

## ÖZET

Eczane, ilaçların firmalardan hastalara ulaştırılmasını sağlayan ve bu amaçla yapılacak faaliyetlerin gerçekleştirilmesinde önemli rol oynayan bir noktadır. Bu çalışmada bir eczane için gözlemlenen ihtiyaçlara ve iş kurallarına göre bir veri tabanı tasarımı yapmak amaçlanmıştır.

Eczane ilaç satış sistemi, genel işleyişi ve birimlerin birbirleri ile ilişkileri hakkında bilgi veren bir veri tabanı tasarımıdır. Bu çalışma temel olarak 5 aşamadan oluşmaktadır; sistemin analizi, iş kuralları ve ilişkilerin belirlenmesi, gerekli kavramsal ve mantıksal tasarımların yapılması, SQL üzerinde ilişkisel şemanın oluşturulması ve bir uygulama ara yüzünün yapılmasıdır. Eczane ilaç satış sistemi tasarımını göstermek amacıyla Chen ER diyagramı ve ilişkisel veritabanı şeması kullanılmıştır.

**Anahtar Kelimeler:** veritabanı tasarımı, eczane ilaç satış sistemi, eczane otomasyonu

## İÇİNDEKİLER

ÖZET .....	0
1. SİSTEM ANALİZİ .....	3
1.1. İş Kuralları.....	3
2. KAVRAMSAL VERİTABANI TASARIMI .....	4
3. MANTIKSAL VERİTABANI TASARIMI.....	5
4. VERİTABANI OLUŞTURMA .....	6
4.1. Veritabanı Diyagramı .....	6
4.2. Tablo Görünümleri .....	7
4.3. Tablo Verilerinin Görünümleri .....	9
5. VERİTABANINDAKİ DÜZENLEMELER .....	12
5.1. Veritabanı İçin Üretilen Gömülü Yordamlar .....	12
5.1.1. Hasta listesi üreten gömülü yordam .....	12
5.1.2. Yeni hasta kaydı yapan gömülü yordam .....	12
5.1.3. Mevcut hasta kaydını güncelleyen gömülü yordam .....	12
5.1.4. Mevcut hasta kaydını silen üreten gömülü yordam.....	13
5.1.5. İlaç listesi üreten gömülü yordam .....	13
5.1.6. Yeni ilaç kaydı yapan gömülü yordam .....	14
5.1.7. Mevcut ilaç kaydını güncelleyen gömülü yordam .....	14
5.1.8. Mevcut ilaç kaydını silen gömülü yordam.....	15
5.1.9. Reçete listesi üreten gömülü yordam .....	16
5.1.10. Yeni reçete kaydı yapan gömülü yordam.....	16
5.1.11. Mevcut reçete kaydını güncelleyen gömülü yordam .....	16
5.1.12. Mevcut reçete kaydını silen gömülü yordam.....	17
5.1.13. İlaç satış listesi üreten gömülü yordam .....	18
5.1.14. Yeni ilaç satış kaydı gömülü yordam .....	18
5.1.15. Mevcut ilaç satış kaydını güncelleyen gömülü yordam.....	18
5.1.16. Mevcut ilaç satış kaydını silen gömülü yordam .....	19
5.1.17. Yabancı anahtar listelerini üreten gömülü yordam.....	19
5.2. Gömülü Yordamlarda Kullanılmak İçin Üretilen Tablo Tipleri .....	20
5.2.1. Reçete satırlarının sunucuya gönderilmesinde kullanılacak tablo tipinin oluşturulması.....	20
5.2.2. İlaç satış satırlarının sunucuya gönderilmesinde kullanılacak tablo tipinin oluşturulması ..	21
5.3. Veritabanı İçin Üretilen Kullanıcı Tanımlı Fonksiyon.....	21
5.3.1. Ödeme tutarını hesaplayan fonksiyon .....	21
5.4. Veritabanı İçin Üretilen Tetikleyici.....	21
5.4.1. Reçete tarihini kontrol eden tetikleyici.....	21
6. ARAYÜZ YAZILIMI.....	22
6.1. Ana Form .....	22

6.1.1.Çalışan Yazılım Ekran Görüntüsü .....	22
6.1.2. Açıklama .....	22
6.1.3. Kaynak Kodları .....	22
6.2. Hasta Listesi .....	23
6.2.1.Çalışan Yazılım Ekran Görüntüsü .....	23
6.2.2. Açıklama .....	24
6.2.3. Kaynak Kodları .....	24
6.3. Hasta Formu .....	25
6.3.1.Çalışan Yazılım Ekran Görüntüsü .....	25
6.3.2. Açıklama .....	25
6.3.3. Kaynak Kodları .....	25
6.4. İlaç Listesi .....	27
6.4.1.Çalışan Yazılım Ekran Görüntüsü .....	27
6.4.2. Açıklama .....	27
6.4.3. Kaynak Kodları .....	28
6.5. İlaç Formu .....	29
6.5.1.Çalışan Yazılım Ekran Görüntüsü .....	29
6.5.2. Açıklama .....	29
6.5.3. Kaynak Kodları .....	29
6.6. Reçete Listesi .....	31
6.6.1.Çalışan Yazılım Ekran Görüntüsü .....	31
6.6.2. Açıklama .....	31
6.6.3. Kaynak Kodları .....	32
6.7. Reçete Formu .....	33
6.7.1.Çalışan Yazılım Ekran Görüntüsü .....	33
6.7.2. Açıklama .....	33
6.7.3. Kaynak Kodları .....	33
6.8. İlaç Satış Listesi .....	35
6.8.1.Çalışan Yazılım Ekran Görüntüsü .....	35
6.8.2. Açıklama .....	36
6.8.3. Kaynak Kodları .....	36
6.9. İlaç Satış Formu .....	37
6.9.1.Çalışan Yazılım Ekran Görüntüsü .....	37
6.9.2. Açıklama .....	37
6.9.3. Kaynak Kodları .....	37
SONUÇ.....	40

## 1. SİSTEM ANALİZİ

Bal eczanesi, ilaç satışlarının bilgisayar ortamında takibini sağlayan bir veritabanı uygulaması geliştirmek istemektedir. Bu uygulama ile hasta ve ilaç bilgileri kayıt altına alınacak ve ilaç satış işlemleri yapılacaktır.

Uygulamada olması istenen bazı özellikler aşağıda maddeler halinde sıralanmıştır:

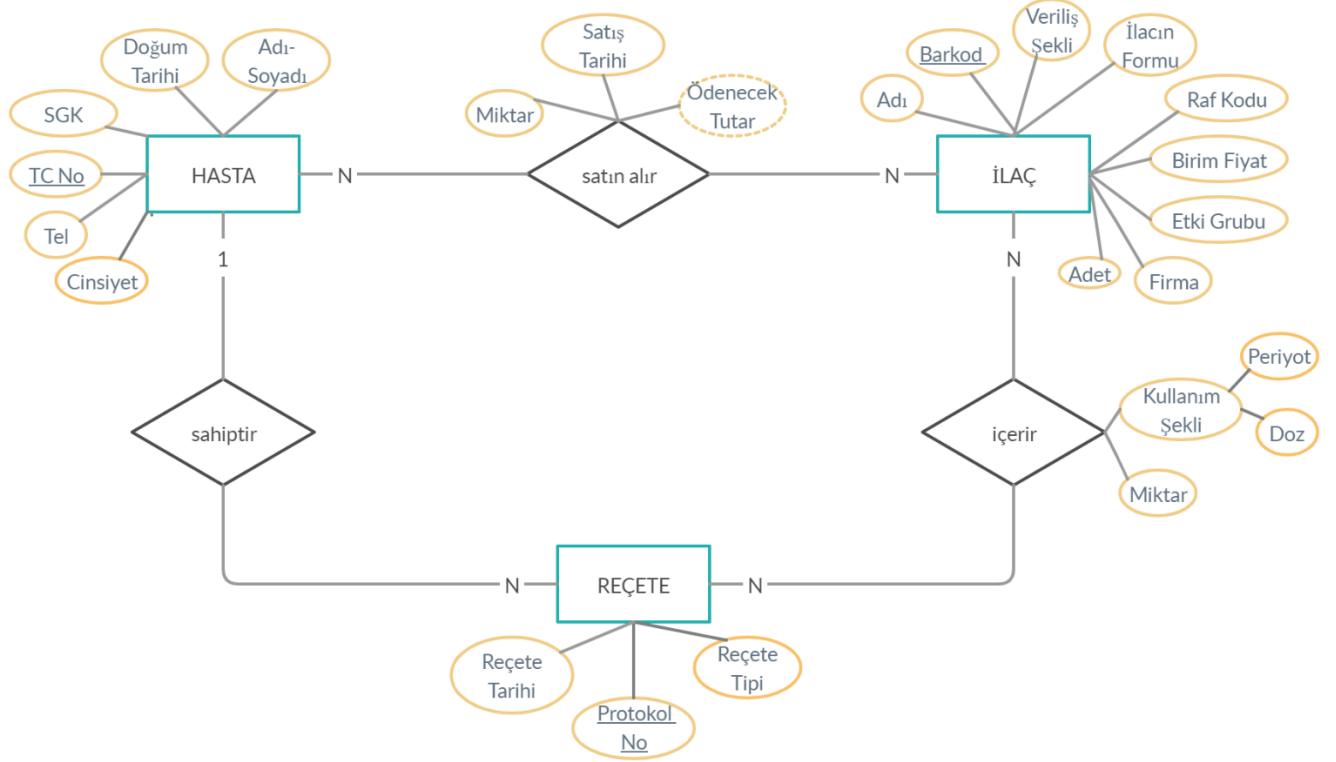
1. Hastalara yönelik bilgilerin takibi yapılmaktadır. Bu bilgiler TC kimlik numarası, adı- soyadı, doğum tarihi, cinsiyeti, telefon numarası ve bağlı olduğu SGK sistemi (Emekli sandığı, Bağ-kur, SSK) olup olmadığı, reçetesi ve aldığı ilaçlardır.
2. Reçeteli ilaç alımında hastanın reçete kaydının alınması gerekmektedir. Reçetesiz ilaç alımında ilaç veriliş şekli bilgisi eczacı tarafından kontrol edilmelidir.
3. Her bir reçetenin protokol numarası, reçete türü (normal, kırmızı, yeşil, mor, turuncu) ve reçetenin yazıldığı tarih uygulama üzerinden takip edilecektir.
4. Reçetede bulunan her bir ilacın miktarı, kullanım dozu ve kullanım periyodunun bilgileri tutulmalıdır.
5. Eczanede bulunan her bir ilacın barkodu, adı, ilacın formu (tablet, damla, sprey vs.), etki grubu (antibiyotik, analjezik vb.), veriliş şekli (reçeteli, reçetesiz, kontrole tabi vb.), üretici firma, birim satış fiyatı, bulunduğu rafın kodu ve stok adeti kayıt altında tutulmalıdır. Stok adeti ilaç girişlerinde manuel olarak güncellenmelidir.
6. Eczanede mevcut ilaç olmaması halinde aynı etki grubuna sahip bir muadili satılabilir.
7. Bir hastanın satın aldığı her bir ilacın satış tarihi, miktarı gerekli raporları almak için kayıt altına alınmalıdır. Hastanın SGK' sı yoksa ödeyeceği tutar, aldığı kutu miktarı ve ilacın birim fiyatı çarpılarak bulunur. Hastanın SGK' sı varsa bu tutarın %20'sini ödeyecektir.

### 1.1. İş Kuralları

- Her bir hastanın birden çok reçetesi olabilir, bir reçete sadece bir hastaya ait olabilir. 1: N ilişkisi vardır.
- Her bir reçetede birden çok ilaç bulunabilir, bir ilaç birden çok reçetede yazılmış olabilir. N: N ilişkisi vardır.
- Her bir hasta birden çok ilaç satın alabilir, her bir ilacı birden çok hasta satın alabilir. N: N ilişkisi vardır.

