

DERS KODU-ADI :SWE202-YAZILIM MİMARİSİ

ÖDEV/PROJE İSMİ :YAZILIM MİMARİSİ DÖNEM PROJESİ

PROJE EKİP ÜYELERİ:

ÖĞRENCİ ADI-SOYADI	ÖĞRENCİ NO	İMZA
Şevval Öveyik	22040301034	
Eslem Berra Özel	22040301016	

BÖLÜMÜ :YAZILIM MÜHENDİSLİĞİ

ÖĞRETİM GÖREVLİSİ :Dr. Suat UĞURLU

TARİH :24.05.2024

Senaryo

Yazılım geliřtiren bir firmada ekip lideri olarak alıřıyorsunuz.

Ara kiralama řirketi olan bir mřteriniz, verdiėi hizmeti internet zerinden online olarak sunmayı planladığını syleyerek sizinle iletiřime geti. Mřteri ile yaptığınız grřmelerden řu bilgileri aldınız;

- Mevcut durumda mřteri sadece telefon ile brořrlerini gren mřterilerine hizmet veriyor.
- Online olarak vereceėi hizmet ile hem internet zerinden tarayıcı ile hem de mobil uygulama ile geliřtirilen siteye eriřim mmkn olmalıdır.
- Bu sitede havuzdaki farklı sınıf ve modellerdeki aralar listelenebilmeli, fiyatları grntlenebilmeli ve mřteri rezervasyon yapabilmeli veya doėrudan aracı kiralayabilmelidir.
- Rezervasyon iin iletilen telefon numarasına gelen mesaj ile mřteri kimliği doėrulanmalıdır. Doėrudan kiralamada deme ise kredi kartı ile yapılabilmelidir.
- Sistem 7/24 alıřmalı, kesintiler uyarı olarak ilgili kiřilere iletilebilmelidir. Sistem eriřim mřteri deyimi ile rakiplerdekinden daha performanslı olmalıdır. Ayrıca mřteri sistemin ok gvenli olmasını istemekte, kiřisel bilgi tutulacağı iin nemli bilgilerin kanun ve ynetmeliklere gre tutulması ve iřlenmesi gerektiğini sylemektedir. Hacker saldırılarına karřı sistem dayanıklı olmalı, sistemdeki her aktivitenin bir kaydı tutularak ne zaman kim tarafından yapıldığı kaydedilmelidir.
- Firma sahibi ve birkaç alıřanı sisteme yetkili olarak girerek, yeni ara tanımlama, mevcut araları kaldırma veya fiyat/bilgi gncelleme gibi iřlemleri yapabilmelidir.
- Mřteri proje boyunca srelerin iinde olmak ve teslim e kadar geri bildirim yaparak istediėi rnn ıkması konusunda riski dřrmek istemektedir.
- rnn nihai halinin 6 ay iinde sonlanması beklenmekte ancak bazı fonksiyonları eksikte olsa henz lansman yapılmasa da sitenin temel zellikler ile alıřması ve hayata geirilmesi istenmektedir.

řirketiniz ekip Lideri olarak projeye sizin liderlik etmenizi ve 5 kiřilik bir ekip kurarak bařlamınızı istemiřtir.

