

# IOS Mobil Uygulama Geliştirme Eğitimi | Swift

## Uygulama Mimarisi - VIPER Pattern

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi

Android - IOS Developer and Trainer

# Uygulama Mimarisi

# Viper Pattern

# VIPER Pattern

VIPER , bir uygulama mimarisidir.

Bu mimari sayesinde daha profesyonel kodlamalar yapabiliriz.

Özellikle büyük projelerde oluşan kodlama karmaşasını azaltır.

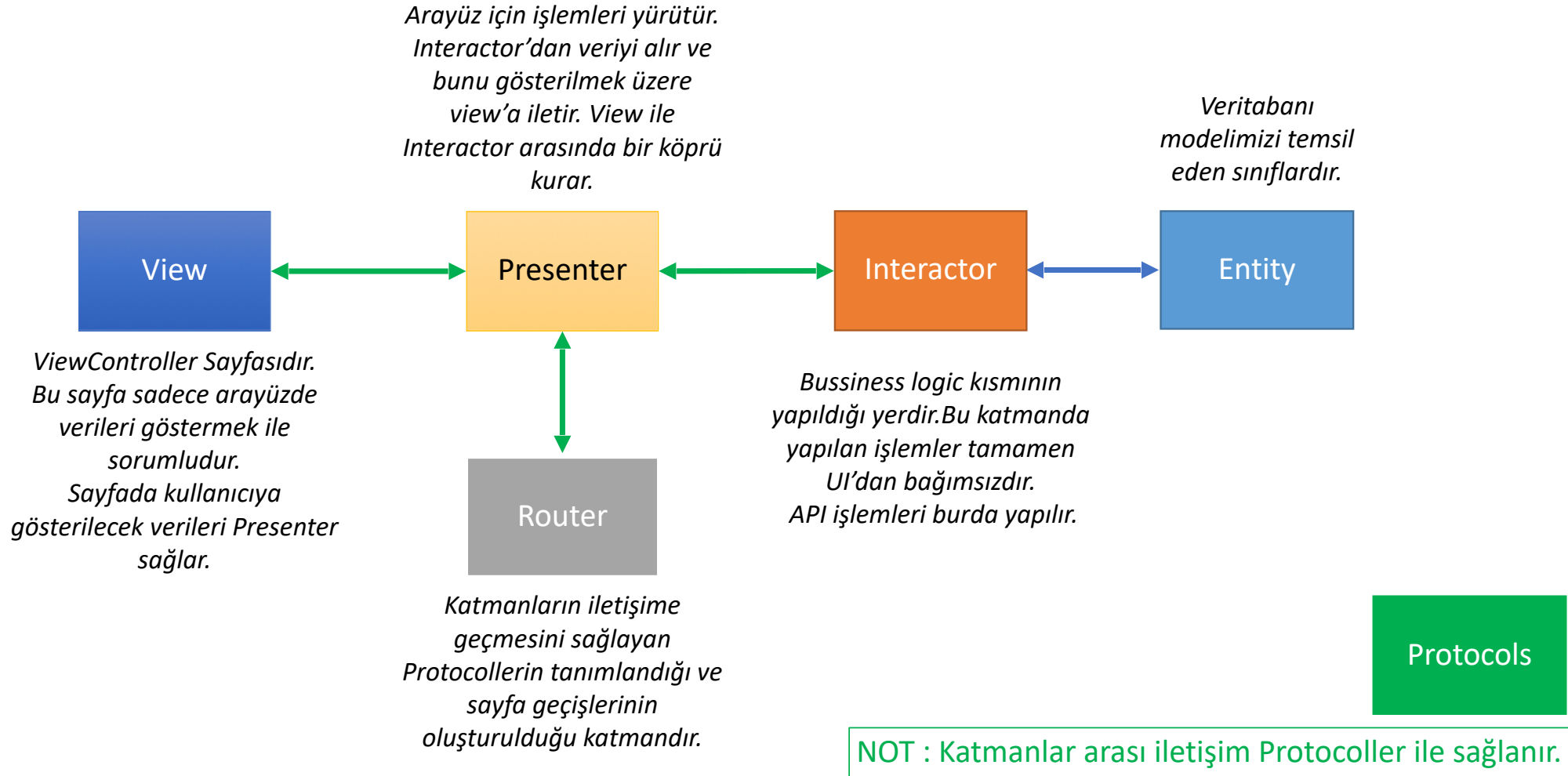
Mimarinin amacı proje içerisindeki işlemsel kodları ayırıp daha modüller ve düzenli hale getirmektir.

Bu mimari kullanımını öğrenmek için mimarinin temel yapılarını öğrenmemiz gereklidir.

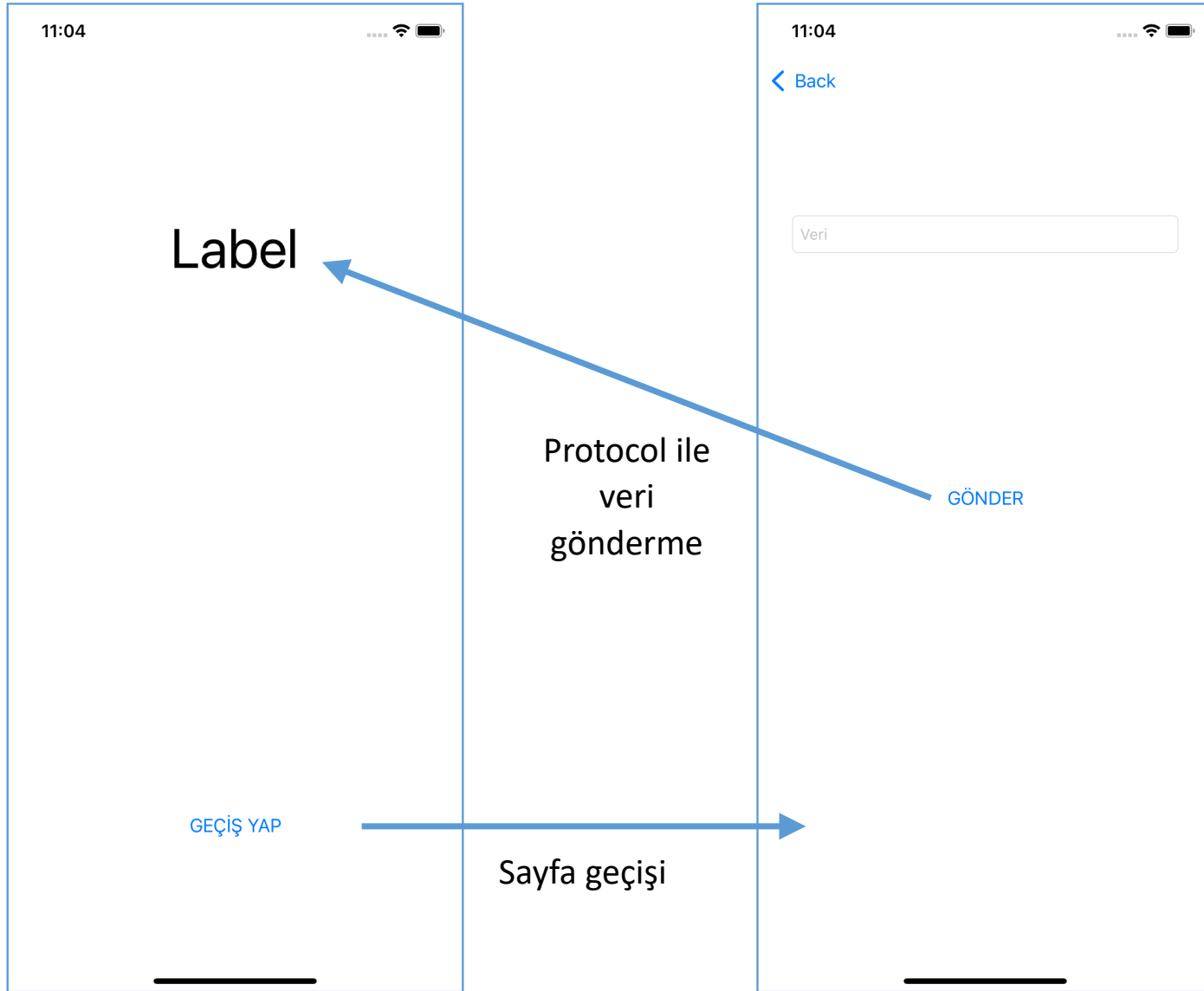
Temelinde protocol kullanımı vardır.

