



**T.C.**

**FIRAT ÜNİVERSİTESİ  
TEKNOLOJİ FAKÜLTESİ  
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

**YMH453 YAZILIM MÜHENDİSLİĞİ GÜNCEL KONULARI**

Proje Dokümantasyonu

**GÜVENİLİR BULUT UYGULAMASI  
(SANDIK)**

**Proje Ekibi**

180541044 Melih ÇİÇEK  
180541027 Berat BAL  
180541050 Oğuzcan BAŞ  
180541006 Harun ŞERLİ  
180541024 Ekrem SİVRİKAYA  
175542012 Serap HERKİLOĞLU

**EKİM – 2021**

---

## 1. GİRİŞ

---

### 1.1 Projenin Amacı

Güvenilir bir bulut tabanlı depolama sistemi tasarlamaktır. Mevcut sistemlerin oluşturduğu iyi yönleri üzerinden geliştirilecek olan sistem de kullanıcı yönelik sade ve anlaşılır arayüz tasarımıyla negatif yönleri minimize edilmiş bir kullanıcı ortamı sunmaktadır.

### 1.2 Projenin Kapsamı

Genel kullanıcı kitlesine hitap etmektedir. Sistem de internet ortamında verilerini depolamak isteyen öğrencilere hizmet vermesi planlanmaktadır. Sistem geliştirme aşamasında genel kullanıcı kitlesi mevcut üniversite öğrencileri için açılması planlanmaktadır.

### 1.3 Tanımlamalar ve Kısaltmalar

---

## 2. PROJE PLANI

---

### 2.1 Giriş

Proje planı ekip yapısında oluşturulan görev dağılımları üzerinden süreç içerisinde sürekli değişime uğramaktadır. Geleneksel proje modeli üzerinden çevik yazılım modeli bir yaklaşım ile aşamalarda süreç içerisinde geri dönüşler ve atlamalar yapılmaktadır.

### 2.2 Projenin Plan Kapsamı

Ekip yapısı oluşturmak mevcut şartların değerlendirilmesi ve ekip iş dağılımı yapılması ön şart ve gerekliliklerin belirtildiği gibi yapılmasını sağlamak çevik bir yazılım süreci içerisinde gerekli programın kontrollü bitirmektir. Projenin istenceleri belli ve kesinlik barındırdığı için geleneksel model yaklaşımı ile ilerlenmektedir.

### 2.3 Proje Zaman-İş Planı

	Oct			Nov				Dec				Jan				Feb		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Research																		
Research softwares for animation																		
Tutorials/ Learning Process																		
Animation skecthes																		
Doing animations																		
Research sortwares																		
language determination																		
Time arrangement btw software and animation																		
coding_																		
Creating informative materials (booklet, posters)																		
Program																		
Documenting																		
Presentation																		
Jury																		

### 2.4 Proje Ekip Yapısı

---

---

Beş ekip üyesi ile öncelikli bir proje değerlendirmesi yapılmış ve teknik ekip bilgi, yeteneklerine göre görev dağılımı yapılmıştır. Süreç içerisinde görevlerde değişimler yaşanmıştır. Görevlerde ki dağılımlar iş akış planlarının süreç içerisindeki dağılımdaki yoğunluğa göre değerlendirilip verilmektedir.

## 2.5 Önerilen Sistemin Teknik Tanımları

Programın nesne kontrolü ve hızlı bir biçimde destek sağlaması, kullanıcı kolaylığı ve kontrol protokollerine uygun çalışmasını sağlamak için: Nesne yapılı diller üzerinden düşünülmüştür. Orta seviyeli bir dillin yeterince hızlı çözümleme sağlayacağı tespit edilmiş ve ekip içinde sistemin NBS / VS üzerinden kodlanması, ara yüz yerine geri dönüt dosyası gönderilmesi ve programın güvenilirliği üzerinde düşünülmektedir.

## 2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları

Proje içerisinde yazılacaktır.

## 2.7 Proje Standartları, Yöntem ve Metodolojiler

Proje güvenlik ve yazılım standartlarımdan olan;

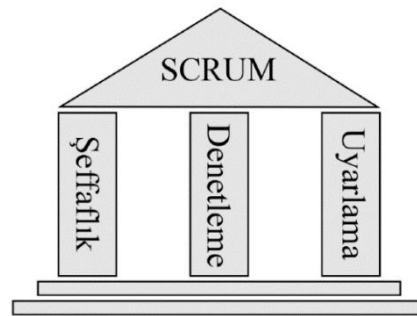
KVKK

CRPYTHON

Standartlar çevresinde oluşturulacaktır.

Yöntem olarak çevik yazılım süreci belirlenmiştir.

Metodoloji SCRUM Üzerinden gidilecektir. Resim1 de bize gerekli olan özellikleri sağlayacaktır. Ek-1 Dosyasında mevcut sistemler için scrumlar araştırması yapılmıştır.



