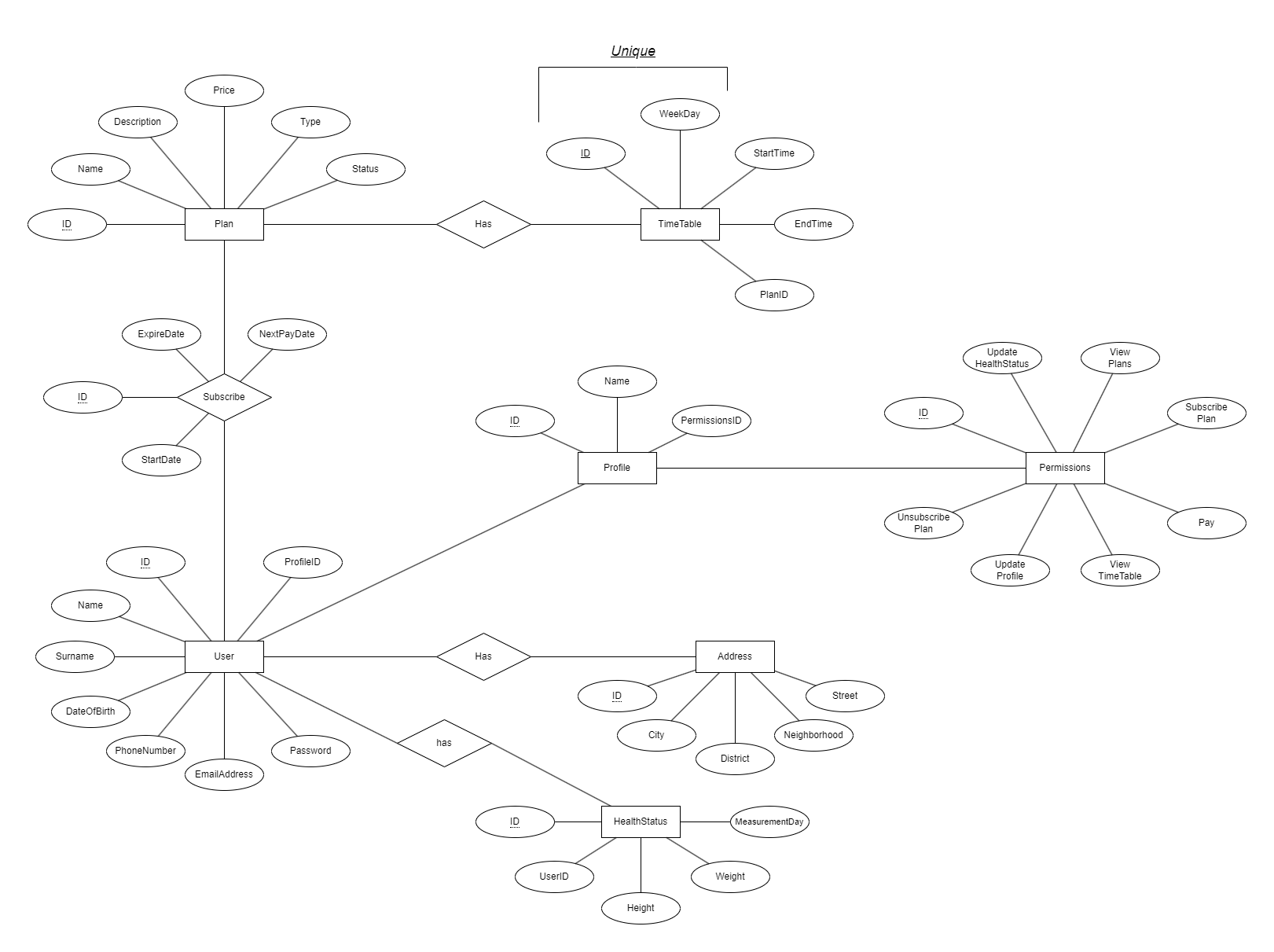


| Ders: | BM309 - Veritabanı Yönetim Sistemleri |
| --- | --- |
| Ödev Adı: | SQL Server Uygulaması - 35 |
| Hazırlayanların Okul No: | 211001012  211001096 |
| Hazırlayanların Adı Soyadı: | Batuhan Özdemir  Ela Elhac Ali |
| Öğrenim Türü: | N.Ö. |