# Viper Pattern Katmanları

# Uygulama Mimarisi - VIPER Pattern



# Protocol Kullanımı



Detay sayfasından girilen veri gönder buttonuna basılarak ilk sayfaya gönderilir ve geri geldiğimizde veriyi görebiliriz.



## DetayVCToViewControllerProtocol

*Metodu kullanmak için  
implement edilir.*

`veriGonder(mesaj:String)`

`detayVC.delegate = self`

ViewController

```
veriGonder(mesaj:String){  
  
}
```

*Metodun dinlenildiği  
yerden tetiklenecek  
sınıftaki nesneye yetki  
verilmelidir. Aksi halde  
yapı çalışmaz.*

`var delegate:DetayVCToViewControllerProtocol?`

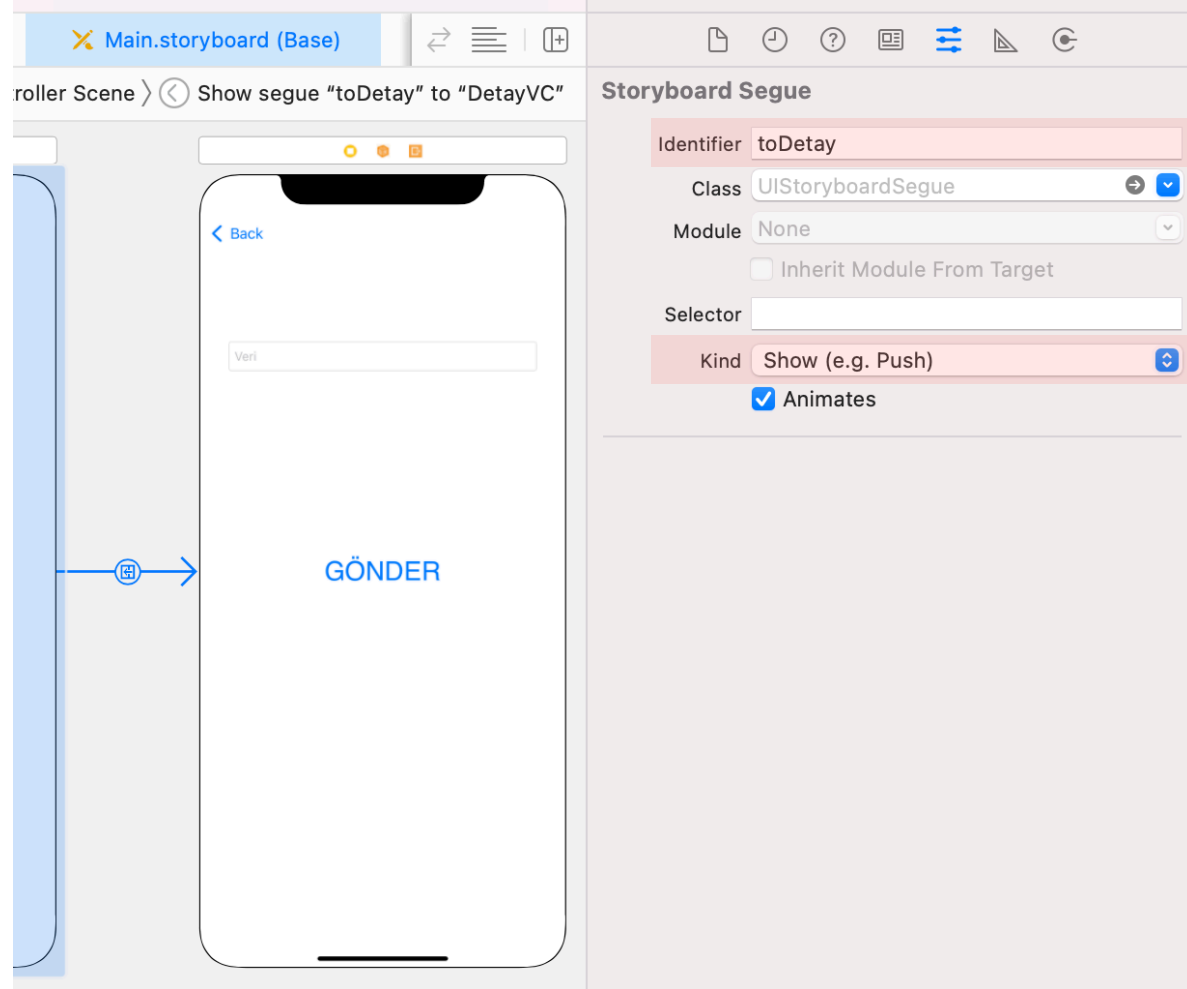
DetayVC

`delegate.veriGonder(mesaj: "Merhaba")`

*Tetikleme*

*Protocol metodu başka sınıftan çalıştırılmak  
için bekler.*

# Sayfa geiři oluřturma



```
protocol DetayVCToViewControllerProtocol{  
    func veriGonder(mesaj:String)  
}
```

Metodu kullanmak için implement edilir.

```
class ViewController: UIViewController {  
    @IBOutlet weak var labelSonuc: UILabel!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
  
    @IBAction func buttonGecisYap(_ sender: Any) {  
        performSegue(withIdentifier: "toDetay", sender: nil)  
    }  
  
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
        if segue.identifier == "toDetay" {  
            let gidilecekVC = segue.destination as! DetayVC  
            gidilecekVC.delegate = self  
        }  
    }  
}  
  
extension ViewController : DetayVCToViewControllerProtocol{  
    func veriGonder(mesaj: String) {  
        labelSonuc.text = mesaj  
    }  
}
```

