**İnönü Üniversitesi**

**Bilgisayar Mühendisliği Bölümü**

**2019 – 2020 Nesneye Yönelik Yazılım Mühendisliği Dersi**

**Proje Analiz Rapor**

|  |  |
| --- | --- |
| **Ders Sorumlusu** | **Dr. Öğr. Üyesi Ahmet Arif AYDIN** |
| **Öğrenci(ler)** | **02160001016 / Ayşe ÖZAVCİ** |
| **Proje İsmi** | **Otobüs Bileti Rezervasyon Sistemi** |
| **Aşama Tanımı** | **Deployment**  **Projenin uygulamasının nasıl gerçekleştirdiğini açıklayınız. Kaç kişi geliştirdiğiniz projeyi kullandı ve bu ortamı nasıl sağladınız? Projenizi kullanıcı erişimine açmak için hangi teknolojileri kullandınız?**  **Seçilen behavioral (davranışsal) tasarım kalıbının kodunun yazılması** |
| **AÇIKLAMA** | |
| *AŞAMA 1*  **Projenin uygulanması nasıl gerçekleşti ?**   * Deployment aşamasında projede eksiklikleri tamamlayıp, yanlışları düzeltip kullanıcıların erişimine sunmam gerekiyordu.Ben de hatalarımı düzeltip, gördüğüm eksiklikleri düzeltip projemi github adresime ekleyip bu şekilde erişime açtım.   **Kaç kişi geliştirdiğim projeyi kullandı, bu erişimi nasıl sağladım ?**   * Projemi github adresime koyarak tüm github kullanıcılarının erişimine açtım.Ayrıca kodlarımın kolay anlaşılması için yazdığım raporları da ekledim.      * Ayrıca evde aileme projemin arayüzü üzerinden işlemler yapmalarını sağladım. (4 kişi) * Son olarak yazılımcı arkadaşlarıma da projemi gönderip denemelerini ve projem hakkındaki görüşlerini ve tavsiyeleri aldım. (5 kişi)   Tüm bu aşamalardan sonra tavsiye ve görüşlere göre projemde küçük çaplı değişiklikler yaptım. (Arayüz gibi)  Projemin linki : <https://github.com/muhendisaysee/OtobusBiletRezervasyonSistemi>  **Projemi kullanıcı erişimine açmak için hangi teknolojileri kullandım?**     * Github adresimi kullandım, ayrıca whatsapp aracılığıyla arkadaşlarımdan tavsiye ve görüşlerini aldım.   *AŞAMA 2*  ***COMMAND TASARIM KALIBI***  Command tasarım kalıbı, İsteklerin nesne olarak kapsüllenmesini ve böylece farklı istek setlerinin parametreler yardımıyla alınmasını sağlar.  Araştırmalarım sonucunda command tasarım kalıbıyla ilgili örnekler buldum.Bu örneklerde metotların direk değilde nesneler şeklinde kapsüllenerek gerçekleştirildiğini gördüm.Işığın kapanıp , açılmasıyla ilgili örnekler gördüm.( <https://www.geeksforgeeks.org/command-pattern/>’ deki gibi)  *Command tasarım kalıbını kullanıcı giriş çıkış işleminde kullandım.Command tasarım kalıbı, metot sayısı arttığında, bir aracı vasıtasıyla çağırmak istediğimizde veya uzaktaki bir metodu çağırmak istediğimde (RPC) bana avantaj sağlar.*  Bunun için 4 önemli kısım var ;   1. Emri alacak nesne olmalı. 2. Her emrin ortak bir arayüzü olmalı. 3. Her emrin alıcısı belli olmalı. 4. Emri iletmesi için bir nesne olmalı ve bu nesne emir dışında bir şey yapmamalı.   Buna göre giriş- çıkış işlemleri için öncelikle bir arayüz(interface) tasarladım.Bu sayede implements edilen tüm sınıflarda giris() metodunun yazılmasını zorunlu kıldım.Çünkü istemciden bir istek varsa bu istek ya giriş işlemi içindir veya çıkış işlemi içindir.    Bu arayüzü giriş ve çıkış isteklerinin alındığı 2 sınıfa implement’e ettim ;   * **GirisOnCommand**   Giriş isteklerinin alındığı sınıftır.Kullanıcı eğer giriş yapmak isterse bu sınıfa istek gönderir ve giriş sınıfında(tüm giriş-çıkış metotlarının yer aldığı sınıf) girisOn metodu çağrılarak giriş için istek oluşturulur;     * **GirisOffCommand**   Çıkış isteklerinin alındığı sınıftır.Kullanıcı eğer çıkış yapmak isterse bu sınıfa istek gönderir ve çıkış sınıfında(tüm giriş-çıkış metotlarının yer aldığı sınıf) girisOff metodu çağrılarak çıkış için istek oluşturulur.Giristen nesne oluşturdum, çünkü o sınıfa erişmem gerekli;     * **Giris**     Giris sınıfım ise giriş çıkış işlemlerinin gerçekleştiren metotların yer aldığı sınıf tüm işlemler bu sınıfta gerçekleşir.Eğer satıcı giriş yapmak istediyse ve şifresi doğruysa kullanıcı ismi ile beraber(Sat|Müsteri) bu sınıfa yönlendirilir ve satıcı giriş yapar, eğer müşteri giriş yapmak istiyorsa müşterinin şifresi kontrol edilir ve şifresi doğruysa bu sınıfa yönlendirilir, müşteri giriş yapar.      Son olarak kullanıcının butona basıp basmadığını kontrol eden KontrolGirisCommand sınıfında butona basıp basılmadığı kontrol ediyorum(Giriş|Çıkış), butona basıldığı an hangisi için istek gelmişse(Giriş veya Çıkış) o işlemi gerçekleştirmek için ilgili metoda gider.    Giriş yapıldığı anda test etmek için LogIn’in arayüzünde ilgili metodu çağırdım ve giriş başarılı bir şekilde gerçekleşti, aynı şekilde çıkış butonum tüm arayüzler de olduğundan her arayüzün controlunda çıkış butonu için commandla çağırdım ;    MüsteriGirisController metodumda ise;  **DÜZELTMELER**  Geri bildirimde facade tasarım kalıbının eksik olduğu ve anlaşılır olmadığı belirtilmişti, bu nedenle Daha fazla class’ı da facade de kullandım.Ayrıca daha da anlaşılır olması için yorum satırlarını arttırdım.Okunabilirlik konusunda ise drive’a yüklediğimde resimler her biri başka tarafa kayıyor, onu düzeltmek için bile yarım günümü harcıyorum, yazının net olmaması da drive’dan kaynaklı, her bir aşamayı titizlikle hazırlamaya çalışıyorum. Güncel facade sınıfım ;  İlgili metotları facadeServer sınıfımda yazdım, güncel facadeServer sınıfım ; | |