Proje : Görev Takip Uygulaması

Projenin Açıklaması: Kullanıcı görevler ekleyerek bu görevleri takip eder.

**1. Amaç**

Görev Takip Uygulaması, kullanıcıların çeşitli görevlerini ekleyip, bu görevlerin durumunu ve ilerlemesini takip etmelerini sağlayan bir masaüstü uygulaması olacaktır. Uygulama, kullanıcıların görev ekleme, takip etme, tamamlama ve görevlerin genel durumunu görmelerine olanak tanır

### 2. ****Kullanıcı Gereksinimleri****

Uygulamayı kullanacak kişinin ihtiyaçlarını karşılamak için belirli işlevler gereklidir:

* **Görev Ekleme**: Kullanıcı, görev adını girebilmelidir. Bu görev bir listeye eklenmeli ve ilerlemesi takip edilebilmelidir.
* **Görevlerin Görüntülenmesi**: Kullanıcı, eklediği tüm görevleri bir liste halinde görebilmeli ve ilerlemesini takip edebilmelidir.
* **Görevlerin Tamamlanması**: Kullanıcı, bir görev tamamlandığında bunu sistemde işaretleyebilmelidir.
* **Görev İlerlemesini Gösterme**: Görevlerin ne kadarının tamamlandığını gösteren bir ilerleme çubuğu olmalıdır.
* **Bilgilendirme Mesajları**: Tüm görevler tamamlandığında, kullanıcıya bir bilgilendirme mesajı gösterilmelidir.
* **Kolay Kullanım**: Arayüz basit ve kullanışlı olmalıdır. Kullanıcılar bir görev eklediğinde veya tamamladığında bunu hızlı ve kolay bir şekilde yapabilmelidir.

**3. Fonksiyonel Gereksinimler**

Bu gereksinimler, uygulamanın yapması gereken işlevleri kapsar:

* **Görev Ekleme Fonksiyonu**:
  + Kullanıcı, bir görev adı girerek "Ekle" butonuna basar.
  + Girilen görev, bir listeye eklenir ve ilerleme çubuğu buna göre güncellenir.
* **Görev Takip Fonksiyonu**:
  + Kullanıcı mevcut görevleri bir Listview içinde görüntüleyebilir.
  + Görevler tamamlandıkça, kullanıcı ilerlemeyi ProgressBar üzerinden takip edebilir.
* **Zamanlayıcı ile Görev Takibi**:
  + Görevlerin ne kadar sürede tamamlanacağı, bir zamanlayıcı ile takip edilir.
  + Zamanlayıcı her bir saniye çalışarak, görevlerin ilerleme durumunu kontrol eder.
* **Görev Tamamlandığında Bilgilendirme**:
  + Tüm görevler tamamlandığında, bir mesaj kutusu ile kullanıcıya bilgi verilir.
  + İlgili etiketler ve mesajlar ekranda gösterilir.
* **İlerlemenin Görsel Olarak Gösterilmesi**:
  + ProgressBar, kullanıcıya görevlerin ne kadarının tamamlandığını gösterir. Tamamlanan her görev, çubuğun ilerlemesini sağlar.

**4. Fonksiyonel Olmayan Gereksinimler**

Bu gereksinimler, uygulamanın performansı ve kullanıcı deneyimini etkileyen faktörleri içerir:

* **Kullanıcı Dostu Arayüz**: Uygulama, kolay anlaşılır ve basit bir arayüze sahip olmalıdır. Görev ekleme ve ilerleme takibi basit butonlar ve form elemanlarıyla yapılmalıdır.
* **Performans**: Uygulama hızlı çalışmalı, zamanlayıcı veya progress bar gibi elemanlar takılmadan işlemleri gerçekleştirmelidir.
* **Genişletilebilirlik**: Uygulama, gelecekteki olası yeni özellikler (örneğin görev silme, düzenleme) için genişletilebilir bir yapıya sahip olmalıdır.
* **Uyumluluk**: Uygulama, Windows işletim sisteminde sorunsuz çalışmalıdır.
* **Kullanıcı Geri Bildirimi**: Görevler tamamlandıkça, kullanıcı bilgilendirilmeli ve bir başarı mesajı gösterilmelidir.

