



Otopark Otomasyonu Yazılım Projesi

Yazılım Mühendisliği Ana Bilim Dalı

Yüksek Lisans

Mehmet Ali OKTAY

ORCID 0000-0000-0000-0000

Tez Danışmanı: Doç. Dr. Vahide BULUT

Haziran 2023

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **Mehmet Ali OKTAY** tarafından hazırlanan **Otopark Otomasyonu Yazılım Projesi** başlıklı bu çalışma tarafımızca okunmuş olup, yapılan savunma sınavı sonucunda kapsam ve nitelik açısından başarılı bulunarak jürimiz tarafından YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

ONAYLAYANLAR:

Tez Danışmanı: **Doç. Dr. Vahide BULUT**
İzmir Kâtip Çelebi Üniversitesi

Tez Eş-danışmanı: **Doç. Dr. Ccccc Ddddd**
İzmir Kâtip Çelebi Üniversitesi

Jüri Üyeleri:

Prof. Dr. Ccccc Ddddd
Ddddd Üniversitesi

Doç. Dr. Ccccc Ddddd
İzmir Kâtip Çelebi Üniversitesi

Dr. Öğr. Üyesi Ccccc Ddddd
Eeeee Üniversitesi

Savunma Tarihi: Haziran 2023

Yazarlık Beyanı

Ben, **Mehmet Ali OKTAY**, başlığı **Otopark Otomasyonu Yazılım Projesi** olan bu tezimin ve tezin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim.

Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: Haziran 2023

Otopark Otomasyonu Yazılım Projesi

ÖZ

OKTAY, Mehmet Ali

Yüksek Lisans Tezi, Yazılım Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Vahide BULUT

Haziran, 2023. 43 Sayfa.

Şehirler için artan araba sahipliği oranları, park yeri mevcudiyetini zorlaştırıyor ve artan karbon emisyonları yoluyla yaşam kalitesini düşürüyor. Bu nedenle, yoğun kasaba ve şehirlerde park yeri aramak, zaman kaybına ve gereksiz yakıt tüketimine yol açan ve daha da önemlisi iklim değişikliğine katkıda bulunan göz korkutucu bir uğraştır.

21. yüzyılda özellikle kentsel alanlarda karşılaşılan bazı sorunların çözümünde teknolojinin en kritik araçlardan biri olarak görüldüğü yeni dönemde, otopark sorunu da geride bırakılmamıştır. Bu bağlamda, Otopark otomasyonu, park etme sürecinin çeşitli yönlerini otomatikleştirmek için teknoloji ve sistemlerin kullanılmasını ifade eder. Bir otopark işletmesindeki araçların kayıtlarını tutmanın yanı sıra, park alanlarının verimliliğini, rahatlığını ve genel yönetimini geliştirmeyi amaçlar. Bu kayıtlar sayesinde işletmenin güvenliğinin sağlanması ve daha ekonomik bir şekilde işletilmesi planlanmıştır.

Otomasyon, park yeri aramak için harcanan zamanı azaltır ve manuel işlemleri en aza indirerek sürücüler için daha sorunsuz ve konforlu bir deneyim sağlar. Araç plakalarının okunmasını sağlamak için gerekli entegre yazılımlar kullanılarak gerçek hayattaki bir yazılım projesi olması hedeflenmiştir. Söz konusu Proje, MySQL veritabanı kullanılarak C# yazılım dili ile Visual Studio üzerinden hazırlanmıştır.

Anahtar Kelimeler: Otopark, Otomasyon, Teknoloji, MySQL, Visual Studio, C#

Parking Automation Software Project

Abstract

OKTAY, Mehmet Ali

Master's Thesis, Software Engineering Anabilim Dalı

Advisor: Doç. Dr. Vahide BULUT

June, 2023. 43 Pages.

For cities, growing rates of car ownership are rendering parking availability a challenge and lowering the quality of life through increased carbon emissions. For this reason, the search for a parking spot in busy towns and cities is a daunting endeavor, leading to time wastage and unwarranted consumption of fuel, and, importantly, contributing to climate change.

In the new era, in which technology is considered one of the most critical tools in solving some problems encountered in the 21st century, especially in urban areas, the parking problem has not been left behind. In this context, Parking automation refers to the use of technology and systems to automate various aspects of the parking process. It aims to improve the efficiency, convenience and general management of parking areas, as well as keeping records of vehicles in a parking lot business. Thanks to these records, it is planned to ensure the safety of the enterprise and to operate it more economically.

Automation reduces the time spent searching for a parking space and minimizes manual operations, providing a smoother and more comfortable experience for drivers. It is aimed to be a real-life software project by using the necessary integrated software to ensure that the license plates are read. The project in question was prepared in C# software language using MySQL database through Visual Studio.

Keywords: Parking, Automation, Technology, MySQL, Visual Studio, C#

Teşekkür

Yüksek Lisans Öğretimi boyunca katkılarından dolayı İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü Yazılım Mühendisliği Anabilim Dalı Bölüm Başkanı Doç. Dr. Aytuğ ONAN' a ve Tez çalışmamdaki desteği ve katkıları için Danışmanım Doç. Dr. Vahide BULUT' a ve tüm öğretmenlerime Teşekkürlerimi iletirim.

İçindekiler

1 Tezin İçeriği	i
Dış Kapak.....	i
Giriş Sayfaları	i
Onay Sayfası	i
Yazarlık Beyanı.....	ii
Öz.....	iii
Abstract	iv
Teşekkür.....	v
İçindekiler	vi
Şekiller Listesi.....	viii
Kısaltmalar Listes	ix
1 Giriş	1
2 Program Veri Tabanı	2
2.1 XAMPP Programı	2
2.1.2 XAMPP Programı'nın Çalıştırılması	2
2.1.3 Sunucuya İstek Yapmak	3
2.1.4 Veri Tabanının Oluşturulması	4
2.1.5 MySQL Veri Tabanının Yüklenmesi.....	4
2.2 Program ile Veri Tabanı Arasındaki Veri Akışı.....	5
2.2.1 Araç Kayıt Ekranına Ait Verilerin Aktarılması.....	6
2.2.1.1 Araç Bilgilerinde Marka ve Seri Ekleme.....	8
2.2.2 Araç Otopark Çıkışı Ekranına Ait Verilerin Aktarılması.....	9
2.2.3 Araç Otopark Satış Listesine Ait Veriler.....	10

3 Otopark Otomasyon Yazılımı.....	12
3.1 Programın Ana Tasarım Ekranı.....	12
3.2 Araç Otopark Kayıt Ekranı.....	15
3.2.1 Araç Marka Ekleme Ekranı.....	21
3.2.2 Araç Modeli Ekleme Ekranı.....	22
3.3 Araç Otopark Yerleri.....	24
3.4 Araç Otopark Çıkış Syafası.....	26
3.5 Araç Otopark Satış Listesi.....	29
4 Sonuçlar Ve Öneriler.....	32
4.1 Sonuçlar.....	32
4.2 Öneriler.....	32
Özgeçmiş.....	33

Şekiller Listesi

Şekil 2.1	XAMPP Kontrol Ekranı	3
Şekil 2.2	XAMPP ile İlgili Ayrıntıları Listeleyen WEB Sayfası	3
Şekil 2.3	phpMyAdmin Sayfasında Veri Tabanının Oluşturulması	4
Şekil 2.4	MySQL Dosyasının Seçimi	4
Şekil 2.5	MySQL Dosyasının İçeri Aktarılması	5
Şekil 2.6	Program Ana Ekranı	5
Şekil 2.7	Veri Tabanı Ana Ekranı	6
Şekil 2.8	Program Araç Kayıt Ekranı	6
Şekil 2.9	Veri Tabanının “car_record” Bölümü	7
Şekil 2.10	Veri Tabanının “car_status” Bölümü	8
Şekil 2.11	Veri Tabanının “car_brand_info” Bölümü	8
Şekil 2.12	Veri Tabanının “car_s” Bölümü	9
Şekil 2.13	Program Araç Otopark Çıkış Sayfası Ekranı	9
Şekil 2.14	Program Araç Otopark Satış Listesi Ekranı	10
Şekil 2.15	Veri Tabanının “car_sales_list” Bölümü	11
Şekil 3.1	Programın Ana Ekranı	12
Şekil 3.2	Form 1 Kısımına Ait Kodlar-1	13
Şekil 3.3	Form 1 Kısımına Ait Kodlar-2	14
Şekil 3.4	Form 1 Kısımına Ait Kodlar-3	15
Şekil 3.5	Araç Otopark Kayıt Ekranı	16
Şekil 3.6	Araç Kayıt Ekranı Kodları-1	16
Şekil 3.7	Araç Kayıt Ekranı Kodları-2	17
Şekil 3.8	Araç Kayıt Ekranı Kodları-3	17
Şekil 3.9	Araç Kayıt Ekranı Kodları-4	18
Şekil 3.10	Araç Kayıt Ekranı Kodları-5	19
Şekil 3.11	Araç Kayıt Ekranı Kodları-6	19
Şekil 3.12	Araç Kayıt Ekranı Kodları-7	20
Şekil 3.13	Araç Kayıt Ekranı Kodları-8	20
Şekil 3.14	Araç Marka Ekleme Ekranı	21
Şekil 3.15	Araç Marka Ekleme Kod Yapısı	22
Şekil 3.16	Araç Model Ekleme Ekranı	23
Şekil 3.17	Araç Model Ekleme Kod Yapısı	23
Şekil 3.18	Araç Otopark Yerleri Uygulama Ekranı	24
Şekil 3.19	Araç Otopark Yerleri Ekran Tasarımı	25
Şekil 3.20	Araç Otopark Yerleri Kod Yapısı	26
Şekil 3.21	Araç Otopark Çıkış Ekranı	27
Şekil 3.22	Araç Otopark Çıkış Ekranı Tasarımı	27
Şekil 3.23	Araç Otopark Çıkış Ekranı Kod Yapısı-1	28
Şekil 3.24	Araç Otopark Çıkış Ekranı Kod Yapısı-2	29
Şekil 3.25	Araç Otopark Satış Listesi Ekranı	30
Şekil 3.26	Araç Otopark Satış Listesi Ekran Tasarımı	30
Şekil 3.27	Araç Otopark Satış Listesi Kod Yapısı	31

Kısaltmalar

Apache	: Açık kaynak kodlu ve özgür bir Web sunucu programıdır.
C#	: Microsoft tarafından .NET Teknolojisi için geliştirilen modern bir programlama dilidir.
Linux	: Linux çekirdeğine dayalı, açık kaynak kodlu, Unix benzeri bir işletim sistemidir.
MacOS	: Mach mikro çekirdeği üzerine kurulu, açık kaynak bir işletim sistemidir.
MySQL	: Veri tabanı yönetim sistemidir.
Perl	: Açık kaynak kodlu bir programlama dilidir.
PHP	: Hypertext Preprocessor, internet için üretilmiş, sunucu tarafı, çok geniş kullanımlı, genel amaçlı, içerisine HTML gömülebilen betik ve programlama dilidir.
phpMyAdmin	: PHP ile yazılmış açık kaynak kodlu bir araçtır. Başlıca kullanım amacı İnternet üzerinden MySQL veri tabanı yönetimidir.
Windows	: Kullanıcıya grafik arabirimler ve görsel iletilerle yaklaşarak, yazılımları çalıştırmak, komut vermek gibi klavyeden yazma zorunluluğunu ortadan kaldıran, Microsoft şirketinin geliştirdiği bir işletim sistemidir.
XAMPP	: Bir web sunucusu yazılımıdır. Yerel sunucu olarak çalışır. Yerel bir sunucudaki verileri yönetmek veya kaydetmek için kullanılır.