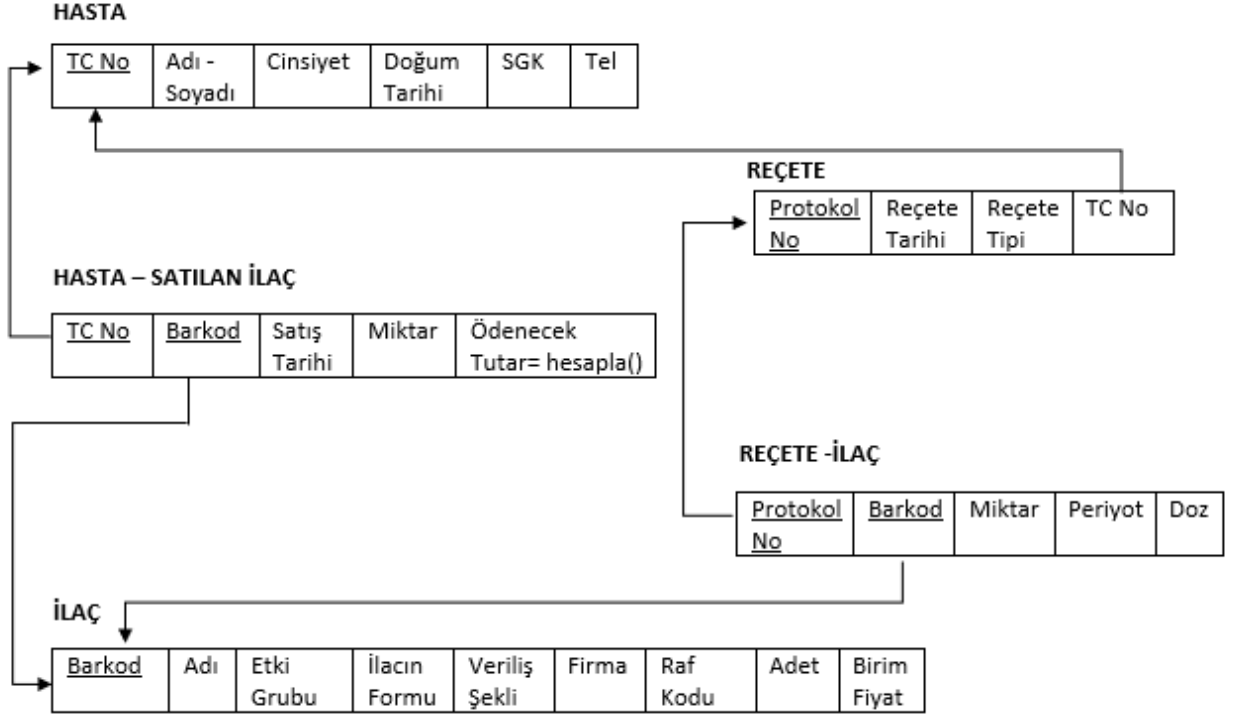
## 2. KAVRAMSAL VERİTABANI TASARIMI

Eczane ilaç satış sistemi projesi kapsamında ihtiyaçları karşılamak için geliştirilecek bilgisayar uygulamasının kullanacağı kavramsal veritabanı, Chen ER diyagramı ile gösterilmiştir.



### 3. MANTIKSAL VERİTABANI TASARIMI

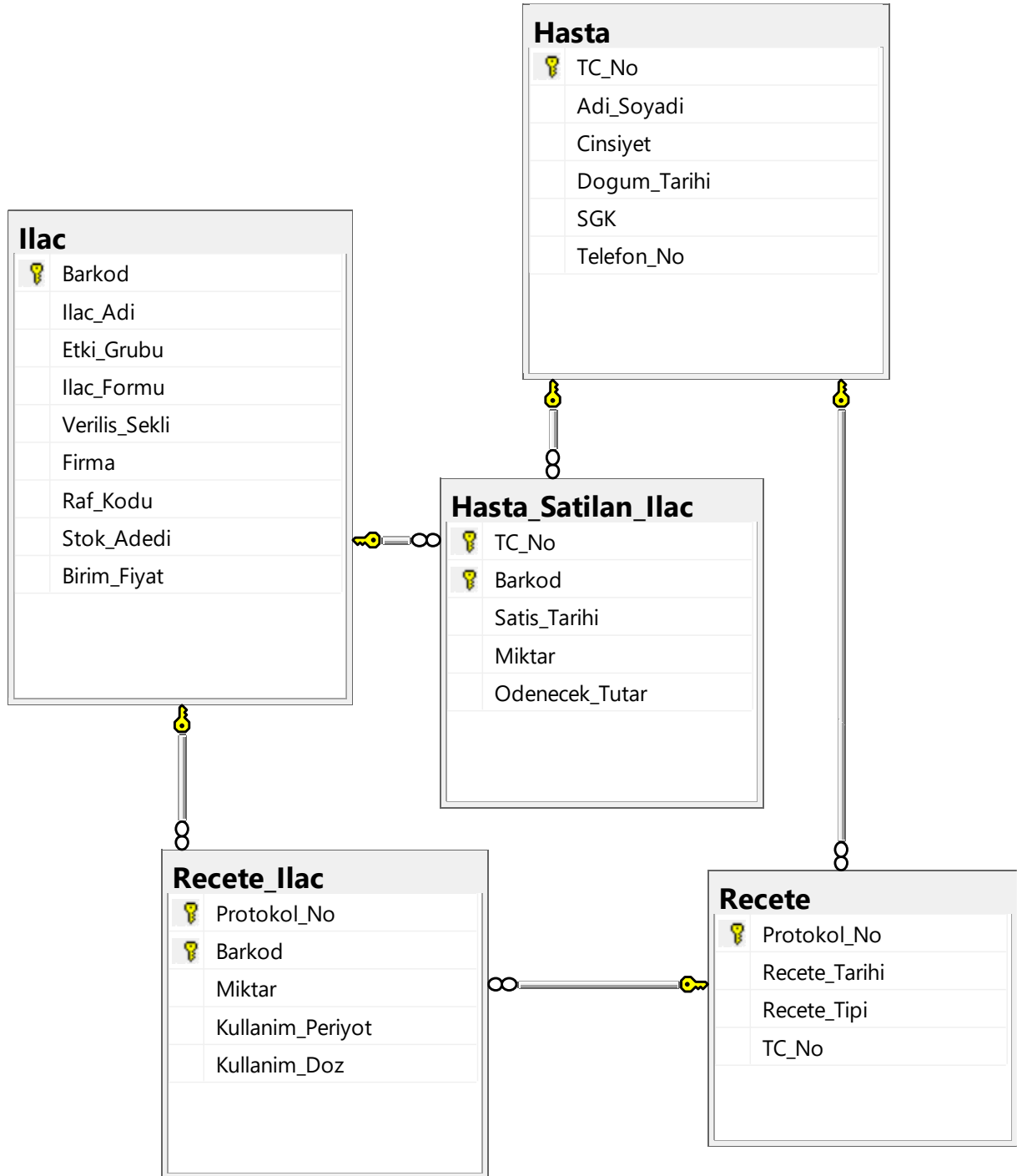
Eczane ilaç satış sisteminin Chen ER diyagramı ilişkisel veritabanı şemasına çevrilmiş ve aşağıda gösterilmiştir.



#### 4. VERİTABANI OLUŞTURMA

Veritabanı Microsoft SQL Server Management Studio 2018 kullanılarak oluşturulmuştur.

##### 4.1. Veritabanı Diyagramı



## 4.2. Tablo Görünümleri

dbo.Hasta
Columns
TC_No (PK, bigint, not null)
Adi_Soyadi (varchar(80), not null)
Cinsiyet (char(1), not null)
Dogum_Tarihi (date, not null)
SGK (varchar(20), not null)
Telefon_No (varchar(15), not null)
Keys
PK_Hasta
Constraints
CK_Cinsiyet
DF_Hasta_SGK
Triggers
Indexes
idx_adisoyadi (Non-Unique, Non-Clustered)
PK_Hasta (Clustered)
Statistics
idx_adisoyadi
PK_Hasta

dbo.Hasta_Satilan_Ilac
Columns
TC_No (PK, FK, bigint, not null)
Barkod (PK, FK, bigint, not null)
Satis_Tarihi (date, not null)
Miktar (smallint, not null)
Odenecek_Tutar (money, not null)
Keys
PK_Hasta_Satilan_Ilac
FK_Hasta_Satilan_Ilac_Hasta
FK_Hasta_Satilan_Ilac_Ilac
Constraints
DF_Hasta_Satilan_Ilac_Odenecek_Tutar
DF_Hasta_Satilan_Ilac_Satis_Tarihi
Triggers
Indexes
PK_Hasta_Satilan_Ilac (Clustered)
Statistics
_WA_Sys_00000002_116A8EFB
_WA_Sys_00000003_116A8EFB
PK_Hasta_Satilan_Ilac



[-]	[-]	[-]	dbo.Ilac
[-]		[-]	Columns
		[-]	Barkod (PK, FK, bigint, not null)
		[-]	Ilac_Adi (varchar(150), not null)
		[-]	Etki_Grubu (varchar(50), not null)
		[-]	Ilac_Formu (varchar(50), not null)
		[-]	Verilis_Sekli (varchar(50), not null)
		[-]	Firma (varchar(50), not null)
		[-]	Raf_Kodu (varchar(10), null)
		[-]	Stok_Adedi (int, not null)
		[-]	Birim_Fiyat (money, not null)
[-]		[-]	Keys
		[-]	PK_Ilac
[-]		[-]	Constraints
		[-]	DF_RafKodu
		[-]	Triggers
[-]		[-]	Indexes
		[-]	idx_IlacAdi (Unique, Non-Clustered)
		[-]	PK_Ilac (Clustered)
[-]		[-]	Statistics
		[-]	_WA_Sys_00000004_61BB7BD9
		[-]	_WA_Sys_00000005_61BB7BD9
		[-]	_WA_Sys_00000007_61BB7BD9
		[-]	idx_IlacAdi
		[-]	PK_Ilac

[-]	[-]	[-]	dbo.Recete
[-]		[-]	Columns
		[-]	Protokol_No (PK, bigint, not null)
		[-]	Recete_Tarihi (date, not null)
		[-]	Recete_Tipi (varchar(30), not null)
		[-]	TC_No (FK, bigint, not null)
[-]		[-]	Keys
		[-]	PK_Recete
		[-]	FK_Recete_Hasta
		[-]	Constraints
[-]		[-]	Triggers
		[-]	Recete_Tarih_Kontrol
[-]		[-]	Indexes
		[-]	idx_ReceteTarihi (Non-Unique, Non-Clustered)
		[-]	PK_Recete (Clustered)
[-]		[-]	Statistics
		[-]	_WA_Sys_00000003_6C390A4C
		[-]	_WA_Sys_00000004_6C390A4C
		[-]	idx_ReceteTarihi
		[-]	PK_Recete

dbo.Recete_Illac
Columns
Protokol_No (PK, FK, bigint, not null)
Barkod (PK, FK, bigint, not null)
Miktar (smallint, not null)
Kullanım_Periyot (varchar(50), not null)
Kullanım_Doz (varchar(50), not null)
Keys
PK_Recetellac
FK_Recete_Illac_Illac
FK_Recete_Illac_Recete
Constraints
Triggers
Indexes
PK_Recetellac (Clustered)
Statistics
_WA_Sys_00000002_0CA5D9DE
PK_Recetellac

### 4.3.Tablo Verilerinin Görünümleri

SQLQuery4.sql - L...eSystemAdmin (51))

SELECT \* FROM Hasta

100 %

Results Messages

	TC_No	Adi_Soyadi	Cinsiyet	Dogum_Tarihi	SGK	Telefon_No
1	10749942744	Belçim Yücel	K	2019-10-09	Bağkur	0562 812 36 50
2	11607189438	Riayet Rüzgar özen	E	2020-06-30	SSK	0522 634 00 13
3	13663833391	Bihter Dede	E	2021-04-04	Yok	0547 503 83 01
4	14548679575	Tanalp İnan	E	2020-01-01	Bağkur	0570 423 62 76
5	15111830835	Safiye Emeksiz	E	2019-07-11	Bağkur	0588 332 68 71
6	16681072363	Ehlullah Yaylacı	K	2020-06-03	Emekli Sandığı	0542 598 38 20
7	17306703444	Üçok Akpınar	K	2019-10-15	Bağkur	0572 678 97 36
8	17454218410	Oktay Işıktel	K	2020-10-16	Emekli Sandığı	0582 629 62 59
9	18165701677	Serdengeçti Tepekıran	K	2020-03-10	Bağkur	0542 600 87 23
10	18295747757	Özinal Kıpır	E	2021-03-01	Bağkur	0530 434 87 90
11	19557688313	Nebi Ağıt	K	2020-01-22	Bağkur	0584 184 91 47
12	20762497748	İlkutay Erkmən	K	2020-09-25	Bağkur	0555 479 51 47
13	22134000460	Özbal Şaksu	E	2019-09-22	SSK	0596 510 27 86
14	22256693985	Renginar Çıtak	K	2021-01-01	SSK	0594 256 27 55
15	22474835869	Mesih Akşahin	E	2021-01-15	Yok	0555 723 96 46
16	22719292446	Tahliye Salkaya	K	2020-08-19	SSK	0528 980 70 12
17	23605619503	Neval Yurdakul ölmez	E	2020-05-01	Emekli Sandığı	0500 545 88 24
18	23729144049	Aytek Okşar	E	2019-06-17	Bağkur	0586 771 87 73
19	24875153499	Ünay Alğan	K	2019-06-26	Emekli Sandığı	0513 311 30 54
20	25511401070	Neşide Özbaysar	E	2020-09-24	Emekli Sandığı	0530 425 19 18

Query executed successfully.

