YÖNETİM BİLİŞİM SİSTEMLERİ

**KÜTÜPHANE WEBAPI YAZILIMI İÇİN SİSTEM ANALİZİ VE TASARIM**

Ümeyir ATASOY

Danışman: Dr. Öğr. Üyesi Muhammed Fatih ALAEDDİNOĞLU

Erzurum

Kasım, 2022

İçindekiler

[1. PLANLAMA 3](#_Toc119957167)

[1.1 Mevcut Sistemin İncelenmesi 3](#_Toc119957168)

[1.1.1 Mevcut Sistemin Adımları 3](#_Toc119957169)

[1.1.1.1 Hazırlık 3](#_Toc119957170)

[1.2 Mevcut Sistemin Uygunluğu 4](#_Toc119957171)

[1.3 Problemlerin ve İyileştirilmelerin Belirlenmesi 4](#_Toc119957172)

[1.3.1 Güvenilirlik 4](#_Toc119957173)

[1.3.2 Veri Kontrolleri 4](#_Toc119957174)

[1.3.3 Teknik ve İnsan İşgücü 4](#_Toc119957175)

[1.4 Mevcut Sistemin İşleyişine Ait Ditagramlar 5](#_Toc119957176)

[1.4.1 Data-Flow Diagram (DFD) 5](#_Toc119957177)

[1.4.2 HIPO Diagram (DFD) 7](#_Toc119957178)

[1.4.2 E-R (Entity Relationship / Varlık İlişki) Diagram 9](#_Toc119957179)

[2. ÇÖZÜMLEME 10](#_Toc119957180)

[2.1 Hedeflerin Belirlenmesi 10](#_Toc119957181)

[2.2Sistemin Önemi ve Gerekçesi 10](#_Toc119957182)

[3. İLGİLİ ARAŞTIRMALAR 10](#_Toc119957183)

[3.1 Katmanlı Mimariler 10](#_Toc119957184)

[3.2 Kullanılan Teknolojİler 10](#_Toc119957185)

[3.3 POSTMAN 10](#_Toc119957186)

[3.4 REST API 11](#_Toc119957187)

[4. UYGULAMA VE TASARIM 12](#_Toc119957188)

[4.1 Uygulama Kullanımı 12](#_Toc119957189)

[4.2 Uygulama Tasarımı 13](#_Toc119957190)

[5. TEST BAKIM VE GÜNCELLEME 14](#_Toc119957191)

[6.SONUÇ VE ÖNERİLER 14](#_Toc119957192)

[6.1 Kurumsal Çerçeve 14](#_Toc119957193)

[KAYNAKÇA 15](#_Toc119957194)

# PLANLAMA

Atatürk Üniversitesi bünyesindeki İktisadi ve İdari Bilimler Fakültesi, Yönetim Bilişim Sistemleri bölümünün Yazılım Mühendisliği dersinin projesi için Kütüphane WebAPI yazılımı yapılmaktadır. Yazılımda Kütüphaneye kitap eklenebilir, kitap silinebilir, kitap verileri güncellenebilir, kitaplar listelenebilir ve bu işlemler anlık ve manuel olarak Postman uygulaması aracılığıyla yapılır. Bireysel olarak projeyi geliştirip, yöneteceğim.

## 1.1 Mevcut Sistemin İncelenmesi

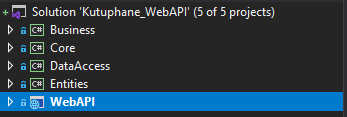
Projedeki yapacağım metotları dinamik bir şekilde kodlayacağım. Veritabanını kullanarak

yazacağım metotları veritabanıyla dinamik hale getireceğim. Yazılımı katmanlı mimari kullanarak yapacağım: Entities, DataAccess, Business, Core, WebAPI Yazılımı yaparken belli başlı teknolojiler kullanacağım bunlar; Autofac, Entity Framework, Validation vb.

