



SAKARYA
ÜNİVERSİTESİ

Sakarya
Üniversitesi

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

NESNE YÖNELİMLİ
ANALİZ VE TASARIM

Proje Çalışması

Lütfü Orhun İNAN

B201210397

1. Öğretim B Grubu

orhun.inan@ogr.sakarya.edu.tr

NESNELERİN İNTERNETİ SİSTEMLERİ İÇİN AKILLI CİHAZ TASARIMI

Lütfü Orhun İNAN¹
¹B201210397 1.B Grubu

Özet

Ödevin amacı Java dili kullanarak internet üzerinden kontrol edilme özelliğine sahip, ortam sıcaklığını kontrol eden akıllı cihaz üretilmesidir. Çalışma kapsamında istenen yazılımın tasarımı, analizi ve gerçekleştirilmesi istenen kriterler üzerinden, OOP kurallarına uymak üzere tasarlanmıştır. Uygulama sanal veya postgres üzerinden gerçek veritabanına bağlanma, oda simülasyonu ve eşzamanlılık sağlamaktadır.

Anahtar Kelimeler: Java, OOP, SOLID

© 2022 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Her hangi bir kopya işleminde sorumluluk bana aittir.

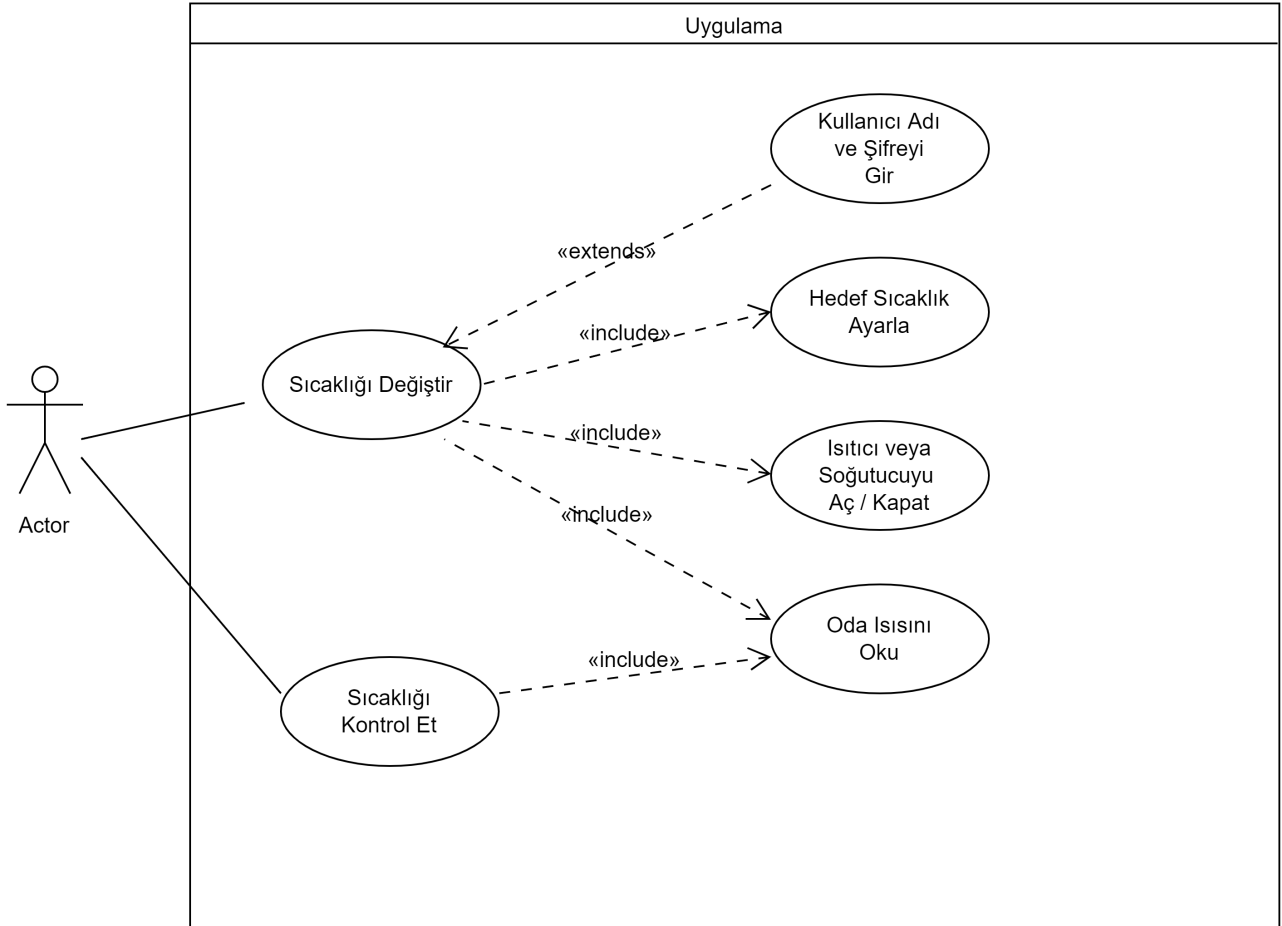
İçindekiler

1. Kullanıcı için Use-Case Diyagramı	2
2. Kullanım durumuna ait metinsel tanımlar	3
3. Kullanım Durumunun Sequence Diyagramı	3
4. Kullanım Durumu için Activity Diyagramı	4
5. Uygulamanın Sınıf Şeması	5
6. İstenen iki sınıfa ait CRC Kartları	6
7. Sistemin State Machine Diyagramı	6
8. Uygulamanın Ekran Görüntüleri	7
9. Veritabanı Görüntüleri	8

10. Açıklamalar

1. Dependency Inversion	
Açıklaması ve Kullanımı	8
2. Factory Method ve Observer	
Açıklaması ve Kullanımı	8
11. Uygulamanın Kaynak Kodu	8
12. Kaynakça	9

