**1. Tanıtım**

**1.1 Projenin amacı**Ülkemizde bireylerin stec ile tanışması çocuk yaşlardadır. Bu etkileşim eğlence odaklı olup bireylerin; kişisel gelişimini ve yeni bilgiler öğrenmelerini aksatmaktadır. Genç kesimin de stec ile etkileşimi kişisel gelişim açısından eksiktir. Stec’larda farklı konular hakkında bilgi edinmek için kullanılabilen bu proje ile bireylerin genel kültürünü ve dolayısı ile kişisel gelişimini olumlu yönde etkilemek hedeflenmektedir.

**1.2 Projenin kapsamı**Kişisel gelişim ve yeni bilgi öğrenmede kullanılacak olan sistem, boş zaman değerlendirmesi için güzel bir fırsat sağlamakta ve yeni bilgiler öğrenmeyi içermektedir.  
Ayrıca değişebilen dünya fotoğrafı bulunan buton ve fotoğraflara göre değişen eğitici sözler sayesinde eğlenceli bir ortam kullanıcıyı beklemektedir.

**1.2 Tanımlar ve kısaltmalar**stec = seyyar taşınabilir elektronik cihaz  
seyyar taşınabilir elektronik cihaz = (telefon, tablet, bilgisayar, masaüstü bilgisayar, … vb.)  
vb. = ve benzeri  
Çocuk birey = (3-11 yaş aralığı)  
Genç birey = (11-25 yaş aralığı)

**2. Proje planı**

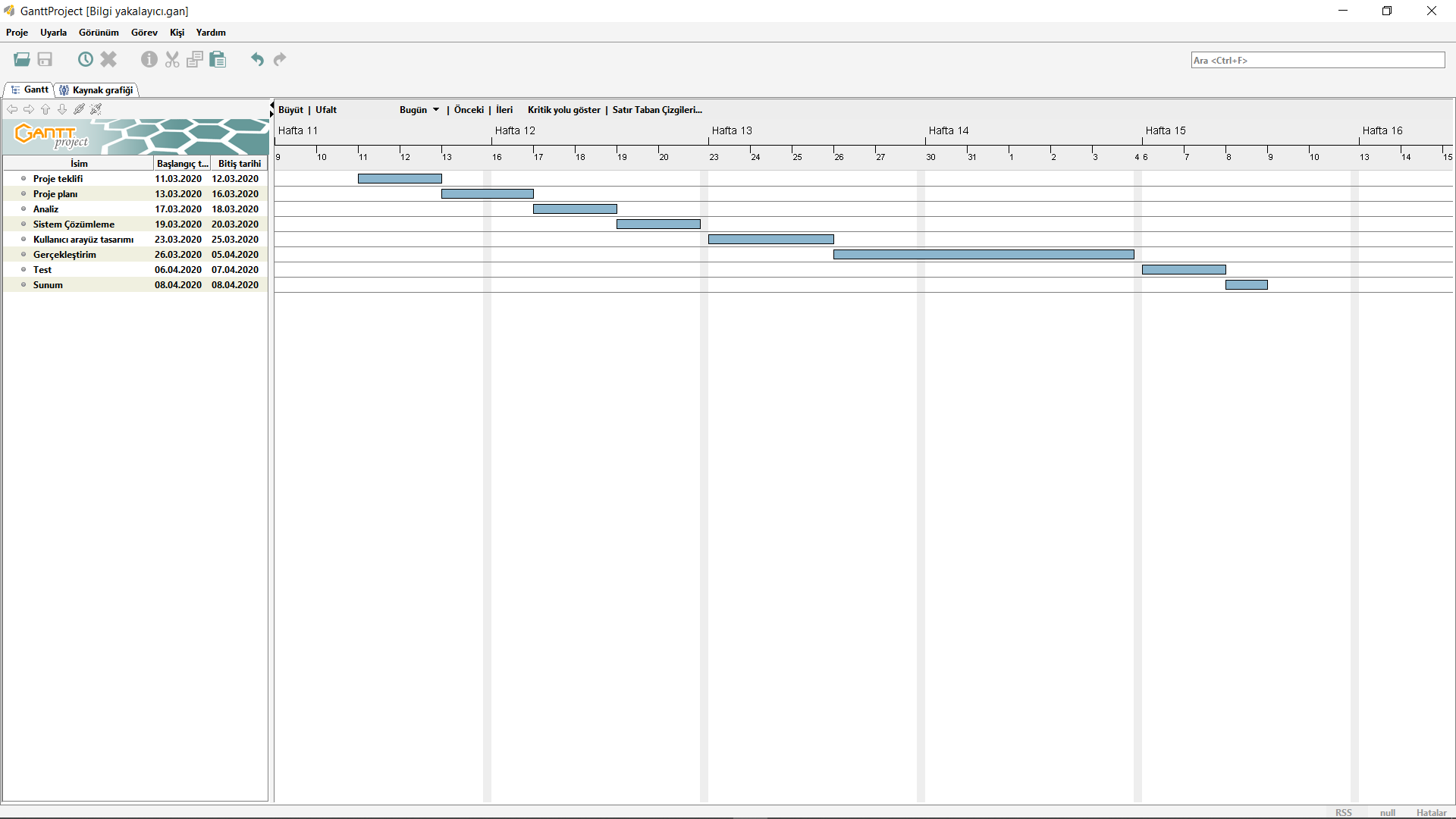
**2.1 Proje plan kapsamı**Geliştirilecek olan sistemde kullanıcıya farklı konular hakkında dikkat çekici bilgiler sunmak hedefleniyor. Kullanıcının arayüz yardımcı butonları sayesinde oluşturulacak menü bar ile üç farklı kategoride bilgi seçme(Havada uçan bilgiler, karada kaçan bilgiler, denize dalan bilgiler), sonraki bilgiye geçme, önceki bilgiye geçme, rastgele bilgi seçme ve seçilen rastgele bilgiden sonraki önceki bilgilere geçiş sağlayabilme, ekranda gösterilen bilgiyi beğenme ya da beğeniyi geri çekme, sağ üst köşede bulunan resimli buton sayesinde kullanıcıya mesaj verme hedeflenmekte son olarak kullanıcının arka plan rengini belirlenen renkler doğrultusunda (Beyaz, cyan, kırmızı, magenta, mavi, pembe, sarı, siyah, yeşil) değiştirebilmesi hedeflenmekte. Fonksiyonlarına sahip olması hedeflenmekte.

|  |  |  |  |
| --- | --- | --- | --- |
| Ölçüm parametresi | Sayı | Ağırlık | Toplam |
| Kullanıcı girdi sayısı | 7 | 3 | 21 |
| Kullanıcı çıktı sayısı | 15 | 4 | 60 |
| Kullanıcı sorgu sayısı | 0 | 3 | 0 |
| Kütük sayısı | 8 | 7 | 56 |
| Dışsal arayüz sayısı | 1 | 5 | 5 |
| Ana işlev nokta sayısı |  |  | 142 |

