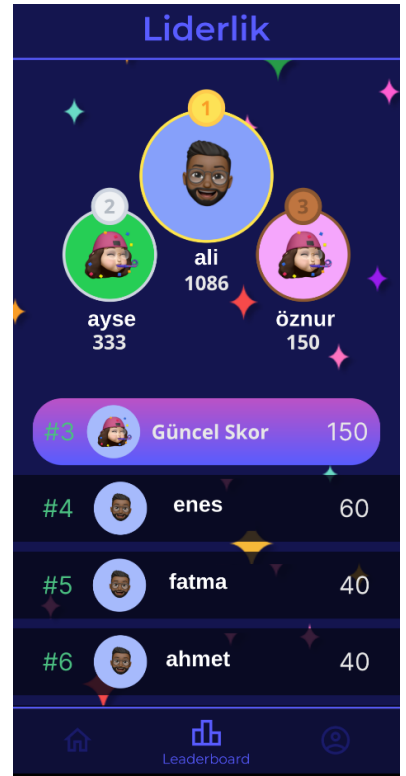
Kullanıcı uygulama üzerinden kendi bilgilerini görebilmektedir. Şekil 3.1’de bu ekran gösterilmiştir. Kullanıcı bu ekrandan favori kategorisini, toplam puanını ve başarı sırası görebilmektedir. Aynı zamanda avatarını seçebilmektedir. Avatar seç sekmesine tıkladığında database’den çekilen resimler görülür ve buradan kullanıcı avatarını değiştirebilir. Aynı zamanda kullanıcı şifresini değiştirebilir ve uygulamadan çıkış yapabilir. Kullanıcı aynı zamanda liderlik tablosunda kendi sıralamasını görebilmektedir.



Şekil 3.1 Profil Ekranı



Şekil 3.2 Avatar Seçme Ekranı



Şekil 3.3 Liderlik Ekranı

```

Row SignOut(BuildContext context) {
  return Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Text(
        "Çıkış Yap",
        style: Theme.of(context)
          .textTheme
          .titleLarge
          ?.copyWith(color: Colors.white),
      ),
      IconButton(
        color: Colors.white,
        iconSize: 35,
        onPressed: () {
          _authService.Logout(context);
        },
        icon: Icon(Icons.exit_to_app))
    ],
  );
}

```

```

InkWell ChangePassword(BuildContext context) {
  return InkWell(
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Text(
          "Şifre Değiştir",
          style: Theme.of(context)
            .textTheme
            .titleLarge
            ?.copyWith(color: Colors.white),
        ),
      ],
    ),
  );
}

```

Şeki 13.4 Change Password ve Sign Out Kod Parçacıkları

```

List<Object> userRank = [];

Future getList() async {
  var data = await FirebaseFirestore.instance
    .collection('users')
    .orderBy('Score', descending: true)
    .get();

  setState(() {
    userRank = List.from(data.docs.map((doc) => doc.id));
    print(userRank.indexOf(FirebaseAuth.instance.currentUser!.uid) + 1);
  });
}

```