**Model-View-Controller Pattern**

Model-View-Controller(MVC) pattern is an Architectural Software Design pattern for implementing software with user interfaces. As the name already suggest, MVC has three components which can be explained as;

* **Model**: Component that contains data, logic and rules. In other words, its the component where most of internal interaction happens.
* **View**: Component that handles output operations, as an example, a button and response warning message on screen.
* **Controller**: Component that interprets inputs and tasks model or view accordingly.

In our design, we used MVC pattern because, our software requires user input for many cases. For the sake of brevity, we used simple names; View, Control and Model for subsystems. Our View Subsystem transfers inputs directly to the Control Subsystem without interpreting them which gives Control Subsystem efficient control over software while it keeps the simplicity of View and Model Subsystems. Separating Model, Controller and View also gives chance for modularity for any future subsystem change. MVC also makes data flow tracing easy for developers. For example, to start an application, data flow of our software generally proceeds like this;

***View*** *(Get user actions on GUI)🡪* ***Control*** *(Interpret inputs accordingly and command model)🡪* ***Model*** *(Start application)*

Furthermore, for getting information from internet (Wikipedia for example) and showing it to the user, data flow proceeds like this;

***View*** *(Get user actions on GUI)🡪* ***Control*** *(Interpret inputs accordingly and command model)🡪* ***Model*** *(Get information, send data to Control to notify)🡪* ***Control*** *(Send data to View to show user)🡪* ***View*** *(Change GUI and show data retrieved from internet)*