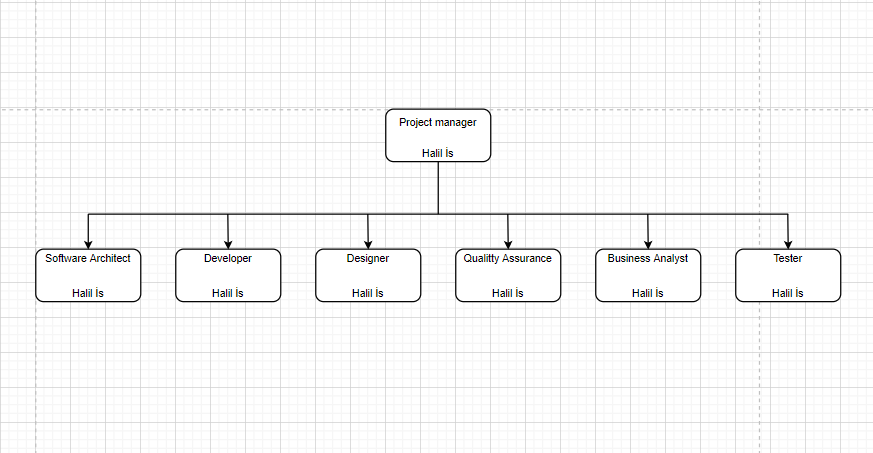
|  |  |
| --- | --- |
| Teknik karmaşıklık sorusu | Puan |
| Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu? | 0 |
| Veri iletişimi gerekiyor mu? | 2 |
| Dağıtık işlem işlevleri var mı? | 2 |
| Performans kritik mi? | 3 |
| Sistem mevcut ve ağır yükü olan bir işletim ortamında mı çalışacak? | 2 |
| Sistem, çevrim içi veri girişi gerektiriyor mu? | 0 |
| Çevrim içi veri girişi, bir ara işlem için birden çok ekran gerektiriyor mu? | 0 |
| Ana kütükler çevrim-içi olarak mı günleniyor? | 0 |
| Girdiler, çıktılar, kütükler ya da sorgular karmaşık mı? | 1 |
| İçsel işlemler karmaşık mı? | 3 |
| Tasarlanacak kod, yeniden kullanılabilir mi olacak? | 2 |
| Dönüştürme ve kurulum, tasarımda dikkate alınacak mı? | 2 |
| Sistem birden çok yerde yerleşik farklı kurumlar için mi geliştiriliyor? | 0 |
| Tasarlanan uygulama, kolay kullanılabilir ve kullanıcı tarafından kolayca değiştirilebilir mi olacak? | 4 |
| TOPLAM | 21 |

0: Hiçbir Etkisi Yok   
1: Çok Az etkisi var   
2: Etkisi Var   
3: Ortalama Etkisi Var   
4: Önemli Etkisi Var   
5: Mutlaka Olmalı, Kaçınılamaz

İN = AİN x (0.65 x 0.01 x TKF)İN = 142 x (0.65 x 0.01 x 21)  
İN = 19.383  
Satır sayısı = İN x 30  
Satır sayısı = 19.383\*30  
Satır sayısı = 581.49  
Etkin Maliyet Modeli – COCOMO  
Proje küçük ekip tarafından geliştirildiği için organik projedir.  
Organik proje: a=2.4, b=1.05, c=2.5, d= 0.38  
Aylık Kişi Başı İş Gücü = E = a x (KSS)^b   
Geliştirme Süresi (Ay) = D = c x (E)d   
Eleman Sayısı = E / D  
  
KSS = Kod Satır Sayısı manasına gelmektedir ve birimi bin satırdır. Projenin tahmini kaç bin satırdan oluşacağını belirtmemizi sağlar.  
  
Aylık Kişi Başı İş Gücü = E = 2.4 x (0.6)^1.05 = 1.4  
Geliştirme Süresi (Ay) = D = 2.5 x (1.4)^0.38 = 2.8 ay  
Eleman Sayısı = E / D = 1.4 / 2.8 = 1 kişi **2.2 Proje zaman-iş planı**

****

**2.3 Proje ekip yapısı**

**  
Project manager:**- Projenin yönetilmesi ve yürütülmesi.  
- Proje ekip yapısının oluşturulması.  
- İş planlamasının yapılması. **Software Architect:**- Projede kullanılacak teknolojinin belirlenmesi.  
- Hız kazandıracak hazır kontrolleri seçmek ve onları takip etmek.  
- Proje metodolojisi ve süreçlerinde tavsiyelerde bulunmak. **Developer:**- Tasarım dökümanının hazırlanması.  
- Sistem arayüzünün tasarlanması.  
- Sistem için gerekli kodların yazılması.  
- Veri tabanının oluşturulması ve sisteme entegre edilmesi.  
**Designer:**-Proje gereksinimlerini tanımlamak için müşterileri ile iletişim kurmak.  
- Bilgisayar yazılımı kullanarak geliştirilen tasarımların sanal modellerini oluşturmak. **Quality Assurance:**- Projenin kalitesinin ölçülmesi.  
- Projenin performansını ölçülmesi. **Business Analyst:**- İş akışı şemasının ve çizelge diyagramının oluşturulması.  
- Tahmini proje bütçesinin hesaplanması. **Tester:**- Projenin gereksinimleri ne ölçüde karşıladığının test edilmesi.  
- Hata potansiyellerini tanımlamak, riskleri değerlendirmek ve yeni yazılım tamamlanmadan önce sorunları çözmek için developer ile işbirliği yapılması.

**2.4 Önerilen sistemin teknik tanımları, özel geliştirme araçları ve ortamları**

**2.4.1 Çözümleme ve tasarım araçları**- Gantt Project  
- draw.io (https://app.diagrams.net)

**2.4.2 Programlama araçları**- JAVA programlama dili  
- MySQL veri tabanı

**2.4.3 Donanım**

|  |  |  |
| --- | --- | --- |
| **Donanım türü** | **Teknik özellikler** | **Kullanım amacı** |
| Masaüstü kişisel bilgisayar | Intel i7, Windows 10 | Verilere erişim |
| Dizüstü kişisel bilgisayar | Intel i7, Windows 10 | Verilere erişim |

**2.4.4 Yazılım**

|  |  |  |
| --- | --- | --- |
| **Yazılım adı** | **Lisans türü** | **Kullanım amacı** |
| İşletim sistemi lisansları | FREE | Bilgisayar yöneticisi |
| Diğer yazılım lisansları | OEM | Çeşitli amaçlara yönelik |
| Kurulum ve entegrasyon | OEM | Sistem kurulumu |
| Diğer | ---- | Çeşitli paket programlar |

**2.5 Proje standartları, yöntem ve metodolojiler**- Uygulama arayüzü bulunmaktadır.  
- Proje masaüstü ve dizüstü bilgisayarlarda çalışacaktır.   
- Uygulama Java programlama dili ve MySQL veri tabanı kullanılarak hazırlanacaktır.  
- Uygulamanın geliştirilmesi için netbeans apache kullanılacaktır.  
- Veri tabanı MySQL olarak tasarlanacaktır.  
- Proje için “Şelale Modeli” kullanılacaktır.  
 **Şelale modeli:  
- İhtiyaç analizi:**Müşteriden alınan detaylı bilgi sayesindeproje taslağı oluşturulur. **- Dizayn:**Programın daha hızlı, etkili, az kodla aynı işlemi yapması hedeflenir. Sonraki aşamalar için kolaylık sağlar. **- Gerçekleştirme:**Planlanan sistemin “Direct implementation method” ile gerçekleştirilmesi hedeflenmektedir. **- Test:**Projenin doğru ve hatasız çalıştığı kontrol edilir.   
Projenin müşterinin isteği doğrultusunda gerçekleştirildiği kontrol edilir. **- Piyasaya sürme:**  
Gerçekleştirilen proje sunucu bilgisayara aktarılır ve kullanıcıya sunulur. **- Bakım ve onarım:**gerçekleştirilen projenin daha efektif çalışması için yeniden düzenlenir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ekip yapısı zaman iş planı** | | | | |
|  | **1. hafta** | **2. hafta** | **3. hafta** | **4.hafta** |
| Project manager | **+** | **+** | **+** | **+** |
| Software Architect | **+** | **+** |  |  |
| Developer | **+** | **+** | **+** | **+** |
| Designer | **+** | **+** |  |  |
| Quality Assurance |  |  | **+** | **+** |
| Business Analyst | **+** |  |  |  |
| Tester |  |  | **+** | **+** |

**2.6 Kalite sağlama planı  
1. Ekonomi:** Ekonomik açıdan yazılımın maliyeti her ne kadar ilk seferde pahalı olsa da ileriye dönük düşünüldüğünde güncelleme masrafları daha düşük olacaktır. **2. Tamlık:** Projede herhangi bir açık yoktur ve programda bulunan tüm butonlar, yazı kutucukları ve menü barlar çalışır durumdadır. **3. Yeniden kullanılabilirlik:** Uygulama her koşulda tekrardan düzenlenip kullanılabilecek.  
**4. Etkinlik:** Kullanıcı sistemin her alanına hakim olduğu için sistemi etkin bir biçimde kullanacak. **5. Güvenilirlik:** Uygulama gerekli güvenlik önlemlerinin alınması yanı sıra şuan devlet bünyesinde bulunan çok yüksek güvenlik önlemli sunucularda saklanacaktır. **6. Belgeleme:** Bu belgeden de anlaşılacağı üzere tam anlamıyla sistemin özeti olacak bu doküman oluşturulmuştur. **7. Kullanılabilirlik:** Kullanılabilirlik olarak her seviyedeki insana hitap edeceğinden zor renkler karmaşık sistemlerden kaçınılmıştır. **8. Genellik:** Proje birçok kişi tarafından kullanılacağı için geneldir. **9. Taşınabilirlik:** Sistem internet üzerinden kullanılacağından herhangi bir özel cihaz gerektirmez ve istenilen cihazlarda taşınabilir ve kullanılabilir. **10. Bütünlük:** Proje belirli bir zaman içerisinde, belirlenen hedefler doğrultusunda gerçekleşmiş ve belirli kaynağı tüketmiştir.