Beklentiler

1. Geliřtirilecek sistem iin fonksiyonel ve Teknik gereksinimler listesini oluřturun. Listedeki her bir gereksinimi kısaca aıklayın
2. Kalite niteliklerinden performans, sreklilik ve gvenlik iin senaryolar oluřturun. Her bir kalite niteliėi iin senaryonuzda en az 3 madde bulunmasına dikkat edin
3. Bu sisteme en uygun olacak yazılım mimari stil 'ini sein ve gereklerinizi aıklayın. Mimari stilinizin diyagramını nemli modlleri gsterecek řekilde izin
4. Ekibiniz ile istenen projeyi yapmak iin hangi yazılım geliřtirme srecini uyguladığınızı gerekleri ile aıklayarak srelerin her ařamasında neler yapacağınızı kısaca yazın
5. Ařaėıdaki tasarım desenlerini projede uygun yerlere uygulayın. Her bir desenin uygulandığı alanı, uyguladığını sınıfları yazın ve ne řekilde uyguladığınız belirtin. Bu desenleri uyguladığınız sınıfların kodunu Java ile gerekleyin. Gereklediğiniz kodlar hatasız derlenmiř alıřan kodlar olmalıdır. Raporunuzda kodlar hari uygulandığı sınıflar, aıklaması ne amala kullandığınızı aıklamalısınız. Kodları ile ayrıca geliřtirip derleyip alıřtırmalı ve kaynak kodları ayrı bir "zip" dosyası ile teslim etmelisiniz

- Builder

- Factory
- Bridge
- Proxy
- Decorator
- Iterator
- Strategy
- Visitor
- Observer

Değerlendirme

No	Açıklama	Değerlendirme Notu	Not
1	Teknik ve fonksiyonel gereksinimlerin açıklaması, beklentileri karşılaması	15	
2	İstenen kalite nitelikleri senaryolarının tanımlanması ve örneklendirilmesi	15	
3	Yazılım mimari stilinin seçilmesi, açıklanması ve uygunluğu	15	
4	Yazılım geliştirme sürecinin uygun seçilmesi ve uygulanması	15	
5	Tasarım desenlerinin kullanımları, uygunluğu ve çalıştırılabilir kodların yazılması	40	

Notlar:

- Proje raporu Word formatında ve java kodları ile birlikte tek bir “zip” dosyası olarak (ogrencino.zip) teslim edilecektir. El ile yazılan diyagram, şemalar kabul edilmeyecek bilgisayar araç ve çıktıları kullanılacaktır. Projedeki kodlar sistemin tamamını içeren kodlar değil, desenlerin uygulandığı örnek kodlar olarak yazılabilir, desenlerin sadece kendileri değil, kullanıldığı örneklerde yazılmalıdır.
- Teslimler “ALMS” üzerinden yapılacak, rapor dosyası ayrıca çıktı alınarak elden imzalı teslim edilmek üzere hazır edilecektir. Raporun kapak sayfası sonrası ilk 2 sayfası bu ödev konusu yer alacak şekilde eklenmelidir.
- Son teslim tarihi 24 Mayıs saat 17:00.
- Projeyi, tek, iki veya en fazla 3 kişilik ekipler ile yapabilirsiniz.
- Kopya ödevler, hazır arama sonuçları ve yapay zekâ çıktıları kabul edilmeyecektir. Ödevler arasında yüksek benzerliklerde benzer ödevler kabul edilmeyecektir.

Fonksiyonel Gereksinimler

- Araç Listeleme ve Fiyat Görüntüleme: Sistem, havuzdaki farklı sınıf ve modellerdeki araçları listelemeli ve fiyatlarını görüntülemelidir.
- Rezervasyon ve Kiralama İşlemleri: Müşteri, rezervasyon yapabilmeli veya doğrudan aracı kiralayabilmelidir.
- Doğrulama ve Ödeme: Rezervasyon için iletilen telefon numarasına gelen mesaj ile müşteri kimliği doğrulanmalıdır. Doğrudan kiralamada ödeme kredi kartı ile yapılmalıdır.
- Kesinti Uyarıları ve Performans: Sistem 7/24 çalışmalı, kesintiler ilgili kişilere uyarı olarak iletilmelidir. Ayrıca sistem, müşteri deneyimi ile rakiplerdekinden daha performanslı olmalıdır.
- Güvenlik ve İzleme: Sistem çok güvenli olmalı, kişisel bilgiler kanun ve yönetmeliklere uygun şekilde tutulmalıdır. Hacker saldırılarına karşı dayanıklı olmalı ve her aktivitenin kaydı tutulmalıdır.
- Müşteri İletişimi ve Geri Bildirim: Müşterinin proje sürecine dahil edilmesi ve geri bildirimleri alınmalıdır. Teslimata kadar müşteri ile aktif iletişim halinde olunması ve isteklerinin dikkate alınmalıdır.