LAPTOP-4VG52K1E (14.0 RTM) | LAPTOP-4VG52K1E\TheSys... | EczaneDB | 00:00:00 | 101 rows

SQLQuery4.sql - L...eSystemAdmin (51))\* ✕

`SELECT * FROM Hasta_Satilan_Ilac`

100 %

Results Messages

	TC_No	Barkod	Satis_Tarihi	Miktar	Odenecek_Tutar
1	10749942744	1050617067919	2019-08-23	1	12,50
2	10749942744	1138569144553	2019-06-17	2	25,00
3	10749942744	1409211194824	2020-04-16	2	90,64
4	11607189438	1138569144553	2019-06-06	2	21,30
5	11607189438	1507218476836	2019-06-23	1	7,95
6	11607189438	1523048989790	2020-04-16	1	17,60
7	13663833391	1409211194824	2020-01-19	2	90,64
8	13663833391	1571745555265	2019-10-09	1	15,50
9	13663833391	1808692307755	2019-06-23	3	55,80
10	14548679575	1507218476836	2020-04-01	1	7,95
11	14548679575	1643111075990	2020-01-08	1	8,45
12	14548679575	2369853945143	2019-10-09	1	15,80
13	15111830835	1523048989790	2019-09-23	3	52,80
14	15111830835	1775784386240	2020-01-07	1	6,00
15	15111830835	2754240753778	2020-01-08	1	7,56
16	16681072363	1571745555265	2020-01-10	1	15,50
17	16681072363	1810428086670	2020-01-04	1	15,25
18	16681072363	2993307245341	2020-01-07	1	49,50
19	17306703444	1643111075990	2020-03-29	1	8,45
20	17306703444	2172373229302	2019-08-21	1	4,60

Query executed successfully. | LAPTOP-4VG52K1E (14.0 RTM) | LAPTOP-4VG52K1E\TheSys... | EczaneDB | 00:00:00 | 300 rows

SQLQuery4.sql - L...eSystemAdmin (51))\* ✕

`SELECT * FROM Ilac`

100 %

Results Messages

	Barkod	Ilac_Adi	Etki_Grubu	Ilac_Fomu	Verilis_Sekli	Firma	Raf_Kodu	Stok_Adedi	Birim_Fiyat
1	1050617067919	POLID 30 Mg	Antidepresan	Losyon	Reçeteli	Biosel	C-01	55	12,50
2	1138569144553	DEQUADIN 30	Bulantı	Losyon	Kontrolle Tabi	Farmed	F-01	51	10,65
3	1409211194824	EFOROL 100 mg 30	Antitussif	Sprey	Turuncu Reçeteli	Dincsa	E-03	48	45,32
4	1507218476836	ABILIFY 30 MG	Ağn Kesici	Losyon	Turuncu Reçeteli	Eczacıbaşı	F-03	40	7,95
5	1523048989790	BIOXININ FORTE %2 60 ml	Antitussif	Kapsül	Turuncu Reçeteli	Generica	F-01	87	17,60
6	1571745555265	DISINOL 10 ml	Antidepresan	Şurup	Reçetesiz	Generica	B-02	94	15,50
7	1643111075990	EVOSTEN 175 mg/5 ml 100 ml	Antitussif	Merhem	Reçetesiz	Eczacıbaşı	D-02	86	8,45
8	1775784386240	ACECAP 30	Antibiyotik	Emülsiyon	Mor Reçeteli	Dincsa	F-02	13	6,00
9	1808692307755	FULDUO 100mg/8MG	Ağn Kesici	Losyon	Reçetesiz	Abott	F-03	48	18,60
10	1810428086670	NORMOTRI FORT 5 mg	Tansiyon Düşürücü	Pomad	Kırmızı Reçeteli	Deva	F-02	98	15,25
11	2172373229302	ANTOKSIC 500 mg	Diğer	Sprey	Kontrolle Tabi	Pfizer	B-02	86	4,60
12	2216868387535	BISMOMAGNESIE 70 gr	Ateş Düşürücü	Emülsiyon	Kontrolle Tabi	Akar	C-03	60	18,60
13	2369853945143	RIZOPROL 1.25 mg (30 )	Ateş Düşürücü	Şurup	Reçeteli	Gunsa	D-02	15	15,80
14	2428133965279	VORANEX 200 mg	Bulantı	Şampuan	Yeşil Reçeteli	Gunsa	C-02	34	5,90
15	2452572778899	ANKEP 50 mg	Analjezik	Merhem	Mor Reçeteli	Gunsa	E-02	100	12,26
16	2714711652839	CANDUR 250 mg/2.5 ml	Antibiyotik	Ampul	Kontrolle Tabi	Biomeks	E-01	37	8,15
17	2754240753778	AVISEF 1 G	Antitussif	Efervesan Tablet	Mor Reçeteli	Abdi İbrahim	C-01	97	7,56
18	2803633738575	EVINOLLE 400 IU	Antitussif	Tablet	Yeşil Reçeteli	Dincsa	F-02	19	10,11
19	2910731612628	ADVANTAN 15 GR	Bulantı	Ampul	Yeşil Reçeteli	Farmed	E-01	82	8,98
20	2937689841530	FOAMIX %2 (100 ml)	Antibiyotik	Sprey	Mor Reçeteli	Dincsa	D-03	85	14,00

Query executed successfully. | LAPTOP-4VG52K1E (14.0 RTM) | LAPTOP-4VG52K1E\TheSys... | EczaneDB | 00:00:00 | 100 rows

SQLQuery4.sql - L...eSystemAdmin (51))\*

`SELECT * FROM Recete`

100 %

Results Messages

	Protokol_No	Recete_Tarihi	Recete_Tipi	TC_No
1	100148101344	2019-01-02	Turuncu	53482387704
2	102291708238	2019-11-30	Yeşil	34180273370
3	103356505571	2018-05-27	Normal	52810956053
4	103455138237	2020-04-11	Mor	52267976120
5	105090684253	2019-08-22	Turuncu	29710635548
6	117241441625	2019-11-29	Turuncu	78110674733
7	120272125406	2020-01-15	Normal	17454218410
8	124557846174	2019-02-03	Kimızı	22719292446
9	130238887207	2019-08-13	Kimızı	86091203467
10	139120915741	2019-01-09	Normal	63234674534
11	142887331659	2020-04-12	Mor	49248533851
12	149064216932	2020-04-18	Mor	79195570581
13	151650991973	2019-12-09	Yeşil	51116282807
14	152312611008	2019-12-06	Yeşil	67136196484
15	153837783382	2019-11-16	Yeşil	99382072672
16	155486232426	2019-03-10	Kimızı	29885202972
17	157706520020	2019-09-24	Turuncu	95972837721
18	159627473889	2019-09-20	Mor	27231694103
19	164104091690	2019-02-13	Yeşil	55099902010
20	170615703153	2018-09-19	Turuncu	49647169020

Query executed successfully. LAPTOP-4VG52K1E (14.0 RTM) LAPTOP-4VG52K1E\TheSys... EczaneDB 00:00:00 250 rows

SQLQuery4.sql - L...eSystemAdmin (51))\*

`SELECT * FROM Recete_Ilac`

100 %

Results Messages

	Protokol_No	Barkod	Miktar	Kullanım_Periyot	Kullanım_Doz
1	100148101344	1050617067919	1	Sabah	2 doz
2	100148101344	6230513169550	2	Günde 6	1x6 doz
3	102291708238	1138569144553	2	Akşam	1 doz
4	102291708238	6269687085412	1	Gün Aşın	1 doz
5	103356505571	1409211194824	3	Sabah Akşam	1x2 doz
6	103356505571	6306710475296	1	Sabah	2 doz
7	103455138237	1507218476836	1	Günde 3	2x3 doz
8	103455138237	6326369279651	1	Akşam	1 doz
9	105090684253	1523048989790	2	Günde 2	1x2 doz
10	105090684253	6501433880492	1	Sabah Akşam	1x2 doz
11	117241441625	1571745555265	3	Günde 6	1x6 doz
12	117241441625	6514727547151	2	Günde 3	2x3 doz
13	120272125406	1643111075990	1	Gün Aşın	1 doz
14	120272125406	6596318579128	1	Günde 2	1x2 doz
15	124557846174	1775784386240	1	Sabah	2 doz
16	124557846174	6947198673184	3	Günde 6	1x6 doz
17	130238887207	1808692307755	2	Akşam	1 doz
18	130238887207	7061969655778	1	Gün Aşın	1 doz
19	139120915741	1810428086670	2	Sabah Akşam	1x2 doz
20	139120915741	7091888882068	2	Sabah	2 doz

Query executed successfully. LAPTOP-4VG52K1E (14.0 RTM) LAPTOP-4VG52K1E\TheSys... EczaneDB 00:00:00 300 rows

## 5. VERİTABANINDAKİ DÜZENLEMELER

### 5.1. Veritabanı İçin Üretilen Gömülü Yordamlar

#### 5.1.1. Hasta listesi üreten gömülü yordam

```
CREATE PROCEDURE [dbo].[HastaListe]
AS
SELECT
TC_No AS [TC Kimlik No],
UPPER(Adi_Soyadi) AS [Ad Soyad],
(CASE Cinsiyet
WHEN 'E' THEN 'Erkek'
WHEN 'K' THEN 'Kadın'
ELSE 'Kaydedilmemiş'
END) AS Cinsiyet,
Dogum_Tarihi AS [Doğum Tarihi],
Telefon_No AS [Telefon No]
FROM Hasta AS H
ORDER BY H.TC_No ASC;
```

#### 5.1.2. Yeni hasta kaydı yapan gömülü yordam

```
CREATE PROCEDURE [dbo].[HastaEkle]
@TC_No bigint,
@Adi_Soyadi varchar(80),
@Cinsiyet char(1),
@Dogum_Tarihi date,
@SGK varchar(20),
@Telefon_No varchar(15)
AS
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
IF(EXISTS(SELECT TOP 1 * FROM Hasta WHERE TC_No=@TC_No))
THROW 50001, 'Girilen TC kimlik numarasına sahip hasta zaten kayıtlı!',1;
IF(LEN(LTRIM(RTRIM(@Adi_Soyadi)))<1)
THROW 50001, 'Ad Soyad bilgisi belirtilmemiş!', 1;

INSERT INTO Hasta(TC_No, Adi_Soyadi, Cinsiyet, Dogum_Tarihi, SGK, Telefon_No)
VALUES(@TC_No, @Adi_Soyadi,@Cinsiyet,@Dogum_Tarihi,@SGK,@Telefon_No);

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem Gerçekleştirildi' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, '(Hata): '+ ERROR_MESSAGE() AS Mesaj;
END CATCH
```

#### 5.1.3. Mevcut hasta kaydını güncelleyen gömülü yordam

```
CREATE PROCEDURE [dbo].[HastaGuncelle]
@TC_No bigint,
@Adi_Soyadi varchar(80),
@Cinsiyet char(1),
@Dogum_Tarihi date,
@SGK varchar(20),
@Telefon_No varchar(15)
AS
BEGIN TRY
BEGIN TRANSACTION;
```

```

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
IF(NOT EXISTS(SELECT TOP 1 * FROM Hasta WHERE TC_No=@TC_No))
THROW 50001, 'Hasta kaydı bulunamadı!.',1;
IF(LEN(LTRIM(RTRIM(@Adi_Soyadi)))<1)
THROW 50001, 'Ad Soyad bilgisi belirtilmemiş!.', 1;

UPDATE Hasta SET Adi_Soyadi=@Adi_Soyadi, Cinsiyet=@Cinsiyet,
Dogum_Tarihi=@Dogum_Tarihi, SGK=@SGK, Telefon_No=@Telefon_No where @TC_No=TC_No;

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem Gerçekleştirildi' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, '(Hata): '+ ERROR_MESSAGE() AS Mesaj;
END CATCH