BÖLÜM 1

GİRİŞ

Dünya genelindeki kent nüfuslarının hızla artması ve aynı zamanda teknolojik gelişmeler ile bağlantılı olarak araç ihtiyacı taleplerinin hızlı bir artış göstermesi sebebiyle, başta metropol şehirlerde olmak üzere karşılaşılan en acil çözülmesi gereken sorunlardan biri araçlar için park yerlerine ihtiyaç duyulmaktadır. İster konutlar ister iş yerleri ister eğlence yerleri... vb. aktiviteler neticesinde artan talep ile sınırlı alan ve geleneksel park etme yöntemleri verimsiz hale gelmiştir. Neyse ki, teknolojideki gelişmeler, arabalarımızı park etme şeklimizdeki devrim yaratmayı vaat eden park otomasyonu kavramını ortaya çıkarmıştır.

Otopark otomasyonu yazılım projeleri, araç kullanıcılarının daha rahat olmaları amacıyla park etme sürecini otomatikleştirmek için, sensörler, kameralar ve diğer sistemlerle entegre edilebilirler. Bu teknoloji, tam otomatik otoparklardan, sürücülerini uygun noktalara yönlendirebilen akıllı park sayaçlarına, araçların güvenli koşullar içerisinde verimli, konforlu ve insanların bulunmadığı sistemler ile park edilmesine kadar çeşitli şekillerde günümüzde uygulanmaktadır.

Projemizdeki Otopark Otomasyon Yazılımını C# yazılım dili ile Visual Studio üzerinde, MySQL veri tabanı kullanarak hazırlanmıştır. Yazılım ile otoparkın merkezi bir veri tabanı çerçevesinde otomatik olarak işletilmesi, kontrol edilmesi ve raporlanması amaçlanmıştır.

Projenin faydalarını incelersek; sürücülerin park yeri bulmaları için geçen süreyi azaltarak park etme operasyonlarındaki verimliliği artırmaya yardımcı olabilmektedir. Yazılım, mevcut park yerleri, park ücretleri ve ödeme seçenekleri hakkında gerçek zamanlı bilgi sağlamakta ve ayrıca park kullanımının izlenebilirliğini ve doluluk oranlarını, otopark kullanım sürelerini ve otopark geliri hakkında rapor oluşturabilmektedir.

BÖLÜM 2

PROGRAM VERİ TABANI

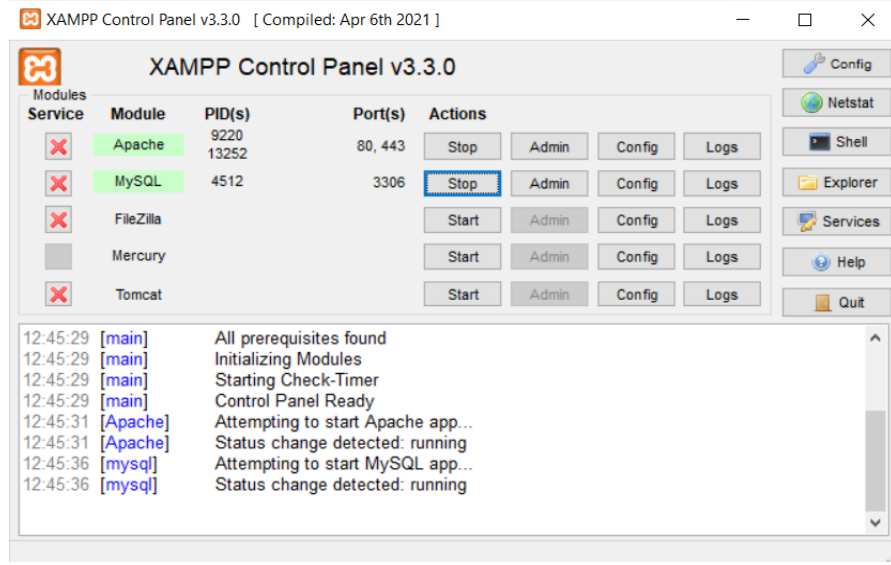
2.1. XAMPP Programı

XAMPP, bir web sunucusu paketi olup, öncelikle PHP, MySQL, Apache ve Perl gibi web geliştirme araçlarını içermektedir. Bu araçların hepsinin bir arada sunulması sebebiyle Program Kullanıcılarının kolayca bir web sunucusu oluşturabilmesi mümkün olmaktadır. XAMPP, Windows, Linux ve MacOS gibi farklı işletim sistemlerinde çalışabilen ücretsiz bir açık kaynaklı yazılımdır.

XAMPP, söz konusu kullanıcılara web uygulamaları geliştirmeleri için gerekli olan tüm araçları sunmaktadır ve MySQL veri tabanını kullanarak veri depolama olanağı sağlamaktadır. Otopark Otomasyon projesinde de XAMPP programı kullanılmıştır.

2.1.2. XAMPP Programının Çalıştırılması

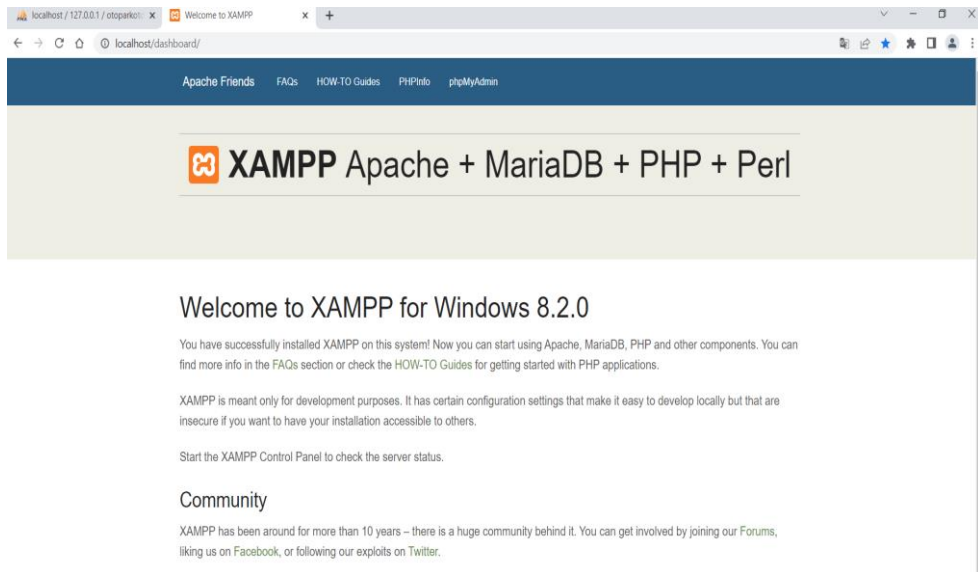
Bilgisayarımızdaki XAMPP Control Panel 'ini (xampp-control.exe) çalıştırdığımızda ekran aşağıdaki Şekil 2.1'deki gibi olacaktır. Sonrasında, ilgili programın başlatılması amacıyla Apache ve MySQL'in yanında yer alan Başlat (Start) düğmesine tıklanmaktadır.



Şekil 2.1. XAMPP Kontrol Ekranı

2.1.3 Sunucuya İstek Yapmak

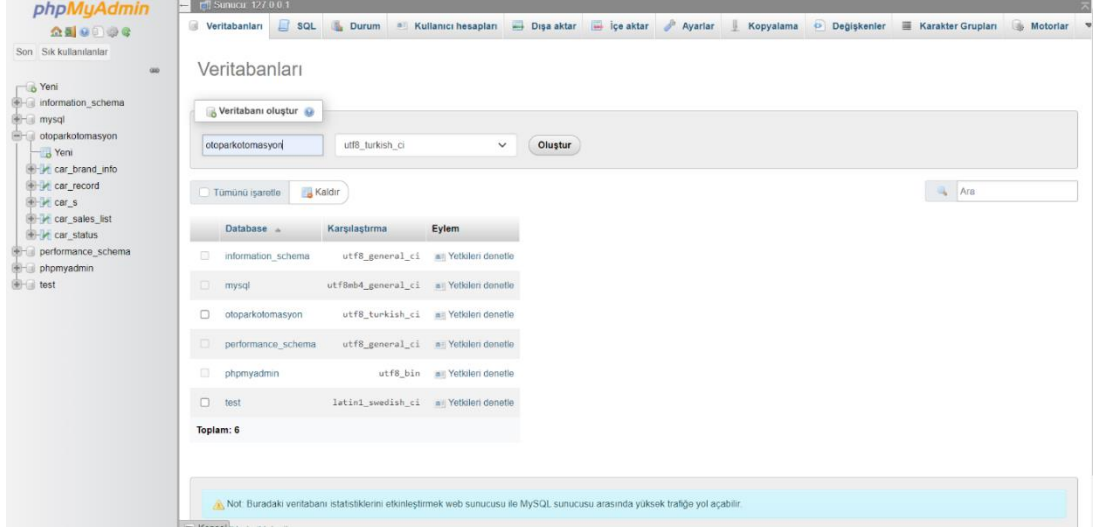
Kontrol panelinde Apache ve MySQL'i başlattıktan sonra web tarayıcımıza <http://localhost> yazmakta ve tarayıcımızda XAMPP ile ilgili ayrıntıları listeleyen bir web sayfası açılmaktadır. Açılan sayfa üzerinde bulunan "phpMyAdmin" sekmesine giriş yapılarak veri tabanımızı aktif hale getirmiş olduğumuz sayfa açılmaktadır.



Şekil 2.2. XAMPP ile İlgili Ayrıntıları Listeleyen Web Sayfası

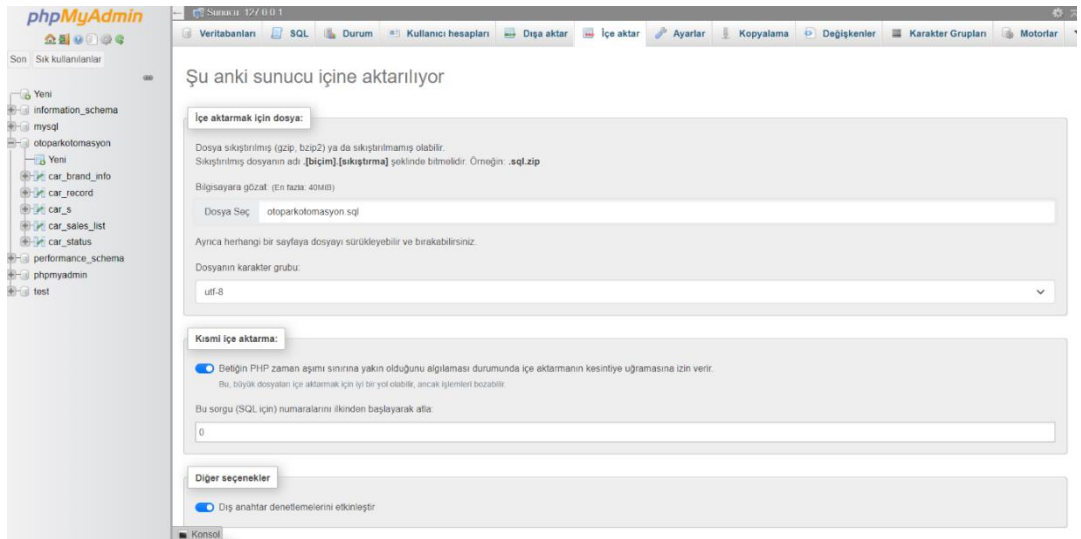
2.1.4 Veri Tabanının Oluşturulması

Açılan sayfada yeni veri tabanımızı oluşturmak için, Veri Tabanımıza ait isim girişinin yapılması istenmektedir. Ayrıca, dil desteği için “utf8_turkish_ci” seçip, Veri Tabanımızı oluşturmak için “oluştur” butonuna basılmaktadır.