TEKNİK GEREKSİNİMLER

- Modülerlik ve Genişletilebilirlik: Var olan modüller daraltılabilir veya genişletilebilir olmalıdır.
- Ölçeklenebilirlik: Sistem ölçeklenebilir olmalıdır.
- Veritabanı Değişikliklerine Direnç: Yaşanabilecek veritabanı değişikliğinden az etkilenmelidir.
- Yeniden Düzenleme Kolaylığı: Sistem kolayca yeniden düzenlenebilmelidir.
- Web ve Mobil Uyumluluk: Hem tarayıcı üzerinden hem de mobil uygulama üzerinden erişim sağlanabilmelidir. Site ve uygulamanın farklı cihazlarda uyumlu olmalıdır.
- Programlama Dilleri ve Teknolojiler: Uygun programlama dillerinin ve teknolojilerinin seçilmelidir. (örneğin, Python, JavaScript, React Native vb.). İhtiyaç duyulan özel kütüphanelerin ve çerçevelerin kullanılmalıdır.
- Yedekleme ve Veri Kurtarma: Veri yedekleme ve kurtarma süreçlerinin otomatik ve düzenli olarak yapılmalıdır. Veri kaybı durumunda hızlı bir şekilde geri yükleme yapılabilirdir.

- Güvenlik ve Saldırı Önleme: Güvenlik duvarları, SSL sertifikaları gibi güvenlik önlemlerinin alınmalıdır. Sistemdeki her aktivitenin loglanması ve denetim izinin sağlanmalıdır.

1. Performans Senaryoları:

Senaryo 1: Yüksek Trafik Altında Hızlı Cevap Süresi:

Senaryo: Eş zamanlı olarak birçok kullanıcının uygulamaya eriştiği bir durumda, araç listeleme ve fiyat görüntüleme işlemlerinin hızlı bir şekilde gerçekleşmesi gerekmektedir.

Performans Hedefi: İşlem süresi 1 saniyenin altında olmalıdır.

Test Planı: Yük testleri ile bu senaryo simüle edilir ve cevap süreleri ölçülür.

Senaryo 2: Rezervasyon İşlemi Performansı:

Senaryo: Kullanıcıların rezervasyon yaparken hızlı ve kesintisiz bir deneyim yaşamaları gerekmektedir.

Performans Hedefi: Rezervasyon işlemi 3 adımda tamamlanmalıdır.

Test Planı: Rezervasyon işlemi simüle edilir ve süreç adımları izlenir.

Senaryo 3: Doğrudan Kiralama İşlemi Performansı:

Senaryo: Kullanıcıların doğrudan araç kiralarken hızlı ve güvenli bir şekilde ödeme yapması gerekmektedir.

Performans Hedefi: Ödeme işlemi 2 dakikadan kısa sürede tamamlanmalıdır.

Test Planı: Ödeme işlemi simüle edilir ve süreç süreleri ölçülür.

2. Süreklilik Senaryoları:

Senaryo 1: Kesintisiz 7/24 Çalışma:

Senaryo: Sistem herhangi bir kesinti olmadan 7/24 çalışmalıdır.

Süreklilik Hedefi: Yılda en fazla 1 saat kesinti süresi olmalıdır.

Test Planı: Yük testleri ve yedekleme senaryoları ile sistemin sürekli çalışma durumu test edilir.

Senaryo 2: Kesinti Uyarıları:

Senaryo: Sistemde bir kesinti olduğunda ilgili kişilere anında uyarı gönderilmelidir.