```

#### 5.1.4. Mevcut hasta kaydını silen üreten gömülü yordam

```

CREATE PROCEDURE [dbo].[HastaSil]
@TC_No bigint
AS
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
IF(NOT EXISTS(SELECT TOP 1 * FROM Hasta WHERE TC_No=@TC_No))
THROW 50001, 'Hasta kaydı bulunamadı!.',1;
IF(EXISTS(SELECT TOP 1 * FROM Recete WHERE Recete.TC_No=@TC_No))
THROW 50001, 'Hastanın kayıtlı reçeteleri var, öncelikle onları silin!.',1;
IF(EXISTS(SELECT TOP 1 * FROM Hasta_Satilan_Ilac WHERE
Hasta_Satilan_Ilac.TC_No=@TC_No))
THROW 50001, 'Hastanın satın aldığı ilaç kayıtları var, öncelikle onları silin!.',1;

DELETE FROM Hasta WHERE TC_No=@TC_No;

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem Gerçekleştirildi' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, '(Hata): '+ ERROR_MESSAGE() AS Mesaj;
END CATCH

```

#### 5.1.5. İlaç listesi üreten gömülü yordam

```

CREATE PROCEDURE [dbo].[IlacListe]
AS
SELECT
Barkod AS [Barkod],
UPPER(Ilac_Adi) AS [Ilacın Adı],
Etki_Grubu AS [Etki Grubu],
Ilac_Formu AS [Ilacın Formu],
Verilis_Sekli AS [Veriliş Şekli],
Firma AS Firma,
UPPER(Raf_Kodu) AS [Raf Kodu],
Stok_Adedi AS [Stok Miktarı],
Birim_Fiyat AS [Birim Fiyat]
FROM Ilac As I
ORDER BY I.Ilac_Adi ASC;

```

### 5.1.6. Yeni ilaç kaydı yapan gömülü yordam

```
CREATE PROCEDURE [dbo].[IlacEkle]
@Barkod bigint,
@Ilac_Adi varchar(150),
@Etki_Grubu varchar(50),
@Ilac_Formu varchar(50),
@Verilis_Sekli varchar(50),
@Firma varchar(50),
@Raf_Kodu varchar(10),
@Stok_Adedi int,
@Birim_Fiyat money

AS
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
IF(EXISTS(SELECT TOP 1 * FROM Ilac WHERE Barkod=@Barkod))
THROW 50001, 'Bu barkoda sahip ilaç zaten kayıtlı!..', 1;
IF(LEN(LTRIM(RTRIM(@Ilac_Adi)))<1)
THROW 50001, 'İlacın ad bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Etki_Grubu)))<1)
THROW 50001, 'İlacın etki grubu belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Ilac_Formu)))<1)
THROW 50001, 'İlacın formu belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Verilis_Sekli)))<1)
THROW 50001, 'İlacın veriliş şekli bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Firma)))<1)
THROW 50001, 'İlacın firma bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Stok_Adedi)))<1)
THROW 50001, 'İlacın stok bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Birim_Fiyat)))<1)
THROW 50001, 'İlacın fiyat bilgisi belirtilmemiş!..', 1;

INSERT INTO
Ilac(Barkod,Ilac_Adi,Etki_Grubu,Ilac_Formu,Verilis_Sekli,Firma,Raf_Kodu,Stok_Adedi,Birim_Fiyat)
VALUES(@Barkod,
@Ilac_Adi,@Etki_Grubu,@Ilac_Formu,@Verilis_Sekli,@Firma,@Raf_Kodu,@Stok_Adedi,@Birim_Fiyat);

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem Gerçekleştirildi' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, '(Hata): ' + ERROR_MESSAGE() AS Mesaj;
END CATCH
```

### 5.1.7. Mevcut ilaç kaydını güncelleyen gömülü yordam

```
CREATE PROCEDURE [dbo].[IlacGuncelle]
@Barkod bigint,
@Ilac_Adi varchar(150),
@Etki_Grubu varchar(50),
@Ilac_Formu varchar(50),
@Verilis_Sekli varchar(50),
@Firma varchar(50),
@Raf_Kodu varchar(10),
@Stok_Adedi int,
@Birim_Fiyat money
```

```

AS
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
IF(NOT EXISTS(SELECT TOP 1 * FROM Ilac WHERE Barkod=@Barkod))
THROW 50001, 'İlaç kaydı bulunamadı!..',1;
IF(EXISTS(SELECT TOP 1 * FROM Ilac WHERE @Ilac_Adi=@Ilac_Adi AND Barkod<>@Barkod))
THROW 50001, 'Aynı isme sahip başka bir ilaç zaten var!..',1;
IF(LEN(LTRIM(RTRIM(@Ilac_Adi)))<1)
THROW 50001, 'İlacın ad bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Etki_Grubu)))<1)
THROW 50001, 'İlacın etki grubu belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Ilac_Formu)))<1)
THROW 50001, 'İlacın formu belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Verilis_Sekli)))<1)
THROW 50001, 'İlacın veriliş şekli bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Firma)))<1)
THROW 50001, 'İlacın firma bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Stok_Adedi)))<1)
THROW 50001, 'İlacın stok bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Birim_Fiyat)))<1)
THROW 50001, 'İlacın fiyat bilgisi belirtilmemiş!..', 1;

UPDATE Ilac SET Ilac_Adi=@Ilac_Adi, Etki_Grubu=@Etki_Grubu, Ilac_Formu=Ilac_Formu,
Verilis_Sekli=@Verilis_Sekli, Firma=@Firma,Raf_Kodu=@Raf_Kodu, Stok_Adedi=@Stok_Adedi
where Barkod=@Barkod ;

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem Gerçekleştirildi' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, '(Hata): ' + ERROR_MESSAGE() AS Mesaj;
END CATCH

```

### 5.1.8. Mevcut ilaç kaydını silen gömülü yordam

```

CREATE PROCEDURE [dbo].[IlacSil]
@Barkod bigint
AS
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
IF(NOT EXISTS(SELECT TOP 1 * FROM Ilac WHERE Barkod=@Barkod))
THROW 50001, 'İlaç kaydı bulunamadı!..',1;
IF(EXISTS(SELECT TOP 1 * FROM Hasta_Satilan_Ilac WHERE Barkod=@Barkod))
THROW 50001, 'Bu ilacın satış kayıtları var, öncelikle onları silin!..',1;
IF(EXISTS(SELECT TOP 1 * FROM Recete_Ilac WHERE Barkod=@Barkod))
THROW 50001, 'Bu ilacın bulunduğu reçeteler var, öncelikle onları silin!..',1;

DELETE FROM Ilac WHERE Barkod=@Barkod;

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem Gerçekleştirildi' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, '(Hata): ' + ERROR_MESSAGE() AS Mesaj;
END CATCH

```



### 5.1.9. Reçete listesi üreten gömülü yordam

```
CREATE PROCEDURE [dbo].[ReceteListe]
AS
SELECT
RI.Protokol_No AS [Protokol No],
R.Recete_Tarihi AS [Reçete Tarihi],
R.Recete_Tipi AS [Reçete Tipi],
R.TC_No AS [TC Kimlik Numarası],
UPPER(H.Adi_Soyadi) AS [Ad Soyad]
FROM Recete AS R, Hasta AS H, Recete_Ilac AS RI
WHERE
R.TC_No=H.TC_No AND R.Protokol_No=RI.Protokol_No
GROUP BY RI.Protokol_No, R.Recete_Tarihi, R.Recete_Tipi,R.TC_No, H.Adi_Soyadi
ORDER BY R.Recete_Tarihi DESC;
```

### 5.1.10. Yeni reçete kaydı yapan gömülü yordam

```
CREATE PROCEDURE [dbo].[ReceteEkle]
@Protokol_No bigint,
@Recete_Tarihi date,
@Recete_Tipi varchar(50),
@TC_No bigint,
@Recete_Ilac RcTabloTipi READONLY

AS
SET NOCOUNT ON;
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
IF(NOT EXISTS(SELECT TOP 1 * FROM Hasta WHERE TC_No=@TC_No))
THROW 50001, 'Hasta kaydı bulunamadı!.Önce hasta kaydı yapın.',1;
IF(LEN(LTRIM(RTRIM(@Recete_Tipi)))<1)
THROW 50001, 'Reçete tipi bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Recete_Tarihi)))<1)
THROW 50001, 'Reçete tarihi bilgisi belirtilmemiş!..', 1;
IF(NOT EXISTS(SELECT TOP 1 * FROM @Recete_Ilac))
THROW 50001, 'Reçete hiç ilaç içermiyor', 1;
IF(EXISTS(SELECT * FROM @Recete_Ilac WHERE (Barkod IS NULL) OR (Miktar IS NULL)OR
(Miktar<=0) OR (Kullanim_Periyyot IS NULL) OR
(Kullanim_Doz IS NULL) OR (NOT EXISTS(SELECT * FROM Ilac WHERE Ilac.Barkod=Barkod))))
THROW 50001, 'İlaç seçilmemiş, Miktar yanlış girilmiş!', 1;

INSERT INTO Recete(Protokol_No,TC_No,Recete_Tarihi,Recete_Tipi)
VALUES(@Protokol_No,@TC_No,@Recete_Tarihi,@Recete_Tipi);

INSERT INTO Recete_Ilac (Protokol_No,Barkod , Miktar, Kullanim_Periyyot,Kullanim_Doz)
SELECT @Protokol_No,Barkod,Miktar, Kullanim_Periyyot,Kullanim_Doz FROM @Recete_Ilac

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem Gerçekleştirildi' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, '(Hata): '+ ERROR_MESSAGE() AS Mesaj;
END CATCH
```

### 5.1.11. Mevcut reçete kaydını güncelleyen gömülü yordam

```
CREATE PROCEDURE [dbo].[ReceteGuncelle]
@Protokol_No bigint,
@TC_No bigint,
```

```

@Recete_Tarihi date,
@Recete_Tipi varchar(50),
@Recete_Ilac RcTabloTipi READONLY

AS
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
IF(NOT EXISTS(SELECT TOP 1 * FROM Recete WHERE Protokol_No=@Protokol_No))
THROW 50001, 'Reçete kaydı bulunamadı!..', 1;
IF(NOT EXISTS(SELECT TOP 1 * FROM Hasta WHERE TC_No=@TC_No))
THROW 50001, 'Hasta tanımlı değil!..', 1;
IF(LEN(LTRIM(RTRIM(@Recete_Tarihi)))<1)
THROW 50001, 'Recete Tarihi bilgisi belirtilmemiş!..', 1;
IF(LEN(LTRIM(RTRIM(@Recete_Tipi)))<1)
THROW 50001, 'Recete tipi bilgisi belirtilmemiş!..', 1;
IF(EXISTS(SELECT * FROM @Recete_Ilac WHERE (Barkod IS NULL) OR (Miktar IS NULL) OR
(Kullanım_Doz IS NULL) OR (Miktar<=0) OR (Kullanım_Periyyot IS NULL) OR
(NOT EXISTS(SELECT * FROM Ilac WHERE Barkod=Ilac.Barkod))
)) THROW 50001, 'İlaç seçilmemiş, miktar yanlış girilmiş!', 1;

UPDATE Recete SET Protokol_No=@Protokol_No, Recete_Tarihi=@Recete_Tarihi,
Recete_Tipi=@Recete_Tipi
WHERE TC_No=@TC_No and Protokol_No=@Protokol_No;

DELETE Recete_Ilac WHERE Protokol_No=@Protokol_No AND Barkod NOT IN (SELECT Barkod
FROM @Recete_Ilac)

UPDATE
Recete_Ilac
SET Recete_Ilac.Barkod=RI2.Barkod,
Recete_Ilac.Miktar=RI2.Miktar,
Recete_Ilac.Kullanım_Periyyot=RI2.Kullanım_Periyyot,
Recete_Ilac.Kullanım_Doz=RI2.Kullanım_Doz
FROM Recete_Ilac AS RI1, @Recete_Ilac AS RI2

INSERT INTO Recete_Ilac(Barkod, Miktar, Kullanım_Periyyot, Kullanım_Doz)
SELECT RI.Barkod, RI.Miktar, RI.Kullanım_Periyyot, RI.Kullanım_Doz FROM @Recete_Ilac AS
RI WHERE RI.Barkod NOT IN (SELECT Barkod FROM Recete_Ilac WHERE
Protokol_No=@Protokol_No);

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem Gerçekleştirildi' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, '(Hata): ' + ERROR_MESSAGE() AS Mesaj;
END CATCH