Resim1: SCRUM Mimarisi

## 2.8 Kalite Sağlama Planı

Güvenilirlik, işlevsellik ve depolama desteği ile sağlanması planlanmıştır. Standart olarak belirlediğimiz standartlar şüphesiz ki proje için kalitenin göstergesi olacaktır. Projede ki sağlamlık kullanılan şifrelemeler ve sistemin güvenliği açısından bakılacaktır.

## 2.9 Konfigürasyon Yönetim Planı

## 2.10 Kaynak Yönetim Planı

Sistem bir bulut depolama sistemi olduğu için kaynak yönetiminde önemli yöntemler denenmektedir. Yöntemler arasında en faydalı yöntem olan;

.....

Seçilmiştir.

### 2.11 Eğitim Planı

Yapılmış ve benzer işlerin ekip olarak incelenmesi, eksik yönlerin tespitinin sağlanmasıyla gerekli dilin belirlenip üzerinde yapılacak çalışmaların başlaması ile bir eğitim planı oluşturulacaktır.

Kullanıcı için kulanınlan ücretli veya ücretsiz programlardan benzerliği, kolaylığı için zorluk olmayacaktır. Her duruma göre bir help butonu ya da aşamasal bilgilendirme yapılabilir halegetirebilir.

### 2.12 Test Planı

Test süreci, analiz -> tasarım -> test hazırlık süreci-> kodlama -> dinamik test süreci -> testin sonlandırılması -> ürün, resim3 şeklinde olacaktır.



Resim3: Test Aşaması

### 2.13 Bakım Planı

Kullanıcı geri dönütlerine ve geliştirme süreci yazılım yaşam döngüsü süreç modeline bağlı olarak bakım aşaması olacaktır. Versiyonlar arasında bakımlar yapılarak iyileştirilmiş bir sistem hizmeti verilecektir. Yıllık maliyet açısından bakıldığında sistemdeki en büyük bakımın depolamaya kaynak kısmına verilen yatırım olacağı için projede de ki versiyonlar ve bakımlar için ücretlendirme bazlı bir yöntem izlenmesi planlanmaktadır.

### 2.14 Projede Kullanılan Yazılım/Donanım Araçlar

Sanal bir bellek kullanımı için sanal donanım/ yazılımı düşünülmektedir. Arayüz kısmını sağlayacak olan html , php ve asp.net kısmıyla jacury de eklenmesi planlanmaktadır.

---

Veri kısmını MSQl düşünölmektedir. Sistemin web kısmında düşünölen yöntemler için ise: html5 ve css şablonları planlanmıştır. Sistemde kullanılacak olan yöntemler maliyet hesabına göre şu şekilde çıkmıştır:

---

### **3. SİSTEM ÇÖZÜMLEME**

---

#### **3.1 Mevcut Sistem İncelemesi**

Mevcut yapıların ücretsiz yazılım kullanımı deneyimleri ve sistem yorumları incelenmiş swot analizleri çıkarılmıştır. Ek-2 dosyasındaki sonuçlar ve ekip planları içerisinde sistem mevcut sistemlerden çıkarımla planlanmıştır. Kullanılan sistemler ve kullanıcılar tarafından oluşturulmuş bakış açıları, mevcut yapı açıkları ve yapılacaklar bu şekilde oluşturulmuştur.

##### **3.1.1 Örgüt Yapısı**

##### **3.1.2 İşlevsel Model**

##### **3.1.3 Veri Modeli**

Mevcut yapı oluşturmak ve bir veri kontrolü için hafıza sağlanmayacaktır. İleriki geliştirme ve güncellemelerde veri modeli oluşturulacaktır.

##### **3.1.4 Varolan Yazılım/Donanım Kaynakları**

İntihal.net temel alınmıştır. Üniversitelerin kullandığı deneyimsel kurallar temel alınmıştır.

##### **3.1.5 Varolan Sistemin Değerlendirilmesi**

Sistemde genel bir kullanıcı kullanımı olsa da genel anlamda belirli kullanma şansı verilmektedir. Yapılan hileler, kullanım açıkları program farkına göre fark edilmektedir. Şuan için nesne temelli bir program düşünmüş olursa da programın yapay zekâ destekli bir yapıda oluşturmasını, tespitin değişikliklerine göre değerlendirecektir.

#### **3.2 Gereksenen Sistemin Mantıksal Modeli**

##### **3.2.1 Giriş**

Model şuan için geliştirilebilir kısım ve geliştirilecek kısmı yapısına göre ikiye ayrılmıştır. Proje destek ve ekip yapısına göre gelişimi ilerleyecektir. Projede şu anlık bu yoldan ilerleyecektir.

##### **3.2.2 İşlevsel Model**

##### **3.2.3 Genel Bakış**

---

---

Dokümanın alınması ile doküman koşullar göre tarandıktan sonra gerekli değerlere döndürülür ve sonunda da bir rapor oluşturulur.

#### 3.2.4 **Bilgi Sistemleri/Nesneler**

Alınan doküman .txt uzantısı olarak alınacaktır. Dokümanın şartlarına göre tarandıktan sonra gerekli özet dokümanı verilir.

#### 3.2.5 **Veri Modeli**

Şuan için bulunmamaktadır.