**2.7 Konfigürasyon yönetim planı**Sistemin ilerde kullanıcının yeni istemlerini karşılayamaması veya sistemin yapısındaki bazı bileşenlerin değişmesi sonucu güncelliğini kaybettiğinde olası konfigürasyon planı hazırlandı.   
Konfigürasyon planını devreye sokacak etmenler.  
 - Kullanıcılardan gelen geri dönüşler dahilinde sisteme güncellemeler yapılacaktır.  
 - Sistemin güncelliğini kaybetmesi.  
 - Kullanıcının yeni özellikler istemesi.

**2.8 Kaynak yönetim planı**Donanım kaynaklarının, yazılım ortamı kaynaklarının, işletim sistemi kaynaklarının, insan kaynaklarının ve finansal kaynakların yönetimi için plan oluşturuldu.

**2.9 Eğitim planı**Proje tamamlandıktan 1 hafta sonra uygulamayla ilgilenecek yöneticiye yeni bilgiler ekleyebilmesi için 1 günlük MySQL workbench eğitimi verilecektir. Uygulamanın yapısı gereği eklenecek olan bilgiler uygulamanın işleyişini bozmadan yürütülebilir kalmasının sağlayacaktır.

**2.10 Sınama planı**Proje test ekibi: Halil İs  
Proje gerçekleştirme kısmından sonra gelecek olan test kısmında yapılacaklar:  
- Hata ayıklama.  
- Projenin müşterinin isteği doğrultusunda gerçekleşip gerçekleşmediğinin kontrolü.  
- Sistemin efektif şekilde çalışması için yapılabilecek değişiklikler.

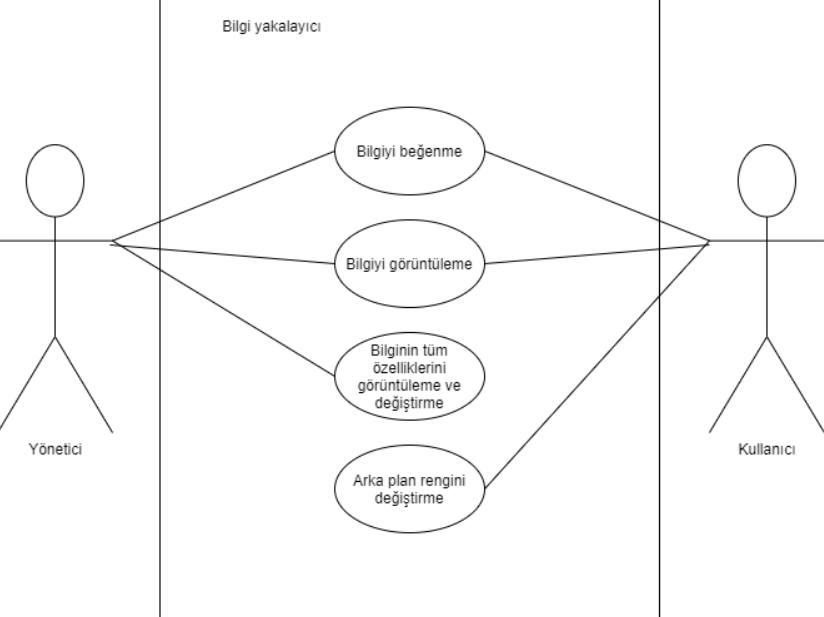
**2.11 Bakım planı**Proje bakım ekibi: Halil İs  
Proje test kısmından sonra gelecek olan bakım kısmında yapılacaklar:  
- Yeni ve güncel bilgiler ekleme veya var olan bilgiyi değiştirme.  
- Kullanıcıdan gelecek olan talepler doğrultusunda uygulamayı güncel tutma.  
- Hatalı çalışan yerlerin düzeltilmesi.

**3. Sistem çözümleme**

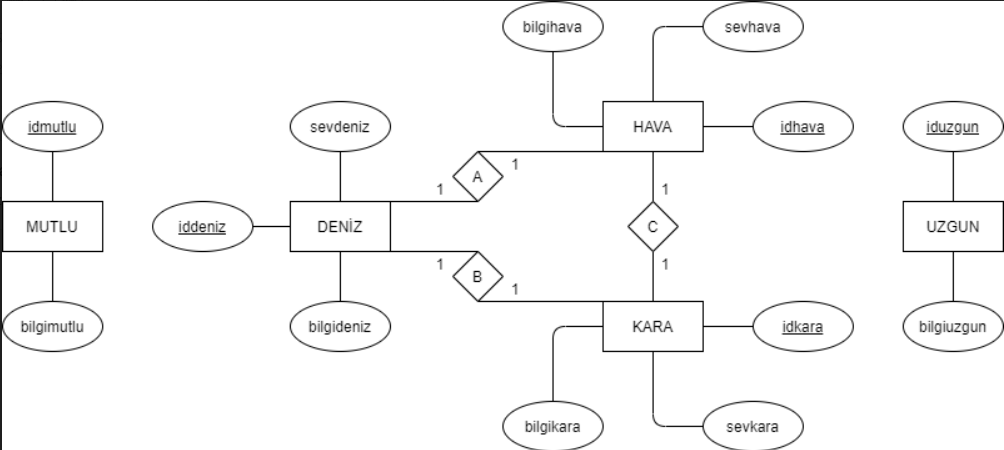
**3.1 Önerilen sistem incelenmesi ve mantıksal modeli**

**3.1.1 Örgüt yapısı**Fırat üniversitesi yazılım mühendisliği temelleri dersi alan öğrenciler bünyesinde bir örgüt yapısı vardır.

**3.1.2 İşlevsel model**

**  
Temel görevlendirme:  
Yönetici:**Sisteme yeni bilgiler ekler.  
Sistemde var olan bilgileri değiştirir. **Kullanıcı:**Uygulamayı yeni bilgi öğrenme amaçlı kullanan bireyler.

**3.1.3 Veri modeli**

****Bir denize dalan bilgi aynı zamanda bir havada uçan bilgi ya da karada kaçan bilgi olabilir.  
Bir havada uçan bilgi aynı zamanda bir denize dalan bilgi ya da karada kaçan bilgi olabilir.  
Bir karada kaçan bilgi aynı zamanda bir havada uçan bilgi ya da denize dalan bilgi olabilir.  
Mutlu dünya fotoğrafının ekrana getireceği bilgi ile üzgün dünya fotoğrafının ekrana getireceği bilgi arasında ortak nokta yoktur.

**3.1.4 Yazılım/Donanım kaynakları  
Yazılım kaynakları:**  
Netbeans apache,  
MySQL workbench,  
draw.io(<https://app.diagrams.net/>).  
**Donanım kaynakları:**Geliştirici bilgisayarı,Yönetici bilgisayarı,  
Kullanıcı bilgisayarı,  
Uygulamanın saklanacağı sunucu bilgisayar.

**3.1.5 Sistemin değerlendirilmesi**Uygulamayı yükleyen kullanıcılar üç farklı kategoride yeni ve dikkat çekici bilgiler öğrenecek. Ayrıca sağ üst köşede bulunan dünya fotoğrafına tıklayıp kendini geliştirecek.

