

*



Öğrenci AD/SOYAD: Ayhan çelik

Numarası : HR210041

Ders: Yazılım geliştirme ve ortam araçları

Konusu: Zonguldak Milletvekili oy giriş sistemi

Projenin Bütün Süreçleri ve Süreç Adımları

Zonguldak iline ait milletvekilleri hakkında bilgi edilmek için Google de arama yaptım, bu arama sonucunda Zonguldak iline ait milletvekili sayısı 5 olduğunu ve hangi partilerin ait olduklarını buldum Bulduğum bilgileri başka kaynaklar(Yüksek Seçim kurulu) teyit ettim. Bilgilerin doğru olduktan sonra bilgileri dokümana ekledim

Projeye Zonguldak Milletvekili Seçimi Oy giriş Sistemi deniyor .Bu sistem sayesinde artık online olarak herhangi bir yere gidilmeden gerek kalmadan , şeffaf bir biçimde sandık görevlilerin milletvekilleri verilen oylar sisteme girebilecektir. Bu oy sistemi sadece Zonguldak ili için geçerli olup , Zonguldak ilinde yaşayan kişilerin verdiği oylar sisteme kayıt olacaktır.

Projeye benzer örneklerin kısıtlı olması nedeniyle daha farklı yapıdaki , seçim üzerine olan sistemleri inceleyerek ve bu sistemler üzerinde gerekli değişiklikleri uygulayarak projenin yapımı için karar.kıldım.

Projeye için Waterfall(Şelale) modelini tercih ettim ve projeye oluşturmaya başladım

Çözümleme adımında Zonguldak halkının 14 mayıstaki milletvekili seçimi için neye ihtiyacı olduğunu , neden ihtiyacı olduğunu araştırdım ve Zonguldak halkının ihtiyacı olan şeyin Zonguldak Milletvekili Seçimi Oy giriş Sistemi olduğunu gördüm.

Zonguldak Milletvekili Seçimi Oy giriş Sisteminde kullanılacak ekranları(sayfaları) belirledim ve bu ekranlar :

1. **Giriş Sayfası:** Görevlilerin giriş bilgilerin gireceğe doğru girdiği taktirde anasayfaya yönlendirileceği ve girişin başarılı olduğunu ifade edecek sayfadır.
2. **Ana Sayfa:** Giriş işlemi sonrası başlangıç sayfasıdır . Burada milletteki seçim hakkında kısa bilgiler, diğer sayfalara yönlendirilme için kutucuklar ve seçim tarihin bitişine kalan süreyi gösteren bir kısım bulunur.
3. **Öneri / Şikâyet Sayfası:** Bu sayfada görevliler yaşadıkları sorunları veya sistemin geliştirmesi için gereken düşüncelerin destek ekibi ile paylaşabilecekler.
4. **Milletvekili /Adayları Sayfası :** Bu sayfada seçme katılmış olan milletvekili adayları listelenir. Kart yapısı şeklinde listelenen milletvekili adaylarına ait resim , isim , soyisim ,bulundukları parti gibi kısa bilgiler bulunur. Tıklandığı taktirde Oy Giriş Sayfasına yönlendirir.
5. **Oy Girişi Sayfası :** Görevliler bu sayfada milletvekili adayının aldığı oyları girecekler .
6. **Oy İstatistik Sayfası :** Bu sayfada görevlilerin girmiş oldukları mevcut oy durumuna göre adayların toplam oy sayıları gibi bilgilerin bulunduğu sayfadır.

Bu ekranların yapımına karar verdim.

Tasarım adımında yazılımın arayüzünü , butonları , görseller , renklerini , temasını , mobil ve web ekran tasarımlarını ve back-end araştırmasını yazdım.

Gerçekleme adımında yazılımı java dili ile yazacağıma kara verdim. Kodlanacak kısımların önemli noktalarını önceliklendirerek ve gerekli kritikleri yaparak kodladım.

Sinama adımında gerekli test senaryolarını oluşturdum , kod yazıldığında gerekli kısımlarda , sürekli olarak testler yaptım . Hem statik hem dinamik testler yaptım ve bu şekilde ilerledim.

Bakım aşamasında sistemi yayınladım ve bakım aşamasında ortaya çıkan hataları düzelttim , görsel açıdan iyileşmeler yaptım . Sistemin performansını iyileştirecek ve kullanıcılara daha kolay ve hızlı bir deneyim sunacak değişiklikler yaptım.

Daha sonra projenin UML Use Case diyagramlarını ve Uml Class diyagramlarını oluşturdum

Diyagram tasarımlarını <https://app.diagrams.net/> sitesinden oluşturdum

Diyagramları UML Class Diyagramları ve Use Case Diyagramları başlıkları altında açıkladım ve görseller ile destekledim.