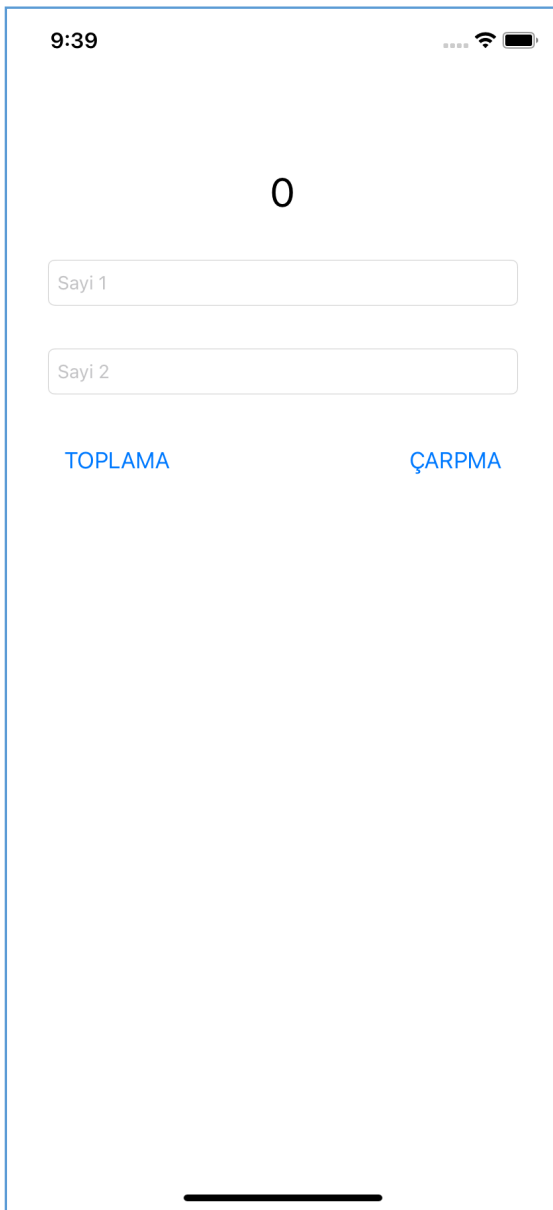
Yetki verme işlemi için önce ilgili sınıfa erişim içindeki delegate nesnesine erişmeliyiz. Daha sonrada bu nesneye yetki vermeliyiz.

Tetikleme için yetki verme

Tetikleme

```
import UIKit  
  
class DetayVC: UIViewController {  
    @IBOutlet weak var textFieldGirdi: UITextField!  
  
    var delegate:DetayVCToViewControllerProtocol?  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
    @IBAction func buttonGonder(_ sender: Any) {  
        if let mesaj = textFieldGirdi.text {  
            delegate?.veriGonder(mesaj: mesaj)  
        }  
    }  
}
```

# Klasik Yöntem



```
class ViewController: UIViewController {

    @IBOutlet weak var labelSonuc: UILabel!
    @IBOutlet weak var textFieldSayi1: UITextField!
    @IBOutlet weak var textFieldSayi2: UITextField!

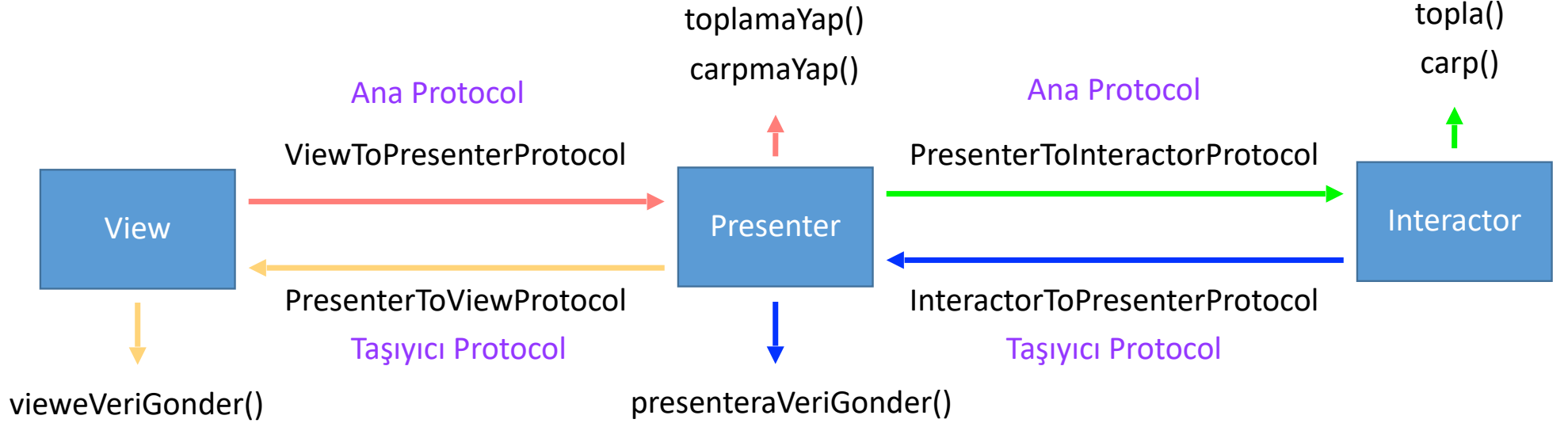
    override func viewDidLoad() {
        super.viewDidLoad()

        labelSonuc.text = "0"
    }

    @IBAction func buttonToplama(_ sender: Any) {
        if let sayi1 = textFieldSayi1.text, let sayi2 = textFieldSayi2.text {
            if let s1 = Int(sayi1), let s2 = Int(sayi2){
                let toplam = s1 + s2
                labelSonuc.text = String(toplam)
            }
        }
    }

    @IBAction func buttonCarpma(_ sender: Any) {
        if let sayi1 = textFieldSayi1.text, let sayi2 = textFieldSayi2.text {
            if let s1 = Int(sayi1), let s2 = Int(sayi2){
                let carpma = s1 * s2
                labelSonuc.text = String(carpma)
            }
        }
    }
}
```

# Viper Pattern Alt Yapımız



*Bir sınıfta neye ihtiyacımız olduğunu anlarsak procolleri kolaylıkla oluşturabiliriz.*

*Örnek : Interactor için topla() ve carp() metodlarını kullanıcaz ozaman PresenterToInteractorProtocol implement etmemiz gereklidir. presenteraVeriGonder() metodunu çalıştırmamız gerekli o zaman InteractorToPresenterProtocol protocolünden nesneye ihtiyacımız vardır.*