Kullanıcı için Use-Case Diyagramı

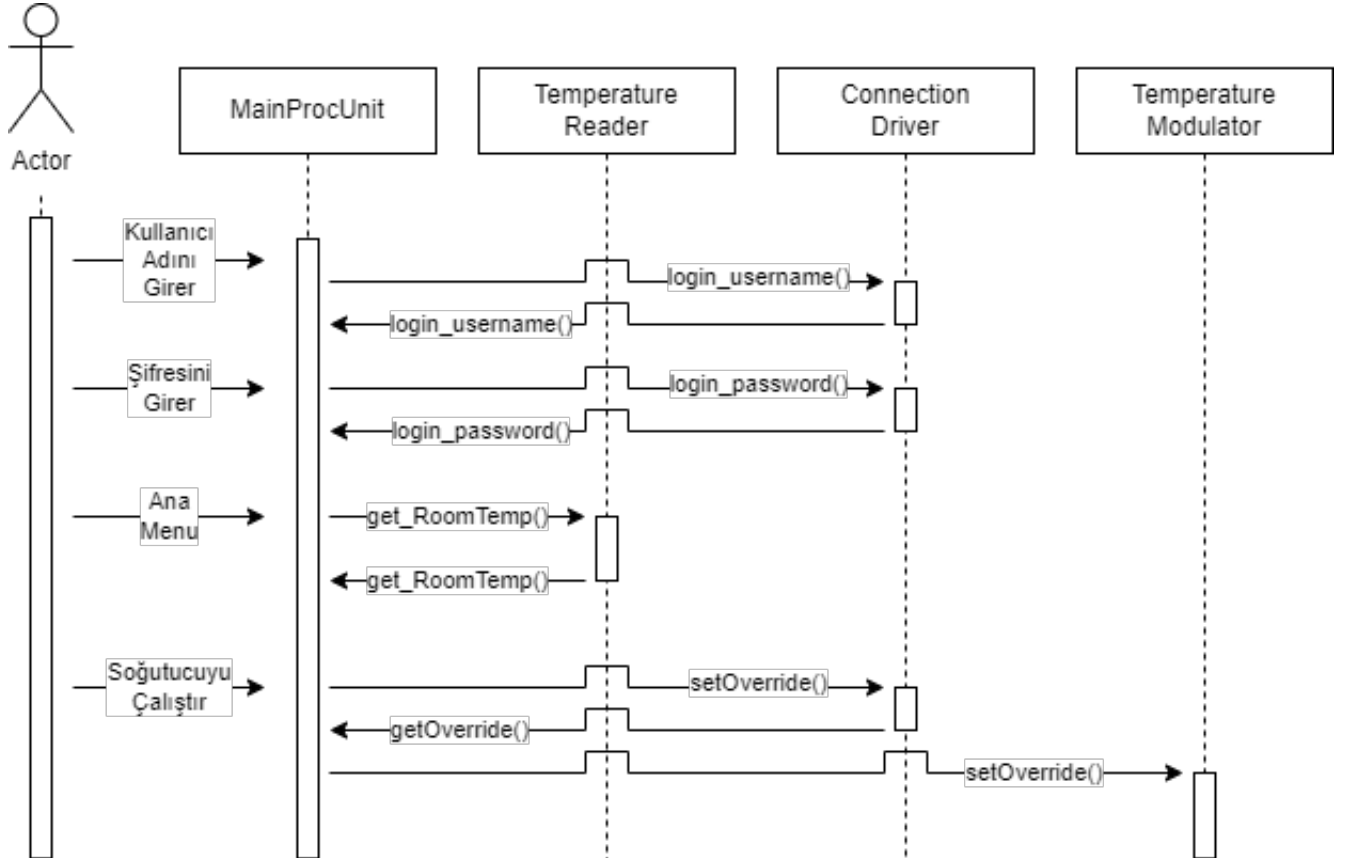


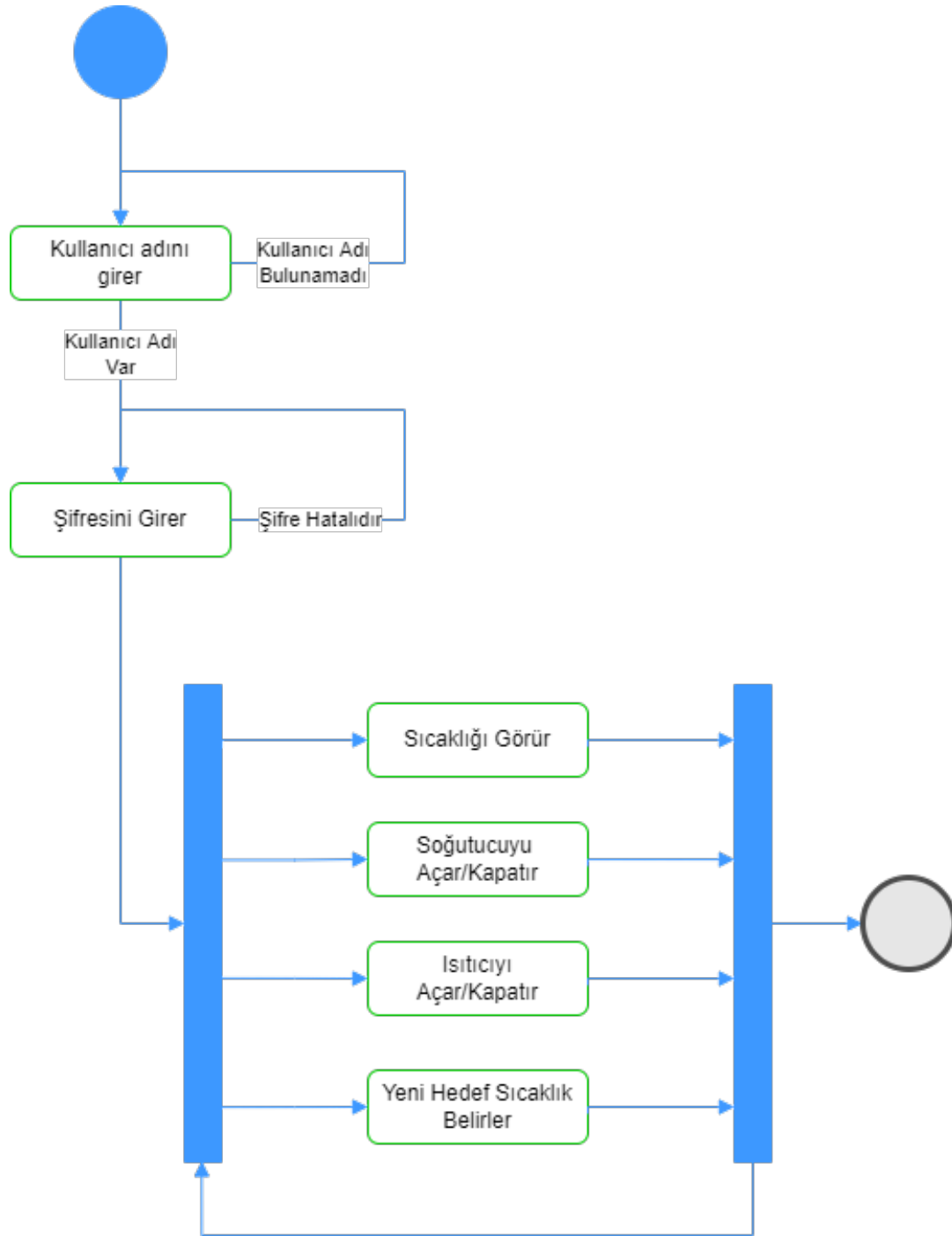
Kullanım durumuna ait metinsel tanımlar

Use Case İsmi	İnternet Aktörünün Uygulama üzerindeki kontrolü
Aktörler	Kullanıcı
Ön Gereklilik	Kullanıcı Adı ve Şifrenin Girilmesi

Ana Olay Akışı (Primary Flow)	Burada kullanıcı oda sıcaklığını ayarlar, Soğutucu veya Isıtıcıyı kontrol eder ve uygulamayı yönetir.
Aktör Adımları	System Cevapları
1. Kullanıcı, Kullanıcı adı ve şifre girer	1.1 Kullanıcı Adı girilince kullanıcı adı doğruluğunu veritabanına göre karar verir. Kullanıcı adı doğruysa şifre menüsüne girilir. 1.2 Şifre girilince kullanıcı adı aynı olan kullanıcıların şifrelerine karşılaştırılır.
2.Sıcaklığı Değiştir	Kullanıcı sıcaklığı değiştirmek için birden fazla seçeneği vardır. İstenen yöntem ana menüden seçilebilir.
3.Hedef Sıcaklık Girilmesi	• Kullanıcı ana menü üzerinden hedef sıcaklık menüsünü seçerek duruma girer • Kullanıcı geçerli sıcaklık girerse girilen sıcaklık değeri veritabanına gönderilir.
4.Sıcaklığı veya Soğutucuyu Aç / Kapat	Kullanıcı ana menü üzerinden ısıtıcıyı veya soğutucuyu açabilir. Kullanıcı herhangi birini açarsa veritabanı içindeki override değeri true haline gelir. Bu Hedef sıcaklık değerine ulaşmayı etkisiz kılar. Eğer kullanıcı hem sıcaklık hem soğutucuyu kapatırsa, yeni sıcaklık girilene kadar override modunda kalır.
5.Sıcaklığı kontrol Et	Oda içerisindeki sıcaklık değeri geri döndürülür.

