

Uygulamamızda MVVM mimarisi kullanılmıştır.

MVVM Nedir?

MVVM bir projenin “sorumlulukların ayrıştırılması” esasına göre geliştirilmesi temeline dayanan bir tasarım kalıbı sunmaktadır. Spagetti kodların oluşmasını engellemek, temiz, okunabilir ve test edilebilir kodlar yazmak için son derece önemlidir.

Temel olarak görünüm ile iş katmanının birbirine karışması işleri git gide zorlaştırır. Bu nedenle bu iki katmanın birbirinden ayrılması gerekir. Yani siz görünüme ihtiyaç duymadan modelinizi kolayca değiştirebilmeli ya da tam tersi, modele ihtiyaç duymadan görünümünüzü güncelleyebilmelisiniz. Bu noktada MVVM bu ayrımı kolay bir şekilde yapıyor ve bu süreci yönetmemizi kolaylaştırıyor.

MVVM, Model, View ve ViewModel olmak üzere üç temel yapıdan oluşmaktadır.

Model: Veritabanından, web servislerinden ya da herhangi bir veri kaynağından gelen verilerimizi temsil etmek için kullanılan POJO ya da entity sınıflarından oluşmaktadır. Ayrıca veri tutarlılığını ve doğruluğunu kontrol eden iş kuralları da burada yer almaktadır.

Projede Firebase veritabanı kullanılmıştır.

View: Bu kısım verilerimizi son kullanıcılara aktardığımız görsel arayüzdür. Son kullanıcı ile uygulama arasında bir köprü görevi görür.

ViewModel: ViewModel ise görsel arayüz ile model arasında köprü görevi görmektedir, yani Model'i View'a bağlayan yapıdır. View ile Model arasında doğrudan bir etkileşim yoktur. View, ilgili işlemleri ViewModel üzerinden yapmaktadır. ViewModel'ın View'a direkt erişimi yoktur ve View ile ilgili hiçbir şey bilmez.

