Annak érdekében, hogy egy látványos különcélú mérőszoftver kerüljön kivitelezésre a grafikus felhasználói felület (GUI - Graphical User Interface) front-en és back-end oldalról egyaránt C# programnyelven íródott. Bár a választott WPF (Windows Presentation Foundation) applikáció nem különíthető csupán kétfelé, mint front és back-end, az érthetőség kedvéért könnyebb így jellemezni.

A szoftver fejlesztése Microsoft Visual Studio fejlesztői környezet alatt valósul/valósult meg és a már említett C# WPF alkalmazásként implementálva.

Mi az és miért pont WPF?

A **WPF** olyan grafikus felületek készítéséhez ajánlott könyvtárrendszer, ami a .NET keretrendszerre épül, egész a .NET 3.0 óta. Különlegessége az elődje, a Windows Form alapú applikációkhoz képest az **MVVM** (Model, View, ViewModel) alapú felépítés.

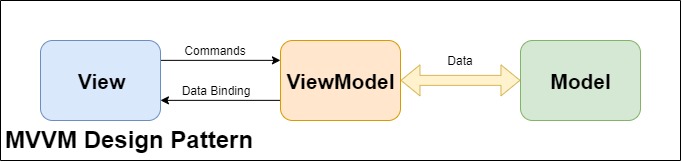
Az alkalmazást három logikai egység építi fel:

A **View**, azaz kinézet (front-end) tartalmazza a megjelenítési réteg elemeit, a felhasználó által látott, könnyen értelmezhető elemeket és paramétereiket (Gombok, legördülő menük-combobox, kapcsolók, csúszkák, szövegmezők, stb), valamint ezek pozícióit és a mögötte található háttérrel (background) kapcsolatos ugyancsak „látható” paramétereket.

A **Model** tartalmazza a deklarációkat, osztályokat, ahol az adatok tárolhatók és objektumorientált szempontból megfelelő, továbbá az úgynevezett üzleti logikát.

A **ViewModel** pedig tartalmazza az összeköttetéseket az előző két csoport között és nyomon követi a változásokat.

A választás azért esett a WPF alapú fejlesztésre, mert - bár nehezebb átlátni a működést – a létrehozott kód generikus lesz, főleg, ami a nézetmodellt illeti. Ebből kifolyólag, ha a nézet oldalon változtatni kell vagy esetleg a nézet helyét egy weboldal veszi át, akkor a már megírt modell és nézetmodell könnyűszerrel átültethető és csak az úgynevezett „bindingokat-kötéseket” kell módosítani.



31. ábra MVVM Modell   
forrás: Szerző által készített ábra