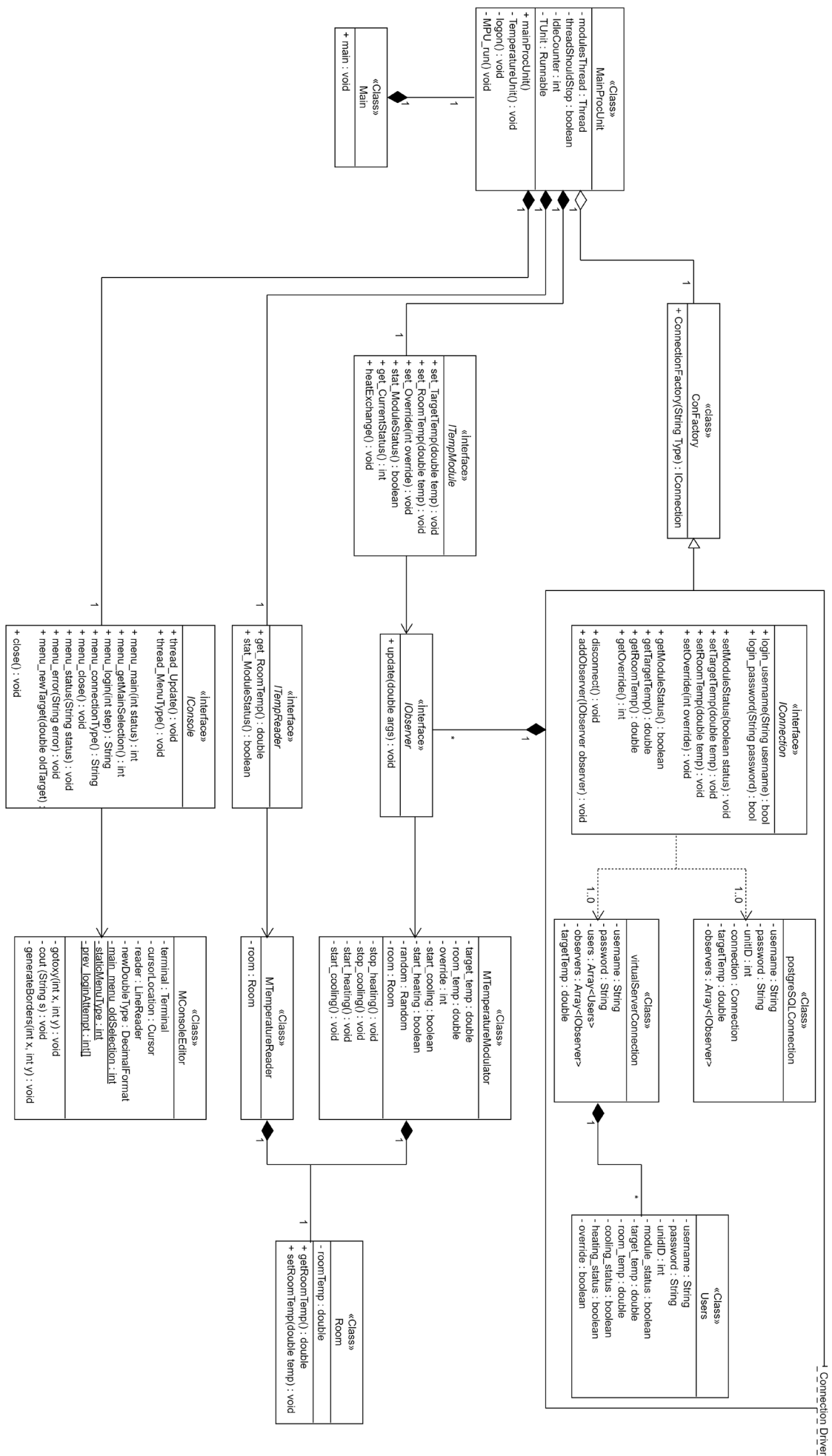
Alternatif Yol	Burada uygulamanın normal dışı durumları gösterilir.
Aktör Adımları	System Cevapları
1. SELF-CHECK Failed	Uygulama eğer içerisindeki kontrol safasında başarısız olursa uyarı menüsü gösterir ve uygulama kapanır.
2. CONNECTION Failed	Uygulama veritabanına bağlantısı başarısız olursa uyarı ekranı gösterildikten sonra uygulama kapanır.