Projenin Yazılım Süreç Modeli ve Süreçler

Öncelikle projem için Waterfall (Şelale) modelin tercih ettim. Bu model tercih etmemin sebebi daha oturmuş ve klasik bir yapı olması. Proje ufak çaplı olduğu için bu modelin kullanılması herhangi Bir sorun teşkil etmiyor. Şelale modelinin adımlarına değinmek gerekirse sırasıyla :

1. Gereksinim Aşaması
2. Tasarım Aşaması
3. Uygulama Aşaması
4. Doğrulama / Test Aşaması
5. Dağıtım Aşaması
6. Bakım Aşaması

Olarak sıralanabilir. Şelale modelinin avantajları olduğu gibi dezavantajları da mevcut bunlarında kısa şu şekilde açıklayabiliriz:

Avantajlar:

- Basit ve küçük projeler için uygundur
- Gereksinimler iyi anlaşılmıştır

- Anlaması kolaydır
- Yönetmesi kolaydır
- Kapsamlı belgeler

Dezavantajlar:

- Kapsam değişikliklerine izin vermez
- Gereksinim değişikliklerine izin vermez
- Projenin bitimine kadar çalışan ürün yoktur
- Beklenmedik riskleri kolayca ele almaz

Adım adım proje aşamalarında yaptıklarımı açıklamak gerekirse:

Gereksinim Aşaması:

Bu aşamada iş ihtiyacını, kullanıcı gereksinimlerini , kısıtlamalar ve riskler gibi projeyle ilgili gereksinimleri araştırdım . Örneğin bu projede iş ihtiyacım partilere verilen oyların sandık görevliler tarafından sisteme yüklenmeseydi ve kullanıcı gereksinimleri oy verme sürecin güvenli ve hızlı Bir hale getirmektir.

Waterfall modelinde geriye dönüş çok maliyetli olduğundan proje ile ilgili olabildiğince fazla Belge topladım. Waterfall modelinde her adım birbirine bağlı olduğundan dolayı mümkün olduğunca öngörüle bulunmaya çalıştım. Gereksinim aşamasının sonunda projenin başından sonuna kadar çok net Bir taslağına sahip olmaya çalıştım. Örneğin sürecin her aşamasını, her aşamada kimin çalıştığını , anahtar bağımlılıkları , gerekli kaynakları ve her aşamanın ne kadar süreceğine dair Bir zaman çizelgesi oluşturdum.

Tasarım Aşaması:

Tasarım aşamasında projenin tüm gereksinimleri, kısıtlamaları ve tasarım Hedeflerini karşılayan bir tasarım oluşturdum. Projede kullanacağımız donanımı, programlama dilini ve kullanıcı arabirimi gibi bilgileri belirledik.

Ürün bilgi teknolojisi ile ilgili olduğu için tasarımı 2 ana bileşene, mantıksal tasarıma ve fiziksel tasarıma böldüm.

Mantıksal tasarımdan biraz bahsetmek gerekirse, üst düzey veri kaynaklarını ve süreçlerin gösteren yazılım soyut bir konseptidir. Mantıksal tasarım, hızlı , kullanıcı dostu ve oy verme sistemi gibi gerçek iş ihtiyacını karşılar . Mantıksal tasarım genellikle bir tasarım genellikle belirli donanımlara başvurmada diyagramları gösterir. Bu yüzden mantıksal tasarım tamamlanmadan proje ekibinin önüne geçilmesine ve fiziksel tasarıma atlamamaya dikkat ettim . Çünkü eğer dikkat etmeseydim bu nihai yazılım ürününde tasarım kusurlarına yol açacaktı.

Fiziksel tasarımda ise veri girişi ve çıkışı, verilerin doğrulanması , kullanıcı arayüzleri ve veri depolama gibi unsurları detaylandırdım. Kısaca bahsetmem gerekirse fiziksel tasarım yazılım tasarımını çalıştırmak için hangi donanımı kullanacağımızı anladığımız yerdir.

Uygulama Aşaması:

Uygulama aşamasını kısaca anlatmak gerekirse her şeyin eyleme geçirildiği aşamadır. Birinci adımdaki gereksinimleri belgesine ve ikinci adımdaki tasarım aşaması sürecine dayanarak , hem gereksinimler aşamasında hem de sistem tasarım aşamasında ana hatları çizilen yazılımı oluşturmak için tam geliştirme sürecine başladım.

Doğrulama / Test Aşaması:

Bu aşamada projeyi gerekli test ekibine teslim ettik. QA test uzmanları proje dağıtılmadan önce düzeltilmesi gereken hataları veya hataları aradılar. Buldukları tüm sorunları belgeleyerek benzer bir hata durumunda sorunu çözmeye yardımcı olması için belgelere başvurdular.

