

TECHNO STUDY Agile Software Testing Project - OpenMRS TR Sprint

Bir Haftalık Sprint

****URL:**** [OpenMRS Project](https://demo.openmrs.org/)

Proje Açıklaması:

OpenMRS, açık kaynaklı bir elektronik sağlık kayıt sistemidir. Sağlık hizmeti sunan kuruluşlar için hasta verilerini yönetmek, saklamak ve paylaşmak amacıyla kullanılır. OpenMRS, özellikle düşük gelirli ülkelerde sağlık hizmetlerini geliştirmek amacıyla tasarlanmıştır. Bu platform, tıbbi kayıtları dijitalleştirme, hasta takibi, tedavi planlaması ve sağlık verilerinin analizi gibi önemli işlevleri destekler.

OpenMRS Projesi açık kaynaklıdır ve bir topluluk tarafından geliştirilen bir projedir. Projede yazılım geliştiricileri, sağlık uzmanları ve gönüllüler bir araya gelir ve sağlık hizmetlerinin veri yönetimini iyileştirmek için çalışırlar. OpenMRS, kullanıcı dostu bir arabirim sunar ve farklı sağlık hizmeti sunan kuruluşlar için uyarlanabilir.

Kullanacağımız environment <https://openmrs.org> adresidir. Bu environment'in alt alanı olan <https://demo.openmrs.org/> 'ta çalışmış olacağız.

User Story Özet:

1. **Sisteme Giriş (Login) Hatalarını Kontrol Etmek.**
2. **Kullanıcı, uygulamaya giriş yapabilmeli.**
3. **Kullanıcı, sistemden güvenli bir şekilde çıkış yapabilmeli.**
4. **Kullanıcı, hasta bilgilerini girerek sisteme hasta kaydı yapabilmeli.**
5. **Kullanıcı, hesap ayarlarına (My Account) ulaşabilecek bir arayüze sahip olmalı.**
6. **Doktor, hastaları listeleyebilmeli ve belirli bir hastayı search yapabilmeli.**
7. **Doktor, belirli bir hastayı sistemden silebilmeli.**
8. **Doktor, tüm hastaları listeleyebilmeli.**
9. **Hasta Kayıtlarını Birleştirme (Merge).**
10. **Hasta Randevusu Alırken Yanlış Saat Dilimi.**

Proje Aşamaları ve Görevler:

Requirements:

Bu projede de, Java ve Selenium gibi test otomasyon araçlarını kullanarak web tabanlı bir uygulamanın testlerini yazmaya devam edeceğiz.

We will use the **Page Object Model** in this project and also implement **logging**. (For this, log4j-core would be sufficient.)

Bu projede **Page Object Modeli** kullanacağız ve ayrıca **logging** uygulayacağız. (Bunun için **log4j-core** yeterli olacaktır.)

Bu arada da **TestNG** ile testleri organize etme konusunda deneyim kazanmayı amaçlıyoruz. Bu sayede **XML** vasıtasıyla testleri gruplamak, teste veri sağlamak (**data provider**), öncelikler belirlemek (**priority**) ve test sonuçları için temel **HTML raporları** oluşturmak konusunda deneyim kazanacağız.

Aynı zamanda **JIRA** ve **Zephyr** gibi test yönetimi araçlarıyla test senaryolarımızı oluşturmaya, hata izleme süreçlerini anlamaya devam edeceğiz.

GitHub üzerinde de takım arkadaşlarımızla işbirliği yapmaya devam ediyor olacağız.

Projede ayrıca **SDLC-STLC** döngüleri açısından vazgeçilmezimiz olan **Scrum** ve **Sprint** seremonileri de uygulanacak ve takım üyeleri olarak birlikte sprintler arasındaki görevleri yönetme konusunda deneyimlerimizi artırmaya çalışacağız...

Aşama 1: Test Senaryolarının Oluşturulması

- JIRA kullanılarak test senaryoları oluşturulacak.
- Zephyr veya benzeri bir eklenti ile bu senaryolar test case'lere dönüştürülecek.
- Her senaryo uygun TestNG testleriyle eşleştirilecek.

Aşama 2: Temel Test Otomasyonu

- Java ve Selenium kullanarak <https://openmrs.org> web uygulamasının otomasyon testleri yazılacak.
- TestNG kullanarak testler gruplandırılacak ve gerekliyse öncelik atanacak.
- Test sonuçları **HTML** raporlarıyla dökümanite edilecek.

Aşama 3: Süreç ve İşbirliği Yönetimi

- GitHub'da bir proje oluşturulacak ve bu proje bir ekip olarak yönetilecek.
- Proje görevleri GitHub üzerinde de açılacak, atanacak ve bu görevler üzerinde çalışılacak. **NOT: Normalde, Scrum ve Sprint yönetimi uygulanarak, proje zaman içinde geliştirilir. Dolayısıyla projelerinizde geriye dönük düzeltme ve bakımlar (maintenance) yapabilirsiniz. Öğrendiğiniz yeni şeyleri eski tasklarda deneyebilirsiniz. Ortak olarak gerçekleştirdiğiniz tüm projeleri kendi github sayfanıza da koyabilirsiniz...**

Aşama 4: Hata İzleme ve Hata Raporlama

- Testler sırasında bulunan hatalar JIRA veya benzeri bir araç üzerinde kaydedilerek prosedür doğrultusunda bug ticket açılacak.
- Bug olduğundan emin olmak için grupça değerlendirme yapacağız.
- Eminsek Discord Proje kanalından da bu durumu paylaşacağız.

Readme File:

- Son olarak projeye alakalı bir **Readme** dosyası oluşturarak, GitHub'daki projenizin kök (root) dizinine ekleyiniz. *Bu dosya, proje tanımı, açıklamaları, sisteminizde kullandığınız teknolojileri (JDK, Selenium versiyonları, library vs.), environment ve collaborator'ları (katılımcılar) içerebilir.*

Notlar:

Bahsettiğimiz gibi, OpenMRS açık kaynaklı bir projedir. Bu nedenle, GitHub ve Jira hesaplarında, yazılım geliştirme süreçleriyle ilgili birçok detay bulunmakta ve bundan faydalanabilirsiniz... Pekçok kaynak gibi orada öylece bizim keşfetmenizi bekliyor :) Bu kaynaklara erişmenin en hızlı yolu ise şu bağlantı: <https://openmrs.org/developers/>

- Her hafta öğrendiğimiz yeni konuları projelerimize uygulamaya devam ediyoruz; her yeni sprint ile bu bilgiyi kullanarak yeni projeler geliştiriyoruz.
- Takım içi iletişim ve işbirliği çok önemlidir. GitHub üzerinde işbirliği yaparak projeyi yönetin.
- Her görevin sonunda bir rapor veya belge oluşturarak yaptığınız işi dökümanete edin.
- Bu tarz projeler, test otomasyonu, test senaryoları oluşturma, süreç yönetimi ve işbirliği becerileri kazandırmak için mükemmel bir fırsat sunmaktadır. Başarılar dilerim!

H A P P Y C O D I N G ! ! ! 😊 🛠️

