

# EBH

## Restaurant Automation

Ensar Akbaş , Hakan Aytuğ Fırat , Berkay Yüce

### ÖZET

*Projemiz en genel anlamıyla bir lokantanın müşteri siparişlerini yönetebilmesi amacıyla yapılmıştır. Kasiyer ve admin kullanıcılarına ait özel kullanım alanları vardır. Burada kasiyer ve admin olarak giriş yapan kullanıcılar lokantaya ait çeşitli işlemler yapabilmektedirler.*

*Anahtar kelimeler: Admin, kasiyer, müşteri, sipariş.*

### ABSTRACT

*In the most general sense, our project is made for a restaurant to manage customer orders. There are special usage areas for cashier and admin users. Here, users who log in as cashier and admin can perform various operations of the restaurant.*

*Keywords : Admin, cashier, customer, order.*

## 1. GİRİŞ

Lokanta otomasyonu projemizde bir restorantda kasiyerin, müşterinin istediği şekilde sipariş oluşturabilmesine olanak sağlayan bir sistem kurmaya çalıştık. Aynı zamanda bir lokantada olabilecek başka özellikleri de eklemeye çalıştık. Lokanta otomasyonunda iki çeşit giriş

Bulunmaktadır. Bunlar kasiyer ve admin kullanıcısı şeklindedir.

Kasiyer, restorandaki boş olan masalara ilgili müşteriyi yönlendirmektedir. Aynı zamanda kasiyer, müşterinin hangi ürünü, hangi üründen ne kadar aldığını ve fiyatlarını görebilmektedir.

Admin olarak giriş yapan kullanıcı personel yönetimi, yeni ürün ekleme gibi belli başlı alanları yönetebilmektedir. Personel yönetimi ile lokantada çalışmakta olan personel bilgilerini görüntüleyebilir. Aynı zamanda yeni personele ait bilgiler de ekleyebilir. Admin, personel bilgilerini güncelleyip personel silme işlemini de gerçekleştirebilir.

## 2. UYGULAMA FOTOĞRAFLARI

Projemizde admin ve kasiyere ait iki adet ana menü ve giriş sayfaları bulunmaktadır. Aynı zamanda sipariş oluşturma, masa yönetimi, ürün seçme, personel yönetimi, ürün ekleme olacak şekilde toplam on adet ekran bulunmaktadır. Sipariş oluşturma, masaların doluluğu, ürün seçimi gibi ekranlar kasiyer kullanıcısı; personel yönetimi, ürün ekleme, ürün görüntüleme gibi ekranlar ise admin kullanıcısı tarafından görüntülenmektedir. Aşağıda uygulama içinden bazı görüntüler bulunmaktadır.

### Personel Giriş Sayfası

Kullanıcı Adı

Şifre

[Kayıt Ol](#) [Şifremi Unuttum](#)

Doğrulama

0 + 0 Sonuç

### Yönetici Giriş Sayfası

Kullanıcı Adı

Şifre

Doğrulama

0 + 0

[Şifremi Unuttum](#)

Görsel 1: Personel ve yönetici giriş sayfaları

Anasayfa
 Siparişler
 Menü
 Masalar
 Çıkış

Personel Giriş  
 Yönetici Giriş

Görsel 2: Admin ve kasiyer ana menü sayfaları

#### Personel Bilgileri

GÖREV ID	İSİM	SOYISİM	MAAŞ

#### Yeni personel bilgileri

Görev:

İsim:

Soyisim:

Maaş:

#### Görev Bilgileri

GÖREV NO	GÖREV ADI

#### Yeni görev bilgileri

Görev No:

Görev Adı:

Anasayfa
 Siparişler
 Menü
 Masalar
 Ürünler
 Personeller
 Çıkış

#### Ürünleri Gözetile

Ürün id:

Ürün Kategorisi:













Ürün Adı:

Ücret:

Stok Durumu:

Resim:

Ürün Ekle

 <p>D11. Hamburger Piyaz: 80 TL</p>	 <p>D12. Manti Piyaz: 150 TL</p>	 <p>D14. Harpaz Sarma Piyaz: 120 TL</p>	 <p>D20. Et Kavurma Piyaz: 210 TL</p>
 <p>Tavuk</p>			
 <p>Ipecek</p>	 <p>D18. Su Piyaz: 5 TL</p>	 <p>D17. Acik Aynan Piyaz: 25 TL</p>	 <p>D18. Baklava Piyaz: 90 TL</p>
 <p>Tatli</p>			 <p>D19. CheeseCake Piyaz: 90 TL</p>
 <p>Salata</p>			
<p>Anahtarli</p>			

Masa No:

Masa Kapasitesi:

Masa Durumu:

☐ Masa boş  
☐ Masa illi

Kapasiteyi Gözetme

Arayışta

1

4

4

14

15

23

1

4

4

14

15

23

1

4

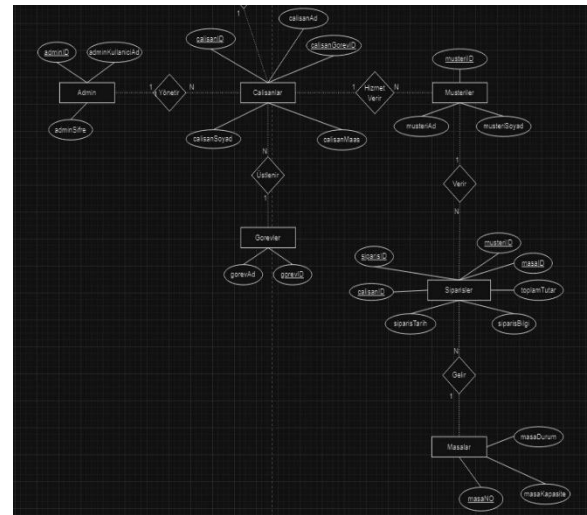
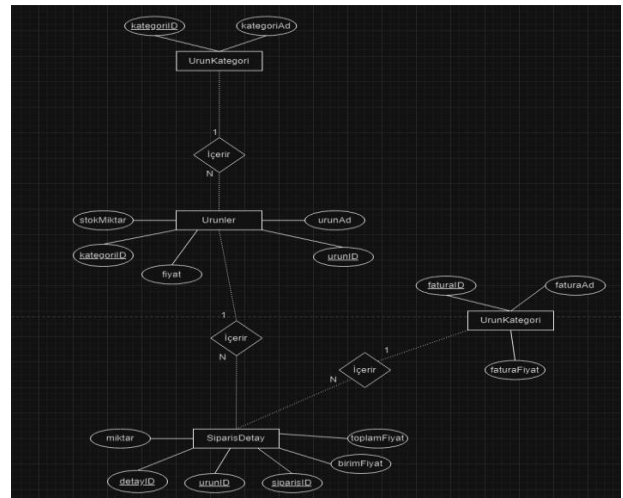
4

14

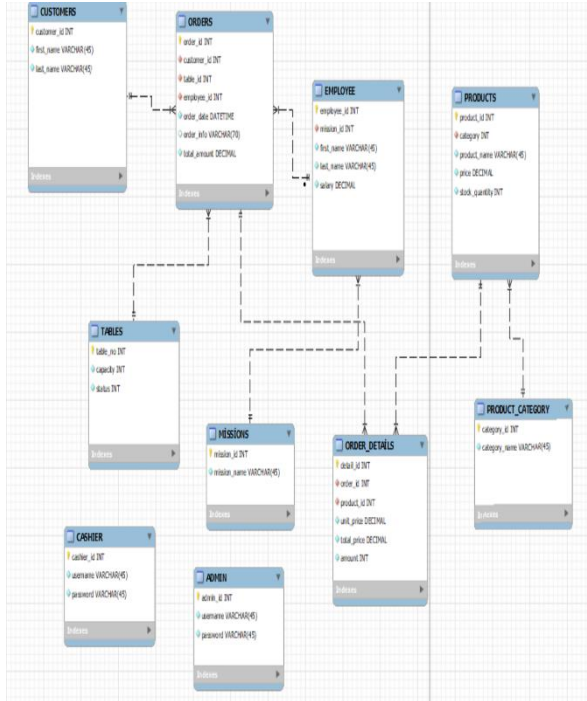
15

23

### 3. VARLIK İLİŞKİ DİYAGRAMI



Projedemizde müşteriler, siparişler, sipariş detayları, ürünler, ürün kategorisi, masalar, admin, kasiyer, personel ve personel görevleri olacak şekilde on bir tane tablo vardır. Ürünler ve ürün kategorisi kendi arasında; personeller ve görevler kendi arasında olacak şekilde birçok tablo kendi arasında ilişkili durumdadır. Bunu gösteren MySQL Workbench üzerinden oluşturduğumuz resim aşağıda verilmiştir.



```
try
{
    connection.Open();
    command.ExecuteNonQuery();
}
catch (MySqlException ex)
{
    MessageBox.Show("Hata: " + ex.Message);
}
```

**INSERT INTO Sorgusu:** INSERT INTO, SQL dilinde yeni bir kaydı bir tabloya eklemek için kullanılan bir deyimdir. Bu deyim, veritabanındaki bir tabloya yeni veri satırları eklemenizi sağlar.

**INSERT INTO** tablo\_adi (sütun1, sütun2, ..., sütunN)

**VALUES** (değer1, değer2, ..., değerN);

Formatında kullanılır.

## 4. AÇIKLAMALAR

**MySqlConnection ile veritabanına bağlantı:** MySqlConnection sınıfı, .NET Framework'te MySQL veritabanlarına bağlantı kurmak ve yönetmek için kullanılır. Bu sınıf, aşağıdaki temel işlevleri sağlar.