```

### 5.1.12. Mevcut reçete kaydını silen gömülü yordam

```

CREATE PROCEDURE [dbo].[ReceteSil]
@Protokol_No bigint
AS
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
IF(NOT EXISTS(SELECT TOP 1 * FROM Recete WHERE Protokol_No=@Protokol_No))
THROW 50001, 'Reçete kaydı bulunamadı!..', 1;
IF(EXISTS(SELECT TOP 1 * FROM Recete_Ilac WHERE Protokol_No=@Protokol_No))
THROW 50001, 'Reçetede kayıtlı ilaçlar var, öncelikle onları silin!..', 1;
DELETE FROM Recete_Ilac WHERE Protokol_No=@Protokol_No;

```

```

DELETE FROM Recete WHERE Protokol_No=@Protokol_No;

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem Gerçekleştirdi' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, '(Hata): '+ ERROR_MESSAGE() AS Mesaj;
END CATCH

```

### 5.1.13. İlaç satış listesi üreten gömülü yordam

```

CREATE PROCEDURE [dbo].[SatilanIlacListe]
AS
SELECT
H.TC_No AS [TC Kimlik No],
H.Adi_Soyadi AS [Ad Soyad],
HS.Satis_Tarihi AS [Satış Tarihi],
I.Barkod AS[Barkod],
I.Ilac_Adi AS [İlaç],
HS.Odenecek_Tutar AS [Toplam Tutar]
FROM Hasta AS H, Hasta_Satilan_Ilac AS HS,Ilac AS I
WHERE
H.TC_No=HS.TC_No AND HS.Barkod=I.Barkod
ORDER BY hs.Satis_Tarihi DESC

```

### 5.1.14. Yeni ilaç satış kaydı gömülü yordam

```

CREATE PROCEDURE [dbo].[SatilanIlacEkle]
@TC_No bigint,
@SatilanIlac IlacSatisTabloTipi READONLY
AS
SET NOCOUNT ON;
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
IF(NOT EXISTS(SELECT TOP 1 * FROM Hasta WHERE TC_No=@TC_No))
THROW 50001, 'Hasta kayıtlı değil, önce kayıt edin', 1;
IF(NOT EXISTS(SELECT TOP 1 * FROM @SatilanIlac))
THROW 50001, 'Satış hiç ilaç içermiyor', 1;

INSERT INTO Hasta_Satilan_Ilac(TC_No,Barkod,Satis_Tarihi,Miktar,Odenecek_Tutar)
SELECT @TC_No,Barkod,(GETDATE()),Miktar,(dbo.odenecektutar()) FROM @SatilanIlac ;

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem tamam' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, ERROR_MESSAGE() AS Mesaj;
END CATCH

```

### 5.1.15. Mevcut ilaç satış kaydını güncelleyen gömülü yordam

```

CREATE PROCEDURE [dbo].[SatilanIlacGuncelle]
@TC_No bigint,
@SatilanIlac IlacSatisTabloTipi READONLY
AS
SET NOCOUNT ON;
BEGIN TRY

```

```

BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
IF(NOT EXISTS(SELECT TOP 1 * FROM Hasta WHERE TC_No=@TC_No))
THROW 50001, 'Hasta bulunamadı', 1;
IF(NOT EXISTS(SELECT TOP 1 * FROM @SatilanIlac))
THROW 50001, 'Satış hiç ilaç içermiyor', 1;

DELETE Hasta_Satilan_Ilac WHERE TC_No=@TC_No AND Barkod NOT IN (SELECT Barkod FROM
@SatilanIlac)

UPDATE Hasta_Satilan_Ilac
SET Hasta_Satilan_Ilac.Barkod=HSI2.Barkod,
Hasta_Satilan_Ilac.Satis_Tarihi=HSI2.Satis_Tarihi,
Hasta_Satilan_Ilac.Miktar=HSI2.Miktar,
Hasta_Satilan_Ilac.Odenecek_Tutar=HSI2.Odenecek_Tutar
FROM Hasta_Satilan_Ilac as HSI1, @SatilanIlac as HSI2
WHERE HSI1.TC_No=HSI2.TC_No;

INSERT INTO Hasta_Satilan_Ilac(TC_No, Barkod,Satis_Tarihi, Miktar, Odenecek_Tutar)
SELECT @TC_No, HSI.Barkod,HSI.Satis_Tarihi, HSI.Miktar, HSI.Odenecek_Tutar FROM
@SatilanIlac AS HSI WHERE HSI.Barkod NOT IN (SELECT Barkod FROM Hasta_Satilan_Ilac
WHERE TC_No=@TC_No);

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem tamam' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, ERROR_MESSAGE() AS Mesaj;
END CATCH

```

#### 5.1.16. Mevcut ilaç satış kaydını silen gömülü yordam

```

CREATE PROCEDURE [dbo].[SatilanIlacSil]
@TC_No bigint
AS
SET NOCOUNT ON;
BEGIN TRY
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
IF(NOT EXISTS(SELECT TOP 1 * FROM Hasta_Satilan_Ilac WHERE TC_No=@TC_No))
THROW 50001, 'Satış bulunamadı', 1;

DELETE Hasta_Satilan_Ilac WHERE TC_No=@TC_No ;

COMMIT TRANSACTION;
SELECT 0 AS Sonuc, 'İşlem tamam' AS Mesaj;
END TRY
BEGIN CATCH
IF(@@TRANCOUNT>0)
ROLLBACK TRANSACTION;
SELECT 1 AS Sonuc, ERROR_MESSAGE() AS Mesaj;
END CATCH

```

#### 5.1.17. Yabancı anahtar listelerini üreten gömülü yordam

```

CREATE PROCEDURE [dbo].[ComboBoxListesi]
@ListeAdi varchar(50)
AS
IF (UPPER(@ListeAdi)='CINSIYET')
BEGIN
SELECT ' K' AS Deger, 'Kadın' AS Baslik

```

```

UNION
SELECT ' E', 'Erkek'
RETURN
END
ELSE IF (UPPER(@ListeAdi)='SGK')
BEGIN
SELECT ' Yok' AS Deger, 'Yok' AS Baslik
UNION
SELECT ' Emekli Sandığı', 'Emekli Sandığı'
UNION
SELECT ' Bağkur', 'Bağkur'
UNION
SELECT ' SSK', 'SSK'
RETURN
END
ELSE IF (UPPER(@ListeAdi)='ILACFORMU')
BEGIN
SELECT DISTINCT Ilac_Formu AS Deger, Ilac_Formu AS Baslik
FROM Ilac
ORDER BY Ilac_Formu
RETURN
END
ELSE IF (UPPER(@ListeAdi)='VERILISSEKLI')
BEGIN
SELECT DISTINCT Verilis_Sekli AS Deger, Verilis_Sekli AS Baslik
FROM Ilac
ORDER BY Verilis_Sekli
RETURN
END
ELSE IF (UPPER(@ListeAdi)='RAFKODU')
BEGIN
SELECT ' ' AS Deger, 'DEPO' AS Baslik
UNION
SELECT Raf_Kodu AS Deger, Raf_Kodu AS Baslik
FROM Ilac
RETURN
END
ELSE IF (UPPER(@ListeAdi)='RECETETIPI')
BEGIN
SELECT DISTINCT Recete_Tipi AS Deger, Recete_Tipi AS Baslik
FROM Recete
ORDER BY Recete_Tipi
RETURN
END
ELSE IF (UPPER(@ListeAdi)='ILACLISTESI')
BEGIN
SELECT
Barkod AS Deger, Ilac_Adi AS Baslik
FROM Ilac
ORDER BY Ilac_Adi
RETURN
END

```

## 5.2. Gömülü Yordamlarda Kullanılmak İçin Üretilen Tablo Tipleri

### 5.2.1. Reçete satırlarının sunucuya gönderilmesinde kullanılacak tablo tipinin oluşturulması

```

CREATE TYPE RcTabloTipi AS TABLE(
[Protokol_No] [bigint] NOT NULL,
[Barkod] [bigint] NOT NULL,
[Miktar] [smallint] NOT NULL,
[Kullanım_Periyo] [varchar(50)] NOT NULL,
[Kullanım_Doz] [varchar(50)] NOT NULL)

```

### 5.2.2. İlaç satış satırlarının sunucuya gönderilmesinde kullanılacak tablo tipinin oluşturulması

```
CREATE TYPE IlacSatisTabloTipi AS TABLE(  
[TC_No] [bigint] NULL,  
[Barkod] [bigint] NULL,  
[Satis_Tarihi][date]NULL,  
[Miktar] [smallint] NULL,  
[Odenecek_Tutar] [money] NULL )
```

### 5.3. Veritabanı İçin Üretilen Kullanıcı Tanımlı Fonksiyon

#### 5.3.1. Ödeme tutarını hesaplayan fonksiyon

```
CREATE FUNCTION [dbo].[odenecektutar] ()  
RETURNS money  
AS  
BEGIN  
DECLARE @odemetutar money;  
DECLARE @sgk varchar(20), @bfiyat money,@TCno bigint, @barkod bigint, @miktar smallint  
SELECT @TCno=Hasta.TC_No,@barkod=Hasta_Satilan_Ilac.Barkod,@miktar=Miktar, @sgk=SGK,  
@bfiyat=Ilac.Birim_Fiyat  
FROM Hasta inner join Hasta_Satilan_Ilac ON Hasta.TC_No=Hasta_Satilan_Ilac.TC_No  
inner join Ilac ON Hasta_Satilan_Ilac.Barkod= Ilac.Barkod  
IF @sgk = ' Yok'  
BEGIN  
SELECT @odemetutar = (@bfiyat*@miktar)  
END  
ELSE  
BEGIN  
SELECT @odemetutar = (@bfiyat*@miktar)/5;  
END  
RETURN @odemetutar;  
END
```

### 5.4. Veritabanı İçin Üretilen Tetikleyici

#### 5.4.1. Reçete tarihini kontrol eden tetikleyici

```
CREATE TRIGGER [dbo].[Recete_Tarih_Kontrol] ON [dbo].[Recete]  
AFTER INSERT , UPDATE  
AS  
IF UPDATE(recete_tarihi)  
BEGIN  
DECLARE @FARK INT  
SELECT @FARK = DATEDIFF(dd, Recete_Tarihi,GETDATE())  
FROM INSERTED  
IF @FARK >= 15  
BEGIN  
RAISERROR('15 Gün Öncesinin Reçetesi Artık Geçerli Değildir.İlaç Satılamaz!',16, 1)  
ROLLBACK  
RETURN  
END  
END
```

## 6. ARAYÜZ YAZILIMI

### 6.1. Ana Form

#### 6.1.1.Çalışan Yazılım Ekran Görüntüsü



#### 6.1.2. Açıklama

Ana formda 4 adet buton ile hasta listesine, ilaç listesine, reçete listesine ve ilaç satışlarını kaydeden İlaç satış listesine bağlantı yapılmıştır.