**3.1.6 Genel bakış**Kullanıcı sisteme herhangi bir müdahalede bulunamıyor ve bilgileri arayüzün izin verdiği kadar görüntüleyebiliyor bunun amacı kullanıcıyı sistemden uzak tutup olası hataları minimuma indirmek. Bunun yanı sıra yöneticinin her şeyi değiştirebilme yetkisi var olası hatalardan kaçınmak için yöneticiye eğitim verilecektir.

**3.1.7 Bilgi sistemleri/nesneler  
Yönetici:**  
Verilen eğitim neticesinde uygulamanın işleyişini iyi kavramış olan yönetici sistemdeki bilgileri güncel tutmakla sorumludur. **Kullanıcı:**  
Uygulama ara yüzünün izin verdiği ölçüde(Bilgi seçme, bilgiyi beğenme, bilgiden beğeniyi geri çekme, başka bilgiye geçme, arka plan rengini değiştirme)işlem yapacak olan kişidir.

**3.1.8 Veri sözlüğü  
iddeniz:** \* 1 den başlayıp artan tam sayılardan oluşur \* **bilgideniz: \***Metinsel ifadeden oluşur**\*  
sevdeniz:** \*Sadece bir veya sıfır değerini alabilen tam sayıdır\* **idhava:** \* 1 den başlayıp artan tam sayılardan oluşur \* **bilgihava: \***Metinsel ifadeden oluşur**\*  
sevhava:** \*Sadece bir veya sıfır değerini alabilen tam sayıdır\* **idkara:** \* 1 den başlayıp artan tam sayılardan oluşur \* **bilgikara: \***Metinsel ifadeden oluşur**\*  
sevkara:** \*Sadece bir veya sıfır değerini alabilen tam sayıdır\* **idmutku:** \* 1 den başlayıp artan tam sayılardan oluşur \* **bilgimutlu: \***Metinsel ifadeden oluşur**\*  
iduzgun:** \* 1 den başlayıp artan tam sayılardan oluşur \* **bilgiuzgun: \***Metinsel ifadeden oluşur**\***

**3.1.9 İşlevlerin sıradüzeni**Yönetici sisteme yeni bilgi ekler.  
Bilgiyi görüntülemek isteyen kullanıcı;  
uygulamayı açar,  
bilgi kategorisini seçer,  
Sonraki, önceki, rastgele butonlarını kullanıp bilgiye erişebilir.

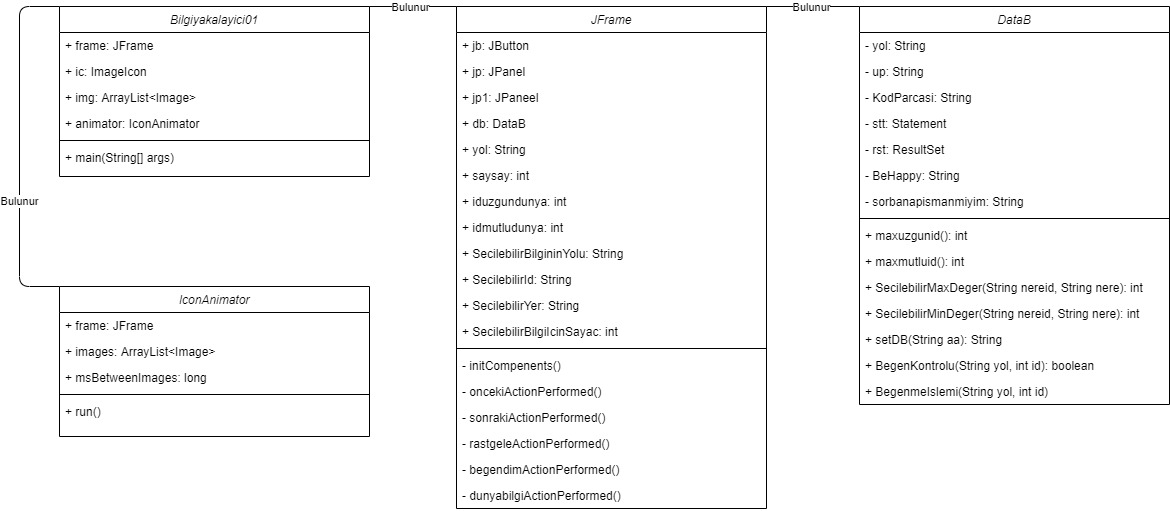
**3.1.10 Başarım gerekleri  
Sistemin temel başarım gerekleri:**- Sistemin sonuç üretim doğrulukları,  
- Tepki sürelerinin en aza indirilmesi ,  
- Mali külfetin azaltılması ,  
- Hile hata ve yanlışlıkların en aza indirilmesi,  
- Kullanım kolaylığı,  
- Anlaşılabilirlik,  
- Tarafsızlık.

**3.2 Arayüz gerekleri**

**3.2.1 Yazılım arayüzü**Projenin çalışması esnasında bir açık verilmemesine özen gösterildi. Gerekli olan her türlü değişiklik kaynak kodları üzerinden yapılıp tekrar derlenecek.

**3.2.2 Kullanıcı arayüzü**Projede kullanıcının arayüzü tasarlanırken ilk açılan ekrandaki renk altın olarak ayarlanmış ve arka plan için kullanıcıya dokuz farklı renk seçeneği sunulmuştur (Beyaz, cyan, kırmızı, magenta, mavi, pembe, sarı, siyah, yeşil).

**3.2.3 Yönetici arayüzü**MySQL workbench

**3.2.4 Bilgi sistemleri ve nesneler** **Sınıfların kullanım amacı ve nasıl kullanıldığı  
DataB:**Veri tabanı işlemleri için oluşturulan bu sınıfta kullanıcının seçtiği kategorideki bilgiye erişim sağlanır ve bu sınıf JFrame sınıfı tarafından kullanılır. **JFrame:**kullanıcının sistemi kullanabilmesi için oluşturulmuştur ve görsel içeriği bulunan tek sınıftır maliyet hesaplamasına katılmıştır. Bilgiyakalayici01 sınıfı tarafından çağırılır. **IconAnimator:**Uygulamanın ikonunu gif yapmak için oluşturulmuştur. Bilgiyakalayici01 sınıfı tarafından kullanılır. Sonradan eklenen bir özelliktir. **Bilgiyakalayici01:**Ana fonksiyonun bulunduğu sınıftır amacı bütün sistemi birleştirip uygulamayı işler hale getirmektir.

**3.3 Belgeleme gerekleri**

**3.3.1 Geliştirme sürecinin belgelenmesi**Geliştirme sürecinde genel olarak belgelendirilmesi hem ileriye dönük hem de şimdiki geliştirme sürecinde projenin tamamlanma yüzdesini nerede kalınıp nerelerde eksikler olduğunu genel hatlarıyla göstermesi amacıyla yapıldı. Bunun yanı sıra projeye yeni dahil olan personellerin olaya hakimiyeti açısından bu yönteme başvuruldu.

**3.3.2 Eğitim belgeleri**Yöneticiye basitçe MySQL workbench kullanımı öğretilecektir.

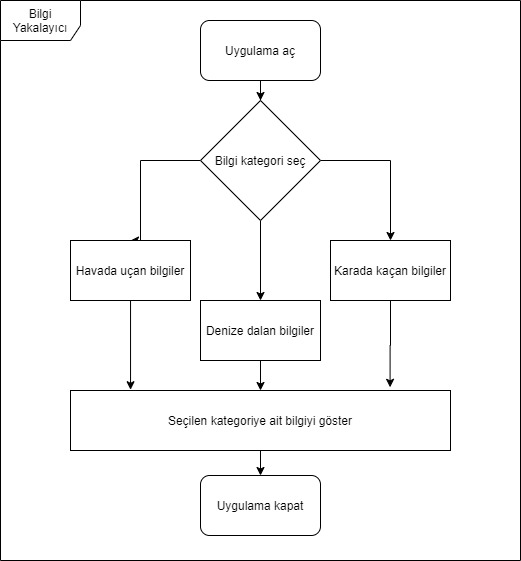
**3.3.3 Kullanıcı el kitapları**Kullanıcı el kitapları projenin en son safhasında kullanıcılara verilecek pilot uygulamalardan yola çıkılarak hazırlanacak. Yani proje sonunda rahat ve kolay kullanımdan dolayı bir eğitim semineri ve bir kullanım kitapçığı hazırlanacaktır.