#### 3.2.6 **Veri Sözlüğü**

Şuan için bulunmamaktadır.

#### 3.2.7 **İşlevlerin Sıradüzeni**

**Resim3** de izlenilecek olan mantıksal düzen belirlenmiştir. İzlenilecek olan yol süreç içerisinde daha net olarak oluşturulacaktır.

### **Resim3: Mantıksal Algoritma Düzeni**

#### 3.2.8 **Başarım Gerekleri**

Yazım kuralları, yazım tipi ve puntosu, sayfa düzeni oluşturulacak olan intihaller dikkate alınacaktır. Gerekli kaynak olarak verilen şartlar genel bir kural kabul edilmektedir.

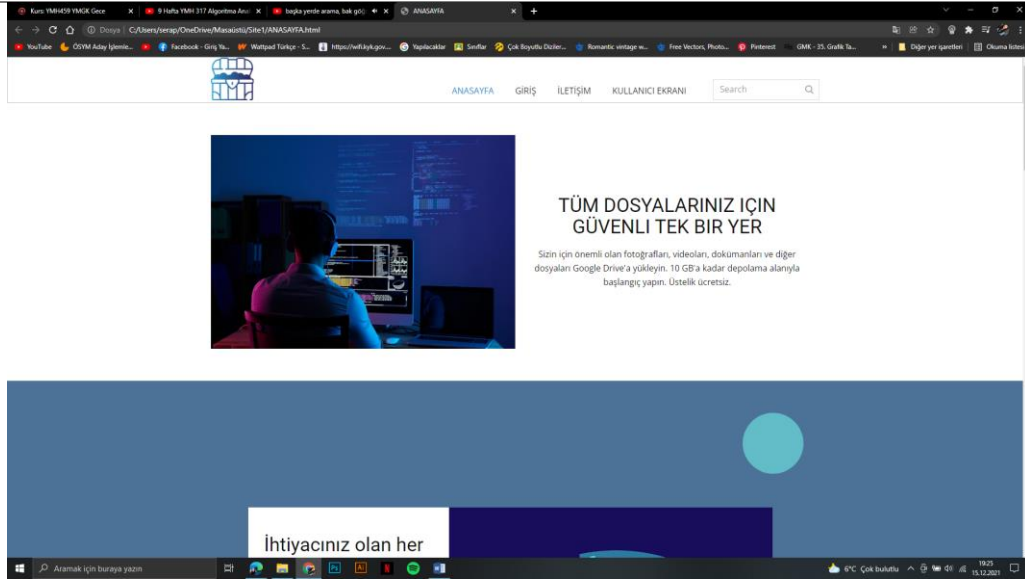
### **3.3 Ara yüz (Modül) Gerekleri**

#### 3.3.1 **Yazılım Arayüzü**

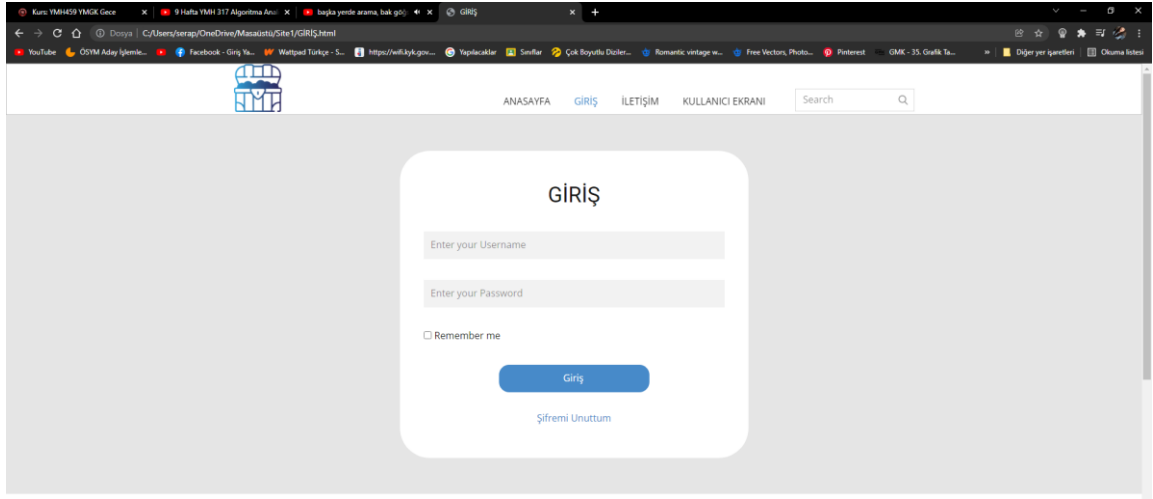
### **Resim4: kod sayfası**

#### 3.3.2 **Kullanıcı Arayüzü**

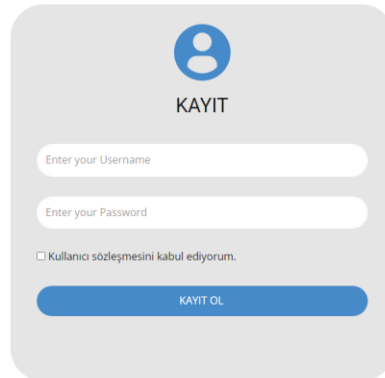
---



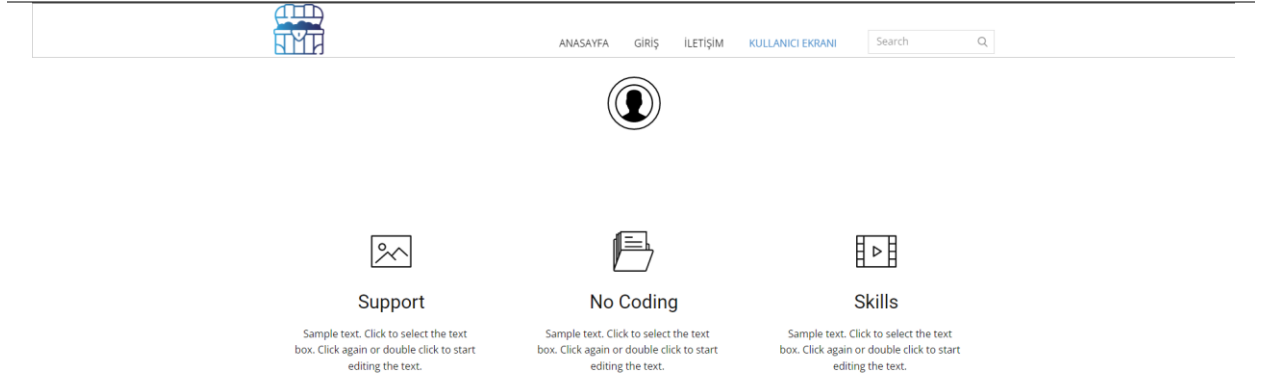
**Resim5: Program İlk Çalışma**



**Resim6: Program Çalışma**



**Resim7: Program kayıt ekranı**



**Resim7:** kullanıcı ekranı

### 3.3.3 Yönetim Arayüzü

Yazılım Arayüzü tarafında sağlanacak olan ara yüzden kullanıcıya aktif edilecektir. Bireysel kullanım sağlanacağı için sistem içinde herhangi bir yönetici(admin) ekranı düşünülmemektedir.