**ÖZET**

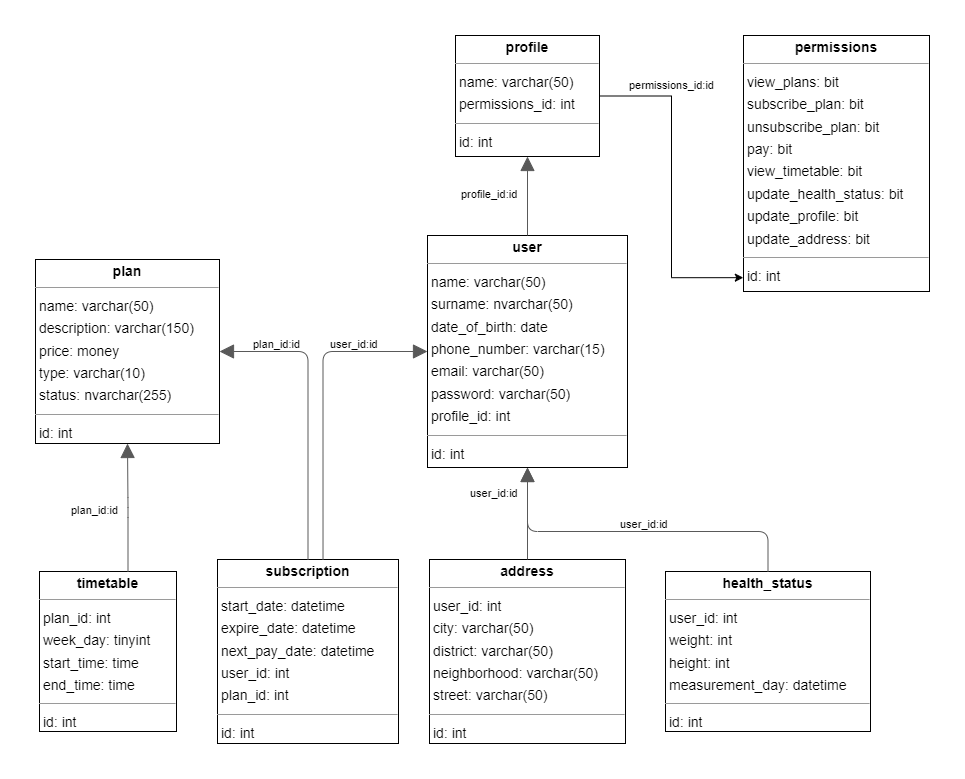
Bu Rapor “Sql Server” veritabanını kullanarak bir spor salonunu ve bu işletmeye gelen kullanıcıları daha iyi yönetmek ve takip etmek için tasarlanmış olan uygulamayı açıklamak amacıyla yazılmıştır. Raporda veritabanını oluşturulmasında kullanılmış ER diyagramına, veritabanına ait şemaya, tablolara ve bu tabloların kullanım amaçlarına, kullanılan procedureler, triggerlar ve kullanım amaçlarına, uygulamanın ekran görüntülerine yer verilmiştir.

**Veritabanı Yapısı**



**Şekil.1:** ER Diyagramı

Öncelikle, bir ER diyagram tasarımı yapılmıştır (Şekil.1). Uygulamanın geliştirilmesinde veritabanında tutulacak ve kullanılacak verilen birbirleriyle ilişkileri gösterilmiştir.



**Şekil.2** : Veritabanı Şeması

ER diyagramından yola çıkılarak veritabanı şeması oluşturulmuş, birincil ve yabancı anahtarlar belirlenmiş ve bunların birbiriyle olan ilişkisi gösterilmiştir.

**Tablolar**

Aşağıda oluşturulan tablo isimleri ve bu tabloların kullanım amaçları yer almaktadır:

**permissions**: Kullanıcı izinlerini tutan tablodur. Kullanıcıların uygulamaya girdiklerinde yapabilecekleri işlemleri belirtir.

**profile**: Kullanıcı profillerini tutan tablodur. Kullanıcıların belirli bir profil ile eşleşmesini sağlar. Kullanım izinleri ve kullanıcıların arasında köprü görevi görür. Belirli izne sahip kullanıcıları aynı profil üzerinde toplar.