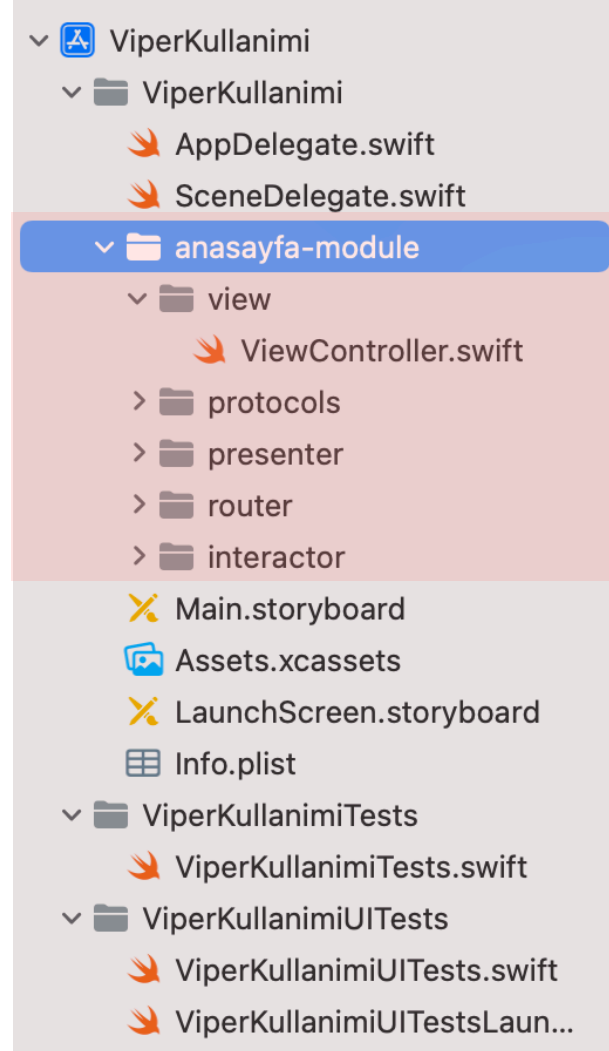
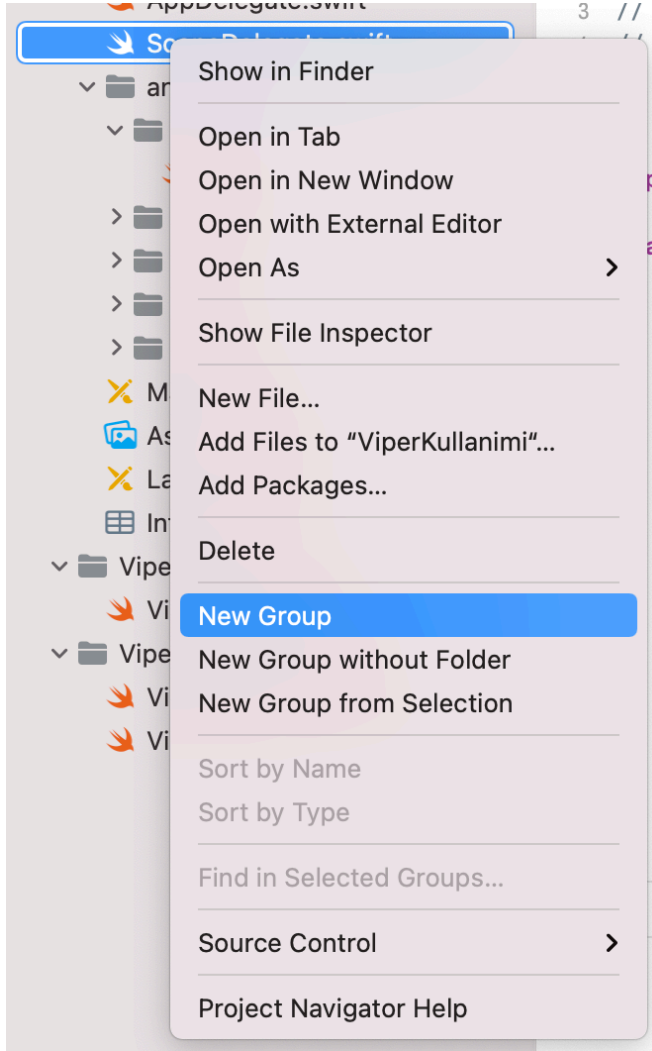
*Not : Okların başlangıç noktası tetikleme yapılacak yeri , bitiş noktası ise metodun kullanılacağı yeri temsil eder.*

*Not : Tetikleme yapılacak yerde , tetiklenecek sınıftan nesne olmalıdır.*

Protocols

NOT : Katmanlar arası iletişim Protocoller ile sağlanır.

# Her sayfaya özel module oluşturmaliyiz



Module dediğimiz kavram  
group oluşturmaktır.

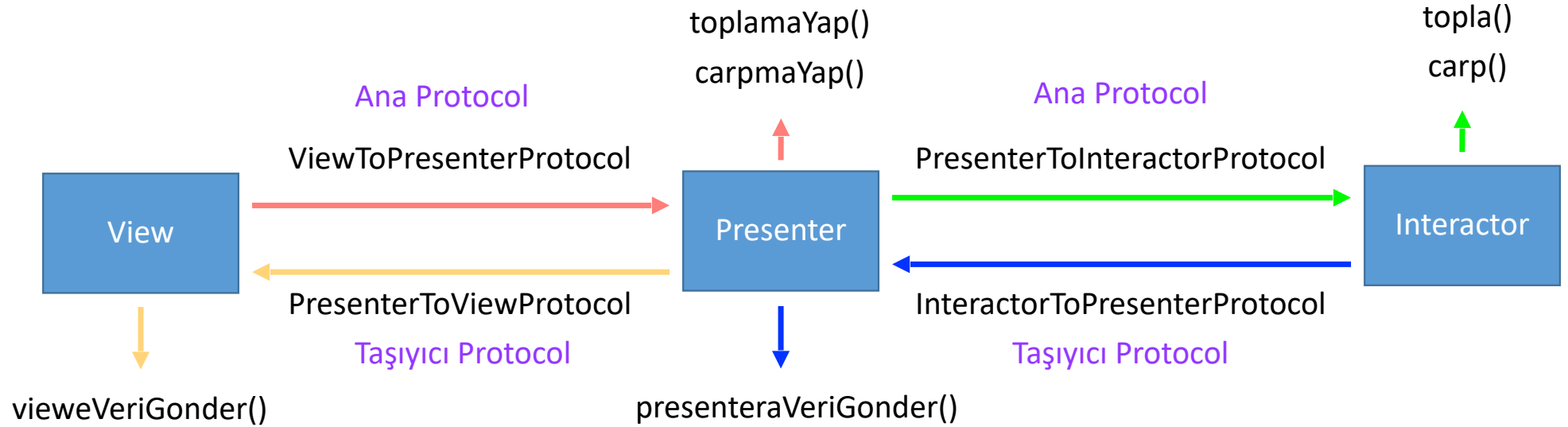
Module ile daha temiz bir  
kodlama düzeni elde ederiz.

Her ViewController için  
oluşturulmalıdır.

Module içinde Viper  
Pattern katmanları  
olacaktır.

Not : Entity module de  
oluşturulabilir tercihe  
bağlıdır.