## 3.4 Belgeleme Gerekları

### 3.4.1 Geliştirme Sürecinin Belgelenmesi

Aşamalar sıralı bir şekilde yazılmaktadır iş – zaman planına dâhil olarak son süreçte oluşturulan belgeler bir dokümantasyon oluşturulacaktır.

### 3.4.2 Eğitim Belgeleri

Sertifika ya da kullanıcı belgelendirilmesi yapılması planlanmamıştır. Eğitim planı da mevcut sistem kullanıcıları için aynı perspektifte olacağı için bu durumdan mali - yetlendirme olanakları için düşünülmemektedir.

### 3.4.3 Kullanıcı El Kitapları

Kolay Ara yüz öğretileri ile sağlanacaktır.

## 4. SİSTEM TASARIMI

### 4.1 Genel Tasarım Bilgileri

Kullanıcı kolaylıklı ve güvenilir bir program oluşturulacaktır. Tasarımda bir yön veya model belirlenmemiştir. Sanal bir depolama programı olarak modellenecek olan projede işlevsel olmayı planlamaktayızdır.

#### 4.1.1 Genel Sistem Tanımı

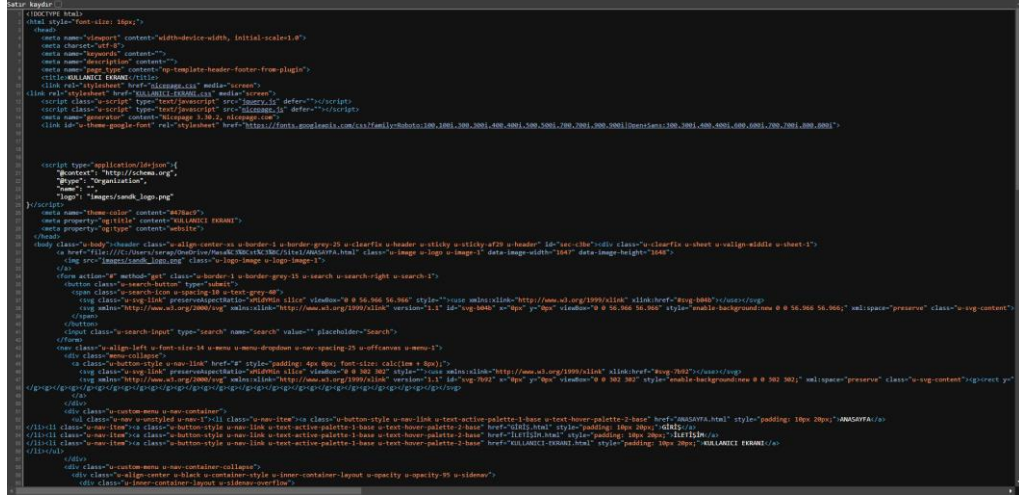


Mevcut sistem swot analizlerindeki benzerliği ile dikkat çekmeyen kendi model ve yapısını gerek hız ve kullanıcı işlevinde göstermektedir. Kural denetimi yazım ve doküman kontrolü yönünde etkin kullanım sağlayacak arka yüzde çalışacak bir projedir.