Kullanım Durumunun Sequence Diyagramı

Kullanım Durumu için Activity Diyagramı

Uygulamanın Sınıf Şeması

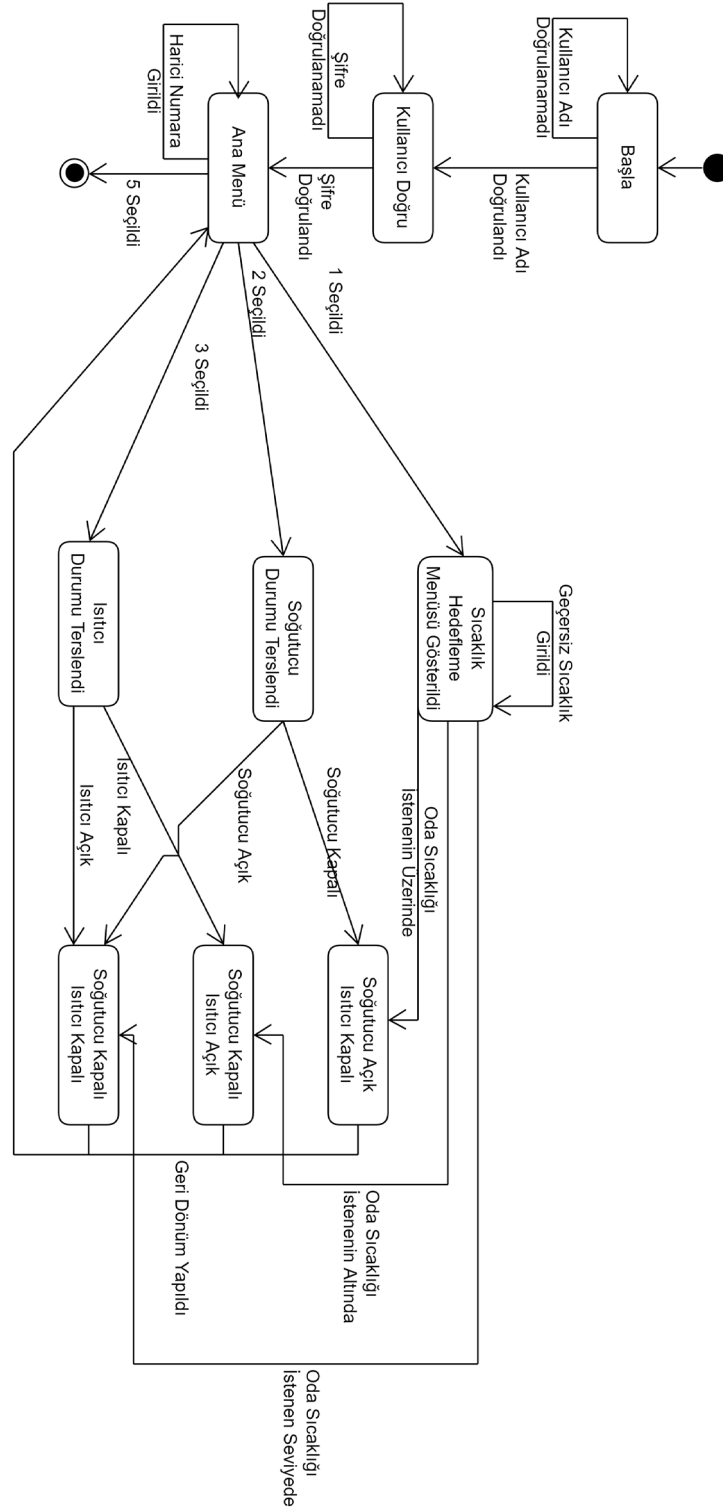
*tam boyutlu diyagramlar GitHub sayfasında diagrams klasöründe bulunabilir.