# Protocollerimiz



## Ana Protocoller

```
protocol PresenterToInteractorProtocol{
    var presenter:InteractorToPresenterProtocol? {get set}

    func topla(sayi1:String,sayi2:String)
    func carp(sayi1:String,sayi2:String)
}
```

```
protocol ViewToPresenterProtocol{
    var interactor:PresenterToInteractorProtocol? {get set}
    var view:PresenterToViewProtocol? {get set}

    func toplamaYap(sayi1:String,sayi2:String)
    func carpmaYap(sayi1:String,sayi2:String)
}
```

*Tetiklenecek  
katmanın adi  
verilir.*

*Ana protocoller  
içinde değişken  
tanımlayarak  
tetikleme  
yapmamız  
gerekir.*

## Taşıyıcı Protocoller

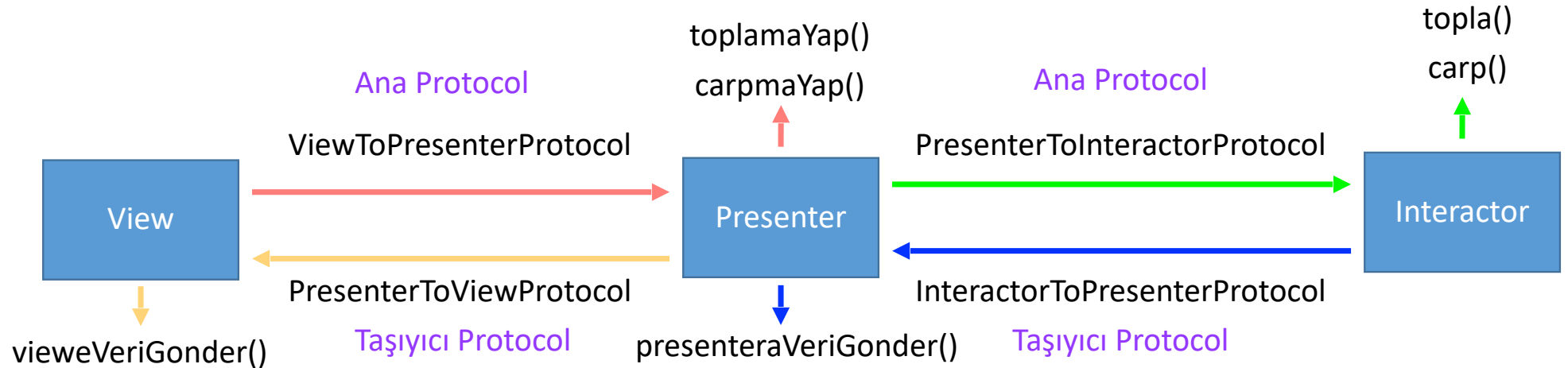
```
protocol InteractorToPresenterProtocol{
    func presenteraVeriGonder(sonuc:String)
}
```

```
protocol PresenterToViewProtocol{
    func vieweVeriGonder(sonuc:String)
}
```

*Taşıyıcı protocoller içinde değişken  
tanımlaya gerek yoktur.Sadece  
metodlar ile veri taşırlar.*



# Interactor Sınıfı



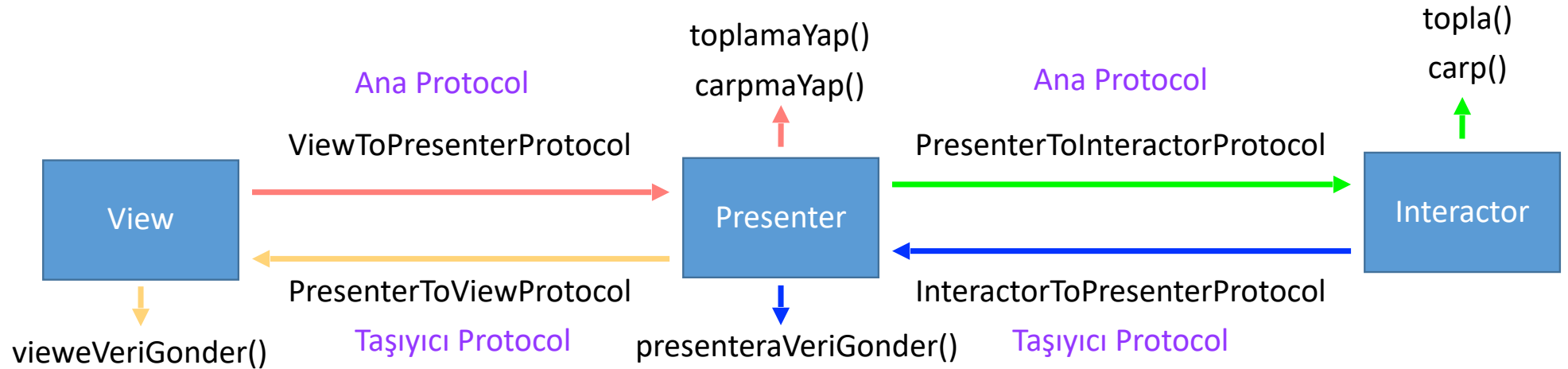
```
class Interactor : PresenterToInteractorProtocol{ Ana Protocol  
    var presenter: InteractorToPresenterProtocol? Taşıyıcı Protocol
```

```
func topla(sayi1: String, sayi2: String) {  
    if let s1 = Int(sayi1), let s2 = Int(sayi2){  
        let toplam = s1 + s2  
        presenter?.presenteraVeriGonder(sonuc: String(toplam))  
    }  
}
```

```
func carp(sayi1: String, sayi2: String) {  
    if let s1 = Int(sayi1), let s2 = Int(sayi2){  
        let carpma = s1 * s2  
        presenter?.presenteraVeriGonder(sonuc: String(carpma))  
    }  
}
```

```
}
```

# Presenter Sınıfı



```
class Presenter : ViewToPresenterProtocol { Ana Protocol
    var interactor: PresenterToInteractorProtocol? Ana Protocol
    var view: PresenterToViewProtocol? Taşıyıcı Protocol
```

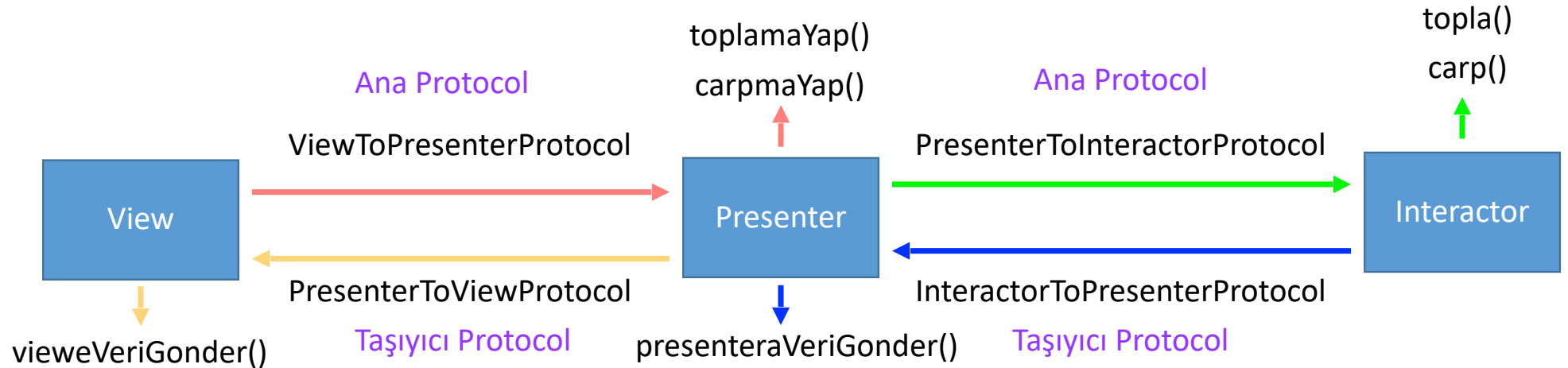
```
    func toplamaYap(sayi1: String, sayi2: String) {
        interactor?.topla(sayi1: sayi1, sayi2: sayi2)
    }

    func carpmaYap(sayi1: String, sayi2: String) {
        interactor?.carp(sayi1: sayi1, sayi2: sayi2)
    }
}
```

```
extension Presenter : InteractorToPresenterProtocol {
    func presenteraVeriGonder(sonuc: String) {
        view?.vieweVeriGonder(sonuc: sonuc)
    }
}
```