**user**: Kullanıcı bilgilerini tutan tablodur.

**address**: Kullanıcıya ait adres bilgilerini tutan tablodur.

**health\_status**: Kullanıcıya ait sağlık durumu bilgilerini tutan tablodur.

**plan**: Spor salonunda kullanıcıların abone olabilecekleri planlarını tutan tablodur.

**timetable**: Plana ait haftalık çalışma tablosunu tutan tablodur. Kişilerin spor salonuna gelecek tarih ve saatlerin planlara göre belirlenmesinde kullanılır.

**subscription**: Kullanıcı abonelik bilgilerini tutan tablodur. Plan ile kullanıcıyı tarih ve ödeme bilgilerini de ekleyerek birbirine bağlar.

**Procedure’lar**

Aşağıda oluşturulan procedure isimleri ve bu procedure’ların kullanım amaçları yer almaktadır:

**GetUsersByKeyword**: Verilen anahtar kelimeye göre eşleşen kullanıcıları gösterir.

**InsertUser:** Verilen kullanıcı bilgileri için veritabanında “user” tablosunda yeni bir kullanıcı oluşturur.

**UpdateUser**: Verilen kullanıcının “id”sine göre güncelleme işlemi yapar.

**GetLatestAddressByUserId:** Kullanıcıya ait son son girilmiş olan adresi gösterir.

**InsertNewAddress:** Kullanıcı tarafından sağlanan adres bilgilerini alır ve bu bilgileri "address" tablosuna ekler.

**InsertHealthStatus:** Kullanıcı tarafından sağlanan sağlık durumu bilgilerini alır ve bu bilgileri "health\_status" tablosuna ekler.

**GetHealthRecordsByUserId:**  Kullanıcının kimliği (ID) ile sağlık durumu tablosundan ilgili kayıtları çeker ve bu kayıtları kullanıcıya ait sağlık kayıtlarını içeren bir sonuç kümesi olarak döndürür.

**GetSubscriptionInfoByUserId:** Belirli bir kullanıcının veritabanındaki "subscription" tablosundaki aktif abonelik bilgilerini getirir. Kullanıcının ID'sine dayanarak, aboneliği hala aktifse, başlangıç tarihinden ve bitiş tarihinden abonelik detaylarını çeker ve bu bilgileri döndürür. Eğer aktif bir abonelik bulunamazsa, bir hata mesajı üretilir.

**GetPlanById:** Belirli bir planın detaylarını almak için kullanılır. Plan ID'si temel alınarak, "plan" tablosundan ilgili planın adı, açıklaması, fiyatı, tipi ve durumu gibi bilgileri getirir.

**GetWeeklyTimetableByPlanId:** Belirli bir plana ait haftalık program bilgilerini getirir. Plan ID'si kullanılarak, "timetable" tablosundan ilgili plana ait haftanın her günü için başlangıç ve bitiş saatlerini çeker.

**DelayNextPayDateByOneMonth:** Belirli bir kullanıcının "subscription" tablosundaki abonelik bilgisinin bir sonraki ödeme tarihini bir ay erteleyerek günceller. Kullanıcı ID'si temel alınarak, ilgili aboneliğin bir sonraki ödeme tarihine bir ay eklenir.

**UnsubscribeUser:** Belirli bir kullanıcının "subscription" tablosundaki abonelik bilgisini siler. Kullanıcı ID'si temel alınarak, ilgili kullanıcının aboneliği veritabanından kaldırılır.

**GetActivePlans:** "plan" tablosundan aktif durumda olan planların detaylarını getirir. Planın adı, açıklaması, fiyatı, tipi ve durumu gibi bilgileri içeren kayıtları çeker.

**StartNewSubscription:** Belirli bir kullanıcıya yeni bir abonelik başlatır. Kullanıcı ID'si ve plan ID'si temel alınarak, "subscription" tablosuna yeni bir abonelik kaydı eklenir. Aboneliğin başlangıç tarihi, bitiş tarihi ve bir sonraki ödeme tarihi otomatik olarak hesaplanır.

**GetPermissionDetailsByProfileId:** Belirli bir profil ID'sine bağlı olarak, "profile" ve "permissions" tablolarından ilgili izin detaylarını getirir. Profil bilgileri ve ilişkili izinlerin durumu gibi detayları içerir.

**GetPlanData:** "plan" tablosundaki tüm planların detaylarını getirir. Plan ID'si, adı, açıklaması, fiyatı, tipi ve durumu gibi bilgileri içeren kayıtları çeker.

**GetTimetableData:** "timetable" tablosundaki tüm programların detaylarını getirir. Program ID'si, plan ID'si, haftanın günü, başlangıç saati ve bitiş saati gibi bilgileri içeren kayıtları çeker.

