

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**PROJE KONUSU: SAYISAL TASARIM PROJESİ**

**DOĞUŞ HAN ALAN 220502014**

**EMRE ÇOLAK 220501011**

**DERS SORUMLUSU:**

**PROF. DR./DR. ÖĞR. ÜYESİ**

**ERCAN ÖLÇER**

**TARİH:02.06.2024**

1. **GİRİŞ**
   1. **Projenin amacı**

**Bu proje, kullanıcıların basit mantık devrelerini tasarlamasına ve simüle etmesine olanak tanıyan bir platform oluşturmayı amaçlar. Kullanıcılar, temel mantık kapılarını (AND, OR, NOT vb.), giriş/çıkış elemanlarını ve bağlantıları tasarlayabilir ve bu devreleri simüle edebilirler. Böylece, kullanıcılar mantıksal devrelerin nasıl çalıştığını anlayabilir ve farklı girişlerle nasıl tepki verdiklerini gözlemleyebilirler. Bu proje aynı zamanda öğrenme ve eğitim amaçlıdır ve elektronik tasarım konusunda temel becerilerin geliştirilmesine yardımcı olabilir**

1. **GEREKSİNİM ANALİZİ**
   1. **Arayüz gereksinimleri**

Araç Kutusu: Kullanıcıların mantık kapıları, giriş/çıkış elemanları ve bağlantıları ekleyebileceği bir araç kutusu. Tasarım Alanı: Kullanıcıların mantık devresini tasarlayabileceği bir alan. Bu alan, öğelerin sürüklenip bırakılarak yerleştirilebileceği bir arayüze sahip olabilir. Kontrol Düğmeleri: Simülasyonun çalıştırılması, durdurulması veya sıfırlanması gibi işlemleri gerçekleştirmek için kontroller içeren düğmeler. Öğe Seçimi ve Taşıma: Kullanıcıların tasarım alanındaki öğeleri seçmesine ve taşımasına olanak tanıyan işlevselliğe sahip olmalıdır. Bilgi Paneli: Kullanıcının seçili öğe hakkında bilgi alabileceği veya gerekli parametreleri ayarlayabileceği bir bilgi paneli. Görünürlük Kontrolleri: Kullanıcıların öğelerin görünürlüğünü veya durumunu değiştirebileceği kontroller. Hata İletişimi: Kullanıcıya olası hataları veya uygulama durumu hakkında bilgi vermek için bir iletişim kutusu veya durum çubuğu. Bu gereksinimler, kullanıcıların kolayca mantık devrelerini tasarlamasını, simüle etmesini ve anlamasını sağlayacak bir kullanıcı arayüzünü kapsar

* 1. **Fonksiyonel gereksinimler**

**Mantık Kapıları Ekleme: Kullanıcıların araç kutusundan mantık kapılarını seçerek tasarım alanına eklemesi.**

**Giriş/Çıkış Elemanları Ekleme: Kullanıcıların giriş ve çıkış elemanlarını tasarım alanına eklemesi.**

**Bağlantı Ekleme: Kullanıcıların elemanlar arasında bağlantılar oluşturması.**

**Düğüm Ekleme: Kullanıcıların bağlantıları kesiştiren veya ayrılan noktaları belirtmek için düğümleri eklemesi.**

**Öğe Seçme ve Taşıma: Kullanıcıların tasarım alanındaki öğeleri seçmesi ve taşıması. Simülasyon Çalıştırma: Kullanıcıların tasarlanan mantık devresini simüle etmesi.**

**Simülasyon Sıfırlama: Kullanıcıların simülasyon sonuçlarını sıfırlaması.**

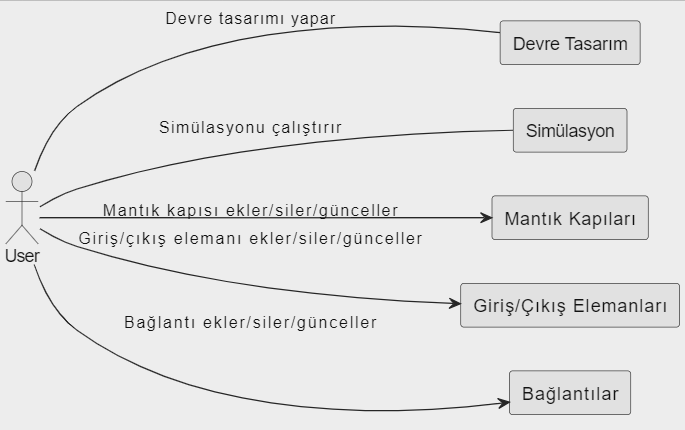
**Simülasyon Durdurma: Kullanıcıların simülasyonu istediği zaman durdurması.**