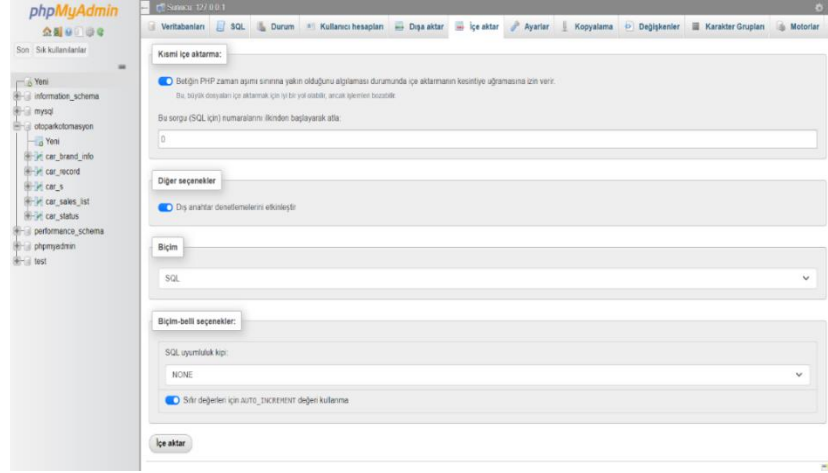
Şekil 2.3. phpMyAdmin Sayfasında Veri Tabanının Oluşturulması

2.1.5. MySQL Veri Tabanının Yüklenmesi



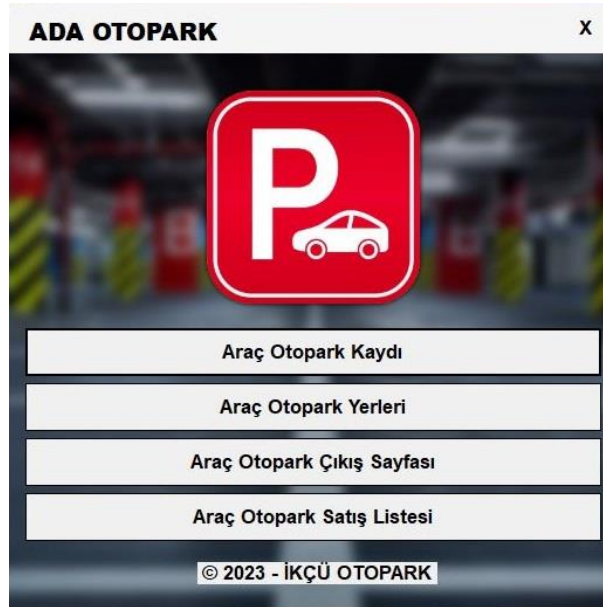
Şekil 2.4. MySQL Dosyasının Seçimi

Verileri içe aktarmak için, phpMyAdmin sayfası içerisinde bulunan “İçe aktar” sekmesine gidilerek “Dosya Seç” kısmından SQL dosyası seçilmektedir. Dosya seçimi yapıldıktan sonra, phpMyAdmin sayfasındaki “İçe aktar” butonuna basılarak veri tabanının aktarımı gerçekleştirilmektedir.



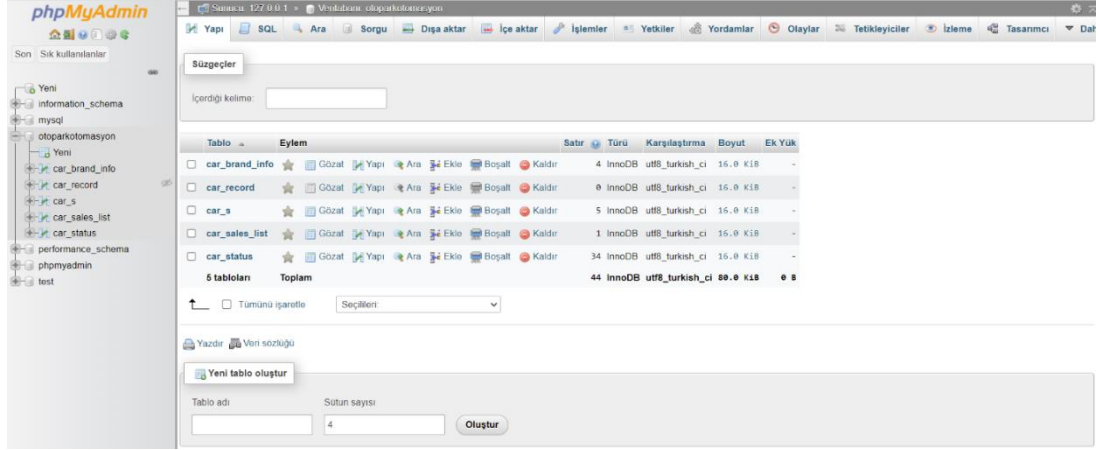
Şekil 2.5. MySQL Dosyasının İçe Aktarımı

2.2. Program ile Veri Tabanı Arasındaki Veri Akışı



Şekil 2.6. Program Ana Ekranı

Programa ait veri tabanının yükleme işlemleri tamamlandıktan sonra söz konusu veriler, ilgili veri tabanında kayıt altına alınmaktadır. Programın tasarım ekranında belirtilen “Araç Otopark Kaydı”, “Araç Otopark Yerleri”, “Araç Otopark Çıkış Sayfası” ve “Araç Otopark Satış Listesi” sekmeleri veri tabanı ile ilişkilendirilmiştir.



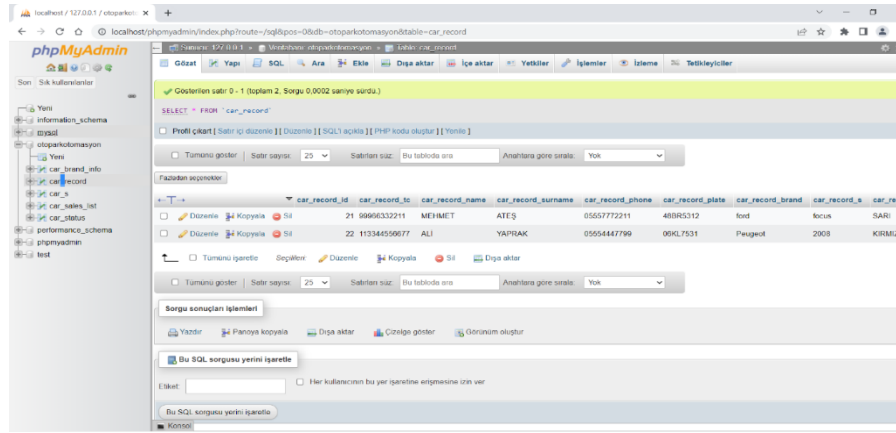
Şekil 2.7. Veri Tabanı Ana ekranı

2.2.1 Araç Kayıt Ekranına Ait Verilerin Aktarılması

The screenshot shows the 'ADA OTO PARK - Car Parking Record' form. The form is divided into two main sections: 'Kişi Bilgileri' (Person Information) and 'Araç Bilgileri' (Car Information). The 'Kişi Bilgileri' section includes fields for TC (113344556677), Ad (ALİ), Soyad (YAPRAK), and Telefon Numarası (0554447799). The 'Araç Bilgileri' section includes fields for Plaka (06KL7531), Marka (Peugeot), Seri (2008), and Renk (KIRMIZI). There is also a 'Park Yeri' dropdown menu with 'P-1' selected. At the bottom right, there are two buttons: 'Bilgileri Temizle' (Clear Information) and 'Bilgileri Kaydet' (Save Information). The footer of the form displays '© 2023 - İKÇÜ OTO PARK'.

Şekil 2.8. Program Araç Kayıt Ekranı

Programa ait Araç Kayıt Ekranı (Car Parking Record)' nda; Kişisel Bilgiler, Araç Bilgileri ve Park Yeri Bilgileri bölümleri vardır. Bunlardan Kişisel Bilgiler penceresinde veri girişi olarak kullanıcıya ait T.C. No, Ad, Soyad ve Telefon Numarası bilgileri girişi yapılmaktadır. Kişi Bilgileri penceresinde ki veri girişleri yapıldıktan sonra, söz konusu programa ait veri tabanımızda bulunan “car_record” kısmındaki tablonun “car_record_tc”, “car_record_name”, “car_record_surname” ve “car_record_phone” bölümleri ile ilgili bilgiler işlenmektedir.



Şekil 2.9. Veri Tabanının “car_record” Bölümü

Araç Kayıt Ekranı üzerinde yer alan Araç Bilgileri penceresinde ise, araca ait Plaka, Marka, Seri ve Renk bilgilerinin veri girişleri yapılmaktadır. Araç Bilgileri penceresinde ki veri tabanımızın “car_record” kısmında yer alan tablonun “car_record_plate”, “car_record_brand”, “car_record_s” ve “car_record_paint” kayıtları ile ilgili bilgiler işlenmektedir.

Araç Kayıt Ekranımızın diğer bölümü olan Park Yeri seçiminde ise otopark alanı içerisindeki boş alanlara ait liste karşımıza çıkmaktadır. Buradan aracı yönlendirmek istediğimiz park yeri alanı seçilerek ilgili araç için, söz konusu park yeri alanına kayıt alınır. Otoparka ait boş ve dolu alanlar veri tabanımızdaki “car_status” kısmındaki tabloya işlenmektedir.

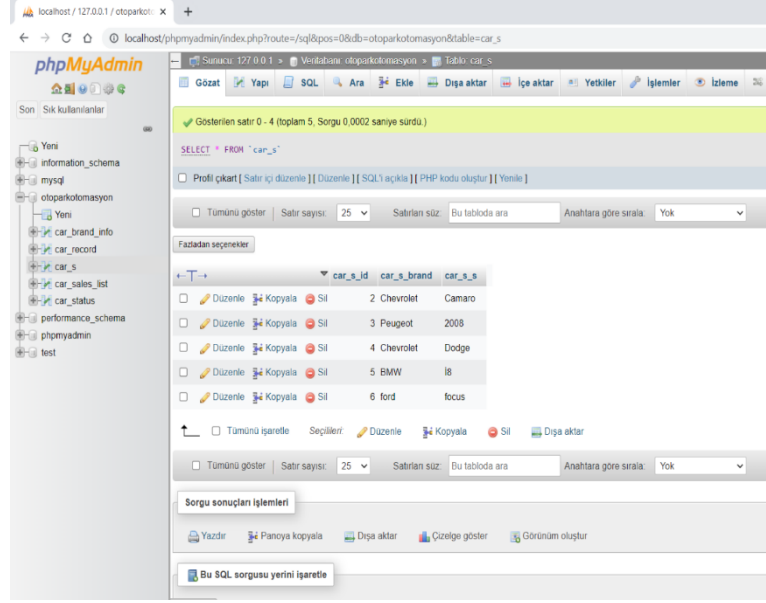
Şekil 2.10. Veri Tabanının “car_status” Bölümü

2.2.1.1 Araç Bilgilerinde Marka ve Seri Ekleme

Yeni araç bilgilerine ait veri girişlerinin yapılması durumunda, ilgili aracın bilgileri, Araç Kayıt Ekranımızda yer alan Marka ve Seri hücrelerine, öncelikle Marka verisinin eklenmesi gerekmektedir. Marka bilgilerine ait veri girişi, veri tabanımızdaki “car_brand_info” kısmına işlenmektedir.