#### 4.1.2 Varsayımlar ve Kısıtlamalar

Bulunmamaktadır. Lakin proje maliyeti için şuan depolama alanı kısıtlamaları yapılarak ilk örnek bir sürüm ile sunulacaktır.

#### 4.1.3 Sistem Mimarisi



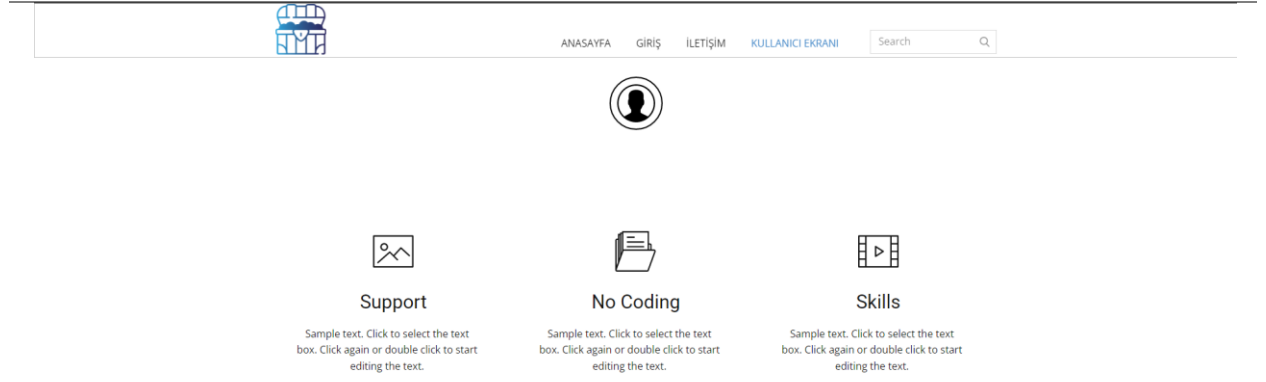
Resim3: Mantıksal Algoritma Düzeni

#### 4.1.4 Dış Arabirimler

Bir birim veya ek bir donanım eklenmemiştir.

##### 4.1.4.1 Kullanıcı Arabirimleri

Projede bir kullanıcı adı ve şifre ile giriş yapıcaktır. Sistme kayıt olur iken oluşturduğu bilgiler ile kendi bir giriş ekranı oluşturacaktır. Ekrandan sürümlerinde kişileştirme ve tasarım üzerinde de bunu sağlanması planlanmaktadır.



**Resim5:** Program İlk Çalışma

#### 4.1.4.2 Veri Arabirimleri

Direk dosya okuma ile alınan evrak dosyaları üzerinden inceleme yapılacaktır. Verileri dosyalamak için şuan depolama alanı ayrılmamıştır, intihal sorgulamak için ilerleyen süreçte dosyalar depolanarak ilerleyen süreçte intihal sorgulaması gerekir.

#### 4.1.4.3 Diğer Sistemlerle Arabirimler

**Resim10:** Aktivite Diyagramı

#### 4.1.5 Veri Modeli

#### 4.1.6 Testler

Genel hatlarıyla testlerimiz iki aşamada gerçekleştirilecek. Bilinen adıyla pilot üniversite uygulaması olarak sistem mevcut öğrencilere Google drivenin sağladığı desteği sağlayacaktır. Alfa Aşaması: Sistemin geliştirildiği yerde kullanıcıların gelerek katkıda bulunması sistemi test etmesi ile yapılacak. Beta Aşaması: Kullanıcı, geliştirilen sistemi kendi kullanıcı potforyosunu oluşturacaktır.

#### 4.1.7 Performans

Sistemin performansını etkileyen faktörlerin test verileri değerlendirilecek Sistemin Tasarıma Uygunluk Performansı; Tasarımı yapılan sistemin stabilizesi ve işleyiş performansı değerlendirilecek. Veri Yapısının Sistemle Performansı; Veri yapısının sistemle stabilizesi ve çalışma zamanındaki uyumluluk düzeyindeki Performansı değerlendirilecek.

## 4.2 Veri Tasarımı

---

#### 4.2.1 Tablo tanımları

#### 4.2.2 Tablo- İlişki Şemaları

### Resim11: Ucase şeması

#### 4.2.3 Veri Tanımları

#### 4.2.4 Değer Kümesi Tanımları

### 4.3 Süreç Tasarımı

#### 4.3.1 Genel Tasarım

Genel olarak tasarımda ilk önce mevcut sistem taraması yapılmıştır. Ardından giriş modülü onun ardından en sonda kullanıcı Arayüzü oluşturulmuştur.