**Sonuçları Görüntüleme: Kullanıcıların simülasyon sonuçlarını görüntülemesi, grafikler veya veri tabloları şeklinde.**

**Öğe Bilgilerini Görüntüleme: Kullanıcıların seçili öğe hakkında bilgi alması veya düzenleme yapması için bir panel görüntülemesi.**

**Hata İletme: Platformda oluşan hataların kullanıcıya bildirilmesi ve uygun şekilde işlenmesi.**

* 1. **Use-Case diyagramı**



1. **TASARIM**
   1. **Mimari tasarım**

**Kullanıcı Arabirimi (UI): Kullanıcıların tasarım araçlarına ve işlevlerine erişebileceği bir arayüz sağlar. Eleman ekleme, bağlantı oluşturma, simülasyon başlatma ve diğer işlevleri içerir.**

**Mantık Devre Tasarımı Motoru: Devre elemanlarını ve bağlantıları yönetir. Kullanıcıların eklediği elemanları saklar, düzenler ve günceller. Simülasyonu yürütür ve sonuçları işler.**

**Veri Yönetimi ve Depolama: Tasarım için gereken verileri saklar. Elemanların özelliklerini, durumlarını ve ilişkilerini yönetir. Simülasyon sonuçlarını geçici veya kalıcı olarak depolar.**

**Simülasyon Modülü: Mantık devresinin simülasyonunu gerçekleştirir. Elemanların durumlarına ve bağlantılara dayanarak çıkışları hesaplar. Sonuçları işler ve grafikler veya tablolar oluşturur.**

**Grafiksel Gösterim Modülü: Tasarım alanını ve elemanları görsel olarak temsil eder. Simülasyon sonuçlarını grafikler veya tablolar olarak gösterir.**

**Kullanıcıların tasarımı ve sonuçları görsel olarak izlemesini sağlar.**

**Hata Yönetimi ve İzleme: Platformda oluşabilecek hataları yakalar ve kullanıcıya bildirir. Hata günlüklerini izler ve geliştiricilere hata ayıklama ve sorun giderme için yardımcı olur**

* 1. **Kullanılacak teknolojiler**

**Tkinter: Python için bir GUI toolkit olan Tkinter, kullanıcı arayüzünü oluşturmak için kullanılabilir. Basit ve kullanımı kolay olması nedeniyle tercih edilebilir.**

**PySpice: Devre simülasyonu yapmak için PySpice kütüphanesi kullanılabilir. Bu kütüphane, Python'da SPICE devre analizini gerçekleştirmek için kullanılan bir araçtır.**

**Matplotlib: Sonuçları grafikler veya tablolar şeklinde görselleştirmek için Matplotlib kütüphanesi kullanılabilir. Bu, simülasyon sonuçlarını görsel olarak analiz etmek için faydalı olabilir.**

**NumPy: NumPy, Python'da bilimsel hesaplama ve veri işleme için kullanılan bir kütüphanedir. Simülasyon sonuçlarını işlemek ve veri analizi yapmak için kullanılabilir.**

**Pillow: Eğer platformda elemanların tasarımını saklamak veya sonuçları raporlamak için görüntü dosyaları oluşturmanız gerekiyorsa, Pillow kütüphanesi resim işleme işlevselliği sağlar**

* 1. **Kullanıcı arayüzü tasarımı**

**Kullanıcı arayüzü, Tkinter kütüphanesi kullanılarak Python dilinde oluşturulabilir. Arayüz, kullanıcının mantık devre elemanlarını sürükleyip bırakarak tasarlayabileceği bir tasarım alanı içerebilir. Bu alanda, kullanıcı devre elemanlarını (AND, OR, NOT kapıları, giriş ve çıkışlar, bağlantılar ve düğümler gibi) sürükleyip bırakarak yerleştirebilir. Ayrıca, arayüz, simülasyonu başlatma, duraklatma, sıfırlama gibi işlevleri gerçekleştirmek için kontrol düğmeleri de içerebilir. Ekran çıktıları, Tkinter arayüzünün farklı durumlarını ve kullanım senaryolarını gösteren ekran görüntüleri veya videoları içerebilir. Örneğin, arayüzün boş bir tasarım alanı, bir AND kapısı eklenmiş bir tasarım alanı, bir bağlantı oluşturulmuş bir tasarım alanı gibi senaryolar görüntülenebilir. Uygulama, Python betiği olarak çalıştırılarak başlatılabilir. Python yürütücüsü yüklendikten sonra, projenin ana dizininde bulunan "logic\_circuit\_platform.py" dosyasını çalıştırarak arayüz başlatılabilir. Bu, Tkinter arayüzünü içeren bir pencerenin açılmasına neden olacaktır. Kullanıcı, arayüz üzerindeki araçları kullanarak mantık devresini tasarlayabilir ve ardından simülasyonu başlatabilir**

