GÖMÜLÜ SİSTEM ÇÖZÜMLEMELERİ

Dr. Öğr. Üyesi Fırat AYDEMİR

Uyku Apnesi Tespit Sitemi

2022-2023 Bahar Dönemi

10/03/2023-24/04/2023

4.Rapor

HASAN MÜNİR DOĞRUEL-201913171801

YUSUF SARUKAN-201813171058

EMİRHAN AKDİN-201813171049

GİRİŞ

Bu hafta ki rapor döneminde, bir önceki rapor döneminde yaptığımız Windows Form Uygulaması ve Google Firebase ile oluşturduğumuz veritabanı ilişkisini kurduk. Doktorun, kayıt olma, giriş yapma, hasta ekleme ve silme gibi yapmak isteyeceği işlemleri uygulamaya kodladık.

İş Planı

Yapılcak İş	Rapor 1	Rapor 2	Rapor 3	Rapor 4	Rapor 5	Rapor 6	Rapor 7
Firebase kurulumu ve mobil için XML tasarımı	**X**						
Firebase ile mobil arası iletişim kuracak Java kodları		**X**					
Doktorun uygulaması için Windows Form Tasarımı			**X**				
Windows Form Uygulamasının Firebase bağlantısı				**X**			
Ardunio devrelerinin bağlanması ve çalıştırılması							
Ardunio, firebase ve mobil bağlantısı							
Testlerin yapılması ve hataların giderilmesi							

YAPILAN İŞLER

1.Firebase Bağlantısı

```
public partial class Login : Form
{
    static List<TheUsers> userList;
    static string referance;
    4 başvuru
    public Login()
    {
        userList = new List<TheUsers>();
        InitializeComponent();

    }
    IFirebaseConfig ifc = new FirebaseConfig() //RealTime Database adresine bağlantı için bir nesne oluşturuldu.
    {
        //Database adresi ve şifresi
        AuthSecret = "UzExMUlmK4nGDVcAQvcxCDUkdoh0DOZDm3PMpxsL",
        BasePath = "https://uykuapnesitespitsistemi-default-rtdb.firebaseio.com/"
    };
    IFirebaseClient client;

    1 başvuru
    private void Login_Load(object sender, EventArgs e)
    {
        try
        {
            client = new FireSharp.FirebaseClient(ifc);
        }
        catch(Exception ex)
        {
            MessageBox.Show($"Bağlantınızı Kontrol Ediniz. \n Hata Kodu => {ex}");
        }
        this.CenterToScreen();

    }
```

Araştırmalarımız sonucunda, FireSharp'ın kolay kodlanabilen ve stabil çalışan bir Firebase kütüphanesi olduğunu öğrendik. Bu sebeple Firebase'ye bağlanmak için FireSharp kütüphanesini kullandık. Firebase'ye bağlanırken ifc adındaki değişkene AuthSecret ve BasePath vererek nesne oluşturduk. Sonrasında bu nesneyi kullanarak Firebase'ye bağlantı kurduk.

2.Kullanıcı Listesi(Doktorlar) Class

```
    1 başvuru
    public TheUsers(string ıD, string name, string alan, List<Hasta> hastaList, string mail, string password)
    {
        ID = ıD; // Buraya bir düzenleme yapılacak!
        Name = name;
        Alan = alan;
        HastaList = hastaList;
        Mail = mail;
        Password = password;

    }

    1 başvuru
    public TheUsers()
    {
    }

    1 başvuru
    public static void ShowError() =>
        MessageBox.Show(error);
    0 başvuru
    public static bool IsEqual(TheUsers user1, TheUsers user2)
    {
        if (user1 == null || user2 == null)
        {
            return false;
        }
        if (user1.Mail != user2.Mail)
        {
            error = "Kullanıcı Bulunamadı";
            return false;
        }
        else if (user1.Password != user2.Password)
        {
            error = "Kullanıcı Adı Veya Şifre Yanlış";
            return false;
        }
        return true;
    }
```

TheUsers adında bir class oluşturduk. Bu Class içinde bir doktorun bilgilerinde olması gereken ne varsa tanımladık. Ekstra olarak bir doktorun hasta listesi olmalıdır. Bunu ise bir liste şeklinde tanımladık.

3.Kayıt Olma Ekranı

```
    {
        List<TheUsers> userList = new();
        List<Hasta> hastaList = new();
        List<Rapor> raporList = new();
        raporList.Add(new Rapor(1,"18.04.2023","deneme","deneme"));
        hastaList.Add(new Hasta(new Guid().ToString(), "Hasan", "Münir", "Apne", 22, "E", raporList));

        string email = MailBox.Text;

        System.Text.RegularExpressions.Regex expr = new System.Text.RegularExpressions.Regex(@"^[a-zA-Z][\w\.-]{2,28}[a-zA-Z0-9]@[a-zA-Z0-9][\w\.-]*[a-zA-Z
        //Kutucukları Kontrol Ediyoruz.
        if (string.IsNullOrWhiteSpace(MailBox.Text) && string.IsNullOrWhiteSpace(PasswordBox.Text))
        {
            MessageBox.Show("Lütfen Tüm Alanları Doldurunuz.");
            return;
        }
        else if (expr.IsMatch(email))
        {


            string mail = MailBox.Text;
            string password = PasswordBox.Text;

            string referance = Guid.NewGuid().ToString();
            userList.Add(new TheUsers(referance, "Münir", "UykuApnesi", hastaList, mail, password));
            SetResponse set = client.Set(@"Users/" + referance, userList.First());
        }
        else
            MessageBox.Show("Geçerli Bir Mail Adresi Giriniz.");

    }
```