Yeni Marka bilgileri veri girişi yapıldıktan sonra, Seri bilgisi kısmında yer alan ilgili araca ait yeni seri bilgisi eklemesi yapılabilmektedir. Seri hücrelerine eklenecek olan araca ait veri girişi; veri tabanımızdaki “car_s” kısmına işlenmektedir.

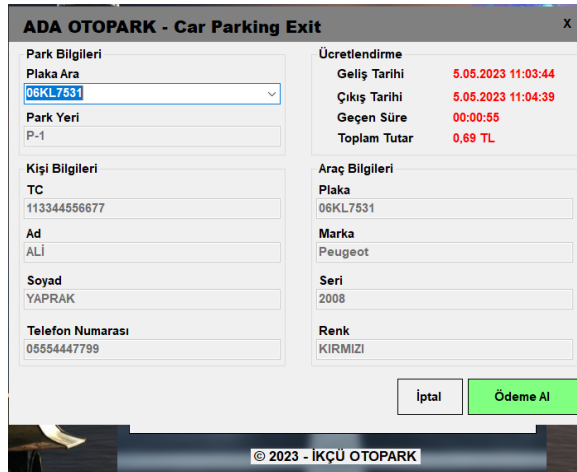
Şekil 2.11. Veri Tabanının “car_brand_info” Bölümü



Şekil 2.12. Veri Tabanının “car_s” Bölümü

2.2.2 Araç Otopark Çıkış Ekranına Ait Verilerin Aktarılması

Programımıza ait Araç Otopark Çıkış Sayfasına giriş yapıldığında, ekranda açılan pencere üzerindeki Plaka Ara hüccresinde, işlem yapılacak plaka seçilmektedir. Plaka seçiminin yapılması ile birlikte, veri tabanımızın “car_record” tablosunda bulunan bilgiler ekranımıza aktarılmaktadır.



Şekil 2.13. Programın Araç Otopark Çıkış Sayfası Ekranı

Veri tabanımızdaki bilgilerin, ekranımıza aktarılması ile birlikte, ilgili aracın otopark kayıt bilgilerindeki Ücretlendirme bölümünün içinde yer alan Geliş Tarihi kısmında belirtilen kayıt bilgisi ile aracın çıkış yapmak istediği zaman bilgileri kullanılarak, belirlenmiş olan mevcut tarife üzerinden aracın ödeyeceği tutar belirlenir. Araç çıkışı işlemi “Ödeme Al” butonuna basılarak tamamlanır.

Araç çıkış işleminin gerçekleşmesi ile birlikte yapılan işleme ait veriler, veri tabanımızın “car_sales_list” kısmında kayıt altına alınmıştır.

2.2.3 Araç Otopark Satış Listesine Ait Veriler

Otopark Otomasyon programı ile ilgili otoparka ait giriş-çıkış işlemlerini gerçekleştirmiş olan araçların bilgilerine, Araç Otopark Satış Listesi bölümünde açılan ekran ile ulaşılabilmektedir. Ekranımıza gelen veriler, veri tabanımızın “car_sales_list” kısmından aktarılmıştır.



ID	FİYAT	TC	İSİM	SOYİSİM	TELEFON	PLAKA
14	0,96	113344556677	ALİ	YAPRAK	05554447799	06KL7531
13	0,69	1236549874	EMRAH	GÖK	05356669933	35AD4612
12	0,10	99999999999999	Burak	Yılmaz	5558889966	66DD99

Şekil 2.14. Programın Araç Otopark Satış Listesi Ekranı

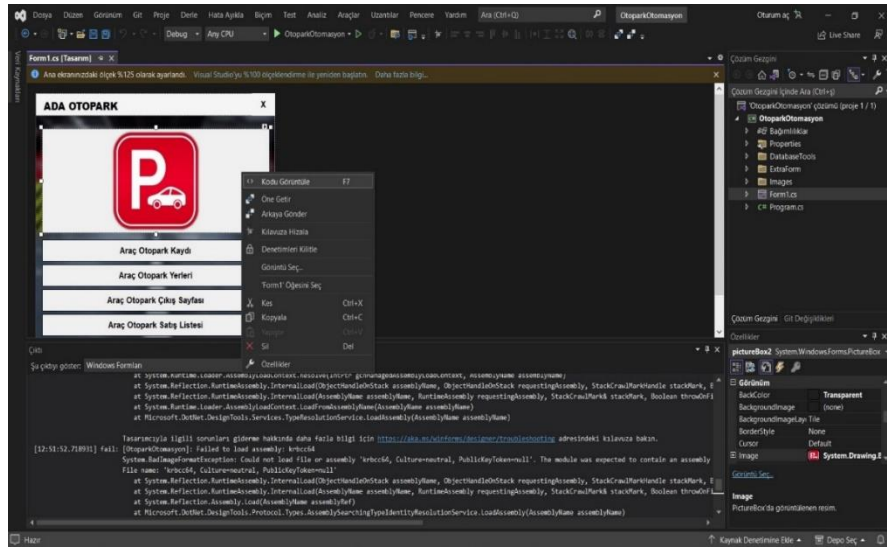
BÖLÜM 3

OTOPARK OTOMASYON YAZILIMI

Hazırlanan yazılım projesini, Microsoft Visual Studio programı ile açıp, ilgili program içerisindeki tasarım ekranları ve bu ekranlara ait proje kodları detaylı bir şekilde incelenebilir.

3.1. PROGRAMIN ANA TASARIM EKSPAN

Microsoft Visual Studio programı ile söz konusu proje açıldığında, “Form1.cs” kısmını açarak programın başlangıç formundaki tasarımı görüntülenmiş olmaktadır. Açılan tasarım pencerelerine ait yazılım kodlarını görüntülemek için, programdaki ilgili tasarım penceresine Mouse ile sağ tık yapıldığında, açılan pencerede ilk sıradaki “Kodu Görüntüle” kısmının seçilmesiyle veya ilgili tasarım penceresi seçili iken F7 tuşuna basılmasının ardından kodlar görüntülenmektedir. (Şekil3.1)



Şekil 3.1. Programın Ana Tasarım Ekspanı

Ayrıca, Microsoft Visual Studio programının sağ alt kısımda bulunan “Özellikler” bölümünde butonlarımıza ait isimlendirmeler ve click yapısı vb. niteliklerin yapılandırılması sağlanmaktadır.

Programdaki Ana Tasarım Ekranına ait kod yapısı aşağıda belirtilmiştir. Bu bölümdeki kodlar incelendiğinde, Form1 sınıfının otopark kayıtlarını yöneten bir uygulamanın grafik kullanıcı arabirimi sağlayan bir Windows Formu olduğu ve söz konusu Formu taşımak ve sürüklemek amacıyla ilgili gerekli yöntemleri içermektedir. Ayrıca, Programdaki diğer form sekmelerini açmayı sağlayan düğmeler için gerekli olay işleyicilerini de kapsamaktadır.

```
... \Desktop\OtoparkOtomasyon\OtoparkOtomasyon\Form1.cs 1
1 using MySql.Data.MySqlClient;
2 using OtoparkOtomasyon.DatabaseTools;
3 using System.Windows.Forms;
4 using OtoparkOtomasyon.ExtraForm;
5
6 namespace OtoparkOtomasyon
7 {
8     public partial class Form1 : Form
9     {
```

Şekil 3.2. Form1 Kısına Ait Kodlar- 1

Şekil 3.2’de yer alan kodlar, “Form1” sınıfına ait düzeni başlatmayı ve ilgili ayarlamaları yapmayı sağlayan içeriklerdir. Ayrıca, yine veri tabanına erişimi ve içe aktarımları da sağlamaktadır. (Şekil 3.2)

```

11 // Move and Drag Setup
12 private bool dragging = false;
13 private Point startPoint = new Point(0, 0);
14
15 public Form1()
16 {
17     InitializeComponent();
18     pictureBox1.Controls.Add(pictureBox2);
19     pictureBox2.BackColor = Color.Transparent;
20 }
21
22 private void closeButton_Click(object sender, EventArgs e)
23 {
24     Application.Exit();
25 }
26
27 private void panel1_MouseDown(object sender, MouseEventArgs e)
28 {
29     MouseDownMethod(e);
30 }
31 private void panel1_MouseMove(object sender, MouseEventArgs e)
32 {
33     MouseMoveMethod(e);
34 }
35 private void panel1_MouseUp(object sender, MouseEventArgs e)
36 {
37     MouseUpMethod();
38 }
39
40 private void label2_MouseDown(object sender, MouseEventArgs e)
41 {
42     MouseDownMethod(e);
43 }
44 private void label2_MouseMove(object sender, MouseEventArgs e)
45 {
46     MouseMoveMethod(e);
47 }
48 private void label2_MouseUp(object sender, MouseEventArgs e)
49 {
50     MouseUpMethod();
51 }
52

```

Şekil 3.3. Form 1 Kısımına Ait Kodlar- 2

Şekil 3.3'te yer alan kodlar ise, Mouse ile pencerenin ekran üzerinde istenilen bir yere taşınmasını sağlamaktadır. Söz konusu kodlarda belirtilen methodlara (MouseDownMethod, MouseMoveMethod vb.) klavyede ki Ctrl tuşu ile birlikte tıklandığında, ilgili methoda ait kod satırına otomatik olarak götürmektedir. Belirtilen kodlar ile diğer tasarım ekranlarında da burada olduğu gibi pencereleri istenilen yerlere taşıma özelliği bulunmaktadır.


```

... \Desktop\OtoparkOtomasyon\OtoparkOtomasyon\Form1.cs 2
54     private void MouseDownMethod(MouseEventArgs e)
55     {
56         dragging = true;
57         startPoint = new Point(e.X, e.Y);
58     }
59
60     private void MouseMoveMethod(MouseEventArgs e)
61     {
62         if (dragging)
63         {
64             Point p = PointToScreen(e.Location);
65             Location = new Point(p.X - this.startPoint.X, p.Y -
                this.startPoint.Y);
66         }
67     }
68
69     private void MouseUpMethod()
70     {
71         dragging = false;
72     }
73

```

Şekil 3.4. Form 1 Kısımına Ait Kodlar - 3

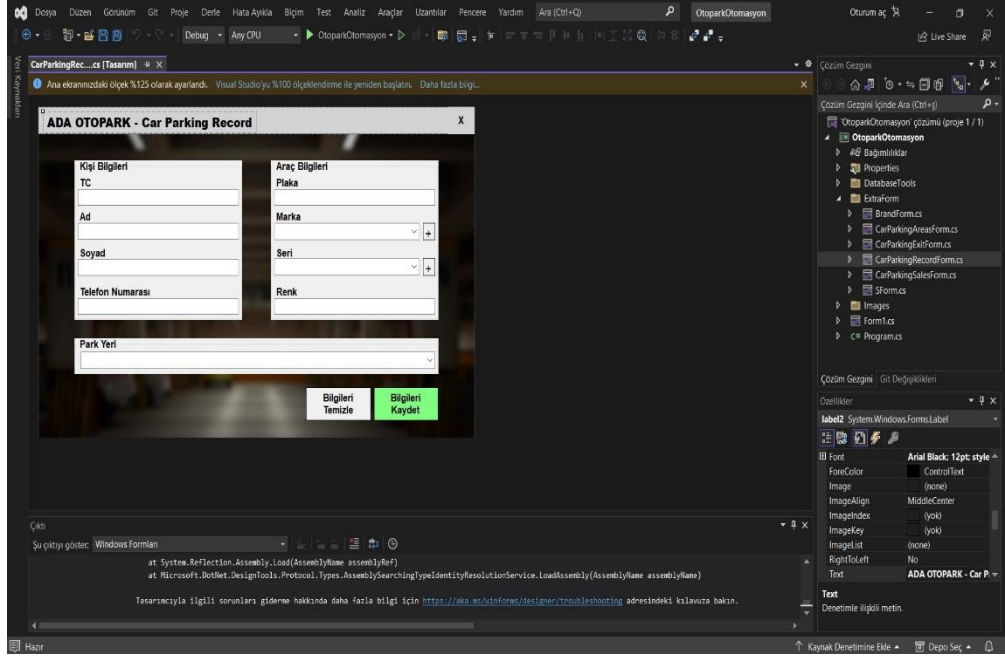
Ana tasarım ekranına ait kodların son kısmını incelediğimizde ise, buradaki kodların amacı programa ait ekranda belirtilen “Araç Otopark Kaydı, Araç Otopark Yerleri, Araç Otopark Çıkış Sayfası ve Araç Otopark Satış Listesi” bölümlerine giriş yapmamızı sağlamaktır.