Taşıyıcı Protocol

# View Sınıfı



```
class ViewController: UIViewController {
    @IBOutlet weak var labelSonuc: UILabel!
    @IBOutlet weak var textFieldSayi1: UITextField!
    @IBOutlet weak var textFieldSayi2: UITextField!

    var presenterNesnesi:ViewToPresenterProtocol? Ana Protocol

    override func viewDidLoad() {
        super.viewDidLoad()
        labelSonuc.text = "0"
        Router.createModule(ref: self) Bütün protocollerin çalışmasını sağlayan yetkilendirme.
    }

    @IBAction func buttonToplama(_ sender: Any) {
        if let sayi1 = textFieldSayi1.text, let sayi2 = textFieldSayi2.text {
            presenterNesnesi?.toplamaYap(sayi1: sayi1, sayi2: sayi2)
        }
    }

    @IBAction func buttonCarpma(_ sender: Any) {
        if let sayi1 = textFieldSayi1.text, let sayi2 = textFieldSayi2.text {
            presenterNesnesi?.carpmaYap(sayi1: sayi1, sayi2: sayi2)
        }
    }
}
```

## Taşıyıcı Protocol

```
extension ViewController : PresenterToViewProtocol {
    func vieweVeriGonder(sonuc: String) {
        labelSonuc.text = sonuc
    }
}
```

# Router Sınıfı

- Router sınıfı bütün protocollerin yetkilendirmesini yapar.

```
protocol PresenterToRouterProtocol{  
    static func createModule(ref:ViewController)  
}
```

```
class Router : PresenterToRouterProtocol {  
    static func createModule(ref: ViewController) {
```

```
        let presenter = Presenter()
```

```
        //View sınıfı değişkeni
```

```
        ref.presenterNesnesi = presenter
```

← View sınıfı içindeki değişkene yetki verme

```
        //Presenter sınıfı değişkenleri
```

```
        ref.presenterNesnesi?.interactor = Interactor()
```

```
        ref.presenterNesnesi?.view = ref
```

← Present sınıfı içindeki değişkene yetki verme

```
        //Interactor değişkeni
```

```
        ref.presenterNesnesi?.interactor?.presenter = presenter
```