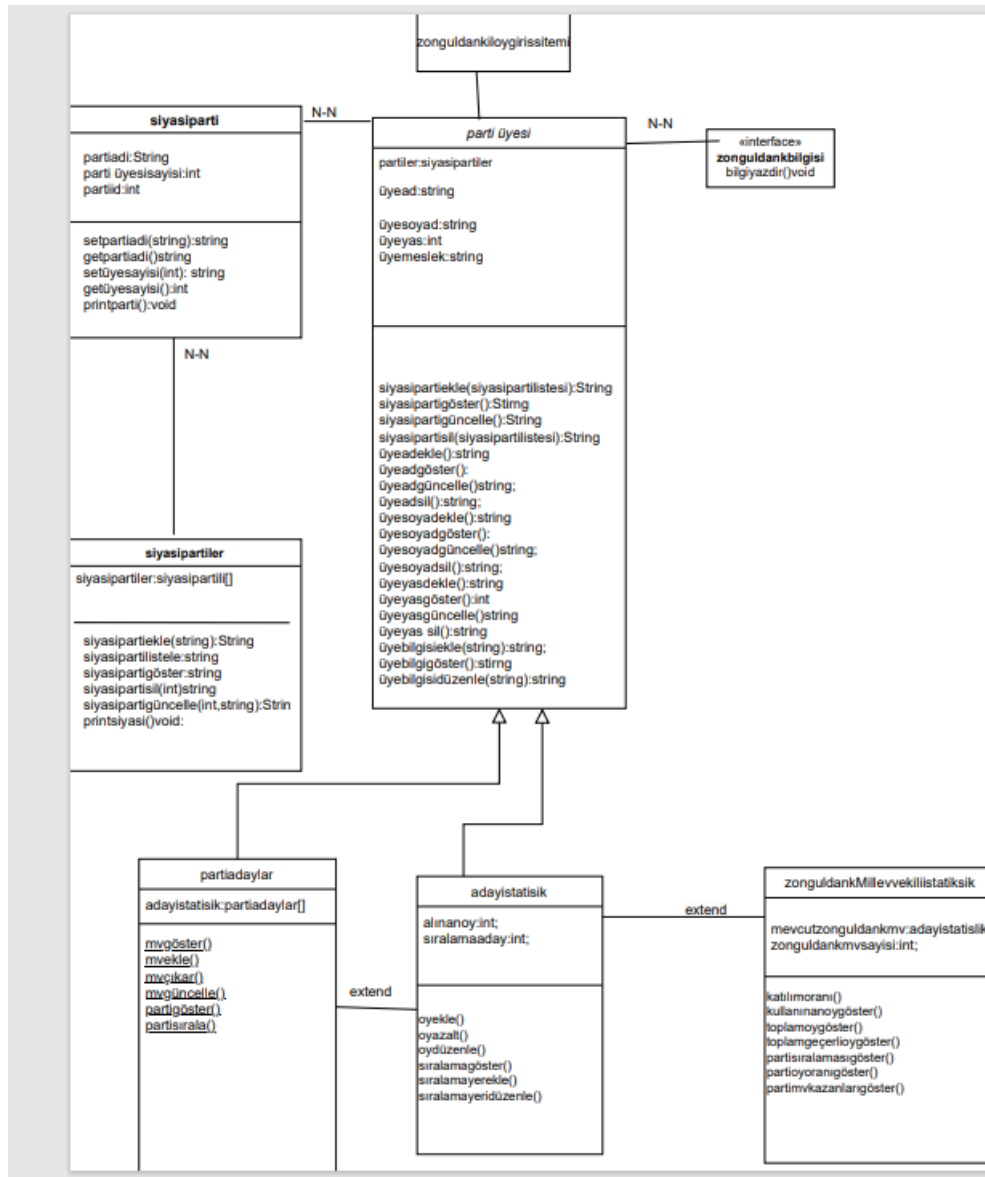
Dağıtım Aşaması:

Bu aşamada proje ile ilgili gerekli adımları takip edip bitirdikten sonra projeyi son kullanıcı için dağıtıma geçirdim. Kısaca projenin nihai aşamasını piyasaya sürdüm, müşteriye teslim ettim diyebiliriz.

Bakım Aşaması:

Bu aşamada kullanıcı müşteri ortamındaki sorunları ele alarak kısa bir izleme sürece geçirdim. Projemdeki bazı sorunları topladım ve bazı yenilikler ekleyerek projeyi güncelledim. Projenin daha optimize çalışmasını sağlayacak hareketlerde bulundum ve bakım aşamasını bununla ilgili birime aktardım.