**5. Teknik Gereksinimler**

Projenin teknik altyapısı ve ihtiyaçları şu şekildedir:

* **Geliştirme Platformu**: Uygulama, .NET Framework kullanılarak C# dili ile geliştirilecektir.
* **Görsel Arayüz (UI)**: Windows Forms kullanılacaktır.
* **Zamanlayıcı**: Görevlerin tamamlanma süresini izlemek için Timer bileşeni kullanılacaktır.
* **İlerleme Çubuğu (ProgressBar)**: Görevlerin ilerleme durumunu göstermek için ProgressBar bileşeni kullanılacaktır.
* **Veri Yönetimi**: Görevler bellekte saklanacak, başlangıç aşamasında bir veri tabanı kullanılmayacaktır (opsiyonel olarak ileride eklenebilir).
* **Yazılım Araçları**:
  + Visual Studio IDE
  + .NET Framework
  + Windows Forms

**6. Riskler**

Uygulama geliştirme sürecinde karşılaşılabilecek olası riskler şunlardır:

* **Zaman Yönetimi**: Projeye ayrılan zaman sınırlı olabilir, bu nedenle gereksinimlerin net ve doğru bir şekilde belirlenmesi önemlidir.
* **Kullanıcı Gereksinimlerinde Değişiklik**: Kullanıcı gereksinimlerinde sonradan değişiklikler talep edilebilir. Bu durum, projeyi yeniden yapılandırma ihtiyacı doğurabilir.
* **Performans Sorunları**: Eğer görev sayısı çok fazla olursa, Timer ve ProgressBar bileşenlerinde performans sorunları yaşanabilir.

**7. Başarı Kriterleri**

Uygulamanın başarılı olarak kabul edilmesi için şu kriterler karşılanmalıdır:

* Görev ekleme ve tamamlama işlevlerinin sorunsuz çalışması.
* ProgressBar’ın her görev tamamlandıkça doğru şekilde ilerlemesi.
* Tüm görevler tamamlandığında kullanıcıya bilgi veren bir mesajın gösterilmesi.
* Uygulamanın kararlı bir şekilde çalışması ve hızlı tepki vermesi.