← Interactor sınıfı içindeki değişkene yetki verme

```
    }
```

```
}
```

Presenter



Router

# View Sınıfı içinde Router Metodunu Çalıştırma

```
class ViewController: UIViewController {
    @IBOutlet weak var labelSonuc: UILabel!
    @IBOutlet weak var textFieldSayi1: UITextField!
    @IBOutlet weak var textFieldSayi2: UITextField!

    var presenterNesnesi:ViewToPresenterProtocol?

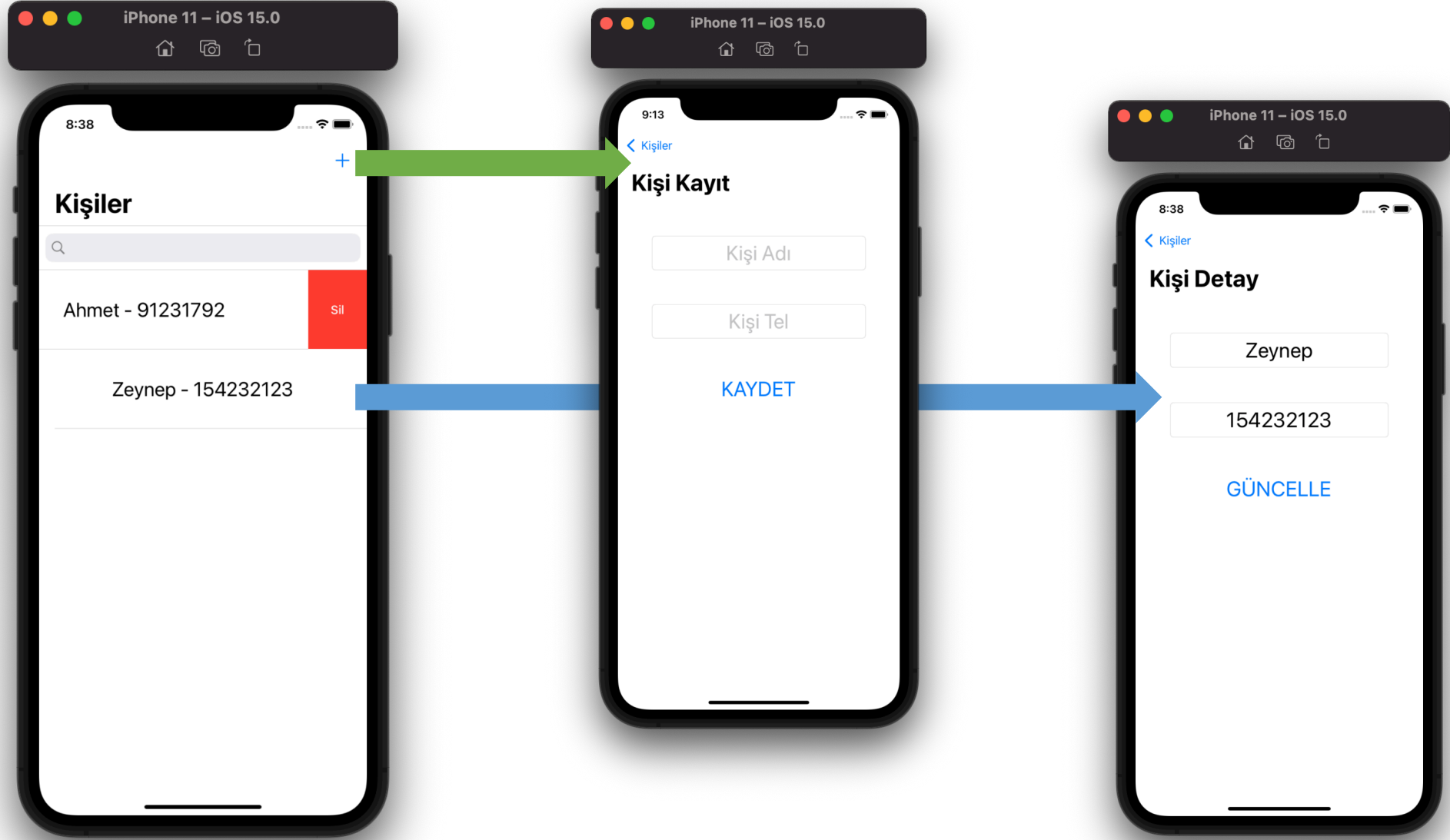
    override func viewDidLoad() {
        super.viewDidLoad()
        labelSonuc.text = "0"
        Router.createModule(ref: self)
    }

    @IBAction func buttonToplama(_ sender: Any) {
        if let sayi1 = textFieldSayi1.text, let sayi2 = textFieldSayi2.text {
            presenterNesnesi?.toplamaYap(sayi1: sayi1, sayi2: sayi2)
        }
    }

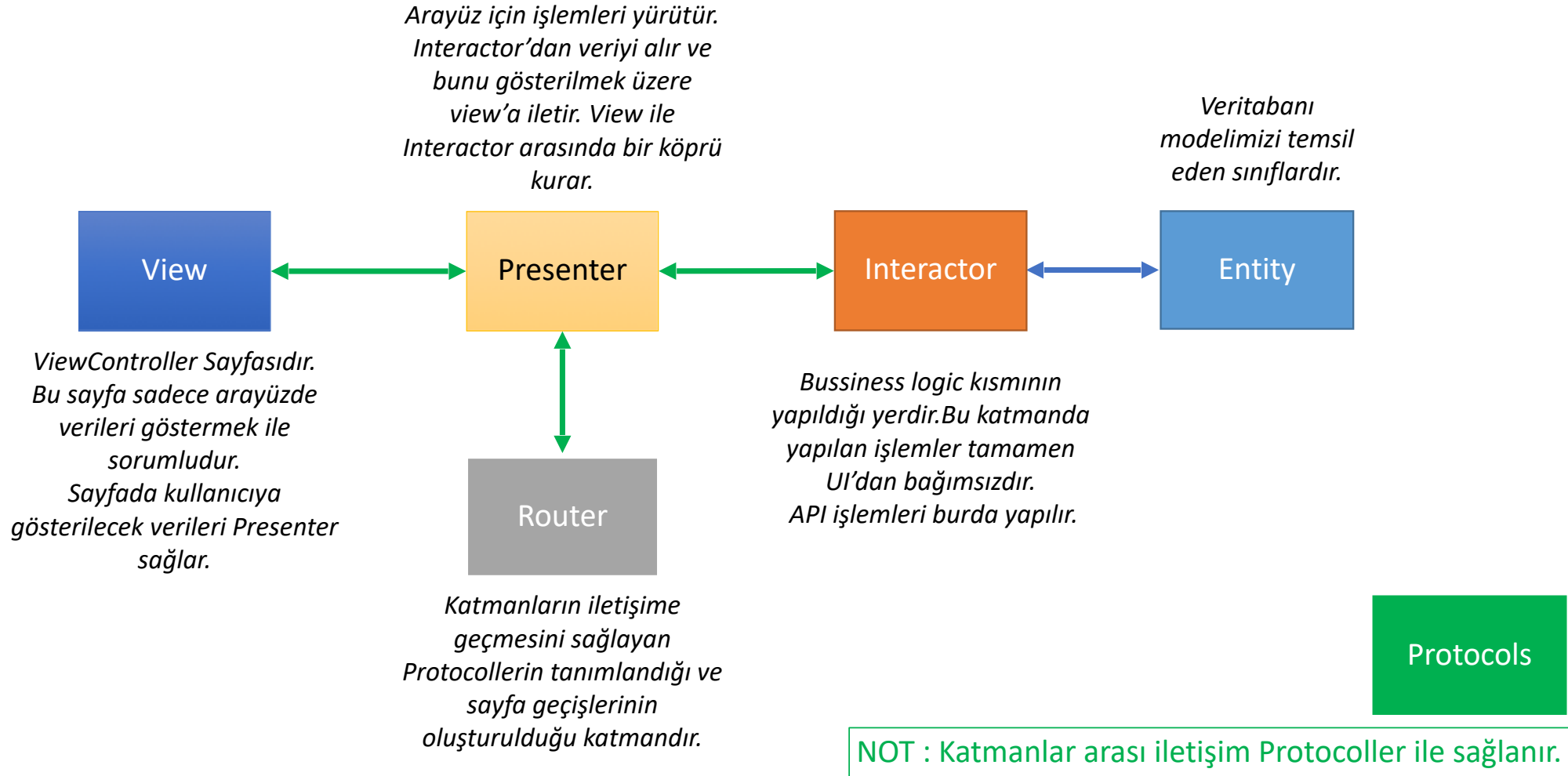
    @IBAction func buttonCarpma(_ sender: Any) {
        if let sayi1 = textFieldSayi1.text, let sayi2 = textFieldSayi2.text {
            presenterNesnesi?.carpmaYap(sayi1: sayi1, sayi2: sayi2)
        }
    }
}
```

# Kişiler Uygulaması Tasarım Viper Pattern

# Kişiler Uygulaması

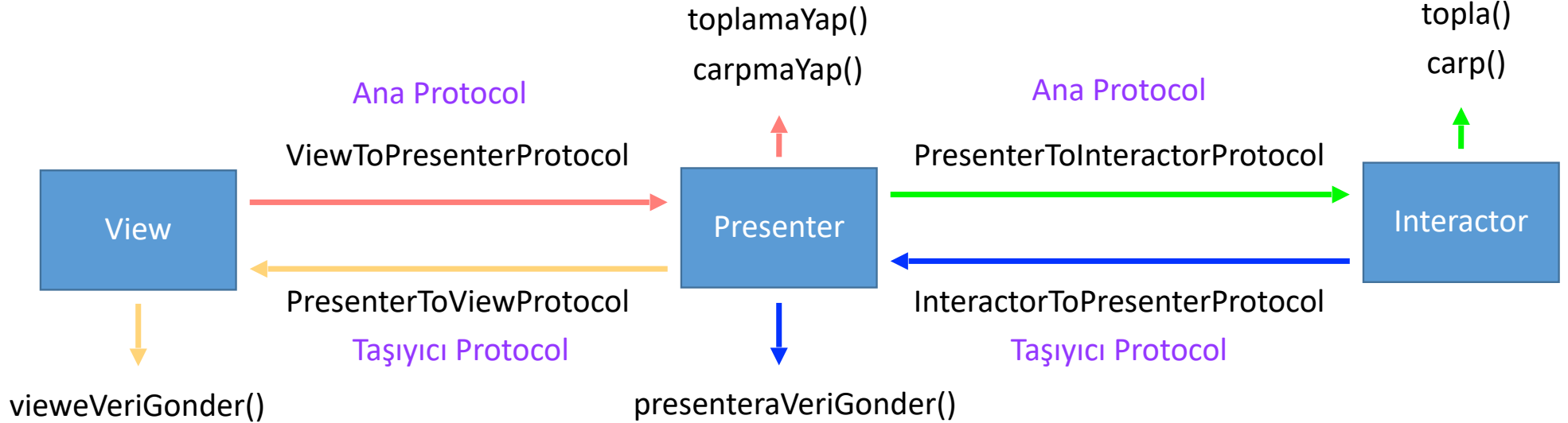


# Uygulama Mimarisi - VIPER Pattern





# Viper Pattern Alt Yapımız



*Bir sınıfta neye ihtiyacımız olduğunu anlarsak procolleri kolaylıkla oluşturabiliriz.*

*Örnek : Interactor için topla() ve carp() metodlarını kullanıcaz ozaman PresenterToInteractorProtocol implement etmemiz gereklidir. presenteraVeriGonder() metodunu çalıştırmamız gerekli o zaman InteractorToPresenterProtocol protocolünden nesneye ihtiyacımız vardır.*

*Not : Okların başlangıç noktası tetikleme yapılacak yeri ,  
bitiş noktası ise metodun kullanılacağı yeri temsil eder.  
Not : Tetikleme yapılacak yerde , tetiklenecek sınıftan  
nesne olmalıdır.*

Protocols

NOT : Katmanlar arası iletişim Protocoller ile sağlanır.