Süreklilik Hedefi: Kesinti anında uyarılar 5 dakika içinde ilgili kişilere iletilmelidir.

Test Planı: Kesinti senaryoları ile uyarı mekanizması test edilir.

Senaryo 3: Güncelleme Sürekliliği:

Senaryo: Sistem güncellemeleri yapılırken kullanıcılar etkilenmemeli ve kesinti yaşamamalıdır.

Süreklilik Hedefi: Güncelleme süreci 30 dakikadan kısa sürmelidir.

Test Planı: Güncelleme senaryoları ile süreç test edilir.

3. Güvenlik Senaryoları:

Senaryo 1: Veri Sızıntısı Önleme:

Senaryo: Hassas verilerin yetkisiz erişimini önlemek için güvenlik duvarları, şifreleme ve erişim kontrolleri uygulanmalıdır.

Güvenlik Hedefi: Veri sızıntısı riski en aza indirilmelidir.

Test Planı: Veri güvenliği senaryoları simüle edilir ve veri sızıntısı riskleri değerlendirilir.

Senaryo 2: Kimlik Doğrulama ve Yetkilendirme:

Senaryo: Kullanıcıların kimlik doğrulama ve yetkilendirme süreçleri güvenli bir şekilde gerçekleştirilmelidir.

Güvenlik Hedefi: Yetkisiz erişim riski en aza indirilmelidir.

Test Planı: Kimlik doğrulama ve yetkilendirme senaryoları simüle edilir ve güvenlik açıkları değerlendirilir.

Senaryo 3: Saldırı Tespit ve İzleme:

Senaryo: Sistemdeki anormal aktiviteleri tespit etmek ve saldırıları izlemek için güvenlik olay günlükleri ve izleme araçları kullanılmalıdır.

Güvenlik Hedefi: Saldırıları hızlıca tespit edilmeli ve önlenmelidir.

Test Planı: Saldırı senaryoları simüle edilir ve izleme mekanizmaları test edilir.

MİKRO SERVİS MİMARİ STİL

Mikroservis mimari stili, bu projenin gereksinimlerini karşılamak için uygun bir seçenektir. İşte nedenleri:

Modülerlik ve Bağımsızlık: Mikroservisler, uygulamayı bağımsız hizmetlere böler. Her mikroservis belirli bir işlevi yerine getirir ve kendi veritabanına sahiptir. Bu, yeni özellikler eklemeyi ve mevcut hizmetleri güncellemeyi kolaylaştırır.

Ölçeklenebilirlik: Mikroservisler, yüksek trafikli sistemlerde ölçeklendirilebilir. Her mikroservisi ayrı ayrı ölçeklendirebilir ve gereksinimlere göre yeni mikroservisler ekleyebilirsiniz.

Hızlı Dağıtım ve Güncelleme: Her mikroservisi bağımsız olarak dağıtabilir ve güncelleyebilirsiniz. Bu, hızlı iterasyonlar ve hızlı teslimat sağlar.

Güvenlik ve İzleme: Her mikroservis, kendi güvenlik önlemlerine sahip olduğu için güvenlik daha iyi yönetilir. Ayrıca, her mikroservisin aktivitelerini izlemek için ayrı kayıtlar tutabilirsiniz.

Diyagram: Aşağıda, önemli modülleri gösteren bir mikroservis mimari diyagramı bulunmaktadır.

Bu mimari, projemizin gereksinimlerini karşılamak için esneklik, ölçeklenebilirlik ve güvenlik sağlayacaktır.



AGİLE YAZILIM DÖNGÜSÜ

Agile Yazılım Geliştirme Döngüsü, bu projeyi yönetmek için iyi bir seçenektir. Agile, esneklik, işbirliği ve sürekli geri bildirim gibi temel prensipleri benimseyen bir yaklaşımdır.

Değişen gereksinimlere hızlı adapte olmanızı sağlar ve kullanıcıların geri bildirimlerine etkili cevaplar vermenizi kolaylaştırır.