### 1.1.1 Mevcut Sistemin Adımları

#### 1.1.1.1 Hazırlık

Projeye başlarken ilk olarak varlık(Entity)lerimizi Entities katmanında belirliyoruz. Belirlediğimiz varlıklarımızın veritabanı ile ilişkisini ve metotların temel kodlarını DataAccess katmanına yazıyoruz. İş kodlarımızı oluşturduğumuz metotların iş kodlarını Business Katmanına yazdıktan sonra. Genel kodlarımızı Code Katmanına yazıyoruz. Hatalar ve eksikler giderildikçe sistem yazılımı tamamlanıyor.



## Mevcut Sistemin Uygunluğu

Mevcut sistemin iş yapma ile alakalı uygunluğu ile ilgili olarak bir probleme rastlanmamaktadır. Ancak işlerin manuel yapılacağından dolayı belli başlı problemler görülebilir.

## 1.3 Problemlerin ve İyileştirilmelerin Belirlenmesi

### 1.3.1 Güvenilirlik

Veriler sürekli manuel kontrol için el altında tutulmakla verilerin güvenliği konusu ihlal edilmiş olunacaktır.

### 1.3.2 Veri Kontrolleri

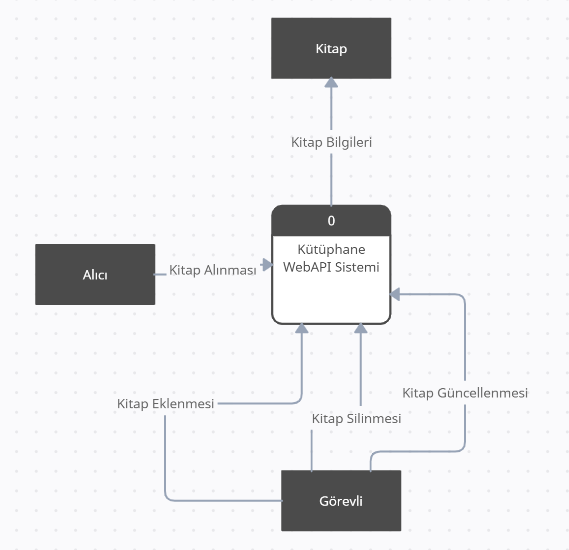
Yazılıma ait bütün işlemleri sadece tek bir kişi yapacağı için yazılımda tek bir kod hatası tüm sistemi etkilemesine sebep olacak.