**4. Sistem tasarımı**

**4.1 Genel tasarım bilgileri**

**4.1.1 Genel sistem tanımı  
Gereksinimler**3 Ayrı ilkokulda yapılacak olan anket ile gereksinimler belirlenecektir. Unutulmamalıdır ki müşteri ile aynı fikri paylaşmayabiliriz onlarla aynı şekilde düşünmeyebiliriz **İşlevsel Belirtimler**Sistem kullanıcının seçtiği kategorideki bilgileri arayüz aracılığı ile kullanıcıya sunacak **Tasarım  
Arayüz Tasarımı**Sistemin arayüzü ihtiyaçlar doğrultusunda tasarlanacak **Yapısal Tasarım**Sistemin yapısı arayüz doğrultusunda tasarlanacak **Veri Tasarımı**Sistemin veri tabanı yapısal tasarım doğrultusunda tasarlanacak

**4.1.2 Varsayımlar ve Kısıtlamalar**Sistemde varsayılan değerler bulunmamakla birlikte varsayılan kısıtlama da yoktur.

**4.1.3 Sistem Mimarisi  
  
Kullanıcı uygulamayı açar:**Kullanıcı uygulama ikonuna basıp uygulamayı başlatır.  
Dilerse belirlenen renkler sınırında arka plan rengini değiştirir. (Beyaz, cyan, kırmızı, magenta, mavi, pembe, sarı, siyah, yeşil)   
Ekranda bulunan dünya butonuna tıklayıp farklı konular hakkında bilgilenebilir. **Bir bilgi kategorisi seçer:**Bilgi kategorileri karada kaçan bilgiler, havada uçan bilgiler ve denize dalan bilgiler olarak belirlenmiştir.  
karada kaçan bilgiler: ağırlıklı olarak kara ortamı hakkında bilgiler verir.  
havada uçan bilgiler: ağırlıklı olarak hava ortamı hakkında bilgiler verir.  
denize dalan bilgiler: ağırlıklı olarak deniz ortamı hakkında bilgiler verir. **Seçilen bilgi:**Kullanıcı, seçilen kategoriye göre ekranda gözükecek olan bilgiyi beğenebilir ve beğeniyi geri çekebilir ayrıca sonraki, önceki ve rastgele bilgilere geçiş yapabilir. **Kullanıcı uygulamayı kapatır:**kullanıcı sağ üst köşede bulunan X tuşuna basıp uygulamayı kapatır.

**4.1.4 Dış Arabirimler**

**4.1.4.1 Kullanıcı Arabirimleri**Kullanıcı arabirimlerinin başında sistem arayüzü bulunacak ve Kullanıcıya bilgileri gösterme görevini üstlenecektir bu yüzden performansının yüksek olması gerekmektedir.

**4.1.4.2 Veri Arabirimleri**Sistem yöneticilerinin kullanacağı MySQL workbench veri arabirimlerini oluşturmakta ve bilgileri depolama görevini üstlenecektir. Kullanıcı arayüzünün performansı yüksek olduğundan, veri arabirimi ile bağlantısından ötürü veri arabiriminin de performansı yüksek olacaktır.

**4.1.4.3 Diğer Sistemlerle Arabirimler**Bu arabirim kullanılmayacaktır.

**4.1.5 Veri Modeli**beş farklı veri tablosu bulunmakta;  
Bunlardan iki tanesi arayüzde kullanılan dünya fotoğrafının durumuna göre bilgi içermekte  
Geri kalan üçü bilgi veren tablolardır.

**4.1.6 Testler**Genel hatlarıyla testler beta testleri olarak gerçekleştirilecek.  
Beta Testleri: Kullanıcı, geliştirilen sistemi kendi yerleşkesinde geri bildirimde bulunacak.

**4.1.7 Performans**Sistemin performansını etkileyen faktörlerin test verileri değerlendirilecek.  
Sistemin stabilitesi ve işleyiş performansı değerlendirilecek.  
Veri yapısının stabilitesi ve çalışma zamanındaki uyumluluk düzeyindeki performansı değerlendirilecek.

**4.2 Veri tasarımı**

**4.2.1 Tablo Tanımları**beş farklı veri tablosu bulunmakta;  
Bunlardan iki tanesi arayüzde kullanılan dünya fotoğrafının durumuna göre bilgi içermekte ve beğenme fonksiyonu bulunmamaktadır.  
Geri kalan üçü bilgi veren tablolardır.  
tablo içerikleri: bilgiid, bilgi, beğendimi,

**4.2.2 Veri Tanımları**Tamsayı (int) veri tipi kullanılmasının nedeni sayısal içerikli veriyi tutmaktır.  
Karakter dizisi (String) veri tipi kullanılmasının nedeni sayısal içerikli veriyi tutmaktır.

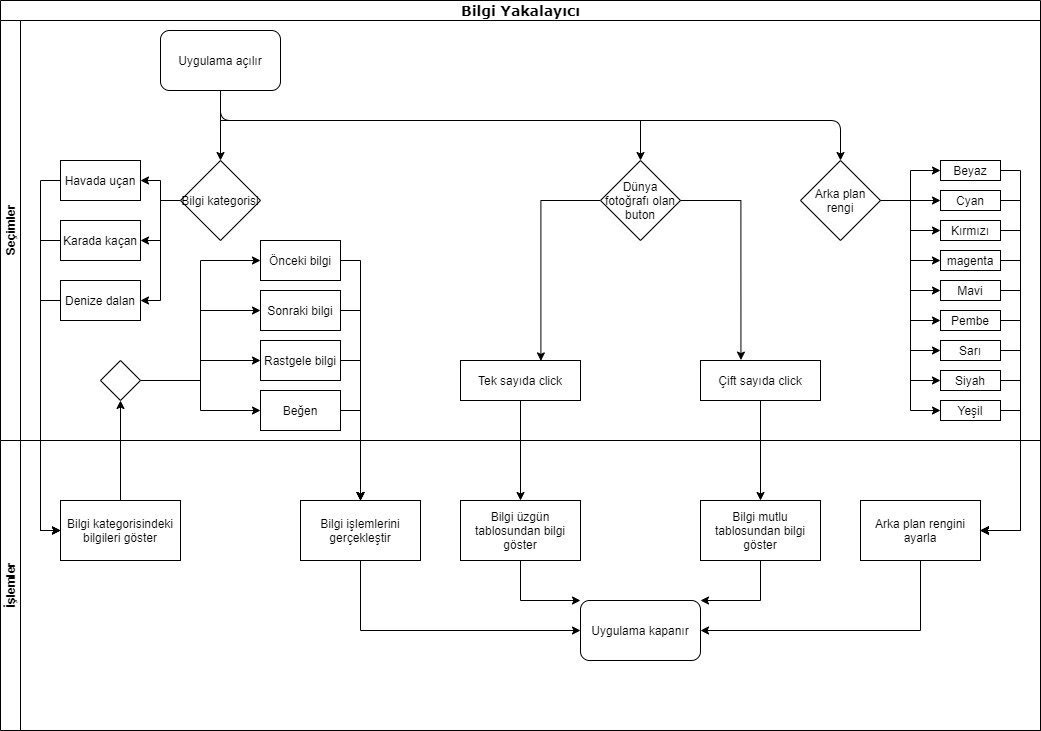
**4.2.3 Değer Kümesi Tanımları  
Bilgi:** İnsan aklının erebileceği olgu, gerçek ve ilkelerin bütününe verilen ad.