#### 4.3.2 Modüller

##### 4.3.2.1 XXX Modülü

###### 4.3.2.1.1 İşlev

Kullanıcının sisteme müdahale edece bileceği ekrana erişmesi için aşması gereken bir birimdir.

###### 4.3.2.1.2 Kullanıcı Arabirimi

Sisteme kayıt/üyelik yaptıktan sonra erişim yapacağı ekrandır.

Kullanıcı olarak versiyonların ve kapasite artırmasıyla sağlanacak kullanıcı birimi versiyonları sağlanacaktır. Şuan için resim 7 de belirtilen tasarım birimi desteklenecektir.

###### 4.3.2.1.3 Modül Tanımı

Tasarım ekranında aşağıda belirtilen modüller düşünülmektedir.

Bunlar;

- Anasayfa
- Giriş/ kayıt
- Kullanıcı ekranı
- İletişim ekranları olarak modüller düşünülmüştür.

Yazılım olarak veri şifreleme ve arka plan yazılımları oluşmaktadır.

Veri tabanı olarak ise;

###### 4.3.2.1.4 Modül iç Tasarımı

RABBİTMQ üzerinde bir planlama düşünülmektedir. Sistem aktivasyonunu daha iyi sağlanması hedeflenmektedir.

---

#### **4.3.2.2 YYY Modülü**

##### **4.3.3 Kullanıcı Profilleri**

Akademisyen ve öğrenciler tarafından kullanılacaktır. Versiyon ve kapasite artırımları ile genel kullanıcı kitlesi oluşturulacaktır. Şuan için mevcut versiyon için akademik destekli bir depolama sağlayacaktır.

##### **4.3.4 Entegrasyon ve Test Gereksinimleri**

Şifreleme ve doğrulama gereksimleri en on plandaki testleri oluşturmaktadır. Entegrasyon sürecinde ise sistemdeki hataların giderilmesi planlanarak oluşan açıkların testleri yapılacaktır. Web üzerindeki açıkların testleri ve kontrolleri yapılacaktır. Mevcut sistemlerin kullanıcılarına sağladığı güvenlik protokol testleri için uygun birimlerde denenecektir.

#### **4.4 Ortak Alt Sistemlerin Tasarımı**

##### **4.4.1 Ortak Alt Sistemler**

Kullanılacak tek ortak sistem üniversitelerin sistemidir bu sistemle kişilerin kullanımı ile gerekli kontrolleri sağlayacaktır. Sistem destek alanı genel kitleye ulaştığında mevcuttaki alt sistem sorunlarının genel kitle içinde avantajlarıyla sistem versiyonlarının artılarıyla kullanıcılarımız için de iyi bir performans sağlayacaktır.

##### **4.4.2 Modüller arası Ortak Veriler**

##### **4.4.3 Ortak Veriler İçin Veri Giriş ve Raporlama Modülleri**

Ortak veriler için böyle bir birim şimdilik kullanılmamıştır.

##### **4.4.4 Güvenlik Altsistemi**

Yazılım sistemlerinin güvenilirliğe ilişkin nicelikleri, kullanıcıların gereksinimlerini Karşılacak şekilde ortaya koymak ve güvenilirliğin hesaplanmasına yönelik verileri toplama, istatistiksel tahminlere, ölçütlerin tespiti, yazılıma ait mimari özelliklerin belirlenmesi, tasarım, geliştirme ve bunlara yönelik çalışma ortamının belirlenmesi ve modellenmesini kapsamaktadır.

- A. Model seçme ve düzenlemeye yönelik faaliyetlerin temelinde uygun hedeflerin tespit edilmesi bulunmaktadır.
  - B. Hata ve aksaklıkların analiz edilmesi için uygun verilerin tanımlanması gerekmektedir.
  - C. Belirtilen hedeflere yönelik veriler modellenir.
  - D. Geçmişe yönelik verilerin zaman bilgilerini de içerecek şekilde elde edilerek yazılım geliştirme sürecine dâhil etmektir.
-

- 
- E. Yazılım geliştirme sürecinin modellenmesi, hata ile karşılaşılıp, test sürecine başlamak ve model doğrulama işlemlerine gerçekleştirmektir.
  - F. Güvenilirlik tahminleme modelinin seçilmesini sağlamaktır.
  - G. Güvenilirlik modeli tarafından kullanılacak olan parametrelerini tespit etmek.
  - H. Verilen bir noktayı kullanarak gelecekteki olası hatalar hakkında tahmin yapmak.
  - I. Tahmin edilen hata ve arıza oranları ile gerçekleşen değerleri kıyaslamaktır.