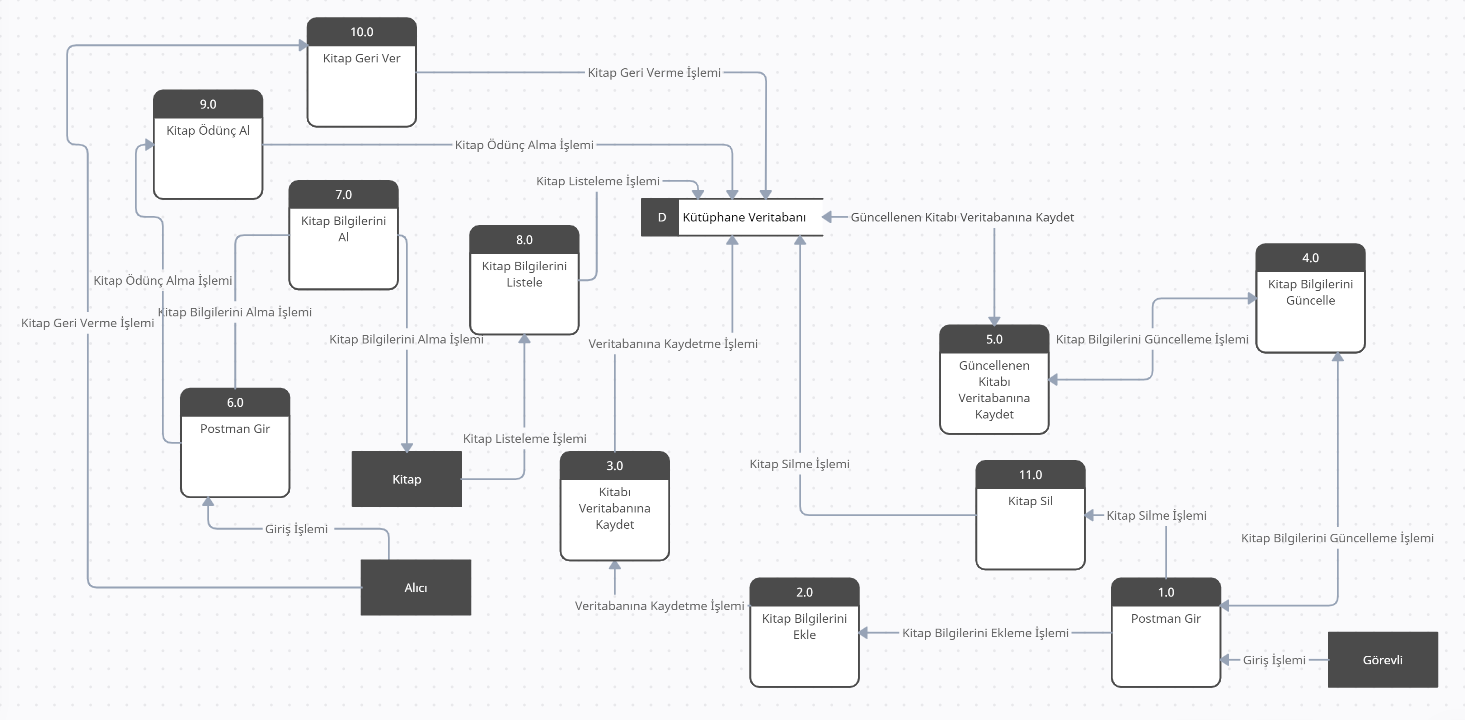
### 1.3.3 Teknik ve İnsan İşgücü

Projenin bu aşaması ile ilgili çalışma sonraki projelere bırakılmıştır. Bu sebeple bu adım ile ilgili problem ve iyileştirme analizi yapılmayacaktır.

## 1.4 Mevcut Sistemin İşleyişine Ait Ditagramlar

### 1.4.1 Data-Flow Diagram (DFD)

Kütüphane WebAPI Sistemi Düzey 0 



1Kütüphane WebAPI Sistemi Düzey 1

### 1.4.2 HIPO Diagram (DFD)

Şekil 2 HIPO Diyagram Şeması

HIPO Özet Diyagramı

* Yapan: Ümeyir ATASOY Sistem: Kütüphane WebAPI Sistemi Tarih: 21.11.2021
* Diyagram No: 0.0 Proses: Kütüphane WebAPI Sistemi Sayfa: 2/2