**4.3 Süreç tasarımı**

**4.3.1 Genel Tasarım**Sistem inşa edilirken sırasıyla gerçekleştirilecek olan işlemler:  
i. Kullanıcı arayüzü oluşturulacak  
ii. Sistem merkezi oluşturulacak  
iii. Veri tabanı sınıfı oluşturulacak

**4.3.2 Modüller**

**4.3.2.1 Kullanıcı Modülü**

**4.3.2.1.1 İşlev**

**4.3.2.1.2 Kullanıcı Arabirimi**Kullanıcının işlemlerini gerçekleştireceği arayüzdür.

**4.3.2.1.3 Modül Tanımı**Kullanıcıların uygulamayı kullanırken işlemlerini yapacakları bilgi platformudur.

**4.3.3 Kullanıcı Profilleri  
Yönetici:** Kullanıcının görmesi için sisteme bilgi ekler. **Kullanıcı:** Sistemi kullanıp bilgilere erişim sağlar.

**4.3.4 Entegrasyon ve Test Gereksinimleri**Sistemin entegre olacağı herhangi bir platform bulunmamakla birlikte test işlemleri sistem piyasaya sürüldükten sonra kullanıcı geri dönüşleri neticesinde yapılacaktır.

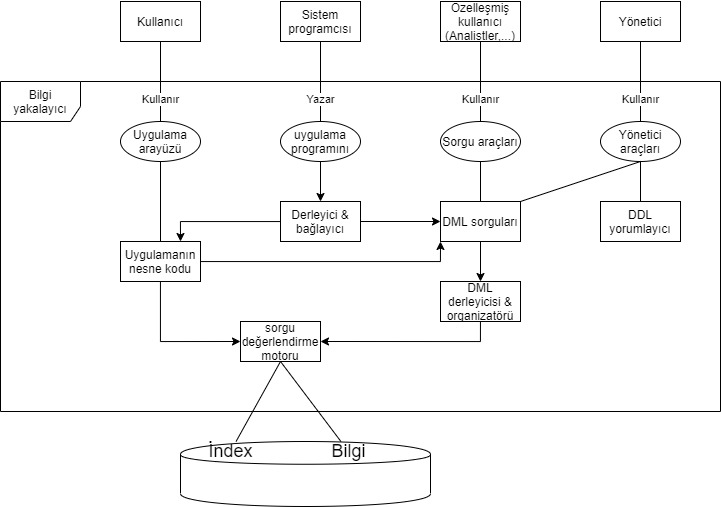
**4.4 Ortak alt sistemlerin tasarımı**Sistemde kullanılacak ortak alt sistem bulunmamaktadır.

**5. Sistem gerçekleştirimi**

**5.1 Giriş**Gerçekleştirim aşaması; planlama, tasarım ve çözümleme aşamalarından sonra gelen aşamadır. Bu Aşamada Tasarlanan arayüzün ve veri tabanı modelinin bilgisayar ortamında çalışan yazılım haline dönüştürülmesi aşamalarını içerir. Sistemin geliştirilmesi için belirli bir yazılım geliştirme ortamı seçilmesi gerekmektedir.

**5.2 Yazılım Geliştirme Ortamları**Yazılım geliştirme ortamı, tasarım sonunda üretilen fiziksel modelin, bilgisayar ortamında çalışması için gerekli olan:  
- Programlama dili  
- Veri yönetim sistemleri  
- Hazır program kitapçıkları  
CASE araçları belirlendi ve yazılım geliştirme ortamı hazırlandı.

**5.2.1 Programlama Dilleri**Sistemde kullanılmak için belirlenen programlama dilleri:  
- JAVA kullanılmasının amacı çok fonksiyonlu platformdan bağımsız güçlü bir programlama dili ve projenin gereklerine karşılık verebilecek kapasitede olması.  
- SQL kullanılmasının amacı veri tabanı işlemleri için gerekli olması ve aynı şekilde beklentiye karşılık verebilmesi.

**5.2.2 Veri Tabanı Yönetim Sistemleri  
**Kullanıcı, arayüzü aracılığıyla veri tabanına erişen aktör oluyor.  
JAVA programlama dili ile veriye erişim sağlanıyor.  
Veri tabanı tanımları ve depolanmış veri tabanı ise SQL ile çözümleniyor.

**5.2.2.1 VTYS Kullanımının Ek Yararları**- Genişleme potansiyeli  
- Esneklik  
- Uygulama geliştirme zamanın azalması  
- Güncel bilgilerin tüm kullanıcılara aynı zamanda ulaşması  
- Ölçüde ekonomi  
- İşletme ortamındaki ortak verilerin tekrarının önlenmesi, verilerin merkezi denetiminin ve tutarlılığının sağlanması  
- yazılımı geliştirmenin kolaylaşması

**5.2.2.2 Veri Modelleri**Üçlü şema mimarisinde görülen yapıların kullanıcı gereksinimlerinden yola çıkılarak aşamalı bir şekilde fiziksel olarak gerçekleştirilmesidir.  
**1. Gereksinimlerin belirlenmesi**- Veri tipleri  
- Veri grupları  
- veriler ile ilgili kurallar  
- Veriler üzerinde yapılması gereken işlemler **2. Kavramsal model**Kullanıcılardan elde edilecek gereksinimler ile ilgili analiz yapılması ve birbiri ile bağlantılı verilerin gruplanarak bir düzenleme içinde modellenmesi gerekir.  **3. Mantıksal model**Veri tabanı tasarımlarının ilişkisel veri tabanı modelinde tablolar ile ifade edilebilmesi için yapılması gereken dönüşümü içerir. **4. Fiziksel model**Fiziksel olarak sistem kurulur. VTYS ile ilk temas burada kurulur.

**5.2.2.3 Şemalar**Veri tabanı tanımlamaları veri tabanı şeması veya meta-veri olarak adlandırılır. Veri tabanı şemaları tasarım aşamasında belirlenip gerekmedikçe değişmez. Diyagramlar her kayıt tipinin yapısını gösterir fakat kaydın gerçek örneğini göstermez.

**5.2.2.4 VTYS Mimarisi   
i. İçsel düzey**Veri tabanının fiziksel saklanma yapısını açıklar. Fiziksel veri modeli kullanır ve veri tabanına erişim yolu ile veri saklamanın tüm detaylarını açıklar. **ii. Kavramsal düzey**Kavramsal şema içerir ve kullanıcılar için veri tabanının yapısını açıklar. Gerçek fiziksel yapının detaylarını kullanıcıdan gizler, veri tipleri, varlıklar, ilişkiler, kullanıcı işlemleri ve sınırlamalar üzerine odaklanır.  **iii. Dışsal düzey**Dış şemalar ve kullanıcı görüşleri içerir. Her dış şema veri tabanının bir bölümünü açıklar. Her gruba sadece ilgilendiği görüşü sunar.

**5.2.2.5 Veri Tabanı Dilleri ve Arabirimleri**Sistemde kullanılmak üzere; SQL dili, veri tabanı programlama için sıklıkla kullanılan bir dil olduğundan seçildi.  
Sistem arabirimi için MySQL seçilmesinin amacı çok fonksiyonlu bir arabirim olması.

**5.2.2.6 Veri Tabanı Sistem Ortamları**Sistem veri tabanı için bir ortam seçilmedi.

**5.2.2.7 VTYS’nin Sınıflandırılması**Çok kullanıcılı ilişkisel veri modeli kullanılmıştır.

**5.2.2.8 Hazır Program Kütüphane Dosyaları**Böyle bir işleme ihtiyaç duyulmamıştır.

**5.2.2.9 CASE Araç ve Ortamları**Case aracı olarak bulut platformu olan draw.io kullanıldı. Draw.io kullanılmasındaki amaç hem fonksiyonel olması hem de kurulumunu yapmadan hızlıca diyagramları çizebilmeyi sağlamsı.

**5.3 Kodlama Stili**Herhangi bir hazır düzene bağlı kalmadan kodlama yapıldı. Bakım geliştiricisine de aynı tarz kodlamada verilen eğitimler sayesinde olası anlaşılmazlık sorunu çözülmüş oldu.

**5.3.1 Açıklama Satırları**Anlaşılması karmaşık olan her satırdan önce açıklama satırı eklendi.

**5.3.2 Kod Biçimlemesi**İç içe biçimde hiyerarşi oluşturulmuştur.

**5.3.3 Anlamlı İsimlendirme**Değişkenler görevlerine göre isimlendirildi. Örneğin sistem ikonunun yerini gösteren değişken “yol” olarak isimlendirildi.