1. **UYGULAMA**

**Kodlanan bileşenler ve açıklamaları:**

**LogicGate: Mantık kapılarını temsil eden sınıf. İlgili mantıksal işlemleri gerçekleştirir.**

**IOElement: Giriş ve çıkış elemanlarını temsil eden sınıf. Başlangıç değeri ile birlikte tanımlanabilir.**

**Connection: Mantık kapıları, giriş/çıkış elemanları veya düğümler arasındaki bağlantıyı temsil eder.**

**ConnectionNode: Bağlantı düğümlerini temsil eder.**

**LogicCircuitPlatform: Tkinter kütüphanesi kullanılarak oluşturulan mantık devreleri tasarım platformunu temsil eder.**

**Görev Dağılımı:**

**Bileşenlerin Tasarım ve Geliştirme Aşamaları: LogicGate, IOElement, Connection, ve ConnectionNode sınıflarının tasarımı ve geliştirilmesi.**

**Raporun Hazırlanması Sürecindeki Görev Dağılımı:**

**Proje tanımı ve gereksinimlerinin belirlenmesi.**

**Kodlanan bileşenlerin açıklamalarının hazırlanması.**

**Görev dağılımı ve proje ilerlemesiyle ilgili raporlama.**

**Karşılaşılan Zorluklar ve Çözüm Yöntemleri:**

**Arayüz Tasarımı: Tkinter ile karmaşık arayüz tasarımı yaparken bazı düzenleme ve konumlandırma zorlukları yaşanabilir. Bu durumda, arayüzün daha basit ve kullanıcı dostu olması için farklı yaklaşımlar denenmelidir.**

**Bağlantıların Yönetimi: Mantık devresindeki bağlantıların düzgün bir şekilde yönetilmesi zor olabilir. Bu nedenle, bağlantıların doğru bir şekilde oluşturulması ve güncellenmesi için dikkatli bir planlama yapılmalıdır.**

**Proje İsterlerine Göre Eksik Yönler: Grafiksel Kullanıcı Arayüzünün Geliştirilmesi: Projenin daha kapsamlı ve kullanıcı dostu bir arayüze sahip olması gerekebilir. Bu nedenle, arayüzün daha estetik ve işlevsel hale getirilmesi için ek geliştirmeler yapılabilir. Simülasyon Özelliğinin Genişletilmesi: Mantık devrelerinin simülasyonu için ek özellikler eklenerek, kullanıcıların devrelerini daha detaylı bir şekilde analiz etmelerine olanak sağlanabilir. Bu, projenin işlevselliğini artırabilir.**

1. **TEST VE DOĞRULAMA**

**Bu test süreci, yazılımın doğruluğunu ve işlevselliğini kontrol etmek için geliştirilmiştir. Testler, her bir bileşenin beklenen davranışını sergileyip sergilemediğini belirlemek için yapılmıştır. Örneğin, LogicGate sınıfının bir örneğinin doğru şekilde oluşturulup oluşturulmadığını kontrol eden bir test bulunmaktadır. Benzer şekilde, giriş/çıkış elemanları, bağlantılar ve bağlantı noktaları için de benzer testler mevcuttur. Bu test uygulaması, yazılımın her bir bileşeninin doğru şekilde çalıştığını doğrulamak için kullanılır. Testlerin başarılı olması durumunda, yazılımın işlevsel olarak doğru olduğu ve istenen işlevleri yerine getirebileceği kabul edilir. Ancak, bir veya daha fazla testin başarısız olması durumunda, bu durumun gözden geçirilmesi ve hataların giderilmesi gerekebilir. Bu şekilde, testler yazılımın güvenilirliğini artırır ve beklenmeyen hataların önüne geçilmesine yardımcı olur.**

[**https://github.com/Emre-Clk**](https://github.com/Emre-Clk)

[**https://github.com/dogussalann**](https://github.com/dogussalann)