Esneklik ve İterasyonlar: Agile, esnek ve iteratif bir yaklaşım sunar. Proje gereksinimleri ve öncelikleri değişebilir, bu nedenle esneklik önemlidir. İterasyonlar sayesinde, projeyi küçük parçalara bölebilir ve her birini ayrı ayrı geliştirebilirsiniz.

Sürekli Geri Bildirim: Müşteri proje boyunca süreçlerin içinde olmak istediğinden, Agile yöntemi sürekli geri bildirim almanızı sağlar. Bu, istenen ürünün doğru şekilde geliştirilmesini ve risklerin azaltılmasını sağlar.

Hızlı Teslimat: Agile, hızlı teslimatı teşvik eder. Her iterasyon sonunda işlevsel bir ürün parçası sunabiliriz. Bu, müşterinin temel özellikleri erken aşamada görmesini sağlar.

Ekip İşbirliği: Agile, ekip üyeleri arasında sıkı işbirliği yapmayı teşvik eder.

1. Gereksinim Analizi: Senaryoyu dikkatli bir şekilde okuyarak analiz ettik. Fonksiyonel ve teknik gereksinimleri belirledik.

2. Tasarım Aşaması: Sistem mimarisini tasarladık (örneğin, mikroservis mimarisi, katmanlı mimari). Kullanıcı arayüzünü tasarladık.

3. Geliştirme: Backend ve frontend geliştirme işlemlerini başlattık. Mikroservisler veya katmanlar arası iletişimi sağladık. Güvenlik önlemlerini uyguladık (örneğin, kimlik doğrulama, yetkilendirme).

4. Test Aşaması: Birim testleri ve entegrasyon testlerini yaptık. Performans testleri gerçekleştirdik. Güvenlik testleri yaptık.

5. Dağıtım ve Yayınlama: Mikroservisleri veya katmanları ayrı ayrı dağıttık. Sunucuları yapılandırdık ve uygulamayı canlıya aldık.

6. İzleme ve Bakım: Sistem performansını izledik ve kesintileri tespit ettik. Güvenlik güncellemelerini takip ettik. Müşteri geri bildirimlerini değerlendirdik ve gerekli düzeltmeleri yaptık.

DESENLER VE KULLANILAN SINIFLAR

Builder desenini müşteri sınıfında kullandık. Burda kullanım amacımız nesneler oluştururken yapıcı metotlara çok sayıda parametre ,özellik ekleyebilmek. Aynı zamanda kodun okunabilirliğini arttırmaya çalıştık. Parametrelerin sırasını veya zorunluluğunu dikkate almadan nesne oluşturma imkanı verir ve nesne oluşturma işlemlerini adım adım yapılandırmamızı sağlar.

Factory desenini Saatlik Kiralama, Günlük Kiralama, Aylık kiralama Kiralama ve KiralamaTuruFac adlı sınıflarda kullandık. Bağımlılıkları azaltmak ve nesne oluşumunun kontrolünü sağlamayı amaçladık. Aynı amaçtan farklı özelliklere sahip nesneler oluşturuldu. Ayrıca, yeni kiralama türleri eklemek gerektiğinde mevcut kodu değiştirmeden kolayca genişletebilme imkanı elde ettik. Desen, kodun esnekliğini artırarak farklı kiralama türlerinin bağımsız olarak ele alınabilmesine olanak tanıdı.

Bridge desenini Bridge İşlem, Bridge İşlem2, Bridge Kiralama, Bridge Rezervasyon sınıflarında kullandık. Bu desen ile rezervasyon ve kiralama arasında bir köprü oluşturduk. Doğrudan nesne erişimi yerine köprü ile erişim sağlandı. Bu sayede,

soyutlama ile implementasyon arasında güçlü bir ayrım yapılmış oldu. Bridge deseni, kodun esnekliğini artırarak farklı implementasyonların bağımsız olarak geliştirilebilmesine ve değiştirilebilmesine olanak tanıdı. Rezervasyon ve kiralama işlemleri arasında ortak bir arayüz oluşturularak, sistemdeki karmaşıklık azaltıldı ve kodun daha modüler hale gelmesi sağlandı.