**UpdatePlanById:** Belirli bir planın detaylarını günceller. Plan ID'si temel alınarak, "plan" tablosundaki ilgili kaydın adı, açıklaması, fiyatı, tipi ve durumu güncellenir.

**UpdateTimetableById:** Belirli bir programın detaylarını günceller. Program ID'si temel alınarak, "timetable" tablosundaki ilgili kaydın haftanın günü, başlangıç saati ve bitiş saati güncellenir.

**CreateNewPlan:** Yeni bir plan oluşturur. Kullanıcının sağladığı ad, açıklama, fiyat, tip ve durum bilgileriyle "plan" tablosuna yeni bir kayıt ekler. Oluşturulan planın ID'si geri döndürülür.

**CreateNewTimetable:** Yeni bir program oluşturur. Kullanıcının sağladığı plan ID'si, haftanın günü, başlangıç saati ve bitiş saati bilgileriyle "timetable" tablosuna yeni bir kayıt ekler. Oluşturulan programın ID'si geri döndürülür.

**SearchUsers:** Kullanıcıları belirli bir anahtar kelimeye göre arar. Kullanıcının adı, soyadı, telefon numarası veya e-posta adresi gibi alanlarda anahtar kelimeye benzer bir eşleşme içeren kullanıcıları getirir.

**GetProfileNamesAndIds:** "profile" tablosundan tüm profil adlarını ve ilişkili profil ID'lerini getirir. Bu bilgiler, kullanıcıların profil seçenekleri arasında seçim yapmalarına yardımcı olabilir.

**InsertNewUser:** Yeni bir kullanıcı ekler. Kullanıcının sağladığı ad, soyad, doğum tarihi, telefon numarası, e-posta, şifre ve profil ID bilgileriyle "user" tablosuna yeni bir kayıt ekler. Oluşturulan kullanıcının ID'si geri döndürülür.

**GetUserById:** belirli bir kullanıcının detaylarını getirir. Kullanıcı ID'si temel alınarak, "user" tablosundaki ilgili kaydın adı, soyadı, doğum tarihi, telefon numarası, e-posta ve profil ID gibi bilgileri getirir.

**GetAllProfiles:** "profile" tablosundan tüm profil detaylarını getirir. Her bir profilin adı ve izinler ID'si gibi bilgileri içerir.

**UpdateProfileAndPermissions:** Belirli bir profilin ve izinlerinin detaylarını günceller. Profil ID'si temel alınarak, "profile" tablosundaki ilgili kaydın adı güncellenir. Aynı zamanda "permissions" tablosundaki ilgili izinler de güncellenir.

**GetPermissionsById:** Belirli bir izin ID'sine göre izin detaylarını getirir. "permissions" tablosundaki izin adı, açıklama ve diğer bilgileri içerir.

**DeleteProfileAndPermissions:** Belirli bir profilin ve izinlerinin kayıtlarını siler. Profil ID'si temel alınarak, "permissions" tablosundaki ilgili izinler ve "profile" tablosundaki ilgili profil kaydı silinir.

**InsertProfileAndPermissions:** Yeni bir profil ve izin seti oluşturur. Kullanıcının sağladığı ad ve izinlerle "permissions" tablosuna yeni bir kayıt ekler. Daha sonra, oluşturulan izinlerin ID'si kullanılarak "profile" tablosuna yeni bir profil kaydı eklenir. Oluşturulan profil ID'si geri döndürülür.

**Trigger’lar**

Aşağıda oluşturulan trigger isimleri ve bu trigger’ların kullanım amaçları yer almaktadır:

**trg\_permissions\_deleted Trigger:** Bu tetikleyici, "permissions" tablosundan bir kayıt silindiğinde çalışır. Silinen izin kaydının detaylarını "deleted\_permissions" tablosuna ekler.

**trg\_permissions\_profile Trigger:** Bu tetikleyici, "profile" tablosundan bir kayıt silindiğinde çalışır. Silinen profil kaydının detaylarını "deleted\_profile" tablosuna ekler.

**trg\_user\_deleted Trigger:** Bu tetikleyici, "user" tablosundan bir kayıt silindiğinde çalışır. Silinen kullanıcı kaydının detaylarını "deleted\_users" tablosuna ekler.