#### **4.4.5 Veri Dağıtım Altsistemi**

#### **4.4.6 Yedekleme ve Arşivleme İşlemleri**

Dokümanın doğruluğu için ilk incelemeler yapılacaktır ilk önce, herhangi bir nedenle zarar görmesi, sistemin çalışma süreçlerinde ciddi zararlar oluşturacağından evrak güvenliği önemlidir. Yaşanabilecek bir intihal durumları için sonrasında, depolanan verilerin geri yüklenememesi, sistemin sağlandığı akademik dosyaya çok ciddi zararlar verebilir. Bu nedenle sistemin çalışma süreçlerine bağlı olarak, yedekleme sistemleri kurulmalı ilerleyen süreçlerde benimsenecektir ve yedekleme işlemleri olarak takip edilmelidir. Yedekleme sistemlerinin kurulumu; yedeklenecek veri miktarı, yedekleme sıklığı, yedeklenen verinin zaman içerisinde değişme oranı ve maksimum veri kaybı gibi parametrelere bağlıdır. Sistemin birden fazla sunucusunun eş zamanlı yedekleme işlemini yapabilmesi, işletim sistemlerinin kayıt dosyalarını tam ve eş zamanlı olarak yedekleyebilmesi ve işletim sistemleri üzerinde çalışan veri tabanı uygulamasının yedeklerini sistem kapatılmadan alabilmesi gerekmektedir.

---

### **5. SİSTEM GERÇEKLEŞTİRİMİ**

#### **5.1. Giriş**

Gerçekleştirim çalışması, tasarım sonucu üretilen süreç ve veri tabanının fiziksel yapısını içeren fiziksel modelin bilgisayar ortamında çalışan yazılım biçimine dönüştürülmesi çalışmalarını içermektedir. Yazılımın geliştirilmesi için her şeyden önce belirli bir yazılım geliştirme ortamını olan Web alanında Html destekli visualStudy seçilmesi gerekmektedir.

#### **Resim7: Şemasal Gösterim**

#### **5.2. Yazılım Geliştirme Ortamları**

Yazılım geliştirme ortamı, tasarım sonunda üretilen fiziksel modelin, bilgisayar ortamında çalıştırılabilmesi için gerekli olan araçlar belirlendi ve yazılım geliştirme ortamı hazırlandı. Proje için Python, Html, css, Php ve JavaScript programlama dilleri ile kullanılmak istendiğinden Visual Studio ortamında geliştirilmiştir.

---

### 5.2.1 Programlama Dilleri

Proje Python programlama dili ile yazılmıştır. Python, makine ile haberleşmede insan hayatında kullanılan terimlerle aynı işler yaptırılmaya çalışılır böylelikle kolayca öğrenim ve uygulama şansı olan basit ve geniş bir kütüphaneli bir programlama dilidir. Modern bir dildir. Günümüzün yüksek seviyeli programlama dillerinden en popülerleri arasında yer alır. Yapay zeka yönelimli ve nesneleri destekleyen bir dildir. Güvenli yazım olanağı sağlar. Birçok işlevsel özelliği sayesinde daha anlaşılır ve kolay değiştirilebilir. Python da yalnızca belirlenen bellek konumuna erişilebilir. Bu da programın güvenliğini artırır. Güncelleme imkânı mevcuttur. Bütünü bozmayan küçük parçalar sayesinde anlaşılabilirlik daha kolay. Python derleme ve yürütme süresi hızlı bir programlama dilidir.

### 5.2.2 Veri Tabanı Yönetim Sistemleri

#### 5.2.2.1 VTYS Kullanımının Ek Yararları

#### 5.2.2.2 Veri Modelleri

#### 5.2.2.3 Şemalar

#### 5.2.2.4 VTYS Mimarisi

#### 5.2.2.5 Veritabanı Dilleri ve Arabirimleri

#### 5.2.2.6 Veri Tabanı Sistem Ortamı

#### 5.2.2.7 VTYS'nin Sınıflandırılması

#### 5.2.2.8 Hazır Program Kütüphane Dosyaları

#### 5.2.2.9 CASE Araç ve Ortamları

### 5.3. Kodlama Stili

Düzenli ve kontrollü bir kodlama stili benimsenmiştir.

#### 5.3.1 Açıklama Satırları

Resim 8 de belirtildiği gibi Açıklama satırları her satırın yanına işlevlerini açıklamak için kullanılacaktır.

**Resim8:** Açıklama Satırı Gösterimi

#### 5.3.2 Kod Biçimlemesi

Programın okunabilirliğini artırmak ve anlaşılabilirliğini kolaylaştırmak amacıyla açıklama satırlarının kullanımının yanı sıra, belirli bir kod yazım düzeninin de

---

---

kullanılması gerekmektedir. Projenin kod biçimlemesine değinmek gerekirse alt alta oluşan kodlarda indexler kullanılarak iç içe bir hiyerarşi oluşturuldu.

### **5.3.3 Anlamlı İsimlendirme**

Gerekli kullanımlara göre adlandırmalar yapılmıştır. Kodlayan ekip yapısı belirlemiştir.

### **5.3.4 Yapısal Programlama Yapıları**

- **Ardışık işlem yapıları:** Bu tür yapılarda genellikle fonksiyon, altprogram ve buna benzer tekrarlı yapıları tek bir seferde çözüldü.