UML CLASS Diyagramı



UML Class diyagramında 6 sınıf ve bir arayüz(interface) bulunuyor bunlar:

- Zonguldak bilgisi
- Parti üyesi
- Siyasi parti
- Siyasi partiler
- Parti adaylar

- Aday istatistik
- Zonguldak milletvekillileri

Bunlar sınıflar birbirine bağlıdır.. Sınıfların ve interface detaylı bir biçimle açıklamak gerekiyor

Zonguldak Bilgisi

- Bu interface ilişkili olan alt sınıf ZonguldakilOysistemi ile implenets ediyor.
- İçerisindeki fonksiyon şunlardır:
 - bilgiYazdir()

ZonguldakilOysistemi

- Bu sınıf Parti üyesi sınıfı tarafından extends edilmiştir ve bu sayede türeyen tüm sınıflara erişebilir.
- Bu sınıfın amacı il bilgisi tutmak ve bilgiYazdir() methodunu içini doldurmaktır.Bu sayede ilgili sınıfların il kontrolü sağlanır.

- Bu sınıfın özellikleri şunlardır:

Zongudak Bilgisi

- Fonksiyonu şunlardır:
 - bilgiYazdir()

Parti Üyesi

- Bu sınıf parti üyelerin bilgileri içeren sınıftır.
- Siyasi Parti sınıfı ile N-.N ilişki içersindedir.
- ZonguldakilOySistemi sınıfının extends etmiştir bu sayede üst sınıfın özelliklere ve fonksiyonlara erişir
- Bu sınıfın özellikleri şunlardır:
 - üye ad
 - üye soyad
 - üye yas

- üye meslek
- Bu sınıfın fonksiyon özellikleri şunlardır:
 - siyasipartiekle(siyasipartilisti)
 - siyasipartigöster()
 - siyasipartigüncelle()
 - siyasipartisi(siyasipartilisti)
 - üyeadekle()
 - üyeadgöster()
 - üyeadgüncelle()
 - üyeadsil()
 - üyesoyadekle()
 - üyesoyadgöster()
 - üyesoyadgüncelle()
 - üyesoyadsil()
 - üyeyasdekle()
 - üyeyasgöster():int
 - üyeyasgüncelle()
 - üyeyas sil()
 - üyebilgisi(siyasipartilisti)
 - üyebilgigöster()
 - üyebilgisi düzenle(siyasipartilisti)

Siyasi Parti

- Bu sınıf siyasi partiye ait bilgilerin olduğu sınıftır.
- Parti üyesi ile N-N ilişkisi içerisindedir.
- Siyasipartiler arasında sınıfı ile N-N ilişkisi içerisindedir.
- Bu sınıfın özellikleri şunlardır
 - Parti adı
 - Parti üye sayısı

- Bu sınıfın fonksiyonları şunlardır:

- setpartiadi(string)
- getpartiadi()
- setüyesayisi(int):
- getüyesayisi():
- printparti():void

Siyasi Partiler

- Bu sınıf siyasi partiler ve bilgilerini tutmaktır.

- Siyasi parti sınıfı N-N ilişkisi içerisinde.

- Bu sınıfın özellikleri şunlardır:

- siyasiparti:siyasipartiler[]

- Bu sınıfın sahip olduğu fonksiyon şunlardır:

- siyasipartiekle(string)
- siyasipartiliste()
- siyasipartigöster()
- siyasipartisil(int)
- siyasipartigüncelle(int,string)
- printsiyasi():void:

Parti Adaylar

- Bu sınıf parti adaylar ve bilgilerini tutmaktır.

- aday istatistik ile N-N ilişkisi içerisinde

- Sınıfın özellikleri şunlardır:

- adayistatistik:partiadaylar[]

- Sınıfın fonksiyonları şunlardır:

- mvgöster()
- mvekle()

- mvçıkak()
- mvgüncelle()
- partigöster()
- partisırala()