**trg\_health\_status\_deleted:** Bu tetikleyici, "health\_status" tablosundan bir kayıt silindiğinde çalışır. Silinen sağlık durumu kaydının detaylarını "deleted\_health\_status" tablosuna ekler.

**trg\_address\_deleted:** Bu tetikleyici, "address" tablosundan bir kayıt silindiğinde çalışır. Silinen adres kaydının detaylarını "deleted\_address" tablosuna ekler.

**trg\_plan\_deleted:** Bu tetikleyici, "plan" tablosundan bir kayıt silindiğinde çalışır. Silinen plan kaydının detaylarını "deleted\_plan" tablosuna ekler.

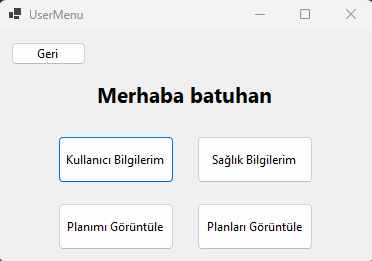
**trg\_timetable\_deleted:** Bu tetikleyici, "timetable" tablosundan bir kayıt silindiğinde çalışır. Silinen program kaydının detaylarını "deleted\_timetable" tablosuna ekler.

**trg\_subscription\_deleted:** Bu tetikleyici, "subscription" tablosundan bir kayıt silindiğinde çalışır. Silinen abonelik kaydının detaylarını "deleted\_subscriptions" tablosuna ekler.

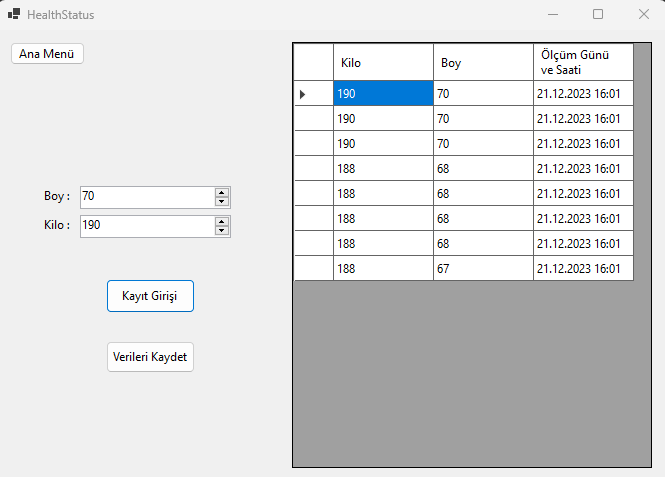
**InsteadOfUpdateAddress:** Bu tetikleyici, "address" tablosunda bir güncelleme işlemi olduğunda çalışır. "deleted\_address" tablosuna güncellenen adres kaydının eski detaylarını ekler ve ardından "address" tablosundaki kaydı günceller. Bu sayede adres güncellemesi takip edilebilir.

**Uygulama**

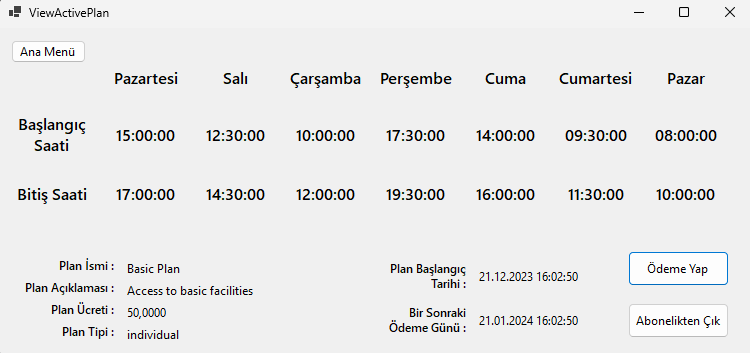
Uygulama C# form ile yapılmıştır, form sayfaları işlevlerine göre özelleşmiştir ve kullanıcılara izinleri ölçüsünde işlem yapma imkanı tanır. Admin kullanıcı ile giriş yapıldığında kullanıcılarla ilgili veri değiştirme, kullanıcı arama gibi bir çok özellik sunulmuştur. Aşağıda Uygulamanın bazı ekran görüntülerine yer verilmiştir.