Microsoft Visual Studio programının sağ tarafında bulunan “ExtraForm” kısmında bu bölüme ait olan Tasarım Ekranları görülmektedir. Söz konusu bölümün içeriğinde CarParkingAreasForm, CarParkingRecordForm, BrandForm ve Sform yapıları bulunmaktadır.

3.2 ARAÇ OTOPARK KAYIT EKRANI

Otopark Otomasyon Programı ana ekranında, “Araç Otopark Kaydı” butonuna tıkladığında yeni bir ekran açılmaktadır. Bu ekran üzerinden otoparka giriş yapan araçlara ve kullanıcılarına ait bilgiler sisteme girilmektedir. Aynı zamanda, boş otopark alanlarına müşterilerin yönlendirilmesi sağlanmaktadır. Programdaki Araç Otopark Kayıt Ekranı; Kişisel Bilgiler, Araç Bilgileri ve Park Yeri Bilgilerinin bulunduğu üç kısma ayrılmıştır.

Bu ekrandaki Araç Bilgileri kısmında yer alan Marka ve Seri hücrelerine yeni eklemeler yapılmak istenildiğinde, söz konusu programda yeni ekranlar açılmaktadır.



Şekil 3.5. Araç Otopark Kayıt Ekranı

Microsoft Visual Studio programı ile projemizi açtığımızda “ExtraForm” klasörünün içinde bulunan “CarParkingRecordForm” yapısı açıldığında ilgili ekrana ait tasarım ve kodlar görüntülenebilmektedir.

```

1 using MySql.Data.MySqlClient;
2 using OtoparkOtomasyon.DatabaseTools;
3 using System;
4 using System.Collections.Generic;
5 using System.ComponentModel;
6 using System.Data;
7 using System.Drawing;
8 using System.Linq;
9 using System.Text;
10 using System.Threading.Tasks;
11 using System.Windows.Forms;
12
13 namespace OtoparkOtomasyon.ExtraForm
14 {
15     public partial class CarParkingRecordForm : Form
16     {
17         private MySqlConnection connection =
18             ConnectionMaker.GetConnection();
19         private MySqlCommand cmd;
20         private MySqlDataReader reader;
21
22         // Move and Drag Setup
23         private bool dragging = false;
24         private Point startPoint = new Point(0, 0);
25
26         public CarParkingRecordForm()
27         {
28             InitializeComponent();
29         }
30     }
31 }

```

Şekil 3.6. Araç Kayıt Ekranı Kodları-1

Şekil 3.6'daki kodlarda "CarParkingRecordForm" yapısına ait düzen başlatılıp, veri tabanı erişimini sağlanmaktadır. Ayrıca, burada belirtilen "ConnectionMarker" yapısındaki kodlar "Database Tools" klasöründe bulunmaktadır.

```
30     private void NormalizeDatas()
31     {
32
33         foreach (Control item in groupBoxPersonalInfo.Controls)
34         {
35             if (item is TextBox)
36                 item.Text = "";
37         }
38
39         foreach (Control item in groupBoxCarInfo.Controls)
40         {
41             if (item is ComboBox)
42             {
43                 ComboBox comboBox = (ComboBox)item;
44                 comboBox.Items.Clear();
45                 comboBox.Items.Add("Seçim Yapınız...");
46                 comboBox.SelectedIndex = 0;
47             }
48             else if (item is TextBox)
49             {
50                 item.Text = "";
51             }
52         }
53
54         foreach (Control item in groupBoxPark.Controls)
55         {
56             if (item is ComboBox)
57             {
58                 ComboBox comboBox = (ComboBox)item;
59                 comboBox.Items.Clear();
60                 comboBox.Items.Add("Seçim Yapınız...");
61                 comboBox.SelectedIndex = 0;
62             }
63         }
64     }
```

Şekil 3.7. Araç Kayıt Ekranı Kodları-2

Şekil 3.7 de yer alan kodlarda ise; Araç Otopark Kayıt ekranında belirtilen Kişi, Araç ve Park Yeri bilgilerinin her birine erişim sağlayan kod yapısı görüntülenmektedir.

```
65     connection.Open();
66     cmd = new MySqlCommand("SELECT * FROM car_status WHERE
67         car_status_status=@x; ", connection);
68     cmd.Parameters.AddWithValue("@x", "BOS");
69     reader = cmd.ExecuteReader();
70     while (reader.Read())
71     {
72         parkingPlaceComboBox.Items.Add(reader
73             ["car_status_parkingplace"].ToString());
74     }
75     reader.Close();
76     cmd = new MySqlCommand("SELECT * FROM car_brand_info",
77         connection);
78     reader = cmd.ExecuteReader();
79     while (reader.Read())
80     {
81         brandComboBox.Items.Add(reader
82             ["car_brand_info_content"]);
83     }
84     connection.Close();
```

Şekil 3.8. Araç Kayıt Ekranı Kodları-3

Programın veri tabanındaki boş otopark alanlarının ve kayıtlı araç markalarının gösterilmesi sağlanarak ihtiyacımız olan bilgilerin seçilebilmesi, Şekil 3.8’de belirtilen kodlar ile gerçekleştirilmektedir.

```
105     private void CarParkingRecordForm_Load(object sender, EventArgs e)
106     {
107         NormalizeDatas();
108     }
109
110     private void MouseDownMethod(MouseEventArgs e)
111     {
112         dragging = true;
113         startPoint = new Point(e.X, e.Y);
114     }
115
116     private void MouseMoveMethod(MouseEventArgs e)
117     {
118         if (dragging)
119         {
120             Point p = PointToScreen(e.Location);
121             Location = new Point(p.X - this.startPoint.X, p.Y - this.startPoint.Y);
122         }
123     }
124
125     private void MouseUpMethod()
126     {
127         dragging = false;
128     }
129
130     private void closeButton_Click(object sender, EventArgs e)
131     {
132         this.Close();
133     }
134
135     private void panel1_MouseDown(object sender, MouseEventArgs e)
136     {
137         MouseDownMethod(e);
138     }
139
140     private void panel1_MouseMove(object sender, MouseEventArgs e)
141     {
142         MouseMoveMethod(e);
143     }
144
145     private void panel1_MouseUp(object sender, MouseEventArgs e)
146     {
147         MouseUpMethod();
148     }
149
150     private void label2_MouseUp(object sender, MouseEventArgs e)
151     {
152         MouseUpMethod();
153     }
154
155     private void label2_MouseMove(object sender, MouseEventArgs e)
156     {
157         MouseMoveMethod(e);
158     }
159
160     private void label2_MouseDown(object sender, MouseEventArgs e)
161     {
162         MouseDownMethod(e);
163     }
164
165     private void confirmButton_Click(object sender, EventArgs e)
166     {
167
```

Şekil 3.9. Araç Kayıt Ekranı Kodları-4

Programa ait açılan pencerelerin ekran üzerinde taşınmasını sağlayan kod yapıları Şekil 3.9 da görüntülenmektedir.

```

168     string tc = txtTc.Text.ToString().Trim();
169     string name = txtName.Text.ToString().Trim();
170     string surname = txtSurname.Text.ToString().Trim();
171     string phone = txtPhoneNumber.Text.ToString().Trim();
172     string plate = txtPlate.Text.ToString().Trim();
173     string brand = brandComboBox.Text.ToString().Trim();
174     string s = sComboBox.Text.ToString().Trim();
175     string color = txtColor.Text.ToString().Trim();
176     string parkingplace = parkingPlaceComboBox.Text.ToString()
    .Trim();
177     string date = DateTime.Now.ToString();
178
179     if (tc.Length <= 0 ||
180         name.Length <= 0 ||
181         surname.Length <= 0 ||
182         phone.Length <= 0 ||
183         plate.Length <= 0 ||
184         brand.Contains("Seçim") ||
185         s.Contains("Seçim") ||
186         color.Length <= 0 ||
187         parkingplace.Contains("Seçim"))
188     {
189         MessageBox.Show("Lütfen boş bilgi bırakmayınız...");
190         return;
191     }
192

```

Şekil 3.10. Araç Kayıt Ekranı Kodları-5

Şekil 3.10’da Programa ait belirtilen kod yapıları incelendiğinde, ekrandaki boşluklar için söz konusu programda veri girişi yapılan bilgilerin kaydedilmesini ve ilgili kontrol yapılarını içermektedir. Kontrol yapılarında, seçim yapılması gereken alanlarda, seçim yapılmadığında veya bilgi girişi olmadığında bizi uyarmaktadır.

```

193     connection.Open();
194     cmd = new MySqlCommand("INSERT INTO car_record
    (car_record_tc, car_record_name, car_record_surname,
    car_record_phone, car_record_plate, car_record_brand,
    car_record_s, car_record_color, car_record_parkingplace,
    car_record_date) VALUES " +
195         "(@tc, @name, @surname, @phone, @plate, @brand, @s,
    @color, @parkingplace, @date)", connection);
196     cmd.Parameters.AddWithValue("@tc", tc);
197     cmd.Parameters.AddWithValue("@name", name);
198     cmd.Parameters.AddWithValue("@surname", surname);
199     cmd.Parameters.AddWithValue("@phone", phone);
200     cmd.Parameters.AddWithValue("@plate", plate);
201     cmd.Parameters.AddWithValue("@brand", brand);
202     cmd.Parameters.AddWithValue("@s", s);
203     cmd.Parameters.AddWithValue("@color", color);
204     cmd.Parameters.AddWithValue("@parkingplace", parkingplace);
205     cmd.Parameters.AddWithValue("@date", date);
206     cmd.ExecuteNonQuery();
207
208     cmd = new MySqlCommand("UPDATE car_status SET
    car_status_status = 'DOLU' WHERE car_status_parkingplace =
    '" + parkingPlaceComboBox.Text.ToString() + "'",
    connection);
209     cmd.ExecuteNonQuery();
210
211     connection.Close();
212
213     NormalizeDatas();
214     MessageBox.Show("Bilgiler sisteme eklendi.");
215
216 }
217

```

Şekil 3.11. Araç Kayıt Ekranı Kodları-6

Şekil 3.11'deki kodlar, Programa tanımlanmış MySQL bağlantı yapısıdır. Program üzerindeki boşluklara girilen bilgilerin veri tabanında oluşturulan tablolardaki yerlerini belirlemektedir. Ayrıca, otopark alanı içerisinde seçmiş olduğumuz boş otopark yerinin veri tabanına kayıt işleminden sonra dolu olarak gösterilmesini sağlayan kod yapılarıdır.

```
218     private void brandAddButton_Click(object sender, EventArgs e)
219     {
220         BrandForm form = new BrandForm();
221         form.ShowDialog();
222         ExtraNormalizeDatas();
223     }
224
225     private void sAddButton_Click(object sender, EventArgs e)
226     {
227         SForm form = new SForm();
228         form.ShowDialog();
229         ExtraNormalizeDatas();
230     }
231
232     private void clearButton_Click(object sender, EventArgs e)
233     {
234         NormalizeDatas();
235     }
236
```

Şekil 3.12. Araç Kayıt Ekranı Kodları-7

Şekil 3.12'de Programın Araç Kayıt Ekranındaki marka seçimi yapılan bölümünde, istediğimiz marka bilgisinin bulunmaması durumunda yeni marka bilgilerini eklememizi sağlayan kod yapısıdır.