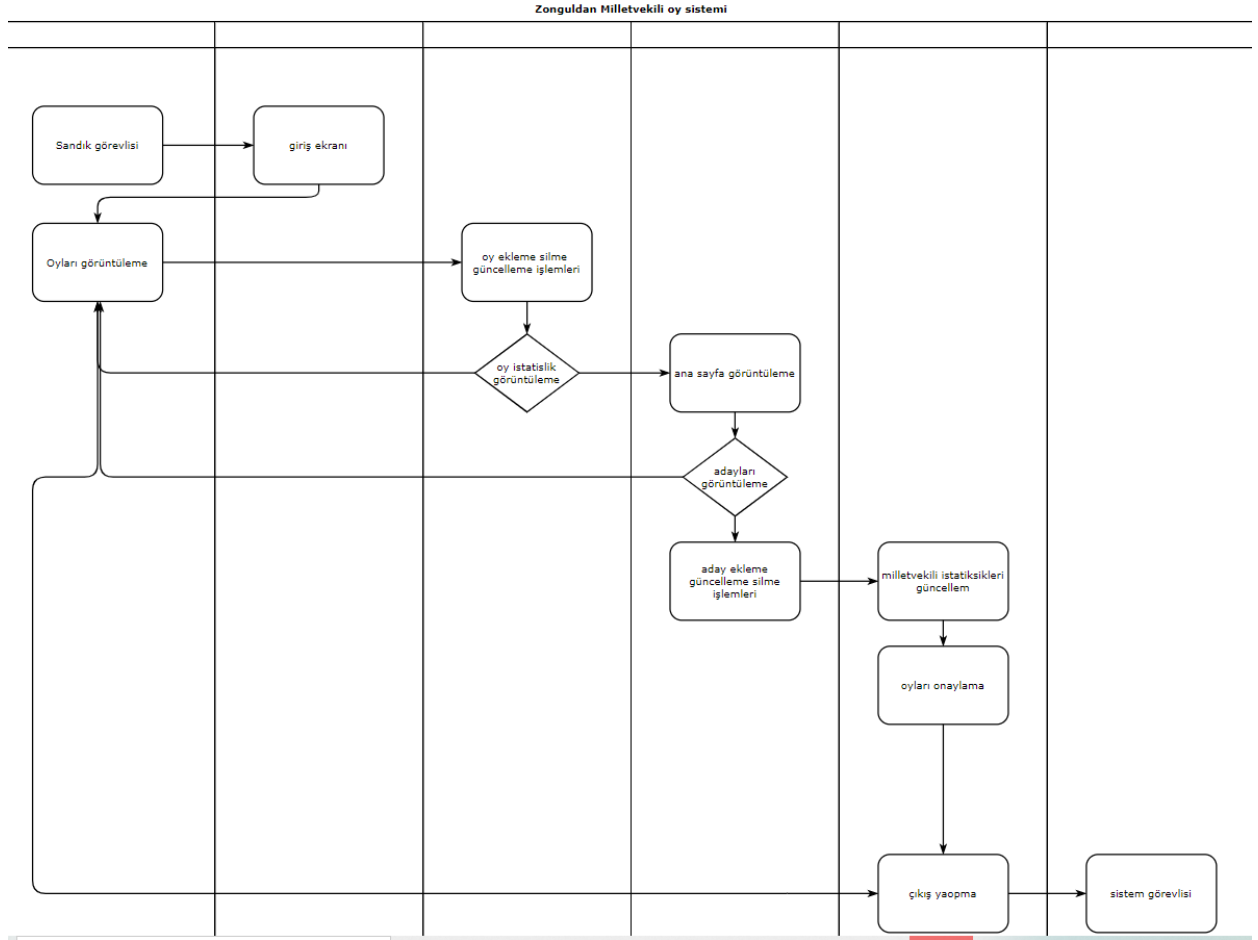
Adayistatistik

- Bu sınıf adayların aldığı oy ve istatistik bilgilerini tutmaktadır.
- Bu sınıf parti adayları sınıfı ile extends etmiştir.
- Bu sınıfın sahip özellikleri şunlardır:
 - alınanoy
 - sıralamaaday
- Bu sınıfın sahip olduğu fonksiyonları şunlardır:
 - oyekle()
 - oyazalt()
 - oydüzenle()
 - sıralamagöster()
 - sıralamayerekle()
 - sıralamayeridüzenle()