GİRDİ PROSES ÇIKTI

1. Gerekli Verileri Oku
2. Verileri İşleme Al
3. Sonuçları Yazdır

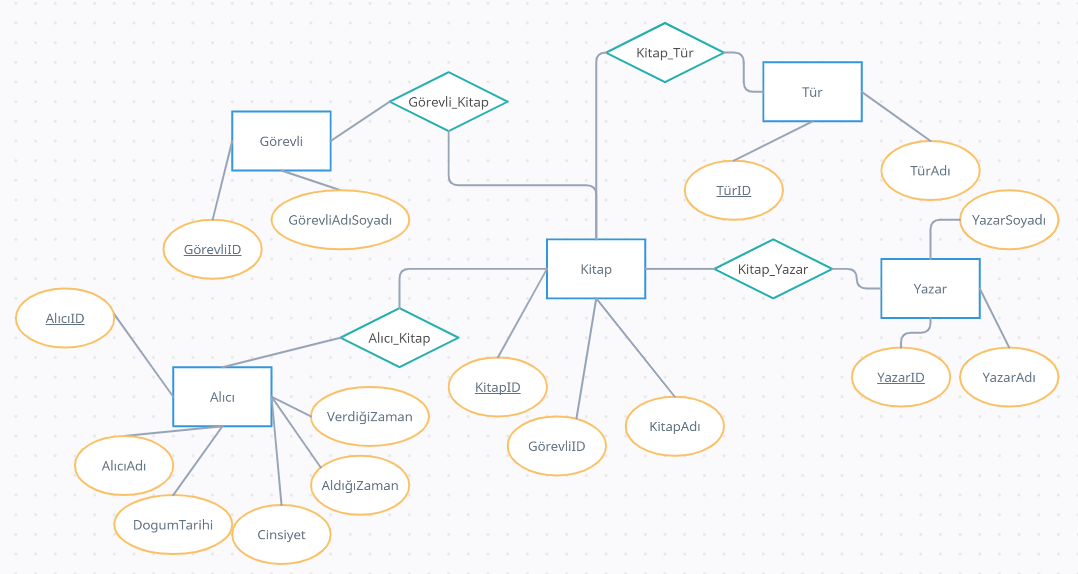
Kütüphane WebAPI Sistemi

Kitap Alımı

Kitap Tabloları

Kütüphane WebAPI Sistemi

### 1.4.2 E-R (Entity Relationship / Varlık İlişki) Diagram



# ÇÖZÜMLEME

## 2.1 Hedeflerin Belirlenmesi

Bu yazılım ile WebAPI kullanarak sistemi daha sistematik hale getiriyoruz. Yazılım ile kitap eklenebilir, kitap silinebilir, kitap verileri güncellenebilir, kitaplar listelenebilir. Bu yazılım ile hem api yazılımını öğreniyoruz hem de c# dilini.

## 2.2Sistemin Önemi ve Gerekçesi

Yazılım Mühendisliği dersinin projesi için Kütüphane WebAPI yazılımı yapıyorum. Hem bu projede yazılım için kendimi geliştiriyorum bu sistem ile birlikte api kullanmak isteyenlerin katmanlı mimariyi öğrenecekler kullanacağım teknolojileri daha yakından görecekler.

# İLGİLİ ARAŞTIRMALAR

## 3.1 Katmanlı Mimariler

Katmanlı Mimariler 3 ana katmandan oluşur bunlar Veri Katmanı, İş Katmanı, Sunum Katmanı olarak sınıflandırılabilir. Bunlardan ayrı olarak Core ve Entites katmanımız vardır bu katmanda ise 3 ana katmanda ortak kullanacağımız kodları Core katmanına, varlıklarımızı Entites yazarız.

Entites: Veritabanındaki tablolarımız projemizde nesne olarak oluşturulmuştur.

Veri Katmanı: Projeyi Veritabanına bağlayan katmandır.

İş Katmanı: Projemizin iş katmanıdır. Çeşitli iş kuralları; Veri kontrolleri, doğrulamalar ve yetkilendirme kontrolleri.

WebAPI: Projenin Dinlendirici API Katmanıdır. Bilinen yöntemler: Al, Gönder, Koy, Sil

## 3.2 Kullanılan Teknolojİler

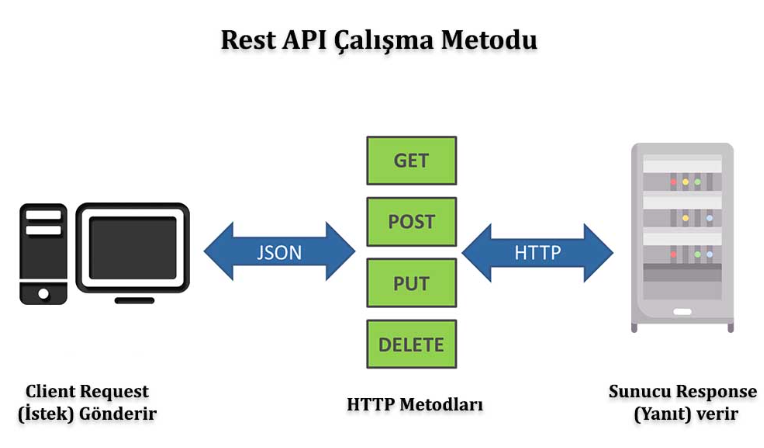
* Result Types
* Interceptor
* Autofac
* Validation
* Fluent Validation
* Validation Error Details

## 3.3 POSTMAN

Yazacağımız yazılımın sistemin ana başlığı API olmaktadır. Bu API’yi test edebilmek için Postman Uygulaması kullanacağım. Bilinen postman metotları: Get, Post, Put, Delete Genel tanımı: Postman, geliştiricilerin API'lerini tasarlaması, oluşturması, test etmesi ve yinelemesi için bir API platformudur.