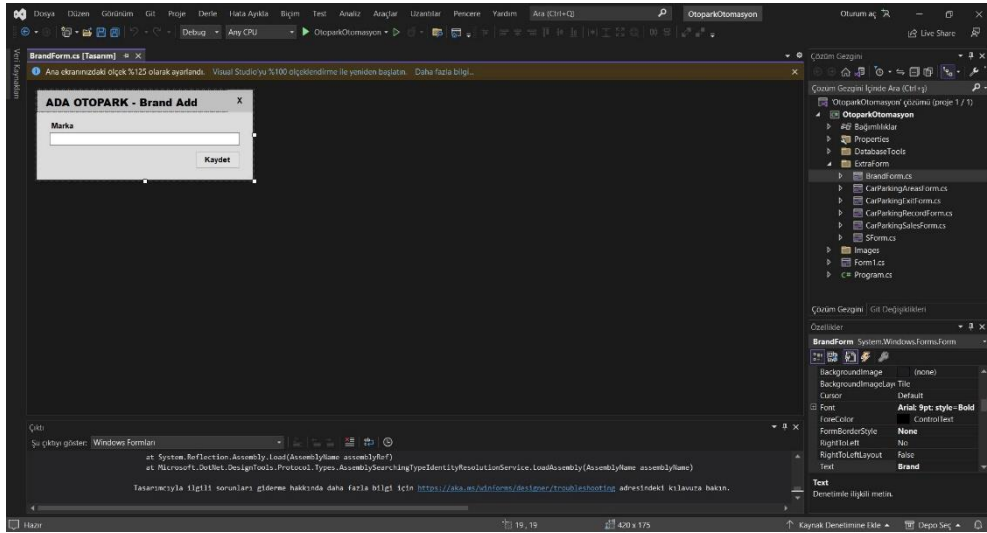
```
237     private void brandComboBox_SelectedIndexChanged(object sender, EventArgs e)
238     {
239         ComboBox comboBox = (ComboBox)sender;
240         string selected = comboBox.SelectedItem.ToString();
241
242         sComboBox.Items.Clear();
243         sComboBox.Items.Add("Seçim Yapınız...");
244         sComboBox.SelectedIndex = 0;
245
246         if (selected.Contains("Seçim")) return;
247
248         connection.Open();
249         cmd = new MySqlCommand("SELECT * FROM car_s WHERE
250                                car_s_brand=@x; ", connection);
251         cmd.Parameters.AddWithValue("@x", selected);
252         reader = cmd.ExecuteReader();
253         while (reader.Read())
254         {
255             sComboBox.Items.Add(reader["car_s_s"].ToString());
256         }
257         connection.Close();
258     }
259 }
260
```

Şekil 3.13. Araç Kayıt Ekranı Kodları-8

Şekil 3.13, Programın Araç Kayıt Ekranındaki marka seçimi yapıldıktan sonra istenilen seri bilgisinin olmaması durumunda, sisteme yeni seri bilgilerini eklemek için kullanılan kod yapısıdır.

3.2.1 ARAÇ MARKA EKLEME EKRANI

Programda, yeni bir araç markası eklenmesine ihtiyaç duyulduğunda, ilgili açılan ekrandır. Microsoft Visual Studio programı ile proje açıldığında, “ExtraForm” klasörünün içinde bulunan “BrandForm” klasörü açıldığında, ekrana ait tasarım ve ilgili kodlar görüntülenebilir. (Şekil 3.14)



Şekil 3.14. Araç Marka Ekleme

Söz konusu Otoparka gelen araçlar için, marka bilgisi eklemek amacıyla açılan ekranın kod yapıları incelendiğinde “Brandform” klasörünün yapısına ait düzen başlatılıp, veri tabanına erişimi sağlayan ve yeni eklenecek marka bilgisi ile birlikte ilgili veri tabanında daha önceden kayıtlı olup olmadığının kontrolü sağlanarak kaydedilen ve diğer formlarda olduğu gibi ekranın taşıma bilirliliğini sağlayan kodlar mevcuttur. (Şekil 3.15)

```

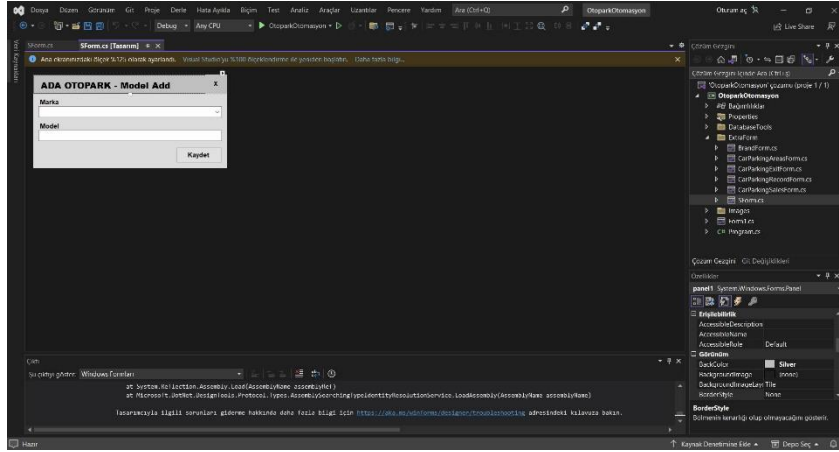
1 using MySql.Data.MySqlClient;
2 using OtoparkOtomasyon.DatabaseTools;
3 using System;
4 using System.Collections.Generic;
5 using System.ComponentModel;
6 using System.Data;
7 using System.Drawing;
8 using System.Linq;
9 using System.Text;
10 using System.Threading.Tasks;
11 using System.Windows.Forms;
12
13
14 namespace OtoparkOtomasyon.ExtraForm
15 {
16     public partial class BrandForm : Form
17     {
18
19         private MySqlConnection connection =
20             ConnectionMaker.GetConnection();
21         private MySqlCommand cmd;
22         private MySqlDataReader reader;
23
24         // Move and Drag Setup
25         private bool dragging = false;
26         private Point startPoint = new Point(0, 0);
27
28         public BrandForm()
29         {
30             InitializeComponent();
31         }
32
33         private void confirmButton_Click(object sender, EventArgs e)
34         {
35             string brand = txtBrand.Text.ToString().Trim();
36             if (brand.Length == 0) {
37                 MessageBox.Show("Lütfen bir marka giriniz.");
38                 return;
39             }
40
41             connection.Open();
42             cmd = new MySqlCommand("SELECT * FROM car_brand_info WHERE
43                 car_brand_info_content = @vbrand", connection);
44             cmd.Parameters.AddWithValue("@vbrand", brand);
45             reader = cmd.ExecuteReader();
46             if (reader.Read()) {
47                 MessageBox.Show("Böyle bir kayıt bulunmaktadır.");
48                 connection.Close();
49                 return;
50             }
51             reader.Close();
52
53             cmd = new MySqlCommand("INSERT INTO car_brand_info
54                 (car_brand_info_content) VALUES (@brand)", connection);
55             cmd.Parameters.AddWithValue("@brand", brand);
56             cmd.ExecuteNonQuery();
57             connection.Close();
58             MessageBox.Show("Marka sisteme eklendi.");
59
60             txtBrand.Text = "";
61         }
62     }
63 }

```

Şekil 3.15. Araç Marka Ekleme Kod Yapısı

3.2.2 ARAÇ MODEL EKLEME EKLANI

Programda, yeni bir araç modeli eklenmesine ihtiyaç duyulduğunda açılan ekrandır. Öncelikle söz konusu program, ilgili kullanıcıdan marka seçimi yapılmasını istemektedir. Marka seçimi yapıldıktan sonra model bilgisi girilerek kayıt işlemi gerçekleştirilmektedir. Microsoft Visual Studio programı ile proje açıldığında, "ExtraForm" klasörünün içinde bulunan "SForm" yapısı açıldığında ekrana ait tasarım ve kodlar görüntülenebilir. (Şekil 3.16)



Şekil 3.16. Araç Model Ekleme

```

1 using MySql.Data.MySqlClient;
2 using OtoparkOtomasyon.DatabaseTools;
3 using System;
4 using System.Collections.Generic;
5 using System.ComponentModel;
6 using System.Data;
7 using System.Drawing;
8 using System.Linq;
9 using System.Text;
10 using System.Threading.Tasks;
11 using System.Windows.Forms;
12
13 namespace OtoparkOtomasyon.ExtraForm
14 {
15     public partial class SForm : Form
16     {
17         private MySqlConnection connection =
18             ConnectionMaker.GetConnection();
19         private MySqlCommand cmd;
20         private MySqlDataReader reader;
21
22         // Move and Drag Setup
23         private bool dragging = false;
24         private Point startPoint = new Point(0, 0);
25
26         public SForm()
27         {
28             InitializeComponent();
29         }
30
31         private void button1_Click(object sender, EventArgs e)
32         {
33             string ss, brand;
34             ss = txtS.Text.ToString().Trim();
35             brand = brandComboBox.Text.ToString().Trim();
36
37             if (ss.Length <= 0 || brand.Contains("Seçim")) {
38                 MessageBox.Show("Lütfen boş bilgi bırakmayınız...");
39                 return;
40             }
41
42             connection.Open();
43             cmd = new MySqlCommand("SELECT * FROM car_s WHERE car_s_s = @vsx", connection);
44             cmd.Parameters.AddWithValue("@vsx", ss);
45             reader = cmd.ExecuteReader();
46             if (reader.Read()) {
47                 MessageBox.Show("Böyle bir seri kaydı bulunmaktadır.");
48                 connection.Close();
49                 return;
50             }
51             reader.Close();
52
53             cmd = new MySqlCommand("INSERT INTO car_s (car_s_brand,
54 car_s_s) VALUES (@brand, @ss)", connection);
55             cmd.Parameters.AddWithValue("@brand", brand);
56             cmd.Parameters.AddWithValue("@ss", ss);
57             cmd.ExecuteNonQuery();
58             connection.Close();
59             MessageBox.Show("Seri kayıt edildi...");
60             NormalizeDatas();
61         }
62
63         private void NormalizeDatas() {
64             txtS.Text = "";
65             brandComboBox.Items.Clear();
66             brandComboBox.Items.Add("Seçim Yapınız...");
67             brandComboBox.SelectedIndex = 0;
68
69             connection.Open();
70             cmd = new MySqlCommand("SELECT * FROM car_brand_info",
71 connection);
72             reader = cmd.ExecuteReader();
73             while (reader.Read())
74                 brandComboBox.Items.Add(reader
75 ["car_brand_info_content"]);
76             connection.Close();
77         }
78     }
79 }

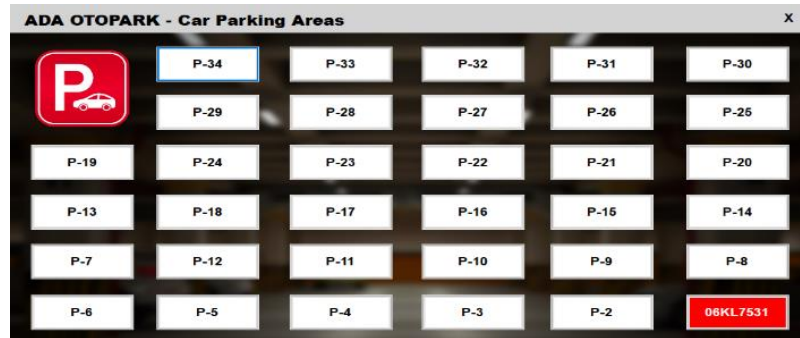
```

Şekil 3.17. Araç Model Ekleme Kod Yapısı

Şekil 3.17’da araç model bilgisi eklemek için, açılan ekranın kod yapıları incelendiğinde, “Sform” yapısına ait düzeni başlatan ve veri tabanı erişimi sağlayan, yeni eklenecek modeller için önce marka seçimi yaptıran, daha sonra da yeni eklenecek model bilgisi ile ilgili olarak veri tabanında önceden kayıtlı olup olmadığının kontrolünü sağlayarak kaydedilen ve diğer formlarda olduğu gibi ekranın taşına bilirliliğini sağlayan ilgili kodlar mevcuttur.