**Veritabanına Bağlanma:** MySqlConnection nesnesini oluşturmak ve bağlantı bilgilerini (sunucu adı, veritabanı adı, kullanıcı adı, parola) ayarlayarak veritabanına bağlantı kurabilirsiniz. Open() metodunu kullanarak bağlantıyı açabilir ve Close() metodunu kullanarak bağlantıyı kapatabilirsiniz.

```
string connectionString = "server=localhost;uid=root;pwd=12345;database=mydb";
// Veritabanına eklenecek değerler
string insertValue1 = grvNO.Text.ToString();
string insertValue2 = grvName.Text;
```

**SELECT Sorgusu:** SELECT sorgusu, SQL dilinde veritabanından veri seçmek için kullanılan bir deyimdir. Bu deyim, belirli bir tablodaki veya tablolardaki verileri filtrelemenize ve görüntülemenize olanak tanır.

string selectSQL = "SELECT \* FROM siparisler";

SELECT sütun1, sütun2, ..., sütunN

FROM tablo\_adi

[WHERE koşul]; Formatında kullanılır.

Karmaşık sorgular oluşturmak için JOIN, UNION, INTERSECT gibi diğer SQL deyimleriyle birlikte kullanılabilir.

SELECT sorgusu ile seçilen veriler, geçici bir sonuç kümesi oluşturur ve bu sonuç kümesi üzerinde işlemler yapılabilir.

#### UPDATE Sorgusu:

Veritabanı yönetiminde kullanılan bir SQL komutudur. Bu komut, bir veya daha fazla kaydın verilerini değiştirmek veya güncellemek için kullanılır. Özellikle "UPDATE" komutu, mevcut bir kaydın verilerini güncellemek veya belirli koşullara uyan kayıtları güncellemek için sıkça kullanılır.

DELETE Sorgusu: DELETE sorgusu, SQL dilinde veritabanından veri silmek için kullanılan bir deyimdir. Bu deyim, belirli bir tablodaki veya tablolardaki verileri filtrelemenize ve silmenize olanak tanır.

DELETE FROM tablo\_adi

[WHERE koşul]; Formatında kullanılır.

```
// Verileri oku
using (MySqlDataReader reader = command.ExecuteReader())
{
    // Her satırı ListView'e ekle
    while (reader.Read())
    {
        int m_id = reader.GetInt32("mission_id");
        string gorevId = m_id.ToString();
        string m_name = reader.GetString("mission_name");

        ListViewItem item = new ListViewItem(gorevId);
        item.SubItems.Add(m_name);
        Gorevler.Items.Add(item);
    }
}
```

Veritabanı işlemleri sırasında olası hataları işlemek için try-catch bloğu kullanılır.

MySqlConnection ve MySqlCommand nesnelerini kullanırken her zaman using

bloğu kullanın. Bu, nesnelerin otomatik olarak kapatır.

#### JOIN (İç Birleştirme):

JOIN terimi, iki veya daha fazla tabloyu belirli bir ortak sütuna göre birleştirme işlemidir.

INNER JOIN olarak da adlandırılır.

İki tabloyu birleştirirken, birleştirme kriteri olarak kullanılan ortak sütun üzerinde eşleşen değerlere sahip satırları döndürür. Eşleşme bulunmayan satırları döndürmez, yalnızca eşleşen satırları getirir. Genellikle ON anahtar kelimesiyle birleştirme kriteri belirtilir.

#### LEFT JOIN (Sol Birleştirme):

LEFT JOIN terimi, sol tablodaki tüm satırları ve sağ tablo ile eşleşen satırları birleştirme işlemidir. Eğer sağ tabloda eşleşme yoksa, sol tablodaki veriler yine getirilir ve sağ tablodaki sütunlar için NULL değerleri atanır.

Sol tablo, ana tablo olarak kabul edilir.

```
string sql2 = "SELECT * " +
    "FROM orders " +
    "LEFT JOIN tables ON tables.table_no = orders.table_id " +
    "JOIN customers ON customers.customer_id = orders.customer_id ";
```

RIGHT JOIN (Sağ Birleştirme):

RIGHT JOIN terimi, sağ tablodaki tüm satırları ve sol tablo ile eşleşen satırları birleştirme işlemidir. Eğer sol tabloda eşleşme yoksa, sağ tablodaki veriler yine getirilir ve sol tablodaki sütunlar için NULL değerleri atanır. Sağ tablo, ana tablo olarak kabul edilir.

Bu JOIN türleri, veritabanında bulunan ilişkili tablolar arasındaki verileri birleştirmek ve istenen sonuçları elde etmek için kullanılır. Hangi JOIN türünün kullanılacağı, birleştirme işlemi sonucunda istenilen veri setine ve ilişkiye bağlıdır.

```
string sql = "SELECT product_id, product_name ,price,category_name,stock_quantity,location FROM products " +  
    "Inner Join product_category on products.category=product_category.category_id where " +  
    "category_id=@categoryid ;";
```

## YARARLANILAN KAYNAKLAR

Youtube: C# & MySQL dersleri

ChatGPT

Gemini

Draw.io: ER Diyagramı

Renderforest: Logo tasarımı