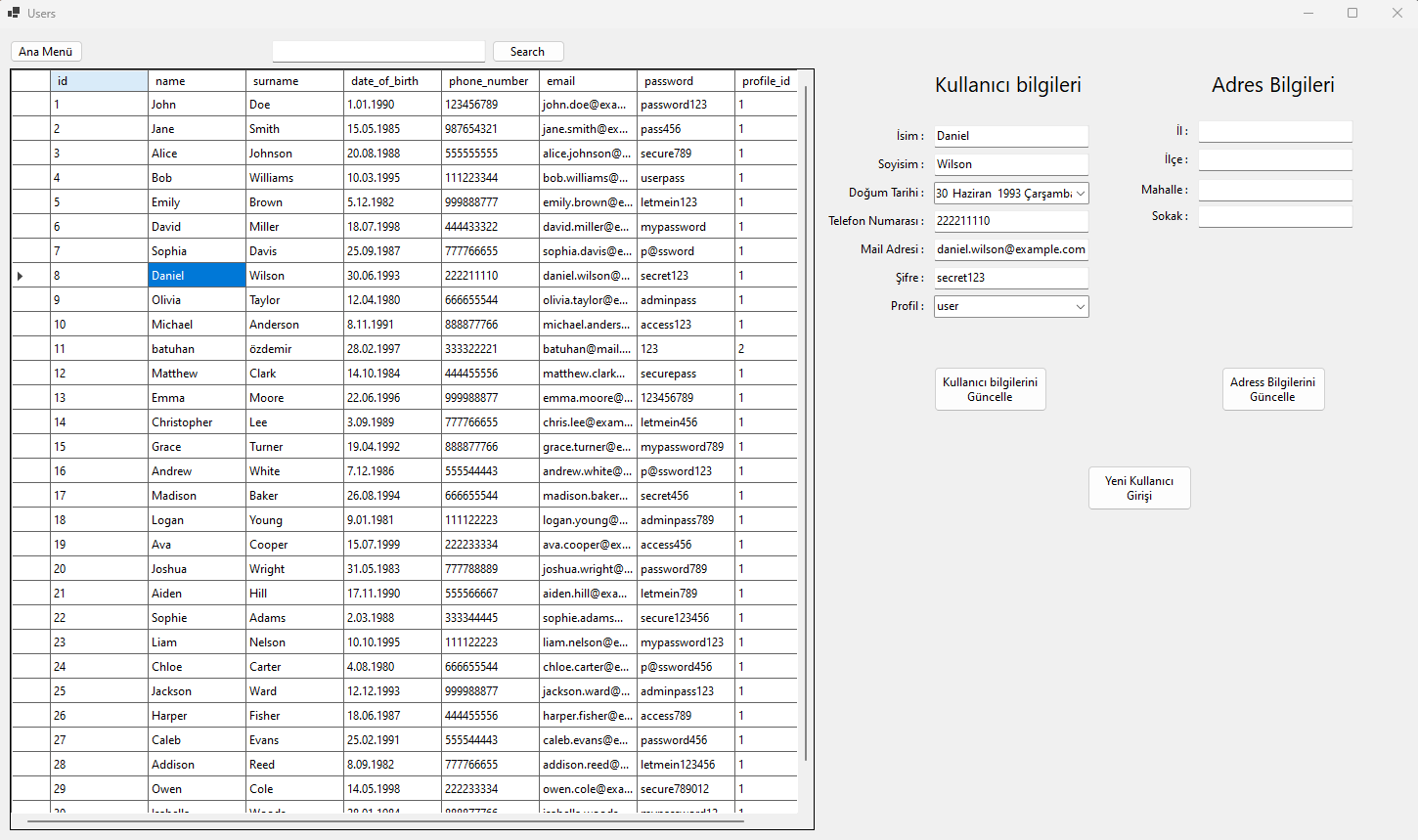
**Şekil.3:** Kullanıcı ile giriş yapıldığında yapılabilen işlemleri gösteren ve bu işlemlerle ilgili form sayfasına yönlendiren menü.