İstenen iki sınıfa ait CRC Kartları

MainProcUnit	
Ekran Çıktısı Alınması	MConsoleEditor
Sıcaklık Ölçümü	MTemperatureReader
Sıcaklık Değiştirme	MTemperatureModulator
Bağlantı Türü Sağlanması	ConFactory
Veritabanına Bağlanma	IConnection
Modüller arası iletişim	Kendisi

MTemperatureModulator	
Oda Sıcaklığının Alınması	MTemperatureReader
Açılıp Kapanma Durumu	IConnection
Modül Durumunun Gösterimi	MConsoleEditor
Sıcaklık Değiştirme	Kendisi

Sistemin State Machine Diyagramı

Uygulamanın Ekran Görüntüleri

Kullanıcı Adınızı Giriniz:

Şifrenizi Giriniz:

Eski Hedef Sıcaklık: 25,550 °C
Yeni Hedef Sıcaklık:Yeni hedef sıcaklık belirlendiğinde $\pm 0.25^\circ$ derece aralığına kadar ulaşmaya çalışır.

Hedef Sıcaklık: 24,000 °C | Oda Sıcaklığı: 30,160 °C

- 1 - Hedef Sıcaklık Ayarlama
- 2 - Soğutucuyu Kapat
- 3 - Isıtıcıyı Aç
- 5 - Kapat

Seçiminiz:

Soğutucu ve Isıtıcının açılıp kapanması ana menü içerisinde kontrol edilebilir. Isıtı veya soğutucu aynı anda açılması kod tarafından engellenir. Soğutucu veya ısıtıcı çalıştığında menu seçenekleri kapat ile değişir. Eğer soğutucu veya ısıtıcı kapatılırsa, hedeflenen sıcaklık özelliği kapatılır yeni hedef sıcaklık girilene kadar.

Hedef Sıcaklık: 24,000 °C | Oda Sıcaklığı: 28,650 °C

- 1 - Hedef Sıcaklık Ayarlama
- 2 - Soğutucuyu Aç
- 3 - Isıtıcıyı Kapat
- 5 - Kapat

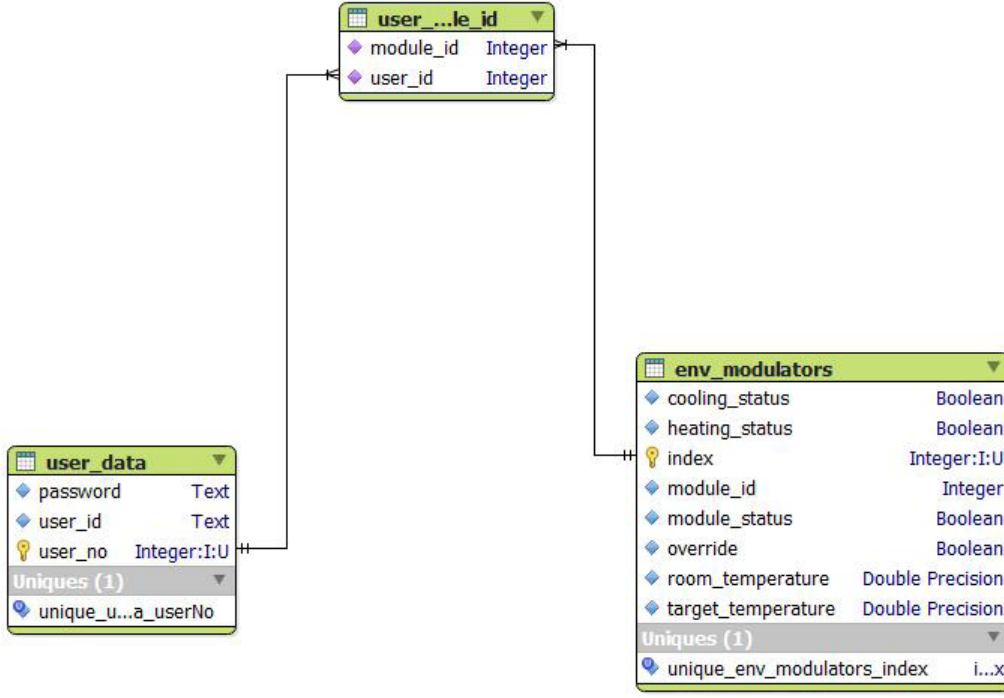
Seçiminiz:

Hedef Sıcaklık: 24,000 °C | Oda Sıcaklığı: 30,170 °C

- 1 - Hedef Sıcaklık Ayarlama
- 2 - Soğutucuyu Aç
- 3 - Isıtıcıyı Aç
- 5 - Kapat

Seçiminiz:

Veritabanı Görüntüleri



	index	module_id	room_temperature	module_status	target_temperature	cooling_status	heating_status	override	user_id	module_id	user_id	password	user_no
1	1	1	30.17	✓	24	✓	✓	✓	1	1	test	test	1
2	2	2	30.06	✓	30	✓	✓	✓	5	2	test2	test2	5

*veritabanı Dump en güncel Dump dosyası GitHub içerisinde bulunabilir.

Dependency Inversion Açıklaması ve Kullanımı

Dependency Inversion ilkesi, SOLID nesne yönelimli programlama ilkelerinden olan, yeniden kullanılabilir ve diğer birimlerden bağımsızlığı sağlayan yüksek derecede ayrılmış kod yazımını öneren programlama paradigmasıdır. Dependency Inversion, Proje içerisindeki; Isı değiştirici, ekran çıktısı alıcısı, ısı okuyucusu, bağlantı sağlayıcı vb bütün modüller Interface yapısı kullanması ile sağlanmıştır.

Factory Method ve Observer Açıklaması ve Kullanımı

Factory Method'u üretilmesi gereken nesne bilinmediğinde, hangi nesnenin oluşturulmasına karar verilmesini sağlar. Bu Proje içerisinde üretilmesi gereken bilinmeyen ve çalışma süresince seçilmesi gereken yapı olduğu için bağlantı driverları için ConFactory sınıfı içerisinde ConnectionFactory methodu kullanıcıdan alınan bağlantı seçeneği seçeneğe göre istenen bağlantı türünü döndürür.

Observer deseni nesneler arasındaki iletişimi belirtir: gözlemlenebilir(Observable) / istemci(Event source) ve gözlemciler(Observer). Bir Observable, durumundaki değişiklikler hakkında Observer'ları bilgilendiren davranışsal desenlerden biridir. Bu proje içerisinde değiştiği durumda diğer yapıları bildirmesi gereken yapı, Bağlantı Veritabanı içerisindeki hedeflenen sıcaklık değişirse, yada Override değeri değişirse ısı değiştirici içerisindeki değerleri update methodu ile değiştirir

Uygulamanın Kaynak Kodu

Uygulamanın kaynak kodu, SQL Dump dosyası, Diyagramlar ve Çalışır jar dosyası aşağıdaki Github repository içinde bulunabilir.

<https://github.com/Dr-Smiles/NYA-Project-Final>

Kaynakça

- Visual Code
Uygulamanın yazıldığı ortam
- Adobe Indesign 2022
Rapor tasarımı ve içeriği
- JLine, JNA, JANSI kütüphaneleri
Uygulamanın GUI tasarımı için
- postgres kütüphanesi
PostgreSQL veritabanı bağlantısını için
- Adobe Illustrator 2020
Kapak Tasarımı