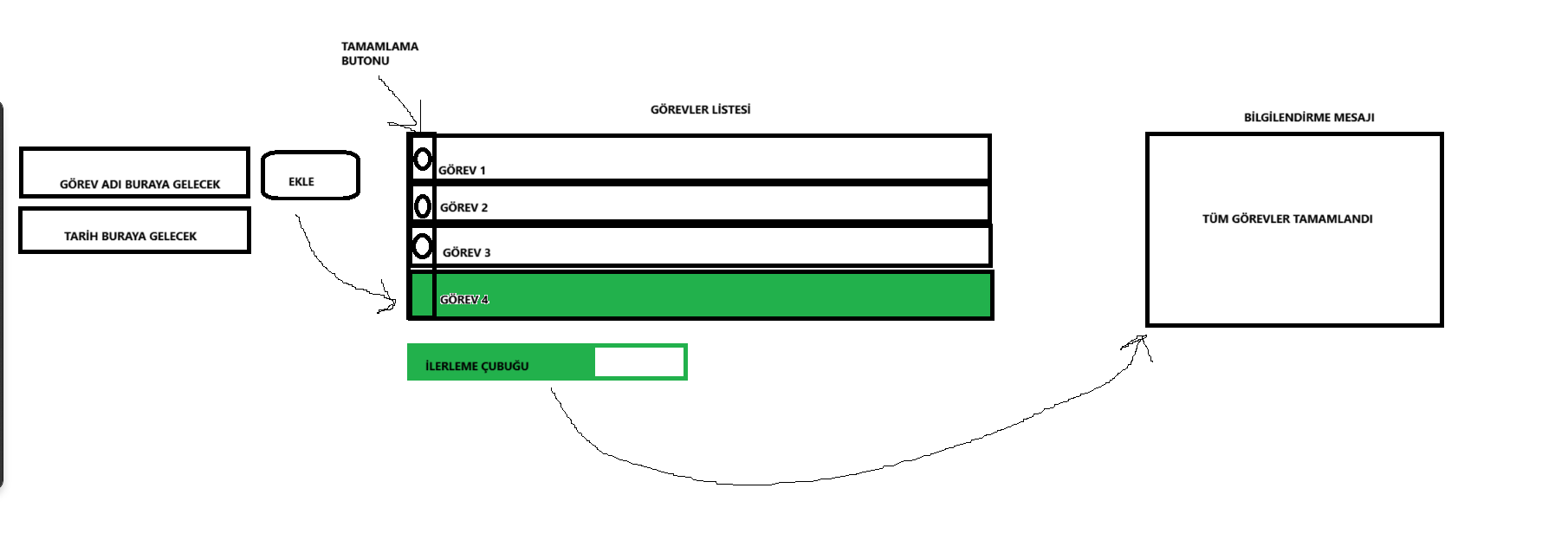
**8. Zaman Çizelgesi**

Projenin tamamlanma süresi kullanıcı deneyimine göre değişebilir, ancak genel olarak:

* **Arayüz Tasarımı**: 1-2 gün
* **Temel Fonksiyonların Geliştirilmesi**: 2-3 gün
* **Testler ve Hata Ayıklama**: 1-2 gün
* **Son İyileştirmeler ve Optimizasyonlar**: 1 gün

Bu gereksinim analizi, Görev Takip Uygulaması'nın nasıl geliştirilmesi gerektiğini ve proje boyunca nelere odaklanılması gerektiğini ortaya koymaktadır. Proje boyunca kullanıcı gereksinimlerine ve olası değişikliklere göre gereksinim analizi de güncellenebilir.

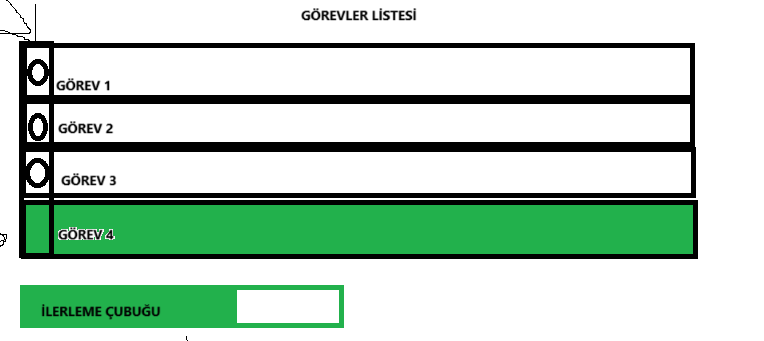
9. TASARIM KURGUSU



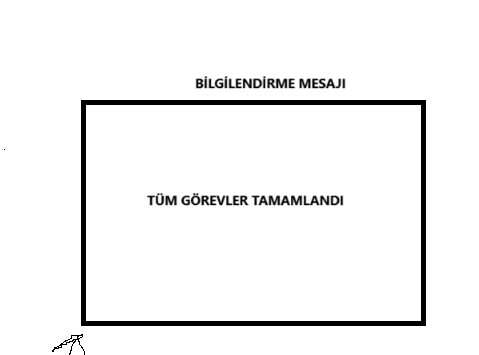
9.1 Görev ekleme işlevi



9.2 Görevleri listeleme ve tamamlama işlevi



9.3 Tüm görevler tamamlandığında bilgilendirme mesajı



GÜNCELLEME 1.0 – DOSYA SİSTEMİNDE VERİLERİ SAKLAMA

Örnek Proje Açıklaması (Türkçe)

To-Do List Uygulaması

Bu proje, günlük görevlerinizi takip etmek ve organize etmek için geliştirilmiş bir masaüstü uygulamasıdır. Uygulama, görevlerin eklenmesi, güncellenmesi, silinmesi ve durumlarının izlenmesini sağlar. Kullanıcı dostu bir arayüzle tasarlanmış olan bu proje, bireylerin zaman yönetimini kolaylaştırmayı hedefler.