ZonguldakMilletvekiliistatistik

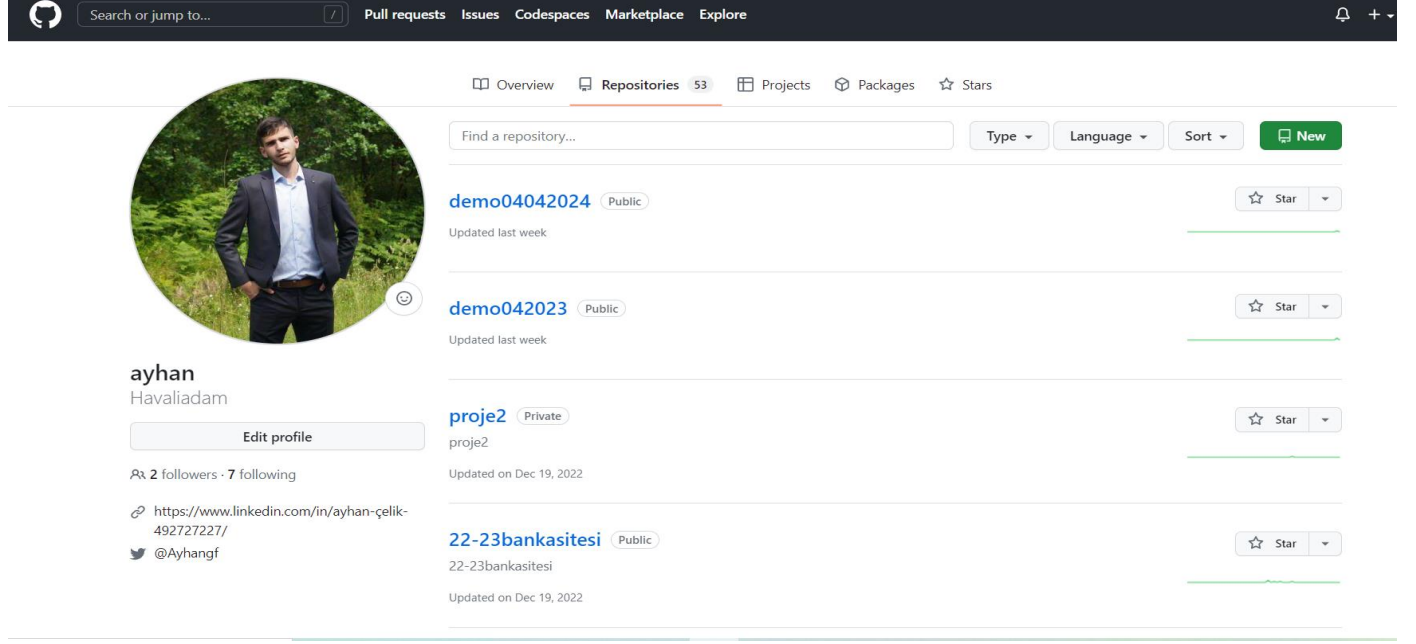
- Bu sınıf ZonguldakMilletvekillerin ve seçimdeki oy bilgileri ve istatistikleri tutan sınıftır.
- Bu sınıf Adayistatistik sınıfı ile N-N ilişki içerisindedir.
- Bu sınıfın sahip olduğu özellikleri şunlardır:
 - mevcutzonguldankmv:adayistatistik[]
 - zonguldankmvsayisi
- Bu sınıfın sahip olduğu fonksiyonlar şunlardır:
 - katılımoranı()
 - kullanınanoygöster()

- o toplamoygöster()
- o toplamgeçerlioygöster()
- o partisıralamasıgöster()
- o partioyoranıgöster()
- o partimvkazanlarıgöster()



Projenin Git Kullanılarak Githuba Yüklmesi

İlk olarak Github sitesinden yeni bir repository oluşturuyoruz. Bunun için resimde belirtilen new yazılı butona basıyoruz .



The screenshot shows the GitHub profile of a user named 'ayhan' (Havaliadam). The profile includes a circular profile picture of a man in a suit, a bio, and links to LinkedIn and Twitter. The 'Repositories' tab is active, showing a list of repositories. The 'New' button is highlighted in green. The repositories listed are 'demo04042024', 'demo042023', 'proje2', and '22-23bankasitesi'. Each repository has a 'Star' button and a dropdown arrow.

Repository Name	Visibility	Updated
demo04042024	Public	Updated last week
demo042023	Public	Updated last week
proje2	Private	Updated on Dec 19, 2022
22-23bankasitesi	Public	Updated on Dec 19, 2022

Şimdi açılan sayfadaki belirtilen kısma repository 'imiz vereceğimiz ismi giriyoruz .

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *



Havalidam ▾

Repository name *

yazilimgelistirmevize ✓

Great repository names are short and memorable. Need inspiration? How about [laughing-giggle?](#)

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.


Initialize this repository with:


Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

İsmi verdikten sonra repository' e kimlerin erişebileceğini belirlememiz gerekiyor. Biz private seçeneğine seçerek diğer insanlar tarafından gizleyeceğiz ve bu sayede yalnızca biz ve yetki verdiğimiz insanlar görebilecek , işlem yapabilecekler .

☐  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☒  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.


☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None** ▼

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

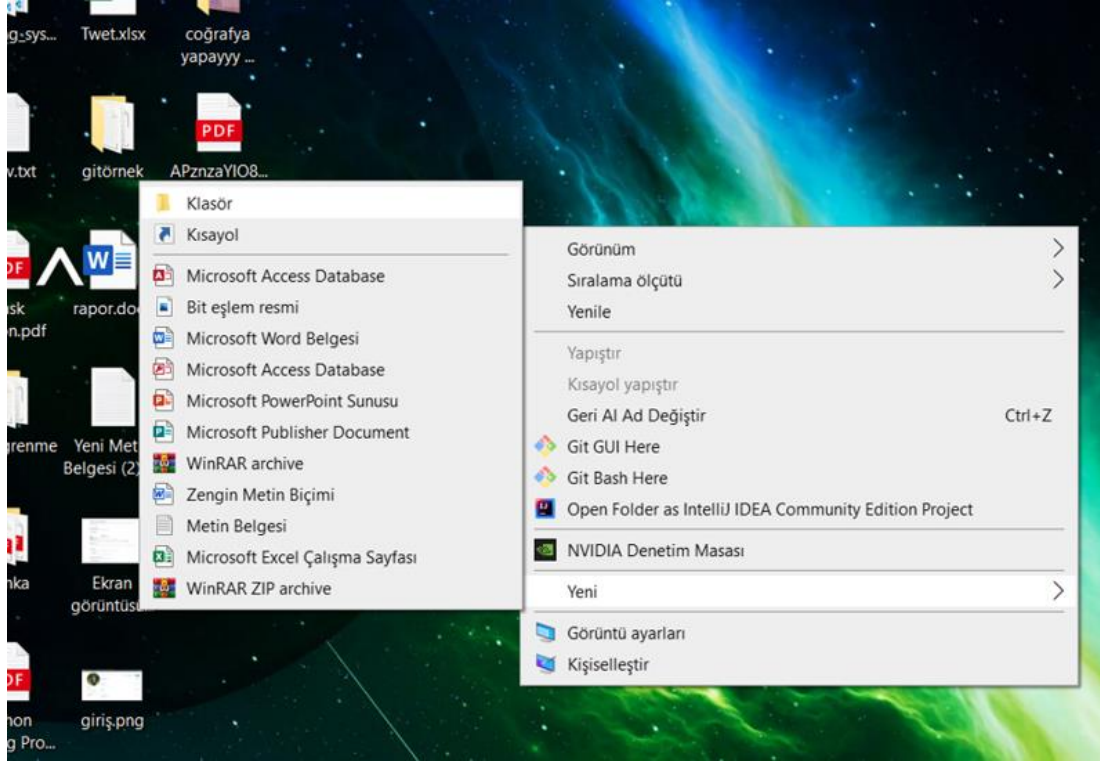
License: **None** ▼

 You are creating a private repository in your personal account.

Create repository

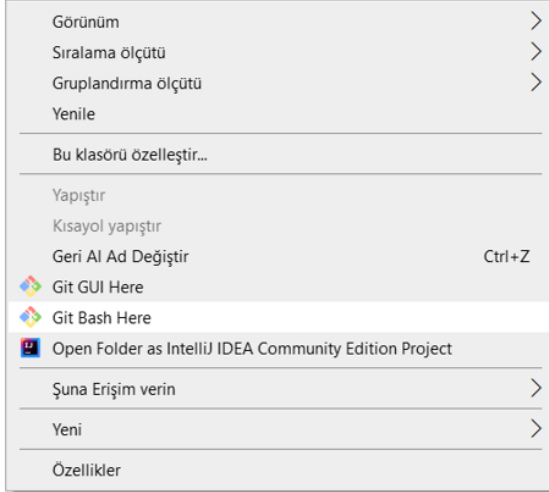
Bu aşamadan sonra artık repository ' miziz oluşmuş oluyor. Şimdi Git işlemleri ile ilgilenmek için bilgisayarına masaüstüne gidiyoruz.

Burada düzenli çalışmak adına Yeni Bir klasör oluşturup dosyalarımızı ve Git işlemlerinin buradan yöneteceğiz.



İlgili adımlardan sonra klasörümüz oluşmuş oluyor ve ona bir isim veriyoruz.

Klasörümüze bir isim verdikten sonra işlem yapacağımız dosyaları , klasörün içeresine taşıyoruz. Şimdi klasör içeresindeyken Git Bash Here yazısına tıklıyoruz.







İşlem sonrası Bir console / terminal ekranı açılıyor.

Açılan terminal ekranına **git init** yazıyoruz .

Yazdıktan sonra klasör içeresine .git adlı bir klasör gelmiş olduğunu görüyoruz. Eğer sizde gözükmiyorsa dosya gezgininden gizli dosyaları göster seçeneğinin seçerek , gözükmesini sağlayabiliriz.

```
MINGW64:/c:/Users/ayhan/Desktop/yazilimgelistirmevize
ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize
$ git init
Initialized empty Git repository in C:/Users/ayhan/Desktop/yazilimgelistirmevize/.git/
ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$
```

 .git	14.04.2023 11:54	Dosya klasörü
 Class.pdf	13.04.2023 14:49	Microsoft Edge PD...
 usecasediagram.png	12.04.2023 17:13	PNG Dosyası
 Yazilimgelistirmeortamaraçları vize.docx	13.04.2023 14:39	Microsoft Word Be...