**5.3.4 Yapısal Programlama Yapıları  
Ardışık işlem yapıları:**Fonksiyon gibi yapılar tek seferde çözüldü. **Koşullu işlem yapıları:**Kontrol yapıları, karşılaştırma yapıları başta olmak üzere birçok yerde kullanılmıştır. **Döngü yapıları:**Ardışık işlem gerektiren yapılarda kullanılmıştır.

**5.4 Program Karmaşıklığı**Program karmaşıklığını ölçmek için Mc Cabe karmaşıklık ölçütü kullanılmıştır.  
1. Programın çizge biçimine dönüştürülmesi.  
2. Mc Cabe karmaşıklık ölçütünün hesaplanması.

**5.5 Olağan Dışı Durum Çözümleme**Sitem gerçekleştirilirken olağan dışı durumların büyük bir kısmı hesaplanmıştır ve mevcut sistemde olağan dışı durum oluşma olasılığı en aza indirilmiştir.

**5.5.1 Olağan Dışı Durum Tanımları**Sistem, try-catch blokları yardımıyla olağan dışı durum karşısında işlevselliğini sürdürecek.

**5.5.2 Farklı Olağan Dışı Durum Çözümleme Yaklaşımları**Olağan dışı durum karşısında sistem durumu görmezden gelip çalışmaya devam edecek.

**5.6 Kod Gözden Geçirme**

**5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi**- Hataların bulunup düzeltilmesi hedeflenir.  
- Kalite sürecinin bir parçası olarak ele alınır.

**5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular**Bir sistem incelenirken aşağıdaki sorulara yanıt aranır.

**5.6.2.1 Öbek Arayüzü**Oluşturulan öbekler için aşağıdaki sorular belirlendi.  
- Her öbek tek bir işlevsel amacı yerine getiriyor mu?  
- Öbek adı işlevini açıklar nitelikte anlamlı olarak verilmiş mi?  
- Öbek tek giriş ve tek çıkış mı?  
- Öbek eğer bir işlev ise parametrelerinin değerini değiştiriyor mu?

**5.6.2.2 Giriş Açıklamaları**Giriş açıklamalarını test etmek için belirli sorular:  
- Öbek doğru biçimde giriş açıklama satırı içeriyor mu?  
- Giriş açıklama satırları, öbeğin amacını yansıtıyor mu?  
- Giriş açıklama satırları, çıktıları ve hata iletilerini tanımlıyor mu?  
- Giriş açıklama satırları, öbeğin algoritma tanımını içeriyor mu?  
- Giriş açıklama satırları, öbekte yapılan değişikliklere ilişkin tanımlamaları içeriyor mu?  
- Giriş açıklama satırları, öbekteki olağan dışı durumları tanımlıyor mu?   
- Giriş açıklama satırları, öbeği yazan kişi ve yazdığı tarih hakkında bilgi içeriyor mu?  
- Her paragrafı açıklayan kısa açıklamalar var mı?

**5.6.2.3 Veri Kullanımı**Veri kullanımlarını test etmek için belirli sorular.  
- İşlevsel olarak ilintili bulunan veri elemanları uygun bir mantıksal veri yapısı içinde gruplanmış mı?  
- Değişken adları işlevlerini yansıtabiliyor mu?  
- Her değişken tek bir amaçla mı kullanılıyor?  
- Dizin değişkenleri kullanıldıkları dizinin sınırları içerisinde mi tanımlanmış?  
- Tanımlanan her gösterge için bellek ataması yapılmış mı?

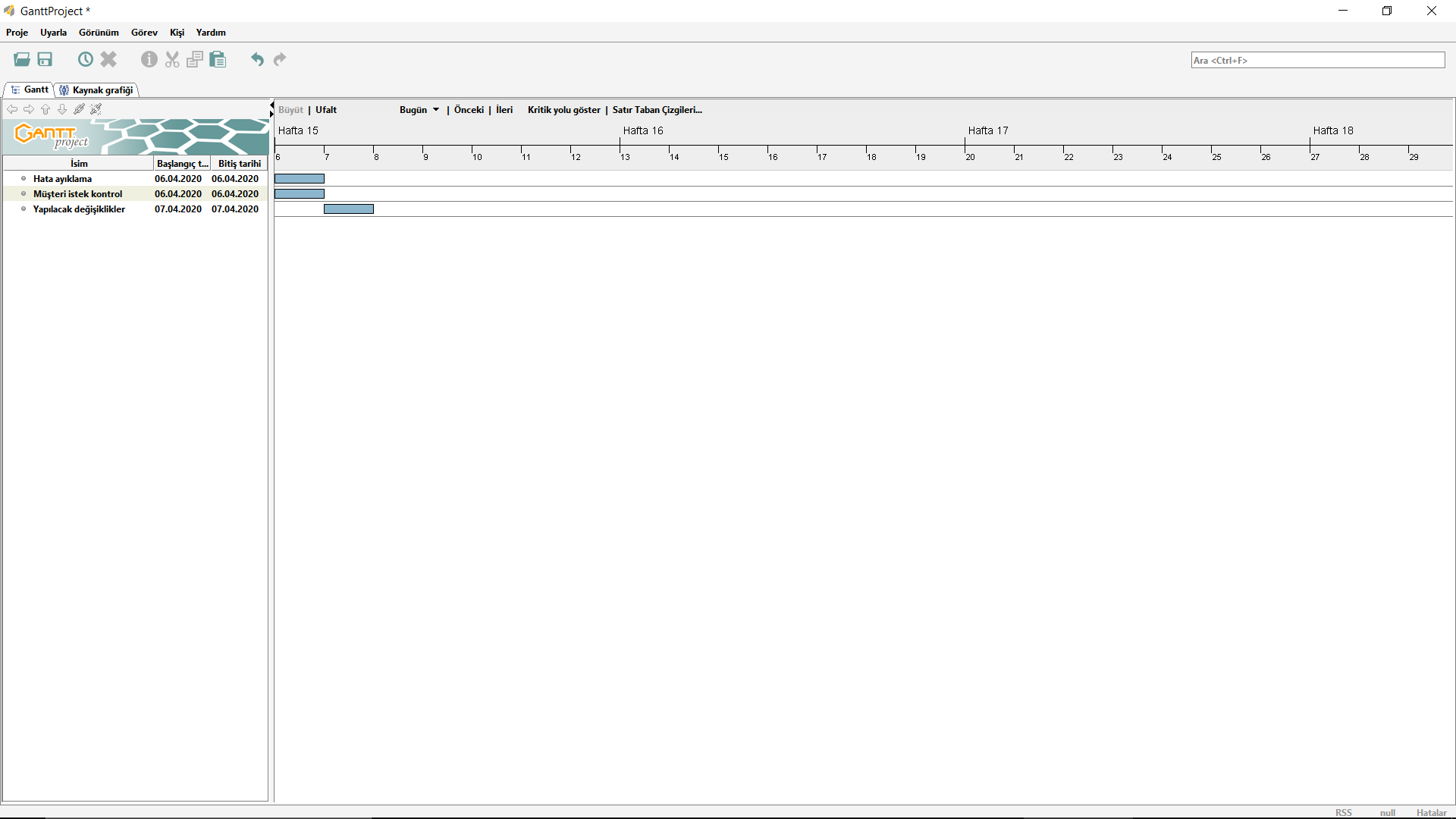
**5.6.2.4 Öbeğin Düzenlenişi**- Modüller birleşimi uyumlu mu?  
- Modüller arası veri aktarımları sağlanıyor mu?  
- Bütün modüller birleştiğinde sistem çalışıyor mu?  
gözden geçirmede referans alınacak sorulardır.

**5.6.2.5 Sunuş**Son kısımda sorulan sorular:  
- Her satır en fazla bir deyim içeriyor mu?  
- Bir deyimin birden fazla satıra taşması durumunda bölünme okunabilirliği kolaylaştıracak biçimde anlamlı mı?  
- Koşullu deyimlerde kullanılan mantıksal işlemler yallın mı?  
- Bütün deyimlerde karmaşıklığı azaltacak parantezler kullanıldı mı?  
- Öbek yapısı içerisinde akıllı programlama hileleri yapılmış mı?  
gibidir.