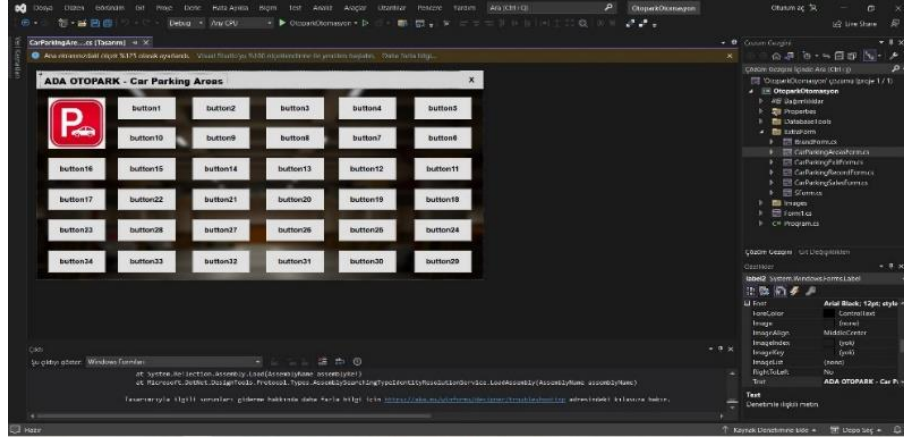
3.3 ARAÇ OTOPARK YERLERİ

Programın bu bölümünde, otopark alanında bulunan araç park yerlerine ait doluluk durumları görülmektedir. Dolu otopark alanlarını programda kırmızı renkli ve plaka bilgisi ile gösterip, boş otopark alanlarını ise otopark numarası ile beyaz renkli göstermektedir. Böylelikle otopark işletmesindeki doluluk oranı ortaya çıkmaktadır. (Şekil 3.18)



Şekil 3.18. Araç Otopark Yerleri Uygulama Ekranı

Microsoft Visual Studio programı ile projeyi açtığımızda, ExtraForm klasörünün içinde bulunan “CarParkingAreasForm” kısmına bakıldığında ekrana ait tasarımı ve kodları görüntüleyebiliriz. (Şekil 3.19)



Şekil 3.19. Araç Otopark Yerleri Ekran Tasarımı

Araç Otopark Yerlerini görebilmek amacıyla açılan ekranda “CarParkingAreasForm” kodlarının yapısı görülmektedir. Bu Form yapısını başlatan ve veri tabanına erişimi sağlayan bu kodların amacı, liste türünün görünümünü butonlar şeklinde göstermektedir. Ayrıca, söz konusu kod yapısı süreci, dolu otopark alanlarının görünümünün plaka bilgisi ile birlikte kırmızı renkli ve boş otopark alanlarının görünümünün ise beyaz renkli olmasını sağlar. Diğer formlarda olduğu gibi ekranda açılan pencerelerin taşınabilirliğini sağlayan kodlarda mevcuttur. (Şekil 3.20)

```

1 using MySql.Data.MySqlClient;
2 using OtoparkOtomasyon.DatabaseTools;
3 using System;
4 using System.Collections.Generic;
5 using System.ComponentModel;
6 using System.Data;
7 using System.Diagnostics;
8 using System.Drawing;
9 using System.Linq;
10 using System.Text;
11 using System.Threading.Tasks;
12 using System.Windows.Forms;
13
14 namespace OtoparkOtomasyon.ExtraForm
15 {
16     public partial class CarParkingAreasForm : Form
17     {
18         private MySqlConnection connection =
19             ConnectionMaker.GetConnection();
20         private MySqlCommand cmd;
21         private MySqlDataReader reader;
22         private List<Button> parkButtonList = new List<Button>();
23
24         // Move and Drag Setup
25         private bool dragging = false;
26         private Point startPoint = new Point(0, 0);
27
28         public CarParkingAreasForm() {
29             InitializeComponent();
30             pictureBox1.Controls.Add(pictureBox2);
31             pictureBox2.BackColor = Color.Transparent;
32         }
33
34         private void closeButton_Click(object sender, EventArgs e) {
35             this.Close();
36         }
37
38         private void CarParkingAreasForm_Load(object sender, EventArgs
39             e) {
40             LoadingPark();
41             connection.Open();
42             cmd = new MySqlCommand("SELECT * FROM car_record",
43                 connection);
44             reader = cmd.ExecuteReader();
45             while (reader.Read()) {
46                 foreach (Button item in parkButtonList) {
47                     if (item.Text.Equals(reader
48                         ["car_record_parkingplace"].ToString())) {
49                         item.BackColor = Color.Red;
50                         item.Text = reader["car_record_plate"].ToString
51                             ();
52                         item.ForeColor = Color.White;
53                     }
54                     continue;
55                 }
56             }
57             connection.Close();
58         }
59
60         private void LoadingPark() {
61             int x = 1;
62             foreach (Control item in Controls) {
63                 if (item is Button && !item.Text.Equals("X")) {
64                     parkButtonList.Add((Button)item);
65                     item.Text = "P-" + x;
66                     item.Name = "p-" + x;
67                     item.BackColor = Color.White;
68                     x++;
69                 }
70             }
71         }
72     }
73 }

```

Şekil 3.20. Araç Otopark Yerleri Kod Yapısı

3.4 ARAÇ OTOPARK ÇIKIŞ SAYFASI

Programın bu bölümünde, açılan ekranda otopark içerisindeki park etmiş araçlara ait çıkış işlemlerinin yapılması sağlanmaktadır. Plaka seçiminin yapılması ile araca ait bilgiler ekrana düşmektedir. Aracın otopark içerisinde bulunduğu süre ve ödenmesi gereken ücret hesabı yapılmaktadır. (Şekil 3.21)

ADA OTOPARK - Car Parking Exit
X

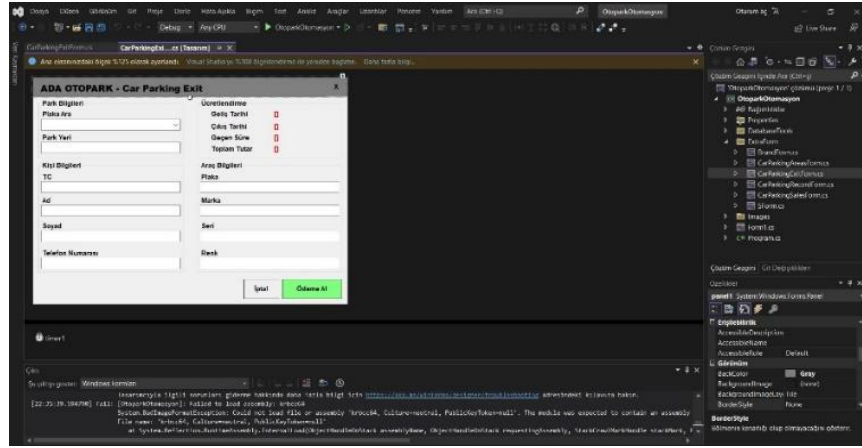
<div style="border: 1px solid gray; padding: 2px;"> Park Bilgileri Plaka Ara <input type="text" value="06KL7531"/> </div> <div style="border: 1px solid gray; padding: 2px; margin-top: 5px;"> Park Yeri <input type="text" value="P-1"/> </div> <div style="border: 1px solid gray; padding: 2px; margin-top: 5px;"> Kişi Bilgileri TC <input type="text" value="113344556677"/> </div> <div style="border: 1px solid gray; padding: 2px; margin-top: 5px;"> Ad <input type="text" value="ALİ"/> </div> <div style="border: 1px solid gray; padding: 2px; margin-top: 5px;"> Soyad <input type="text" value="YAPRAK"/> </div> <div style="border: 1px solid gray; padding: 2px; margin-top: 5px;"> Telefon Numarası <input type="text" value="05554447799"/> </div>	<div style="border: 1px solid gray; padding: 2px;"> Ücretlendirme Geliş Tarihi 5.05.2023 11:03:44 Çıkış Tarihi 5.05.2023 11:04:39 Geçen Süre 00:00:55 Toplam Tutar 0,69 TL </div> <div style="border: 1px solid gray; padding: 2px; margin-top: 5px;"> Araç Bilgileri Plaka <input type="text" value="06KL7531"/> </div> <div style="border: 1px solid gray; padding: 2px; margin-top: 5px;"> Marka <input type="text" value="Peugeot"/> </div> <div style="border: 1px solid gray; padding: 2px; margin-top: 5px;"> Seri <input type="text" value="2008"/> </div> <div style="border: 1px solid gray; padding: 2px; margin-top: 5px;"> Renk <input type="text" value="KIRMIZI"/> </div>
--	--

İptal

Ödeme Al

Şekil 3.21. Araç Otopark Çıkış Ekranı

Microsoft Visual Studio programı ile proje açıldığında, “ExtraForm” klasörünün içinde bulunan “CarParkingExitForm” yapısı açıldığında, ekrana ait tasarım ve ilgili kodlar görüntülenebilir. (Şekil 3.19)