## **5.4. Program Karmaşıklığı**

Program karmaşıklığını ölçmek için birçok teorik model geliştirilmiştir. Bu modellerin en eskisi ve yol göstericisi McCabe karmaşıklık ölçütüdür. Bu bölümde bu ölçüt anlatılmaktadır. Söz konusu ölçüt 1976 yılında McCabe tarafından geliştirilmiştir. Bu konuda geliştirilen diğer ölçütlerin çoğu, bu ölçütten esinlenmiştir. McCabe ölçütü, bir programda kullanılan "koşul" deyimlerinin program karmaşıklığını etkileyen en önemli unsur olduğu esasına dayanır ve iki aşamada uygulanır:

- i. Programın Çizge Biçimine Dönüştürülmesi
- ii. McCabe Karmaşıklık Ölçütü Hesaplama

### **5.4.1 Programın Çizge Biçimine Dönüştürülmesi**

### **5.4.2 McCabe Karmaşıklık Ölçütü Hesaplama**

Proje henüz yapım aşamasında olduğundan bu kısımlar final döneminde doldurulup iletilecektir.

## **5.5. Olağan Dışı Durum Çözümleme**

Olağan dışı durum, bir programın çalışmasının, geçersiz ya da yanlış veri oluşumu ya da başka nedenlerle istenmeyen bir biçimde sonlanmasına neden olan durum olarak tanımlanmaktadır. Olağan dışı oluşan bu durumların çözümü de programı yazan programcı ve proje ekibindeki kişiler tarafından kontrol edilebilecektir.

### **5.5.1 Olağandışı Durum Tanımları**

### **5.5.2 Farklı Olağandışı Durum Çözümleme Yaklaşımları**

## **5.6. Kod Gözden Geçirme**

---

---

Bir programın önceki sürümleri gözden geçirilmeden ve incelenmeden okunabilir bir program yazılamaz. Hiçbir program incelenmeden, gözden geçirilmeden işleme alınmamalıdır.

Kod gözden geçirme ile program sına işlemlerini birbirinden ayırmak gerekir. Program sına, programın işlemini sırasında ortaya çıkabilecek yanlışları veya hataları yakalamak amacıyla yapılır. Kod gözden geçirme işlemleri ise, programın kaynak kodu üzerinde yapılan bir incelemedir. Kod gözden geçirmelerinde program hatalarının %3-5 oranındaki kesimi yakalanabilmektedir. Eğer programı yazan kişi, yazdığı programın hemen sonra bir kod inceleme sürecine girdi olacağını bilerek program yazdığında daha etkin, az hatalı ve okunabilir programlar elde edilebilmektedir.

### **5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi**

Gözden geçirme sürecinin amacı;

- Hataların bulunması, ancak düzeltilmemesi hedeflenir.
- Olabildiğince küçük bir grup tarafından yapılmalıdır. En iyi durum deneyimli bir inceleyci kullanılmasıdır. Birden fazla kişi gerektiğinde, bu kişilerin, ileride program bakımı yapacak ekipten seçilmesinde yarar vardır.
- Kalite çalışmalarının bir parçası olarak ele alınmalı ve sonuçlar düzenli ve belirlenen bir biçimde saklanmalıdır. Biçiminde özetlenebilir. Burada yanıtı aranan temel soru, programın yazıldığı gibi çalışıp çalışmayacağını belirlemesidir.

### **5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular**

Program incelenirken, programın her bir işlevi için bir takım sorular sorulmalı ve bu soruların yanıtları aranmalıdır. Bazı soruların yanıtlarının olumsuz olması programın reddedileceği anlamına gelmemelidir.

Programı gözden geçirmek için sorulabilecek bir takım sorular:

1. Program çalışıyor mu?
2. Program gereksinimleri karşılıyor mu?
3. Program doğru çalışıyor mu?
4. Program geçerli mi?

#### **5.6.2.1 Öbek Arayüzü**

#### **5.6.2.2 Giriş Açıklamaları**

#### **5.6.2.3 Veri Kullanımı**



---

#### **5.6.2.4 Öbeğin Düzenlenişi**

#### **5.6.2.5 Sunuş**

---

### **6. DOĞRULAMA VE GEÇERLEME**

---

#### **6.1. Giriş**

#### **6.2. Sınama Kavramları**

#### **6.3. Doğrulama ve Geçerleme Yaşam Döngüsü**

#### **6.4. Sınama Yöntemleri**

##### **6.4.1 Beyaz Kutu Sınaması**

##### **6.4.2 Temel Yollar Sınaması**

#### **6.5.Sınama ve Bütünleştirme Stratejileri**

##### **6.5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme**

##### **6.5.2 Aşağıdan Yukarıya Sınama ve Bütünleştirme**

#### **6.6. Sınama Planlaması**

#### **6.7. Sınama Belirtileri**

#### **6.8. Yaşam Döngüsü Boyunca Sınama Etkinlikleri**

---

### **7. BAKIM**

---

#### **7.1 Giriş**

#### **7.2 Kurulum**

#### **7.3 Yerinde Destek Organizasyonu**

#### **7.4 Yazılım Bakımı**

##### **7.4.1 Tanım**

##### **7.4.2 Bakım Süreç Modeli**

---

### **8. SONUÇ**

---

### **9. KAYNAKLAR**

---