**6. Doğrulama ve geçerleme (Test)**

**6.1 Giriş**Bu aşamada gerçekleştirilen işlemler iki soru ile özetlenebilir.  
Doğru ürünü mü üretiyoruz? **Doğrulama**  
Ürünü doğru olarak mı üretiyoruz? **Geçerleme  
Doğrulama:** Ürünü kullanacak kişilerin isteklerinin karşılanıp karşılanmadığına dair etkinliklerden oluşur. **Geçerleme:** Ürünün içsel niteliğine ilişkin izleme ve denetim etkinliklerinden oluşur.

**6.2 Sınama Kavramları  
Birim Sınama:**   
Sistemde sınanacak birim bulunmamaktadır. **Alt sistem sınama:**Sistemde sınanacak alt sistem bulunmamaktadır. **Sistem sınama:**Sadece kendini içeren sistemin sınaması yapıldı ve sistemin eksiksiz olduğu onaylandı. **Kabul sınama:**Sistem prototipten çıkartılıp gerçek veriler girildi ve sorunsuz olduğu bir kez daha onaylandı.

**6.3 Doğrulama ve Geçerleme Zaman-İş planı**

**6.4 Sınama Yöntemleri**Sınama işlemi, geliştirmeyi izleyen bir düzeltme görevi olmak ile sınırlı değildir. Bir "sonra" operasyonu olmaktan çok, geliştirme öncesinde planlanan ve tasarımı yapılması gereken bir çaba türüdür.

**6.4.1 Beyaz Kutu Sınaması**- Bağımsız yollar en az bir kere denendi.  
- Bütün mantıksal karar noktalarında iki değişik karar için sınamalar yapıldı.  
- Bütün döngüler sınır değerlerinde sınandı.  
- İç veri yapılarının denenmesi yapıldı.

**6.4.2 Temel Yollar Sınaması**Sistemin tümüne yönelik işlevlerin doğru yürütüldüğü testtir. Şartname gerekleri incelenir.  
- Eş değerlerde bölme  
- Uç değerler analizi  
- Karar tablosu  
- Sonlu durum makinesi  
- Belgelenmiş özelliklere göre test  
- Rastgele test  
- Kullanım profili

**6.5Aşağıdan Yukarıya Sınama ve Bütünleştirme**Sistem sınanırken aşağıdan yukarıya sınama tekniği kullanılmıştır. Yani önce alt düzey birimler sınanmış sonra üstler sınanmıştır. Bu durum sınama yapmayı kolaylaştırdığı için seçilmiştir.

**6.6 Sınama Planlaması**Test raporu hazırlanırken belirtilecek özellikler; **Test tanı kimliği:** Test planının adı veya belge numarası **Giriş:** Test edilecek yazılımın elemanlarının genel tanıtım özetleri. Ayrıca bu plan kapsamı ve başvurulan belgeler. Kısaltmalar ve terim açıklamaları bu bölümde bildirilmelidir. **Test edilecek sistem:** Sistemde bileşenleri sürüm sayıları olarak sıralar ve sistemin özelliklerini bileşenlerini ve nasıl kullanıldıkları açıklanmalıdır. Ayrıca sistemde test edilmeyecek parçalar belirtilmelidir. **Test edilecek ana fonksiyonlar:** Sistemin test edilecek ana fonksiyonlarının kısa bir tanıtımı yapılmalıdır. **Test edilmeyecek ana fonksiyonlar:** Sistemde test edilmeyecek fonksiyonları ve bunların neden test edilmedikleri açıklanacaktır. **Geçti/Kaldı kriteri:** Bir test sonucunda sistemin geçmiş veya kalmış sayılacağını açıklanmalıdır. **Test dökümanı:** Test süresince yapılan işlemleri alınan raporları elde edilen bilgileri rapor içinde sunulmalıdır. **Sorumluluklar:** Hangi kişilerin nelerden sorumlu olduğu ve test takım lideri bilgileri mutlaka raporda belirtilmelidir. **Riskler ve önlemler:** Test planında varsayılan ve olası yüksek riskli durumları belirtir ve bu durumların olması durumunda, etkilerinin en aza indirilebilmesi için alınması gereken önlemleri açıklar.

**6.7 Yaşam Döngüsü Boyunca Sınama Etkinlikleri**Planlama aşamasında genel planlama sınaması gerçekleştirildi.  
Çözümleme aşamasında sınama planı uygulandı.  
Tasarım aşamasında detaylanan sınama planı, daha sonra oluşturulacak olan el kitabında kullanılacak.  
Gerçekleştirim aşamasında teknik sınamalar uygulandı.  
Kurulum aşamasında son sınamalar gerçekleştirildi.

**7. Bakım**

**7.1 Giriş**Sistemin tasarımı bittikten sonra sistemin güncel kalması ve kullanıcı beklentilerine karşılık vermesi amacıyla sisteme aylık bakım yapılacaktır.

**7.2 Kurulum**Github sunucusuna yüklenen uygulamanın indirilmesi ve kullanılması herhangi bir karmaşık işlem içermemektedir.

**7.3 Yerinde Destek Organizasyonu**Sunuculara yüklenen sistem Türkiye’nin farklı illerinde eş zamanlı olarak kullanılabileceğinden Kullanıcılara yerinde destek vermek mümkün olmayacaktır. Bunun yerine sistemle ilgili sorun yaşayan kullanıcılar elektronik posta ile bana ulaşabileceklerdir.

**7.4 Yazılım Bakımı**

**7.4.1 Tanım**Kullanıcı geri dönüşleri dahilinde yapılacak olan güncellemeler sistem bakımı olarak nitelendirilebilir.

**7.4.2 Bakım Süreç Modeli  
Adım 1:** Kullanıcı uygulama beklentisi hakkında geri dönüşte bulunur.  
**Adım 2:** Geliştirici ekip geri dönüşleri inceleyip beklenen sistem hakkında gereksinimleri belirler.  
**Adım 3:** Geliştirici ekip belirlenen gereksinimler doğrultusunda sistem planlaması ve analizi yapar.  **Adım 4:** Geliştirici ekip sistem gerçekleştirilmeden önce sistem tasarımını yapıp süreç işleyişi hakkında bilgi edinir.  **Adım 5:** Geliştirici ekip belirlenen nicelikler doğrultusunda sistemi günceller. **Adım 6:** Geliştirici ekip kullanıcıdan gelen geri dönüşler ve istekler doğrultusunda sistemin test ve bakımını gerçekleştirir.

**8. Sonuç**Takip edilen maddelerde Bilgi Yakalayıcı adlı uygulamayı ve işlevlerini sundum.   
 Bilginin güç olduğu çağımızda gerçekleştirilen bu uygulama ile kullanıcıların bilgi düzeyinde ve genel kültür seviyesinde önceki bilgi düzeyi ve genel kültür seviyesine oranla artış beklenmekte.  
Bu uygulamayı gerçekleştirirken herhangi bir sistemi referans almadım.  
**Sistemin dezavantajları:**- Yöneticinin bilgileri arayüz değil de MySQL workbench ile girmesi.  
- Bilgilerin görsel açıdan eksik olması.  
**Sistemin gelecekteki geliştirmeleri:**- Verilen bilgiler için görsel ekleme de yapılabilecek.  
- Sistem yöneticisi için bilgileri girebileceği arayüz oluşturulacak.  
- Yeni ve eğlenceli bilgi kategorileri eklenecek.  
olarak belirlendi bu geliştirmelere ek olarak kullanıcı geri dönüşleri de dikkate alınıp sisteme güncelleme yapılacak.

**9. Kaynaklar**[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529– 551, April 1955. (references)   
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.   
[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350. [4] K. Elissa, “Title of paper if known,” unpublished.   
[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.   
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magnetooptical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740– 741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].   
[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University   
**İnternet kaynakları:**1-http://www.fao.org, (25.03.2011)   
2-www.atsdr.cdc.gov, (03.06.2010)   
3-http://en.wikipedia.org/wiki/File:4-Hydroxybenzoic\_acid.svg, (07.12.2011)