## 3.4 REST API

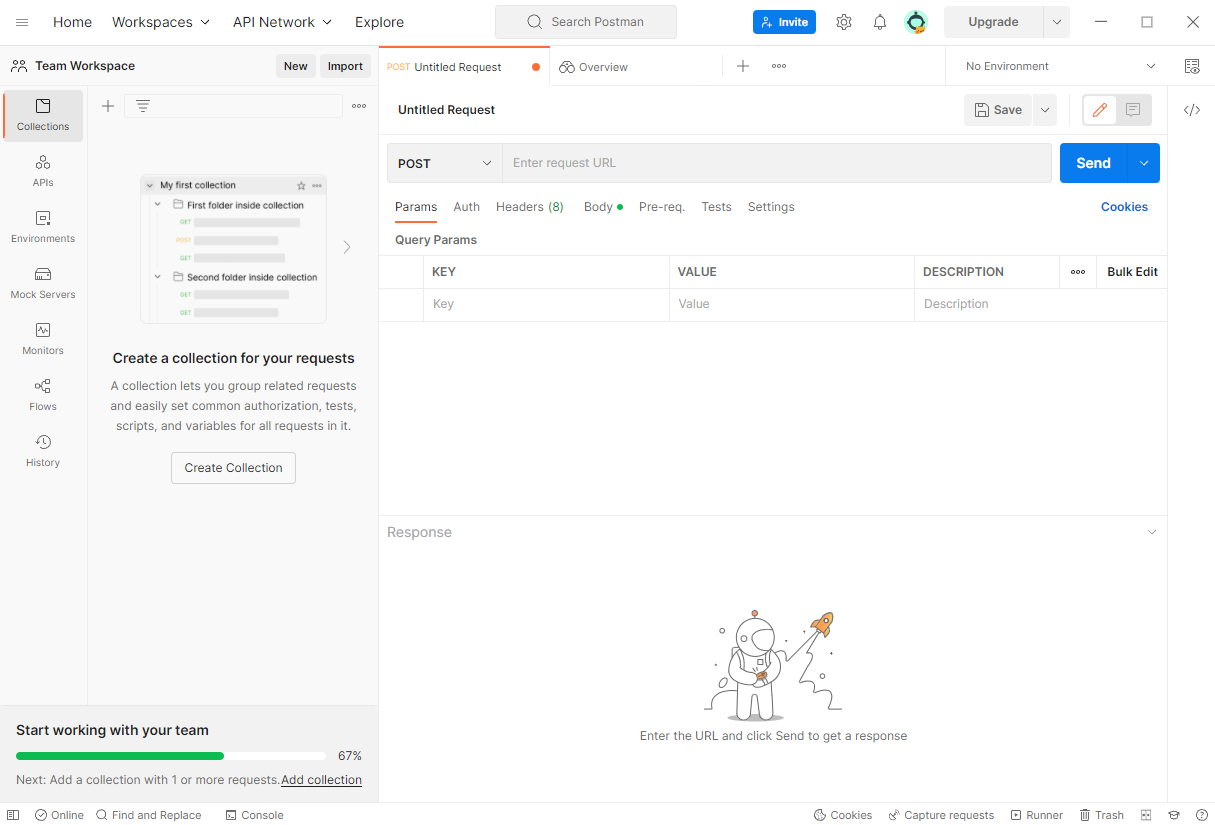
Günümüzde yaygın olarak REST API (Representational State Transfer) olarak da bilinen bir mimari yaklaşım tercih edilmektedir. Sadece HTTP protokolü üzerinden cevap alıp genellikle JSON formatında dönüş sağlar(isteğe göre XML de dönüş sağlayabilir).



