

The Model-View-Controller is a software design architecture. It divides the program into three structural parts. The model, the view, and the controller.

The Model: This is where most of the data will be stored. It is completely independent of the user interface, and commonly uses databases and permanent storage. It is usually easy to isolate, like a controller sending and receiving data from a remote server(the model).

The View: This is where the user will interact with the program. It is often a user interface where they can perform actions that the controller can interpret. It is also where the user is displayed information. This is independent from the Model.

The Controller: This is where most of the program logic occurs. The controller acts as a bridge between the View and Model, where any interaction between the two needs to come from the controller. It converts commands from the View into actions for the Model and converts data from the model into information to be displayed to the user.

I find this architecture to be very intuitive and effective at eliminating many problems that other architectures may have like security, integrity, modularity, and usability. This project was my first time building something with the model-view-controller architecture, but I found it relatively easy to understand and effective at what it was supposed to accomplish. However, this was a simple project, so this architecture might get very complex and difficult to navigate on more difficult projects.