#### 6.1.3. Kaynak Kodları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ECZANE
{
    public partial class AnaForm : Form
    {
        public AnaForm()
        {
            InitializeComponent();
        }
    }
}
```

```

{
    InitializeComponent();
}
private void HastalListesiGetir_Click(object sender, EventArgs e)
{
    HastaListe hstlst = new HastaListe();
    hstlst.Show();
}
private void IlacListesiGetir_Click(object sender, EventArgs e)
{
    IlacListe ilclst = new IlacListe();
    ilclst.Show();
}
private void Recete_Click(object sender, EventArgs e)
{
    ReceteListe rctlst = new ReceteListe();
    rctlst.Show();
}
private void IlacSatıs_Click(object sender, EventArgs e)
{
    IlacSatısListesi isl = new IlacSatısListesi();
    isl.Show();
}
}
}

```

## 6.2. Hasta Listesi

### 6.2.1.Çalışan Yazılım Ekran Görüntüsü

Hasta Listesi					
	TC Kimlik No	Ad Soyad	Cinsiyet	Doğum Tarihi	Telefon No
▶	10749942744	BELÇİM YÜCEL	Kadın	9.10.2019	0562 812 36 50
	11607189438	RİAYET RÜZGA...	Erkek	30.06.2020	0522 634 00 13
	13663833391	BİHTER DEDE	Erkek	4.04.2021	0547 503 83 01
	14548679575	TANALP İNAN	Erkek	1.01.2020	0570 423 62 76
	15111830835	SAFİYE EMEKSİZ	Erkek	11.07.2019	0588 332 68 71
	16681072363	EHLULLAH YAY...	Kadın	3.06.2020	0542 598 38 20
	17306703444	ÜÇÖK AKPINAR	Kadın	15.10.2019	0572 678 97 36
	17454218410	OKTAY İŞİKTEL	Kadın	16.10.2020	0582 629 62 59
	18165701677	SERDENGEÇTİ ...	Kadın	10.03.2020	0542 600 87 23
	18295747757	ÖZİNAL KAPAR	Erkek	1.03.2021	0530 434 87 90
	19557688313	NEBİ AŞUT	Kadın	22.01.2020	0584 184 91 47
	20762497748	İLKUTAY ERKM...	Kadın	25.09.2020	0555 479 51 47
	22134000460	ÖZBAL ŞAKSU	Erkek	22.09.2019	0596 510 27 86
	22256693985	RENGİNAR ÇITAK	Kadın	1.01.2021	0594 256 27 55
	22474835869	MESİH AKŞAHİN	Erkek	15.01.2021	0555 723 96 46
	22719292446	TAHLİYE SALK...	Kadın	19.08.2020	0528 980 70 12
	23605619503	NEVAL YURDAK...	Erkek	1.05.2020	0500 545 88 24

Hasta Ekle
Degistir
Sil



### 6.2.2. Açıklama

Hasta listesi, hastanın TC kimlik numarasını, adını ve soyadını, cinsiyetini, doğum tarihini ve telefon numarasını içeren bir listedir. Hasta ekleme, güncelleme ve silme işlemlerini yapmak için 3 buton eklenmiştir.

### 6.2.3. Kaynak Kodları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ECZANE
{
    public partial class HastaListe : Form
    {
        public HastaListe()
        {
            InitializeComponent();
            ListeGetir();
        }
        void ListeGetir()
        {
            HastaListeGrid.DataSource =
VeriTabani_SQLServer.ProsedurCalistir_TabloGelir("HastaListe", null);
        }
        private void YeniHastaKayit_Click(object sender, EventArgs e)
        {
            HastaFormu hstfrm = new HastaFormu();
            if (hstfrm.yeniKayiEkle())
                ListeGetir();
        }
        private void HastaGuncelle_Click(object sender, EventArgs e)
        {
            if(HastaListeGrid.SelectedRows.Count<1)
            {
                MessageBox.Show("Önce Güncellenecek Kaydı Seçin.");
                return;
            }
            long f_HastaGuncelle =
Convert.ToInt64(HastaListeGrid.SelectedRows[0].Cells["TC Kimlik No"].Value);
            HastaFormu hstfrm = new HastaFormu();
            if (hstfrm.kayitGuncelle(f_HastaGuncelle))
                ListeGetir();
        }
        private void HastaSil_Click(object sender, EventArgs e)
        {
            if(HastaListeGrid.SelectedRows.Count<1)
            {
                MessageBox.Show("Önce Silinecek Kaydı Seçin.");
                return;
            }
            long f_TCno =Convert.ToInt64(HastaListeGrid.SelectedRows[0].Cells["TC
Kimlik No"].Value);
            HastaFormu hstfrm = new HastaFormu();
        }
    }
}
```

```

        if (hstfrm.kayitSil(f_TCno))
            ListeGetir();
    }
}

```

### 6.3. Hasta Formu

#### 6.3.1.Çalışan Yazılım Ekran Görüntüsü

	TC Kimlik No	Ad Soyad	Cinsiyet	Doğum Tarihi	Telefon No
▶	10749942744	BELÇİM YÜCEL	Kadın	9.10.2019	0562 812 36 50
	11607189438	RIAYET RÜZGA...	Erkek	30.06.2020	0522 634 00 13
	13663833391	BIHTER DEDE	Erkek	4.04.2021	0547 503 83 01
	14548				
	15111				
	16681				
	17306				
	17454				
	18165				
	18295				
	19557				
	20762				
	22134				
	22256				
	22474				
	22719292446	TAHLİYE SALKA...	Kadın	19.08.2020	0528 980 70 12
	23605619503	NEVAL YURDAK...	Erkek	1.05.2020	0500 545 88 24

#### 6.3.2. Açıklama

Hasta ekleme ve güncelleme işlemlerinde hasta formu açılmaktadır. Hasta eklemede boş form açılır. Hasta güncelleme de seçilen satırdaki hasta bilgileri hasta formunda görülür. Silmek için hasta listesinden sütun seçilip sil butonuna basılır.

#### 6.3.3. Kaynak Kodları

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ECZANE
{
    public partial class HastaFormu : Form

```

```

{
    int islem;
    long TC_No;
    bool Sonuc = false;
    public HastaFormu()
    {
        InitializeComponent();
        ComboBoxListesiAyarla(cbxCinsiyet, "CINSIYET");
        ComboBoxListesiAyarla(cbxSGK, "SGK");
    }
    void ComboBoxListesiAyarla(ComboBox combo, string listeAdi)
    {
        combo.DisplayMember = "Baslik";
        combo.ValueMember = "Deger";
        ArrayList parametreler = new ArrayList();
        parametreler.Add(new DictionaryEntry("ListeAdi", listeAdi));
        combo.DataSource =
VeriTabani_SQLServer.ProsedurCalistir_TabloGelir("ComboBoxListesi", parametreler);
    }
    public bool yeniKayiEkle()
    {
        islem = 0;
        this.ShowDialog();
        return Sonuc;
    }
    public bool kayitGuncelle(long HastaTCNo)
    {
        islem = 1;
        TC_No = HastaTCNo;
        DataTable f_kayit = VeriTabani_SQLServer.SELECTCalistir("SELECT*FROM Hasta
WHERE TC_No=" + HastaTCNo);
        if (f_kayit.Rows.Count < 1)
        {
            MessageBox.Show("Hasta Kaydı Bulunamadı!");
            return false;
        }
        nudTCNo.Value = Convert.ToInt64(f_kayit.Rows[0]["TC_No"]);
        txtAdSoyad.Text = f_kayit.Rows[0]["Adi_Soyadi"].ToString();
        cbxCinsiyet.SelectedValue = f_kayit.Rows[0]["Cinsiyet"].ToString();
        dtpDogumTarihi.Value =
Convert.ToDateTime(f_kayit.Rows[0]["Dogum_Tarihi"]);
        cbxSGK.SelectedValue = f_kayit.Rows[0]["SGK"].ToString();
        txtTelefonNo.Text = f_kayit.Rows[0]["Telefon_No"].ToString();
        this.ShowDialog();
        return Sonuc;
    }
    public bool kayitSil(long HastaTCNo)
    {
        if (MessageBox.Show("Seçilen hastanın kaydı silinecek. Emin misiniz?",
"Onayla", MessageBoxButtons.YesNo) != DialogResult.Yes)
            return false;
        ArrayList parametreler = new ArrayList();
        parametreler.Add(new DictionaryEntry("TC_No", HastaTCNo));
        return (VeriTabani_SQLServer.ProsedurCalistir_MesajDegerlendir("HastaSil",
parametreler));
    }
    private void HastaKaydet_Click(object sender, EventArgs e)
    {
        ArrayList parametreler = new ArrayList();
        parametreler.Add(new DictionaryEntry("TC_No", nudTCNo.Value));
        parametreler.Add(new DictionaryEntry("Adi_Soyadi", txtAdSoyad.Text));
        parametreler.Add(new DictionaryEntry("Cinsiyet", cbxCinsiyet.Text));
    }
}

```

```

        parametreler.Add(new DictionaryEntry("Dogum_Tarihi",
dtpDogumTarihi.Value));
        parametreler.Add(new DictionaryEntry("SGK",cbxSGK.Text));
        parametreler.Add(new DictionaryEntry("Telefon_No", txtTelefonNo.Text));
        string f_prosedurAdi = "dbo.HastaEkle";
        if (islem == 1) f_prosedurAdi = "dbo.HastaGuncelle";
        if(VeriTabani_SQLServer.ProsedurCalistir_MesajDegerlendir(f_prosedurAdi,
parametreler))
        {
            Sonuc = true;
            Close();
        }
    }
    private void KayitIptal_Click(object sender, EventArgs e)
    {
        Close();
    }
}
}

```

## 6.4. İlaç Listesi

### 6.4.1.Çalışan Yazılım Ekran Görüntüsü

İlaç Listesi							
	Barkod	İlacın Adı	Etki Grubu	İlacın Formu	Veriliş Şekli	Firma	Raf Kodu
▶	4264152354181	5 FU-INJ 250 MG...	Antidepresan	Şurup	Reçetesiz	Biomeks	C-02
	9962864291128	ABELCET 100 M...	Antibiyotik	Şampuan	Turuncu Reçeteli	Farmed	C-01
	1507218476836	ABILIFY 30 MG	Ağn Kesici	Losyon	Turuncu Reçeteli	Eczacıbaşı	F-03
	3084546379495	ACCUZIDE 20 M...	Diğer	Emülsiyon	Kontrolle Tabi	Biomeks	C-03
	1775784386240	ACECAP 30	Antibiyotik	Emülsiyon	Mor Reçeteli	Dinçsa	F-02
	2910731612628	ADVANTAN 15 ...	Bulantı	Ampul	Yeşil Reçeteli	Farmed	E-01
	5366660915289	AKNEFUG BP 5 ...	Antidepresan	Suspansiyon	Yeşil Reçeteli	Liba	E-01
	8000387449673	AKSİL-2.5 20 GR	Bulantı	Şampuan	Turuncu Reçeteli	Abott	A-02
	8513335204928	AKSİL-5 20 GR	Diğer	Sprey	Kırmızı Reçeteli	Pfizer	D-03
	4338100694024	ALFUTU 10 MG	Diğer	Pomad	Turuncu Reçeteli	Sanovel	E-01
	7298421674858	ALLO-URIK 300 ...	Antidepresan	Şurup	Kontrolle Tabi	Bayer	B-01
	3193606826871	ALOFIN 1 MG 28	Antidepresan	Tablet	Kırmızı Reçeteli	Fama	C-02
	4243017327535	ALOMIDE 0,1 M...	Diğer	Damla	Mor Reçeteli	Sanovel	D-01
	9541955606368	ALOPEXY T %5 ...	Tansiyon Düşürücü	Krem	Kırmızı Reçeteli	Biomeks	E-02
	6230513169550	ALOXI 250 MCG	Analjezik	Granul	Reçetesiz	Gunsa	D-03
	9225964483251	ALPHA D3 1 MG...	Antitussif	Suspansiyon	Turuncu Reçeteli	Generica	D-03
<div> <div>İlaç Ekle</div> <div>Değiştir</div> <div>Sil</div> </div>							

### 6.4.2. Açıklama

İlaç listesinde ilacın barkodu, adı, etki grubu, formu, veriliş şekli, üretici firma, raf kodu, stok miktarı ve birim fiyat bilgileri bulunur.İlaç ekleme, güncelleme ve silme işlemlerini yapmak için 3 buton eklenmiştir.

### 6.4.3. Kaynak Kodları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ECZANE
{
    public partial class IlacListe : Form
    {
        public IlacListe()
        {
            InitializeComponent();
            ListeGetir();
        }
        void ListeGetir()
        {
            IlacListeGrid.DataSource =
VeriTabani_SQLServer.ProsedurCalistir_TabloGelir("IlacListe",
            null);
        }
        private void yeniIlacEkle_Click(object sender, EventArgs e)
        {
            IlacFormu ilcfrm = new IlacFormu();
            if (ilcfrm.yeniKayitEkle())
                ListeGetir();
        }
        private void IlacGuncelle_Click(object sender, EventArgs e)
        {
            if (IlacListeGrid.SelectedRows.Count < 1)
            {
                MessageBox.Show("Önce Güncellenecek Kaydı Seçin.");
                return;
            }
            long f_IlacGuncelle =
Convert.ToInt64(IlacListeGrid.SelectedRows[0].Cells["Barkod"].Value);
            IlacFormu ilcfrm = new IlacFormu();
            if (ilcfrm.kayitGuncelle(f_IlacGuncelle))
                ListeGetir();
        }
        private void IlacSil_Click(object sender, EventArgs e)
        {
            if (IlacListeGrid.SelectedRows.Count < 1)
            {
                MessageBox.Show("Önce Silinecek Kaydı Seçin.");
                return;
            }
            long f_Barkod =
Convert.ToInt64(IlacListeGrid.SelectedRows[0].Cells["Barkod"].Value);
            IlacFormu ilcfrm = new IlacFormu();
            if (ilcfrm.kayitSil(f_Barkod))
                ListeGetir();
        }
    }
}
```

## 6.5. İlaç Formu

### 6.5.1.Çalışan Yazılım Ekran Görüntüsü

The screenshot shows a software application window titled 'İlaç Listesi'. It contains a table with columns: Barkod, İlaçın Adı, Etki Grubu, İlaçın Formu, Veriliş Şekli, Firma, and Raf Kodu. The table lists several medicines, including '5 FU-INJ 250 MG...' and 'ABELCET 100 M...'. An 'İlaçFormu' dialog box is open, displaying the details of the selected medicine: Barkod (4264152354181), Adı (5 FU-INJ 250 Mg 10), Etki Grubu (Antidepresan), Formu (Şurup), Veriliş Şekli (Reçetesiz), Firma (Biomeks), Raf Kodu (C-02), Stok Adedi (81), and Birim Fiyat (60). The dialog box has 'Kaydet' (Save) and 'İptal' (Cancel) buttons. At the bottom of the application window, there are buttons for 'İlaç Ekle' (Add Medicine), 'Değiştir' (Change), and 'Sil' (Delete).