Şekil 3.22. Araç Otopark Çıkış Ekran Tasarımı

Araç Otopark Çıkış Sayfası gösterimi için açılan ekranın kodları incelendiğinde “CarParkingExitForm” yapısına ait düzeni başlatan ve veri tabanı erişimi sağlandıktan sonra Programda plaka seçiminin yapılması ile birlikte ilgili plakaya ait verilerin getirilmesini sağlayan ve ücret hesabını yapan kod yapısıdır. (Şekil 3.20)

```

110
111     bool plaStatus;
112     private void plakaAraComboBox_SelectedIndexChanged(object sender, EventArgs e)
113     {
114         ComboBox comboBox = (ComboBox)sender;
115         string plates = (string)comboBox.SelectedItem;
116         ExtraNormalizeDatas();
117         plaStatus = false;
118         if (plates.Contains("Seçin"))
119             return;
120
121         connection.Open();
122         cmd = new MySqlCommand("SELECT * FROM car_record WHERE car_record_plate = @plate", connection);
123         cmd.Parameters.AddWithValue("@plate", plates);
124         reader = cmd.ExecuteReader();
125         while (reader.Read())
126         {
127             txtTc.Text = reader["car_record_tc"].ToString();
128             txtName.Text = reader["car_record_name"].ToString();
129             txtSurname.Text = reader["car_record_surname"].ToString();
130             txtPhoneNumber.Text = reader["car_record_phone"].ToString();
131             txtPlate.Text = reader["car_record_plate"].ToString();
132             txtBrand.Text = reader["car_record_brand"].ToString();
133             txtModel.Text = reader["car_record_s"].ToString();
134             txtColor.Text = reader["car_record_color"].ToString();
135             parkVeriTxt.Text = reader["car_record_parkingplace"].ToString();
136             gelisTarihLabel.Text = reader["car_record_date"].ToString();
137         }
138         connection.Close();
139         plaStatus = true;
140
141     string totalprice = "";
142     private void timer1_Tick(object sender, EventArgs e)
143     {
144         if (plaStatus)
145         {
146             cikisTarihLabel.Text = DateTime.Now.ToString();
147             DateTime x, y;
148             x = DateTime.Parse(gelisTarihLabel.Text);
149             y = DateTime.Parse(cikisTarihLabel.Text);
150             TimeSpan time;
151             time = y - x;
152             gecensureLabel.Text = time.ToString();
153             totalprice = (time.TotalMinutes * 0.75).ToString("0.00");
154             toplamTutarLabel.Text = totalprice + " TL";
155         }
156     }
157     private void clearButton_Click(object sender, EventArgs e)
158     {
159         NormalizeDatas();
160     }

```

Şekil 3.23. Araç Otopark Çıkış Ekranı Kod Yapısı -1

Programda, ödeme al butonuna basılması ile birlikte veri tabanımızda kayıtlı olan araç bilgilerinin silinmesi, aracın kayıtlı olduğu otopark alanının boş olarak gösterilmesi ve programın diğer ekranlarından biri olan Araç Otopark Satış Listesi bölümündeki raporlama için verileri tablo formundan satır formunda aktarılması sağlanan kod yapısıdır. (Şekil 3.20)

```

158
159 private void getPaidButton_Click(object sender, EventArgs e){
160     // id ->
161     // string tprice, tc, name, surname, phone, plate,
162     brand, s, color, parkingplace, recorddate, exitdate;
163     string tprice, tc, name, surname, phone, plate, brand, s,
164     color, parkingplace, recorddate, exitdate;
165     tprice = totalprice;
166     tc = txtTc.Text;
167     name = txtName.Text;
168     surname = txtSurname.Text;
169     phone = txtPhoneNumber.Text;
170     plate = txtPlate.Text;
171     brand = txtBrand.Text;
172     s = txtModel.Text;
173     color = txtColor.Text;
174     parkingplace = parkYeriTxt.Text;
175     recorddate = gelisTarihiLabel.Text;
176     exitdate = cikisTarihiLabel.Text;
177
178     string fullstring = tprice + "#" + tc + "#" + name + "#" +
179     surname + "#" + phone + "#" + plate + "#" + brand + "#" +
180     s + "#" + color + "#" + parkingplace + "#" + recorddate +
181     "#" + exitdate;
182
183     connection.Open();
184     cmd = new MySqlCommand("INSERT INTO car_sales_list
185     (car_sales_list_content) VALUES (@fullstring)",
186     connection);
187     cmd.Parameters.AddWithValue("@fullstring", fullstring);
188     cmd.ExecuteNonQuery();
189     cmd = new MySqlCommand("DELETE FROM car_record WHERE
190     car_record_plate = @plate", connection);
191     cmd.Parameters.AddWithValue("@plate", plate);
192     cmd.ExecuteNonQuery();
193
194     cmd = new MySqlCommand("UPDATE car_status SET
195     car_status_status = 'BOS' WHERE car_status_parkingplace =
196     '" + parkingplace + "'", connection);
197     cmd.ExecuteNonQuery();
198     connection.Close();
199
200     NormalizeDatas();
201     MessageBox.Show(totalprice+" TL Ödeme alındı ve sisteme

```

Şekil 3.24. Araç Otopark Çıkış Ekranı kod Yapısı -2

Ayrıca, söz konusu programın kod yapısında, diğer ekranlarda olduğu gibi pencerelerin taşınmasını sağlayan ve “İptal” butonuna basılması durumunda ekran üzerindeki verileri temizleyen kodlarda bulunmaktadır.

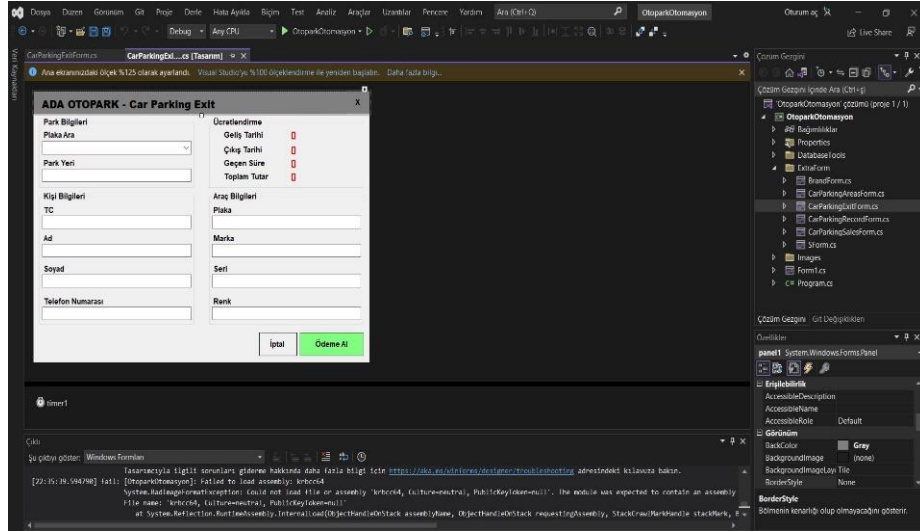
3.5 ARAÇ OTOPARK SATIŞ LİSTESİ

Programın son kısmı olan Araç Otopark Satış Listesinde otopark işletmesine giriş ve çıkış işlemi yapılmış araçlara ilişkin bilgiler ve tahsil edilen ücretlerin bilgileri gösterilmektedir. Böylece, söz konusu işletme kontrol edilebilir bir yapıya sahip olmaktadır. (Şekil 3.25)

ADA OTOPARK - Car Parking Sales List						
ID	FIYAT	TC	İSİM	SOYİSİM	TELEFON	PLAKA
14	0,96	113344556677	ALİ	YAPRAK	05554447799	06KL7531
13	0,69	1236549874	EMRAH	GÖK	05356669933	35AD4612
12	0,10	99999999999999	Burak	Yılmaz	5558889966	66DD99

Şekil 3.25. Araç Otopark Satış Listesi Ekranı

Microsoft Visual Studio programı ile proje açıldığında, “ExtraForm” klasörünün içinde bulunan “CarParkingSalesForm” yapısı tıklandığında, ekrana ait tasarım ve kodlar görüntülenebilir.



Şekil 3.26. Araç Otopark Satış Listesi Ekran Tasarımı

Programın bu kısmında ise, kod yapılarının diğer ekranlarında olduğu gibi ekranın çalışmasını, veri tabanı bağlantısını ve açılan ekranın taşıma birliğini sağlayan yapılar mevcuttur. Ayrıca, Araç Otopark Çıkış Ekranında ödeme al butonuna basılması ile

birlikte veri tabanının satır olarak kaydedilen bilgilerinin ekranda tablo olarak gösterilmesini sağlayan kod yapısı da bulunmaktadır. (Şekil.23)

```
47
48 private void CarParkingSalesForm_Load(object sender, EventArgs e){
49     // id ,tprice, tc, name, surname, phone, plate, brand, s,
        color, parkingplace, recorddate, exitdate;
50     dataGridView1.ColumnCount = 13;
51     dataGridView1.Columns[0].Name = "ID";
52     dataGridView1.Columns[1].Name = "FİYAT";
53     dataGridView1.Columns[2].Name = "TC";
54     dataGridView1.Columns[3].Name = "İSİM";
55     dataGridView1.Columns[4].Name = "SOYİSİM";
56     dataGridView1.Columns[5].Name = "TELEFON";
57     dataGridView1.Columns[6].Name = "PLAKA";
58     dataGridView1.Columns[7].Name = "MARKA";
59     dataGridView1.Columns[8].Name = "SERİ";
60     dataGridView1.Columns[9].Name = "RENK";
61     dataGridView1.Columns[10].Name = "PARK YERİ";
62     dataGridView1.Columns[11].Name = "KAYIT TARİHİ";
63     dataGridView1.Columns[12].Name = "ÇIKIŞ TARİHİ";
64
65     dataGridView1.Columns[11].Width = 180;
66     dataGridView1.Columns[12].Width = 180;
67
68
69     connection.Open();
70     cmd = new MySqlCommand("SELECT * FROM car_sales_list ORDER
        BY car_sales_list_id DESC", connection);
71     reader = cmd.ExecuteReader();
72     while (reader.Read()){
73         string demos = reader["car_sales_list_id"].ToString()
            +"#" +reader["car_sales_list_content"].ToString();
74         string[] fulls = demos.Split("#");
75         dataGridView1.Rows.Add(fulls);
76     }
77     connection.Close();
78 }
```

Şekil 3.27. Araç Otopark Satış Listesi Kod Yapısı

BÖLÜM 4

SONUÇLAR VE ÖNERİLER

4.1 SONUÇLAR

Bu çalışmada oluşturulan program sayesinde, otopark işletmesinin hem ekonomik hem de hızlı bir şekilde işletilebileceği görülmüştür. Kullanım kolaylığı sayesinde, işletme otomatik olarak rapor alabilmektedir. Ayrıca otoparkın anlık kapasite kontrolü de sağlanmaktadır. Bu sayede araçlar söz konusu otoparkta doğru yönlendirilmekte ve fazla müşteri kabul ederek araç sahiplerinin gereksiz yere zamanın harcanmasının önüne geçilmektedir.

Araç sahipleri için uygulanacak uygun bir tarife ile birlikte otopark otomasyonu kullanan otopark işletmelerine talep artacaktır. Böylelikle hem şehir trafiği rahatlayacak hem de araçlar daha güvenli alanlarda park etmiş olacaktır.

4.2. ÖNERİLER

Araçların durmalarına gerek kalmadan otoparka giriş ve çıkış yapabilecekleri ilave entegre sistemler ile otopark işletmesi daha modern bir yapıya sahip olabilir. Böyle bir sistem ile hem personel maliyetinin azalması sağlanabilir hem de otopark giriş ve çıkışlarındaki zaman kaybının önüne geçilebilir.

Bu model, bir şehrin tüm otoparklarında kullanılması durumunda, şehrin otopark yönetiminin kontrol edilebilirliğini sağlayacaktır. Şehirdeki otopark arama sorununun önüne geçerek doğru yönlendirilebilir bir sistem haline gelebilir. Bu sayede, ilgili uygulamayı kullanan şehirler daha modern bir görünüme sahip olacaktır.

Özgeçmiş

Adı Soyadı: Mehmet Ali Oktay

Eğitim:

2005–2009 Süleyman Demirel Üniversitesi, Yapı Tasarım Ögt. Bölümü

İş Deneyimi:

2009 – 2011 Proje Etüt Mimarlık Ltd. Şti.

2011 – İzmir Büyükşehir Belediyesi

Yayınlar (varsa):

1. "Modifiye Çökme Deney Cihazı ile Klasik Çökme Deneyinin Karşılaştırılması",
Uluslararası Teknolojik Bilimler Dergisi, c. 1, sayı. 1, ss. 18-28, Mar. 2009