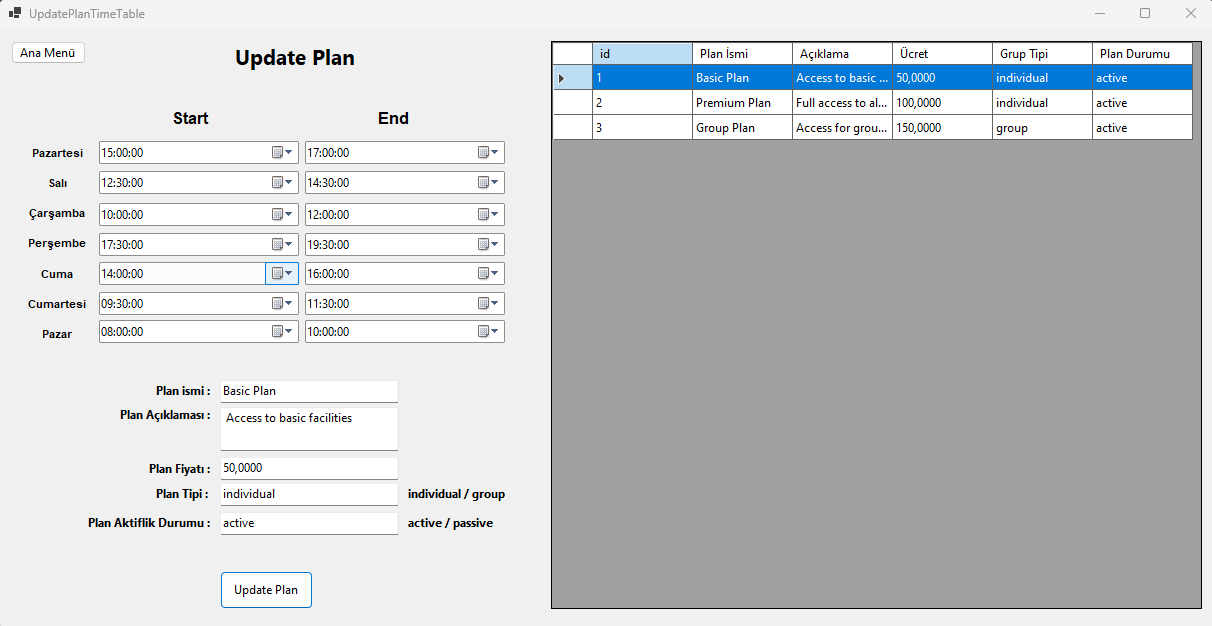
**Şekil.4:** Kullanıcıya ait boy ve kilo bilgisini girilen tarih ve saat bilgisiyle birlikte kaydeden ve gerektiği durumda .xlsx uzantılı tablo çıktısı almayı sağlayan ekran.



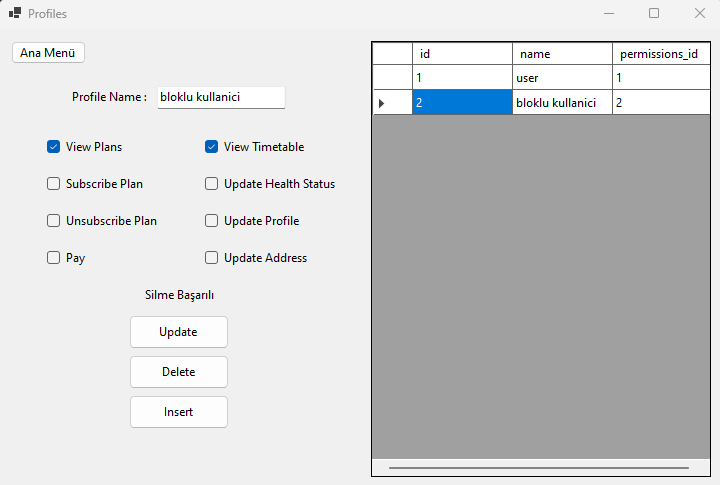
**Şekil.5:** Kullanıcıya ait aktif abonelik bilgisini ve bu aboneliğe dair haftalık takvimi gösteren ekran.



**Şekil.6:** Admin içinkullanıcı aranan ve istenen kullanıcıya ait bilgileri güncelleyen ya da yeni kullanıcı ekleme işlevini yapan ekran.



**Şekil.7:** Admin kullanıcısı için kullanıcıların abone oldukları planlara güncelleme yapabilen ve haftalık programı değiştirilebilen ekran.



**Şekil.8:** Admin kullanıcısı için profil eklenebilen ya da var olan profilleri ve bunlara bağlı kullanıcı izinlerinin güncellenebildiği ekran.

**Kaynak Dosyaları**

Projeye ait kaynak dosyaları bu rapor ile verilmiştir. Verilen dosyalarda “/db/setup.ddl” Sql Server’da istenen proje için veritabanının oluşturulmasını sağlayan kodları içermektedir. Form uygulamasının kaynak dosyaları ise “/app” dosyasının içerisinde bulunabilir.