Özellikler:

Görev Ekleme: Kullanıcılar yeni görevler oluşturabilir.

Görev Düzenleme: Seçilen bir görevin adı, durumu ve son tarihi kolayca güncellenebilir.

Görev Tamamlama: Görevlerin durumları (tamamlandı/tamamlanmadı) bir checkbox ile işaretlenebilir.

Görev Silme: Gereksiz hale gelen görevler kolayca silinebilir.

Görev Durum İzleme: Görevlerin tamamlanma durumu, bir yüzde çubuğu ve tamamlanan görevlerin strikeout (üstü çizili) olarak gösterilmesiyle görselleştirilir.

Tarihe Göre Renk Kodlama: Son tarih yaklaşan veya geçen görevler renklerle belirtilir.

Kullanıcı Arayüzü: Modern ve minimal bir tasarımla kullanıcıların kolayca etkileşimde bulunabileceği bir arayüz.

Kullanılan Teknolojiler:

C#: Masaüstü uygulamasının temel programlama dili.

WinForms: Kullanıcı arayüzü tasarımı ve geliştirilmesi.

Custom Drawing: Görev durumlarına göre özelleştirilmiş görseller (örneğin, strikeout metin).

Öğrendiklerim:

Bu proje sırasında masaüstü uygulama geliştirme becerilerimi geliştirdim ve aşağıdaki konuları öğrendim:

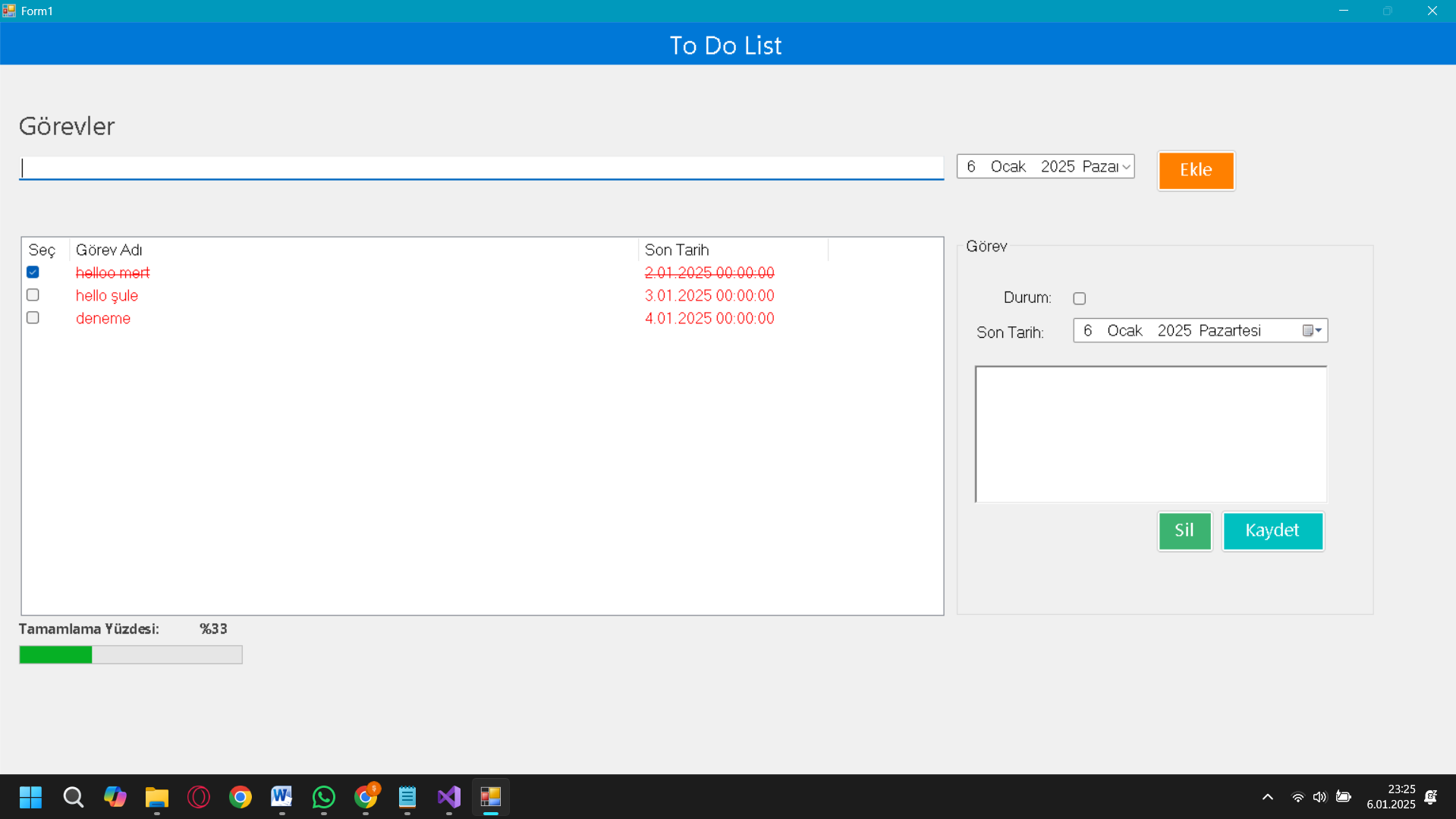
Olay Tabanlı Programlama: ListView ve diğer kontrollerin olaylarını etkin bir şekilde kullanma.