## tum\_kisiler.php

GET

```
<?php
// array for JSON response
$response = array();

//DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
require_once __DIR__ . '/db_config.php';

// Bağlantı oluşturuluyor.
$baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

// Bağlantı kontrolü yapılır.
if (!$baglanti) {
    die("Hatalı bağlantı : " . mysqli_connect_error());
}

$sqlsorgu = "SELECT * FROM kisiler";
$result = mysqli_query($baglanti, $sqlsorgu);

// result kontrolü yap
if (mysqli_num_rows($result) > 0) {

    $response["kisiler"] = array();

    while ($row = mysqli_fetch_assoc($result)) {
        // temp user array
        $kisiler = array();

        $kisiler["kisi_id"] = $row["kisi_id"];
        $kisiler["kisi_ad"] = $row["kisi_ad"];
        $kisiler["kisi_tel"] = $row["kisi_tel"];

        // push single product into final response array
        array_push($response["kisiler"], $kisiler);
    }
    // success
    $response["success"] = 1;

    // echoing JSON response
    echo json_encode($response);
}
```



```
} else {
    // no products found
    $response["success"] = 0;
    $response["message"] = "No data found";

    // echo no users JSON
    echo json_encode($response);
}
//bağlantı koparılır.
mysqli_close($baglanti);
?>
```

```
{
  "kisiler": [
    {
      "kisi_id": "376",
      "kisi_ad": "Ahmet",
      "kisi_tel": "5348564412"
    },
    {
      "kisi_id": "379",
      "kisi_ad": "Zeynep",
      "kisi_tel": "05375483214"
    },
    { },
    { },
    { },
    { },
    { }
  ],
  "success": 1
}
```

```
<?php
```

```
$response = array();

if (isset($_POST['kisi_ad'])) {
    $kisi_ad = $_POST['kisi_ad'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "SELECT * FROM kisiler WHERE kisiler.kisi_ad like '%$kisi_ad%'";
    $result = mysqli_query($baglanti, $sqlsorgu);

    if (mysqli_num_rows($result) > 0) {

        $response["kisiler"] = array();

        while ($row = mysqli_fetch_assoc($result)) {

            $kisiler = array();

            $kisiler["kisi_id"] = $row["kisi_id"];
            $kisiler["kisi_ad"] = $row["kisi_ad"];
            $kisiler["kisi_tel"] = $row["kisi_tel"];

            array_push($response["kisiler"], $kisiler);
        }

        $response["success"] = 1;
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
}
```



tum\_kisiler\_arama.php

POST

```
{
  "kisiler": [
    {
      "kisi_id": "376",
      "kisi_ad": "Ahmet",
      "kisi_tel": "5348564412"
    },
    {
      "kisi_id": "379",
      "kisi_ad": "Zeynep",
      "kisi_tel": "05375483214"
    },
    { },
    { },
    { },
    { },
    { }
  ],
  "success": 1
}
```

```
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>
```

```

<?php
$response = array();

if (isset($_POST['kisi_id'])) {
    $kisi_id = $_POST['kisi_id'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }
    $sqlsorgu = "DELETE FROM kisiler WHERE kisiler.kisi_id = $kisi_id";

    if (mysqli_query($baglanti, $sqlsorgu)) {

        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {

        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>

```

delete\_kisiler.php

POST

Çalıştırmak istediğimiz  
PHP kod

```

{
  "success":0,
  "message":"Required field(s) is missing"
}

```

POST

insert\_kisiler.php

## Çalıştırmak istediğimiz PHP kod

```
<?php
$response = array();

if (isset($_POST['kisi_ad']) && isset($_POST['kisi_tel'])) {
    $kisi_ad = $_POST['kisi_ad'];
    $kisi_tel = $_POST['kisi_tel'];

    //DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.
    require_once __DIR__ . '/db_config.php';

    // Bağlantı oluşturuluyor.
    $baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);

    // Bağlantı kontrolü yapılır.
    if (!$baglanti) {
        die("Hatalı bağlantı : " . mysqli_connect_error());
    }

    $sqlsorgu = "INSERT INTO kisiler (kisi_ad,kisi_tel) VALUES ('$kisi_ad','$kisi_tel')";

    if (mysqli_query($baglanti, $sqlsorgu)) {
        $response["success"] = 1;
        $response["message"] = "successfully ";
        echo json_encode($response);
    } else {
        $response["success"] = 0;
        $response["message"] = "No product found";
        echo json_encode($response);
    }
    //bağlantı koparılır.
    mysqli_close($baglanti);
} else {
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";
    echo json_encode($response);
}
?>
```

```
{
  "success":0,
  "message":"Required field(s) is missing"
}
```

```
<?php
```

```
$response = array();
```

```
if (isset($_POST['kisi_id']) && isset($_POST['kisi_ad']) && isset($_POST['kisi_tel']) ) {  
    $kisi_id = $_POST['kisi_id'];  
    $kisi_ad = $_POST['kisi_ad'];  
    $kisi_tel = $_POST['kisi_tel'];
```

```
//DB_SERVER,DB_USER,DB_PASSWORD,DB_DATABASE değişkenleri alınır.  
require_once __DIR__ . '/db_config.php';
```

```
// Bağlantı oluşturuluyor.  
$baglanti = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);
```

```
// Bağlantı kontrolü yapılır.  
if (!$baglanti) {  
    die("Hatalı bağlantı : " . mysqli_connect_error());  
}
```

```
$sqlsorgu = "UPDATE kisiler SET kisiler.kisi_ad = '$kisi_ad',kisiler.kisi_tel = '$kisi_tel' WHERE  
    kisiler.kisi_id = $kisi_id ";
```

```
if (mysqli_query($baglanti, $sqlsorgu)) {  
    $response["success"] = 1;  
    $response["message"] = "successfully ";  
    echo json_encode($response);  
} else {  
    $response["success"] = 0;  
    $response["message"] = "No product found";  
    echo json_encode($response);  
}
```

```
//bağlantı koparılır.  
mysqli_close($baglanti);
```

```
} else {  
    $response["success"] = 0;  
    $response["message"] = "Required field(s) is missing";  
    echo json_encode($response);  
}  
?>
```

update\_kisiler.php

POST

## Çalıştırmak istediğimiz PHP kod

```
{  
    "success":0,  
    "message":"Required field(s) is missing"  
}
```

Teşekkürler...



kasim-adalan



kasimadalan@gmail.com



kasimadalan