Şimdi terminal 'e git status yazarak dosyaların durumunu kontrol ediyoruz

```
ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  "Yaz\304\261l\304\261mgeli\305\237tirmeortamara\303\247lar\304\261 vize.docx"
  usecasediagram.png

nothing added to commit but untracked files present (use "git add" to track)

ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$
```

Görüldüğü üzere dosyalar git 'e eklenmemişe bu nedenle isimleri kırmızı renkte yazıyor.

Şimdi dosyaları git add . komutu ile git 'e ekliyoruz.

```
ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$ git add .

ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$ |
```

Git add , komutundan sonra git status ile kontrol sağlıyoruz

```
ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Class.pdf
    new file:   "Yaz\304\261l\304\261mgeli\305\237tirmeortamara\303\247lar\304\261 vize.docx"
    new file:   usecasediagram.png

ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$
```

Görüldüğü üzere dosya isimleri yeşil yazmakta bu durumda dosyaların git 'e eklendiğini görmüş olduk.

Şimdi dosyalarımızı **git commit -m "firstCommit"** komutu ile commit edeceğiz.

```
ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$ git commit -m "firstCommit"
[master (root-commit) cd7680d] firstCommit
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Class.pdf
create mode 100644 "Yaz\304\261l\304\261mgeli\305\237tirmeortamara\303\247lar\304\261 vize.docx"
create mode 100644 usecasediagram.png

ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$
```

İşlemleri **git status** ile kontrol ediyoruz ve başarılı olduğunu görüyoruz.

Şimdi **git branch -M Main** komutu ile Yeni Bir branch oluşturuyoruz.

```
ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$ git status
On branch master
nothing to commit, working tree clean

ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (master)
$ git branch -M main

ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (main)
$ git remote add origin https://github.com/Havaliadam/yazilimgelistirmevize.git

ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.21 MiB | 367.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Havaliadam/yazilimgelistirmevize.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

ayhan@LAPTOP-LA2D8TC9 MINGW64 ~/Desktop/yazilimgelistirmevize (main)
$ |
```

Şimdi github repository 'imiz ile bağlantı kurmak için **git remote add origin** komutunun yanına repository adresini yazıyoruz.

Bu şekilde bağlantıyı sağlamış olduk.

Şimdi **git push -u origin main** komutu ile repository imize dosyaları push etmiş oluyoruz.

Görüldüğü üzere dosyalar gelmiş bulunuyor.

The screenshot shows the GitHub interface for a repository named 'Havaliadam / yazilimgelistirmevize' (Private). The repository is on the 'main' branch, which has 1 branch and 0 tags. The commit history shows a single commit 'firstCommit' by 'Havaliadam' with the hash 'cd7688d', made 1 minute ago. The commit includes three files: 'Class.pdf', 'Yazilimgelistirmeortamaraçları vize.d...', and 'usecasediagram.png', all committed 1 minute ago. A blue banner at the bottom of the commit list prompts the user to 'Add a README with an overview of your project.' with a green 'Add a README' button. The footer of the page shows the GitHub logo, copyright notice '© 2022 GitHub, Inc.', and links for Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, and Training.

Şimdi ozkansari (Hocamız) github repository 'imize erişmesi için yetki vereceğiz. Bunun için öncelikle settings 'e tıklıyoruz.

Açılan pencerede collaborators kısmına tıklıyoruz.

Havaliadam / yazilingelistirmevize (Private) Unwatch

<> Code Issues Pull requests Actions Projects Security Insights Settings

General

Access

Collaborators

Code and automation

Branches

Tags

Actions

Webhooks

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Who has access

PRIVATE REPOSITORY Only those with access to this repository can view it. [Manage](#)

DIRECT ACCESS 0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet


[Add people](#)

Tıkladıktan sonra açılan sayfada Add people butonuna tıklıyoruz.

Actions Projects Security Insights Settings

Add a collaborator to yazilingelistirmevize

Q Ozkan Sari

 **Ozkan Sari**
ozkansari • Invite collaborator

Select a collaborator above

İsmin, girdikten sonra tıklayarak kişiyi seçiyoruz.

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

[Manage](#)

DIRECT ACCESS

1 has access to this repository. 0 collaborators. 1 invitation.


Manage access


Add people

☐ Select all

Type ▾

☐

**Ozkan Sari**
Awaiting ozkansari's response

Pending Invite 

Remove

< Previous Next >

Butona tıklıyoruz ve kişiye collaborators olması için Bir istek gönderiliyor.

Kişiye gönderilen istek şuan beklemede , kabul ettiği taktirde collaborators olarak tanımlanacak