Barkod	İlaçın Adı	Etki Grubu	İlaçın Formu	Veriliş Şekli	Firma	Raf Kodu
4264152354181	5 FU-INJ 250 MG...	Antidepresan	Şurup	Reçetesiz	Biomeks	C-02
9962864291128	ABELCET 100 M...	Antibiyotik	Şampuan	Turuncu Reçeteli	Farmed	C-01
1507218476						
3084546379						
1775784386						
2910731612						
5366660915						
8000387449						
8513335204						
4338100694						
7298421674						
3193606826						
4243017327						
9541955606						
6230513169						
9225964483						

### 6.5.2. Açıklama

İlaç ekleme ve güncelleme işlemlerinde ilaç formu açılmaktadır. İlaç eklemede boş form açılır. İlaç güncelleme de seçilen satırdaki ilaç bilgileri ilaç formunda görülür. Silmek için ilaç listesinden sütun seçilip sil butonuna basılır.

### 6.5.3. Kaynak Kodları

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace ECZANE
{
    public partial class IlacFormu : Form
    {
        int islem;
        long BarkodNo;
        bool Sonuc = false;
        public IlacFormu()
        {
            InitializeComponent();
        }
    }
}
```

```

        ComboBoxListesiAyarla(cmbIlacFormu, "ILACFORMU");
        ComboBoxListesiAyarla(cmbVerilisSekli, "VERILISSEKLI");
        ComboBoxListesiAyarla(cmbRafKodu, "RAFKODU");
    }
    void ComboBoxListesiAyarla(ComboBox combo, string listeAdi)
    {
        combo.DisplayMember = "Baslik";
        combo.ValueMember = "Deger";
        ArrayList parametreler = new ArrayList();
        parametreler.Add(new DictionaryEntry("ListeAdi", listeAdi));
        combo.DataSource =
VeriTabani_SQLServer.ProsedurCalistir_TabloGelir("ComboBoxListesi", parametreler);
    }
    public bool yeniKayitEkle()
    {
        islem = 0;
        this.ShowDialog();
        return Sonuc;
    }
    public bool kayitGuncelle(long IlacBarkodNo)
    {
        islem = 1;
        BarkodNo = IlacBarkodNo;
        DataTable f_kayit = VeriTabani_SQLServer.SELECTCalistir("SELECT*FROM Ilac
WHERE Barkod=" + IlacBarkodNo);
        if (f_kayit.Rows.Count < 1)
        {
            MessageBox.Show("İlaç Kaydı Bulunamadı!");
            return false;
        }
        nudBarkod.Value = Convert.ToInt64(f_kayit.Rows[0]["Barkod"]);
        txtIlacAdi.Text = f_kayit.Rows[0]["Ilac_Adi"].ToString();
        txtEtkiGrubu.Text = f_kayit.Rows[0]["Etki_Grubu"].ToString();
        cmbIlacFormu.SelectedValue = f_kayit.Rows[0]["Ilac_Formu"].ToString();
        cmbVerilisSekli.SelectedValue =
f_kayit.Rows[0]["Verilis_Sekli"].ToString();
        txtFirmaAdi.Text = f_kayit.Rows[0]["Firma"].ToString();
        cmbRafKodu.SelectedValue = f_kayit.Rows[0]["Raf_Kodu"].ToString();
        nudStokAdedi.Value=Convert.ToInt32(f_kayit.Rows[0]["Stok_Adedi"]);
        nudBirimFiyat.Value= Convert.ToDecimal(f_kayit.Rows[0]["Birim_Fiyat"]);
        this.ShowDialog();
        return Sonuc;
    }
    public bool kayitSil(long IlacBarkodNo)
    {
        if (MessageBox.Show("Seçilen ilaç kaydı silinecek. Emin misiniz?",
"Onayla", MessageBoxButtons.YesNo) != DialogResult.Yes)
            return false;
        ArrayList parametreler = new ArrayList();
        parametreler.Add(new DictionaryEntry("Barkod", IlacBarkodNo));
        return (VeriTabani_SQLServer.ProsedurCalistir_MesajDegerlendir("IlacSil",
parametreler));
    }
    private void IlacKaydet_Click(object sender, EventArgs e)
    {
        ArrayList parametreler = new ArrayList();
        parametreler.Add(new DictionaryEntry("Barkod", nudBarkod.Value));
        parametreler.Add(new DictionaryEntry("Ilac_Adi", txtIlacAdi.Text));
        parametreler.Add(new DictionaryEntry("Etki_Grubu", txtEtkiGrubu.Text));
        parametreler.Add(new DictionaryEntry("Ilac_Formu",
cmbIlacFormu.SelectedValue));
    }

```

```

        parametreler.Add(new DictionaryEntry("Verilis_Sekli",
cmbVerilisSekli.SelectedValue));
        parametreler.Add(new DictionaryEntry("Firma", txtFirmaAdi.Text));
        parametreler.Add(new DictionaryEntry("Raf_Kodu",
cmbRafKodu.SelectedValue));
        parametreler.Add(new DictionaryEntry("Stok_Adedi", nudStokAdedi.Value));
        parametreler.Add(new DictionaryEntry("Birim_Fiyat", nudBirimFiyat.Value));
        string f_prosedurAdi = "dbo.IlacEkle";
        if (islem == 1) f_prosedurAdi = "dbo.IlacGuncelle";
        if (VeriTabani_SQLServer.ProsedurCalistir_MesajDegerlendir(f_prosedurAdi,
parametreler))
        {
            Sonuc = true;
            Close();
        }
    }
    private void Iptal_Click(object sender, EventArgs e)
    {
        Close();
    }
}
}

```

## 6.6. Reçete Listesi

### 6.6.1.Çalışan Yazılım Ekran Görüntüsü

Protokol No	Reçete Tarihi	Reçete Tipi	TC Kimlik Numarası	Ad Soyad
718712485231	26.04.2020	Turuncu	88113553821	ZIYNET ŞAVRAN
439683647166	23.04.2020	Yeşil	24875153499	ÖNAY ALĞAN
149064216932	18.04.2020	Mor	79195570581	KÜHSAR BAL
340561514070	18.04.2020	Turuncu	79750567681	DURDU DURAK
400870712864	18.04.2020	Turuncu	79195570581	KÜHSAR BAL
212041380608	16.04.2020	Turuncu	91357363066	DAREKUTNİ TEPEHAN
941387498116	14.04.2020	Normal	78594133145	TANDORUK GEYİK
142887331659	12.04.2020	Mor	49248533851	UZTEKİN DEMİRDOĞAN
103455138237	11.04.2020	Mor	52267976120	TÜMKAN BİŞİ
540573666346	10.04.2020	Normal	58330359981	YİĞİTKAN TUĞRA
403731568478	7.04.2020	Normal	73972031571	SABİKUN MERAL
256427452836	3.04.2020	Yeşil	14548679575	TANALP İNAN
471976298968	3.04.2020	Kırmızı	64406899255	FAYSAL TAYFUN
524236201480	2.04.2020	Kırmızı	91357363066	DAREKUTNİ TEPEHAN
648134793220	2.04.2020	Kırmızı	17454218410	OKTAY İŞİKTEL
344885010949	29.03.2020	Turuncu	90074168981	ABADI KABAN

### 6.6.2. Açıklama

Reçete listesinde reçetenin protokol numarası, reçete tarihi, reçete tipi, reçete sahibinin TC kimlik numarası, adı ve soyadı bulunmaktadır. Reçete ekleme, güncelleme ve silme işlemlerini yapmak için 3 buton eklenmiştir.



### 6.6.3. Kaynak Kodları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ECZANE
{
    public partial class ReceteListe : Form
    {
        public ReceteListe()
        {
            InitializeComponent();
            ListeGetir();
        }
        void ListeGetir()
        {
            ReceteListeGrid.DataSource =
VeriTabani_SQLServer.ProsedurCalistir_TabloGelir("ReceteListe", null);
        }
        private void ReceteEkle_Click(object sender, EventArgs e)
        {
            ReceteFormu rctfrm = new ReceteFormu();
            if (rctfrm.yeniKayitEkle())
                ListeGetir();
        }
        private void ReceteGuncelle_Click(object sender, EventArgs e)
        {
            if (ReceteListeGrid.SelectedRows.Count < 1)
            {
                MessageBox.Show("Önce Güncellenecek Kaydı Seçin.");
                return;
            }
            long f_ReceteGuncelle =
Convert.ToInt64(ReceteListeGrid.SelectedRows[0].Cells["Protokol No"].Value);
            ReceteFormu rctfrm = new ReceteFormu();
            if (rctfrm.kayitGuncelle(f_ReceteGuncelle))
                ListeGetir();
        }
        private void ReceteSil_Click(object sender, EventArgs e)
        {
            if (ReceteListeGrid.SelectedRows.Count < 1)
            {
                MessageBox.Show("Önce Silinecek Kaydı Seçin.");
                return;
            }
            long f_Protokolno =
Convert.ToInt64(ReceteListeGrid.SelectedRows[0].Cells["Protokol No"].Value);
            ReceteFormu rctfrm = new ReceteFormu();
            if (rctfrm.kayitSil(f_Protokolno))
                ListeGetir();
        }
    }
}
```

## 6.7. Reçete Formu

### 6.7.1.Çalışan Yazılım Ekran Görüntüsü

### 6.7.2. Açıklama

Reçete ekleme ve güncelleme işlemlerinde reçete formu açılmaktadır. Reçete eklemede boş form açılır. Reçete güncellemede seçilen satırdaki ilaçların bilgileri ve reçete bilgileri reçete formunda görülür. Silmek için reçete listesinden sütun seçilip sil butonuna basılır.

### 6.7.3. Kaynak Kodları

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlTypes;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ECZANE
{
    public partial class ReceteFormu : Form
    {
        int islem;
        long ProtokolNo;
        bool Sonuc = false;
        public ReceteFormu()
    }
}
```

```

{
    InitializeComponent();
    ComboBoxListesiAyarla(cbxReceteTipi, "RECETETIPI");
    ComboBoxListesiAyarla(Barkod, "ILACLISTESI");
}
void ComboBoxListesiAyarla(object combo, string listeAdi)
{
    ArrayList parametreler = new ArrayList();
    parametreler.Add(new DictionaryEntry("ListeAdi", listeAdi));
    if (combo is DataGridViewComboBoxColumn)
    {
        (combo as DataGridViewComboBoxColumn).DisplayMember = "Baslik";
        (combo as DataGridViewComboBoxColumn).ValueMember = "Deger";
        (combo as DataGridViewComboBoxColumn).DataSource =
            VeriTabani_SQLServer.ProsedurCalistir_TabloGelir("ComboBoxListesi",
parametreler);
    }
    else
    {
        (combo as ComboBox).DisplayMember = "Baslik";
        (combo as ComboBox).ValueMember = "Deger";
        (combo as ComboBox).DataSource =
            VeriTabani_SQLServer.ProsedurCalistir_TabloGelir("ComboBoxListesi",
parametreler);
    }
}
public bool yeniKayitEkle()
{
    islem = 0;
    ReceteIlacGrid.DataSource = VeriTabani_SQLServer.SELECTCalistir("SELECT
*FROM Recete_Ilac WHERE Barkod = 000000000000");
    this.ShowDialog();
    return Sonuc;
}
public bool kayitGuncelle(long ri_ProtokolNo)
{
    islem = 1;
    ProtokolNo = ri_ProtokolNo;
    DataTable f_kayit = VeriTabani_SQLServer.SELECTCalistir("SELECT * FROM
Recete WHERE Protokol_No=" + ProtokolNo);
    if (f_kayit.Rows.Count < 1)
    {
        MessageBox.Show("İlaç kaydı bulunamadı!..");
        return false;
    }
    dtpReceteTarihi.Value =
Convert.ToDateTime(f_kayit.Rows[0]["Recete_Tarihi"]);
    cbxReceteTipi.SelectedValue = f_kayit.Rows[0]["Recete_Tipi"].ToString();
    nudHastaTcNo.Value = Convert.ToInt64(f_kayit.Rows[0]["TC_No"]);
    ReceteIlacGrid.DataSource = VeriTabani_SQLServer.SELECTCalistir("SELECT *
FROM Recete_Ilac WHERE Protokol_No = " + ProtokolNo);
    this.ShowDialog();
    return Sonuc;
}
public bool kayitSil(long ri_ProtokolNo)
{
    if (MessageBox.Show("Seçilen reçete kaydı silinecek. Emin misiniz?",
"Onayla",
MessageBoxButtons.YesNo) != DialogResult.Yes)
        return false;
    ArrayList parametreler = new ArrayList();
    parametreler.Add(new DictionaryEntry("Protokol_No", ri_ProtokolNo));
}

```