Decarator desenini Decarator Arac, DecaratorAracInterface , SporAraba, LuksAraba ve KlasikAraba sınıflarında kullandık. Bu deseni kullanmamızdaki amac koda esnek ve dinamik bir yapı kazandırmak. Açık kapalı prensibi ile sınıfları değiştirmeden yeni özellikler ekleme imkanı sağlar.

Iterator desenini Iterator ve AracYönetimi sınıfında kullandık. AracYönetiminde listeden arac kaldırma metoduna entegre ettik. Burada iterator deseni olmasaydı döngü her adımda listenin boyutunu kontrol edicekti ve bir öge kaldırıldığında boyut değişecekti. Iterator deseni burada listenin yapısını ve ögeler üzerinde güvenli işlem yapmamızı sağlıyor.

Proxy desenini MudurErisim, IsletmeSahibiErişim, Erişim , Proxy Erişim, Personel sınıflarında kullandık. Sisteme erişim kontrollerini denetlemek amaçlandı. Gerçek nesne yerine proxy nesnesi koyulur. Nesneye erişim kontrol edilir veya değiştirmek için kullanılır.Bu sayede güvenlik kontrolleri yapılır. Yetkilendirme ve kimlik doğrulama işlemlerini merkezi bir noktada toplanır. Ayrıca, sistem performansını iyileştirmek için nesneye gereksiz erişimlerin de önüne geçmiş olur. Proxy nesneleri, gerçek nesneye yalnızca gerekli olduğunda erişim sağlar ve bu da sistem kaynaklarının verimli kullanılmasına yardımcı olur.

Strategy desenini ArabaFiyatlandırmaStrategy, KrediKartıStrategy , HavaleStrategy, Odeme sınıflarında kullandık. Odeme için iki farklı görev için iki farklı algoritmasını bu desenle yaptık. Bu kodun modülerliğini artırdı ve her sınıfın yalnızca bir alitmadan sorumlu olması karmaşıklığında önüne geçmiş oldu. Desen farklı alitmaların sistemin kararına göre dinamik olarak seçilmesini sağlar.

Visitor deseni Visitor ,KrediKartıStrategy, ConcreteVisit, Odeme Kontrol sınıflarında kullandık. Bu deseni kullanmamızdaki amaç müşterinin ödemesini doğrulamaktır. Burada işlem mantığını ayırmış olduk. Concrete Visit sınıfı ödeme doğrulama işlemlerini genişletmek için alt sınıflar oluşturabilir bu da koda modülerlik sağlar. visit metodu ziyaretçi nesnesini visit metodunda çağırıyor ve mevcut nesne ziyaretçiye geçiyor. Ziyaretçi nesnenin işlenmesi için ziyaretçi nesnesi kabul edilir.

Observer desenini Observer, ObserverInterface, Subject, SubjectInterface, MessageStream sınıflarında kullandık. Bu desen nesnenin durumunu bildirir. Nesne de herhangi bir değişiklik olduğunda bize bildirim gönderir. Başka bir nesnenin durumunu

izleyen nesne gözlemci, izlenen nesneyede subject denir. Izleyen ve izlenen nesneleri sınıflardan anlayabiliriz. Sistemimizde her hangi bir kesinti yaşanması durumunda bize bildirim gönderilecektir. Bu da kodumuzun 7/24 saat aktif çalışmasını sağlayacaktır. Kodumuza süreklilik kazandırmış olur.

Main sınıfında ise tanımladığımız işlemleri menüleştirdik.