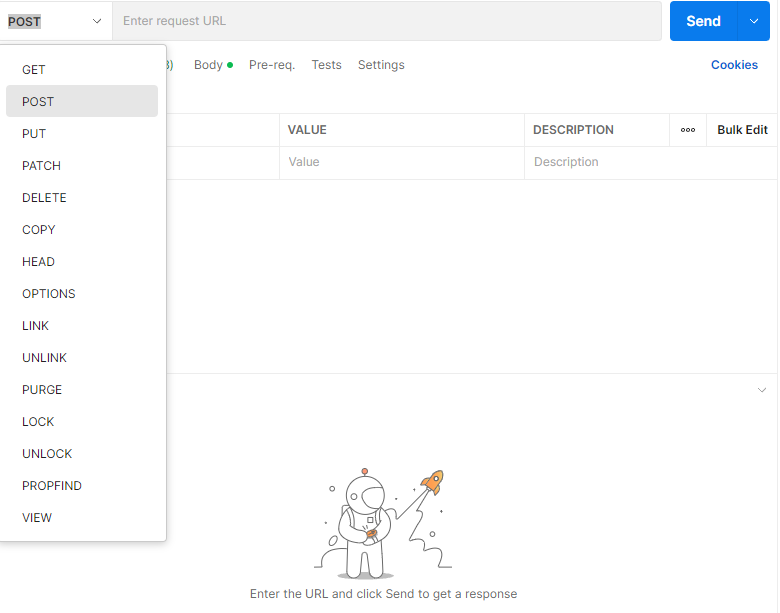
# UYGULAMA VE TASARIM

## 4.1 Uygulama Kullanımı

Yazılan yazılımı kullanım şekli Postman uygulamasından geçiyor. Yazılan metotları kullanabilmek için postman uygulamasını kullanıyoruz. Host için localhost kullanacağız.



Postman uygulaması sayesinde yazdığımız kodları get post vb. edebileceğiz.



## 4.2 Uygulama Tasarımı

Sistemin yapılışı:

|  |  |
| --- | --- |
| **Yapılan İş** | **Kim** |
| Varlıkların Belirlenmesi | Yazılımı Yazan Kişi |
| Veri bağlantıları ve iş kodlarının yazımı | Yazılımı Yazan Kişi |
| WebAPI kullanıma hazırlanması | Yazılımı Yazan Kişi |

Sistemin hazırlığı:

# TEST BAKIM VE GÜNCELLEME

Proje kodlarının tamamlanması sonucunda test edilebilir hale gelecektir.

Sistemin bakımı yazılımı yapan kişi tarafından yapılacak olup sistem bakımı kodlanan ara yüzde yapılacaktır.

Sistemin güncellenmesi yazılımı yapan kişi tarafından yapılacak olup sistem güncellemelerin kolaylığı açısından kullanılan teknolojileri sisteme iyice yerleştirilecek.

# 6.SONUÇ VE ÖNERİLER

## 6.1 Kurumsal Çerçeve

Bu sistemde amacımız webapi kullanarak kütüphaneye api görevlerini yazmak olucak. Webapi sayesinde daha hızlı bir şekilde verileri işleyeceğiz. Bu sistemde temel sorunlarımız API nedir, nasıl çalışır ve nasıl kullanılır gibi sorulara cevap aramak olmalı.

Bu projeye daha fazla **HTTP Metotları (HTTP Request Methods)** eklenebilir.

# KAYNAKÇA

SİSTEM ANALİZİ KAYNAK

https://aws.amazon.com/tr/what-is/api/#:~:text=API'nin%20a%C3%A7%C4%B1l%C4%B1m%C4%B1%20olan%20Application,aras%C4%B1ndaki%20hizmet%20s%C3%B6zle%C5%9Fmesi%20gibi%20d%C3%BC%C5%9F%C3%BCn%C3%BClebilir.

https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/08/%C3%A7ok-katmanl%C4%B1-mimari

https://www.hosting.com.tr/bilgi-bankasi/api/

https://github.com/engindemirog/NetCoreBackend

https://en.wikipedia.org/wiki/Postman\_(software)

https://www.hosting.com.tr/bilgi-bankasi/rest-api/

<https://mbilgil0.medium.com/http-metotlar%C4%B1-http-request-methods-90d57d574dfa>

DİAGRAM KAYNAK

https://www.ndonmez.com/wp-content/uploads/2017/03/sistem-analizi-ve-tasarimi.pdf

<https://kontrolotomasyon.files.wordpress.com/2015/02/sistem-analizi-1.pdf>

<https://kevserkose.wordpress.com/2017/05/26/varlik-iliski-modeli-entity-relationship-model/>