Liste Yönetimi: Görevlerin durumlarına ve tarihlerine göre liste güncellemeleri.

UI Özelleştirme: Kullanıcı deneyimini artırmak için özelleştirilmiş çizimler ve düzenler.

Proje Hedefi:

Bu proje, bireysel görev yönetimini kolaylaştırmak için bir araç sunmayı ve aynı zamanda masaüstü uygulama geliştirme konusundaki yetkinliğimi artırmayı amaçladı.



Bundan Sonra Geliştirilebilecek Kısımlar

Eğer daha fazla özellik eklemek isterseniz veya geliştirmek isterseniz, şu önerileri düşünebilirsiniz:

1. **Görev Arama:** Kullanıcıların görev isimleri veya son tarihlerine göre arama yapmalarını sağlayabilirsiniz.
2. **Kategori Ekleme:** Görevler için farklı kategoriler (iş, kişisel, alışveriş vs.) eklenebilir.
3. **Hatırlatıcı Özelliği:** Belirli bir tarihte yaklaşan görevler için hatırlatma bildirimi eklenebilir.
4. **Kullanıcı Deneyimini İyileştirme:** Görev eklerken ya da düzenlerken daha kullanıcı dostu bir arayüz tasarımı yapılabilir.
5. **Veritabanı Kullanımı:** Verileri bir dosya yerine SQLite gibi bir veritabanında saklamak, daha büyük ve daha esnek bir sistem için ideal olabilir.
6. **Verilerin Görselleştirilmesi:** Görevlerin tamamlanma yüzdesini grafiklerle görselleştirebilirsiniz (örneğin, bir pie chart).
7. **Datetimepickerda yeni görev eklerken bugünün tarihinden itibaren tarih seçilebilmesi.** Geçmiş tarihlerin enable özelliğinin false yapılması

Ekspertiz Seviyesi Unsurları (Bu projede olmayan)

Veritabanı Entegrasyonu: Eğer SQLite ya da SQL Server kullanarak görevleri bir veritabanında saklamak isteseydiniz, proje zorluk seviyesi daha da artardı.

Çok Katmanlı Mimari: İş mantığı (business logic) ve veri erişimi (data access) gibi katmanları ayırmak projeyi ileri düzeye taşıyabilir.

Modern UI Tasarımı: Daha modern ve kullanıcı dostu bir arayüz (örneğin, WPF ya da dış UI kütüphaneleri kullanılarak) geliştirmek uzmanlık gerektirebilir.