MAIN ÇIKTILARI

```
Menü:
1. Tüm araçları listele
2. Yeni araç ekle
3. Decorator araçları göster
4. Rezervasyon yap
5. Tüm rezervasyonları listele
6. Müşteri oluştur
7. Ödeme yap
8. Çıkış
Bir seçenek girin: 2
Araç ID'si girin: 9
Marka girin: volkswagen
Model girin: tiguan
Vites türü girin: otomatik
Renk girin: beyaz
Yakıt türü girin: dizel
Yıl girin: 2020
Kilometre girin: 19000
Müsait mi (true/false): true
Araç başarıyla eklendi.

Menü:
1. Tüm araçları listele
2. Yeni araç ekle
3. Decorator araçları göster
4. Rezervasyon yap
5. Tüm rezervasyonları listele
6. Müşteri oluştur
7. Ödeme yap
8. Çıkış
Bir seçenek girin: 3
Decorator Araç Özellikleri:
Klasik araç.
Klasik araç.
Lüks araç özellikleri ekleniyor.
Klasik araç.
Spor araç özellikleri ekleniyor.
```

```
Menü:
1. Tüm araçları listele
2. Yeni araç ekle
3. Decorator araçları göster
4. Rezervasyon yap
5. Tüm rezervasyonları listele
6. Müşteri oluştur
7. Ödeme yap
8. Çıkış
Bir seçenek girin: 4
Rezervasyon yapmak için araç ID'sini girin: 9
Rezervasyon tarihi (yyyy-MM-dd): 24-05-2024
İade tarihi (yyyy-MM-dd): 30-05-2024
Geçersiz araç ID'si veya araç müsait değil!

Menü:
1. Tüm araçları listele
2. Yeni araç ekle
3. Decorator araçları göster
4. Rezervasyon yap
5. Tüm rezervasyonları listele
6. Müşteri oluştur
7. Ödeme yap
8. Çıkış
Bir seçenek girin: 6
Ad girin: Suat
Soyad girin: Ugurlu
Araç plaka numarası girin: 34 İST 34
Kiralama türü girin: günlük
Adres girin: İstanbul
Telefon numarası girin: 0544 444 22 33
E-posta girin: suatugurlu@topkapi.edu.tr
```

Menü:

1. Tüm araçları listele
2. Yeni araç ekle
3. Decorator araçları göster
4. Rezervasyon yap
5. Tüm rezervasyonları listele
6. Müşteri oluştur
7. Ödeme yap
8. Çıkış

Bir seçenek girin: 7

Ödeme yöntemi seçin:

1. Kredi Kartı
2. Havale

1

Kart üzerindeki isim: suat

Kart numarası: 5165

CW: 156

Son kullanma tarihi (AA/YY): 10/28

Tutar: 1500

İsminizi giriniz: suat

Kart numarası: 5165

Son kullanma tarihi: 1500.0

Ödeme işlemi tamamlandı.

1500.0TL kredi kartı ile ödendi

Menü:

1. Tüm araçları listele
2. Yeni araç ekle
3. Decorator araçları göster
4. Rezervasyon yap
5. Tüm rezervasyonları listele
6. Müşteri oluştur
7. Ödeme yap
8. Çıkış

Bir seçenek girin: 7

Ödeme yöntemi seçin:

1. Kredi Kartı
2. Havale

2

Banka adı: ziraat

Hesap numarası: 5464646

IBAN: 55454664

Tutar: 1500

Havale yapılan banka: ziraat

Hesap numarası: 5464646

Ödeme miktarı: 1500.0

Havale işlemi tamamlandı.

1500.0TL havale ile ödendi

Menü:

1. Tüm araçları listele
2. Yeni araç ekle
3. Decorator araçları göster
4. Rezervasyon yap
5. Tüm rezervasyonları listele
6. Müşteri oluştur
7. Ödeme yap
8. Çıkış

Bir seçenek girin: 8

Çıkış yapılıyor...