```

        return
    (VeriTabani_SQLServer.ProsedurCalistir_MesajDegerlendir("ReceteSil", parametreler));
    }
    private void ReceteKaydet_Click_1(object sender, EventArgs e)
    {
        ArrayList parametreler = new ArrayList();
        parametreler.Add(new DictionaryEntry("@Protokol_No", nudPNO.Value));
        parametreler.Add(new DictionaryEntry("Recete_Tarihi",
dtpReceteTarihi.Value));
        parametreler.Add(new DictionaryEntry("Recete_Tipi",
cbxReceteTipi.SelectedValue));
        parametreler.Add(new DictionaryEntry("TC_No", nudHastaTcNo.Value));
        DataTable f_ilacilar = (ReceteILacGrid.DataSource as DataTable);
        f_ilacilar.AcceptChanges();
        parametreler.Add(new DictionaryEntry("Recete_Ilac", f_ilacilar));
        string f_prosedurAdi = "dbo.ReceteEkle";
        if (islem == 1) f_prosedurAdi = "dbo.ReceteGuncelle";
        if (VeriTabani_SQLServer.ProsedurCalistir_MesajDegerlendir(f_prosedurAdi,
parametreler))
        {
            Sonuc = true;
            Close();
        }
    }
    private void Iptal_Click(object sender, EventArgs e)
    {
        Close();
    }
}
}

```

## 6.8. İlaç Satış Listesi

### 6.8.1.Çalışan Yazılım Ekran Görüntüsü

İlaç Satış Listesi						
	TC Kimlik No	Ad Soyad	Satış Tarihi	Barkod	İlaç	Toplam Tutar
▶	85484060986	Asya Siper	29.04.2020	1643111075990	EVOSTEN 175 mg/5 ml 100 ml	3,3800
	86091203467	Fahamet Erkaya	29.04.2020	7804752474934	AVISEF 0.5 G	4,5300
	35877699178	Volkan Apaydın	29.04.2020	4253184835355	BEVITIN-C	2,9000
	63234674534	Mamur Yetkin tulu	27.04.2020	8343177924489	ULSETIDIN 40 mg	2,0440
	17454218410	Oktay İşiktel	27.04.2020	1775784386240	ACECAP 30	1,2000
	64406899255	Faysal Tayfun	27.04.2020	5402002724872	JINOMED 600 mg (2 )	5,4400
	43505416856	Direm Uğurlu çimen	27.04.2020	5594715844513	UPTRAVI 400 mcg (60)	4,8600
	43505416856	Direm Uğurlu çimen	27.04.2020	8499176775010	R-LATE %15+%5 Deri (6.5 ml)	44,9000
	45935971534	Cihan Banu Barlak	27.04.2020	4253184835355	BEVITIN-C	2,9000
	34798081727	Oyanalp Karakuzu	18.04.2020	5087719282379	NAC-C 900-300 mg	12,5000
	35877699178	Volkan Apaydın	18.04.2020	2754240753778	AVISEF 1 G	3,0240
	82149308638	Sina Sakaoğlu	18.04.2020	7804752474934	AVISEF 0.5 G	45,3000
	11607189438	Riayet Rüzgar özen	16.04.2020	1523048989790	BIOXININ FORTE %2 60 ml	3,5200
	10749942744	Belçim Yücel	16.04.2020	1409211194824	EFOROL 100 mg 30	18,1280
	99382072672	Ekiner Solmazgül ergör	16.04.2020	4569268624234	SELKAP 200 mg	5,5320
	79184450445	Belen Emiş	14.04.2020	5594715844513	UPTRAVI 400 mcg (60)	9,7200
<	16005717757	Asya Siper	29.04.2020	1643111075990	EVOSTEN 175 mg/5 ml 100 ml	3,3800

Satış Ekle
Güncelle
Sil

### 6.8.2. Açıklama

İlaç satış listesinde hastanın TC kimlik numarası, adı ve soyadı, ilacın satış tarihi, barkodu, adı ve ödeme koşuluna göre tutarı listelenmiştir. İlaç satış ekleme, güncelleme ve silme işlemlerini yapmak için 3 buton eklenmiştir.

### 6.8.3. Kaynak Kodları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ECZANE
{
    public partial class IlacSatisListesi : Form
    {
        public IlacSatisListesi()
        {
            InitializeComponent();
            ListeGetir();
        }
        void ListeGetir()
        {
            IlacSatisGrid.DataSource =
            VeriTabani_SQLServer.ProsedurCalistir_TabloGelir("SatilanIlacListe",null);
        }

        private void IlacSatisEkle_Click(object sender, EventArgs e)
        {
            IlacSatisFormu hlc = new IlacSatisFormu();
            if (hlc.yeniKayitEkle())
                ListeGetir();
        }

        private void IlacSatisGuncelle_Click(object sender, EventArgs e)
        {
            if (IlacSatisGrid.SelectedRows.Count < 1)
            {
                MessageBox.Show("Önce Güncellenecek Kaydı Seçin.");
                return;
            }
            long f_SatisGuncelle =
            Convert.ToInt64(IlacSatisGrid.SelectedRows[0].Cells["TC Kimlik No"].Value);
            IlacSatisFormu ilcsts = new IlacSatisFormu();
            if (ilcsts.kayitGuncelle(f_SatisGuncelle))
                ListeGetir();
        }

        private void IlacSatisSil_Click(object sender, EventArgs e)
        {
            if (IlacSatisGrid.SelectedRows.Count < 1)
            {
                MessageBox.Show("Önce Silinecek Kaydı Seçin.");
                return;
            }
        }
    }
}
```

```

        long f_TCno = Convert.ToInt64(IlacSatisGrid.SelectedRows[0].Cells["TC
Kimlik No"].Value);
        IlacSatisFormu rctfrm = new IlacSatisFormu();
        if (rctfrm.kayitSil(f_TCno))
            ListeGetir();
    }
}
}

```

## 6.9. İlaç Satış Formu

### 6.9.1.Çalışan Yazılım Ekran Görüntüsü

### 6.9.2. Açıklama

İlaç satış ekleme ve güncelleme işlemlerinde ilaç satış formu açılmaktadır. İlaç satış eklemede boş form açılır. İlaç satış güncellemede seçilen satırdaki ilaçların bilgileri ilaç satış formunda görülür. Formda hastanın satın aldığı ilaçlar aynı anda sisteme girilebilir. Fakat her ilaç, ilaç listesinde ayrı satırlara kaydedilmektedir. Aynı zamanda bir hastanın farklı zamanlarda aldığı ilaçları topluca bu listede görüntüleyebildiğimiz için ilaç kullanımları takip edilmesi gereken hastalar kontrol edilebilir. Hastanın bir satış bilgisi silmek istendiğinde ilaç satış listesinden satırın seçilip silinmesi yeterlidir.

### 6.9.3. Kaynak Kodları

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;

```

```

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ECZANE
{
    public partial class IlacSatisFormu : Form
    {
        int islem;
        long hastaTC;
        bool Sonuc = false;
        public IlacSatisFormu()
        {
            InitializeComponent();
            ComboBoxListesiAyarla(Barkod, "ILACLISTESI");
        }
        void ComboBoxListesiAyarla(object combo, string listeAdi)
        {
            ArrayList parametreler = new ArrayList();
            parametreler.Add(new DictionaryEntry("ListeAdi", listeAdi));
            if (combo is DataGridViewComboBoxColumn)
            {
                (combo as DataGridViewComboBoxColumn).DisplayMember = "Baslik";
                (combo as DataGridViewComboBoxColumn).ValueMember = "Deger";
                (combo as DataGridViewComboBoxColumn).DataSource =
                    VeriTani_SQLServer.ProsedurCalistir_TabloGelir("ComboBoxListesi",
parametreler);
            }
            else
            {
                (combo as ComboBox).DisplayMember = "Baslik";
                (combo as ComboBox).ValueMember = "Deger";
                (combo as ComboBox).DataSource =
                    VeriTani_SQLServer.ProsedurCalistir_TabloGelir("ComboBoxListesi",
parametreler);
            }
        }
        public bool yeniKayitEkle()
        {
            islem = 0;
            IlacSatisGrid.DataSource = VeriTani_SQLServer.SELECTCalistir("SELECT *
FROM Hasta_Satilan_Ilac WHERE TC_No = 0000000000");
            this.ShowDialog();
            return Sonuc;
        }
        public bool kayitGuncelle(long hsi_TC)
        {
            islem = 1;
            hastaTC = hsi_TC;
            DataTable f_kayit = VeriTani_SQLServer.SELECTCalistir("SELECT * FROM
Hasta_Satilan_Ilac WHERE TC_No=" + hastaTC);
            nudTCNo.Value = Convert.ToInt64(f_kayit.Rows[0]["TC_No"]);
            IlacSatisGrid.DataSource = VeriTani_SQLServer.SELECTCalistir("SELECT *
FROM Hasta_Satilan_Ilac WHERE TC_No = " + hastaTC);
            this.ShowDialog();
            return Sonuc;
        }
        public bool kayitSil(long hsi_TC)
        {

```

```

        if (MessageBox.Show("Seçilen satılan ilaç kaydı silinecek. Emin misiniz?",
"Onayla",
    MessageBoxButtons.YesNo) != DialogResult.Yes)
        return false;
        ArrayList parametreler = new ArrayList();
        parametreler.Add(new DictionaryEntry("TC_No", hsi_TC));
        return
        (VeriTabani_SQLServer.ProsedurCalistir_MesajDegerlendir("SatilanIlacSil",
parametreler));
    }
    private void SatisKaydet_Click(object sender, EventArgs e)
    {
        ArrayList parametreler = new ArrayList();
        parametreler.Add(new DictionaryEntry("@TC_No", nudTCNo.Value));
        DataTable f_ilacsatis = (IlacSatisGrid.DataSource as DataTable);
        f_ilacsatis.AcceptChanges();
        parametreler.Add(new DictionaryEntry("SatilanIlac", f_ilacsatis));
        string f_prosedurAdi = "dbo.SatilanIlacEkle";
        if (islem == 1)
            f_prosedurAdi = "dbo.SatilanIlacGuncelle";
        if (VeriTabani_SQLServer.ProsedurCalistir_MesajDegerlendir(f_prosedurAdi,
parametreler))
        {
            Sonuc = true;
            Close();
        }
    }
    private void SatisIptal_Click(object sender, EventArgs e)
    {
        Close();
    }
}
}

```



## SONUÇ

Eczane, ilaçların firmalardan hastalara ulaştırılmasını sağlayan ve bu amaçla yapılacak faaliyetlerin gerçekleştirilmesinde önemli rol oynayan bir noktadır. Bu çalışmada bir eczane için gözlemlenen ihtiyaçlara ve iş kurallarına göre bir veri tabanı tasarımı yapılmıştır. Tasarlanan veritabanı C# ile eczane bilgi sistemi otomasyonuna dönüştürülmüştür.

Eczane ilaç satış sistemi tasarımında genel işleyişi ve birimlerin birbirleri ile ilişkileri hakkında bilgi veren bir veri tabanı tasarımı yapılmak istenmiştir. Bu çalışma temel olarak 5 aşamadan oluşmaktadır; sistemin analizi, iş kuralları ve ilişkilerin belirlenmesi, gerekli kavramsal ve mantıksal tasarımların yapılması, SQL üzerinde ilişkisel şemanın oluşturulması ve oluşturulan prosedürler ile bir uygulama ara yüzünün yapılmasıdır. Eczane ilaç satış sistemi tasarımını göstermek amacıyla Chen ER diyagramı ve ilişkisel veritabanı şeması kullanılmıştır.