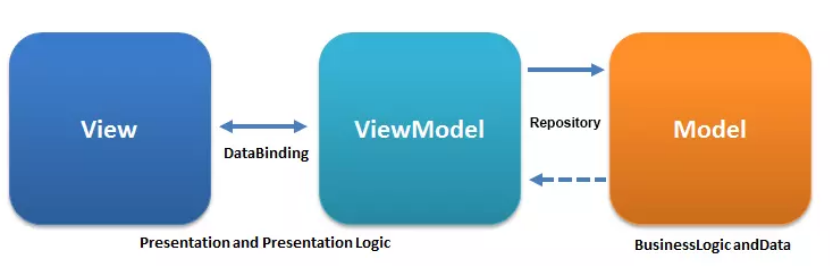
Trong MVVM (Model – View – ViewModel)

* Model: chứa các data classes, database classes, API và repository
* View: thành phần UI, đại diện cho trạng thái hiện tại của thông tin mà người dùng có thể nhìn thấy
* ViewModel: chứa data cần thiết trong View và chuyển dịch data được lưu trữ trong model mà sau đó có thể có trong View.
* ViewModel và View được liên kết thông qua Databinding và observable Livedata



Giải thích:

* Các phần Business Logic và Presentation Logic là tách biệt hoàn toàn với nhau
* View không nhận biết được những gì đang diễn ra bên trong ViewModel => vậy nên có thể sử dụng lại ViewModel
* Khi sử dụng Repository giữa ViewModel và Model, ViewModel chỉ biết cách tạo data request và repository sẽ quan tâm tới cách mà data được tìm gọi cho dù đó là từ API hay là Local database
* Sử dụng ViewModel và Livedata để nhận biết vòng đời của activity dẫn đến là ít crashes và memory leaks