Doktor kayıt olma formunda bir mail ve password belirler; uygulama ise bir random guild id oluşturarak veritabanına doktoru kaydeder. Guild id ise referance adlı bir değişkene daha sonra giriş yaptıktan sonra doktorun hasta listesini listemek için TheUsers listesine kaydettik.

4.Giriş Yapma Ekranı

```
1 başvuru
private void GirisYap_Click(object sender, EventArgs e)
{


    //Kutucukları Kontrol Ediyoruz.
    if (string.IsNullOrWhiteSpace(MailBox.Text) && string.IsNullOrWhiteSpace(PasswordBox.Text))
    {
        MessageBox.Show("Lütfen Tüm Alanları Doldurunuz.", "HATA");
        return;
    }
    string email = MailBox.Text;

    System.Text.RegularExpressions.Regex expr = new System.Text.RegularExpressions
        .Regex(@"^[a-zA-Z][\w\.-]{2,28}[a-zA-Z0-9]@[a-zA-Z0-9][\w\.-]*[a-zA-Z0-9]\.[a-zA-Z][a-zA-Z\.]*[a-zA-Z]$");
    //Database tepkiyi TheUsers sınıfına göre kontrol edecek.
    FirebaseResponse res = client.Get(@"Users/");
    var deneme = res.Body;


    var data = JsonConvert.DeserializeObject<Dictionary<string, TheUsers>>(deneme);


    foreach (var item in data)
    {
        TheUsers user = item.Value;
        userList.Add(user);
    }

    TheUsers CurUser = new TheUsers()
    {
        Mail = MailBox.Text,
        Password = PasswordBox.Text,
    };

    TheUsers finaluser = userList.FirstOrDefault(x => x.Mail.Equals(CurUser.Mail) && x.Password.Equals(CurUser.Password));

    if (finaluser!=null && expr.IsMatch(email))
    {
        UykuApnesiTespitSistemi uykuApnesiTespitSistemi = new UykuApnesiTespitSistemi(finaluser.HastaList,finaluser.ID);
        uykuApnesiTespitSistemi.Show();
        this.Visible = false;
    }
    else
        TheUsers.ShowError();
}
```

Doktorun textboxlara yazdığı maili ve parolayı CurUser adlı bir değişkene atadık. Sonrasında veritabanından çektiğimiz mail ve password ile karşılaştırdık. Son olarak bu kısımda daha sonra doktorun hasta listesini görüntüleyebilmesi için hastaList'i bir sonraki forma gönderdik.

5.Listeleme, Ekleme ve Veri Silme

```
1 başvuru
private void dataGridView1_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)
{
    removeAt = e.RowIndex;
    HastaAdıTextBox.Text = hastaList[removeAt].Name;
    HastaSoyAdıTextBox.Text = hastaList[removeAt].Soyad;

}

1 başvuru
private void HastaEkleButton_Click(object sender, EventArgs e)
{
    string hastaadi = HastaAdıTextBox.Text;
    string hastasoyadi = HastaSoyAdıTextBox.Text;
    int dogumtarihi =2023-HastaYasi.Value.Year;
    bool cinsiyet = checkBox1.Checked;
    Guid random = new Guid();
    if (hastaList == null)
    {
        hastaList = new List<Hasta>();
    }

    hastaList.Add(HastaEkle(random.ToString(), hastaadi, hastasoyadi, dogumtarihi, cinsiyet));
    SetResponse set = client.Set(@"Users/" + id+"/HastaList",hastaList);
    dataGridView1.DataSource = null;
    dataGridView1.Update();
    dataGridView1.Refresh();
    Task.Delay(100);
    dataGridView1.DataSource = hastaList;
    dataGridView1.Update();
    dataGridView1.Refresh();
```

Örnek olarak yeni hasta ekleme özelliği. Daha önce uygulamanın rastgele oluşturduğu guild id verisini references değişkenine atamıştık. Bu references değişkeni bize veritabanındaki bütün hastalar yerine daha spesifik olma imkanı verdi. Uygulama doktorun sadece guild id (references)'ın hasta listesini listeliyor, listesine ekleme yapıyor veya siliyor.

YAPILACAK OLAN ÇALIŞMALAR

Bir sonraki rapor döneminde Ardunio devresini oluşturacağız. Devreyi oluşturmak için gerekli sensörleri, kartları ve kabloları temin edeceğiz. Bu devreyi kurarken kullanmak istediğimiz Ardunio kartı ise ESP32.